Acta Numerica (2013), pp. 133–288
doi:10.1017/S0962492913000044

High-dimensional integration: The quasi-Monte Carlo way*†

Josef Dick
University of New South Wales,
Sydney, NSW 2052, Australia
E-mail: josef.dick@unsw.edu.au

Frances Y. Kuo
University of New South Wales,
Sydney, NSW 2052, Australia
E-mail: f.kuo@unsw.edu.au

Ian H. Sloan
University of New South Wales,
Sydney, NSW 2052, Australia
E-mail: i.sloan@unsw.edu.au
and
King Fahd University of Petroleum and Minerals,
Dhahran 34463, Saudi Arabia

This paper is a contemporary review of QMC ('quasi-Monte Carlo') methods, that is, equal-weight rules for the approximate evaluation of high-dimensional integrals over the unit cube $[0,1]^s$, where s may be large, or even infinite. After a general introduction, the paper surveys recent developments in lattice methods, digital nets, and related themes. Among those recent developments are methods of construction of both lattices and digital nets, to yield QMC rules that have a prescribed rate of convergence for sufficiently smooth functions, and ideally also guaranteed slow growth (or no growth) of the worst-case error as s increases. A crucial role is played by parameters called 'weights', since a careful use of the weight parameters is needed to ensure that the worst-case errors in an appropriately weighted function space are bounded, or grow only slowly, as the dimension s increases. Important tools for the analysis are weighted function spaces, reproducing kernel Hilbert spaces, and discrepancy, all of which are discussed with an appropriate level of detail.

* Colour online for monochrome figures available at journals.cambridge.org/anu.
† The URLs cited in this work were correct at the time of going to press, but the publisher and the authors make no undertaking that the citations remain live or are accurate or appropriate.

CONTENTS

1. Introduction

Numerical integration in more than one dimension has for a century been a topic that presents many challenges, as witnessed by the classical monograph by Stroud (1971). Among the challenges are those posed by the rich choice of multi-dimensional geometries, whether the cube, ball, sphere, simplex, or something more elaborate. Another challenge is the common occurrence of singularities of the integrand at special points or sub-manifolds of the region. But of all the challenges the one that stands above all others is the difficulty of doing anything effective when the number of integration variables (the 'dimension') passes beyond the number of fingers on two hands. Certainly the simple strategy of using a 'product' of one-dimensional rules becomes infeasible, since even a mere 10-fold product of say two-point Gauss rules requires 2^{10} points. Yet these days problems arise, and are tackled, in which the number of dimensions is in the hundreds, or even the thousands or tens of thousands.

The aim of this survey is to describe and analyse effective methods for numerical integration, with a special emphasis on a high number of dimensions. The price that has to be paid is that we consider only the simplest geometry (namely the unit cube in an arbitrary number s of dimensions), somewhat smooth functions, and only equal-weight integration rules.

Thus the task is to approximate the integral

$$I_s(f) = \int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \tag{1.1}$$

where s is greater than 1 and possibly large, for some integrable function f, by an n-point integration rule of the form

$$Q_{n,s}(f) = \frac{1}{n}\sum_{i=0}^{n-1} f(\boldsymbol{t}_i), \tag{1.2}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

thus a rule that uses the prescribed sample points $\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{n-1} \in [0,1]^s$. Of course, we shall have much to say about the choice of these sample points. A rule of this equal-weight form is called a *quasi-Monte Carlo* (or QMC) rule.

The name quasi-Monte Carlo comes from a certain analogy with the *Monte Carlo* (MC) method: in its simplest form the MC approximation to the integral (1.1) takes exactly the same form as (1.2), but with one crucial difference, that the sample points are chosen randomly (and independently), from a uniform distribution on the cube $[0,1]^s$. The MC method (Hammersley and Handscomb 1964, Lemieux 2009, Woźniakowski 2013) has some very attractive features, most notably that there exists a probabilistic (root-mean-square) error estimate of the form

$$\frac{\sigma(f)}{\sqrt{n}},$$

where $\sigma^2(f)$ is the variance of f,

$$\sigma^2(f) := I_s(f^2) - (I_s(f))^2,$$

which is finite whenever f is square-integrable. Moreover, $\sigma(f)$ can be estimated by the same MC rule used to estimate $I_s(f)$. But for all the virtues of the MC method, its $O(1/\sqrt{n})$ rate of convergence is often considered to be impossibly slow, especially when f is smooth. For example, one needs four times as many points to reduce the error by a factor of two, or 100 times as many points to reduce the error by an order of magnitude.

The QMC methods were born in the 1950s and 1960s from the (successful) desire to achieve faster convergence than the MC rate of $O(1/\sqrt{n})$. In the next sections we shall meet QMC methods that depend on constructing points sets with small 'discrepancy'. These methods were indeed successful in improving upon the MC convergence rate, achieving a convergence rate of order $O((\log n)^s/n)$ or better if sufficient smoothness of f is assumed. In particular, we will meet the so-called 'lattice rules', which for certain periodic functions (periodic with respect to each component of $\boldsymbol{x}$) could achieve convergence rates of $O((\log n)^s/n^2)$ or faster, and so-called 'higher-order digital nets', which could achieve convergence rates of $O((\log n)^{2s}/n^2)$ or faster even for non-periodic functions.

In the early days of QMC research, the emphasis was mainly on the rate of convergence as n increases, and less attention was paid to what happens as the dimensionality s increases; those early QMC researchers surely never envisaged QMC methods being applied to integrands in hundreds of dimensions. For lattice methods there was much interest in the 1960s in periodization strategies (for converting non-periodic integrands into periodic ones), which can then achieve fast convergence for appropriate QMC rules, but regrettably those periodization strategies often turn easy high-dimensional problems into ones that are impossibly hard. In any case, the

rate of convergence is only part of the story: the size of the implied constant is also important.

In the last two decades there has been great research interest in developing QMC methods and theory that can cope with arbitrarily high dimensions, if the circumstances are right. (Note that we do *not* say that all high-dimensional problems can be successfully tackled by QMC methods. Rather, the interest is in recognizing and analysing mathematically the particular features that make *some* high-dimensional problems manageable.)

Applications have also played an important role in the development of QMC methods suitable for high-dimensional problems. In financial mathematics, the numerical experiments of Paskov and Traub (1995), which used low-discrepancy QMC methods in 360 dimensions to value parcels of mortgage-backed obligations, were successful to a degree that caused universal surprise. This led to many theoretical developments, as researchers struggled to understand how such high-dimensionality could be handled successfully (for example Caflisch, Morokoff and Owen 1997, Wang and Fang 2003). Option pricing problems have spurred many developments (for example Acworth, Broadie and Glasserman 1998, L'Ecuyer 2004, Giles, Kuo, Sloan and Waterhouse 2008, Wang and Sloan 2011, Griebel, Kuo and Sloan 2013). A challenging class of problems in multivariate statistics led Kuo *et al.* (2008*a*) to a new understanding of the importance of proper handling of the transformation process from $\mathbb{R}^s$ into $[0,1]^s$ for problems over unbounded regions. The problem of evaluating high-dimensional expected values arising from partial differential equations with random coefficients, typified by the flow of a liquid (oil or water) through a porous material, with the permeability treated as a random field, is the newest driver of innovation (Graham *et al.* 2011, Kuo, Schwab and Sloan 2012). However, in this survey the focus will be on the theoretical innovations rather than on the applications themselves.

The structure of the survey is as follows. In Section 2 we introduce QMC methods, and develop basic definitions and concepts without much theory. In Section 3 we begin to develop key tools (such as reproducing kernel Hilbert spaces) needed for error analysis of QMC methods. In Section 4 we introduce the modern theory of weighted function spaces, in which certain parameters ('weights') are introduced to describe the circumstance that some variables (or groups of variables) are more important than others. In Sections 5 and 6 we introduce more sophisticated material on lattice rules and digital nets. Then in Section 7 we introduce a setting which is currently attracting great interest, namely that of numerical integration in an infinite number of dimensions. Finally in Section 8 we give concluding remarks. At the end of each section we provide notes with references to further related material.

2. Introducing quasi-Monte Carlo methods

2.1. Multivariate numerical integration

Recall from (1.1) that our goal is to approximate a multiple integral over the s-dimensional unit cube $[0,1]^s$,

$$I_s(f) = \int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} := \int_0^1 \cdots \int_0^1 f(x_1,\ldots,x_s)\,\mathrm{d}x_1 \cdots \mathrm{d}x_s,$$

where s is large, possibly in the hundreds or thousands. In practice most integrals over bounded or unbounded regions can be transformed into this desired form with a suitable change of variables. The transformation plays an important role as it determines the behaviour of the transformed integrand f. We shall defer the choice and strategy of transformation to be discussed later in Section 2.11.

For our analysis in this survey, we will generally assume that f is continuous and has a certain level of smoothness, as will be made precise in the next section.

In the classical theory of numerical integration (Davis and Rabinowitz 1984), an integral in one dimension can be approximated by a *quadrature rule*, as follows:

$$\int_0^1 f(x)\,\mathrm{d}x \approx \sum_{i=0}^{n-1} w_i f(t_i),$$

where $t_0,\ldots,t_{n-1} \in [0,1]$ are the *quadrature points*, and $w_0,\ldots,w_{n-1} \in \mathbb{R}$ are the *quadrature weights* satisfying $\sum_{i=0}^{n-1} w_i = 1$. Well-known examples include the (left) rectangle rule, which uses equally spaced points $t_i = i/n$ and equal weights $w_i = 1/n$ and has the error $f'(\xi)/(2n)$ for some $\xi \in (0,1)$; the trapezoidal rule (which has second-order convergence); Simpson's rule (with fourth-order convergence); and the Gauss rules (which integrate polynomials of degree $2n-1$ exactly).

For integration in s dimensions with $s \geq 2$, we have *cubature rules* (Stroud 1971),

$$\int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \approx \sum_{i=0}^{n-1} w_i f(\boldsymbol{t}_i),$$

where $\boldsymbol{t}_0,\ldots,\boldsymbol{t}_{n-1} \in [0,1]^s$ are the *cubature points*, $\boldsymbol{t}_i = (t_{i,1},\ldots,t_{i,s})$, and $w_0,\ldots,w_{n-1} \in \mathbb{R}$ are the *cubature weights* satisfying $\sum_{i=0}^{n-1} w_i = 1$. Explicit cubature rules are available in low dimensions (Cools 1997). An obvious way to approximate a high-dimensional integral is to use a *product rule*,

$$\int_0^1 \cdots \int_0^1 f(x_1,\ldots,x_s)\,\mathrm{d}x_1 \cdots \mathrm{d}x_s \approx \sum_{i_1=0}^{m-1} \cdots \sum_{i_s=0}^{m-1} u_{i_1} \cdots u_{i_s} f(v_{i_1},\ldots,v_{i_s}),$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

in effect applying a one-dimensional quadrature rule $\sum_{i=0}^{m-1} u_i f(v_i)$ to each of the s integrals. The cubature points $\boldsymbol{t}_i$ are given by the s-fold product of the quadrature points v_i, and the cubature weights w_i are the corresponding products of the quadrature weights u_i. The total number of points is $n = m^s$, which is enormous when s is large, while (since f could be a function of only one component, such as $f(x_1, \ldots, x_s) = x_1^2$) the error is only the sth root of the error of the one-dimensional quadrature rule: for example, the product rectangle rule has error of order $O(n^{-1/s})$.

As the title suggests, this survey will focus on *quasi-Monte Carlo methods* for high-dimensional integration. This section is devoted to introducing the basic principles behind these methods. We begin in the next subsection with a brief introduction to the related and widely used classical *Monte Carlo method*.

Another competing technique for high-dimensional integration is based on *sparse grid methods* (Bungartz and Griebel 2004), but we will not discuss these methods in this survey.

2.2. Monte Carlo method

The classical *Monte Carlo* (*MC*) *method* is an equal-weight cubature rule of the form

$$Q_{n,s}(f) = \frac{1}{n} \sum_{i=0}^{n-1} f(\boldsymbol{t}_i),$$

where $\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{n-1}$ are i.i.d. (independent and identically distributed) *uniform* random samples from $[0,1]^s$.

Theorem 2.1 (Monte Carlo root-mean-square error). For all *square-integrable* functions f, we have

$$\sqrt{\mathbb{E}[|I_s(f) - Q_{n,s}(f)|^2]} = \frac{\sigma(f)}{\sqrt{n}},$$

where the expectation is taken with respect to the uniform random samples $\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{n-1}$, and

$$\sigma^2(f) := I_s(f^2) - (I_s(f))^2$$

is the *variance* of f.

Although this result is well known, we give a proof since it will serve as a model for later arguments.

Proof. We have

$$\mathbb{E}[|Q_{n,s}(f) - I_s(f)|^2] = \mathbb{E}[(Q_{n,s}(f))^2] - 2\,\mathbb{E}[Q_{n,s}(f)]I_s(f) + (I_s(f))^2,$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

where

$$\begin{aligned}\mathbb{E}[Q_{n,s}(f)] &= \int_{[0,1]^s} \cdots \int_{[0,1]^s} \left(\frac{1}{n} \sum_{i=0}^{n-1} f(\boldsymbol{t}_i) \right) \mathrm{d}\boldsymbol{t}_0 \ldots \mathrm{d}\boldsymbol{t}_{n-1} \\ &= \frac{1}{n} \sum_{i=0}^{n-1} \int_{[0,1]^s} f(\boldsymbol{t}_i)\, \mathrm{d}\boldsymbol{t}_i \\ &= I_s(f)\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}[(Q_{n,s}(f))^2] &= \int_{[0,1]^s} \cdots \int_{[0,1]^s} \left(\frac{1}{n} \sum_{i=0}^{n-1} f(\boldsymbol{t}_i) \right)^2 \mathrm{d}\boldsymbol{t}_0 \ldots \mathrm{d}\boldsymbol{t}_{n-1} \\ &= \int_{[0,1]^s} \cdots \int_{[0,1]^s} \left(\frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{k=0}^{n-1} f(\boldsymbol{t}_i) f(\boldsymbol{t}_k) \right) \mathrm{d}\boldsymbol{t}_0 \ldots \mathrm{d}\boldsymbol{t}_{n-1} \\ &= \int_{[0,1]^s} \cdots \int_{[0,1]^s} \left(\frac{1}{n^2} \sum_{i=0}^{n-1} f^2(\boldsymbol{t}_i) + \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{\substack{k=0 \\ k \neq i}}^{n-1} f(\boldsymbol{t}_i) f(\boldsymbol{t}_k) \right) \mathrm{d}\boldsymbol{t}_0 \ldots \mathrm{d}\boldsymbol{t}_{n-1} \\ &= \frac{1}{n^2} \sum_{i=0}^{n-1} \int_{[0,1]^s} f^2(\boldsymbol{t}_i)\, \mathrm{d}\boldsymbol{t}_i + \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{\substack{k=0 \\ k \neq i}}^{n-1} \int_{[0,1]^s} f(\boldsymbol{t}_i)\, \mathrm{d}\boldsymbol{t}_i \int_{[0,1]^s} f(\boldsymbol{t}_k)\, \mathrm{d}\boldsymbol{t}_k \\ &= \frac{1}{n} I_s(f^2) + \frac{n-1}{n} (I_s(f))^2.\end{aligned}$$

Hence

$$\mathbb{E}[|Q_{n,s}(f) - I_s(f)|^2] = \frac{I_s(f^2) - (I_s(f))^2}{n},$$

as claimed. □

Treating $Q_{n,s}(f)$ as a random variable, we see from the above proof that its *mean* is

$$\mathbb{E}[Q_{n,s}(f)] = I_s(f)$$

(that is to say, the MC method is unbiased), and its *variance* is

$$\mathrm{Var}[Q_{n,s}(f)] = \mathbb{E}[|Q_{n,s}(f) - I_s(f)|^2] = \frac{\sigma^2(f)}{n}.$$

By the central limit theorem, if $0 < \sigma(f) < \infty$ then

$$\lim_{n \to \infty} \mathbb{P}\left(|I_s(f) - Q_{n,s}(f)| \leq c \frac{\sigma(f)}{\sqrt{n}} \right) = \frac{1}{\sqrt{2\pi}} \int_{-c}^{c} \mathrm{e}^{-x^2/2}\, \mathrm{d}x.$$

In other words, we have a 'probabilistic' error bound with a convergence rate

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

of $O(n^{-1/2})$. The convergence rate is *independent of the dimension*. Comparing this with, for example, the product rectangle rule whose convergence rate is $O(n^{-1/s})$, we see that the MC method has the faster convergence when $s \geq 3$ even for smooth functions.

Another advantage of the MC method is that it is easy to obtain a practical error estimation. Indeed, an unbiased estimator for $\mathrm{Var}[Q_{n,s}(f)]$ is given by

$$\frac{1}{n(n-1)} \sum_{i=0}^{n-1} (f(\boldsymbol{t}_i) - Q_{n,s}(f))^2 = \frac{1}{n(n-1)} \left(\sum_{i=0}^{n-1} f^2(\boldsymbol{t}_i) - n[Q_{n,s}(f)]^2 \right). \tag{2.1}$$

The square root of this quantity provides an estimate for the root-mean-square MC error.

Although widely applicable, MC methods suffer from the slow $O(n^{-1/2})$ convergence rate. Variance reduction techniques (*e.g.*, importance sampling, stratified sampling, correlated sampling) can be used to improve the efficiency of MC, but in practice MC methods often remain distressingly slow. Bakhvalov (1959) proved that the $O(n^{-1/2})$ rate of convergence cannot be improved for general square-integrable or continuous functions f. For functions with more smoothness, this slow convergence rate is the main motivation for the switch to quasi-Monte Carlo methods.

2.3. Quasi-Monte Carlo methods

Quasi-Monte Carlo (QMC) methods are equal-weight cubature rules of the form

$$Q_{n,s}(f) = \frac{1}{n} \sum_{i=0}^{n-1} f(\boldsymbol{t}_i),$$

just like MC methods, but now the points $\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{n-1} \in [0,1]^s$ are *chosen deterministically to be better than random*, in the sense that the deterministic nature of QMC leads to guaranteed error bounds, and that the convergence rate may be faster than the MC rate of $O(n^{-1/2})$ for sufficiently smooth functions.

There are two types of QMC methods.

- The 'open' type: this uses the first n points of an *infinite sequence*. Thus to increase n one only needs to evaluate the integrand at the additional cubature points.
- The 'closed' type: this uses a *finite point set* which depends on n. Thus a new value of n means a completely new set of cubature points.

(Here and in the following, a point set is understood to be a multiset, that is, we count points according to their multiplicity.)

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

The following definition is needed before we provide examples of QMC methods.

Definition 2.2 (radical inverse function). For integers $i \geq 0$ and $b \geq 2$, we define the *radical inverse function* $\phi_b(i)$ as follows:

$$\text{if} \quad i = \sum_{a=1}^{\infty} i_a b^{a-1}, \quad \text{where} \quad i_a \in \{0, 1, \ldots, b-1\}, \quad \text{then} \quad \phi_b(i) := \sum_{a=1}^{\infty} \frac{i_a}{b^a}.$$

In other words, if $i = (\cdots i_2 i_1)_b$ denotes the base b representation of i, then $\phi_b(i) := (0.i_1 i_2 \cdots)_b$.

Example 2.3 (van der Corput sequence). The *van der Corput sequence in base b* is the one-dimensional sequence

$$\phi_b(0), \phi_b(1), \phi_b(2), \ldots.$$

For example, take $b = 2$. First we write down the natural numbers $0, 1, 2, \ldots$ in base 2

$$0, 1_2, 10_2, 11_2, 100_2, 101_2, 110_2, \ldots.$$

Then we apply the radical inverse function ϕ_2 to each number, to obtain the sequence

$$0, 0.1_2, 0.01_2, 0.11_2, 0.001_2, 0.101_2, 0.011_2, \ldots,$$

which in decimal form is the sequence

$$0, 0.5, 0.25, 0.75, 0.125, 0.625, 0.375, \ldots.$$

Example 2.4 (Halton sequence). Let $p_1, p_2, \ldots, p_s$ be the first s prime numbers. The *Halton sequence* $\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots$ *in s dimensions* is given by

$$\boldsymbol{t}_i = \big(\phi_{p_1}(i), \phi_{p_2}(i), \ldots, \phi_{p_s}(i)\big), \quad i = 0, 1, \ldots,$$

that is, the jth components of points in the Halton sequence form the van der Corput sequence in base p_j, where p_j is the jth prime. The Halton sequence leads to an 'open' QMC method. We have explicitly

$$\begin{aligned}
\boldsymbol{t}_0 &= (0, 0, 0, \ldots, 0),\\
\boldsymbol{t}_1 &= (0.1_2, 0.1_3, 0.1_5, \ldots, 0.1_{p_s}),\\
\boldsymbol{t}_2 &= (0.01_2, 0.2_3, 0.2_5, \ldots, 0.2_{p_s}),\\
\boldsymbol{t}_3 &= (0.11_2, 0.01_3, 0.3_5, \ldots, 0.3_{p_s})\\
&\vdots
\end{aligned}$$

The Halton sequence satisfies the error bound

$$|I_s(f) - Q_{n,s}(f)| \leq C_s \frac{(\log n)^s}{n} V(f), \quad \text{for all } n \geq 2,$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

where C_s depends only on s, and $V(f)$ is the *variation of f in the sense of Hardy and Krause*; see for instance Kuipers and Niederreiter (1974, p. 147, Definition 5.2). For now we simply observe that, although the convergence rate appears to beat the MC rate of $O(n^{-1/2})$, the MC rate is independent of s, whereas for fixed s, the function $(\log n)^s/n$ increases with increasing n for all $n < \exp(s)$.

Example 2.5 (Hammersley point set). Let $p_1, p_2, \ldots, p_{s-1}$ be the first $s-1$ prime numbers. The *Hammersley point set $\{\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{n-1}\}$ with n points in s dimensions* is given by

$$\boldsymbol{t}_i = \left(\frac{i}{n}, \phi_{p_1}(i), \phi_{p_2}(i), \ldots, \phi_{p_{s-1}}(i)\right), \quad i = 0, 1, \ldots, n-1.$$

The Hammersley point set leads to a 'closed' QMC method. We have explicitly

$$\begin{aligned}
\boldsymbol{t}_0 &= (0, 0, \ldots, 0),\\
\boldsymbol{t}_1 &= \left(\tfrac{1}{n}, 0.1_2, 0.1_3, \ldots, 0.1_{p_{s-1}}\right),\\
\boldsymbol{t}_2 &= \left(\tfrac{2}{n}, 0.01_2, 0.2_3, \ldots, 0.2_{p_{s-1}}\right)\\
&\vdots\\
\boldsymbol{t}_{n-1} &= \left(\tfrac{n-1}{n}, \ldots\right).
\end{aligned}$$

The QMC method based on the Hammersley point set satisfies the error bound

$$|I_s(f) - Q_{n,s}(f)| \le C_s' \frac{(\log n)^{s-1}}{n} V(f), \quad \text{for all } n \ge 2,$$

where C_s' depends only on s. Note that there is one less power of $\log n$ compared to the error bound for the Halton sequence. Typically the error bounds for QMC methods based on 'closed' point sets are better than those based on 'open' sequences.

Example 2.6 (Kronecker sequence). Let $1, \alpha_1, \alpha_2, \ldots, \alpha_s \in \mathbb{R}$ be linearly independent over $\mathbb{Q}$. The *Kronecker sequence $\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots$ in s dimensions* is given by

$$\boldsymbol{t}_i = \big(\{i\alpha_1\}, \{i\alpha_2\}, \ldots, \{i\alpha_s\}\big), \quad i = 0, 1, \ldots,$$

where the braces indicate that we take the fractional part of a real number, that is,

$$\{x\} := x - \lfloor x \rfloor. \tag{2.2}$$

For example, a good choice of parameters is $\alpha_j = \sqrt{p_j}$ where p_j is the jth prime number. The error bound for the Kronecker sequence takes the same form as that of the Halton sequence, but with a different multiplying constant, again depending on s.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Two main families of QMC methods are currently under investigation:

- *digital nets* ('closed') and *digital sequences* ('open'),
- *lattice rules* ('closed' and 'open').

We will introduce these two families in the next few subsections.

2.4. Lattice rules

A *lattice* in $\mathbb{R}^s$ is a discrete subset of $\mathbb{R}^s$ which is closed under addition and subtraction. An *integration lattice* in $\mathbb{R}^s$ is a lattice which contains $\mathbb{Z}^s$ as a subset. A *lattice rule* is an equal-weight cubature rule whose cubature points are those points of an integration lattice that lie in the half-open unit cube $[0,1)^s$.

Every lattice point set includes the origin $\mathbf{0}$. The projection of the lattice points onto each axis gives equally spaced points. In a sense the integral in each dimension is approximated by a rectangle rule (or a trapezoidal rule if the integrand is periodic). Every lattice rule can be written as a *multiple sum* involving one or more generating vectors. The minimal number of generating vectors required to generate a lattice rule is known as the *rank* of the rule. Besides rank-one lattice rules involving just one generating vector, there exist lattice rules having rank up to s. Lattice rules were introduced by Korobov (1959). They were originally designed for periodic integrands.

Definition 2.7 (rank-one lattice rule). An n-point *rank-one lattice rule* in s dimensions, also known as the *method of good lattice points*, is a QMC method with cubature points

$$\boldsymbol{t}_i = \left\{ \frac{i\boldsymbol{z}}{n} \right\}, \quad i = 0, 1, \ldots, n-1, \tag{2.3}$$

where $\boldsymbol{z} \in \mathbb{Z}^s$, known as the *generating vector*, is an s-dimensional integer vector having no factor in common with n, and the braces around the vector indicate, as in (2.2), that we take the fractional part of each component in the vector.

Example 2.8 (Fibonacci lattice). Let $\boldsymbol{z} = (1, F_k)$ and $n = F_{k+1}$, where F_k and F_{k+1} are consecutive Fibonacci numbers. Then the resulting two-dimensional lattice point set is called a *Fibonacci lattice*; see Figure 2.1 for two examples. Fibonacci lattices in two dimensions have a certain optimality property, but there is no obvious generalization to higher dimensions that retains the optimality property.

Since we are only interested in the fractional part of $i\boldsymbol{z}/n$, the components of $\boldsymbol{z}$ can be restricted to the set

$$\mathbb{Z}_n := \{0, 1, 2, \ldots, n-1\}.$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

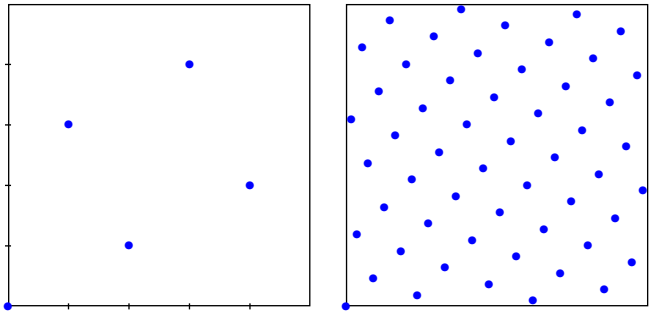


Figure 2.1. The Fibonacci lattices with 5 points and 55 points. The corresponding generating vectors are $(1, 3)$ and $(1, 34)$, respectively.

Moreover, since we would prefer that every one-dimensional projection of the lattice rule has n distinct values, the components of $\boldsymbol{z}$ can be further restricted to the set

$$\mathbb{U}_n := \{z \in \mathbb{Z} : 1 \le z \le n-1 \text{ and } \gcd(z, n) = 1\}.$$

The number of elements in the set $\mathbb{U}_n$ is $\varphi(n) := |\mathbb{U}_n|$, the *Euler totient function*. If $n = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k}$ is the prime factorization of n, then

$$\varphi(n) = (p_1^{a_1} - p_1^{a_1-1})(p_2^{a_2} - p_2^{a_2-1}) \cdots (p_k^{a_k} - p_k^{a_k-1}).$$

Asymptotically, $\varphi(n)$ grows at a rate close to n: $1/\varphi(n) = O((\log\log n)/n)$.

For simplicity, we often assume that n is prime and thus $\varphi(n) = n - 1$. This implies that:

- there are $n - 1$ choices for each component of $\boldsymbol{z}$,
- there are $(n-1)^s$ choices for the generating vector $\boldsymbol{z}$.

For large n and s, an exhaustive search to find a generating vector that minimizes some desired error criterion is practically impossible. We now present two approaches for constructing lattice generating vectors in high dimensions.

Example 2.9 (Korobov construction). Given an integer a satisfying $1 \le a \le n-1$ and $\gcd(a, n) = 1$, we define

$$\boldsymbol{z} = \boldsymbol{z}(a) := (1, a, a^2, \ldots, a^{s-1}) \bmod n.$$

There are (at most) $n - 1$ choices for the Korobov parameter a, leading to (at most) $n - 1$ choices for the generating vector $\boldsymbol{z}$. Thus it is feasible in practice to search through the (at most) $n-1$ choices and take the one that minimizes the desired error criterion.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Example 2.10 (component-by-component (CBC) construction). Given n, we construct a generating vector $\boldsymbol{z} = (z_1, z_2, \ldots)$ as follows.

(1) Set $z_1 = 1$.

(2) With z_1 held fixed, choose z_2 from $\mathbb{U}_n$ to minimize a desired error criterion in 2 dimensions.

(3) With z_1, z_2 held fixed, choose z_3 from $\mathbb{U}_n$ to minimize a desired error criterion in 3 dimensions.

(4) With z_1, z_2, z_3 held fixed, choose z_4 from $\mathbb{U}_n$ to minimize a desired error criterion in 4 dimensions.

(5) ...

The generating vector obtained by the CBC construction is *extensible in* s. In Section 5 we will consider two important aspects of the CBC construction: error analysis and computational cost. We will also discuss how to obtain *extensible lattice sequences* that are extensible in both s and n.

2.5. (t, m, s)-nets and (t, s)-sequences

Nets and sequences provide another method for obtaining well-distributed point sets in the unit cube $[0,1)^s$ that are useful for QMC integration. The concept of (t, m, s)-net is based on subdividing the unit cube into intervals and placing points in the cube such that each interval of a certain size and shape contains the 'correct' number of points.

Definition 2.11 ((t, m, s)-net). Let $t \geq 0$, $m \geq 1$, $s \geq 1$, and $b \geq 2$ be integers with $t \leq m$. A *(t, m, s)-net in base b* is a point set P consisting of b^m points in $[0,1)^s$ such that every *elementary interval* of the form

$$\prod_{j=1}^{s} \left[\frac{a_j}{b^{d_j}}, \frac{a_j + 1}{b^{d_j}} \right) \tag{2.4}$$

with each $d_j \geq 0$, $0 \leq a_j < b^{d_j}$, and $d_1 + d_2 + \cdots + d_s = m - t$, contains exactly b^t points of P.

An elementary interval (2.4) has volume $b^{-(d_1+d_2+\cdots+d_s)} = b^{t-m}$, which is precisely the proportion of the points from P that lie in this elementary interval. One would expect such a property to hold if the point set is uniformly distributed. Figure 2.2 provides an illustration of a two-dimensional net with 16 points.

Note that any point set consisting of b^m points in $[0,1)^s$ is trivially an (m, m, s)-net in base b, indicating that the concept of (t, m, s)-net is only useful when $t < m$. For fixed b and m, the (t, m, s)-net condition gets stronger as t gets smaller. Hence the aim is to find (t, m, s)-nets in base

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

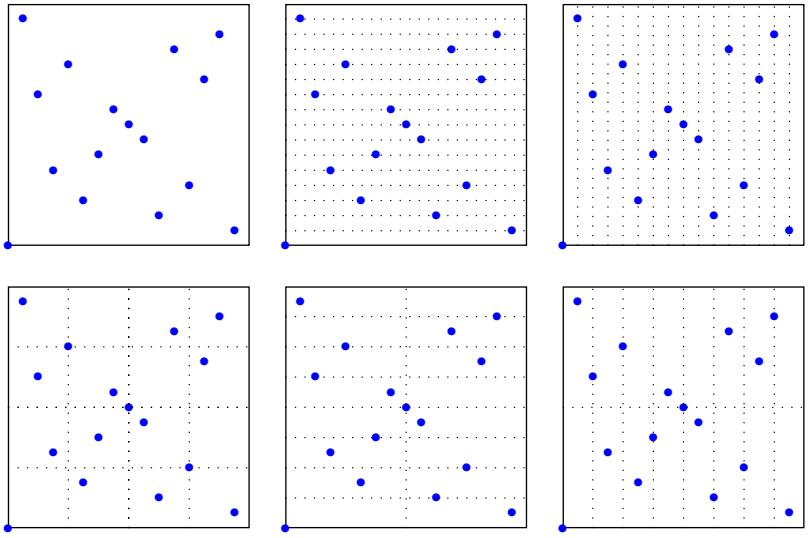


Figure 2.2. Illustration of a $(0, 4, 2)$-net in base 2: every elementary interval of volume $1/16$ contains exactly one of the 16 points. A point that lies on the dividing line counts toward the interval above or to the right.

b where t is small. A (t, m, s)-net is said to be a *strict* (t, m, s)-net if it is not a $(t-1, m, s)$-net. The 't-value' is often referred to as the *quality parameter*. It is also not useful to choose the base b too large, noting that in the extreme case in which $b = n$ we have $m = 1$, so that even when $t = 0$ the only elementary intervals are the 'boxes' of side length 1 and thickness $1/n$, in which case even the set of points $k(1/n, 1/n, \ldots, 1/n)$ for $k = 0, \ldots, n-1$ lying on the main diagonal is a $(0, m, s)$-net. The relevant quantity for fixed n is the *strength* $k = m - t$ of the (t, m, s)-net. The aim is then to find nets with large strength k.

There is an analogous concept for infinite sequences.

Definition 2.12 ((t, s)-sequences). Let $t \geq 0$ and $s \geq 1$ be integers. A (t, s)*-sequence in base* b is a sequence of points $S = (\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots)$ in $[0, 1)^s$ such that for any integers $m > t$ and $\ell \geq 0$, every block of b^m points

$$\boldsymbol{t}_{\ell b^m}, \ldots, \boldsymbol{t}_{(\ell+1)b^m-1}$$

in the sequence S forms a (t, m, s)-net in base b.

The van der Corput sequence in base b is a $(0, 1)$-sequence in base b. Higher-dimensional examples include the *Sobol′ sequence* (Sobol′ 1967), which is a (t, s)-sequence in base 2, where t is a non-decreasing function of s; the *Faure sequence* (Faure 1982); the *Niederreiter sequence* (Niederreiter 1987); and the *Niederreiter–Xing sequence* (Niederreiter and Xing 1995).

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

More details about these sequences will be given in Section 2.7. These sequences are also commonly referred to as 'low-discrepancy sequences' due to their high uniformity. We will explain the term 'discrepancy' in Section 3.

2.6. Digital construction scheme

We now introduce a method for the construction of (t, m, s)-nets and (t, s)-sequences. This method is based on linear algebra over finite fields.

Let b be prime and let $\mathbb{Z}_b := \{0, 1, \ldots, b-1\}$ denote the equivalence classes of integers modulo b. Using addition $+$ and multiplication $*$ of elements in $\mathbb{Z}_b$ modulo the prime b, we obtain a finite field of order b. In the following we identify the element k in the finite field $\mathbb{Z}_b$ with the integer $0 \le k < b$.

Let $C_1, \ldots, C_s$ be m-by-m matrices with entries in $\mathbb{Z}_b$. Then $t_{i,j}$, the jth component of the ith point in $P = \{\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{b^m-1}\}$, can be constructed as follows.

(1) Write i in its base b representation:

$$i = (i_m \cdots i_2 i_1)_b = i_1 + i_2 b + \cdots + i_m b^{m-1}.$$

(2) Compute

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = C_j \begin{pmatrix} i_1 \\ i_2 \\ \vdots \\ i_m \end{pmatrix},$$

where all additions and multiplications are done in the finite field $\mathbb{Z}_b$, *i.e.*, modulo b.

(3) Set

$$t_{i,j} = (0.y_1 y_2 \cdots y_m)_b = \frac{y_1}{b} + \frac{y_2}{b^2} + \cdots + \frac{y_m}{b^m}.$$

The resulting point set $P = \{\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{b^m-1}\}$ is called a *digital net* over $\mathbb{Z}_b$ and the matrices $C_1, \ldots, C_s$ are called the *generating matrices* of the digital net. The following definition connects (t, m, s)-nets in base b with digital nets.

Definition 2.13 (digital (t, m, s)-net). Let b be prime, $t \ge 0$, $m \ge 1$, and $s \ge 1$ be integers. If a digital net P constructed over $\mathbb{Z}_b$ is a (t, m, s)-net in base b, then P is called a *digital (t, m, s)-net over $\mathbb{Z}_b$*.

The following lemma provides the connection between the generating matrices of a digital net and the (t, m, s)-net property.

Lemma 2.14 (t-value for a digital net). A digital net over $\mathbb{Z}_b$ with generating matrices $C_1, \ldots, C_s$ is a digital (t, m, s)-net over $\mathbb{Z}_b$ if and only if,

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

for all $d_1, \dots, d_s \geq 0$ with $d_1 + \cdots + d_s = m - t$, the set of vectors

$$c_{1,1}, c_{1,2}, \dots, c_{1,d_1}, c_{2,1}, c_{2,2}, \dots, c_{2,d_2}, \dots, c_{s,1}, c_{s,2}, \dots, c_{s,d_s}$$

is linearly independent over $\mathbb{Z}_b$, where $c_{j,\ell}$ denotes the ℓth row vector of the matrix C_j.

Proof. Let $d_1, \dots, d_s \geq 0$ such that $d_1 + \cdots + d_s = m - t$, let

$$J = \prod_{j=1}^{s} \left[\frac{a_j}{b^{d_j}}, \frac{a_j + 1}{b^{d_j}} \right)$$

be an elementary interval, and let

$$\frac{a_j}{b^{d_j}} = \frac{u_{j,1}}{b} + \frac{u_{j,2}}{b^2} + \cdots + \frac{u_{j,d_j}}{b^{d_j}}.$$

Thus, for given $d_1, \dots, d_s$, we can uniquely associate the vector

$$\vec{u} = (u_{1,1}, \dots, u_{1,d_1}, \dots, u_{s,1}, \dots, u_{s,d_s})^\top$$

to the elementary interval J. Let $\{\boldsymbol{t}_0, \boldsymbol{t}_1, \dots, \boldsymbol{t}_{b^m-1}\}$ be a digital net with generating matrices $C_1, \dots, C_s \in \mathbb{Z}_b^{m \times m}$. Let $\boldsymbol{t}_i = (t_{i,1}, t_{i,2}, \dots, t_{i,s})$ with $t_{i,j} = \tau_{i,j,1} b^{-1} + \tau_{i,j,2} b^{-2} + \cdots + \tau_{i,j,m} b^{-m}$. Then $\boldsymbol{t}_i \in J$ if and only if $u_{j,\ell} = \tau_{i,j,\ell}$ for $1 \leq \ell \leq d_j$ and $1 \leq j \leq s$. Let $C_j = (c_{j,1}^\top, \dots, c_{j,m}^\top)^\top$, that is, $c_{j,\ell}$ is the ℓth row vector of C_j. Let $A = (c_{1,1}^\top, \dots, c_{1,d_1}^\top, \dots, c_{s,1}^\top, \dots, c_{s,d_s}^\top)^\top$. Then the number of points of the digital net which lie in J is given by the number of solutions $(i_1, \dots, i_m)$ of the equation over $\mathbb{Z}_b$:

$$A \begin{pmatrix} i_1 \\ i_2 \\ \vdots \\ i_m \end{pmatrix} = \vec{u}. \tag{2.5}$$

For each vector $\vec{u}$ the number of solutions of (2.5) is b^{m-t} if and only if the rows of A are linearly independent. Thus the result follows. □

Constructions of good generating matrices for digital nets yielding digital (t, m, s)-nets with small t-value will be discussed in Section 2.7. In the following we give a two-dimensional example of digital $(0, m, 2)$-nets over $\mathbb{Z}_b$ for arbitrary prime b.

Example 2.15. Let b be a prime and let $m \geq 1$ be an integer. Let

$$C_1 = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix} \in \mathbb{Z}_b^{m \times m}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

and

$$C_2 = \begin{pmatrix} 0 & \cdots & \cdots & 0 & 1 \\ 0 & \cdots & 0 & 1 & 0 \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & \cdots & 0 \end{pmatrix} \in \mathbb{Z}_b^{m \times m}.$$

Then Lemma 2.14 implies that C_1 and C_2 are generating matrices of a digital $(0, m, 2)$-net over $\mathbb{Z}_b$.

The concept of digital nets can be extended to infinite sequences $S = (\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots)$ in $[0,1)^s$. To do so, we need infinite matrices $C_1, \ldots, C_s \in \mathbb{Z}^{\mathbb{N} \times \mathbb{N}}$, with $C_j = (c_{j,k,\ell})_{k,\ell \in \mathbb{N}}$ and $c_{j,k,\ell} \in \mathbb{Z}_b$, such that for every ℓ there is a k_ℓ such that $c_{j,k,\ell} = 0$ for all $k > k_\ell$. Then $t_{i,j}$, the jth component of the ith point in the sequence S is constructed in the following way.

(1) Write i in its base b representation:

$$i = (\cdots i_2 i_1)_b = i_1 + i_2 b + \cdots .$$

(Note that only finitely many digits i_k are different from 0.)

(2) Compute

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \end{pmatrix} = C_j \begin{pmatrix} i_1 \\ i_2 \\ \vdots \end{pmatrix},$$

where all additions and multiplications are done in the finite field $\mathbb{Z}_b$.

(3) Set

$$t_{i,j} = (0.y_1 y_2 \cdots)_b = \frac{y_1}{b} + \frac{y_2}{b^2} + \cdots .$$

The resulting sequence $S = (\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots)$ is called a *digital sequence* over $\mathbb{Z}_b$ and the matrices $C_1, \ldots, C_s$ are called the *generating matrices* of the digital sequence. The following definition connects (t, s)-sequences in base b with digital sequences.

Definition 2.16 (digital (t, s)-sequences). Let b be prime, and let $t \geq 0$ and $s \geq 1$ be integers. If a digital sequence S constructed over $\mathbb{Z}_b$ is a (t, s)-sequence in base b, then S is called a *digital (t, s)-sequence over $\mathbb{Z}_b$*.

Constructions of good generating matrices for digital sequences yielding digital (t, s)-sequences with small t-value will be discussed in Section 2.7. In the following we give a one-dimensional example of a $(0, 1)$-sequence over $\mathbb{Z}_b$ for arbitrary prime b.

Example 2.17. Let b be a prime. Let $C \in \mathbb{Z}_b^{\mathbb{N} \times \mathbb{N}}$ be the matrix with ones along the main diagonal and zeros everywhere else (*i.e.*, the identity

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

matrix). Then C generates a digital $(0,1)$-sequence over $\mathbb{Z}_b$. In fact, C generates the van der Corput sequence in base b.

2.7. Constructions of digital nets and sequences

In this subsection we discuss explicit constructions of digital nets and sequences with small t value.

In the previous subsection we introduced (digital) (t,m,s)-nets and (digital) (t,s)-sequences and we gave explicit constructions for small dimensions. The first constructions in arbitrary dimensions are due to Sobol′ (1967) and Faure (1982), before the general concept of (digital) (t,s)-sequence was introduced in Niederreiter (1987). In the following we discuss these explicit constructions in detail.

We use the following notation. Let $\mathbb{Z}_b[x]$ denote the set of polynomials over the finite field $\mathbb{Z}_b$. The polynomial $p(x) = a_0 + a_1 x + \cdots + a_m x^m$ with $a_m \neq 0$ is said to have degree $\deg(p) = m$. For $p = 0$ we set $\deg(p) = -\infty$. A polynomial $p \in \mathbb{Z}_b[x]$ is *irreducible* if it cannot be written as a product of two non-constant polynomials, that is, there are no $q, q' \in \mathbb{Z}_b[x]$ such that $p(x) = q(x)q'(x)$ with $\deg(q), \deg(q') > 0$. A polynomial $p(x) = a_0 + a_1 x + \cdots + a_m x^m \in \mathbb{Z}_b[x]$ of degree $m \geq 1$ is *primitive* if and only if $a_m = 1$, $a_0 \neq 0$ and the smallest integer k such that there is a polynomial $q \in \mathbb{Z}_b[x]$ with $p(x)q(x) = x^k - 1 \in \mathbb{Z}_b[x]$ is $k = b^m - 1$. It can be shown that every primitive polynomial is irreducible. Further, if p is primitive, then the polynomial x is a primitive element in $\mathbb{Z}_b[x]/p$, that is, $\{x^k \bmod p : k = 0, 1, \ldots, b^m - 2\} = (\mathbb{Z}_b[x]/p) \setminus \{0\}$.

Example 2.18 (Sobol′ sequence). Sobol′ (1967) was the first to introduce a construction of (t,s)-sequences over $\mathbb{Z}_2$, which are now referred to as *Sobol′ sequences*. These are digital sequences over the finite field $\mathbb{Z}_2$.

(1) Let $p_1, \ldots, p_s \in \mathbb{Z}_2[x]$ be distinct primitive polynomials ordered according to their degree, and let

$$p_j(x) = x^{e_j} + a_{1,j} x^{e_j - 1} + a_{2,j} x^{e_j - 2} + \cdots + a_{e_j - 1, j} x + 1, \quad \text{for } 1 \leq j \leq s,$$

where $a_{j,k} \in \mathbb{Z}_b$. Note that e_j denotes the degree of the polynomial p_j.

(2) Choose odd natural numbers $1 \leq m_{1,j}, \ldots, m_{e_j,j}$ such that $m_{k,j} < 2^k$ for $1 \leq k \leq e_j$, and for all $k > e_j$ define $m_{k,j}$ recursively by

$$m_{k,j} = 2a_{1,j} m_{k-1,j} \oplus \cdots \oplus 2^{e_j - 1} a_{e_j - 1, j} m_{k - e_j + 1, j} \oplus 2^{e_j} m_{k - e_j, j} \oplus m_{k - e_j, j},$$

where $\oplus$ is the bit-by-bit exclusive-or operator.

(3) The so-called *direction numbers* are defined by

$$v_{k,j} := \frac{m_{k,j}}{2^k}, \quad \text{for } k \geq 1.$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

(4) Then for $i \in \mathbb{N}_0$ with dyadic expansion $i = i_0 + 2i_1 + \cdots + 2^{r-1}i_{r-1}$ we define

$$t_{i,j} = i_0 v_{1,j} \oplus i_1 v_{2,j} \oplus \cdots \oplus i_{r-1} v_{r,j}.$$

The Sobol′ sequence is then the sequence of points $\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots$, where $\boldsymbol{t}_i = (t_{i,1}, \ldots, t_{i,s})$.

Sobol′ sequences are digital (t, s)-sequences with

$$t = \sum_{j=1}^{s} (e_j - 1).$$

This result holds for all choices of direction numbers, and therefore the direction numbers do not influence the overall quality of Sobol′ sequences. Note, however, that the t-value for a Sobol′ net of b^m points for different m can be smaller than the t-value of the full sequence, and hence the choice of direction numbers can affect the quality of a Sobol′ net.

Example 2.19 (Faure sequence). Faure (1982) introduced a construction of $(0, s)$-sequences over prime fields $\mathbb{Z}_b$ with $s \le b$, which are now called *Faure sequences*. The generating matrices $C_1, \ldots, C_s$ are given by

$$C_j = (P^\top)^{j-1} \pmod b \quad \text{for } 1 \le j \le s,$$

where P is the *Pascal matrix* given by

$$P = \begin{pmatrix} \binom{0}{0} & \binom{0}{1} & \binom{0}{2} & \cdots \\ \binom{1}{0} & \binom{1}{1} & \binom{1}{2} & \cdots \\ \binom{2}{0} & \binom{2}{1} & \binom{2}{2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

where we set $\binom{k}{\ell} = 0$ for $\ell > k$. A Faure sequence is a digital $(0, s)$-sequence.

Example 2.20 (Niederreiter sequence). Niederreiter (1987) introduced the general concept of digital (t, s)-sequences, which includes both the Sobol′ and Faure constructions as special cases. We describe a special case of this sequence in the following.

Let $\mathbb{Z}_b = \{0, 1, \ldots, b-1\}$ denote the finite field of prime order b with addition and multiplication modulo b. Let $\mathbb{Z}_b[x]$ denote the set of polynomials with coefficients in $\mathbb{Z}_b$. Addition and multiplication of polynomials in $\mathbb{Z}_b$ is defined by using arithmetic in $\mathbb{Z}_b$ (*i.e.*, modulo b). This can also be extended to division of polynomials in $\mathbb{Z}_b$, which gives rise to the set of formal series $\mathbb{Z}_b((x^{-1}))$, which are series of the form

$$\sum_{\ell=w}^{\infty} a_\ell x^{-\ell}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

for some integer w and coefficients $a_\ell \in \mathbb{Z}_b$. Note that this set contains the set of polynomials $\mathbb{Z}_b[x]$. This set now permits addition, subtraction, multiplication and division (except dividing by 0 of course) of formal series where the coefficients are added, subtracted, multiplied and divided modulo b. Thus $\mathbb{Z}_b((x^{-1}))$ is a field, which is called the field of *formal Laurent series*. The construction of the generating matrices is based on arithmetic in this field.

(1) Let $p_1, p_2, \ldots, p_s \in \mathbb{Z}_b[x]$ be distinct monic irreducible polynomials over $\mathbb{Z}_b$ (monic means that the coefficient of the monomial in p_j with the highest degree is 1). Let $e_j = \deg(p_j)$ for $1 \le j \le s$. The best results are obtained by choosing the degree of the polynomials $p_1, p_2, \ldots, p_s$ as small as possible.

(2) For integers $1 \le j \le s$, $u \ge 1$ and $0 \le k < e_j$, consider the expansions

$$\frac{x^{e_j-k-1}}{p_j(x)^u} = \sum_{\ell=0}^{\infty} a^{(j)}(u,k,\ell)x^{-\ell-1} \tag{2.6}$$

over the field of formal Laurent series $\mathbb{Z}_b((x^{-1}))$.

(3) Then we define the entries in the matrix $C_j = (c_{j,i,\ell})_{i,\ell\in\mathbb{N}}$ in the following way:

$$c_{j,i,\ell} = a^{(j)}(Q+1,k,\ell) \in \mathbb{Z}_b \quad \text{for } 1 \le j \le s, i \ge 1, \ell \ge 0,$$

where $i-1 = Qe_j + k$ with integers $Q = Q(i,j)$ and $k = k(i,j)$, with k satisfying $0 \le k < e_j$.

We briefly describe how to compute the coefficients $a^{(j)}(u,k,\ell)$ recursively. Let u, j be fixed and let $v_\ell = a^{(j)}(u, e_j - 1, \ell)$ for all $\ell \ge 0$. Then $a^{(j)}(u,k,\ell) = v_{\ell+e_j-1-k}$ for $0 \le k < e_j$. Let $p_j(x)^u = x^{ue_j} - b_{ue_j-1}x^{ue_j-1} - \cdots - b_0$, where we assume without loss of generality that $p_j(x)$ is monic (*i.e.*, the coefficient of x^{e_j} is 1). Then (2.6) can be written as

$$1 = (v_0x^{-1} + v_1x^{-2} + \cdots)(x^{ue_j} - b_{ue_j-1}x^{ue_j-1} - \cdots - b_0).$$

By comparing coefficients we obtain $v_0 = \cdots = v_{ie_j-2} = 0$, $v_{ie_j-1} = 1$ and

$$v_{ue_j+\ell} = b_{ue_j-1}v_{ue_j+\ell-1} + \cdots + b_0v_\ell \in \mathbb{Z}_b,$$

for all $\ell \ge 0$.

The Niederreiter sequence is a digital (t,s)-sequence with

$$t = \sum_{j=1}^{s}(e_j - 1).$$

The Faure sequence can be obtained as a special case of the Niederreiter sequence, which corresponds to the case where the base b is a prime number

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

such that $b \geq s$ and $p_j(x) = x - j + 1$ for $1 \leq j \leq s$. The Sobol′ sequence is obtained by choosing $p_1(x) = x$ and $p_2, p_3, \ldots, p_s$ as primitive polynomials. This yields a Sobol′ sequence with a special choice of direction numbers. The Niederreiter sequence can also be generalized, where x^{e_j-k-1} is replaced by a polynomial of degree $e_j - k - 1$. The choice of polynomials then corresponds to choosing different direction numbers. This way one can obtain the generating matrices for a Sobol′ sequence. On the other hand, an algorithm similar to the one introduced above for Sobol′ sequences can also be obtained for Niederreiter sequences, which yields a fast implementation of Niederreiter sequences.

A practical implementation of digital nets and sequences can use a Gray code ordering to improve efficiency.

2.8. Polynomial lattice rule construction

In this subsection we discuss polynomial lattice rules, a concept analogous to lattice rules, but based on linear algebra over finite fields; they form a special class of digital nets.

Let b be prime, let p be a polynomial of degree m with coefficients in $\mathbb{Z}_b$, and let $q_1, \ldots, q_s$ be polynomials of degree at most $m - 1$ with coefficients in $\mathbb{Z}_b$. Then C_j, the jth generating matrix, can be chosen as follows.

(1) Let $u_1, u_2, \ldots \in \mathbb{Z}_b$ be such that

$$\frac{q_j(x)}{p(x)} = \frac{u_1}{x} + \frac{u_2}{x^2} + \cdots .$$

For given $q_j(x)$ and $p(x)$, the values of $u_1, u_2, \ldots$ can be obtained by equating coefficients in $q_j(x) = (u_1/x + u_2/x^2 + \cdots)p(x)$, noting that all additions and multiplications are to be done in $\mathbb{Z}_b$.

(2) Set

$$C_j = \begin{pmatrix} u_1 & u_2 & u_3 & \cdots & u_m \\ u_2 & u_3 & \iddots & \iddots & u_{m+1} \\ u_3 & \iddots & \iddots & \iddots & u_{m+2} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ u_m & u_{m+1} & u_{m+2} & \cdots & u_{2m-1} \end{pmatrix} \in \mathbb{Z}_b^{m \times m}.$$

The digital net with generating matrices $C_1, \ldots, C_s$ as defined above is called a *polynomial lattice point set*, and a QMC rule using a polynomial lattice point set is called a *polynomial lattice rule*. The polynomial p is referred to as the *modulus*, and the vector of polynomials $(q_1, \ldots, q_s)$ is referred to as the *generating vector*.

There is also a faster method for generating the cubature points of a polynomial lattice rule which does not use the generating matrices. For

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

the following we need to introduce some definitions and some notation. As before, let $\mathbb{Z}_b[x]$ denote the set of polynomials with coefficients in $\mathbb{Z}_b$ and let $\mathbb{Z}_b((x^{-1}))$ denote the set of formal Laurent series

$$\sum_{\ell=w}^{\infty} a_\ell x^{-\ell},$$

where $a_\ell \in \mathbb{Z}_b$. For $m \in \mathbb{N}$ let υ_m be the map from $\mathbb{Z}_b((x^{-1}))$ to the interval $[0,1)$ defined by

$$\upsilon_m\left(\sum_{\ell=w}^{\infty} t_\ell x^{-\ell}\right) = \sum_{\ell=\max(1,w)}^{m} t_\ell b^{-\ell}.$$

We frequently associate a non-negative integer k, with b-adic expansion $k = \kappa_0 + \kappa_1 b + \cdots + \kappa_a b^a$, with the polynomial $k(x) = \kappa_0 + \kappa_1 x + \cdots + \kappa_a x^a \in \mathbb{Z}_b[x]$ and *vice versa*. Further, for arbitrary $\boldsymbol{k} = (k_1, \ldots, k_s) \in \mathbb{Z}_b[x]^s$ and $\boldsymbol{q} = (q_1, \ldots, q_s) \in \mathbb{Z}_b[x]^s$, we define the 'inner product'

$$\boldsymbol{k} \cdot \boldsymbol{q} = \sum_{i=1}^{s} k_i q_i \in \mathbb{Z}_b[x],$$

and we write $q \equiv 0 \pmod{p}$ if p divides q in $\mathbb{Z}_b[x]$.

With these definitions we can give the following equivalent but simpler form of the construction of $\mathcal{P}(\boldsymbol{q}, p)$.

Theorem 2.21. Let b be a prime and let $m, s \in \mathbb{N}$. For $p \in \mathbb{Z}_b[x]$ with $\deg(p) = m$ and $\boldsymbol{q} = (q_1, \ldots, q_s) \in \mathbb{Z}_b[x]^s$, the polynomial lattice point set $\mathcal{P}(\boldsymbol{q}, p)$ is the point set consisting of the b^m points

$$\boldsymbol{t}_i = \left(\upsilon_m\left(\frac{i(x)q_1(x)}{p(x)}\right), \ldots, \upsilon_m\left(\frac{i(x)q_s(x)}{p(x)}\right)\right) \in [0,1)^s,$$

for $i \in \mathbb{Z}_b[x]$ with $\deg(i) < m$.

The task of choosing good generating matrices is now replaced by the task of choosing a good generating vector $(q_1, \ldots, q_s)$. The total number of possible choices of generating matrices is reduced from $b^{m^2 s}$ to b^{ms}.

Example 2.22. In this example we show the polynomial lattice rule analogue of Fibonacci lattice rules; they are constructed via the continued fraction expansion of the ratio $q_2(x)/p(x)$. Let b be prime and let $m \geq 1$ be an integer. Set $q_1(x) = 1$. Let $p(x)$ be a polynomial over $\mathbb{Z}_b$ of degree m and let $q_2(x)$ be the polynomial over $\mathbb{Z}_b$ defined by

$$\frac{q_2(x)}{p(x)} = \cfrac{1}{1 + x + \cfrac{1}{1+x+\cfrac{1}{1+x+\ddots}}}.$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

The corresponding polynomial lattice point set is a digital $(0, m, 2)$-net over $\mathbb{Z}_b$. For instance, let $b = 2$. Then for $m = 2$ we obtain

$$\frac{q_2(x)}{p(x)} = \frac{1}{1 + x + \frac{1}{1+x}} = \frac{1+x}{(1+x)^2 + 1} = \frac{1+x}{x^2},$$

and thus $p(x) = x^2$ and $q_2(x) = 1 + x$. For $m = 3$ we obtain

$$\frac{q_2(x)}{p(x)} = \frac{1}{1 + x + \frac{1}{1+x+\frac{1}{1+x}}} = \frac{x^2}{(1+x)x^2 + (1+x)} = \frac{x^2}{x^3 + x^2 + x + 1},$$

and thus $p(x) = x^3 + x^2 + x + 1$ and $q_2(x) = x^2$.

2.9. Randomization and error estimation: shifted lattice rules

We recall that the mean-square MC error can be estimated in practice using (2.1). In comparison, a fully deterministic QMC method, although having a faster rate of convergence, is biased and lacks a practical error estimate. 'Randomized' QMC methods combine the best of both worlds; their advantages are as follows.

- Randomization yields an unbiased estimator.
- Randomization provides a practical error estimate.
- Randomized QMC methods enjoy faster rates of convergence than the MC method for smooth functions.
- Some randomization technique (see 'scrambling' below) can further improve the QMC rate of convergence by an additional $O(n^{-1/2})$.

Here we discuss the simplest form of randomization called *shifting*, and we explain how an error estimate can be obtained in practice. We begin with a remark that *shifting preserves the lattice structure*, and therefore this randomization technique typically goes with lattice rules, to yield so-called *shifted lattice rules*. Having said that, shifting can be used together with any QMC method to obtain a practical error estimate (however, it need not preserve the original structure of the QMC point set).

The idea behind shifting is to move all the points in the same direction by the same amount. If any point falls outside the unit cube then it is 'wrapped' back into the cube from the opposite side. More precisely, given a vector $\boldsymbol{\Delta} \in [0,1]^s$, known as the *shift*, the $\boldsymbol{\Delta}$-shift of the QMC points $\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{n-1}$ yields points

$$\{\boldsymbol{t}_i + \boldsymbol{\Delta}\}, \quad i = 0, 1, \ldots, n-1, \tag{2.7}$$

where, as in (2.2), the braces indicate taking the fractional parts. Figure 2.3 illustrates how shifting is done.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

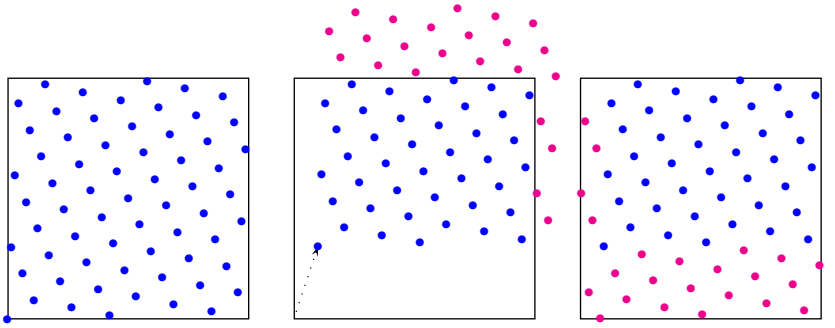


Figure 2.3. Applying a $(0.1, 0.3)$-shift to a 64-point lattice rule in two dimensions: (a) original lattice rule, (b) moving all points by $(0.1, 0.3)$, (c) wrapping the points back inside the unit cube.

For a random shift $\boldsymbol{\Delta} \in [0,1]^s$, the shifted QMC points $\{\boldsymbol{t}_i + \boldsymbol{\Delta}\}$, $i = 0, 1, \ldots, n-1$ are correlated. Therefore we cannot estimate the variance of the shifted QMC rule using the sample variance as in (2.1). Instead, we need to use a number of independent random shifts as follows.

(1) We generate q independent random shifts $\boldsymbol{\Delta}_0, \boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_{q-1}$ from the uniform distribution on $[0,1]^s$.

(2) For a given QMC rule, we form the approximations $Q_{n,s}^{(0)}(f)$, $Q_{n,s}^{(1)}(f)$, $\ldots$, $Q_{n,s}^{(q-1)}(f)$, where

$$Q_{n,s}^{(k)}(f) = \frac{1}{n} \sum_{i=0}^{n-1} f(\{\boldsymbol{t}_i + \boldsymbol{\Delta}_k\}), \quad k = 0, 1, \ldots, q-1,$$

is the approximation of the integral using a $\boldsymbol{\Delta}_k$-shift of the original QMC rule.

(3) We take the average

$$\bar{Q}_{n,s,q}(f) = \frac{1}{q} \sum_{k=0}^{q-1} Q_{n,s}^{(k)}(f)$$

as our final approximation to the integral.

(4) An unbiased estimate for the mean-square error of $\bar{Q}_{n,s,q}(f)$ is given by

$$\frac{1}{q(q-1)} \sum_{k=0}^{q-1} (Q_{n,s}^{(k)}(f) - \bar{Q}_{n,s,q}(f))^2.$$

Typically we take n in the thousands or more while keeping q small, say around 10–50. To obtain a fair comparison between the MC method and

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

a randomized QMC method (*e.g.*, for the two methods to have the same number of function evaluations), we should take $n^{\mathrm{MC}} = q \cdot n^{\mathrm{QMC}}$.

Scrambling is a popular but more complicated randomization method. The idea is to randomly and recursively permute the points between elementary intervals so that *the digital net structure is preserved*. More details will be given in the next subsection. For now we just introduce a related randomization technique called *digital shift*, which is similar to shifting. We just need to replace (2.7) by

$$\boldsymbol{t}_i \oplus \boldsymbol{\Delta}, \quad i = 0, 1, \ldots, n-1,$$

where $\oplus$ denotes the digit-wise addition operator in base b defined as follows. If $x, \sigma \in [0,1)$ have base b representations

$$x = (0.x_1 x_2 \cdots)_b \quad \text{and} \quad \sigma = (0.\sigma_1 \sigma_2 \cdots)_b,$$

then $y = x \oplus \sigma = (0.y_1 y_2 \cdots)_b$, where

$$y_i = (x_i + \sigma_i) \bmod b.$$

The procedure for estimating the error is the same as for shifting.

For later use we also define digit-wise subtraction $y = x \ominus \sigma = (0.y_1 y_2 \cdots)_b$, which is defined by $y_i = (x_i - \sigma_i) \bmod b$.

2.10. Scrambled nets and sequences

Randomization methods are designed to introduce randomness into the point sets but at the same time should preferably keep the relevant structure intact. In the case of (t, m, s)-nets this means that the net should still be a (t, m, s)-net after the randomization (with probability 1). This can be achieved by the following algorithm introduced by Owen (1997*a*).

We first introduce *Owen's scrambling algorithm*, which is most easily described for some generic point $\boldsymbol{x} \in [0,1)^s$, with $\boldsymbol{x} = (x_1, \ldots, x_s)$ and $x_j = x_{j,1} b^{-1} + x_{j,2} b^{-2} + \cdots$. The scrambled point will be denoted by $\boldsymbol{y} \in [0,1)^s$, where $\boldsymbol{y} = (y_1, \ldots, y_s)$ and $y_j = y_{j,1} b^{-1} + y_{j,2} b^{-2} + \cdots$. The point $\boldsymbol{y}$ is obtained by applying permutations to each digit of each coordinate of $\boldsymbol{x}$. The permutation applied to $x_{j,\ell}$ depends on $x_{j,k}$ for $1 \le k < \ell$. Specifically, $y_{j,1} = \pi_j(x_{j,1})$, $y_{j,2} = \pi_{j,x_{j,1}}(x_{j,2})$, $y_{j,3} = \pi_{j,x_{j,1},x_{j,2}}(x_{j,3})$, and in general

$$y_{j,k} = \pi_{j,x_{j,1},\ldots,x_{j,k-1}}(x_{j,k}), \tag{2.8}$$

where $\pi_{j,x_{j,1},\ldots,x_{j,k-1}}$ is a random permutation of $\{0, \ldots, b-1\}$. We assume that permutations with different indices are chosen mutually independent from each other and that each permutation is chosen with the same probability. In this case the scrambled point $\boldsymbol{y}$ is uniformly distributed in $[0,1)^s$. We show this fact in Section 6.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

We now introduce some convenient notation to describe Owen's scrambling. For $1 \le j \le s$ let

$$\Pi_j = \{\pi_{j,x_{j,1},\ldots x_{j,k-1}} : k \in \mathbb{N}, x_{j,1}, \ldots, x_{j,k-1} \in \{0, \ldots, b-1\}\}$$

(where for $k = 1$ we set $\pi_{j,x_{j,1},\ldots x_{j,k-1}} = \pi_j$) be a given set of permutations, and let $\mathbf{\Pi} = (\Pi_1, \ldots, \Pi_s)$. Then, when applying Owen's scrambling using these permutations to some point $\boldsymbol{x} \in [0,1)^s$, we write $\boldsymbol{y} = \mathbf{\Pi}(\boldsymbol{x})$, where $\boldsymbol{y}$ is the point obtained by applying Owen's scrambling to $\boldsymbol{x}$ using the permutations $\Pi_1, \ldots, \Pi_s$. For $x \in [0,1)$ we drop the subscript j and just write $y = \Pi(x)$.

To scramble a (t, m, s)-net or a (t, s)-sequence, we choose the permutations in $\mathbf{\Pi}$ for each coordinate independently from each other and then apply these permutations to each point of the (t, m, s)-net or (t, s)-sequence as explained above. For instance, for a given (t, m, s)-net $\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{b^m-1}$, the Owen-scrambled (t, m, s)-net is given by

$$\mathbf{\Pi}(\boldsymbol{t}_0), \mathbf{\Pi}(\boldsymbol{t}_1), \ldots, \mathbf{\Pi}(\boldsymbol{t}_{b^m-1}).$$

In practice, the number of permutations one has to choose is large, but there are simplifications of the randomization procedure which greatly reduce this number. This yields a fast scrambling algorithm. More details are given in Section 6.

In Section 6 we show that scrambling also yields an unbiased estimate of the integral, that is,

$$\mathbb{E}\left(\frac{1}{n}\sum_{i=0}^{n-1} f(\mathbf{\Pi}(\boldsymbol{t}_i))\right) = \int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.$$

In the same way as for random (digital) shifts, one can also obtain an unbiased estimate of the mean-square error for scrambling. It is sufficient to choose only one set of random permutations $\mathbf{\Pi}$. Let

$$Q_{n_1,n_2,s}(f) = \frac{1}{n_2 - n_1}\sum_{i=n_1}^{n_2-1} f(\mathbf{\Pi}(\boldsymbol{t}_i)).$$

Let $1 \le m' < m$. The mean-square error of the cubature rule applied to f can now be estimated by

$$\frac{1}{b^{m'}(b^{m'}-1)}\sum_{i=0}^{b^{m'}-1}\left(Q_{ib^{m-m'},(i+1)b^{m-m'},s}(f) - Q_{0,b^m,s}(f)\right)^2,$$

where m' can be chosen such that, say, $b^{m'} \approx 30$.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Figure 2.4. Owen's scrambling in base 2: (a) original; (b) swap left and right halves; (c) swap 3rd and 4th vertical quarters; (d) swap 3rd and 4th, 7th and last vertical eighths; (e) swap 3rd and 4th, 7th and 8th, 9th and 10th, 15th and last sixteenths; (f) swap 1st and 2nd horizontal quarters; (g) swap 1st and 2nd, 5th and 6th, 7th and last horizontal eighths; (h) swap 3rd and 4th, 7th and 8th, 9th and 10th, 15th and last horizontal sixteenths.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

2.11. Transformation to the unit cube

In this subsection we discuss some simple strategies for transforming an integral over $\mathbb{R}^s$ to an integral over $[0,1]^s$. We begin with the univariate case.

Consider an integral

$$\int_{-\infty}^{\infty} g(y)\,\phi(y)\,\mathrm{d}y,$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is some univariate probability density function, that is, $\phi(y) \geq 0$ for all $y \in \mathbb{R}$ and $\int_{-\infty}^{\infty} \phi(y)\,\mathrm{d}y = 1$. Let $\Phi : \mathbb{R} \to [0,1]$ denote the cumulative distribution function of ϕ, that is, $\Phi(y) := \int_{-\infty}^{y} \phi(t)\,\mathrm{d}t$, and let $\Phi^{-1} : [0,1] \to \mathbb{R}$ denote the inverse of the cumulative distribution function. Then we can use the substitution (or change of variables)

$$x = \Phi(y) \iff y = \Phi^{-1}(x),$$

to obtain

$$\int_{-\infty}^{\infty} g(y)\,\phi(y)\,\mathrm{d}y = \int_0^1 g(\Phi^{-1}(x))\,\mathrm{d}x = \int_0^1 f(x)\,\mathrm{d}x,$$

with the transformed integrand defined by $f := g \circ \Phi^{-1}$.

Note that this is not the only way to obtain a transformed integrand. Indeed, we can divide and multiply the original integrand by any other probability density function $\tilde{\phi}$, and then map to $[0,1]$ using its inverse cumulative distribution function $\tilde{\Phi}^{-1}$:

$$\begin{aligned}\int_{-\infty}^{\infty} g(y)\,\phi(y)\,\mathrm{d}y &= \int_{-\infty}^{\infty} \frac{g(y)\,\phi(y)}{\tilde{\phi}(y)}\,\tilde{\phi}(y)\,\mathrm{d}y \\ &= \int_{-\infty}^{\infty} \tilde{g}(y)\,\tilde{\phi}(y)\,\mathrm{d}y = \int_0^1 \tilde{g}(\tilde{\Phi}^{-1}(x))\,\mathrm{d}x = \int_0^1 \tilde{f}(x)\,\mathrm{d}x,\end{aligned}$$

where $\tilde{g}(y) := g(y)\,\phi(y)/\tilde{\phi}(y)$, giving a different transformed integrand $\tilde{f} := \tilde{g} \circ \tilde{\Phi}^{-1}$. Ideally we would like to choose a density function $\tilde{\phi}$ that leads to a 'nice' integrand in the unit cube. This is related to the concept of *importance sampling* for MC methods.

The MC approximation for the integral over $\mathbb{R}$ is

$$\int_{-\infty}^{\infty} g(y)\,\phi(y)\,\mathrm{d}y \approx \frac{1}{n}\sum_{i=0}^{n-1} g(\tau_i),$$

where the MC points $\tau_0, \ldots, \tau_{n-1}$ are randomly sampled following the distribution of ϕ. Depending on the choice of ϕ, there may exist an algorithm to sample from the distribution directly, or it might be necessary to generate random samples from the uniform distribution on $[0,1]$and then map

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

them back to $\mathbb{R}$ using Φ^{-1}. The latter approach is consistent with the MC method over $[0,1]$,

$$\int_{-\infty}^{\infty} g(y)\,\phi(y)\,\mathrm{d}y = \int_0^1 g(\Phi^{-1}(x))\,\mathrm{d}x \approx \frac{1}{n}\sum_{i=0}^{n-1} g(\Phi^{-1}(t_i)),$$

where $t_0,\ldots,t_{n-1}$ are uniform random samples from $[0,1]$. If we use a different density function $\tilde{\phi}$ to sample the points from, then we evaluate the points for a different function $\tilde{g}$:

$$\int_{-\infty}^{\infty} g(y)\,\phi(y)\,\mathrm{d}y = \int_{-\infty}^{\infty} \tilde{g}(y)\,\tilde{\phi}(y)\,\mathrm{d}y \approx \frac{1}{n}\sum_{i=0}^{n-1} \tilde{g}(\tau_i).$$

The idea behind importance sampling is to choose a sampling density $\tilde{\phi}$ to minimize the variance of the resulting function $\tilde{g}$.

This transformation strategy can be generalized to s dimensions as follows. If we have a product of univariate densities, then we can apply the mapping Φ^{-1} component-wise,

$$\boldsymbol{y} = \Phi^{-1}(\boldsymbol{x}) := (\Phi^{-1}(x_1),\ldots,\Phi^{-1}(x_s)),$$

to obtain

$$\int_{\mathbb{R}^s} g(\boldsymbol{y}) \prod_{j=1}^{s} \phi(y_j)\,\mathrm{d}\boldsymbol{y} = \int_{[0,1]^s} g(\Phi^{-1}(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} = \int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.$$

Note that we can always multiply and divide any given integrand by such a product provided that $\phi(y) > 0$ for all $y \in \mathbb{R}$:

$$\int_{\mathbb{R}^s} h(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} = \int_{\mathbb{R}^s} \frac{h(\boldsymbol{y})}{\prod_{j=1}^{s}\phi(y_j)} \prod_{j=1}^{s} \phi(y_j)\,\mathrm{d}\boldsymbol{y}.$$

Thus, this strategy always works in principle, but the resulting transformed integrand might not be 'nice'.

Many integrals from practical models involve the multivariate normal density. If the multivariate normal density is the dominating part of the entire integrand, then a good strategy is to *factorize the covariance matrix* Σ, that is, to find an $s \times s$ matrix A such that

$$\Sigma = AA^{\mathrm{T}}, \tag{2.9}$$

and then use the substitutions (treating all vectors as column vectors)

$$\boldsymbol{y} = A\boldsymbol{z} \quad \text{followed by} \quad \boldsymbol{z} = \Phi^{-1}(\boldsymbol{x}),$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

to obtain

$$\begin{aligned}\int_{\mathbb{R}^s} g(\boldsymbol{y})\,\frac{\exp(-\frac{1}{2}\boldsymbol{y}^{\mathsf{T}}\Sigma^{-1}\boldsymbol{y})}{\sqrt{(2\pi)^s\det(\Sigma)}}\,\mathrm{d}\boldsymbol{y} &= \int_{\mathbb{R}^s} g(A\boldsymbol{z})\,\frac{\exp(-\frac{1}{2}\boldsymbol{z}^{\mathsf{T}}\boldsymbol{z})}{\sqrt{(2\pi)^s}}\,\mathrm{d}\boldsymbol{z}\\ &= \int_{\mathbb{R}^s} g(A\boldsymbol{z})\,\prod_{j=1}^{s}\frac{\exp(-\frac{1}{2}z_j^2)}{\sqrt{2\pi}}\,\mathrm{d}\boldsymbol{z}\\ &= \int_{[0,1]^s} g(A\Phi^{-1}(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} = \int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.\end{aligned}$$

The factorization (2.9) is *not* unique. Two obvious choices are the *Cholesky factorization* with lower triangular matrix A, and the *principal components factorization*, which is given by

$$A = \big[\sqrt{\lambda_1}\boldsymbol{\eta}_1;\cdots;\sqrt{\lambda_s}\boldsymbol{\eta}_s\big],$$

where $(\lambda_j,\boldsymbol{\eta}_j)_{j=1}^s$ denotes the set of eigenpairs of Σ, with ordered eigenvalues $\lambda_1\ge\lambda_2\ge\cdots\ge\lambda_s$ and unit-length column eigenvectors $\boldsymbol{\eta}_1,\ldots,\boldsymbol{\eta}_s$. Other factorizations are possible: for example, in finance applications the covariance matrix arising from the time-discretized Brownian paths can also be factorized using the 'Brownian bridge' technique. In general, factorization of Σ in very high dimensions can be very costly, and the problem can be poorly conditioned.

In some applications the multivariate normal density is *not* the dominating part of the entire integrand. In those cases, other transformation steps (such as recentering and rescaling) might be required to capture the main feature of the integrand: see, for example, Kuo *et al.* (2008*a*).

2.12. Notes

Halton (1960) introduced what is now called the Halton sequence. Discrepancy bounds for Halton sequences where the constant decreases with the dimension have been shown by Atanassov (2004*b*). Kronecker sequences were studied by Niederreiter (1978). Lattice rules were introduced by Korobov (1959) and have also been discovered by Hlawka (1962). For the early history of QMC point sets and related concepts see Niederreiter (1978), Kuipers and Niederreiter (1974), Niederreiter (1992*a*) and Sloan and Joe (1994). More recent results are covered by Dick and Pillichshammer (2010).

For an efficient implementation of Sobol′ sequences see Antonov and Saleev (1979) and Bratley and Fox (1988), and for questions concerning the choice of primitive polynomials and direction numbers, see, for example, Joe and Kuo (2008). The implementation of Faure sequences has been discussed in Atanassov (2004*a*) and Fox (1986). Implementation of the Niederreiter sequence has been discussed in Bratley, Fox and Niederreiter (1992). For digital sequences with improved t-value see Niederreiter and

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Xing (1995, 1996*a*, 1996*b*) and Xing and Niederreiter (1995). For an implementation of such sequences see Pirsic (2002).

It has been observed that the quality of QMC point sets usually decreases as the dimension increases. This has led to the suggestion to study 'mixed' point sets, that is, the first few coordinates are QMC point sets with the remaining coordinates filled up by, say, random numbers. This fits in with the structure of integrands whose importance is concentrated in the first few coordinates. The idea of combining the advantages of QMC methods and MC methods was first proposed by Spanier, who applied mixed QMC–MC sequences to particle transport problems. Probabilistic discrepancy bounds for such 'mixed sequences' have been provided in Ökten (1996), Ökten, Tuffin and Burago (2006), Gnewuch (2009), Gnewuch and Roşca (2009) and Aistleitner and Hofer (2012); in these papers it is assumed that the Monte Carlo components of the points consist of idealized random numbers. There have also been studies of mixed point sets whose remaining coordinates are filled up by deterministic pseudo-random numbers or by the components of other deterministic point sets. Deterministic discrepancy bounds for such point sets can be found in Niederreiter (2009, 2010*a*, 2010*b*), Niederreiter and Winterhof (2011), Hofer and Kritzer (2011) and Niederreiter (2012). Further mixed sequences using Halton, Kronecker and digital sequences have been studied by Hofer and Larcher (2010, 2012) and Hofer, Kritzer, Larcher and Pillichshammer (2009). So-called $(t, \boldsymbol{e}, s)$-sequences have been introduced by Tezuka (2013). This concept captures the dependence on the dimension of digital nets more precisely and leads to improvements of discrepancy bounds in terms of their dependence on the dimension. Digital sequences in which the generating matrices have only a finite number of non-zero elements in each row have been studied, for instance, by Hofer and Pirsic (2011) and Hofer and Niederreiter (2013). The latter paper constructs digital (t, s)-sequences with finite-row generating matrices and asymptotically optimal t-values (in the sense of fixed base and dimension s going to ∞). Such sequences with finite-row generating matrices may be advantageous in implementations.

More information on transformations from $\mathbb{R}^s$ to the unit cube $[0,1]^s$, variance reduction techniques, and other information for the practical use of MC and QMC, can be found in Lemieux (2009). The monographs by Devroye (1986) and Hörmann, Leydold and Derflinger (2010) are primarily concerned with transformations. Integration of functions with singularities has been discussed by Owen (2006). Kuo *et al.* (2008*a*) considered transformation strategies for applying QMC methods to maximum likelihood integrals from statistics. Kuo, Sloan, Wasilkowski and Waterhouse (2010*a*) analysed lattice rules for unbounded integrands arising from the transformation. A construction of digital nets constructed in $\mathbb{R}^s$ has been studied by Dick (2011*b*).

Information relevant to statistical applications can be found in Fang and Wang (1994). Control variates for QMC have been discussed by Hickernell, Lemieux and Owen (2005). QMC methods have been applied to computer graphics by Keller (2006, 2013), to transport problems by Spanier, and to experimental designs by Hickernell (1999).

An introduction to lattice rules can be found in Sloan and Joe (1994); see also Niederreiter (1992*a*, Chapter 5). Dick and Pillichshammer (2010) provide a comprehensive introduction to (digital) (t, m, s)-nets, (digital) (t, s)-sequences and related point sets and sequences. A survey of QMC methods and their randomization strategies is given by Hickernell and Hong (2002).

Sparse grids were discovered independently by Smolyak (1963) and Zenger (1991). See also Wasilkowski and Woźniakowski (1995) and the review article by Bungartz and Griebel (2004).

3. Error, discrepancy, and reproducing kernel

In this section we present the necessary ingredients for the error analysis of QMC methods. To provide an accessible entry point for readers who are new to QMC, we begin by studying QMC in its simplest setting, namely, equal-weight quadrature rules for integration over the unit interval $[0, 1]$.

We will meet the useful notion of reproducing kernel Hilbert spaces, and meet specific examples of function spaces that have proved useful for designing and analysing QMC methods. All of the spaces we deal with in this and the next section are spaces of non-periodic functions. We follow contemporary practice in referring to such spaces as 'Sobolev' spaces, to distinguish them from 'Korobov' spaces of periodic functions, to be considered later in Section 5.8.

3.1. Error analysis in one dimension

We consider real-valued functions f defined on the interval $[0, 1]$. As is common in numerical analysis, we require that the functions have some smoothness and that the fundamental theorem of calculus holds, so that

$$f(x) = f(1) - \int_x^1 f'(y)\,\mathrm{d}y = f(1) - \int_0^1 f'(y) 1_{[0,y]}(x)\,\mathrm{d}y, \tag{3.1}$$

where the indicator function $1_{[0,y]}$ is defined by

$$1_{[0,y]}(x) = \begin{cases} 1 & \text{if } x \in [0, y], \\ 0 & \text{if } x \notin [0, y]. \end{cases}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

We consider QMC rules of the form

$$Q(f) = \frac{1}{n}\sum_{i=0}^{n-1} f(t_i),$$

and study their integration error given by

$$\text{error}_n(f;Q) = \int_0^1 f(x)\,\mathrm{d}x - \frac{1}{n}\sum_{i=0}^{n-1} f(t_i).$$

Upon substituting (3.1) in the equation above and changing the order of integration, we obtain

$$\text{error}_n(f;Q) = \int_0^1 \left(-\int_0^1 1_{[0,y]}(x)\,\mathrm{d}x + \frac{1}{n}\sum_{i=0}^{n-1} 1_{[0,y]}(t_i) \right) f'(y)\,\mathrm{d}y,$$

or

$$\text{error}_n(f;Q) = \int_0^1 \Delta_P(y) f'(y)\,\mathrm{d}y, \tag{3.2}$$

where $P := \{t_0, \ldots, t_{n-1}\} \subset [0,1]$ denotes the set of quadrature points, and $\Delta_P(y)$ is the *local discrepancy* of the point set P, defined by

$$\Delta_P(y) := \frac{1}{n}\sum_{i=0}^{n-1} 1_{[0,y]}(t_i) - \int_0^1 1_{[0,y]}(x)\,\mathrm{d}x, \quad y \in [0,1]. \tag{3.3}$$

Note that the local discrepancy function Δ_P is just the integration error for the indicator function $1_{[0,y]}$.

Thus from (3.2) the integration error can be viewed as the L_2 inner product of the local discrepancy function Δ_P and the derivative f' of the integrand.

Using Hölder's inequality we obtain from (3.2)

$$\begin{aligned} |\text{error}_n(f;Q)| &\le \left(\int_0^1 |\Delta_P(y)|^p\,\mathrm{d}y \right)^{1/p} \left(\int_0^1 |f'(y)|^q\,\mathrm{d}y \right)^{1/q} \\ &= \|\Delta_P\|_{L_p} \|f'\|_{L_q}, \quad \frac{1}{p} + \frac{1}{q} = 1, \end{aligned} \tag{3.4}$$

where $\|g\|_{L_p} = (\int_0^1 |g(y)|^p\,\mathrm{d}y)^{1/p}$, and we make the usual modification if $p = \infty$. If $p = \infty$ and $q = 1$, this is a special case of *Koksma's inequality* (Koksma's inequality holds when one replaces $\|f'\|_{L_1}$ with the variation of f). The quantity $\|\Delta_P\|_{L_p}$ is referred to as the *discrepancy* of the point set P, and is closely related to the so-called *worst-case error*, which we shall introduce formally in Section 3.3.

Several conclusions can be drawn from this elementary inequality. First, the upper bound (3.4) is best possible: indeed, equality holds if $\Delta_p(y) f'(y) \ge 0$ and $|f'(y)|^q$ is a multiple of $|\Delta_P(y)|^p$. Second, to minimize the integration

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

error for all functions for which $\|f'\|_{L_q} < \infty$, one should choose quadrature points P for which the discrepancy $\|\Delta_P\|_{L_p}$ is, in some sense, small. Third, it is enough to study the integration error of the indicator functions $1_{[0,y]}$ for $y \in [0,1]$, since the local discrepancy function Δ_P evaluated at y is the integration error of the indicator function $1_{[0,y]}$.

The above analysis of the integration error rests entirely on the integral representation of the function f given by

$$f(x) = f(1) - \int_0^1 f'(y) 1_{[0,y]}(x)\,\mathrm{d}y. \tag{3.5}$$

For two functions f, g that permit such a representation, we can define an inner product

$$\langle f, g\rangle := f(1)g(1) + \int_0^1 f'(y)g'(y)\,\mathrm{d}y, \tag{3.6}$$

and a corresponding norm

$$\|f\| = \sqrt{|f(1)|^2 + \int_0^1 |f'(y)|^2\,\mathrm{d}y}.$$

Then (3.4) holds with $p = q = 2$ for all functions in the function space

$$H := \{f : [0,1] \to \mathbb{R} : f \text{ is absolutely continuous and } \|f\| < \infty\}.$$

3.2. Reproducing kernel Hilbert spaces

We have seen above that the integration error for functions $f \in H$ has the integral representation (3.2) involving the local discrepancy function. This rests in turn on the representation of f given by (3.5). We now express (3.5) in the language of *reproducing kernels*. The idea is to find a function $K : [0,1] \times [0,1] \to \mathbb{R}$ such that

$$\langle f, K(\cdot, x)\rangle = f(x) \quad \text{for all } f \in H \text{ and all } x \in [0,1].$$

For this to hold we must have, from (3.6) and (3.5),

$$f(1)K(1,x) + \int_0^1 f'(y)\frac{\partial K}{\partial y}(x,y)\,\mathrm{d}y = f(1) - \int_0^1 f'(y)1_{[0,y]}(x)\,\mathrm{d}y,$$

which is clearly satisfied for all $f \in H$ if

$$K(1,x) = 1 \quad \text{and} \quad \frac{\partial K}{\partial y}(x,y) = -1_{[0,y]}(x) \quad \text{for all } x, y \in [0,1].$$

This can be achieved by taking

$$K(x,y) = 2 - \max(x,y). \tag{3.7}$$

The function of two variables $K(x,y)$ given by (3.7) is our first example of a reproducing kernel, and H is our first example of a reproducing kernel Hilbert space.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

We now formally define a reproducing kernel Hilbert space of real-valued functions defined on $[0,1]^s$.

Definition 3.1 (reproducing kernel Hilbert space). The Hilbert space $H(K)$ with inner product $\langle \cdot, \cdot \rangle_H$ is a *reproducing kernel Hilbert space* (RKHS) with kernel $K : [0,1]^s \times [0,1]^s \to \mathbb{R}$ if:

- $K(\cdot, \boldsymbol{x}) \in H$ for all $\boldsymbol{x} \in [0,1]^s$,
- (the *reproducing property*) $f(\boldsymbol{x}) = \langle f, K(\cdot, \boldsymbol{x}) \rangle_H$ for all $f \in H$ and all $\boldsymbol{x} \in [0,1]^s$.

For a given RKHS there is a uniquely defined reproducing kernel satisfying the above two properties. In fact, every Hilbert space where *point evaluation is a bounded linear functional* has a reproducing kernel. This follows from the Riesz representation theorem. Reproducing kernels have the additional properties that:

- (*symmetry*) $K(\boldsymbol{x}, \boldsymbol{y}) = K(\boldsymbol{y}, \boldsymbol{x})$ for all $\boldsymbol{x}, \boldsymbol{y} \in [0,1]^s$,
- (*positive semidefiniteness*) $\sum_{i=1}^{M} \sum_{k=1}^{M} a_i a_k K(\boldsymbol{t}_i, \boldsymbol{t}_k) \geq 0$ for all $M \geq 1$ and all $a_1, \ldots, a_M \in \mathbb{R}$ and $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_M \in [0,1]^s$.

Indeed, any function $K : [0,1]^s \times [0,1]^s \to \mathbb{R}$ which is symmetric and positive semidefinite is a reproducing kernel to which there corresponds a uniquely defined inner product and RKHS. A comprehensive theory on reproducing kernels can be found in Aronszajn (1950).

The thing that makes an RKHS so useful in the problem of numerical integration is that, as we shall demonstrate in Section 3.3, there exists a closed-form expression for the worst-case error. That expression for the worst-case error is expressed in terms of the kernel of the RKHS, and so is particularly useful when there exists a simple closed expression for the kernel K.

We remark that it is not always easy to find the explicit kernel of an RKHS with a given inner product. The particular example above, where the kernel is not only known but is of very simple piecewise linear form, has played an important part, as we shall see, in the recent development of QMC methods. We shall meet other reproducing kernels in Section 3.4.

We now show how to use a reproducing kernel for a space of functions of a single variable as a building block for a multivariate RKHS. First we need the following definition.

Definition 3.2 (tensor product Hilbert space). A Hilbert space H_s of functions on $[0,1]^s$ is a *tensor product* of Hilbert spaces $H_{1,1}, H_{1,2}, \ldots, H_{1,s}$ of functions on $[0,1]$, written as

$$H_s = H_{1,1} \otimes H_{1,2} \otimes \cdots \otimes H_{1,s},$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

if it is the completion of the span of products $\prod_{j=1}^{s} f_j(x_j)$, where $f_j \in H_{1,j}$, under the norm in H_s. The H_s norm of the simple product $\prod_{j=1}^{s} f_j(x_j)$ is just the product of the norms $\|f_j\|_{H_{1,j}}$.

Tensor product spaces provide a popular setting for studying QMC methods, for the very good reason that *the reproducing kernel for a tensor product of reproducing kernel Hilbert spaces is just the product of the kernels.*

Example 3.3 (a key tensor product RKHS). Take each one-dimensional RKHS $H_{1,j}$ to be the space of single-variable absolutely continuous functions with reproducing kernel (3.7). Then the tensor product space H_s is the s-variable RKHS with reproducing kernel

$$K_s(\boldsymbol{x}, \boldsymbol{y}) = \prod_{j=1}^{s} (2 - \max(x_j, y_j)).$$

The inner product in this space is

$$\langle f, g \rangle_{H_s} = \sum_{\mathfrak{u} \subseteq \{1:s\}} \int_{[0,1]^{|\mathfrak{u}|}} \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{x}_{\mathfrak{u}}} f(\boldsymbol{x}_{\mathfrak{u}}; \mathbf{1}) \, \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{x}_{\mathfrak{u}}} g(\boldsymbol{x}_{\mathfrak{u}}; \mathbf{1}) \, \mathrm{d}\boldsymbol{x}_{\mathfrak{u}}, \tag{3.8}$$

and the norm is

$$\|f\|_{H_s} = \left(\sum_{\mathfrak{u} \subseteq \{1:s\}} \int_{[0,1]^{|\mathfrak{u}|}} \left| \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{x}_{\mathfrak{u}}} f(\boldsymbol{x}_{\mathfrak{u}}; \mathbf{1}) \right|^2 \mathrm{d}\boldsymbol{x}_{\mathfrak{u}} \right)^{1/2}.$$

Here $\{1 : s\}$ is a shorthand notation for $\{1, 2, \ldots, s\}$, and the sum is over all subsets $\mathfrak{u} \subseteq \{1 : s\}$, including the empty set, while for $\boldsymbol{x} \in [0, 1]^s$ the symbol $\boldsymbol{x}_{\mathfrak{u}}$ denotes the set of components x_j of $\boldsymbol{x}$ with $j \in \mathfrak{u}$, and $(\boldsymbol{x}_{\mathfrak{u}}; \mathbf{1})$ indicates that the components of $\boldsymbol{x}$ for $j \notin \mathfrak{u}$ are replaced by 1. The partial derivative $\partial^{|\mathfrak{u}|}/\partial \boldsymbol{x}_{\mathfrak{u}}$ denotes the mixed first partial derivative with respect to the components $\boldsymbol{x}_{\mathfrak{u}}$. Which functions lie in this space? The answer is all real-valued functions on $[0, 1]^s$ that have square-integrable mixed first derivatives and that are expressible in the form

$$\begin{aligned} f(\boldsymbol{x}) &= \langle f, K(\cdot, \boldsymbol{x}) \rangle_{H_s} \\ &= \sum_{\mathfrak{u} \subseteq \{1:s\}} (-1)^{|\mathfrak{u}|} \int_{[0,1]^{|\mathfrak{u}|}} \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} f(\boldsymbol{y}_{\mathfrak{u}}; \mathbf{1}) \, 1_{[\mathbf{0}, \boldsymbol{y}_{\mathfrak{u}}]}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{y}_{\mathfrak{u}}, \quad \boldsymbol{x} \in [0, 1]^s. \end{aligned}$$

We refer to this space as an *anchored Sobolev space* with the anchor point 1.

3.3. Worst-case error in an RKHS

We have foreshadowed in Section 3.1 that the discrepancy of a point set is closely related to the worst-case error. We now give the formal definition.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Definition 3.4 (worst-case error). The *worst-case error* of a QMC rule $Q_{n,s}$ using the point set $P \subset [0,1]^s$ in a normed space H (not necessarily a Hilbert space) is

$$e_{n,s}(P;H) := \sup_{\|f\|_H \le 1} |I_s(f) - Q_{n,s}(f)|,$$

that is, it is the worst error attained by $Q_{n,s}$ for f in the unit ball of H. The *initial error* is

$$e_{0,s}(H) := \sup_{\|f\|_H \le 1} |I_s(f)|,$$

which is the error obtained with the QMC rule replaced by zero.

Due to linearity, for any function $f \in H$ we have

$$|I_s(f) - Q_{n,s}(f)| \le e_{n,s}(P;H)\, \|f\|_H,$$

which takes the same form as the inequality in (3.4). Later in Section 3.5 we shall see other inequalities of the same form.

Worst-case errors are in general hard to compute, except for the case of an RKHS. In every RKHS of functions defined on the unit cube (not just a tensor product space), the following theorem gives a formula for the worst-case error in terms of the reproducing kernel.

Theorem 3.5 (formula for the squared worst-case error). Let $K : [0,1]^s \times [0,1]^s \to \mathbb{R}$ be a reproducing kernel that satisfies

$$\int_{[0,1]^s} \int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{y})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y} < \infty.$$

The *squared worst-case error* and *initial error* for a QMC rule in an RKHS $H_s(K)$ with reproducing kernel K satisfy

$$e^2_{n,s}(P;H_s(K)) = \int_{[0,1]^s} \int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{y})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y} - \frac{2}{n} \sum_{i=0}^{n-1} \int_{[0,1]^s} K(\boldsymbol{t}_i,\boldsymbol{y})\, \mathrm{d}\boldsymbol{y} + \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{k=0}^{n-1} K(\boldsymbol{t}_i,\boldsymbol{t}_k), \tag{3.9}$$

and

$$e^2_{0,s}(H_s(K)) = \int_{[0,1]^s} \int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{y})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y}. \tag{3.10}$$

Proof. We give here the proof for $s = 1$. The proof for general s can be easily obtained by replacing the unit interval $[0,1]$ in the argument below by the unit cube $[0,1]^s$.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

For an $f \in H$ we write the reproducing property $f(x) = \langle f, K(\cdot, x)\rangle_H$ with successive arguments $x = t_1, t_2, \cdots$ and average the results to obtain, for the quadrature sum,

$$\frac{1}{n}\sum_{i=0}^{n-1} f(t_i) = \frac{1}{n}\sum_{i=0}^{n-1}\langle f, K(\cdot, t_i)\rangle_H = \left\langle f, \frac{1}{n}\sum_{i=0}^{n-1} K(\cdot, t_i)\right\rangle_H.$$

In a similar way we find

$$\int_0^1 f(x)\,\mathrm{d}x = \int_0^1 \langle f, K(\cdot, x)\rangle_H\,\mathrm{d}x = \left\langle f, \int_0^1 K(\cdot, x)\,\mathrm{d}x\right\rangle_H,$$

provided that integration from 0 to 1 is a bounded linear functional in H, or equivalently that $\int_0^1 K(\cdot, x)\,\mathrm{d}x \in H$. In turn this requires

$$\begin{aligned}\left\|\int_0^1 K(\cdot, x)\,\mathrm{d}x\right\|_H^2 &= \int_0^1\int_0^1 \big\langle K(\cdot, x), K(\cdot, y)\big\rangle_H\,\mathrm{d}x\,\mathrm{d}y\\ &= \int_0^1\int_0^1 K(x, y)\,\mathrm{d}x\,\mathrm{d}y < \infty,\end{aligned}$$

which will hold for all the kernels we shall consider. Note that in the last step we used the reproducing property. By subtraction, the integration error is

$$\int_0^1 f(x)\,\mathrm{d}x - \frac{1}{n}\sum_{i=0}^{n-1} f(t_i) = \left\langle f, \int_0^1 K(\cdot, x)\,\mathrm{d}x - \frac{1}{n}\sum_{i=0}^{n-1} K(\cdot, t_i)\right\rangle_H = \langle f, \xi\rangle_H,$$

where

$$\xi(y) := \int_0^1 K(x, y)\,\mathrm{d}x - \frac{1}{n}\sum_{i=0}^{n-1} K(y, t_i), \quad y \in [0, 1]. \tag{3.11}$$

We call the function ξ the *representer* of the integration error. Thus

$$e_{n,1}(P; H) = \sup_{\|f\|_H \le 1} |\langle f, \xi\rangle_H| = \|\xi\|_H,$$

since the supremum is attained by the choice $f = \xi/\|\xi\| \in H$. Therefore

$$\begin{aligned}e^2_{n,1}(P; H) &= \langle \xi, \xi\rangle_H\\ &= \left\langle \int_0^1 K(\cdot, x)\,\mathrm{d}x - \frac{1}{n}\sum_{i=0}^{n-1} K(x\cdot, t_i), \int_0^1 K(\cdot, x)\,\mathrm{d}x - \frac{1}{n}\sum_{i=0}^{n-1} K(x, t_i)\right\rangle_H\\ &= \int_0^1\int_0^1 K(x, y)\,\mathrm{d}x\,\mathrm{d}y - \frac{2}{n}\sum_{i=0}^{n-1}\int_0^1 K(x, t_i)\,\mathrm{d}x + \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{k=0}^{n-1} K(t_i, t_k),\end{aligned}$$

where we again used the reproducing property of the kernel. □

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

For the particular Hilbert space with reproducing kernel $K(x,y) = 2 - \max(x,y)$, it is easily seen that

$$\int_0^1 K(x,y)\,\mathrm{d}y = \frac{3-x^2}{2} \quad \text{and} \quad \int_0^1\int_0^1 K(x,y)\,\mathrm{d}x\,\mathrm{d}y = \frac{4}{3},$$

so that the worst-case error is

$$e_{n,1}^2(P;H) = \frac{4}{3} - \frac{2}{n}\sum_{i=0}^{n-1}\left(\frac{3-t_i^2}{2}\right) + \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{k=0}^{n-1}(2-\max(t_i,t_k)).$$

For future purposes we note that the derivative of the error representer ξ is $\xi' = \Delta_P$, the local discrepancy function.

For the corresponding tensor product RKHS the worst-case error is

$$e_{n,s}^2(P;H_s) \tag{3.12}$$
$$= \left(\frac{4}{3}\right)^s - \frac{2}{n}\sum_{i=0}^{n-1}\prod_{j=1}^{s}\left(\frac{3-t_{i,j}^2}{2}\right) + \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{k=0}^{n-1}\prod_{j=1}^{s}\bigl(2-\max(t_{i,j},t_{k,j})\bigr),$$

where $t_{i,j}$ denotes the jth component of the ith cubature point $\boldsymbol{t}_i$. This is sometimes called the *squared L_2 discrepancy* of the point set P, for reasons that will soon become clear.

It proves to be very useful (and surprisingly easy) to compute the average of the squared worst-case error over all cubature points,

$$E_{n,s}^2(H_s(K)) := \int_{[0,1]^s}\cdots\int_{[0,1]^s} e_{n,s}^2(\boldsymbol{t}_0,\ldots,\boldsymbol{t}_{n-1};H_s(K))\,\mathrm{d}\boldsymbol{t}_0\cdots\mathrm{d}\boldsymbol{t}_{n-1}.$$

We refer to the quantity $E_{n,s}(K)$ as the *QMC mean* of the worst-case error. (*Why* this is a useful quantity will be explained after the theorem.)

Theorem 3.6 (QMC mean). Let $K : [0,1]^s \times [0,1]^s \to \mathbb{R}$ be a reproducing kernel that satisfies

$$\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} < \infty.$$

The *QMC mean* of the worst-case error in a reproducing kernel Hilbert space $H_s(K)$ of functions on $[0,1]^s$ and with kernel K satisfies

$$E_{n,s}^2(H_s(K)) = \frac{1}{n}\left(\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} - \int_{[0,1]^s}\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y}\right). \tag{3.13}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Proof. Using Theorem 3.5 we have

$$\begin{aligned}
&E_{n,s}^2(H_s(K)) \\
&= \int_{[0,1]^s} \cdots \int_{[0,1]^s} \Bigg(\int_{[0,1]^s} \int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y} - \frac{2}{n}\sum_{i=0}^{n-1} \int_{[0,1]^s} K(\boldsymbol{t}_i,\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \\
&\qquad + \frac{1}{n^2}\sum_{i=0}^{n-1} K(\boldsymbol{t}_i,\boldsymbol{t}_i) + \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{\substack{k=0\\k\neq i}}^{n-1} K(\boldsymbol{t}_i,\boldsymbol{t}_k)\Bigg)\mathrm{d}\boldsymbol{t}_0\cdots\mathrm{d}\boldsymbol{t}_{n-1},
\end{aligned}$$

noting that in the double sum we separated the diagonal and off-diagonal terms as in the proof of Theorem 2.1. The first term inside the parentheses is independent of $\boldsymbol{t}_0,\ldots,\boldsymbol{t}_{n-1}$, thus these variables can all be integrated out to give the result 1. In the second term, after interchanging the outer integrations and the sum, all variables but $\boldsymbol{t}_i$ can be integrated out. The situation is similar for the third term, while for the fourth term all but $\boldsymbol{t}_i$ and $\boldsymbol{t}_k$ can be integrated out. We are left with

$$\begin{aligned}
E_{n,s}^2(K) &= \int_{[0,1]^s}\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y} - \frac{2}{n}\sum_{i=0}^{n-1}\int_{[0,1]^s}\int_{[0,1]^s} K(\boldsymbol{t}_i,\boldsymbol{y})\,\mathrm{d}\boldsymbol{t}_i\,\mathrm{d}\boldsymbol{y} \\
&\quad + \frac{1}{n^2}\sum_{i=0}^{n-1}\int_{[0,1]^s} K(\boldsymbol{t}_i,\boldsymbol{t}_i)\,\mathrm{d}\boldsymbol{t}_i + \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{\substack{k=0\\k\neq i}}^{n-1}\int_{[0,1]^s}\int_{[0,1]^s} K(\boldsymbol{t}_i,\boldsymbol{t}_k)\,\mathrm{d}\boldsymbol{t}_i\,\mathrm{d}\boldsymbol{t}_k \\
&= \int_{[0,1]^s}\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y} - 2\int_{[0,1]^s}\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y} \\
&\quad + \frac{1}{n}\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \frac{n-1}{n}\int_{[0,1]^s}\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y} \\
&= \frac{1}{n}\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} - \frac{1}{n}\int_{[0,1]^s}\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y}. \qquad \square
\end{aligned}$$

The importance of Theorem 3.6 is that it gives us an immediate existence result for QMC integration: by applying the principle that *there is always at least one choice as good as the average*, we conclude that there exists a QMC point set P for which

$$e_{n,s}^2(P;H_s(K)) \le E_{n,s}^2(H_s(K)).$$

Thus, provided that both integrals on the right-hand side in Theorem 3.6 are finite (as they are in all the cases we will consider), there exists a QMC point set with worst-case error of order $O(n^{-1/2})$, that is, the same rate of convergence as Monte Carlo. Later we will not be satisfied with this result, for two reasons: first, we want not just the Monte Carlo rate of convergence,

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

but rather a rate of convergence close to $O(n^{-1})$, or even better; second, we want not just existence, but also a method of constructing the points.

3.4. Other univariate reproducing kernels

Note that (3.1) can be viewed as a Taylor series with integral remainder, developed at 1. As such it is clear that we could choose an arbitrary anchor point $c \in [0,1]$ instead of 1. Or we can consider function spaces whose members have more than one derivative. Or we can depart from a Taylor series representation of f to obtain other examples of reproducing kernels. All of the following examples can be used as building blocks for multivariate Sobolev spaces.

Example 3.7 (anchored reproducing kernel of smoothness α). If the $(\alpha-1)$th derivative of f is absolutely continuous for some natural number α, then the Taylor series of a univariate function f anchored at 0 is

$$f(x) = \sum_{r=0}^{\alpha-1} \frac{f^{(r)}(0)}{r!} x^r + \int_0^1 f^{(\alpha)}(y) \frac{(x-y)_+^{\alpha-1}}{(\alpha-1)!}\,\mathrm{d}y,$$

where $f^{(r)}$ denotes the rth derivative of f and

$$(x-y)_+^{\alpha-1} = \begin{cases} (x-y)^{\alpha-1} & \text{if } x > y, \\ 0 & \text{if } x \le y. \end{cases}$$

For functions f, g permitting such a Taylor series presentation, we can define the inner product

$$\langle f, g\rangle = \sum_{r=0}^{\alpha-1} f^{(r)}(0) g^{(r)}(0) + \int_0^1 f^{(\alpha)}(y) g^{(\alpha)}(y)\,\mathrm{d}y.$$

The set of functions whose norm corresponding to this inner product is finite is another RKHS, with kernel given by

$$K_\alpha(x,y) = \sum_{r=0}^{\alpha-1} \frac{x^r}{r!}\frac{y^r}{r!} + \int_0^1 \frac{(x-z)_+^{\alpha-1}}{(\alpha-1)!}\frac{(y-z)_+^{\alpha-1}}{(\alpha-1)!}\,\mathrm{d}z.$$

We call the corresponding RKHS the *anchored Sobolev space of smoothness α with anchor* 0.

Example 3.8 (unanchored reproducing kernel of smoothness α). For a univariate function f belonging to the same smoothness class as in the preceding example, consider the representation given by

$$f(x) = \sum_{r=0}^{\alpha} \frac{B_r(x)}{r!} \int_0^1 f^{(r)}(y)\,\mathrm{d}y - (-1)^\alpha \int_0^1 \frac{B_\alpha(|x-y|)}{\alpha!} f^{(\alpha)}(y)\,\mathrm{d}y.$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Here, B_r denotes the Bernoulli polynomial of degree r: see Chapter 24 of the *Digital Library of Mathematical Functions* (2012). In this case the inner product is

$$\langle f, g\rangle = \sum_{r=0}^{\alpha-1}\left(\int_0^1 f^{(r)}(y)\,\mathrm{d}y\right)\left(\int_0^1 g^{(r)}(y)\,\mathrm{d}y\right) + \int_0^1 f^{(\alpha)}(y)g^{(\alpha)}(y)\,\mathrm{d}y,$$

and the reproducing kernel is

$$K_\alpha(x,y) = \sum_{r=0}^{\alpha}\frac{B_r(x)}{r!}\frac{B_r(y)}{r!} - (-1)^\alpha\frac{B_{2\alpha}(|x-y|)}{(2\alpha)!}.$$

The corresponding RKHS is called the *unanchored Sobolev space of smoothness* α.

3.5. Geometric discrepancy

We now return to the local discrepancy function

$$\Delta_P(x) = \frac{1}{n}\sum_{i=0}^{n-1} 1_{[0,x]}(t_i) - \int_0^1 1_{[0,x]}(y)\,\mathrm{d}y,$$

which we encountered already in (3.3), and which we also obtained as the derivative ξ' of the representer ξ, where ξ is given by (3.11) with K as in (3.7).

The local discrepancy function has a geometric interpretation: since

$$\sum_{i=0}^{n-1} 1_{[0,x]}(t_i)$$

is the number of points of P lying in the interval $[0,x]$, therefore

$$\frac{1}{n}\sum_{i=0}^{n-1} 1_{[0,x]}(t_i)$$

is the proportion of points of P lying in the interval $[0,x]$; and the local discrepancy is the departure of this proportion from the 'ideal' proportion, which is the length of the interval. By taking the L_p norm of Δ_P we obtain the L_p *discrepancy* of the point set P, given by

$$\|\Delta_P\|_p := \left(\int_0^1 |\Delta_P(x)|^p\,\mathrm{d}x\right)^{1/p}$$

for $1 \le p < \infty$, with the obvious modifications for $p = \infty$. The L_p discrepancy can therefore be understood as a measure for how uniformly the point set P is distributed, that is, it measures the discrepancy between the empirical distribution of the point set P and the uniform distribution.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

We consider now a generalization to dimensions $s > 1$. In one dimension, the local discrepancy function can be derived as the derivative of the representer ξ based on the reproducing kernel $K(x, y) = 2 - \max(x, y)$. As before we consider the product kernel $K_s(\boldsymbol{x}, \boldsymbol{y}) = \prod_{j=1}^{s} K(x_j, y_j)$. The representer of the error is given by

$$\begin{aligned}\xi(\boldsymbol{x}) &= \int_{[0,1]^s} K_s(\boldsymbol{x}, \boldsymbol{y})\,\mathrm{d}\boldsymbol{y} - \frac{1}{n}\sum_{i=0}^{n-1} K_s(\boldsymbol{x}, \boldsymbol{t}_i) \\ &= \prod_{j=1}^{s}\left(\frac{3 - x_j^2}{2}\right) - \frac{1}{n}\sum_{i=0}^{n-1}\prod_{j=1}^{s}(2 - \max(x_j, t_{i,j})).\end{aligned}$$

The mixed first partial derivatives are then given by

$$\frac{\partial^{|\mathfrak{u}|}\xi}{\partial \boldsymbol{x}_{\mathfrak{u}}}(\boldsymbol{x}_{\mathfrak{u}}; \mathbf{1}) = (-1)^{|\mathfrak{u}|}\left(\prod_{j\in\mathfrak{u}} x_j - \frac{1}{n}\sum_{i=0}^{n-1} 1_{[\mathbf{0}_{\mathfrak{u}}, \boldsymbol{x}_{\mathfrak{u}}]}(\boldsymbol{t}_{i,\mathfrak{u}})\right),$$

where

$$1_{[\mathbf{0}_{\mathfrak{u}}, \boldsymbol{x}_{\mathfrak{u}}]}(\boldsymbol{t}_{i,\mathfrak{u}}) = \prod_{j\in\mathfrak{u}} 1_{[0,x_j]}(t_{i,j}).$$

We define the local discrepancy function Δ_P in s dimensions by

$$\Delta_P(\boldsymbol{x}) := \frac{1}{n}\sum_{i=0}^{n-1} 1_{[\mathbf{0},\boldsymbol{x}]}(\boldsymbol{t}_i) - \prod_{j=1}^{s} x_j.$$

Then

$$\frac{\partial^{|\mathfrak{u}|}\xi}{\partial \boldsymbol{x}_{\mathfrak{u}}}(\boldsymbol{x}_{\mathfrak{u}}; \mathbf{1}) = (-1)^{|\mathfrak{u}|+1}\Delta_P(\boldsymbol{x}_{\mathfrak{u}}; \mathbf{1}).$$

Analogously to the one-dimensional case, we can show that

$$\frac{1}{n}\sum_{i=0}^{n-1} f(\boldsymbol{t}_i) - \int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \sum_{\mathfrak{u}\subseteq\{1:s\}} (-1)^{|\mathfrak{u}|}\int_{[0,1]^{|\mathfrak{u}|}} \frac{\partial^{|\mathfrak{u}|} f}{\partial \boldsymbol{x}_{\mathfrak{u}}}(\boldsymbol{x}_{\mathfrak{u}}; \mathbf{1})\Delta_P(\boldsymbol{x}_{\mathfrak{u}}; \mathbf{1})\,\mathrm{d}\boldsymbol{x}_{\mathfrak{u}},$$

where the $\mathfrak{u} = \emptyset$ term is actually zero since $\Delta_P(\mathbf{1}) = 0$. This equality is called by different people the *Hlawka identity* or the *Zaremba identity*.

By applying Hölder's inequality for integrals and sums, we then obtain the following inequality.

Theorem 3.9 (Koksma–Hlawka inequality). We have

$$\left|\frac{1}{n}\sum_{i=0}^{n-1} f(\boldsymbol{t}_i) - \int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\right| \le \|\Delta_P\|_{p,p'}\|f\|_{q,q'}, \tag{3.14}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

where $1 \le p, p', q, q' \le \infty$, $\frac{1}{p} + \frac{1}{q} = 1$, $\frac{1}{p'} + \frac{1}{q'} = 1$, and

$$\|\Delta_P\|_{p,p'} = \left(\sum_{\mathfrak{u} \subseteq \{1:s\}} \left(\int_{[0,1]^{|\mathfrak{u}|}} \left| \Delta_P(\boldsymbol{x}_\mathfrak{u}; \mathbf{1}) \right|^p \mathrm{d}\boldsymbol{x}_\mathfrak{u} \right)^{p'/p} \right)^{1/p'}$$

and

$$\|f\|_{q,q'} = \left(\sum_{\mathfrak{u} \subseteq \{1:s\}} \left(\int_{[0,1]^{|\mathfrak{u}|}} \left| \frac{\partial^{|\mathfrak{u}|} f}{\partial \boldsymbol{x}_\mathfrak{u}}(\boldsymbol{x}_\mathfrak{u}; \mathbf{1}) \right|^q \mathrm{d}\boldsymbol{x}_\mathfrak{u} \right)^{q'/q} \right)^{1/q'},$$

with the obvious modifications if one or more of p, p', q, q' are infinite.

In the theorem the norm $\|f\|_{q,q'}$ can obviously be replaced by the corresponding seminorm $|f|_{q,q'}$, defined in exactly the same way but with the $\mathfrak{u} = \emptyset$ term omitted. (This term can be omitted because the QMC rule is exact for constants.)

The inequality (3.14) says that the integration error is bounded by the product of the norm $\|f\|_{q,q'}$ or seminorm $|f|_{q,q'}$ of the integrand and the discrepancy $\|\Delta_P\|_{p,p'}$ of the cubature points. It is often called a Koksma–Hlawka inequality, especially for the case $p = p' = \infty$. If the integrand is given, and cannot be changed, the inequality motivates the search for cubature points with small discrepancy. It also connects QMC with the field of discrepancy theory, since it shows that point sets with low discrepancy are attractive to use as QMC integration points.

The classical Koksma–Hlawka inequality states that

$$\left| \frac{1}{n} \sum_{i=0}^{n-1} f(\boldsymbol{t}_i) - \int_{[0,1]^s} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \right| \le D_n^*(P) V(f), \tag{3.15}$$

where $D_n^*(P)$ is the *star discrepancy*

$$D_n^*(P) := \sup_{\boldsymbol{x} \in [0,1]^s} |\Delta_P(\boldsymbol{x})|, \tag{3.16}$$

and $V(f)$ is the *variation of f in the sense of Hardy and Krause*. If f has continuous mixed first partial derivatives, the variation $V(f)$ coincides with

$$|f|_{1,1} = \sum_{\emptyset \ne \mathfrak{u} \subseteq \{1:s\}} \int_{[0,1]^{|\mathfrak{u}|}} \left| \frac{\partial^{|\mathfrak{u}|} f}{\partial \boldsymbol{x}_\mathfrak{u}}(\boldsymbol{x}_\mathfrak{u}; \mathbf{1}) \right| \mathrm{d}\boldsymbol{x}_\mathfrak{u}.$$

3.6. Notes

The classical reference for reproducing kernels is Aronszajn (1950). See also Thomas-Agnan (1996) and Wahba (1990). In the context of QMC, reproducing kernels were introduced by Hickernell (1996*a*); see also Hickernell (1998*a*). For another elementary introduction to RKHSs in the context of QMC integration see Dick and Pillichshammer (2010, Chapter 2).

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Theorem 3.5 can be found in Hickernell (1998*a*). Mercer's theorem implies that each reproducing kernel has an expansion in terms of its eigenfunctions

$$K(\boldsymbol{x}, \boldsymbol{y}) = \sum_{r=0}^{\infty} \lambda_r \psi_r(\boldsymbol{x}) \psi_r(\boldsymbol{y}),$$

where $\lambda_r > 0$ are the eigenvalues and ψ_r are the eigenfunctions of the integral operator on $[0,1]^s$ with kernel K. For many reproducing kernels considered in QMC theory these eigenfunctions are explicitly known: see Dick, Nuyens and Pillichshammer (2013), Wasilkowski and Woźniakowski (1999) and Werschulz and Woźniakowski (2009).

The one-dimensional form of (3.15) is due to Koksma (1942/43) and the higher-dimensional result is due to Hlawka (1961).

Geometric discrepancy was studied long before QMC integration. The classical reference to uniform distribution modulo 1 is Weyl (1916). Further monographs dealing with discrepancy theory and various aspects of it are due to Kuipers and Niederreiter (1974), Niederreiter (1992*a*), Drmota and Tichy (1997), Matoušek (1999), Chazelle (2000) and Dick and Pillichshammer (2010).

Novak and Woźniakowski (2009) studied whether general L_2 discrepancies (in Euclidean spaces) are related to the worst-case error of multivariate integration on an RKHS. In particular, they provided a concrete formula for the corresponding reproducing kernel for arbitrary L_2 discrepancies. Gnewuch (2012*a*) extended these results to weighted L_2 discrepancies and weighted RKHS. Here the underlying domain can be more general as in Novak and Woźniakowski (2009). In particular, Gnewuch (2012*a*) covers the relation between infinite-dimensional integration on weighted Hilbert spaces and the corresponding weighted (limiting) L_2 discrepancies.

4. Weighted spaces and tractability

4.1. Meeting the high-dimensional challenge

High-dimensional problems, as noted already in the Introduction, can be very hard. Yet some problems have caused surprise in the opposite direction by turning out to be easier than might have been expected. Especially influential in this respect were certain 360-dimensional calculations carried out in the mid-1990s by Paskov and Traub (1995) at Columbia University. That paper described a problem coming directly from Wall Street, on the valuation of a class of financial derivatives known as mortgage-backed obligations. While the details of the model or the calculation are not important here, broadly the problem was to evaluate a parcel of mortgages held by a bank, where each month borrowers make individual choices as to whether or not to repay the mortgage, and are assumed to make this choice according

to some probability distribution. And of course the value of a mortgage held by the bank is affected by whether or not the loan is repaid. Because there are 360 months in the 30-year period of the loan, the problem is to evaluate a 360-dimensional expected value. The standard approach to such a problem is to model the process month by month by a Monte Carlo method. Paskov and Traub (1995), in contrast, treated the problem as one of 360-dimensional integration. To the surprise of most observers, they were able to obtain satisfactory results with a QMC method, at much smaller computational cost than with a Monte Carlo method.

Following the success of those experiments, the idea began to emerge that perhaps problems such as the one described in the previous paragraph are in some hidden sense not really as difficult as might be expected from their nominal high-dimensionality. In this spirit Caflisch, Morokoff and Owen (1997) defined two notions of 'effective dimension' (namely 'truncation dimension' and 'superposition dimension'), with the idea that either or both might be much smaller than the nominal dimension. Sloan and Woźniakowski (1998) sought to build this notion into the mathematics, by introducing function spaces containing 'weights'. It is fair to say that in some form or another 'weighted spaces' have become a standard part of the QMC framework. We will define such spaces formally in the next subsection, but first we describe the underlying idea, and indicate why it has become so popular.

Sloan and Woźniakowski speculated that perhaps the reason that QMC calculations such as those in Paskov and Traub (1995) were successful is that some coordinate directions are more important than others, in the sense of being in some way more difficult. Assuming then that the coordinate directions are ordered in order of difficulty, they sought to quantify this decreasing importance by associating with each coordinate direction a positive number γ_j, where

$$\gamma_1 \geq \gamma_2 \geq \gamma_3 \cdots > 0.$$

By introducing a function space incorporating these weights, they were able to obtain a rather precise result, namely that the worst-case error (defined already in Definition 3.4) is bounded independently of dimension if and only if

$$\sum_{j=0}^{\infty} \gamma_j < \infty.$$

This means that a classical choice of weights (with all weights equal) certainly fails to have worst-case errors bounded independently of the dimension. So too, though only marginally, do weights $\gamma_j = 1/j$. On the other hand weights of the form $\gamma_j = 1/j^a$ lead to uniformly bounded worst-case errors if (and only if) $a > 1$.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

The weights described in the preceding paragraph are now called 'product weights', for reasons that will soon become clear. Since that time many other forms of weights have been studied (we shall meet 'general weights', 'finite-order' weights, 'order-dependent' weights, and the most recent addition, 'POD' (for 'product and order-dependent') weights. The driving motivation for this flowering of possibilities has been the desire to describe in a more precise way the influence of particular combinations of the variables. At the (unrealistic) extreme, the greatest freedom attaches to 'general weights', which allow a different weight $\gamma_{\mathfrak{u}}$ for each of the 2^s subsets $\mathfrak{u} \subseteq \{1:s\}$.

For all these choices of weights the question of 'tractability' arises: loosely, we ask under what conditions on the weights are the worst-case errors uniformly bounded in dimension; or if they are not bounded, then at worst grow only slowly with dimension. We shall discuss tractability in Section 4.5.

4.2. Weighted reproducing kernel Hilbert spaces: anchored Sobolev spaces

In this subsection we first introduce a weighted RKHS in its simplest product form. But we then quickly develop weighted spaces of a more general form.

Arguing as in Example 3.3, we build our first weighted s-variable space out of single-variable weighted spaces with simple kernels. Our one-dimensional space $H_{1,\gamma}$ is almost the same as before (in particular, it is always the space of absolutely continuous functions f defined on $[0,1]$ whose first derivatives are square-integrable), except that we now associate with the space a weight $\gamma > 0$, and in addition we allow the 'anchor' value to be any number $c \in [0,1]$, instead of just 1 as before. Our building block is now the one-dimensional kernel

$$K_{1,\gamma}(x,y) = 1 + \gamma\,\eta(x,y),$$

where

$$\eta(x,y) = \begin{cases} \min(x,y) - c & \text{if } x,y > c, \\ c - \max(x,y) & \text{if } x,y < c, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

If $c = 1$ and $\gamma = 1$ it is easily seen that this reduces to the reproducing kernel (3.7). For general c and γ it is easy to verify, using no more than the fundamental theorem of calculus, that $K_{1,\gamma}(x,y)$ is the reproducing kernel associated with the inner product

$$\langle f, g\rangle_{1,\gamma} = f(c)g(c) + \frac{1}{\gamma}\int_0^1 f'(x)\,g'(x)\,\mathrm{d}x,$$

which is a generalization of (3.6). The corresponding norm is

$$\|f\|_{1,\gamma} = \langle f, f\rangle_{1,\gamma}^{1/2}.$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

The s-dimensional tensor product RKHS corresponding to the above one-dimensional space is

$$H_{s,\boldsymbol{\gamma}} := H_{1,\gamma_1} \otimes H_{1,\gamma_2} \otimes \cdots \otimes H_{1,\gamma_s},$$

if we allow a different weight γ_j for each component x_j. It has the reproducing kernel

$$K_{s,\boldsymbol{\gamma}}(\boldsymbol{x},\boldsymbol{y}) = \prod_{j=1}^{s} (1+\gamma_j \eta(x_j,y_j)).$$

The identity

$$\prod_{j=1}^{s}(1+a_j) = \sum_{\mathfrak{u}\subseteq\{1:s\}} \prod_{j\in\mathfrak{u}} a_j = 1 + \sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:s\}} \prod_{j\in\mathfrak{u}} a_j \tag{4.2}$$

can now be used to rewrite the product as a sum over all subsets of $\{1:s\}$,

$$K_{s,\boldsymbol{\gamma}}(\boldsymbol{x},\boldsymbol{y}) = \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \eta(x_j,y_j), \tag{4.3}$$

where for this *product weight* case we have

$$\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_j,$$

with the convention that the empty product has the value 1. The reason for the 'product' tag for these weights now becomes clear: the weight associated with the subset $\mathfrak{u}$ of the variables is the product of the weights for the individual components. One implication of this is that *the scaling of product weights is crucially important*. If, for example, all the weights γ_j are halved then the weights for the subset $\mathfrak{u}=\{1,3\}$ with two elements are reduced by a factor of 4, and those for the subset $\mathfrak{u}=\{2,4,5\}$ by a factor of 8.

Because of the limited flexibility of product weights, other choices of weights have been considered. In principle the formula (4.3) allows a different weight $\gamma_{\mathfrak{u}}$ to be chosen for each subset $\mathfrak{u}$ of the variables, but the cost of even one evaluation of the reproducing kernel would then be prohibitive when s is large. This is the *general weight* case (Dick, Sloan, Wang and Woźniakowski 2006), where we take $\gamma_\emptyset \equiv 1$. There is much interest (as well as some unease (Sloan 2007)) in *finite-order weights* (Sloan, Wang and Woźniakowski 2004, Wasilkowski and Woźniakowski 2004), in which $\gamma_{\mathfrak{u}} = 0$ for all $|\mathfrak{u}|$ greater than some number q^*. *Order-dependent weights*, introduced by Dick *et al.* (2006), take the form

$$\gamma_{\mathfrak{u}} = \Gamma_{|\mathfrak{u}|}$$

for some non-negative numbers $\Gamma_1, \Gamma_2, \ldots$. The most recent addition to the menagerie of weights are the *POD weights* (*i.e.*, *product and order-dependent weights*), in which two sequences $\gamma_1, \gamma_2, \ldots$ and $\Gamma_1, \Gamma_2, \ldots$ of non-negative

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

numbers are defined, and we take

$$\gamma_{\mathfrak{u}} = \Gamma_{|\mathfrak{u}|} \prod_{j\in\mathfrak{u}} \gamma_j. \tag{4.4}$$

Weights of the POD form were found to arise naturally in a recent study of PDEs with random coefficients (Kuo *et al.* 2012).

For general weights $\gamma_{\mathfrak{u}}$ the inner product in our anchored space is

$$\langle f, g\rangle_{s,\gamma} = \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}^{-1} \int_{[0,1]^{|\mathfrak{u}|}} \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{x}_{\mathfrak{u}}} f(\boldsymbol{x}_{\mathfrak{u}}; \boldsymbol{c})\, \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{x}_{\mathfrak{u}}} g(\boldsymbol{x}_{\mathfrak{u}}; \boldsymbol{c})\, \mathrm{d}\boldsymbol{x}_{\mathfrak{u}},$$

where the notation follows that in (3.8), and the term labelled by $\mathfrak{u}$ is to be omitted for $\mathfrak{u} = \emptyset$.

The worst-case errors for spaces with reproducing kernel of the form (4.3) and η given by (4.1) can be found from (3.9). To make use of that expression one needs the single and double integrals of the kernel. For completeness we give the necessary integrals here, as well as the 'diagonal' integral needed for the QMC mean (3.13):

$$\begin{aligned}
\int_{[0,1]^s} K_{s,\boldsymbol{\gamma}}(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} &= \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \big(\alpha(x_j)+\beta\big),\\
\int_{[0,1]^s}\int_{[0,1]^s} K_{s,\boldsymbol{\gamma}}(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y} &= \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}\, \beta^{|\mathfrak{u}|},\\
\int_{[0,1]^s} K_{s,\boldsymbol{\gamma}}(\boldsymbol{x},\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} &= \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \big(\beta+\tfrac{1}{6}\big)^{|\mathfrak{u}|},
\end{aligned} \tag{4.5}$$

where

$$\alpha(x) := \max(x,c) - \tfrac{x^2}{2} - \tfrac{c^2}{2} - \tfrac{1}{3} \quad\text{and}\quad \beta := c^2 - c + \tfrac{1}{3}. \tag{4.6}$$

When the weights are of the product form, these three integrals simplify (by using the identity (4.2)) to

$$\prod_{j=1}^{s}\big(1+\gamma_j\big(\alpha(x_j)+\beta\big)\big), \quad \prod_{j=1}^{s}\big(1+\gamma_j\beta\big), \quad \prod_{j=1}^{s}\big(1+\gamma_j\big(\beta+\tfrac{1}{6}\big)\big),$$

respectively.

For example, with product weights and $c = 1$, we have from (3.9) and (3.10) that the worst-case error and initial error are given by

$$\begin{aligned}
e_{n,s}^2(P; H(K_{s,\boldsymbol{\gamma}})) = \prod_{j=1}^{s}\left(1+\frac{\gamma_j}{3}\right) - \frac{2}{n}\sum_{i=0}^{n-1}\prod_{j=1}^{s}\left(1+\frac{\gamma_j}{2}(1-t_{i,j}^2)\right)\\
+ \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{k=0}^{n-1}\prod_{j=1}^{s}\big(1+\gamma_j\big[1-\max(t_{i,j},t_{k,j})\big]\big),
\end{aligned}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

which reduces to (3.12) if $\gamma_j = 1$ for all j, and

$$e^2_{0,s}(H(K_{s,\boldsymbol{\gamma}})) = \prod_{j=1}^{s}\left(1+\frac{\gamma_j}{3}\right).$$

Moreover, we have from (3.13) that the QMC mean satisfies

$$E^2_{n,s}(H(K_{s,\boldsymbol{\gamma}})) = \frac{1}{n}\left(\prod_{j=1}^{s}\left(1+\frac{\gamma_j}{2}\right) - \prod_{j=1}^{s}\left(1+\frac{\gamma_j}{3}\right)\right).$$

We remark that it is also possible to use a different anchor value c_j for each coordinate direction x_j, thus leading to different η_j, α_j, β_j for each index j. All results can be trivially generalized to that case.

4.3. Unanchored Sobolev spaces

The reproducing kernels considered so far in this section have been of the so-called 'anchored' variety, in which there is a special number $c \in [0,1]$, called the 'anchor', at which the components of $\boldsymbol{x}$ that are not active (*i.e.*, that are not in the active set $\mathfrak{u}$) are fixed. However, there are other RKHSs with interesting properties (Sloan and Woźniakowski 2002). In particular, the so-called 'unanchored' spaces, in which the inactive variables are integrated over rather than fixed, have some advantages. We give the necessary formulas here; all can be checked by similar arguments to those given previously for the anchored spaces (or see the reference above).

In the one-dimensional weighted case the inner product is now (*cf.* Example 3.8 for $\alpha = 1$)

$$\langle f, g\rangle_{1,\gamma} = \left(\int_0^1 f(x)\,\mathrm{d}\boldsymbol{x}\right)\left(\int_0^1 g(x)\,\mathrm{d}\boldsymbol{x}\right) + \frac{1}{\gamma}\int_0^1 f'(x)\,g'(x)\,\mathrm{d}x,$$

and the corresponding norm is

$$\|f\|_{1,\gamma} = \langle f, f\rangle^{1/2}_{1,\gamma}.$$

The corresponding reproducing kernel is, as before,

$$K_{1,\gamma}(x,y) = 1 + \gamma\,\eta(x,y),$$

but now η is given by

$$\eta(x,y) = \tfrac{1}{2}B_2(|x-y|) + \left(x-\tfrac{1}{2}\right)\left(y-\tfrac{1}{2}\right), \tag{4.7}$$

where $B_2(x) := x^2 - x + \frac{1}{6}$ is the Bernoulli polynomial of degree 2, which has the useful property

$$\int_0^1 B_2(|x-y|)\,\mathrm{d}y = \int_0^1 B_2(y)\,\mathrm{d}y = 0.$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

The corresponding kernel for the s-dimensional general weight case is again given by (4.3), but now with η given by (4.7). The inner product for the s-dimensional general weight unanchored space is

$$\langle f, g\rangle_{s,\gamma} = \sum_{\mathrm{u}\subseteq\{1:s\}} \gamma_{\mathrm{u}}^{-1} \int_{[0,1]^{|\mathrm{u}|}} \left(\int_{[0,1]^{s-|\mathrm{u}|}} \frac{\partial^{|\mathrm{u}|}}{\partial \boldsymbol{x}_{\mathrm{u}}} f(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}_{-\mathrm{u}} \right) \times \left(\int_{[0,1]^{s-|\mathrm{u}|}} \frac{\partial^{|\mathrm{u}|}}{\partial \boldsymbol{x}_{\mathrm{u}}} g(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}_{-\mathrm{u}} \right) \mathrm{d}\boldsymbol{x}_{\mathrm{u}},$$

where $\boldsymbol{x}_{-\mathrm{u}}$ stands for all the components of the s-dimensional vector $\boldsymbol{x}$ that are not included in $\boldsymbol{x}_{\mathrm{u}}$.

The range of choices for the weights is exactly the same as for the anchored case.

The worst-case error is again given by (3.9). The necessary single and double integrals of the kernel, along with the diagonal integral needed for the QMC mean, now take the simpler forms

$$\int_{[0,1]^s} K_{s,\boldsymbol{\gamma}}(\boldsymbol{x},\boldsymbol{y})\, \mathrm{d}\boldsymbol{y} = 1,$$

$$\int_{[0,1]^s} \int_{[0,1]^s} K_{s,\boldsymbol{\gamma}}(\boldsymbol{x},\boldsymbol{y})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y} = 1,$$

$$\int_{[0,1]^s} K_{s,\boldsymbol{\gamma}}(\boldsymbol{x},\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \sum_{\mathrm{u}\subseteq\{1:s\}} \gamma_{\mathrm{u}} \left(\tfrac{1}{6}\right)^{|\mathrm{u}|},$$

and in the case of product weights the third integral simplifies to

$$\prod_{j=1}^{s} \left(1 + \frac{\gamma_j}{6}\right).$$

To avoid having to treat the anchored and unanchored spaces separately, we can write $\eta(x,y)$ from (4.1) and (4.7) in the common form

$$\eta(x,y) = \tfrac{1}{2} B_2(|x-y|) + (x-\tfrac{1}{2})(y-\tfrac{1}{2}) + \alpha(x) + \alpha(y) + \beta, \tag{4.8}$$

where $\alpha(\cdot)$ and β are defined by (4.6) for the anchored case, and $\alpha \equiv 0$ and $\beta = 0$ for the unanchored case.

4.4. Why weighted spaces are interesting

We have defined different kinds of weighted spaces for functions defined on the s-dimensional cube, but we have not so far said why weighted spaces are interesting. The answer is that *the worst-case errors in any of our weighted reproducing kernel Hilbert spaces are bounded independently of the dimension s if (and only if) the weights decay in a suitable way.*

For product weights, the following result, first obtained by Sloan and Woźniakowski (1998) for the anchored case $c = 1$, is particularly striking.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Theorem 4.1. For product weights, and for the weighted anchored or unanchored Sobolev space of Sections 4.2 and 4.3, there exist point sets $P_n \subset [0,1]^s$ for $n = 1, 2, \ldots$ such that the worst-case error $e_{n,s}(P_n; H(K_{s,\boldsymbol{\gamma}}))$ is bounded independently of s if and only if

$$\sum_{j=1}^{\infty} \gamma_j < \infty. \tag{4.9}$$

There are two parts to this theorem: one part is an existence result, stating that 'good' point sets exist if (4.9) holds, without saying how to find them; and the other part is a negative result, saying that if the assumption (4.9) does not hold then, no matter how the points are chosen, the worst-case error is unbounded as $s \to \infty$. We shall address each result separately in the following two lemmas.

Lemma 4.2. For product weights, and for the weighted anchored or unanchored Sobolev space of Sections 4.2 and 4.3, there exist point sets $P_n \subset [0,1]^s$ for $n = 1, 2, \ldots$ such that the worst-case error $e_{n,s}(P_n; H(K_{\gamma,s}))$ satisfies

$$e_{n,s}(P_n; H(K_{s,\boldsymbol{\gamma}})) \le \frac{1}{\sqrt{n}} \exp\left(B \sum_{j=1}^{s} \gamma_j\right) \le \frac{1}{\sqrt{n}} \exp\left(B \sum_{j=1}^{\infty} \gamma_j\right),$$

where $B = (c^2 - c + 1/2)/2$ for the anchored space with anchor value $c \in [0,1]$, and $B = 1/12$ for the unanchored space. (In particular, $B = 1/4$ for the anchor $c = 1$ and $B = 1/8$ for the midpoint anchor $c = 1/2$.)

Thus the worst-case error is bounded independently of s, and has the Monte Carlo rate of convergence, if the infinite sum in (4.9) converges.

Proof. We give the proof for the anchored space with $c = 1$. The other cases follow similarly. It follows from the *averaging argument* (there is always at least one choice as good as the average) that there exists a QMC point set P_n for which the worst-case error is less than or equal to the QMC mean given by (3.13). We produce an upper bound on the latter by omitting the negative term, and then use the third equation in (4.5) to evaluate the integral, obtaining

$$\begin{aligned} e^2_{n,s}(P_n; H(K_{s,\boldsymbol{\gamma}})) \le E^2_{n,s}(H(K_{s,\boldsymbol{\gamma}})) &\le \frac{1}{n} \prod_{j=1}^{s} \left(1 + \frac{\gamma_j}{2}\right) \\ &= \frac{1}{n} \exp\left(\sum_{j=1}^{s} \log\left(1 + \frac{\gamma_j}{2}\right)\right) \le \frac{1}{n} \exp\left(\frac{1}{2} \sum_{j=1}^{s} \gamma_j\right), \end{aligned}$$

where in the last step we used the property that $\log(1+x) \le x$ for all $x > 0$. The rest of the claim in the lemma follows immediately. □

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

The 'only if' part of Theorem 4.1 for the anchored case comes from the following result established by Sloan and Woźniakowski (1998). The argument there is relatively elementary, starting from (3.9) with the off-diagonal terms of the last term dropped (which is justifiable by the non-negativity assumption).

Lemma 4.3. If the reproducing kernel K_s of some Hilbert space $H(K_s)$ is non-negative, then for any point set $P_n \subset [0,1]^s$ we have

$$e^2_{n,s}(P_n; H(K_s)) \;\geq\; e^2_{0,s}(H(K_s))(1 - n\,\kappa_s^2),$$

where

$$\kappa_s := \left(\sup_{\boldsymbol{x} \in [0,1]^s} \frac{a_s(\boldsymbol{x})}{\sqrt{K_s(\boldsymbol{x},\boldsymbol{x})}} \right) \frac{1}{\|a_s\|_s},$$

and

$$a_s(\boldsymbol{x}) := \int_{[0,1]^s} K_s(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}.$$

Hence $e_{n,s}(P_n, H(K_s)) \leq \varepsilon e_{0,s}(H(K_s))$ can happen only if

$$n \geq \frac{1-\varepsilon^2}{\kappa_s^2}.$$

This lemma applies to the anchored space of Section 4.2 because the kernel $K_{s,\boldsymbol{\gamma}}$ is in this case manifestly positive. The proof of the second part of Theorem 4.1 then follows in this case by showing that $\kappa_s \to 0$ if the sum in (4.9) diverges. For the unanchored case a different proof is needed, because the kernel $K_{s,\boldsymbol{\gamma}}$ in that case is not necessarily positive. The result for that case is proved in Sloan and Woźniakowski (2002, Theorem 1), but in any case is superseded by the stronger result stated in Theorem 4.5 below.

In the case of our original anchored but unweighted Sobolev space discussed in Example 3.3, it can be shown (Sloan and Woźniakowski 1998) that $\kappa_s \approx 1.055^{-s}$, thus in this case n must be exponentially large to reduce the initial error by some fixed percentage, say 50%.

The upper bound on the worst-case error in Lemma 4.2 is independent of s under the condition (4.9), but the apparent rate of convergence in that lemma is just the Monte Carlo rate of $O(n^{-1/2})$. Fortunately, as we shall see in Sections 5 and 6, a convergence rate arbitrarily close to $O(n^{-1})$ can be attained if the (product) weights satisfy the stronger condition

$$\sum_{j=1}^{\infty} \gamma_j^{1/2} < \infty.$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

4.5. Tractability of multivariate integration

In this subsection we frame the earlier discussion in terms of the valuable notion of tractability. Loosely speaking, the tractability of a problem relates to the question of how quickly the difficulty of a problem increases as the dimension increases. Note that it relates to the difficulty of the *problem*, not to the cost of any particular algorithm.

The standard reference in this area is the recent three-volume work by Novak and Woźniakowski (2008, 2010, 2012). In the present brief discussion we limit ourselves to a very small part of the subject, touching only parts that are relevant to our concerns with multivariate integration over the unit cube.

Thus our problem is that of evaluating the s-dimensional integral $I_s(f)$ defined in (1.1). We will assume that f belongs to a Banach space F_s of real-valued functions defined on $[0,1]^s$. The space F_s could be one of our reproducing kernel spaces H_s, but could also (as in the next subsection) be a non-Hilbert space. The setting is that we are allowed to approximate $I_s(f)$ by any deterministic algorithm $A_{n,s}(f)$ that uses at most n function values of F_s. (In the language of information-based complexity (Traub, Wasilkowski and Woźniakowski 1988), the algorithm is restricted to 'standard information'.) Thus $A_{n,s}$ could be a QMC integration rule, but could also be an integration rule that uses unequal weights, including even negative weights, and $A_{n,s}$ is even allowed to be a nonlinear combination of $f(\boldsymbol{t}_0), \dots, f(\boldsymbol{t}_{n-1})$.

For a given $s \geq 1$ and a given $\varepsilon \in (0,1]$ we define the nth *minimal number* for this problem by

$$n(\varepsilon, s) := \min\left\{ n \geq 1 : \inf_{A_{n,s}} \sup_{\|f\|_{F_s} \leq 1} |I_s(f) - A_{n,s}(f)| \leq \varepsilon \right\},$$

where the infimum is over all algorithms $A_{n,s}$ that use no more than n function values of f (and no other information about f).

Definition 4.4 (tractability). The integration problem is said to be *polynomially tractable* if there is some $C > 0$ and some $p, q > 0$ and $q \geq 0$ such that

$$n(\varepsilon, s) \leq C \left(\frac{1}{\varepsilon}\right)^p s^q$$

for all $s \in \mathbb{N}$ and $\varepsilon \in (0,1]$. It is *strongly polynomially tractable* if this inequality holds with $q = 0$, that is, if

$$n(\varepsilon, s) \leq C \left(\frac{1}{\varepsilon}\right)^p.$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Many other variants of tractability are defined in the cited works, but these two will suffice for our present purposes.

The following theorem mimics Theorem 4.1 in the preceding subsection, but there is an important difference: whereas for the 'only if' part of the theorem we previously allowed only QMC integration rules, here *all* algorithms $A_{n,s}$ are allowed, making the statement considerably stronger.

Theorem 4.5 (strong polynomial tractability). For product weights, and for the weighted anchored or unanchored Sobolev space of Sections 4.2 and 4.3, the integration problem is strongly polynomially tractable if and only if

$$\sum_{j=1}^{\infty} \gamma_j < \infty.$$

A necessary and sufficient condition for polynomial tractability is as follows.

Theorem 4.6 (polynomial tractability). For product weights, and for the weighted anchored or unanchored Sobolev space of Sections 4.2 and 4.3, the integration problem is polynomially tractable if and only if

$$\limsup_{s\to\infty} \frac{\sum_{j=1}^{s} \gamma_j}{\log(s+1)} < \infty.$$

The preceding theorem may give the impression that non-trivial weights are needed in order to have (polynomial) tractability, but the remarkable result discussed in the next subsection shows that, while this is true for the Hilbert spaces we have met so far, it is not true for the classical L_1 version of the anchored spaces.

But the Hilbert spaces retain one great advantage, namely that the worst-case error for a given QMC rule is computable, something that turns out (as we shall see in the next two sections) to be helpful in construction. The star discrepancy (whether weighted or not), on the other hand, is notoriously difficult to compute: see Gnewuch, Srivastav and Winzen (2009).

In this subsection we have so far restricted ourselves to product weights. Tractability results for general weights were first considered by Sloan *et al.* (2004). For example, for general weights $\gamma_{\mathfrak{u}}$ and the unanchored Sobolev space it is shown that the integration problem is strongly polynomially tractable if

$$\sum_{|\mathfrak{u}|<\infty} \gamma_{\mathfrak{u}} \left(\tfrac{1}{6}\right)^{|\mathfrak{u}|} < \infty,$$

where now the sum is over all finite subsets of the natural numbers. For other tractability results for general weights and variants of the weights, we refer the reader to Novak and Woźniakowski (2010).

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

4.6. Tractability of star discrepancy

A surprising result established in Heinrich, Novak, Wasilkowski and Woźniakowski (2001) is that there exists a point set P in $[0,1]^s$ consisting of n points such that its star discrepancy, given by (3.16), satisfies

$$D_n^*(P) \le C\sqrt{\frac{s}{n}}, \tag{4.10}$$

for some constant $C > 0$ which is independent of n and s. Aistleitner (2011) showed that C can be chosen as 10, for instance. A lower bound by Hinrichs (2004) states that the infimum of $D_n^*(P)$ over all point sets P is bounded by

$$\inf_{P\subset[0,1]^s,|P|=n} D_n^*(P) \ge \min\left(c_0, c\frac{s}{n}\right),$$

where the constant $c > 0$ is independent of s and n and

$$c_0 = \frac{1}{32\mathrm{e}^2},$$

as stated in Gnewuch and Roşca (2009).

In the following we prove a slightly weaker version of (4.10). The proof is based on Hoeffding's inequality, a special case of which we describe in the following. Let $X_0, X_1, \dots, X_{n-1}$ be independent and identically distributed random variables with mean μ and $X_i \in [-1,1]$ almost surely for $0 \le i < n$. Let

$$X = \frac{1}{n}\sum_{i=0}^{n-1} X_i.$$

Then

$$\mathbb{P}(|X-\mu| \ge \delta) \le 2\,\mathrm{e}^{-\delta^2 n/2}, \quad \text{for all } \delta \ge 0.$$

We need an additional lemma.

Lemma 4.7. Let $P = \{\boldsymbol{t}_0, \boldsymbol{t}_1, \dots, \boldsymbol{t}_{n-1}\} \subset [0,1]^s$ be a point set consisting of n elements. Let $\delta > 0$ and $m = \lceil s/\delta \rceil$. Let Γ_m be the equidistant grid on $[0,1]^s$ with mesh size $1/m$ (and therefore cardinality $(m+1)^s$). Then

$$D_n^*(P) \le \max_{\boldsymbol{x}\in\Gamma_m} |\Delta_P(\boldsymbol{x})| + \delta.$$

Proof. Let $\eta > 0$. Then there is a vector $\boldsymbol{x}^* = (x_1^*, \dots, x_s^*) \in [0,1]^s$ such that

$$D_n^*(P) \le |\Delta_P(\boldsymbol{x}^*)| + \eta.$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Let $\boldsymbol{y}, \boldsymbol{z} \in \Gamma_m$ be such that $y_j \le x_j^* \le z_j$ and $z_j - y_j = m^{-1}$. Then we have

$$\begin{aligned}\prod_{j=1}^{s} z_j - \prod_{j=1}^{s} y_j &= \sum_{r=1}^{s}\left(\left(\prod_{j=r}^{s} z_j\right)\left(\prod_{j=1}^{r-1} y_j\right) - \left(\prod_{j=r+1}^{s} z_j\right)\left(\prod_{j=1}^{r} y_j\right)\right)\\ &= \sum_{r=1}^{s}\left(\prod_{j=r+1}^{s} z_j\right)\left(\prod_{j=1}^{r-1} y_j\right)(z_r - y_r) \le \frac{s}{m} \le \delta,\end{aligned}$$

where the empty product is set to 1. Thus we have

$$\begin{aligned}\prod_{j=1}^{s} z_j - \frac{1}{n}\sum_{i=0}^{n-1} 1_{[0,\boldsymbol{z})}(\boldsymbol{t}_i) - \delta &\le \prod_{j=1}^{s} x_j^* - \frac{1}{n}\sum_{i=0}^{n-1} 1_{[\boldsymbol{0},\boldsymbol{x}^*)}(\boldsymbol{t}_i)\\ &\le \prod_{j=1}^{s} y_j - \frac{1}{n}\sum_{i=0}^{n-1} 1_{[\boldsymbol{0},\boldsymbol{y})}(\boldsymbol{t}_i) + \delta,\end{aligned}$$

which implies the result. □

We are now ready to prove the following slightly weaker version of (4.10), which is Theorem 1 of Heinrich *et al.* (2001).

Theorem 4.8. For every $n, s \in \mathbb{N}$ there exists a point set P consisting of n points in $[0,1]^s$, whose star discrepancy satisfies

$$D_n^*(P) \le \frac{2\sqrt{2}}{\sqrt{n}}\sqrt{s\log\left(\left\lceil\frac{s\sqrt{n}}{\sqrt{4\log 2}}\right\rceil + 1\right) + \log 2}.$$

Proof. Let $\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{n-1}$ be independently and uniformly distributed random variables in $[0,1]^s$. Let $P = \{\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{n-1}\}$. Then for each $\boldsymbol{x} \in [0,1]^s$

$$\Delta_{\boldsymbol{t}_i}(\boldsymbol{x}) = 1_{[\boldsymbol{0},\boldsymbol{x})}(\boldsymbol{t}_i) - x_1 \cdots x_s$$

is a random variable with mean 0 and $|\Delta_{\boldsymbol{t}_i}(\boldsymbol{x})| \le 1$ for $0 \le i < n$. Further, we have

$$\Delta_P(\boldsymbol{x}) = \frac{1}{n}\sum_{i=0}^{n-1} \Delta_{\boldsymbol{t}_i}(\boldsymbol{x}).$$

Thus we can use Hoeffding's inequality, which implies that

$$\begin{aligned}\mathbb{P}\big(D_n^*(P)) \le 2\delta\big) &\ge \mathbb{P}\Big(\max_{\boldsymbol{x}\in\Gamma_m} |\Delta_P(\boldsymbol{x})| \le \delta\Big)\\ &\ge 1 - 2(m+1)^s \mathrm{e}^{-\delta^2 n/2}.\end{aligned}$$

We now choose the parameters such that $1 - 2(m+1)^s \mathrm{e}^{-\delta^2 n/2} > 0$, or equivalently

$$\log 2 + s\log(m+1) - \frac{\delta^2 n}{2} < 0.$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Since $m = \lceil s/\delta \rceil$, this holds for all $\delta > \delta_0 = \delta_0(n,s)$, where δ_0 satisfies

$$\delta_0^2 = 2n^{-1}(s\log(\lceil s/\delta_0 \rceil + 1) + \log 2).$$

This implies that

$$\frac{1}{\delta_0} \le \sqrt{\frac{n}{4\log 2}},$$

and therefore

$$\delta_0^2 \le 2n^{-1}\left(s\log\left(\left\lceil \frac{s\sqrt{n}}{\sqrt{4\log 2}} \right\rceil + 1\right) + \log 2\right).$$

Thus for any $\delta > \delta_0$ there exist points $\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{n-1}$ such that

$$D_n^*(\{\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{n-1}\}) \le 2\delta.$$

Hence there is a point set $P = \{\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{n-1}\}$ such that $D_n^*(P) \le 2\delta_0$. □

4.7. Notes

Weighted spaces were first introduced by Sloan and Woźniakowski (1998), and subsequently generalized by Sloan and Woźniakowski (2001, 2002), Dick *et al.* (2006) and Sloan *et al.* (2004). A non-technical introduction to the weighted space setting and lattice rules is given in Kuo and Sloan (2005). For the standard reference on tractability see the recent three-volume work by Novak and Woźniakowski (2008, 2010, 2012).

The 'only if' parts of Theorems 4.5 and 4.6 were proved by Novak and Woźniakowski (2001) for the anchored Sobolev space, and also for the more general class of 'decomposable' kernels. For the unanchored Sobolev space the second parts of the theorems were proved by Sloan and Woźniakowski (2002), using results from Hickernell and Woźniakowski (2001) and in turn Hickernell and Woźniakowski (2000). See also Wasilkowski and Woźniakowski (2004).

A classic paper on numerical integration and discrepancy is by Hickernell (1998*a*), who defined discrepancies which include lower-dimensional projections and introduced the reproducing kernel machinery for numerical integration. Sufficient conditions on the weights for which Sobol′, Halton, and Niederreiter sequences achieve strong tractability have been studied by Wang (2002, 2003). Tractability of so-called 'finite-order weights' has been shown by Sloan *et al.* (2004). Strong tractability of scrambled digital nets and sequences has been studied by Yue and Hickernell (2005, 2006). Tractability questions for the weighted star discrepancy have been considered by Hinrichs, Pillichshammer and Schmid (2008). A strategy for choosing the weights in finance applications has been considered in Wang and Sloan (2006, 2007). The concepts of 'effective dimension', 'truncation

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

dimension' and 'superposition dimension' have been introduced in Caflisch *et al.* (1997). The effective dimension of problems arising from financial applications have been studied by Wang and Fang (2003) and Wang and Sloan (2005) in the context of QMC.

The equidistant grid Γ_m used in Lemma 4.7 is a special δ-cover or (essentially) bracketing cover. The notion of bracketing is well established in empirical process theory (see van der Vart and Wellner 2009, for example), and to achieve better results than in Theorem 4.8 one needs better bracketing covers than Γ_m. In fact, the results from Aistleitner (2011), together with the results on probabilistic discrepancy estimates from Gnewuch and Roşca (2009), Aistleitner and Hofer (2012) and Gnewuch (2009), as well as theoretical arguments from Doerr, Gnewuch and Wahlström (2009) and Doerr, Gnewuch, Kritzer and Pillichshammer (2008) for constructing small discrepancy samples, all rely on the constructive bracketing covers and the induced upper bounds on bracketing numbers from Gnewuch (2008). Aistleitner (2011) and Aistleitner and Hofer (2012) also use Bernstein's inequality and the technique of dyadic chaining.

Attempts have been made to find constructions of point sets whose star discrepancy grows only polynomially with the dimension: see for instance Doerr, Gnewuch and Wahlström (2010) and Doerr *et al.* (2008). An implementation and numerical experiments have been reported in Doerr *et al.* (2009). However, the construction cost of these algorithms depends exponentially on the dimension. An essential problem in these algorithms is the estimation of the star discrepancy of a given point set, which is itself intractable as shown by Gnewuch *et al.* (2009). Algorithms for estimating the star discrepancy have been further investigated by Gnewuch, Wahlström and Winzen (2012).

Let $n(\varepsilon, d)$ be the smallest number necessary to reduce the d-dimensional L_2 discrepancy $\|\Delta_P\|_2$ by a factor of ε. The exponent of discrepancy is then the smallest number p such that $n(\varepsilon, d) \leq C\varepsilon^{-p}$ for all $d \geq 1$. It has been shown that $1.0669 \leq p \leq 1.41274$ for the classical L_2 discrepancy, where the lower bound is by Matoušek (1998*a*) and the upper bound by Wasilkowski and Woźniakowski (2010).

5. Lattice rules

In Section 2 we gave a brief introduction to lattice rules. We defined *rank-one lattice rules* and explained how *randomly shifted lattice rules* can be used for practical error estimation. We also introduced the *component-by-component* (CBC) construction for obtaining good lattice rule generating vectors. Here in this section we provide the theory behind the CBC construction, including error analysis of randomly shifted lattice rules and the fast implementation of the CBC algorithm. We will also discuss *extensible*

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

lattice sequences, which turn lattice rules from 'closed' point sets to 'open' sequences, making them more flexible for practical applications.

For most of this section, we will consider the weighted anchored or unanchored spaces of Section 4, which contain integrands with square-integrable mixed *first* derivatives. The lattice rules constructed for these function spaces can achieve close to $O(n^{-1})$ convergence rate, with implied constants that can be independent of the dimension s if the weights of the function space satisfy a certain condition. We will also discuss the use of the *baker's transformation* to obtain close to $O(n^{-2})$ convergence rate when the integrands have square-integrable mixed *second* derivatives, again with implied constants that can be independent of s. We will also outline other strategies for obtaining even higher orders of convergence using lattice rules, but now the error bounds known so far have exponential growth in s.

5.1. A brief background on the classical theory of lattice rules

Lattice rules were originally designed for periodic functions. It is customary to assume that the integrand f has an *absolutely convergent Fourier series*

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{h}\in\mathbb{Z}^s} \widehat{f}(\boldsymbol{h})\, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{h}\cdot\boldsymbol{x}}, \quad \mathrm{i}^2 = -1, \tag{5.1}$$

with Fourier coefficients

$$\widehat{f}(\boldsymbol{h}) = \int_{[0,1]^s} f(\boldsymbol{x})\, \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{h}\cdot\boldsymbol{x}}\, \mathrm{d}\boldsymbol{x},$$

where $\boldsymbol{h}\cdot\boldsymbol{x} = h_1x_1 + \cdots + h_sx_s$ denotes the usual vector dot product. It follows from the absolute (and hence uniform) convergence of (5.1) that f is necessarily continuous and also *one-periodic* with respect to each variable, that is, $f(\boldsymbol{x})|_{x_j=0} = f(\boldsymbol{x})|_{x_j=1}$ for all $j = 1, \ldots, s$.

Theorem 5.1 (the lattice rule error). Let $Q_{n,s}$ denote a lattice rule (not necessarily rank-one) and let $\mathcal{L}$ denote the associated integration lattice. If f has an absolutely convergent Fourier series (5.1), then

$$Q_{n,s}(f) - I_s(f) = \sum_{\boldsymbol{h}\in\mathcal{L}^\perp\setminus\{\boldsymbol{0}\}} \widehat{f}(\boldsymbol{h}),$$

where $\mathcal{L}^\perp := \{\boldsymbol{h}\in\mathbb{Z}^s : \boldsymbol{h}\cdot\boldsymbol{x}\in\mathbb{Z} \text{ for all } \boldsymbol{x}\in\mathcal{L}\}$ is the *dual lattice* associated with $\mathcal{L}$.

We will prove this result for rank-one lattice rules (see (2.3)), that is, for QMC rules of the form

$$Q_{n,s}(f) = \frac{1}{n}\sum_{i=0}^{n-1} f\left(\left\{\frac{i\boldsymbol{z}}{n}\right\}\right).$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

In this case, $\mathcal{L}^\perp = \{\boldsymbol{h} \in \mathbb{Z}^s : \boldsymbol{h} \cdot \boldsymbol{z} \equiv 0 \pmod{n}\}$. A proof for general lattice rules is given in Sloan and Joe (1994).

Theorem 5.2 (rank-one lattice rule error). Let $Q_{n,s}$ denote a rank-one lattice rule with generating vector $\boldsymbol{z}$. If f has an absolutely convergent Fourier series (5.1), then

$$Q_{n,s}(f) - I_s(f) = \sum_{\substack{\boldsymbol{h} \in \mathbb{Z}^s \setminus \{\boldsymbol{0}\} \\ \boldsymbol{h} \cdot \boldsymbol{z} \equiv 0 \,(\mathrm{mod}\, n)}} \widehat{f}(\boldsymbol{h}).$$

Proof. Using the Fourier series (5.1), we can write

$$\begin{aligned} Q_{n,s}(f) - I_s(f) &= \frac{1}{n} \sum_{k=0}^{n-1} f\left(\left\{ \frac{k\boldsymbol{z}}{n} \right\} \right) - \int_{[0,1]^s} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \\ &= \sum_{\boldsymbol{h} \in \mathbb{Z}^s} \widehat{f}(\boldsymbol{h}) \left(\frac{1}{n} \sum_{k=0}^{n-1} \mathrm{e}^{2\pi \mathrm{i} k \boldsymbol{h} \cdot \boldsymbol{z}/n} \right) - \sum_{\boldsymbol{h} \in \mathbb{Z}^s} \widehat{f}(\boldsymbol{h}) \int_{[0,1]^s} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{h} \cdot \boldsymbol{x}} \, \mathrm{d}\boldsymbol{x}. \end{aligned}$$

The result then follows from the elementary properties

$$\int_{[0,1]^s} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{h} \cdot \boldsymbol{x}} \, \mathrm{d}\boldsymbol{x} = \begin{cases} 1 & \text{if } \boldsymbol{h} = \boldsymbol{0}, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\frac{1}{n} \sum_{k=0}^{n-1} \mathrm{e}^{2\pi \mathrm{i} k \boldsymbol{h} \cdot \boldsymbol{z}/n} = \begin{cases} 1 & \text{if } \boldsymbol{h} \cdot \boldsymbol{z} \equiv 0 \pmod{n}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

□

The property (5.2) is sometimes referred to as the 'character property' of lattice rules with respect to the exponential functions. It is one of the key properties we need for the error analysis of lattice rules.

In the classical theory of lattice rules, one considers a class $E_\alpha(c)$ of functions whose Fourier coefficients satisfy, for $\alpha > 1$ and $c > 0$,

$$|\widehat{f}(\boldsymbol{h})| \le \frac{c}{(\overline{h}_1 \cdots \overline{h}_s)^\alpha}, \quad \text{with } \overline{h} := \max(1, |h|).$$

It follows that

$$|Q_{n,s}(f) - I_s(f)| \le c \sum_{\substack{\boldsymbol{h} \in \mathbb{Z}^s \setminus \{\boldsymbol{0}\} \\ \boldsymbol{h} \cdot \boldsymbol{z} \equiv 0 \,(\mathrm{mod}\, n)}} \frac{1}{(\overline{h}_1 \cdots \overline{h}_s)^\alpha}, \quad \text{for } f \in E_\alpha(c).$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

This leads to the definition of a quality measure called P_α,

$$P_\alpha(\boldsymbol{z}, n) := \sum_{\substack{\boldsymbol{h}\in\mathbb{Z}^s\setminus\{\boldsymbol{0}\} \\ \boldsymbol{h}\cdot\boldsymbol{z}\equiv 0 \,(\mathrm{mod}\, n)}} \frac{1}{(\overline{h}_1\cdots\overline{h}_s)^\alpha}, \quad \alpha > 1, \tag{5.3}$$

and the goal is to choose $\boldsymbol{z}$ so that $P_\alpha(\boldsymbol{z}, n)$ is as small as possible. By using the character property (5.2) (or by recognizing that P_α is the integration error for a function f with Fourier coefficients $\widehat{f}(\boldsymbol{h}) = 1/(\overline{h}_1\cdots\overline{h}_s)^\alpha$), one can express P_α as

$$\begin{aligned} P_\alpha(\boldsymbol{z}, n) &= -1 + \frac{1}{n}\sum_{k=0}^{n-1}\prod_{j=1}^{s}\left(1 + \sum_{h\in\mathbb{Z}\setminus\{0\}} \frac{\mathrm{e}^{2\pi\mathrm{i}khz_j/n}}{|h|^\alpha}\right) \\ &= -1 + \frac{1}{n}\prod_{j=1}^{s}\big(1 + 2\zeta(\alpha)\big) + \frac{1}{n}\sum_{k=1}^{n-1}\prod_{j=1}^{s}\left(1 + \sum_{h\in\mathbb{Z}\setminus\{0\}} \frac{\mathrm{e}^{2\pi\mathrm{i}khz_j/n}}{|h|^\alpha}\right), \end{aligned} \tag{5.4}$$

where

$$\zeta(x) := \sum_{h=1}^{\infty}\frac{1}{h^x}, \quad x > 1, \tag{5.5}$$

is the Riemann zeta function. For practical computation, α is often taken to be an even integer. This is because the *Bernoulli polynomial of degree* α, with α an even integer, has the Fourier series

$$B_\alpha(x) = \frac{(-1)^{\frac{\alpha}{2}+1}\alpha!}{(2\pi)^\alpha}\sum_{h\in\mathbb{Z}\setminus\{0\}}\frac{\mathrm{e}^{2\pi\mathrm{i}hx}}{h^\alpha}, \quad \text{for } x\in[0,1], \tag{5.6}$$

so that in this case we can write

$$\sum_{h\in\mathbb{Z}\setminus\{0\}} \frac{\mathrm{e}^{2\pi\mathrm{i}khz_j/n}}{|h|^\alpha} = \frac{(-1)^{\frac{\alpha}{2}+1}(2\pi)^\alpha}{\alpha!}B_\alpha\left(\left\{\frac{kz_j}{n}\right\}\right). \tag{5.7}$$

This allows P_α in (5.4) to be computed in $O(n\,s)$ operations. One may therefore use the Korobov construction in Example 2.9 to search for a lattice generating vector $\boldsymbol{z}$, using the criterion that P_α be as small as possible.

It is well known that the rate of decay of the Fourier coefficients of a function is related to the smoothness of the function. For instance, if $\alpha > 1$ is an integer and all partial derivatives

$$\frac{\partial^{q_1+\cdots+q_s} f}{\partial x_1^{q_1}\cdots\partial x_s^{q_s}}, \quad 0 \le q_1,\ldots,q_s \le \alpha,$$

exist and are continuous on $[0,1]^s$, then there exists $c > 0$ for which $f \in E_\alpha(c)$. Thus the class $E_\alpha(c)$ is essentially a class of functions with smoothness determined by α, therefore the parameter α is called the *smoothness*

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

parameter. It is known that a convergence rate of

$$P_\alpha(\boldsymbol{z}, n) = O(n^{-\alpha}\,(\log n)^{\alpha\, s})$$

can be achieved. There are also other related quality measures for lattice rules (Niederreiter 1992*a*, Sloan and Joe 1994).

We shall not discuss the classical theory further, because in the classical error analysis all error bounds grow exponentially with dimension. In the following, we will carry out error analysis in the weighted function space setting of Section 4.

5.2. Shift-averaged worst-case error

In this subsection we discuss a general strategy for the error analysis of *randomly shifted QMC rules*, as a preparation for our later analysis of randomly shifted lattice rules.

For any QMC point set $P = \{\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{n-1}\}$ and any shift $\boldsymbol{\Delta} \in [0,1]^s$, let

$$P + \boldsymbol{\Delta} = \{\{\boldsymbol{t}_i + \boldsymbol{\Delta}\} : i = 0, 1, \ldots, n-1\}$$

denote the shifted QMC point set, and let $Q_{n,s}(\boldsymbol{\Delta}; f)$ denote the corresponding shifted QMC rule. Then, for any integrand f belonging to some normed space H, it follows from the definition of the worst-case error (see Definition 3.4) that

$$|I_s(f) - Q_{n,s}(\boldsymbol{\Delta}, f)| \le e_{n,s}(P + \boldsymbol{\Delta}; H)\, \|f\|_H.$$

We deduce a bound for the *root-mean-square* error

$$\sqrt{\mathbb{E}\,|I_s(f) - Q_{n,s}(\boldsymbol{\Delta}, f)|^2} \le e^{\rm sh}_{n,s}(P; H)\, \|f\|_H,$$

where the expectation $\mathbb{E}$ is taken over the random shift $\boldsymbol{\Delta}$ which is uniformly distributed over $[0,1]^s$, and where the quantity

$$e^{\rm sh}_{n,s}(P; H) := \sqrt{\int_{[0,1]^s} e^2_{n,s}(P + \boldsymbol{\Delta}; H)\, {\rm d}\boldsymbol{\Delta}} \tag{5.8}$$

is referred to as the *shift-averaged worst-case error*.

The shift-averaged worst-case error (5.8) will be used as our quality measure for randomly shifted QMC rules. For any given point set P, the averaging argument guarantees the existence of at least one shift $\boldsymbol{\Delta}$ for which

$$e_{n,s}(P + \boldsymbol{\Delta}; H) \le e^{\rm sh}_{n,s}(P; H).$$

The following theorem gives an explicit formula for the shift-averaged worst-case error when the function space is an RKHS.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Theorem 5.3 (formula for the shift-averaged worst-case error). The *shift-averaged* worst-case error (5.8) for a QMC point set P in an RKHS $H_s(K)$ satisfies

$$[e^{\rm sh}_{n,s}(P; H_s(K))]^2 = -\int_{[0,1]^s}\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{y})\,{\rm d}\boldsymbol{x}\,{\rm d}\boldsymbol{y} + \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{k=0}^{n-1} K^{\rm sh}(\boldsymbol{t}_i,\boldsymbol{t}_k), \tag{5.9}$$

where

$$K^{\rm sh}(\boldsymbol{x},\boldsymbol{y}) := \int_{[0,1]^s} K(\{\boldsymbol{x}+\boldsymbol{\Delta}\},\{\boldsymbol{y}+\boldsymbol{\Delta}\})\,{\rm d}\boldsymbol{\Delta} \quad \text{for all } \boldsymbol{x},\boldsymbol{y}\in[0,1]^s. \tag{5.10}$$

Proof. Using the definition (5.8) and applying the formula (3.9) for the worst-case error $e_{n,s}(P+\boldsymbol{\Delta}; H_s(K))$, we obtain

$$\begin{aligned}
&[e^{\rm sh}_{n,s}(P; H_s(K))]^2\\
&= \int_{[0,1]^s}\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{y})\,{\rm d}\boldsymbol{x}\,{\rm d}\boldsymbol{y} - \frac{2}{n}\sum_{i=0}^{n-1}\int_{[0,1]^s}\int_{[0,1]^s} K(\{\boldsymbol{t}_i+\boldsymbol{\Delta}\},\boldsymbol{y})\,{\rm d}\boldsymbol{\Delta}\,{\rm d}\boldsymbol{y}\\
&\quad + \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{k=0}^{n-1}\int_{[0,1]^s} K(\{\boldsymbol{t}_i+\boldsymbol{\Delta}\},\{\boldsymbol{t}_k+\boldsymbol{\Delta}\})\,{\rm d}\boldsymbol{\Delta}.
\end{aligned}$$

With a change of variables $\boldsymbol{x} = \{\boldsymbol{t}_i+\boldsymbol{\Delta}\}$, the double integrals in the second term turn into the double integral in the first term. The result then follows from the definition (5.10). □

The function $K^{\rm sh}$ defined by (5.10) is actually a reproducing kernel, with the *shift-invariant property*

$$K^{\rm sh}(\boldsymbol{x},\boldsymbol{y}) = K^{\rm sh}(\{\boldsymbol{x}+\boldsymbol{\Delta}\},\{\boldsymbol{y}+\boldsymbol{\Delta}\}) \quad \text{for all } \boldsymbol{x},\boldsymbol{y},\boldsymbol{\Delta}\in[0,1]^s,$$

or equivalently

$$K^{\rm sh}(\boldsymbol{x},\boldsymbol{y}) = K^{\rm sh}(\{\boldsymbol{x}-\boldsymbol{y}\},\boldsymbol{0}) \quad \text{for all } \boldsymbol{x},\boldsymbol{y}\in[0,1]^s.$$

Furthermore, it can be verified that the shift-averaged worst-case error $e^{\rm sh}_{n,s}(P; H_s(K))$ is precisely the worst-case error of the QMC point set P in the RKHS with $K^{\rm sh}$ as the reproducing kernel, that is,

$$\int_{[0,1]^s} e^2_{n,s}(P+\boldsymbol{\Delta}; H_s(K))\,{\rm d}\boldsymbol{\Delta} = e^2_{n,s}(P; H_s(K^{\rm sh})).$$

In words, *the average over all possible shifts of the squared worst-case error for a shifted QMC rule in a Hilbert space with reproducing kernel K* is equal to *the squared worst-case error of the original unshifted QMC rule in a Hilbert space with reproducing kernel $K^{\rm sh}$*.This important connection

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

provides a powerful tool for analysing randomly shifted QMC rules, because it is often easier to work with a shift-invariant kernel. We refer to the kernel K^{sh} as the *shift-invariant kernel associated with* K.

Further, we note that for the reproducing kernels we consider below, the reproducing kernel K^{sh} has the same smoothness properties as K. As the smoothness properties of the reproducing kernel determine the convergence rate of the worst-case error, one obtains the same convergence rate of the worst-case error for $H_s(K^{\mathrm{sh}})$ as for $H_s(K)$.

5.3. Randomly shifted lattice rules in weighted Sobolev spaces

We are now ready to analyse randomly shifted lattice rules in the weighted anchored or unanchored Sobolev space of Section 4. First we derive the associated shift-invariant kernel needed for the shift-averaged worst-case error.

Lemma 5.4. Let $H(K_{s,\boldsymbol{\gamma}})$ be the weighted anchored or unanchored Sobolev space of Section 4. The shift-invariant kernel associated with $K_{s,\boldsymbol{\gamma}}$ is

$$K^{\mathrm{sh}}_{s,\boldsymbol{\gamma}}(\boldsymbol{x},\boldsymbol{y}) = \sum_{\mathrm{u}\subseteq\{1:s\}} \gamma_{\mathrm{u}} \prod_{j\in\mathrm{u}} \big(B_2(|x_j - y_j|) + \beta\big), \tag{5.11}$$

where $\beta = c^2 - c + 1/3$ for the anchored variant with anchor c, and $\beta = 0$ for the unanchored variant.

Proof. From the definitions (5.10) and (4.3), we have

$$K^{\mathrm{sh}}_{s,\boldsymbol{\gamma}}(\boldsymbol{x},\boldsymbol{y}) = \sum_{\mathrm{u}\subseteq\{1:s\}} \gamma_{\mathrm{u}} \prod_{j\in\mathrm{u}} \int_0^1 \eta(\{x_j+\Delta_j\},\{y_j+\Delta_j\})\,\mathrm{d}\Delta_j,$$

where the function η can be written in the common form (4.8) for both the anchored and unanchored spaces. Using the symmetry $B_2(x) = B_2(1-x)$ and $\int_0^1 \alpha(x)\,\mathrm{d}x = 0$, we obtain

$$\begin{aligned}&\int_0^1 \eta(\{x+\Delta\},\{y+\Delta\})\,\mathrm{d}\Delta\\ &= \tfrac{1}{2}B_2(|x-y|) + \int_0^1 \big(\{x+\Delta\}-\tfrac{1}{2}\big)\big(\{y+\Delta\}-\tfrac{1}{2}\big)\mathrm{d}\Delta + \beta.\end{aligned}$$

Therefore it remains to show that

$$\mathcal{I}(x,y) := \int_0^1 \big(\{x+\Delta\}-\tfrac{1}{2}\big)\big(\{y+\Delta\}-\tfrac{1}{2}\big)\mathrm{d}\Delta = \tfrac{1}{2}B_2(|x-y|).$$

Since $\mathcal{I}(x,y) = \mathcal{I}(y,x)$, without loss of generality we assume that $x \le y$. There are three possible arrangements of the values of $x+\Delta$ and $y+\Delta$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

for $x, y, \Delta \in [0,1)$:

$$\begin{aligned} &x+\Delta \le y+\Delta < 1 \Rightarrow \Delta < 1-y,\\ &x+\Delta < 1 \le y+\Delta \Rightarrow 1-y \le \Delta < 1-x,\\ &1 \le x+\Delta \le y+\Delta \Rightarrow 1-x \le \Delta. \end{aligned}$$

Thus we have

$$\begin{aligned} \mathcal{I}(x,y) &= \int_0^{1-y} \big(x+\Delta-\tfrac{1}{2}\big)\big(y+\Delta-\tfrac{1}{2}\big)\mathrm{d}\Delta \\ &\quad + \int_{1-y}^{1-x} \big(x+\Delta-\tfrac{1}{2}\big)\big(y+\Delta-1-\tfrac{1}{2}\big)\mathrm{d}\Delta \\ &\quad + \int_{1-x}^{1} \big(x+\Delta-1-\tfrac{1}{2}\big)\big(y+\Delta-1-\tfrac{1}{2}\big)\mathrm{d}\Delta \\ &= \tfrac{1}{2}(y-x)^2 - \tfrac{1}{2}(y-x) + \tfrac{1}{12} = \tfrac{1}{2}\big(|x-y|^2 - |x-y| + \tfrac{1}{6}\big). \end{aligned}$$

This completes the proof. □

For simplicity, let

$$e^{\mathrm{sh}}_{n,s}(\boldsymbol{z}) := e^{\mathrm{sh}}_{n,s}(P; H(K_{s,\boldsymbol{\gamma}}))$$

denote the shift-averaged worst-case error for a rank-one lattice rule with point set P and generating vector $\boldsymbol{z}$ in our weighted anchored or unanchored space. The following lemma gives an explicit expression for $[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2$.

Lemma 5.5 (shift-averaged worst-case error for lattice rules). The shift-averaged worst-case error for a rank-one lattice rule in the weighted anchored or unanchored Sobolev space satisfies

$$[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2 = \sum_{\emptyset \ne \mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}} \left(\frac{1}{n} \sum_{k=0}^{n-1} \prod_{j \in \mathfrak{u}} \left[B_2\left(\left\{ \frac{k z_j}{n} \right\} \right) + \beta \right] - \beta^{|\mathfrak{u}|} \right), \tag{5.12}$$

where $\beta = c^2 - c + 1/3$ for the anchored variant with anchor c, and $\beta = 0$ for the unanchored variant. For product weights, the expression simplifies to

$$[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2 = -\prod_{j=1}^{s}\big(1+\gamma_j\beta\big) + \frac{1}{n}\sum_{k=0}^{n-1}\prod_{j=1}^{s}\left(1+\gamma_j\left[B_2\left(\left\{\frac{kz_j}{n}\right\}\right)+\beta\right]\right). \tag{5.13}$$

Proof. Substituting $\boldsymbol{t}_i = \{i\boldsymbol{z}/n\}$ into (5.9) and using (4.5) and (5.11), we obtain

$$[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2 = -\sum_{\mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}} \beta^{|\mathfrak{u}|} + \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{k=0}^{n-1} \sum_{\mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}} \prod_{j \in \mathfrak{u}} \left[B_2\left(\left\{ \frac{(i-k) z_j}{n} \right\} \right) + \beta \right].$$

As i and k range from 0 to $n-1$, the values of $(i-k) \bmod n$ are just $0, \dots, n-1$ in some order, with each value occurring n times. Thus we can

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

reduce the double sum in $[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2$ to a single sum. Grouping the sum over $\mathfrak{u}$ and then cancelling the $\mathfrak{u} = \emptyset$ term yields the formula for general weights above. The formula for product weights follows from the identity (4.2). □

5.4. Component-by-component construction

Recall that the components of the generating vector $\boldsymbol{z}$ can be restricted to the set

$$\mathbb{U}_n = \{z \in \mathbb{Z} : 1 \le z \le n-1 \text{ and } \gcd(z,n) = 1\},$$

whose cardinality is given by the Euler totient function

$$\varphi(n) := |\mathbb{U}_n| = |\{z \in \mathbb{Z} : 1 \le z \le n-1 \text{ and } \gcd(z,n) = 1\}|. \tag{5.14}$$

When n is prime $\varphi(n)$ takes its largest value $n-1$, hence there are altogether up to $(n-1)^s$ possible choices for $\boldsymbol{z}$, far too many to allow an exhaustive search for the best one.

The CBC construction below, already foreshadowed in Example 2.10, provides a feasible way to obtain good lattice generating vectors. The first component z_1 is arbitrarily set to 1, since all choices in one dimension lead to the same rectangle rule.

Algorithm 5.6 (CBC construction). Given: n, $s_{\max}$, and weights $\gamma_{\mathfrak{u}}$.

(1) Set $z_1 = 1$.

(2) For $s = 2, 3, \ldots, s_{\max}$, choose z_s in $\mathbb{U}_n$ to minimize $[e^{\mathrm{sh}}_{n,s}(z_1, \ldots, z_s)]^2$.

With general weights $\gamma_{\mathfrak{u}}$, the cost of the CBC algorithm is prohibitively expensive, thus when discussing implementation we will always assume there is some special structure, such as product weights, order-dependent weights, finite-order weights, or POD weights. We will discuss the implementation of CBC in later subsections.

In this subsection we focus on error bounds resulting from the CBC construction. The CBC error bounds are proved by induction: under the induction hypothesis that the desired bound holds in $s-1$ dimensions, we show by the averaging argument that the algorithm picks a next component z_s, for which the bound holds in s dimensions.

To illustrate the general argument, we first prove that in the simplest case of product weights and prime n, a generating vector constructed by the CBC algorithm beats the QMC mean (see (3.13) together with (4.5))

$$E^2_{n,s}(K_{s,\boldsymbol{\gamma}}) = \frac{1}{n}\left(\prod_{j=1}^{s}\left(1 + \gamma_j\left(\beta + \frac{1}{6}\right)\right) - \prod_{j=1}^{s}(1 + \gamma_j\beta)\right),$$

which yields a convergence rate of $O(n^{-1/2})$. We will prove a better, if more complicated, result in Theorem 5.8 below.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Theorem 5.7 (CBC error bound beats QMC mean). Consider the weighted anchored or unanchored Sobolev space with product weights $\gamma_1 \geq \gamma_2 \geq \cdots > 0$, and suppose that n is a prime number satisfying

$$n \geq \frac{\gamma_1}{6(1+\gamma_1\beta)}, \tag{5.15}$$

where $\beta = c^2 - c + 1/3$ for the anchored variant with anchor c, and $\beta = 0$ for the unanchored variant. The generating vector $\boldsymbol{z} \in \mathbb{U}_n^s$ constructed by the CBC algorithm, minimizing the squared shift-averaged worst-case error $[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2$ in the weighted anchored or unanchored Sobolev space in each step, satisfies

$$[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2 < E^2_{n,s}(K_{s,\boldsymbol{\gamma}}). \tag{5.16}$$

Proof. For the inductive step we need to show, under the assumption $[e^{\mathrm{sh}}_{n,s-1}(z_1,\ldots,z_{s-1})]^2 < E^2_{n,s-1}(K_{s-1,\boldsymbol{\gamma}})$, that $[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2 < E^2_{n,s}(K_{s,\boldsymbol{\gamma}})$ when $z_s \in \mathbb{U}_n$ is chosen to minimize $[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2$.

To proceed, we rewrite (5.13) as

$$[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2 = \big(1+\gamma_s\beta\big)[e^{\mathrm{sh}}_{n,s-1}(z_1,\ldots,z_{s-1})]^2 + \frac{\gamma_s}{6n}\prod_{j=1}^{s-1}\left(1+\gamma_j\left(\beta+\frac{1}{6}\right)\right)$$
$$+ \frac{\gamma_s}{n}\sum_{k=1}^{n-1}\left[B_2\left(\left\{\frac{kz_s}{n}\right\}\right)\prod_{j=1}^{s-1}\left(1+\gamma_j\left[B_2\left(\left\{\frac{kz_j}{n}\right\}\right)+\beta\right]\right)\right],$$

where we separated out the $k=0$ term and used $B_2(0) = 1/6$. Next we average over all possible choices of z_s, forming

$$\mu(z_1,\ldots,z_{s-1}) := \frac{1}{n-1}\sum_{z_s=1}^{n-1}[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2.$$

Since $B_2(x) = (1/2\pi^2)\sum_{h\in\mathbb{Z}\setminus\{0\}} \mathrm{e}^{2\pi\mathrm{i}hx}/h^2$ for $x \in [0,1]$, for any integer k from 1 to $n-1$ we can write

$$\frac{2\pi^2}{n-1}\sum_{z_s=1}^{n-1} B_2\left(\left\{\frac{kz_s}{n}\right\}\right) = \frac{1}{n-1}\sum_{z=1}^{n-1}\sum_{h\in\mathbb{Z}\setminus\{0\}}\frac{\mathrm{e}^{2\pi\mathrm{i}hkz/n}}{h^2}$$
$$= \sum_{\substack{h\in\mathbb{Z}\setminus\{0\}\\ h\equiv 0 \,(\mathrm{mod}\, n)}}\frac{1}{h^2} - \frac{1}{n-1}\sum_{\substack{h\in\mathbb{Z}\setminus\{0\}\\ h\not\equiv 0 \,(\mathrm{mod}\, n)}}\frac{1}{h^2}$$
$$= \sum_{\substack{h\in\mathbb{Z}\setminus\{0\}\\ h\equiv 0 \,(\mathrm{mod}\, n)}}\frac{1}{h^2} - \frac{1}{n-1}\left(\sum_{h\in\mathbb{Z}\setminus\{0\}}\frac{1}{h^2} - \sum_{\substack{h\in\mathbb{Z}\setminus\{0\}\\ h\equiv 0 \,(\mathrm{mod}\, n)}}\frac{1}{h^2}\right)$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

$$= \frac{n}{n-1} \sum_{\substack{h \in \mathbb{Z}\setminus\{0\} \\ h \equiv 0 \,(\mathrm{mod}\, n)}} \frac{1}{h^2} - \frac{1}{n-1} \sum_{h \in \mathbb{Z}\setminus\{0\}} \frac{1}{h^2}$$
$$= \frac{2\pi^2 n}{6n^2(n-1)} - \frac{2\pi^2}{6(n-1)} = -\frac{2\pi^2}{6n},$$

where we used $\sum_{h=1}^{\infty} 1/h^2 = \pi^2/6$. Thus we conclude for any integer $1 \le k \le n-1$ that

$$\frac{1}{n-1} \sum_{z_s=1}^{n-1} B_2\left(\left\{\frac{kz_s}{n}\right\}\right) = -\frac{1}{6n}. \tag{5.17}$$

Combining the expressions yields

$$\mu(z_1, \ldots, z_{s-1}) = \big(1 + \gamma_s\beta\big)[e^{\mathrm{sh}}_{n,s-1}(z_1, \ldots, z_{s-1})]^2$$
$$+ \frac{\gamma_s}{6n} \prod_{j=1}^{s-1}\left(1 + \gamma_j\left(\beta + \frac{1}{6}\right)\right) - \frac{\gamma_s}{6n^2} \sum_{k=1}^{n-1} \prod_{j=1}^{s-1}\left(1 + \gamma_j\left[B_2\left(\left\{\frac{kz_j}{n}\right\}\right) + \beta\right]\right).$$

Rewriting the third expression in terms of $e^2_{n,s-1}(z_1, \ldots, z_{s-1})$ and then collecting similar expressions, we obtain

$$\mu(z_1, \ldots, z_{s-1}) = \left(1 + \gamma_s\beta - \frac{\gamma_s}{6n}\right)[e^{\mathrm{sh}}_{n,s-1}(z_1, \ldots, z_{s-1})]^2$$
$$- \frac{\gamma_s}{6n} \prod_{j=1}^{s-1}(1 + \gamma_j\beta) + \left(\frac{\gamma_s}{6} + \frac{\gamma_s}{6n}\right)\frac{1}{n} \prod_{j=1}^{s-1}\left(1 + \gamma_j\left(\beta + \frac{1}{6}\right)\right).$$

The condition (5.15) ensures that $1 + \gamma_s\beta - \gamma_s/(6n) \ge 0$. Applying the induction hypothesis and dropping a negative term, we finally arrive at $\mu(z_1, \ldots, z_{s-1}) < E^2_{n,s}(K_{s,\boldsymbol{\gamma}})$.

Since $\mu(z_1, \ldots, z_{s-1})$ is the average of $[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2$ over all $z_s \in \mathbb{U}_n$, the choice of z_s that minimizes $[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2$ must satisfy

$$[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2 \le \mu(z_1, \ldots, z_{s-1}) < E^2_{n,s}(K_{s,\boldsymbol{\gamma}}).$$

To complete the proof by induction we note that the error bound (5.16) is easily satisfied for $s = 1$: indeed we may adapt (5.17) to obtain

$$[e^{\mathrm{sh}}_{n,1}(1)]^2 = \frac{\gamma_1}{n} \sum_{k=0}^{n-1} B_2\left(\frac{k}{n}\right) = \frac{\gamma_1}{n}\left(\frac{1}{6} - \frac{n-1}{6n}\right) = \frac{\gamma_1}{6n^2} \le \frac{\gamma_1}{6n}.$$

This completes the proof. □

We now show, through a more complicated averaging argument in the induction proof, that the CBC algorithm yields a convergence rate arbitrarily close to $O(n^{-1})$. The result holds for general n (not necessarily prime),

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

and holds for the unanchored space with general weights (including product weights), but holds for the anchored space only with product weights. For the anchored space with general non-product weights, the same error bound holds, but a modification to the search criterion in the CBC construction is needed. We therefore state the results for the unanchored and anchored spaces separately.

In the following theorems, the choice of z_s at each step is independent of the parameter λ, and the error bound holds for all values of $\lambda \in (1/2, 1]$. The optimal convergence rate close to $O(n^{-1})$ is obtained with $\lambda \to 1/2$, but note that $\lambda = 1/2$ is not permitted because $\zeta(2\lambda) \to \infty$ as $\lambda \to 1/2$. The implied constant in the big-O bound can be independent of s under an appropriate condition on the weights. For example, in the case of product weights, a sufficient condition to obtain close to $O(n^{-1})$ convergence with the implied constant independently of s is $\sum_{j=1}^{\infty} \gamma_j^{1/2} < \infty$.

Theorem 5.8 (optimal CBC error bound: the unanchored case). The generating vector $\boldsymbol{z} \in \mathbb{U}_n^s$ constructed by the CBC algorithm, minimizing the squared shift-averaged worst-case error $[e_{n,s}^{\mathrm{sh}}(\boldsymbol{z})]^2$ for the weighted unanchored Sobolev space in each step, satisfies

$$[e_{n,s}^{\mathrm{sh}}(\boldsymbol{z})]^2 \le \left(\frac{1}{\varphi(n)} \sum_{\emptyset \ne \mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}}^{\lambda} \left(\frac{2\zeta(2\lambda)}{(2\pi^2)^{\lambda}} \right)^{|\mathfrak{u}|} \right)^{1/\lambda}, \tag{5.18}$$

for all $\lambda \in (1/2, 1]$, where $\zeta(\cdot)$ is the Riemann zeta function (5.5), and $\varphi(n)$ is the Euler totient function (5.14).

Proof. We prove this by induction on s. The base step $s = 1$ is straightforward to verify for all $\lambda \in (1/2, 1]$. Assume now that we have chosen the first $s-1$ components $z_1, \ldots, z_{s-1}$ and that (5.18) holds with s replaced by $s-1$. We separate the terms in (5.12) (remembering that $\beta = 0$ for this unanchored space), depending on whether or not the element s is included in the set $\mathfrak{u}$, to obtain the recursive expression

$$[e_{n,s}^{\mathrm{sh}}(z_1, \ldots, z_{s-1}, z_s)]^2 = [e_{n,s-1}^{\mathrm{sh}}(z_1, \ldots, z_{s-1})]^2 + \theta(z_s), \tag{5.19}$$

with (suppressing the dependence of θ on $z_1, \ldots, z_{s-1}$)

$$\begin{aligned} \theta(z_s) &:= \sum_{s \in \mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}} \left(\frac{1}{n} \sum_{k=0}^{n-1} \prod_{j \in \mathfrak{u}} B_2\left(\left\{ \frac{k z_j}{n} \right\} \right) \right) \\ &= \sum_{s \in \mathfrak{u} \subseteq \{1:s\}} \frac{\gamma_{\mathfrak{u}}}{(2\pi^2)^{|\mathfrak{u}|}} \left(\frac{1}{n} \sum_{k=0}^{n-1} \sum_{\boldsymbol{h}_{\mathfrak{u}} \in (\mathbb{Z} \setminus \{0\})^{|\mathfrak{u}|}} \frac{\mathrm{e}^{2\pi \mathrm{i} k \boldsymbol{h}_{\mathfrak{u}} \cdot \boldsymbol{z}_{\mathfrak{u}}/n}}{\prod_{j \in \mathfrak{u}} h_j^2} \right) \end{aligned}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

$$= \sum_{s\in\mathfrak{u}\subseteq\{1:s\}} \frac{\gamma_{\mathfrak{u}}}{(2\pi^2)^{|\mathfrak{u}|}} \left(\sum_{\substack{\boldsymbol{h}_{\mathfrak{u}}\in(\mathbb{Z}\setminus\{0\})^{|\mathfrak{u}|} \\ \boldsymbol{h}_{\mathfrak{u}}\cdot\boldsymbol{z}_{\mathfrak{u}}\equiv 0 \,(\mathrm{mod}\, n)}} \frac{1}{\prod_{j\in\mathfrak{u}} h_j^2} \right)$$

$$= \sum_{s\in\mathfrak{u}\subseteq\{1:s\}} \frac{\gamma_{\mathfrak{u}}}{(2\pi^2)^{|\mathfrak{u}|}} \left(\sum_{h_s\in\mathbb{Z}\setminus\{0\}} \frac{1}{h_s^2} \sum_{\substack{\boldsymbol{h}_{\mathfrak{u}\setminus\{s\}}\in(\mathbb{Z}\setminus\{0\})^{|\mathfrak{u}|-1} \\ \boldsymbol{h}_{\mathfrak{u}\setminus\{s\}}\cdot\boldsymbol{z}_{\mathfrak{u}\setminus\{s\}}\equiv -h_s z_s \,(\mathrm{mod}\, n)}} \frac{1}{\prod_{j\in\mathfrak{u}\setminus\{s\}} h_j^2} \right),$$

where we made use of the Fourier expansion of B_2 and the character property (5.2).

If z_s^* denotes the value chosen by the CBC algorithm in dimension s, then (since the minimum is always smaller than or equal to the average) we have for all $\lambda \in (0,1]$ that

$$[\theta(z_s^*)]^\lambda \le \frac{1}{\varphi(n)} \sum_{z_s\in\mathbb{U}_n} [\theta(z_s)]^\lambda \le \frac{1}{\varphi(n)} \sum_{z_s\in\mathbb{U}_n} \sum_{s\in\mathfrak{u}\subseteq\{1:s\}} \frac{\gamma_{\mathfrak{u}}^\lambda}{(2\pi^2)^{|\mathfrak{u}|\lambda}}$$
$$\times \left(\sum_{h_s\in\mathbb{Z}\setminus\{0\}} \frac{1}{|h_s|^{2\lambda}} \sum_{\substack{\boldsymbol{h}_{\mathfrak{u}\setminus\{s\}}\in(\mathbb{Z}\setminus\{0\})^{|\mathfrak{u}|-1} \\ \boldsymbol{h}_{\mathfrak{u}\setminus\{s\}}\cdot\boldsymbol{z}_{\mathfrak{u}\setminus\{s\}}\equiv -h_s z_s \,(\mathrm{mod}\, n)}} \frac{1}{\prod_{j\in\mathfrak{u}\setminus\{s\}} |h_j|^{2\lambda}} \right),$$

where we used the inequality (sometimes referred to as Jensen's inequality)

$$\left(\sum_k a_k\right)^\lambda \le \sum_k a_k^\lambda, \quad a_k \ge 0,\ \lambda\in(0,1]. \tag{5.20}$$

Next we separate the terms depending on whether or not h_s is a multiple of n, to obtain

$$[\theta(z_s^*)]^\lambda \le \sum_{s\in\mathfrak{u}\subseteq\{1:s\}} \frac{\gamma_{\mathfrak{u}}^\lambda}{(2\pi^2)^{|\mathfrak{u}|\lambda}} \cdot \frac{2\zeta(2\lambda)}{n^{2\lambda}} \left(\sum_{\substack{\boldsymbol{h}_{\mathfrak{u}\setminus\{s\}}\in(\mathbb{Z}\setminus\{0\})^{|\mathfrak{u}|-1} \\ \boldsymbol{h}_{\mathfrak{u}\setminus\{s\}}\cdot\boldsymbol{z}_{\mathfrak{u}\setminus\{s\}}\equiv 0 \,(\mathrm{mod}\, n)}} \frac{1}{\prod_{j\in\mathfrak{u}\setminus\{s\}} |h_j|^{2\lambda}} \right)$$
$$+ \frac{1}{\varphi(n)} \sum_{z_s\in\mathbb{U}_n} \sum_{c=1}^{n-1} \sum_{s\in\mathfrak{u}\subseteq\{1:s\}} \frac{\gamma_{\mathfrak{u}}^\lambda}{(2\pi^2)^{|\mathfrak{u}|\lambda}}$$
$$\times \left(\sum_{\substack{h_s\in\mathbb{Z}\setminus\{0\} \\ h_s\equiv -cz_s^{-1} \,(\mathrm{mod}\, n)}} \frac{1}{|h_s|^{2\lambda}} \sum_{\substack{\boldsymbol{h}_{\mathfrak{u}\setminus\{s\}}\in(\mathbb{Z}\setminus\{0\})^{|\mathfrak{u}|-1} \\ \boldsymbol{h}_{\mathfrak{u}\setminus\{s\}}\cdot\boldsymbol{z}_{\mathfrak{u}\setminus\{s\}}\equiv c \,(\mathrm{mod}\, n)}} \frac{1}{\prod_{j\in\mathfrak{u}\setminus\{s\}} |h_j|^{2\lambda}} \right),$$

where z_s^{-1} denotes the multiplicative inverse of z_s in $\mathbb{U}_n$, that is, $z_s z_s^{-1} \equiv 1 \,(\mathrm{mod} n)$. For fixed c satisfying $1 \le c \le n-1$, we have $\{cz_s^{-1} \bmod n : z_s \in \mathbb{U}_n\} = \{cz \bmod n : z \in \mathbb{U}_n\}$. Let $g = \gcd(c,n)$. Then $\gcd(c/g, n/g) = 1$,

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

and

$$\sum_{z_s\in\mathbb{U}_n}\sum_{\substack{h_s\in\mathbb{Z}\setminus\{0\}\\ h_s\equiv -cz_s^{-1}\,(\mathrm{mod}\,n)}}\frac{1}{|h_s|^{2\lambda}}=\sum_{z\in\mathbb{U}_n}\sum_{\substack{h_s\in\mathbb{Z}\setminus\{0\}\\ h_s\equiv -cz\,(\mathrm{mod}\,n)}}\frac{1}{|h_s|^{2\lambda}}$$
$$=\sum_{z\in\mathbb{U}_n}\sum_{m\in\mathbb{Z}}\frac{1}{|mn-cz|^{2\lambda}}=g^{-2\lambda}\sum_{z\in\mathbb{U}_n}\sum_{m\in\mathbb{Z}}\frac{1}{|m(n/g)-(c/g)z|^{2\lambda}}$$
$$=g^{-2\lambda}\sum_{z\in\mathbb{U}_n}\sum_{\substack{h\in\mathbb{Z}\setminus\{0\}\\ h\equiv -(c/g)z\,(\mathrm{mod}\,n/g)}}\frac{1}{|h|^{2\lambda}}\le g^{-2\lambda}g\sum_{a=1}^{n/g-1}\sum_{\substack{h\in\mathbb{Z}\setminus\{0\}\\ h\equiv a\,(\mathrm{mod}\,n/g)}}\frac{1}{|h|^{2\lambda}}$$
$$=g^{1-2\lambda}\cdot 2\zeta(2\lambda)\big(1-(n/g)^{-2\lambda}\big)\le 2\zeta(2\lambda),$$

where the last step holds because $g\ge 1$ and $\lambda>1/2$. (The condition $\lambda>1/2$ is needed to ensure that $\zeta(2\lambda)<\infty$.) Hence

$$[\theta(z_s^*)]^\lambda\le\sum_{s\in\mathfrak{u}\subseteq\{1:s\}}\frac{\gamma_\mathfrak{u}^\lambda}{(2\pi^2)^{|\mathfrak{u}|\lambda}}\cdot\frac{2\zeta(2\lambda)}{n^{2\lambda}}\Bigg(\sum_{\substack{\boldsymbol{h}_{\mathfrak{u}\setminus\{s\}}\in(\mathbb{Z}\setminus\{0\})^{|\mathfrak{u}|-1}\\ \boldsymbol{h}_{\mathfrak{u}\setminus\{s\}}\cdot\boldsymbol{z}_{\mathfrak{u}\setminus\{s\}}\equiv 0\,(\mathrm{mod}\,n)}}\frac{1}{\prod_{j\in\mathfrak{u}\setminus\{s\}}|h_j|^{2\lambda}}\Bigg)$$
$$+\frac{1}{\varphi(n)}\sum_{s\in\mathfrak{u}\subseteq\{1:s\}}\frac{\gamma_\mathfrak{u}^\lambda}{(2\pi^2)^{|\mathfrak{u}|\lambda}}\cdot 2\zeta(2\lambda)\Bigg(\sum_{\substack{\boldsymbol{h}_{\mathfrak{u}\setminus\{s\}}\in(\mathbb{Z}\setminus\{0\})^{|\mathfrak{u}|-1}\\ \boldsymbol{h}_{\mathfrak{u}\setminus\{s\}}\cdot\boldsymbol{z}_{\mathfrak{u}\setminus\{s\}}\not\equiv 0\,(\mathrm{mod}\,n)}}\frac{1}{\prod_{j\in\mathfrak{u}\setminus\{s\}}|h_j|^{2\lambda}}\Bigg)$$
$$\le\frac{1}{\varphi(n)}\sum_{s\in\mathfrak{u}\subseteq\{1:s\}}\gamma_\mathfrak{u}^\lambda\left(\frac{2\zeta(2\lambda)}{(2\pi^2)^\lambda}\right)^{|\mathfrak{u}|}.$$

Now we use (5.19), the induction hypothesis and another use of (5.20) to obtain (5.18). □

Theorem 5.9 (optimal CBC error bound: the anchored case). A generating vector $\boldsymbol{z}\in\mathbb{U}_n^s$ can be constructed by a (modified) CBC algorithm such that the squared shift-averaged worst-case error $e_{n,s}^{\mathrm{sh}}(\boldsymbol{z})$ in the weighted anchored Sobolev space with anchor c satisfies

$$[e_{n,s}^{\mathrm{sh}}(\boldsymbol{z})]^2\le\left(\frac{1}{\varphi(n)}\sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:s\}}\gamma_\mathfrak{u}^\lambda\left(\frac{2\zeta(2\lambda)}{(2\pi^2)^\lambda}+\beta^\lambda\right)^{|\mathfrak{u}|}\right)^{1/\lambda}$$

for all $\lambda\in(1/2,1]$, where $\beta=c^2-c+1/3$, $\zeta(\cdot)$ is the Riemann zeta function (5.5), and $\varphi(n)$ is the Euler totient function (5.14). For product weights, the algorithm minimizes $[e_{n,s}^{\mathrm{sh}}(\boldsymbol{z})]^2$ step by step as usual. For general non-product weights, the algorithm minimizes an auxiliary quantity $\widetilde{e}_{n,s,d}^2(\boldsymbol{z})$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

depending on s,

$$\widetilde{e}^2_{n,s,d}(\boldsymbol{z}) := \sum_{\emptyset\neq\mathfrak{v}\subseteq\{1:d\}} \widetilde{\gamma}_{s,\mathfrak{v}} \left(\frac{1}{n}\sum_{k=0}^{n-1}\prod_{j\in\mathfrak{v}} B_2\left(\left\{\frac{kz_j}{n}\right\}\right)\right), \tag{5.21}$$

step by step for each $d = 2, 3, \ldots, s$, with auxiliary weights defined by

$$\widetilde{\gamma}_{s,\mathfrak{v}} := \sum_{\mathfrak{v}\subseteq\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}\, \beta^{|\mathfrak{u}|-|\mathfrak{v}|}, \quad \mathfrak{v}\subseteq\{1:s\}. \tag{5.22}$$

Note that the latter algorithm is not extensible in s.

Proof. In the previous proof we made use of the Fourier expansion of B_2, which has no constant term. To allow us to use essentially the same argument here, we start from (5.12), adapting the identity (4.2) and swapping the order of sums, to obtain

$$\begin{aligned}
[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2 &= \frac{1}{n}\sum_{k=0}^{n-1}\sum_{\mathfrak{u}\subseteq\{1:s\}}\gamma_{\mathfrak{u}}\sum_{\mathfrak{v}\subseteq\mathfrak{u}}\beta^{|\mathfrak{u}|-|\mathfrak{v}|}\prod_{j\in\mathfrak{v}}B_2\left(\left\{\frac{kz_j}{n}\right\}\right) - \sum_{\mathfrak{u}\subseteq\{1:s\}}\gamma_{\mathfrak{u}}\beta^{|\mathfrak{u}|}\\
&= \frac{1}{n}\sum_{k=0}^{n-1}\sum_{\mathfrak{v}\subseteq\{1:s\}}\sum_{\mathfrak{v}\subseteq\mathfrak{u}\subseteq\{1:s\}}\gamma_{\mathfrak{u}}\beta^{|\mathfrak{u}|-|\mathfrak{v}|}\prod_{j\in\mathfrak{v}}B_2\left(\left\{\frac{kz_j}{n}\right\}\right) - \sum_{\mathfrak{u}\subseteq\{1:s\}}\gamma_{\mathfrak{u}}\beta^{|\mathfrak{u}|}\\
&= \frac{1}{n}\sum_{k=0}^{n-1}\sum_{\mathfrak{v}\subseteq\{1:s\}}\widetilde{\gamma}_{s,\mathfrak{v}}\prod_{j\in\mathfrak{v}}B_2\left(\left\{\frac{kz_j}{n}\right\}\right) - \widetilde{\gamma}_{s,\emptyset}\\
&= \sum_{\emptyset\neq\mathfrak{v}\subseteq\{1:s\}}\widetilde{\gamma}_{s,\mathfrak{v}}\left(\frac{1}{n}\sum_{k=0}^{n-1}\prod_{j\in\mathfrak{v}}B_2\left(\left\{\frac{kz_j}{n}\right\}\right)\right),
\end{aligned} \tag{5.23}$$

where we introduced the auxiliary weights defined in (5.22).

This last expression takes the same form as the squared shift-averaged worst-case error in the unanchored space. However, there is a vital difference that the auxiliary weights $\widetilde{\gamma}_{s,\mathfrak{v}}$ depend on the dimension s. Consequently, a recursive formula of the form (5.19) does not hold.

Case 1. Consider first the case of product weights $\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}}\gamma_j$. It follows from the definition (5.22) that, for $s\notin\mathfrak{v}$, we have

$$\widetilde{\gamma}_{s,\mathfrak{v}} = \sum_{\mathfrak{v}\subseteq\mathfrak{u}\subseteq\{1:s-1\}}\gamma_{\mathfrak{u}}\,\beta^{|\mathfrak{u}|-|\mathfrak{v}|} + \sum_{\mathfrak{v}\subseteq\mathfrak{u}\subseteq\{1:s-1\}}\gamma_{\mathfrak{u}}\gamma_s\,\beta^{|\mathfrak{u}|-|\mathfrak{v}|+1} = (1+\gamma_s\beta)\widetilde{\gamma}_{s-1,\mathfrak{v}}.$$

Thus, by separating the terms depending on whether or not the element s is included in the set $\mathfrak{v}$, we obtain

$$[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2 = (1+\gamma_s\beta)[e^{\mathrm{sh}}_{n,s-1}(z_1,\ldots,z_{s-1})]^2 + \theta(z_s),$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

where

$$\theta(z_s) := \sum_{s\in\mathfrak{v}\subseteq\{1:s\}} \widetilde{\gamma}_{s,\mathfrak{v}} \left(\frac{1}{n} \sum_{k=0}^{n-1} \prod_{j\in\mathfrak{v}} B_2\left(\left\{\frac{kz_j}{n}\right\}\right)\right).$$

The induction hypothesis is now

$$[e^{\mathrm{sh}}_{n,s-1}(z_1,\ldots,z_{s-1})]^2 \le \left(\frac{1}{\varphi(n)} \sum_{\emptyset\ne\mathfrak{v}\subseteq\{1:s-1\}} \widetilde{\gamma}^\lambda_{s-1,\mathfrak{v}} \left(\frac{2\zeta(2\lambda)}{(2\pi^2)^\lambda}\right)^{|\mathfrak{v}|}\right)^{1/\lambda}$$

for all $\lambda \in (1/2, 1]$. For the quantity $\theta(z_s)$ we follow the averaging argument in the previous proof, to obtain

$$[\theta(z_s^*)]^\lambda \le \frac{1}{\varphi(n)} \sum_{s\in\mathfrak{v}\subseteq\{1:s\}} \widetilde{\gamma}^\lambda_{s,\mathfrak{v}} \left(\frac{2\zeta(2\lambda)}{(2\pi^2)^\lambda}\right)^{|\mathfrak{v}|}.$$

Combining the estimates, we obtain

$$\begin{aligned}[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2 \le (1+\gamma_s\beta) &\left(\frac{1}{\varphi(n)} \sum_{\emptyset\ne\mathfrak{v}\subseteq\{1:s-1\}} \widetilde{\gamma}^\lambda_{s-1,\mathfrak{v}} \left(\frac{2\zeta(2\lambda)}{(2\pi^2)^\lambda}\right)^{|\mathfrak{v}|}\right)^{1/\lambda} \\ &+ \left(\frac{1}{\varphi(n)} \sum_{s\in\mathfrak{v}\subseteq\{1:s\}} \widetilde{\gamma}^\lambda_{s,\mathfrak{v}} \left(\frac{2\zeta(2\lambda)}{(2\pi^2)^\lambda}\right)^{|\mathfrak{v}|}\right)^{1/\lambda}.\end{aligned}$$

Applying (5.20) then yields

$$[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2 \le \left(\frac{1}{\varphi(n)} \sum_{\emptyset\ne\mathfrak{v}\subseteq\{1:s\}} \widetilde{\gamma}^\lambda_{s,\mathfrak{v}} \left(\frac{2\zeta(2\lambda)}{(2\pi^2)^\lambda}\right)^{|\mathfrak{v}|}\right)^{1/\lambda}. \tag{5.24}$$

Finally we express the result in terms of the original weights. Using (5.22) and (5.20), we have

$$\begin{aligned}&\sum_{\emptyset\ne\mathfrak{v}\subseteq\{1:s\}} \widetilde{\gamma}^\lambda_{s,\mathfrak{v}} \left(\frac{2\zeta(2\lambda)}{(2\pi^2)^\lambda}\right)^{|\mathfrak{v}|} \\ &\quad\le \sum_{\mathfrak{v}\subseteq\{1:s\}} \sum_{\mathfrak{v}\subseteq\mathfrak{u}\subseteq\{1:s\}} \gamma^\lambda_{\mathfrak{u}}\, \beta^{(|\mathfrak{u}|-|\mathfrak{v}|)\lambda} \left(\frac{2\zeta(2\lambda)}{(2\pi^2)^\lambda}\right)^{|\mathfrak{v}|} - \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma^\lambda_{\mathfrak{u}}\, \beta^{|\mathfrak{u}|\lambda} \\ &\quad= \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma^\lambda_{\mathfrak{u}} \sum_{\mathfrak{v}\subseteq\mathfrak{u}} \beta^{(|\mathfrak{u}|-|\mathfrak{v}|)\lambda} \left(\frac{2\zeta(2\lambda)}{(2\pi^2)^\lambda}\right)^{|\mathfrak{v}|} - \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma^\lambda_{\mathfrak{u}}\, \beta^{|\mathfrak{u}|\lambda}\end{aligned}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

$$= \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}^{\lambda} \left(\frac{2\zeta(2\lambda)}{(2\pi^2)^{\lambda}} + \beta^{\lambda}\right)^{|\mathfrak{u}|} - \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}^{\lambda}\, \beta^{|\mathfrak{u}|\lambda}$$

$$\le \sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}^{\lambda} \left(\frac{2\zeta(2\lambda)}{(2\pi^2)^{\lambda}} + \beta^{\lambda}\right)^{|\mathfrak{u}|}. \tag{5.25}$$

That the bound holds for $s = 1$ follows as before, completing the proof for the case of product weights.

Remark. For the case of product weights, the use of auxiliary weights is only needed in the proof and has no bearing on the actual CBC construction. It is also possible to prove the result by working directly with the simplified worst-case error expression for product weights (see (5.13)) without introducing auxiliary weights. We have chosen to prove the result this way to provide valuable insights into the more complicated case of general non-product weights.

Case 2. Now we consider the case of general non-product weights. Due to the lack of structure in general weights, we are unable to relate $[e_{n,s}^{\mathrm{sh}}(\boldsymbol{z})]^2$ to $[e_{n,s-1}^{\mathrm{sh}}(z_1,\dots,z_{s-1})]^2$ in a meaningful way. We therefore switch to a modified CBC construction based on the auxiliary quantity (5.21). Note that the auxiliary weights depend only on the final dimension s and not on the induction index d in (5.21). This allows us to write

$$\widetilde{e}_{n,s,d}^{\,2}(z_1,\dots,z_d) = \widetilde{e}_{n,s,d-1}^{\,2}(z_1,\dots,z_{d-1}) + \widetilde{\theta}(z_d),$$

where

$$\widetilde{\theta}(z_d) := \sum_{d\in\mathfrak{v}\subseteq\{1:d\}} \widetilde{\gamma}_{s,\mathfrak{v}} \left(\frac{1}{n}\sum_{k=0}^{n-1}\prod_{j\in\mathfrak{v}} B_2\left(\left\{\frac{kz_j}{n}\right\}\right)\right).$$

Note that

$$\widetilde{e}_{n,s,s}^{\,2}(\boldsymbol{z}) = e_{n,s}^2(\boldsymbol{z}), \quad \text{but} \quad \widetilde{e}_{n,s,d}^{\,2}(z_1,\dots,z_d) \ne e_{n,d}^2(z_1,\dots,z_d) \quad \text{for } d < s.$$

Hence the CBC construction based on the auxiliary quantity is *not* the same as the original CBC construction. We prove by induction that the CBC construction based on the auxiliary quantity yields, for each $d = 1, 2, \dots, s$,

$$\widetilde{e}_{n,s,d}^{\,2}(z_1,\dots,z_d) \le \left(\frac{1}{\varphi(n)} \sum_{\emptyset\ne\mathfrak{v}\subseteq\{1:d\}} \widetilde{\gamma}_{s,\mathfrak{v}}^{\lambda} \left(\frac{2\zeta(2\lambda)}{(2\pi^2)^{\lambda}}\right)^{|\mathfrak{v}|}\right)^{1/\lambda}$$

for all $\lambda \in (1/2, 1]$. Since the auxiliary weights do not change with the induction index d, the proof is essentially the same as the previous proof for the unanchored space. At the final step $d = s$ we recover the error bound (5.24), which can be expressed in terms of the original weights following (5.25). This completes the proof. □

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

The following theorem illustrates how we can apply the theory of this section to a given integration problem.

Theorem 5.10 (CBC integration error). Suppose that a given integrand f belongs to either our weighted anchored or unanchored Sobolev space. A lattice rule generating vector $\boldsymbol{z} \in \mathbb{U}_n^s$ can be constructed by a CBC algorithm such that, for all $\lambda \in (1/2, 1]$,

$$\sqrt{\mathbb{E}\,|I_s(f) - Q_{n,s}(\cdot; f)|^2} \le \left(\frac{1}{\varphi(n)} \sum_{\emptyset \ne \mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}}^{\lambda} \left(\frac{2\zeta(2\lambda)}{(2\pi^2)^{\lambda}} + \beta^{\lambda} \right)^{|\mathfrak{u}|} \right)^{1/(2\lambda)} \|f\|_{s,\boldsymbol{\gamma}},$$

where the expectation is taken with respect to the random shift which is uniformly distributed over $[0,1]^s$, $\beta = c^2 - c + 1/3$ for the anchored variant with anchor c and $\beta = 0$ for the unanchored variant, $\zeta(\cdot)$ is the Riemann zeta function (5.5), and $\varphi(n)$ is the Euler totient function (5.14).

We conclude this subsection with the remark that a different analysis can be used for the anchored Sobolev space if, roughly speaking, the auxiliary weights (5.22) are dominated by the original weights, that is, if

$$\sum_{\substack{|\mathfrak{u}|<\infty \\ \mathfrak{v} \subseteq \mathfrak{u} \subset \mathbb{N}}} \gamma_{\mathfrak{u}} \beta^{|\mathfrak{u}|-|\mathfrak{v}|} \le C \gamma_{\mathfrak{v}}$$

for all finite subsets $\mathfrak{v} \subset \mathbb{N}$, where $C > 0$ is independent of $\mathfrak{v}$. A similar result holds if the weights are of the following special POD form:

$$\gamma_{\mathfrak{u}} = ((|\mathfrak{u}| + \ell)!)^a \prod_{j \in \mathfrak{u}} \gamma_j,$$

where ℓ is a non-negative integer and a is a positive real number. We explain the alternative analysis for this case below.

Recall that for the anchored space the CBC algorithm must work with auxiliary weights (5.22) which do not preserve the POD structure of the original weights. We can write

$$\begin{aligned}
\widetilde{\gamma}_{s,\mathfrak{v}} &= \sum_{\mathfrak{v} \subseteq \mathfrak{u} \subseteq \{1:s\}} ((|\mathfrak{u}| + \ell)!)^a \prod_{j \in \mathfrak{u}} \gamma_j \\
&= ((|\mathfrak{v}| + \ell)!)^a \left(\prod_{j \in \mathfrak{v}} \gamma_j \right) \sum_{\mathfrak{v} \subseteq \mathfrak{u} \subseteq \{1:s\}} \left(\frac{(|\mathfrak{u}| + \ell)!}{(|\mathfrak{v}| + \ell)!} \right)^a \prod_{j \in \mathfrak{u} \setminus \mathfrak{v}} \gamma_j \\
&= ((|\mathfrak{v}| + \ell)!)^a \left(\prod_{j \in \mathfrak{v}} \gamma_j \right) \sum_{\mathfrak{w} \subseteq \{1:s\} \setminus \mathfrak{v}} \left(\frac{(|\mathfrak{v}| + |\mathfrak{w}| + \ell)!}{(|\mathfrak{v}| + \ell)!} \right)^a \prod_{j \in \mathfrak{w}} \gamma_j.
\end{aligned}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Using the simple inequality

$$\frac{(p+q)!}{p!\,q!} = \binom{p+q}{p} \le \sum_{k=0}^{p+q} \binom{p+q}{k} = 2^{p+q},$$

we obtain

$$\begin{aligned}\widetilde{\gamma}_{s,\mathfrak{v}} &\le ((|\mathfrak{v}|+\ell)!)^a \Bigg(\prod_{j\in\mathfrak{v}} \gamma_j\Bigg) \sum_{\mathfrak{w}\subseteq\{1:s\}\setminus\mathfrak{v}} \big(|\mathfrak{w}|!\, 2^{|\mathfrak{v}|+|\mathfrak{w}|+\ell}\big)^a \prod_{j\in\mathfrak{w}} \gamma_j \\ &\le ((|\mathfrak{v}|+\ell)!)^a \Bigg(\prod_{j\in\mathfrak{v}} (2^a\gamma_j)\Bigg) 2^{a\ell} \sum_{\mathfrak{w}\subseteq\{1:s\}} (|\mathfrak{w}|!)^a \prod_{j\in\mathfrak{w}} (2^a\gamma_j) = c_{s,\boldsymbol{\gamma}}\, \widetilde{\widetilde{\gamma}}_{\mathfrak{v}},\end{aligned}$$

where

$$\widetilde{\widetilde{\gamma}}_{\mathfrak{v}} := ((|\mathfrak{v}|+\ell)!)^a \prod_{j\in\mathfrak{v}} (2^a\gamma_j), \quad \text{and} \quad c_{s,\boldsymbol{\gamma}} := 2^{a\ell} \sum_{\mathfrak{w}\subseteq\{1:s\}} (|\mathfrak{w}|!)^a \prod_{j\in\mathfrak{w}} (2^a\gamma_j).$$

Thus from (5.23) we can write

$$[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2 \le c_{s,\boldsymbol{\gamma}} \sum_{\emptyset\ne\mathfrak{v}\subseteq\{1:s\}} \widetilde{\widetilde{\gamma}}_{\mathfrak{v}} \left(\frac{1}{n}\sum_{k=0}^{n-1} \prod_{j\in\mathfrak{v}} B_2\Big(\Big\{\frac{kz_j}{n}\Big\}\Big)\right). \tag{5.26}$$

Hence, we can use the expression on the right-hand side of (5.26) (without the factor $c_{s,\boldsymbol{\gamma}}$) as the search criterion for the CBC construction. The benefit is that the new weights $\widetilde{\widetilde{\gamma}}_{\mathfrak{v}}$ are also of the POD form, and they do *not* depend on the dimension s. This means that a fast implementation is possible (see Section 5.6) and that the algorithm is extensible in s. The resulting error bound would be slightly worse (each factor in the product part of the weights would be scaled by 2^a, and there is an additional factor of $c_{s,\boldsymbol{\gamma}}^{1/2}$ in the overall error bound in Theorem 5.10), but the convergence rate and the dependence of the implied constant on the dimension s remain the same as before.

5.5. Fast CBC construction for product weights

In this subsection we explain how to implement the CBC construction efficiently for product weights. We begin by presenting a naive implementation of Algorithm 5.6 as a pseudocode (Pseudocode 1, overleaf). Then we discuss the techniques to speed up the computation.

The reader should keep in mind that the general approach described in this subsection applies to all shift-invariant kernels: we can replace $B_2(x)$ by any generic function whose integral over $[0,1]$ is 0.

Recall that

$$\begin{aligned}\mathbb{Z}_n &:= \{z \in \mathbb{Z} : 0 \le z \le n-1\},\\ \mathbb{U}_n &:= \{z \in \mathbb{Z}_n : \gcd(z,n) = 1\},\end{aligned}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Pseudocode 1 (CBC: naive implementation)

for s **from** 1 **to** $s_{\max}$ **do**
  for all $z_s \in \mathbb{U}_n$ **do**

$$e_s^2(z_s) = -\prod_{j=1}^{s}(1+\gamma_j\beta) + \frac{1}{n}\sum_{k=0}^{n-1}\prod_{j=1}^{s}\left(1+\gamma_j\left[B_2\left(\left\{\frac{kz_j}{n}\right\}\right)+\beta\right]\right)$$

  end for
  $z_s = \operatorname{argmin}_{z\in\mathbb{U}_n} e_s^2(z)$
end for

with $\varphi(n) := |\mathbb{U}_n|$. In the pseudocode we used the simplified notation

$$e_s^2(z_s) := [e_{n,s}^{\mathrm{sh}}(z_1,\ldots,z_s)]^2$$

to highlight the fact that z_s is the only component to be determined at step s; all previous components $z_1,\ldots,z_{s-1}$ have already been fixed. To simplify the exposition, we included z_1 in the search as well, even though all choices in the first dimension are equivalent.

The computational cost of this naive implementation is $O(n^2\, s_{\max}^2)$ operations. There are a number of ways to reduce this cost. Firstly, due to the symmetry $B_2(x) = B_2(1-x)$, we have $e_s^2(z_s) = e_s^2(n-z_s)$, thus we only need to search through half of the elements in $\mathbb{U}_n$. Secondly, at step s we can write

$$e_s^2(z_s) = (1+\gamma_s\beta)\, e_{s-1}^2 + \frac{\gamma_s}{n}\sum_{k=0}^{n-1}\underbrace{\left[B_2\left(\left\{\frac{kz_s}{n}\right\}\right)\right]}_{\Omega_n(z_s,k)}\underbrace{\prod_{j=1}^{s-1}\left(1+\gamma_j\left[B_2\left(\left\{\frac{kz_j}{n}\right\}\right)+\beta\right]\right)}_{p_{s-1}(k)}. \tag{5.27}$$

The n values $p_{s-1}(k)$ do not depend on z_s, and can be stored during the search. This reduces the construction cost to $O(n^2\, s_{\max})$ operations, at the expense of $O(n)$ storage.

The values of (5.27) for all $z_s \in \mathbb{U}_n$ can be expressed as a $\varphi(n)\times 1$ vector

$$\boldsymbol{e}_s^2 = [e_s^2(z_s)]_{z_s\in\mathbb{U}_n},$$

which can be computed in terms of a matrix–vector product, with the $\varphi(n)\times n$ matrix

$$\boldsymbol{\Omega}_n = [\Omega_n(z,k)]_{\substack{z\in\mathbb{U}_n\\ k\in\mathbb{Z}_n}} = \left[B_2\left(\frac{kz \bmod n}{n}\right)\right]_{\substack{z\in\mathbb{U}_n\\ k\in\mathbb{Z}_n}}, \tag{5.28}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Pseudocode 2 (CBC: matrix–vector form)

$\boldsymbol{p}_0 = \mathbf{1}$
$e_0^2 = 0$
for s **from** 1 **to** $s_{\max}$ **do**
  $\boldsymbol{e}_s^2 = (\mathbf{1} + \gamma_s \boldsymbol{\beta}) e_{s-1}^2 + \dfrac{\gamma_s}{n} \boldsymbol{\Omega}_n \boldsymbol{p}_{s-1}$ ▷ compute
  $z_s = \operatorname{argmin}_{z \in \mathbb{U}_n} e_s^2(z)$ ▷ select
  $e_s^2 = e_s^2(z_s)$ ▷ set
  $\boldsymbol{p}_s = \big(\mathbf{1} + \gamma_s(\boldsymbol{\Omega}_n(z_s, :) + \boldsymbol{\beta})\big) .* \boldsymbol{p}_{s-1}$ ▷ update
end for

and the $n \times 1$ vector

$$\boldsymbol{p}_{s-1} = [p_{s-1}(k)]_{k \in \mathbb{Z}_n}.$$

This yields our second pseudocode. Here $\boldsymbol{\Omega}_n(z_s, :)$ means taking a particular row of the matrix, while $.*$ means element-wise vector multiplication. Also, $\mathbf{1} = (1, \ldots, 1)$ and $\boldsymbol{\beta} = (\beta, \ldots, \beta)$ are vectors with the appropriate length.

At the update stage, the vector $\boldsymbol{p}_{s-1}$ can be overwritten by the vector $\boldsymbol{p}_s$. Thus the memory requirement is of order $O(n)$.

The trick now is to order the indices $z \in \mathbb{U}_n$ and $k \in \mathbb{Z}_n$ in (5.28) in a clever way in order to allow fast matrix–vector multiplication. Below we explain how this is achieved for the simpler case when n is prime.

When n is prime and using the *Rader factorization*, the matrix $\boldsymbol{\Omega}_n$ can be reordered such that it has a *circulant submatrix* $\boldsymbol{C}_n$ of size $(n-1) \times (n-1)$. This is achieved by taking the indices in the order

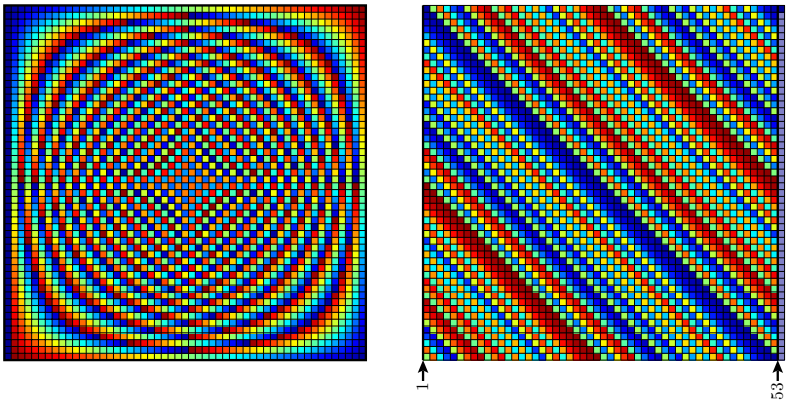
$$z = g^i \quad \text{and} \quad k = (g^{-1})^{i'} \quad \text{for } 0 \le i, i' \le n-2,$$

where g is a primitive root of n (*i.e.*, g is a generator for the cyclic group $\mathbb{U}_n$) and g^{-1} denotes its multiplicative inverse. We denote this particular ordering of $\boldsymbol{\Omega}_n$ by $\boldsymbol{\Omega}_n^{\langle g \rangle}$. Thus, for n prime, the element $\Omega_n^{\langle g \rangle}(i, i')$ in the $(i+1)$th row and $(i'+1)$th column of $\boldsymbol{\Omega}_n^{\langle g \rangle}$ is given by

$$\Omega_n^{\langle g \rangle}(i, i') = \begin{cases} \Omega_n(g^i, (g^{-1})^{i'}) & \text{if } 0 \le i, i' \le n-2, \\ 0 & \text{if } i' = n-1, \end{cases}$$

where $g^i, (g^{-1})^{i'} \in \mathbb{U}_n$ for $0 \le i, i' \le n-2$. A graphical illustration of the effect of reordering the indices is given in Figure 5.1. We present a concrete illustration of the reordering in the next example.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press



(a) (b)

Figure 5.1. (Image by Dirk Nuyens.) The structure of the matrix Ω_{53} for the fast CBC construction of lattice rules. (a) Original ordering of indices, and (b) after a reordering of indices.

Example 5.11. Take $n = 11$. Then $\mathbb{U}_{11} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. The natural ordering of the indices $z \in \mathbb{U}_{11}$ and $k \in \mathbb{Z}_{11}$ gives the 10×11 matrix

$$\boldsymbol{\Omega}_{11} = B_2 \left(\frac{1}{11} \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 0 & 2 & 4 & 6 & 8 & 10 & 1 & 3 & 5 & 7 & 9 \\ 0 & 3 & 6 & 9 & 1 & 4 & 7 & 10 & 2 & 5 & 8 \\ 0 & 4 & 8 & 1 & 5 & 9 & 2 & 6 & 10 & 3 & 7 \\ 0 & 5 & 10 & 4 & 9 & 3 & 8 & 2 & 7 & 1 & 6 \\ 0 & 6 & 1 & 7 & 2 & 8 & 3 & 9 & 4 & 10 & 5 \\ 0 & 7 & 3 & 10 & 6 & 2 & 9 & 5 & 1 & 8 & 4 \\ 0 & 8 & 5 & 2 & 10 & 7 & 4 & 1 & 9 & 6 & 3 \\ 0 & 9 & 7 & 5 & 3 & 1 & 10 & 8 & 6 & 4 & 2 \\ 0 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{bmatrix} \right).$$

The generator for the group $\mathbb{U}_{11}$ can be taken to be $g = 2$. The ordering for the z indices is then

$$[g^i : 0 \le i \le 9] = [1, 2, 4, 8, 5, 10, 9, 7, 3, 6].$$

For the ordering of the indices $k \ne 0$ we take the powers of $g^{-1} = 6$,

$$[(g^{-1})^{i'} : 0 \le i' \le 9] = [1, 6, 3, 7, 9, 10, 5, 8, 4, 2],$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

which, leaving aside the index 1, is the reverse order of the z indices. This reordering, together with a column of zeros for $k = 0$, gives the matrix

$$\boldsymbol{\Omega}_{11}^{\langle 2\rangle} = B_2\left(\frac{1}{11}\left[\begin{array}{cccccccccc|c} 1 & 6 & 3 & 7 & 9 & 10 & 5 & 8 & 4 & 2 & 0 \\ 2 & 1 & 6 & 3 & 7 & 9 & 10 & 5 & 8 & 4 & 0 \\ 4 & 2 & 1 & 6 & 3 & 7 & 9 & 10 & 5 & 8 & 0 \\ 8 & 4 & 2 & 1 & 6 & 3 & 7 & 9 & 10 & 5 & 0 \\ 5 & 8 & 4 & 2 & 1 & 6 & 3 & 7 & 9 & 10 & 0 \\ 10 & 5 & 8 & 4 & 2 & 1 & 6 & 3 & 7 & 9 & 0 \\ 9 & 10 & 5 & 8 & 4 & 2 & 1 & 6 & 3 & 7 & 0 \\ 7 & 9 & 10 & 5 & 8 & 4 & 2 & 1 & 6 & 3 & 0 \\ 3 & 7 & 9 & 10 & 5 & 8 & 4 & 2 & 1 & 6 & 0 \\ 6 & 3 & 7 & 9 & 10 & 5 & 8 & 4 & 2 & 1 & 0 \end{array}\right]\right), \quad \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\boldsymbol{C}_{11}}$$

where the 10×10 submatrix $\boldsymbol{C}_{11}$ is circulant.

We can halve the number of rows and the number of columns of the circulant submatrix $\boldsymbol{C}_{11}$ due to the symmetry $B_2(x) = B_2(1-x)$, to obtain the reduced form (denoted by a $\sim$ instead of $=$)

$$\boldsymbol{\Omega}_{11}^{\langle 2\rangle} \sim B_2\left(\frac{1}{11}\left[\begin{array}{ccccc\|\|c} 1 & 6 & 3 & 7 & 9 & 0 \\ 2 & 1 & 6 & 3 & 7 & 0 \\ 4 & 2 & 1 & 6 & 3 & 0 \\ 8 & 4 & 2 & 1 & 6 & 0 \\ 5 & 8 & 4 & 2 & 1 & 0 \end{array}\right]\right), \quad \underbrace{\qquad\qquad\qquad\qquad\qquad}_{\widetilde{\boldsymbol{C}}_{11}}$$

where we remember that the full matrix $\boldsymbol{\Omega}_{11}^{\langle 2\rangle}$ can be obtained by doubling the 5×5 circulant submatrix $\widetilde{\boldsymbol{C}}_{11}$ vertically and horizontally up to the double bar. This concludes our simple example.

If we express the reordering of indices induced by g as permutations Π_g^{T} and $\Pi_{g^{-1}}$ on the rows and columns of the matrix $\boldsymbol{\Omega}_n$, then we obtain

$$\boldsymbol{y} = \boldsymbol{\Omega}_n\, \boldsymbol{p}_{s-1} \iff \Pi_g^{\mathsf{T}}\, \boldsymbol{y} = (\Pi_g^{\mathsf{T}}\, \boldsymbol{\Omega}_n\, \Pi_{g^{-1}})\, (\Pi_{g^{-1}}^{\mathsf{T}}\, \boldsymbol{p}_{s-1}) = \boldsymbol{\Omega}_n^{\langle g\rangle}\, \boldsymbol{p}_{s-1}^{\langle g^{-1}\rangle}.$$

Since the initialization of $\boldsymbol{p}_0 = \mathbf{1}$ is unaffected by any permutation, we only need to work with the permuted vector throughout. We just need to make sure that the correct index z_s is chosen at the select stage.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

A matrix–vector multiplication with a circulant matrix $\boldsymbol{C}_m$ of size $m \times m$, with first column $\boldsymbol{c}$, can be obtained by

$$\begin{aligned}\boldsymbol{C}_m\,\boldsymbol{x} &= \boldsymbol{F}_m^{-1}\,\mathrm{diag}(\boldsymbol{F}_m\,\boldsymbol{c})\,\boldsymbol{F}_m\,\boldsymbol{x}\\ &= \mathrm{IFFT}(\mathrm{FFT}(\boldsymbol{c}).*\,\mathrm{FFT}(\boldsymbol{x})),\end{aligned}$$

where

$$\boldsymbol{F}_m = [\exp(-2\pi\mathrm{i}\,p\,q/m)]_{0\le p,q\le m-1}$$

is the Fourier matrix of size $m \times m$, and

$$\boldsymbol{F}_m^{-1} = \frac{1}{m}[\exp(2\pi\mathrm{i}\,p\,q/m)]_{0\le p,q\le m-1}.$$

Using FFT and IFFT, this matrix–vector multiplication takes $O(m \log m)$ operations instead of the usual $O(m^2)$, and uses $O(m)$ memory.

Using permutation together with FFT, the cost for the CBC construction can be reduced to $O(n \log n\, s_{\max})$ operations, at the expense of $O(n)$ storage. This approach is called the *fast CBC construction*, and is due to Nuyens and Cools (2006*a*, 2006*b*).

For composite n, the general idea is that the complete matrix $\boldsymbol{\Omega}_n$ can be partitioned in blocks which have a circulant or block-circulant structure.

We summarize the fast CBC algorithm in the pseudocode below. We stress again that the key point is to use FFT to compute the matrix–vector product $\boldsymbol{\Omega}_n^{\langle g\rangle}\,\boldsymbol{p}_{s-1}$, taking care to select the correct z_s under the corresponding permuted set of indices.

Pseudocode 3 (Fast CBC: matrix–vector form with permuted matrix)

$\boldsymbol{p}_0 = \boldsymbol{1}$
$e_0^2 = 0$
for s **from** 1 **to** $s_{\max}$ **do**
  $\boldsymbol{e}_s^{2\langle g\rangle} = (\boldsymbol{1} + \gamma_s\boldsymbol{\beta})e_{s-1}^2 + \dfrac{\gamma_s}{n}\,\boldsymbol{\Omega}_n^{\langle g\rangle}\,\boldsymbol{p}_{s-1}$ ▷ compute – use FFT
  $z_s = \mathrm{argmin}_{z\in\mathbb{U}_n} e_s^2(z)$ ▷ select – pick the correct index
  $e_s^2 = e_s^2(z_s)$ ▷ set
  $\boldsymbol{p}_s = \big(\boldsymbol{1} + \gamma_s(\boldsymbol{\Omega}_n^{\langle g\rangle}(z_s,:) + \boldsymbol{\beta})\big).*\,\boldsymbol{p}_{s-1}$ ▷ update
end for

5.6. *Fast CBC construction for POD weights*

Here we explain how the fast CBC construction can be extended to POD (product and order-dependent) weights (see (4.4)), covering order-dependent weights as a special case. For completely general weights $\gamma_{\mathfrak{u}}$ the cost for evaluating $[e_{n,s}^{\mathrm{sh}}(\boldsymbol{z})]^2$ in (5.12) is prohibitively expensive, whereas for

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

POD weights their special structure enables us to compute $[e^{\mathrm{sh}}_{n,s}(\boldsymbol{z})]^2$ using a recursive argument starting from dimension one.

We restrict ourselves to the unanchored Sobolev space. (Recall that the CBC algorithm for the anchored Sobolev space minimizes an auxiliary quantity that depends on the auxiliary weights (5.22). The POD form of the original weights is not preserved under the auxiliary weights, thus the strategy to be described here is not applicable to the anchored space setting; see also the remark at the end of Section 5.4 for cases where fast CBC for POD weights can be used.) However, the approach can be generalized to other shift-invariant kernels if B_2 is replaced by another function that integrates to zero.

With POD weights $\gamma_{\mathfrak{u}} = \Gamma_{|\mathfrak{u}|} \prod_{j\in\mathfrak{u}} \gamma_j$, we can write the shift-averaged worst-case error at step s (using again the simplified notation of the previous subsection) as

$$\begin{aligned} e_s^2(z_s) &= \frac{1}{n}\sum_{k=0}^{n-1} \sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:s\}} \Gamma_{|\mathfrak{u}|} \prod_{j\in\mathfrak{u}} \left(\gamma_j\, B_2\left(\left\{\frac{kz_j}{n}\right\}\right)\right) \\ &= \frac{1}{n}\sum_{k=0}^{n-1} \sum_{\ell=1}^{s} \underbrace{\sum_{\substack{\mathfrak{u}\subseteq\{1:s\}\\ |\mathfrak{u}|=\ell}} \Gamma_\ell \prod_{j\in\mathfrak{u}} \left(\gamma_j\, B_2\left(\left\{\frac{kz_j}{n}\right\}\right)\right)}_{p_{s,\ell}(k)}. \end{aligned} \tag{5.29}$$

We split the sum over $\mathfrak{u}$ into two depending on whether $s \in \mathfrak{u}$, to obtain

$$\begin{aligned} e_s^2(z_s) = \frac{1}{n}\sum_{k=0}^{n-1} \sum_{\ell=1}^{s} \Bigg[& \underbrace{\sum_{\substack{\mathfrak{u}\subseteq\{1:s-1\}\\ |\mathfrak{u}|=\ell}} \Gamma_\ell \prod_{j\in\mathfrak{u}} \left(\gamma_j\, B_2\left(\left\{\frac{kz_j}{n}\right\}\right)\right)}_{p_{s-1,\ell}(k)} \\ & + \frac{\Gamma_\ell}{\Gamma_{\ell-1}}\,\gamma_s\, \underbrace{B_2\left(\left\{\frac{kz_s}{n}\right\}\right)}_{\Omega_n(z_s,k)} \underbrace{\sum_{\substack{\mathfrak{u}\subseteq\{1:s-1\}\\ |\mathfrak{u}|=\ell-1}} \Gamma_{\ell-1} \prod_{j\in\mathfrak{u}} \left(\gamma_j\, B_2\left(\left\{\frac{kz_j}{n}\right\}\right)\right)}_{p_{s-1,\ell-1}(k)} \Bigg] \\ = e_{s-1}^2 + \frac{\gamma_s}{n}\sum_{k=0}^{n-1} \Omega_n(z_s,k) & \left(\sum_{\ell=1}^{s} \frac{\Gamma_\ell}{\Gamma_{\ell-1}}\, p_{s-1,\ell-1}(k)\right). \end{aligned} \tag{5.30}$$

The computation of (5.30) for all $z_s \in \mathbb{U}_n$ can be expressed in matrix–vector form as

$$\boldsymbol{e}_s^2 = e_{s-1}^2\mathbf{1} + \frac{\gamma_s}{n}\boldsymbol{\Omega}_n\left(\sum_{\ell=1}^{s} \frac{\Gamma_\ell}{\Gamma_{\ell-1}}\, \boldsymbol{p}_{s-1,\ell-1}\right),$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

with the update formula

$$\boldsymbol{p}_{s,\ell} = \boldsymbol{p}_{s-1,\ell} + \frac{\Gamma_\ell}{\Gamma_{\ell-1}} \gamma_s \, \boldsymbol{\Omega}_n(z_s, :) .* \, \boldsymbol{p}_{s-1,\ell-1} \quad \text{for } \ell = 1, \ldots, s,$$

starting from

$$\boldsymbol{p}_{s,0} = \boldsymbol{1} \quad \text{and} \quad \boldsymbol{p}_{s,\ell} = \boldsymbol{0} \quad \text{for all } s \geq 1 \text{ and } \ell > s.$$

We need to store the vectors $\boldsymbol{p}_{s,\ell}$ for $\ell = 1, \ldots, s$, each of length n. These s vectors correspond to the s possible values of $|\mathfrak{u}|$ for $\emptyset \neq \mathfrak{u} \subseteq \{1:s\}$. These vectors can be overwritten for dimension $s+1$. Thus we require $O(n\, s_{\max})$ storage overall. The update cost for dimension s is $O(n\, s)$ operations.

If we permute the rows and columns of the matrix $\boldsymbol{\Omega}_n$ and use FFT to carry out the matrix–vector multiplication as in the case of product weights, then the search cost for each dimension is $O(n \log n)$ operations. The overall cost, combining search and update in all dimensions up to $s_{\max}$, is

$$O(n \log n \, s_{\max} + n \, s_{\max}^2)$$

operations.

If the weights are of *finite order* q, that is, $\Gamma_\ell = 0$ for all $\ell > q$, then the update cost for every dimension is capped at $O(n\, q)$ operations, with $O(n\, q)$ memory requirement, and the overall cost combining search and update in all dimensions up to $s_{\max}$ is reduced to

$$O(n \log n \, s_{\max} + n \, q \, s_{\max})$$

operations.

The reader may question why we did not leave out the factor Γ_ℓ from the definition of $p_{s,\ell}(k)$ in (5.29). Indeed, the expression (5.30) and the update formula would have been simpler if we had defined $p_{s,\ell}(k)$ that way. The problem is that for certain POD weights we may have Γ_ℓ growing very quickly with increasing ℓ and γ_j decaying with increasing j: for example,

$$\gamma_{\mathfrak{u}} = (|\mathfrak{u}|!)^{4/3} \prod_{j \in \mathfrak{u}} j^{-2.1}.$$

For weights of this kind, the alternative approach would be numerically unstable. The approach we outlined here works with the ratio $\Gamma_\ell / \Gamma_{\ell-1}$, thus avoiding the potential problem of overflow when working with Γ_ℓ.

5.7. Extensible lattice sequences

The CBC construction yields a lattice rule which is extensible in dimension, but not extensible in the number of points. Recall that the formula for obtaining the ith point of an n-point lattice rule with generating vector $\boldsymbol{z}$ is

$$\boldsymbol{t}_i = \left\{ \frac{i}{n} \boldsymbol{z} \right\}. \tag{5.31}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

In an *extensible lattice sequence* we take $\boldsymbol{z}$ to be a vector of b-adic numbers $\sum_{r=0}^{\infty} a_r b^r$, where $a_r \in \{0, 1, \ldots, b-1\}$, and allow n to increase. In an extensible lattice sequence with base $b \geq 2$, the formula (5.31) is changed to

$$\boldsymbol{t}_i = \{\phi_b(i)\, \boldsymbol{z}\}, \tag{5.32}$$

where $\phi_b(\cdot)$ is the radical inverse function in base b (or the *Gray code variant* in which the successive indices differ only in one b-ary digit). The formula (5.32) does not require you to know n in advance, and so in practice you can add more points to your lattice rule approximation until you are satisfied with the error.

When $n = b^m$ for any $m \geq 1$, the formulas (5.31) and (5.32) produce the same set of points, only the ordering of the points is different. Therefore, to obtain the points of an extensible lattice rule only at exact powers of the base, one can avoid the radical inverse function and still use the formula (5.31). For example, if we take $b = 2$ and denote by L_m the lattice point set with 2^m points, then we can illustrate the lattice sequence as follows:

$$\underbrace{\underbrace{\underbrace{\underbrace{\underbrace{\boldsymbol{t}_0 = \boldsymbol{0}}_{L_0},\ \underbrace{\boldsymbol{t}_1 = \tfrac{\boldsymbol{1}}{\boldsymbol{2}}}_{\Delta L_1}}_{L_1},\ \underbrace{\boldsymbol{t}_2, \boldsymbol{t}_3}_{\Delta L_2}}_{L_2},\ \underbrace{\boldsymbol{t}_4, \boldsymbol{t}_5, \boldsymbol{t}_6, \boldsymbol{t}_7}_{\Delta L_3}}_{L_3},\ \underbrace{\boldsymbol{t}_8, \boldsymbol{t}_9, \boldsymbol{t}_{10}, \boldsymbol{t}_{11}, \boldsymbol{t}_{12}, \boldsymbol{t}_{13}, \boldsymbol{t}_{14}, \boldsymbol{t}_{15}}_{\Delta L_4}}_{L_4},\ \ldots.$$

The number of points in L_m doubles as we move along the sequence, and the set of additional points $\Delta L_m = L_m \setminus L_{m-1}$ is generated by $\{i\boldsymbol{z}/2^m\}$ with i running through all the odd numbers up to $2^m - 1$. In Table 5.1 we show how the points are ordered under radical inverse and the Gray code variant.

Empirical searches for extensible lattice sequences were considered in Hickernell, Hong, L'Ecuyer and Lemieux (2000). Then it was proved by Hickernell and Niederreiter (2003) that good generating vectors exist for extensible lattice sequences, but the proof was non-constructive.

Cools, Kuo and Nuyens (2006) constructed *embedded lattice rules* that are good for n in a large practical range $b^{m_1} \leq n \leq b^{m_2}$. They used a modified CBC algorithm to construct a generating vector $\boldsymbol{z}$, minimizing as much as possible the search criterion

$$X_{m_1,m_2,s}(\boldsymbol{z}) := \max_{m_1 \leq m \leq m_2} \frac{e^{\mathrm{sh}}_{b^m,s}(\boldsymbol{z})}{e^{\mathrm{sh}}_{b^m,s}(\boldsymbol{z}^{(m)})},$$

where $e^{\mathrm{sh}}_{b^m,s}(\boldsymbol{z}^{(m)})$ denotes the shifted-averaged worst-case error for the generating vector obtained by the original CBC construction for b^m points,

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Table 5.1. Ordering of lattice points.

	i	Natural order	Radical inverse	Gray code variant
L_0	0	0	0	0
ΔL_1	1	$1/2 = 0.5$	$(0.1)_2 = 0.5$	$(0.1)_2 = 0.5$
ΔL_2	2	$1/4 = 0.25$	$(0.01)_2 = 0.25$	$(0.11)_2 = 0.75$
	3	$3/4 = 0.75$	$(0.11)_2 = 0.75$	$(0.01)_2 = 0.25$
ΔL_3	4	$1/8 = 0.125$	$(0.001)_2 = 0.125$	$(0.011)_2 = 0.375$
	5	$3/8 = 0.375$	$(0.101)_2 = 0.625$	$(0.111)_2 = 0.875$
	6	$5/8 = 0.625$	$(0.011)_2 = 0.375$	$(0.101)_2 = 0.625$
	7	$7/8 = 0.875$	$(0.111)_2 = 0.875$	$(0.001)_2 = 0.125$
ΔL_4	8	$1/16 = 0.0625$	$(0.0001)_2 = 0.0625$	$(0.0011)_2 = 0.1875$
	9	$3/16 = 0.1875$	$(0.1001)_2 = 0.5625$	$(0.1011)_2 = 0.6875$
	10	$5/16 = 0.3125$	$(0.0101)_2 = 0.3125$	$(0.1111)_2 = 0.9375$
	11	$7/16 = 0.4375$	$(0.1101)_2 = 0.8125$	$(0.0111)_2 = 0.4375$
	12	$9/16 = 0.5625$	$(0.0011)_2 = 0.1875$	$(0.0101)_2 = 0.3125$
	13	$11/16 = 0.6875$	$(0.1011)_2 = 0.6875$	$(0.1101)_2 = 0.8125$
	14	$13/16 = 0.8125$	$(0.0111)_2 = 0.4375$	$(0.1001)_2 = 0.5625$
	15	$15/16 = 0.9375$	$(0.1111)_2 = 0.9375$	$(0.0001)_2 = 0.0625$

which is used as the reference error to provide a good normalization. The components of $\boldsymbol{z}$ can take values up to $b^{m_2} - 1$. Due to many structures in the embedding, the computational cost for the modified CBC algorithm is only a factor of $O(m_2)$ more than the cost for constructing a lattice rule with a fixed number of points. Figure 5.2 illustrates the embedding structure.

Numerically, it is found in Cools *et al.* (2006) for $2^{10} \le n \le 2^{20}$, s up to 360, and a number of different choices of product weights and order-dependent weights, that the quantity $X_{m_1,m_2,s}(\boldsymbol{z})$ resulting from the modified CBC construction is at most 1.6, indicating that the embedded lattice rules achieve the near-optimal rate of convergence of lattice rules with a fixed number of points, but with an implied constant up to 1.6 times larger.

Dick, Pillichshammer and Waterhouse (2008) introduced a *sieve algorithm* to obtain extensible lattice sequences. The general principle is to begin with a number of generating vectors that are *good* for some value of n, drop those vectors that are not good for a larger value of n, and then repeat this process for larger and larger values of n. Some parameters need to be specified beforehand to ensure that the sieve process leaves a non-empty set of generating vectors. Note that this sieve algorithm yields lattice sequences

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

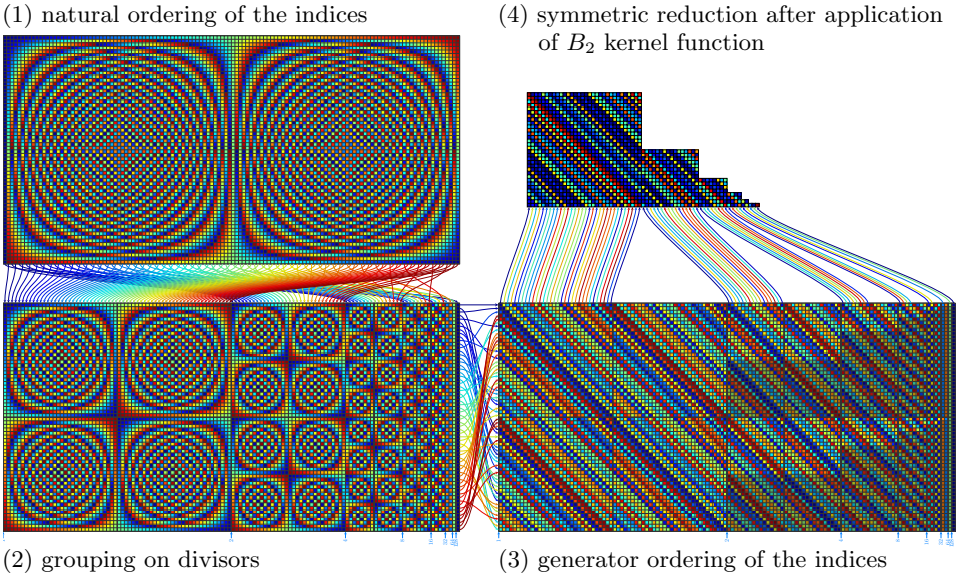


Figure 5.2. (Image by Dirk Nuyens.) The structure of the matrix Ω_{128} for the fast CBC construction of lattice rules, illustrating the reordering of indices and the reduction due to symmetry. The matrices Ω_n for $n = 2^m$ with $m < 7$ are embedded in Ω_{128}.

that are extensible in n but the algorithm is very expensive to implement. Dick *et al.* (2008) also introduced a cheaper version called the *CBC sieve algorithm*, which yields embedded lattice rules that work for a finite set of n and are extensible in s. This theory can also be modified to provide the theoretical justification for the algorithm from Cools *et al.* (2006).

5.8. Lattice rules in weighted Korobov spaces

In the previous subsections we have presented the lattice rule results for weighted anchored or unanchored variants of Sobolev spaces containing functions with square-integrable mixed first derivatives. Analogous results were originally obtained in a different function space setting, the *weighted Korobov spaces* of periodic functions.

We now briefly review the essential ingredients and results for weighted Korobov spaces, with the motive of understanding how one might achieve better than order-one convergence rate for non-periodic integrands. In the Korobov space setting, there is a *smoothness parameter* $\alpha > 1/2$. (The parameter α as used here differs by a factor of two from the usage in Sloan and Woźniakowski 2001.) The Korobov space can be viewed as an L_2 version of the Korobov class $E_\alpha(c)$ mentioned in Section 5.1. It again contains one-periodic functions with absolutely convergent Fourier series (5.1) but with

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

the inner product now given by

$$\langle f, g\rangle_{s,\alpha,\gamma} = \sum_{\boldsymbol{h}\in\mathbb{Z}^s} \frac{\prod_{j\in\mathfrak{u}(\boldsymbol{h})} |h_j|^{2\alpha}}{\gamma_{\mathfrak{u}(\boldsymbol{h})}} \widehat{f}(\boldsymbol{h})\,\overline{\widehat{g}(\boldsymbol{h})},$$

with $\mathfrak{u}(\boldsymbol{h}) := \{j \in \{1:s\} : h_j \neq 0\}$. The reproducing kernel is

$$\begin{aligned} K_{s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{x},\boldsymbol{y}) &= \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \left(\sum_{h_j\in\mathbb{Z}\setminus\{0\}} \frac{\mathrm{e}^{2\pi\mathrm{i}h_j(x_j-y_j)}}{|h_j|^{2\alpha}} \right) \\ &= \sum_{\boldsymbol{h}\in\mathbb{Z}^s} \gamma_{\mathfrak{u}(\boldsymbol{h})} \prod_{j\in\mathfrak{u}(\boldsymbol{h})} \frac{\mathrm{e}^{2\pi\mathrm{i}h_j(x_j-y_j)}}{|h_j|^{2\alpha}} \\ &= \sum_{\boldsymbol{h}\in\mathbb{Z}^s} \frac{\gamma_{\mathfrak{u}(\boldsymbol{h})}}{\prod_{j\in\mathfrak{u}(\boldsymbol{h})} |h_j|^{2\alpha}}\, \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{h}\cdot(\boldsymbol{x}-\boldsymbol{y})}. \end{aligned}$$

From this last expression it is easy to verify the reproducing property.

The parameter α moderates the rate of decay of the Fourier coefficients, and is related to the smoothness of the functions. To illustrate this point, we note that when α is an integer, the norm in one dimension can be written as

$$\|f\|_{1,\alpha,\gamma}^2 = \left(\int_0^1 f(t)\,\mathrm{d}t\right)^2 + \frac{1}{(2\pi)^{2\alpha}\gamma} \int_0^1 \left(f^{(\alpha)}(t)\right)^2 \mathrm{d}t.$$

Thus α is precisely the number of available square-integrable derivatives.

When α is an integer, we can use the property (5.7) to rewrite the kernel as

$$K_{s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{x},\boldsymbol{y}) = \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \left(\frac{(2\pi)^{2\alpha}}{(-1)^{\alpha+1}(2\alpha)!} B_{2\alpha}\big(|x_j - y_j|\big) \right), \tag{5.33}$$

where $B_{2\alpha}(\cdot)$ is the Bernoulli polynomial of degree 2α. Our earlier results in the Sobolev space setting were originally obtained by observing that the associated shift-invariant kernel (see (5.11)) is in fact the kernel for a Korobov space with $\alpha = 1$ but with some redefined weights.

It is straightforward to deduce, using Theorem 3.5, that the worst-case error for a rank-one lattice rule in a weighted Korobov space satisfies

$$e_{n,s}^2(\boldsymbol{z}) = \frac{1}{n} \sum_{k=0}^{n-1} \sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \sum_{h\in\mathbb{Z}\setminus\{0\}} \frac{\mathrm{e}^{2\pi\mathrm{i}hkz_j/n}}{|h|^{2\alpha}}.$$

For product weights, the expression simplifies to

$$e_{n,s}^2(\boldsymbol{z}) = -1 + \frac{1}{n} \sum_{k=0}^{n-1} \prod_{j=1}^{s} \left(1 + \gamma_j \sum_{h\in\mathbb{Z}\setminus\{0\}} \frac{\mathrm{e}^{2\pi\mathrm{i}hkz_j/n}}{|h|^{2\alpha}} \right),$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

which is precisely the formula for P_α when all $\gamma_j = 1$ (see (5.4)), but with α replaced by 2α.

The following theorem summarizes the error bound for lattice rules constructed by the CBC algorithm in weighted Korobov space. We remark that the result holds for unshifted lattice rules, but since the reproducing kernel is shift-invariant, the result applies also to shifted lattice rules. The best convergence rate is obtained by taking $\lambda \to 1/(2\alpha)$, which yields close to $O(n^{-\alpha})$ convergence.

Theorem 5.12 (optimal CBC error bound in Korobov spaces). The generating vector $\boldsymbol{z}$ constructed by the CBC algorithm in Korobov spaces, minimizing $e^2_{n,s}(\boldsymbol{z})$ in each step, satisfies

$$e^2_{n,s}(\boldsymbol{z}) \le \left(\frac{1}{\varphi(n)} \sum_{\emptyset \ne \mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}}^{\lambda} (2\zeta(2\alpha\lambda))^{|\mathfrak{u}|} \right)^{1/\lambda}$$

for all $\lambda \in (1/(2\alpha), 1]$, where $\zeta(\cdot)$ is the Riemann zeta function (5.5) and $\varphi(n)$ is the Euler totient function (5.14).

Proof. Using the character property (5.2), we can write

$$e^2_{n,s}(\boldsymbol{z}) = \sum_{\emptyset \ne \mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}} \left(\sum_{\substack{\boldsymbol{h}_{\mathfrak{u}} \in (\mathbb{Z} \setminus \{0\})^{|\mathfrak{u}|} \\ \boldsymbol{h}_{\mathfrak{u}} \cdot \boldsymbol{z}_{\mathfrak{u}} \equiv 0 \,(\mathrm{mod}\, n)}} \frac{1}{\prod_{j \in \mathfrak{u}} |h_j|^{2\alpha}} \right).$$

The rest of the proof follows the proof of Theorem 5.8, which is essentially the case $\alpha = 1$, but with weights scaled by $(2\pi)^{|\mathfrak{u}|}$. □

The results for Korobov spaces are not often usable in practice because integrands are typically not fully periodic. However, the analysis in Korobov spaces provides the fundamental framework for the lattice rule analysis in Sobolev spaces. There we achieved order-one convergence via *shifting*: the shift-averaged worst-case error in Sobolev spaces of smoothness one is related to the Korobov space with $\alpha = 1$. Our aim now is to achieve higher convergence rates for non-periodic functions by linking with the results in Korobov space with $\alpha > 1$.

5.9. The baker's transformation

Extending Example 3.8 to s dimensions and introducing weights as in Section 4, we obtain the reproducing kernel for the *weighted and unanchored Sobolev space of smoothness* α:

$$K_{s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}} \prod_{j \in \mathfrak{u}} \left(\frac{B_{2\alpha}(|x_j - y_j|)}{(-1)^{\alpha+1}\,(2\alpha)!} + \sum_{r=1}^{\alpha} \frac{B_r(x_j)\, B_r(y_j)}{(r!)^2} \right). \quad (5.34)$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Comparing with (5.33), we see that this kernel has additional terms involving the lower-degree Bernoulli polynomials $B_1, \dots, B_\alpha$. We already know that in the case of smoothness one ($\alpha = 1$) the associated shift-invariant kernel does not involve the lower-degree polynomial B_1; in short, *shifting removes* B_1. In this subsection we explain how *shifting followed by folding removes* B_2.

The *baker's transformation* makes use of the function

$$\phi(t) := 1 - |2t - 1| = \begin{cases} 2t & \text{if } t \le \frac{1}{2}, \\ 2 - 2t & \text{if } t > \frac{1}{2}. \end{cases}$$

It is called the baker's transformation since it emulates how a baker stretches and folds bread dough. In one dimension, if we apply the baker's transformation to the n points of a left-rectangle rule $0, 1/n, 2/n, \dots, (n-1)/n$, where n is even, then we obtain the points

$$0, \frac{2}{n}, \frac{4}{n}, \dots, 1, \frac{n-2}{n}, \frac{n-4}{n}, \dots, \frac{2}{n},$$

which are the points for a trapezoidal rule. A rectangle rule has quadrature error $O(n^{-1})$, but a trapezoidal rule has quadrature error $O(n^{-2})$ if the integrand is sufficiently smooth. This is the underlying motivation for considering the baker's transformation. Its application in the context of QMC is due to Hickernell (2002). In the following, we refer to the process of applying the baker's transformation as *folding* and the resulting point set as the *folded* point set.

A shifted and then folded QMC rule takes the form

$$Q^{\phi}_{n,s}(\boldsymbol{\Delta}; f) = \frac{1}{n} \sum_{i=0}^{n-1} f\big(\phi\big(\{\boldsymbol{t}_i + \boldsymbol{\Delta}\}\big)\big).$$

Note that this can also be viewed as applying a shifted lattice rule to a transformed integrand $g = f \circ \phi$. However, this interpretation is not helpful because the transformed integrand g does not have sufficient smoothness to get higher-order convergence.

As in Section 5.2, for a given point set P we let $\phi(P + \boldsymbol{\Delta})$ denote the shifted and then folded QMC point set. Then, for a given shift $\boldsymbol{\Delta}$ we have

$$|I_s(f) - Q^{\phi}_{n,s}(\boldsymbol{\Delta}; f)| \le e_{n,s}(\phi(P + \boldsymbol{\Delta}); H)\, \|f\|_H,$$

and for a randomly shifted and folded rule we have

$$\sqrt{\mathbb{E}|I_s(f) - Q^{\phi}_{n,s}(\boldsymbol{\Delta}; f)|^2} \le e^{\mathrm{sh},\phi}_{n,s}(P; H)\, \|f\|_H,$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

where

$$e_{n,s}^{\mathrm{sh},\phi}(P;H) := \sqrt{\int_{[0,1]^s} e_{n,s}^2(\phi(P+\boldsymbol{\Delta});H)\,\mathrm{d}\boldsymbol{\Delta}}. \tag{5.35}$$

Theorem 5.13 (formula for the shift-averaged and folded worst-case error). The *shift-averaged and folded* worst-case error (5.35) for a QMC point set P in an RKHS $H_s(K)$ satisfies

$$[e_{n,s}^{\mathrm{sh},\phi}(P;H_s(K))]^2 = -\int_{[0,1]^s}\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y} + \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{k=0}^{n-1} K^{\mathrm{sh},\phi}(\boldsymbol{t}_i,\boldsymbol{t}_k),$$

where

$$K^{\mathrm{sh},\phi}(\boldsymbol{x},\boldsymbol{y}) := \int_{[0,1]^s} K(\phi\{\boldsymbol{x}+\boldsymbol{\Delta}\},\phi\{\boldsymbol{y}+\boldsymbol{\Delta}\})\,\mathrm{d}\boldsymbol{\Delta} \quad \text{for all } \boldsymbol{x},\boldsymbol{y}\in[0,1]^s. \tag{5.36}$$

Proof. Using the definition (5.35) and applying the formula (3.9) for the worst-case error $e_{n,s}(\phi(P+\boldsymbol{\Delta});H_s(K))$, we obtain

$$\begin{aligned}
&[e_{n,s}^{\mathrm{sh},\phi}(P;H_s(K))]^2 \\
&= \int_{[0,1]^s}\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y} - \frac{2}{n}\sum_{i=0}^{n-1}\int_{[0,1]^s}\int_{[0,1]^s} K(\phi(\{\boldsymbol{t}_i+\boldsymbol{\Delta}\}),\boldsymbol{y})\,\mathrm{d}\boldsymbol{\Delta}\,\mathrm{d}\boldsymbol{y} \\
&\quad + \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{k=0}^{n-1}\int_{[0,1]^s} K(\phi(\{\boldsymbol{t}_i+\boldsymbol{\Delta}\}),\phi(\{\boldsymbol{t}_k+\boldsymbol{\Delta}\}))\,\mathrm{d}\boldsymbol{\Delta}.
\end{aligned}$$

With a change of variables $\boldsymbol{w} = \{\boldsymbol{t}_i+\boldsymbol{\Delta}\}$, the double integral in the second term becomes $\int_{[0,1]^s}\int_{[0,1]^s} K(\phi(\boldsymbol{w}),\boldsymbol{y})\,\mathrm{d}\boldsymbol{w}\,\mathrm{d}\boldsymbol{y}$, which is easily seen to equal $\int_{[0,1]^s}\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y}$. The result then follows from the definition (5.36). □

We now demonstrate the effect of shifting followed by folding for the case $\alpha = 2$. To obtain the shifted and folded kernel $K^{\mathrm{sh},\phi}$ associated with (5.34), we need to evaluate the following integrals:

$$\begin{aligned}
\mathcal{I}_1(x,y) &:= \int_0^1 B_1(\phi(\{x+\Delta\}))\,B_1(\phi(\{y+\Delta\}))\,\mathrm{d}\Delta, \\
\mathcal{I}_2(x,y) &:= \int_0^1 B_2(\phi(\{x+\Delta\}))\,B_2(\phi(\{y+\Delta\}))\,\mathrm{d}\Delta, \\
\mathcal{J}_4(x,y) &:= \int_0^1 B_4(|\phi(\{x+\Delta\}) - \phi(\{y+\Delta\}))\,\mathrm{d}\Delta.
\end{aligned}$$

By using the Fourier series expansion (5.6) of the Bernoulli polynomials,

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

it can be shown (Hickernell 2002) that

$$\begin{aligned}
\mathcal{I}_1(x,y) &= -\frac{4}{3}B_4(\{x-y\}) + \frac{4}{3}B_4\big(\{x-y-\tfrac{1}{2}\}\big),\\
\mathcal{I}_2(x,y) &= -\frac{4}{3}B_4(\{x-y\}) - \frac{4}{3}B_4\big(\{x-y-\tfrac{1}{2}\}\big),\\
\mathcal{J}_4(x,y) &= \frac{32}{3}B_4(\{x-y\}) - \frac{8}{3}B_4\big(\{x-y-\tfrac{1}{2}\}\big)\\
&\qquad + \frac{128}{15}B_6(\{x-y\}) - \frac{128}{15}B_6\big(\{x-y-\tfrac{1}{2}\}\big).
\end{aligned}$$

Since the lowest-degree Bernoulli polynomial in the above expressions is B_4, we conclude that the shifted and folded kernel is related to the kernel of the Korobov space with smoothness parameter $\alpha = 2$. This indicates that we get a convergence rate close to $O(n^{-2})$.

In addition, it can be shown that the CBC construction yields lattice rules achieving a convergence rate close to $O(n^{-2})$, and that the implementation can be done in a fast way for product or POD weights. We omit the details.

It is possible to generalize the baker's transformation to obtain even higher rates of convergence, but there is a seemingly inevitable dependence on s, prohibiting the strategy for practical use in high dimensions.

5.10. Periodization

From the results for the Korobov space, we know that lattice rules can yield higher rates of convergence for periodic integrands. This brings us to consider strategies for transforming a non-periodic integrand to a periodic integrand.

We now briefly explain one periodization strategy. Let $\omega : [0,1] \to \mathbb{R}$ be a smooth function, with

$$\omega(0) = \omega(1) = 0, \quad \omega(y) > 0 \quad \text{for all } y \in (0,1), \quad \text{and} \quad \int_0^1 \omega(y)\,\mathrm{d}y = 1.$$

Define

$$\psi(y) := \int_0^y \omega(u)\,\mathrm{d}u.$$

Then ψ is an increasing function satisfying

$$\psi'(y) = \omega(y), \quad \psi(0) = 0 \quad \text{and} \quad \psi(1) = 1.$$

The periodization transformation is essentially a change of variables $\boldsymbol{x} = \psi(\boldsymbol{y}) = (\psi(y_1), \ldots, \psi(y_s))$ so that

$$I_s(f) = \int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_{[0,1]^s} F(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} = I_s(F),$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

with

$$F(\boldsymbol{y}) := f(\psi(\boldsymbol{y})) \prod_{j=1}^{s} \omega(y_j).$$

The function F is periodic since ω vanishes at the end points.

Popular choices of ω include the family of *polynomial transformations* of Korobov (1963) and the family of *trigonometric transformations* of Sidi (1993); see also Laurie (1996). For example, the former family is given by

$$\omega(y) = (2\alpha + 1)\binom{2\alpha}{\alpha} y^\alpha (1-y)^\alpha$$

for a positive integer α. In particular, we have $\omega^{(r)}(0) = \omega^{(r)}(1)$ for all $r \leq \alpha - 1$. With this choice of ω, and if f is sufficiently smooth, the transformed integrand F belongs to the Korobov space with smoothness parameter α, and the lattice rule convergence rate is therefore close to $O(n^{-\alpha})$.

The problem with the periodization strategy is that the norm of the transformed integrand F can be exponentially large in s, thus is generally only feasible for small to moderate s.

As we explained in the previous subsection, the baker's transformation can also be viewed as a transformation of the integrand. The baker's transformation makes the end points of the function take the same value by reflection. The periodization strategy, on the other hand, makes the function value and perhaps also some derivatives vanish at the end points.

5.11. Notes

Lattice rules date back to the works of number theorists Korobov (1959) and Hlawka (1962). There is an extensive literature up to the late 1990s, which we refer to as the classical results; these are reviewed in detail in the books by Niederreiter (1992*a*) and Sloan and Joe (1994); see also Hickernell (1998*b*). The modern consideration of lattice rules can be said to have begun with Sloan and Woźniakowski (2001), who proved the existence of good lattice rules for periodic integrands in weighted Korobov spaces, and via the shift-invariant kernel, they showed that there exist good shifted lattice rules for non-periodic integrands in weighted Sobolev space of smoothness one. Although the results were non-constructive, they demonstrated that lattice rules have a role to play for truly high-dimensional integration as well as for non-periodic integrands.

The CBC construction was first considered by Korobov (1959) (see also Korobov 1963), who proved error bounds for P_α of order $n^{-\alpha+\delta}$ for any $\delta > 0$ and for arbitrary n. The modern analysis of CBC construction for the lattice rule generating vector began with Sloan and Reztsov (2002) for the classical (unweighted) criterion P_α in the periodic setting. Sloan,

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Kuo and Joe (2002*a*) then introduced the CBC algorithm to construct a deterministic shift alongside the generating vector for integration in the non-periodic weighted Sobolev spaces. Subsequently, the CBC algorithm for constructing randomly shifted lattice rules in weighted Sobolev spaces (see Algorithm 5.6) was introduced in Sloan, Kuo and Joe (2002*b*). In these earlier works only $O(n^{-1/2})$ convergence rate was proved. The fact that the CBC algorithm achieves the (optimal) convergence rate close to $O(n^{-\alpha})$ in weighted Korobov and close to $O(n^{-1})$ in weighted Sobolev spaces with smoothness one was later proved by Kuo (2003). Extensions of these results from prime n to composite n were given in Kuo and Joe (2002) with convergence rate of $O(n^{-1/2})$, and Dick (2004) with convergence rate close to $O(n^{-1})$. Other considerations include the CBC construction of intermediate-rank lattice rules (Kuo and Joe 2003), and a modified CBC algorithm making use of the prime factorization of n for the purpose of speeding up the construction (Dick and Kuo 2004*a*, 2004*b*). All of the above results were originally proved in the setting of product weights. Results for general weights in Korobov spaces were obtained in Dick *et al.* (2006), and the corresponding results for weighted Sobolev spaces were obtained in Sloan *et al.* (2004), again via the shift-invariant kernel.

The fast CBC construction was first introduced for prime n in Nuyens and Cools (2006*a*) and later extended to composite n in Nuyens and Cools (2006*b*). The special case of n being a power of 2 is discussed in Cools *et al.* (2006). Fast CBC for POD weights was first considered in Kuo, Schwab and Sloan (2011). Some good generating vectors for lattice rules can be downloaded from http://www.maths.unsw.edu.au/∼fkuo/lattice/. Implementation of fast CBC constructions for various forms of weights can be obtained from http://people.cs.kuleuven.be/∼dirk.nuyens/fast-cbc/.

Extensible lattice sequences were considered in Maize (1980), Korobov (1982), and Hickernell *et al.* (2000), but the first theoretical result proving that good extensible lattice sequences exist was proved by Hickernell and Niederreiter (2003). Cools *et al.* (2006) introduced a fast CBC construction for embedded lattice rules, while Dick *et al.* (2008) analysed a sieve algorithm and also provided the theoretical justification for the algorithm in Cools *et al.* (2006). An algorithm was introduced in Niederreiter and Pillichshammer (2009) which constructs the lattice rule digit by digit. Hickernell, Kritzer, Kuo and Nuyens (2012) investigated strategies for applying a lattice sequence point by point.

Joe (2004) introduced a CBC construction of lattice rules based on an error criterion called R, which is part of an upper bound to the classical star discrepancy. This was later extended to the weighted setting for product weights in Joe (2006), and to composite n in Sinescu and Joe (2008). General weights were considered by Sinescu and Joe (2007) for prime n, and by Sinescu and L'Ecuyer (2011) for composite n. The analysis for general

weights presented difficulties. (Recall that the CBC construction for the anchored space with general non-product weights must work with an auxiliary quantity. There is a similar difficulty here.) Thus, instead of working with weighted-R, the last two papers considered a related quantity which we shall refer to as weighted-$\widetilde{R}$, and which forms part of a different upper bound to the weighted star discrepancy. This argument holds under a restrictive monotonicity assumption on the weights that

$$\gamma_{\mathfrak{v}} \geq \gamma_{\mathfrak{u}} \quad \text{whenever} \quad \mathfrak{v} \subseteq \mathfrak{u}. \tag{5.37}$$

Switching from weighted-R to weighted-$\widetilde{R}$ does not require a redefinition of weights. Thus any structure in the weights, such as the POD form, would be preserved. Note, however, that the given POD weights may not satisfy the monotonicity requirement (5.37) needed to allow for this switch from weighted-R to weighted-$\widetilde{R}$. (We remark that imposing condition (5.37) does not appear to eliminate the need to work with the auxiliary quantity in the anchored space with general non-product weights.)

The baker's transformation (which is called 'tent transformation' in ergodic theory) for shifted lattice rules was first introduced by Hickernell (2002) to gain an extra order of convergence for non-periodic integrands, that is, from nearly $O(n^{-1})$ to nearly $O(n^{-2})$. The recent work by Dick *et al.* (2013) generalizes this strategy to obtain even higher rates of convergence. There is, however, an inevitable dimension dependence. This is similar to the issue with periodization in high dimensions, which was discussed in Kuo, Sloan and Woźniakowski (2007). An interesting discovery by Dick *et al.* (2013) is that the baker's transformation can be applied to an unshifted lattice rule, to achieve close to $O(n^{-1})$ convergence for non-periodic integrands, *without the need to use random shifts*. This is different from most of the results discussed in this section.

Randomly shifted lattice rules for integrands over unbounded domains (including $\mathbb{R}^s$) were considered in Kuo, Wasilkowski and Waterhouse (2006*b*) and Kuo *et al.* (2010*a*). The strategy is to transform the integrals to the unit cube by a change of variables, using the inverse of the cumulative distribution function of some univariate probability density function. The transformed integrands over the unit cube typically do not belong to the standard Sobolev space setting, and this is why new theory is needed. It is proved that good lattice rules can be constructed by a CBC algorithm in a similar way to the standard theory in Sobolev spaces.

Another branch of lattice rule analysis focuses on the trigonometric degree and similar quantities; see for example Cools and Lyness (2001), Lyness and Sørevik (2006), Cools and Nuyens (2008) and the references therein. Recent constructions in terms of trigonometric degrees include the works by Cools, Kuo and Nuyens (2010), Achtsis and Nuyens (2012) and Kämmerer, Kunis and Potts (2012).

Other search criteria for lattice rules were reviewed in L'Ecuyer and Lemieux (2000), Lemieux and L'Ecuyer (2001) and L'Ecuyer and Munger (2012), while L'Ecuyer, Munger and Tuffin (2010) considered the distribution of the integration error with respect to the random shift.

Lattice rules with exponential convergence rates in (modified) weighted Korobov spaces were considered in Dick, Larcher, Pillichshammer and Woźniakowski (2011).

Approximation of functions by lattice rules was considered in Li and Hickernell (2003) and Zeng, Leung and Hickernell (2006). CBC constructions of lattice rules for the approximation of functions were analysed in Kuo, Sloan and Woźniakowski (2006*a*, 2008*b*) and Kuo, Wasilkowski and Woźniakowski (2009).

The choice of weights in relation to the application of lattice rules were considered in Wang and Sloan (2006), Dick (2012) and Kuo *et al.* (2012).

The character property (5.2) is based on the following general algebraic structure: The set $([0,1)^s, \oplus)$ with addition modulo 1 given by $\boldsymbol{x} \oplus \boldsymbol{y} := \{\boldsymbol{x}+\boldsymbol{y}\}$ forms an abelian group, and the set of lattice points $P_{\rm lat} = \{\boldsymbol{t}_k = \{k\boldsymbol{z}/n\} : 0 \le k < n\}$ is a finite subgroup. The set $\mathcal{K}$ of functions $\chi_{\boldsymbol{h}} : [0,1)^s \to \{z \in \mathbb{C} : |z| = 1\}$, $\boldsymbol{h} \in \mathbb{Z}$, given by

$$\chi_{\boldsymbol{h}}(\boldsymbol{x}) = \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{h}\cdot\boldsymbol{x}},$$

constitute a group homomorphism of the group $([0,1)^s, \oplus)$ called characters. They themselves form a group via the multiplication $\chi_{\boldsymbol{h}}\chi_{\boldsymbol{k}} = \chi_{\boldsymbol{h}+\boldsymbol{k}}$ which is isomorphic to the group $(\mathbb{Z}^s, +)$. The character-theoretic dual lattice of the lattice point set $P_{\rm lat}$ is given by the set of characters

$$\mathcal{L}^\perp = \{\chi_{\boldsymbol{h}} : \boldsymbol{h} \in \mathbb{Z}^s, \chi_{\boldsymbol{h}}(\{k\boldsymbol{z}/n\}) = 1 \text{ for all } 0 \le k < n\},$$

which is a subgroup of the set of characters. This can be used to define the quotient group $P_{\rm lat}^\perp = \mathcal{K}/\mathcal{L}^\perp$, which can be viewed as the dual group of characters of a lattice point set $P_{\rm lat}$. Since the characters of $([0,1)^s, \oplus)$ are the basis functions of Fourier series, it is natural to study lattice rules for numerical integration of Fourier series. We will use an analogous algebraic structure in the next section.

6. Digital nets and sequences

In the previous section we discussed the modern theory of lattice rules. Many of the ideas there apply also to polynomial lattice rules, or more generally, to nets and sequences. For lattice rules we have seen that Fourier series play an important role. For polynomial lattice rules and nets this role is played by *Walsh series*, which we introduce in the following subsection.

Throughout this section we denote by $\mathbb{N}_0$ the set of non-negative integers and $\mathbb{N}$ the set of positive integers.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

6.1. A brief discussion of numerical integration of Walsh series

Walsh (1923) introduced a system of functions denoted by $\{\mathrm{wal}_k : k \in \mathbb{N}_0\}$, which is in some way similar to the trigonometric function system $\{\mathrm{e}^{2\pi \mathrm{i} k x} : k \in \mathbb{Z}\}$ that is connected to the well-known Fourier theory.

Definition 6.1 (Walsh functions in one dimension). For $b \geq 2$ we denote by $\omega_b := \mathrm{e}^{2\pi \mathrm{i}/b}$ the primitive bth root of unity. Let $k \in \mathbb{N}_0$ with b-adic expansion $k = \kappa_0 + \kappa_1 b + \cdots + \kappa_{r-1} b^{r-1}$. The *$k$th b-adic Walsh function* $\mathrm{wal}_k : \mathbb{R} \to \mathbb{C}$, periodic with period one, is defined by

$$\mathrm{wal}_k(x) := \omega_b^{\kappa_0 \xi_1 + \kappa_1 \xi_2 + \cdots + \kappa_{r-1} \xi_r},$$

for $x \in [0, 1)$ with b-adic expansion $x = \xi_1 b^{-1} + \xi_2 b^{-2} + \cdots$ (unique in the sense that infinitely many of the digits ξ_i must be different from $b - 1$).

In the literature the function system defined above is often called the *generalized Walsh function system* and only in the case $b = 2$ one speaks of Walsh functions.

One of the main differences between Walsh functions and the trigonometric functions is that Walsh functions are only piecewise continuous. This is clear, since Walsh functions are step functions as we now show.

Let $k \in \mathbb{N}$ with b-adic expansion $k = \kappa_0 + \kappa_1 b + \cdots + \kappa_{r-1} b^{r-1}$. Let $J = [a/b^r, (a+1)/b^r)$, with integers $r \geq 1$ and $0 \leq a < b^r$, be a so-called *elementary b-adic interval of order r*. Let a have b-adic expansion of the form $a = \alpha_0 + \alpha_1 b + \cdots + \alpha_{r-1} b^{r-1}$. Then any $x \in J$ has b-adic expansion

$$x = \alpha_{r-1} b^{-1} + \alpha_{r-2} b^{-2} + \cdots + \alpha_0 b^{-r} + \xi_{r+1} b^{-(r+1)} + \cdots$$

for some digits $0 \leq \xi_i \leq b - 1$ for $i \geq r + 1$, and hence

$$\mathrm{wal}_k(x) = \omega_b^{\kappa_0 \alpha_{r-1} + \kappa_1 \alpha_{r-2} + \cdots + \kappa_{r-1} \alpha_0} = \mathrm{wal}_k(a/b^r).$$

We summarize this result in the following proposition.

Proposition 6.2 (Walsh functions are step functions). Let $k \in \mathbb{N}$ satisfy $b^{r-1} \leq k < b^r$ for some $r \in \mathbb{N}$. Then the kth Walsh function wal_k is constant on elementary b-adic intervals of order r of the form $[a/b^r, (a+1)/b^r)$ with value $\mathrm{wal}_k(a/b^r)$. Further, $\mathrm{wal}_0 \equiv 1$.

Now we generalize the definition of Walsh functions to higher dimensions.

Definition 6.3 (Walsh functions in s dimensions). For dimension $s \geq 2$, and $k_1, \ldots, k_s \in \mathbb{N}_0$ we define the *s-dimensional b-adic Walsh function* $\mathrm{wal}_{k_1,\ldots,k_s} : \mathbb{R}^s \to \mathbb{C}$ by

$$\mathrm{wal}_{k_1,\ldots,k_s}(x_1, \ldots, x_s) := \prod_{j=1}^{s} \mathrm{wal}_{k_j}(x_j).$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

For vectors $\boldsymbol{k} = (k_1, \ldots, k_s) \in \mathbb{N}_0^s$ and $\boldsymbol{x} = (x_1, \ldots, x_s) \in [0,1)^s$ we write, with some abuse of notation,

$$\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}) := \mathrm{wal}_{k_1,\ldots,k_s}(x_1, \ldots, x_s).$$

The system $\{\mathrm{wal}_{\boldsymbol{k}} : \boldsymbol{k} \in \mathbb{N}_0^s\}$ is called the *s-dimensional b-adic Walsh function system*.

Since any s-dimensional Walsh function is a product of one-dimensional Walsh functions, it is clear that s-dimensional Walsh functions are step functions too.

Let $P = \{\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{b^m-1}\}$ be a digital (t, m, s)-net constructed over $\mathbb{Z}_b$ as introduced in Section 2.6. The set P is an additive group whose dual group of characters is given by the Walsh functions in the same base b, that is, we have the following character property:

$$\frac{1}{b^m} \sum_{i=0}^{b^m-1} \mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{t}_i) = \begin{cases} 1 & \text{if } C_1^\top \vec{k}_1 + \cdots + C_s^\top \vec{k}_s = \vec{0}, \\ 0 & \text{otherwise,} \end{cases} \tag{6.1}$$

where for a vector $k \in \mathbb{N}_0$ with b-adic expansion $k = \kappa_0 + \kappa_1 b + \cdots + \kappa_{m-1} b^{m-1}$ we write $\vec{k} = (\kappa_0, \kappa_1, \ldots, \kappa_{m-1})^\top$. This leads to a similar result to Theorems 5.1 and 5.2 for lattice rules.

Theorem 6.4 (integration error for digital net). Assume we are given a Walsh series

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in \mathbb{N}_0^s} \widehat{f}(\boldsymbol{k}) \mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}),$$

where the Walsh coefficients $\widehat{f}(\boldsymbol{k})$ are given by

$$\widehat{f}(\boldsymbol{k}) = \int_{[0,1]^s} f(\boldsymbol{x}) \overline{\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x})} \, \mathrm{d}\boldsymbol{x}.$$

The integration error of approximating the integral of f using a QMC rule $Q_{b^m,s}$ based on a digital net with b^m points is given by

$$Q_{b^m,s}(f) - I_s(f) = \sum_{\boldsymbol{k} \in \mathcal{D} \setminus \{\boldsymbol{0}\}} \widehat{f}(\boldsymbol{k}),$$

where

$$\mathcal{D} := \{\boldsymbol{k} \in \mathbb{N}_0^s : C_1^\top \vec{k}_1 + \cdots + C_s^\top \vec{k}_s = \vec{0}\}. \tag{6.2}$$

is the *dual net* associated with the digital net.

Proof. We have

$$\frac{1}{b^m} \sum_{i=0}^{b^m-1} f(\boldsymbol{t}_i) - \int_{[0,1]^s} f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \sum_{\boldsymbol{k} \in \mathbb{N}_0^s} \widehat{f}(\boldsymbol{k}) \frac{1}{b^m} \sum_{i=0}^{b^m-1} \mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{t}_i) - \widehat{f}(\boldsymbol{0}),$$

The result then follows immediately from the character property (6.1). □

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

We consider now the case where the Walsh coefficients $\widehat{f}(\boldsymbol{k})$ decay with a certain rate. For $k \in \mathbb{N}_0$ let

$$\mu_1(k) := \begin{cases} 0 & \text{if } k = 0, \\ a & \text{if } k = \kappa_0 + \cdots + \kappa_{a-1} b^{a-1}, \kappa_{a-1} \neq 0. \end{cases} \tag{6.3}$$

(Later in Section 6.6 we generalize the function μ_1 to μ_α.) For vectors $\boldsymbol{k} = (k_1, \ldots, k_s) \in \mathbb{N}_0^s$ let

$$\mu_1(\boldsymbol{k}) := \mu_1(k_1) + \cdots + \mu_1(k_s).$$

Let

$$E_{\vartheta,s}(c) := \bigl\{ f : [0,1]^s \to \mathbb{R} : |\widehat{f}(\boldsymbol{k})| \leq c\, b^{-\vartheta \mu_1(\boldsymbol{k})} \ \text{ for all } \boldsymbol{k} \in \mathbb{N}_0^s \bigr\},$$

for some $\vartheta > 1$ and $c > 0$. Then we obtain

$$|Q_{b^m,s}(f) - I_s(f)| \leq c \sum_{\boldsymbol{k} \in \mathcal{D} \setminus \{\boldsymbol{0}\}} b^{-\vartheta \mu_1(\boldsymbol{k})} \quad \text{for all } f \in E_{\vartheta,s}(c).$$

The quantity

$$T_\vartheta := \sum_{\boldsymbol{k} \in \mathcal{D} \setminus \{\boldsymbol{0}\}} b^{-\vartheta \mu_1(\boldsymbol{k})}, \quad \vartheta > 1, \tag{6.4}$$

depends only on the function class $E_{\vartheta,s}(1)$ and the digital net and can be understood as the 'worst-case error' for this function class. Thus, for the function class $E_{\vartheta,s}(1)$ one aims at finding digital nets for which T_ϑ is small. Thus the quantity T_ϑ plays a similar role to the quantity P_α for lattice rules (see (5.3)).

We briefly describe the connection between the quantity T_ϑ and the quality parameter t of the digital net. Lemma 2.14 implies that if $\boldsymbol{k} \in \mathcal{D} \setminus \{\boldsymbol{0}\}$, then $\mu_1(\boldsymbol{k}) > m - t$. If the digital net is a strict (t, m, s)-net, then there exists a $\boldsymbol{k} \in \mathcal{D} \setminus \{\boldsymbol{0}\}$ such that $\mu_1(\boldsymbol{k}) = m - t + 1$. Thus, the largest summand in (6.4) is of the form $b^{-\vartheta(m-t+1)}$. On the other hand it can also be shown that the quantity T_ϑ is dominated by its largest term, that is, there is a constant $\tilde{c} > 0$ such that

$$T_\vartheta \leq \tilde{c} \frac{(m-t)^{s-1}}{b^{\vartheta(m-t+1)}}.$$

Thus digital nets for which $m - t$ is large yield a small value of T_ϑ and are therefore useful for numerical integration of Walsh series.

6.2. Digitally shift-averaged worst-case error

In this subsection we discuss the general strategy for the error analysis of randomly digitally shifted QMC rules not specific to digital nets (similar to Section 5.2), and then focus on digital nets in the next subsection.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

For any QMC point set $P = \{\boldsymbol{t}_0, \dots, \boldsymbol{t}_{n-1}\}$ and any digital shift $\boldsymbol{\Delta} \in [0,1]^s$, let

$$P \oplus \boldsymbol{\Delta} = \{\boldsymbol{t}_i \oplus \boldsymbol{\Delta} : i = 0, 1, \dots, n-1\}$$

denote the digitally shifted QMC point set, and let $Q_{n,s}(\oplus\boldsymbol{\Delta}; f)$ denote the corresponding digitally shifted QMC rule, where $\oplus$ denotes digit-wise addition modulo b already defined in Section 2.9. Then, for any integrand f belonging to some normed space H, it follows from the definition of the worst-case error that

$$|I_s(f) - Q_{n,s}(\oplus\boldsymbol{\Delta}, f)| \le e_{n,s}(P \oplus \boldsymbol{\Delta}; H)\, \|f\|_H.$$

We consider the *root-mean-square* error

$$\sqrt{\mathbb{E}\,|I_s(f) - Q_{n,s}(\oplus\boldsymbol{\Delta}, f)|^2} \le e^{\mathrm{dsh}}_{n,s}(P; H)\, \|f\|_H,$$

where the expectation $\mathbb{E}$ is taken over the random digital shift $\boldsymbol{\Delta}$ which is uniformly distributed over $[0,1]^s$, and where the quantity

$$e^{\mathrm{dsh}}_{n,s}(P; H) := \sqrt{\int_{[0,1]^s} e^2_{n,s}(P \oplus \boldsymbol{\Delta}; H)\, \mathrm{d}\boldsymbol{\Delta}} \tag{6.5}$$

is referred to as the *digitally shift-averaged worst-case error*.

The digitally shift-averaged worst-case error (5.8) will be used as our quality measure for randomly digitally shifted QMC rules. The averaging argument guarantees the existence of at least one digital shift $\boldsymbol{\Delta}$ for which $e_{n,s}(P \oplus \boldsymbol{\Delta}; H) \le e^{\mathrm{dsh}}_{n,s}(P; H)$.

Analogously to Theorem 5.3, the following theorem gives an explicit formula for the digitally shift-averaged worst-case error when the function space is an RKHS.

Theorem 6.5 (formula for the digitally shift-averaged worst-case error). The *digitally shift-averaged* worst-case error (6.5) for a QMC point set P in an RKHS $H_s(K)$ satisfies

$$[e^{\mathrm{dsh}}_{n,s}(P; H_s(K))]^2 = -\int_{[0,1]^s}\int_{[0,1]^s} K(\boldsymbol{x}, \boldsymbol{y})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y} + \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{k=0}^{n-1} K^{\mathrm{dsh}}(\boldsymbol{t}_i, \boldsymbol{t}_k),$$

where

$$K^{\mathrm{dsh}}(\boldsymbol{x}, \boldsymbol{y}) := \int_{[0,1]^s} K(\boldsymbol{x} \oplus \boldsymbol{\Delta}, \boldsymbol{y} \oplus \boldsymbol{\Delta})\, \mathrm{d}\boldsymbol{\Delta} \quad \text{for all } \boldsymbol{x}, \boldsymbol{y} \in [0,1]^s. \tag{6.6}$$

The proof follows along the same lines as the proof of Theorem 5.3.

The function K^{dsh} defined by (6.6) is actually a reproducing kernel with the *digitally shift-invariant property*

$$K^{\mathrm{dsh}}(\boldsymbol{x}, \boldsymbol{y}) = K^{\mathrm{dsh}}(\boldsymbol{x} \oplus \boldsymbol{\Delta}, \boldsymbol{y} \oplus \boldsymbol{\Delta}) \quad \text{for all } \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\Delta} \in [0,1]^s,$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

or equivalently,

$$K^{\mathrm{dsh}}(\boldsymbol{x},\boldsymbol{y}) = K^{\mathrm{dsh}}(\boldsymbol{x}\ominus\boldsymbol{y},\boldsymbol{0}) \quad \text{for all } \boldsymbol{x},\boldsymbol{y}\in[0,1]^s,$$

where $\ominus$ denotes digit-wise subtraction modulo b already defined in Section 2.9. As in Section 5.2, it can be verified that the digitally shift-averaged worst-case error $e^{\mathrm{dsh}}_{n,s}(P;H_s(K))$ is precisely the worst-case error of the QMC point set P in the RKHS with K^{dsh} as the reproducing kernel, that is,

$$\int_{[0,1]^s} e^2_{n,s}(P\oplus\boldsymbol{\Delta};H_s(K))\,\mathrm{d}\boldsymbol{\Delta} = e^2_{n,s}(P;H_s(K^{\mathrm{dsh}})).$$

This important connection provides a powerful tool for analysing randomly digitally shifted QMC rules. We refer to the kernel K^{dsh} as the *digital shift-invariant kernel associated with* K.

6.3. Randomly digitally shifted digital nets in weighted Sobolev spaces

The digital shift-invariant kernel K^{dsh} in the previous subsection has Walsh series expansion

$$K^{\mathrm{dsh}}(\boldsymbol{x},\boldsymbol{y}) = \sum_{\boldsymbol{k}\in\mathbb{N}_0^s} \widehat{K^{\mathrm{dsh}}}(\boldsymbol{k})\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}\ominus\boldsymbol{y})$$

for some coefficients $\widehat{K^{\mathrm{dsh}}}(\boldsymbol{k})\ge 0$. If K is the unanchored Sobolev space, then it can be shown that the Walsh coefficients of the digital shift-invariant kernel satisfy

$$|\widehat{K^{\mathrm{dsh}}}(\boldsymbol{k})| = b^{-2\mu_1(\boldsymbol{k})}\prod_{j=1}^{s}\upsilon_b(k_j),$$

for some function $\upsilon_b(k)$ (see (6.9) below). This leads to an expression of the digitally shift-averaged worst-case error for the unanchored Sobolev space using a QMC rule based on a digital net.

Theorem 6.6 (digitally shift-averaged worst-case error for digital nets). The digitally shift-averaged worst-case error for a digital net P in the weighted unanchored Sobolev space $H(K_{s,\boldsymbol{\gamma}})$ satisfies

$$[e^{\mathrm{dsh}}_{b^m,s}(P;H(K_{s,\boldsymbol{\gamma}}))]^2 = \sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:s\}}\gamma_{\mathfrak{u}}\sum_{\boldsymbol{k}_{\mathfrak{u}}\in\mathcal{D}^*_{\mathfrak{u}}} b^{-2\mu_1(\boldsymbol{k}_{\mathfrak{u}})}\prod_{j\in\mathfrak{u}}\upsilon_b(k_j), \tag{6.7}$$

where

$$\mathcal{D}^*_{\mathfrak{u}} = \left\{\boldsymbol{k}_{\mathfrak{u}}\in\mathbb{N}^{|\mathfrak{u}|} : \sum_{j\in\mathfrak{u}} C_j^{\top}\vec{k}_j \equiv \vec{0}\right\}, \tag{6.8}$$

and

$$\upsilon_b(k) := \frac{1}{2}\left(\frac{1}{\sin^2(\kappa_{a-1}\pi/b)} - \frac{1}{3}\right), \tag{6.9}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

where $k = \kappa_{a-1}b^{a-1} + \kappa_{a-2}b^{a-2} + \cdots + \kappa_0$ with $\kappa_{a-1} \in \{1, 2, \ldots, b-1\}$. In particular, if $b = 2$ we have $\kappa_{a-1} = 1$ and $\upsilon_b(k)$ is a constant independent of k.

For the weighted anchored Sobolev space $H(K_{s,\boldsymbol{\gamma}})$ with anchor c we have

$$[e^{\mathrm{dsh}}_{b^m,s}(P; H(K_{s,\boldsymbol{\gamma}}))]^2 = \sum_{\emptyset \neq \mathfrak{v} \subseteq \{1:s\}} \widetilde{\gamma}_{s,\mathfrak{v}} \sum_{\boldsymbol{k}_{\mathfrak{v}} \in \mathcal{D}^*_{\mathfrak{v}}} b^{-2\mu_1(\boldsymbol{k}_{\mathfrak{v}})} \prod_{j \in \mathfrak{v}} \upsilon_b(k_j), \tag{6.10}$$

where $\upsilon_b(k)$ is defined as in (6.9) and $\widetilde{\gamma}_{s,\mathfrak{v}}$ are the auxiliary weights defined in (5.22).

Note that (6.7) and (6.10) are essentially weighted versions of the quantity T_ϑ in (6.4), with some additional factors $\upsilon_b(k)$ and a change of weights for the anchored Sobolev space.

The proof of Theorem 6.6 is somewhat intricate and was shown by Dick and Pillichshammer (2005) for the unanchored Sobolev space and by Dick, Kuo, Pillichshammer and Sloan (2005) for the anchored Sobolev space. Instead of giving the proof, we provide an example to show how the decay rate of the Walsh coefficients is connected to the smoothness of the integrand.

Example 6.7. To illustrate how to prove a bound on the decay rate of the Walsh coefficients we consider the simplest possible example. Let $f : [0,1] \to \mathbb{R}$ be absolutely continuous with

$$f(x) = f(1) - \int_0^1 f'(t) 1_{[x,1]}(t)\,\mathrm{d}t.$$

In this example let $b = 2$, and let $k = 2^{a-1} + \kappa_{a-2}2^{a-2} + \cdots + \kappa_0 \in \mathbb{N}$, for some $a \geq 1$. Then $\mathrm{wal}_k(x) \in \{-1, 1\}$ for all $x \in [0,1)$. Note that wal_k changes sign on intervals of the form $[\ell 2^{-a+1}, (\ell+1)2^{-a+1})$ for $0 \leq \ell < 2^{a-1}$, *i.e.*, it has the opposite sign on the interval $[\ell 2^{-a+1}, \ell 2^{-a+1} + 2^{-a})$ compared to the interval $[\ell 2^{-a+1} + 2^{-a}, (\ell+1)2^{-a+1})$. Thus

$$\begin{aligned}
|\widehat{f}(k)| = \left| \int_0^1 f(x)\overline{\mathrm{wal}_k(x)}\,\mathrm{d}x \right| &\leq \sum_{\ell=0}^{2^{a-1}-1} \int_{\ell 2^{-a+1}}^{\ell 2^{-a+1}+2^{-a}} |f(x) - f(x+2^{-a})|\,\mathrm{d}x \\
&\leq \int_0^{1-2^{-a}} |f(x) - f(x+2^{-a})|\,\mathrm{d}x \\
&\leq \int_0^1 |f'(t)| \int_0^{1-2^{-a}} 1_{[x,x+2^{-a})}(t)\,\mathrm{d}x\,\mathrm{d}t \\
&\leq 2^{-a} \int_0^1 |f'(t)|\,\mathrm{d}t < k^{-1} \left(\int_0^1 |f'(t)|^2 \mathrm{d}t \right)^{1/2}.
\end{aligned}$$

Thus for any $k \in \mathbb{N}_0$ we have

$$|\widehat{f}(k)| \leq \frac{1}{\max(1,k)} \|f\|_H,$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

where $\|\cdot\|_H$ denotes the norm in the unanchored Sobolev space. This can be generalized to arbitrary base b and dimensions $s > 1$.

In particular, if $K : [0,1]^2 \to \mathbb{R}$ is the reproducing kernel for the unanchored Sobolev space, then K has Walsh coefficients $\widehat{K}(k,\ell)$ which satisfy

$$|\widehat{K}(k,\ell)| \le Cb^{-\mu_1(k)-\mu_1(\ell)}.$$

For the corresponding digital shift-invariant kernel defined by (6.6), it can be shown that $\widehat{K^{\mathrm{dsh}}}(k) = \widehat{K}(k,k)$. We therefore obtain

$$|\widehat{K^{\mathrm{dsh}}}(k)| = |\widehat{K}(k,k)| \le Cb^{-2\mu_1(k)}.$$

6.4. CBC construction of polynomial lattice rules

Analogous to lattice rules, no optimal explicit construction of polynomial lattice rules is known beyond dimension 2. On the other hand, a CBC approach is also possible for polynomial lattice rules. An essential ingredient of the CBC algorithm is an easily computable quality criterion. We now show that the digitally shift-averaged worst-case error (6.7) can be computed efficiently.

Lemma 6.8. The digitally shift-averaged worst-case error for a digital net $P = \{\boldsymbol{t}_0, \dots, \boldsymbol{t}_{b^m-1}\}$ in the weighted unanchored Sobolev space $H(K_{s,\boldsymbol{\gamma}})$ satisfies

$$[e^{\mathrm{dsh}}_{b^m,s}(P; H(K_{s,\boldsymbol{\gamma}}))]^2 = \frac{1}{b^m} \sum_{i=0}^{b^m-1} \sum_{\emptyset \ne \mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}} \prod_{j \in \mathfrak{u}} \phi_b(t_{i,j}), \tag{6.11}$$

with

$$\phi_b(t) := \begin{cases} 6^{-1} - \frac{\tau_{a_0}(b-\tau_{a_0})}{b^{a_0+1}} & \text{if } 0 < t < 1, \\ 6^{-1} & \text{if } t = 0, \end{cases}$$

where, for $t \in (0,1)$ with b-adic expansion $t = \tau_1 b^{-1} + \tau_2 b^{-2} + \cdots$ (unique in the sense that infinitely many τ_r are different from $b-1$), $a_0 = -\lfloor \log_b t \rfloor$ is the smallest positive integer such that $\tau_1 = \tau_2 = \cdots = \tau_{a_0-1} = 0$ and $\tau_{a_0} \ne 0$.

Proof. We can write from (6.7) that

$$\begin{aligned} &[e^{\mathrm{dsh}}_{b^m,s}(P; H(K_{s,\boldsymbol{\gamma}}))]^2 \\ &= \sum_{\emptyset \ne \mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}} b^{-m} \sum_{i=0}^{b^m-1} \sum_{\boldsymbol{k}_{\mathfrak{u}} \in \mathbb{N}^{|\mathfrak{u}|}} b^{-2\mu_1(\boldsymbol{k}_{\mathfrak{u}})} \prod_{j \in \mathfrak{u}} \upsilon_b(k_j) \mathrm{wal}_{k_j}(t_{i,j}) \\ &= b^{-m} \sum_{i=0}^{b^m-1} \sum_{\emptyset \ne \mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}} \prod_{j \in \mathfrak{u}} \sum_{k=1}^{\infty} b^{-2\mu_1(k)} \upsilon_b(k) \mathrm{wal}_k(t_{i,j}). \end{aligned} \tag{6.12}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

To simplify the proof we only consider the case $b = 2$ in the following. Then $\upsilon_b(k) = 1/3$ for all $k \in \mathbb{N}$. Let $\omega_2 = \mathrm{e}^{2\pi \mathrm{i}/2} = -1$. Then we have

$$\sum_{k=2^{a-1}}^{2^a-1} \mathrm{wal}_k(t) = \prod_{r=1}^{a-1} \sum_{\kappa_{r-1}=0}^{1} \omega_2^{\kappa_{r-1}\tau_r} \sum_{\kappa_{a-1}=1}^{1} \omega_2^{\kappa_{a-1}\tau_a}$$
$$= \begin{cases} 2^{a-1} & \text{if } \tau_1 = \tau_2 = \cdots = \tau_a = 0, \\ -2^{a-1} & \text{if } \tau_1 = \tau_2 = \cdots = \tau_{a-1} = 0, \tau_a \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Thus

$$\sum_{k=1}^{\infty} 2^{-2\mu_1(k)} \mathrm{wal}_k(t) = \sum_{a=1}^{\infty} 2^{-2a} \sum_{k=2^{a-1}}^{2^a-1} \mathrm{wal}_k(t)$$
$$= \sum_{a=1}^{a_0-1} 2^{-2a} 2^{a-1} - 2^{-2a_0} 2^{a_0-1} = 2^{-1} - 2^{-a_0}(1 + 2^{-1}).$$

For $t = 0$ we have

$$\sum_{k=1}^{\infty} 2^{-2\mu_1(k)} \mathrm{wal}_k(0) = \sum_{a=1}^{\infty} 2^{-2a} 2^{a-1} = \frac{1}{2}.$$

The result now follows from (6.12). □

The expression (6.11) can be easily evaluated by a computer. Thus we can use the following CBC algorithm to construct polynomial lattice rules. In the following we write

$$[e^{\mathrm{dsh}}_{b^m,s}(P; H(K_{s,\boldsymbol{\gamma}}))]^2 = [e^{\mathrm{dsh}}_{b^m,s}(\boldsymbol{q})]^2 = [e^{\mathrm{dsh}}_{b^m,s}(q_1, \ldots, q_s)]^2$$

if the digital net is a polynomial lattice rule with generating vector $\boldsymbol{q} = (q_1, \ldots, q_s) \in (G^*_{b,m})^s$, where $G^*_{b,m} = \{q \in \mathbb{Z}_b[x] \setminus \{0\} : \deg(q) < m\}$.

Algorithm 6.9 (CBC construction). Given: m, $s_{\max}$, and weights $\gamma_{\mathfrak{u}}$.

(1) Set $q_1 = 1$.

(2) For $s = 2, 3, \ldots, s_{\max}$, choose q_s in $G^*_{b,m}$ to minimize $[e^{\mathrm{dsh}}_{b^m,s}(q_1, \ldots, q_s)]^2$.

The next theorem states that the CBC algorithm yields a convergence rate close to $O(b^{-m})$.

Theorem 6.10 (optimal CBC error bound). The generating vector $\boldsymbol{q}$ constructed by the CBC algorithm, minimizing the shift-averaged worst-case error for the unanchored Sobolev space $e^{\mathrm{dsh}}_{b^m,s}(\boldsymbol{q})$ in each step, satisfies

$$[e^{\mathrm{dsh}}_{b^m,s}(\boldsymbol{q})]^2 \leq \left(\frac{1}{b^m - 1} \sum_{\emptyset \neq \mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}}^{\lambda} (\rho_b(\lambda))^{|\mathfrak{u}|} \right)^{1/\lambda}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

for any $1/2 < \lambda \le 1$, where

$$\rho_b(\lambda) := \begin{cases} \dfrac{1}{3^\lambda(2^{2\lambda}-2)} & \text{for } b = 2, \\ \dfrac{(b-1)(4b^2-9)^\lambda}{54^\lambda(b^{2\lambda}-b)} & \text{for } b > 2. \end{cases}$$

Proof. Since the algorithm constructs the polynomial lattice rule inductively with respect to the dimension, the proof also uses induction. The argument follows along the same lines as for the lattice rule case. To simplify the proof we only consider $b = 2$ in the following. In this case $\upsilon_2(k) = 1/3$ for all $k \in \mathbb{N}$ in (6.7).

First note that we have for $1/2 < \lambda \le 1$ that

$$\sum_{k=1}^{\infty} 2^{-2\lambda\mu_1(k)} = \sum_{a=1}^{\infty} 2^{-2\lambda a} 2^{a-1} = \frac{1}{2^{2\lambda}-2}$$

and

$$\sum_{\substack{k=1 \\ 2^m \mid k}}^{\infty} 2^{-2\lambda\mu_1(k)} = \sum_{\ell=1}^{\infty} 2^{-2\lambda\mu_1(b^m\ell)} = \sum_{\ell=1}^{\infty} 2^{-2\lambda m} 2^{-2\lambda\mu_1(\ell)} = 2^{-2\lambda m} \frac{1}{2^{2\lambda}-2}.$$

For a polynomial lattice rule with generating vector $(q_1, \ldots, q_s)$ and modulus p, the set $\mathcal{D}^*_{\mathfrak{u}}$ in (6.8) is given by

$$\mathcal{D}^*_{\mathfrak{u}} = \left\{ \boldsymbol{k}_{\mathfrak{u}} \in \mathbb{N}^{|\mathfrak{u}|} : \sum_{j \in \mathfrak{u}} \mathrm{tr}_m(k_j) q_j \equiv 0 \pmod{p} \right\},$$

where for $k_j = \kappa_{j,0} + \kappa_{j,1} 2 + \cdots \in \mathbb{N}_0$ we set $\mathrm{tr}_m(k_j) = \kappa_{j,0} + \kappa_{j,1} x + \cdots + \kappa_{j,m-1} x^{m-1}$.

In dimension one we have

$$\begin{aligned} [e^{\mathrm{dsh}}_{2^m,s}(q_1)]^2 &= \gamma_{\{1\}} 3^{-1} \sum_{\substack{k=1 \\ 2^m \mid k}}^{\infty} 2^{-2\mu_1(k)} \\ &= 2^{-2m} \gamma_{\{1\}} 3^{-1} \sum_{\ell=1}^{\infty} 2^{-2\mu_1(\ell)} = 3^{-1} 2^{-2m-1} \gamma_{\{1\}}. \end{aligned}$$

Thus the result holds for dimension one.

Suppose, for some $1 \le d < s$, we have $(q_1, \ldots, q_d) \in (G^*_{2,m})^d$ and

$$[e^{\mathrm{dsh}}_{2^m,d}(q_1, \ldots, q_d)]^2 \le \left(\frac{1}{2^m - 1} \sum_{\emptyset \ne \mathfrak{u} \subseteq \{1:d\}} \gamma^\lambda_{\mathfrak{u}} \left(\frac{1}{3^\lambda(2^{2\lambda}-2)} \right)^{|\mathfrak{u}|} \right)^{1/\lambda}, \quad (6.13)$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

for all $1/2 < \lambda \le 1$. Now we consider $(q_1, \dots, q_d, q_{d+1})$. We have

$$\begin{aligned}
&[e^{\mathrm{dsh}}_{2^m,d+1}(q_1,\dots,q_d,q_{d+1})]^2 \\
&= \sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:d\}} \gamma_{\mathfrak{u}} 3^{-|\mathfrak{u}|} \sum_{\boldsymbol{k}_{\mathfrak{u}}\in\mathcal{D}^*_{\mathfrak{u}}} 2^{-2\mu_1(\boldsymbol{k}_{\mathfrak{u}})} + \sum_{\{d+1\}\subseteq\mathfrak{u}\subseteq\{1:d+1\}} \gamma_{\mathfrak{u}} 3^{-|\mathfrak{u}|} \sum_{\boldsymbol{k}_{\mathfrak{u}}\in\mathcal{D}^*_{\mathfrak{u}}} 2^{-2\mu_1(\boldsymbol{k}_{\mathfrak{u}})} \\
&= [e^{\mathrm{dsh}}_{2^m,d}(q_1,\dots,q_d)]^2 + \theta(q_{d+1}), \qquad (6.14)
\end{aligned}$$

where

$$\theta(q_{d+1}) = \sum_{\{d+1\}\subseteq\mathfrak{u}\subseteq\{1:d+1\}} \gamma_{\mathfrak{u}} \sum_{\boldsymbol{k}_{\mathfrak{u}}\in\mathcal{D}^*_{\mathfrak{u}}} b^{-2\mu_1(\boldsymbol{k}_{\mathfrak{u}})}.$$

According to the algorithm, q_{d+1} is chosen such that the mean-square worst-case error $[e^{\mathrm{dsh}}_{2^m,d+1}(q_1,\dots,q_d,q_{d+1})]^2$ is minimized. Since the only dependency on q_{d+1} is in $\theta(q_{d+1})$, we have $\theta(q_{d+1}) \le \theta(q)$ for all $q \in G^*_{2,m}$, which implies that for any $1/2 < \lambda \le 1$ we have $\theta(q_{d+1})^\lambda \le \theta(q)^\lambda$ for all $q \in G^*_{2,m}$. This leads to

$$\theta(q_{d+1}) \le \left(\frac{1}{2^m-1} \sum_{q\in G^*_{2,m}} \theta(q)^\lambda\right)^{1/\lambda}.$$

We obtain a bound on $\theta(q_{d+1})$ through this last inequality.

For λ satisfying $1/2 < \lambda \le 1$ it follows from Jensen's inequality that

$$\theta(q)^\lambda \le \sum_{\{d+1\}\subseteq\mathfrak{u}\subseteq\{1:d+1\}} \gamma_{\mathfrak{u}}^\lambda 3^{-\lambda|\mathfrak{u}|} \sum_{\boldsymbol{k}_{\mathfrak{u}}\in\mathcal{D}^*_{\mathfrak{u}}} 2^{-2\lambda\mu_1(\boldsymbol{k}_{\mathfrak{u}})}.$$

The condition $\boldsymbol{k}_{\mathfrak{u}} \in \mathcal{D}^*_{\mathfrak{u}}$ is equivalent to the equation

$$\mathrm{tr}_m(k_1)q_1 + \cdots + \mathrm{tr}_m(k_d)q_d \equiv -\mathrm{tr}_m(k_{d+1})q \pmod p.$$

If k_{d+1} is a multiple of 2^m, then $\mathrm{tr}_m(k_{d+1}) = 0$ and the corresponding term in the sum is independent of q. If k_{d+1} is not a multiple of 2^m, then $\mathrm{tr}_m(k_{d+1})$ can be any polynomial in $G^*_{2,m}$. Moreover, since $q \ne 0$ and p is irreducible, $\mathrm{tr}_m(k_{d+1})q$ is never a multiple of p.

By averaging over all $q \in G^*_{2,m}$, with the above discussion in mind, we obtain

$$\begin{aligned}
&\frac{1}{2^m-1} \sum_{q\in G^*_{2,m}} \theta(q)^\lambda \\
&\le \sum_{\substack{k_{d+1}=1\\ 2^m\mid k_{d+1}}}^{\infty} 2^{-2\lambda\mu_1(k_{d+1})} \sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:d\}} \gamma^\lambda_{\mathfrak{u}\cup\{d+1\}} 3^{-\lambda(|\mathfrak{u}|+1)} \sum_{\substack{\boldsymbol{k}_{\mathfrak{u}}\in\mathbb{N}^{|\mathfrak{u}|}\\ \mathrm{tr}_m(\boldsymbol{k}_{\mathfrak{u}})\cdot\boldsymbol{q}_{\mathfrak{u}}\equiv 0 \,(\mathrm{mod}\, p)}} 2^{-2\lambda\mu_1(\boldsymbol{k}_{\mathfrak{u}})} \\
&\quad + \sum_{\substack{k_{d+1}=1\\ 2^m\nmid k_{d+1}}}^{\infty} \frac{2^{-2\lambda\mu_1(k_{d+1})}}{2^m-1} \sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:d\}} \gamma^\lambda_{\mathfrak{u}\cup\{d+1\}} 3^{-\lambda(|\mathfrak{u}|+1)} \sum_{\substack{\boldsymbol{k}_{\mathfrak{u}}\in\mathbb{N}^{|\mathfrak{u}|}\\ \mathrm{tr}_m(\boldsymbol{k}_{\mathfrak{u}})\cdot\boldsymbol{q}_{\mathfrak{u}}\not\equiv 0 \,(\mathrm{mod}\, p)}} 2^{-2\lambda\mu_1(\boldsymbol{k}_{\mathfrak{u}})}
\end{aligned}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

$$\le 2^{-2\lambda m}\frac{1}{3^{\lambda(|\mathfrak{u}|+1)}(2^{2\lambda}-2)}\sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:d\}}\gamma_{\mathfrak{u}\cup\{d+1\}}^{\lambda}\sum_{\substack{\boldsymbol{k}_{\mathfrak{u}}\in\mathbb{N}^{|\mathfrak{u}|}\\ \mathrm{tr}_m(\boldsymbol{k}_{\mathfrak{u}})\cdot\boldsymbol{q}_{\mathfrak{u}}\equiv 0\,(\mathrm{mod}\,p)}}2^{-2\lambda\mu_1(\boldsymbol{k}_{\mathfrak{u}})}$$

$$+\frac{1}{2^m-1}\frac{1-2^{-2\lambda m}}{3^{\lambda(|\mathfrak{u}|+1)}(2^{2\lambda}-2)}\sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:d\}}\gamma_{\mathfrak{u}\cup\{d+1\}}^{\lambda}\sum_{\substack{\boldsymbol{k}_{\mathfrak{u}}\in\mathbb{N}^{|\mathfrak{u}|}\\ \mathrm{tr}_m(\boldsymbol{k}_{\mathfrak{u}})\cdot\boldsymbol{q}_{\mathfrak{u}}\not\equiv 0\,(\mathrm{mod}\,p)}}2^{-2\lambda\mu_1(\boldsymbol{k}_{\mathfrak{u}})}$$

$$\le\frac{1}{2^m-1}\frac{1}{3^{\lambda(|\mathfrak{u}|+1)}(2^{2\lambda}-2)}\sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:d\}}\gamma_{\mathfrak{u}\cup\{d+1\}}^{\lambda}\sum_{\boldsymbol{k}_{\mathfrak{u}}\in\mathbb{N}^{|\mathfrak{u}|}}2^{-2\lambda\mu_1(\boldsymbol{k}_{\mathfrak{u}})}$$

$$=\frac{1}{2^m-1}\sum_{\{d+1\}\subseteq\mathfrak{u}\subseteq\{1:d+1\}}\gamma_{\mathfrak{u}}^{\lambda}\left(\frac{1}{3^{\lambda}(2^{2\lambda}-2)}\right)^{|\mathfrak{u}|}.$$

Thus we conclude that

$$\theta(q_{d+1})\le\left(\frac{1}{2^m-1}\sum_{\{d+1\}\subseteq\mathfrak{u}\subseteq\{1:d+1\}}\gamma_{\mathfrak{u}}^{\lambda}\left(\frac{1}{3^{\lambda}(2^{2\lambda}-2)}\right)^{|\mathfrak{u}|}\right)^{1/\lambda}$$

which, together with (6.13) and (6.14), yields

$$[e^{\mathrm{dsh}}_{2^m,d+1}(q_1,\ldots,q_d,q_{d+1})]^{2\lambda}=[e^{\mathrm{dsh}}_{b^m,d}(q_1,\ldots,q_d)]^{2\lambda}+\theta(q_{d+1})^{\lambda}$$
$$\le\frac{1}{2^m-1}\sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:d+1\}}\gamma_{\mathfrak{u}}^{\lambda}\left(\frac{1}{3^{\lambda}(2^{2\lambda}-2)}\right)^{|\mathfrak{u}|}.$$

Hence the result follows by induction. □

The choice of q_d at each step is independent of λ and therefore the error bound holds for all values of $\lambda\in(1/2,1]$. The optimal convergence rate close to $O(b^{-m})$ is obtained with $\lambda\to 1/2$.

6.5. Fast CBC construction of polynomial lattice rules

In the previous subsection we showed how to construct, for a given modulus p, a generating vector $\boldsymbol{q}$ which yields a polynomial lattice rule that achieves a small integration error using a CBC algorithm. We have seen in Section 5.5 that FFT can be used to reduce the construction cost of the CBC construction of lattice rules. We now show that the same technique can be used for polynomial lattice rules by outlining the key differences from lattice rules. For product weights, it is possible to construct, for a given polynomial p with $\deg(p)=m$, an s-dimensional generating vector $\boldsymbol{q}$ in $O(s\,m\,b^m)$ operations, compared to $O(s^2\,b^{2m})$ operations for a naive implementation of the CBC algorithm.

Let b be a prime. Throughout this subsection we consider the polynomial $p\in\mathbb{Z}_b[x]$ with $\deg(p)=m$ to be irreducible. We restrict ourselves to

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

product weights $\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_j$, for a sequence of positive real numbers γ_j. The extension to POD weights can be done as in the lattice rule case: see Section 5.6.

Using Algorithm 6.9 we construct, component by component, a generating vector $\boldsymbol{q} = (q_1, \ldots, q_s) \in (G^*_{b,m})^s$ such that for all $1 \le d \le s$ the quantity $e_d^2(q_d) := [e^{\mathrm{dsh}}_{b^m,d}(q_1, \ldots, q_{d-1}, q_d)]^2$ is minimized with respect to q_d for fixed $q_1, \ldots, q_{d-1}$. Assume that $q_1, \ldots, q_{d-1}$ are already constructed. Then we have to find $q \in G^*_{b,m}$ which minimizes

$$e_d^2(q) = [e^{\mathrm{dsh}}_{b^m,d-1}(q_1, \ldots, q_{d-1})]^2 + \gamma_d \phi_b(0) \frac{\prod_{j=1}^{d-1}(1+\gamma_j\phi_b(0))}{b^m} + \frac{\gamma_d}{b^m} \sum_{i=1}^{b^m-1} \eta_{d-1}(i) \phi_b\left(\upsilon_m\left(\frac{q(x)i(x)}{p(x)}\right)\right),$$

where we used Theorem 2.21, separated out the $i = 0$ term and introduced the column vector $\boldsymbol{\eta}_{d-1} = \big(\eta_{d-1}(1), \ldots, \eta_{d-1}(b^m-1)\big)^\top$, where

$$\eta_{d-1}(i) = \prod_{j=1}^{d-1}(1+\gamma_j\phi_b(t_{i,j})), \quad i = 1, \ldots, b^m - 1.$$

Note that the vector $\boldsymbol{\eta}_{d-1}$ remains fixed for fixed d regardless of the choice of polynomial q. We set $\boldsymbol{\eta}_0 = (1, \ldots, 1) \in \mathbb{R}^{b^m-1}$. The vector $\boldsymbol{\eta}_{d-1}$ can be computed as part of the CBC algorithm with the update formula $\eta_{d-1}(i) = \eta_{d-2}(i)(1+\gamma_{d-1}\phi_b(t_{i,d-1}))$.

For short we write $\omega := \phi_b \circ \upsilon_m$ from now on. We define the $(b^m-1) \times (b^m-1)$ matrix

$$\boldsymbol{\Omega}_p := \left(\omega\left(\frac{q(x)i(x)}{p(x)}\right)\right)_{\substack{q=1,\ldots,b^m-1\\ i=1,\ldots,b^m-1}}.$$

Further, let $\boldsymbol{e}_d^2 = (e_d^2(1), \ldots, e_d^2(b^m-1))^\top$ be the column vector collecting the quantities $e_d^2(q)$ and let $\mathbf{1}$ be the (b^m-1)-dimensional column vector whose components are all 1. Then we have

$$\boldsymbol{e}_d^2 = \boldsymbol{e}_{d-1}^2 + \frac{\prod_{j=1}^{d}(1+\gamma_j\phi_b(0))}{b^m}\mathbf{1} + \frac{\gamma_d}{b^m}\boldsymbol{\Omega}_p\boldsymbol{\eta}_{d-1}.$$

As we are only interested in which q minimizes $e_d^2(q)$, but not the value of $e_d^2(q)$ itself, we only need to compute $\boldsymbol{\Omega}_p\boldsymbol{\eta}_{d-1}$.

The entries of the matrix $\boldsymbol{\Omega}_p$ are of the form $\omega\big(\frac{q(x)i(x)}{p(x)}\big)$, where the product of the non-zero polynomials q and i has to be evaluated in the field $\mathbb{Z}_b[x]/(p)$, and thus modulo p. Since the multiplicative group of every finite field is cyclic, we can find a primitive element g which generates all elements of the multiplicative group $(\mathbb{Z}_b[x]/(p))^*$ $(= (\mathbb{Z}_b[x]/(p)) \setminus \{0\})$. That is, there

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

is a $g \in (\mathbb{Z}_b[x]/(p))^*$ such that $(\mathbb{Z}_b[x]/(p))^* = \{g^0, g^1, g^2, \ldots, g^{b^m-1}\}$. Thus, we can write the product of any non-zero polynomials $q, i \in \mathbb{Z}_b[x]/(p)$ as a power of the polynomial g. Nuyens and Cools (2006*a*, 2006*b*) then suggest permuting the rows of $\mathbf{\Omega}_p$ by the positive powers of the primitive polynomial g and the columns by the negative powers of the same primitive polynomial. This procedure is often called Rader transform, since it goes back to an idea of Rader (1968).

We now describe the Rader transform in detail. Let $g(x)$ be a primitive element in $(\mathbb{Z}_b[x]/(p))^*$. We define a $(b^m-1) \times (b^m-1)$ matrix $\Pi(g) = (\pi_{k,\ell}(g))_{1 \le k,\ell < b^m}$, where

$$\pi_{k,\ell}(g) = \begin{cases} 1 & \text{if } k(x) \equiv g(x)^\ell \pmod{p(x)}, \\ 0 & \text{otherwise.} \end{cases}$$

Here $k(x)$ denotes the polynomial which is associated with the integer k. Since g is a primitive element it follows that each row and each column of $\Pi(g)$ has exactly one entry, which is 1, and the remaining entries are 0. Further, $\Pi(g)\Pi(g)^\top = I$, the identity matrix. In fact, the matrix $\Pi(g)$ is a permutation matrix. That is, for any $(b^m-1) \times (b^m-1)$ matrix C, $\Pi(g)C$ just changes the order of the rows of C and $C\Pi(g)$ only changes the order of the columns of C.

Let $C = (c_{k,\ell})_{1 \le k,\ell < b^m}$ and

$$C = (\Pi(g))^\top \mathbf{\Omega}_p \Pi(g^{-1}).$$

Then

$$c_{k,\ell} = \sum_{u,v=1}^{b^m-1} \pi_{u,k}(g)\omega\left(\frac{u(x)v(x)}{p(x)}\right)\pi_{v,\ell}(g^{-1}) = \omega\left(\frac{g(x)^k g(x)^{-\ell}}{p(x)}\right).$$

Let

$$c_r = \omega\left(\frac{g(x)^r}{p(x)}\right).$$

Observe that $c_r = c_{r'}$ for all $r, r' \in \mathbb{Z}$ with $r \equiv r' \pmod{b^m-1}$, since $g(x)^{b^m-1} = 1$. Then we have $c_{k,\ell} = c_{k-\ell}$, and therefore C is circulant (*cf.* Section 5.5).

Note that the circulant matrix C is fully determined by its first column $\boldsymbol{c} = (c_0, c_1, c_2, \ldots, c_{b^m-2})^\top$. Such matrices have a similarity transform which has the Fourier matrix as its eigenvectors. For $n \in \mathbb{N}$, let $F_n := (f_{k,\ell})_{k,\ell=0}^{n-1}$ be the Fourier matrix of order n given by $f_{k,\ell} = \omega_n^{-k\ell}$, where $\omega_n = \mathrm{e}^{2\pi\mathrm{i}/n}$. Note that F_n is symmetric and $\frac{1}{n}F_n\overline{F}_n = I$, the identity matrix. Furthermore, let $\mathrm{diag}(a_1, \ldots, a_n)$ be the $n \times n$ diagonal matrix $A = (A_{i,j})_{i,j=1}^n$ with $A_{i,i} = a_i$ for $1 \le i \le n$ and $A_{i,j} = 0$ for $i \ne j$. Then we have (*cf.* Section 5.5)

$$\mathbf{\Omega}_p = \frac{1}{b^m-1}\Pi(g)\overline{F}_{b^m-1}DF_{b^m-1}\Pi(g^{-1})^\top,$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

where $\Pi(g), \Pi(g^{-1})^\top$ are permutation matrices, F_{b^m-1} is a Fourier matrix, $\overline{F}_{b^m-1}$ its complex conjugate, and D is the diagonal matrix $D = \mathrm{diag}(\boldsymbol{F}_{b^m-1}\,\boldsymbol{c})$.

For any vector $\boldsymbol{x} = (x_1, \dots, x_{b^m-1})^\top \in (\mathbb{C}^{b^m-1})^\top$, the matrix–vector multiplications $\Pi(g)\boldsymbol{x}$, $D\boldsymbol{x}$, and $\Pi(g^{-1})^\top\boldsymbol{x}$ can be done in $O(b^m)$ operations (in fact, in the implementation of the algorithm the permutations using $\Pi(g)$ and $\Pi(g^{-1})$ can be avoided altogether: see Section 5.5). Further, $F_{b^m-1}\boldsymbol{x}$ and $\overline{F}_{b^m-1}\boldsymbol{x}$ can be computed in $O(mb^m)$ operations using the fast Fourier transform.

Therefore a fast CBC algorithm requires only $O(s\,m\,b^m)$ operations by using $O(b^m)$ memory space. This is a significant speed-up compared to a straightforward implementation. Only through this reduction of the construction cost does the CBC algorithm become applicable for the generation of polynomial lattice point sets with reasonably large cardinality.

6.6. Integration of smooth functions

It is well understood that the convergence rate for numerical integration depends on the smoothness of the integrand. In the following we consider the weighted unanchored Sobolev space of smoothness $\alpha > 1$, $H(K_{s,\boldsymbol{\gamma},\alpha})$, already encountered in Section 5.9; see also Example 3.8 for the one-dimensional case. The reproducing kernel is given by (5.34), and the inner product is

$$\begin{aligned}\langle f, g\rangle_{s,\boldsymbol{\gamma},\alpha} = \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}^{-1} \sum_{\mathfrak{v}\subseteq\mathfrak{u}} \sum_{\boldsymbol{r}_{\mathfrak{v}}\in\{1:\alpha-1\}^{|\mathfrak{v}|}} &\int_{[0,1]^{|\mathfrak{u}\setminus\mathfrak{v}|}} \left(\int_{[0,1]^{|\{1:s\}\setminus\mathfrak{u}|}} \int_{[0,1]^{|\mathfrak{v}|}} \frac{\partial^{|\boldsymbol{r}_{\mathfrak{v}}|_1+\alpha|\mathfrak{u}\setminus\mathfrak{v}|} f}{\partial \boldsymbol{x}_{(\boldsymbol{0},\boldsymbol{r},\boldsymbol{\alpha})}}\,\mathrm{d}\boldsymbol{x}_{\mathfrak{v}}\,\mathrm{d}\boldsymbol{x}_{\{1:s\}\setminus\mathfrak{u}}\right) \\ &\times \left(\int_{[0,1]^{|\{1:s\}\setminus\mathfrak{u}|}} \int_{[0,1]^{|\mathfrak{v}|}} \frac{\partial^{|\boldsymbol{r}_{\mathfrak{v}}|_1+\alpha|\mathfrak{u}\setminus\mathfrak{v}|} g}{\partial \boldsymbol{x}_{(\boldsymbol{0},\boldsymbol{r},\boldsymbol{\alpha})}}\,\mathrm{d}\boldsymbol{x}_{\mathfrak{v}}\,\mathrm{d}\boldsymbol{x}_{\{1:s\}\setminus\mathfrak{u}}\right) \mathrm{d}\boldsymbol{x}_{\mathfrak{u}\setminus\mathfrak{v}},\end{aligned}$$

where $|\boldsymbol{r}_{\mathfrak{v}}|_1 := \sum_{j\in\mathfrak{v}} |r_j|$, and

$$\frac{\partial^{|\boldsymbol{r}_{\mathfrak{v}}|_1+\alpha|\mathfrak{u}\setminus\mathfrak{v}|} f}{\partial \boldsymbol{x}_{(\boldsymbol{0},\boldsymbol{r},\boldsymbol{\alpha})}}$$

stands for the partial derivatives of f of order r_j for $j \in \mathfrak{v}$, of order α for $j \in \mathfrak{u}\setminus\mathfrak{v}$ and order 0 otherwise, where $\boldsymbol{r}_{\mathfrak{v}} = (r_j)_{j\in\mathfrak{v}}$. Let the corresponding norm be denoted by $\|f\|_{s,\boldsymbol{\gamma},\alpha} = \sqrt{\langle f, f\rangle_{s,\boldsymbol{\gamma},\alpha}}$.

For functions in $H(K_{s,\boldsymbol{\gamma},\alpha})$ it is known that a convergence rate of order $n^{-\alpha+\delta}$ for any $\delta > 0$ can be achieved with a suitable algorithm. In the previous subsections we have used the decay of the Walsh coefficients for functions in $H(K_{s,\boldsymbol{\gamma},1})$ to show a convergence rate of order $n^{-1+\delta}$. To generalize this approach to higher order we need to investigate how the

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Walsh coefficients of smooth functions decay. Since the proof is involved we describe the main idea, leaving out technical details.

To simplify the notation we restrict ourselves to base $b = 2$. Let $k \in \mathbb{N}$ be given by $k = 2^{a_1-1} + 2^{a_2-1} + \cdots + 2^{a_\nu-1}$ for $a_1 > a_2 > \cdots > a_\nu > 0$. We define for $\alpha \geq 2$ an integer

$$\mu_\alpha(k) := \begin{cases} a_1 + a_2 + \cdots + a_{\min\{\alpha,\nu\}} & \text{if } k \in \mathbb{N}, \\ 0 & \text{if } k = 0. \end{cases}$$

This generalizes the definition of $\mu_1(k)$ given by (6.3). Note that the Walsh functions in base $b = 2$ only take on the values 1 or -1. Thus in the following we write wal_k instead of $\overline{\mathrm{wal}_k}$.

Let $f \in H(K_{s,\boldsymbol{\gamma},\alpha})$ and $J_k(x) := \int_0^x \mathrm{wal}_k(t)\,\mathrm{d}t$. Then

$$\begin{aligned} \widehat{f}(k) = \int_0^1 f(x)\mathrm{wal}_k(x)\,\mathrm{d}x &= \big[f(x)J_k(x)\big]_{x=0}^1 - \int_0^1 f'(x)J_k(x)\,\mathrm{d}x \\ &= -\int_0^1 f'(x)J_k(x)\,\mathrm{d}x, \end{aligned} \tag{6.15}$$

as $\int_0^1 \mathrm{wal}_k(x)\,\mathrm{d}x = 0$.

We would now like to relate the Walsh coefficient $\widehat{f}(k)$ to some Walsh coefficient of f'. This is done by using the Walsh series expansion of J_k.

Let $k' := k - 2^{a_1-1}$, and hence $0 \leq k' < 2^{a_1-1}$. Then it can be shown that

$$J_k(x) = 2^{-a_1-1}\left(\mathrm{wal}_{k'}(x) - \sum_{c=1}^{\infty} 2^{-c}\mathrm{wal}_{2^{a_1+c-1}+k}(x)\right).$$

Substituting this into (6.15), we obtain approximately

$$\widehat{f}(k) \approx -2^{-a_1}\int_0^1 f'(x)\mathrm{wal}_{k'}(x)\,\mathrm{d}x = -2^{-a_1}\widehat{f'}(k').$$

In fact we obtain an infinite sum on the right-hand side, but the main term is the first one: the remaining terms can be bounded by a suitable expression.

The above step can be repeated. To do so, let $k'' := k' - 2^{a_2-1}$, $k''' := k'' - 2^{a_3-1}$ and so on, where $k^{(\nu)} = 0$ and $k^{(\tau)} = 0$ for $\tau \geq \nu$. We can repeat the last step τ times until either $f^{(\tau)}$ is no longer differentiable, or $k^{(\tau)} = 0$, that is, we can repeat it $\min(\alpha, \nu)$ times. Hence

$$\begin{aligned} \widehat{f}(k) &\approx 2^{-a_1}\widehat{f'}(k^{(1)}) \approx 2^{-a_1-a_2}\widehat{f''}(k^{(2)}) \\ &\;\;\vdots \\ &\approx 2^{-a_1-\cdots-a_{\min(\alpha,\nu)}}\widehat{f^{(\min(\alpha,\nu))}}(k^{(\min(\alpha,\nu))}). \end{aligned}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Taking the absolute value and using some estimation we obtain

$$|\widehat{f}(k)| \lessapprox 2^{-a_1-\cdots-a_{\min(\alpha,\nu)}} |\widehat{f}^{(\min(\alpha,\nu))}(k^{(\min(\alpha,\nu))})|$$
$$\le 2^{-\mu_\alpha(k)} |\widehat{f}^{(\min(\alpha,\nu))}(k^{(\min(\alpha,\nu))})| \le 2^{-\mu_\alpha(k)} \int_0^1 \left| f^{(\min(\alpha,\nu))}(x) \right| \mathrm{d}x,$$

where we used

$$|\widehat{f}^{(\min(\alpha,\nu))}(k^{(\min(\alpha,\nu))})| = \left| \int_0^1 f^{(\min(\alpha,\nu))}(x) \overline{\mathrm{wal}_{k^{(\min(\alpha,\nu))}}(x)} \, \mathrm{d}x \right|$$
$$\le \int_0^1 \left| f^{(\min(\alpha,\nu))}(x) \right| |\overline{\mathrm{wal}_{k^{(\min(\alpha,\nu))}}(x)}| \, \mathrm{d}x$$
$$= \int_0^1 \left| f^{(\min(\alpha,\nu))}(x) \right| \mathrm{d}x.$$

Thus if f is α-times differentiable, we obtain

$$|\widehat{f}(k)| \lessapprox C_f 2^{-\mu_\alpha(k)},$$

where $C_f > 0$ is a constant which depends only on f and α. By some modification of the above approach it can be shown that the constant C_f can be replaced by a constant which depends only on α (but not on f) and the norm of f, that is, we have

$$|\widehat{f}(k)| \le C_\alpha \|f\|_{1,(1),\alpha} 2^{-\mu_\alpha(k)}.$$

The same holds for dimension $s > 1$ and integer base $b > 2$, where the constant additionally depends on s and b, that is,

$$|\widehat{f}(\boldsymbol{k})| \le C_{\alpha,b,s} \|f\|_{s,\mathbf{1},\alpha} b^{-\mu_\alpha(\boldsymbol{k})},$$

where $\mu_\alpha(\boldsymbol{k}) = \mu_\alpha(k_1) + \cdots + \mu_\alpha(k_s)$ for $\boldsymbol{k} = (k_1, \ldots, k_s)$ and $\mathbf{1}$ is the set of weights which are 1 for each projection. For some values of b, this constant $C_{\alpha,b,s}$ goes to 0 exponentially as s increases.

We can now begin to investigate numerical integration of functions in $H(K_{s,\boldsymbol{\gamma},\alpha})$ using digital nets. From Theorem 6.4 we conclude that the integration error for a digital net $P = \{\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{b^m-1}\}$ satisfies

$$\left| \int_{[0,1]^s} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} - \frac{1}{b^m} \sum_{i=0}^{b^m-1} f(\boldsymbol{t}_i) \right| \le C_{\alpha,b,s} \|f\|_{s,\mathbf{1},\alpha} \sum_{\boldsymbol{k} \in \mathcal{D} \setminus \{\mathbf{0}\}} b^{-\mu_\alpha(\boldsymbol{k})},$$

where $\mathcal{D}$ is the dual net defined by (6.2).

The last inequality separates the contribution of the function from the contribution of the QMC rule, that is, $\|f\|_{s,\mathbf{1},\alpha}$ depends only on the function f but not on the digital net, whereas $\sum_{\boldsymbol{k} \in \mathcal{D} \setminus \{\mathbf{0}\}} b^{-\mu_\alpha(\boldsymbol{k})}$ depends only on the generating matrices of the digital net and not on the function itself (only on the smoothness of f, *i.e.*, it is the same for all functions that have

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

smoothness α). Therefore, when considering the integration error we can now focus on the term $\sum_{\boldsymbol{k}\in\mathcal{D}\setminus\{\mathbf{0}\}} b^{-\mu_\alpha(\boldsymbol{k})}$.

It is possible to also introduce weights in the approach above. A precise result for the unanchored weighted Sobolev space $H(K_{s,\boldsymbol{\gamma},\alpha})$ for $\alpha > 1$ is the following.

Theorem 6.11. The square worst-case error for multivariate integration in the weighted unanchored Sobolev space $H(K_{s,\boldsymbol{\gamma},\alpha})$ of smoothness $\alpha > 1$ using a QMC rule based on a digital net P can be bounded by

$$[e(P, H(K_{s,\boldsymbol{\gamma},\alpha}))]^2 \le \sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}(D_{\alpha,b})^{|\mathfrak{u}|} \left(\sum_{\boldsymbol{k}_{\mathfrak{u}}\in\mathcal{D}^*_{\mathfrak{u}}} b^{-\mu_\alpha(\boldsymbol{k}_{\mathfrak{u}})}\right)^2,$$

where $\mathcal{D}^*_{\mathfrak{u}}$ is defined by (6.8) and where

$$D_{\alpha,b} = \max_{1\le\nu\le\alpha}\left(\sum_{\tau=\nu}^{\alpha}\left(\left(1+\frac{1}{b}+\frac{1}{b(b+1)}\right)^{(\tau-2)_+}(2\sin(\pi/b))^{-\tau}\right)^2 b^{-(\tau-\nu)} + 2\left(1+\frac{1}{b}+\frac{1}{b(b+1)}\right)^{2\alpha-2}(2\sin(\pi/b))^{-2\alpha}b^{-2(\alpha-\nu)}\right).$$

Baldeaux and Dick (2009) investigated the value of the constant $D_{\alpha,b}$ and showed that for $b = 2$ and 3 and $2 \le \alpha \le 8$ the constant $D_{\alpha,b} < 1$.

Theorem 6.11 holds for any digital net, that is, any point set obtained via the digital construction scheme as described in Section 2.6, including the higher-order digital nets to be introduced in the next subsection.

6.7. Higher-order digital nets

The aim is now to find digital nets, that is, generating matrices $C_1, \dots, C_s \in \mathbb{Z}_b^{w\times m}$ such that

$$\sum_{\boldsymbol{k}\in\mathcal{D}\setminus\{\mathbf{0}\}} b^{-\mu_\alpha(\boldsymbol{k})} = O(n^{-\alpha}(\log n)^{\alpha s}),$$

where the number of cubature points is $n = b^m$. Note that we now use generating matrices with w rows, and we usually choose $w \approx \alpha m$ as explained below.

Roughly speaking, the sum $\sum_{\boldsymbol{k}\in\mathcal{D}\setminus\{\mathbf{0}\}} b^{-\mu_\alpha(\boldsymbol{k})}$ is dominated by its largest term. To find this largest term, define

$$\mu^*_\alpha(C_1, \dots, C_s) = \min_{\boldsymbol{k}\in\mathcal{D}\setminus\{\mathbf{0}\}} \mu_\alpha(\boldsymbol{k}).$$

The dependence on the generating matrices $C_1, \dots, C_s$ on the right-hand side of the above equation is via the dual net $\mathcal{D} = \mathcal{D}(C_1, \dots, C_s)$. The largest term in $\sum_{\boldsymbol{k}\in\mathcal{D}\setminus\{\mathbf{0}\}} b^{-\mu_\alpha(\boldsymbol{k})}$ is then $b^{-\mu^*_\alpha(C_1,\dots,C_s)}$.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

In order to achieve a convergence of almost $n^{-\alpha} = b^{-\alpha m}$ we must have that the largest term in $\sum_{\boldsymbol{k}\in\mathcal{D}\setminus\{\boldsymbol{0}\}} b^{-\mu_\alpha(\boldsymbol{k})}$ is also of this order, that is, we must have $\mu^*_\alpha(C_1,\dots,C_s) \approx \alpha m$ (or say $\mu^*_\alpha(C_1,\dots,C_s) > \alpha m - t$ for some constant t independent of m). That this condition is also sufficient is quite technical and was shown in Dick (2008, Lemma 5.2).

We can use some analogy to find matrices $C_1,\dots,C_s \in \mathbb{Z}_b^{w\times m}$ which achieve $\mu^*_\alpha(C_1,\dots,C_s) \approx \alpha m$. Let $C_j = (\vec{c}_{j,1}^{\top},\dots,\vec{c}_{j,w}^{\top})^\top$, that is, $\vec{c}_{j,\ell} \in \mathbb{Z}_b^m$ is the ℓth row of C_j. Then the matrices $C_1,\dots,C_s$ generate a classical digital (t,m,s)-net if, for all $i_1,\dots,i_s \geq 0$ with $i_1+\cdots+i_s \leq m-t$, the vectors

$$\vec{c}_{1,1},\dots,\vec{c}_{1,i_1},\dots,\vec{c}_{s,1},\dots,\vec{c}_{s,i_s}$$

are linearly independent over $\mathbb{Z}_b$.

Now assume $C_1,\dots,C_s$ generate a classical digital (t,m,s)-net and that we are given a $\boldsymbol{k} \in \mathbb{N}_0^s \setminus \{\boldsymbol{0}\}$ with $\mu_1(\boldsymbol{k}) \leq m-t$. Let $i_j = \mu_1(k_j)$ for $j=1,\dots,s$, then $C_1^\top\vec{k}_1+\cdots+C_s^\top\vec{k}_s$ is a linear combination of the vectors $\vec{c}_{1,1},\dots,\vec{c}_{1,i_1},\dots,\vec{c}_{s,1},\dots,\vec{c}_{s,i_s}$. As $\boldsymbol{k}\neq\boldsymbol{0}$ and $i_1+\cdots+i_s \leq m-t$, which implies that $\vec{c}_{1,1},\dots,\vec{c}_{1,i_1},\dots,\vec{c}_{s,1},\dots,\vec{c}_{s,i_s}$ are linearly independent, it follows that $C_1^\top\vec{k}_1+\cdots+C_s^\top\vec{k}_s \neq \boldsymbol{0} \in \mathbb{Z}_b^m$. Thus $\boldsymbol{k}\notin\mathcal{D}$. This shows that if $C_1,\dots,C_s$ generate a classical digital (t,m,s)-net and $\boldsymbol{k}\in\mathcal{D}\setminus\{\boldsymbol{0}\}$, then $\mu_1(\boldsymbol{k}) > m-t$. This is precisely the type of result described above which we also want to have for $\alpha>1$.

We now want to generalize this linear independence condition to $\alpha>1$, that is, we want to have that if $\boldsymbol{k}\in\mathbb{N}_0^s\setminus\{\boldsymbol{0}\}$ with $\mu_\alpha(\boldsymbol{k}) \leq \alpha m - t$, then the generating matrices should have linearly independent rows such that $C_1^\top\vec{k}_1+\cdots+C_s^\top\vec{k}_s \neq \boldsymbol{0}\in\mathbb{Z}_b^m$. Let $\boldsymbol{k}=(k_1,\dots,k_s)$, where $k_j = \kappa_{j,1}b^{a_{j,1}-1}+\cdots+\kappa_{j,\nu_j}b^{a_{j,\nu_j}-1}$, with $a_{j,1}>\cdots>a_{j,\nu_j}>0$ and $0<\kappa_{j,1},\dots,\kappa_{j,\nu_j}<b$. First note that if $w<\alpha m-t$, then $\boldsymbol{k}=(b^w,0,\dots,0)\in\mathcal{D}$, but $\mu_\alpha(\boldsymbol{k}) = w+1 \leq \alpha m - t$. In order to avoid this problem we may choose $w=\alpha m$. Hence we may now assume that $a_{j,1}\leq w=\alpha m$ for $j=1,\dots,s$, as otherwise $\mu_\alpha(\boldsymbol{k}) > \alpha m$ already and no independence condition on the generating matrices is required in this case.

Now $C_1^\top\vec{k}_1+\cdots+C_s^\top\vec{k}_s$ is a linear combination of the rows

$$\vec{c}_{1,a_{1,1}},\dots,\vec{c}_{1,a_{1,\nu_1}},\dots,\vec{c}_{s,a_{s,1}},\dots,\vec{c}_{s,a_{s,\nu_s}}.$$

Thus, if these rows are linearly independent, then

$$C_1^\top\vec{k}_1+\cdots+C_s^\top\vec{k}_s \neq \boldsymbol{0}\in\mathbb{Z}_b^m,$$

and therefore $\boldsymbol{k}\notin\mathcal{D}$.

Therefore, if $C_1,\dots,C_s\in\mathbb{Z}_b^{\alpha m\times m}$ are such that for all choices of $a_{j,1}>\cdots>a_{j,\nu_j}>0$ for $j=1,\dots,s$, with

$$a_{1,1}+\cdots+a_{1,\min(\alpha,\nu_1)}+\cdots+a_{s,1}+\cdots+a_{s,\min(\alpha,\nu_s)} \leq \alpha m - t,$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

the rows

$$\vec{c}_{1,a_{1,1}},\dots,\vec{c}_{1,a_{1,\nu_1}},\dots,\vec{c}_{s,a_{s,1}},\dots,\vec{c}_{s,a_{s,\nu_s}}$$

are linearly independent, then $\boldsymbol{k} \in \mathcal{D} \setminus \{\mathbf{0}\}$ implies that $\mu_\alpha(\boldsymbol{k}) > \alpha m - t$. (Note that we also include the case where some $\nu_j = 0$, in which case we just set $a_{j,1} + \cdots + a_{j,\min(\alpha,\nu_j)} = 0$.)

We can now formally define such digital nets for which the generating matrices satisfy such a property. The following definition is a special case of Dick (2008, Definition 4.3).

Definition 6.12 (digital (t, α, m, s)-nets). Let $m, \alpha \geq 1$, and $0 \leq t \leq \alpha m$ be natural numbers. Let $\mathbb{Z}_b$ be the finite field of prime order b and let $C_1,\dots,C_s \in \mathbb{Z}_b^{\alpha m \times m}$ with $C_j = (\vec{c}_{j,1}^{\top},\dots,\vec{c}_{j,\alpha m}^{\top})^{\top}$. If for all $0 < a_{j,\nu_j} < \cdots < a_{j,1}$, where $0 \leq \nu_j$ for all $j = 1,\dots,s$, with

$$\sum_{j=1}^{s} \sum_{\ell=1}^{\min(\nu_j,\alpha)} a_{j,\ell} \leq \alpha m - t,$$

the vectors

$$\vec{c}_{1,a_{1,\nu_1}},\dots,\vec{c}_{1,a_{1,1}},\dots,\vec{c}_{s,a_{s,\nu_s}},\dots,\vec{c}_{s,a_{s,1}}$$

are linearly independent over $\mathbb{Z}_b$, then the digital net with generating matrices $C_1,\dots,C_s$ is called a *digital (t, α, m, s)-net over $\mathbb{Z}_b$*.

The need for a more general definition in Dick (2008) arises as we assume therein that the smoothness α of the integrand is not known, so one cannot choose $w = \alpha m$ in this case.

We have the following result.

Theorem 6.13 (worst-case error bound for digital (t, α, m, s)-nets). Let $\alpha \geq 2$ be an integer, and let $b \geq 2$ be a prime number. Then the worst-case error of multivariate integration in the weighted unanchored Sobolev space $H(K_{s,\boldsymbol{\gamma},\alpha})$ of smoothness $\alpha > 1$, using a QMC rule with a digital (t, α, m, s)-net P over $\mathbb{Z}_b$ as cubature points, is bounded by

$$e(P, H(K_{s,\boldsymbol{\gamma},\alpha})) \leq b^{-(\alpha m - t)} \left(\sum_{\emptyset \neq \mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}} (D''_{|\mathfrak{u}|,\alpha,b})^2 (\alpha m - t + \alpha + 2)^{2|\mathfrak{u}|\alpha} \right)^{\frac{1}{2}},$$

where $D''_{|\mathfrak{u}|,\alpha,b} = (D_{\alpha,b})^{\frac{|\mathfrak{u}|}{2}} b^{|\mathfrak{u}|\alpha} (b^{-1} + (1 - b^{1/\alpha - 1})^{-|\mathfrak{u}|\alpha})$.

We conclude from this theorem that a digital (t, α, m, s)-net used as cubature points in a QMC rule will yield a convergence of the integration error of order $n^{-\alpha}(\log n)^{\alpha s}$ for integrands with $\|f\|_{s,\boldsymbol{\gamma},\alpha} < \infty$. The remaining question now is: Do digital (t, α, m, s)-nets exist for all given $\alpha, s \geq 1$ and some fixed t (which may depend on α and s but not on m) for all $m \in \mathbb{N}$? An affirmative answer to this question will be given in the next subsection.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

6.8. Construction of higher-order digital nets

We now present explicit constructions of digital (t, α, m, s)-nets. Central to this method is the *digit interlacing function*.

Definition 6.14 (digit interlacing function). The digit interlacing function with interlacing factor $\alpha \in \mathbb{N}$ is given by

$$\begin{aligned} \mathscr{D}_\alpha : [0,1)^\alpha &\to [0,1) \\ (x_1, \ldots, x_\alpha) &\mapsto \sum_{a=1}^{\infty} \sum_{r=1}^{\alpha} \xi_{r,a} b^{-r-(a-1)\alpha}, \end{aligned}$$

where $x_r = \xi_{r,1} b^{-1} + \xi_{r,2} b^{-2} + \cdots$ for $1 \le r \le \alpha$. We also define this function for vectors by setting

$$\begin{aligned} \mathscr{D}_\alpha : [0,1)^{\alpha s} &\to [0,1)^s \\ (x_1, \ldots, x_{\alpha s}) &\mapsto (\mathscr{D}_\alpha(x_1, \ldots, x_\alpha), \ldots, \mathscr{D}_\alpha(x_{(s-1)\alpha+1}, \ldots, x_{s\alpha})). \end{aligned}$$

Example 6.15 (construction of higher-order digital nets using point sets). Let $\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{b^m-1} \in [0,1)^{s\alpha}$ be a digital $(t', m, s\alpha)$-net in base b. Then the point set

$$\mathscr{D}_\alpha(\boldsymbol{t}_0), \mathscr{D}_\alpha(\boldsymbol{t}_1), \ldots, \mathscr{D}_\alpha(\boldsymbol{t}_{b^m-1})$$

is a digital (t, α, m, s)-net in base b, where t is given by Theorem 6.17.

Higher-order digital sequences can be constructed in an analogous way.

To analyse the properties of higher-order digital nets, it is more convenient to describe the above construction using the generating matrices rather than the point set.

Example 6.16 (construction of higher-order digital nets using generating matrices). Let $C_1, \ldots, C_{s\alpha}$ be the generating matrices of a digital $(t', m, s\alpha)$-net. Let $C_j = (\vec{c}_{j,1}^{\,\top}, \ldots, \vec{c}_{j,m}^{\,\top})^\top$ for $j = 1, \ldots, s\alpha$; that is, $\vec{c}_{j,\ell}$ are the row vectors of C_j. Now let the matrix $C_j^{(\alpha)}$ be made of the first rows of the matrices $C_{(j-1)\alpha+1}, \ldots, C_{j\alpha}$, then the second rows of $C_{(j-1)\alpha+1}, \ldots, C_{j\alpha}$, and so on. The matrix $C_j^{(\alpha)}$ is then an $\alpha m \times m$ matrix, that is,

$$C_j^{(\alpha)} = (\vec{c}_{j,1}^{\,(\alpha)}, \ldots, \vec{c}_{j,\alpha m}^{\,(\alpha)})^\top,$$

where

$$\vec{c}_{j,\ell}^{\,(\alpha)} = \vec{c}_{u,v},$$

with $\ell = (v-j)\alpha + u$, $1 \le v \le m$, and $(j-1)\alpha < u \le j\alpha$ for $\ell = 1, \ldots, \alpha m$ and $j = 1, \ldots, s$. Then the matrices $C_1^{(\alpha)}, \ldots, C_s^{(\alpha)}$ are the generating matrices of a digital (t, α, m, s)-net over $\mathbb{Z}_b$, where t is given by Theorem 6.17.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

As we will see later, the choice of the underlying $(t', m, s\alpha)$-net has a direct impact on the bound on the t-value of the digital (t, α, m, s)-net.

To give the idea why this construction works we may consider the case $s = 1$. Let $\alpha > 1$. To simplify the notation we drop the j (which denotes the coordinate) from the notation for a moment. Let $C^{(\alpha)}$ be constructed from a classical digital (t', m, α)-net with generating matrices $C_1, \ldots, C_\alpha$ as described above. Let $\alpha m \geq a_1 > a_2 > \cdots > a_\nu \geq 1$. Then we need to consider the row vectors $\vec{c}_{a_1}^{(\alpha)}, \ldots, \vec{c}_{a_\nu}^{(\alpha)}$. Now by the construction above, the vector $\vec{c}_{a_1}^{(\alpha)}$ may stem from any of the generating matrices $C_1, \ldots, C_\alpha$. Without loss of generality assume that $\vec{c}_{a_1}^{(\alpha)}$ stems from C_1, *i.e.*, it is the i_1th row of C_1, where $i_1 = \lceil a_1/\alpha \rceil$. Next consider $\vec{c}_{a_2}^{(\alpha)}$. This row vector may again stem from any of the matrices $C_1, \ldots, C_\alpha$. If $\vec{c}_{a_2}^{(\alpha)}$ also stems from C_1, then $\lceil a_2/\alpha \rceil < i_1$. If not, we may assume without loss of generality that it stems from C_2. Indeed, it will be the i_2th row of C_2, where $i_2 = \lceil a_2/\alpha \rceil$. We continue in this fashion and define numbers $i_3, i_4, \ldots, i_\ell$, where $1 \leq \ell \leq \alpha$. Further, we set $i_{\ell+1} = \cdots = i_\alpha = 0$. Then by the (t', m, α)-net property of $C_1, \ldots, C_\alpha$, it follows that $\vec{c}_{a_1}^{(\alpha)}, \ldots, \vec{c}_{a_\nu}^{(\alpha)}$ are linearly independent provided that $i_1 + \cdots + i_\alpha \leq m - t'$. Hence, if we choose t such that $a_1 + \cdots + a_{\min(\alpha,\nu)} \leq \alpha m - t$ implies that $i_1 + \cdots + i_\alpha \leq m - t'$ for all admissible choices of $a_1, \ldots, a_\nu$, then the digital $(t, \alpha, m, 1)$-net property of $C^{(\alpha)}$ follows.

Note that $i_1 = \lceil a_1/\alpha \rceil$ and $i_\ell \leq \lceil a_\ell/\alpha \rceil$ for $\ell = 2, \ldots, \alpha$. Thus

$$\begin{aligned} i_1 + \cdots + i_\alpha &\leq \lceil a_1/\alpha \rceil + \cdots + \lceil a_\alpha/\alpha \rceil \leq (a_1 + \cdots + a_\alpha + \alpha(\alpha - 1))/\alpha \\ &= \frac{a_1 + \cdots + a_\alpha}{\alpha} + \alpha - 1 \leq m - t/\alpha + \alpha - 1. \end{aligned}$$

Thus, if we choose t such that $m - t/\alpha + \alpha - 1 \leq m - t'$, then the result follows. Simple algebra then shows that

$$t = \alpha t' + \alpha(\alpha - 1)$$

will suffice.

A more general and improved result is given in the following.

Theorem 6.17. Let $\alpha \geq 1$ be a natural number and let $C_1, \ldots, C_{s\alpha}$ be the generating matrices of a digital $(t', m, s\alpha)$-net over the finite field $\mathbb{Z}_b$ of prime order b. Let $C_1^{(\alpha)}, \ldots, C_s^{(\alpha)}$ be defined as in Example 6.16. Then the matrices $C_1^{(\alpha)}, \ldots, C_s^{(\alpha)}$ are the generating matrices of a digital (t, α, m, s)-net over $\mathbb{Z}_b$ with

$$t = \alpha \min\left(m, t' + \left\lfloor \frac{s(\alpha - 1)}{2} \right\rfloor\right).$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

This shows that digital (t, α, m, s)-nets can be explicitly constructed for all $\alpha, m, s \geq 1$ with t bounded independently of m. Indeed, the dependence of t on α and s is also known from Dick and Baldeaux (2009): namely, there are constants $c, C > 0$ independent of α and s such that $c\alpha^2 s \leq t \leq C\alpha^2 s$.

Using Theorem 6.17 we therefore obtain explicit constructions of higher-order quasi-Monte Carlo rules which satisfy Theorem 6.13.

6.9. Geometric properties of higher-order digital nets

Figures 6.1–6.4 illustrate the properties of a digital $(2, 2, 4, 2)$-net in base 2. Figure 6.2 shows a partition of the square for which each union of the shaded rectangles contains exactly two points. Figures 6.3 and 6.4 show that also other partitions of the unit square are possible where each union of shaded rectangles contains the 'fair' number of points. Many other partitions of the square are possible where the point set always contains the fair number of points in each union of rectangles (see Dick and Baldeaux 2009), but there are too many to show all of them here. Even in the simple case considered here there are 12 partitions possible, for each of which the point set is fair: this is quite remarkable since the point set itself has only 16 points (we exclude all those partitions for which the fairness would follow already from some other partition, otherwise there would be 34 of them). In the classical case we have 4 such partitions, all of which are shown in Figure 6.1. (The partitions from the classical case are included in the generalized case; so out of the 12 partitions 4 are shown in Figure 6.1, one is shown in Figure 6.2, one is shown in Figure 6.4 and one is indicated in Figure 6.3.)

The subsets of $[0, 1)^s$ which form a partition, each having a fair number of points, are of the form

$$J(\vec{a}_{\vec{\nu}}, \vec{d}_{\vec{\nu}}) = \prod_{j=1}^{s} \bigcup_{\substack{d_{j,\ell}=0 \\ \ell \in \{1,\ldots,\alpha m\} \setminus \{a_{j,1},\ldots,a_{j,\nu_j}\}}}^{b-1} \left[\frac{d_{j,1}}{b} + \cdots + \frac{d_{j,n}}{b^{\alpha m}}, \frac{d_{j,1}}{b} + \cdots + \frac{d_{j,n}}{b^{\alpha m}} + \frac{1}{b^{\alpha m}} \right),$$

where $b \geq 2$ is the base and where $\sum_{j=1}^{s} \sum_{\ell=1}^{\nu_j} a_{j,\ell} \leq \alpha m - t$. For $j = 1, \ldots, s$ we again assume $1 \leq a_{j,\nu_j} < \cdots < a_{j,1} \leq \alpha m$ for $\nu_j > 0$ and $\{a_{j,1}, \ldots, a_{j,\nu_j}\} = \emptyset$ for $\nu_j = 0$. Further, we also use the following notation:

$$\vec{\nu} = (\nu_1, \ldots, \nu_s), \quad |\vec{\nu}|_1 = \sum_{j=1}^{s} \nu_j,$$

$$\vec{a}_{\vec{\nu}} = (a_{1,1}, \ldots, a_{1,\nu_1}, \ldots, a_{s,1}, \ldots, a_{s,\nu_s}),$$

$$\vec{d}_{\vec{\nu}} \in \{0, \ldots, b-1\}^{|\vec{\nu}|_1},$$

$$\vec{d}_{\vec{\nu}} = (d_{1,i_{1,1}}, \ldots, d_{1,i_{1,\nu_1}}, \ldots, d_{s,i_{s,1}}, \ldots, d_{s,i_{s,\nu_s}}),$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

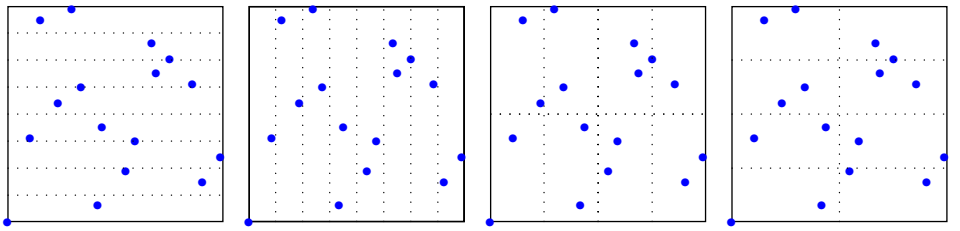


Figure 6.1. Higher-order Sobol′ points: these 16 points are from a Sobol′ point set in dimension four with interlacing factor two (which yields a two-dimensional point set).

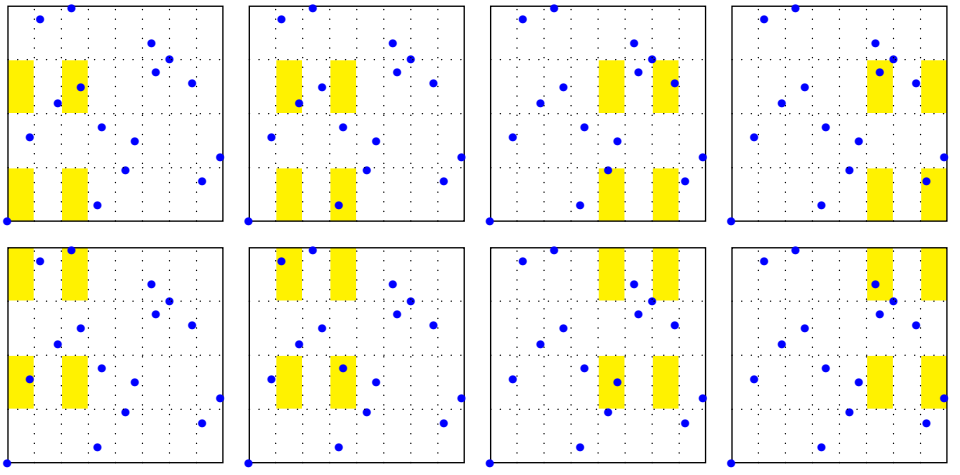


Figure 6.2. Higher-order Sobol′ points: the shaded area in each square contains exactly two points.

where the components $a_{j,\ell}$ and $d_{j,\ell}$, $\ell = 1, \ldots, \nu_j$, do not appear in the vectors $\vec{a}_{\vec{\nu}}$ and $\vec{d}_{\vec{\nu}}$ for $\nu_j = 0$.

Figures 6.1–6.4 give only a few examples of unions of intervals for which each subset of the partition contains the right number of points. As the subsets $J(\vec{a}_{\vec{\nu}}, \vec{d}_{\vec{\nu}})$ for fixed $\vec{\nu}$ and $\vec{a}_{\vec{\nu}}$ (with $\vec{d}_{\vec{\nu}}$ running through all possibilities) form a partition of $[0,1)^s$, it is clear that the right number of points in $J(\vec{a}_{\vec{\nu}}, \vec{d}_{\vec{\nu}})$ has to be $b^m \mathrm{Vol}(J(\vec{a}_{\vec{\nu}}, \vec{d}_{\vec{\nu}}))$. For example, the digital net in Figure 6.2 has 16 points and the partition consists of 8 different subsets $J(\vec{a}_{\vec{\nu}}, \vec{d}_{\vec{\nu}})$, hence each $J(\vec{a}_{\vec{\nu}}, \vec{d}_{\vec{\nu}})$ contains exactly $16/8 = 2$ points. (In general, the volume of $J(\vec{a}_{\vec{\nu}}, \vec{d}_{\vec{\nu}})$ is given by $b^{-|\vec{\nu}|_1}$: see Dick and Baldeaux 2009.)

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

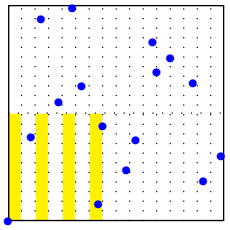


Figure 6.3. Higher-order Sobol′ points: the shaded area contains exactly two points.

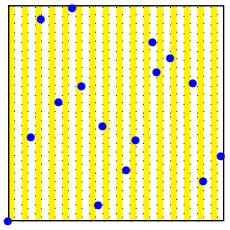


Figure 6.4. Higher-order Sobol′ points: the shaded area contains exactly half of the points.

6.10. Scrambled digital nets and sequences

We now discuss properties of Owen's scrambling algorithm, which was introduced in Section 2.10. First we show that a point $\boldsymbol{x}$ scrambled using Owen's algorithm is uniformly distributed in $[0,1)^s$.

Proposition 6.18. Let $\boldsymbol{x} \in [0,1)^s$ and let $\mathbf{\Pi}$ be a uniformly and i.i.d. set of permutations. Then $\mathbf{\Pi}(\boldsymbol{x})$ is uniformly distributed in $[0,1)^s$, that is, for any Lebesgue-measurable set $G \subseteq [0,1)^s$, the probability that $\mathbf{\Pi}(\boldsymbol{x}) \in G$, denoted by $\mathbb{P}[\mathbf{\Pi}(\boldsymbol{x}) \in G]$, satisfies $\mathbb{P}[\mathbf{\Pi}(\boldsymbol{x}) \in G] = \lambda_s(G)$, where λ_s denotes the s-dimensional Lebesgue measure.

Proof. We follow Owen (1995, Proof of Proposition 2) in our exposition. We use the notation from above and set $\boldsymbol{y} := \mathbf{\Pi}(\boldsymbol{x})$. Consider the case $s = 1$ first and let

$$E = \left[\frac{a}{b^\ell}, \frac{a+1}{b^\ell}\right)$$

be an elementary interval where $\ell \geq 0$ and $0 \leq a < b^\ell$. A technical problem which can arise in the proof below is when y is of the form

$$y_1 = y_{1,1}b^{-1} + \cdots + y_{1,\ell}b^{-\ell} + (b-1)b^{-\ell-1} + (b-1)b^{-\ell-1} + \cdots,$$

since then we have $y_1 = y_{1,1}b^{-1} + \cdots + (y_{1,\ell}+1)b^{-\ell}$. We show that this only happens with probability 0.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

The probability that there are $u \geq j_0 \geq 1$ such that $y_{1,j_0} = y_{1,j_0+1} = \cdots = y_{1,u} = b-1$ is given by $((b-1)!)^{-(u-j_0)}$. Hence the probability that $y_{1,j_0} = y_{1,j_0+1} = \cdots = b-1$, that is, all digits of y_1 are $b-1$ from some index j_0 onwards, is 0.

Let $ab^{-\ell} = a_1 b^{-1} + a_2 b^{-2} + \cdots + a_\ell b^{-\ell}$. Then $y_1 \in E$ if and only if $y_1 = a_1, y_2 = a_2, \ldots, y_\ell = a_\ell$. Using (2.8), this is equivalent to

$$\pi_{1,x_{1,1},\ldots,x_{1,k-1}}(x_{1,k}) = a_k, \quad \text{for } 1 \leq k \leq \ell. \tag{6.16}$$

For each $1 \leq k \leq \ell$, the probability that (6.16) holds is b^{-1}. Hence the probability that $y_1 \in E$ is $b^{-\ell}$. The result therefore holds for all elementary intervals of $[0,1)$.

We now extend the result to the general case. First notice that the result also holds for all subintervals $[ub^{-\ell}, vb^{-\ell})$, where $\ell \geq 0$ and $0 \leq u < v \leq b^{-\ell}$. The endpoints of these intervals are dense in $[0,1)$. A corollary of Chung (1974, p. 28) extends the result $\mathbb{P}[y_1 \in B] = \lambda_1(B)$ to all Borel-measurable subsets $B \subseteq [0,1)$. The equality $\mathbb{P}[y_1 \in B] = \lambda_1(B)$ extends to Lebesgue-measurable sets B since subsets of sets of measure zero have probability zero of containing y_1.

Consider now $s > 1$. Let $B_1, \ldots, B_s$ be measurable subsets of $[0,1)$. Because the components $y_1, \ldots, y_s$ of $\boldsymbol{y}$ are independent, it follows that

$$\mathbb{P}[y_i \in B_i, 1 \leq i \leq s] = \prod_{i=1}^{s} \lambda_1(B_i). \tag{6.17}$$

Finally, λ_s is the unique measure on $[0,1)^s$ which satisfies (6.17). □

Consider a (t,m,s)-net in base b consisting of points $\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{b^m-1}$, where $\boldsymbol{t}_i = (t_{i,1}, \ldots, t_{i,s})$ and $t_{i,j} = t_{i,j,1}b^{-1} + t_{i,j,2}b^{-2} + \cdots$. We shall denote the scrambled points by $\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{b^m-1}$, where $\boldsymbol{y}_i = (y_{i,1}, \ldots, y_{i,s})$ and $y_{i,j} = y_{i,j,1}b^{-1} + y_{i,j,2}b^{-2} + \cdots$. Specifically, the scrambled points are given by $y_{i,j,k} = \pi_{j,t_{i,j,1},\ldots,t_{i,j,k-1}}(t_{i,j,k})$, for $0 \leq i < b^m$, $1 \leq j \leq s$, and $k \geq 1$.

Similarly, if $(\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots)$ is a (t,s)-sequence, then the scrambled sequence will be denoted by $(\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots)$, where, using the same notation as above, again $y_{i,j,k} = \pi_{j,t_{i,j,1},\ldots,t_{i,j,k-1}}(t_{i,j,k})$, for all $i \geq 0$, $1 \leq j \leq s$, and $k \geq 1$.

We consider now the expected value of $\frac{1}{n}\sum_{i=0}^{n-1} f(\boldsymbol{y}_i)$. For any measurable function f we have

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=0}^{n-1} f(\boldsymbol{y}_i)\right] = \frac{1}{n}\sum_{i=0}^{n-1} \mathbb{E}[f(\boldsymbol{y}_i)] = \int_{[0,1]^s} f(\boldsymbol{y})\mathrm{d}\boldsymbol{y},$$

since each point $\boldsymbol{y}_i$ is uniformly distributed in $[0,1)^s$ and hence

$$\mathbb{E}[f(\boldsymbol{y}_i)] = \int_{[0,1]^s} f(\boldsymbol{y})\mathrm{d}\boldsymbol{y} \quad \text{for } 0 \leq i < n.$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

In other words, this means that a scrambled point set (note that the above applies even if the underlying point set $\boldsymbol{t}_0, \dots, \boldsymbol{t}_{n-1}$ is not a digital net) used in a QMC rule yields an unbiased estimator.

The second important property of the randomization algorithm we require is that the (t, m, s)-net structure of the points $\boldsymbol{t}_0, \dots, \boldsymbol{t}_{b^m-1}$ is retained after applying the scrambling algorithm. For some technical reason this does not quite hold, but it holds with probability one, which is still sufficient. The following proposition was first shown by Owen (1995).

Proposition 6.19. If $\boldsymbol{t}_0, \dots, \boldsymbol{t}_{b^m-1}$ form a (t, m, s)-net in base b, then $\boldsymbol{y}_0, \dots, \boldsymbol{y}_{b^m-1}$ is a (t, m, s)-net in base b with probability one. If $\boldsymbol{t}_0, \boldsymbol{t}_1, \dots$ are obtained from a (t, s)-sequence, then the scrambled points $\boldsymbol{y}_0, \boldsymbol{y}_1, \dots$ form a (t, s)-sequence with probability one.

Proof. The probability that, for some $0 \le i < b^m$, $1 \le j \le s$, and $\ell \in \mathbb{N}$, all $y_{i,j,k} = b - 1$ for all $k \ge \ell$ is 0. Equivalently, with probability one, infinitely many digits in the b-adic expansion of $y_{i,j}$ are different from $b-1$. Therefore, the probability that $y_{i,j}$ has infinitely many digits in the b-adic expansion of $y_{i,j}$ are different from $b-1$ for all $0 \le i < b^m$ and $1 \le j \le s$ is 0, since the union of a finite number of zero probability events has probability zero. Hence this holds for each component of each point of a (t, m, s)-net.

For a (t, s)-sequence the same applies, since a countable union of probability zero events itself has probability zero. Hence this also holds for each component of each point of a (t, s)-sequence.

Therefore we may, in the following, assume that infinitely many digits in the b-adic expansion of $y_{i,j}$ differ from $b-1$ for all $i \in \mathbb{N}_0$ and $1 \le j \le s$.

Assume we are given an elementary interval

$$J = \prod_{j=1}^{s} [a_j b^{-d_j}, (a_j + 1) b^{-d_j}),$$

where $0 \le a_j < b^{d_j}$, $d_j \in \mathbb{N}_0$, and $d_1 + \cdots + d_s \le m - t$. Let $a_j b^{-d_j} = a_{j,1} b^{-1} + a_{j,2} b^{-2} + \cdots + a_{j,d_j} b^{-d_j}$.

Then $\boldsymbol{y}_i \in J$ if and only if $y_{i,j,k} = a_{j,k}$ for all $1 \le k \le d_j$ and all $1 \le j \le s$. Further, $y_{i,j,k} = a_{j,k}$ if and only if $x_{i,j,k} = \pi^{-1}_{j,t_{i,j,1},\dots,t_{i,j,k-1}}(a_{j,k})$. Let $a'_{j,k} = \pi^{-1}_{j,t_{i,j,1},\dots,t_{i,j,k-1}}(a_{j,k})$. Then $\boldsymbol{y}_i \in J$ if and only if

$$\boldsymbol{t}_i \in J' = \prod_{j=1}^{s} [a'_j b^{-d_j}, (a'_j + 1) b^{-d_j}),$$

where $a'_j b^{-d_j} = a'_{j,1} b^{-1} + \cdots + a'_{j,d_j} b^{-d_j}$. As the points $\boldsymbol{t}_0, \dots, \boldsymbol{t}_{b^m-1}$ form a (t, m, s)-net, it follows that there are exactly b^{m-t} points of this net in J' and hence there are exactly b^{m-t} points of $\boldsymbol{y}_0, \dots, \boldsymbol{y}_{b^m-1}$ in J. Thus $\boldsymbol{y}_0, \dots, \boldsymbol{y}_{b^m-1}$ form a (t, m, s)-net with probability one.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

For a (t,s)-sequence $\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots$ for all $k \in \mathbb{N}_0$ and $m \geq t$ the point set consisting of $\boldsymbol{t}_{kb^m}, \ldots, \boldsymbol{t}_{kb^m+b^m-1}$ forms a (t,m,s)-net which is again a (t,m,s)-net after scrambling with probability one. Since the union of countably many zero probability events has probability zero, the result for (t,s)-sequences follows as well. □

6.11. Error analysis of scrambled digital nets

In this subsection we study upper bounds on the variance of scrambled digital nets. To explain the result for scrambled digital nets, we introduce the so-called *nested ANOVA decomposition*.

Theorem 6.20 (nested ANOVA decomposition). Consider a function $f \in L_2([0,1])$ with Walsh series expansion

$$f(x) \sim \sum_{k=0}^{\infty} \widehat{f}(k)\mathrm{wal}_k(x), \tag{6.18}$$

where

$$\widehat{f}(k) = \int_0^1 f(x)\overline{\mathrm{wal}_k(x)}\,\mathrm{d}x.$$

Define $\beta_0 := \int_0^1 f(y)\,\mathrm{d}y$, and for $\ell \geq 1$,

$$\beta_\ell(x) := \sum_{k=b^{\ell-1}}^{b^\ell-1} \widehat{f}(k)\mathrm{wal}_k(x) \quad \text{and} \quad \sigma_\ell^2(f) := \mathrm{Var}[\beta_\ell].$$

Then we have the *nested ANOVA decomposition* of f

$$\mathrm{Var}[f] = \sum_{\ell=1}^{\infty} \sigma_\ell^2(f).$$

Notice that we do not necessarily have equality in (6.18), since the function f is only assumed to be in $L_2([0,1])$ (and hence may for example be changed arbitrarily on a set of measure zero without changing $\widehat{f}(k)$ for any $k \geq 0$).

Proof. Consider the b^ℓ-term approximation of f given by

$$\sum_{k=0}^{b^\ell-1} \widehat{f}(k)\mathrm{wal}_k(x) = \sum_{k=0}^{b^\ell-1} \int_0^1 f(y)\mathrm{wal}_k(x \ominus y)\,\mathrm{d}y = \int_0^1 f(y)D_\ell(x \ominus y)\,\mathrm{d}y,$$

where D_ℓ is the Walsh–Dirichlet kernel given by

$$D_\ell(z) = \sum_{k=0}^{b^\ell-1} \mathrm{wal}_k(z) = \begin{cases} b^\ell & \text{if } z \in [0, b^{-\ell}), \\ 0 & \text{otherwise.} \end{cases}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Hence we have

$$\sum_{k=0}^{b^\ell-1} \widehat{f}(k)\mathrm{wal}_k(x) = b^\ell \int_{\lfloor yb^\ell\rfloor=\lfloor xb^\ell\rfloor} f(y)\,\mathrm{d}y,$$

where the integration is over all y such that $\lfloor yb^\ell\rfloor = \lfloor xb^\ell\rfloor$, that is, the first ℓ digits of x and y coincide. Therefore, for $\ell \geq 1$ we have

$$\beta_\ell(x) = b^\ell \int_{\lfloor yb^\ell\rfloor=\lfloor xb^\ell\rfloor} f(y)\,\mathrm{d}y - b^{\ell-1} \int_{\lfloor yb^{\ell-1}\rfloor=\lfloor xb^{\ell-1}\rfloor} f(y)\,\mathrm{d}y.$$

(Owen 1997*b* defined this function using Haar wavelets.) Notice that β_ℓ is constant on intervals of the form $[ub^{-\ell}, (u+1)b^{-\ell})$, so $\beta_\ell(x) = \beta_\ell(\lfloor b^\ell x\rfloor b^{-\ell})$.

Then, because of the orthogonality of the Walsh functions we obtain

$$\sigma_\ell^2(f) = \int_0^1 |\beta_\ell(x)|^2\mathrm{d}x = \sum_{k=b^{\ell-1}}^{b^\ell-1} |\widehat{f}(k)|^2$$

and also

$$\int_0^1 \beta_\ell(x)\overline{\beta_{\ell'}(x)}\mathrm{d}x = 0, \quad \text{for } \ell \neq \ell'.$$

Since $f \in L_2([0,1])$ and the Walsh function system is complete, we can use Plancherel's identity to obtain

$$\mathrm{Var}[f] = \int_0^1 |f(y) - \mathbb{E}(f)|^2\mathrm{d}y = \sum_{k=1}^{\infty} |\widehat{f}(k)|^2 = \sum_{\ell=1}^{\infty} \sigma_\ell^2(f).$$

Therefore we have obtained a decomposition of the variance of f in terms of the variances of β_ℓ. □

Notice that $\sum_{k=1}^{\infty} |\widehat{f}(k)|^2 = \mathrm{Var}\big[\sum_{k=0}^{\infty} \widehat{f}(k)\mathrm{wal}_k\big]$. Hence, as a by-product, we obtain that for any $f \in L_2([0,1])$, the variance of f and the variance of its Walsh series coincide, that is,

$$\mathrm{Var}[f] = \mathrm{Var}\left[\sum_{k=0}^{\infty} \widehat{f}(k)\mathrm{wal}_k\right].$$

For functions $f : [0,1]^s \to \mathbb{R}$ the nested ANOVA decomposition can be applied component-wise. Let $f \in L_2([0,1]^s)$ have the following Walsh series expansion:

$$f(\boldsymbol{x}) \sim \sum_{\boldsymbol{k}\in\mathbb{N}_0^s} \widehat{f}(\boldsymbol{k})\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}) =: S(\boldsymbol{x}, f).$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Let $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_s) \in \mathbb{N}_0^s$ and $L_{\boldsymbol{\ell}} = \{\boldsymbol{k} = (k_1, \ldots, k_s) \in \mathbb{N}_0^s : \lfloor b^{\ell_j - 1} \rfloor \le k_j < b^{\ell_j} \text{ for } 1 \le j \le s\}$. Then let

$$\beta_{\boldsymbol{\ell}}(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in L_{\boldsymbol{\ell}}} \widehat{f}(\boldsymbol{k}) \mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x})$$

and

$$\sigma_{\boldsymbol{\ell}}^2(f) := \mathrm{Var}[\beta_{\boldsymbol{\ell}}] = \int_{[0,1]^s} |\beta_{\boldsymbol{\ell}}(\boldsymbol{x})|^2 \mathrm{d}\boldsymbol{x} = \sum_{\boldsymbol{k} \in L_{\boldsymbol{\ell}}} |\widehat{f}(\boldsymbol{k})|^2.$$

For $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_s) \in \mathbb{N}_0^s \setminus \{\boldsymbol{0}\}$ let

$$G_{\boldsymbol{\ell}} = \frac{1}{N^2} \sum_{i,i'=0}^{n-1} \prod_{j=1}^{s} \left(\frac{b}{b-1} 1_{\lfloor b^{\ell_j} t_{i,j} \rfloor = \lfloor b^{\ell_j} t_{i',j} \rfloor} - \frac{1}{b-1} 1_{\lfloor b^{\ell_j - 1} t_{i,j} \rfloor = \lfloor b^{\ell_j - 1} t_{i',j} \rfloor} \right),$$

where $\boldsymbol{t}_i = (t_{i,1}, \ldots, t_{i,s})$ for $0 \le i \le n-1$ and where $1_{A=B}$ is 1 if $A = B$ and 0 otherwise.

If $\boldsymbol{k} \in L_{\boldsymbol{\ell}}$, then

$$\frac{1}{n^2} \sum_{i,i'=0}^{n-1} \mathbb{E}\big[\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{y}_i \ominus \boldsymbol{y}_{i'})\big] = G_{\boldsymbol{\ell}},$$

since the coordinates are randomized independently from each other. Owen (1997*b*) called the numbers $\Gamma_{\boldsymbol{\ell}} := nG_{\boldsymbol{\ell}}$ *gain coefficients*, since, as we see below, they determine how much one gains compared to the classical Monte Carlo algorithm.

We now estimate the variance of the estimator

$$\widehat{Q}_{n,s}(f) = \frac{1}{n} \sum_{i=0}^{n-1} f(\boldsymbol{y}_i) \tag{6.19}$$

for integrands $f \in L_2([0,1]^s)$ when the points $\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{n-1}$ are obtained by applying Owen's scrambling to a (digital) (t, m, s)-net over $\mathbb{Z}_b$. (To emphasize that (6.19) is a random variable, we use the notation $\widehat{Q}_{n,s}$ rather than $Q_{n,s}$.)

For the following results see Owen (1997*a*, 1997*b*, 1998).

Theorem 6.21 (variance of the integral estimator using scrambled nets). Let $f \in L_2([0,1]^s)$ and $\widehat{Q}_{n,s}(f)$ be given by (6.19). Let the point set $\{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{n-1}\} \subseteq [0,1)^s$ be obtained by applying Owen's scrambling algorithm to the point set $\{\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{n-1}\} \subseteq [0,1)^s$. Then the variance of the estimator $\widehat{Q}_{n,s}(f)$ is given by

$$\mathrm{Var}[\widehat{Q}_{n,s}(f)] = \frac{1}{n} \sum_{\boldsymbol{\ell} \in \mathbb{N}_0^s \setminus \{\boldsymbol{0}\}} \Gamma_{\boldsymbol{\ell}} \sigma_{\boldsymbol{\ell}}^2(f).$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Theorem 6.22 (a bound on the variance in terms of gain coefficients). Let $f \in L_2([0,1]^s)$ and $\widehat{Q}_{n,s}(f)$ be given by (6.19). Let the points $\{\boldsymbol{t}_0, \dots, \boldsymbol{t}_{b^m-1}\}$ be a digital (t,m,s)-net over $\mathbb{Z}_b$. Then

$$\mathrm{Var}[\widehat{Q}_{n,s}(f)] \le b^{-m+t}\left(\frac{b+1}{b-1}\right)^s \sum_{\substack{\boldsymbol{\ell}\in\mathbb{N}_0^s \\ |\boldsymbol{\ell}|_1 > m-t}} \sigma_{\boldsymbol{\ell}}^2(f).$$

For MC one obtains a variance $\mathrm{Var}[\widehat{Q}_{n,s}(f)] = \frac{1}{n}\sum_{\boldsymbol{\ell}\in\mathbb{N}_0^s\setminus\{\boldsymbol{0}\}} \sigma_{\boldsymbol{\ell}}^2(f)$. Hence the gain of scrambled digital nets lies in the fact that we only sum over $\sigma_{\boldsymbol{\ell}}^2(f)$ for which $|\boldsymbol{\ell}|_1 > m-t$, although one incurs a penalty factor of $b^t((b+1)/(b-1))^s$ using scrambled digital (t,m,s)-nets. Notice that the gain coefficients $\Gamma_{\boldsymbol{\ell}}$ are 0 for $\boldsymbol{\ell} \in \mathbb{N}_0^s$ with $|\boldsymbol{\ell}|_1 \le m-t$ and $\Gamma_{\boldsymbol{\ell}} = b^t((b+1)/(b-1))^s$ for $\boldsymbol{\ell} \in \mathbb{N}_0^s$ with $|\boldsymbol{\ell}|_1 > m-t$.

Theorem 6.22 shows that $\mathrm{Var}[\widehat{Q}_{n,s}(f)]$ for a scrambled $(0,m,s)$-net is always at most by a factor of $((b+1)/(b-1))^s$ larger than the variance for a plain MC algorithm (see also Owen (1997*a*, Theorem 1), who shows that the gain coefficients are always bounded by e in this case). Further, for scrambled (t,m,s)-nets we have

$$b^m\mathrm{Var}[\widehat{Q}_{n,s}(f)] = b^t\left(\frac{b+1}{b-1}\right)^s \sum_{\substack{\boldsymbol{\ell}\in\mathbb{N}_0^s \\ |\boldsymbol{\ell}|_1 > m-t}} \sigma_{\boldsymbol{\ell}}^2(f) \to 0 \quad \text{as } m \to \infty.$$

Hence, scrambled (t,m,s)-nets outperform MC with respect to $\mathrm{Var}[\widehat{Q}_{n,s}(f)]$ asymptotically, as for MC we have $n\mathrm{Var}[\widehat{Q}_{n,s}(f)] = \sum_{\boldsymbol{\ell}\in\mathbb{N}_0^s\setminus\{\boldsymbol{0}\}} \sigma_{\boldsymbol{\ell}}^2(f)$ for all $n \in \mathbb{N}$.

If the integrand f has smoothness $0 < \alpha \le 1$, then one can show that $|\sigma_{\boldsymbol{\ell}}(f)| \le C_f b^{-\alpha\mu_1(\boldsymbol{\ell})}$. Scrambled (t,m,s)-nets can take advantage now, since $\Gamma_{\boldsymbol{\ell}} = 0$ for all $\boldsymbol{\ell} \in \mathbb{N}_0^s$ with $\mu_1(\boldsymbol{\ell}) \le m-t$ and $\Gamma_{\boldsymbol{\ell}}$ is bounded otherwise. Thus, asymptotically one obtains an improved rate of convergence in this case. If the integrand is only in $L_2([0,1]^s)$, then one still gets the MC rate of convergence of $n^{-1/2}$.

Definition 6.23 (generalized Vitali variation). We define the *generalized variation in the sense of Vitali of order* $0 < \alpha \le 1$ by

$$V_\alpha^{(s)}(f) = \sup_{\mathcal{P}} \left(\sum_{J\in\mathcal{P}} \mathrm{Vol}(J)\left|\frac{\Delta(f,J)}{\mathrm{Vol}(J)^\alpha}\right|^2\right)^{1/2},$$

where the supremum is extended over all partitions $\mathcal{P}$ of $[0,1]^s$ into subintervals and $\mathrm{Vol}(J)$ denotes the volume of the subinterval J.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

For $\alpha = 1$ and if the partial derivatives of f are continuous on $[0,1]^s$, we also have the formula

$$V_1^{(s)}(f) = \left(\int_{[0,1]^s} \left| \frac{\partial^s f}{\partial x_1 \cdots \partial x_s}(\boldsymbol{x}) \right|^2 \mathrm{d}\boldsymbol{x} \right)^{1/2}.$$

Indeed we have

$$|\Delta(f,J)| = \left| \int_J \frac{\partial^s f}{\partial x_1 \cdots \partial x_s}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \right| = \mathrm{Vol}(J) \left| \frac{\partial^s f}{\partial x_1 \cdots \partial x_s}(\boldsymbol{\zeta}_J) \right|$$

for some $\boldsymbol{\zeta}_J \in \overline{J}$, which follows by applying the mean value theorem to the inequality

$$\begin{aligned} \min_{\boldsymbol{x} \in \overline{J}} \left| \frac{\partial^s f}{\partial x_1 \cdots \partial x_s}(\boldsymbol{x}) \right| &\le \mathrm{Vol}(J)^{-1} \left| \int_J \frac{\partial^s f}{\partial x_1 \cdots \partial x_s}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \right| \\ &\le \max_{\boldsymbol{x} \in \overline{J}} \left| \frac{\partial^s f}{\partial x_1 \cdots \partial x_s}(\boldsymbol{x}) \right|. \end{aligned}$$

Therefore we have

$$\sum_{J \in \mathcal{P}} \mathrm{Vol}(J) \left| \frac{\Delta(f,J)}{\mathrm{Vol}(J)} \right|^2 = \sum_{J \in \mathcal{P}} \mathrm{Vol}(J) \left| \frac{\partial^s f}{\partial x_1 \cdots \partial x_s}(\boldsymbol{\zeta}_J) \right|^2,$$

which is just a Riemann sum for the integral $\int_{[0,1]^s} \left| \frac{\partial^s f}{\partial x_1 \cdots \partial x_s} \right|^2 \mathrm{d}\boldsymbol{x}$, and thus the equality follows.

So far we have not taken into account projections to lower-dimensional faces.

Definition 6.24 (generalized Hardy and Krause variation). For $\emptyset \ne \mathfrak{u} \subseteq \{1:s\}$, let $V_\alpha^{(|\mathfrak{u}|)}(f_\mathfrak{u};\mathfrak{u})$ be the generalized Vitali variation with coefficient $0 < \alpha \le 1$ of the $|\mathfrak{u}|$-dimensional function

$$f_\mathfrak{u}(\boldsymbol{x}_\mathfrak{u}) = \int_{[0,1]^{s-|\mathfrak{u}|}} f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}_{\{1:s\}\setminus\mathfrak{u}}.$$

For $\mathfrak{u} = \emptyset$ we have $f_\emptyset = \int_{[0,1]^s} f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}_{\{1:s\}}$ and we define $V_\alpha^{(|\emptyset|)}(f_\emptyset;\emptyset) = |f_\emptyset|$. Then

$$V_\alpha(f) = \left(\sum_{\mathfrak{u} \subseteq \{1:s\}} \left(V_\alpha^{(|\mathfrak{u}|)}(f_\mathfrak{u};\mathfrak{u}) \right)^2 \right)^{1/2}$$

is called the *generalized Hardy and Krause variation* of f on $[0,1]^s$.

A function f for which $V_\alpha(f) < \infty$ is said to be of *finite variation of order* α.

Theorem 6.25 (a bound on the variance in terms of variation). Let $f : [0,1]^s \to \mathbb{R}$ have bounded variation $V_\alpha(f) < \infty$ of order $0 < \alpha \le 1$.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Then the variance of the estimator $\mathrm{Var}[\widehat{Q}_{n,s}(f)]$ using a randomly scrambled digital (t,m,s)-net over $\mathbb{Z}_b$ is bounded by

$$\mathrm{Var}[\widehat{Q}_{n,s}(f)] \le V_\alpha^2(f) b^{-(1+2\alpha)(m-t)} \frac{(b-1)^{(2\alpha-1)_+ s} b^{2s}}{b^{2\alpha}(b-1)^s} \binom{m-t+s}{s-1},$$

where again $(2\alpha-1)_+ = \max(2\alpha-1,0)$.

6.12. Simplifications of the scrambling scheme: affine matrix scrambling

The scrambling algorithm as introduced by Owen requires one to generate and store all the necessary permutations used for the scrambling scheme. Since the number of permutations needed is very large, this is not practical, and therefore simplifications of this algorithm have been found for which the main result still holds.

One version of such a simplified scrambling is the following: Let $x_j = x_{j,1}b^{-1} + x_{j,2}b^{-2} + \cdots$ denote the base b expansion of the jth coordinate of $\boldsymbol{x} = (x_1,\ldots,x_s) \in [0,1)^s$. Let $\boldsymbol{y} \in [0,1)^s$ denote the point which is obtained after scrambling $\boldsymbol{x}$. Assume that the jth coordinate has base b expansion $y_j = y_{j,1}b^{-1} + y_{j,2}b^{-2} + \cdots$. To obtain a simplified scrambling, choose $m_{j,k,\ell}, \sigma_{j,k} \in \mathbb{Z}_b$ for $1 \le \ell < k$ and $m_{j,k,k} \in \mathbb{Z}_b \setminus \{0\}$ for $1 \le j \le s$ independently and uniformly distributed over their ranges. Then we set

$$y_{j,k} = \sum_{\ell=1}^{k} m_{j,k,\ell} x_{j,\ell} + \sigma_{j,k} \pmod b \quad \text{for } 1 \le j \le s.$$

This can also be written in matrix form. Set $M_j = (m_{j,k,\ell})_{k,\ell}$, where $m_{j,k,\ell} = 0$ for $\ell > k$ and $\vec{\sigma}_j = (\sigma_{j,k})_k^\top$. Note that M_j is a lower triangular matrix. Further, let $\vec{x}_j = (x_{j,1}, x_{j,2}, \ldots)$ and $\vec{y}_j = (y_{j,1}, y_{j,2}, \ldots)^\top$. Then we have

$$\vec{y}_j = M\vec{x}_j + \vec{\sigma}.$$

This method reduces the number of permutations significantly while still preserving the main properties of Owen's scrambling. This scrambling scheme is called *affine matrix scrambling*. The results for Owen's scrambling presented above still hold for the affine matrix scrambling.

6.13. Higher-order scrambling

We have seen that scrambling can improve the convergence rate of the variance to $n^{-3/2+\delta}$, for any $\delta > 0$, for instance for functions whose generalized variation is bounded. Since there are higher-order nets which yield deterministic QMC rules with convergence rates of $n^{-\alpha+\delta}$ for arbitrary $\delta > 0$ for functions with smoothness $\alpha \ge 1$, the question also arises of how to generalize Owen's scrambling and its simplifications to achieve higher-order convergence. We briefly describe this in the following.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Let $\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{b^m-1} \in [0,1)^{ds}$ be a randomly scrambled digital (t, m, ds) net over the finite field $\mathbb{Z}_b$ of prime order b (here scrambling can mean Owen's original scrambling or any of the simplifications for which the same results hold as for Owen's scrambling). Then one simply uses the sample points

$$\boldsymbol{y}_i = \mathscr{D}_\alpha(\boldsymbol{x}_i) \in [0,1)^s \quad \text{for } 0 \le i < b^m,$$

where $\mathscr{D}_\alpha$ is the digit interlacing function of order $\alpha \ge 1$. The integral is then estimated using

$$\widehat{Q}_{n,s}(f) = \frac{1}{b^m} \sum_{n=0}^{b^m-1} f(\boldsymbol{y}_n).$$

This method again yields an unbiased estimator, that is,

$$\mathbb{E}(\widehat{Q}_{n,s}(f)) = \int_{[0,1]^s} f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}.$$

If the integrand has square-integrable partial mixed derivatives up to order $\alpha \ge 1$ in each variable, then the variance of $\widehat{Q}_{n,s}(f)$ satisfies

$$\mathrm{Var}[\widehat{Q}_{n,s}(f)] = O(n^{-2\min(d,\alpha)-1+\delta})$$

for any $\delta > 0$, where $n = b^m$ is the number of sample points.

6.14. Notes

Section 6.1 is taken partly from Dick and Pillichshammer (2010). Sections 6.4 and 6.5 are adapted from Dick and Pillichshammer (2010, Chapter 10). Section 6.6 is based on Dick and Baldeaux (2009). Sections 6.7 and 6.8 are based on Dick (2009*b*). Sections 6.10 and 6.11 are based on Dick and Pillichshammer (2010, Chapter13).

The monographs by Niederreiter (1992*a*) and by Dick and Pillichshammer (2010) give a comprehensive introduction to (digital) nets and (digital) sequences, discrepancy theory and QMC rules based on such point sets and sequences. More references can be found therein.

Walsh functions were introduced by Walsh (1923); see also Chrestenson (1955) and Fine (1949). For information on Walsh functions in the context of QMC integration, see Dick and Pillichshammer (2010).

Numerical integration of Walsh series using QMC rules based on digital nets was studied, for instance, by Larcher and Traunfellner (1994) and Larcher, Schmid and Wolf (1994, 1996*b*), whereas numerical integration in Haar wavelet spaces was studied by Sobol′ (1969) and Heinrich, Hickernell and Yue (2004). Numerical integration in anchored and unanchored Sobolev spaces using QMC rules based on digital nets was first studied by Dick and Pillichshammer (2005).

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Polynomial lattice rules have been introduced by Niederreiter (1992*b*). Shift nets are a subclass of digital nets which were introduced by Schmid (1996) and further studied by Pillichshammer (2002). The so-called Salzburg tables provide a table of digital nets found by computer search with small t-value. These computations are described by Hansen, Mullen and Niederreiter (1993), Larcher, Lauß, Niederreiter and Schmid (1996*a*) and Schmid (2000). An improvement of the method for computing the t-value of digital nets has been studied in Pirsic and Schmid (2001). A table with many of the currently best known t-values for (t, m, s)-nets and (t, s)-sequences for many values of m and s can be found at http://mint.sbg.ac.at/.

The CBC construction of polynomial lattice rules was introduced by Dick *et al.* (2005) for anchored and unanchored Sobolev spaces and by Dick, Kritzer, Leobacher and Pillichshammer (2007*a*) for the weighted star-discrepancy. The case where the modulus is not necessarily irreducible was considered by Kritzer and Pillichshammer (2007).

The fast CBC algorithm was introduced by Nuyens and Cools (2006*a*) using FFT methods for the construction of classical lattice point sets. Due to the similarities between ordinary and polynomial lattice point sets it turned out that their methods can also be carried over to the polynomial case: see Nuyens and Cools (2006*b*). Implementations of the fast algorithm using Matlab can be found in Nuyens and Cools (2006*c*).

Cyclic nets are another subclass of digital nets of size b^m which were introduced by Niederreiter (2004). Hyperplane nets are a subclass of digital nets of size b^{ms} which were introduced by Pirsic, Dick, Pillichshammer (2006). The discrepancy of cyclic and hyperplane nets was considered in Pillichshammer and Pirsic (2009). Extensible polynomial lattice rules were studied by Niederreiter (2003) and a construction algorithm was introduced by Dick (2007*b*). Extensible hyperplane nets were studied by Pirsic and Pillichshammer (2011). Constructions of (digital) nets and (digital) sequences based on existing constructions are called *propagation rules*: see Dick and Pillichshammer (2010, Chapter 9) for a summary.

The basic construction principle of higher-order digital nets and sequences appeared first in Dick (2007*a*) and was slightly modified in Dick (2008). A bound on the decay of the Walsh coefficients has been studied in more detail in Dick (2009*a*). Theorem 6.17 is a special case of Dick (2008, Theorem 4.11), with an improvement for some cases from Dick and Baldeaux (2009) (a proof of this result can be found in Dick and Baldeaux 2009, Dick 2007*a* and Dick and Kritzer 2010). The CBC construction of higher-order polynomial lattice rules was studied by Baldeaux, Dick, Greslehner and Pillichshammer (2011) and Baldeaux *et al.* (2012). A CBC construction of higher-order scrambled polynomial lattice rules was studied by Goda and Dick (2013). Propagation rules for higher-order digital nets were studied by Dick and Kritzer (2010). Tractability of higher-order polynomial

lattice rules was studied by Dick and Pillichshammer (2007) and the existence of higher-order polynomial lattice rules with small t-value was studied by Dick, Kritzer, Pillichshammer and Schmid (2007*b*). Recently Matsumoto, Saito and Matoba (2013) and Matsumoto and Yoshiki (2013) constructed QMC rules which achieve higher-order convergence for function classes of very high smoothness.

Scrambling was introduced by Owen (1995) and further studied in Owen (1997*a*, 1997*b*). Scrambled Niederreiter and Xing sequences were studied by Owen (1998). The mean-square discrepancy of scrambled nets was studied by Hickernell and Yue (2000), whereas integration and approximation using scrambled nets were studied by Yue and Hickernell (2001). The gain coefficients of digital nets were further investigated by Yue and Hickernell (2002). Strong tractability of scrambled Niederreiter sequences was studied by Yue and Hickernell (2005), whereas strong tractability in Banach spaces was studied by Yue and Hickernell (2006). Of particular interest in the context of scrambled nets is also the result by Loh (2003), who shows that a central limit theorem holds for the estimate $\widehat{Q}_{n,s}(f)$ for which the cubature points are based on a scrambled $(0, m, s)$-net. This allows one to obtain an approximate confidence interval from the variance estimates of $\widehat{Q}_{n,s}(f)$.

Simplifications of Owen's scrambling algorithm were studied by Hickernell (1996*b*), Matoušek (1998*b*) and Tezuka and Faure (2003). In particular, the affine matrix scrambling is from Matoušek (1998*b*) and is implemented in Matlab. Higher-order scrambling has been studied in Dick (2011*a*). A construction of randomly scrambled polynomial lattice rules which have a better dependence on the dimension has been studied in Baldeaux and Dick (2011) for functions of smoothness at most 1.

7. Infinite-dimensional integration

In this section we consider briefly numerical integration for problems with an infinite number of dimensions. We do not try to be definitive, because the subject is developing and changing rapidly. Rather, we introduce some themes likely to be of continuing importance. We concentrate here on 'anchored' function spaces, because anchoring is a natural idea for functions that depend on an infinite number of variables. The section concludes with a brief discussion of an application that is driving much recent interest in infinite-dimensional problems.

7.1. The infinite-dimensional problem

The problem is to integrate real-valued functions f that have a countably infinite number of variables,

$$f(x_1, x_2, \ldots),$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

with each component lying on the unit interval, $x_j \in [0,1], j \in \mathbb{N}$. Equivalently we write $f(\boldsymbol{x})$ with

$$\boldsymbol{x} = (x_j)_{j\in\mathbb{N}} \in [0,1]^{\mathbb{N}} := \prod_{j=1}^{\infty}[0,1].$$

To avoid formal problems associated with functions of an infinite number of variables, we choose an 'anchor' $c \in [0,1]$, and in our algorithms only allow function evaluation when at most a finite number of components of $\boldsymbol{x}$ have values different from the anchor value c. Often $c = 0$ is the most natural choice for the anchor, but in some applications it is better to take the midpoint $c = 1/2$ as anchor. The components x_j that have the value c are considered to be 'inactive variables' for the particular function evaluation, in contrast to the 'active variables' $x_j \neq c$.

If the set of active variables is $\boldsymbol{x}_{\mathfrak{u}}$, we write the value of f as $f(\boldsymbol{x}_{\mathfrak{u}}; \boldsymbol{c})$. In this section we shall assume that $f(\boldsymbol{x}_{\mathfrak{u}}; \boldsymbol{c})$ is a continuous function of $\boldsymbol{x}_{\mathfrak{u}}$ on $[0,1]^{|\mathfrak{u}|}$ for every finite subset $\mathfrak{u}$ of the natural numbers. The infinite-dimensional integral may now be defined by

$$I_\infty(f) := \lim_{s\to\infty} \int_{[0,1]^s} f(\boldsymbol{x}_{\{1:s\}}; \boldsymbol{c})\, \mathrm{d}\boldsymbol{x}_{\{1:s\}}. \tag{7.1}$$

In Section 7.4 we shall introduce a reproducing kernel Hilbert space setting that ensures that this limit exists, and ensures also that $f(\boldsymbol{x})$ is well defined for all $\boldsymbol{x} \in [0,1]^{\mathbb{N}}$. For the present, however, we concentrate on algorithms and costs.

Our algorithms for approximating this infinite-dimensional integral all have the form

$$Q(f) = \sum_{i=0}^{n-1} w_i f(\boldsymbol{t}^{(i)}_{\mathfrak{u}_i}; \boldsymbol{c}), \tag{7.2}$$

where $w_i \in \mathbb{R}$, $\mathfrak{u}_i$ is a finite subset of $\mathbb{N}$, and $\boldsymbol{t}^{(i)}_{\mathfrak{u}_i} \in [0,1]^{|\mathfrak{u}_i|}$. Three kinds of algorithms have been studied in the literature, which we introduce in the following subsection.

7.2. Three kinds of algorithms

In this subsection we describe *single-level* (or *fixed-dimension*), *multi-level* and *changing-dimension* algorithms.

Example 7.1 (single-level algorithm). The *single-level* (SL) or *fixed-dimension* algorithm approximates the infinite-dimensional integral by an s-dimensional QMC rule,

$$Q^{\mathrm{SL}}(f) := \frac{1}{n}\sum_{i=0}^{n-1} f(\boldsymbol{t}^{(i)}_{\{1:s\}}; \boldsymbol{c}), \tag{7.3}$$

in which the active variables are the leading variables $x_1, x_2, \ldots, x_s$.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Example 7.2 (multi-level algorithm). A different kind of algorithm comes from adopting a *multi-level* (ML) approach (Heinrich 1998, Giles 2008). We define an infinite sequence of dimensions

$$0 = s_0 < s_1 < s_2 < \cdots < \infty,$$

and write the function as a collapsing sum,

$$f(\boldsymbol{x}) = \sum_{\ell=1}^{\infty} \bigl(f(\boldsymbol{x}_{\{1:s_\ell\}}; \boldsymbol{c}) - f(\boldsymbol{x}_{\{1:s_{\ell-1}\}}; \boldsymbol{c})\bigr) + f(\boldsymbol{c}),$$

and similarly write the integral (7.1) also as a collapsing sum,

$$I_\infty(f) = \sum_{\ell=1}^{\infty} \bigl(I_{s_\ell}(f) - I_{s_{\ell-1}}(f)\bigr) + f(\boldsymbol{c}),$$

with $I_0(f) = f(\boldsymbol{c})$, and

$$I_{s_\ell}(f) := \int_{[0,1]^{s_\ell}} f(\boldsymbol{x}_{\{1:s_\ell\}}; \boldsymbol{c})\, \mathrm{d}\boldsymbol{x}_{\{1:s_\ell\}}.$$

We then approximate the collapsing sum up to level $L > 0$ by the cubature formula

$$Q^{\mathrm{ML}}(f) := \sum_{\ell=1}^{L} Q_{n_\ell,\ell}(\psi_\ell(f) - \psi_{\ell-1}(f)) + f(\boldsymbol{c}), \tag{7.4}$$

where $\psi_\ell(f) := f(\boldsymbol{x}_{\{1:s_\ell\}}; \boldsymbol{c})$, and

$$Q_{n_\ell,\ell}(\psi_\ell(f) - \psi_{\ell-1}(f)) = \frac{1}{n_\ell} \sum_{i=0}^{n_\ell-1} \Bigl(f(\boldsymbol{t}^{(i)}_{\{1:s_\ell\}}; \boldsymbol{c}) - f(\boldsymbol{t}^{(i)}_{\{1:s_{\ell-1}\}}; \boldsymbol{c})\Bigr),$$

and where $\boldsymbol{t}^{(i)}_{\{1:s_\ell\}} \in [0,1]^{s_\ell}$, and $\boldsymbol{t}^{(i)}_{\{1:s_{\ell-1}\}}$ contains the first $s_{\ell-1}$ components of $\boldsymbol{t}^{(i)}_{\{1:s_\ell\}}$. For $n_\ell = 0$ we assume that $Q_{n_\ell,\ell}(\psi_\ell(f) - \psi_{\ell-1}(f)) = 0$. The idea of an ML scheme is that the successive terms in the collapsing sum for the integral should be smaller and smaller, and hence can be well approximated by a smaller number of points n_ℓ as ℓ increases, leaving only relatively low-dimensional integrals needing a large number of points. We assumed in (7.4) that the cubature rules $Q_{n_\ell,\ell}$ are QMC rules, but other choices are clearly possible.

Roughly speaking, ML algorithms are of particular advantage if for a particular f the important subsets $\mathfrak{u}$ involve only a small number of leading variables (that is, in the language of Caflisch *et al.* 1997, if the 'truncation dimension' is small).

Example 7.3 (changing-dimension algorithm). A third kind of algorithm is the *changing-dimension* (CD) algorithm (Kuo, Sloan, Wasilkowski and Woźniakowski 2010*c*). In this approach a different cubature rule is

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

applied to each term $f_{\mathfrak{u}}$ of the so-called 'anchored' decomposition of f. The anchored decomposition has the form

$$f = \sum_{|\mathfrak{u}|<\infty} f_{\mathfrak{u}}, \tag{7.5}$$

where for a finite set $\mathfrak{u}$ the function $f_{\mathfrak{u}}$ depends only on the set $\boldsymbol{x}_{\mathfrak{u}}$ of active variables $x_j, j \in \mathfrak{u}$, and vanishes if any of those active variables has the value c. The anchored decomposition has appeared under many different guises: see for example Sobol′ (1969), Li, Schoendorf, Ho and Rabitz (2004) and Griebel (2006). The terms in the anchored decomposition may be defined recursively by

$$f_{\mathfrak{u}}(\boldsymbol{x}_{\mathfrak{u}}) = f(\boldsymbol{x}_{\mathfrak{u}}; \boldsymbol{c}) - \sum_{\mathfrak{v}\subset\mathfrak{u}} f_{\mathfrak{v}}(\boldsymbol{x}_{\mathfrak{v}}), \tag{7.6}$$

where the sum is over the strict subsets of $\mathfrak{u}$. An explicit formula for $f_{\mathfrak{u}}$ in terms of f is given by

$$f_{\mathfrak{u}}(\boldsymbol{x}_{\mathfrak{u}}) = \sum_{\mathfrak{v}\subseteq\mathfrak{u}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|} f(\boldsymbol{x}_{\mathfrak{v}}; \boldsymbol{c}), \tag{7.7}$$

where now the sum is over all subsets of $\mathfrak{u}$. The formula is established, for example, in Theorem 1 of Kuo, Sloan, Wasilkowski and Woźniakowski (2010*b*).

The CD algorithms then take the form

$$Q^{\mathrm{CD}}(f) := \sum_{|\mathfrak{u}|<\infty} Q_{n_{\mathfrak{u}},\mathfrak{u}}(f_{\mathfrak{u}}), \tag{7.8}$$

where, for a finite subset $\mathfrak{u}$,

$$Q_{n_{\mathfrak{u}},\mathfrak{u}}(f_{\mathfrak{u}}) = \frac{1}{n_{\mathfrak{u}}} \sum_{i=0}^{n_{\mathfrak{u}}-1} f_{\mathfrak{u}}(\boldsymbol{t}_{\mathfrak{u}}^{(i)}),$$

with $\boldsymbol{t}_{\mathfrak{u}}^{(i)} \in [0,1]^{|\mathfrak{u}|}$, and we use the convention that $Q_{n_{\mathfrak{u}},\mathfrak{u}}(f_{\mathfrak{u}})$ vanishes if $n_{\mathfrak{u}} = 0$. Thus in the CD algorithm a different cubature rule is applied to each component of the anchored decomposition. We assumed for simplicity that the cubature rule $Q_{n_{\mathfrak{u}},\mathfrak{u}}$ is a QMC rule, but other choices are clearly possible. Roughly speaking, this algorithm has a significant advantage if for a particular f the important subsets $\mathfrak{u}$ all have small cardinality (that is, in the language of Caflisch *et al.* 1997, if the 'superposition dimension' is small).

7.3. Cost models and the choice of algorithms

In earlier sections we implicitly assumed that the cost of a single function evaluation is always the same, since in assessing cost we only counted the number of function evaluations. Now that the number of variables is

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

unbounded, it seems reasonable to allow the cost of evaluating $f(\boldsymbol{x}_{\mathfrak{u}}; \boldsymbol{c})$ to depend on the set $\mathfrak{u}$. We consider two different cost models.

- Model A: the cost depends on the highest index of the active variables
$$\mathrm{cost}_A(\mathfrak{u}) = \$(\max \mathfrak{u}).$$
- Model B: the cost depends on the number of active variables
$$\mathrm{cost}_B(\mathfrak{u}) = \$(|\mathfrak{u}|).$$

Here the 'dollar' function $\$: \mathbb{N}_0 \to \mathbb{R}_+$ is assumed to be positive and non-decreasing. For example, $\$(k) = [\max(1,k)]^\sigma$ for some $\sigma > 0$. We note that Model A is a simplification of a model introduced in Creutzig, Dereich, Müller-Gronbach and Ritter (2009) and Model B was introduced in Kuo *et al.* (2010*c*).

Whereas in previous sections we assumed that the cost of an algorithm is simply the number of function evaluations, we now assume that the cost of the algorithm given by (7.2) is

$$\mathrm{cost}_X(Q) := \sum_{i=0}^{n-1} \mathrm{cost}_X(\mathfrak{u}_i), \quad X \in \{A, B\}. \tag{7.9}$$

The aim now is to approximate the integral with accuracy $\varepsilon > 0$ by an algorithm with small cost (rather than a small number of function evaluations).

The costs of the SL and ML algorithms under both cost models are, respectively,

$$\mathrm{cost}_X(Q^{\mathrm{SL}}) = n\,\$(s) \quad \text{and} \quad \mathrm{cost}_X(Q^{\mathrm{ML}}) = \sum_{\ell=1}^{L} n_\ell\big(\$(s_\ell) + \$(s_{\ell-1})\big),$$

while the cost of the CD algorithm under each cost model is

$$\begin{aligned}
\mathrm{cost}_A(Q^{\mathrm{CD}}) &= \sum_{|\mathfrak{u}|<\infty} n_{\mathfrak{u}} \sum_{\mathfrak{v}\subseteq\mathfrak{u}} \$(\max \mathfrak{v}) \le \sum_{|\mathfrak{u}|<\infty} n_{\mathfrak{u}}\,\$(\max \mathfrak{u})\,2^{|\mathfrak{u}|},\\
\mathrm{cost}_B(Q^{\mathrm{CD}}) &= \sum_{|\mathfrak{u}|<\infty} n_{\mathfrak{u}} \sum_{\mathfrak{v}\subseteq\mathfrak{u}} \$(|\mathfrak{v}|) \le \sum_{|\mathfrak{u}|<\infty} n_{\mathfrak{u}}\,\$(|\mathfrak{u}|)\,2^{|\mathfrak{u}|},
\end{aligned}$$

where the sum over $\mathfrak{v}$ appears due to the need to compute $f_{\mathfrak{u}}$ using (7.7).

One cost model may be more appropriate than the other, depending on the integrand for the application at hand. The choice of algorithm should therefore also depend on the cost model for the given application.

7.4. A reproducing kernel Hilbert space setting

Now we restrict the function class further. Just as in the finite-dimensional case, we restrict our considerations to functions that lie in a certain reproducing Hilbert space (RKHS), because it is then possible to compute worst-case errors, and to study tractability of the integration problem.

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

Let $\eta : [0,1]^2 \to \mathbb{R}$ be a reproducing kernel on the interval $[0,1]$ such that there exists a $c \in [0,1]$ with $\eta(c,c) = 0$. Examples of reproducing kernels satisfying this condition are $\min(x,y)$ (see (4.1) with $c = 0$) or $1 - \max(x,y)$ (see (4.1) with $c = 1$). Let the RKHS be denoted by $H(\eta)$, the inner product by $\langle \cdot, \cdot \rangle$ and the corresponding norm by $\| \cdot \|$. Assume that

$$M := \sup_{x \in [0,1]} |\eta(x,x)| < \infty.$$

Then, since

$$|\eta(x,y)| = |\langle \eta(\cdot,x), \eta(\cdot,y) \rangle| \le \|\eta(\cdot,x)\| \|\eta(\cdot,y)\| = \sqrt{\eta(x,x)\eta(y,y)} \quad (7.10)$$

for all $x, y \in [0,1]$, we have $|\eta(x,c)| \le \sqrt{M\eta(c,c)} = 0$.Thus

$$\eta(x,c) = 0, \quad \text{for all } x \in [0,1], \quad (7.11)$$

and we have

$$f(c) = \langle f, \eta(\cdot,c) \rangle = \langle f, 0 \rangle = 0, \quad \text{for all } f \in H(\eta).$$

Thus all functions $f \in H(\eta)$ vanish at c, in particular, the only constant function in $H(\eta)$ is $f \equiv 0$. This implies that the space of constant functions $H(1)$, defined by the kernel 1, is orthogonal to $H(\eta)$. Moreover, the kernel $1 + \eta$ defines an RKHS which is the direct sum of constant functions and functions in $H(\eta)$. This implies that the space of constant functions $H(1)$, defined by the kernel 1, is orthogonal to $H(\eta)$ in the space $H(1 + \eta)$.

To define the infinite-dimensional RKHS, for each set $\mathfrak{u} \subset \mathbb{N}$ with $|\mathfrak{u}| < \infty$ let $\gamma_\mathfrak{u}$ be a non-negative real number, and for points $\boldsymbol{x}, \boldsymbol{y} \in [0,1]^\mathbb{N}$ let

$$K_\mathfrak{u}(\boldsymbol{x}_\mathfrak{u}, \boldsymbol{y}_\mathfrak{u}) := \prod_{j \in \mathfrak{u}} \eta(x_j, y_j) \quad \text{and} \quad K_{\boldsymbol{\gamma}}(\boldsymbol{x}, \boldsymbol{y}) := \sum_{|\mathfrak{u}| < \infty} \gamma_\mathfrak{u} K_\mathfrak{u}(\boldsymbol{x}_\mathfrak{u}, \boldsymbol{y}_\mathfrak{u}), \quad (7.12)$$

where again the empty product is defined to be 1. In the following we assume that

$$\sum_{|\mathfrak{u}| < \infty} \gamma_\mathfrak{u} M^{|\mathfrak{u}|} < \infty. \quad (7.13)$$

Condition (7.13) ensures that the series expression for $K_{\boldsymbol{\gamma}}(\boldsymbol{x}, \boldsymbol{y})$ converges absolutely and uniformly, since from (7.10) we have $|\eta(x,y)| \le M$. The kernel $K_\mathfrak{u}$ defines an RKHS $H(K_\mathfrak{u})$ on $[0,1]^{|\mathfrak{u}|}$, and the kernel $K_{\boldsymbol{\gamma}}$ defines an RKHS $H(K_{\boldsymbol{\gamma}})$ on $[0,1]^\mathbb{N}$. The inner products in $H(K_{\boldsymbol{\gamma}})$ and $H(K_\mathfrak{u})$ will be denoted by $\langle \cdot, \cdot \rangle_{\boldsymbol{\gamma}}$ and $\langle \cdot, \cdot \rangle_\mathfrak{u}$ respectively, and the corresponding norms by $\| \cdot \|_{\boldsymbol{\gamma}}$ and $\| \cdot \|_\mathfrak{u}$.

Note that a condition significantly weaker than (7.13) would be sufficient to define infinite-dimensional integration, but to avoid some technicalities we shall be content with (7.13). In particular, our assumptions imply that function evaluation is continuous at every point $\boldsymbol{x} \in [0,1]^\mathbb{N}$. The

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

idea for weakening the conditions is to impose conditions such that infinite-dimensional integration and function evaluation at finite-dimensional projections are continuous linear functionals. See the work by Gnewuch, Mayer and Ritter (2013) for even weaker conditions.

A function $f \in H(K_{\boldsymbol{\gamma}})$ can be decomposed into a sum of functions $f_{\mathfrak{u}} \in H(K_{\mathfrak{u}})$ by setting

$$f_{\mathfrak{u}}(\boldsymbol{x}_{\mathfrak{u}}) := \langle f, \gamma_{\mathfrak{u}} K_{\mathfrak{u}}(\cdot, \boldsymbol{x}_{\mathfrak{u}})\rangle_{\boldsymbol{\gamma}}, \tag{7.14}$$

which implies

$$\sum_{|\mathfrak{u}|<\infty} f_{\mathfrak{u}}(\boldsymbol{x}_{\mathfrak{u}}) = \sum_{|\mathfrak{u}|<\infty} \langle f, \gamma_{\mathfrak{u}} K_{\mathfrak{u}}(\cdot, \boldsymbol{x}_{\mathfrak{u}})\rangle_{\boldsymbol{\gamma}} = \left\langle f, \sum_{|\mathfrak{u}|<\infty} \gamma_{\mathfrak{u}} K_{\mathfrak{u}}(\cdot, \boldsymbol{x}_{\mathfrak{u}}) \right\rangle_{\boldsymbol{\gamma}} = f(\boldsymbol{x}). \tag{7.15}$$

The decomposition $f = \sum_{|\mathfrak{u}|<\infty} f_{\mathfrak{u}}$ is orthogonal in $H(K_{\boldsymbol{\gamma}})$, since $H(\eta)$ is orthogonal to the space of constant functions and for $\mathfrak{u} \neq \mathfrak{v}$ either there exists $j \in \mathfrak{u}, j \notin \mathfrak{v}$ or there exists $j \in \mathfrak{v}, j \notin \mathfrak{u}$, implying that $\langle K_{\mathfrak{u}}(\cdot, \boldsymbol{x}_{\mathfrak{u}}), K_{\mathfrak{v}}(\cdot, \boldsymbol{y}_{\mathfrak{v}})\rangle_{\boldsymbol{\gamma}} = 0$ and hence by (7.14) we have $\langle f_{\mathfrak{u}}, K_{\mathfrak{v}}(\cdot, \boldsymbol{y}_{\mathfrak{v}})\rangle_{\boldsymbol{\gamma}} = 0$.

By orthogonality, the norm in $H(K_{\boldsymbol{\gamma}})$ is related to the norm of the projections by the formula

$$\|f\|_{\boldsymbol{\gamma}}^2 = \sum_{|\mathfrak{u}|<\infty} \|f_{\mathfrak{u}}\|_{\boldsymbol{\gamma}}^2 = \sum_{|\mathfrak{u}|<\infty} \gamma_{\mathfrak{u}}^{-1} \|f_{\mathfrak{u}}\|_{\mathfrak{u}}^2.$$

We now study integration of functions defined on $H(K_{\boldsymbol{\gamma}})$. Let

$$h_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}) := \int_{[0,1]^{|\mathfrak{u}|}} K_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}, \boldsymbol{x}_{\mathfrak{u}})\, \mathrm{d}\boldsymbol{x}_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \int_0^1 \eta(y_j, x_j)\, \mathrm{d}x_j,$$

and

$$h(\boldsymbol{y}) := \sum_{|\mathfrak{u}|<\infty} \gamma_{\mathfrak{u}} h_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}).$$

From (7.13) we conclude that this series converges absolutely and uniformly. The functions $h_{\mathfrak{u}}$ are orthogonal in $H(K_{\boldsymbol{\gamma}})$ (since $h_{\mathfrak{u}} \in H(K_{\mathfrak{u}})$), from which it follows that

$$\begin{aligned} \|h\|_{\boldsymbol{\gamma}}^2 &= \sum_{|\mathfrak{u}|<\infty} \gamma_{\mathfrak{u}}^2 \|h_{\mathfrak{u}}\|_{\boldsymbol{\gamma}}^2 = \sum_{|\mathfrak{u}|<\infty} \gamma_{\mathfrak{u}}^2 \gamma_{\mathfrak{u}}^{-1} \|h_{\mathfrak{u}}\|_{\mathfrak{u}}^2 \\ &= \sum_{|\mathfrak{u}|<\infty} \gamma_{\mathfrak{u}}^2 \gamma_{\mathfrak{u}}^{-1} \int_{[0,1]^{|\mathfrak{u}|}} \int_{[0,1]^{|\mathfrak{u}|}} \langle K_{\mathfrak{u}}(\cdot, \boldsymbol{x}), K_{\mathfrak{u}}(\cdot, \boldsymbol{y})\rangle_{\mathfrak{u}}\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y} \\ &= \sum_{|\mathfrak{u}|<\infty} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \int_0^1 \int_0^1 \eta(x_j, y_j)\, \mathrm{d}x_j\, \mathrm{d}y_j < \sum_{|\mathfrak{u}|<\infty} \gamma_{\mathfrak{u}} M^{|\mathfrak{u}|} < \infty, \end{aligned} \tag{7.16}$$

and thus $h \in H(K_{\boldsymbol{\gamma}})$. Note that this decomposition of h differs from the

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

decomposition (7.15) (which holds for all functions in $H(K_{\boldsymbol{\gamma}})$) by a scaling factor $\gamma_{\mathfrak{u}}$ for each term $h_{\mathfrak{u}}$.

For $f \in H(K_{\boldsymbol{\gamma}})$ it follows from (7.1) and (7.12) that

$$\begin{aligned}\int_{[0,1]^{\mathbb{N}}} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} &= \lim_{s\to\infty} \int_{[0,1]^s} f(\boldsymbol{x}_{\{1:s\}};\boldsymbol{c})\,\mathrm{d}\boldsymbol{x}\\ &= \lim_{s\to\infty} \int_{[0,1]^s} \left\langle f, \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} K_{\mathfrak{u}}(\boldsymbol{x}_{\mathfrak{u}},\cdot)\right\rangle_{\boldsymbol{\gamma}} \mathrm{d}\boldsymbol{x}\\ &= \lim_{s\to\infty} \left\langle f, \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} h_{\mathfrak{u}}\right\rangle_{\boldsymbol{\gamma}} = \langle f, h\rangle_{\boldsymbol{\gamma}}. \end{aligned} \tag{7.17}$$

Since $\|I_\infty\| = \|h\|_{\boldsymbol{\gamma}} < \infty$, it follows that I_∞ is a bounded linear functional on $H(K_{\boldsymbol{\gamma}})$.

7.5. Error analysis in RKHS

The analysis of the worst-case error is based on the orthogonal decomposition in the space $H(K_{\boldsymbol{\gamma}})$. The representer (see (3.11)) of the integration error is given by

$$\xi(\boldsymbol{x}) = \int_{[0,1]^{\mathbb{N}}} K_{\boldsymbol{\gamma}}(\boldsymbol{y},\boldsymbol{x})\,\mathrm{d}\boldsymbol{y} - Q(K_{\boldsymbol{\gamma}}(\cdot,\boldsymbol{x})),$$

where, in general, Q is a cubature algorithm of the form (7.2) which is applied to the first variable of $K_{\boldsymbol{\gamma}}$ (later we will consider only the SL, ML, CD algorithms). As explained in Section 3.3, the worst-case integration error is given by

$$e(Q; H(K_{\boldsymbol{\gamma}})) = \|\xi\|_{\boldsymbol{\gamma}}.$$

(In this subsection we use the notation $e(Q; H(K_{\boldsymbol{\gamma}}))$ instead of $e(P; H(K_{\boldsymbol{\gamma}}))$ as in Section 3.3, since we now allow unequal cubature weights in the algorithm.) Analogously to (7.16), we can obtain an orthogonal decomposition of the worst-case error

$$\begin{aligned} e^2(Q; H(K_{\boldsymbol{\gamma}})) = \|\xi\|_{\boldsymbol{\gamma}}^2 &= \sum_{|\mathfrak{u}|<\infty} \gamma_{\mathfrak{u}}^2 \|\xi_{\mathfrak{u}}\|_{\boldsymbol{\gamma}}^2\\ &= \sum_{|\mathfrak{u}|<\infty} \gamma_{\mathfrak{u}} \|\xi_{\mathfrak{u}}\|_{\mathfrak{u}}^2 = \sum_{|\mathfrak{u}|<\infty} \gamma_{\mathfrak{u}}\, e^2(Q_{\mathfrak{u}}; H(K_{\mathfrak{u}})), \end{aligned}$$

where $e(Q_{\mathfrak{u}}; H(K_{\mathfrak{u}}))$ is the worst-case error in the space $H(K_{\mathfrak{u}})$ using the cubature rule

$$Q_{\mathfrak{u}}(f) = \sum_{i=0}^{n-1} w_i f_{\mathfrak{u}}(\boldsymbol{t}^{(i)}_{\mathfrak{u}_i\cap\mathfrak{u}};\boldsymbol{c}).$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

The relationship between f and its projection $f_{\mathfrak{u}}$ is given by (7.6) and (7.7). Note that if for $\mathfrak{u} \neq \emptyset$ there does not exist at least one $0 \leq i < n$ such that $\mathfrak{u} \subseteq \mathfrak{u}_i$, then $Q_{\mathfrak{u}}(f) = 0$ since $f_{\mathfrak{u}}(\boldsymbol{t}_{\mathfrak{v}}; \boldsymbol{c}) = 0$ for $\mathfrak{v} \subseteq \mathfrak{u}$ and $\mathfrak{v} \neq \mathfrak{u}$. This orthogonal decomposition of the worst-case error is an essential tool in analysing the integration error. We consider now the orthogonal decompositions of the worst-case error for the SL, ML and CD algorithms in more detail.

For the SL algorithm we obtain

$$\begin{aligned} e^2(Q^{\mathrm{SL}}; H(K_{\boldsymbol{\gamma}})) &= \sum_{\mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}}\, e^2(Q_{\mathfrak{u}}; H(K_{\mathfrak{u}})) + \sum_{\substack{|\mathfrak{u}|<\infty \\ \mathfrak{u} \not\subseteq \{1:s\}}} \gamma_{\mathfrak{u}}\, e^2(0; H(K_{\mathfrak{u}})) \\ &= e^2(Q^{\mathrm{SL}}; H(K_{s,\boldsymbol{\gamma}})) + \sum_{\substack{|\mathfrak{u}|<\infty \\ \mathfrak{u} \not\subseteq \{1:s\}}} \gamma_{\mathfrak{u}}\, m^{|\mathfrak{u}|}, \end{aligned}$$

where $e^2(0; H(K_{\mathfrak{u}}))$ is the squared initial error (see (3.10))

$$e^2(0; H(K_{\mathfrak{u}})) = m^{|\mathfrak{u}|}, \quad m := \int_0^1 \int_0^1 \eta(x, y)\, \mathrm{d}x\, \mathrm{d}y \leq M,$$

and $e^2(Q^{\mathrm{SL}}; H(K_{s,\boldsymbol{\gamma}}))$ is the squared worst-case error of the cubature rule for the finite-dimensional weighted space $H(K_{s,\boldsymbol{\gamma}})$ with reproducing kernel

$$K_{s,\boldsymbol{\gamma}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}} K_{\mathfrak{u}}(\boldsymbol{x}_{\mathfrak{u}}, \boldsymbol{y}_{\mathfrak{u}}), \quad \boldsymbol{x}, \boldsymbol{y} \in [0,1]^s.$$

We can then apply any estimate for $e^2(Q^{\mathrm{SL}}; H(K_{s,\boldsymbol{\gamma}}))$ based on lattice rules or digital nets from the previous sections. We can also use general cubature rules other than QMC rules.

The ML algorithm approximates each level ℓ using a cubature rule with n_ℓ points. To bound the ML cubature error, note that from (7.15) we have

$$f(\boldsymbol{x}_{\{1:s_\ell\}}; \boldsymbol{c}) - f(\boldsymbol{x}_{\{1:s_{\ell-1}\}}; \boldsymbol{c}) = \sum_{\mathfrak{u} \subseteq \{1:s_\ell\}} f_{\mathfrak{u}}(\boldsymbol{x}_{\mathfrak{u}}) - \sum_{\mathfrak{u} \subseteq \{1:s_{\ell-1}\}} f_{\mathfrak{u}}(\boldsymbol{x}_{\mathfrak{u}}).$$

Thus the error for the ML algorithm satisfies

$$\begin{aligned} &e^2(Q^{\mathrm{ML}}; H(K_{\boldsymbol{\gamma}})) \\ &= \sum_{\ell=1}^{L} \sum_{\substack{\mathfrak{u} \subseteq \{1:s_\ell\} \\ \mathfrak{u} \not\subseteq \{1:s_{\ell-1}\}}} \gamma_{\mathfrak{u}} e^2(Q_{\mathfrak{u}}^{\mathrm{ML}}; H(K_{\mathfrak{u}})) + \sum_{\substack{|\mathfrak{u}|<\infty \\ \mathfrak{u} \not\subseteq \{1:s_L\}}} \gamma_{\mathfrak{u}}\, e^2(0; H(K_{\mathfrak{u}})) \end{aligned}$$

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

$$= \sum_{\ell=1}^{L} e^2(Q_{n_\ell,\ell}; H(K_{s_\ell,\boldsymbol{\gamma}} - K_{s_{\ell-1},\boldsymbol{\gamma}})) + \sum_{\substack{|\mathfrak{u}|<\infty \\ \mathfrak{u}\not\subseteq\{1:s_L\}}} \gamma_\mathfrak{u}\, m^{|\mathfrak{u}|}$$

$$\le \sum_{\ell=1}^{L} e^2(Q_{n_\ell,\ell}; H(K_{s_\ell,\boldsymbol{\gamma}})) + \sum_{\substack{|\mathfrak{u}|<\infty \\ \mathfrak{u}\not\subseteq\{1:s_L\}}} \gamma_\mathfrak{u}\, m^{|\mathfrak{u}|}.$$

We then apply previous results for $e^2(Q_{n_\ell,\ell}; H(K_{s_\ell,\boldsymbol{\gamma}} - K_{s_{\ell-1},\boldsymbol{\gamma}}))$ or simply $e^2(Q_{n_\ell,\ell}; H(K_{s_\ell,\boldsymbol{\gamma}}))$.

For the CD algorithm we have

$$e^2(Q^{\mathrm{CD}}; H(K_{\boldsymbol{\gamma}})) = \sum_{|\mathfrak{u}|<\infty} \gamma_\mathfrak{u}\, e^2(Q_{n_\mathfrak{u},\mathfrak{u}}; H(K_\mathfrak{u}))$$

$$= \sum_{\substack{|\mathfrak{u}|<\infty \\ n_\mathfrak{u}>0}} \gamma_\mathfrak{u}\, e^2(Q_{n_\mathfrak{u},\mathfrak{u}}; H(K_\mathfrak{u})) + \sum_{\substack{|\mathfrak{u}|<\infty \\ n_\mathfrak{u}=0}} \gamma_\mathfrak{u}\, m^{|\mathfrak{u}|}.$$

Again, we then apply known estimates for $e^2(Q_{n_\mathfrak{u},\mathfrak{u}}; H(K_\mathfrak{u}))$.

The idea now is to choose the number of cubature points (n for SL, n_ℓ for ML, $n_\mathfrak{u}$ for CD) and the dimensions or sets of variables (s for SL, s_ℓ for ML, active sets $\mathfrak{u}$ for CD), such that the error is small for a given cost, or conversely, the cost is small for a given required level of accuracy. One needs to balance the error obtained by truncating the infinite-dimensional integral against the integration error. This leads to a constrained optimization problem which in some cases can be solved using the technique of Lagrange multipliers. For a given application, one would also need to determine suitable weights $\gamma_\mathfrak{u}$ so that the integrand not only belongs to the RKHS setting but also has small norm $\|\cdot\|_{\boldsymbol{\gamma}}$.

7.6. An infinite-dimensional application

Do infinite-dimensional problems arise in practice? A class of infinite-dimensional problems arises from partial differential equations with random fields as coefficients, with one problem being the flow of a liquid (oil or water) through a porous medium (such as rock). For a given pressure gradient, the rate of flow at a particular location depends on the local 'permeability' of the medium, which can vary rapidly from point to point. Engineers studying the overall flow properties then have two choices: either to model the true permeability as a rapidly varying function over the region, or (as is often done in practice) to model the permeability as a 'random field' whose general characteristics match those of the physical problem. A random field over a two- or three-dimensional region may require a countably infinite number of independent random variables for its complete description, hence

https://doi.org/10.1017/S0962492913000044 Published online by Cambridge University Press

the infinite-dimensionality of the problem. Such problems are presently tackled by a variety of methods, beginning with Norbert Wiener's 'polynomial chaos', and continuing to 'generalized polynomial chaos', 'stochastic Galerkin', and 'stochastic collocation' methods. In all those methods the probabilistic aspect of the problem is parametrized by a continuous variable $\boldsymbol{x}$ with a large (possibly infinite) number of components. The solution is approximated by truncation to a manageable number of components of $\boldsymbol{x}$, and an approximate solution depending on both the truncated $\boldsymbol{x}$ and the physical variables is then sought in a finite-dimensional function space. All such methods face great challenges when the effective dimension is large (where the effective dimension may be loosely defined as the number of components of $\boldsymbol{x}$ needed to obtain a reliable approximation). For hard problems of this type MC or QMC methods may be the engineer's only choice (Ghanem and Spanos 1991).

In a recent paper (Kuo *et al.* 2012) a QMC method (randomly shifted lattice rule) was applied to a simple (but infinite-dimensional) model of a partial differential equation with a random coefficient, using the methodology of Section 5, after truncation to a finite dimension s. An interesting feature of this work was the appearance for the first time of POD (product and order-dependent) weights (see (4.4)), obtained by minimizing a certain upper bound on the error of some functional (for example, the overall effective permeability) of the solution. An ML version of the same problem was studied in Kuo, Schwab and Sloan (2013), using an ML scheme more complicated than that above, in that a different spatial discretization was employed at each level ℓ.

7.7. Notes

Early studies of numerical integration for infinite-dimensional problems (and ML methods) include the following: Wasilkowski and Woźniakowski (1995) considered Feynman–Kac path integration, Wasilkowski and Woźniakowski (1996) considered path integration, Heinrich (1998) studied ML in the context of integral equations, Heinrich and Sindambiwe (1999) studied ML in the context of parametric integration, Hickernell and Wang (2002) studied QMC algorithms for infinite-dimensional integration. Giles (2008) considered ML MC schemes for stochastic partial differential equations.

Creutzig *et al.* (2009) introduced the variable subspace sampling model (which generalizes cost model A) and obtained optimality results for ML algorithms for this model, which apply in particular to infinite-dimensional integration of stochastic differential equations. Infinite-dimensional integration using randomized MC algorithms in a hierarchy of finite-dimensional subspaces was studied by Hickernell, Müller-Gronbach, Niu and Ritter (2010). Niu, Hickernell, Müller-Gronbach and Ritter (2011) studied

deterministic ML algorithms. Niu and Hickernell (2009) studied MC simulation of infinite-dimensional stochastic integrals arising in financial applications. The CD algorithm combined with QMC methods was introduced in Kuo *et al.* (2010*c*). The above results for ML and CD algorithms were improved by Gnewuch (2012*b*). Plaskota and Wasilkowski (2011) considered the CD algorithm combined with Smolyak algorithms and studied tractability in the worst-case and randomized setting. Higher-order convergence using higher-order polynomial lattice rules in infinite-dimensional anchored spaces was studied by Dick and Gnewuch (2013). Baldeaux (2012) and Baldeaux and Gnewuch (2013) studied randomly scrambled polynomial lattice rules for infinite-dimensional integration, the latter in unanchored Sobolev spaces. Lower bounds were considered by Gnewuch (2013). A probabilistic ML algorithm which yields an unbiased estimator was introduced by Rhee and Glynn (2012).

The mathematics of random fields is well described in the book by Adler (1981). For engineering applications of the MC method to problems of porous flow: see for instance Ghanem and Spanos (1991). The ML MC approach has recently been developed for elliptic problems with random input data in Barth, Schwab and Zollinger (2011), Charrier, Scheichl and Teckentrup (2011), Cliffe, Giles, Scheichl and Teckentrup (2011), Schwab and Gittelson (2011) and Teckentrup, Scheichl, Giles and Ullmann (2012). Application of QMC methods to these PDE problems has been considered in Graham *et al.* (2011), Kuo *et al.* (2012, 2013) and Graham *et al.* (2013).

8. Concluding remarks

In this survey we have tried to capture key concepts and methods in the rapidly developing field of high-dimensional numerical integration. We have concentrated on equal-weight rules, not because they will always be the best rules for a particular problem (indeed, in low dimensions we know that this is certainly not the case), but because they can be used in practice for very high-dimensional problems. We have not covered sparse grid methods, reviewed in a previous *Acta Numerica* article by Bungartz and Griebel (2004), which also provide a well-understood approach to integration in moderately high dimensions, and which are especially attractive for the related topic of approximation of functions.

In this rapidly developing subject it is almost certain that future research will give a different emphasis to the material covered in this survey, but we think it likely that topics such as reproducing kernels, weighted function spaces, component-by-component constructions, and discrepancy in all its forms, will continue to play starring roles in the science of high-dimensional numerical integration.

Acknowledgements

We gratefully acknowledge the the support of the Australian Research Council. J. Dick is supported by an Australian Research Council QEII Fellowship.

We have received valuable comments and corrections from Johann Brauchart, Michael Gnewuch, Mario Hefter, Hernan Leovey, Sebastian Mayer, Harald Niederreiter, Dirk Nuyens, Steffen Omland, Art Owen, Klaus Ritter, Grzegorz Wasilkowski, Markus Weimar, and Henryk Woźniakowski.

REFERENCES

N. Achtsis and D. Nuyens (2012), A component-by-component construction for the trigonometric degree. In *Monte Carlo and Quasi-Monte Carlo Methods 2010* (L. Plaskota and H. Woźniakowski, eds), Springer, pp. 235–253.

P. Acworth, M. Broadie and P. Glasserman (1998), A comparison of some Monte Carlo and quasi-Monte Carlo techniques for option pricing. In *Monte Carlo and Quasi-Monte Carlo Methods 1996* (P. Hellekalek, G. Larcher, H. Niederreiter and P. Zinterhof, eds), Springer, pp. 1–18.

R. J. Adler (1981), *The Geometry of Random Fields*, Wiley.

C. Aistleitner (2011), 'Covering numbers, dyadic chaining and discrepancy', *J. Complexity* **27**, 531–540.

C. Aistleitner and M. Hofer (2012), 'Probabilistic error bounds for the discrepancy of mixed sequences', *Monte Carlo Methods Appl.* **18**, 181–200.

I. A. Antonov and V. M. Saleev (1979), 'An effective method for the computation of λP_τ-sequences', *Ž. Vyčisl. Mat. i Mat. Fiz.* **19**, 243–245, 271.

N. Aronszajn (1950), 'Theory of reproducing kernels', *Trans. Amer. Math. Soc.* **68**, 337–404.

E. I. Atanassov (2004*a*), Efficient CPU-specific algorithm for generating the generalized Faure sequences. In *Large-Scale Scientific Computing*, Vol. 2907 of *Lecture Notes in Computer Science*, Springer, pp. 121–127.

E. I. Atanassov (2004*b*), 'On the discrepancy of Halton sequences', *Math. Balkanica* **18**, 15–32.

N. S. Bakhvalov (1959), 'On approximate computation of integrals' (in Russian), *Vestnik MGU, Ser. Math. Mech. Astron. Phys. Chem.* **4**, 3–18.

J. Baldeaux (2012), Scrambled polynomial lattice rules for infinite-dimensional integration. In *Monte Carlo and Quasi-Monte Carlo Methods 2010* (L. Plaskota and H. Woźniakowski, eds), Springer, pp. 255–263.

J. Baldeaux and J. Dick (2009), 'QMC rules of arbitrary high order: Reproducing kernel Hilbert space approach', *Constr. Approx.* **30**, 495–527.

J. Baldeaux and J. Dick (2011), 'A construction of polynomial lattice rules with small gain coefficients', *Numer. Math.* **119**, 271–297.

J. Baldeaux and M. Gnewuch (2013), Optimal randomized multilevel algorithms for infinite-dimensional integration on function spaces with ANOVA-type decomposition. Submitted.

J. Baldeaux, J. Dick, J. Greslehner and F. Pillichshammer (2011), 'Construction algorithms for higher order polynomial lattice rules', *J. Complexity* **27**, 281–299.

J. Baldeaux, J. Dick, G. Leobacher, D. Nuyens and F. Pillichshammer (2012), 'Efficient calculation of the worst-case error and (fast) component-by-component construction of higher order polynomial lattice rules', *Numer. Algorithms* **59**, 403–431.

A. Barth, C. Schwab and N. Zollinger (2011), 'Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients', *Numer. Math.* **119**, 123–161.

P. Bratley and B. L. Fox (1988), 'Algorithm 659: Implementing Sobol′s quasirandom sequence generator', *ACM Trans. Math. Softw.* **14**, 88–100.

P. Bratley, B. L. Fox, and H. Niederreiter (1992), 'Implementation and tests of low-discrepancy sequences', *ACM Trans. Model. Comput. Simul.* **2**, 195–213.

H. Bungartz and M. Griebel (2004), Sparse grids. In *Acta Numerica*, Vol. 13, Cambridge University Press, pp. 147–269.

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

J. Charrier, R. Scheichl and A. L. Teckentrup (2011), Finite element error analysis of elliptic PDEs with random coefficients and its application to multilevel Monte Carlo methods. Preprint 2/11, Bath Institute For Complex Systems.

B. Chazelle (2000), *The Discrepancy Method: Randomness and Complexity*, Cambridge University Press.

H. E. Chrestenson (1955), 'A class of generalized Walsh functions', *Pacific J. Math.* **5**, 17–31.

K. L. Chung (1974), *A Course in Probability Theory*, second edition, Vol. 21 of *Probability and Mathematical Statistics*, Academic Press.

K. A. Cliffe, M. B. Giles, R. Scheichl and A. L. Teckentrup (2011), 'Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients', *Comput. Vis. Sci.* **14**, 3–15.

R. Cools (1997), Constructing cubature formulae: The science behind the art. In *Acta Numerica*, Vol. 6, Cambridge University Press, pp. 1–54.

R. Cools and J. N. Lyness (2001), 'Three- and four-dimensional K-optimal lattice rules of moderate trigonometric degree', *Math. Comp.* **70**, 1549–1567.

R. Cools and D. Nuyens (2008), A Belgian view on lattice rules. In *Monte Carlo and Quasi-Monte Carlo Methods 2006* (A. Keller, S. Heinrich and H. Niederreiter, eds), Springer, pp. 3–21.

R. Cools, F. Y. Kuo and D. Nuyens (2006), 'Constructing embedded lattice rules for multivariate integration', *SIAM J. Sci. Comput.* **28**, 2162–2188.

R. Cools, F. Y. Kuo and D. Nuyens (2010), 'Constructing lattice rules based on weighted degree of exactness and worst case error', *Computing* **87**, 63–89.

J. Creutzig, S. Dereich, T. Müller-Gronbach and K. Ritter (2009), 'Infinite-dimensional quadrature and approximation of functions', *Found. Comp. Math.* **9**, 391–429.

P. J. Davis and P. Rabinowitz (1984), *Methods of Numerical Integration*, second edition, Academic.

L. Devroye (1986), *Nonuniform Random Variate Generation*, Springer.

J. Dick (2004), 'On the convergence rate of the component-by-component construction of good lattice rules', *J. Complexity* **20**, 493–522.

J. Dick (2007*a*), 'Explicit constructions of quasi-Monte Carlo rules for the numerical integration of high-dimensional periodic functions', *SIAM J. Numer. Anal.* **45**, 2141–2176.

J. Dick (2007*b*), 'The construction of extensible polynomial lattice rules with small weighted star discrepancy', *Math. Comp.* **76**, 2077–2085.

J. Dick (2008), 'Walsh spaces containing smooth functions and quasi-Monte Carlo rules of arbitrary high order', *SIAM J. Numer. Anal.* **46**, 1519–1553.

J. Dick (2009*a*), 'The decay of the Walsh coefficients of smooth functions', *Bull. Aust. Math. Soc.* **80**, 430–453.

J. Dick (2009*b*), On Quasi-Monte Carlo rules achieving higher order convergence. In *Monte Carlo and Quasi-Monte Carlo Methods 2008* (P. L'Ecuyer and A. B. Owen, eds), Springer, pp. 73–96.

J. Dick (2011*a*), 'Higher order scrambled digital nets achieve the optimal rate of the root mean square error for smooth integrands', *Ann. Statist.* **39**, 1372–1398.

J. Dick (2011*b*), 'Quasi-Monte Carlo numerical integration on $\mathbb{R}^s$: Digital nets and worst-case error', *SIAM J. Numer. Anal.* **49**, 1661–1691.

J. Dick (2012), 'Random weights, robust lattice rules and the geometry of the cbc*r*c algorithm', *Numer. Math.* **122**, 443–467.

J. Dick and J. Baldeaux (2009), Equidistribution properties of generalized nets and sequences. In *Monte Carlo and Quasi-Monte Carlo Methods 2008* (P. L'Ecuyer and A. B. Owen, eds), Springer, pp. 305–322.

J. Dick and M. Gnewuch (2013), Infinite-dimensional integration in weighted Hilbert spaces: Anchored decompositions, optimal deterministic algorithms, and higher order convergence. Submitted.

J. Dick and P. Kritzer (2010), 'Duality theory and propagation rules for generalized digital nets', *Math. Comp.* **79**, 993–1017.

J. Dick and F. Y. Kuo (2004*a*), 'Reducing the construction cost of the component-by-component construction of good lattice rules', *Math. Comp.* **73**, 1967–1988.

J. Dick and F. Y. Kuo (2004*b*), Constructing good lattice rules with millions of points. In *Monte Carlo and Quasi-Monte Carlo Methods 2002* (H. Niederreiter, ed.), Springer, pp. 181–197.

J. Dick and F. Pillichshammer (2005), 'Multivariate integration in weighted Hilbert spaces based on Walsh functions and weighted Sobolev spaces', *J. Complexity* **21**, 149–195.

J. Dick and F. Pillichshammer (2007), 'Strong tractability of multivariate integration of arbitrary high order using digitally shifted polynomial lattice rules', *J. Complexity* **23**, 436–453.

J. Dick and F. Pillichshammer (2010), *Digital Nets and Sequences*, Cambridge University Press.

J. Dick, P. Kritzer, G. Leobacher and F. Pillichshammer (2007*a*), 'Constructions of general polynomial lattice rules based on the weighted star discrepancy', *Finite Fields Appl.* **13**, 1045–1070.

J. Dick, P. Kritzer, F. Pillichshammer and W. C. Schmid (2007*b*), 'On the existence of higher order polynomial lattices based on a generalized figure of merit', *J. Complexity* **23**, 581–593.

J. Dick, F .Y. Kuo, F. Pillichshammer and I. H. Sloan (2005), 'Construction algorithms for polynomial lattice rules for multivariate integration', *Math. Comp.* **74**, 1895–1921.

J. Dick, G. Larcher, F. Pillichshammer and H. Woźniakowski (2011), 'Exponential convergence and tractability of multivariate integration for Korobov spaces', *Math. Comp.* **80**, 905–930.

J. Dick, D. Nuyens and F. Pillichshammer (2013), Lattice rules for nonperiodic smooth integrands. Submitted.

J. Dick, F. Pillichshammer, and B. J. Waterhouse (2008), 'The construction of good extensible rank-1 lattices', *Math. Comp.* **77**, 2345–2374.

J. Dick, I. H. Sloan, X. Wang and H. Woźniakowski (2006), 'Good lattice rules in weighted Korobov spaces with general weights', *Numer. Math.* **103**, 63–97.

Digital Library of Mathematical Functions, National Institute of Standards and Technology.

B. Doerr, M. Gnewuch, P. Kritzer and F. Pillichshammer (2008), 'Component-by-component construction of low-discrepancy point sets of small size', *Monte Carlo Methods Appl.* **14**, 129–149.

B. Doerr, M. Gnewuch and M. Wahlström (2009), Implementation of a component-by-component algorithm to generate small low-discrepancy samples. In *Monte Carlo and Quasi-Monte Carlo Methods 2008* (P. L'Ecuyer and A. B. Owen, eds), Springer, pp. 323–338.

B. Doerr, M. Gnewuch and M. Wahlström (2010), 'Algorithmic construction of low-discrepancy point sets via dependent randomized rounding', *J. Complexity* **26**, 490–507.

M. Drmota and R. F. Tichy (1997), *Sequences, Discrepancies and Applications*, Vol. 1651 of *Lecture Notes in Mathematics*, Springer.

K. T. Fang and Y. Wang (1994), *Number-Theoretic Methods in Statistics*, Chapman & Hall.

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

N. J. Fine (1949), 'On the Walsh functions', *Trans. Amer. Math. Soc.* **65**, 372–414.

B. L. Fox (1986), 'Algorithm 647: Implementation and relative efficiency of quasirandom sequence generators', *ACM Trans. Math. Softw.* **12**, 362–376.

R. G. Ghanem and P. D. Spanos (1991), *Stochastic Finite Elements: A Spectral Approach*, Springer.

M. B. Giles (2008), 'Multilevel Monte Carlo path simulation', *Oper. Res.* **56**, 607–617.

M. Giles, F. Y. Kuo, I. H. Sloan and B. J. Waterhouse (2008), 'Quasi-Monte Carlo for finance applications', *ANZIAM J.* **50** (CTAC2008), C308–C323.

M. Gnewuch (2008), 'Bracketing numbers for axis-parallel boxes and applications to geometric discrepancy', *J. Complexity* **24**, 154–172.

M. Gnewuch (2009), 'On probabilistic results for the discrepancy of a hybrid-Monte Carlo sequence', *J. Complexity* **23**, 312–317.

M. Gnewuch (2012*a*), 'Weighted geometric discrepancies and numerical integration on reproducing kernel Hilbert spaces', *J. Complexity* **28**, 2–17.

M. Gnewuch (2012*b*), 'Infinite-dimensional integration on weighted Hilbert spaces', *Math. Comp.* **280**, 2175–2205.

M. Gnewuch (2013), Lower error bounds for randomized multilevel and changing dimension algorithms. In *Monte Carlo and Quasi-Monte Carlo Methods 2012* (J. Dick, F. Y. Kuo, G. W. Peters and I. H. Sloan, eds), Springer, to appear.

M. Gnewuch and A. V. Roşca (2009), 'On G-discrepancy and mixed Monte Carlo and quasi-Monte Carlo sequences', *Acta Univ. Apulensis Math. Inform.* **18**, 97–110.

M. Gnewuch, S. Mayer and K. Ritter (2013), On weighted Hilbert spaces and integration of functions of infinitely many variables. Submitted.

M. Gnewuch, A. Srivastav and C. Winzen (2009), 'Finding optimal volume subintervals with k-points and calculating the star discrepancy are NP-hard problems', *J. Complexity* **25**, 115–127.

M. Gnewuch, M. Wahlström and C. Winzen (2012), 'A new randomized algorithm to approximate the star discrepancy based on threshold accepting', *SIAM J. Numer. Anal.* **50**, 781–807.

T. Goda and J. Dick (2013), Construction of interlaced scrambled polynomial lattice rules of arbitrary high order. Submitted.

I. G. Graham, F. Y. Kuo, J. Nichols, R. Scheichl, C. Schwab and I. H. Sloan (2013), Quasi-Monte Carlo finite element methods for elliptic PDEs with log-normal random coefficients. In preparation.

I. G. Graham, F. Y. Kuo, D. Nuyens, R. Scheichl and I. H. Sloan (2011), 'Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications', *J. Comput. Phys.* **230**, 3668–3694.

M. Griebel (2006), Sparse grids and related approximation schemes for higher dimensional problems. In *Foundations of Computational Mathematics, Santander, 2005*, Cambridge University Press, pp. 106–161.

M. Griebel, F. Y. Kuo and I. H. Sloan (2013), 'The smoothing effect of integration in $\mathbb{R}^d$ and the ANOVA decomposition', *Math. Comp.* **82**, 383–400.

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

J. M. Hammersley and D. C. Handscomb (1964), *Monte Carlo Methods*, Methuen.

T. Hansen, G. L. Mullen and H. Niederreiter (1993), 'Good parameters for a class of node sets in quasi-Monte Carlo integration', *Math. Comp.* **61**, 225–234.

S. Heinrich (1998), 'Monte Carlo complexity of global solution of integral equations', *J. Complexity* **14**, 151–175.

S. Heinrich and E. Sindambiwe (1999), 'Monte Carlo complexity of parametric integration', *J. Complexity* **15**, 317–341.

S. Heinrich, F. Hickernell and R. X. Yue (2004), 'Optimal quadrature for Haar wavelet spaces', *Math. Comp.* **73**, 259–277.

S. Heinrich, E. Novak, G. W. Wasilkowski, H. Woźniakowski (2001), 'The inverse of the star-discrepancy depends linearly on the dimension', *Acta Arith.* **96**, 279–302.

F. J. Hickernell (1996*a*), 'Quadrature error bounds with applications to lattice rules', *SIAM J. Numer. Anal.* **33**, 1995–2016. Erratum: 'Quadrature error bounds with applications to lattice rules', *SIAM J. Numer. Anal.* **34**, 853–866 (1997).

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

F. J. Hickernell (1998*a*), 'A generalized discrepancy and quadrature error bound', *Math. Comp.* **67**, 299–322.
F. J. Hickernell (1998*b*), Lattice rules: How well do they measure up? In *Random and Quasi-Random Point Sets* (P. Hellekalek and G. Larcher, eds), Springer, pp. 109–166.
F. J. Hickernell (1999), 'Goodness-of-fit statistics, discrepancies and robust designs', *Statist. Probab. Lett.* **44**, 73–78.
F. J. Hickernell (2002), Obtaining $O(N^{-2+\epsilon})$ convergence for lattice quadrature rules. In *Monte Carlo and Quasi-Monte Carlo Methods 2000* (K. T. Fang, F. J. Hickernell and H. Niederreiter, eds), Springer, pp. 274–289.
F. J. Hickernell and H. S. Hong (2002), Quasi-Monte Carlo methods and their randomisations. In *Applied Probability: AMS/IP Studies in Advanced Mathematics*, Vol. 26 (R. Chan, Y.-K. Kwok, D. Yao and Q. Zhang, eds), AMS, pp. 59–77.
F. J. Hickernell and H. Niederreiter (2003), 'The existence of good extensible rank-1 lattices', *J. Complexity* **19**, 286–300.
F. J. Hickernell and X. Wang (2002), 'The error bounds and tractability of quasi-Monte Carlo algorithms in infinite dimension', *Math. Comp.* **71**, 1641–1661.
F. J. Hickernell and H. Woźniakowski (2000), 'Integration and approximation in arbitrary dimensions', *Adv. Comput. Math.* **12**, 25–58.
F. J. Hickernell and H. Woźniakowski (2001), 'Tractability of multivariate integration for periodic functions', *J. Complexity* **17**, 660–682.
F. J. Hickernell and R. X. Yue (2000), 'The mean square discrepancy of scrambled (t, s)-sequences', *SIAM J. Numer. Anal.* **38**, 1089–1112.
F. J. Hickernell, H. S. Hong, P. L'Ecuyer, and C. Lemieux (2000), 'Extensible lattice sequences for quasi-Monte Carlo quadrature', *SIAM J. Sci. Comput.* **22**, 1117–1138.
F. J. Hickernell, P. Kritzer, F. Y. Kuo and D. Nuyens (2012), 'Weighted compound integration rules with higher order convergence for all N', *Numer. Algorithms* **59**, 161–183.
F. J. Hickernell, C. Lemieux and A. B. Owen (2005), 'Control variates for quasi-Monte Carlo', with comments by Pierre L'Ecuyer and Xiao-Li Meng and a rejoinder by the authors. *Statist. Sci.* **20**, 1–31.
F. Hickernell, T. Müller-Gronbach, B. Niu and K. Ritter (2010), 'Monte Carlo algorithms for infinite-dimensional integration on $\mathbb{R}^{\mathbb{N}}$', *J. Complexity* **26**, 229–254.
A. Hinrichs (2004), 'Covering numbers, Vapnik–Červonenkis classes and bounds for the star-discrepancy', *J. Complexity* **20**, 477–483.
A. Hinrichs, F. Pillichshammer and W. C. Schmid (2008), 'Tractability properties of the weighted star discrepancy', *J. Complexity* **24**, 134–143.
E. Hlawka (1961), 'Funktionen von beschränkter Variation in der Theorie der Gleichverteilung', *Ann. Mat. Pura Appl.* **54**, 325–333.
E. Hlawka (1962), 'Zur angenäherten Berechnung mehrfacher Integrale', *Monatsh. Math.* **66**, 140–151.
R. Hofer and P. Kritzer (2011), 'On hybrid sequences built from Niederreiter–Halton sequences and Kronecker sequences', *Bull. Aust. Math. Soc.* **84**, 238–254.

R. Hofer and G. Larcher (2010), 'On existence and discrepancy of certain digital Niederreiter–Halton sequences', *Acta Arith.* **141**, 369–394.
R. Hofer and G. Larcher (2012), 'Metrical results on the discrepancy of Halton–Kronecker sequences', *Math. Z.* **271**, 1–11.
R. Hofer and H. Niederreiter (2013), 'A construction of (t, s)-sequences with finite-row generating matrices using global function fields', *Finite Fields Appl.* **21**, 97–110.
R. Hofer and G. Pirsic (2011), 'An explicit construction of finite-row digital $(0, s)$-sequences', *Unif. Distrib. Theory* **6**, 13–30.
R. Hofer, P. Kritzer, G. Larcher and F. Pillichshammer (2009), 'Distribution properties of generalized van der Corput–Halton sequences and their subsequences', *Int. J. Number Theory* **5**, 719–746.
W. Hörmann, J. Leydold and G. Derflinger (2010), *Automatic Nonuniform Random Variate Generation*, Springer.
S. Joe (2004), Component by component construction of rank-1 lattice rules having $O(n^{-1}(\ln(n))^d)$ star discrepancy. In *Monte Carlo and Quasi-Monte Carlo Methods 2002* (H. Niederreiter, ed.), Springer, pp. 293–298.
S. Joe (2006), Construction of good rank-1 lattice rules based on the weighted star discrepancy. In *Monte Carlo and Quasi-Monte Carlo Methods 2004* (H. Niederreiter and D. Talay, eds), Springer, pp. 181–196.
S. Joe and F. Y. Kuo (2008), 'Constructing Sobol′ sequences with better two-dimensional projection', *SIAM J. Sci. Comput.* **30**, 2635–2654.
L. Kämmerer, S. Kunis and D. Potts (2012), 'Interpolation lattices for hyperbolic cross trigonometric polynomials', *J. Complexity* **28**, 76–92.
A. Keller (2006), Myths of computer graphics. In *Monte Carlo and Quasi-Monte Carlo Methods 2004* (H. Niederreiter and D. Talay, eds), Springer, pp. 217–243.
A. Keller (2013), Quasi-Monte Carlo image synthesis in a nutshell. In *Monte Carlo and Quasi-Monte Carlo Methods 2012* (J. Dick, F. Y. Kuo, G. W. Peters and I. H. Sloan, eds), Springer, to appear.
J. F. Koksma (1942/43), 'Een algemeene stelling uit de theorie der gelijkmatige verdeeling modulo 1', *Mathematica B* (Zutphen) **11**, 7–11.
N. M. Korobov (1959), 'The approximate computation of multiple integrals' (in Russian), *Dokl. Akad. Nauk SSSR* **124**, 1207–1210.
N. M. Korobov (1963), *Number-Theoretic Methods in Approximate Analysis*, Fizmatgiz.
N. M. Korobov (1982), 'On the calculation of optimal coefficients' (in Russian), *Dokl. Akad. Nauk SSSR* **267**, 289–292.
P. Kritzer and F. Pillichshammer (2007), 'Constructions of general polynomial lattices for multivariate integration', *Bull. Austral. Math. Soc.* **76**, 93–110.
L. Kuipers and H. Niederreiter (1974), *Uniform Distribution of Sequences*, Wiley.
F. Y. Kuo (2003), 'Component-by-component constructions achieve the optimal rate of convergence for multivariate integration in weighted Korobov and Sobolev spaces', *J. Complexity* **19**, 301–320.
F. Y. Kuo and S. Joe (2002), 'Component-by-component construction of good lattice rules with a composite number of points', *J. Complexity* **18**, 943–976.

F. Y. Kuo and S. Joe (2003), 'Component-by-component construction of good intermediate-rank lattice rules', *SIAM J. Numer. Anal.* **41**, 1465–1486.

F. Y. Kuo and I. H. Sloan (2005), 'Lifting the curse of dimensionality', *Not. Amer. Math. Soc.* **52**, 1320–1328.

F. Y. Kuo, W. T. M. Dunsmuir, I. H. Sloan, M. P. Wand, R. S. Womersley (2008*a*), 'Quasi-Monte Carlo for highly structured generalised response models', *Method. Comput. Appl. Probab.* **10**, 239–275.

K. Y. Kuo, C. Schwab and I. H. Sloan (2011), 'Quasi-Monte Carlo methods for high-dimensional integration: The standard (weighted Hilbert space) setting and beyond', *ANZIAM J.* **53**, 1–37.

F. Y. Kuo, C. Schwab and I. H. Sloan (2012), Quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients, *SIAM J. Numer. Anal.* **50**, 3351–3374.

F. Y. Kuo, C. Schwab and I. H. Sloan (2013), Multi-level quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients. Submitted.

F. Y. Kuo, I. H. Sloan, G. W. Wasilkowski, and B. J. Waterhouse (2010*a*), 'Randomly shifted lattice rules with the optimal rate of convergence for unbounded integrands', *J. Complexity* **26**, 135–160.

F. Y. Kuo, I. H. Sloan, G. W. Wasilkowski and H. Woźniakowski (2010*b*), 'On decompositions of multivariate functions', *Math. Comp.* **79**, 953–966.

F. Y. Kuo, I. H. Sloan, G. W. Wasilkowski and H. Woźniakowski (2010*c*), 'Liberating the dimension', *J. Complexity* **26**, 422–454.

F. Y. Kuo, I. H. Sloan, and H. Woźniakowski (2006*a*), Lattice rules for multivariate approximation in the worst case setting. In *Monte Carlo and Quasi-Monte Carlo Methods 2004* (H. Niederreiter and D. Talay, eds), Springer, pp. 289–330.

F. Y. Kuo, I. H. Sloan and H. Woźniakowski (2007), 'Periodization strategy may fail in high dimensions', *Numer. Algorithms* **46**, 369–391.

F. Y. Kuo, I. H. Sloan, and H. Woźniakowski (2008*b*), 'Lattice rule algorithms for multivariate approximation in the average case setting', *J. Complexity* **24**, 283–323.

F. Y. Kuo, G. W. Wasilkowski, and B. J. Waterhouse (2006*b*), 'Randomly shifted lattice rules for unbounded integrals', *J. Complexity* **22**, 630–651.

F. Y. Kuo, G. W. Wasilkowski, and H. Woźniakowski (2009), 'Lattice algorithms for multivariate L_∞ approximation in the worst case setting', *Constr. Approx.* **30**, 475–493.

G. Larcher and C. Traunfellner (1994), 'On the numerical integration of Walsh series by number-theoretic methods', *Math. Comp.* **63**, 277–291.

G. Larcher, A. Lauß, H. Niederreiter and W. C. Schmid (1996*a*), 'Optimal polynomials for (t, m, s)-nets and numerical integration of multivariate Walsh series', *SIAM J. Numer. Anal.* **33**, 2239–2253.

G. Larcher, W. C. Schmid and R. Wolf (1994), 'Representation of functions as Walsh series to different bases and an application to the numerical integration of high-dimensional Walsh series', *Math. Comp.* **63**, 701–716.

G. Larcher, W. C. Schmid and R. Wolf (1996*b*), 'Quasi-Monte Carlo methods for the numerical integration of multivariate Walsh series: Monte Carlo and quasi-Monte Carlo methods', *Math. Comput. Modelling* **23**, 55–67.

D. Laurie (1996), 'Periodizing transformations for numerical integration', *J. Comput. Appl. Math.* **66**, 337–344.

P. L'Ecuyer (2004), Quasi-Monte Carlo methods in finance. In *Proc. 2004 Winter Simulation Conference* (R. G. Ingalls, M. D. Rossetti, J. S. Smith and B. A. Peters, eds), IEEE Computer Society Press, pp. 1645–1655.

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

P. L'Ecuyer and D. Munger (2012), On figures of merit for randomly shifted lattice rules. In *Monte Carlo and Quasi-Monte Carlo Methods 2010* (L. Plaskota and H. Woźniakowski, eds), Springer, pp. 133–159.

P. L'Ecuyer, D. Munger and B. Tuffin (2010), 'On the distribution of integration error by randomly-shifted lattice rules'. *Electron. J. Stat.* **4**, 950–993.

C. Lemieux (2009), *Monte Carlo and Quasi-Monte Carlo Sampling*, Springer.

C. Lemieux and P. L'Ecuyer (2001), 'On selection criteria for lattice rules and other quasi-Monte Carlo point sets', *Math. Comput. Simulation* **55**, 139–148.

D. Li and F. J. Hickernell (2003), Trigonometric spectral collocation methods on lattices. In *Recent Advances in Scientific Computing and Partial Differential Equations* (S. Y. Cheng, C.-W. Shu and T. Tang, eds), Vol. 330 of *AMS Series in Contemporary Mathematics*, AMS, pp. 121–132.

G. Li, J. Schoendorf, T.-S, Ho and H. Rabitz (2004), 'Multicut-HDMR with an application to an ionospheric model', *J. Comput. Chem.* **25**, 1149–1156.

W. L. Loh (2003), 'On the asymptotic distribution of scrambled net quadrature', *Ann. Statist.* **31**, 1282–1324.

J. Lyness and T. Sørevik (2006), 'Five-dimensional k-optimal lattice rules', *Math. Comp.* **75**, 1467–1480.

E. Maize (1980), Contributions to the theory of error reduction in quasi-Monte Carlo methods. PhD thesis, Claremont Graduate School.

J. Matoušek (1998*a*), 'The exponent of discrepancy is at least 1.0669', *J. Complexity* **14**, 448–453.

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

J. Matoušek (1999), *Geometric Discrepancy: An Illustrated Guide*, Vol. 18 of *Algorithms and Combinatorics*, Springer.

M. Matsumoto and T. Yoshiki (2013), Existence of higher order convergent quasi-Monte Carlo rules via Walsh figure of merit. In *Monte Carlo and Quasi-Monte Carlo Methods 2012* (J. Dick, F. Y. Kuo, G. W. Peters and I. H. Sloan, eds), Springer, to appear.

M. Matsumoto, M. Saito, K. Matoba (2013), 'A computable figure of merit for quasi-Monte Carlo point sets', *Math. Comp.*, to appear.

H. Niederreiter (1978), 'Quasi-Monte Carlo methods and pseudo-random numbers', *Bull. Amer. Math. Soc.* **84**, 957–1041.

H. Niederreiter (1987), 'Point sets and sequences with small discrepancy', *Monatsh. Math.* **104**, 273–337.

H. Niederreiter (1988), 'Low-discrepancy and low-dispersion sequences', *J. Number Theory* **30**, 51–70.

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

H. Niederreiter (1992*b*), 'Low-discrepancy point sets obtained by digital constructions over finite fields', *Czechoslovak Math. J.* **42**, 143–166.

H. Niederreiter (2003), 'The existence of good extensible polynomial lattice rules', *Monatsh. Math.* **139**, 295–307.

H. Niederreiter (2004), Digital nets and coding theory. In *Coding, Cryptography and Combinatorics* (K. Q. Feng, H. Niederreiter and C. P. Xing, eds), Birkhäuser, pp. 247–257.

H. Niederreiter (2009), 'On the discrepancy of some hybrid sequences', *Acta Arith.* **138**, 373–398.

H. Niederreiter (2010*a*), 'A discrepancy bound for hybrid sequences involving digital explicit inversive pseudorandom numbers', *Unif. Distrib. Theory* **5**, 53–63.

H. Niederreiter (2010*b*), 'Further discrepancy bounds and an Erdős–Turán–Koksma inequality for hybrid sequences', *Monatsh. Math.* **161**, 193–222.

H. Niederreiter (2012), 'Improved discrepancy bounds for hybrid sequences involving Halton sequences', *Acta Arith.* **155**, 71–84.

H. Niederreiter and F. Pillichshammer (2009), 'Construction algorithms for good extensible lattice rules, *Constr. Approx.* **30**, 361–393.

H. Niederreiter and A. Winterhof (2011), 'Discrepancy bounds for hybrid sequences involving digital explicit inversive pseudorandom numbers', *Unif. Distrib. Theory* **6**, 33–56.

H. Niederreiter and C. P. Xing (1995), 'Low-discrepancy sequences obtained from algebraic function fields over finite fields', *Acta Arith.* **72**, 281–298.

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

H. Niederreiter and C. P. Xing (1996*b*), Quasirandom points and global function fields. In *Finite Fields and Applications* Vol. 233 of *London Mathematical Society Lecture Note Series*, Cambridge University Press, pp. 269–296.

B. Niu and F. J. Hickernell (2009), Monte Carlo simulation of stochastic integrals when the cost of function evaluation is dimension dependent. In *Monte Carlo and Quasi-Monte Carlo Methods 2008* (P. L'Ecuyer and A. B. Owen, eds), Springer, pp. 545–560.

B. Niu, F. J. Hickernell, T. Müller-Gronbach and K. Ritter (2011), 'Deterministic multi-level algorithms for infinite-dimensional integration on $\mathbb{R}^{\mathbb{N}}$', *J. Complexity* **27**, 331–351.

E. Novak and H. Woźniakowski (2001), 'Intractability results for integration and discrepancy', *J. Complexity*, **17**, 388–441.

E. Novak and H. Woźniakowski (2008), *Tractability of Multivariate Problems, Vol. I: Linear Information*, EMS.

E. Novak and H. Woźniakowski (2009), L_2 discrepancy and multivariate integration. In *Analytic Number Theory: Essays in Honour of Klaus Roth* (W. W. L. Chen, W. T. Gowers, H. Halberstam, W. M. Schmidt and R. C. Vaughan, eds), Cambridge University Press, pp. 359–388.

E. Novak and H. Woźniakowski (2010), *Tractability of Multivariate Problems, Vol. II: Standard Information for Functionals*, EMS.

E. Novak and H. Woźniakowski (2012), *Tractability of Multivariate Problems, Vol. III: Standard Information for Operators*, EMS.

D. Nuyens and R. Cools (2006*a*), 'Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel Hilbert spaces', *Math. Comp.* **75**, 903–920.

D. Nuyens and R. Cools (2006*b*), 'Fast component-by-component construction of rank-1 lattice rules with a non-prime number of points', *J. Complexity* **22**, 4–28.

D. Nuyens and R. Cools (2006*c*), Fast component-by-component construction, a reprise for different kernels. In *Monte Carlo and Quasi-Monte Carlo Methods 2004* (H. Niederreiter and D. Talay, eds), Springer, pp. 373–387.

G. Ökten (1996), 'A probabilistic result on the discrepancy of a hybrid Monte Carlo sequence and applications', *Monte Carlo Methods Appl.* **2**, 255–270.

G. Ökten, B. Tuffin and V. Burago (2006), 'A central limit theorem and improved error bounds for a hybrid-Monte Carlo sequence with applications in computational finance', *J. Complexity* **22**, 435–458.

A. B. Owen (1995), Randomly permuted (t, m, s)-nets and (t, s)-sequences. In *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing* (Las Vegas, NV, 1994), Vol. 106 of *Lecture Notes in Statistics*, Springer, pp. 299–317.

A. B. Owen (1997*a*), 'Scrambled net variance for integrals of smooth functions', *Ann. Statist.* **25**, 1541–1562.

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

A. B. Owen (1998), 'Scrambling Sobol′ and Niederreiter–Xing points', *J. Complexity* **14**, 466–489.

A. B. Owen (2006), Quasi-Monte Carlo for integrands with point singularities at unknown locations. In *Monte Carlo and Quasi-Monte Carlo Methods 2004* (H. Niederreiter and D. Talay, eds), Springer, pp. 403–417.

S. H. Paskov and J. Traub (1995), 'Faster evaluation of financial derivatives', *J. Portfolio Management* **22**, 113–120.

F. Pillichshammer (2002), 'Bounds for the quality parameter of digital shift nets over Z_2', *Finite Fields Appl.* **8**, 444–454.

F. Pillichshammer and G. Pirsic (2009), Discrepancy of hyperplane nets and cyclic nets. In *Monte Carlo and Quasi-Monte Carlo Methods 2008* (P. L'Ecuyer and A. B. Owen, eds), Springer, pp. 573–587.

G. Pirsic (2002), A software implementation of Niederreiter–Xing sequences. In *Monte Carlo and Quasi-Monte Carlo Methods 2000* (K. T. Fang, F. J. Hickernell and H. Niederreiter, eds), Springer, pp. 434–445.

G. Pirsic and F. Pillichshammer (2011), 'Extensible hyperplane nets', *Finite Fields Appl.* **17**, 407–423.

G. Pirsic and W. C. Schmid (2001), 'Calculation of the quality parameter of digital nets and application to their construction', *J. Complexity* **17**, 827–839.

G. Pirsic, J. Dick and F. Pillichshammer (2006), 'Cyclic digital nets, hyperplane nets, and multivariate integration in Sobolev spaces', *SIAM J. Numer. Anal.* **44**, 385–411.

L. Plaskota and G. Wasilkowski (2011), 'Tractability of infinite-dimensional integration in the worst case and randomized setting', *J. Complexity* **27**, 505–518.

L. Plaskota, G. Wasilkowski and H. Woźniakowski (2000), 'A new algorithm and worst case complexity for Feynman–Kac path integration', *J. Comput. Phys.* **164**, 335–353.

C. M. Rader (1968), 'Discrete Fourier transforms when the number of data samples is prime', *Proc. IEEE* **5**, 1107–1108.

C.-H. Rhee and P. W. Glynn (2012), A new approach to unbiased estimation for SDE's. In *Proc. 2012 Winter Simulation Conference* (C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose and A. M. Uhrmacher, eds).

W. C. Schmid (1996), Shift-nets: A new class of binary digital (t, m, s)-nets. In *Monte Carlo and Quasi-Monte Carlo Methods 1996* (H. Niederreiter, P. Hellekalek, G. Larcher and P. Zinterhof, eds), Vol. 127 of *Lecture Notes in Statistics*, Springer, pp. 369–381.

W. C. Schmid (2000), Improvements and extensions of the 'Salzburg Tables' by using irreducible polynomials. In *Monte Carlo and Quasi-Monte Carlo Methods 1998* (H. Niederreiter and J. Spanier, eds), Springer, pp. 436–447.

C. Schwab and C. J. Gittelson (2011), Sparse tensor discretizations of high-dimensional parametric and stochastic PDEs. In *Acta Numerica*, Vol. 20, Cambridge University Press, pp. 291–467.

A. Sidi (1993), A new variable transformation for numerical integration. In *Numerical Integration IV: Oberwolfach, 1992* (H. Brass and G. Hämmerlin, eds), Birkhäuser, pp. 359–373.

V. Sinescu and S. Joe (2007), 'Good lattice rules based on the general weighted star discrepancy', *Math. Comp.* **76**, 989–1004.

V. Sinescu and S. Joe (2008), Good lattice rules with a composite number of points based on the product weighted star discrepancy. In *Monte Carlo and Quasi-Monte Carlo Methods 2006* (A. Keller, S. Heinrich and H. Niederreiter, eds), Springer, pp. 645–658.

V. Sinescu and P. L'Ecuyer (2011), 'Existence and construction of shifted lattice rules with an arbitrary number of points and bounded weighted star discrepancy for general decreasing weights', *J. Complexity* **27**, 449–465.

I. H. Sloan (2007), 'Finite order integration weights can be dangerous', *Comput. Meth. Appl. Math.* **7**, 239–254.

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

I. H. Sloan and A, V. Reztsov (2002), 'Component-by-component construction of good lattice rules', *Math. Comp.* **71**, 263–273.

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

I. H. Sloan and H. Woźniakowski (2001), 'Tractability of multivariate integration for weighted Korobov classes', *J. Complexity* **17**, 697–721.

I. H. Sloan and H. Woźniakowski (2002), 'Tractability of integration in non-periodic and periodic weighted tensor product Hilbert spaces', *J. Complexity* **18**, 479–499.

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

I. H. Sloan, F. Y. Kuo and S. Joe (2002*b*) 'Constructing randomly shifted lattice rules in weighted Sobolev spaces', *SIAM J. Numer. Anal.* **40**, 1650–1665.

I. H. Sloan, X. Wang and H. Woźniakowski (2004), 'Finite-order weights imply tractability of multivariate integration', *J. Complexity* **20**, 46–74.

I. M. Sobol′ (1967), 'Distribution of points in a cube and approximate evaluation of integrals' (in Russian), *Ž. Vyčisl. Mat. i Mat. Fiz.* **7**, 784–802.

I. M. Sobol′ (1969), *Multidimensional Quadrature Formulas and Haar Functions* (in Russian), Nauka.

S. Smolyak (1963), 'Quadrature and interpolation formulas for tensor products of certain classes of functions', *Soviet Math. Dokl.* **4**, 240–243. Russian original in *Dokl. Akad. Nauk SSSR* **148** (1963), 1042–1045.

A. H. Stroud (1971), *Approximate Calculation of Multiple Integrals*, Prentice-Hall.

A. L. Teckentrup, R. Scheichl, M. B. Giles and E. Ullmann (2012), Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficient. Technical Report, University of Bath.

S. Tezuka (2013), 'On the discrepancy of generalized Niederreiter sequences, *J. Complexity*, to appear.

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

C. Thomas-Agnan (1996), 'Computing a family of reproducing kernels for statistical applications', *Numer. Algorithms* **13**, 21–32.

J. F. Traub, G. W. Wasilkowski and H. Woźniakowski (2008), *Information-Based Complexity*, Academic Press.

A. W. van der Vart and J. A. Wellner (2009), *Weak Convergence and Empirical Processes*, Springer Series in Statistics, Springer.

G. Wahba (1990), *Spline Models for Observational Data*, Vol. 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*, SIAM.

J. L. Walsh (1923), 'A closed set of normal orthogonal functions', *Amer. J. Math.* **45**, 5–24.

X. Wang (2002), 'A constructive approach to strong tractability using quasi-Monte Carlo algorithms', *J. Complexity* **18**, 683–701.

X. Wang (2003), 'Strong tractability of multivariate integration using quasi-Monte Carlo algorithms', *Math. Comp.* **72**, 823–838.

X. Wang and K.-T. Fang (2003), 'Effective dimension and quasi-Monte Carlo integration', *J. Complexity* **19**, 101–124.

X. Wang and I. H. Sloan (2005), 'Why are high-dimensional finance problems often of low effective dimension?', *SIAM J. Sci. Comput.* **27**, 159–183.

X. Wang and I. H. Sloan (2006), 'Efficient weighted lattice rules with applications to finance', *SIAM J. Sci. Comput.* **28**, 728–750.

X. Wang and I. H. Sloan (2007), 'Brownian bridge and principal component analysis: Towards removing the curse of dimensionality', *IMA J. Numer. Anal.* **27**, 631–654.

X. Wang and I. H. Sloan (2011), 'Quasi-Monte Carlo methods in financial engineering: An equivalence principle and dimension reduction', *Oper. Res.* **59**, 80–95.

G. W. Wasilkowski and H. Woźniakowski (1995), 'Explicit cost bounds of algorithms for multivariate tensor product problems', *J. Complexity* **11**, 1–56.

G. W. Wasilkowski and H. Woźniakowski (1996), 'On tractability of path integration', *J. Math.Phys.* **37**, 2071–2088.

G. W. Wasilkowski and H. Woźniakowski (1999), 'Weighted tensor product algorithms for linear multivariate problems', *J. Complexity* **15**, 402–447.

G. W. Wasilkowski and H. Woźniakowski (2004), 'Finite-order weights imply tractability of linear multivariate problems', *J. Approx. Theory* **130**, 57–77.

G. W. Wasilkowski and H. Woźniakowski (2010), 'On the exponent of discrepancy', *Math. Comp.* **79**, 983–992.

A. Werschulz and H. Woźniakowski (2009), 'Tractability of multivariate approximation over a weighted unanchored Sobolev space', *Constr. Approx.* **30**, 395–421.

H. Weyl (1916), 'Über die Gleichverteilung von Zahlen mod. Eins', *Math. Ann.* **77**, 313–352.

H. Woźniakowski (2013), Monte Carlo integration. In *Encyclopedia of Numerical Analysis*.

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

R. X. Yue and F. J. Hickernell (2001), 'Integration and approximation based on scramble sampling in arbitrary dimensions', *J. Complexity* **17**, 881–897.

R. X. Yue and F. J. Hickernell (2002), 'The discrepancy and gain coefficients of scrambled digital nets', *J. Complexity* **18**, 135–151.

R. X. Yue and F. J. Hickernell (2005), 'Strong tractability of integration using scrambled Niederreiter points', *Math. Comp.* **74**, 1871–1893.

R. X. Yue and F. J. Hickernell (2006), 'Strong tractability of quasi-Monte Carlo quadrature using nets for certain Banach spaces', *SIAM J. Numer. Anal.* **44**, 2559–2583.

X. Y. Zeng, K. T. Leung, and F. J. Hickernell (2006), Error analysis of splines for periodic problems using lattice designs. In *Monte Carlo and Quasi-Monte Carlo Methods 2004* (H. Niederreiter and D. Talay, eds), Springer, pp. 501–514.

C. Zenger (1991), Sparse grids. In *Parallel Algorithms for Partial Differential Equations* (W. Hackbusch, ed.), Vol. 31 of *Notes on Numerical Fluid Mechanics*, Vieweg.