Monte Carlo Book: the Quasi-Monte Carlo parts

Art B. Owen
Stanford University

July 2019

Contents

15

Quasi-Monte Carlo

Monte Carlo computation usually begins with points sampled from a uniform distribution on the unit cube transformed as needed to other spaces and different distributions on those spaces. Those uniform points $\boldsymbol{x}_i$ tend to form clumps in some parts of $[0,1]^d$ and leave voids in others. Whether any given region of the unit cube gets a clump or a void is of course random. The idea in ***quasi-Monte Carlo*** (QMC) sampling is to choose points that, to the extent possible, are spread out uniformly through $[0,1]^d$ with minimal clumps and voids. We still estimate $\mu = \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ by

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i), \tag{15.1}$$

but now $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are deterministic points designed to fill $[0,1]^d$ as evenly as mathematically possible, while f incorporates our transformations as well as our original integrand. Roughly speaking, QMC is stratification taken to extremes. For QMC it is easy to use transformations like inversion and not so simple to use acceptance-rejection because the necessary d is not fixed.

The best use case for QMC arises when the integrand f has a high enough dimension d that classic quadratures are infeasible, yet f is itself well approximated by a sum of functions of one or two or a handful of its inputs. When QMC works well on high dimensional functions, it can be a surprise. A famous surprise found empirically by Paskov and Traub (1995) was integrands from finance with d in the hundreds could be well integrated by QMC. QMC is also used in computer graphics (Keller, 1997) and in solving partial differential equations over random environments (Graham et al., 2015).

In this chapter we look at how to measure the uniformity of a set of points. Such measures, of which there are many, are called discrepancies. Then we compare QMC to MC, by considering the counterparts in QMC to the LLN and CLT from MC. A key result, the Koksma-Hlawka theorem, shows how bounds on discrepancy can be turned into bounds on quadrature error. It is possible to achieve discrepancies that are $O(n^{-1+\epsilon})$ for any $\epsilon > 0$, and we will see conditions under which $|\hat{\mu} - \mu| = O(n^{-1+\epsilon})$ too. This, roughly speaking, is as good as Monte Carlo would be with on the order of n^2 function evaluations.

15.1 Introduction to QMC

To begin, it is important to point out one commonly overlooked difference between MC and QMC. With QMC, there are usually significant benefits to using certain special values of n, such as powers of 2, large primes, and more generally, powers of primes. Powers of ten are almost never especially good choices. Using $n = 100{,}000$ could be much worse than using $2^{17} = 131{,}072$ and could even be worse than using $2^{16} = 65{,}536$. This distinction does not show up in the asymptotic error rates for QMC. Those are usually given as some power of n often with a power of $\log(n)$ and they hide the importance of special sample sizes. Some QMC methods have errors that are $o(1/n)$ for special n. Then adding an $n+1$'st point changes $\hat{\mu}$ by $O(1/n)$ which is of larger magnitude than the prior error, thus destroying the convergence rate. Even when the error rate is just slightly worse than $O(1/n)$, using arbitrary sample sizes is often detrimental. With QMC as with antibiotics, it is best to use the complete sequence.

There are two main forms of QMC rule, lattices and digital nets. Small examples of each are shown in Figure 15.1. After introducing QMC concepts, this chapter looks at digital constructions. Then Chapter 16 presents lattice rules. QMC methods are deterministic and it is hard to estimate their errors. Randomized QMC (RQMC), presented in Chapter 17, provides a solution. Some other advantages of RQMC over QMC are mentioned within this chapter.

Quasi-Monte Carlo algorithms may seem complicated at first. But they are essentially the same algorithms that are used in pseudo-random number generators. The digital constructions are similar to feedback shift register random number generators while lattice rules are similar to congruential generators. One useful way to think of QMC is that we are taking a small random number generator and using it in its entirety (Niederreiter, 1986). Because the algorithms have so much in common, QMC points are not materially slower to generate than pseudo-random numbers.

When describing QMC, the unit cube is variously presented as $(0,1)^d$, $[0,1)^d$ or $[0,1]^d$. Because $\int_{(0,1)^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_{[0,1)^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$, the choice would seem to make no difference. Sometimes it doesn't matter and we can make an arbitrary choice. At other times, there are useful distinctions. When f might be infinite on the boundary of the unit cube, then choosing $(0,1)^d$ lets us avoid having any $f(\boldsymbol{x}) = \pm\infty$. When f, defined on $\mathbb{R}^d$, is periodic with period 1 in every variable, then we can define f on $[0,1)^d$ and know that we have not introduced any contradictions. A further advantage of $[0,1)^d$ is that it can be split into similar pieces into which we will place the same number of sample

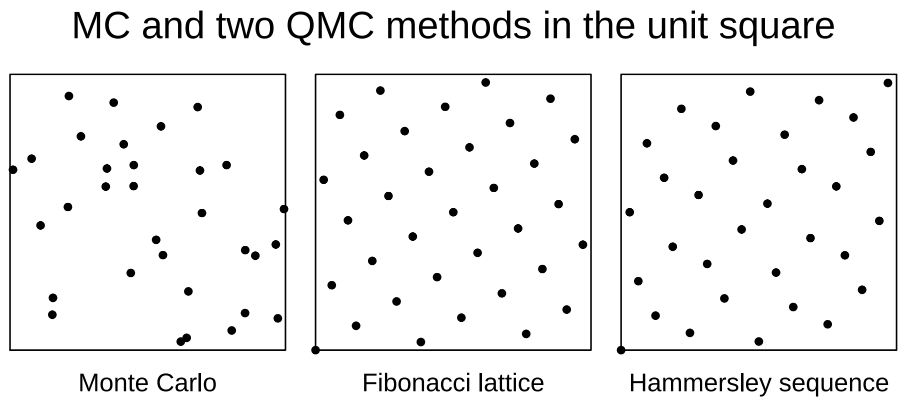

Figure 15.1: The left panel shows 32 points sampled independently from the $\mathbf{U}[0,1]^2$ distribution. The center panel shows the 34 points of a Fibonacci lattice from Chapter 16. The right panel shows the 32 point Hammersley sequence in base 2 from §15.5. Reference lines show the boundary of the unit square.

points. For instance, when $d = 1$ we have

$$[0,1) = \Big[0,\frac{1}{3}\Big) \cup \Big[\frac{1}{3},\frac{2}{3}\Big) \cup \Big[\frac{2}{3},1\Big)$$

and we might put $n/3$ points into each of those subsets on the right. The intervals $[0,1]$ and $(0,1)$ are more awkward to partition. Finally, we will see that the total variation of a function plays an important role in QMC, and total variation is defined for functions on the closed unit cube $[0,1]^d$. In short, being consistent about the unit cube to use would be a bigger nuisance than being inconsistent.

15.2 Discrepancy measures

The first task in QMC is to define what it means for points to be more uniform than uniform random points. We do this by making a numerical measure of the non-uniformity of our points.

Our goal is to estimate $\mu = \int f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ for $\boldsymbol{x} \in [0,1]^d$, that is $\mathbb{E}(f(\boldsymbol{x}))$ for $\boldsymbol{x} \sim \mathbf{U}[0,1]^d$. Our estimate is $\hat\mu = (1/n)\sum_{i=1}^n f(\boldsymbol{x}_i)$ for points $\boldsymbol{x}_i \in [0,1]^d$, which (assuming the $\boldsymbol{x}_i$ are distinct) is $\mathbb{E}(f(\boldsymbol{x}))$ for $\boldsymbol{x} \sim \mathbf{U}\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\}$. The intuition behind QMC is that if the discrete uniform distribution $\mathbf{U}\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\}$ is somehow close to the continuous distribution $\mathbf{U}[0,1]^d$, then at least for reasonable f, $\hat\mu$ should be close to μ.

If any two of the $\boldsymbol{x}_i$ are equal, then a technicality arises. The set $\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\}$ has fewer than n points and then $\hat\mu$ is no longer the mean of $f(\boldsymbol{x})$ for $\boldsymbol{x} \sim \mathbf{U}\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\}$. We would instead need a mean that weights each QMC point by its multiplicity. The QMC estimate is still an expectation, namely $\mathbb{E}(f(\boldsymbol{x}_I))$ for a random index $I \sim \mathbf{U}\{1,2,\dots,n\}$. We will assume that the $\boldsymbol{x}_i$ are distinct and work with $\hat{\mu} = \mathbb{E}(f(\boldsymbol{x}))$ for $\boldsymbol{x} \sim \mathbf{U}\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\}$. This case is simpler and covers the great majority of applications.

There are many ways to define a distance between distributions on $[0,1]^d$. One that has served well in the theory of quasi-Monte Carlo is the star discrepancy, developed next. First we generalize the notion of an interval to d dimensions.

For $\boldsymbol{a},\boldsymbol{b} \in \mathbb{R}^d$, with $a_j \leqslant b_j$, the half-open interval $[\boldsymbol{a},\boldsymbol{b})$ is the set

$$\prod_{j=1}^d [a_j,b_j) = \bigl\{\boldsymbol{x} \in \mathbb{R}^d \mid a_j \leqslant x_j < b_j,\ j=1,\dots,d\bigr\}.$$

Half-open intervals are convenient here for their partitioning property mentioned at the beginning of this chapter. We take special interest in intervals of the form $[\mathbf{0},\boldsymbol{a})$. Such an interval is often called an ***anchored box*** where the more general interval is an ***un-anchored box***.

The ***local discrepancy*** of $\boldsymbol{x}_1,\dots,\boldsymbol{x}_n$ at $\boldsymbol{a} \in [0,1]^d$ is

$$\delta(\boldsymbol{a}) = \delta(\boldsymbol{a};\boldsymbol{x}_1,\dots,\boldsymbol{x}_n) = \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{\boldsymbol{x}_i\in[\mathbf{0},\boldsymbol{a})} - \prod_{j=1}^d a_j.$$

The ratio $(1/n)\sum_{i=1}^n \mathbb{1}_{\boldsymbol{x}_i\in[\mathbf{0},\boldsymbol{a})}$ is the fraction of our n points inside $[\mathbf{0},\boldsymbol{a})$. Ideally, that fraction would match $\mathbf{vol}([\mathbf{0},\boldsymbol{a})) = \prod_{j=1}^d a_j$. Then $\delta(\boldsymbol{a})$ is positive for anchored boxes containing an excess of points $\boldsymbol{x}_i$, compared to their volume, and is negative for anchored boxes with a deficit of points. If $\delta(\boldsymbol{a}) = 0$, then the points have sampled $[\mathbf{0},\boldsymbol{a})$ in a perfectly balanced way. We may interpret $\delta(\boldsymbol{a})$ as $\widehat{\mathbf{vol}}([\mathbf{0},\boldsymbol{a})) - \mathbf{vol}([\mathbf{0},\boldsymbol{a}))$, with $\widehat{\mathbf{vol}}(A) = (1/n)\sum_{i=1}^n \mathbb{1}_{\boldsymbol{x}_i\in A}$, an estimated volume of A using points $\boldsymbol{x}_1,\dots,\boldsymbol{x}_n$. Put another way, $\delta(\boldsymbol{a})$ is the difference between $\mathbb{P}(\boldsymbol{x} \in [\mathbf{0},\boldsymbol{a}))$ under $\boldsymbol{x} \sim \mathbf{U}\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\}$ versus $\boldsymbol{x} \sim \mathbf{U}([0,1]^d)$. Figure 15.2 illustrates the local discrepancy function.

The ***star discrepancy*** of $\boldsymbol{x}_1,\dots,\boldsymbol{x}_n \in [0,1]^d$ is

$$D_n^* = D_n^*(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n) = \sup_{\boldsymbol{a}\in[0,1)^d} |\delta(\boldsymbol{a};\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)|. \tag{15.2}$$

When D_n^* is small, then the fraction of n points in each anchored box is very close to the proportion of the unit cube taken up by that box. For $d=1$, the star discrepancy reduces to the well-known Kolmogorov-Smirnov test statistic for whether $x_1,\dots,x_n$ have been sampled from $\mathbf{U}[0,1]$.

The origin plays a special role in the star discrepancy, because all the anchored boxes include it. There may be nothing about f to make the origin any more important than the other 2^d-1 corners of $[0,1]^d$. The next discrepancy measure does not treat the origin specially.

The ***extreme discrepancy*** of $\boldsymbol{x}_1,\dots,\boldsymbol{x}_n \in [0,1]^d$ is

$$D_n = D_n(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n) = \sup_{\boldsymbol{a},\boldsymbol{b}} \left|\frac{1}{n}\sum_{i=1}^n \mathbb{1}_{\boldsymbol{x}_i\in[\boldsymbol{a},\boldsymbol{b})} - \prod_{j=1}^d (b_j-a_j)\right| \tag{15.3}$$

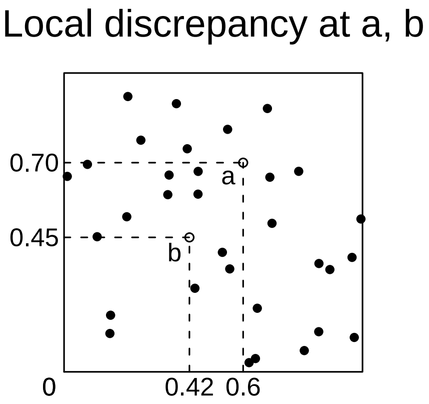

Figure 15.2: The plot illustrates the local discrepancy $\delta(\cdot)$ at points $\boldsymbol{a} = (0.6, 0.7)$ and $\boldsymbol{b} = (0.42, 0.45)$ for 32 points $\boldsymbol{x}_i$ in $[0,1]^2$. Here $|\delta(\boldsymbol{a})| = |13/32 - 0.6 \times 0.7| = 0.01375$ and $|\delta(\boldsymbol{b})| = |2/32 - 0.42 \times 0.45| = 0.1265$.

where the supremum is taken over $\boldsymbol{a}, \boldsymbol{b} \in [0,1]^d$ with $0 \leqslant a_j \leqslant b_j \leqslant 1$ for $j = 1, \dots, d$.

Sometimes the extreme discrepancy is simply called the discrepancy. It is in this sense the default discrepancy, although the star discrepancy is more frequently used. One reason for the popularity of the star discrepancy is that it has a simple and direct connection to integration error. The connection is easiest to see when $d = 1$ and $f(x)$ is a continuously differentiable function on $[0,1]$.

Theorem 15.1. *Let f have a continuous first derivative on $[0,1]$. Let $x_1, \dots, x_n \in [0,1]$. Then*

$$\frac{1}{n}\sum_{i=1}^{n} f(x_i) - \int_0^1 f(x)\,\mathrm{d}x = -\int_0^1 \delta(x) f'(x)\,\mathrm{d}x, \tag{15.4}$$

where δ is the local discrepancy function for $x_1, \dots, x_n$.

Proof. Integrating by parts,

$$\int_0^1 f(x)\,\mathrm{d}x = xf(x)\Big|_0^1 - \int_0^1 xf'(x)\,\mathrm{d}x = f(1) - \int_0^1 xf'(x)\,\mathrm{d}x.$$

An analogous summation by parts gives us

$$\sum_{i=1}^{n} f(x_i) = nf(1) - \sum_{i=0}^{n} i(f(x_{i+1}) - f(x_i))$$

using $x_0 = 0$ and $x_{n+1} = 1$. Now suppose without loss of generality that $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_n$. Then

$$\frac{1}{n}\sum_{i=1}^n f(x_i) - \int_0^1 f(x)\,\mathrm{d}x = \int_0^1 xf'(x)\,\mathrm{d}x - \sum_{i=0}^n \frac{i}{n}(f(x_{i+1}) - f(x_i)).$$

We can write the sum as an integral

$$\begin{aligned}\sum_{i=0}^n \frac{i}{n}(f(x_{i+1}) - f(x_i)) &= \sum_{i=0}^n \frac{i}{n}\int_{[x_i,x_{i+1})} f'(x)\,\mathrm{d}x \\ &= \int_0^1 \sum_{i=0}^n \frac{i}{n}\mathbb{1}_{x_i\leqslant x<x_{i+1}} f'(x)\,\mathrm{d}x \\ &= \int_0^1 \frac{1}{n}\#\{1\leqslant i\leqslant n \mid x_i < x\} f'(x)\,\mathrm{d}x \\ &= \int_0^1 \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{x_i<x} f'(x)\,\mathrm{d}x. \qquad (15.5)\end{aligned}$$

Finally

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^n f(x_i) - \int_0^1 f(x)\,\mathrm{d}x &= \int_0^1 \Bigl(x - \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{x_i<x}\Bigr) f'(x)\,\mathrm{d}x \\ &= -\int_0^1 \delta(x)f'(x)\,\mathrm{d}x,\end{aligned}$$

where δ is the local discrepancy function for $x_1, \dots, x_n$. □

From (15.4) we see that when we are lucky enough to have f' orthogonal to the local discrepancy function δ, then the integration error is zero. On the other hand, if $f' = c\delta$ for $c \neq 0$, then we get no cancellation in (15.4) and a large error is the result. Strictly speaking, f' cannot be $c\delta$ because we assumed that f' is continuous and δ is discontinuous at each x_i. But f' might be a continuous function arbitrarily close to $c\delta$, and so Theorem 15.1 does let us find integrands that will be poorly handled by $x_1, \dots, x_n$.

Discrepancies have been defined as $\sup_{S\in\mathcal{S}} |\widehat{\mathbf{vol}}(S) - \mathbf{vol}(S)|$ for various classes $\mathcal{S}$ of sets. Examples include the set of hyper-rectangles not necessarily parallel to the sides of $[0,1)^d$, the set of simplices and the set of balls. Some references to that literature are in the chapter end notes. One of the most comprehensive discrepancies is the isotropic discrepancy. Let $\mathcal{C}$ be the set of convex subsets of $[0,1)^d$. The ***isotropic discrepancy*** of $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in [0,1]^d$ is

$$J_n(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \sup_{C\in\mathcal{C}} \Bigl| \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{\boldsymbol{x}_i\in C} - \mathbf{vol}(C) \Bigr|. \qquad (15.6)$$

Not all discrepancies are defined as suprema of $|\widehat{\mathbf{vol}}(S)-\mathbf{vol}(S)|$ over classes of sets S. The ***L^2-star discrepancy*** of $\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\in[0,1]^d$ is

$$D^*_{n,2}=D^*_{n,2}(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)=\Bigl(\int_{\boldsymbol{a}\in[0,1]^d}\delta(\boldsymbol{a})^2\,\mathrm{d}\boldsymbol{a}\Bigr)^{1/2}$$

where $\delta(\boldsymbol{a})=\delta(\boldsymbol{a};\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)$ is the local discrepancy of $\boldsymbol{x}_1,\dots,\boldsymbol{x}_n$ at $\boldsymbol{a}\in[0,1]^d$. Warnock's (1972) formula for the square of $D^*_{n,2}$ is

$$\bigl(D^*_{n,2}\bigr)^2=\Bigl(\frac{1}{3}\Bigr)^d-\frac{2}{n}\sum_{i=1}^n\prod_{j=1}^d\Bigl(\frac{(1-x_{ij})^2}{2}\Bigr)+\frac{1}{n^2}\sum_{i=1}^n\sum_{i'=1}^n\prod_{j=1}^d(1-\max(x_{ij},x_{i'j})).\tag{15.7}$$

The cost of Warnock's formula grows like n^2d. It is useful for investigating small QMC rules.

Thus while the star discrepancy is $\|\delta\|_\infty=\sup_{\boldsymbol{a}\in[0,1]^d}|\delta(\boldsymbol{a})|$, the L^2-star discrepancy is $\|\delta\|_2$. General L^p norms $\|\delta\|_p=\bigl(\int_{\boldsymbol{a}\in[0,1]^d}|\delta(\boldsymbol{a})|^p\,\mathrm{d}\boldsymbol{a}\bigr)^{1/p}$ have also been used as discrepancies. For $d=1$, the L^2-star discrepancy reduces to the Cramer-von Mises distance between $\mathbf{U}[0,1]$ and $\mathbf{U}\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\}$.

The practical use of discrepancies is in proving bounds on the integration error. The most important one is the Koksma-Hlawka inequality in §15.4. In these error bounds, the first thing we look at is the rate at which discrepancy decreases as $n\to\infty$. The ordinary and star discrepancy attain the same rate:

Proposition 15.1. *For $\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\in[0,1]^d$,*

$$D^*_n\leqslant D_n\leqslant 2^dD^*_n.$$

Note: The key to proving the upper bound in Proposition 15.1 is to express the unanchored box $[\boldsymbol{a},\boldsymbol{b})$ in terms of 2^d anchored boxes, each extending from the origin to one of the vertices of $[\boldsymbol{a},\boldsymbol{b})$. Figure 15.3 illustrates the decomposition for $d=2$.

Proof of Proposition 15.1. The left side is immediate.

For the right side, the indicator function of the un-anchored box $[\boldsymbol{a},\boldsymbol{b})$ is

$$\mathbb{1}_{[\boldsymbol{a},\boldsymbol{b})}(\boldsymbol{x})=\prod_{j=1}^d\bigl(\mathbb{1}_{[0,b_j)}(x_j)-\mathbb{1}_{[0,a_j)}(x_j)\bigr).\tag{15.8}$$

We will write this function as a sum of 2^d signed indicator functions of anchored boxes.

For $u\subseteq\{1,\dots,d\}$ let $\boldsymbol{c}^{(u)}=\boldsymbol{a}_u{:}\boldsymbol{b}_{-u}$, a merger of components from $\boldsymbol{a}$ and $\boldsymbol{b}$, given by $c^{(u)}_j=a_j$ for $j\in u$ and $c^{(u)}_j=b_j$ for $j\not\in u$. Expanding (15.8) we get

$$\mathbb{1}_{[\boldsymbol{a},\boldsymbol{b})}(\boldsymbol{x})=\sum_{u\subseteq\{1,\dots,d\}}(-1)^{|u|}\mathbb{1}_{[\boldsymbol{0},\boldsymbol{c}^{(u)})}(\boldsymbol{x}).$$

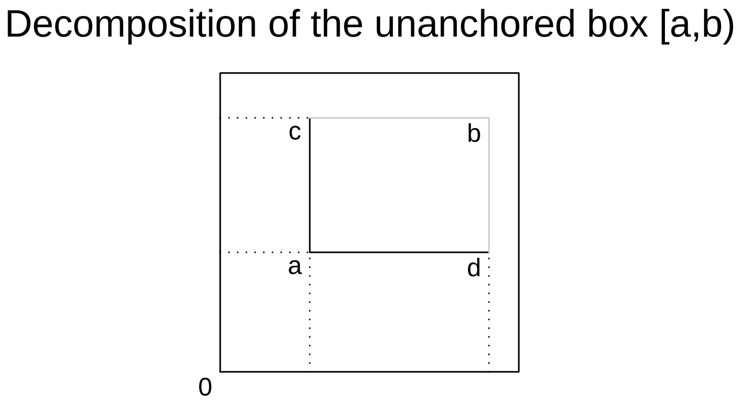

Figure 15.3: An unanchored box $[\boldsymbol{a},\boldsymbol{b})\subset[0,1]^2$ is shown. Its indicator function can be written $\mathbb{1}_{[\boldsymbol{a},\boldsymbol{b})}(\boldsymbol{x})=\mathbb{1}_{[\boldsymbol{0},\boldsymbol{b})}(\boldsymbol{x})-\mathbb{1}_{[\boldsymbol{0},\boldsymbol{c})}(\boldsymbol{x})-\mathbb{1}_{[\boldsymbol{0},\boldsymbol{d})}(\boldsymbol{x})+\mathbb{1}_{[\boldsymbol{0},\boldsymbol{a})}(\boldsymbol{x})$ in terms of indicators of anchored boxes at $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ and $\boldsymbol{d}$.

Now

$$
\begin{aligned}
&\Bigl|\widehat{\mathbf{vol}}([\boldsymbol{a},\boldsymbol{b}))-\mathbf{vol}([\boldsymbol{a},\boldsymbol{b}))\Bigr|\\
&=\Bigl|\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{[\boldsymbol{a},\boldsymbol{b})}(\boldsymbol{x}_i)-\int_{[0,1]^d}\mathbb{1}_{[\boldsymbol{a},\boldsymbol{b})}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\Bigr|\\
&=\Bigl|\frac{1}{n}\sum_{i=1}^{n}\sum_{u}(-1)^{|u|}\mathbb{1}_{[\boldsymbol{0},\boldsymbol{c}^{(u)})}(\boldsymbol{x}_i)-\int_{[0,1]^d}\sum_{u}(-1)^{|u|}\mathbb{1}_{[\boldsymbol{0},\boldsymbol{c}^{(u)})}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\Bigr|\\
&\leqslant\sum_{u}\Bigl|\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{[\boldsymbol{0},\boldsymbol{c}^{(u)})}(\boldsymbol{x}_i)-\int_{[0,1]^d}\mathbb{1}_{[\boldsymbol{0},\boldsymbol{c}^{(u)})}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\Bigr|\\
&\leqslant 2^dD_n^*.
\end{aligned}
$$

Since $[\boldsymbol{a},\boldsymbol{b})$ is arbitrary, $D_n\leqslant 2^dD_n^*$. □

Discrepancies are mainly used to get rates of convergence. We will see below that those rates show how QMC can be much better than MC. The factor 2^d can be quite large, but it does not change rates in n for fixed d. It seems unlikely that $\boldsymbol{x}_i$ would really have a ratio of D_n^*/D_n anywhere close to 2^d for QMC points in use.

15.3 Discrepancy rates

The star discrepancy of random points $\boldsymbol{x}_i\sim\mathbf{U}[0,1]^d$ is well studied. For any point $\boldsymbol{a}\in[0,1]^d$, we easily find that $\mathbb{E}(\delta^2(\boldsymbol{a};\boldsymbol{x}_1,\dots,\boldsymbol{x}_n))^{1/2}=\sqrt{p(1-p)/n}$ where $p=\mathbf{vol}([\boldsymbol{0},\boldsymbol{a}])$, and so the local discrepancy decreases like $1/\sqrt{n}$ at any

$\boldsymbol{a}$. The star discrepancy, which must account for finding the least favorable anchored box $[\mathbf{0}, \boldsymbol{a}]$ for the sampled values $\boldsymbol{x}_i$ is of just slightly larger order. It is eventually no larger than $\sqrt{\log\log n}/\sqrt{2n}$ with probability 1 as Theorem 15.2 shows.

Theorem 15.2. *Let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \sim \mathbf{U}[0,1]^d$ be independent. Then*

$$\mathbb{P}\biggl(\limsup_{n\to\infty} \frac{\sqrt{2n}\, D_n^*(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)}{\sqrt{\log\log n}} = 1\biggr) = 1.$$

Proof. Chung (1949) proved this for $d = 1$ and Kiefer (1961) proved it for $d \geqslant 1$. □

It is possible to attain much lower discrepancies than random points do. We will see constructions of sequences that have low discrepancy, according to the following criterion. The infinite sequence $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots \in [0,1]^d$ is a ***low discrepancy sequence*** if

$$D_n^*(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = O(n^{-1}(\log n)^d)$$

as $n \to \infty$. Any finite positive power of $\log(n)$ is asymptotically negligible compared to any finite positive power of n. Thus a low discrepancy sequence has

$$D_n^*(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = O(n^{-1+\epsilon})$$

for any $\epsilon > 0$ as $n \to \infty$. It is also $o(n^{-1+\epsilon})$ but $O(n^{-1+\epsilon})$ is more commonly used.

Even modest powers of $\log(n)$ like $\log(n)^{10}$ can be quite large compared to n when n is small enough to be a feasible sample size. We return to this point later when discussing how discrepancy affects the accuracy of QMC integration.

In practice, we only use a finite value of n, not an entire infinite sequence. For finite n, we can find constructions that are better than $O(n^{-1}(\log n)^d)$. Here $O(\cdot)$ refers to asymptotics as $n \to \infty$ so we need to reconcile $n \to \infty$ with finite n. We consider an infinite sequence of finite sequences. The finite sequences increase in length, and we take the limit as this length goes to infinity. Specifically, let $\boldsymbol{x}_{in} \in [0,1]^d$ for all $i = 1, \dots, n$ and all $n \in \mathcal{N}$ where $\mathcal{N} = \{n_1, n_2, \dots\}$ is an infinite set of positive integers with $n_j < n_{j+1}$. We call such an arrangement a ***triangular array*** because it can be displayed as a table

$$\begin{array}{ccccccccc}
\boldsymbol{x}_{1n_1} & \boldsymbol{x}_{2n_1} & \cdots & \boldsymbol{x}_{n_1n_1} & & & & & \\
\boldsymbol{x}_{1n_2} & \boldsymbol{x}_{2n_2} & \cdots & \boldsymbol{x}_{n_1n_2} & \cdots & \boldsymbol{x}_{n_2n_2} & & & \\
\boldsymbol{x}_{1n_3} & \boldsymbol{x}_{2n_3} & \cdots & \boldsymbol{x}_{n_1n_3} & \cdots & \boldsymbol{x}_{n_2n_3} & \cdots & \boldsymbol{x}_{n_3n_3} & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{array}$$

of infinitely many rows of increasing length. The j'th row has n_j elements. In examples we may have $n_j = j$ which would give the table a truly triangular shape, or perhaps $n_j = 2^j$, among other choices.

Some triangular arrays of points in $[0,1]^d$ have

$$D^*_{n_j}(\boldsymbol{x}_{1n_j},\dots,\boldsymbol{x}_{n_jn_j}) = O(n_j^{-1}(\log(n_j))^{d-1})$$

as $j\to\infty$. Compared to a low discrepancy sequence, the triangular array saves a factor of $\log(n)$, which is like reducing the dimension d by one.

A potential drawback to a triangular array construction is that the points in the $j+1$'st row need not include the ones in the j'th row. If we find that n_j points are not enough to get an accurate answer, then we may have to start all over again computing f at n_{j+1} new values and discarding the previous ones. A triangular array is ***extensible*** if $\boldsymbol{x}_{in_j} = \boldsymbol{x}_{in_{j+1}}$ for all $j\geqslant 1$ and $i=1,\dots,n_j$. An extensible array lets us reuse all the previous points. We can extend from one good sample size n_j to a larger good sample size n_{j+1} computing only $n_{j+1}-n_j$ new function values. Extensible rules just use the first n_j points of an infinite sequence of $\boldsymbol{x}_i$.

The best possible rate for discrepancies is not known. Roth (1954) gives a celebrated lower bound $D^*_{n,2}\geqslant c_d(\log(n))^{(d-1)/2}/n$ for the L^2-star discrepancy which holds for any set of n points in $[0,1]^d$. The constant $c_d>0$ does not depend on n. Roth's result implies that $D^*_n\geqslant c_d(\log(n))^{(d-1)/2}/n$ too. It is widely believed that the rate $D^*_n = o(n^{-1}(\log n)^{d-1})$ cannot be attained by any triangular array. This has been proved for $d\leqslant 2$. Dick and Pillichshammer (2010, Chapter 2) give more information on bounds for discrepancies.

15.4 The Koksma-Hlawka Inequality

When we replace randomly sampled points $\boldsymbol{x}_i$ by deterministic ones, we can no longer rely upon the law of large numbers to ensure convergence. We also lose the central limit theorem. Here we look at replacement concepts for deterministic quadrature rules. We'll work with the star discrepancy. Qualitatively similar results exist for many other discrepancies.

Definition 15.1. The infinite sequence $\boldsymbol{x}_1,\boldsymbol{x}_2,\dots\in[0,1]^d$ is ***uniformly distributed*** if $D^*_n(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)\to 0$ as $n\to\infty$.

Theorem 15.3. *Let f be a Riemann integrable function on $[0,1]^d$. If $\boldsymbol{x}_1,\boldsymbol{x}_2,\dots\in[0,1]^d$ are uniformly distributed then*

$$\Bigl|\frac1n\sum_{i=1}^n f(\boldsymbol{x}_i)-\int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\Bigr|\to 0 \tag{15.9}$$

as $n\to\infty$.

Proof. Kuipers and Niederreiter (1974) give this as Exercise 6.1. □

Using Theorem 15.1 we can easily verify Theorem 15.3 for $d=1$ and f' continuous. We write

$$\Bigl|\frac1n\sum_{i=1}^n f(x_i)-\int_{[0,1]} f(x)\,\mathrm{d}x\Bigr|\leqslant\int_0^1|\delta(x)f'(x)|\,\mathrm{d}x\leqslant D^*_n\int_0^1|f'(x)|\,\mathrm{d}x$$

and then $D_n^* \to 0$ by the definition of uniformly distributed x_i.

Theorem 15.3 is the QMC counterpart to the law of large numbers. Our estimate will converge to the right answer if we use a uniformly distributed sequence of points. There is a new condition that we did not require for MC: the function f must now be Riemann integrable. That rules out some functions we might not have cared about. One such is the function which is 1 if all components of $\boldsymbol{x}$ are rational and is 0 otherwise. Another is a classic pathological example, the function $f(\boldsymbol{x})$ which is 1 at each of the infinitely many sample points $\boldsymbol{x}_i$, and is 0 everywhere else. But requiring Riemann integrability also rules out unbounded functions, including many that are important to applications. For example, we commonly apply the inverse Gaussian CDF Φ^{-1} to one or more components of $\boldsymbol{x}$ and subsequent steps don't always leave us with a bounded quantity. Theorem 15.3 has a converse:

Theorem 15.4. *If the limit* (15.9) *holds for all uniformly distributed sequences $\boldsymbol{x}_i \in [0,1]^d$, then f is Riemann integrable.*

Proof. The case $d = 1$ is due to de Bruijn and Post (1968) and Binder (1970) proves it for $d \geqslant 1$. □

Quasi-Monte Carlo often attains good empirical results on unbounded functions. From Theorem 15.4 we know that conditions beyond uniform distribution must be imposed on $\boldsymbol{x}_i$. There are more remarks and references about QMC for unbounded integrands in the chapter end notes, and §17.12 considers randomized QMC for unbounded integrands.

In Monte Carlo sampling, the central limit theorem is used to study the error. For QMC, there is the Koksma-Hlawka inequality. It requires a new quantity, $V_{\mathrm{HK}}(f)$, which is the total variation of f in the sense of Hardy and Krause. For $d = 1$, $V_{\mathrm{HK}}(f)$ is the familiar total variation of f. See the chapter end notes for a discussion of total variation, including the d-dimensional case.

Theorem 15.5 (Koksma-Hlawka inequality)**.** *For $d \geqslant 1$ and $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in [0,1]^d$,*

$$\Bigl|\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i) - \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\Bigr| \leqslant D_n^*(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)V_{\mathrm{HK}}(f), \tag{15.10}$$

where $V_{\mathrm{HK}}(f)$ denotes the total variation of f in the sense of Hardy and Krause.

Proof. This was proved by Koksma (1943) for $d = 1$ and Hlawka (1961) for $d \geqslant 1$. Kuipers and Niederreiter (1974, Chapter 5) include a proof. □

Theorem 15.5 gives control over the quadrature error $|\hat{\mu} - \mu|$. The upper bound is the product of a measure of roughness of f times a measure of non-uniformity of $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$. While it is a counterpart of the CLT, there are some important differences. First of all, the Koksma-Hlawka inequality is not probabilistic. It holds with certainty, or 100% confidence in statistical language. We ordinarily prefer 100% confidence to 99%, except perhaps when the former interval is far wider than the latter. Second, the Koksma-Hlawka inequality holds for finite n, while the CLT only holds in the limit as $n \to \infty$.

Having a 100% confidence interval for the n specific points we use may sound too good to be true. There is indeed a problem. While we are sure that the interval $\hat\mu \pm D_n^* V_{\mathrm{HK}}(f)$ contains μ, outside of very special cases, neither D_n^* nor $V_{\mathrm{HK}}(f)$ is known to us. Therefore we don't get a usable 100% confidence interval. The star discrepancy is very hard to compute for modestly large d and no practical algorithms for it can handle n as large as we want to use in QMC. While we could in principle compute D_n^* once and then use it for many integrands f, we still would not know the value of $V_{\mathrm{HK}}(f)$. The total variation is ordinarily harder to compute than μ. It involves $2^d - 1$ multidimensional integrals of certain partial derivatives of f as described in the chapter end notes.

Theorem 15.5 is however an extremely important result. It shows that if we use a low discrepancy sequence then we will achieve $|\hat\mu - \mu| = O(n^{-1+\epsilon})$, for any $\epsilon > 0$. As a result, we know that if $V_{\mathrm{HK}}(f) < \infty$, then for large enough n we should get much better accuracy from QMC than from MC. Also, the search for good QMC methods may be organized around reducing D_n^*, and other similar figures of merit.

The Koksma-Hlawka inequality is tight. We cannot replace the right hand side of (15.10) by $\gamma D_n^* V_{\mathrm{HK}}$ for any $\gamma < 1$, because given $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in [0,1]^d$ there is always some function f for which

$$\Bigl|\frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i) - \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\Bigr| > \gamma D_n^*(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n) V_{\mathrm{HK}}(f).$$

Being tight (in the sense above) does not prevent the inequality from also being loose in a given application. Equality holds in (15.10) for a worst case function that is allowed to take account of the locations of the sampling points. For some other function f we might well have $|\hat\mu - \mu| \ll D_n^* V_{\mathrm{HK}}(f)$.

The quantity $\log(n)^{d-1}/n$ causes a lot of difficulty even for moderate dimensions, like $d = 10$. It can require quite enormous n before that quantity is below $n^{-1/2}$, and we have not yet considered the lead constant. Here we rule out $n = 1$ which is clearly not relevant to an asymptotic bound. One never actually sees an error behaving like $\log(n)^9/n$, at least in published papers, for the usual QMC points and real 10-dimensional integrands. That rate applies to the worst function of bounded variation on $[0,1]^{10}$ that could be chosen by an adversary who knew the locations of the points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ to be used. That makes the criterion quite pessimistic. Furthermore, it is not feasible for one and the same function to be worst case for all of the different values of n used when plotting absolute error versus n.

We know that for some n, QMC will be better than MC, but we cannot tell a user which n that will be. An additional difficulty is that the coefficient of

$\log(n)^{d-1}/n$ includes $V_{\mathrm{HK}}(f)$. Morokoff and Caflisch (1995) compare

$$f_1(\boldsymbol{x}) = \prod_{j=1}^{d} x_j \quad \text{and} \quad f_2(\boldsymbol{x}) = \prod_{j=1}^{d}(1 - x_j). \tag{15.11}$$

These certainly appear to be about equally challenging to integrate numerically, yet $V_{\mathrm{HK}}(f_1) = 2^d - 1$, while $V_{\mathrm{HK}}(f_2) = 1$. The difference stems from the way that V_{HK} is defined (see the end notes). There is thus an exponential in d difference in the lead constants for the error bounds of these two quite similar integrands.

These problems with the theoretical accuracy of QMC have lead to some empirical alternatives. Many authors fit a linear regression model $\log|\hat{\mu} - \mu| \doteq \alpha_0 - \alpha_1 \log(n)$ to example data where μ is known, from which it will then appear that the error is $O(n^{-\alpha_1})$ with α_1 commonly between 1/2 and 1. We know theoretically that such rates are not the true asymptotic rates, while at the same time, they can be much more realistic for a given range of n than the asymptotic rates. It is then difficult to know for which other integrands and sample sizes is the empirical rate $O(n^{-\alpha_1})$ a good guide.

A less common empirical investigation looks at alternatives to using $V_{\mathrm{HK}}(f)$ to describe performance at finite n for different functions f. That is like seeking an empirical α_0 in the regression above. In a set of examples, Schlier (2004) finds that $V_{\mathrm{HK}}(f)$ has little to do with the QMC accuracy, confirming what seemed clear in the discussion of f_1 and f_2 from (15.11). He then finds that $\sigma^2 = \mathrm{Var}(f(\boldsymbol{x}))$ provides a more reliable scaling. This measure is problematic theoretically because we could choose f completely lacking any of the regularity that QMC uses without that irregularity being reflected in $\mathrm{Var}(f(\boldsymbol{x}))$. We therefore cannot know to which other integrands his findings might apply. His test functions all had bounded variation and most were differentiable.

The empirical answers are not aligned with known theory. By the same token, the theoretical guidelines fail empirically. Schlier (2004) reports inaccuracies of "tens of orders of magnitude" from using the Koksma-Hlawka bound. Improved descriptions of QMC performance are available by considering coordinate projections of the QMC points in §15.8. That connects to the notions of effective dimension in §17.2 and weighted spaces in §7.7. Those concepts narrow the gap between theoretical and empirical performance.

Some forms of randomized QMC provide control on those logarithmic powers. They ensure that the mean squared error in RQMC sampling cannot be above a fixed known multiple of the mean squared error of MC at a given value of n, even where $\log(n)^{d-1}$ is enormous. That bound holds even for worst case square integrable functions specifically chosen to make RQMC have a large variance relative to MC.

15.5 van der Corput and Halton sequences

Given a sample size n, a natural way to evenly distribute n points in $[0,1]$ is to form n congruent intervals $[(i-1)/n, i/n]$ for $i = 1, \dots, n$, and take their center points $(i-1/2)/n$. Niederreiter (1992b) shows that for $x_i \in [0,1]$,

$$D_n^*(x_1, \dots, x_n) = \frac{1}{2n} + \max_{1 \leqslant i \leqslant n} \Bigl| x_{(i)} - \frac{i-1/2}{n} \Bigr|$$

where $x_{(i)}$ is the i'th smallest of the x_i. Thus the midpoint rule $x_i = (i-1/2)/n$ minimizes D_n^* attaining the value $1/(2n)$. A similar representation shows that the midpoint rule also minimizes D_n.

One problem with the midpoint rule is that it is awkward to extend. The midpoint rule with $n+1$ points does not contain the n point rule. Neither does the one with $2n$ points. The midpoint rule with $3n$ sample points does extend the one with n points. But if we start with n_1 points and keep extending our rule this way we get a sequence of quadrature rules of size $n_j = 3^{j-1}n_1$ which grows uncomfortably fast. Extending a QMC rule by one point is not very effective, but seeing how it is done helps to understand how digital constructions work, and of course we can extend it from one good sample size to another.

We would like to find an infinite sequence $x_i \in [0,1]$ for $i \geqslant 1$ with a small discrepancy D_n or D_n^* for all n. The most reasonable one point rule is $x_1 = 1/2$. This splits $[0,1]$ into two equal intervals, left and right. The next point x_2 might as well be in the middle of one such interval. If we take $x_2 = 1/4$ then it is reasonable to put $x_3 = 3/4$ to recover some balance. Now we have four intervals of equal length so it is reasonable to split one of them in two. If we've split a subinterval of $[0,1/2]$ with x_4 then it seems fair to split a subinterval of $[1/2,1]$ with x_5.

The van der Corput sequence carries out just such a myopic equidistribution algorithm. To define it, we introduce a ***digit retrieval*** function. For integers $i \geqslant 0$, $k \geqslant 0$, and $b \geqslant 2$, let $\mathsf{d}_{k,b}(i) \in \{0, 1, \dots, b-1\}$ be the coefficient of b^k in the base b expansion of i. That is

$$i = \sum_{k=0}^{\infty} \mathsf{d}_{k,b}(i) b^k, \tag{15.12}$$

where only finitely many of the $\mathsf{d}_{k,b}$ are nonzero. Equation (15.12) uniquely determines $\mathsf{d}_{k,b}(i)$ given i, k, and b. When b is understood, we use $\mathsf{d}_k(i)$ as shorthand for $\mathsf{d}_{k,b}(i)$.

The ***radical inverse function*** ϕ_b in base $b \geqslant 2$ is defined as

$$\phi_b(i) = \sum_{k=0}^{\infty} \mathsf{d}_{k,b}(i) b^{-k-1}. \tag{15.13}$$

The radical inverse function flips the base b expansion of i around the decimal point (b-minal point), mapping the nonnegative integers into $[0,1)$.

i				$\phi_2(i)$
1	1	0.1	1/2	0.5
2	10	0.01	1/4	0.25
3	11	0.11	3/4	0.75
4	100	0.001	1/8	0.125
5	101	0.101	5/8	0.625
6	110	0.011	3/8	0.375
7	111	0.111	7/8	0.875
8	1000	0.0001	1/16	0.0625
9	1001	0.1001	9/16	0.5625

Table 15.1: The table illustrates computation of the base 2 radical inverse function ϕ_2 used in the van der Corput sequence. From left to right: The integer i is converted to base 2. Then its binary digits are reflected about the binary point. The result is then re-expressed as a fraction and as a number in base 10.

The ***van der Corput sequence*** is defined by $x_i = \phi_2(i-1)$ for $i \geqslant 1$. See Table 15.1 for an illustration. It is customary to start the van der Corput sequence with $x_1 = \phi_2(0) = 0$, instead of taking the first point to be $x_1 = 1/2$ as discussed above. However, having $x_1 = 0$ often causes problems with integrands that are unbounded. As a result we often take $x_i = \phi_2(i)$ instead, in applications.

Reading down the second and third column of Table 15.1 we see how van der Corput's sequence remains balanced. The integers i alternate between odd and even, ending in 1 or 0 base 2. When flipped at the binary point, they therefore alternate between subintervals $[1/2, 1)$ and $[0, 1/2)$. If n is even, half the points are on the left and half are on the right, while if n is odd, the disparity between the half intervals is just one point. Similarly, the last k binary digits of i cycle through 2^k possible endings, and every consecutive 2^k points are equally stratified among 2^k intervals $[\ell 2^{-k}, (\ell+1)2^{-k})$ for $0 \leqslant \ell < 2^k$.

The same idea works in any integer base $b \geqslant 2$. The ***van der Corput sequence in base b ⩾ 2*** is defined by $x_i = \phi_b(i-1)$ for $i \geqslant 1$. As with base 2, we often take $x_i = \phi_b(i)$ to avoid having $x_1 = 0$. The van der Corput sequences are low discrepancy sequences:

Theorem 15.6. *For $i \geqslant 1$ and $b \geqslant 2$ let $x_i = \phi_b(i-1) \in [0,1]$. Then*

$$\limsup_{n\to\infty} \frac{nD_n^*(x_1,\dots,x_n)}{\log n} = \begin{cases} \dfrac{b-1}{4\log b} & b \text{ odd} \\ \dfrac{b^2}{4(b+1)\log b} & b \text{ even.}\end{cases}$$

Proof. Faure (1982). □

The same asymptotic star discrepancies apply if we start the sequence at $\phi_b(1)$ instead of $\phi_b(0)$. Indeed, we could skip ahead any number of places,

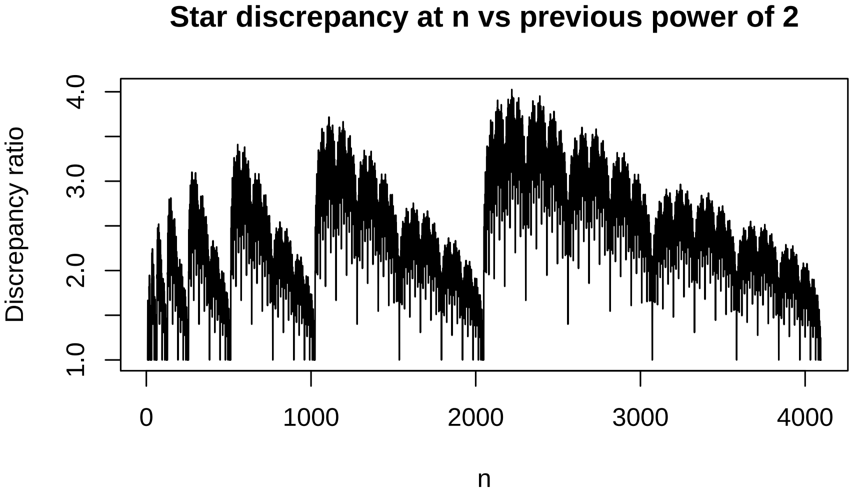

Figure 15.4: For $1 \leqslant n \leqslant 4096$, we see the star discrepancy of the first n points of the van der Corput sequence divided by that of the first 2^m points for $m = m(n) = \max\{k \mid 2^k \leqslant n\}$.

taking $x_i = \phi_b(N + i - 1)$ for $i \geqslant 1$ and $N \geqslant 0$. The limit in Theorem 15.6 is strictly increasing in b for $b \geqslant 3$. The value for $b = 2$ is just slightly worse than the one for $b = 3$, and so $b = 3$ attains the best limit.

Samples sizes $n = 2^m$ are especially good for the van der Corput sequence. Figure 15.4 shows

$$D_n^*(\phi_2(1), \dots, \phi_2(n))/D_{m(n)}^*(\phi_2(1), \dots, \phi_2(m(n)))$$

where $m(n) = 2^{\lfloor \log_2(n) \rfloor}$ is the greatest power of 2 that is less than or equal to n. That ratio is never below 1 for $1 \leqslant n \leqslant 2^{14}$. In that range, adding points after a power of 2 cannot make an improvement until one reaches the next power of 2. Perhaps that ratio is never below 1 for any $n \geqslant 1$. The figure includes $1 \leqslant n \leqslant 4096 = 2^{12}$. Exercise 15.7 asks you to investigate $b = 3$.

The greatest need for QMC methods is not for $d = 1$, but for d large enough that iterated one dimensional rules are ineffective. One of the simplest methods for higher d is the Halton sequence. The Halton sequence uses radical inverse generators in bases $b_j \geqslant 2$ for $j = 1, \dots, d$. In order for these points to be equidistributed it is necessary for b_j to be relatively prime to each other. That is, for $j \neq k$ the bases b_j and b_k should not both be divisible by any positive integer other than 1. The definition below uses the usual choice.

Definition 15.2. The ***Halton sequence*** $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots \in [0, 1)^d$ has

$$x_{ij} = \phi_{p_j}(i - 1), \quad i \geqslant 1, \quad 1 \leqslant j \leqslant d,$$

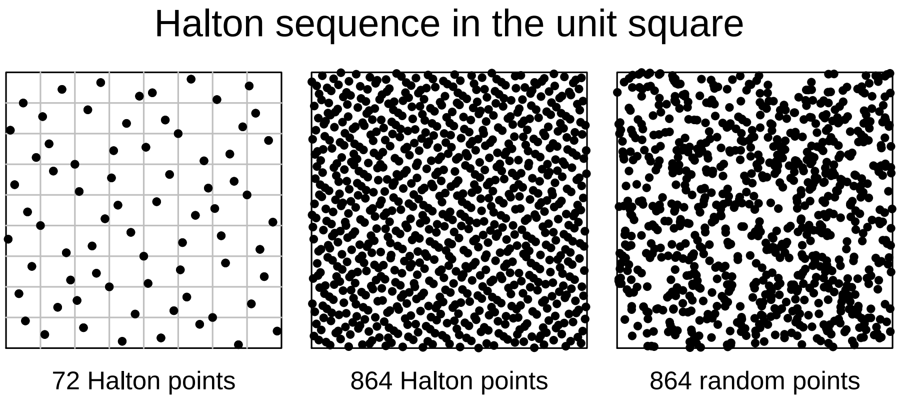

Figure 15.5: The left panel shows the first $72 = 2^3 3^2$ points of the Halton sequence $\boldsymbol{x}_i = (\phi_2(i), \phi_3(i))$ for $i = 1, \dots, 72$. The reference lines divide the unit square into a grid of 8 columns and 9 rows. Each grid rectangle has one Halton point. The middle panel shows the first $864 = 2^5 3^3$ Halton points. The right panel shows 864 random points for comparison.

where $p_1 = 2$, $p_2 = 3$, and more generally, p_j is the j'th prime number.

Figure 15.5 illustrates the Halton sequence for $d = 2$, skipping the point at the origin. For $d = 2$, if we take $n = 2^a 3^b$ consecutive points from the Halton sequence, for positive integers a and b, then from the radical inverse construction, each interval $[(\ell - 1)/2^a, \ell/2^a)$ for $\ell = 1, \dots, 2^b$ has 3^b of the x_{i1}. Similarly each interval $[(\ell-1)/3^b, \ell/3^b)$ for $\ell = 1, \dots, 3^b$ has 2^a of the x_{i2}. Even better, if we intersect these strata in the natural way, we get n rectangles each with exactly one of the n Halton points.

More generally, the projection of $n = \prod_{j=1}^d p_j^{a_j}$ consecutive Halton points onto components $j \in u \subset \{1, \dots, d\}$ places $\prod_{j \notin u} p_j^{a_j}$ points into each of $\prod_{j \in u} p_j^{a_j}$ congruent hyper-rectangular regions.

The reason for using the first d primes is that smaller bases give finer stratification than larger ones. The smallest d relatively prime natural numbers (ruling out 1 because it can't be used as a base) are the first d primes.

For large d it would be cumbersome to have n be a multiple of a power of each p_j used. We then find that no values of n are especially good for Halton sequences. Powers of 10 may then be ok, not because they are especially good, but instead because no other sample sizes are especially good either. The variables getting base 2 or 3 in the Halton sequence will tend to have the best equidistribution and so it makes sense to use them on the input dimensions thought to be most important.

The Halton sequence is extensible. If we don't need an extensible sequence, then a scheme of Hammersley is preferable.

Definition 15.3. The ***Hammersley sequence*** $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \in [0, 1)^d$ has

$x_{i1} = (i-1)/n$ and $x_{ij} = \phi_{p_{j-1}}(i-1)$ for $j = 2, \dots, d$ where p_j is the j'th prime number.

The Hammersley sequence samples the first variable x_{i1} with equispaced points and then uses a $d-1$-dimensional Halton sequence for the rest of the variables. By taking smaller bases than the Halton points use, better equidistribution is obtained. In practice, the first dimension can instead be $x_{i1} = (i-1/2)/n$ and the others can be any n consecutive values from a $d-1$-dimensional Halton sequence. The Halton and Hammersley sequences both achieve low discrepancy.

Theorem 15.7. *For the Halton sequence with $n \geqslant 2$,*

$$D_n^*(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \leqslant \frac{1}{nd!} \prod_{j=1}^{d} \left(\frac{\lfloor p_j/2 \rfloor \log(n)}{\log(p_j)} + d \right) + O\left(\frac{\log(n)^{d-1}}{n} \right). \qquad (15.14)$$

For the Hammersley sequence with $n \geqslant 1$,

$$D_n^*(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \leqslant \frac{1}{n(d-1)!} \prod_{j=1}^{d-1} \left(\frac{\lfloor p_j/2 \rfloor \log(n)}{\log(p_j)} + d - 1 \right) + O\left(\frac{\log(n)^{d-2}}{n} \right) \qquad (15.15)$$

Proof. These are derived from theorems presented in Chapter 2 of Dick and Pillichshammer (2010). They are based on the work of Atanassov (2004) who attained a notable reduction in the lead constant, compared to the original results of Halton (1960) and Hammersley (1960). □

Equation (15.14) shows $D_n^* = O(n^{-1}(\log n)^d)$ for the Halton sequence. The non-extensible Hammersley sequence attains the slightly better rate $D_n^* = O(n^{-1}(\log n)^{d-1})$. Using bounds on the size of the j'th prime number, Dick and Pillichshammer (2010) show that the lead term in D_n^* is at most $7\log(n)^d/(2^d dn)$ for the Halton sequence and $7\log(n)^{d-1}/(2^{d-1}(d-1)n)$ for the Hammersley sequence.

The Halton sequence has a problem for large values of d. Figure 15.6 shows three pairwise projections of the first 1000 points. They correspond to the last two dimensions when $d = 10$ or 20 or 30. The projection of $\boldsymbol{x}_i$ onto their 29'th and 30'th dimensions will not be stratified if we use fewer than $n = 109 \times 113 = 12317$ points. It will be exactly stratified if we use a multiple of 12317 points and approximately stratified if $n \gg 12317$, but otherwise the projection might be bad, as shown.

The bad projection in the third panel of Figure 15.6 becomes even worse when we adjoin the 28'th prime and look at $\boldsymbol{x}_{i,28:30}$ in three dimensions. The two dimensional projection shows a handful of nearly diagonal stripes. The three dimensional projection similarly shows such stripes as a handful of line segments in the unit cube, leaving a large void.

The Halton sequence can be improved, by scrambling its digits as described next. First we introduce a ***generalized van der Corput*** sequence with $x_i =$

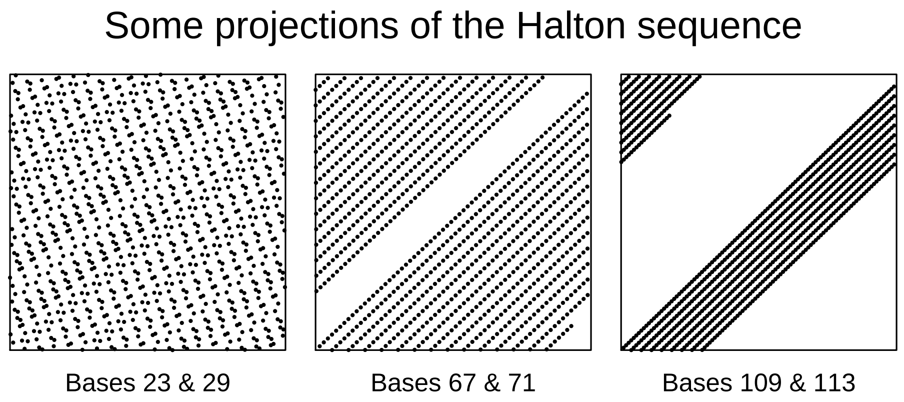

Figure 15.6: Each panel shows a two dimensional view of 1000 points of the Halton sequence, starting at $\phi_b(1)$. From left to right, the bases are for the 9'th and 10'th primes, the 19'th and 20'th primes, and the 29'th and 30'th primes.

$\phi_{b,\pi}(i-1)$ where

$$\phi_b(i;\pi) = \sum_{k=0}^{\infty} \pi\bigl(\mathsf{d}_{k,b}(i)\bigr) b^{-k-1}. \tag{15.16}$$

where π is a permutation of $\{0, 1, \dots, b-1\}$, and as before, i has base b digits $\mathsf{d}_{k,b}(i)$. There is a reason to prefer permutations with $\pi(0) = 0$. The integer i has only finitely many nonzero digits $\mathsf{d}_{k,b}(i)$, and taking $\pi(0) = 0$ means we only need to sum finitely many terms to compute $\phi_b(i;\pi)$. The alternative is to sum only those terms that affect a finite precision floating point representation of x_i.

A ***scrambled Halton sequence*** $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots$ has $x_{ij} = \phi_{p_j}(i-1;\pi_j)$ for $i \geqslant 1$ and $j = 1, \dots, d$, where p_j is the j'th prime number and π_k is a permutation of $\{0, 1, \dots, k-1\}$ for which $\pi_k(0) = 0$. Taking $x_{ij} = \phi_{p_j}(i;\pi_j)$ instead avoids starting at $(0, \dots, 0)$. There are numerous proposals for the permutations π_k.

A proposal due to Faure (1992) is widely used and one of the simplest to describe. The first few permutations are:

$$\begin{aligned}
\pi_2 &= (0\ 1)\\
\pi_3 &= (0\ 1\ 2)\\
\pi_4 &= (0\ 2\ 1\ 3)\\
\pi_5 &= (0\ 3\ 2\ 1\ 4)\\
\pi_6 &= (0\ 2\ 4\ 1\ 3\ 5)\\
\pi_7 &= (0\ 2\ 5\ 3\ 1\ 4\ 6)\\
\pi_8 &= (0\ 4\ 2\ 6\ 1\ 5\ 3\ 7).
\end{aligned} \tag{15.17}$$

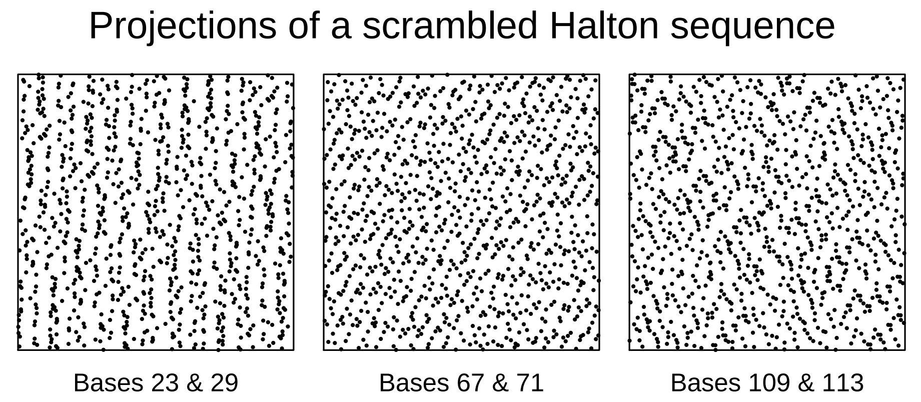

Figure 15.7: This figure shows the points from Figure 15.6 after applying Faure's permutations to their digits.

These permutations may be defined recursively. For even b, there is a simple pattern relating π_b to $\pi_{b/2}$. Letting $b = 2k$ with an integer $k \geqslant 2$, the rule is

$$\pi_b = (2\pi_k, 2\pi_k + 1).$$

For odd b the rule is a bit more complicated. If $b = 2k + 1$ for integer $k \geqslant 1$, then for $j = 0, \dots, 2k - 1$ let

$$\eta(j) = \begin{cases} \pi_{2k}(j), & \pi_{2k}(j) < k \\ \pi_{2k}(j) + 1, & \pi_{2k}(j) \geqslant k, \end{cases}$$

and put

$$\pi_{2k+1} = (\eta(0 : (k-1)),\ k,\ \eta(k : (2k-1))).$$

The bad projections from Figure 15.6 are replotted in Figure 15.7 after applying the permutations from Faure (1992) to their digits. The result is a substantial improvement, though Exercise 15.8 has a cautionary note. There have been many more proposals for deterministic scrambling of the digits of the Halton sequence. A related idea is to use ***leaped sequences***, defined by $x_{ij} = \phi_{p_j}(\ell(i-1))$. Here $\ell > 1$ is a leaping constant that should be relatively prime to all the p_j that are used. In §17.10 we will look at a proposal that chooses the permutation at random.

The Halton sequence is somewhat out of favor compared to digital nets presented in §15.7 as well as the lattice methods in Chapter 16. It remains popular, in part because it is very easy to program. It can be used for any number of sample points n in any dimension d.

The Halton sequence is ***extensible in dimension***, meaning we can add a $d+1$'st dimension to our input points. Suppose for example that $f(\boldsymbol{x}) = f_d(\boldsymbol{x})$ follows a process through d time steps using $\boldsymbol{x} \in [0,1]^d$. If we later want to update our n values $y_{i,d} = f_d(x_{i,1}, \dots, x_{i,d})$ to get $y_{i,d+1} = f_{d+1}(x_{i,1}, \dots, x_{i,d+1})$

Variable	Range	Meaning
S_w	[150, 200]	wing area (ft^2)
W_fw	[220, 300]	weight of fuel in the wing (lb)
A	[6, 10]	aspect ratio
Λ	[−10, 10]	quarter-chord sweep (degrees)
q	[16, 45]	dynamic pressure at cruise (lb/ft^2)
λ	[0.5, 1]	taper ratio
t_c	[0.08, 0.18]	aerofoil thickness to chord ratio
N_z	[2.5, 6]	ultimate load factor
W_dg	[1700, 2500]	flight design gross weight (lb)
W_p	[0.025, 0.08]	paint weight (lb/ft^2)

Table 15.2: Variables and their ranges for the wing weight function.

we can use input points $x_{i,d+1}$ for $i = 1, \dots, n$. To be more concrete, if $f_{d+1}(x_{i,1}, \dots, x_{i,d+1})$ is of the form $g_{d+1}(y_{i,d}, x_{i,d+1})$ for some function g_{d+1} then the updates are simple and we don't even need to store the prior x_{ij} values. The Hammersley sequence is similarly extensible in dimension d, but it is not extensible in n.

15.6 Example: the wing weight function

The following function is a model for the weight of a wing of an aircraft

$$0.036 S_\mathrm{w}^{0.758} W_\mathrm{fw}^{0.0035} \Bigl(\frac{A}{\cos^2(\Lambda)}\Bigr)^{0.6} q^{0.006} \lambda^{0.04} \Bigl(\frac{100 t_\mathrm{c}}{\cos(\Lambda)}\Bigr)^{-0.3} (N_\mathrm{x} W_\mathrm{dg})^{0.49} + S_\mathrm{w} W_\mathrm{p}$$

taken from the virtual library of simulation experiments test functions of Surjanovic and Bingham (2013). The variables' meanings and ranges are given in Table 15.2. The virtual library contains code to implement this function as well as references to its origin. Note that Λ is given in degrees, from -10 to 10. It then lies between $\pm 10\pi/180$ radians, so $\cos(\Lambda)$ does not approach zero, and the wing weights are bounded.

We will study the average of this function over the 10-dimensional hypercube defined by its input variables' ranges. Our integrand on $[0,1]^{10}$ first scales each variable to its range and then computes the wing weight. One would not ordinarily seek the average weight of a randomly designed airplane wing. This example is useful for illustration because it has a scientific/engineering origin while not requiring access to specialized proprietary software to compute it. We will also ignore the fact that nine of the ten input variables can be integrated out to yield an elementary closed form. The exception is Λ.

We can apply plain Monte Carlo as well as quasi-Monte Carlo sampling to this integrand. Figure 15.8 shows cumulative averages of the wing weight function using the first 20,000 points of the Halton sequence in 10 dimensions. Only every 200'th point is plotted and we start plotting at $n = 1000$. The

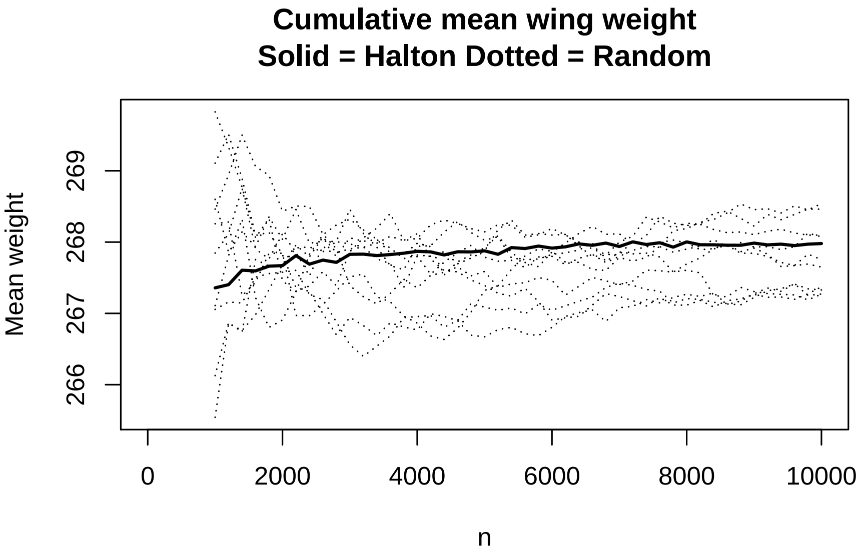

Figure 15.8: The horizontal axis is the sample size n from 1000 to 10,000 in steps of 200. The vertical axis is the cumulative average of the first n wing weight values. A solid line is used for the Halton sequence. Ten dotted lines show plain Monte Carlo.

Halton sequence in 10 dimensions does not have any especially good sample sizes, so little to no harm is done by using round numbers for n.

From Figure 15.8 it appears that the QMC rule is doing better than plain MC. The Halton cumulative values stabilize more quickly than the MC ones and they fluctuate less. Of course, we don't know the error because we don't know the true integral μ, and if we did know μ we would not be using QMC. By comparison, for MC, the fluctuations within curves are about $O(1/n)$ while those between curves are about $O(1/\sqrt{n})$, the same as our MC error. For QMC, we do not have an estimate of between curve error until we randomize as in Chapter 17. As noted above, the Koksma-Hlawka bound does not tell us how accurate $\hat{\mu}$ is and we cannot be sure whether $n = 20{,}000$ is large enough for the asymptotic rate to be relevant. Despite this doubt, we are left thinking that QMC is probably better in this instance, but we don't have evidence as strong as we would like much less a numerical estimate of error. This is a fundamental difficulty with QMC and it is the primary motivation for RQMC in Chapter 17.

The Halton cumulative means in Figure 15.8 appear to be drifiting up as n increases. A possible explanation is that the cumulative means of the inputs tend to approach 0.5 from below, and the wing weight function is monotone increasing in most of its inputs. Perhaps antithetic sampling with the Halton sequence would improve the estimation of mean wing weight. Exercise 15.9 asks you to investigate that possibility.

15.7 Digital nets and sequences

One problem with the Halton sequence is that as d increases, a larger value of n is required to get meaningful stratification. For $d = 5$, consecutive blocks of $2 \times 3 \times 5 \times 7 \times 11 = 2310$ points have a full 5-dimensional stratification. For $d = 10$, the product of the first 10 primes is 6,469,693,230, so that no 10-dimensional stratification appears until over 6 billion points have been used.

A second problem with the Halton sequence is that pairs (x_{i1}, x_{i2}) are stratified in consecutive blocks of 6 points while pairs (x_{i1}, x_{i3}) are stratified every 10 points and pairs (x_{i2}, x_{i3}) are stratified every 15 points. It would be better to have a rule where all $\binom{d}{2}$ pairs of variables can be stratified with the same value of n.

For large d it may be unrealistic to expect that we attain a full d-dimensional stratification. But it should be feasible to stratify all the two dimensional marginal distributions simultaneously using about d^2 points. For instance, using randomized orthogonal arrays (see §10.4) it is possible to stratify all $\binom{d}{s}$ s-dimensional coordinate projections using p^s points for any prime number $p \geqslant d - 1$.

What is needed is something like a Halton sequence with the same base b used for all dimensions. The solution is found in digital nets as described below. The digital nets we present are known as (t, m, s)-nets in base b, for integer parameters t, m, s and b, with s corresponding to the dimension of the space for $\boldsymbol{x}$. Usually $s = d$, that is we sample on a (t, m, d)-net. It is useful to let s differ from d, because there are ways to use an s-dimensional set of points while solving a d-dimensional problem. For example, the higher order nets in §15.12 as well as Latin supercube sampling in §17.9 use $s \neq d$.

Let $d \geqslant 1$ and $b \geqslant 2$ be integers. An ***elementary interval in base b*** is a subinterval of $[0, 1)^s$ of the form

$$E = \prod_{j=1}^{s} \Bigl[\frac{c_j}{b^{k_j}}, \frac{c_j + 1}{b^{k_j}}\Bigr)$$

for integers k_j and c_j, with $k_j \geqslant 0$ and $0 \leqslant c_j < b^{k_j}$. For $d = 1$, these are the usual sort of interval.

Figure 15.9 shows some elementary intervals in base $b = 5$ and dimension $s = 2$. In the upper left corner we have the entire unit square $[0, 1)^2$ which is, trivially, an elementary interval. The more interesting ones are those that impose some restrictions on one or more components of $\boldsymbol{x}$. We say that E is ***genuinely r-dimensional*** if $k_j > 0$ holds for at least r of the indices $j = 1, \dots, s$. Elementary intervals in base b are also called b-ary boxes, b-adic intervals, or cells.

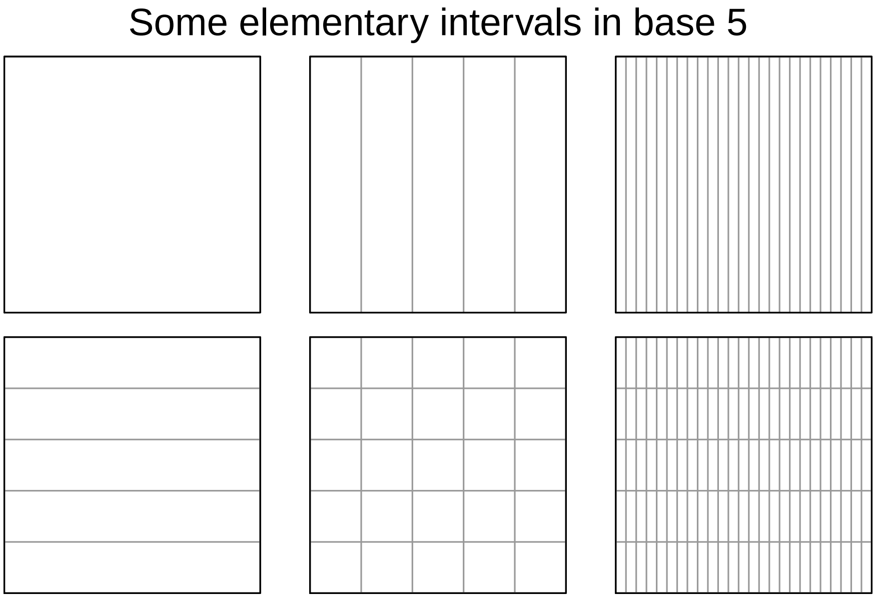

Figure 15.9: Each panel shows the unit square divided into elementary intervals in base 5. Panels in the left, middle and right columns are divided into 1, 5, and 25 vertical strips respectively. Panels in the top and bottom rows are divided into 1 and 5 horizontal strips respectively.

Definition 15.4. Let $m \geqslant 0$, $b \geqslant 2$ and $s \geqslant 1$ be integers. The sequence $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{b^m} \in [0,1)^s$ is a ***(0, m, s)*-net in base *b*** if every elementary interval E in base b of volume b^{-m} contains exactly 1 of the points $\boldsymbol{x}_i$.

Figure 15.10 shows some $(0, m, 2)$-nets in base 5. The number of elementary intervals balanced by a net can be much larger than n. The $(0, 3, 2)$-net in Figure 15.10 shows the first two dimensions of a $(0, 3, 5)$-net in base 5. For each vector of scales $(k_1, \dots, k_5)$ with $k_j \geqslant 0$ and $\sum_{j=1}^5 k_j = 3$, there are 125 rectangular cells of volume 1/125 in $[0,1)^5$ that each contain exactly 1 of the 125 points. Some combinatorial arguments show that there are 35 such tilings, and so $n = 125$ points of the net manage to balance $35 \times 125 = 4375$ cells of volume 1/125. Of these, only $5 \times 125 = 625$ would have been balanced in a Latin hypercube sample. The method of control variates §8.9 can be used to take account of known stratum volumes by introducing regression coefficients. But it would be difficult to use 4375 control variate regression parameters with only $n = 125$ data points. As m increases, the number of elementary intervals balanced grows more quickly than $n = b^m$ does.

The very strong multiple stratification that $(0, m, s)$-nets have is not always possible. For some choices of m, s and b, no such net exists. By weakening the criterion somewhat, more constructions become available.

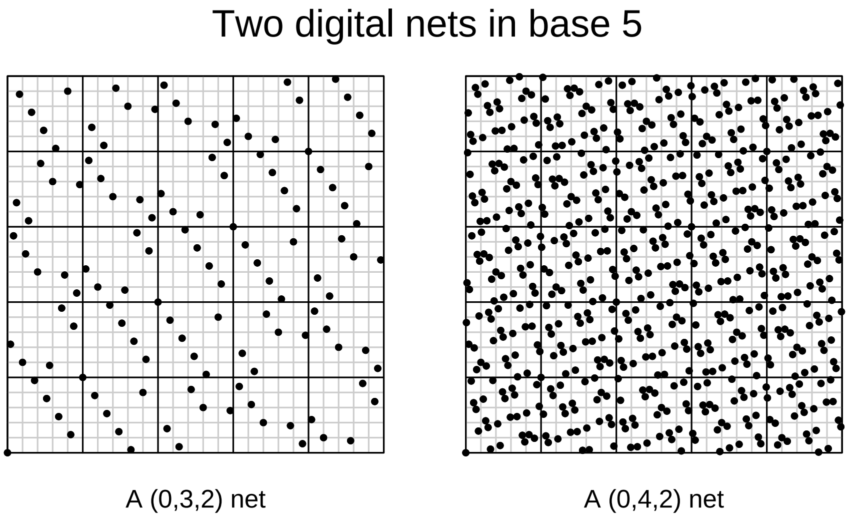

Figure 15.10: This figure shows two digital nets in the unit square in base 5. The one on the left has 125 points. The one on the right has 625 points. Dark reference lines 1/5 apart and light ones 1/25 apart show boundaries of some elementary intervals.

Definition 15.5. Let $m \geqslant t \geqslant 0$ be integers. The sequence $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{b^m} \in [0,1)^s$ is a ***(t,m,s)-net in base b*** if every elementary interval in base b of volume b^{t-m} contains exactly b^t points of the sequence.

Cells with volume b^t/n contain exactly b^t of the n sample points, matching their proportion of the volume of $[0,1)^s$. Smaller values of t imply better equidistribution. The upper limit on t is from the trivial case $t = m$, which only states that all points of the sequence are in $[0,1)^s$. A (t,m,s)-net in base b is ordinarily a $(t+1,m,s)$-net in the same base. The only exceptions are from cases where t is at the upper limit m and so cannot be raised. A ***strict*** (t,m,s)-net in base b is one that is not also a $(t-1,m,s)$-net in base b. Digital nets have low discrepancy:

Proposition 15.2. *The star discrepancy of a (t,m,s)-net in base b with $m > 0$ satisfies*

$$D_n^* \leqslant B(s,b) b^t \frac{(\log n)^{s-1}}{n} + O((\log n)^{s-2})$$

where

$$B(s,b) = \begin{cases} \Bigl(\dfrac{b-1}{2\log b}\Bigr)^{s-1} & s=2, \quad \text{or } b=2,\ s=3,4 \\ \dfrac{1}{(s-1)!}\Bigl(\dfrac{\lfloor b/2 \rfloor}{\log b}\Bigr)^{s-1} & \text{otherwise.} \end{cases}$$

Proof. This is Theorem 4.10 of Niederreiter (1992b). □

The (t, m, s)-nets are finite sequences. There is an extensible version of them as follows.

Definition 15.6. For $t \geqslant 0$, the infinite sequence $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots \in [0, 1)^s$ is a ***(t, s)*-sequence in base *b*** if for all $k \geqslant 0$ and $m \geqslant t$ the sequence $\boldsymbol{x}_{kb^m+1}, \dots, \boldsymbol{x}_{(k+1)b^m}$ is a (t, m, s)-net in base b.

A (t, s)-sequence is really an astonishing object. It is the concatenation of an infinite sequence of (t, m, s)-nets for any $m \geqslant t$. Those nets can be grouped into blocks of b consecutive ones. Each such block is a $(t, m + 1, s)$-net. Similarly, those $(t, m + 1, s)$-nets are nested within $(t, m + 2, s)$-nets within $(t, m + 3, s)$-nets and so on. As n increases through powers of b, the volume of balanced elementary intervals falls off as $b^{t-m} = b^t/n$ and their number increases rapidly.

The construction and analysis of digital nets and sequences is a very specialized topic. We will look at the properties and algorithms for some nets, but not delve into how they are constructed, apart from §15.10 which gives an elementary example.

The ***Faure sequences*** are $(0, s)$-sequences in base p, where $p \geqslant s$ is a prime number. An early implementation of the Faure sequence is in Fox (1986). The ***Faure net*** is a $(0, m, s)$-net in base p obtained as the first p^m points of the Faure sequence. The nets in Figure 15.10 are leading subsequences of Faure's $(0, 5)$-sequence in base 5. The Hammersley device of adding one equispaced variable also works for Faure's $(0, m, s)$-net allowing the construction of a $(0, m, p+1)$-net in base p for prime p.

Nets from the Faure sequences have a disadvantage when d is large. We need p to be a prime number at least equal to d (or $d - 1$ if using the Hammersley device). We may use the first d components of the base p Faure sequence, but that sequence balances no genuinely 2-dimensional elementary intervals unless n is a multiple of $p^2 \geqslant d^2$. If n is much below p^2, then some two dimensional projections of the Faure points will be very unevenly sampled. The appearance is quite similar to stripes that we see in Figure 15.6 for the Halton sequence projected on the j'th and k'th variables when $n/(p_j p_k)$ is somewhat smaller than 1.

Even with $n = p^2$, there can be bad higher dimensional projections. For example, the first 121 points of the Faure sequence in base 11 have some strange projections. From Figure 15.11 we see that $x_{i4} + x_{i6} - x_{i10} + x_{i11}$ takes on only 3 distinct values -1, 0 and 1, for $1 \leqslant i \leqslant 121$. As $\boldsymbol{x}$ varies through the unit cube, this projection takes values from -2 to 2 (not uniformly distributed) and so the sampled values are not only clustered but are also confined to a central subregion. There are other undesirable projections and some pairs of them reveal very structured patterns.

The first multidimensional digital sequences to be constructed were those of Sobol' (1967). He called them LP_τ squences but now they are more more widely known as ***Sobol' sequences***. They are (t, s)-sequences in base 2. Here $t = t_s$ is a non-decreasing function of s. The first few values are given in

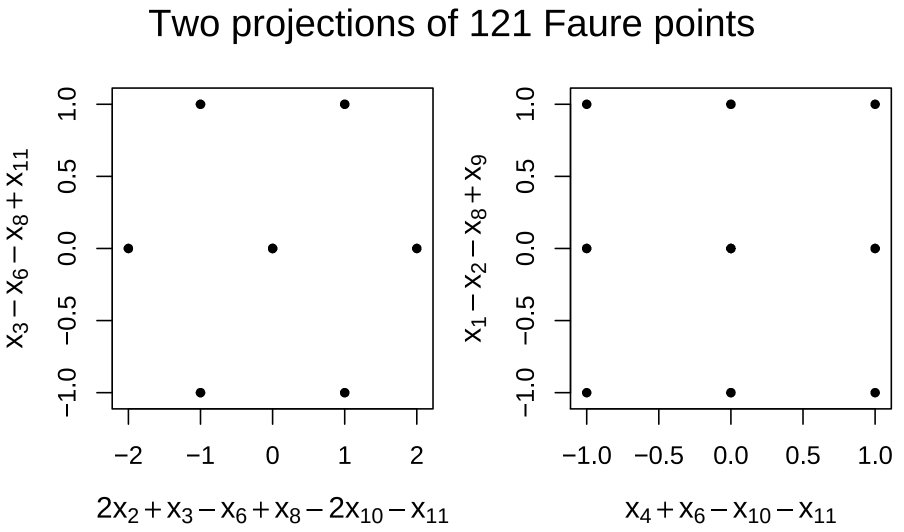

Figure 15.11: This figure shows two projections of the first 121 points of the 11-dimensional Faure sequence in base 11. In the left panel, there are 61 points at the center and 10 in each of the other sites. In the right panel, 57 points project to the origin, 4 points project to each corner, and 12 points project to the center of each side.

Table 15.3. The Sobol' construction for dimension $s + 1$ is obtained by adding the $s + 1$'st variable to the points of the Sobol' construction for dimension s. That is, Sobol' sequences are extensible in dimension. The earlier dimensions are constructed to have better equidistribution properties than the later ones. When we are able to order the inputs to f from most important to least, then we should use the first components of the Sobol' points on the most important inputs to f.

A (t, m, s)-net in base 2 can be formed from the first $n = 2^m$ points of Sobol's (t, s)-sequence. Such nets are not necessarily strict (t, m, s)-nets. The value of t can be lower for a net than the sequence it came from. For each $j = 1, \dots, s$, the points $\{x_{1j}, \dots, x_{2^m j}\} \subset [0, 1)$ of a Sobol' net are in fact a $(0, m, 1)$-net in base 2. That is, the Sobol' points have very uniform univariate projections. The Sobol' points can have some bad 2 dimensional projections. Bad projections of Sobol' points have quite a different appearance than bad projections of Halton or Faure points. Figure 15.13 shows some of them, based on the code from Bratley and Fox (1988).

There are multiple implementations of Sobol's idea and they differ in which projections are problematic, but they generally have more uniformity in projections of lower numbered dimensions. Because Sobol's points are defined in base 2, some of the implementations exploit bit level operations to gain greater speed.

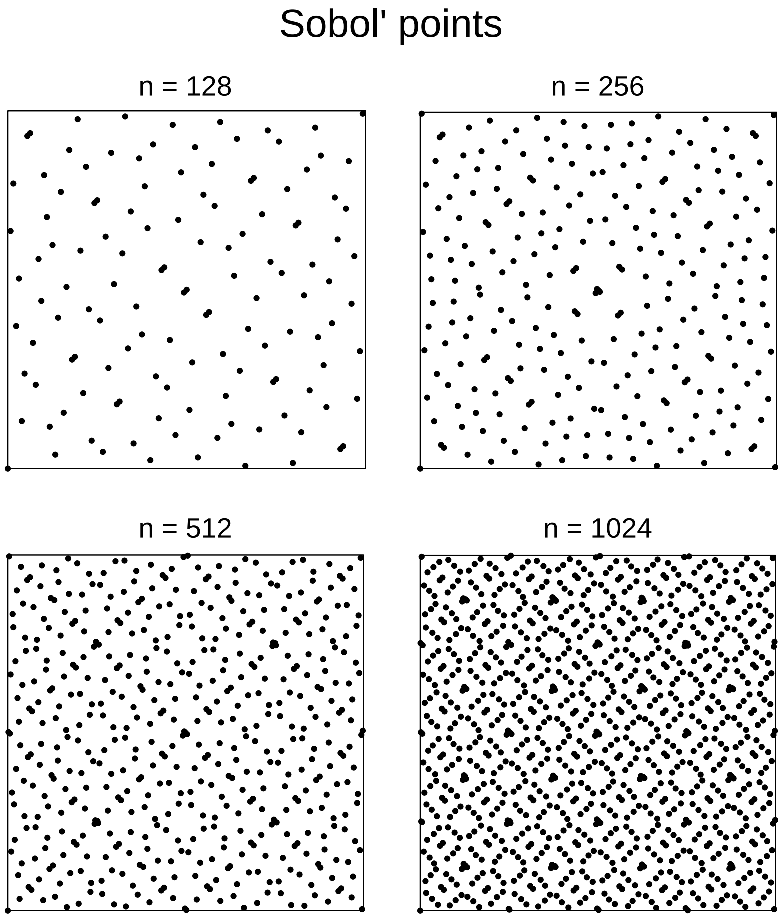

Figure 15.12: Points (x_{i1}, x_{i2}) for $i = 1, \dots, n$ of a Sobol' sequence.

There are many versions of Sobol's construction differing in what are called 'direction numbers'. The points in Figure 15.13 use direction numbers from Bratley and Fox (1988). Those provide Sobol' sequences for dimensions up to 40. A greatly expanded set of direction numbers going to much higher dimensions and paying attention to two dimensional projections has been produced by Joe and Kuo (2008). They give 21201 as the 'target dimension' of their searches for direction numbers. Figure 15.14 shows greatly improved projection for $x_{i,31}$ versus $x_{i,26}$ that was problematic in Figure 15.13. It also includes two of the subjectively worst projections of the first 1024 points for $\boldsymbol{x}_i \in [0,1]^{40}$. Those problematic projections fill in shortly after, with a complementary set of points placed in the gaps. This takes place at sample sizes that are still not large for 40-dimensional sampling. Sobol' et al. (2011) provide direction numbers for up

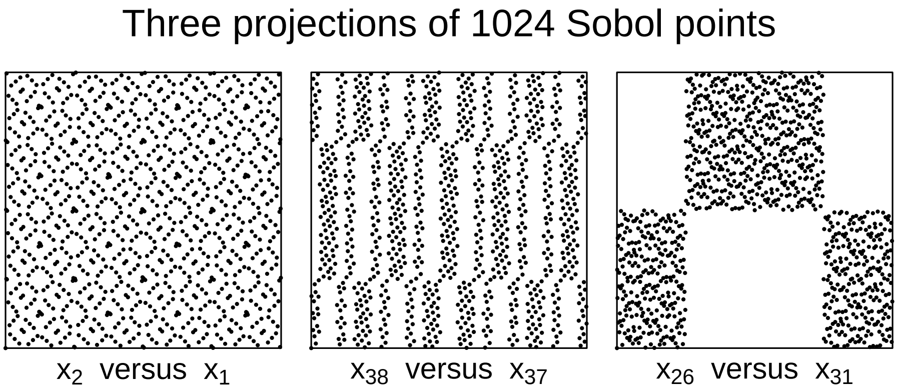

Figure 15.13: This figure shows three projections of the first 1024 points of the Sobol' sequence in $[0,1]^{40}$ using direction numbers from Bratley and Fox (1988). The left panel shows a very good projection of the first two components. The middle panel shows shows a less satisfactory projection and the right panel shows one with a serious flaw (that disappears when $n = 2^{14}$).

to 65,536 dimensions and cite several other published papers providing direction numbers.

The lead constant in the discrepancy bound for digital nets used to be much better, for large s, than that for the Halton and Hammersley sequences. That was all changed by Atanassov (2004) who sharpened the bounds for those sequences. He reduced the upper bounds on their leading constants by a factor of about $s!$. The sequences themselves did not change, and it is possible that a sharper bound could yet be found for digital nets.

Digital sequences are extensible, though we should not extend them one point at a time. If we use $n = b^m$ points from a (t, s)-sequence then (for $m \geqslant t$) all elementary intervals of volume b^{t-m} are balanced. The next sample size that retains all the balance we had at $n = b^m$ is $n' = 2b^m$.

If we increase n along a sequence of values of the form λb^m, where $1 \leqslant \lambda < b$ and $m \geqslant t$, then any elementary interval that was balanced at some value of n remains balanced for all future values of n. The first $n = \lambda b^m$ points of the (t, s)-sequence are (when $m \geqslant t$) equidistributed over the same set of elementary intervals that a (t, m, s)-net is. For $1 < \lambda < b$, those points do not form a (t, m, s)-net because λb^m is not a power of b. A second equidistribution property of $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{\lambda b^m}$ is as follows: no elementary interval of volume b^{t-m-1} has more than b^t points of the sequence. This holds because such an elementary interval has only b^t points of the first b^{m+1} points of the (t, s)-sequence.

Definition 15.7. Let λ, t, m, s, b be integers with $s \geqslant 1$, $m \geqslant t \geqslant 0$, $b \geqslant 2$ and $1 \leqslant \lambda < b$. A sequence $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{\lambda b^m} \in [0, 1)^s$ is called a ***$(\boldsymbol{\lambda}, \boldsymbol{t}, \boldsymbol{m}, \boldsymbol{s})$-net in base*** b if every elementary interval in base b of volume b^{t-m} contains λb^t points

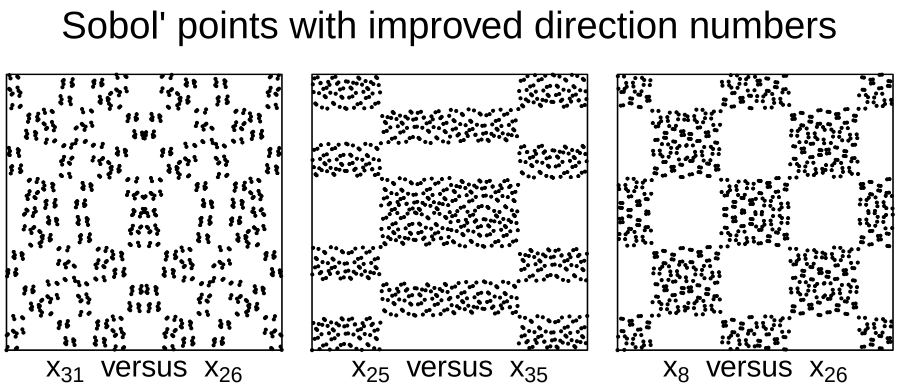

Figure 15.14: The first panel shows improved projection for $x_{i,31}$ versus $x_{i,26}$ using projection numbers of Joe and Kuo (2008). The next two panels show subjectively poor projections of those points. Most projections are much better. The holes in the first panel 'fill in' when $n = 4096$. The second and third ones fill in for $n = 2048$.

Dimension s	1	2	3	4	5	6	7	8	9	10
Sobol's t	0	0	1	3	5	8	11	15	19	23
Niederreiter-Xing's t	0	0	1	1	2	3	4	5	6	8

Table 15.3: This table shows the quality parameter t for Sobol' and Niederreiter-Xing (t, s)-sequences in base 2, where $s \leqslant 10$.

of the sequence and no elementary interval in base b of volume b^{t-m-1} contains more than b^t points of the sequence.

The smallest known values of t for digital nets come from a construction of Niederreiter and Xing (1996). Pirsic (2002) describes a computer implementation. Table 15.3 shows some of the resulting t values for the version in base 2.

The Niederreiter-Xing nets and sequences (NX-nets and NX-sequences below) have superior t parameters that are close to known lower bounds for t as a function of dimension s and base b. They are not as widely used as the Sobol' points. In some empirical comparisons, they do not seem to give much more accurate results than other methods. For example, see Hong and Hickernell (2003). Part of the reason is that the high t value for Sobol' sequences is somewhat misleading. Sobol' nets (finite n) usually have better t parameters than their corresponding infinitely long sequences. Moreover, lower dimensional projections of a net can have smaller t values than the net itself. These considerations may play more strongly in favor of the Sobol' sequences than the NX sequences.

If the t parameter of projections really explains why NX-sequences do not attain greatly improved accuracy, then it may be that NX-nets will prove to be superior on the very hardest fully d-dimensional problems as described in the project in Exercise 15.13.

15.8 Effect of projections

When investigating QMC points, we often consider their one, two, and three-dimensional coordinate projections, that is, their marginal distributions. The bivariate projections are most frequently investigated because we usually know that the univariate projections are very good, and bivariate projections are easier to investigate than the trivariate projections. In general, the lower the dimension we project points into, the better the equidistribution. It is easy to see, for example, that $D_n^*(\boldsymbol{x}_{1,1:(d-1)},\dots,\boldsymbol{x}_{n,1:(d-1)}) \leqslant D_n^*(\boldsymbol{x}_{1,1:d},\dots,\boldsymbol{x}_{n,1:d})$. Furthermore, the asymptotic bounds on D_n^* attain more favorable rates in low dimensions than in high.

The role of coordinate projections can be understood through the ANOVA decomposition of f (see Appendix §A). We write

$$f(\boldsymbol{x}) = \sum_{u\subseteq\{1,\dots,d\}} f_u(\boldsymbol{x}) \tag{15.18}$$

where f_u depends only on the components x_j with $j \in u$. The component $f_\varnothing$ is a constant function, $f_\varnothing(\boldsymbol{x}) = \mu$, which gets correctly averaged over any sample. The other f_u integrate to 0. Therefore the QMC error is

$$\begin{aligned} |\hat\mu - \mu| &= \Bigl|\sum_{u\neq\varnothing}\frac{1}{n}\sum_{i=1}^n f_u(\boldsymbol{x}_i)\Bigr| \leqslant \sum_{u\neq\varnothing}\Bigl|\frac{1}{n}\sum_{i=1}^n f_u(\boldsymbol{x}_i)\Bigr| \\ &\leqslant \sum_{u\neq\varnothing} D_n^*(\boldsymbol{x}_{1u},\dots,\boldsymbol{x}_{nu})V_{\mathrm{HK}}(f_u). \end{aligned} \tag{15.19}$$

Now let $|u|$ denote the cardinality of u. In examples, it is common to find that subsets u with large $|u|$ have effects f_u that are so small that they contribute little to the sum in (15.19). Although f is of nominal dimension d, it may be closely approximated by a sum of functions of much lower dimension. It is in this sense of lower effective dimension than d. See §17.2.

For subsets u of small cardinality, the effects f_u may be large, but our points $\boldsymbol{x}_i$ may have low dimensional projections $\boldsymbol{x}_{iu}$ with small discrepancy. The discrepancy bound for projected points is $O(\log(n)^{|u|-1}/n)$ not $O(\log(n)^{d-1}/n)$. For instance, if $\boldsymbol{x}_i$ form a (t,m,s)-net in base b then $\boldsymbol{x}_{iu}$ form a $(t',m,|u|)$-net in base b too, where t' is at most t and could be lower. Even with $t'=t$, if $m > t+|u|$ then the $\boldsymbol{x}_{iu}$ have some nontrivial stratification over elementary intervals while the $\boldsymbol{x}_i \in [0,1]^d$ may fail to balance any d-dimensional elementary interval smaller than $[0,1]^d$ itself. When $\boldsymbol{x}_{iu}$ have small discrepancy then the term for u in (15.19) is small if $V_{\mathrm{HK}}(f_u)$ is not large. In the best case, every term on the right of (15.19) is small because at one or the other of its factors is small. Then QMC delivers an estimate for our high dimensional problem with the accuracy we would have expected for a lower dimensional one.

It is not a theorem that f must be dominated by low dimensional parts that are amenable to QMC sampling. It is a common though not universal empirical finding and it provides the best use case for QMC methods. Sets of such f can be described through the weighted spaces in §7.7. QMC methods can be customized to a specific weighted space; see the end notes on polynomial lattice rules. Equation (15.18) decomposes f into ANOVA components but there are other such decompositions in §xxx and the argument behind (15.19) applies to any of them.

15.9 Example: synthetic integrands

We know from Theorem 15.5 that QMC is much better than Monte Carlo when n is large enough and f is of bounded variation in the sense of Hardy and Krause. Unfortunately, the proven bounds are hard to apply for specific n and f. Here we look at some examples with known integrals to get a sense of whether the advantage of QMC applies to realistic sample sizes n, or is purely asymptotic.

Numerical examples serve as spot checks on the theory. To investigate every important issue numerically would require an unmanageable number of examples. Instead, we consider a small number of examples seeking qualitative insights. In some examples, the ANOVA representation (15.19) makes it reasonable that QMC should do well. In others, such as functions of unbounded variation, poor performance is quite plausible but by no means assured.

In the examples here, we see QMC beating MC by enormous factors when the function is smooth and low dimensional. Also, MC does much better than QMC on some very high dimensional functions. For a high dimensional function dominated by smooth low dimensional ANOVA components, QMC holds a strong advantage.

It is convenient to take f to be a product of univariate functions. Let $\boldsymbol{g} = (g_1, \dots, g_d)$ be a vector of functions on $[0,1]$ satisfying $\int_0^1 g_j(x)\,\mathrm{d}x = 0$ and $\int_0^1 g_j(x)^2\,\mathrm{d}x = 1$. For $\boldsymbol{\beta} \in \mathbb{R}^d$ define

$$f(\boldsymbol{x}) = f_{\boldsymbol{\beta},\boldsymbol{g}}(\boldsymbol{x}) = \prod_{j=1}^d (1 + \beta_j g_j(x_j)). \tag{15.20}$$

We know that $\mu = \int f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 1$ and so it is easy to study the error of a QMC rule. Similarly

$$\sigma^2 = \int f(\boldsymbol{x})^2\,\mathrm{d}\boldsymbol{x} - 1 = \prod_{j=1}^d (1 + \beta_j^2) - 1$$

is known, so the Monte Carlo RMSE $\sigma/\sqrt{n}$, is available for comparison, without having to actually do any MC sampling. What makes product functions very convenient is that their entire ANOVA decomposition (Appendix §A) is available:

$$f(\boldsymbol{x}) = \sum_{u\subseteq\{1,\dots,d\}} f_u(\boldsymbol{x}) \quad \text{where} \quad f_u(\boldsymbol{x}) = \prod_{j\in u} \beta_j g_j(x_j),$$

which includes $f_\varnothing(\boldsymbol{x}) = 1$ by convention. For $u \neq \varnothing$, the variance of f_u is $\sigma_u^2 = \prod_{j\in u} \beta_j^2$.

Increasing the magnitude of $\boldsymbol{\beta}$ makes the higher dimensional ANOVA components relatively larger and makes the quadrature problem harder. Also, individual variables x_j with larger values of $|\beta_j|$ are more important than the others.

We begin with an easy problem taking $g_j(x) = \sqrt{12}(x-1/2)$ and $\beta_j = 1/5$ for $j = 1,\dots,5$, leading to

$$f_1(\boldsymbol{x}) = \prod_{j=1}^{5}\Bigl(1 + \frac{\sqrt{12}}{5}(x-1/2)\Bigr).$$

This choice of $\boldsymbol{\beta}$ is one where even Latin hypercube sampling, which only stratifies the one dimensional projections, makes a meaningful improvement. From the ANOVA decomposition we obtain the best additive approximation to f,

$$f_{\text{add}}(\boldsymbol{x}) = \sum_{|u|\leqslant 1} f_u(\boldsymbol{x}) = 1 + \sum_{j=1}^{5} \beta_j g_j(x_j).$$

This additive approximation has variance $\sigma_{\text{add}}^2 = \sum_{j=1}^5 \beta_j^2$. Latin hypercube sampling (§10.3) has variance $\sigma_{\text{lhs}}^2/n + o(1/n)$ where $\sigma_{\text{lhs}}^2 = \sigma^2 - \sigma_{\text{add}}^2 = \prod_{j=1}^5(1+\beta_j^2) - 1 - \sum_{j=1}^5 \beta_j^2$. For f_1, Latin hypercube sampling reduces the variance by a factor of about 13.

Figure 15.15 shows results for the Halton, Faure and Sobol' sequences with f_1 for $n \leqslant 5^4 = 3125$. They all have errors smaller than $\sigma/\sqrt{n}$. The Halton sequence has an error comparable to the Latin hypercube sampling RMSE (for these n) while the other sequences yield smaller errors.

To judge the attained convergence rate for QMC, it is better to look at errors on a logarithmic scale. One difficulty with the logarithmic scale is that when two consecutive errors $\hat\mu_n - \mu$ and $\hat\mu_{n+1} - \mu$ have opposite signs, one or both is typically very close to zero. We can't know in practice when our error has changed sign and so, when looking at errors on a log scale we should ignore a few stray values that are far below the others; they don't correspond to actionable information. We can also mitigate this difficulty by plotting $|\hat\mu_n - \mu|$ for every k'th value of n.

Figure 15.16 shows the QMC errors for f_1 on a logarithmic scale for n up to $5^6 = 15{,}625$. The Halton sequence makes steady progress, showing a rate better than $n^{-1/2}$ though not, on this range of sample sizes, as good as n^{-1}. It eventually gets better than Latin hypercube sampling and by $n = 15{,}625$ it

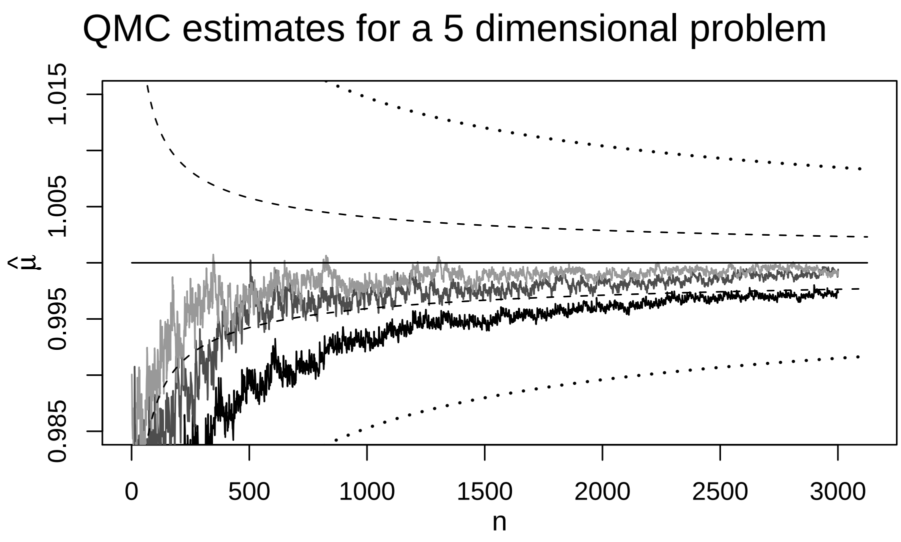

Figure 15.15: This figure shows quasi-Monte Carlo estimates $\hat{\mu}$ of $\mu = \int_{[0,1]^5} f_1(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ for the example function f_1. The horizontal axis has the sample size n over the range from $5^2 = 125$ to $5^4 = 3125$. From top to bottom, lightest gray to darkest, the results are for the Sobol' sequence, the Faure sequence (base 5) and the Halton sequence. The horizontal reference line is at the true mean $\mu = 1$. The dotted reference curves are at μ plus or minus one Monte Carlo standard deviation. The dashed curves are at μ plus or minus one Latin hypercube standard deviation.

shows an error between 1/10 and 1/100 of $\sigma/\sqrt{n}$. The Faure sequence attains better results than the Halton sequence. It makes uneven progress resembling stair steps. Its efficiency increases greatly as n approaches a power of 5, where a new set of elementary intervals become balanced. In this example, both Faure and Sobol' sequences perform close to the theoretical $1/n$ rate when n is a power of their respective bases. The Sobol' sequence performs better than the Faure sequence between powers of its base. Its error changes sign numerous times near the end of the run with the zero crossings complicating a logarithmic plot of the errors.

It is interesting to consider the effects of dimension on accuracy for this example. We can inspect the purely 5-dimensional component $(0.2\sqrt{12})^5 \prod_{j=1}^{5}(x_j - 1/2)$ of f_1 and see how close its average is to 0. The results for $n \leqslant 15{,}625$ (not plotted) are that the Halton sequence makes absolute errors that fluctuate around $\sigma/\sqrt{n}$. The Faure sequence has errors generally above $\sigma/\sqrt{n}$. The Sobol' sequence has errors, at powers of 2, that trend more steeply downwards than $1/\sqrt{n}$, ending up below $\sigma/\sqrt{1000n}$. The Sobol' sequence in $[0,1]^5$ for $n \leqslant 15{,}625$ has more thorough 5 dimensional stratification than either of the other two sequences. This brings it better performance on the highest ANOVA

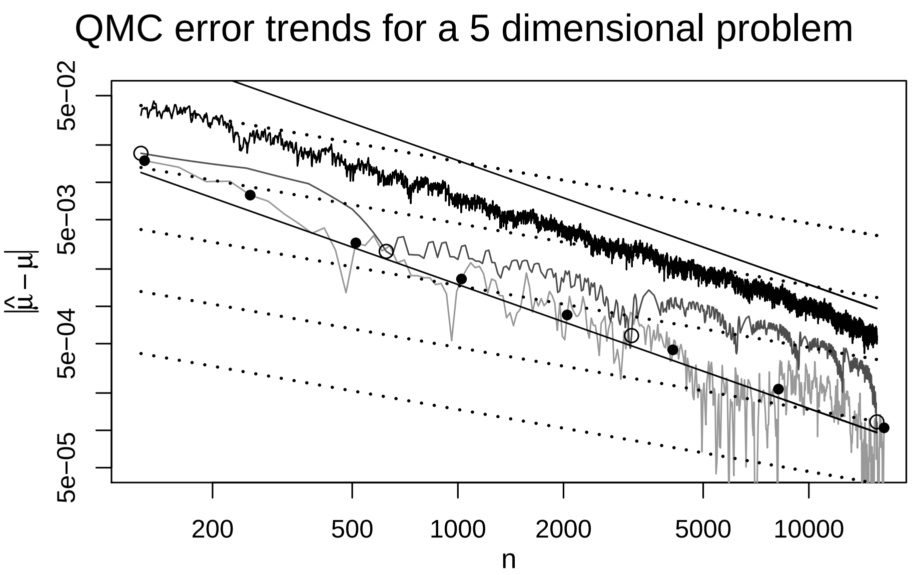

Figure 15.16: This figure shows quasi-Monte Carlo errors $|\hat{\mu}-\mu|$ for the example function f_1. The horizontal axis has the sample size n from $5^2 = 125$ to $5^6 =$ 15,625. From top to bottom, darkest gray to lightest, the results are for the Halton sequence, the Faure sequence (at multiples of 25) and the Sobol' sequence (at multiples of 32). The solid reference lines are proportional to $1/n$, the approximate asymptotic convergence rate for QMC. The dotted reference lines are the Monte Carlo RMSEs for sample sizes n, $10n$, $100n$, $1000n$ and $10{,}000n$. The open dots show Sobol' errors when n is a power of 2. The solid dots show Faure errors when n is a power of 5. At the final sample size the QMC errors are just below $10^{-3}\sigma/\sqrt{n}$.

component of f_1. The Faure sequence remains competitive on f_1 because the highest ANOVA component has small magnitude.

For a five dimensional and very smooth integrand, it would be possible to use a quadrature rule based on a 5-dimensional grid. QMC is easier to use than such 5-dimensional product rules. For example, the Sobol' sequence works well at values of n that are powers of 2. The simplest product rules would require n to be the 5'th powers of an integer, so while usable they would be quite cumbersome.

Next we consider a function f_2 in 25 dimensions, where product rules are completely infeasible. We suppose this time that each successive component of $\boldsymbol{x}$ is less important than the previous one. We take $\beta_j = 1/(2j)$ and retain $g_j(x) = \sqrt{12}(x - 1/2)$. That is

$$f_{2,d}(\boldsymbol{x}) = \prod_{j=1}^{d}\biggl(1 + \frac{\sqrt{3}}{j}(x_j - 1/2)\biggr).$$

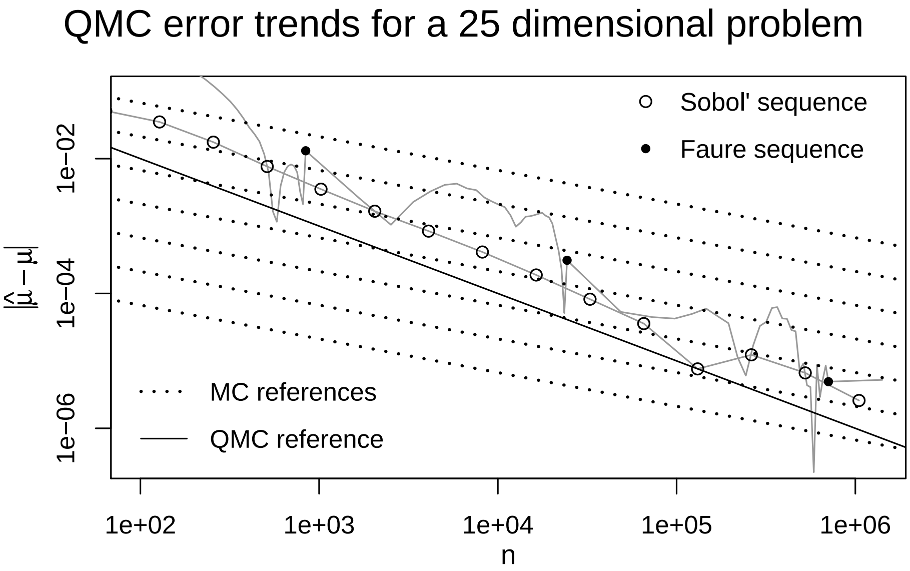

Figure 15.17: This figure shows quasi-Monte Carlo errors $|\hat{\mu}-\mu|$ for the example function f_2. The horizontal axis has the sample size n from 1 to almost 2 million. Results from the Sobol' sequence are plotted at $n = 2^k$ for $k = 0, \dots, 20$. Results from the Faure sequence in base 29 are plotted for $n = \lambda 29^k \leqslant 2\times 29^4$ for integer n and λ. A solid point is shown for $n = 29^k$. The dotted reference lines are at $\sigma/\sqrt{10^k n}$ for (top to bottom) $0 \leqslant k \leqslant 6$. The solid reference line is $1/n$.

This function has finite variance for any $d < \infty$ because for $\boldsymbol{x} \sim \mathbf{U}(0,1)^d$,

$$\log(\mathbb{E}(f_{2,d}(\boldsymbol{x})^2)) = \sum_{j=1}^{d} \log\Bigl(1 + \frac{1}{4j^2}\Bigr) \leqslant \frac{1}{4}\sum_{j=1}^{\infty}\frac{1}{j^2} = \frac{\pi^2}{24}. \tag{15.21}$$

That is $\sigma^2 \leqslant \exp(\pi^2/24) - 1 \doteq 0.51$. The small magnitude of this variance does not make f_2 unrealistic, because effective (or otherwise) QMC methods for integrating f_2 are similarly effective on cf_2 for $c \neq 0$. In particular, their relative error $|\hat{\mu}-\mu|/\mu$ is unaffected by c as is the comparison between QMC and MC. The variance at $d = 25$ is roughly 90% of the variance bound (15.21). For $d = 25$, Latin hypercube sampling reduces the Monte Carlo variance by about 9.1 fold.

Figure 15.17 shows results for this function using the Sobol' sequence as well as the Faure sequence in base 29. The Sobol' sequence starts out with an error equal to about $\sigma/\sqrt{n}$ but turns the corner around $n = 100$ where the plot begins. It makes steady progress roughly in proportion to $1/n$ from then on. The Faure sequence has very large errors below $n = 100$, but we ordinarily would not contemplate using fewer than 100 points in 25 dimensions so that shortcoming is not serious. The Faure sequence at powers of 29 is nearly as good as the

Sobol' sequence. Between powers of 29, the Faure sequence is not steady, and makes some relatively large errors. It has one very small error, presumably a lucky outcome, near $n = 700{,}000$. The Sobol' sequence has the advantage here of using better equidistribution on the earlier and more important input variables.

In this 25-dimensional example, QMC is able to attain, for n near 10^6, errors comparable to the RMSE that Monte Carlo would have with n between 10^{10} and 10^{11}.

For 25 variables there are $2^{25}-1$ (over 33 million) ANOVA components that contribute to the function and so we can partition the error into that many parts. Table 15.4 illustrates some of them. For the highest order ANOVA component, the full 25 way interaction, the Sobol' sequence has an error of about 900 times the Monte Carlo standard deviation $\sigma_u/\sqrt{n}$. The Faure sequence has an error of about $1090\sigma_u/\sqrt{n}$.

In this example, the 25 factor interaction $\prod_{j=1}^{25} \beta_j g_j(x_j)$ is integrated with an error equal to roughly 1000 times the plain Monte Carlo RMSE. No fully 25-dimensional elementary interval was balanced by either set of QMC points, so perhaps we should not have expected them to be better than MC. For Faure points, that balance could not happen until $n = 29^{25} \approx 3.6 \times 10^{36}$. For base 2 sequences like Sobol's, the best available value of t, from the minT project (Schürer and Schmid, 2009), is 31. So a 25-dimensional elementary interval could be balanced by $n = 2^{31+25} \approx 7.2 \times 10^{16}$ points. From that same source, there do exist $(28, 53, 25)$-nets in base 2 which could balance a 25-dimensional elementary interval with $n = 2^{28+25} \approx 9.0 \times 10^{15}$ points. Even for d as low as 25, getting nontrivial d-dimensional stratification with these digital nets is unreasonably expensive.

This bad performance on that 25-dimensional interaction hardly matters because that highest interaction accounts for only about 1.32×10^{-33} of the variance of f. The interaction of the first 4 variables is relatively much more important. The Sobol' sequence makes an error about $10^{-9}\sigma_u/\sqrt{n}$ for this component while the Faure sequence error is about $1.67 \times 10^{-4}\sigma_u/\sqrt{n}$.

The Halton sequence was left out of this example. Exercise 15.11 is about implementing Halton points for this example.

In these product functions, the accuracy promised by QMC is attained at modest sample sizes for the low dimensional ANOVA components. We might expect good results for QMC when f is dominated by smooth low dimensional ANOVA components. We should not expect similar results for every function, not even every product function. A spiky product using functions such as $g_j(x) = \sqrt{50}(\mathbb{1}_{x<0.01} - \mathbb{1}_{x>0.99})$ will obviously require larger n to get good results. Similarly, we expect that highly oscillatory functions such as $g_j(x) = \sqrt{2}\sin(2K\pi x)$ with large $K > 0$ will require larger n before the QMC rate is observed.

We have used Latin hypercube sampling as a yardstick. To live up to its promise, QMC should at least be better than LHS. For product functions (15.20) with monotonic $g_j(x)$, we know that antithetic sampling will improve on plain

Variable		Sobol'		Faure	
subset u	σ_u^2/σ^2	Error	vs MC	Error	vs MC
First 1	3.45×10^{-1}	8.26×10^{-7}	1.69×10^{-3}	1.22×10^{-6}	2.06×10^{-3}
First 2	8.62×10^{-2}	8.80×10^{-11}	7.21×10^{-7}	2.11×10^{-10}	1.42×10^{-6}
First 3	1.44×10^{-2}	2.55×10^{-11}	1.25×10^{-6}	1.76×10^{-9}	7.09×10^{-5}
First 4	1.80×10^{-3}	2.68×10^{-15}	1.05×10^{-9}	5.16×10^{-10}	1.67×10^{-4}
First 5	1.80×10^{-4}	7.54×10^{-15}	2.97×10^{-8}	1.70×10^{-8}	5.50×10^{-2}
First 10	1.85×10^{-10}	1.33×10^{-14}	5.05×10^{-2}	1.26×10^{-13}	3.93×10^{-1}
First 15	1.61×10^{-17}	6.09×10^{-20}	2.67×10^{0}	1.24×10^{-19}	4.46×10^{0}
First 20	2.70×10^{-25}	2.23×10^{-26}	5.84×10^{1}	5.34×10^{-26}	1.15×10^{2}
Last 1	1.38×10^{-2}	3.30×10^{-8}	1.69×10^{-3}	4.90×10^{-8}	2.06×10^{-3}
Last 2	2.87×10^{-4}	2.27×10^{-15}	5.59×10^{-9}	7.02×10^{-13}	1.42×10^{-6}
Last 3	6.24×10^{-6}	2.44×10^{-19}	2.76×10^{-11}	7.64×10^{-13}	7.09×10^{-5}
Last 4	1.42×10^{-7}	6.75×10^{-21}	3.36×10^{-11}	4.08×10^{-14}	1.67×10^{-4}
Last 5	3.38×10^{-9}	2.01×10^{-17}	4.20×10^{-6}	3.20×10^{-13}	5.50×10^{-2}
Last 10	5.67×10^{-17}	5.74×10^{-21}	7.14×10^{-2}	3.84×10^{-20}	3.93×10^{-1}
Last 15	4.92×10^{-24}	2.26×10^{-26}	3.24×10^{0}	3.78×10^{-26}	4.46×10^{0}
Last 20	5.09×10^{-30}	4.14×10^{-31}	5.74×10^{1}	1.00×10^{-30}	1.15×10^{2}
All 25	1.32×10^{-33}	1.69×10^{-33}	9.00×10^{2}	2.50×10^{-33}	1.09×10^{3}

Table 15.4: QMC results for selected ANOVA components of the function $f_{2,25}$. The first column gives $u \subseteq \{1, 2, \dots, 25\}$. The second column shows σ_u^2/σ^2, the fraction of variance from the interaction u. The third column shows the error of the Sobol' sequence $|\hat\mu_u| = |(1/n)\sum_{i=1}^n f_u(\boldsymbol{x}_i)|$. The fourth column has $\sqrt{n}|\hat\mu_u|/\sigma_u$ for the Sobol' sequence. The next two columns give accuracy for the Faure sequence. The Sobol' data are for $n = 2^{20} = 1{,}048{,}576$. The Faure data are for $n = 29^4 = 707{,}281$.

Monte Carlo providing another yardstick. We can work out the antithetic sampling variance of such products and they take a simple form for functions like $g_j = \sqrt{12}(x - 1/2)$.

Proposition 15.3. *Let $f(\boldsymbol{x})$ have the product form* (15.20) *in which each function g_j is antisymmetric: $g_j(x) = -g_j(1-x)$. For an even number $n \geqslant 2$, let $\hat\mu_{\mathrm{anti}} = (1/n)\sum_{i=1}^{n/2}(f(\boldsymbol{x}_i) + f(\mathbf{1} - \boldsymbol{x}_i))$ where $\boldsymbol{x}_i \sim \mathbf{U}[0,1]^d$ for $i = 1, \dots, n/2$. Then*

$$\mathrm{Var}(\hat\mu_{\mathrm{anti}}) = \frac{2}{n}\sum_{k=1}^{\lfloor d/2\rfloor}\sum_{\substack{u\subseteq\{1,\dots,d\}\\|u|=2k}}\prod_{j\in u}\beta_j^2. \tag{15.22}$$

Proof. See Exercise 15.12. □

15.10 How digital constructions work

This section gives a simple illustration of the construction of a digital net. It can be skipped by readers who simply want to use those nets. For a complete account, see Dick and Pillichshammer (2010).

Digital nets are constructed by working with the base b expansion of the digits of integers. As a simple example, we can construct a $(0, m, 1)$-net in base b by using the first b^m points of the van der Corput sequence in base b.

To get the idea of how a multidimensional digital net can be constructed we look first at a small two dimensional example, a $(0, 4, 2)$-net in a prime base p, used by Dick and Pillichshammer (2010). Then we consider the more general setting.

We begin with the matrices

$$C^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad C^{(2)} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Now suppose that we construct another matrix placing the first k_1 rows of $C^{(1)}$ above the first k_2 rows of $C^{(2)}$, where $k_j \geqslant 0$ and $k_1 + k_2 = 4$. The resulting matrix is

$$C_{k_1,k_2} \equiv \begin{pmatrix} C^{(1)}_{1:k_1} \\ C^{(2)}_{1:k_2} \end{pmatrix}$$

where $C^{(j)}_{1:k}$ has the first k rows of $C^{(j)}$. In this small example we will need each C_{k_1,k_2} to be invertible in arithmetic modulo p.

Here we find that C_{k_1,k_2} is a permutation matrix: the product $C_{k_1,k_2}v$ reverses the order of the last k_2 elements of v. As a result C_{k_1,k_2} is its own inverse matrix. This can be seen by squaring C_{k_1,k_2} in arithmetic modulo p.

More generally, matrices whose leading rows can be extracted and reassembled into a combined matrix of sufficiently high rank are the crucial ingredient in digital net constructions.

Now we construct our digital net. For integers $i \geqslant 0$, write $i = \sum_{k=0}^{\infty} \mathsf{d}_k(i)p^k$ with the digit retrieval function of §15.5. For $0 \leqslant i < p^4$, the expansion of i requires at most 4 digits. We put them in a vector of length 4, writing

$$\widetilde{i} = \begin{pmatrix} \mathsf{d}_0(i) \\ \mathsf{d}_1(i) \\ \mathsf{d}_2(i) \\ \mathsf{d}_3(i) \end{pmatrix} \quad \text{for} \quad 0 \leqslant i < p^4.$$

Now let $\boldsymbol{y}_{i1} = C^{(1)}\widetilde{i}$ and $\boldsymbol{y}_{i2} = C^{(2)}\widetilde{i}$ in arithmetic modulo p. For our very simple example

$$\boldsymbol{y}_{i1} = \begin{pmatrix} \mathsf{d}_0(i) \\ \mathsf{d}_1(i) \\ \mathsf{d}_2(i) \\ \mathsf{d}_3(i) \end{pmatrix} \quad \text{and} \quad \boldsymbol{y}_{i2} = \begin{pmatrix} \mathsf{d}_3(i) \\ \mathsf{d}_2(i) \\ \mathsf{d}_1(i) \\ \mathsf{d}_0(i) \end{pmatrix}.$$

Finally, the point $\boldsymbol{x}_i = (x_{i1}, x_{i2})$ is made from the digits in $\boldsymbol{y}_{i-1}$ via

$$x_{ij} = \sum_{k=1}^{m} y_{(i-1)jk} p^{-k} \tag{15.23}$$

where $m = 4$ in our example.

By construction, $x_{ij} \in [0, 1)$ for $i = 1, \dots, p^4$ and $j = 1, 2$. Now let's check that $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{p^4}$ are a $(0, 4, 2)$-net in base p. Consider the elementary interval

$$E = \Bigl[\frac{c_1}{p^{k_1}}, \frac{c_1+1}{p^{k_1}}\Bigr) \times \Bigl[\frac{c_2}{p^{k_2}}, \frac{c_2+1}{p^{k_2}}\Bigr)$$

where $k_j \geqslant 0$ with $k_1 + k_2 = 4$ and $0 \leqslant c_j < p^{k_j}$. If every such E contains exactly one of the $\boldsymbol{x}_i$, then the $\boldsymbol{x}_i$ are a $(0, 4, 2)$-net in base p.

We need to solve for i such that $\boldsymbol{x}_i \in E$. If such i exists then we know it satisfies $c_1 \leqslant p^{k_1} x_{i1} < c_1 + 1$. That is $c_1 \leqslant \sum_{k=1}^{4} y_{(i-1)1k} p^{k_1-k} < c_1 + 1$. Therefore

$$c_1 = \sum_{k=1}^{k_1} y_{(i-1)1k} p^{k_1-k} = \sum_{k=0}^{k_1-1} y_{(i-1)1(k_1-k)} p^{k}.$$

In other words, the digits of c_1 are $\mathsf{d}_k(c_1) = y_{(i-1)1(k_1-k+1)}$ for $k = 0, \dots, k_1 - 1$. Notice that the order of the digits is reversed.

We require that i satisfies $y_{(i-1)1(k+1)} = \mathsf{d}_{k-k_1}(c_1)$ for $0 \leqslant k < k_1$. Taking account of the second dimension as well, the value i must also satisfy $y_{(i-1)2(k+1)} = \mathsf{d}_{k-k_2}(c_2)$ for $0 \leqslant k < k_2$.

For $0 \leqslant i < p^4$, let $\boldsymbol{y}_{i;k_1,k_2}$ be made up of the first k_1 elements of $\boldsymbol{y}_{i1}$ and the first k_2 elements of $\boldsymbol{y}_{i2}$. Then

$$\boldsymbol{y}_{i;k_1,k_2} \equiv \begin{pmatrix} \boldsymbol{y}_{i1,1:k_1} \\ \boldsymbol{y}_{i2:1:k_2} \end{pmatrix} = C_{k_1,k_2} \begin{pmatrix} \mathsf{d}_0(i) \\ \mathsf{d}_1(i) \\ \mathsf{d}_2(i) \\ \mathsf{d}_3(i) \end{pmatrix}.$$

We find i by solving

$$\boldsymbol{y}_{i;k_1,k_2} = \begin{pmatrix} \mathsf{d}_{k_1-1}(c_1) \\ \vdots \\ \mathsf{d}_0(c_1) \\ \mathsf{d}_{k_2-1}(c_2) \\ \vdots \\ \mathsf{d}_0(c_2) \end{pmatrix}$$

so that

$$\begin{pmatrix} \mathsf{d}_0(i) \\ \mathsf{d}_1(i) \\ \mathsf{d}_2(i) \\ \mathsf{d}_3(i) \end{pmatrix} = C_{k_1,k_2}^{-1} \begin{pmatrix} \mathsf{d}_{k_1-1}(c_1) \\ \vdots \\ \mathsf{d}_0(c_1) \\ \mathsf{d}_{k_2-1}(c_2) \\ \vdots \\ \mathsf{d}_0(c_2) \end{pmatrix}$$

in arithmetic modulo p. The solution exists because C_{k_1,k_2} is invertible. From the digits we recover the integer $i = \sum_{k=0}^{3} \mathsf{d}_k(i)p^k$. Now $\boldsymbol{x}_{i+1} \in E$ is the point we needed to find, and we have shown that $\boldsymbol{x}_i$ are a $(0, 4, 2)$-net in base p.

A general digital net construction in a prime base p starts with $s \geqslant 1$ matrices $C^{(1)}, \dots, C^{(s)} \in \{0, 1, \dots, p-1\}^{m\times m}$ for $m \geqslant 1$. Suppose that the matrix

$$C_{k_1,k_2,\dots,k_s} = \begin{pmatrix} C^{(1)}_{1:k_1} \\ C^{(2)}_{1:k_2} \\ \vdots \\ C^{(s)}_{1:k_s} \end{pmatrix}$$

containing the first $k_j \geqslant 0$ rows of $C^{(j)}$ always has rank at least $m - t$ when $\sum_{j=1}^{s} k_j = m$. Then we may construct a (t, m, s)-net in base p as follows:

1) place the base p digits of $i - 1$ into the vector $\widetilde{i}$,
2) for $j = 1, \dots, s$, multiply $\boldsymbol{y}_j = C^{(j)}\widetilde{i}$, in arithmetic modulo p, and,
3) for $j = 1, \dots, s$, form x_{ij} from digits of $\boldsymbol{y}_j$ as in equation (15.23).

When $t > 0$, the $m \times m$ matrices above have rank $m - t < m$. As a result we expect a t-dimensional space of solutions. Working in integers mod p that leads to p^t solutions corresponding to the p^t points that the (t, m, s)-net places in a given elementary interval.

A (t, m, s)-net requires s matrices of size m by m and it generates b^m points with m digits each. A (t, s)-sequence uses s matrices with infinitely many rows and columns both indexed from 1 to ∞. Only finitely many rows and columns are needed in practice because $n < \infty$ and floating point representations use only finitely many bits.

More information on these constructions, and especially on how to find suitable matrices may be found in the text by Dick and Pillichshammer (2010). When the base b is a prime power, but not a prime number, then similar constructions are available, but they do not use arithmetic modulo b.

15.11 Infinite variation

The Koksma-Hlawka inequality (Theorem 15.5) does not help us when $V_{\mathrm{HK}}(f) = \infty$. It reduces to $|\hat{\mu} - \mu| \leqslant \infty$, which we already knew. We have $V_{\mathrm{HK}}(f) = \infty$ whenever $|f|$ is unbounded, and that is a common occurence when f first transforms $\boldsymbol{x}$ into one or more Gaussian variables. If f is unbounded, then there exists a point $\boldsymbol{x}_{2^m}$ with $|f(\boldsymbol{x}_{2^m})|$ large enough to make $|\hat{\mu} - \mu| > 1$ for $n = 2^m$. Changing $\boldsymbol{x}_{2^m}$ this way for all $m \geqslant 1$ would not stop D_n^* from converging to zero but it would stop $\hat{\mu}$ from converging to μ. By contrast, unbounded integrands are not a severe problem for plain Monte Carlo, so long as they are square integrable.

In applications, it is common that $|f|$ only diverges to ∞ as $\boldsymbol{x}$ approaches the boundary of $[0, 1]^d$. Then QMC samples that approach the boundary, but not too quickly, can make $\hat\mu$ converge to μ. We will see in Chapter 17 that having each $\boldsymbol{x}_i \sim \mathbf{U}[0,1]^d$ induces about the right amount of singularity avoidance to get convergence, and one does not have to know where the singularities are.

It is not just unbounded functions that have $V_{\mathrm{HK}}(f)=\infty$. The indicator function of $T_d(\theta)=\{\boldsymbol{x}\in[0,1]^d \mid \sum_{j=1}^d x_j \leqslant \theta\}$ has infinite variation when $d\geqslant 2$ and $0<\theta<d$. More generally, we typically find that $V_{\mathrm{HK}}(1_S)=\infty$, for a set $S\subset[0,1]^d$, unless the boundary of S is formed from hyperplanes parallel to the coordinate axes of $[0,1]^d$. There are more details in the chapter end notes. If S is well enough behaved that $f(\boldsymbol{x})=1_S$ is Riemann integrable, then we know that $D_n^*\to 0$ implies that $\hat\mu\to\mu$ as $n\to\infty$. Also, because both $\hat\mu$ and μ are in $[0,1]$ we know that $|\hat\mu-\mu|\leqslant 1$, but Koksma-Hlawka does not refine this bound.

It is not just unbounded or discontinuous functions that have infinite variation. The function $(1-\sum_{j=1}^d x_j)_+ = \max(0,1-\sum_{j=1}^d x_j)$ has infinite variation when $d\geqslant 3$. Functions of the form $f(\boldsymbol{x})=\max(g(\boldsymbol{x}),h(\boldsymbol{x}))$ for two smooth functions g and h commonly arise in finance where one has the option to choose either outcome g or h. There is typically a cusp at points $\boldsymbol{x}$ where $g(\boldsymbol{x})=h(\boldsymbol{x})$ and this cusp leads to infinite variation when $d\geqslant 3$.

Infinite variation in f is still compatible with good results from QMC. Griebel et al. (2010, 2013) give conditions where integrands f that have infinite variation due to cusps or even discontinuities can be dominated by low dimensional ANOVA components that are smooth and of finite variation. The high dimensional non-smooth components then have a small norm and QMC works well despite the infinite variation of f.

The case where $\int |f(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{x} < V_{\mathrm{HK}}(f)=\infty$ has a parallel in ordinary Monte Carlo when $\int |f(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{x} < \int f(\boldsymbol{x})^2\,\mathrm{d}\boldsymbol{x}=\infty$. In that case Monte Carlo estimates satisfy $\hat\mu_n\to\mu$ as $n\to\infty$ but the central limit theorem fails to give a usable confidence interval. This trap is comparatively rare in Monte Carlo, though it can be brought on by a poorly chosen importance sampling distribution. We will see in Chapter 17 that RQMC methods will still asymptotically outperform plain MC on integrands with finite variance even if $V_{\mathrm{HK}}(f)=\infty$.

15.12 Higher order nets

Digital nets attain a quadrature error that is $O(n^{-1+\epsilon})$ for any $\epsilon>0$ as the number n of points tends to infinity. This rate is achieved when the function f has bounded variation in the sense of Hardy and Krause. A sufficient condition is that the mixed partial derivative of f taken once with respect to all components of $\boldsymbol{x}$ be continuous on $[0,1]^d$.

When the integrand f is even smoother, with continuous mixed partial derivatives of order two (or more) with respect to each component, digital nets still only attain the rate $O(n^{-1+\epsilon})$. Higher order nets described here are able to attain better convergence rates for smoother integrands.

We begin with the interleaving function. Suppose that $x=0.x_1x_2x_3\dots$ and $y=0.y_1y_2y_3\dots$ are two points in $[0,1)$, written in base b. For definiteness, suppose that neither ends in an infinite sequence of the digit $b-1$. The ***digit***

interleaving function yields the point

$$z = \mathbf{inter}(x, y) = 0.x_1y_1x_2y_2x_3y_3\dots$$

also in base b.

A higher order digital net is constructed from the variables of an ordinary digital net via the interleaving function. For example, a ***second order net*** is constructed by interleaving pairs of variables from an ordinary digital net:

$$z_{ij} = \mathbf{inter}(x_{i,2j-1}, x_{i,2j}), \quad 1 \leqslant i \leqslant n, \quad 1 \leqslant j \leqslant d$$

where $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are a (t, m, s)-net in base b and $s \geqslant 2d$.

A second order net can attain the error rate $O(n^{-2+\epsilon})$ for integrands as smooth as those described in Theorem 15.8 below for $k = 2$. Even better rates can be attained by interleaving more than two variables from a digital net. For $y_j = 0.y_{j1}y_{j2}y_{j3}\dots$ let

$$\mathbf{inter}(y_1, y_2, \dots, y_k) = 0.\underbrace{y_{11}y_{21}\dots y_{k1}}_{\text{1st digits}}\underbrace{y_{12}y_{22}\dots y_{k2}}_{\text{2nd digits}}y_{13}y_{23}\cdots.$$

A k'th order net has

$$z_{ij} = \mathbf{inter}(x_{i,kj-k+1}, \cdots, x_{i,kj}), \quad 1 \leqslant i \leqslant n, \quad 1 \leqslant j \leqslant d$$

where $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are points of a (t, m, s)-net in base b with $s \geqslant kd$.

Theorem 15.8. *Let $k \geqslant 1$ be an integer. Let f be a function on $[0, 1]^d$ such that any mixed partial derivatives of f taken up to k times with respect to each component x_j is square integrable. Let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ be a k'th order digital net. Then*

$$\left|\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i) - \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\right| = O(n^{-k}(\log n)^{kd}).$$

Proof. See Dick (2008, page 1120). □

The larger k is, the better the asymptotic rate of convergence is. The improved asymptote comes at a cost. To solve a d-dimensional problem with an order k net requires an ordinary net in kd dimensions. For larger k we can expect the asymptotic rate to take hold at larger n.

15.13 Haar wavelets and Walsh functions

Haar wavelets and Walsh functions provide some insight into how digital nets can improve on Monte Carlo. This section looks at the case $b = 2$. There are generalizations to integers $b \geqslant 2$. Those work similarly to the case $b = 2$ but are more complicated to present. This section presents an intuitive sketch. For details of wavelets, see Owen (1997a). For Walsh functions, see Dick and Pillichshammer (2010, Appendix A).

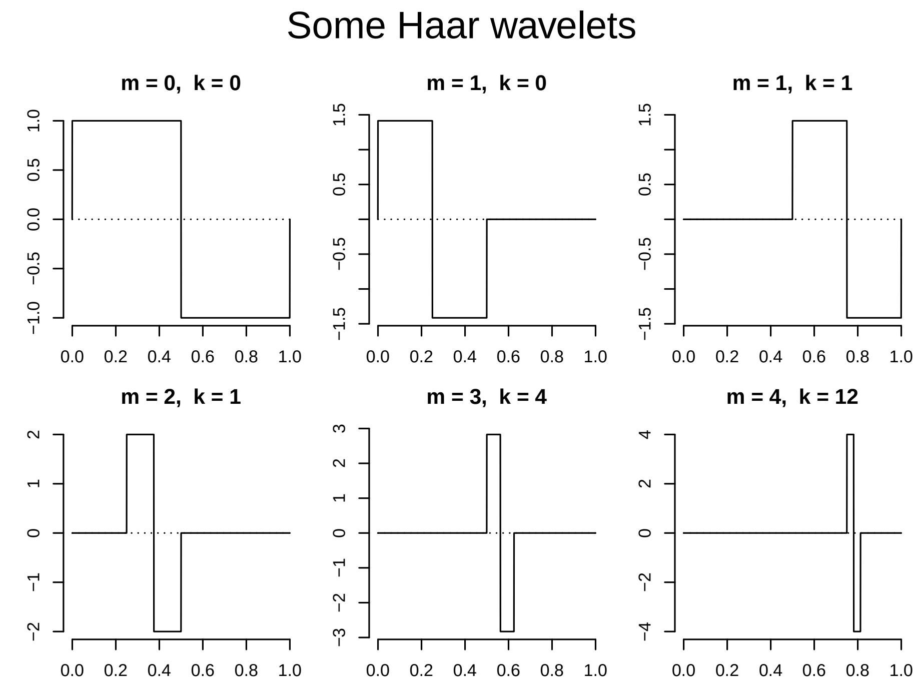

Figure 15.18: This figure shows a selection of Haar wavelets $\psi_{m,k}(x)$ on $[0,1)$.

The Haar analysis begins with a 'mother wavelet',

$$\psi(x) = \begin{cases} 1, & 0 \leqslant x < 1/2 \\ -1, & 1/2 \leqslant x < 1 \\ 0, & \text{else.} \end{cases}$$

Haar wavelets take the form

$$\psi_{m,k}(x) = 2^{m/2}\psi(2^m x - k), \quad 0 \leqslant k < 2^m, \quad m \geqslant 0.$$

Figure 15.18 shows some Haar wavelets on $[0,1)$. The factor 2^m makes the nonzero part of the wavelet take place over an interval of width 2^{-m}. Subtracting k shifts the wavelet. They all integrate to 0 over the unit interval. The external factor $2^{m/2}$ scales them so that $\int_0^1 \psi_{m,k}(x)^2 \,\mathrm{d}x = 1$. These wavelets are orthogonal: if $m \neq m'$ or $k \neq k'$, then $\int_0^1 \psi_{m,k}(x)\psi_{m',k'}(x)\,\mathrm{d}x = 0$. We say that wavelets with small m are coarse while those with large m are fine.

For $d = 1$, if $\int_0^1 f(x)^2 \,\mathrm{d}x < \infty$, we may write

$$f(x) = \mu + \sum_{m=0}^{\infty} \sum_{k=0}^{2^m-1} \beta_{k,m}\psi_{m,k}(x), \quad \text{where}$$

$$\beta_{k,m} = \int_0^1 \psi_{m,k}(x)f(x)\,\mathrm{d}x.$$

Then for $x_1,\dots,x_n \in [0,1]$

$$\hat\mu = \frac{1}{n}\sum_{i=1}^n f(x_i) = \mu + \sum_{m=0}^\infty \sum_{k=0}^{2^m-1} \frac{\beta_{k,m}}{n}\sum_{i=1}^n \psi_{m,k}(x_i),$$

and so the quadrature error satisfies

$$|\hat\mu-\mu| \leqslant \sum_{m=0}^\infty \sum_{k=0}^{2^m-1} |\beta_{k,m}| \times \biggl|\frac{1}{n}\sum_{i=1}^n \psi_{m,k}(x_i)\biggr|. \tag{15.24}$$

The wavelet $\psi_{m,k}(x)$ is piecewise constant on intervals $[\ell/2^{m+1},(\ell{+}1)/2^{m+1})$ for $0 \leqslant \ell < 2^{m+1}$. If $n = 2^{m+1+t}$ for $t \geqslant 0$ and each of those intervals contains 2^t of the x_i, then $(1/n)\sum_{i=1}^n \psi_{m',k}(x_i) = 0$ holds for all $0 \leqslant m' \leqslant m$ and $0 \leqslant k < 2^{m'}$. Sampling on a $(t,m+1,1)$-net in base 2 would then leave us with an error determined only by the fine contributions $\beta_{k,m'}\psi_{m',k}(x)$ for $m' > m$. As the sample size increases through powers of 2, more and more of the Haar wavelets are integrated without error.

Using the mean value theorem for integrals, we can get a rough idea of the magnitude of $\beta_{k,m}$ for large m. If f is continuous, then

$$\begin{aligned}\beta_{k,m} &= 2^{m/2}\int_0^1 \psi(2^m x - k)f(x)\,\mathrm{d}x\\ &= 2^{m/2}\int_{2^{-m}k}^{2^{-m}(k+1/2)} f(x)\,\mathrm{d}x - 2^{m/2}\int_{2^{-m}(k+1/2)}^{2^{-m}(k+1)} f(x)\,\mathrm{d}x\\ &= 2^{m/2}(f(x_1)-f(x_2))2^{-m-1} = 2^{-m/2-1}(f(x_1)-f(x_2)),\end{aligned}$$

for some points $x_1 \in [2^{-m}k, 2^{-m}(k+1/2)]$ and $x_2 \in [2^{-m}(k+1/2), 2^{-m}(k+1)]$. If f' is continuous on $[0,1]$, then $f(x_1)-f(x_2) = f'(x_3)(x_1-x_2)$ for some point $x_3 \in [x_1,x_2]$. Then

$$|\beta_{k,m}| \leqslant 2^{-m/2-1}|f'(x_3)||x_1-x_2| \leqslant 2^{-3m/2-1}|f'(x_3)|,$$

and so the contribution from fine wavelets decays for smooth f.

For functions on $[0,1)^d$ we form wavelets by taking products of the one dimensional wavelets above. For nonempty $u \subset \{1,2,\dots,d\}$, for vectors $\boldsymbol{m} \in \mathbb{N}^{|u|}$ and for vectors $\boldsymbol{k}$ with $k_j \in \{0,1,\dots,2^{m_j}-1\}$, we use the Haar wavelets

$$\psi_{u,\boldsymbol{m},\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j\in u}\psi_{m_j,k_j}(x_j).$$

The notation $\sum_{\boldsymbol{m}|u}$ below indicates that we sum only over the values of $\boldsymbol{m}$ that are legal for the given u. Similarly for $\sum_{\boldsymbol{k}|u,\boldsymbol{m}}$. Then if $\int_{[0,1)^d} f(\boldsymbol{x})^2\,\mathrm{d}\boldsymbol{x} < \infty$,

$$f(\boldsymbol{x}) = \mu + \sum_{u\neq\varnothing}\sum_{\boldsymbol{m}|u}\sum_{\boldsymbol{k}|u,k} \beta_{u,\boldsymbol{m},\boldsymbol{k}}\psi_{u,\boldsymbol{m},\boldsymbol{k}}(\boldsymbol{x}), \quad \text{for}$$

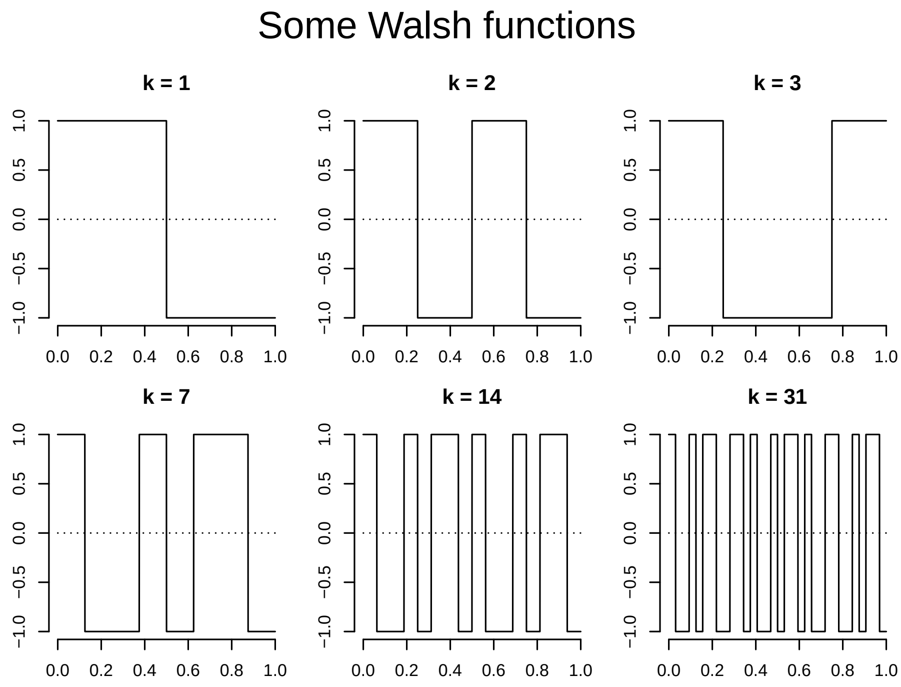

Figure 15.19: This figure shows a selection of Walsh functions $\mathrm{wal}_k(x)$ on $[0,1)$.

$$\beta_{u,\boldsymbol{m},\boldsymbol{k}} = \int_{[0,1)^d} \psi_{u,\boldsymbol{m},\boldsymbol{k}}(\boldsymbol{x}) f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.$$

Digital nets in base 2 correctly integrate $\psi_{u,\boldsymbol{m},\boldsymbol{k}}$ for the 'coarse' wavelets with a small value for $\sum_{j\in u} m_j$. The fine wavelets tend to have small coefficients $\beta_{u,\boldsymbol{m},\boldsymbol{k}}$ when f is smooth.

A very similar understanding of digital nets can be obtained via Walsh functions. For an integer $k \geqslant 0$, we can write $k = \kappa_0 + 2\kappa_1 + 4\kappa_2 + \cdots + 2^m\kappa_m$ where each $\kappa_j \in \{0,1\}$, for some finite m depending on k. We let $m(k)$ denote the smallest m for which this can be done. Now for $x \in [0,1)$ write $x = \xi_1/2 + \xi_2/4 + \xi_3/8 + \ldots$, taking care to choose an expansion that does not end in an infinite tail of 1s. For instance $x = 1/4$ is represented by $0.01000\cdots$ not $0.001111\cdots$ in base 2.

Using these expansions of $x \in [0,1)$ and $k \geqslant 0$, the k'th Walsh function is

$$\mathrm{wal}_k(x) = (-1)^{\kappa_0\xi_1+\kappa_1\xi_2+\kappa_2\xi_3+\cdots+\kappa_{m(k)}\xi_{m(k)+1}}.$$

The Walsh functions only take values ±1. They are constant in elementary intervals of width $2^{-m(k)-1}$. Figure 15.19 shows some Walsh functions.

Each Haar wavelet can be written as a linear combination of Walsh functions and vice versa. Like Haar wavelets, Walsh functions include coarse ones over wide intervals (for small k) and fine ones over narrow intervals (for larger k). Walsh functions are not localized in space like the Haar wavelets. The Walsh functions are orthogonal to each other.

The multivariable version of Walsh functions is slightly easier to write than the one for Haar wavelets because the above development of Walsh functions includes the constant function via $\mathrm{wal}_0(x) = 1$, and because we have not used two parameters, one for $m(k)$ and one for k given $m(k)$. For a vector $\boldsymbol{k} \in \mathbb{N}^d$ and a point $\boldsymbol{x} \in [0,1)^d$, we may define

$$\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^{d} \mathrm{wal}_{k_j}(x_j).$$

Similarly to Haar wavelets, we expand square integrable f as

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{N}^d} \gamma_{\boldsymbol{k}} \mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}), \quad \text{where} \quad \gamma_{\boldsymbol{k}} = \int_{[0,1)^d} \mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}) f(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}.$$

For smooth f, the coefficients $\gamma_{\boldsymbol{k}}$ tend to decay as the components of $\boldsymbol{k}$ increase, but not in precisely the same way that Haar wavelet coefficients $\beta_{u,\boldsymbol{m},\boldsymbol{k}}$ do. Dick (2009) shows that for a smooth function f on $[0,1)$, the vast majority of coefficients γ_k for a given value of $m(k)$, must decrease rapidly as $m(k)$ increases. Any exceptions are for those k that have only one 1 in their binary expansion.

15.14 Kronecker sequences

The term quasi-Monte Carlo is due to Richtmyer (1952). The points he used are sometimes called Richtmyer sequences and are perhaps better known as Kronecker sequences. They are included primarily for their historical interest. We begin with the Weyl criterion: $x_1, x_2, \dots \in [0,1)$ are uniformly distributed if and only if

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} e^{2\pi\sqrt{-1}\ell x_i} = 0 \tag{15.25}$$

for all integers $\ell \neq 0$. There is a d-dimensional version where non-zero integers ℓ are generalized to vectors $\boldsymbol{\ell} \in \mathbb{Z}^d \setminus \{\boldsymbol{0}\}$ and ℓx_i generalizes to $\boldsymbol{\ell}^\mathsf{T}\boldsymbol{x}_i$ for $\boldsymbol{x}_i \in [0,1)^d$ (Kuipers and Niederreiter, 1974, page 48).

Now let $x_i = \{\alpha i\} \equiv \alpha i - \lfloor \alpha i \rfloor$ for some $\alpha > 0$. This is the fractional part of αi also called its remainder modulo 1 and should not be confused with the set containing αi. If α is a rational number, then the values x_i will eventually start repeating in a cycle. If instead, α is irrational then x_i are uniformly distributed as can be shown by applying the Weyl criterion. Popular choices for α are square roots of prime numbers.

For $d > 1$, we can use $\boldsymbol{x}_i = (\{i\alpha_1\}, \{i\alpha_2\}, \dots, \{i\alpha_d\})$ for distinct irrational numbers α_j. It would not work to have $\alpha_1 = \sqrt{2}$ and $\alpha_2 = 2\sqrt{2}/3$. Then x_{i1}

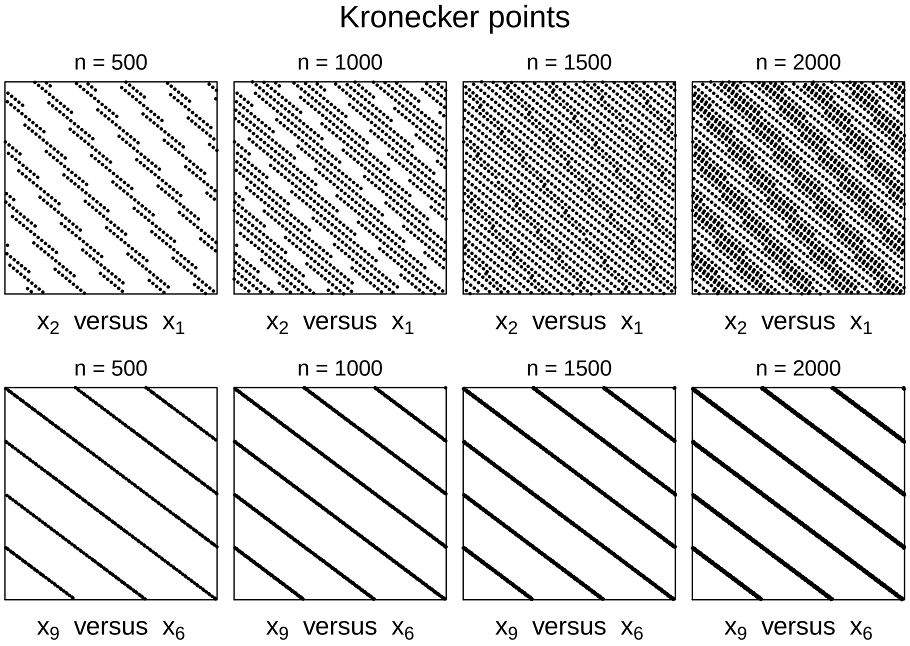

Figure 15.20: The top row plots $x_{i2} = \{i\sqrt{3}\}$ versus $x_{i1} = \{i\sqrt{2}\}$ for $n = 500, 1000, 1500, 2000$. The bottom row plots $x_{i9} = \{i\sqrt{23}\}$ versus $x_{i6} = \{i\sqrt{13}\}$ for the same value of n.

and x_{i2} would each be uniformly distributed in $[0, 1)$ but (x_{i1}, x_{i2}) would fail to be uniformly distributed in $[0, 1)^2$. We need α_j where

$$a_0 + \sum_{j=1}^{d} a_j\alpha_j = 0$$

does not hold for any rational numbers $a_0, \dots, a_d$. Then 1 and $\alpha_1, \dots, \alpha_d$ are said to be linearly independent over the rational numbers.

Theorem 15.9. *If* 1 *and* $\alpha_1, \dots, \alpha_d$ *are linearly independent over the rational numbers and* $\boldsymbol{x}_i = (\{i\alpha_1\}, \{i\alpha_2\}, \dots, \{i\alpha_d\})$*, then* $\boldsymbol{x}_i$ *for* $i \geqslant 1$ *are uniformly distributed over* $[0, 1)^d$.

Proof. Kuipers and Niederreiter (1974, Chapter 6). □

Figure 15.20 show some two dimensional projections of Kronecker points. The bottom row has x_{i9} versus x_{i6}, which looks to be one of the worst pair plots among the first 10 dimensions. As n increases, the diagonal stripes there become wider and eventually fill in the plane. As n continues to increase, some diagonal stripes get about double the sampling intensity of the rest of the figure and those double wide stripes grow slowly to cover the square, before a small triple wide stripe appears.

The discrepancy bounds for Kronecker sequences involve somewhat higher powers of $\log(n)$ than for the digital sequences in this chapter. It is also hard to make a good choice of α_j when $d > 1$. See Niederreiter (1992b) for both of these points. The most widely used choices are $\alpha_j = \sqrt{p_j}$ where p_j is the j'th largest prime number but as we see in Figure 15.20, even some two dimensional projections look bad. Like Halton sequences, Kronecker sequences do not seem to have any especially good values of n.

Kronecker sequences resemble lattice rules of Chapter 16 except that they are extensible instead of being periodic. These sequences have been criticized because their theoretical properties depend on irrationality of α_j and in floating point computations rational approximations to α_j must be used. Some authors, for example Vandewoestyne (2008), report good results from the Richtmyer sequence despite this concern.

The year of Richtmyer's technical report is variously given as 1951 or 1952. It was written in October 1951 but published in April 1952. Richtmyer also refers to the effective number of dimensions in an integrand, in what we would now call the truncation sense. That is, a notion of effective dimension appears already in the first QMC paper. Richtmyer (1952) finds theoretical superiority for his quasi-Monte Carlo points but concludes that there is no practical superiority. His example functions were of high, in fact indefinite dimension and were discontinuous. The function f implicit in his computation had infinite variation. Richtmyer's technical report was not optimistic about the performance of QMC. The poor performance he saw could have been due to a lack of smoothness in his integrands or to the poor finite sample equidistribution of the Kronecker points.

Chapter end notes

Dick et al. (2013) present QMC using methods from reproducing kernel Hilbert spaces. They pay special attention to the weighted spaces of §7.7.

Acceptance-rejection

Suppose that acceptance-rejection is used to generate one or more of the components of a random vector in $\mathbb{R}^d$. We use some number of uniform random variables to generate the proposal and one or more others to make the acceptance-rejection decision or decisions. Doing this we use a point in $[0,1]^s$ where generally $s > d$ to sample $\boldsymbol{x}_i$. If the i'th point in $[0,1]^s$ is rejected then we ignore it and only evaluate f on the accepted points. There is a set $A \subset [0,1]^s$ for which $\boldsymbol{x} \in A$ implies acceptance. Let f subsume the transformations from $[0,1]^s$ to proposed points and the ultimate integrand applied to an accepted proposals.

Then, using QMC this way estimates $\mu = \int_A f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ by

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\boldsymbol{x}_i \in A} f(\boldsymbol{x}_i) \Bigm/ \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\boldsymbol{x}_i \in A},$$

for low discrepancy points $\boldsymbol{x}_i$. That is, we have a ratio estimate with a numerator estimaing $\int \mathbb{1}_{\boldsymbol{x}\in A} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ and a denominator estimating $\int \mathbb{1}_{\boldsymbol{x}\in A}\,\mathrm{d}\boldsymbol{x}$. When A has an arbitrary boundary that is not a box parellel to coordinate axes, then the numerator and denominator integrands ordinarily have infinite variation in the sense of Hardy and Krause, as discussed below. QMC sampling will converge to the right answer if both $\mathbb{1}_A(\boldsymbol{x})$ and $f(\boldsymbol{x})\mathbb{1}_A(\boldsymbol{x})$ are Riemann integrable. Zhu and Dick (2014) study this process and find empirical evidence that it has better than $O(n^{-1/2})$ errors despite the infinite variation.

Another approach is to make the first $r \geqslant 1$ proposals and decisions based on a point in $[0,1]^s$ for some s. If that ends in a rejection, carry on from there using pseudo-random numbers to propose and accept or reject until an acceptance occurs. The result is a hybrid of MC and QMC. The hybrid might still be better than plain MC, but the Koksma-Hlawka theorem would not be applicable to it because the dimension is not bounded.

Discrepancy

Discrepancy as a branch of mathematics is older than quasi-Monte Carlo. It is sometimes called ‘irregularities of distribution’. It goes back at least to Weyl (1914, 1916) and the Weyl criterion (15.25). There are texts by Beck and Chen (1987), Matoušek (1999) and Chazelle (2000). The latter emphasizes applications to theoretical computer science. Many authors use $n \times D_n^*$, an integer count, instead of D_n^*. We call those ‘integer discrepancies’ below. One of the first problems was to show that this integer discrepancy could not remain bounded as $n \to \infty$.

Integer discrepancies taken over sets other than axis-parallel boxes generally cannot be made as small as $\log(n)^{d-1}$. Lower bounds worse than that are known for circular disks in $[0,1]^d$, axis parallel triangles in $[0,1]^2$, rotated d-dimensional boxes and many more geometrical quantities. See Alexander et al. (2018) for results and references. Axis-parallel boxes are much easier to sample uniformly than those other sets. Fortunately, low discrepancy over axis-parallel boxes is already sufficient to provide good numerical integration for functions of bounded variation.

Doerr et al. (2014) give a comprehensive survey of methods to compute L^2 and star discrepancies. The L^2 discrepancy formula (15.7) of Warnock (1972) requires $O(dn^2)$ computation, if performed as written. Heinrich (1996) presents an algorithm to compute it in $O(n\log(n)^d)$ work as $n \to \infty$ for fixed d. There is more interest in computing the star discrepancy, which is much harder. Gnewuch et al. (2009) report that all known algorithms are exponential in d. They show that the problem is NP-hard when $n = d$ and both go to infinity.

Doerr et al. (2014) describe some algorithms that approach $O(n^{d/2})$ cost as well as some faster algorithms that approximate the star discrepancy.

Hickernell (1998) points out some connections between discrepancy measures and goodness of fit tests in statistics.

Nets

The first digital nets were the Sobol' sequences (Sobol', 1967). The next major family of nets were the Faure sequences (Faure, 1982). Niederreiter (1987) merged Faure's and Sobol's concepts to produce the definitions of digital nets and sequences used here. He also created additional constructions, including a generalization of Faure's $(0, s)$-sequences to prime power bases $q = p^r \geqslant s$. The best available t parameters are for the NX nets of Niederreiter and Xing (1996). The minT project (Schürer and Schmid, 2009) maintains an online reference to net constructions. Higher order nets are due to Dick (2008).

The value of t in a digital net can be strictly less than the value of t in a digital sequence of which the net is the first b^m points. The attained t value of digital nets extracted from digital sequences has been studied by Schmid (1999, 2001).

The projections shown in Figure 15.11 were found using the projection pursuit option in Ggobi (Swayne et al., 2003). Projection pursuit is a numerical optimization designed to find projections of data that are highly structured. There are several ways to measure the strength of structure. Of these, the central mass option seemed most useful at finding bad projections of QMC points.

Although nets have been presented as an improvement on Halton sequences, there is still interest in generalized Halton sequences employing permutations to break up the striping artifacts. Vandewoestyne and Cools (2006) compare permutations via the resulting mean squared discrepancy and find good results for a 'reversed Halton' scramble that permutes $0, 1, \ldots, b-1$ via $\pi_b = (0, b-1, b-2, \ldots, 2, 1)$. Faure and Lemieux (2009) study several proposals and make their own. See also Chi et al. (2005). Random scrambles of Halton points are considered in §17.10.

QMC versus MC

An interesting point of view, advanced by Zaremba (1968), is that Theorem 15.3 is the real reason that Monte Carlo sampling works. By that line of reasoning, it does not matter that we tried to get independent $\mathbf{U}[0, 1]^d$ points. All that matters for $\hat{\mu}$ is which points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ we actually got. At the time Zaremba wrote, random number generators were not as well tested as they are now, and he remarked that the only really worthwhile tests should be for properties like discrepancy that affect the accuracy. By now random number generators are much more thoroughly tested, theoretically and empirically, and many of those tests refer to properties like discrepancy. The tests however verify that the discrepancies behave as predicted under randomness. Zaremba would have preferred discrepancies far smaller than under genuine randomness. The randomness in Monte Carlo does serve a very practical purpose in letting us quantify the uncertainty in our estimates.

Total variation

When f is a continuously differentiable function on $[0,1]$ then it has total variation $V(f)=\int_0^1 |f'(x)|\,\mathrm{d}x$. Variation sounds like variance, and the concepts are similar, but with important differences. For a constant c, $V(cf)=|c|V(f)$ while $\mathrm{Var}(cf)=c^2\mathrm{Var}(f)$ so it is more reasonable to compare $V(f)$ to the standard deviation $\sqrt{\mathrm{Var}(f)}$. When f' is continuous on $[0,1]$

$$V(f)=\int_{(0,1)}\lim_{\epsilon\downarrow 0}\frac{1}{\epsilon}\bigl|f(x+\epsilon)-f(x)\bigr|\,\mathrm{d}x,$$

while the standard deviation can be written

$$\sqrt{\mathrm{Var}(f)}=\biggl[\frac{1}{2}\int_0^1\int_0^1\bigl(f(x)-f(\widetilde{x})\bigr)^2\,\mathrm{d}x\,\mathrm{d}\widetilde{x}\biggr]^{1/2}.$$

Variation is based on local differences (closer than ϵ) while variance is based on global differences $x-\widetilde{x}\in[-1,1]$.

We need to define the total variation for functions that are not necessarily continuously differentiable or even differentiable at all. Let $\mathcal{X}_n=\{\boldsymbol{x}\in[0,1]^n\mid 0<x_1<x_2<\cdots<x_n=1\}$ and by convention take $x_0=0$. The total variation of a function on $[0,1]$ is

$$V(f)=\sup_{n\geqslant 1}\sup_{\boldsymbol{x}\in\mathcal{X}_n}\sum_{i=1}^n|f(x_i)-f(x_{i-1})|. \tag{15.26}$$

When f' is continuous on $[0,1]$, then $V(f)=\int_0^1|f'(x)|\,\mathrm{d}x$. For the nondifferentiable function $f(x)=\mathbb{1}_{x>1/2}$ the total variation (15.26) is easily seen to be 1. The supremum in (15.26) can be infinite. For example $V(f)=\infty$ for the function f with $f(x)=1/x$ for $x>0$ and $f(0)=0$. A standard bounded function of infinite variation is $f(x)=\sin(1/x)$ for $x>0$ and $f(0)=0$.

There are numerous generalizations of total variation to functions on $[0,1]^d$ for $d\geqslant 1$. Clarkson and Adams (1933) consider 6 of them and Adams and Clarkson (1934) include two more. In quasi-Monte Carlo, we use total variation in the sense of Hardy and Krause. That in turn is based on total variation in Vitali's sense.

Here is a brief synopsis of total variation for QMC that avoids some cumbersome d-dimensional notation. The full details are in Owen (2005). Vitali's variation $\mathrm{Vit}(f)$ is a d-dimensional version of (15.26). The list of points $x_i\in[0,1]$ is generalized to a d-dimensional grid within $[0,1]^d$. In each d-dimensional cell of that grid, the difference $f(x_i)-f(x_{i-1})$ is replaced by an alternating sum. For $d=2$, the alternating sum is a difference of differences like $f(a_1,a_2)-f(a_1,b_2)-f(b_1,a_2)+f(b_1,b_2)$. For general $d\geqslant 1$ it is a d-fold difference of differences. Vitali's variation is the supremum over d-dimensional grids in $[0,1]^d$ of the sum over grid cells of the absolute values of the alternating sums. If $f^{1:d}(\boldsymbol{x}) = \partial^d f(\boldsymbol{x})/\partial \boldsymbol{x}$ exists, then $\int |f^{1:d}(\boldsymbol{x})| \mathrm{d}\boldsymbol{x} \geqslant \mathrm{Vit}(f)$ with equality when $f^{1:d}$ is continuous on $[0,1]^d$ (Fréchet, 1910).

Vitali's variation is unsuitable for QMC because $\mathrm{Vit}(f) = 0$ if f does not depend on one of its components. For example, $f(\boldsymbol{x}) = \sin(1/x_2)\mathbb{1}_{x_2>0}$ does not depend on x_1 and so it has $\mathrm{Vit}(f) = 0$.

To get a suitable definition of variation, we begin by specifying an anchor point $\boldsymbol{c} \in [0,1]^d$. Next, for every $u \subseteq \{1,\dots,d\}$, let $\boldsymbol{x}_u{:}\boldsymbol{c}_{-u}$ be the point $\boldsymbol{y} \in [0,1]^d$ with $y_j = x_j$ for $j \in u$ and $y_j = c_j$ for $j \notin u$. Now define the function $f_{\boldsymbol{c},u}$ on $[0,1]^{|u|}$ through $f_{\boldsymbol{c},u}(\boldsymbol{x}_u) = f(\boldsymbol{x}_u{:}\boldsymbol{c}_{-u})$. This function is not the same as the ANOVA component f_u. The total variation of f in the sense of Hardy and Krause with anchor $\boldsymbol{c}$ is

$$V_{\mathrm{HK},\boldsymbol{c}}(f) = \sum_{u \neq \varnothing} \mathrm{Vit}(f_{\boldsymbol{c},u}). \tag{15.27}$$

In the prior example, $V_{\mathrm{HK},\boldsymbol{c}}(\sin(1/x_2)\mathbb{1}_{x_2>0}) = \infty$ for any $\boldsymbol{c} \in [0,1]^d$. The original and customary definition of V_{HK} uses $\boldsymbol{c} = \mathbf{1}$, the vector of d ones. That is

$$V_{\mathrm{HK}}(f) = V_{\mathrm{HK},\mathbf{1}}(f), \tag{15.28}$$

is the measure of variation used in the Koksma-Hlawka Theorem. Aistleitner and Dick (2015) have found it useful to move the anchor to $\mathbf{0}$ when studying discrepancies with respect to distributions other than $\mathbf{U}[0,1]^d$.

Unbounded integrands

Theorem 15.3 ensures that $\hat{\mu} \to \mu$ when $\boldsymbol{x}_i \in [0,1]^d$ are a low discrepancy sequence and f is a Riemann integrable function on $[0,1]^d$. From the converse Theorem 15.4, we see that if f is not Riemann integrable then low discrepancy alone is not enough to ensure convergence. Unbounded functions on $[0,1]^d$ are not Riemann integrable.

The Riemann integral can be extended to some unbounded functions by taking appropriate limits. For some unbounded functions f the integral $\mu_\epsilon = \int_{[\epsilon,1-\epsilon]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ will converge to μ as $\epsilon \to 0^+$. That limiting process cannot help us if $\boldsymbol{x}_1 = 0$ and $f(\boldsymbol{x}_1) = \infty$. Random sampling handles singular integrands well because if $\mathbb{E}(f(\boldsymbol{x}))$ exists then $\mathbb{P}(f(\boldsymbol{x}) = \infty) = 0$.

For unbounded integrands, the uniformly distributed points must also avoid the singularity to some extent. Sobol' (1973a,b) shows that some of his sequences tend to avoid a region around the origin, and this helps when the integrand has a known singularity at the origin. The Halton points, if started with $x_{1j} = \phi_{p_j}(1)$ and not $x_{1j} = \phi_{p_j}(0) = 0$ have a tendency to avoid the origin (Owen, 2006a) and, to a lesser extent, every corner of $[0,1]^d$. Hartinger et al. (2005) study corner avoidance properties of some generalized Niederreiter sequences and Faure sequences. In these problems, there is a delicate interplay between the speed at which the points approach the origin and/or the boundary of $[0,1]^d$ and the rate at which the (integrable) function diverges near its singularity; discrepancy alone is not sharp enough to give a sufficient condition for convergence.

Polynomial lattice rules

Polynomial lattice rules (Niederreiter, 1992a) are a beautiful generalization of ideas from lattice rules (see Chapter 16) but they produce digital nets instead of such lattices. Their presentation requires methods beyond the prerequisites for this book. The interested reader may see Dick and Pillichshammer (2010, Chapter 10) for details. As with lattice rules, polynomial lattice rules require a search process to pick parameters. Compared to Halton, Faure or Sobol' nets that are more or less automatic, that choice is a burden. On the other hand, having that choice allows one to select a digital net customized to a given problem instance. See Nuyens (2013) and also Kuo and Nuyens (2016) who customize polynomial lattice rules for problems involving partial differential equations over random environments (Graham et al., 2015).

Exercises

15.1. This is based on an observation of Sobol' who recommends increasing sample sizes through a geometric progression, not an arithmetic one. Let $\hat\mu_n = (1/n)\sum_{i=1}^n f(\boldsymbol{x}_i)$. Suppose that for some $A < \infty$ and $\epsilon > 0$ and all integers $n \geqslant 1$ we have $|\hat\mu_n - \mu| \leqslant An^{-1-\epsilon}$ and $|\hat\mu_{n+1} - \mu| \leqslant A(n+1)^{-1-\epsilon}$. Using these facts find an upper bound on $|f(\boldsymbol{x}_{n+1}) - \mu|$ strong enough to show that $\lim_{n\to\infty} |f(\boldsymbol{x}_{n+1}) - \mu| = 0$. The interpretation is that if we could really get $O(n^{-1-\epsilon})$ error for every sample size n, then we really only need to use one sample point $\boldsymbol{x}_n$ for a very large n.

15.2. The left endpoint rule is $x_i = (i-1)/n$ for $i = 1, \dots, n$. Find D_n^* and D_n for this rule. For $n \geqslant 1$, does the rule with $3n$ points extend the rule with n points? Is there a smaller rule that extends the left endpoint rule of size n?

15.3. This exercise assumes that you have already solved Exercise 15.2.

The Hammersley sequence of length $n \geqslant 1$ in dimension $d \geqslant 2$ combines a left endpoint rule for the first component, with a $d-1$-dimensional Halton sequence for the other components. The Halton sequence is extensible. The left endpoint sequence can be extended too, not from sample size n to $n+1$, but along a rapidly growing sequence of values n. Having extensible parts does not necessarily make the Hammersley sequence extensible!

For this exercise: devise a method that extends the Hammersley sequence in $[0,1]^3$ by combining an extensible left endpoint rule for the first component with the extensible Halton sequence for the second and third components. Using graphical investigations, find a flaw with this construction.

Note: It is not known whether this attempt to produce an extensible Hammersley sequence yields a low discrepancy sequence. It has visually apparent flaws in two dimensional projections that may or may not make it fail to be uniformly distributed. The Halton sequence should be used instead of the Hammersley when extensibility is required.

15.4. The function $\log(n)^{d-1}/n$ over $2 \leqslant n < \infty$ first increases and then decreases as $n \to \infty$ for fixed $d \geqslant 2$. At what value of n does it start decreasing? Non-integer answers are ok.

15.5. The QMC bound is predicted to stay below the root mean squared error when $C\log(n)^{d-1}/n < n^{-1/2}$ holds for all $n \geqslant N$, for some $C > 0$.

a) For what n does that happen when $C = 1$?

b) For what n does that happen when $C = 10^{-6}$?

c) For what n does that happen when $C = 10^{6}$?

Here C is the total variation of the integrand in the sense of Hardy and Krause, multiplied by the implied constant in the discrepancy bound. This is a 'prediction' in the sense that the discrepancy is only asymptotically of the given form and the integrand is not necessarily worst case.

15.6. By using Warnock's formula (15.7), find $\mathbb{E}((D^*_{n,2})^2)$ when $\boldsymbol{x}_i$ are sampled values of $\boldsymbol{x}_i \sim \mathbf{U}(0,1)^d$.

15.7. Let $x_i = \phi_3(i)$ be the i'th point of the van der Corput sequence in base $b = 3$. Let $D^*_{n,3}$ be the star discrepancy of $x_1, \dots, x_n$ for $1 \leqslant n \leqslant 6561 = 3^8$. Over this range, is $D^*_{n,3}$ ever below $D^*_{m(n),3}$ where $m(n)$ is the greatest power of 3 that is not larger than n?

15.8. The Faure scrambled Halton sequence is known (Ökten et al., 2009) to produce some bad projections in higher dimensions.

a) Plot the points $(\phi_{1031}(i-1), \phi_{1033}(i-1))$ for $1 \leqslant i \leqslant 500$.

b) Repeat the plot, this time using Faure's scramble of the Halton sequence, as described at and just below Equation (15.17).

c) Repeat the two previous plots for $n = 1000$ and $n = 10{,}000$. Do either of them look satisfactory?

d) Replace the Faure permutations by permutations chosen at random, subject to the constraint that $\pi(0) = 0$. Plot the first 1000 randomly scrambled points. Compare 5 such randomly scrambled images to the Faure scrambled points.

e) Repeat the previous exercise, but this time do not force $\pi(0) = 0$.

15.9. The wing weight function appears to be increasing in all of its inputs except Λ. That one varies over a small range, so maybe it is not important. Cumulative means of van der Corput points used in the Halton sequence tend to approach 0.5 from below. Figure 15.21 shows the first 1000 dimensions.

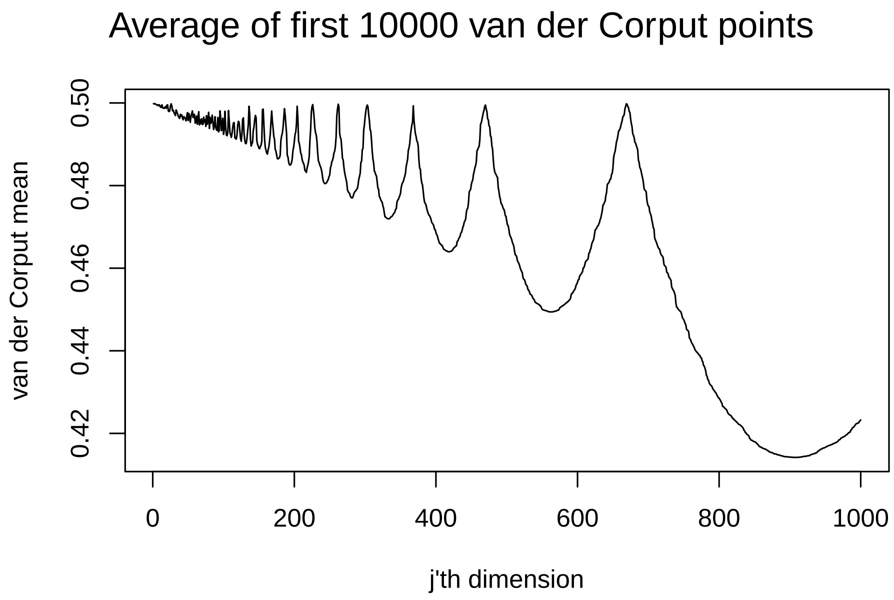

Figure 15.21: The vertical axis is the average of the first 10,000 of a van der Corput sequence in base p_j, where p_j is the j'th prime. The horizontal axis is $j = 1, \dots, 1000$. See Exercise 15.9.

The bias is present but smaller in the first 10 dimensions that the wing weight function uses.

These two observations suggest that antithetic sampling of the Halton points might be an improvement. Evaluate the wing weight function on the first $n = 5000$ points of the Halton sequence. Repeat on antithetic pairs $\tilde{\boldsymbol{x}}_i = 1 - \boldsymbol{x}_i$, componentwise for $i = 1, \dots, n$. Make a plot comparing the cumulative mean wing weight under antithetic sampling of the Halton sequence to cumulative mean wing weight without antithetic sampling. Take care to have the same number of evaluation points in the horizontal axis for both methods.

15.10. A $(0, 5, 7)$-net in base 7 has $n = 7^5$ points. How many distinct elementary intervals of volume 7^{-5} does it balance?

15.11. The Halton sequence was left out of the comparison for the test function $f_{2,25}$ of §15.9. It has some very bad projections in high dimensions, so it might do worse than the Faure and Sobol' sequences. Then again, the Halton sequence uses its most equidistributed components on the first and most important variables of $f_{2,25}$, and so it might do very well.

a) Apply the Halton sequence to the function $f_{2,25}$ of §15.9 over the range $125 \leqslant n \leqslant 1.5 \times 10^6$. Determine how best to display its accuracy and then compare it to the other methods. Is it superior, inferior, or comparable to those other methods? You can refer to the values displayed in Figure 15.17 without recomputing them.

b) Repeat part **a)** using Faure's scrambling of the Halton sequence, as described at and just below Equation (15.17). Also comment on whether the scrambling improves the Halton sequence for this function. If there is a clearly quantifiable trend then measure it (as for example a typical ratio of absolute errors).

c) Suppose that through bad luck or bad planning we had the variables in reverse order of importance. That is we used instead

$$\widetilde{f}_{2,d}(\boldsymbol{x}) = \prod_{j=1}^{d}\Bigl(1 + \frac{\sqrt{3}}{j}(x_{d-j+1} - 1/2)\Bigr)$$

for $d = 25$. Make a graphical evaluation of the Faure sequence. Determine whether Faure's scrambling improves it. For both methods compare (graphically) how well they do on $\widetilde{f}_{2,d}$ to how well they do on $f_{2,d}$.

15.12. We have studied QMC on some test functions formed by products of univariate functions. Here we investigate the effectiveness of antithetic sampling on such product test functions.

a) Prove Proposition 15.3.

b) For the function f_1 of §15.9 determine whether antithetic sampling is better than Latin hypercube sampling.

c) Suppose now that $\int_0^1 g_j(x)g_j(1-x)\,\mathrm{d}x = \rho_j \in [-1, 1]$. What now is the variance (15.22)?

15.13. It is possible that NX-nets and NX-sequences should be held in reserve for the very hardest quadrature problems. Their extremely low t parameter shows that for some dimensions d, no other known construction for (t, m, d)-nets will be able to balance any genuinely d-dimensional elementary intervals (as defined on page 25) without requiring a larger sample size n than the NX-nets use.

This exercise is a project to verify whether NX sequences are superior for very hard problems. To do so:

- write or obtain some code for Niederreiter-Xing sequences,
- construct some test functions that are smooth and fully d-dimensional in that $\sigma^2 = \sigma^2_{\{1,2,\dots,d\}}$ from the ANOVA decomposition of f,
- compare the accuracy of Niederreiter-Xing quadrature to ordinary MC and to some other QMC methods using your test functions.

This project requires you to use some judgment. You must select the range of dimensions d, sample sizes n, test functions f, and the competing QMC method or methods. You also have to decide how to present the results, and give an interpretation of what you find. There is even an interesting issue in choosing the base b. Most of the literature is for $b = 2$. Results of Kritzer (2006) suggest that $b = 3$ may be better, just as Faure (1982) finds base 3 best for van der

Corput sequences in general bases. It is impossible to make perfect choices for these decisions. A proper writeup should however explain the choices made.

15.14. For $n \geqslant 1$ let $x_1, \dots, x_n$ be fixed distinct points in $(0, 1)$. Let $f(x) = 1$ if $x = x_i$ for one of these points and let $f(x) = 0$ otherwise. This seems like an unfavorable integrand for the given set of integration points, but perhaps it is not the worst case.

a) Find both $\mu = \int_0^1 f(x)\,\mathrm{d}x$ and $\hat\mu = (1/n)\sum_{i=1}^n f(x_i)$.

b) What is the total variation of f on the interval $[0, 1]$?

c) The function f is a worst case function if $|\hat\mu - \mu| = D_n^* V(f)$ holds. For what point sets $x_1, \dots, x_n$, if any, does this happen?

d) [Harder] Suppose instead that have used a closed rule with $x_1 = 0$ and $x_n = 1$. Repeat parts b and c.

15.15. Let $f(\boldsymbol{x}) = \prod_{j=1}^d x_j^{A_j}$ on $[0, 1]^d$ where $A_j > 0$. Similarly, let $\widetilde{f}(\boldsymbol{x}) = \prod_{j=1}^d (1 - x_j)^{A_j}$ on $[0, 1]^d$.

a) Show that $V_{\mathrm{HK}}(f) = 2^d - 1$.

b) Show that $V_{\mathrm{HK}}(\widetilde{f}) = 1$.

15.16. Consider the function $f(\boldsymbol{x}) = 12^{d/2}\prod_{j=1}^d (x_j - 1/2)$. Show that if $\boldsymbol{x} \sim \mathbf{U}(0, 1)^d$ then $f(\boldsymbol{x})$ has variance $\sigma^2 = 1$ regardless of d. Show that $V_{\mathrm{HK}}(f)$ increases exponentially in d. For $d = 10$, find the smallest integer n_0 such that the bound on D_n^* in (15.15) times $V_{\mathrm{HK}}(f)$ is smaller than $n^{-1/2}$ whenever $n \geqslant n_0$.

15.17. Show that the function

$$f(\boldsymbol{x}) = \begin{cases} 1, & x_2 \leqslant x_1 \\ 0, & \text{else} \end{cases}$$

for $\boldsymbol{x} = (x_1, x_2) \in [0, 1]^2$, has infinite variation in the sense of Hardy and Krause.

15.18. Construct a function f on $[0, 1]^2$ such that $V_{\mathrm{HK}}(f) = \infty$ but $V_{\mathrm{HK}}(f^2) < \infty$.

15.19. Construct a function f on $[0, 1]$ that is discontinuous but for which f_{anti} has infinitely many continuous derivatives.

16

Lattice rules

Lattice rules are a second major family of QMC methods. They have developed in parallel with digital nets and sequences of Chapter 15. The points we use in a lattice rule have the same geometric structure as the multiplicative congruential random number generators that we saw in Chapter 3, just as algorithms for digital nets are like shift register random number generators. In both cases, QMC can be likened to finding a small random number generator and using all of its points. Lattice rules are well suited to Fourier methods of analysis and periodic integrands, but they work well more generally. The presentation in this chapter follows the text by Sloan and Joe (1994) and Chapter 5 of Niederreiter (1992b), and incorporates some more recent developments.

As in Chapter 16, we are going to create n points in the d-dimensional unit cube. For lattice rules it is most convenient to use $[0,1)^d$ for that cube, instead of $[0,1]^d$, because periodic functions will play a critical role. As before, $\mathbf{U}[0,1)^d$ random variables might first be transformed into some other distribution before applying an integrand of interest. Letting f incorporate both our transformations and the integrand of interest, we will estimate $\mu = \int_{[0,1)^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ by $(1/n)\sum_{i=1}^n f(\boldsymbol{x}_i)$ as before, using $\boldsymbol{x}_i$ from the lattice rules described here.

16.1 Rank one lattices

Given a strategically chosen vector of integers $\boldsymbol{z} = (z_1, \dots, z_d)$ and a compatibly chosen sample size $n \geqslant 1$, the ***rank-1 lattice rule*** has points $\boldsymbol{x}_i \in [0,1)^d$ with components

$$x_{ij} = \frac{(i-1)z_j}{n} \mathsf{\ mod\ 1} \tag{16.1}$$

for $i = 1, \dots, n$ and $j = 1, \dots, d$. As usual, $y \bmod 1$ means $y - \lfloor y \rfloor$ and we adopt a convenient shorthand $\{y\}$ for this. The intent $\{y\} = y \bmod 1$ will always be clearly distinct from the set containing y. These lattices have rank 1 because they use only one vector $\boldsymbol{z}$ of integers. Lattices of rank-2 and higher using more than one vector are described in the end notes. They are not commonly used now.

We could replace equation (16.1) by $x_{ij} = iz_j/n \bmod 1$ for $i = 1, \dots, n$. We would get the same set of points in a different order. It is however standard to present lattice rules with $\boldsymbol{x}_1 = (0, 0, \dots, 0)$ as in (16.1) instead of $\boldsymbol{x}_n = (0, 0, \dots, 0)$.

It is critically important to use all n points of a lattice rule. For instance, using just the first $n/2$ points omits the whole region $\{\boldsymbol{x} \in [0,1)^d \mid 1/2 < x_1 < 1\}$. This requirement is not much of a problem, because can choose the sample size n to fit within our computing budget.

It is clear at the outset that z_j and n should not share a common factor $k > 1$. If they did, then there would be at most n/k distinct values for $\{(i-1)z_j/n\}$. Instead, choosing z_j to have no common factor with n means that the values $x_{1j}, \dots, x_{nj}$ will take on all n values 0, $1/n$, $2/n$, …, $(n-1)/n$. It is typical to take $z_1 = 1$, so that $\boldsymbol{z} = (1, z_2, \dots, z_d)$.

Using the points $\boldsymbol{x}_i$ of a rank-1 lattice rule, we estimate $\mu = \int_{[0,1)^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ by

$$\hat{\mu}_{\mathrm{rlat}} = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i) = \frac{1}{n}\sum_{i=0}^{n-1} f\left(\left\{\frac{i}{n}\boldsymbol{z}\right\}\right). \tag{16.2}$$

That is, the indices of $\boldsymbol{x}_i$ run from 1 to n but the argument of f uses i from 0 to $n-1$. Obviously, the difficult part will be to choose parameters n and $\boldsymbol{z}$ that lead to a good rule. Figure 16.1 shows three small rank-1 lattices, one good lattice, one not so good, and one clearly flawed because it leaves wide empty diagonal gaps.

Before discussing how to choose $\boldsymbol{z}$ well, it is worth mentioning some special cases. The ***Fibonacci lattices*** are especially good for $d = 2$, as described in the end notes. The Fibonacci numbers are defined by $F_1 = F_2 = 1$ and $F_j = F_{j-1} + F_{j-2}$ for $j \geqslant 3$. A Fibonacci lattice has $n = F_m$ and $\boldsymbol{z} = (1, F_{m-1})$ for $m \geqslant 3$. The illustrations in this section include many Fibonacci lattices. Some other lattices with $n = F_m$ but z_2 other than F_{m-1} are then used to illustrate what can happen with a poorly chosen $\boldsymbol{z}$. For $d \geqslant 3$, there is no such singularly good choice.

A second special case of rank-1 lattices are the ***Korobov rules***. These have $\boldsymbol{z} = (1, a, a^2 \bmod n, \dots, a^{d-1} \bmod n)$ for a carefully chosen integer $a \in \{2, 3, \dots, n-1\}$. Having $z_j = a^{j-1} \bmod n$ means that given n, the search for a good Korobov rule only requires a search for a, instead of a search for $z_2, z_3, \dots, z_d$. Furthermore, a choice for a can be used with more than one dimension d, unlike a choice for $(1, z_2, \dots, z_d)$. That means we do not have to have a table of vectors $\boldsymbol{z} \in \mathbb{Z}^d$ indexed by n, with a separate table for each d.

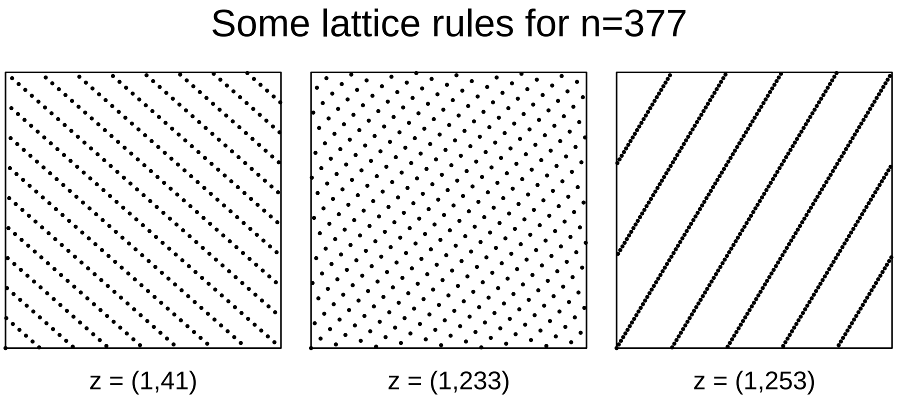

Figure 16.1: Each panel shows a lattice rule $\boldsymbol{x}_i = ((i-1)\boldsymbol{z} \pmod n)/n$ in $[0,1)^2$ with $n = 377$ (a Fibonacci number) and $\boldsymbol{z} = (1, z_2)$. The values of z_2 are, from left to right: 41, 233 and 253. The middle panel is a Fibonacci lattice. The other panels show poor lattices, that one should avoid using.

The numbers $a^j \mathsf{mod}\ 1$ eventually repeat as j increases. The smallest j for which this happens is $j = \phi(n)$ where ϕ is Euler's ***totient function***, the number of integers i from 1 to $n-1$ inclusive with $\gcd(i,n) = 1$. We would not use a Korobov rule with $d \geqslant \phi(n)$, because then we would find that $x_{ij} = x_{i1}$ for all $i = 1, \dots, n$ and $j = \phi(n)$. If $n = p^k$ is a large prime power then it can be shown that $\phi(n) = p^{k-1}(p-1)$, which is also large.

Table 16.1 shows some examples of Korobov rules from L'Ecuyer and Lemieux (2000). They searched for combinations of a and n that produced high quality lattices. We will consider some quality criteria for lattices in 16.4. The quality of a Korobov rule depends on the dimension d. The rules in Table 16.1 were

n	a
1021	76
2039	1487
4093	1516
8191	5130
16381	4026
32749	14251
65521	8950
131071	28823

Table 16.1: This table shows the parameters of some Korobov rules listed in Table 1 of L'Ecuyer and Lemieux (2000). For $k = 10, \dots, 17$, the largest prime $n < 2^k$ is shown along with one of their recommended values a for z_2.

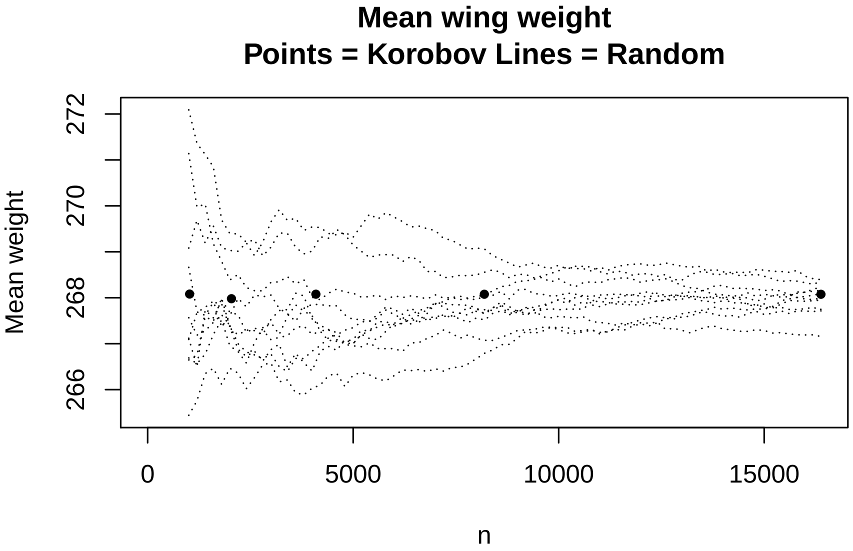

Figure 16.2: The horizontal axis is the sample size n from 1000 to just over 16,000. The vertical axis is the average of the first n wing weight values. Solid points show Korobov values. Ten dotted lines show cumulative Monte Carlo estimates.

constructed using criteria that considered $d \in \{8, 12, 24, 32\}$. With the material presented above it is already possible to implement lattice rules such as those in Table 16.1 to see empirically how they behave.

16.2 Example: wing weight revisited

In §15.6 we compared Halton points to plain Monte Carlo on a wing weight function in 10 dimensions. Here we make the same comparison using Korobov points.

Figure 16.2 shows the results. We don't connect the points between the Korobov estimates because the sequences used there are not extensions of each other. It seems pretty clear from the figure that the Korobov estimates are better than the Monte Carlo ones. The value for $n = 8191$ is pretty close to that for $n = 16{,}381$ and both look to be near the central value that the Monte Carlo points produce. We cannot get a good estimate of the accuracy of a lattice rule from the sample values. Chapter 17 presents randomized QMC which makes it possible to estimate how much better, if any, the lattice rules are than plain MC.

We can see numerical estimates of μ in Table 16.2. Because the values don't

n	Korobov	Halton
1021	268.0803	267.4654
2039	267.9789	267.5688
4093	268.0776	267.8209
8191	268.0763	267.9668
16381	268.0753	268.0193

Table 16.2: Sample size and estimates for the mean wing weight, using both Korobov and Halton points at the given values of n. The Halton estimates are from §15.6.

change much as one moves down the columnn, it appears that the estimates are accurate. Table 16.2 also includes some estimates based on the Halton sequence for these same sample sizes. Judging by eye, it appears that the Korobov points are doing better than the Halton ones, for this function. As we noted in Chapter 15, the Halton points tend to be below 0.5 on average and this function seems to be increasing in most of its input variables. The Korobov rules have mean $1/2-1/(2n)$ in each coordinate. Perhaps this is better than the mean of the Halton points. See Exercise 16.1.

16.3 Lattices and lattice rules

Before presenting the criteria that separate good from bad lattice rules, it is useful to consider lattices in more generality.

Definition 16.1. A ***lattice*** is a nonempty set $L\subset\mathbb{R}^d$ for $d\geqslant 1$, with these properties:
1) $\boldsymbol{x},\boldsymbol{y}\in L$ implies that $\boldsymbol{x}+\boldsymbol{y}\in L$ and $\boldsymbol{x}-\boldsymbol{y}\in L$, and,
2) there is an $\epsilon>0$ such $\|\boldsymbol{x}-\boldsymbol{y}\|>\epsilon$ for all $\boldsymbol{x},\boldsymbol{y}\in L$ with $\boldsymbol{x}\neq\boldsymbol{y}$.

The set of integer vectors $\mathbb{Z}^d$ is a lattice. The set $\{\mathbf{0}\}$ is a lattice by a convention, that we might prefer to call a technicality. It never has distinct points $\boldsymbol{x}$ and $\boldsymbol{y}$ with $\|\boldsymbol{x}-\boldsymbol{y}\|\leqslant\epsilon$ because it has no pairs of distinct points at all. Every other lattice has countably infinitely many points. The set of rational vectors $\mathbb{Q}^d$ is not a lattice because it fails the second clause.

Our lattice rules will always have a finite number of points. We arrange this by intersecting a lattice with the unit cube, as illustrated in Figure 16.3 and defined below.

Definition 16.2. An ***lattice rule*** in dimension $d\geqslant 1$ is a finite set of points formed as $L\cap[0,1)^d$ where L is a lattice such that $\mathbb{Z}^d\subset L$.

Lattice rules are sometimes called ***integration lattices*** but that term is potentially confusing, because these lattice rules are not lattices. They are just the parts of a lattice that lie inside the unit cube $[0,1)^d$. They always have a

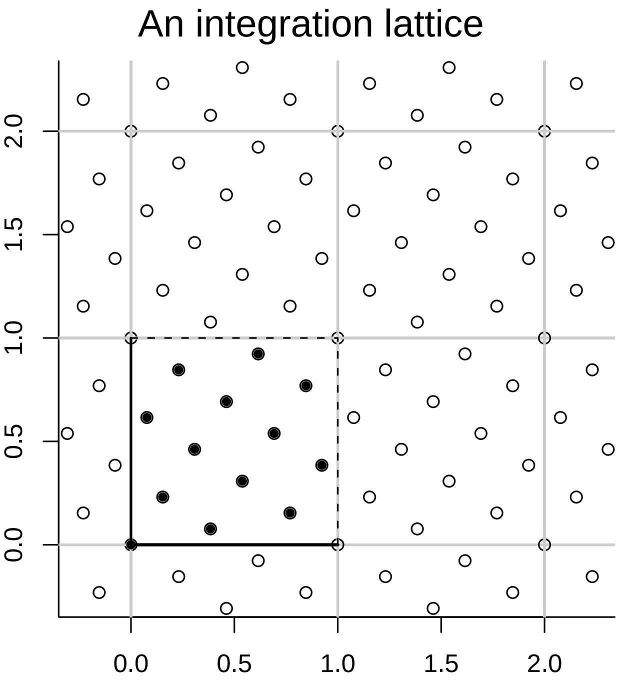

Figure 16.3: The plotted points are a subset of an infinite lattice in the plane. The 13 solid points are a lattice rule, that is, the points of the lattice which belong to $[0,1)^2$.

finite number of points. If they had infinitely many points in $[0,1)^d$, then some pair of them would be closer than ϵ for any $\epsilon > 0$ that we choose.

Definition 16.2 has an extra clause that L must include the integer lattice $\mathbb{Z}^d$. That clause has several functions. First, it rules out some very unsuitable lattices. For example, with $d = 3$ there are lattices that lie completely within a plane, or even a line, such as $\{(i,i,i) \mid i \in \mathbb{Z}\}$. Forcing L to contain $\mathbb{Z}^d$ makes L fully d dimensional. The second advantage of having L contain $\mathbb{Z}^d$ is that if we then shift the points of the lattice L by $\Delta \in \mathbb{R}^d$, the shifted lattice $L + \Delta = \{\boldsymbol{x} + \Delta \mid \boldsymbol{x} \in L\}$ will place the same number of points in $[0,1)^d$ as L does. The lattice $L = \{(i/10, j/\sqrt{5}) \mid i,j \in \mathbb{Z}\}$ does not yield a lattice rule when intersected with $[0,1)^2$, and shifting it can change the number of points that intersect $[0,1)^2$. Finally, for $\Delta \in \mathbb{Z}^d$ we find that $L + \Delta$ is L shifted on top of itself: $L + \Delta = L$. Equivalently, L looks the same in every integer cell $[\boldsymbol{a}, \boldsymbol{a} + \boldsymbol{1})$ for $\boldsymbol{a} \in \mathbb{Z}^d$.

We can recover the lattice L of a given lattice rule by shifting the points of that rule through all possible integer offsets. In Figure 16.3, that operation would produce the open circle points from the solid ones. In the case of a rank-1 lattice defined by n and $\boldsymbol{z} = (1, z_2, \dots, z_d)$, any point in L can be written as a linear combination, with integer coefficients, of the rows of

$$\begin{pmatrix} 1/n & z_2/n & z_3/n & \cdots & z_d/n \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \in \mathbb{R}^{(d+1)\times d}. \tag{16.3}$$

We would not expect to need $d+1$ rows to span a d dimensional space. It is easy to see that the second row of the matrix in (16.3) is an integer linear combination of the other rows with coefficient n on the first row and $-z_j$ on the $j+1$'st row for $j = 2, \dots, d$. We can drop it and generate the lattice using integer linear combinations of the remaining rows. The first row is also a linear combination of the other rows, but it is not an integer linear combination of those other rows, so we cannot drop it and still generate the lattice. After dropping the second row, the lattice L may be written

$$L = \left\{ \sum_{j=1}^{d} a_j \boldsymbol{g}_j \mid \boldsymbol{a} \in \mathbb{Z}^d \right\} \tag{16.4}$$

where the vectors $\boldsymbol{g}_j$ are $\boldsymbol{z}$ and $\boldsymbol{e}_2$ through $\boldsymbol{e}_d$ where $\boldsymbol{e}_j$ has a 1 in the j'th position and is zero elsewhere. Because of (16.4), we say that the vectors $\boldsymbol{g}_j$ generate L.

The matrix

$$A = A(L) = \begin{pmatrix} 1/n & z_2/n & z_3/n & \cdots & z_d/n \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

whose rows are the generating vectors $\boldsymbol{g}_j$ is called a ***generator matrix*** of L. For a nontrivial lattice in dimension $d \geqslant 2$ there is more than one possibility for its generator matrix. The number of points in a lattice rule with generator matrix A is $|\det(A)|^{-1}$. This is easy to verify for the matrix above. Geometrically it is reasonable: if we take a large bounded cubical region R in $\mathbb{R}^d$ and map it via A^T to $\widetilde{R} = \{A^\mathsf{T}\boldsymbol{x} \in \mathbb{R}^d \mid \boldsymbol{x} \in R\}$ then the mapping has Jacobian A^T and so $\mathbf{vol}(\widetilde{R}) = \mathbf{vol}(R)|\det(A^\mathsf{T})| = \mathbf{vol}(R)|\det(A)|$. As a result, the image $\widetilde{R}$ should have about $1/|\det(A)|$ times as many integer lattice points in it as R has.

16.4 Quality criteria for lattices

For $d = 1$, the rank-1 lattice rule reduces to an equal weight rule with evaluation points i/n for $i = 0, \dots, n-1$. This is a left endpoint rule. For $d \geqslant 1$ all the univariate projections of $\boldsymbol{x}_i$ are left endpoint rules. Left endpoint rules typically have error $O(1/n)$ while midpoint rules can attain error $O(1/n^2)$. See Chapter 7. Things change for smooth periodic functions f with period 1. Then a left endpoint rule is equivalent to a trapezoid rule with evaluation points i/n for $i = 0, \dots, n$ and relative weights $1/2, 1, 1, \dots, 1, 1/2$. Trapezoid rules are very accurate for smooth functions. If f'' is continuous on $[0, 1]$, then the trapezoid rule has error $O(n^{-2})$ just like the midpoint rule.

A strategy for designing lattice rules is as follows. Because lattice rules are extremely well suited to periodic functions, we will first suppose that f is a periodic function on $\mathbb{R}^d$, with period 1 as defined below. Then we develop an upper bound for the error $|\hat{\mu}_{\mathrm{rlat}} - \mu|$ when we use a lattice rule to integrate a periodic function. Next we select rank-1 lattices for which the error bound is small. Of course a problem remains: we cannot count on the real world problem we face to involve a periodic function f. Therefore we look for ways to make our function periodic. That is, we replace f by a periodic function $\widetilde{f}$ constructed so that $\int \widetilde{f}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \mu = \int f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$. Finally, we estimate μ by $(1/n)\sum_{i=1}^n \widetilde{f}(\boldsymbol{x}_i)$.

Definition 16.3. The function $f : \mathbb{R}^d \to \mathbb{R}$ is ***periodic*** (with period 1) if $f(\boldsymbol{x} + \boldsymbol{z}) = f(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{z} \in \mathbb{Z}^d$.

If f is periodic and $\boldsymbol{x}_i$ are from a rank one lattice (16.1), then

$$\hat{\mu}_{\mathrm{rlat}} = \frac{1}{n}\sum_{i=0}^{n-1} f\Bigl(\Bigl\{\frac{i}{n}\boldsymbol{z}\Bigr\}\Bigr) = \frac{1}{n}\sum_{i=0}^{n-1} f\Bigl(\frac{i}{n}\boldsymbol{z}\Bigr).$$

We don't have to reduce the argument of f modulo 1 because f is periodic.

We study lattice methods by expanding f into a d-dimensional Fourier series. Fourier series are commonly defined for functions on $(-\pi, \pi]^d$ or $[0, 2\pi)^d$. For QMC, it is more convenient to work with functions on $[0, 1)^d$ by scaling $\boldsymbol{x}$. For each vector of integers $\boldsymbol{h} \in \mathbb{Z}^d$ we define the function $\psi_{\boldsymbol{h}}(\boldsymbol{x}) = e^{2\pi\sqrt{-1}\boldsymbol{h}^\mathsf{T}\boldsymbol{x}} = \cos(2\pi\boldsymbol{h}^\mathsf{T}\boldsymbol{x}) + \sqrt{-1}\sin(2\pi\boldsymbol{h}^\mathsf{T}\boldsymbol{x})$. These functions are periodic. They are orthonormal in that

$$\int_{[0,1)^d} \psi_{\boldsymbol{h}'}(\boldsymbol{x})\overline{\psi}_{\boldsymbol{h}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \begin{cases} 1, & \boldsymbol{h} = \boldsymbol{h}', \\ 0, & \text{else.} \end{cases}$$

Here $\overline{\psi}_{\boldsymbol{h}}(\boldsymbol{x})$ is the complex conjugate of $\psi_{\boldsymbol{h}}(\boldsymbol{x})$, and $\overline{\psi}_{\boldsymbol{h}}(\boldsymbol{x}) = \psi_{-\boldsymbol{h}}(\boldsymbol{x})$.

The Fourier coefficients of f are defined to be

$$\hat{f}(\boldsymbol{h}) = \int_{[0,1)^d} f(\boldsymbol{x})\psi_{-\boldsymbol{h}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \quad \boldsymbol{h} \in \mathbb{Z}^d, \tag{16.5}$$

and the Fourier series for f is

$$\tilde{f}(\boldsymbol{x}) = \sum_{\boldsymbol{h}\in\mathbb{Z}^d} \hat{f}(\boldsymbol{h})\psi_{\boldsymbol{h}}(\boldsymbol{x}). \tag{16.6}$$

We say that $\tilde{f}$ represents f. Our study of lattice rules would be simpler if $f(\boldsymbol{x}) = \tilde{f}(\boldsymbol{x})$ always held, but reality is more complicated, as we discuss next.

If we were to change f at a finite number of points $\boldsymbol{x}_k \in [0,1)^d$, producing say a function g, then none of the $\hat{f}(\boldsymbol{h})$ would change and hence $\tilde{f}$ would not change either, yet $\tilde{f}$ could not then equal both f and g. We will say that functions f and g on $[0,1)^d$ are ***equal with probability one*** (abbreviated w.pr. 1) if $\mathbb{P}(f(\boldsymbol{x}) \neq g(\boldsymbol{x})) = 0$ for $\boldsymbol{x} \sim \mathbf{U}[0,1)^d$. If we did change f to g at a finite number of points then we would have $f = g$ with probability one.

As remarked above, if f and g are integrable functions that are equal with probability one, then they have the same Fourier coefficients. Conversely, if f and g are integrable functions that have the same Fourier coefficients, then they are equal with probability one (Grafakos, 2004, Proposition 3.1.13). Next we have a condition for such equality.

Theorem 16.1. *Let f be an integrable function on $[0,1)^d$. If*

$$\sum_{\boldsymbol{h}\in\mathbb{Z}^d} |\hat{f}(\boldsymbol{h})| < \infty, \tag{16.7}$$

then $f(\boldsymbol{x}) = \tilde{f}(\boldsymbol{x})$ from (16.6) with probability one.

Proof. This is Proposition 3.1.14 of Grafakos (2004). □

Condition (16.7) is that the Fourier coefficients of f are absolutely summable. We will then say that f has an ***absolutely convergent Fourier series***. By Theorem 16.1, when f has an absolutely convergent Fourier series, it equals that Fourier series with probability one. In that case, integrating $\tilde{f}$ will give us the integral of f.

The sufficient conditions to get absolute convergence can be strict. One sufficient condition described in the end notes involves even more smoothness than having continuous partial derivatives of all orders up to $d/2$. Below we will use some derivatives of order d or higher.

Some weaker smoothness conditions with a weaker connection between f and $\tilde{f}$ are useful. For integers $N \geqslant 0$, define the truncated Fourier series

$$\tilde{f}_N(\boldsymbol{x}) = \sum_{\boldsymbol{h}\in\mathbb{Z}^d, |h_j|\leqslant N} \hat{f}(\boldsymbol{x})\psi_{\boldsymbol{h}}(\boldsymbol{x}).$$

The next theorem describes a mean square convergence property of square integrable f. Theorem 17.2 for randomized lattice rules in Chapter 17 only requires mean square integrability of f, not absolute converence of $\tilde{f}$.

Theorem 16.2. *Let f be a square integrable function on $[0,1)^d$. Then*

$$\lim_{N\to\infty} \tilde{f}_N(\boldsymbol{x}) = f(\boldsymbol{x}), \quad \textit{w.pr. 1}, \quad \textit{and} \tag{16.8}$$

$$\lim_{N\to\infty} \int_{[0,1)^d} (\tilde{f}_N(\boldsymbol{x}) - f(\boldsymbol{x}))^2 \,\mathrm{d}\boldsymbol{x} = 0. \tag{16.9}$$

Proof. This is Proposition 3.1.16, part 2, of Grafakos (2004). □

The 'with probability one' clause in (16.8) covers the possibility that for some points $\boldsymbol{x}$, the values $\tilde{f}_N(\boldsymbol{x})$ may fail to converge as $N \to \infty$. The set of $\boldsymbol{x}$ where that happens has probability zero. When the sequence does converge, it converges to $f(\boldsymbol{x})$ (Grafakos, 2004, Proposition 3.1.15). We won't always add the "with probability one" clause, nor keep saying that f must be integrable.

Next, we develop lattice rules assuming that $\sum_{\boldsymbol{h}\in\mathbb{Z}^d} |\hat{f}(\boldsymbol{h})| < \infty$ so that integrating $\tilde{f}$ is the same as integrating f. Substituting the Fourier representation of such an f into the rank-1 lattice rule, and reversing the order of summation, we find that

$$\hat{\mu}_{\mathrm{rlat}} = \sum_{\boldsymbol{h}\in\mathbb{Z}^d} \hat{f}(\boldsymbol{h})\,\frac{1}{n}\sum_{i=1}^n \psi_{\boldsymbol{h}}(\boldsymbol{x}_i). \tag{16.10}$$

Proposition 16.1. *Let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in [0,1)^d$ be a rank-1 lattice rule defined by equation (16.1) using $\boldsymbol{z} \in \mathbb{Z}^d$. Then*

$$\frac{1}{n}\sum_{i=1}^n \psi_{\boldsymbol{h}}(\boldsymbol{x}_i) = \begin{cases} 1, & \boldsymbol{h}^\mathsf{T}\boldsymbol{z} = 0 \mod n\\ 0, & \text{else.}\end{cases}$$

Proof. Expanding the left hand side and using periodicity of $\psi_{\boldsymbol{h}}$,

$$\frac{1}{n}\sum_{i=1}^n \psi_{\boldsymbol{h}}(\boldsymbol{x}_i) = \frac{1}{n}\sum_{i=0}^{n-1} \psi_{\boldsymbol{h}}\Bigl(\frac{i\boldsymbol{z}}{n}\Bigr) = \frac{1}{n}\sum_{i=0}^{n-1} \exp\Bigl(2\pi\sqrt{-1}\,\frac{i\boldsymbol{h}^\mathsf{T}\boldsymbol{z}}{n}\Bigr) = \frac{1}{n}\sum_{i=0}^{n-1}\omega^i$$

where $\omega = \exp(2\pi\sqrt{-1}\boldsymbol{h}^\mathsf{T}\boldsymbol{z}/n)$. If $\boldsymbol{h}^\mathsf{T}\boldsymbol{z} = 0 \mod n$, then $\boldsymbol{h}^\mathsf{T}\boldsymbol{z}/n \in \mathbb{Z}$ so that $\omega = 1$, and the first case is proved. Now suppose that $\boldsymbol{h}^\mathsf{T}\boldsymbol{z} \neq 0 \mod n$. Then $\omega \neq 1$ and so

$$\frac{1}{n}\sum_{i=0}^{n-1}\omega^i = \frac{1-\omega^n}{1-\omega} = 0,$$

because then $\omega^n = \exp(2\pi\sqrt{-1}\boldsymbol{h}^\mathsf{T}\boldsymbol{z}) = 1$, proving the second case. □

From Proposition 16.1, we find that

$$\hat{\mu}_{\mathrm{rlat}} = \sum_{\boldsymbol{h}\in\mathbb{Z}^d} \hat{f}(\boldsymbol{h})\frac{1}{n}\sum_{i=1}^n \psi_{\boldsymbol{h}}(\boldsymbol{x}_i) = \sum_{\boldsymbol{h}\in L^\perp} \hat{f}(\boldsymbol{h})$$

where

$$L^\perp = L^\perp(\boldsymbol{z}) = \{\boldsymbol{h}\in\mathbb{Z}^d \mid \boldsymbol{h}^\mathsf{T}\boldsymbol{z} = 0 \mod n\}. \tag{16.11}$$

A similar expansion of μ when $\tilde{f}$ is absolutely convergent yields

$$\mu = \sum_{\boldsymbol{h}\in\mathbb{Z}^d} \hat{f}(\boldsymbol{h}) \int_{[0,1)^d} \psi_{\boldsymbol{h}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \hat{f}(\boldsymbol{0}),$$

because $\psi_{\boldsymbol{h}}$ integrates to 1 if $\boldsymbol{h} = \boldsymbol{0}$ and integrates to 0 otherwise. As a result, the error in the lattice rule is

$$\hat{\mu}_{\mathrm{rlat}} - \mu = \sum_{\boldsymbol{h} \in L_*^{\perp}} \hat{f}(\boldsymbol{h}) \tag{16.12}$$

where

$$L_*^{\perp} = \{\boldsymbol{h} \in L^{\perp} \mid \boldsymbol{h} \neq \boldsymbol{0}\}. \tag{16.13}$$

The set $L^{\perp}$ is a lattice (Exercise 16.2). It is known as the ***dual lattice*** of L. Its nonzero elements comprise the Fourier coefficients $\boldsymbol{h}$ for which the lattice rule gets $\int \psi_{\boldsymbol{h}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ completely wrong. In the words of Sloan and Joe (1994, page 32) "the dual lattice represents a graphic picture of failure". The generator matrix for the dual lattice $L^{\perp}$ is $(A(L)^{\mathsf{T}})^{-1}$ where $A(L)$ is the generator matrix of L. Figure 16.4 depicts three small lattices with their corresponding dual lattices.

A good lattice for f has a small value for the infinite sum in (16.12) of Fourier coefficients $\hat{f}(\boldsymbol{h})$. A good latice overall has a small infinite sum for a large collection of integrands f that we wish to handle.

A vector $\boldsymbol{h}$ that is far from the origin corresponds to a sinusoidal function $\psi_{\boldsymbol{h}}(\boldsymbol{x})$ that oscillates very quickly. Suppose that f is smooth and nearly constant over some rectangular box $R \subset [0,1)^d$. For large enough $\boldsymbol{x}$ the function $\psi_{\boldsymbol{h}}$ is a rapidly oscillating sinusoid over R and then $f(\boldsymbol{x})\psi_{\boldsymbol{h}}(\boldsymbol{x})$ will have a near zero integral over R. We would then expect $|\hat{f}(\boldsymbol{h})|$ to be small. The more derivatives f has, the faster $|\hat{f}(\boldsymbol{h})|$ must decay. The Riemann-Lebesgue theorem has $|\hat{f}(\boldsymbol{x})| \to 0$, without requiring f to be differentiable at all.

Theorem 16.3. *Let f be integrable on $[0,1)^d$. Then $|\hat{f}(\boldsymbol{h})| \to 0$ as $\|\boldsymbol{h}\| \to \infty$. If*

$$\frac{\partial^{q_1+\cdots+q_d} f}{\partial x_1^{q_1} \cdots \partial x_d^{q_d}}(\boldsymbol{x})$$

exists and is integrable for any $q_j \geqslant 0$ with $\sum_{j=1}^d q_j \leqslant \alpha$, then

$$|\hat{f}(\boldsymbol{h})|(1 + \|\boldsymbol{h}\|^{\alpha}) \to 0$$

as $\|\boldsymbol{h}\| \to 0$.

Proof. The first part is the Riemann-Lebesgue theorem, Proposition 3.2.1 of Grafakos (2004). The second part is from Theorem 3.2.9 of Grafakos (2004). □

We see from Theorem 16.3 that the large errors will come from $\boldsymbol{h}$ close to zero and so we should prefer a lattice L where the vectors in $L_*^{\perp}$ are far from the origin. For $d = 1$, the measure of large or small $\boldsymbol{h}$ is simply $|h_1|$. For $d \geqslant 2$, there is no unique way to order the nonzero vectors $\boldsymbol{h}$. Lyness (1989) lists the three most commonly studied measures

$$P(\boldsymbol{h}) = \prod_{j=1}^d \max(1, |h_j|), \quad S(\boldsymbol{h}) = \sum_{j=1}^d |h_j| \quad \text{and} \quad M(\boldsymbol{h}) = \max_{1 \leqslant j \leqslant d} |h_j|,$$

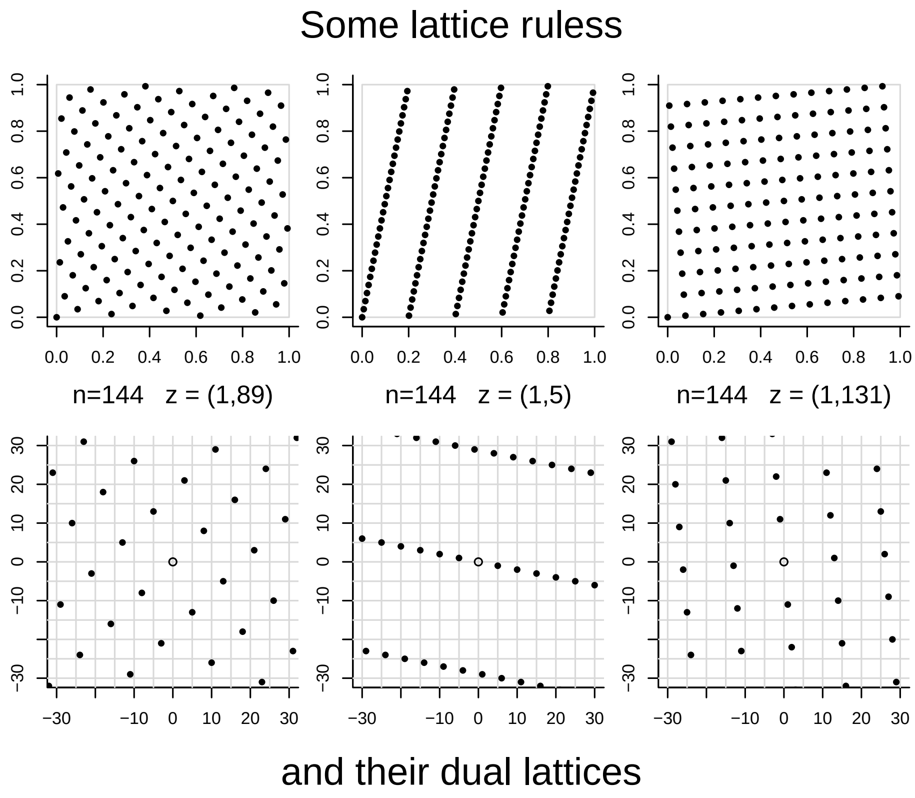

Figure 16.4: The top row shows lattice rules in the unit square with $n = 144$, $z_1 = 1$ and z_2 equal to 89, 5, and 131 from left to right. The bottom row shows the corresponding dual lattices with reference lines at multiples of 5. A good lattice rule, like the one on the left, has few non-zero points near the origin in its dual lattice.

which use $|h_j|$ within a product, sum and maximum, respectively. The measure P is most commonly used. Lyness (1989) remarks that S may be a good choice for extremely smooth (analytic) periodic functions and that M has little to recommend it. The criterion $P(\boldsymbol{h})$ can be written

$$P(\boldsymbol{h}) = \prod_{j=1}^{d} \bar{h}_j, \quad \text{for} \quad \bar{h} = \max(1, |h|).$$

Usage of $P(\boldsymbol{h})$ may be justified by Zaremba's theorem.

Theorem 16.4 (Zaremba's Theorem)**.** *Let f be a periodic function on $\mathbb{R}^d$ and let $\alpha > 1$ be an integer. Suppose that*

$$\frac{\partial^{q_1+\cdots+q_d} f}{\partial x_1^{q_1} \cdots \partial x_d^{q_d}}$$

exists and has bounded variation over $[0,1]^d$ *in the sense of Hardy and Krause for any integers* $q_j \geqslant 0$ *with* $\sum_{j=1}^d q_j \leqslant \alpha - 1$. *Then for some* $c > 0$

$$|\hat{f}(\boldsymbol{h})| \leqslant cP(\boldsymbol{h})^{-\alpha}. \tag{16.14}$$

Niederreiter (1992b) states this theorem and also notes that we can replace bounded variation whenever $\sum_{j=1}^d q_j \leqslant \alpha - 1$ by continuous differentiability whenever $\sum_{j=1}^d q_j \leqslant \alpha$.

Definition 16.4. For $c > 0$ and $\alpha > 1$, let $\mathcal{E}_\alpha(c)$ be the set of periodic functions f on $\mathbb{R}^d$ with period 1, according to Definition 16.3, for which (16.14) holds for all $\boldsymbol{h} \in \mathbb{Z}^d$.

Any $f \in \mathcal{E}_\alpha(c)$ has an absolutely convergent Fourier series, and

$$|\hat{\mu} - \mu| \leqslant c \sum_{\boldsymbol{h} \in L_*^\perp} P(\boldsymbol{h})^{-\alpha}. \tag{16.15}$$

A lattice that is good for $c = 1$ will be good for any $c > 0$ because μ, $\hat{\mu}$ and $\hat{f}(\boldsymbol{h})$ all scale by a multiple of $1/c$ when f is replaced by f/c, so we focus now on finding a good lattice for $f \in \mathcal{E}_\alpha(1)$.

For the function

$$f_{\alpha,n}(\boldsymbol{x}) = \sum_{\boldsymbol{h} \in \mathbb{Z}^d} \frac{e^{2\pi\sqrt{-1}\boldsymbol{h}^\mathsf{T}\boldsymbol{x}}}{(\bar{h}_1\bar{h}_2\cdots\bar{h}_d)^\alpha}, \tag{16.16}$$

equality holds in (16.15) for $c = 1$. Thus $f_{\alpha,n}$ is a worst case integrand from $\mathcal{E}_\alpha(1)$ and the worst error is then

$$P_\alpha(\boldsymbol{z}; n) \equiv \sum_{\boldsymbol{h} \in L_*^\perp} \prod_{j=1}^d \bar{h}_j^{-\alpha} = \frac{1}{n}\sum_{i=0}^{n-1} f_{\alpha,n}\Bigl(\Bigl\{\frac{i\boldsymbol{z}}{n}\Bigr\}\Bigr) - 1. \tag{16.17}$$

The second expression for $P_\alpha(\boldsymbol{z}; n)$ is convenient because the definition of $f_{\alpha,n}$ sums over $\boldsymbol{h}$ over $\mathbb{Z}^d$ instead of $L_*^\perp$. Now for a given n, we look for $\boldsymbol{h} \in \mathbb{Z}^d$ to minimize

$$\begin{aligned} 1 + P_\alpha(\boldsymbol{z}; n) &= \frac{1}{n}\sum_{i=0}^{n-1}\sum_{\boldsymbol{h} \in \mathbb{Z}^d} \frac{e^{2\pi\sqrt{-1}\boldsymbol{h}^\mathsf{T}\boldsymbol{z}i/n}}{(\bar{h}_1\bar{h}_2\cdots\bar{h}_d)^\alpha} \\ &= \frac{1}{n}\sum_{i=0}^{n-1}\prod_{j=1}^d\Bigl(1 + \sum_{h \neq 0}\frac{e^{2\pi\sqrt{-1}hz_ji/n}}{|h|^\alpha}\Bigr). \end{aligned} \tag{16.18}$$

If $\alpha \geqslant 2$ is an even integer, then the infinite sum in (16.18) can be written in terms of certain Bernoulli polynomials b_α of degree α. See Sloan and Joe (1994, Appendix C). The case $\alpha = 2$ is most frequently used. There

$$1 + P_2(\boldsymbol{z}; n) = \frac{1}{n}\sum_{i=0}^{n-1}\prod_{j=1}^d\bigl\{1 + 2\pi^2(x_{ij}^2 - x_{ij} + 1/6)\bigr\}$$

$$= \frac{1}{n}\sum_{i=0}^{n-1}\prod_{j=1}^{d}\Bigl\{1+2\pi^2\Bigl(\Bigl(\frac{iz_j}{n}\Bigr)^2-\frac{iz_j}{n}+\frac{1}{6}\Bigr)\Bigr\}, \tag{16.19}$$

where as before $\{y\} = y - \lfloor y \rfloor$. Good values for $\boldsymbol{z}$ are then found by computerized search. See §16.7.

16.5 Convergence rates

As $\alpha > 1$ increases, lattice rules can achieve much better convergence rates for $f \in \mathcal{E}_\alpha(1)$. Here we summarize some of those results. Following Niederreiter (1993), we consider $\boldsymbol{z}$ belonging to

$$\mathcal{G}_d(n) = \{\boldsymbol{z} \in \mathbb{Z}^d \mid -n/2 < z_j \leqslant n/2,\ 1 \leqslant j \leqslant d\}.$$

Theorem 16.5. *For every $d \geqslant 2$ and $n \geqslant 2$ and $\alpha > 1$, there exists $\boldsymbol{z} \in \mathcal{G}_d(n)$ with*

$$P_\alpha(\boldsymbol{z};n) \leqslant c(d,\alpha)\frac{\log(n)^{\alpha(d-1)+1}}{n^\alpha}\Bigl(\frac{n}{\phi(n)}\Bigr)^{(\alpha-1)(d-1)}\Bigl(1+O\Bigl(\frac{(\log\log n)^{b(d)}}{\log n}\Bigr)\Bigr)$$

where

$$c(d,\alpha) = 2^{\alpha(d-1)+1}\alpha\Bigl(\frac{\alpha}{(d-1)!(\alpha-1)}\Bigr)^{\alpha-1}$$

with $b(2) = 3$ and $b(d) = d-1$ for $d \geqslant 3$, and $\phi(n)$ is Euler's totient function.

Proof. This is Theorem 1 of Niederreiter (1993). □

A lattice rule with $-n/2 < z_j < 0$ gives the same points as one using $z_j + n$ instead, because $(i-1)z_j \equiv (i-1)(z_j+n) \mathsf{\ mod\ } n$. As a result, Theorem 16.5 also holds when $\mathcal{G}_g(n)$ is replaced by $\{\boldsymbol{z} \in \mathbb{Z}^d \mid 0 \leqslant z_j < n\}$.

Theorem 16.5 shows that we can get error $O(n^{-\alpha+\epsilon})$ for any $\epsilon > 0$ from a lattice rule if $f \in \mathcal{E}_\alpha(c)$ for some $c < \infty$. The factor $n/\phi(n) = n/(n-1)$ if n is prime and $2^k/\phi(2^k) = 2$, so powers of 2 are also reasonable choices, though they do introduce a factor $2^{(\alpha-1)(d-1)}$ that is not present for prime numbers. The constant $c(d,\alpha)$ decreases rapidly with d. Niederreiter (1993, Theorem 2) also shows that the usual practice of taking $z_1 = 1$ does not greatly change the bound.

Niederreiter (1992b) uses an alternative to $P_\alpha(\boldsymbol{z};n)$, that we can write as

$$R_\alpha(\boldsymbol{z};n) = \sum_{\boldsymbol{h}\in L_*^\perp\cap(-n/2,n/2]^d}\prod_{j=1}^{d}\bar{h}_j^{-\alpha}. \tag{16.20}$$

While P_α is only defined for $\alpha > 1$, he makes use of $R_1(\boldsymbol{z};n)$ and notes that $R_\alpha(\boldsymbol{z};n) \leqslant R_1(\boldsymbol{z};n)^\alpha$, for $\alpha > 1$. For $\alpha > 1$ and $\boldsymbol{z} \in \mathcal{G}_d(n)$,

$$R_\alpha(\boldsymbol{z};n) \leqslant P_\alpha(\boldsymbol{z};n) \leqslant R_\alpha(\boldsymbol{z};n) + O(n^{-\alpha})$$

holds. The upper bound is from Niederreiter (1993, Lemma 2). In a non-asymptotic result (Niederreiter, 1992b, page 115), the average of $R_1(\boldsymbol{z};n)$ over $\boldsymbol{z} \in \mathcal{G}_d(n)$ is below $(2\log(n)+7/5)^d$ for all $d \geqslant 2$ and all $n \geqslant 2$. Larcher (1987) proves that $R_1(\boldsymbol{z};n) \geqslant c_d(\log n)^d/n$ always holds for some $c_d > 0$ when $n \geqslant 2$ and $d \geqslant 2$. It then follows that $P_1(\boldsymbol{z};n)$ cannot be $o((\log n)^d/n)$.

For Korobov rules, the search space is smaller. For $0 \leqslant a < n$, let $\boldsymbol{z}(a) = (1, a, a^2 \mathsf{\ mod\ } n, \dots, a^{n-1} \mathsf{\ mod\ } n)$ over $a < n$. If n is prime and $d \geqslant 2$, then the average value of $R_1(\boldsymbol{z};n)$, for

$$\frac{1}{n}\sum_{a=0}^{n-1} R_1(\boldsymbol{z}(a);n) < \frac{d-1}{n}(2\log(n)+1)^d,$$

by Niederreiter (1992b, Theorem 5.18). Of course, one would never use $a = 0$.

It is worth remembering that a better rate in n does not mean a better method for practical sample sizes. First, the implied constant can increase rapidly with smoothness α. Also, when the result is asymptotic, the accuracy it presents might not be a good description of attained accuracy for feasible sample sizes n.

We can bound the discrepancy of a rank-1 lattice rule.

Theorem 16.6. *For $\boldsymbol{z} \in \mathbb{Z}^d$ for $d \geqslant 2$, and integer $n \geqslant 2$, let $\boldsymbol{x}_i = \{(i-1)\boldsymbol{z}/n\}$ for $i = 1, \dots, n$. Then*

$$D_n(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \leqslant \frac{d}{n} + \frac{1}{2}R_1(\boldsymbol{z};n),$$

where R_1 is from (16.20) *with $\alpha = 1$.*

Proof. This is from Theorem 5.6 of Niederreiter (1992b). □

16.6 Periodizing transformations

Lattice rules are designed for smooth periodic integrands on $\mathbb{R}^d$. Given a smooth integrand f on $[0,1)^d$, the natural way to extend it to $\mathbb{R}^d$ is given in Definition 16.5 below, but the result isn't necessarily smooth. Here we consider ways to define an integrand $\widetilde{f}$ satisfying

$$\int_{[0,1)^d} \widetilde{f}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_{[0,1)^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

and for which the natural periodic extension of $\widetilde{f}$ is smooth.

Definition 16.5. Given $f : [0,1)^d \to \mathbb{R}$ the ***periodic extension*** of f to $\mathbb{R}^d$ is $f(\{\boldsymbol{x}\})$ applied componentwise.

We begin with $d = 1$. Consider the function $(x - 1/4)^3$ on $[0,1)$. This function is very smooth on that interval but its periodic extension has discontinuities at every $x \in \mathbb{Z}$. The first panel of Figure 16.5 shows $(x-1/4)^3$ overlaid on a portion of its periodic extension.

For a function f on $[0,1)$ to have a continuous periodic extension requires that $\lim_{x\to 1-} f(x) = f(0)$ in addition to continuity for $0 < x < 1$. It is convenient to refer to $f(1) = f^{(0)}(1)$, $f'(1)$ and $f^{(r)}(1)$ which mean, respectively, $f(1-) = \lim_{x\to 1-} f(x)$, $f'(1-) = \lim_{x\to 1-} f'(x)$ and $f^{(r)}(1-) = \lim_{x\to 1-} f^{(r)}(x)$. With this understanding, for f to extend to a function with r continuous derivatives the required boundary condition is $f^{(j)}(0) = f^{(j)}(1)$ for $0 \leqslant j \leqslant r$. More generally, when f is defined on $[0,1)^d$ we treat any boundary point by taking limits as necessary.

We consider several ways to replace f by a suitable $\widetilde{f}$ for $d=1$. They are distinguished by how well they preserve smoothness of f and by how effective they are in higher dimensions.

The function $\widetilde{f}(x) \equiv f_{\mathrm{E}}(x) = (f(x)+f(1-x))/2$ extends to a one-periodic function on $\mathbb{R}$. This function was used in §8.2 on antithetic sampling. By symmetry $f_{\mathrm{E}}(0) = f_{\mathrm{E}}(1)$. The derivative of f_{E} is $f_{\mathrm{E}}'(x) = (f'(x) - f'(1-x))/2$. For $f_{\mathrm{E}}'(x)$ to extend continuously, we need $f'(0) = f'(1)$.

Antithetic periodization is awkward to extend to higher dimensions. The function $(f(\boldsymbol{x}) + f(\tilde{\boldsymbol{x}}))/2$ with $\tilde{\boldsymbol{x}} = \boldsymbol{1} - \boldsymbol{x}$ taken componentwise is not necessarily periodic (Exercise 16.3). The ***reflection method*** produces a d-dimensional periodization of f by averaging 2^d reflections of $\boldsymbol{x}$. Letting $g_0(x) = x$ and $g_1(x) = 1-x$, we take

$$\begin{aligned}\widetilde{f}_{\mathrm{refl}}(\boldsymbol{x}) &= \frac{1}{2^d}\sum_{j_1=0}^{1}\cdots\sum_{j_d=0}^{1} f\bigl(g_{j_1}(x_1), g_{j_2}(x_2), \dots, g_{j_d}(x_d)\bigr)\\ &= \frac{1}{2^d}\sum_{u\subseteq\{1,\dots,d\}} f(\tilde{\boldsymbol{x}}_u{:}\boldsymbol{x}_{-u})\end{aligned}$$

where $\tilde{\boldsymbol{x}}_u{:}\boldsymbol{x}_{-u}$ is the point $\boldsymbol{z}$ with $z_j = 1 - x_j$ for $j \in u$ and $z_j = x_j$ for $j \notin u$. By symmetry, $\int \widetilde{f}_{\mathrm{refl}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$. The obvious disadvantage of reflection is that to compute $\widetilde{f}_{\mathrm{refl}}$ at one point requires 2^d evaluations of f.

The reflection method does not yield a very smooth function. We saw this already for $d=1$ where $f_{\mathrm{refl}} = f_{\mathrm{E}}$. The second panel of Figure 16.5 shows the reflection periodization of $(x-1/4)^3$.

A second way to periodize a function uses the baker transformation which is defined in terms of the ***tent function***

$$t(x) = \min(2x, 1-2x) = 1 - 2|x - 1/2| = \begin{cases} 2x, & 0 \leqslant x \leqslant 1/2\\ 2(1-x), & 1/2 \leqslant x \leqslant 1.\end{cases} \qquad (16.21)$$

The version using absolute value can be useful when writing software. The absolute value function does the testing of $x \leqslant 1/2$ for us, and we don't then need to put a conditional branch in our code. The tent function is also called the hat function.

Definition 16.6. The d-dimensional ***baker transformation*** is a function $B : [0,1]^d \to [0,1]^d$ with $B(\boldsymbol{x}) = \widetilde{\boldsymbol{x}}$ where $\tilde{x}_j = t(x_j)$ from (16.21), for $j = 1, \dots, d$. For $f : [0,1]^d \to \mathbb{R}$, the ***baker periodization of f*** is

$$\widetilde{f}_{\text{baker}}(\boldsymbol{x}) = f(B(\boldsymbol{x})). \tag{16.22}$$

The name "baker" comes from the resemblance of this function to the folding or kneading of bread dough, especially for $d = 2$. The function $\widetilde{f}_{\text{baker}}(\boldsymbol{x})$ has a continuous periodic extension. As x goes from 0 to 1, the value of $t(x)$ goes from 0 to 1 and then back to 0. Thus $\widetilde{f}_{\text{baker}}(0) = \widetilde{f}_{\text{baker}}(1) = f(0)$.

The baker periodization satisfies $\int \widetilde{f}_{\text{baker}}(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} = \int f(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}$ (Exercise 16.4). We can compute $\widetilde{f}_{\text{baker}}(\boldsymbol{x})$ with only one evaluation of f instead of the 2^d required for $\widetilde{f}_{\text{refl}}$.

The third panel of Figure 16.5 shows the periodic extension of the baker periodization of $(x - 1/4)^3$. It clearly has cusps (first derivative discontinuities) at 0 and 1. By the chain rule, $\widetilde{f}'_{\text{baker}}(0) = 2f'(0)$ and $\widetilde{f}'_{\text{baker}}(1) = -2f'(0)$ so that we expect these cusps whenever $f'(0) \neq f'(1)$. There is an additional cusp at $x = 1/2$. This arises because $\widetilde{f}'_{\text{baker}}(1/2-) = f'(1)$ while $\widetilde{f}'_{\text{baker}}(1/2+) = -f'(1)$. For $d > 1$ there are cusps in $\widetilde{f}_{\text{baker}}$ at points $\boldsymbol{x}$ where $x_j = 0$ or $1/2$ or 1 for one or more $j \in \{1, \dots, d\}$.

A third periodization method is to subtract certain polynomials from f in order to get the desired number of smooth derivatives at integer values of the periodization. For continuity of $\widetilde{f}$ (i.e. the zeroth derivative) when $d = 1$ we may use the ***linear periodization***

$$\widetilde{f}_{\text{linear}}(x) = f(x) - (f(1) - f(0))(x - 1/2).$$

Clearly $\int_0^1 \widetilde{f}_{\text{linear}}(x) \,\mathrm{d}x = \int_0^1 f(x) \,\mathrm{d}x$ because $x - 1/2$ integrates to 0. Also $\widetilde{f}_{\text{linear}}(0) = \widetilde{f}_{\text{linear}}(1) = (f(0) + f(1))/2$. The fourth panel of Figure 16.5 shows the periodic extension of the linear periodization of $(x - 1/4)^3$.

The linear periodization can be extended to smooth functions on $[0,1)$. Sloan and Joe (1994) describe

$$\widetilde{f}_{\text{Bern}}(x) = f(x) - \sum_{j=1}^{r} \bigl(f^{(j-1)}(1) - f^{(j-1)}(0)\bigr) b_j(x)$$

where b_j are the Bernoulli polynomials, which they define in their Appendix C. Unfortunately, higher dimensional generalizations of the Bernoulli periodization method are awkward already for $d = 2$ and their complexity grows exponentially in d. We do not consider them further.

Of the methods considered above, the baker transformation method is clearly the best for large d. It does not give rise to smooth integrands. It is possible to obtain smooth periodic integrands using a change of variable formula.

Let $\phi(x)$ be a differentiable increasing function from $[0,1]$ onto $[0,1]$. Then

$$\int_0^1 f(x) \,\mathrm{d}x = \int_0^1 f(\phi(x))\phi'(x) \,\mathrm{d}x. \tag{16.23}$$

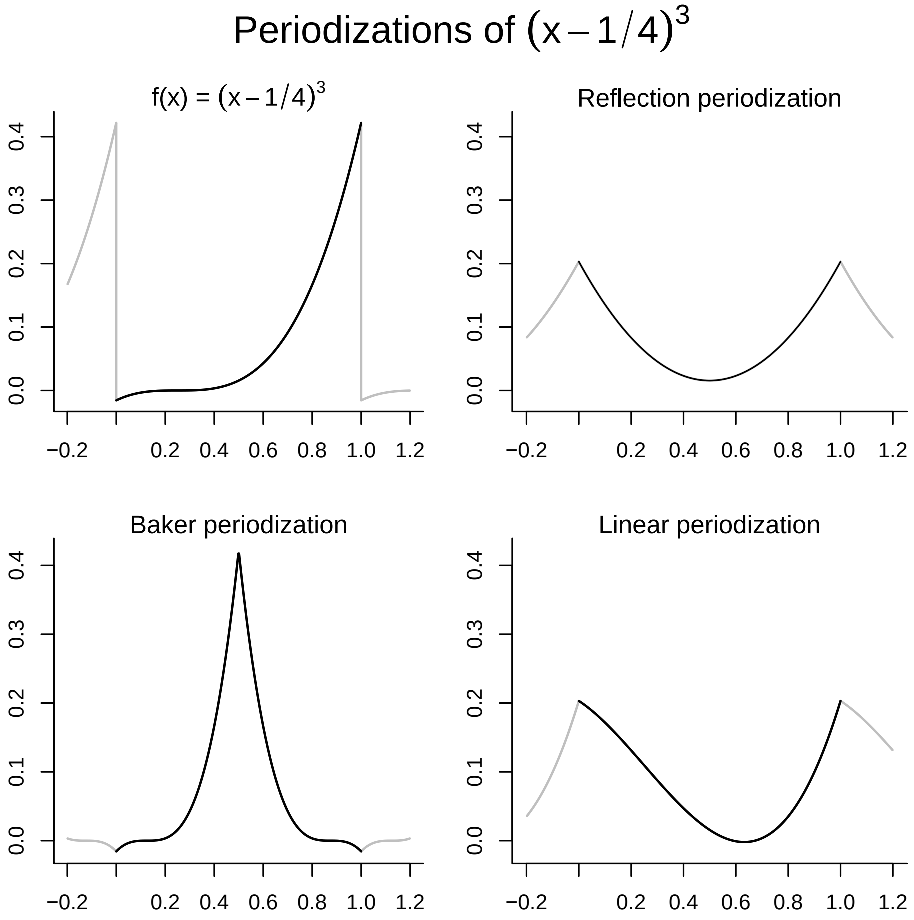

Figure 16.5: The first panel shows $f(x) = (x - 1/4)^3$ on $[0, 1)$ with a portion of its periodic extension. The next three panels show periodizations of $f(x)$ described in the text.

The function $\widetilde{f}(x) = f(\phi(x))\phi'(x)$ has $\widetilde{f}(0) = \widetilde{f}(1)$ if $\phi'(0) = \phi'(1) = 0$. The function ϕ is a cumulative distribution function and ϕ' is the corresponding probability density function.

The function ϕ must satisfy $\phi(0) = 0$, $\phi(1) = 1$, $\phi'(0) = 0$ and $\phi'(1) = 0$. These four constraints can be satisfied by a cubic polynomial $\phi_3(x) = a + bx + cx^2 + dx^3$. Solving

$$\begin{pmatrix}0\\1\\0\\0\end{pmatrix} = \begin{pmatrix}1&0&0&0\\1&1&1&1\\0&1&0&0\\0&1&2&3\end{pmatrix}\begin{pmatrix}a\\b\\c\\d\end{pmatrix}$$

we find that $\phi_3(x) = 3x^2 - 2x^3$, which upon inspection is increasing on $[0, 1]$. We recognize the derivative $\phi_3'(x) = 6x(1-x)$ as the density function of a Beta$(2,2)$ random variable. The resulting periodization method is

$$\widetilde{f}_{\mathrm{Beta}(2,2)}(x) = f(\phi_3(x))\phi_3'(x).$$

The periodization $\widetilde{f}_{\mathrm{Beta}(2,2)}$ does not have a continuous derivative at $x = 1$. Adding the further constraints $\phi''(0) = \phi''(1) = 0$ makes $\widetilde{f}'(0) = \widetilde{f}'(1) = 0$. The quintic polynomial $\phi_5(x) = x^3(10 - 15x + 6x^2)$ satisfies these constraints. It has derivative $\phi_5'(x) = 30x^2(1-x)^2$, the Beta$(3,3)$ density. The periodizing transformation

$$\widetilde{f}_{\mathrm{Beta}(3,3)}(x) = f(\phi_5(x))\phi_5'(x)$$

is the first one we have considered whose periodic extension has a continuous derivative when f does. It is illustrated for $f(x) = (x - 1/4)^3$ in the second panel of Figure 16.6. The Beta(k,k) density $[x(1-x)]^{k-1}(2k)!/(k!)^2$ has $k-1$ vanishing derivatives at 0 and at 1.

More generally, we may obtain a periodization with $k-1$ derivatives vanishing at integer values x by taking $\phi = \phi_{2k-1}$ equal to the cumulative distribution function of the Beta(k,k) distribution. Larger k are smoother but we will soon see a drawback for large k.

Periodization by such transformations as this can be extended to d dimensions without undue computational cost. In d dimensions we may use a ***monotone change of variable*** periodization

$$\widetilde{f}(\boldsymbol{x}) = f(\phi(\boldsymbol{x}))\prod_{j=1}^d \phi'(x_j), \tag{16.24}$$

where $\phi(\boldsymbol{x})$ is applied componentwise. If $\phi'(0) = \phi'(1) = 0$ then $\widetilde{f}(\boldsymbol{x})$ equals 0 on the boundary of $[0,1)^d$ and has a continuous periodic extension. Choosing ϕ with more vanishing derivatives makes $\widetilde{f}(\{\boldsymbol{x}\})$ smoother.

Sloan and Joe (1994) advocate the transformation

$$\phi_{\mathrm{Sidi}}(x) = x - \frac{\sin(2\pi x)}{2\pi}$$

with $\phi_{\mathrm{Sidi}}'(x) = 1 - \cos(2\pi x)$ due to Sidi (1993). This choice of ϕ helps us find a periodization of lattice rules that integrates constant functions without error. To see why that is an issue, suppose that $f(\boldsymbol{x}) = c$, a constant. Then $\widetilde{f}(\boldsymbol{x})$ from (16.24) equals $c \times \prod_{j=1}^d \phi'(x_j)$ is not constant. When our rule integrates constants correctly, and $f_1(\boldsymbol{x}) + f_2(\boldsymbol{x}) = 1$, then we will surely have $\hat\mu_1 + \hat\mu_2 = 1$ where $\hat\mu_j$ is the lattice rule estimate of $\mu_j = \int f_j(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$. When $f(\boldsymbol{x}) = 1$, then $\widetilde{f}$ for Sidi's transformation becomes

$$\prod_{j=1}^d (1 - \cos(2\pi x_j)) = \prod_{j=1}^d \Bigl(1 - \frac{1}{2}\Bigl(e^{\sqrt{-1}x_j} + e^{-\sqrt{-1}x_j}\Bigr)\Bigr)$$

which expands into a sum of 3^d sinusoids $\psi_{\boldsymbol{h}}(\boldsymbol{x})$, all with $\max_{1\leqslant j\leqslant d}|h_j|\leqslant 1$. Unless the dual lattice of (16.1) has a nonzero $\boldsymbol{h}$ with all components in $\{-1,0,1\}$ the lattice rule will correctly integrate $f(\boldsymbol{x})=1$.

For large d, the product $\prod_{j=1}^d\phi'(x_j)$ can cause difficulties. It typically introduces a significant spike into $\widetilde{f}(\boldsymbol{x})$ near $\boldsymbol{x}_c=(1/2,\dots,1/2)$. For example $\widetilde{f}(\boldsymbol{x}_c)=f(\boldsymbol{x}_c)\phi'(1/2)^d$. With $\phi'_{\mathrm{Beta}(2,2)}(1/2)=1.5$, $\phi'_{\mathrm{Beta}(3,3)}(1/2)=1.875$, $\phi'_{\mathrm{Beta}(4,4)}(1/2)=2.1875$, and $\phi'_{\mathrm{Sidi}}(1/2)=2$, the spike can grow quickly with d.

When d is large, then $\widetilde{f}(\boldsymbol{x})$ will have a prominent spike near the center of the cube, unless $f(\phi(\boldsymbol{x}))$ somehow vanishes there. When n is small, the entire sample may miss the spike. A moderately large sample may hit the spike once or a few times, with the result that the estimate $\hat{\mu}$ is dominated by those few function evaluations. A very large sample is required so that the spike region is properly covered.

We cannot solve the spike problem by choosing a function with $\phi'(1/2)\leqslant 1$. The function ϕ' has to have an average value of 1 in order that $\phi(0)=0$ and $\phi(1)=1$. But ϕ' must be close to zero near 0 and 1 to bring about periodicity of $\widetilde{f}$. Therefore ϕ' must be larger than 1 somewhere and the usual choices for ϕ' have a maximum at 1/2.

In §9.1, importance sampling was proposed as a means of handling integrands with spikes in them. Suppose that we apply importance sampling to the integrand in (16.24), sampling from the density $\prod_{j=1}^d\phi'(x_j)$. The result is to replace $\widetilde{f}(\boldsymbol{x})$ by

$$\frac{\widetilde{f}(\phi^{-1}(\boldsymbol{x}))}{\prod_{j=1}^d\phi'(x_j)}=f(\phi(\phi^{-1}(\boldsymbol{x})))\frac{\prod_{j=1}^d\phi'(x_j)}{\prod_{j=1}^d\phi'(x_j)}=f(\boldsymbol{x}). \tag{16.25}$$

This importance sampling undoes the periodization transformation. Put another way: the periodization that caused our problem is itself a kind of importance sampling.

The problem with spikes is not limited to transformations applied componentwise to $\boldsymbol{x}$. Suppose that $\phi(\boldsymbol{x})$ transforms $[0,1]^d$ to $[0,1]^d$ and that we replace $\int f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ by $\int f(\phi(\boldsymbol{x}))\det(J(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x}$ where J is the Jacobian of ϕ. We can make $\widetilde{f}(\boldsymbol{x})=f(\phi(\boldsymbol{x}))\det(J(\boldsymbol{x}))$ periodic by making $\det(J(\boldsymbol{x}))$ equal zero on the boundary of $[0,1)^d$. It is clear that $\det(J(\boldsymbol{x}))$ must average to 1 over $[0,1)^d$. To see this consider the function f with $f(\boldsymbol{x})=1$. Now if $|\det(J(\boldsymbol{x}))|\leqslant\epsilon$ for some $0<\epsilon<1/2$ whenever $\boldsymbol{x}\not\in[\epsilon,1-\epsilon]^d$ then $|\det(J(\boldsymbol{x}))|$ has to be very large somewhere inside the tiny region $(\epsilon,1-\epsilon)^d$.

The method of choice for periodization of high dimensional functions remains the baker transformation. For smaller d, a smooth monotone change of variable transformation, such as Sidi's, may be better. A strong advantage of the baker transformation was discovered by Hickernell (2002). He shows that the baker transformation can produce errors of $O(n^{-2+\epsilon})$ if f can be continuously differentiated up to twice with respect to each component x_j. This holds even though $f(B(\cdot))$ fails to be smooth at points $\boldsymbol{x}$ with any $x_j=1/2$. Lack of

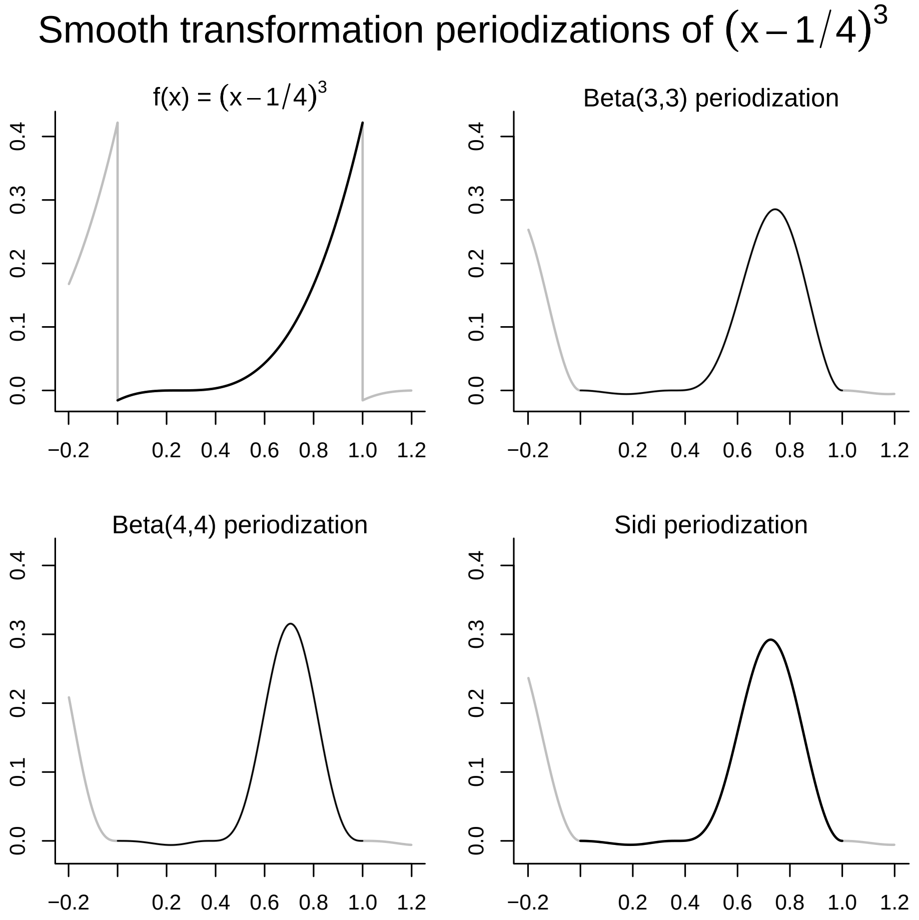

Figure 16.6: The first panel shows $f(x) = (x - 1/4)^3$ on $[0, 1)$ with a portion of its periodic extension. The next three panels show smooth transformation periodizations of $f(x)$ based on two Beta CDFs and a transformation of Sidi.

smoothness along axis parallel directions is a ‘QMC-friendly’ lack of smoothness as discussed by Wang and Sloan (2011).

Table 16.3 adds a column for a Korobov rule with the baker function to the estimates of the wing weight integral from Table 16.2. We see a much more stable estimate for Korobov points using the baker’s transformation as n increases. By that standard, the baker’s transformation appears to have improved the accuracy of the Korobov lattice as predicted by Hickernell (2002).

n	Korobov	K.+baker	Halton
1021	268.0803	268.0743	267.4654
2039	267.9789	268.0739	267.5688
4093	268.0776	268.0750	267.8209
8191	268.0763	268.0753	267.9668
16381	268.0753	268.0752	268.0193

Table 16.3: Sample sizes n and integral estimates for the mean wing weight. The methods are Korobov points, Korobov points with a baker's transformation, and Halton points.

16.7 Lattice parameter search

Searching for good lattice parameters is a job for specialists. Fortunately, they publish tables with values that they find work well. See for example Hua and Wang (1981), Sloan and Joe (1994), and L'Ecuyer and Lemieux (2000). As computers get more powerful, sample sizes grow, and static lists of tables become obsolete. As a result, we can expect to need new searches as long as computers keep improving.

Before doing the search, one shows theoretically that there are good lattices to be found. Somewhat disturbingly, this step proceeds by showing that the average quality for a randomly chosen lattice is acceptable. For rank-1 rules, the average might be taken over all vectors $(z_1, z_2, \dots, z_d)$ subject only to each z_j being an integer between 1 and $n-1$ inclusive with $\mathrm{GCD}(z_j, n) = 1$. Sloan and Joe (1994, Chapter 4.4) note that it is enough to search with $z_j \leqslant \lfloor n/2 \rfloor$. Then we can conclude that there must be at least one such good parameter vector $\boldsymbol{z}$. The reason that this argument is disturbing is that it shows existence of good parameters but does not point out any single specific good parameter vector. If we really did pick the vector at random, then we might get one that is much worse than average and then use it on every quadrature problem. Once the search has begun on the computer, we do get numerical values of the figure of merit in use and we can control the probability of a bad result. If we choose $\boldsymbol{z}$ uniformly at random from the specified set, then there is at most 0.5 probability that our criterion is worse than average. If we choose 10 times at random, then there is less than $2^{-10} < 0.001$ probability that the best of those 10 is worse than the average. There is also at most 10^{-10} probability that the best one exceeds 10 times the average.

Some of the searches are done with criteria other than P_α. Sometimes it is possible to compute the ratio of the attained criterion to a bound on the best possible value for that criterion. L'Ecuyer and Munger (2016) include such a relative quality option for a spectral criterion that describes the spacings between lattice planes.

What makes the search much more feasible is that a greedy component-by-component (CBC) strategy is now known to find a good lattice. CBC search was proposed by Korobov (1959). It was long forgotten and then reinvented by Sloan and Reztsov (2002). We could reasonably be concerned that a greedy search, choosing one z_j at a time, would be suboptimal. Kuo (2003) shows that CBC search produces lattices that attain the same convergence rate as optimizing all of $z_2, \dots, z_d$ jointly. Given n, we pick $(z_1, \dots, z_k)$ to get a good k dimensional rule, working up from $k = 1$ to d. When searching for z_k we retain the values $z_1, \dots, z_{k-1}$ from the earlier searches. The starting point is easy. Because any z_1 relatively prime to n will give the same 1 dimensional lattice we may take $z_1 = 1$. Nuyens and Cools (2006) brought a significant speedup to CBC searches by employing fast Fourier transformations.

16.8 Embedded, extensible and shifted lattices

The lattice rules presented so far are not extensible. If n proves to be too small, then we have to start over with a larger number $n' > n$ of points and may even have to repeat a parameter search for lattices of size n'. While rank-1 lattice rules are an improvement over the Kronecker rules of §15.14, they have given up the extensibility of Kronecker rules in return for having especially good performance at certain special values of n such as prime numbers or powers of 2. Here we consider ways to produce rank-1 lattice rules with more than one especially good sample size.

Embedded lattice rules of rank-1 in $[0,1]^d$ are constructed to be extensible through a finite sequence of sample sizes n for a finite list of dimensions d. Most commonly

$$n = b^m, \quad m_1 \leqslant m \leqslant m_2 \quad \text{and } 1 \leqslant d \leqslant d_{\max}.$$

Here $b \geqslant 2$ is an integer, and $b = 2$ is the usual choice. We will use the term 'embedded' to mean that the number of levels of n or d is finite, but greater than 1. By contrast, 'extensible' means that there are an infinite number of levels. The component-by-component constructions in §16.7 produce rules with a fixed n that are extensible in d. They have been generalized to produce rules that are embedded with respect to n, but are extensible in d. Some lattice rules are extensible in both n and d.

It is common for embedded and extensible lattice rules to be constructed using a shift modulo one. For $\Delta \in [0,1)^d$, the ***shifted lattice rule*** has

$$\boldsymbol{x}_i = \Bigl\{\frac{(i-1)\boldsymbol{z}}{n} + \Delta\Bigr\}, \quad i = 1, \dots, n$$

for a vector $\boldsymbol{z} \in \mathbb{Z}^d$. We will consider random Δ in §17.3. The integration error of a lattice from (16.12) becomes

$$\sum_{\boldsymbol{h} \in L_*^\perp} \hat{f}(\boldsymbol{h}) e^{2\pi\sqrt{-1}\Delta^\mathsf{T}\boldsymbol{h}}. \tag{16.26}$$

See Exercise 16.7.

Cools et al. (2006) have a strategy for embedded lattice rules of size $n = 2^m$ for $m_1 \leqslant m \leqslant m_2$. Let $\mathrm{wce}(n, d, \boldsymbol{z})$ be the worst case error when using the vector $\boldsymbol{z} \in \mathbb{Z}^d$ in a rank-1 lattice rule with n points in dimension d. This quantity could be the $P_2(\boldsymbol{z}; n)$ from §16.4 but those authors include more general criteria including some designed for the weighted spaces discussed in §16.9. The search is for a good vector $\boldsymbol{z} \in \mathbb{Z}^d$ among those with all $\gcd(z_j, b) = 1$. For $m = m_1, \dots, m_2$, let $\boldsymbol{z}^{(m)}$ minimize $\mathrm{wce}(b^m, d, \boldsymbol{z})$. Then let

$$\mathrm{wcerel}(\boldsymbol{z}) = \max_{m_1 \leqslant m \leqslant m_2} \frac{\mathrm{wce}(b^m, d, \boldsymbol{z})}{\mathrm{wce}(b^m, d, \boldsymbol{z}^{(m)})}.$$

Given $z_1, z_2, \dots, z_{d-1}$, they choose z_d to minimize the worst case relative error $\mathrm{wcerel}(\boldsymbol{z})$ above. They report that those worst case relative errors are typically smaller than 2. They actually tune their performance measures to account for a random Δ, so the story is a bit more complicated than the above account.

The whole search can be done in time $O(nd(\log(n))^2)$ time, for prime b where $n = b^{m_2}$. For fixed d, the cost to compute $f(\boldsymbol{x}_1), \cdots, f(\boldsymbol{x}_{b^m})$ will ordinarily be $O(b^m)$ for a value of m between m_1 and m_2, so for very large sample sizes the search cost will not be negligible. The cost to compute each $f(\boldsymbol{x}_i)$ could depend on d in many ways. If it is sublinear in d, then the construction cost is asymptotically dominant, while if it it superlinear in d, then the construction cost can be negligible.

An ***extensible shifted lattice rule*** with shift $\Delta \in [0, 1)^d$ is an infinite sequence $\boldsymbol{x}_i = \{\phi_b(i-1)\boldsymbol{z} + \Delta\}$ for $i \geqslant 1$, where $\phi_b(\cdot)$ is the radical inverse function that we used to generate the van der Corput sequence in §15.5. That is, we use the points

$$\{\phi_b(i)\boldsymbol{z} + \Delta\}, \quad i \geqslant 0.$$

As before $\boldsymbol{z} \in \mathbb{Z}^d$. We choose $\boldsymbol{z}$ in order to get a good lattice rule on $n = b^m$ points. Now consider indices $i = \ell b^m, \ell b^m + 1, \ell b^m + 2, \dots, (\ell + 1)b^m - 1$ for an integer $\ell \geqslant 0$. Over this range $\phi_b(i)$ takes the values

$$\phi_b(\ell)b^{-m-1} + \phi_b(0), \phi_b(\ell)b^{-m-1} + \phi_b(1), \dots, \phi_b(\ell)b^{-m-1} + \phi_b(b^m - 1).$$

These are a reordering of

$$\phi_b(\ell)b^{-m-1} + j/b^m, \quad 0 \leqslant j < b^m.$$

As a result, the ℓ'th block of consecutive point is a shifted lattice rule with shift $\Delta + \phi_b(\ell)b^{-m-1}$. We get an infinite sequence of shifted lattice rules, each of length b^m. They do not repeat. If we choose $\Delta = 0$, then the first block is a usual rank-1 lattice rule, while all subsequent blocks are shifted lattice rules.

Extensible lattice rules were proposed by Maize (1981) and rediscovered by Hickernell and Hong (1997) and further studied by Hickernell et al. (2000). Hickernell and Niederreiter (2003) prove, using an averaging argument, that good extensible rank-1 lattices exist. Table 16.4 gives some example rules. They are for points $\{\phi_2(i)\boldsymbol{z} + \Delta\}$ for a Korobov vector $\boldsymbol{z} = (1, a, \dots, a^{s-1})$ in dimension $s \leqslant d$. They are designed for $i = 0, \dots, 2^m$ for $m_0 \leqslant m \leqslant m_1$ and

Criterion	m_0	m_1	d	a
Weighted P_2	0	17	32	17797
Weighted P_2	13	20	32	407641
Spectral	0	17	25	1267
Spectral	15	24	32	4450341

Table 16.4: Selected extensible Korobov rules from Table 4.1 of Hickernell et al. (2000). They use $\boldsymbol{z} = (1, a, \dots, a^{s-1})$ to integrate over $[0,1)^s$ for $s \leqslant d$. The intended sample sizes are $n = b^m$ for $m_0 \leqslant m \leqslant m_1$.

thereafter for $i = 0, \dots, \ell 2^{m_1}$, for $\ell \geqslant 1$. The criterion 'weighted P_2' refers to a criterion designed for functions in a weighted space model that downweights the importance of higher order interactions in f. The article describes them as using $\alpha = 1$ but ordinarily $\alpha > 1$ is required for lattices, and they do use the Bernoulli polynomial of degree two. The ones labeled 'Spectral' use a criterion similar to the ones used to design random number generators. The selected rules in Table 16.4 cover two ranges of sample sizes.

16.9 Weighted spaces

Lattices can be custom designed to integrate functions in the weighted spaces of §7.7. The weights are incorporated into a figure of merit and then it is possible to do a custom search for a lattice rule just before evaluating an integrand, though that does raise the cost.

To describe these methods, for $u \subseteq \{1, 2, \dots, d\}$, let $\boldsymbol{x}_u$ be the components of $\boldsymbol{x}$ for $j \in u$. L'Ecuyer and Munger (2016) consider criteria of the form

$$\sum_{\varnothing \neq u \subseteq \{1,2,\dots,d\}} \gamma_{u,q} \times \mathcal{D}_u(\boldsymbol{x}_{1,u}, \dots, \boldsymbol{x}_{n,u})^q$$

where $q \geqslant 1$ and $\gamma_{u,q}$ are real numbers and $\mathcal{D}_u$ is a badness measure for points in $[0,1]^{|u|}$. It could be the worst case error from §16.4, via $\mathcal{D}_u^2 = P_2(\boldsymbol{z}; n)$. They include several other performance measures. They write their weights γ_u^q but then remark that their methods allow negative weights, so writing $\gamma_{u,q}$ makes it clear that the weights need not be the q'th power of a real number. Their optimization takes account of a fixed set of sample sizes, but not an infinite set. That is, their rules are embedded but not extensible in n.

Section 7 of Kuo and Nuyens (2016) describes software for constructing lattices in weighted spaces. They also consider polynomial lattice rules mentioned in the end notes of Chapter 15.

Chapter end notes

Lattice rules were proposed by Korobov (1959), with early contributions by Hua and Wang (1960) and Hlawka (1962). They were earlier called the ***number theoretic method*** because of the use of number theory in the searches for good parameter values. More information on lattice rules may be found in the monographs by Sloan and Joe (1994), Hua and Wang (1981) and Niederreiter (1992b) as well as the article by Lyness (1989). Fang and Wang (1994) give applications to statistics. Most of the literature on lattice rules emphasizes periodic integrands. Dick et al. (2014) are an exception. In place of functions $\psi_{\boldsymbol{h}}(\boldsymbol{x}) = \exp(2\pi\sqrt{-1}\boldsymbol{h}^\mathsf{T}\boldsymbol{x})$, they use $\prod_{j\in u}\sqrt{2}\cos(\pi k_j x_j)$ for $u \subseteq \{1,2,\dots,d\}$ and integers $k_j \geqslant 1$.

The literature on lattice rules refers to both Korobov spaces and Sobolev spaces. Korobov spaces have periodic integrands and integrands in Sobolev spaces are not necessarily periodic.

Fibonacci lattices attain optimal discrepancy and L_2-discrepancy among lattice rules in $[0,1)^2$. Breneis and Hinrichs (2019) describe several optimality results for Fibonacci lattices. For $d \geqslant 3$, there is no best family of lattice rules comparable to Fibonacci lattices for $d = 2$.

Weighted spaces were introduced by Hickernell (1996b) to improve the quality of lattice rules in their lower dimensional projections. Wang and Sloan (2006) describe a sense in which lattice rule constructions that are not designed for weighted spaces are more sensitive to equidistribution of higher dimensional projections than they are to lower dimensional projections.

Higher rank lattice rules

An estimate from a rank-2 lattice rule takes the form

$$\hat{\mu} = \frac{1}{n}\sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2} f\biggl(\biggl\{\frac{i_1-1}{n_1}\boldsymbol{z}_1 + \frac{i_2-1}{n_2}\boldsymbol{z}_2\biggr\}\biggr) \tag{16.27}$$

where $n = n_1 n_2$ and $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{Z}^d$ are carefully chosen vectors of integers. More generally, for $1 \leqslant r \leqslant d$, a rank-$r$ rule takes the form

$$\hat{\mu} = \frac{1}{n}\sum_{i_1=1}^{n_1}\cdots\sum_{i_r=0}^{n_r} f\Biggl(\Biggl\{\sum_{j=1}^{r}\frac{i_j-1}{n_j}\boldsymbol{z}_r\Biggr\}\Biggr) \tag{16.28}$$

where $n = \prod_{j=1}^r n_j$ and $\boldsymbol{z}_1, \dots, \boldsymbol{z}_r \in \mathbb{Z}^d$.

The rank-1 lattice rules are formed as n consecutive integer multiples of a single vector $\boldsymbol{z}$. Geometrically they are formed by taking equispaced points along a line through the origin and then putting them into $[0,1)^d$ by a wraparound operation corresponding to taking the points modulo 1. Rank-2 lattices are obtained by taking a rectangular grid of points in a plane through the origin and reducing them modulo 1 to lie within $[0,1)^d$. Rank-r rules exist for any integer r between 1 and d inclusive. Some of them are presented in Sloan and Joe (1994). Higher rank rules also have dual lattices and they satisfy Proposition 16.1, though the proof is more subtle in the general case.

Rules of rank 2 and higher can be shown to achieve the higher order accuracy that rank-1 rules obtain for smooth periodic functions. Joe and Disney (1993) describe how the average rank $r+1$-rule is better than the average rank-r rule for $1 \leqslant r < d$. There is numerical evidence that well chosen rank-2 rules can be somewhat better than rank-1 rules (see Sloan and Joe (1994)), but it in the examples they do not appear to be much better. So far, higher rank rules have not displaced rank-1 rules in practice. One disadvantage of higher order rules is that they require a search for good choices of $(\boldsymbol{z}_j, n_j)$ for $j = 1, \dots, r$. If $r > 1$, then the search is more challenging.

Fourier convergence

Many convergence results for multidimensional Fourier series are in (Grafakos, 2004, Chapter 3) and yet more are in Golubov (1984) who cites 474 references on the topic. A sufficient condition for f to have an absolutely summable Fourier expansion is that $f(\{\boldsymbol{x}\})$ have $\alpha > d/2$ derivatives. Conversely, there exist functions with exactly $d/2$ derivatives and divergent Fourier coefficient sums. The condition allows for non-integer α as next, but the more important point is that for larger d ever more smoothness is required of f by this sufficient condition. Differentiability of non-integer order $\alpha > 0$ then means that every partial derivative of f of order $\lfloor \alpha \rfloor$ is Hölder continuous of order $\beta = \alpha - \lfloor \alpha \rfloor$. The function g is Hölder continuous of order β if $|g(\boldsymbol{x}) - g(\boldsymbol{x} + \boldsymbol{\delta})| = O(\|\boldsymbol{\delta}\|^{\beta})$ as $\boldsymbol{\delta} \to 0$.

Lattices versus nets

Prior to the 1990s, lattice rules could be designed to exploit increased smoothness of the integrand, while digital nets could not. Digital nets were known to be part of extensible in n sequences while lattices were not extensible. Finally, lattices required challenging parameter searches while digital nets were almost automatic: Faure sequences require no search, and while Sobol' sequences require a choice of direction numbers, there are just a small number of commonly used choices.

Now the features of each family of methods have found parallels in the other. The polynomial lattice rules of Niederreiter (1992a) provide a mechanism to search among digital net constructions. Hickernell et al. (2000) brought extensibility to lattice rules. The advent of higher order nets, also called interlaced nets by Dick (2008) yielded digital net constructions that could exploit increased smoothness. The expansion into Walsh functions for digital nets in §15.13 is a natural parallel to the Fourier expansions used for lattices.

It is difficult to choose between lattices or digital nets, at least based on accuracy. The difference between QMC and MC is much more important than the choice of which sort of MC to use. Even for one specific domain, the valuation of Asian options in finance, Lemieux and L'Ecuyer (1998) found lattices working best in some examples and nets working best in others.

Exercises

16.1. Prove that the mean of $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in [0,1)^d$ from a rank-1 lattice rule has all d components equal to $1/2-1/(2n)$. For the sample sizes n in Table 16.3, find the mean of $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in [0,1]^{10}$ when $\boldsymbol{x}_i$ are taken from the Halton sequence. Subtract each component of the mean from $1/2$ and multiply the absolute value of the difference by n. Compare the result to corresponding results for lattice points.

16.2. Prove that the set $L^\perp$ is a lattice.

16.3. Let f be defined on $[0,1)^d$ for $d > 1$. Show that the function $(f(\boldsymbol{x}) + f(\mathbf{1} - \boldsymbol{x}))/2$ with $\mathbf{1} - \boldsymbol{x}$ taken componentwise is not necessarily periodic.

16.4. Prove that the baker periodization satisfies $\int \widetilde{f}_{\mathrm{baker}}(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} = \int f(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}$.

16.5. Sidi's change of variable periodization results in a lattice rule that correctly integrates constant functions.

a) Show with a small example that the Beta(2,2) periodization does not always correctly integrate constant functions.

b) Determine whether lattice rules incorporating the baker transformation always correctly integrate constant functions. Do the same for the reflection periodization.

c) Now consider the linear functions $x_j - 1/2$ for $j = 1, \dots, d$. These of course integrate to 0. Which of the change of variable transformations we considered will lead to correct integration of these linear functions.

16.6. Prove that equality holds in (16.15) for f given by (16.16).

16.7. Prove equation (16.26). This expression involves complex numbers even though the integration error must be real for $f(\boldsymbol{x}) \in \mathbb{R}$. Show that the expression is indeed real, in some other way than simply observing that $\hat{\mu} - \mu$ must be real when $\hat{\mu}, \mu \in \mathbb{R}$.

17

Randomized quasi-Monte Carlo

From Chapters 15 and 16 we see that quasi-Monte Carlo (QMC) methods can vastly outperform Monte Carlo (MC). Under the right regularity on f, QMC can attain an error of $O(n^{-1+\epsilon})$ or even $O(n^{-\alpha+\epsilon})$ for some $\alpha \geqslant 1$ and all $\epsilon > 0$, compared to a root mean squared error (RMSE) of $O(n^{-1/2})$ for MC.

A major difficulty with QMC is that we cannot estimate the size of the error from the QMC sample values $f(\boldsymbol{x}_1), \dots, f(\boldsymbol{x}_n)$. The theory provides estimates for $|\hat{\mu} - \mu|$, but they depend on virtually unknowable quantities, may apply to worst case functions quite different from f, and the estimates are usually asymptotic in n.

The situation for plain MC, while not perfect, is much more satisfactory. The RMSE is exactly $\sigma/\sqrt{n}$ for the n we used, where σ^2 is the variance of the f we studied. While σ^2 is unknown, MC provides a useful unbiased estimate s^2 of it, and the central limit theorem gives us asymptotic confidence statements. Those account for the estimation error in both $\hat{\mu}$ and s. When we want 99% coverage, we get $99\% + O(1/n)$ coverage. Our uncertainty quantification is in this sense even more precise than our estimate.

In this chapter we consider randomized QMC (RQMC) methods to get the accuracy of QMC with the error estimation advantage of MC. In an RQMC method, the points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are individually $\mathbf{U}[0,1]^d$, but collectively of low discrepancy. It will follow that $\hat{\mu}$ is then an unbiased estimate of μ with at least QMC accuracy. Then independent replications of the QMC rule provide an MC basis for error estimation.

We will see some circumstances where RQMC ends up being even more accurate than plain QMC. Randomization also helps to protect against some worst case outcomes, or at least to make their probabilities negligibly small. Randomization even helps to make the sample points avoid singularities, whether their locations are known or unknown and RQMC can still be better than MC even if the integrand does not have bounded variation.

As we noted in Chapter 15, the points $\boldsymbol{x}_i$ in RQMC are variously defined as elements of $[0,1]^d$, $(0,1)^d$ or $[0,1)^d$, even though $\mathbf{U}[0,1]^d$, $\mathbf{U}(0,1)^d$ and $\mathbf{U}[0,1)^d$ are all the same distribution. The cube $[0,1)^d$ is convenient when f is periodic or when $\boldsymbol{x}_i$ have to be placed into congruent strata, while it is better to define f on $[0,1]^d$ when we need to consider its total variation or Riemann integrability, and $(0,1)^d$ is convenient for some unbounded integrands.

17.1 RQMC definitions and basic properties

Random variables $\boldsymbol{x}_i \in [0,1]^d$ for $i \geqslant 1$ comprise a ***randomized quasi-Monte Carlo*** rule if there exist $B < \infty$ and $N > 0$ with

$$\mathbb{P}\bigl(D_n^*(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n) < B(\log n)^d/n \bigr) = 1, \quad \text{for all } n \geqslant N, \quad \text{and,} \tag{17.1}$$

$$\boldsymbol{x}_i \sim \mathbf{U}[0,1]^d, \quad \text{for all } i \geqslant 1. \tag{17.2}$$

This definition applies to an infinite sequence. We can also define a triangular array version of RQMC. There $\boldsymbol{x}_{n_j i} \sim \mathbf{U}[0,1]^d$ for $j \geqslant 1$ and $i = 1,\dots,n_j$ and $\mathbb{P}\bigl(D_{n_n}^*(\boldsymbol{x}_{n_j 1},\dots,\boldsymbol{x}_{n_j n_j}) < B(\log n_j)^d/n_j \bigr) = 1$ for $j \geqslant 1$. The sample sizes satisfy $n_j \to \infty$ as $j \to \infty$, with $n_{j+1} > n_j$. For example, they may be a sequence of prime numbers, or powers of 2.

Constructions of RQMC points begin with QMC points $\boldsymbol{a}_1,\dots,\boldsymbol{a}_n \in [0,1]^d$. Then we apply randomizations, generating $\boldsymbol{x}_i \sim \mathbf{U}[0,1]^d$ from $\boldsymbol{a}_i$, while preserving in $\boldsymbol{x}_i$ some of the QMC structure from $\boldsymbol{a}_i$. Before describing specific constructions, we look at their basic properties, as well as how to use randomization to estimate error.

Given n points $\boldsymbol{x}_1,\dots,\boldsymbol{x}_n \in [0,1]^d$ of an RQMC rule, the estimate of $\mu = \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ is the usual average

$$\hat\mu = \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i).$$

The RQMC estimate is unbiased, because

$$\begin{aligned}\mathbb{E}(\hat\mu) &= \frac{1}{n}\sum_{i=1}^n \mathbb{E}(f(\boldsymbol{x}_i)), \quad \text{and}\\ \mathbb{E}(f(\boldsymbol{x}_i)) &= \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \mu.\end{aligned}$$

If f has bounded variation in the sense of Hardy and Krause, then

$$\begin{aligned}\mathrm{Var}(\hat\mu) = \mathbb{E}((\hat\mu-\mu)^2) &\leqslant \mathbb{E}\bigl((D_n^*(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)V_{\mathrm{HK}}(f))^2\bigr)\\ &< B^2 V_{\mathrm{HK}}(f)^2\frac{\log(n)^{2d}}{n^2}\end{aligned}$$

for large enough n, so RQMC is asymptotically better than Monte Carlo.

RQMC provides an unbiased estimate of μ for which the QMC error bounds apply. RQMC estimates have an RMSE that is $O(n^{-1+\epsilon})$ for any $\epsilon > 0$. The process that turns $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ into random points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ can be repeated independently $R \geqslant 2$ times, giving $\hat\mu_1, \dots, \hat\mu_R$. Then we may form the pooled estimate,

$$\hat\mu_{\mathrm{pool}} = \frac{1}{R}\sum_{r=1}^{R} \hat\mu_r$$

and its associated variance estimate

$$\widehat{\mathrm{Var}}(\hat\mu_{\mathrm{pool}}) = \frac{1}{R(R-1)}\sum_{r=1}^{R} (\hat\mu_r - \hat\mu_{\mathrm{pool}})^2.$$

Because $\hat\mu_1, \dots, \hat\mu_R$ are independent and identically distributed, we find that $\mathbb{E}(\hat\mu_{\mathrm{pool}}) = \mu$ and $\mathbb{E}(\widehat{\mathrm{Var}}(\hat\mu_{\mathrm{pool}})) = \mathrm{Var}(\hat\mu_{\mathrm{pool}})$.

A replicated RQMC estimate requires nR function evaluations. When f is of bounded variation, the error in $\hat\mu_{\mathrm{pool}}$ is $O(n^{-1+\epsilon}R^{-1/2})$. Given an upper bound on nR, the most accurate estimate of μ is obtained by taking n large and R small. Taking R small will result in a poor variance estimate. The estimate of $\mathrm{Var}(\hat\mu_{\mathrm{pool}})$ is based on a sample of R independent replicates. The relative error $\widehat{\mathrm{Var}}(\hat\mu_{\mathrm{pool}})/\mathrm{Var}(\hat\mu_{\mathrm{pool}}) - 1$ decreases at the rate $O(R^{-1/2})$, for any fixed n, when $\mathbb{E}(\hat\mu_r^4) < \infty$.

Confidence intervals are better than variance estimates for quantifying the uncertainty in $\hat\mu_{\mathrm{pool}}$. If R is large, then an asymptotic 99% confidence interval for μ is $\hat\mu_{\mathrm{pool}} \pm 2.58\widehat{\mathrm{Var}}(\hat\mu_{\mathrm{pool}})$. For smaller R it is better to use $\hat\mu_{\mathrm{pool}} \pm t^{0.995}_{(R-1)}\widehat{\mathrm{Var}}(\hat\mu_{\mathrm{pool}})$ where $t^{\alpha}_{(k)}$ is the α-quantile of the t distribution on k degrees of freedom. The attained coverage of an asymptotic 99% confidence interval is typically $0.99 + O(1/R)$, compared to the $O(1/\sqrt{R})$ error in the variance estimate. The accuracy of these approximate confidence intervals depends strongly on the third central moment of $\hat\mu_r$. When we suspect that $\hat\mu_r$ has a very skewed distribution, then more replicates are needed. It is also possible to replace the standard CLT-based confidence intervals by those based on the bootstrap t method described in the chapter end notes. The coverage error in bootstrap t intervals is typically smaller than other nonparametric confidence interval methods.

The choice of n and R thus depends on the relative importance of the accuracy of $\hat\mu_{\mathrm{pool}}$ and the accuracy of our confidence interval. Even when accuracy of $\hat\mu_{\mathrm{pool}}$ takes precedence, it is reasonable to do some replicates. It would be an odd use case indeed, if we needed utmost accuracy in μ, but were completely uninterested in verifying what accuracy we had achieved.

When we want to estimate $\hat\mu_{\mathrm{pool}}$ well and can accept a rough estimate of its error, then we could take $R = 5$ or 10. Like any rule of thumb this guideline could give poor results in extreme cases. For example, when $\hat\mu_{\mathrm{pool}}$ has a very long-tailed distribution, we might get poor coverage. There is more discussion of uncertainty quantification for RQMC in §17.4, which has a worked example.

Sometimes we may want a very good estimate of $\mathrm{Var}(\hat\mu_{\mathrm{pool}})$ in its own right. For example, when we need to decide which of two RQMC methods to adopt for future problems, it would be worthwhile to carefully investigate their variances on a collection of similar test problems. Then we might want R as large as 300 or even 1000 during the tests, though smaller R would be used later on with the selected method.

Sometimes a central limit theorem holds for each $\hat\mu_r$ as $n \to \infty$. Then we may find that the individual $\hat\mu_r$ values are approximately normally distributed. In that case, a smaller value of R is likely to be large enough to give a good confidence interval.

17.2 Effective dimension for RQMC

The ANOVA decomposition (Appendix §A) of a square integrable function f on $(0,1)^d$ is

$$f(\boldsymbol{x}) = \sum_{u\subseteq 1:d} f_u(\boldsymbol{x}) \tag{17.3}$$

where $f_u(\boldsymbol{x})$ only depends on $\boldsymbol{x}$ through $\boldsymbol{x}_u$. This f_u also satisfies $\int_0^1 f_u(\boldsymbol{x})\,\mathrm{d}x_j = 0$ whenever $j \in u$. Here $f_\varnothing(\boldsymbol{x})$ is a constant function always equal to μ. For $\boldsymbol{x} \sim \mathbf{U}(0,1)^d$, $\mathrm{Var}(f(\boldsymbol{x})) = \sum_u \sigma_u^2$ where $\sigma_u^2 = \mathrm{Var}(f_u(\boldsymbol{x}))$. Under RQMC we can decompose the variance of $\hat\mu$ into components too.

Theorem 17.1. *Let f be square integrable with ANOVA decomposition* (17.3). *If $\boldsymbol{x}_1,\dots,\boldsymbol{x}_n$ are an RQMC point set, then*

$$\mathrm{Var}\biggl(\frac1n\sum_{i=1}^n f(\boldsymbol{x}_i)\biggr) = \sum_{u\subseteq 1:d} \mathrm{Var}\biggl(\frac1n\sum_{i=1}^n f_u(\boldsymbol{x}_i)\biggr). \tag{17.4}$$

Proof. Because

$$\mathrm{Var}\biggl(\frac1n\sum_{i=1}^n f(\boldsymbol{x}_i)\biggr) = \sum_{u\subseteq 1:d}\sum_{v\subseteq 1:d} \mathrm{Cov}\biggl(\frac1n\sum_{i=1}^n f_u(\boldsymbol{x}_i), \frac1n\sum_{i=1}^n f_v(\boldsymbol{x}_i)\biggr),$$

it is enough to show that

$$\mathrm{Cov}\biggl(\sum_{i=1}^n f_u(\boldsymbol{x}_i), \sum_{i=1}^n f_v(\boldsymbol{x}_i)\biggr) = 0$$

for any two distinct subsets u and v of $1{:}d$. This is automatically true if either u or v is $\varnothing$ because $f_\varnothing$ is constant. Without loss of generality, let $j \in u$ with $j \notin v$ with $v \ne \varnothing$. Then $\boldsymbol{x}_i \sim \mathbf{U}(0,1)^d$ implies that $\mathbb{E}(f_u(\boldsymbol{x}_i)) = \mathbb{E}(f_v(\boldsymbol{x}_i)) = 0$. Let $\boldsymbol{x}_{i',-j}$ be composed of x_{ik} for all $k \ne j$. The covariance above is then

$$\sum_{i=1}^n\sum_{i'=1}^n \mathbb{E}(f_u(\boldsymbol{x}_i)f_v(\boldsymbol{x}_{i'})) = \sum_{i=1}^n\sum_{i'=1}^n \mathbb{E}\bigl(\mathbb{E}(f_u(\boldsymbol{x}_i)f_v(\boldsymbol{x}_{i'}) \mid \boldsymbol{x}_{i',-j})\bigr)$$

$$= \sum_{i=1}^{n} \sum_{i'=1}^{n} \mathbb{E}\bigl(f_u(\boldsymbol{x}_i)\mathbb{E}(f_v(\boldsymbol{x}_{i'}) \,|\, \boldsymbol{x}_{i',-j})\bigr) = 0$$

because f_v integrates to zero over x_j. □

By Theorem 17.1, we can write the RQMC variance as

$$\mathrm{Var}\Bigl(\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i)\Bigr) = \frac{1}{n}\sum_{u\neq\varnothing} \Gamma_u \sigma_u^2 \tag{17.5}$$

for ***gain coefficients*** $\Gamma_u \geqslant 0$ that quantify how much better RQMC is than MC for the given integrand f and variables $\boldsymbol{x}_u$. If all $\Gamma_u = 1$, then RQMC has exactly the same variance as MC. A common feature in RQMC is that $\Gamma_u \ll 1$ for subsets u with small cardinality $|u|$. It is possible to have $\Gamma_u > 1$ for some sets, especially those with large cardinality. When f is dominated by effects f_u with small $|u|$, then RQMC can bring a great improvement. There are several ways to measure the extent to which an integrand f is dominated by small subsets u.

Definition 17.1. The function $f : (0,1)^d \to \mathbb{R}$ has effective dimension $s \geqslant 1$ in the ***superposition sense*** at level 0.99 if s is the smallest integer with

$$\sum_{u:|u|\leqslant s} \sigma_u^2 \geqslant 0.99\sigma^2.$$

Another notion of effective dimension has f depending primarily on the first s input variables. We let $\lceil u \rceil = \max\{1 \leqslant j \leqslant d \mid j \in u\}$ with $\lceil\varnothing\rceil = 0$.

Definition 17.2. The function $f : (0,1)^d \to \mathbb{R}$ has effective dimension $s \geqslant 1$ in the ***truncation sense*** at level 0.99 if s is the smallest integer with

$$\sum_{u:\lceil u \rceil\leqslant s} \sigma_u^2 \geqslant 0.99\sigma^2.$$

The threshold 0.99 in effective dimension is arbitrary. It is motivated by the idea that if we could remove about 99% of the variance through methods that are very good for some u, then we might be able to speed up estimation by a factor of about 100. It can be difficult to estimate the effective dimension in specific examples. The ***mean dimensions*** of f in the superposition and truncation senses are

$$\nu_s(f) = \frac{1}{\sigma^2}\sum_{u\subseteq 1:d} |u|\sigma_u^2, \quad \text{and} \tag{17.6}$$

$$\nu_t(f) = \frac{1}{\sigma^2}\sum_{u\subseteq 1:d} \lceil u \rceil\sigma_u^2 \tag{17.7}$$

respectively. These are well defined unless $\sigma^2 = 0$. In that case we could take $\nu_s(f) = \nu_t(f) = 0$, or we may simply ignore this exception, as we seldom need to integrate a function with $\sigma^2 = 0$.

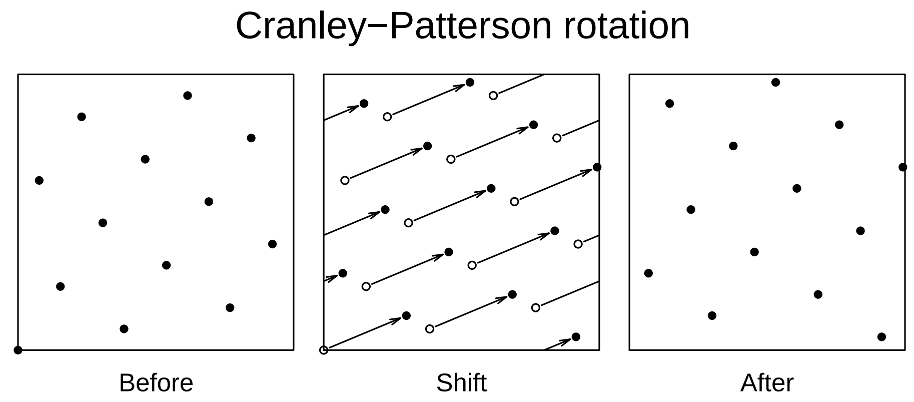

Figure 17.1: The left panel has 13 points in the unit square. The center panel shows them shifted right by 0.3 and up by 0.125 with wraparound. The right panel shows the resulting points.

The value ν_s is comparatively easy to estimate by Sobol' indices. We can use the identity

$$\sigma^2\nu_s(f) = \frac{1}{2}\sum_{j=1}^{d}\int_0^1\int_{[0,1]^d}(f(\boldsymbol{x}) - f(\boldsymbol{x}_{-j}{:}z_j)^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}z_j \tag{17.8}$$

See Appendix §A. If ν_s is close to one, then it means that f is well approximated by an additive function.

17.3 Cranley-Patterson rotation and lattices

A simple random shift modulo 1 is often used to randomize lattice rules. Let $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n \in [0,1]^d$. A ***Cranley-Patterson rotation*** of these points takes the form

$$\boldsymbol{x}_i = \boldsymbol{a}_i + \boldsymbol{U} \ \mathsf{mod}\ 1$$

interpreted componentwise, where $\boldsymbol{U} \sim \mathbf{U}(0,1)^d$. The method is named for the authors of the paper Cranley and Patterson (1976) where the idea was proposed for lattice rules. Figure 17.1 illustrates Cranley-Patterson rotation for $d = 2$.

The Cranley-Patterson rotation of any point $\boldsymbol{a} \in [0,1]^d$ is uniformly distributed. This is geometrically reasonable. The value $\boldsymbol{a}+\boldsymbol{U}$ mod 1 is the random point $\boldsymbol{U}$ shifted right with wraparound by the amount $\boldsymbol{a}$. The result $\boldsymbol{a} + \boldsymbol{U}$ is in a region E if and only if $\boldsymbol{U}$ is in E shifted left by $\boldsymbol{a}$, again with wraparound. Shifting the region left might break it into pieces but does not change the total volume of the pieces. Therefore we expect $\mathbb{P}(\boldsymbol{a}+\boldsymbol{U} \in E) = \mathbf{vol}(E) = \mathbb{P}(\boldsymbol{U} \in E)$. The proof is as follows.

Proposition 17.1. *Let $\boldsymbol{a} \in [0,1]^d$ for $d \geqslant 1$. If $\boldsymbol{x} = \boldsymbol{a} + \boldsymbol{U} \bmod 1$ for $\boldsymbol{U} \sim \mathbf{U}(0,1)^d$ then $\boldsymbol{x} \sim \mathbf{U}(0,1)^d$.*

Proof. We begin with $d = 1$. If $x \in (0,1)$ then $\mathbb{P}(\{a+U\} < x)$ equals

$$\int_0^1 \mathbb{1}_{\{a+u\}<x} \,\mathrm{d}u = \int_0^1 \mathbb{1}_{a+u<x}\mathbb{1}_{a+u<1} + \mathbb{1}_{a+u-1<x}\mathbb{1}_{a+u\geqslant 1} \,\mathrm{d}u.$$

Now

$$\begin{aligned}\int_0^1 \mathbb{1}_{a+u<x}\mathbb{1}_{a+u<1} \,\mathrm{d}u &= \int_0^1 \mathbb{1}_{u<x-a} \,\mathrm{d}u = \max(0, x-a), \quad \text{and}\\ \int_0^1 \mathbb{1}_{1-a\leqslant u<1-a+x} \,\mathrm{d}u &= \min(1, 1-a+x) - (1-a) = \min(a,x).\end{aligned}$$

Either $x > a$ or $x \leqslant a$, but $\max(0, x-a) + \min(a,x) = x$ holds in both cases. Therefore $\mathbb{P}(\{a+U\} < x) = x$ and the result is established for $d = 1$. For $d \geqslant 1$ each component $\{a_j + U_j\} \sim \mathbf{U}(0,1)$. Then, because U_j are independent, so are $\{a_j + U_j\}$, and therefore $\{\boldsymbol{a} + \boldsymbol{U}\} \sim \mathbf{U}(0,1)^d$. □

A Cranley-Patterson rotation of low discrepancy points has low discrepancy. For example, if $\boldsymbol{x}_i = \boldsymbol{a}_i + \boldsymbol{U} \bmod 1$ then

$$D_n(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \leqslant 2^d D_n(\boldsymbol{a}_1, \dots, \boldsymbol{a}_n) \tag{17.9}$$

holds for any $\boldsymbol{U}$. See Exercise 17.2. Combining equation (17.9) with Proposition 17.1 shows that a Cranley-Patterson rotation of low discrepancy points yield a randomized quasi-Monte Carlo rule so that the unbiasedness and variance estimation properties in §17.1 apply. The factor 2^d in (17.9) turns into 4^d in a variance bound. It is however extremely conservative stemming from a worst case or even impossible pattern among the rotated points. The sample variance of $\hat{\mu}_r$ will depend on the specific points $\boldsymbol{a}_i$ and integrand f not on worst case $\boldsymbol{a}_i$ or f. It is also not clear whether the factor 2^d is even close to best possible for any QMC points that one might use.

While Cranley-Patterson rotation of low discrepancy points will retain their low discrepancy, Cranley-Patterson rotation of badly non-uniform points cannot meaningfully improve them. For one thing, the original bad points would be an inverse Cranley-Patterson rotation of our new good points and we argued above that these rotations could not turn good points into bad ones. If there is a dense cluster of points somewhere, after rotation that dense cluster appears in another place, perhaps split at the boundary of the unit cube. The same applies to a void. The unpleasant stripes and gaps from the Kronecker points of §15.14 would simply move to new locations parallel to their old ones under Cranley-Patterson rotation.

A Cranley-Patterson rotation of a (t,m,d)-net in base b has low discrepancy, but the result is not usually another (t,m,d)-net. Cranley-Patterson rotations are more commonly applied to lattice rules. The resulting points are then a randomly shifted lattice rule. The variance of lattice rules under Cranley-Patterson rotation can be expressed in terms of the dual lattice of the sampling points and the Fourier coefficients of the integrand as follows.

Theorem 17.2. *Let $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n \in [0,1)^d$ be a lattice rule with dual lattice $D \in \mathbb{Z}^d$. Let f be a square integrable function on $[0,1)^d$ with Fourier coefficients $\hat{f}(\boldsymbol{h})$ for $\boldsymbol{h} \in \mathbb{Z}^d$. Let $\boldsymbol{x}_i = \boldsymbol{a}_i + \boldsymbol{U} \bmod 1$ for $\boldsymbol{U} \sim \mathbf{U}(0,1)^d$ and $i = 1, \dots, n$. Then*

$$\mathrm{Var}\Bigl(\frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)\Bigr) = \sum_{\boldsymbol{h}\in D_*} \hat{f}(\boldsymbol{h})^2 \tag{17.10}$$

where $D_* = D \setminus \{\mathbf{0}\}$.

Proof. Tuffin (1998) proved it assuming f has an absolutely convergent Fourier series. L'Ecuyer and Lemieux (2000) proved the version above. □

It is instructive to compare equation (17.10) with the result for crude Monte Carlo. In that case the variance is

$$\mathrm{Var}(\hat{\mu}_{\mathrm{IID}}) = \frac{\sigma^2}{n} = \frac{1}{n}\sum_{\boldsymbol{h}\in\mathbb{Z}^d\setminus\{\mathbf{0}\}} \hat{f}(\boldsymbol{h})^2.$$

Letting $\hat{\mu}_{\mathrm{CranPat}}$ refer to Cranley-Patterson rotation of a lattice rule we obtain

$$0 \leqslant \frac{\mathrm{Var}(\hat{\mu}_{\mathrm{CranPat}})}{\mathrm{Var}(\hat{\mu}_{\mathrm{IID}})} \leqslant n, \quad \text{and so} \quad 0 \leqslant \mathrm{Var}(\hat{\mu}_{\mathrm{CranPat}}) \leqslant \sigma^2. \tag{17.11}$$

The upper bound is disconcerting. In such worst cases the Cranley-Patterson method with n points is as effective as crude Monte Carlo with just 1 point.

At first sight, equation (17.11) may seem to contradict $n\mathrm{Var}(\hat{\mu}) \to 0$. The resolution is as follows: if we fix a function f of bounded variation and take a low discrepancy sequence of lattices with $n \to \infty$ then indeed $n\mathrm{Var}(\hat{\mu}_{\mathrm{CranPat}}) \to 0$. If instead, we fix an integration lattice on n points and then look for a worst case function $f = f_n$ for that given lattice, then we can find one with $\mathrm{Var}(\hat{\mu}_{\mathrm{CranPat}}) = \sigma^2 = n \times \mathrm{Var}(\hat{\mu}_{\mathrm{IID}})$, but we could not use the same f for all n.

The worst case is a consequence of shifted lattices being a cluster sample as described in §10.7. The nasty integrands are constant within clusters and vary between clusters. In one dimension, this case arises if $a_i = (i-1)/n$ and f happens to be a periodic function with period $1/n$.

These worst case functions are extremely unlikely to arise in real applications. On the other hand, a ratio like $\mathrm{Var}(\hat{\mu}_{\mathrm{CranPat}})/\mathrm{Var}(\hat{\mu}_{\mathrm{IID}}) = 100$ or 1000 is very far from the worst case when n is large and would be quite disappointing in practice. It is hard to know where the line is between realistic and unrealistic values for $\mathrm{Var}(\hat{\mu}_{\mathrm{CranPat}})/\mathrm{Var}(\hat{\mu}_{\mathrm{IID}})$ for lattice points.

n	Halton	Halton-b	Korobov	Korobov-b	MC
1021	0.340	0.180	0.160	0.0014	1.70
2039	0.270	0.086	0.058	0.0029	1.20
4093	0.150	0.035	0.084	0.00076	0.87
8191	0.066	0.052	0.021	0.00015	0.61
16381	0.054	0.028	0.011	0.000038	0.43

Table 17.1: Half-widths of approximate 99% confidence intervals for the mean wing weight, to two significant figures. Baker transformations indicated by 'b'. MC half-width quantities are described in the text.

n	Halton	Halton-b	Korobov	Korobov-b
1021	268.0775	268.0556	268.0946	268.0744
2039	268.1679	268.1686	268.0584	268.0755
4093	268.1102	268.0153	268.1123	268.0757
8191	268.0588	268.0723	268.0757	268.0752
16381	268.0752	268.0780	268.0763	268.0752

Table 17.2: Estimated mean wing weight based on 5 Cranley-Patterson rotations.

17.4 Example: wing weight function

We can use Cranley-Patterson rotations to estimate the accuracy of QMC on the wing weight function of §16.2. Table 17.1 results using $R = 5$ Cranley-Patterson rotations for Halton and Korobov points, with and without the baker transformation. In each case an approximate 99% confidence interval was constructed using as $\hat\mu_{\mathrm{pool}} \pm t_{(4)}^{0.995}\widehat{\mathrm{Var}}(\hat\mu_{\mathrm{pool}})^{1/2}$. The half-widths reported are $t_{(4)}^{0.995}\widehat{\mathrm{Var}}(\hat\mu_{\mathrm{pool}})^{1/2}/2$.

Table 17.2 gives the estimated values. For this problem RQMC has estimated the mean to greater accuracy than might be required. The Korobov method with a baker transformation has done particularly well. Using some replicates of that sequence we can estimate the Monte Carlo variance. For this function $\sigma \approx 48.08$ and so the RMSE for MC would be roughly $48.08/\sqrt{n}$. A rough counterpart to the half-widths reported in Table 17.1 would be $2.58 \times 48.08/\sqrt{5n}$ where the factor of 5 is there to give MC the same number of sample evaluations that the RQMC methods had.

It is remarkable how well the baker transformation applied to the Korobov points has done. There is some theoretical reason to expect this in §16.6. Also, the mean dimension of this function is small. Equation (17.8) describes an integral that for the wing weight function would have 11 dimensional input. Using some 11-dimensional RQMC points we find that the mean dimension in the superposition sense for the wing weight function is about 1.012. This means that at least 98.8% of the variance of f comes from an additive approximation

(Exercise 17.6). We might not have guessed from the formula in §16.2 that this function is so nearly additive. Taylor's theorem implies that it would be nearly linear and hence additive over a small region where the gradient was not zero, but it is not obvious that the region of interest is that small. Indeed, it might not be, because the best additive approximation to the function might not be linear. For a smooth integrand that is about 99% additive, we would ordinarily find Latin hypercube sampling to have about 1/100 times the variance of plain MC and yield half-widths about 1/10 times as large. The RQMC methods are doing even better than that, so they must be accurately estimating the integrals of f_u for some u with $|u| \geqslant 2$.

From the replicates, we have a much better idea of the sampling error than we got from just computing $\hat\mu$ for varying n. We might still wonder whether the widths for the confidence intervals were accurately estimated. If we would estimate the accuracy of those widths, then we would face a higher order question about the accuracy of the estimates of accuracy. Mosteller and Tukey (1968) refer to a staircase of inference with primary, secondary, tertiary and even higher order quantities each one a more challenging estimate of the accuracy of the preceding one. In statistics, it is common to just stop with the secondary quantity, here a confidence interval. The QMC estimates of Chapters 15 and 16 stop with the primary quantity, $\hat\mu$.

In this instance, getting 5 replicates was not so expensive. We can do 200 times the work and see what happens for $R = 1000$. For Korobov and Halton points, with and without the baker transformation, and for all 5 sample sizes, we get 20 histograms. Figure 17.2 shows four of them. One is for Korobov points with the baker transformation at the largest sample size, $n = 16381$. This was the most accurate method. Another is for Korobov and baker, with $n = 2039$, the second smallest sample size. This was, subjectively, the most visibly non-Gaussian histogram. It has two clear modes. The second most visibly non-Gaussian histogram was for Halton with baker and $n = 16381$. The histogram for shifted Halton points at that sample size looks nearly Gaussian.

The central limit theorem applies quite well to averages from any of those distributions, even the bimodal one, as $R \to \infty$, though we could reasonably doubt whether $R = 5$ is asymptotic. Using our 1000 replicates we can see how accurate the confidence intervals were. We will treat the average of all 1000 estimates from the Korobov points with the baker transformation and $n = 16381$ as if it were the true integral μ. Then we can inspect the distribution of

$$t \equiv \frac{\hat\mu_{\mathrm{pool}} - \mu}{\sqrt{\widehat{\mathrm{Var}}(\hat\mu_{\mathrm{pool}})}} \tag{17.12}$$

by repeatedly computing with a simple random sample of R of those 1000 estimates.

Table 17.3 shows the coverage levels attained by our approximate 99% confidence intervals among 100,000 repeated samplings from the histograms in Figure 17.2. The worst one is for $R = 5$ and the bimodal histogram discussed

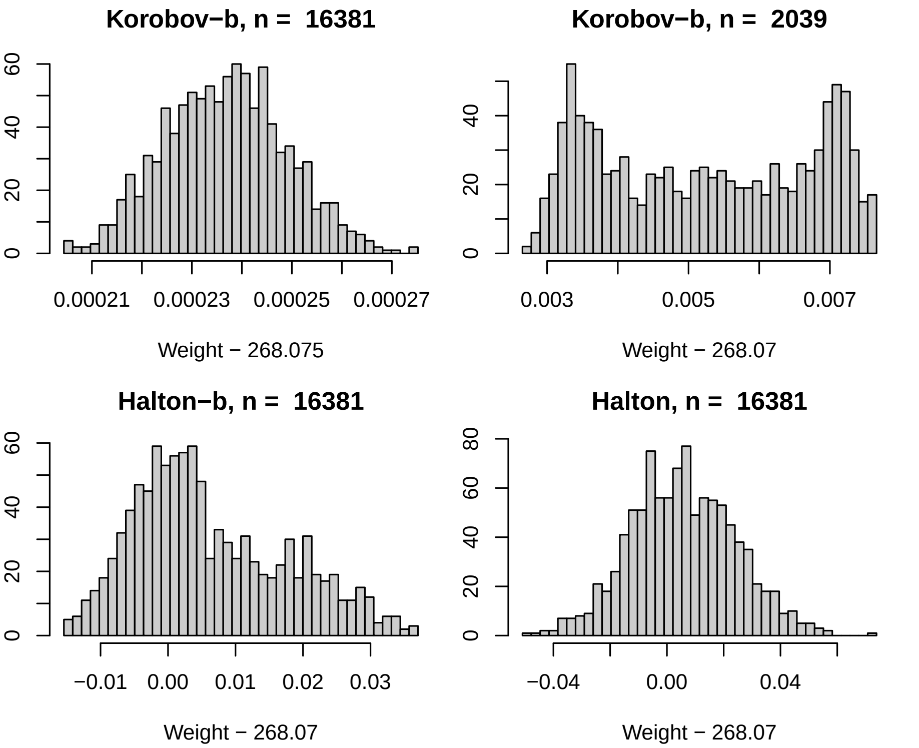

Figure 17.2: Histograms of $R = 1000$ replicated estimates of mean wing weight for four RQMC estimates.

above. A user in that situation would only have about 96.50% coverage not 99%. Coverage would be much better for a user who had $R = 10$ and it would be quite excellent for a user with $R = 30$.

Table 17.4 gives another way to judge the accuracy of the confidence intervals. It shows the estimated 99'th percentiles of the distribution of $|t|$. For $R = 30$, the t-tables give 2.75 as the 99'th percentile and the more appropriate values are very close to 2.75. The worst case in that table is for the bimodal histogram with $R = 5$. The t-tables give 4.03 and one would have needed nearly double that to get 99% coverage.

The practical problem we face in choosing R is that we don't know ahead of time what the histogram of $\hat{\mu}_r$ will look like. The choice $R = 30$ is a commonly quoted rule of thumb in statistics. There is however the usual rule of thumb arms race: for any $\hat{\mu}_r$ with finite variance there's an $R < \infty$ where the CLT gives good coverage, while for any $R < \infty$, there is a finite variance distribution for $\hat{\mu}_r$ where the CLT will give poor coverage. There is still a role for judgment in choosing R.

Attained coverage	$R = 5$	$R = 10$	$R = 30$
Korobov-b 16831	98.29	98.84	99.01
Korobov-b 2039	96.50	98.41	98.97
Halton-b 16831	97.63	98.48	98.68
Halton 16831	98.24	98.83	99.08

Table 17.3: Attained coverage percents of approximate 99% confidence intervals based on 100,000 samples of size R from the histograms in Figure 17.2.

$\lvert t\rvert^{0.99}$	$R = 5$	$R = 10$	$R = 30$	$R = \infty$
Korobov-b 16831	4.72	3.26	2.75	2.58
Korobov-b 2039	7.55	3.63	2.76	2.58
Halton-b 16831	5.31	3.53	2.86	2.58
Halton 16831	4.83	3.28	2.71	2.58
Gaussian	4.03	3.17	2.75	2.58

Table 17.4: Ninety-ninth percentile of $|t|$ for t given by (17.12) based on 100,000 samples of size R from the histograms in Figure 17.2. The last column is from the central limit theorem. The bottom row is from the $t_{(R-1)}$ distribution for sampling Gaussian values.

17.5 Scrambled nets

A Cranley-Patterson rotation of a digital net does not preserve the stratification properties that define a digital net. Those properties can however be preserved through certain strategic randomizations of the digits of the points. The scrambling method in this section is the most studied, but a direct implementation requires storage proportional to nd. Computer memory is much less expensive now than it was when that scramble was proposed, so this issue is less pressing. After presenting results for this scrambling we will look in §17.6 at alternative scrambles.

Suppose that $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ are a (t, m, d)-net in base b. Imagine that those points are firmly embedded specks in a d dimensional solid cube $[0, 1)^d$. If we could split that cube $[0, 1)^d$ into b congruent slabs $[\ell/b, (\ell + 1)/b) \times [0, 1)^{d-1}$ for $\ell = 0, 1, \dots, b - 1$ and shuffle those slabs in random order, then the final positions of $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ would still yield a (t, m, d)-net.

The geometric reasoning is as follows. Suppose first that an elementary interval E in base b is contained within one of the b slabs. Then it has a counterpart of equal volume in each of the other $b - 1$ slabs. The shuffling operation moves points $\boldsymbol{a}_i$ into E that were formerly in either E (with probability $1/b$) or one of its counterparts. If $\mathbf{vol}(E) \geqslant b^{t-m}$ then the net property ensures that E ends up with the correct number of shuffled points $\boldsymbol{a}_i$. If instead, E is not contained within a slab then it extends across all b slabs. Shuffling the slabs

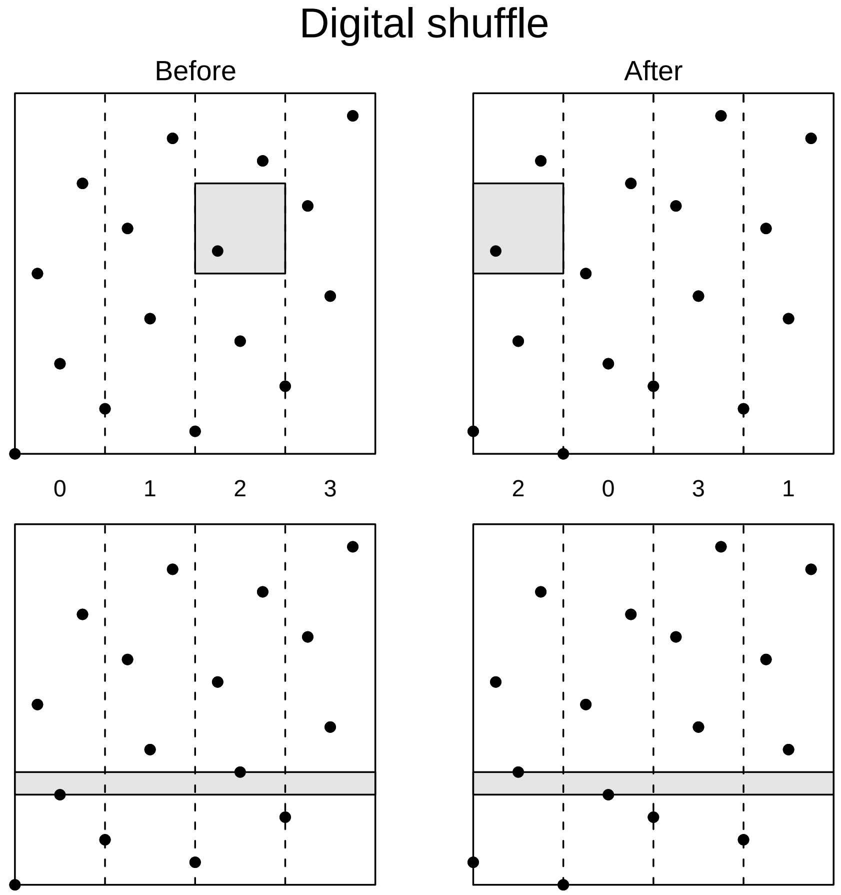

Figure 17.3: This figure illustrates the first step of a base 4 scramble of 16 points in the unit square. The square is split into vertical slabs 0, 1, 2 and 3. The slabs are rearranged in order 2, 0, 3 and 1. The top panels show how the point that ends up in $[0, 1/4) \times [1/2, 3/4)$ started in $[1/2, 3/4) \times [1/2, 3/4)$. The bottom panels show a point being reordered within $[0, 1) \times [1/4, 5/16)$.

moves some points $\boldsymbol{a}_i$ around within E but does not change their number and so equidistribution is preserved in this case too.

A ***nested uniform scramble*** proceeds by slicing each of the b slabs into b thinner ones and scrambling the thinner slabs within their respective original slabs. Then slabs within those slabs are scrambled and the process continues recursively. Conceptually this shuffling goes on forever, but in practice the process can stop when the slabs are too thin to affect the floating point representation of x_{ij}. Finally, the other $d-1$ dimensions are sliced and scrambled independently, in the same way as the first.

The scrambling operation can be represented in terms of base b digits. For simplicity we consider $d = 1$ and so instead of scrambling each a_{ij} into x_{ij} we drop the subscript j and scramble a point a_i into a point x_i. Since we will apply the same operation to all of $a_1, \dots, a_n$ we drop the subscript i as well and scramble one single point

$$a = \sum_{k=0}^{\infty} a_k b^{-k-1} \in [0,1) \quad \text{into} \quad x = \sum_{k=0}^{\infty} x_k b^{-k-1},$$

with the understanding that (temporarily) a_k and x_k refer to the k'th digits of a and x, and not the k'th points of a sequence. The digits $x_k \in \{0, 1, \dots, b-1\}$ are obtained by scrambling as follows:

$$\begin{aligned} x_0 &= \pi_\bullet(a_0) \\ x_1 &= \pi_\bullet\, a_0(a_1) \\ x_2 &= \pi_\bullet\, a_0, a_1(a_2) \\ &\vdots \\ x_k &= \pi_\bullet\, a_0, a_1, \cdots, a_{k-1}(a_k) \end{aligned} \tag{17.13}$$

where the various subscripted $\pi(\cdot)$'s are independent uniform random permutations of $\{0, 1, \dots, b-1\}$. The permutation applied to digit a_k depends on digits 0 through $k-1$ of a.

To scramble n points, the same set of permutations is applied to all of $a_1, \dots, a_n$ creating $x_1, \dots, x_n$. To scramble d dimensional points $\boldsymbol{a}_i \in [0,1)^d$, the j'th components are scrambled using independently generated permutations $\pi_{j\bullet}$, $\pi_{j\bullet\, a_0}$, $\pi_{j\bullet\, a_0,a_1}$, and so on.

check: digit convention, starting at 0 vs at 1

Potentially b^k permutations are needed for digit $k \geqslant 0$ in each of d components of the net. But a (t,m,d)-net in base b has only b^m points. Thus only $b^{\max(k,m)}$ permutations are needed for the digits of each component $j = 1, \dots, d$. For $b = 2$, we then need nd permutations and each permutation is either $(0,1)$ or $(1,0)$, so only nd bits are needed.

Proposition 17.2. *Let $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ be a (t,m,d)-net in base b, and suppose that $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are a nested uniform scramble of $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$. Then $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are a (t,m,d)-net in base b, with probability 1. Let $\boldsymbol{a}_i$ for $i \geqslant 1$ be a (t,d)-sequence in base b, and suppose that $\boldsymbol{x}_i$ are a nested uniform scramble of $\boldsymbol{a}_i$. Then $\boldsymbol{x}_i$ are a (t,d)-sequence in base b, with probability 1.*

Proof. This is proved in Owen (1995). □

The clause 'with probability 1' merits some explanation. Suppose that $a_1 = 0$ and $a_2 = 1/2$. Then a_1 and a_2 taken together comprise a $(0,1,1)$-net in base 2. The digits of a_1 are $0.0000\cdots$ and those of a_2 are $0.1000\cdots$, both in base 2. Suppose that every digit a_{11k} (for $k \geqslant 1$) in the infinite tail of 0s for a_1 were permuted to the value 1. Then we would get $x_1 = 1/2$ or $x_1 = 1$, depending on whether digit x_{110} of x_1 came out to 0 or 1. In the first case, $x_1 = 1/2$ and the interval $[0, 1/2)$ would contain neither x_1 nor x_2. The second case would put $x_1 = 1$ leaving either $[0, 1/2)$ or $[1/2, 1)$ without a sample point.

More generally, if for some $i \geqslant 1$ and $j \in \{1, \dots, d\}$ we should ever get an infinite sequence of consecutive $b - 1$'s as permuted values of $a_{ijk} = 0$ for $k \geqslant k_*$ then the resulting points could fail to properly populate some elementary interval in base b. The probability of this ever happening in a finite (or even countably infinite) number of trials is 0 and that is why the probability that $\boldsymbol{x}_i$ are a digital net (or sequence) is 1. Also, putting a point at 1/2 that should have been inside $[0, 1/2)$ is not a large error. The point is at distance 0 from the desired interval!

An RQMC method also requires $\boldsymbol{x}_i \sim \mathbf{U}[0, 1]^d$. Since this is a property of the individual points, it suffices to verify that any single point $\boldsymbol{a} \in [0, 1)^d$ when scrambled yields a point $\boldsymbol{x} \sim \mathbf{U}[0, 1]^d$.

Proposition 17.3. *For $\boldsymbol{a} \in [0, 1)^d$ let $\boldsymbol{x}$ be a nested uniform scramble of $\boldsymbol{a}$. Then $\boldsymbol{x} \sim \mathbf{U}[0, 1]^d$.*

Proof. This is proved in Owen (1995). □

The idea of the proof is as follows. Scrambling the first k digits of a_j places x_j into one of b^k intervals $[\ell b^{-k}, (\ell + 1)b^{-k})$, for $\ell = 0, 1, \dots, b^k - 1$, each with probability b^{-k}. Letting $k \to \infty$ this means that $x_j \sim \mathbf{U}[0, 1]$. The d components of $\boldsymbol{x}$ are independent, so $\boldsymbol{x} \sim \mathbf{U}[0, 1]^d$.

Scrambled nets have some significant advantages over randomly shifted lattice rules. First, their worst case performance relative to plain Monte Carlo is better. Second, they have an error cancellation property that, for smooth enough f, makes them attain a better rate of convergence than unscrambled nets obtain.

Theorem 17.3. *Let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ be a nested uniform scramble of a $(0, m, d)$-net in base $b \geqslant \max(d, 2)$. Let f be a function on $[0, 1]^d$ such that $f(\boldsymbol{x})$ has variance $\sigma^2 < \infty$ when $\boldsymbol{x} \sim \mathbf{U}[0, 1]^d$. Then*

$$\mathrm{Var}\Bigl(\frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)\Bigr) \leqslant \Bigl(\frac{b}{b-1}\Bigr)^{\min(m,d-1)}\frac{\sigma^2}{n} \leqslant \Bigl(\frac{b}{b-1}\Bigr)^{b-1}\frac{\sigma^2}{n} \leqslant \frac{e\sigma^2}{n}.$$

Proof. See Owen (1997a). □

Corollary 17.1. *Let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in [0, 1]^d$ be a Latin hypercube sample. Let f be a function on $[0, 1]^d$ such that $f(\boldsymbol{x})$ has variance $\sigma^2 < \infty$ when $\boldsymbol{x} \sim \mathbf{U}[0, 1]^d$. Then*

$$\mathrm{Var}\Bigl(\frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)\Bigr) \leqslant \frac{\sigma^2}{n-1}.$$

Proof. For $i = 1, \dots, n$, let $\boldsymbol{a}_i = ((i - 1)/n, \dots, (i - 1)/n)$ which is a $(0, 1, d)$-net in base n. Scrambling $\boldsymbol{a}_i$ generates a Latin hypercube sample. The result follows from Theorem 17.3. □

A common way to get a $(0,m,d)$-net in base b is to take the first b^m points from one of Faure's $(0,d)$-sequences. This requires a prime base $b \geqslant d$, or a prime power $b = p^r \geqslant d$ if we use the generalization of Faure's construction in Niederreiter (1987). The result is a variance that is never more than $e \doteq 2.7183$ times the Monte Carlo variance σ^2/n, because $(b/(b-1))^{b-1}$ increases from 2 to e as b goes from 2 to ∞. Like shifted lattice sampling, this worst case requires a quite unusual function f. Unlike shifted lattices, the worst performance relative to plain Monte Carlo is a variance inflation factor of at most e instead of n.

Bounds are also available for digital nets, like Sobol's, with $t > 0$.

Theorem 17.4. *Let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in [0,1)^d$ be a nested uniform scramble of a (t,m,d)-net in base b. Let f be a function on $[0,1]^d$ such that $f(\boldsymbol{x})$ has variance $\sigma^2 < \infty$ when $\boldsymbol{x} \sim \mathbf{U}[0,1]^d$. Then*

$$\mathrm{Var}\Bigl(\frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)\Bigr) \leqslant b^t \Bigl(\frac{b+1}{b-1}\Bigr)^{d-1} \frac{\sigma^2}{n}.$$

Proof. See Owen (1997a). □

One way to construct a net with $t > 0$ is to merge b^t identical copies of a net with $t = 0$. Of course nobody would do that, but the example serves to show that we cannot remove the leading factor of b^t from Theorem 17.4 and get a result that will hold for all scrambled nets. It is not known whether that factor of b^t applies to commonly used digital nets, such as those of Sobol'.

Suppose that $f(\boldsymbol{x})$ is a sum of some other functions, each of which is constant inside an elementary interval in base b of volume b^{m-t}. Then $\hat\mu = \mu$ with probability 1 under scrambled net sampling. More realistically, we can approximate f by such a sum of functions. If f is smooth enough, then as m increases the best such approximation rapidly converges to f.

For our purposes here, the function f on $[0,1]^d$ is a ***smooth function*** if

$$\frac{\partial}{\partial x_{j_1}}\frac{\partial}{\partial x_{j_2}}\cdots\frac{\partial}{\partial x_{j_d}} f(\boldsymbol{x})$$

is continuous on $[0,1]^d$ for any distinct $j_1, \dots, j_d \in \{1, \dots, d\}$. This condition also ensures that the order of partial differentiation does not matter.

Theorem 17.5. *Let $f(\boldsymbol{x})$ be a smooth function defined on $[0,1]^d$ and suppose that $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ is a (λ, t, m, d)-net in base $b \geqslant 2$ (so $n = \lambda b^m$). If $\boldsymbol{x}_i$ are a nested uniform scramble of $\boldsymbol{a}_i$, then as $n \to \infty$ with $1 \leqslant \lambda < b$,*

$$\mathrm{Var}\Bigl(\frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)\Bigr) = O\Bigl(\frac{\log(n)^{d-1}}{n^3}\Bigr).$$

Proof. This is from Owen (1997b), originally with a Lipshitz condition on d-fold partial derivative of f taken once with respect to each component. Owen (2008) weakens the assumptions to the ones given here and corrects a Lemma from the earlier paper. □

The root mean square error in Theorem 17.5 is $O(n^{-3/2+\epsilon})$ which compares favorably to the rate $O(n^{-1+\epsilon})$ for unscrambled nets. The reduction of about $O(n^{-1/2})$ may be interpreted as arising from random error cancellations. Random errors tend to cancel, while deterministic ones need not.

To compare nested uniform scrambling and Cranley-Patterson rotations, consider the points $a_i = (i-1)/n$ for $i = 1, \dots, n$. These are simultaneously a lattice rule with $z = (1)$ as well as a $(0, 1, 1)$-net in base n. Applying a Cranley-Patterson rotation shifts them all the same distance U (with wraparound). As a result, Cranley-Patterson rotations give points with the same joint distribution as $X_i = (i - 1 + U)/n$ for $U \sim \mathbf{U}[0, 1)$. Applying a nested uniform scramble in base n is quite different. The first permutation shuffles the intervals $[j/n, (j+1)/n)$ changing nothing. The subsequent permutations take the point at j/n and distribute it uniformly in $[j/n, (j + 1)/n)$ with different intervals being independent. Nested uniform sampling thus delivers a stratified sample $X_i = (i-1+U_i)/n$ for independent $U_i \sim \mathbf{U}[0, 1)$. Stratified sampling achieves a variance of order $O(n^{-3})$ for smooth f due to error cancellation between strata.

Scrambled nets with $t = 0$ obey a central limit theorem as $n \to \infty$. It is not known whether such a limit holds for $t > 0$.

Theorem 17.6. *Let f be a function on $[0, 1]^d$ with $\int f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \mu$,*

$$\biggl|\frac{\partial^d}{\partial x_1 \cdots \partial x_d} f(\boldsymbol{x}) - \frac{\partial^d}{\partial x_1 \cdots \partial x_d} f(\widetilde{\boldsymbol{x}})\biggr| \leqslant B\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\|^{\beta}$$

for some $B \geqslant 0$ and $0 < \beta \leqslant 1$, and $\int (\partial^d f(\boldsymbol{x})/\prod_{j=1}^d x_j)^2\,\mathrm{d}\boldsymbol{x} > 0$. Let $\hat{\mu} = (1/n)\sum_{i=1}^n f(\boldsymbol{x}_i)$ where $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ is a scrambled $(0, m, d)$-net in base b. Then

$$\mathbb{P}\biggl(\frac{\hat{\mu} - \mu}{\sqrt{\mathrm{Var}(\hat{\mu})}} \leqslant z\biggr) \to \Phi(z)$$

as $m \to \infty$.

Proof. Loh (2003). □

Theorem 17.6 assumes some smoothness for f, quite unlike the usual central limit theorem. It is clear that some smoothness condition on f is necessary in a scrambled net central limit theorem. For example, consider $f(x) = \sum_{k=1}^{\infty} \alpha_k \mathbb{1}_{x<b^{-k}}$ and points x of a scrambled van der Corput sequence in base b. The value of $\hat{\mu}$ depends entirely on how close to zero the smallest of $x_1, \dots, x_n$ happens to be. As a result, $\hat{\mu}$ is not normally distributed since the most probable value for $\hat{\mu}$ will have probability $(b-1)/b$.

Scrambling also improves higher order nets. These are constructed by the interleaving method of §7.3.

Theorem 17.7. *Let f be a function on $[0, 1]^d$ whose partial derivatives of order up to $k \geqslant 1$ in each component have finite mean square. Let $\boldsymbol{z}_1, \dots, \boldsymbol{z}_n$ be formed as a nested uniform scramble of a digital (t, m, kd)-net in base b and let*

$\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ be a k'th order digital net formed by interleaving the components of $\boldsymbol{z}_1, \dots, \boldsymbol{z}_n$. Letting $\hat\mu = (1/n)\sum_{i=1}^n f(\boldsymbol{x}_i)$, we have

$$\mathrm{Var}(\hat\mu) = O(n^{-2k-1}\log(n)^{ks+k}) = O(n^{-2k-1+\epsilon}),$$

for any $\epsilon > 0$.

Proof. See Dick (2011). □

The RMSE for scrambled high order nets is $O(n^{-k-1/2+\epsilon})$ which compares favorably to the deterministic error $O(n^{-k})$ for higher order nets. If the smoothness of the function is described by an integer k' and the net is of order k then the RMSE is $O(n^{-\min(k,k')-1/2+\epsilon})$.

17.6 More scrambles

The scrambling method used in §17.5 requires an amount of storage proportional to n. Computers have much more memory now than when that scramble was proposed, and simpler methods were devised to cope. The most important one is a partial derandomization of that scramble, due to Matoušek, which attains the same variance, with less storage. This section may be skipped on first reading.

Digital shift scrambling is the first alternative we consider. It is very easy to apply, requires the same storage as Cranley-Patterson rotations, and preserves the digital net structure. It does not satisfy the same variance bounds or have the same convergence rates that nested uniform scrambling does.

The digital shift is a digital analogue of the random shift modulo 1 used with lattice samples. First, we describe a digital addition operation $\oplus_b$ acting on points $x, y \in [0, 1)$ for integer base $b \geqslant 2$. Let $x = \sum_{k=0}^{\infty} x_k b^{-k-1}$ and $y = \sum_{k=0}^{\infty} y_k b^{-k-1}$ for $x_k, y_k \in \{0, 1, \dots, b-1\}$. Then

$$\begin{aligned} x \oplus_b y = z \equiv \sum_{k=0}^{\infty} z_k b^{-k-1}, &\quad \text{where} \\ z_k = x_k + y_k \ \mathsf{mod}\ b. & \end{aligned} \tag{17.14}$$

When b is understood, we may write $x \oplus y$. We will use $\oplus$ to digitally add a random point to our QMC points as described below.

Before using $\oplus$, we need to add a condition that makes it well defined. In base 10, the number 1/2 can be written two ways, as 0.5 or as $0.4999\cdots$ with an infinite tail of 9s. Similarly, in base b any number ℓ/b^k for $k \geqslant 1$ and $1 \leqslant \ell < b^k$ has two representations, one ending in an infinite tail of 0s and the other ending in $b-1$s. When applying $\oplus_b$ to points $x, y \in [0, 1)$ we always choose the representation ending in 0s over the one ending in $b-1$s.

The number 1 is awkward to handle digitally. Representing it as $0.d_0d_1d_2\cdots$ requires precisely the infinite tail of $b-1$s that we have excluded. This is why we work with $x, y \in [0, 1)$. It is however still possible to get 1 as a sum. For example, if $x = 0.1111\cdots$ and $y = 0.3333\cdots$ both in base 5 then $x \oplus_5 y = 0.4444\cdots = 1$.

i	a_i	a_i	U	$a_i \oplus U_i$
1	0	0.000	0.110101	0.110101
2	1/8	0.001	0.110101	0.111101
3	1/4	0.010	0.110101	0.100101
4	3/8	0.011	0.110101	0.101101
5	1/2	0.100	0.110101	0.010101
6	5/8	0.101	0.110101	0.011101
7	3/4	0.110	0.110101	0.000101
8	7/8	0.111	0.110101	0.001101

Table 17.5: This table illustrates a digital shift of a small net in $[0, 1)$. The original net is $a_1, \dots, a_8$, shown in the second column and (in base 2) in the third column. The random shift is $U = 0.110101$ (base 2), that is, $U = 0.828125$. The resulting points are in the final column.

We choose to handle this problem by treating a sum equal to 1 as if it were 0. This is the same choice we make for shifted lattices when we add numbers modulo 1.

A ***digital shift*** randomization of $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ has

$$\boldsymbol{x}_i = \boldsymbol{a}_i \oplus_b \boldsymbol{U} \quad \text{where} \quad \boldsymbol{U} \sim \mathbf{U}[0, 1)^d.$$

In practice we generate only the first k digits of U_j for $j = 1, \dots, d$ with each such digit $u_{jk} \sim \mathbf{U}\{0, 1, \dots, b-1\}$. Then we add them modulo b to the corresponding digits of a_{ij}.

Like the nested uniform scrambling of §17.5, digital shifts yield $\boldsymbol{x}_i \sim \mathbf{U}[0, 1)^d$. Digital shifts also preserve the digital net properties of $\boldsymbol{a}_i$ in base b (with probability one). As a result, digital shifts of (t, m, d)-nets and (t, d)-sequences in base b provide an RQMC method.

A small random digital shift is illustrated in Table 17.5. It starts with a_i in a $(0, 3, 1)$-net in base 2 defined by $a_i = (i - 1)/8$. The random shift U is only taken to 6 base 2 places for simplicity of exposition. The first 3 bits of a_i go through all 8 possible values and after adding U the first 3 bits of the result also go through all 8 possible values though the order has now changed. The original points end with a tail of 0s after the first 3 bits. As a result, the generated points all end in the same tail of digits that U has. A random digital shift, in this small example, gives the same distribution of points that we would get from a Cranley-Patterson rotation.

Digitally shifted nets do not have the same variance properties as fully scrambled nets. Their worst case performance relative to simple Monte Carlo is not very good. The root cause is that digital shifts, like Cranley-Patterson rotations, do not randomize the points enough. Operationally, if we knew $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ and one randomized point $\boldsymbol{x}_1$ then we could reconstruct $\boldsymbol{x}_2, \dots, \boldsymbol{x}_n$ for a digital shift, but not (outside of trivial cases) for a nested uniform scramble.

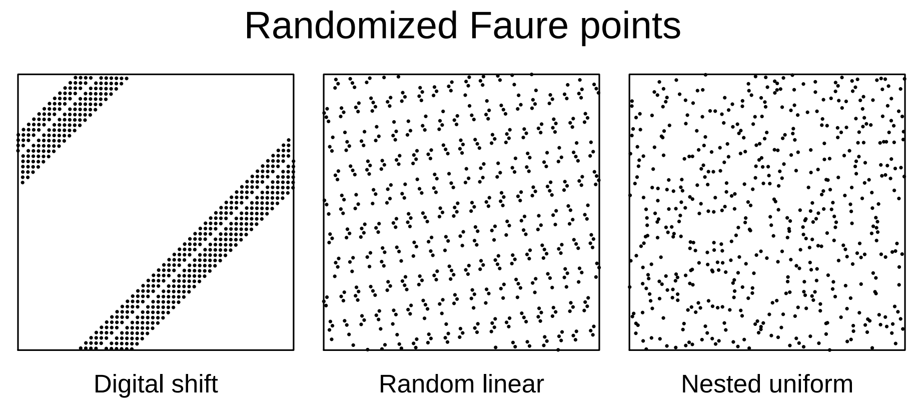

Figure 17.4: The left panel has a digital shift of the first 530 points of the first two components of Faure's $(53, 0)$-sequence in base 53. The center panel shows a random linear scramble. The right panel shows a nested uniform scramble.

Theorem 17.8. *Let $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n \in [0,1)^d$ be a (t, m, d)-net in base $b \geqslant 2$. For $i = 1, \dots, n$ let $\boldsymbol{x}_i = \boldsymbol{a}_i \oplus_b \boldsymbol{U}$ where $\boldsymbol{U} \sim \mathbf{U}[0,1)^d$. Then there exist functions $f(\boldsymbol{x})$ defined on $[0,1)^d$ such that*

$$\mathrm{Var}\biggl(\frac{1}{n}\sum_{i=1}^{n} f\bigl(\boldsymbol{a}_i + \boldsymbol{U}\bigr)\biggr) = \sigma^2$$

where σ^2 is the variance of $f(\boldsymbol{U})$.

Proof. This follows from Proposition 6.3 of Lemieux (2009). □

Theorem 17.8 is the digital scrambling counterpart to Theorem 17.2 for shifted lattices. As with shifted lattices and fully scrambled nets, the worst case functions for digital shifts are of a type quite unlikely to arise in real applications. Once again, while the very worst functions are implausible for applications, little is known about where to draw the line between realistic and implausibly pessimistic cases.

Digital shifts do not introduce enough randomness to get the error cancellation properties of scrambled nets. They do not attain a root mean squared error of $O(n^{-3/2+\epsilon})$ the way that scrambled nets do for smooth integrands.

The key to reducing the memory requirements of nested uniform scrambling, while randomizing the points enough, is to replace the uniform random scrambles by something simpler.

If p is a prime number, then a random linear permutation of $\{0, 1, \dots, p-1\}$ takes the form $\pi(a) = g + ha \,\mathsf{mod}\, p$ where $g \sim \mathbf{U}\{0, 1, \dots, p-1\}$ independently of $h \sim \mathbf{U}\{1, 2, \dots, p-1\}$. We only need to store g and h to represent this permutation. Like uniform random permutations, random linear permutations can be applied in a nested manner, yielding the random linear scrambles described next.

Once again we describe the scrambling of a single point $a \in [0,1)$ yielding $x \in [0,1]$. The same scramble gets applied to a sequence $a_1,\dots,a_n$ and independent scrambles are used for components $a_{1j},\dots,a_{nj}$ for $j=1,\dots,s$. A ***random linear scramble*** of $a=\sum_{\ell=0}^{\infty} a_\ell p^{-\ell-1}$ in a prime base p has digits

$$x_k = \sum_{\ell=0}^{k} M_{k\ell}\, a_\ell + C_k \mod p$$

for $k \geqslant 0$, where

$$\begin{aligned} M_{kk} &\sim \mathbf{U}\{1,\dots,b-1\}, \quad k \geqslant 0 \\ M_{k\ell} &\sim \mathbf{U}\{0,1,\dots,b-1\}, \quad k > \ell \geqslant 0, \quad \text{and} \\ C_k &\sim \mathbf{U}\{0,1,\dots,b-1\}, \quad k \geqslant 0 \end{aligned} \tag{17.15}$$

are all independent. The resulting point is $x=\sum_{k=0}^{\infty} x_k b^{-k-1}$. We can also write $x_k = h_k a_k + g_k \mod p$ where $h_k = M_{kk}$ and $g_k = \sum_{0\leqslant \ell<k} M_{k\ell}\, a_\ell + C_k \mod p$, the summation being 0 for $k=0$.

It is easy to see that $x \sim \mathbf{U}[0,1)$, because the terms C_k add a digital shift. The points retain their properties as a net because the permutations simply move elementary intervals around without altering the number of sample points in them. As a result, random linear scrambling of digital nets yields an RQMC method.

Theorem 17.9. *Let $\boldsymbol{x}_1,\dots,\boldsymbol{x}_n$ be a random linear scramble of a $(0,m,d)$-net in a prime base p. Let $\widetilde{\boldsymbol{x}}_1,\dots,\widetilde{\boldsymbol{x}}_n$ be a nested uniform scramble of the same net in base p. Then $\mathbb{E}((D^*_{n,2}(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)^2) = \mathbb{E}((D^*_{n,2}(\widetilde{\boldsymbol{x}}_1,\dots,\widetilde{\boldsymbol{x}}_n)^2)$.*

Proof. Matoušek (1998) shows that the random linear scramble satisfies conditions enumerated in Hickernell (1996a) for expected square discrepancy to match that of nested uniform scrambles. □

From Theorem 17.9 we see that random linear scrambles and nested uniform scrambles lead to the same expected square L^2-star discrepancies. The proofs work by showing that the joint distribution of any pair x_{ijk} and $x_{i'j'k'}$ of the RQMC digits is the same under both kinds of scrambling. It then follows that both scrambles result in the same variance for integral estimates $\hat\mu$. Instead of storing about nd permutations that nested uniform scrambling requires, we need instead about $K(K+1)/2$ base p digits (counting both M's and C's) where K is the number of base p digits we use to represent each number $x_{ij} \in [0,1)$.

The first 530 points of the Faure sequence in base 53 project into a small band containing 10 parallel lines of points with wraparound, when we select the first two components. A Cranley-Patterson rotation would simply move the band around. A digital shift of these points, as illustrated in the leftmost panel of Figure 17.4 looks similar to a Cranley-Patterson rotation. Both of these randomizations deliver points that are individually $\mathbf{U}[0,1]^d$ but they do nothing to improve the joint behavior of the points. A random linear scramble shakes up the points much more as shown by the middle panel. A nested uniform scramble randomizes the points and ends up with a less structured appearance than the other randomizations.

There is a peculiar blank stripe in the digital shift data which makes it look like two disjoint bands have wrapped around. The first 53^2 points of the Faure sequence have a similar blank region wrapping around the boundary of the unit square. That blank region maps onto the stripe in the first panel, under the digital shift.

Some digital scrambles (e.g., nested uniform and random linear scrambles) are excellent for breaking up clumps of points concentrated in lines or planes. Such clumps are the common flaw for Halton and Faure sequences. Digit scrambling is not very effective at countering the rectangular clumps and voids that appear in bad projections of the Sobol' sequence. The fundamental reason is that digital scrambles are constructed to preserve the number of points of a net within every sufficiently large elementary interval. They move points from any given elementary interval into some other one congruent to the first. If a list of points has bad coverage of an elementary interval, then their scrambled version will have bad coverage of some interval of the same size and shape, in a possibly different location.

Digital shifts failed to even separate the stripes that we see in small subsequences of the Faure sequence. Some other scrambles, simpler than random linear ones, do separate the stripes.

A ***positional scramble*** of $a \in [0,1)$ in base $b \geqslant 2$ takes the value $x = \sum_{k=0}^{\infty} \pi_k(a_k)b^{-k-1}$ where π_k are permutations of $\{0,1,\dots,b-1\}$. The permutations π_k in a positional scramble could all be independent. Or, we could make use of a positional scramble in which just one random permutation is used: $\pi_k = \pi_0$ for $k \geqslant 0$. In either kind of positional scramble, the permutations could be linear, when b is a prime number p, or they could be uniform. We have already seen one kind of positional scramble. A digital shift is a positional scramble with $\pi_k(a) = g_k + a \ \mathsf{mod}\ b$ where $g_k \sim \{0,\dots,b-1\}$ are independent.

Uniform random positional scrambles break up the stripes in leading subsequences of the Faure sequence. See Exercises 17.3 and 17.4. These are very easy to program and take little space, making them a better choice than digital shifts for scrambling the Faure and Halton sequences. They do not however attain the same variance that nested uniform and random linear scrambles do.

The points of a (t,m,d)-net in base b, for the usual constructions, have components that are integer multiples of b^{-m}. That is, their base b expansions have up to m nonzero digits followed by an infinite tail of zeros. When this happens, we do not have to explicitly scramble the infinite tail of zeros. The infinite tail of zeros will scramble into a term that adds independent $\mathbf{U}[0,b^{-m})^d$ distributed vectors to the generated points. For nested uniform scrambling and random linear scrambling, we can simply scramble the first m digits of $\boldsymbol{a}_i$ into $\widetilde{\boldsymbol{x}}_i$ and then deliver $\boldsymbol{x}_i = \widetilde{\boldsymbol{x}}_i + b^{-m}\boldsymbol{u}_i$ where $\boldsymbol{u}_i \sim \mathbf{U}[0,1)^d$ are independent. For digital scrambles and positional scrambles, we scramble the first m digits of $\boldsymbol{a}_i$ into $\widetilde{\boldsymbol{x}}_i$ and then deliver $\boldsymbol{x}_i = \widetilde{\boldsymbol{x}}_i + b^{-m}\boldsymbol{U}$ for one single point $\boldsymbol{U} \sim \mathbf{U}[0,1)^d$.

The only scramble for which a central limit theorem is known is the nested uniform scramble. To satisfy the central limit theorem a scramble must have an asymptotically negligible skewness: that is $\mathbb{E}((\hat\mu-\mu)^3)/\mathbb{E}((\hat\mu-\mu)^2)^{3/2}\to 0$. This condition can be met by arranging for the joint distribution of any three digits x_{ijk}, $x_{i'j'k'}$ and $x_{i''j''k''}$ used in the construction of $\boldsymbol{x}_1,\dots,\boldsymbol{x}_n$ to be the same as their joint distribution in nested uniform scrambling. Higher moment conditions are also necessary and they may be satisfied by methods that have the same higher order joint distributions (of digits) as nested uniform scrambling. It seems likely that the other scrambles considered here do not satisfy a central limit theorem.

17.7 Reducing effective dimension

Sometimes we can change the integrand in a way that is favorable to RQMC sampling. What we do is find another integrand $\tilde f$ that we know has $\int \tilde f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$, where we think that f has lower effective dimension as described by Definition 17.1 or 17.2. For instance, if we can find a way to make $\tilde f$ nearly a function of its first few input variables, then we may well have greatly improved RQMC accuracy. It is hard to be sure ahead of time that accuracy will increase. However, it is often easy to implement RQMC both ways with replicates and then see whether accuracy improved. Coding $\tilde f$ could take some care, but then measuring whether it is better might only take minutes or even seconds. An effective strategy is to use intuition and domain knowledge to devise alternative functions $\tilde f$, and then measure empirically whether the anticipated improvement materialized.

Many of the best examples of reducing effective dimension come from problems where f is a function of Brownian motion at d points, or more generally, a function of a high dimensional Gaussian random vector. Let $\boldsymbol{y}\sim\mathcal{N}(\mu,\Sigma)$ for a non-singular covariance matrix $\Sigma\in\mathbb{R}^{d\times d}$, and suppose that we want to find

$$\mu = \mathbb{E}(g(\boldsymbol{y})) = (2\pi)^{-d/2}|\Sigma|^{-1/2}\int_{\mathbb{R}^d} f(\boldsymbol{y})e^{-(\boldsymbol{y}-\mu)^{\mathsf{T}}\Sigma^{-1}(\boldsymbol{y}-\mu)/2}\,\mathrm{d}\boldsymbol{y}.$$

A Monte Carlo approach takes $\boldsymbol{x}_i \overset{\text{iid}}{\sim} \mathbf{U}(0,1)^d$, then $\boldsymbol{z}_i=\Phi^{-1}(\boldsymbol{x}_i)$ (componentwise), then $\boldsymbol{y}_i=\mu+C\boldsymbol{z}_i$ where $CC^{\mathsf{T}}=\Sigma$, and it averages $g(\boldsymbol{y}_i)$. The Monte Carlo estimate of $\mu=\mathbb{E}(f(\boldsymbol{x}))$ is

$$\hat\mu=\frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i),\quad\text{for}\quad f(\boldsymbol{x})=f(\boldsymbol{x};C)=g(\mu+C\Phi^{-1}(\boldsymbol{x})),\tag{17.16}$$

with $\boldsymbol{x}_i \overset{\text{iid}}{\sim} \mathbf{U}(0,1)^d$. For QMC, we replace $\boldsymbol{x}_i \overset{\text{iid}}{\sim} \mathbf{U}(0,1)^d$ by low discrepancy points $\boldsymbol{x}_1,\dots,\boldsymbol{x}_n$. We may get good results, but there is the risk of a bad result, even failure to converge, when g is an unbounded function on $\mathbb{R}^d$, because then $V_{\mathrm{HK}}(f)=\infty$. When f has finite variance, then both MC and RQMC will converge and RQMC will have an RMSE of $o(n^{-1/2})$. In that case, unboundedness of f is no longer a problem and neither are other ways (e.g., discontinuities) that $V_{\mathrm{HK}}(f) = \infty$ could arise.

If we replace C by $\tilde{C} = CQ$ for an orthogonal matrix Q, then

$$\boldsymbol{y} = \mu + CQ\boldsymbol{z} \sim \mathcal{N}(\mu, CQQ^{\mathsf{T}}C^{\mathsf{T}}) = \mathcal{N}(\mu, CC^{\mathsf{T}}) = \mathcal{N}(\mu, \Sigma).$$

We will get the same Monte Carlo mean and variance using $f(\boldsymbol{x}; C)$ or $f(\boldsymbol{x}; \tilde{C})$, because $\boldsymbol{y}_i$ have the same distribution either way. There may be speed differences between these choices arising from different costs of computing $C\boldsymbol{z}$ and $\tilde{C}\boldsymbol{z}$, but there is no difference in mean squared error for fixed n.

With (R)QMC, $f(\boldsymbol{x}; C)$ and $f(\boldsymbol{x}; \tilde{C})$ can be very different functions of $\boldsymbol{x}$ even when $CC^{\mathsf{T}} = \tilde{C}\tilde{C}^{\mathsf{T}} = \Sigma$. In Chapter 6 we considered generating a Brownian motion path in three ways: sampling the increments in time order, sampling them in arbitrary order using the Brownian bridge construction, and using principal components. The form of the matrix C for each of those choices can be found in that chapter.

Figure 17.5 illustrates these three constructions for Brownian motion at points $t/512$ for $t = 1, 2, \dots, 512$. Each construction takes a point $\boldsymbol{x} \in [0,1]^{512}$ to generate the sample path. The top panel shows a curve generated by the first 8 principal components. That curve uses the first 8 components of $\boldsymbol{x}$. The next $512 - 8 = 504$ components of $\boldsymbol{x}$ are used to complete the Brownian path. Three independent completions are shown. The first 8 components provide a 'skeleton' that is refined by the next 504 components. In terms of the gross outline of that sample path, those first 8 variables appear to be much more important than the others. The second panel shows the same quantities, replacing the principal components skeleton by a piecewise linear skeleton formed by Brownian bridge sampling. Again, there are three independent completions. The bottom panel shows a standard construction where the first 8 inputs generate the curve up to time $8/512 = 1/64$, along with three completions. The first 8 inputs do not greatly influence the path.

When the function $g(\cdot)$ depends on the coarse outline of the Brownian path, the principal components and Brownian bridge constructions can be expected to concentrate importance into the first few components of $\boldsymbol{x}$ reducing effective dimension in the truncation sense. Conversely, if we knew that $g(\cdot)$ depended only on details of how the skeleton is completed to form the path, and had nothing to do with the skeleton itself, then we would not expect these constructions to reduce effective dimension.

There is more to gain by reducing effective dimension in the superposition sense than the truncation sense, because RQMC points normally have good equidistribution in all projections onto one or two or a handful of coordinates, and the truncation approach does not take account of that property. It is hard to devise a way to reduce superposition dimension because that requires considering how the components of $\boldsymbol{x}$ interact to produce f. Strategies to concentrate importance into the first few components of $\boldsymbol{x}$ are more plentiful, probably because it is easier to think of how to make a few variables very important.

For any sampling strategy we come up with, there will be unfavorable integrands. If $\tilde{f}$ depends on $\boldsymbol{x}$ only through its final component, then the truncation

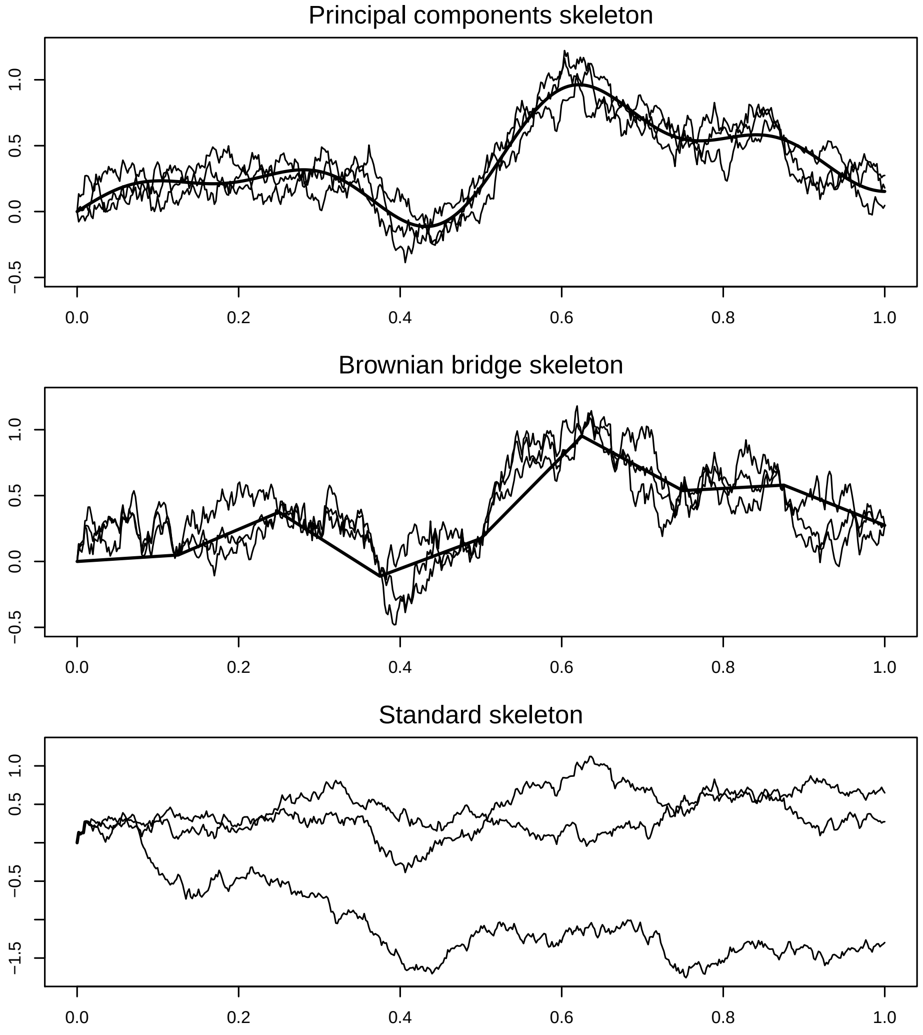

Figure 17.5: The top panel shows Brownian motion generated by principal components at 512 points. The thicker curve shows the skeleton from the first 8 principal components. There are three realizations completing the process using the remaining $512 - 8 = 504$ principal components. The second panel shows a piecewise linear skeleton of Brownian motion generated by 8 increments. There are three realizations completing the process. The bottom panel shows three sample paths sharing the same first 8 increments, with a different vertical scale.

dimension will be the largest possible value d. If $\tilde{f}$ depends on $\boldsymbol{x}$ only through the d-fold interaction $\tilde{f}_{\{1,2,\dots,d\}}$ then the superposition will be the largest possible value d. These outcomes seem unduly pessimistic, and they would be easily detected in practice.

A multivariate Gaussian random vector can be sampled in any order that we like, but the cost of the algebra and bookkeeping may depend on the order we choose. The principal components construction is available for any covariance matrix, even singular ones, though it does require a one time computation of up to $O(d^3)$ cost if the matrix Σ has no special structure to exploit. For Brownian motion it is inexpensive to sample time points in any order.

The constructions for Gaussian vectors can be generalized to multivariate t random vectors (see §5.2), either to sample via the t-copula of §5.6, or because the problem is defined in terms of multivariate t vectors. Sampling from a t distribution requires an additional component of $\boldsymbol{x}$ to generate the χ^2 random variable used in the denominator. Further strategies for reducing effective dimension are described in the chapter end notes.

17.8 Example: valuing an Asian option

Here we consider a well known test problem: valuing an option that depends on geometric Brownian motion. In this option, $S(t)$ is the value of some traded asset at time t. If the average of S over d time periods exceeds a strike price K, then the holder of the option is paid the difference. This provides a hedge against unaffordable upward price rises in the asset. The problem is to find a fair price to pay for that potential benefit. The price depends on an interest rate r, a measure σ^2 of the asset's price volatility, the time T at which the option is to be paid, and also the strike price K. We want to find $\mu = \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$, where

$$f(\boldsymbol{x}) = e^{-rT}\max\Bigl(\frac{1}{d}\sum_{j=1}^{d} S(t_j,\boldsymbol{x}) - K, 0\Bigr), \quad \text{for} \tag{17.17}$$

$$S(t_j,\boldsymbol{x}) = S(0)\exp\Bigl[(r-\sigma^2/2)t_j + \sigma\sqrt{T/d}\sum_{\ell=1}^{j}\Phi^{-1}(x_\ell)\Bigr]$$

with $t_j = jT/d$ for $j = 1,\dots,d$. Averaging over d time points is reasonable when the buyer needs to make regular purchase of the asset. A classic example is an airline hedging against price rises in jet fuel. This is called an Asian option because it was invented in Tokyo. We use the values $T = 16$, $S(0) = K = 100$, $r = 0.05$ and $\sigma = 0.3$ that were used in Hickernell et al. (2005).

The integrand (17.17) has infinite variation in the sense of Hardy and Krause. There are two causes. First, f is unbounded. Second, there is a kink at the set of $\boldsymbol{x}$ values for which $(1/d)\sum_{j=1}^{d} S(t_j,\boldsymbol{x}) = K$. However, because $\int_{[0,1]^d} f(\boldsymbol{x})^2\,\mathrm{d}\boldsymbol{x} < \infty$ we know that scrambled nets will provide an unbiased estimate with variance $o(1/n)$. This integrand is not smooth enough to satisfy the sufficient condition for variance $O(n^{-3+\epsilon})$ nor is it smooth enough to satisfy the sufficient condition for Loh's central limit theorem.

Roughly half of the time, this option ends up at value $f(\boldsymbol{x}) = 0$ and the rest of the time it is positive. Lowering the strike price K to well below $S(0)$ reduces the chance of a zero payout and raising K increases the chance of a zero payout. For very large K, a nonzero payout becomes such a rare event that importance sampling would be helpful.

The process $S(t, \boldsymbol{x})$ is a geometric Brownian motion. It depends on a plain Brownian motion sampled at times t_j, that is $B(t_j, \boldsymbol{x}) = \sqrt{T/d}\sum_{\ell=1}^{j}\Phi(x_\ell)$. We could as well replace that standard construction of Brownian motion at $t_j = jT/d$ by Brownian motion sampled by the principal components construction at those time points. Figure 17.6 shows some results using Sobol' points with a nested uniform scramble in $[0,1]^{16}$ to evaluate this option. It is based on 30 replicates of up $n = 2^{12}$ points. We do 30 replicates here because we can then get an indication of how the variances differ between standard and principal components constructions. By $n = 2^{12}$, the standard construction has a standard deviation about ten times as large as the principal components construction has. That corresponds to a variance ratio of about 100 in favor of principal compnents. The plain Monte Carlo variance of this integrand is the same under either method of sampling Brownian motion. It was estimated from $n = 2^{20}$ IID geometric Brownian motion paths. The dotted line in the right panel of Figure 17.6 gives the estimated standard deviation for the average n plain Monte Carlo samples.

In Figure 17.7, we repeat the problem, but this time taking $d = 250$ time steps. We then need a scrambled Sobol' sequence in $[0,1]^{250}$. The direction numbers of Joe and Kuo (2008) were used to construct Sobol' points that were given a nested uniform scramble. Once again RQMC outperforms MC and the principal components construction works better than the standard one.

Comparing Figures 17.6 and 17.7 shows that an option at 250 time points is much less valuable than one at 16 time points. An average over 16 times points has greater variance than one over 250 times points. When by chance the average is unusually far above K, the holder benefits. There is no compensating cost to the holder when the average is far below K. Therefore high variance in $(1/d)\sum_{j=1}^{d} S(Tj/d; \boldsymbol{x})$ is beneficial to the option holder, and the option value decreases as d increases.

17.9 Padding, hybrids and supercube sampling

It becomes harder to apply digital nets as the dimension d increases. Either the quality parameter t must grow, as in Sobol' and NX nets, or the base b must grow, as in Faure nets. Similarly, as the dimension increases, the quality of a rank 1 lattice can decrease but not increase.

Here we look at ways to use a high quality RQMC method in s dimensions on a problem that has $d > s$ dimensions, or even $d \gg s$ dimensions. We suppose as usual that the function f is defined on $[0,1]^d$, that we seek $\mu = \int f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$,

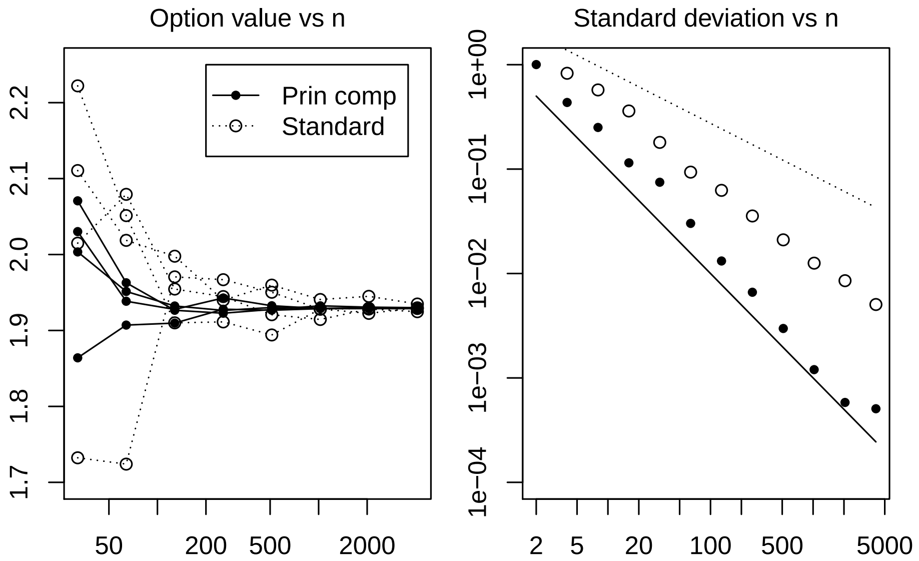

Figure 17.6: The left panel shows 4 RQMC estimates of the option value from (17.17) versus the number n of Sobol' points used. The standard construction is shown with open circles connected by dotted lines. The principal components construction is shown with solid circles and lines. The right panel plots an estimated standard deviation versus n based on $R = 30$ replicates. The dashed reference line is parallel to $n^{-1/2}$ the solid is parallel to n^{-1}.

and that $\sigma^2 = \int (f(\boldsymbol{x}) - \mu)^2 \, \mathrm{d}\boldsymbol{x} < \infty$.

The methods can succeed when $f(\boldsymbol{x})$ depends very strongly on s of the components of $\boldsymbol{x}$ and only weakly on the other $d - s$ components. Those other components get sampled by some lower quality method. We assume that f is defined, using subject matter knowledge, in such a way that the importance of x_j is generally thought to decrease as j increases. In §17.7 we discuss techniques for increasing the importance of the leading components of $\boldsymbol{x}$.

In this setting we may combine RQMC points $\widetilde{\boldsymbol{x}}_1, \dots, \widetilde{\boldsymbol{x}}_n \in [0,1]^s$ for $s < d$ with some kind of filler method on the other $d - s$ dimensions. For example, we could take

$$x_{ij} = \begin{cases} \widetilde{x}_{ij}, & j \leqslant s \\ u_{ij}, & s < j \leqslant d, \end{cases} \tag{17.18}$$

where $u_{ij} \sim \mathbf{U}(0,1)$ are independent of each other and of all the $\widetilde{x}_{ij}$.

The method (17.18) is called ***padding***. It produces ***hybrid points*** $\boldsymbol{x}_i$. Each individual point $\boldsymbol{x}_i \sim \mathbf{U}(0,1)^d$ and so $\hat{\mu} = (1/n)\sum_{i=1}^n f(\boldsymbol{x}_i)$ is unbiased for $\mu = \int_{[0,1]^d} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$. We could also form hybrids using ordinary QMC points $\widetilde{\boldsymbol{x}}_i$ instead of RQMC points $\widetilde{\boldsymbol{x}}_i$ or replace U_{ij} by 1/2. Those combinations are

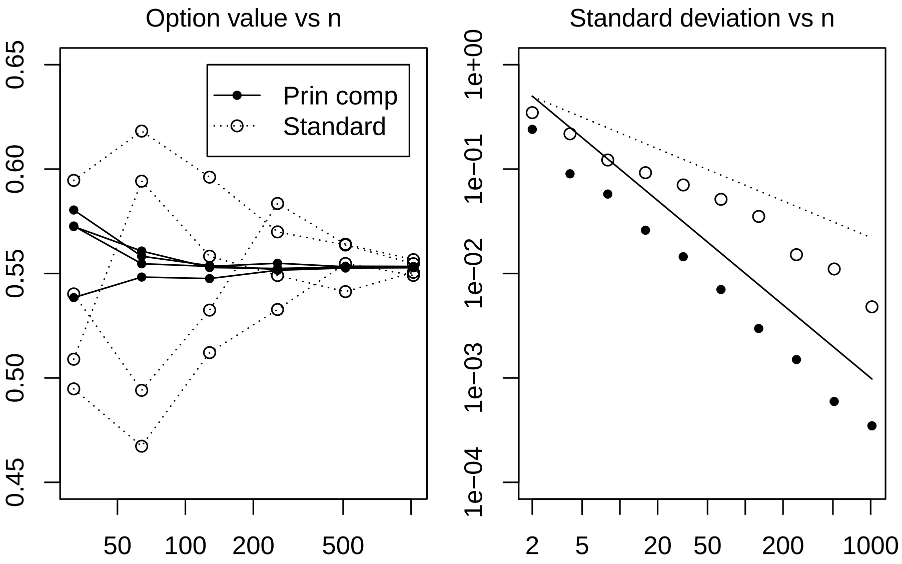

Figure 17.7: This is like Figure 17.6 except that the option is averaged over 250 points instead of 16. The computations are based on a 250 dimensional scrambled Sobol' sequence.

harder to study than (17.18) because they merge deterministic and random components, and they do not give an unbiased estimate of μ.

It is natural to try to replace the plain Monte Carlo portion u_{ij} by points with better equidistribution properties. One simple improvement is to replace the IID components by a Latin hypercube sample. That is

$$x_{ij} = \begin{cases} \widetilde{x}_{ij}, & j \leqslant s \\ \dfrac{\pi_j(i) - U_{ij}}{n}, & s < j \leqslant d, \end{cases} \tag{17.19}$$

where π_j are uniform random permutations of $\{1, \dots, n\}$, independent of the u_{ij} and the $\widetilde{x}_{ij}$ and each other. The resulting points $\boldsymbol{x}_i$ will now be stratified in all d univariate projections, under the very reasonable assumption that we have chosen RQMC points $\widetilde{\boldsymbol{x}}_i$ with good univariate projections.

The next idea we consider is to replace the MC points by one or more other sets of RQMC points. If we have an s-dimensional QMC rule and we want points in dimension $d = ks$ then it is tempting to use k independent scrambles of (t, m, s)-net points $\boldsymbol{a}_i$ with the j'th scramble producing components $(j-1)s+1$ through js of $\boldsymbol{x}_i$. Unfortunately, multiple scrambles of the same underlying point set have a severe flaw that is illustrated in Figure 17.8.

To understand the problem with placing multiple scrambles of the same QMC points side by side, we can look at components 1 and $s+1$ of $\boldsymbol{x}_i$. Let

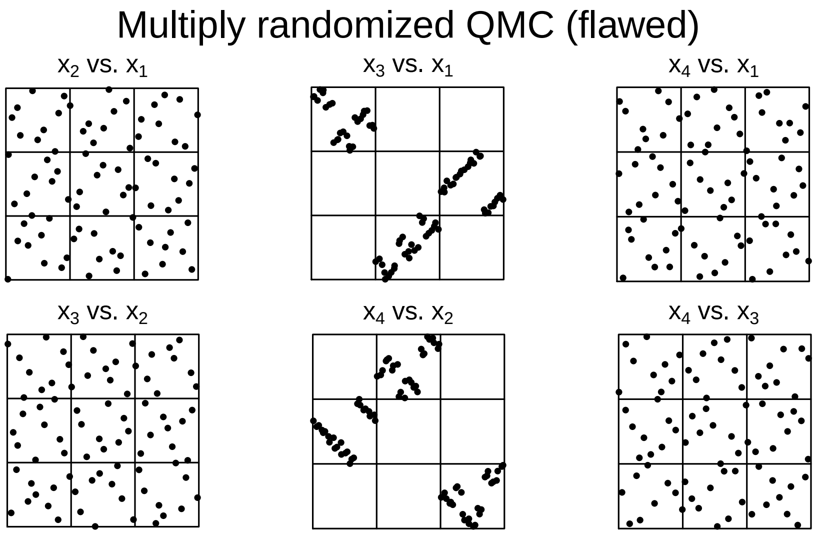

Figure 17.8: This figure shows pairwise scatterplots of 81 points in $\boldsymbol{x}_i \in [0,1]^4$, with horizontal and vertical reference lines at 1/3 and 2/3. Components 1 and 2 are a scrambled $(0,4,2)$-net in base 3. Components 3 and 4 are an independent scramble of the same net. The joint behavior of (x_1, x_3) is flawed because they are scrambles of the same points. The same holds for (x_2, x_4). Latin supercube sampling repairs the flaw.

$a_{i1} = 0.a_{i11}a_{i12}a_{i13}\cdots$ in base b, and similarly

$$x_{i1} = 0.x_{i11}x_{i12}x_{i13}\cdots, \quad \text{and}$$
$$x_{i,s+1} = 0.x_{i,s+1,1}x_{i,s+1,2}x_{i,s+1,3}\cdots.$$

Then $x_{i11} = \pi^1_{1\bullet 1}(a_{i11})$ and $x_{i,s+1,1} = \pi^2_{1\bullet 1}(a_{i11})$ where $\pi^g_{1\bullet 1}$ is the permutation applied to the first digit of the first component of $\boldsymbol{a}_i$ in group $g = 1,\dots,k$. Consider all of the points x_{i11} that lie in the interval $[\ell/b, (\ell+1)/b)$ for some $\ell \in \{0,1,\dots,b-1\}$. That is, their first digit is ℓ. All such points have the same value for a_{i11}, namely $a_{i11} = (\pi^1_{1\bullet 1})^{-1}(\ell)$. Therefore they all have the same value $\pi^2_{1\bullet 1}((\pi^1_{1\bullet 1})^{-1}(\ell))$ for $x_{i,s+1,1}$. It follows that there are b squares, each of area b^{-2} in $[0,1]^2$ whose union contains all of the n points $(x_{i1}, x_{i,s+1})$. In Figure 17.8 we see that all 81 points (x_{i1}, x_{i3}) lie within 3 squares having total area 1/9.

By considering the second digit of $\boldsymbol{a}_i$ we find that all of the points lie within b^2 squares with side length b^{-2} and total area b^{-4} and from the r'th digit they lie inside the union of b^r squares of total area b^{-2r}. Even if we could apply the scramble to an entire infinite (t,s)-sequence there would still be a set of b^r small squares of total area b^{-2r} that contained the entire infinite sequence $(x_{i1}, x_{i,s+1})$ for $i \geqslant 1$. As a result, we should not expect $\hat{\mu}$ to converge to μ as $n \to \infty$ when we use multiply randomized QMC points as described above.

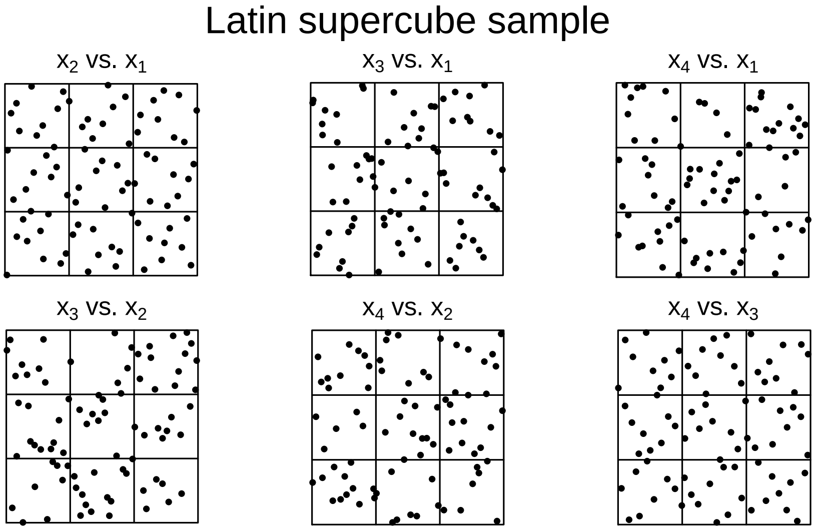

Figure 17.9: This figure shows Latin supercube sampling of the data from Figure 17.8. The run order of each block of points has been randomized. The projections (x_j, x_ℓ) for $j \in \{1,2\}$ and $\ell \in \{3,4\}$ are now comparable to Latin hypercube sample projections.

It is not just scrambling methods that have this flaw. Cranley-Patterson rotations have a version of it. Suppose that $x_{i1} = a_{i1} + U_1 \bmod 1$ and that $x_{i,s+1} = a_{i1} + U_2 \bmod 1$ where U_1 and U_2 are independent $\mathbf{U}(0,1)$ random variables. If $U_2 \geqslant U_1$, then $x_{i,s+1} - x_{i1} \in \{U_2 - U_1, U_2 - U_1 - 1\}$ holds for all $i = 1, \dots, n$. If instead $U_2 < U_1$, then $x_{i,s+1} - x_{i1} \in \{U_2 - U_1, U_2 - U_1 + 1\}$ for all $i = 1, \dots, n$. Either way, the points $(x_{i1}, x_{i,s+1})$ all lie on one line (with wraparound) in the unit square.

We can avoid such extremely bad projections by using Latin supercube sampling, described next. For $j \in 1, \dots, k$ let $\widetilde{\boldsymbol{x}}_1^{(j)}, \dots, \widetilde{\boldsymbol{x}}_n^{(j)} \in [0,1]^{s_j}$ where $s_j \geqslant 1$ and $\sum_{j=1}^k s_j = d$. A ***Latin supercube sample*** has points

$$\boldsymbol{x}_i = (\widetilde{\boldsymbol{x}}_{\pi_1(i)}^{(1)}, \widetilde{\boldsymbol{x}}_{\pi_2(i)}^{(2)}, \cdots \widetilde{\boldsymbol{x}}_{\pi_k(i)}^{(k)}) \in [0,1]^d, \quad i = 1, \dots, n,$$

where $\pi_1, \dots, \pi_k$ are independent uniform random permutations of $\{1, \dots, n\}$. Ordinarily $\widetilde{\boldsymbol{x}}_1^{(j)}, \dots, \widetilde{\boldsymbol{x}}_n^{(j)} \in [0,1]^{s_j}$ comprise an RQMC rule for each $j = 1, \dots, k$ and the permutations $\pi_1, \dots, \pi_k$ are also independent of any randomizations in these RQMC rules. Latin hypercube sampling §10.3 is a special case where all of the $s_j = 1$ and the points $\widetilde{\boldsymbol{x}}_1^{(j)}, \dots, \widetilde{\boldsymbol{x}}_n^{(j)}$ comprise a midpoint rule (for centered LHS) or a stratified sample of $[0,1]$ (for unbiased LHS).

Figure 17.9 shows Latin supercube sampling applied to the points displayed in Figure 17.8. The projections of variable subsets $\{1,2\}$ and $\{3,4\}$ are the same as in Figure 17.8. They are the same points in different order. The projections for subsets $\{1,3\}$ and $\{2,4\}$ are substantially improved. They are not of low discrepancy: they are instead a Latin hypercube sample.

The projections for subsets $\{2,3\}$ and $\{1,4\}$ appear worse for LSS than for multiple RQMC. With multiple RQMC, x_3 is a closely related to x_1 and so the (x_2,x_3) projection inherits the high quality of the (x_1,x_2) projection. While multiple RQMC has some projections that are better than LSS in this case, the flawed projections for multiple RQMC are serious enough to prevent it giving the correct answer as $n\to\infty$.

If we use k RQMC methods to get $\widetilde{\boldsymbol{x}}_i^{(j)}$ then each $\widetilde{\boldsymbol{x}}_{\pi_j(i)}^{(j)}\sim\mathbf{U}[0,1]^{s_j}$ and because they are independent, $\boldsymbol{x}_i\sim\mathbf{U}[0,1]^d$. As a result, LSS yields an unbiased estimate $\hat\mu=(1/n)\sum_{i=1}^n f(\boldsymbol{x}_i)$ of μ. If we use QMC points instead of RQMC points then LSS is biased, though the bias may be very small.

The points $\widetilde{\boldsymbol{x}}_i^{(j)}\in[0,1]^{s_j}$ are from an RQMC rule and so we should expect them to be at least as good, and asymptotically much better, than simple Monte Carlo points in s_j dimensions. To quantify their quality, introduce

$$\varepsilon_n^j(f)=\sup_{\boldsymbol{z}\in[0,1]^{d-s_j}}\biggl|\frac1n\sum_{i=1}^n f\bigl(\boldsymbol{z}{:}\widetilde{\boldsymbol{x}}_i^{(j)}\bigr)-\int_{[0,1]^{s_j}}f(\boldsymbol{z}{:}\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\biggr|,$$

where $\boldsymbol{z}{:}\boldsymbol{x}$ is the point in $[0,1]^d$ formed by using $\boldsymbol{x}\in[0,1]^{s_j}$ for the s_j components corresponding to group j and $\boldsymbol{z}\in[0,1]^{d-s_j}$ for the other $k-1$ groups. Then $\varepsilon_n^j(f)$ is the largest error we would make averaging f over the j'th RQMC points with other components held fixed. Notice that $\varepsilon_n^j(f)$ is a random variable. We will suppose that it is bounded, and that it is $o(n^{-1/2})$, which captures the superiority of the RQMC rules over simple Monte Carlo.

We can analyze LSS via the ANOVA decomposition of f. The error is

$$\hat\mu-\mu=\sum_{|u|>0}\frac1n\sum_{i=1}^n f_u(\boldsymbol{x}_i)\equiv\sum_{|u|>0}\hat\mu_u$$

where the sum is over non-empty $u\subseteq\{1,2,\dots,d\}$. Now let $A_j\subset\{1,\dots,d\}$ be the set of s_j indices corresponding to the input values $\widetilde{\boldsymbol{x}}_i^{(j)}$.

Theorem 17.10. *Let f be a square integrable function on $[0,1]^d$. Suppose that the RQMC rules $\widetilde{\boldsymbol{x}}_i^{(j)}$ satisfy $\varepsilon_n^j(f)=o(n^{-1/2})$ and $\varepsilon_n^j(f_uf_v)=o(n^{-1/2})$ for $u,v\subseteq\{1,\dots,d\}$. Then*

$$\mathrm{Var}(\hat\mu)=\frac1n\biggl(\sigma^2-\sum_{j=1}^k\sum_{u\subseteq A_j}\sigma_u^2+o\Bigl(\frac1{\sqrt n}\Bigr)\biggr)+o\Bigl(\frac1n\Bigr).$$

Proof. This follows from Theorem 2 of Owen (1998). □

Simple Monte Carlo sampling has an error variance that is σ^2/n. By using Latin supercube sampling, with RQMC points, we are able to reduce the asymptotic variance. Specifically, those ANOVA effects σ_u^2 for $u\subseteq A_j$ are asymptotically removed from the variance. ANOVA effects σ_u^2 with $u \cap A_j \neq \varnothing$ and $u \cap A_{j'} \neq \varnothing$ for $j \neq j'$ are handled no better or no worse than under simple Monte Carlo.

While LSS can reduce the constant in the variance of $\hat{\mu}$, it does not improve the rate in n. To get a large reduction in the constant, we would need to arrange for the groups of inputs randomized together to contain the bulk of the interactions in f.

Latin hypercube sampling corresponds to Latin supercube sampling with singleton sets $A_j = \{j\}$, for $j = 1, \dots, d$. In LHS the asymptotic variance comes from all the interactions in f, that is σ_u^2 for $|u| \geqslant 2$.

Theorem 17.10 shows how best to take advantage of Latin supercube sampling. Where subject matter knowledge and computational convenience allow, we should arrange for the variables with the strongest interactions to be grouped together within the same subset A_j. The emphasis should be on grouping together the variables with strong low order interactions, because the asymptotic advantage of RQMC for the higher order interactions may require larger n to take hold.

17.10 Randomized Halton sequences

Halton sequences have mostly been left behind by progress in lattices, digital nets and polynomial lattice rules. They may still have a role to play, and they do have very good discrepancy bounds.

There have been several proposals to randomize them, mostly by scrambling their digits. Let the unscrambled Halton points be $\boldsymbol{a}_i = (a_1, \dots, a_d) \in [0,1]^d$ with

$$a_{ij} = \sum_{k=0}^{K_{ij}} a_{ijk} p_j^{-k-1}, \quad 0 \leqslant a_{ijk} < p_j,$$

where p_j is the j'th prime. The sum is finite, with K_{ij} just large enough that $p_j^{K_{ij}+1} \geqslant i$. Perhaps the most straightforward way to randomize these points is to take

$$x_{ij} = \sum_{k=0}^{K_{ij}} \pi_j(a_{ijk}) p_j^{-k-1} + p_j^{K_{ij}-2} U_{ij}, \quad 0 \leqslant a_{ijk} < p_j, \tag{17.20}$$

where π_j is a uniform random permutation of $(0, 1, \dots, p_j - 1)$, $U_{ij} \sim \mathbf{U}(0,1)$ and all the π_j and U_{ij} are independent. That is, the same permutation is used for all of the digits in the j'th variable. There have been many efforts to find good deterministic permutations π_j. Some of those are in §15.5. However, making the permutations random gives unbiased estimates suitable for replication.

Another very innovative randomization due to Wang and Hickernell (2000) uses the von Neumann-Kakutani transformation in base p_j. Figure 17.10 shows this transformation in bases $p_1 = 2$ and $p_2 = 3$. For the radical inverse function $\phi_b(i)$ in base $b \geqslant 2$, the value $\phi_b(i+1)$ is a deterministic function of $\phi_b(i)$.

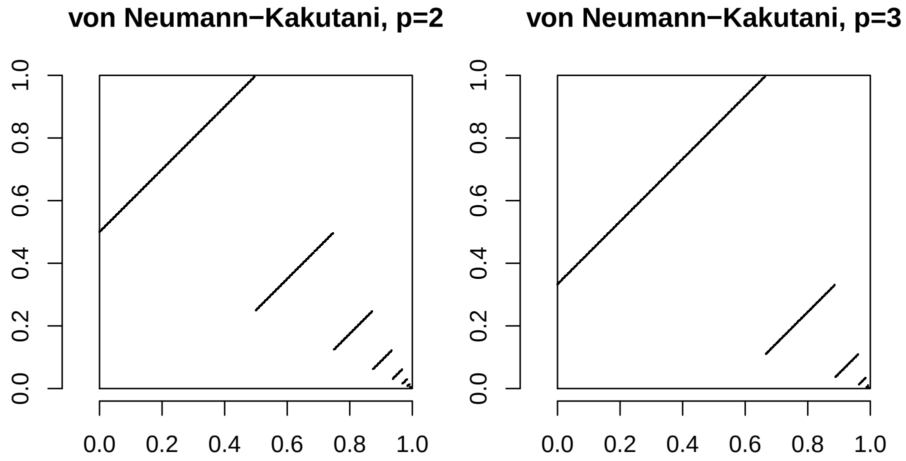

Figure 17.10: These are von Neumann-Kakutani transformations from $[0,1]$ to $[0,1]$. The left panel plots $\phi_2(i+1)$ versus $\phi_2(i)$ for integers $i \geqslant 0$. The right panel shows $\phi_3(i+1)$ versus $\phi_3(i)$.

Write it $\phi_b(i+1) = \mathrm{vnk}_b(\phi_b(i))$. They choose their first point $\boldsymbol{x}_1 \sim \mathbf{U}(0,1)^d$. Then, for $i \geqslant 2$ they take $x_{i+1,j} = \mathrm{vnk}_{p_j}(x_{ij})$. Each $\boldsymbol{x}_i \sim \mathbf{U}(0,1)^d$. One way to implement it, is to solve for N_j such that $\phi_{p_j}(N_j) \doteq x_{1j}$ for $j = 1,\dots,d$ and then take $x_{ij} = \phi_{p_j}(N_j + i - 1)$. If x_{1j} would really be random then N_j might not be bounded, but rounding x_{ij} to machine precision will give a finite N_j. This random start Halton can produce unwanted stripes (Chi et al., 2005).

Matoušek (1998) considers nested uniform scrambling (component j scrambled in base p_j, the j'th prime). He numerically evaluates mean squared discrepancy. Scrambling does not offer a consistent advantage or disadvantage for the dimensions and samples sizes he investigates.

Ökten et al. (2012) make a study of scrambled Halton sequences. They compare mean square discrepancy (Warnock's formula) at $n = 100$ as well as accuracy for larger n on a standard test integrand $f(\boldsymbol{x}) = \prod_{j=1}^d (|4x_j - 2| + a_j)/(1+a_j)$ for several different vectors $\boldsymbol{a} = (a_1,\dots,a_d)$ and dimensions d. One of their conclusions is that the simple scramble in (17.20) is hard to beat. They find that it gives results that are at least competitive with and perhaps better than purpose built deterministic scrambles.

17.11 RQMC and variance reduction

Randomized quasi-Monte Carlo sampling is a kind of variance reduction method. It can be combined with other variance reduction methods, such as control variates, antithetic sampling, and importance sampling.

Consider a control variate $h(\boldsymbol{x}) \in \mathbb{R}^J$ for which $\int h(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \theta$ is known.

Using RQMC points $\boldsymbol{x}_i$, we may construct the unbiased estimate

$$\hat{\mu}_\beta = \frac{1}{n}\sum_{i=1}^n \bigl(f(\boldsymbol{x}_i) - \beta^{\mathsf{T}} h(\boldsymbol{x}_i)\bigr) + \beta^{\mathsf{T}}\theta$$

of $\mu = \int f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$.

The optimal coefficient is

$$\beta_n^{\mathrm{opt}} = \mathrm{Cov}_{\mathrm{RQMC}}(\bar{h},\bar{h})^{-1}\mathrm{Cov}_{\mathrm{RQMC}}(\bar{h},\bar{f}) \quad \text{where}$$
$$\bar{h} = \frac{1}{n}\sum_{i=1}^n h(\boldsymbol{x}_i) \quad \text{and} \quad \bar{f} = \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i),$$

and the covariances are with respect to the randomizations in the RQMC points $\boldsymbol{x}_i$. The optimal value of β can be arbitrarily different from the one in plain Monte Carlo and it ordinarily changes with n. The usual regression formula for estimating β estimates the optimal value for MC, not for RQMC. We can estimate $\hat{\beta}_n^{\mathrm{opt}}$ for RQMC by using independent replicates of the RQMC points as described in Hickernell et al. (2005).

A common reason for the difference is as follows. The variance of $\hat{\mu}$ under ordinary Monte Carlo sampling may be dominated by low order ANOVA components or low order terms in a Fourier, wavelet or Walsh expansion (as in §16.4 and §15.13). A good control variate is then one that correlates with those low order components to allow us to remove them as a source of variance. In RQMC sampling, we often get very accurate results for low order terms and then have an error variance dominated by somewhat higher order terms, perhaps the lowest order ones not well handled by the RQMC points. In that case, a good control variate h is one whose higher order components correlate well with those of f.

In a numerical example of Hickernell et al. (2005, Table 4), the variance reduction from using both RQMC and control variates is smaller than the product of their individual variance reduction factors. In that 16 dimensional option valuation problem, an RQMC method reduced variance by a factor of 142. The best of four control variate strategies, all based on the close connection between geometric means and arithmetic means for an Asian call option, reduced variance by a factor of 450. Combining the control variate strategy with RQMC sampling reduced variance by about 1800 fold, better than either method individually, but far short of 142 × 450. A different control variate strategy, which was not the best for MC, yielded a variance reduction of 3600 fold when used with RMQC.

It is possible to construct examples for which a control variate is very useful for RQMC. But in real applications it can be very hard to identify a control variate whose very high order behavior matches that of f. It is extremely rare to find a control variate that is as good as the geometric average option is for a problem of valuing an arithmetic option. Plain Monte Carlo control variates are comparatively easier to find because it is easier to understand and match the coarse low order behavior of f and h.

The combination of RQMC with antithetic sampling faces similar issues. Antithetic sampling yields exact answers for the odd part of f while doubling the variance for the even part of f. Randomized QMC provides very accurate integration for low dimensional coarse parts of the integrand while yielding more like the Monte Carlo rate on high dimensional and high frequency parts of the integrand. As a result, antithetic sampling with RQMC points will be of great benefit if the high order and high frequency parts of f are dominated by their odd parts. A combination of antithetic sampling with RQMC is then ideally suited for f if its high order and high frequency components are approximately odd functions.

Caflisch et al. (1997, §6.1) study a 360 dimensional integrand motivated by mortgage valuation. By comparing the variances of Latin hypercube sampling and plain Monte Carlo they estimate that the integrand has roughly 99.98% of its variance in its additive component. Comparing Monte Carlo with antithetic sampling they estimate that roughly 99.96% of the variance comes from its odd part. The integrand is therefore nearly a sum of odd one dimensional functions. The higher dimensional components were not overwhelmingly odd functions, because the combination of antithetic sampling with Latin hypercube sampling was not much more effective than Latin hypercube sampling on its own.

Somewhat better results are available with local antithetic sampling. If f is nearly linear within each small rectangular patch, then local antithetic versions of stratified sampling from §10.2 reduce the variance from $O(n^{-1})$ to $O(n^{-1+2/d})$ in d dimensional problems. A similar reduction is available for scrambled net quadrature. Given sufficient smoothness, a locally antithetic version of randomized nets yields mean square errors of $O(n^{-3-2/d+\epsilon})$ in d dimensional problems compared to $O(n^{-3+\epsilon})$ for scrambled nets. See Owen (2008). Much better asymptotic orders are obtainable via higher order nets of §15.12.

17.12 Singular integrands

Many problems involve finding the expectation of an unbounded, that is singular, integrand. For example, Gaussian random variables are unbounded and integrands on $[0,1]^d$ constructed by transforming to a multivariate Gaussian vector may well be unbounded too. That is common in financial valuation problems (Glasserman, 2004). Kollig and Keller (2006) describe some singular integrands in computer graphics. The function f may diverge to $\pm\infty$ in places and yet $\mu = \int f(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ is well defined so long as $\int |f(\boldsymbol{x})|p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} < \infty$. These problems then have integrable singularities. If also $\int f(\boldsymbol{x})^2 p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} < \infty$, then Monte Carlo sampling of $\boldsymbol{x}_i \sim p$ will lead to an estimate of μ with root mean squared error $O(1/\sqrt{n})$.

Plain QMC is not designed for such problems. If f defined on $[0,1]^d$ is unbounded then it has infinite variation in the sense of Hardy and Krause. Averages of f over a low discrepancy point set could fail to converge to $\mu = \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$. Randomized QMC can work well on singular integrands. If

$\int_{[0,1]^d} f(\boldsymbol{x})^2 \,\mathrm{d}\boldsymbol{x} < \infty$ then sampling along digital sequences, such as Sobol's or Faure's, with a nested uniform scramble attains $\mathrm{Var}(\hat{\mu}) = o(1/n)$. This rate holds whether or not we know where the singularity or singularities are.

Very often the singularities arise on the boundary of $[0,1]^d$ and then we can study the problem in more detail. We look first at how QMC can work with such singularities. Despite the potential noncovergence of QMC, Ilya Sobol' noticed by the early 1970s that his colleagues were using QMC on singular integrands without any apparent problems, and found an explanation: sometimes QMC points manage to avoid the area of the singularity.

Suppose that there is a region $K \subset [0,1]^d$, such that $|f|$ is bounded on K and all of the QMC points $\boldsymbol{x}_i$ are inside K. Next, let $\tilde{f}$ be a function on $[0,1]^d$ with $V_{\mathrm{HK}}(\tilde{f}) < \infty$ that satisfies $\tilde{f}(\boldsymbol{x}) = f(\boldsymbol{x})$ whenever $\boldsymbol{x} \in K$. Then

$$\begin{aligned}
|\hat{\mu} - \mu| &= \left| \frac{1}{n}\sum_{i=1}^n \tilde{f}(\boldsymbol{x}_i) + (f(\boldsymbol{x}_i) - \tilde{f}(\boldsymbol{x}_i)) - \int_{[0,1]^d} \tilde{f}(\boldsymbol{x}) + (f(\boldsymbol{x}) - \tilde{f}(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} \right| \\
&= \left| \frac{1}{n}\sum_{i=1}^n \tilde{f}(\boldsymbol{x}_i) - \int_{[0,1]^d} \tilde{f}(\boldsymbol{x}) + (f(\boldsymbol{x}) - \tilde{f}(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} \right| \\
&\leqslant \left| \frac{1}{n}\sum_{i=1}^n \tilde{f}(\boldsymbol{x}_i) - \int_{[0,1]^d} \tilde{f}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \right| + \int_{[0,1]^d} |\tilde{f}(\boldsymbol{x}) - f(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{x} \\
&\leqslant D_n^*(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) V_{\mathrm{HK}}(\tilde{f}) + \int_{K^c} |\tilde{f}(\boldsymbol{x}) - f(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{x}.
\end{aligned}$$

Now, given a sequence of regions K_n containing $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ and a corresponding sequence of extensions $\tilde{f}_n$, QMC will converge to the right answer if

$$\lim_{n\to\infty} D_n^*(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) V_{\mathrm{HK}}(\tilde{f}_n) + \int_{K_n^c} |\tilde{f}_n(\boldsymbol{x}) - f(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{x} = 0.$$

Sometimes the integral has a singularity at the origin or along the 'lower boundary' of $[0,1]^d$. Points $\boldsymbol{x}_i$ can avoid the singularity by being confined to

$$\begin{aligned}
K_{\min}^{\mathrm{orig}}(\epsilon) &= \{\boldsymbol{x} \in [0,1]^d \mid \min_{1\leqslant j\leqslant d} x_j \geqslant \epsilon\}, \\
K_{\mathrm{prod}}^{\mathrm{orig}}(\epsilon) &= \{\boldsymbol{x} \in [0,1]^d \mid \prod_{1\leqslant j\leqslant d} x_j \geqslant \epsilon\}, \quad \text{or} \\
K_{\max}^{\mathrm{orig}}(\epsilon) &= \{\boldsymbol{x} \in [0,1]^d \mid \max_{1\leqslant j\leqslant d} x_j \geqslant \epsilon\}.
\end{aligned} \tag{17.21}$$

For $d = 1$, these all reduce to $[\epsilon, 1]$.

For $d > 1$, we may have to rearrange our integrand to ensure that the corner containing the singularity is placed at the origin. If the singularity can be at

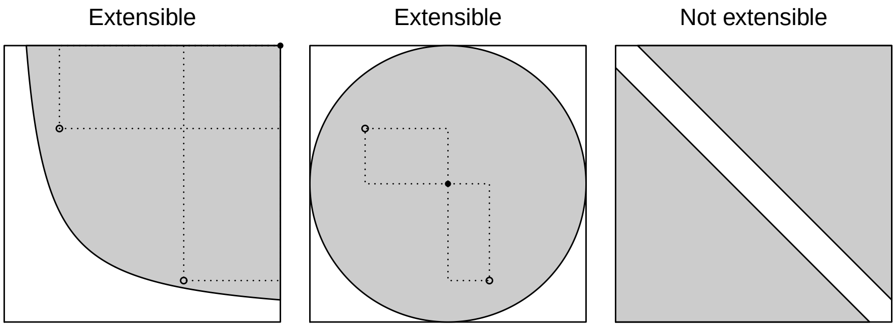

Figure 17.11: The first panel shows a Sobol'-extensible region above a hyperbola. The second panel shows a Sobol'-extensible circle. Their anchors $\boldsymbol{c}$ are marked with a solid point and two bounding boxes are drawn. The third panel shows a non-extensible region that omits a strip along a diagonal.

any of the corners or along any of the boundaries then we may instead use

$$\begin{aligned}
K^{\mathrm{corn}}_{\min}(\epsilon) &= \{\boldsymbol{x}\in[0,1]^d \mid \min_{1\leqslant j\leqslant d}\min(x_j,1-x_j)\geqslant\epsilon\},\\
K^{\mathrm{corn}}_{\mathrm{prod}}(\epsilon) &= \{\boldsymbol{x}\in[0,1]^d \mid \prod_{1\leqslant j\leqslant d}\min(x_j,1-x_j)\geqslant\epsilon\},\quad\text{or}\\
K^{\mathrm{corn}}_{\max}(\epsilon) &= \{\boldsymbol{x}\in[0,1]^d \mid \max_{1\leqslant j\leqslant d}\min(x_j,1-x_j)\geqslant\epsilon\}.
\end{aligned}\tag{17.22}$$

For $d=1$, these all reduce to $[\epsilon,1-\epsilon]$.

Sobol' (1973a) found a way to extend f defined on certain regions $K\subset[0,1]^d$ to $\tilde f$ on $[0,1]^d$ keeping $V_{\mathrm{HK}}(\tilde f)$ under some control. The set $K\subset[0,1]^d$ is ***Sobol' extensible*** if there is some anchor point $\boldsymbol{c}\in[0,1]^d$ such that the hyper-rectangle

$$R(\boldsymbol{x})\equiv\prod_{j=1}^d[\min(x_j,c_j),\max(x_j,c_j)]$$

satisfies $R(\boldsymbol{x})\subset K$ for all $\boldsymbol{x}\in K$. The region $R(\boldsymbol{x})$ is a rectangular bounding box or rectangular hull of the points $\boldsymbol{x}$ and $\boldsymbol{c}$. Figure 17.11 shows two Sobol'-extensible regions and one other that is not extensible. The regions in (17.21) and (17.22) are Sobol' extensible.

If the partial derivatives of f taken once with respect to each component of $\boldsymbol{x}$ is continuous on K, then Sobol's extension can be made. We illustrate it for $d=2$. For the more general treatment see Basu and Owen (2015). For $\boldsymbol{x}\in K$ we can write

$$f(\boldsymbol{x})=f(\boldsymbol{c})+\int_{c_1}^{x_1}\frac{\partial f(\boldsymbol{x})}{\partial x_1}\,\mathrm{d}x_1+\int_{c_2}^{x_2}\frac{\partial f(\boldsymbol{x})}{\partial x_2}\,\mathrm{d}x_2\pm\int_{R(\boldsymbol{x})}\frac{\partial^2 f(\boldsymbol{x})}{\partial x_1\partial x_2}\,\mathrm{d}\boldsymbol{x}.$$

The sign in the final integral is positive if $x_j > c_j$ for an even number of j, that is for zero or two such j, and is negative otherwise. The two univariate integrals must be interpreted with a similar care on their signs. For instance, if $c_1 > x_1$, then the first one is $-\int_{x_1}^{c_1} \partial f(\boldsymbol{x})/\partial x_1 \,\mathrm{d}x_1$. For $d = 2$, the Sobol' extension of f to $\boldsymbol{x} \notin K$ is

$$\begin{aligned}\tilde{f}(\boldsymbol{x}) = f(\boldsymbol{c}) &+ \int_{c_1}^{x_1} \mathbb{1}\{(z_1, x_2) \in K\} \frac{\partial f((z_1, x_2))}{\partial z_1} \,\mathrm{d}z_1 \\ &+ \int_{c_2}^{x_2} \mathbb{1}\{(x_1, z_2) \in K\} \frac{\partial f((x_1, z_2))}{\partial z_2} \,\mathrm{d}z_2 \pm \int_{R(\boldsymbol{x})} \mathbb{1}\{\boldsymbol{z} \in K\} \frac{\partial^2 f(\boldsymbol{z})}{\partial z_1 \partial z_2} \,\mathrm{d}\boldsymbol{z}.\end{aligned}$$

For $\boldsymbol{x} \in K$, the fundamental theorem of calculus gives $\tilde{f}(\boldsymbol{x}) = f(\boldsymbol{x})$. Points $\boldsymbol{x} \notin K$ don't add any variation to $\tilde{f}$ beyond what it must have to match f on K. For instance, the Vitali variation of $\tilde{f}$ is $\int_K |\partial^d f(\boldsymbol{x}) / \prod_{j=1}^d \partial x_j| \,\mathrm{d}\boldsymbol{x}$.

Sobol' (1973a) showed that some of his sequences avoid a hyperbolic region, $K_{\mathrm{prod}}^{\mathrm{orig}}$ around the origin. An unfortunate typo in that paper makes it look he is considering $K_{\mathrm{min}}^{\mathrm{orig}}$. For $d = 1$, he finds that the van der Corput sequence (not including $x = 0$) integrates x^{-A} with error $O(n^{A-1}\log(n))$ for $A < 1$. For larger d, products of negative powers of x_j are integrated correctly by Sobol' sequences as $n \to \infty$.

Halton points are quite good at avoiding the origin, assuming that they don't start with $\phi_{p_j}(0)$. For x_{ij} to come close to the origin, i must be a multiple of a power of p_j, the j'th prime. For $\boldsymbol{x}_i$ to come close to 0, i must be a multiple of powers of all the primes $p_1, \dots, p_d$. There are details in Theorem 3.1 of Owen (2006a).

Uniform random points are good at avoiding small regions containing integrable singularities. If they were not, then the law of large numbers could fail. The next Lemma shows that for RQMC points there can be only finitely many n for which one or more of the $\boldsymbol{x}_i$ was within $K_{\mathrm{prod}}^{\mathrm{orig}}(Cn^{-r})$ when $r > 1$.

Lemma 17.1. *For $i = 1, \dots, n$, let $\boldsymbol{x}_i \sim \mathbf{U}[0,1]^d$. Then for $C > 0$ and $r > 1$,*

$$\mathbb{P}\Bigl(\min_{1 \leqslant i \leqslant n} \prod_{j=1}^{d} x_{ij} \leqslant Cn^{-r} \quad \text{infinitely often}\Bigr) = 0.$$

Proof. This is part of Lemma 4.1 of Owen (2006a). □

Lemma 17.1 follows from the Borel-Cantelli theorem and it does not require that $\boldsymbol{x}_i$ be independent of each other. The same holds for all 2^d corners of $[0,1]^d$ and so RQMC points also remain within $K_{\mathrm{prod}}^{\mathrm{corn}}(Cn^{-r})$ all but finitely often. We can get rates for some RQMC points, using just that avoidance behavior, their discrepancy, and assumptions about the integrand.

Definition 17.3. The function f on $[0,1]^d$ has corner singularities ***no worse than*** $\prod_{j=1}^d x_j^{-A_j}$ if

$$\left|\frac{\partial^{|u|}f(\boldsymbol{x})}{\prod_{j\in u}\partial x_j}\right| \leqslant B\prod_{j=1}^d \min(x_j,1-x_j)^{-A_j-\mathbb{1}\{j\in u\}} \tag{17.23}$$

holds for all $u\subseteq\{1,2,\dots,d\}$, some $A_j\in(0,1)$ and some $B<\infty$.

We need $A_j<1$ because other wise f might not be integrable. We assume that $A_j>0$ because otherwise f might not be singular.

Theorem 17.11. *Let* $\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\sim\mathbf{U}[0,1]^d$ *with* $\mathbb{E}(D_n^*(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n))=O(n^{-1+\epsilon})$ *for all* $\epsilon>0$. *If* f *satisfies* (17.23), *then*

$$\mathbb{E}\bigl(|\hat\mu-\mu|\bigr)=O(n^{-1+\epsilon+\max_j A_j}).$$

Proof. This is Theorem 5.7 of Owen (2006a). □

When one or more of the $A_j\geqslant 1/2$ then f does not necessarily have finite variance. The Monte Carlo rate in that case is not usually known, (though perhaps we could find it, see Exercise 17.8). Theorem 17.11 gives a known rate. It is better than the MC rate when $\max_j A_j<1/2$. Some slowly growing singularities derived by inverting the Gaussian CDF can satisfy (17.23) for any $A_j>0$. Then the RQMC expected error can be $O(n^{-1+\epsilon})$ for any $\epsilon>0$.

Isolated point singularities, even at unknown locations, can be handled by RQMC if they are not too severe. Owen (2006b) considers singularities at unknown points $\boldsymbol{z}\in[0,1]^d$ that are 'no worse' than $\|\boldsymbol{x}-\boldsymbol{z}\|_p^{-A}$ for $1<p<\infty$. For such integrands, RQMC estimates $\int_{[0,1]^d}f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ with

$$\mathbb{E}(|\hat\mu-\mu|)=O(n^{(-1+\epsilon)(d-A)/d}).$$

The proof uses Sobol' extensions from sets $K_u=\{\boldsymbol{x}\in[0,1]^d\mid\|\boldsymbol{x}-\boldsymbol{z}\|_p\geqslant\epsilon\}\cap O_u$ for all 2^d orthants $O_u\equiv\{\boldsymbol{x}\in[0,1]^d\mid x_j>z_j\iff j\in u\}$ defined by $u\subseteq\{1,2,\dots,d\}$.

Very little is known about (R)QMC for singularities along arbitrary manifolds. For instance, for a singularity along $\{(t,1-t)\mid 0\leqslant t\leqslant 1\}\subset[0,1]^2$, Figure 17.11 shows a region that we might wish to extend f from. That region is not Sobol'-extensible so some other construction of $\tilde f$ would be necessary. Basu and Owen (2018) consider some approaches to this problem.

Hartinger et al. (2005) study corner avoidance properties of QMC points. Hartinger and Kainhofer (2006) consider QMC integration of $f(\boldsymbol{x})p(\boldsymbol{x})$ for integrands f with singularities and non-uniform probability density functions p.

17.13 (R)QMC for MCMC

Here we consider what happens if we try to use QMC methods in Markov chain Monte Carlo (MCMC). Then we extend it to RQMC.

QMC and MCMC are in some ways opposites. QMC is done with n points in d dimensions, with $n \gg d$, possibly $d = 1$, and studied as $n \to \infty$. The inputs can be arranged in an $n \times d$ matrix with a row per sample and a column per variable. In Bayesian applications, MCMC is done with some large number n of generated points and R replicated chains, perhaps with $R = 1$. For $R = 1$, MCMC is estimating an integral by just one average over n data points. If we picture the inputs to MCMC as one row per sample and one column per variable used they form an $R \times ns$ matrix where s is the average number of uniform random variables needed to advance the Markov chain one step. Then because $R \ll ns$, the input shape for MCMC looks like the transpose of what we use for QMC.

The justifications for QMC and MCMC are also different. QMC uses discrepancy of a collection of points. MCMC uses ergodicity of a sequence.

The first thing to realize is that the combination, done badly, would fail dramatically. Caflisch and Moskowitz (1995) described replacing the stream of random numbers in MCMC by a van der Corput sequence. For random walk Metropolis, a simple proposal like $x_i \to x_i + \Phi^{-1}(u_{2i-1})$ could be followed by an acceptance-rejection decision based on whether u_{2i} is below the Hastings ratio. Because large u_{2i-1} are followed by small u_{2i} and vice versa in the van der Corput sequence, we could find that positive proposed changes are usually accepted while negative ones are usually rejected, producing a random walk that drifts off to infinity instead of being stationary.

If we are to replace $u_1, u_2, \dots$ from a random number generator by a QMC sequence, then it is clear that having $D_n^*(u_1, \dots, u_n) \to 0$ is not enough, because that holds for van der Corput. To fix the that flaw with the van der Corput sequence we would also want $D_n^*(\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n) \to 0$ for $\boldsymbol{v}_i = (u_i, u_{i+1})$. More generally, we want

$$D_n^*(\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n) \to 0, \quad \text{for} \quad \boldsymbol{v}_i = (u_i, u_{i+1}, \dots, u_{i+k-1}) \in [0,1]^k \tag{17.24}$$

to hold for all $k \geqslant 1$. An infinite sequence $u_1, u_2, \dots$ that satisfies (17.24) is ***completely uniformly distributed***, or CUD. Definition (17.24) uses overlapping k-tuples. Chentsov (1967) shows that we can also define CUD via non-overlapping k-tuples, with $\boldsymbol{v}_i = (u_{k(i-1)+1}, \dots, u_{ki})$.

One of the definitions of a random sequence in Knuth (1998) is that it be CUD. Some of the criteria for random number generators in Chapter 3 involve the full period of the RNG having uniformly distributed k-tuples, though that is only possible for k small compared to the period of the generator. The idea behind putting QMC into MCMC is to use the entire period of an RNG. Of course, one would then need to choose a small RNG.

For finite n, CUD sequences are constructed using similar algorithms to those used for RNGs. Tribble (2007) uses some Korobov points, which are small congruential RNGs, as well some small linear feedback shift register (LFSR) generators. Chen et al. (2012) present some LFSRs on 2^m points for each integer m from 10 to 30.

For RQMC, we would use a random sequence u_i instead of a deterministic one. An infinite random sequence u_i is ***weakly CUD*** or WCUD if

$$\mathbb{P}(D_n^*(\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n) \to 0) = 1, \quad \text{for} \quad \boldsymbol{v}_i = (u_i, u_{i+1}, \dots, u_{i+k-1}) \tag{17.25}$$

holds for all $k \geqslant 1$. Tribble and Owen (2008) give some constructions of WCUD sequences. A Cranley-Patterson rotation of a CUD sequence is WCUD.

Theoretical understanding of MCMC driven by (W)CUD points is more complicated than when the driving sequence of u_i has IID elements. The k-tuples $\boldsymbol{v}_i$ have some negative dependencies which means that the output of the simulation is not Markov, just as RQMC for finite d produces outputs that are not independent. We saw the use of non-Markov simulations for MCMC in adaptive MCMC.

What is known about (R)QMC inside MCMC is that it is consistent, that is, various laws of large numbers have been proved for it. Chentsov (1967) proved one for sampling a Markov chain on a discrete space by inversion. Owen and Tribble (2005) handled discrete Markov chains by Metropolis-Hastings. Chen et al. (2011) considered MCMC for continuous random variables by Metropolis-Hastings and by Gibbs. Empirically, placing QMC within MCMC is often seen to give a better convergence rate, especially for Gibbs sampling which avoids the step discontinuities that Metropolis-Hastings has.

Chen (2011) proves that a better rate is possible, but under much stronger assumptions than those rates have been observed empirically. Chen et al. (2016) introduce a herded Gibbs sampler for problems on Markov random fields. It is deterministic and they show $O(1/n)$ convergence. Schwedes and Calderhead (2018) obtain variance nearly $O(1/n^2)$ using QMC within parallelized MCMC.

17.14 Array-RQMC

Array-RQMC uses RQMC methods to sample a large number n of Markov chains through T time steps each. For full details, see L'Ecuyer et al. (2018). At time $t \geqslant 1$, chain i visits

$$\boldsymbol{x}_{i,t} = \Psi(\boldsymbol{x}_{i-1,t}, \boldsymbol{u}_{i,t}), \quad \boldsymbol{u}_{i,t} \in (0,1)^d,$$

for an update function $\Psi(\cdot, \cdot)$. There is a common starting value $\boldsymbol{x}_{i,0} = \boldsymbol{x}_0$ for all of the chains, and the quantity of interest is

$$\mu = \mathbb{E}\biggl(\sum_{t=1}^{T} c_t(\boldsymbol{x}_{i,t})\biggr), \quad \text{for } \boldsymbol{u}_{i,t} \overset{\text{iid}}{\sim} \mathbf{U}(0,1)^d. \tag{17.26}$$

For instance, $\boldsymbol{x}$ might describe the state of a inventory system or a queue of customers at time t and $c_t(\cdot)$ could be a corresponding cost function, perhaps discounting future costs using an interest rate. Policy changes would then amount to changing Ψ, and we might want to know what the expected cost of a proposed policy is. Puchhammer et al. (2019) report using array-RQMC on a chemical kinetics problem with τ-leaping. In that case only the last state needs to be summarized and so $c_t = 0$ for all $t < T$.

While (17.26) is defined in terms of plain MC sampling, we can instead apply the variance reduction or RQMC methods to improve the quality of an estimate. The most straightforward way to apply RQMC is to use n points $\boldsymbol{v}_i \in (0,1)^{Td}$. The first d components of $\boldsymbol{v}_i$ are then $\boldsymbol{u}_{i,1}$ and more generally $\boldsymbol{v}_i = (\boldsymbol{u}_{i,1}, \boldsymbol{u}_{i,2}, \dots, \boldsymbol{u}_{i,T})$. That is, we take

$$\mu = \mathbb{E}(f(\boldsymbol{v})), \quad \boldsymbol{v} \sim \mathbf{U}(0,1)^{Td}$$

for $\boldsymbol{v} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_T)$ with f incorporating both the costs c_t and the updates Ψ. The problem with this plain approach is that it may require a very high dimensional RQMC point set.

We could instead use Latin supercube sampling (LSS) of §17.9, with T independent reorderings of some d-dimensional RQMC points. The reordering for time t would be used to update all n Markov chains from time $t-1$. We could even use d independent reorderings of a T-dimensional RQMC point set. The j'th reordering for $j = 1, \dots, d$ would then give the j'th components of $\boldsymbol{u}_{i,t}$ for $t = 1, \dots, T$, though that involves more difficult bookkeeping. Unfortunately, LSS only gives the plain MC rate, $O(n^{-1/2})$. If there are important interactions between variables receiving different random reorderings, then LSS only averages those interactions at the MC rate.

What is missing from LSS is a way to have the updates at step t be almost independent of the prior state $\boldsymbol{x}_{i,t-1} \in \mathbb{R}^s$. That prior state captures everything relevant about the prior update variables $\boldsymbol{u}_{i,t'}$ for $t' < t$, and so updating the chains nearly independently of their prior states should be effective.

Array-RQMC has a simple way to fill this weakness in LSS when $s = 1$, that is, when $x_{i,t-1}$ is scalar. It uses low discrepancy points $\boldsymbol{w}_{i,t} = (a_{i,t}, \boldsymbol{u}_{i,t}) \in (0,1)^{d+1}$, for $i = 1, \dots, n$. The first component $a_{i,t}$ is used to decide which simulated Markov chain gets updated by which of the $\boldsymbol{u}_{i,t}$. The k'th largest $x_{i,t-1}$ gets updated by $\boldsymbol{u}_{i(k),t}$ where $a_{i(k),t}$ is the k'th largest of the $a_{i,t}$.

The points $\boldsymbol{w}_{i,t}$ are RQMC points with a different, independent randomization at each time point t. This runs counter to the admonition in §17.9 that such points do not uniformly sample. The situation is not precisely the same. The points $\boldsymbol{u}_{i,t}$ and $\boldsymbol{u}i, t'$ for $t' \neq t$ do not necessarily update the same Markov chain, because of the way the ordering develops. It may be enough for the values $\boldsymbol{x}_{i,t-1}$ at time t to have low discrepancy with respect to the true distribution at that time. Then the specific random inputs that produced them can be forgotten.

When $s > 1$, then it is more challenging to decide which of the Markov chains should be updated with a given input $\boldsymbol{u}_{i,t}$ at time $t-1$. One approach is to take $\boldsymbol{w}_{i,t} \in (0,1)^{s+d}$ and make the match based on some sort of similarity between points $\boldsymbol{x}$ at time $t-1$ and the coordinates of $\boldsymbol{w}_{i,t}$. Because s-dimensional space does not have a natural ordering for $s > 1$, there are many ways to do this. See Puchhammer et al. (2019).

Gerber and Chopin (2015) develop a sequential quasi-Monte Carlo sampler that can be viewed as a form of array-RQMC. We will revisit it in Chapter xxx on sequential Monte Carlo and particle methods. They smoothly transform $\boldsymbol{x}_{i,t-1} \in \mathbb{R}^s$ to $(0,1)^s$ via a logistic function. Then they run a Hilbert space-filling curve through $[0,1]^s$. That is a continuous curve mapping $[0,1]$ onto

$[0,1]^s$. Each point $\boldsymbol{x}_{i,t-1}$ lies on that curve and so they can be sorted in order of their pre-image in $[0,1]$ under the Hilbert curve. The updates come from the last d components of points $\boldsymbol{w}_{i,t} \in (0,1)^{1+d}$, after sorting them to have their first component in the same order as the Hilbert sort of $\boldsymbol{x}_{i,t-1}$.

The empirical results for array-RQMC so far outstrip what has been proved theoretically. Gerber and Chopin (2015) showed that their method has variance $o(n^{-1})$, using scrambled nets for their RQMC points. L'Ecuyer et al. (2008) show that for scalar x's it is possible to achieve variance of $O(n^{-3/2})$ by a version of array-RQMC.

Array-RQMC builds on the quasi-random walk methods of Lécot and Ogawa (2002), which are a form of array-QMC. They use unrandomized QMC and a reordering strategy to solve problems in chemistry where particles undergo diffusion and convection. They find empirical error rates that are better than for plain Monte Carlo. Examples 1, 2 and 3 in dimensions 1, 2 and 3 respectively attain error rates $n^{-0.73}$, $n^{-0.64}$ and $n^{-0.57}$. Changes in problems, methods and sample sizes could make these rates better or worse.

Chapter end notes

L'Ecuyer et al. (2010) make a study of the distribution of RQMC estimates for randomly shifted lattice rules. They include an analysis of one and two dimensional problems and some examples in higher dimensions. They especially note that no central limit theorem applies to the individual estimates.

The comparison between Cranley-Patterson rotations and digital shifts for $d=1$ is from an observation by F. J. Hickernell.

Bootstrap t

We would like a confidence interval for μ based on a small sample R of IID random variables $\hat\mu_1,\dots,\hat\mu_r$ with mean μ. Ideally the confidence interval would not require $\hat\mu_r$ to have a distribution in a known parametric family of distributions. Unfortunately, Bahadur and Savage (1956) describe a sense in which it is impossible to get such a nonparametric confidence interval for the mean of a random variable. We can however get confidence intervals asymptotic in R. The best of these is the bootstrap t method of Efron (1982) described here. The asymptotics for R presented here describe finite R much more closely than the Koksma-Hlawka asymptotics do.

Suppose that we have independent and identically distributed estimates $\hat\mu_1,\dots,\hat\mu_R$, unbiased for μ. We can form an asymptotic 99% confidence interval for μ as

$$\hat\mu_{\mathrm{pool}} \pm \frac{2.58s}{\sqrt{R}}. \tag{17.27}$$

for

$$\hat\mu = \frac{1}{R}\sum_{r=1}^R \hat\mu_r \quad\text{and}\quad s^2 = \frac{1}{R-1}\sum_{r=1}^R(\hat\mu_r - \hat\mu_{\mathrm{pool}})^2.$$

If the distribution of $\hat\mu_r$ has enough moments and is not supported on integer values or some other lattice in $\mathbb{R}$, then $\mathbb{P}(|\hat\mu_{\mathrm{pool}} - \mu| > 2.58s/\sqrt{R}) = 0.99 + O(1/R)$ Hall (1988). The implied constant in $O(\cdot)$ depends on the third and fourth moments of $\hat\mu_r$. Replacing 2.58 by the 0.995 quantile of the t distribution on $R-1$ degrees of freedom helps for small R but does not change the convergence rate.

The interval (17.27) is derived assuming that $\hat\mu_r$ have a Gaussian distribution, in which case using the t distribution removes the $O(1/R)$ coverage error entirely. Bootstrap confidence intervals are designed without assuming a specific distribution. In the percentile bootstrap, we sample $\hat\mu^{*1}, \hat\mu^{*2}, \dots, \hat\mu^{*R}$ with replacement from $(\hat\mu_1, \hat\mu_2, \dots, \hat\mu_R)$ and compute

$$\hat\mu^*_{\mathrm{pool}} = \frac{1}{R}\sum_{r=1}^R \hat\mu^*_r.$$

We do this independently $B \gg 1$ times, getting $\hat\mu^{*1}_{\mathrm{pool}}, \hat\mu^{*2}_{\mathrm{pool}}, \dots, \hat\mu^{*B}_{\mathrm{pool}}$. We then sort the values into $\hat\mu^{*(1)}_{\mathrm{pool}} \leqslant \hat\mu^{*(2)}_{\mathrm{pool}} \leqslant \cdots \leqslant \hat\mu^{*(B)}_{\mathrm{pool}}$. The central 99% of these is an approximate 99% percentile confidence interval for μ. That is, we use

$$[\hat\mu^{*(0.005B)}_{\mathrm{pool}}, \hat\mu^{*(0.995B)}_{\mathrm{pool}}].$$

We might have to round $0.005B$ down to an integer or $0.995B$ up to an integer, or because B is under our control, we can choose for B a multiple of 200, such as 10^5 or 10^6. A large value like this is reasonable when we want the 0.5 and 99.5 percentiles. The coverage accuracy of the bootstrap confidence interval is usually studied in the $B \to \infty$ limit, especially when it is inexpensive to take large B. The coverage error is however still $O(1/R)$, or $O(1/R + 1/\sqrt{B})$ for finite B.

The bootstrap t method of Efron (1982) generates $\hat\mu^*_1, \hat\mu^*_2, \dots, \hat\mu^*_R$ as before by resampling the original data. It then computes

$$t^* = \frac{\hat\mu^*_{\mathrm{pool}} - \hat\mu_{\mathrm{pool}}}{s^*/\sqrt{R}}, \quad\text{for}\quad s^{*2} = \frac{1}{R-1}\sum_{r=1}^R(\hat\mu^*_r - \hat\mu^*_{\mathrm{pool}})^2.$$

In the bootstrap t method we compute $t^{*1}, t^{*2}, \dots, t^{*B}$ from the resampled data, again for $B = 10^5$ or more, sort them as $t^{*(1)}, t^{*(2)}, \dots, t^{*(B)}$, and record $t^{*(0.005B)}$ and $t^{*(0.995B)}$. As $B \to \infty$ the distribution of the t^* becomes the exact distribution of the t statistic in a world where our QMC estimates had the $\mathbf{U}\{\hat\mu_1, \dots, \hat\mu_R\}$ distribution. For the bootstrap t we reason that t^* under resampling has almost the same distribution that t has under the unknown distribution of $\hat\mu_r$. Setting

$$0.01 = \mathbb{P}\bigl(t^{*(0.005B)} \leqslant t \leqslant t^{*(0.995B)}\bigr) = \mathbb{P}\Bigl(t^{*(0.005B)} \leqslant \frac{\hat\mu_{\mathrm{pool}} - \mu}{s/\sqrt{R}} \leqslant t^{*(0.995B)}\Bigr)$$

and solving we get

$$\hat{\mu}_{\text{pool}} - \frac{t^{*(0.995B)}s}{\sqrt{R}} \leqslant \mu \leqslant \hat{\mu}_{\text{pool}} - \frac{t^{*(0.005B)}s}{\sqrt{R}} \tag{17.28}$$

as the bootstrap t 99% approximate confidence interval for μ.

Hall (1988) finds expressions for the coverage error in the bootstrap t and other nonparametric approximate confidence intervals. The coverage error in central confidence intervals is $(A+B\gamma^2+C\kappa)/R+o(1/R^2)$, where $\gamma = \mathbb{E}((\hat{\mu}_r - \mu)^3)/\mathrm{Var}(\hat{\mu}_r)^{3/2}$ and $\kappa = \mathbb{E}((\hat{\mu}_r-\mu)^4)/\mathrm{Var}(\hat{\mu}_r)^2-3$, are the skewess and kurtosis respectively of $\hat{\mu}_r$. The values of γ and κ are hard to know for $\hat{\mu}_r$ from RQMC. For MC with n observations, $\gamma(\hat{\mu}) = \gamma(f(\boldsymbol{x}))/\sqrt{n}$ and $\kappa(\hat{\mu}) = \kappa(f(\boldsymbol{x}))/n$.

Alone among the nonparametric methods that Hall considers, the bootstrap t has $A = 0$. The others have $A < 0$ which makes then tend to cover μ less often than they should. If $f(\boldsymbol{x})$ has a Gaussian distribution, then $\gamma = \kappa = 0$ making the coverage error $O(1/R^2)$. Hall (1988) also describes a sense in which the bootstrap t chooses the right length for its confidence intervals.

Extreme cases for coverage error have γ^2 zero or large and κ below, equal to, or larger than the 0 we would have from a Gaussian distribution. That makes for 6 possibilities, but it is impossible to have an extremely large γ^2 with a small κ, so there are really 5 possibilities.

Owen (1992) simulates 95% confidence intervals for seven distributions (given in Exercise 17.1) including examples of all five (γ^2, κ) types and nine different confidence interval methods. The bootstrap t had the most reliable coverage of all methods tested. The coverage was reasonably good for sample sizes $n \geqslant 4$ except for log normally distributed data where none of the methods did well. The intervals were very long unless $n \geqslant 6$ and they had quite variable length unless $n \geqslant 7$. Additionally, Hall's asymptotic formula for coverage error was accurate already for $n = 18$ (except for lognormal data). Confidence intervals can also be judged by the coverage level attained at their given length. By that criterion, empirical likelhood intervals were best.

Bootstrap t confidence intervals can be very long because sometimes s^* is tiny. For small R, methods that do not generate some long confidence intervals typically fail to achieve the desired coverage. When as usual, all R values of $\hat{\mu}_r$ are distinct, and none is exactly equal to $\hat{\mu}_{\text{pool}}$, then there is an R^{1-R} chance of getting $s^{*b} = 0$ and $t^{*b} = \pm\infty$. We must choose R so that $R^{1-R} \lll 0.01$ in order to get finite values for $t^{*(0.005B)}$ and $t^{*(0.99B)}$, which leads to $R \geqslant 6$ or perhaps $R \geqslant 7$.

A slightly sharper version of the bootstrap uses $B = 99{,}999$ or some other multiple of 200, less one. The B bootstrap values of t^* partition the real line into B intervals, two of which have infinite length. Some will have length 0 due to ties among the t^{*b} values $b = 1, \dots, B$. We can then define $t^{*(0.005B)}$ and $t^{*(0.995B)}$ as the end points from the union of the central $0.99B$ of these intervals. See Davison and Hinkley (1997). For large enough B it won't make a practical difference.

For small R we can enumerate all of the different bootstrap samples t^* in a combinatorial problem. There are $\binom{2R-1}{R}$ of them with unequal weights.

Scrambles

Nested uniform scrambling of digital nets and sequences was introduced in Owen (1995). The digital scramble was mentioned by L'Ecuyer and Lemieux (2002) who attribute it to R. Couture. A taxonomy of scrambles appears in Owen (2003).

The generalized Faure and generalized Niederreiter sequences of Tezuka (1995) are non-random scrambles of those sequences designed to improve their performance. They take the form of a random linear scramble of Matoušek (1998), but use a deterministic choice for M_{kh} and they take $C_k = 0$. As a result, random linear scrambling is simultaneously a randomization of generalized Faure/Niederreiter sequences and a derandomization of nested uniform scrambling.

The I-binomial scramble of Tezuka and Faure (2003) is a further derandomization of random linear scrambling that uses $O(k)$ numbers to scramble k digits instead of $O(k^2)$. Owen (2003) presents an affine striped matrix scrambling that induces a local antithetic sampling pattern in the generated points. That leads to an improved convergence rate for one dimensional problems and for the one dimensional main effects in higher dimensions. But it does not improve the convergence rate for $d \geqslant 2$ overall.

L'Ecuyer and Lemieux (2005) report on a strategy by Morohosi to cache random seeds to reduce the space requirement of nested uniform sampling, by instead regenerating permutations as needed. They link to a web page that is no longer present.

Dimension reduction methods

Moskowitz and Caflisch (1996) presented the Brownian bridge construction for QMC evaluation of integrals involving a discretely sampled Brownian motion. To cover the time interval $[0, T]$, they sample at times T, $T/2$, $T/4$, $3T/4$ et cetera, continuing at times formed by multiplying T by numbers of the van der Corput sequence. The Brownian bridge construction also appears in Chapter 2 of Buslenko et al. (1966) which was written by I. M. Sobol'. That chapter includes MC and QMC but the Brownian bridge example there uses MC. Morokoff (1998) develops Brownian bridge sampling for stochastic differential equations of the form

$$\mathrm{d}S(t) = (a(t) + b(t)S)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}B(t)$$

where $B(t)$ is Brownian motion.

Acworth et al. (1997) proposed the principal components construction for problems of valuing financial options under a geometric Brownian motion model. There are also spatial versions of the principal components decomposition for regions in two or three or more dimensions. Heat or water might be flowing through a region and meeting a spatially random resistance as it moves. One can simulate that randomness by Monte Carlo and then measure some quantity of interest, often determined by solving a partial differential equation over the region of interest. Repeating the process several times gives an estimate the expected value of that quantity. A zero mean Gaussian spatial process on such a region can be written, in a Karhunen-Loève expansion, as $f(\boldsymbol{t}) = \sum_{\ell=1}^{\infty} \gamma_\ell \psi_\ell(\boldsymbol{t}) z_\ell$ for functions $\psi_\ell(\boldsymbol{t})$ random variables $z_\ell \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$ and coefficients $\gamma_1 \geqslant \gamma_2 \geqslant \cdots \geqslant 0$. QMC or RQMC can be used on the first L components z_ℓ. Graham et al. (2015) consider partial differential equations in a random lognormal environment using QMC to sample their environment.

There is no reason to expect that either the Brownian bridge or the principal component construction is necessarily best for a given application. Indeed, Papageorgiou (2002) shows that for certain digital options the Brownian bridge can be outperformed by the standard construction. Åkesson and Lehoczky (2000) consider a weighted principal components method for financial problems where future values of the security of interest should be discounted by an interest rate. Imai and Tan (2006) present a method that searches numerically for the best square root C of the Gaussian covariance matrix Σ.

Another difference between RQMC and MC arises in the transformations we use to create random variables. We can sample Gaussian variables $\boldsymbol{z}_i \sim \mathcal{N}(0, I)$ by inverting the Gaussian CDF, or by the Box-Muller transformation. These choices will produce estimates $\hat{\mu}$ with identical mean and identical variance in MC but they will be different in general under RQMC. Many authors, e.g., Morokoff and Caflisch (1993), advocate for inversion. Ökten and Göncü (2011) take the contrary view, especially for integrands that depend on the norm of the Gaussian random vector. For even d, the norm of the Gaussian vector will have been determined by only $d/2$ of the components in $\boldsymbol{x}_i$ compared to all d of them under inversion.

Padding and hybrid methods

Spanier (1995) describes a hybrid method in which deterministic quasi-Monte Carlo points are padded out with ordinary Monte Carlo. Ökten (1996) analyzes the discrepancy of such hybrid schemes. Owen (1994) considers padding randomized orthogonal array samples with Latin hypercube samples.

Exercises

17.1. This exercise is a mini-project to calibrate the bootstrap t for 99% intervals. Consider sample sizes $n = 6, 7, \dots, 30$ for random variables x with these distributions:

i) $x \sim \text{Exp}(1)$ (Exponential),
ii) $x \sim \mathcal{N}(0,1)$ (Gaussian),
iii) $x \sim \exp(\mathcal{N}(0,1))$ (Lognormal),
iv) $x \sim 0.25\mathcal{N}(3,1) + 0.75\mathcal{N}(-1,1)$ (Mixture of normals),
v) $x \sim t_{(4)}$ (Student's t),
vi) $x \sim x - 2x, 0 < x < 1$ (Triangular density function), and
vii) $x \sim \mathbf{U}(0,1)$ (Uniform).

In RQMC, each x is a $\hat{\mu}$ and n is R. The lognormal distribution is especially challenging.

a) Write or find code to sample n numbers $j^*(1), j^*(2), \dots, j^*(n)$ uniformly from integers $i = 1, 2, \dots, n$, with replacement. (In bootstrapping x_i^* will be $x_{j^*(i)}$.) For $n = 3$, there are only 27 possible ordered samples. Sample many times to verify that all 27 appear approximately the right number of times.

b) For our bootstrap purposes getting observations $(1, 3, 2, 1)$, when $n = 4$, is the same as getting $(1, 1, 2, 3)$. That is, the order in which the samples were taken does not count. Enumerate the distinct samples for $n = 4$, compute their true exact probabilities as an integer multiple of 4^{-4} and compare their frequencies in sampling to their true probabilities.

c) For $n = 6$ and $x_i \sim \mathrm{Exp}(1)$, using $B = 100{,}000$ bootstrap samples, what fraction of the time does the 99% bootstrap confidence interval contain the true mean? Use a large number N of samples of sizes $n = 6$ to get your answer. Turn in your code including the seed you used so it can be reproduced. If necessary reduce B in order to use a large N.

d) Repeat the previous computation for the other six distributions above. Turn in your code.

e) Repeat the previous computation for $n = 7, 8, \cdots, 30$. Turn in your code.

f) Instead of coverage, report width calibration. Specifically for each distribution and sample size n find a factor w such that

$$wt^{*(0.005B)} \leqslant \frac{\hat{\mu} - \mu}{s/\sqrt{n}} \leqslant wt^{*(0.995B)}$$

holds with estimated probability 0.99. Such factors are commonly called ***fudge factors***.

g) Find fudge factors for the usual, non bootstrapped, t distribution for the same sample sizes and distributions.

Some parts can be done by writing code that wraps the prior parts. Instructors may want to skip some parts or change recommendations for N and B or change the subset of n values.

17.2. Prove equation (17.9) bounding the discrepancy of Cranley-Patterson rotation.

17.3. This requires considers some positional scrambles of the Faure sequence and therefore requires code for the Faure sequence. Exercise 17.4 is similar but based on the more easily programmed Halton sequence.

a) Compute the first 2 components of the first 530 points of Faure's $(0, 53)$-sequence in base 53. Plot the second component versus the first for these points.

b) Apply a positional scramble to both components above, using a uniform random permutation π_{jk} for the k'th base b digit of the j'th component. The π_{jk} are mutually independent. For this exercise you may truncate the expansion at 4 digits in base 53. (In applications, more digits should be accounted for.) Plot the scrambled second component versus the scrambled first.

c) Repeat the previous part but now use the same uniform random permutation π_j for all digits of the j'th component of the points. Take π_1 and π_2 to be independent uniform random permutations of $\{0,\dots,52\}$.

17.4. Do Exercise 17.3 with the following substitutions: Replace the first two components of the Faure sequence by the 19'th and 20'th components of the Halton sequence (prime bases 67 and 71). Replace the base 53 expansions by ones in bases 67 and 71 as appropriate. Use random permutations of $\{0,\dots,66\}$ and $\{0,\dots,70\}$ as appropriate.

17.5. xxx distn of $\hat\mu - \mu$ for fixed pts under cpr with/without Baker

17.6. Let the mean dimension of f on be $1+\epsilon$ for $\epsilon > 0$ from its ANOVA decomposition on $[0,1]^d$ be $1+\epsilon$ for $\epsilon > 0$. Show that that

$$\sum_{j=1}^{d} \sigma^2_{\{j\}} \geqslant (1-\epsilon)\sigma^2.$$

17.7. For $i \geqslant 1$, let $a_i = \phi_2(i)$ be the van der Corput sequence in base 2. Let x_i be a nested uniform scramble of a_i, in base 2. Let y_i be a second, independent, nested uniform scramble of a_i, also in base 2.

a) Does (x_{88}, y_{88}) have the uniform distribution in the unit square?

b) Would (x_{88}, y_{88}) be uniformly distributed if digital shifts were used?

17.8. Theory project. Theorem 17.11 requires low discrepancy points. Find an asymptotic rate for $\mathbb{E}(|\hat\mu - \mu|)$, if instead the points have exactly the same discrepancy bounds that plain Monte Carlo points have.

17.9. Chapter 6 contains pseudocode for the Brownian bridge construction of Brownian motion. Compare the Brownian bridge construction to the principal components construction for the Asian option problem of §17.8. Try both dimensions $d = 16$ and $d = 250$.

17.10. Perhaps the Box-Muller algorithm would be better for the Asian option problem than the method of inversion used in §17.8.

Bibliography

Acworth, P., Broadie, M., and Glasserman, P. (1997). A comparison of some Monte Carlo techniques for option pricing. In Niederreiter, H., Hellekalek, P., Larcher, G., and Zinterhof, P., editors, *Monte Carlo and quasi-Monte Carlo methods '96*, pages 1–18. Springer.

Adams, C. R. and Clarkson, J. A. (1934). Properties of functions $f(x, y)$ of bounded variation. *Transactions of the American Mathematical Society*, 36(4):711.

Aistleitner, C. and Dick, J. (2015). Functions of bounded variation, signed measures, and a general Koksma–Hlawka inequality. *Acta Arithmetica*, 167:143–171.

Åkesson, F. and Lehoczky, J. P. (2000). Path generation for quasi-Monte Carlo simulation of mortgage-backed securities. *Management Science*, 46(9):1171–1187.

Alexander, J. R., Beck, J., and Chen, W. W. L. (2018). Geometric discrepancy theory and uniform distribution. In Toth, C. D., O'Rourke, J., and Goodman, J. E., editors, *Handbook of discrete and computational geometry*, page 279. CRC Press, Boca Raton, FL.

Atanassov, E. (2004). On the discrepancy of the Halton sequence. *Mathematica Balkanica*, 18:15–32.

Bahadur, R. R. and Savage, L. J. (1956). The nonexistence of certain statistical procedures in nonparametric problems. *The Annals of Mathematical Statistics*, 27(4):1115–1122.

Basu, K. and Owen, A. B. (2015). Scrambled geometric net integration over general product spaces. *Foundations of Computational Mathematics*, pages 1–30.

Basu, K. and Owen, A. B. (2018). Quasi-Monte Carlo for an integrand with a singularity along a diagonal in the square. In Dick, J., Kuo, F. Y., and Woźniakowski, H., editors, *Contemporary Computational Mathematics-A Celebration of the 80th Birthday of Ian Sloan*, pages 119–130. Springer.

Beck, J. and Chen, W. L. (1987). *Irregularities of Distribution.* Cambridge University Press, Cambridge.

Binder, C. (1970). Über einen Satz von de Bruijn und Post. *Österreichische Akademie der Wissenschaften Mathematisch-Naturwissenschaftliche Klasse. Sitzungsberichte. Abteilung II*, 179:233–251.

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

Breneis, S. and Hinrichs, A. (2019). Fibonacci lattices have minimal dispersion on the two-dimensional torus. Technical Report arXiv:1905.03856, Johannes Kepler Universität Linz.

Buslenko, N. P., Golenko, D. I., Schreider, Yu. A., Sobol', I. M., and Sragovich, V. G. (1966). *The Monte Carlo Method: the Method of Statistical Trials.* Pergamon Press, New York. Translated by G. J. Tee, translation edited by D. M. Parkyn.

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

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

Chazelle, B. (2000). *The Discrepancy Method: Randomness and Complexity.* Cambridge University Press, Cambridge.

Chen, S. (2011). *Consistency and convergence rate of Markov chain quasi Monte Carlo with examples.* PhD thesis, Stanford University.

Chen, S., Dick, J., and Owen, A. B. (2011). Consistency of Markov chain quasi-Monte Carlo on continuous state spaces. *The Annals of Statistics*, 39(2):673–701.

Chen, S., Matsumoto, M., Nishimura, T., and Owen, A. B. (2012). New inputs and methods for Markov chain quasi-Monte Carlo. In Plaskota, L. and Woźniakowski, H., editors, *Monte Carlo and quasi-Monte Carlo methods 2010*, pages 313–327. Springer.

Chen, Y., Bornn, L., De Freitas, N., Eskelin, M., Fang, J., and Welling, M. (2016). Herded Gibbs sampling. *The Journal of Machine Learning Research*, 17(1):263–291.

Chentsov, N. N. (1967). Pseudorandom numbers for modelling Markov chains. *Computational Mathematics and Mathematical Physics*, 7:218–233.

Chi, H., Mascagni, M., and Warnock, T. (2005). On the optimal Halton sequence. *Mathematics and computers in simulation*, 70(1):9–21.

Chung, K.-L. (1949). An estimate concerning the Kolmogoroff limit distribution. *Transactions of the American Mathematical Society*, 67(1):36–50.

Clarkson, J. A. and Adams, C. R. (1933). On definitions of bounded variation for functions of two variables. *Transactions of the American Mathematical Society*, 35(4):824–854.

Cools, R., Kuo, F., and Nuyens, D. (2006). Constructing embedded lattice rules for multivariate integration. *SIAM Journal on Scientific Computing*, 28(6):2162–2188.

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

Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap methods and their application*. Cambridge University Press, Cambridge.

de Bruijn, N. G. and Post, K. A. (1968). A remark on uniformly distributed sequences and Riemann integrability. *Indagationes Mathematicae*, 30:149–150.

Dick, J. (2008). Walsh spaces containing smooth functions and quasi-Monte Carlo rules of arbitrarily high order. *SIAM Journal of Numerical Analysis*, 46(3):1519–1553.

Dick, J. (2009). The decay of the Walsh coefficients of smooth functions. *Bulletin of the Australian Mathematical Society*, 80(3):430–453.

Dick, J. (2011). Higher order scrambled digital nets achieve the optimal rate of the root mean square error for smooth integrands. *The Annals of Statistics*, 39(3):1372–1398.

Dick, J., Kuo, F. Y., and Sloan, I. H. (2013). High-dimensional integration: the quasi-Monte Carlo way. *Acta Numerica*, 22:133–288.

Dick, J., Nuyens, D., and Pillichshammer, F. (2014). Lattice rules for nonperiodic smooth integrands. *Numerische Mathematik*, 126(2):259–291.

Dick, J. and Pillichshammer, F. (2010). *Digital sequences, discrepancy and quasi-Monte Carlo integration*. Cambridge University Press, Cambridge.

Doerr, C., Gnewuch, M., and Wahlström, M. (2014). Calculation of discrepancy measures and applications. In W., C., A., S., and G., T., editors, *A Panorama of Discrepancy Theory*, pages 621–678. Springer.

Efron, B. (1982). *The jackknife, the bootstrap, and other resampling plans*, volume 38. SIAM, Philadelphia, PA.

Fang, K.-T. and Wang, Y. (1994). *Number Theoretic Methods in Statistics.* Chapman & Hall.

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

Faure, H. (1992). Good permutations for extreme discrepancy. *Journal of Number Theory*, 42(1):47–56.

Faure, H. and Lemieux, C. (2009). Generalized Halton sequences in 2008: A comparative study. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 19(4):15.

Fox, B. L. (1986). Algorithm 647: implementation and relative efficiency of quasirandom sequence generators. *ACM Transactions on Mathematical Software*, 12(4):362–376.

Fréchet, M. (1910). Extension au cas des intégrales multiples d'une définition de l'intégrale due à Stieltjes. *Nouvelles Annales de Mathématiques*, 10:241–256.

Gerber, M. and Chopin, N. (2015). Sequential quasi-Monte Carlo. *Journal of the Royal Statistical Society, Series B*, 77(3):509–579.

Glasserman, P. G. (2004). *Monte Carlo methods in financial engineering.* Springer, New York.

Gnewuch, M., Srivastav, A., and Winzen, C. (2009). Finding optimal volume subintervals with k points and calculating the star discrepancy are NP-hard problems. *Journal of Complexity*, 25(2):115–127.

Golubov, B. I. (1984). Multiple Fourier series and integrals. *Journal of Soviet Mathematics*, 24(6):639–673.

Grafakos, L. (2004). *Classical and Modern Fourier Analysis.* Prentice Education Inc., Upper Saddle River, NJ.

Graham, I. G., Kuo, F. Y., Nichols, J. A., Scheichl, R., Schwab, Ch., and Sloan, I. H. (2015). Quasi-Monte Carlo finite element methods for elliptic PDEs with lognormal random coefficients. *Numerische Mathematik*, 131(2):329–368.

Griebel, M., Kuo, F. Y., and Sloan, I. H. (2010). The smoothing effect of the ANOVA decomposition. *Journal of Complexity*, 26(5):523–551.

Griebel, M., Kuo, F. Y., and Sloan, I. H. (2013). The smoothing effect of integration in $\mathbb{R}^d$ and the ANOVA decomposition. *Mathematics of Computation*, 82(281):383–400.

Hall, P. (1988). Theoretical comparisons of bootstrap confidence intervals. *The Annals of Statistics*, 16(3):927–953.

Halton, J. H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2(1):84–90.

Hammersley, J. M. (1960). Monte Carlo methods for solving multivariable problems. *Annals of the New York Academy of Sciences*, 86(3):844–874.

Hartinger, J. and Kainhofer, R. (2006). Non-uniform low-discrepancy sequence generation and integration of singular integrands. In Niederreiter, H. and Talay, D., editors, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 163–179. Springer.

Hartinger, J., Kainhofer, R., and Ziegler, V. (2005). On the corner avoidance properties of various low-discrepancy sequences. *INTEGERS: Electronic Journal of Combinatorial Number Theory*, 5(3):A10.

Heinrich, S. (1996). Efficient algorithms for computing the l_2-discrepancy. *Mathematics of Computation of the American Mathematical Society*, 65(216):1621–1633.

Hickernell, F. J. (1996a). The mean square discrepancy of randomized nets. *ACM Transactions on Modeling and Computer Simulation*, 6(4):274–296.

Hickernell, F. J. (1996b). Quadrature error bounds with applications to lattice rules. *SIAM Journal on Numerical Analysis*, 33(5):1995–2016. Erratum: 1997, v. 34, n. 2, pp 853–866.

Hickernell, F. J. (1998). Goodness-of-fit statistics, discrepancies and robust designs. *Statistics and Probability Letters*, 44(1):73–78.

Hickernell, F. J. (2002). Obtaining $o(n^{-2+\epsilon})$ convergence for lattice quadrature rules. In Fang, K.-T., Hickernell, F. J., and Niederreiter, N., editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 434–445, New York. Springer-Verlag.

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

Hickernell, F. J., Hong, H. S., L'Ecuyer, P., and Lemieux, C. (2000). Extensible lattice sequences for quasi-Monte Carlo quadrature. *SIAM Journal on Scientific Computing*, 22(3):1117–1138.

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

Hickernell, F. J. and Niederreiter, H. (2003). The existence of good extensible rank-1 lattices. *Journal of Complexity*, 19(3):286–300.

Hlawka, E. (1961). Funktionen von eschränkter variation in der theorie der gleichverteilung. *Annali di Matematica Pura Applicata*, 54(1):325–333.

Hlawka, E. (1962). Zur angenäherten Berechnung mehrfacher Integrale. *Monatshefte fur mathematik*, 66(2):140–151.

Hong, H. S. and Hickernell, F. J. (2003). Algorithm 823: Implementing scrambled digital sequences. *AMS Transactions on Mathematical Software*, 29(2):95–109.

Hua, L. K. and Wang, Y. (1960). Remarks concerning numerical integration. *Scientific Record (New Series)*, 4(1):8–11.

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

Imai, J. and Tan, K. S. (2006). A general dimension reduction technique for derivative pricing. *Journal of Computational Finance*, 10(2):129.

Joe, S. and Disney, S. A. R. (1993). Intermediate rank lattice rules for multidimensional integration. *SIAM journal on numerical analysis*, 30(2):569–582.

Joe, S. and Kuo, F. Y. (2008). Constructing Sobol' sequences with better two-dimensional projections. *SIAM Journal on Scientific Computing*, 30(5):2635–2654.

Keller, A. (1997). Instant radiosity. Technical report, Universität Kauserslautern.

Kiefer, J. (1961). On large deviations of the empirical d.f. of vector chance variables and a law of the iterated logarithm. *Pacific Journal of Mathematics*, 11(2):649–660.

Knuth, D. E. (1998). *The Art of Computer Programming*, volume 2: Seminumerical algorithms. Addison-Wesley, Reading MA, 3rd edition.

Koksma, J. F. (1942/1943). Een algemeene stelling uit de theorie der gelijkmatige verdeeling modulo 1. *Mathematica B (Zutphen)*, 11:7–11.

Kollig, T. and Keller, A. (2006). Illumination in the presence of weak singularities. In Niederreiter, H. and Talay, D., editors, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 245–257. Springer.

Korobov, N. M. (1959). The approximate computation of multiple integrals (Russian). *Doklady Akademii Nauk SSSR*, 124(6):1207–1210.

Kritzer, P. (2006). Improved upper bounds on the star discrepancy of (t, m, s)-nets and (t, s)-sequences. *Journal of Complexity*, 22(3):336–347.

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

Kuo, F. Y. (2003). Component-by-component constructions achieve the optimal rate of convergence for multivariate integration in weighted Korobov and Sobolev spaces. *Journal of Complexity*, 19(3):301–320.

Kuo, F. Y. and Nuyens, D. (2016). Application of quasi-Monte Carlo methods to elliptic PDEs with random diffusion coefficients: a survey of analysis and implementation. *Foundations of Computational Mathematics*, 16(6):1631–1696.

Larcher, G. (1987). A best lower bound for good lattice points. *Monatshefte für Mathematik*, 104(1):45–51.

Lécot, C. and Ogawa, S. (2002). Quasirandom walk methods. In Fang, K.-T., Hickernell, F. J., and Niederreiter, H., editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000.* Springer.

L'Ecuyer, P., Lécot, C., and Tuffin, B. (2008). A randomized quasi-Monte Carlo simulation method for Markov chains. *Operations Research*, 56(4):958–975.

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

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

L'Ecuyer, P. and Lemieux, C. (2005). Recent advances in randomized quasi-Monte Carlo methods. In Dror, M., L'Ecuyer, P., and Szidarovszky, F., editors, *Modeling uncertainty*, pages 419–474. Kluwer Academic Publishers, New York.

L'Ecuyer, P. and Munger, D. (2016). Algorithm 958: Lattice builder: A general software tool for constructing rank-1 lattice rules. *ACM Transactions on Mathematical Software (TOMS)*, 42(2):15.

L'Ecuyer, P., Munger, D., Lécot, C., and Tuffin, B. (2018). Sorting methods and convergence rates for Array-RQMC: some empirical comparisons. *Mathematics and Computers in Simulation*, 143:191–201.

L'Ecuyer, P., Munger, D., and Tuffin, B. (2010). On the distribution of integration error by randomly-shifted lattice rules. *Electronic Journal of Statistics*, 4:950–993.

Lemieux, C. (2009). *Monte Carlo and quasi-Monte Carlo Sampling.* Springer, New York.

Lemieux, C. and L'Ecuyer, P. (1998). Efficiency improvement by lattice rules for pricing Asian options. In *Proceedings of the 1998 Winter Simulation Conference*, pages 579–585.

Loh, W.-L. (2003). On the asymptotic distribution of scrambled net quadrature. *Annals of Statistics*, 31(4):1282–1324.

Lyness, J. N. (1989). An introduction to lattice rules and their generator matrices. *IMA Journal of Numerical Analysis*, 9(3):405–149.

Maize, E. (1981). *Contributions to the Theory of Error Reduction in quasi-Monte Carlo methods.* PhD thesis, The Claremont Graduate School.

Matoušek, J. (1998). On the L^2–discrepancy for anchored boxes. *Journal of Complexity*, 14(4):527–556.

Matoušek, J. (1999). *Geometric discrepancy: An illustrated guide*, volume 18. Springer-Verlag, Berlin.

Morokoff, W. J. (1998). Generating quasi-random paths for stochastic processes. *SIAM Review*, 40(4):765–788.

Morokoff, W. J. and Caflisch, R. E. (1993). A quasi-Monte Carlo approach to particle simulation of the heat equation. *SIAM Journal on Numerical Analysis*, 30(6):1558–1573.

Morokoff, W. J. and Caflisch, R. E. (1995). Quasi-Monte Carlo integration. *Journal of computational physics*, 122(2):218–230.

Moskowitz, B. and Caflisch, R. E. (1996). Smoothness and dimension reduction in quasi-Monte Carlo methods. *Mathematical and Computer Modelling*, 23(8-9):37–54.

Mosteller, F. and Tukey, J. W. (1968). Data analysis, including statistics. *Handbook of social psychology*, 2:80–203.

Niederreiter, H. (1986). Multidimensional numerical integration using pseudo-random numbers. In *Stochastic Programming 84 Part I*, volume 27, pages 17–38. Springer, Berlin.

Niederreiter, H. (1987). Point sets and sequences with small discrepancy. *Monatshefte fur mathematik*, 104(4):273–337.

Niederreiter, H. (1992a). Low-discrepancy point sets obtained by digital constructions over finite fields. *Czechoslovak Mathematical Journal*, 42(1):143–166.

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

Niederreiter, H. (1993). Improved error bounds for lattice rules. *Journal of Complexity*, 9(1):60–75.

Niederreiter, H. and Xing, C. (1996). Low-discrepancy sequences and global function fields with many rational places. *Finite Fields and their applications*, 2(3):241–273.

Nuyens, D. (2013). The construction of good lattice rules and polynomial lattice rules. Technical Report arXiv:1308.3601, KU Leuven.

Nuyens, D. and Cools, R. (2006). Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel hilbert spaces. *Mathematics of Computation*, 75(254):903–920.

Ökten, G. (1996). A probabilistic result on the discrepancy of a hybrid-Monte Carlo sequence and applications. *Monte Carlo Methods and Applications*, 2(4):255–270.

Ökten, G. and Göncü, A. (2011). Generating low-discrepancy sequences from the normal distribution: Box–Muller or inverse transform? *Mathematical and Computer Modelling*, 53(5-6):1268–1281.

Ökten, G., Shah, M., and Goncharov, Y. (2009). Random and deterministic digit permutations of the Halton sequence. Technical report, University of Florida.

Ökten, G., Shah, M., and Goncharov, Y. (2012). Random and deterministic digit permutations of the Halton sequence. In Plaskota, L. and Woźniakowski, H., editors, *Monte Carlo and Quasi-Monte Carlo Methods 2010*, pages 609–622. Springer.

Owen, A. B. (1992). Empirical likelihood and small samples. In *Computing Science and Statistics*, pages 79–88. Springer.

Owen, A. B. (1994). Lattice sampling revisited: Monte Carlo variance of means over randomized orthogonal arrays. *Annals of Statistics*, 22:930–945.

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

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

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

Owen, A. B. (1998). Latin supercube sampling for very high dimensional simulations. *ACM Transactions on Modeling and Computer Simulation*, 8(2):71–102.

Owen, A. B. (2003). Variance with alternative scramblings of digital nets. *ACM Transactions on Modeling and Computer Simulation*, 13(4):363–378.

Owen, A. B. (2005). Multidimensional variation for quasi-Monte Carlo. In Fan, J. and Li, G., editors, *International Conference on Statistics in honour of Professor Kai-Tai Fang's 65th birthday*.

Owen, A. B. (2006a). Halton sequences avoid the origin. *SIAM review*, 48(3):487–503.

Owen, A. B. (2006b). Randomized QMC and point singularities. In Niederreiter, H. and Talay, D., editors, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 403–418. Springer.

Owen, A. B. (2008). Local antithetic sampling with scrambled nets. *Annals of Statistics*, 36(5):2319–2343.

Owen, A. B. and Tribble, S. D. (2005). A quasi-Monte Carlo Metropolis algorithm. *Proceedings of the National Academy of Sciences*, 102(25):8844–8849.

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

Paskov, S. and Traub, J. (1995). Faster valuation of financial derivatives. *The Journal of Portfolio Management*, 22:113–120.

Pirsic, G. (2002). A software implementation of the Niederreiter-Xing sequences. In Fang, K.-T., Hickernell, F. J., and Niederreiter, N., editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 434–445, New York. Springer-Verlag.

Puchhammer, F., Ben Abdellah, A., and L'Ecuyer, P. (2019). Variance reduction for chemical reaction netowrks with array-RQMC. MCM 2019 slides.

Richtmyer, R. D. (1952). The evaluation of definite integrals, and a quasi-Monte Carlo method based on the properties of algebraic numbers. Technical Report LA-1342, University of California.

Roth, K. F. (1954). On irregularities of distribution. *Mathematica*, 1(2):73–79.

Schlier, Ch. (2004). Error trends in quasi-Monte Carlo integration. *Computer physics communications*, 159(2):93–105.

Schmid, W. C. (1999). The exact quality parameter of nets derived from Sobol' and Niederreiter sequences. In Illiev, O., Kaschiev, M. S., Margenov, S. D., Sendov, B. H., and Vassilevski, P. S., editors, *Recent advances in numerical methods and applications*, pages 287–295, Singapore. World Scientific.

Schmid, W. C. (2001). Projections of digital nets and sequences. *Mathematics and computers in simulation*, 55(1–3):239–247.

Schürer, R. and Schmid, W. C. (2009). MinT–new features and new results. In L'Ecuyer, P. and Owen, A. B., editors, *Monte Carlo and Quasi-Monte Carlo Methods 2008*, pages 501–512, Berlin. Springer-Verlag.

Schwedes, T. and Calderhead, B. (2018). Quasi Markov chain Monte Carlo methods. Technical Report arXiv:1807.00070, Imperial College, London.

Sidi, A. (1993). A new variable transformation for numerical integration. In Bräss, H. and Hämmerlin, B., editors, *Numerical Integration IV*. Birkhäuser Verlag, Basel.

Sloan, I. H. and Joe, S. (1994). *Lattice Methods for Multiple Integration.* Oxford Science Publications, Oxford.

Sloan, I. H. and Reztsov, A. V. (2002). Component-by-component construction of good lattice rules. *Mathematics of Computation*, 71(237):263–273.

Sobol', I. M. (1967). The distribution of points in a cube and the accurate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112.

Sobol', I. M. (1973a). Calculation of improper integrals using uniformly distributed sequences. *Soviet Mathematics Doklady*, 14:734–738.

Sobol', I. M. (1973b). On the use of uniformly distributed sequences for approximate computations of improper integrals. *Theory of cubature formulas and applications to certain problems in mathematical physics*, pages 62–66. In Russian.

Sobol', I. M., Asotsky, D., Kreinin, A., and Kucherenko, S. (2011). Construction and comparison of high-dimensional Sobol' generators. *Wilmott magazine*, 2011(56):64–79.

Spanier, J. (1995). Quasi-Monte Carlo Methods for Particle Transport Problems. In Niederreiter, H. and Shiue, P. J.-S., editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 121–148, New York. Springer-Verlag.

Surjanovic, S. and Bingham, D. (2013). Virtual library of simulation experiments: test functions and datasets. `https://www.sfu.ca/~ssurjano/`.

Swayne, D. F., Lang, D. T., Buja, A., and Cook, D. (2003). GGobi: evolving from XGobi into an extensible framework for interactive data visualization. *Computational Statistics and Data Analysis*, 43(4):423–444.

Tezuka, S. (1995). *Uniform random numbers: theory and practice.* Kluwer Academic Publishers, Boston.

Tezuka, S. and Faure, H. (2003). I-binomial scrambling of digital nets and sequences. *Journal of Complexity*, 19(6):744–757.

Tribble, S. D. (2007). *Markov chain Monte Carlo algorithms using completely uniformly distributed driving sequences.* PhD thesis, Stanford University.

Tribble, S. D. and Owen, A. B. (2008). Construction of weakly CUD sequences for MCMC sampling. *Electronic Journal of Statistics*, 2:634–660.

Tuffin, B. (1998). Variance reduction order using good lattice points in Monte Carlo methods. *Computing*, 61(4):371–378.

Vandewoestyne, B. (2008). *Quasi-Monte Carlo techniques for approximation of high-dimensional integrals.* PhD thesis, Katholieke Universiteit, Leuven.

Vandewoestyne, B. and Cools, R. (2006). Good permutations for deterministic scrambled Halton sequences in terms of L2-discrepancy. *Journal of computational and applied mathematics*, 189(1-2):341–361.

Wang, X. and Hickernell, F. J. (2000). Randomized Halton sequences. *Mathematical and Computer Modelling*, 32(7-8):887–899.

Wang, X. and Sloan, I. H. (2006). Efficient weighted lattice rules with applications to finance. *SIAM Journal on Scientific Computing*, 28(2):728–750.

Wang, X. and Sloan, I. H. (2011). Quasi-Monte Carlo methods in financial engineering: An equivalence principle and dimension reduction. *Operations Research*, 59(1):80–95.

Warnock, T. T. (1972). Computational investigations of low discrepancy point sets. In Zaremba, S. K., editor, *Applications of number theory to numerical analysis*, pages 319–343. Academic Press, New York.

Weyl, H. (1914). Über ein problem aus dem gebiete der diophantischen approximationen. *Nachrichten der Akademie der Wissenschaften in Göttingen. II. Mathematisch-Physikalische Klasse*, pages 234–244.

Weyl, H. (1916). Über die gleichverteilung von zahlen mod. eins. *Mathematische Annalen*, 77:313–352.

Zaremba, S. K. (1968). Mathematical basis of Monte Carlo and quasi-Monte Carlo methods. *SIAM Review*, 10(3):303–314.

Zhu, H. and Dick, J. (2014). Discrepancy bounds for deterministic acceptance-rejection samplers. *Electronic Journal of Statistics*, 8(1):678–707.