ON DROPPING THE FIRST SOBOL' POINT

By

Art B. Owen

Technical Report No. 2020-12
September 2020

Department of Statistics
STANFORD UNIVERSITY
Stanford, California 94305-4065

ON DROPPING THE FIRST SOBOL' POINT

By

Art B. Owen
Stanford University

Technical Report No. 2020-12
September 2020

**This research was supported in part by
National Science Foundation grant IIS-1837931.**

Department of Statistics
STANFORD UNIVERSITY
Stanford, California 94305-4065

http://statistics.stanford.edu

On dropping the first Sobol' point

Art B. Owen
Stanford University

August 2020

Abstract

Quasi-Monte Carlo (QMC) points are a substitute for plain Monte Carlo (MC) points that greatly improve integration accuracy under mild assumptions on the problem. Because QMC can give errors that are $o(1/n)$ as $n \to \infty$, changing even one point can change the estimate by an amount much larger than the error would have been and worsen the convergence rate. As a result, certain practices that fit quite naturally and intuitively with MC points are very detrimental to QMC performance. These include thinning, burn-in, and taking sample sizes such as powers of 10, other than the ones for which the QMC points were designed. This article looks at the effects of a common practice in which one skips the first point of a Sobol' sequence. The retained points ordinarily fail to be a digital net and when scrambling is applied, skipping over the first point can increase the numerical error by a factor proportional to $\sqrt{n}$ where n is the number of function evaluations used.

1 Introduction

A Sobol' sequence is an infinite sequence of points $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots \in [0,1]^d$ constructed to fill out the unit cube with low discrepancy, meaning that a measure of the distance between the discrete uniform distribution on $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ and the continuous uniform distribution on $[0,1]^d$ is made small. These points are ordinarily used to approximate

$$\mu = \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \quad \text{by} \quad \hat{\mu} = \hat{\mu}_{\boldsymbol{u},1} = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{u}_i).$$

The reason for calling this estimate $\hat{\mu}_{\boldsymbol{u},1}$ will become apparent later. Sobol' sequences are often used to estimate expectations with respect to unbounded random variables, such as Gaussians. In such cases f subsumes a transformation from the uniform distribution on $[0,1]^d$ to some other more appropriate distribution. This article uses 1-based indexing, so that the initial point is $\boldsymbol{u}_1$. Sometimes 0-based indexing is used, and then the initial point is denoted $\boldsymbol{u}_0$. Both indexing conventions are widespread in mathematics and software for Sobol' points.

The initial point of the Sobol' sequence is $\boldsymbol{u}_1 = (0, 0, \ldots, 0)$. A common practice is to skip that point, similar to the burn-in practice in Markov chain Monte Carlo (MCMC). One then estimates μ by

$$\hat{\mu} = \hat{\mu}_{\boldsymbol{u},2} = \frac{1}{n}\sum_{i=2}^{n+1} f(\boldsymbol{u}_i).$$

One reason to skip the initial point is that a transformation to a Gaussian distribution might make the initial Gaussian point infinite. That is problematic not just for integration problems but also when f is to be evaluated at the design points to create surrogate models for Bayesian optimization [9, 1]. If one skips the initial point, then the next point in a Sobol' sequence is usually $(1/2, 1/2, \ldots, 1/2)$. While that is an intuitively much more reasonable place to start, it has detrimental consequences and there are better remedies, described here.

A discussion about whether to drop the initial point came up in the plenary tutorial of Fred Hickernell at MCQMC 2020 about QMCPy [5] software for QMC, at `https://github.com/QMCSoftware/QMCSoftware`. The issue has been discussed by the pytorch community at `https://github.com/pytorch/pytorch/issues/32047`, and the scipy community at `https://github.com/scipy/scipy/pull/10844`, which are both incorporating QMC methods. This article shows that skipping even one point of the Sobol' sequence can be very detrimental. The resulting points are no longer a digital net in general, and in the case of scrambled Sobol' points, skipping a point can bring about an inferior rate of convergence, making the estimate less accurate by a factor that is roughly proportional to $\sqrt{n}$.

A second difficulty with Sobol' sequence points is that it is difficult to estimate the size $|\hat{\mu} - \mu|$ of the integration error from the data. The well-known Koksma-Hlawka inequality [11] bounds $|\hat{\mu} - \mu|$ by the product of two unknown and essentially uncomputable quantities and, while tight for some worst case integrands, it can yield an extreme overestimate the error, growing ever more conservative as the dimension d increases.

Randomly scrambling the Sobol' sequence points preserves their balance properties and provides a basis for uncertainty quantification. Scrambling turns points $\boldsymbol{u}_i$ into random points $\boldsymbol{x}_i \sim \mathsf{U}[0,1]^d$. The points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are not independent. Instead they retain the digital net property of Sobol' points and consequent accuracy properties. The result is randomized QMC (RQMC) points. RQMC points also have some additional accuracy properties stemming from the randomization. With scrambled Sobol' points, we estimate μ by

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

One can estimate the mean squared error using R independent replicates of the n-point RQMC estimate $\hat{\mu}_{\boldsymbol{x},1}$. It is also possible to drop the first point in

RQMC, estimating μ by

$$\hat{\mu} = \hat{\mu}_{\boldsymbol{x},2} = \frac{1}{n}\sum_{i=2}^{n+1} f(\boldsymbol{x}_i).$$

The purpose of this paper is to show that $\hat{\mu}_{\boldsymbol{x},1}$ is a much better choice than $\hat{\mu}_{\boldsymbol{x},2}$.

Many implementations of a Sobol' sequence will produce $n = 2^m$ points $\boldsymbol{u}_i \in \{0, 1/n, 2/n, \ldots, (n-1)/n\}^d \subset [0,1)^d$. In that case, a safer way to avoid having a point at the origin is to take $\boldsymbol{u}_i + 1/(2n)$ componentwise. This is reasonable if one has already decided on the value of n to use. It does not work to add that same value $1/(2n)$ to the next 2^m points and subsequent values. For one thing, the result may produce values on the upper boundary of $[0,1]^d$ in the very next batch and will eventually place points outside of $[0,1]^d$. It remains better to scramble the Sobol' points.

This paper is organized as follows. Section 2 defines digital nets and shows that skipping over the first point can destroy the digital net property underlying the analysis of Sobol' sequences. It also presents properties of scrambled digital nets. Section 3 shows some empirical investigations on some very simple and favorable integrands where $\mathbb{E}((\hat{\mu}_{\boldsymbol{x},1} - \mu)^2)$ shows decays very nearly to the rate $O(n^{-3})$ while $\mathbb{E}((\hat{\mu}_{\boldsymbol{x},2} - \mu)^2)$ decays very nearly to $O(n^{-2})$. These are both in line with what we expect from asymptotic theory. The relevance is not that our integrands are as trivial as those examples, but rather that when realistic integrands are well approximated by such simple ones we stand to gain from using the original scrambled Sobol' points. They essentially provide the value we would have had from using those simple functions as control variates [12] but without us having to search for control variates. There is also no theoretical reason to expect $\mathbb{E}((\hat{\mu}_{\boldsymbol{x},2} - \mu)^2)$ to be smaller than $\mathbb{E}((\hat{\mu}_{\boldsymbol{x},1} - \mu)^2)$ and so there is a Pascal's wager argument against dropping the first point. Section 4 looks at a ten dimensional function representing the weight of an airplane wing as a function of the way it was made. We see there also that the theoretically expected rates match the empirical data. Section 5 considers some very special cases where burn-in might be harmless and also recommends against using round number sample sizes and thinning QMC points.

2 Digital nets and scrambling

In this section we review digital nets and describe properties of their scrambled versions. The points from Sobol' sequences provide the most widely used example of digital nets. For details of their construction and analysis see the monographs [7, 20]. There are numerous implementations of Sobol' sequences [2, 14, 35]. They differ in what are called 'direction numbers' and they can also deliver the points in different orderings. The numerical results here use direction numbers from [14] via an implementation from Nuyens' magic point shop, described in [16] and scrambled as in [21].

We begin with the notion of elementary intervals, which are special hyper-rectangular subsets of $[0,1)^d$. For an integer base $b \geqslant 2$, a dimension $d \geqslant 1$, a vector $\boldsymbol{k} = (k_1, \ldots, k_d)$ of integers $k_j \geqslant 0$ and a vector $\boldsymbol{c} = (c_1, \ldots, c_d)$ of integers with $0 \leqslant c_j < b^{k_j}$, the Cartesian product

$$E(\boldsymbol{k}, \boldsymbol{c}) = \prod_{j=1}^{d} \Big[\frac{c_j}{b^{k_j}}, \frac{c_j + 1}{b^{k_j}} \Big)$$

is an elementary interval in base b. It has volume $b^{-|\boldsymbol{k}|}$ where $|\boldsymbol{k}| = \sum_{j=1}^{d} k_j$.

Speaking informally, the set $E(\boldsymbol{k}, \boldsymbol{c})$ has a proportion $b^{-|\boldsymbol{k}|}$ of the volume of $[0,1]^d$ and so it 'deserves' to get (i.e., contain) $nb^{-|\boldsymbol{k}|}$ points when we place n points inside $[0,1]^d$. Digital nets satisfy that condition for certain $\boldsymbol{k}$. We use the following definitions from Niederreitter [19].

Definition 1. For integers $m \geqslant t \geqslant 0$, the $n = b^m$ points $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n \in [0,1]^d$ are a (t, m, d)-net in base $b \geqslant 2$, if every elementary interval $E(\boldsymbol{k}, \boldsymbol{c}) \subset [0,1]^d$ of volume b^{t-m} contains exactly b^t of the points $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$.

Every elementary interval that 'deserves' b^t points of the digital net, gets that many of them. When we speak of digital nets we ordinarily mean (t, m, d)-nets though some authors reserve the term 'digital' to refer to specific construction algorithms rather than just the property in Definition 1.

Definition 2. For integers $t \geqslant 0$, $b \geqslant 2$ and $d \geqslant 1$, the infinite sequence $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots \in [0,1]^d$ is a (t, d)-sequence in base b if $\boldsymbol{u}_{(r-1)b^m+1}, \ldots, \boldsymbol{u}_{rb^m}$ is a (t, m, d)-net in base b for any integers $m \geqslant t$ and $r \geqslant 1$.

Sobol' sequences [32] are (t, d)-sequences in base 2. From Definition 2, we see that the first 2^m points of a Sobol' sequence are a (t, m, d)-net in base 2 for any $m \geqslant t$. So are the second 2^m points, and if we merge both of those point sets, we get a $(t, m+1, d)$-net in base 2. We can merge the first two of those to get a $(t, m+2, d)$-net in base 2 and so on ad infinitum.

Given b, m and d, smaller values of t are better. It is not always possible to have $t = 0$. The best known values of t for (t, d)-sequences and (t, m, d)-nets are given in the online MinT web site [30]. The published t value for a Sobol' sequence might be conservative in that the first b^m points of the Sobol' sequence can possibly be a (t', m, d)-net for some $t' < t$.

The proven properties of digital nets including those taken from Sobol' sequences derive from their balanced sampling of elementary intervals. The analysis path can be via discrepancy [20] or Haar wavelets [33] or Walsh functions [7].

The left panel in Figure 1 shows the first 16 points of a Sobol' sequence in two dimensions. Fifteen of them are small solid disks and one other is represented by concentric circles at the origin. Those points form a $(0, 4, 2)$-net in base 2. Reference lines divide the unit square into a 4×4 grid of elementary intervals of size $1/4 \times 1/4$. Each of those has one of the 16 points, often at the lower left corner. Recall that elementary intervals include their lower boundary but not their upper boundary. Finer reference lines partition the unit square into

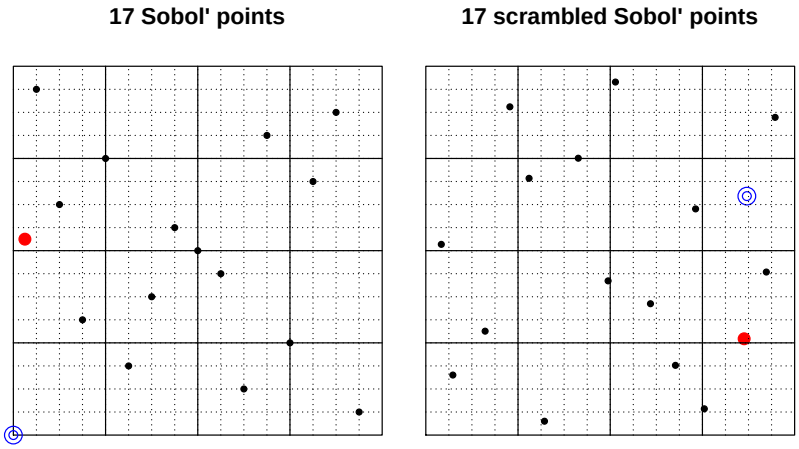

Figure 1: The left panel shows the first 17 Sobol' points in $[0,1]^2$. The initial point at $(0,0)$ is shown in concentric circles. The 17'th point is shown as a large disk. Solid reference lines partition $[0,1]^2$ into 16 congruent squares. Dashed reference lines partition it into 256 congruent squares. The right panel shows a nested uniform scramble of these 17 points.

16 strips of size $1 \times 1/16$. Each of those has exactly one point of the digital net. The same holds for the 16 rectangles of each of these shapes: $1/2 \times 1/8$, $1/8 \times 1/2$ and $1/16 \times 1$. All told, those 16 points have balanced 80 elementary intervals and the number of balanced intervals grows rapidly with m and d.

The point $\boldsymbol{u}_1 = (0,0)$ is problematic as described above. If we skip it and take points $\boldsymbol{u}_2, \dots, \boldsymbol{u}_{17}$ then we replace it with the large solid disk at $(1/32, 17/32)$. Doing that leaves the lower left $1/4 \times 1/4$ square empty and puts two points into a square above it. The resulting 16 points now fail to be a $(0,4,2)$-net.

The introduction mentioned some randomizations of digital nets. There is a survey of RQMC in [17]. For definiteness, we consider the nested uniform scramble from [21]. Applying a nested uniform scramble to a (t,d)-sequence $\boldsymbol{u}_1, \boldsymbol{u}_2, \dots$ in base b yields points $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots$ that individually satisfy $\boldsymbol{x}_i \sim \mathsf{U}[0,1]^d$ and collectively are a (t,d)-net in base b with probability one. The estimate $\hat{\mu}_{\boldsymbol{x},1}$ then satisfies $\mathbb{E}(\hat{\mu}_{\boldsymbol{x},1}) = \mu$ by uniformity of $\boldsymbol{x}_i$. If $f \in L^{1+\epsilon}[0,1]^d$ for some $\epsilon > 0$ then [29] show that $\Pr(\lim_{m\to\infty} \hat{\mu}_{\boldsymbol{x},1} = \mu) = 1$, where the limit is through (t,m,d)-nets formed by initial b^m subsequences the (t,d)-sequence of $\boldsymbol{x}_i$. If $f \in L^2[0,1]^d$ then $\mathrm{var}(\hat{\mu}_{\boldsymbol{x},1}) = o(1/n)$ as $n = b^m \to \infty$ [22]. When a plain Monte Carlo average of $f(\cdot)$ at $n = b^m$ IID $\mathsf{U}[0,1]^d$ points has variance σ^2/n then $\mathrm{var}(\hat{\mu}_{\boldsymbol{x},1}) \leqslant \Gamma\sigma^2/n$ for some $\Gamma < \infty$ whenever $f \in L^2[0,1]^d$ [24]. In particular $\mathrm{var}(\hat{\mu}_{\boldsymbol{x},1})/(\sigma^2/n)$ contains none of the powers of $\log(n)$ that one sees in the Koksma-Hlawka based bounds. Under further smoothness conditions on f (a square integrable mixed partial derivative taken once with respect to

each component of $\boldsymbol{x}$), $\mathrm{var}(\hat{\mu}_{\boldsymbol{x},1}) = O(n^{-3}(\log n)^{d-1})$ [23, 25]. To reconcile the appearance and non-appearance of logarithmic factors, those results are that $\mathrm{var}(\hat{\mu}_{\boldsymbol{x},1}) \leqslant \min(\Gamma\sigma^2/n, A_n)$ for some sequence $A_n = O(n^{-3}\log(n)^{d-1})$. The logarithmic factor can degrade the n^{-3} rate but only subject to a cap on performance relative to plain Monte Carlo. Finally, Loh [18] proves a central limit theorem for $\hat{\mu}_{\boldsymbol{x},1}$ when $t = 0$.

The right panel of Figure 1 shows a nested uniform scramble of the points in the left panel. The problematic point $\boldsymbol{u}_1$ becomes a uniformly distributed point in the square, and is no longer on the boundary. If we replace it by $\boldsymbol{u}_{17}$ then just as in the unscrambled case, there is an empty $1/4 \times 1/4$ elementary interval, and another one with two points.

There is a disadvantage to $\hat{\mu}_{\boldsymbol{x},2}$ compared to $\hat{\mu}_{\boldsymbol{x},1}$ when the latter attains a root mean squared error $O(n^{-3/2+\epsilon})$, for then

$$\hat{\mu}_{\boldsymbol{x},2} = \hat{\mu}_{\boldsymbol{x},1} + \frac{1}{n}\big(f(\boldsymbol{x}_{n+1}) - f(\boldsymbol{x}_1)\big). \tag{1}$$

The term $(f(\boldsymbol{x}_{n+1}) - f(\boldsymbol{x}_1))/n = O(1/n)$ will ordinarily decay more slowly than $|\hat{\mu}_{\boldsymbol{x},1} - \mu|$. Then skipping the first point will actually make the rate of convergence worse. A similar problem happens if one simply ignores $\boldsymbol{x}_1$ and averages the $n-1$ points $f(\boldsymbol{x}_2)$ through $f(\boldsymbol{x}_n)$. A related problem is that when equally weighed integration rules have errors $O(n^{-r})$ for $r > 1$, that can only realistically take place at geometrically separated values of n. See [34, 26].

3 Synthetic examples

Here we look at some very simple modest dimensional integrands. They fit into a 'best case' case analysis for integration, motivated as follows. We suppose that some sort of function $g(\boldsymbol{x})$ is extremely favorable for a method and also that it resembles the actual integrand. We may write

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) + \varepsilon(\boldsymbol{x}).$$

In the favorable cases, ε is small and g is easily integrated. For classical quadratures g may be a polynomial [6]. For digital nets, some functions g may have rapidly converging Walsh series [7], others are sums of functions of only a few variables at a time [3]. For lattice rules [31], a favorable g has a rapidly converging Fourier series. The favorable cases work well because

$$\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i) = \frac{1}{n}\sum_{i=1}^{n} g(\boldsymbol{x}_i) + \frac{1}{n}\sum_{i=1}^{n} \varepsilon(\boldsymbol{x}_i)$$

with the first term having small error because it is well suited to the method and the second term having small error because $\varepsilon(\cdot)$ has a small norm and we take an equal weight sample of it instead of using large weights of opposite signs. A good match between method and g saves us the chore of searching for one or more control variates. Choosing cases where a method ought to work is like

the positive controls used in experimental science. We can use them to verify that the method or its numerical implementation work as expected on the cases they were designed for. There can and will be unfavorable cases in practice. Measuring the sample variance under replication provides a way to detect that.

Here we consider some cases where scrambled nets should work well. The first is

$$g_0(\boldsymbol{x}) = \sum_{j=1}^{d} \bigl(e^{x_j} - e + 1\bigr), \tag{2}$$

which clearly has $\mu = 0$. This sum of centered exponentials is smooth and additive. It is thus very simple for QMC and RQMC. It is unlikely that anybody turns to RQMC for this function but as remarked above the integrand one has may be close to such a simple function.

Figure 2 shows the root mean squared error for this function g_0 based on $R = 10$ independent replicates of both $\hat{\mu}_{\boldsymbol{x},1}$ and $\hat{\mu}_{\boldsymbol{x},2}$. Ordinarily 10 replicates are not enough to estimate a mean squared error well but this case is an exception because the underlying variances span many orders of magnitude and that signal then swamps the sampling uncertainty. Reference lines show a clear pattern. The error follows a reference line parallel to $n^{-3/2}$ on a log-log plot for $\hat{\mu}_{\boldsymbol{x},1}$. For $\hat{\mu}_{\boldsymbol{x},2}$, the reference line is parallel to n^{-1}. These slopes are exactly what we would expect from the underlying theory, the first from [23] and the second from equation (1). In both cases the line goes through the data for $n = 32$ and is then extrapolated to $n = 2^{14} = 16{,}384$ with the given slopes. That is a more severe test for the asymptotic theory than fitting by least squares would be. In this instance, the asymptotic theory is close to the measurements. Both theory and measurements show a strong advantage to retaining the scrambled initial point.

An earlier version of this article used $g_0(\boldsymbol{x}) = \sum_{j=1}^{d} x_j$ instead of the function g_0 above. The RMSEs for that function also closely follow the predicted rates. It is not however as good a test case because it is antisymmetric about $\boldsymbol{x} = (1/2, \ldots, 1/2)$, meaning that $(g_0(\boldsymbol{x}) + g_0(\tilde{\boldsymbol{x}}))/2 = \mu$ for all $\boldsymbol{x}$, where $\tilde{\boldsymbol{x}} = 1 - \boldsymbol{x}$ componentwise. If we use such an antisymmetric function, then we will get highly accurate results just from having a nearly antithetic set of evaluation points that may or may not be equidistributed.

The second function is

$$g_1(\boldsymbol{x}) = \left(\sum_{j=1}^{d} x_j \right)^2. \tag{3}$$

Unlike g_0 this function is not additive. It has interactions of order 2 but no higher in the functional ANOVA decomposition [13, 33] and it also has a substantial additive component. It is not antisymmetric about $(1/2, 1/2, \ldots, 1/2)$. It has $\mu = d/3 + d(d-1)/4$. Figure 3 shows the root mean squared error for $\hat{\mu}_{\boldsymbol{x},1}$ and $\hat{\mu}_{\boldsymbol{x},2}$. Once again they follow reference lines parallel to $n^{-3/2}$ and n^{-1} respectively. Asymptotic theory predicts a squared error with a component

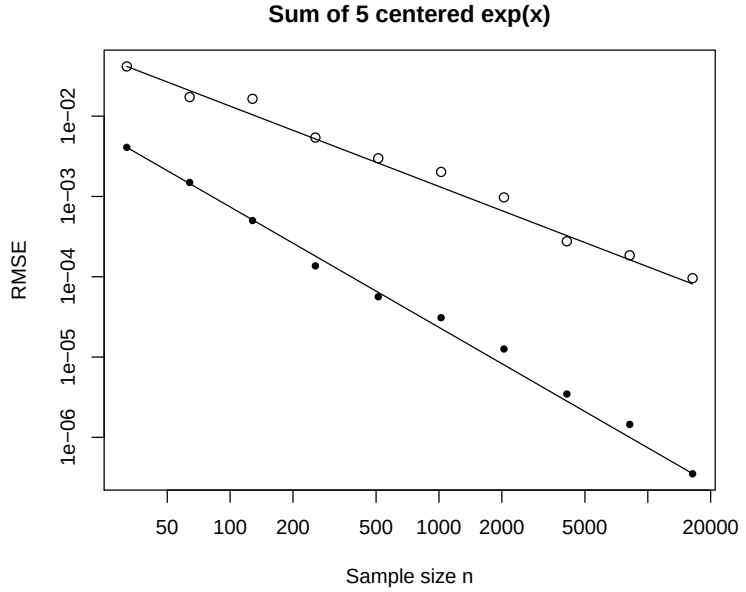

Figure 2: Solid points show RMSE for scrambled Sobol' estimate $\hat{\mu}_{\boldsymbol{x},1}$ versus n from $R = 10$ replicates. A reference line parallel to $n^{-3/2}$ goes through the first solid point. Open points show RMSE for scrambled Sobol' estimates $\hat{\mu}_{\boldsymbol{x},2}$ which drop the initial zero. A reference line parallel to n^{-1} goes through the first open point.

proportional to n^{-3} and a second one proportional to $\log(n)n^{-3}$ that would eventually dominate the first.

Next we look at a product

$$g_2(\boldsymbol{x}) = \prod_{j=1}^{d} (e^{x_j} - e + 1).$$

This function has $\mu = 0$ for any d. It is surprisingly hard for (R)QMC to handle this function for modest d, much less large d. It is dominated by 2^d spikes of opposite signs around the corners of $[0,1]^d$. It may also be extra hard for Sobol' points compared to alternatives, because Sobol' points often have rectangular blocks that alternate between double the uniform density and emptiness. In a functional ANOVA decomposition, it is purely d-dimensional in that the only non-zero variance component is the one involving all d variables. Asymptotic theory predicts an RMSE that is $O(\log(n)^{(d-1)/2}/n^{3/2})$.

Figure 4 shows results for $d = 3$ and this $g_2(\boldsymbol{x})$. The rate for $\hat{\mu}_{\boldsymbol{x},1}$ shows up as slightly worse than $n^{-3/2}$ while the one for $\hat{\mu}_{\boldsymbol{x},2}$ appears to be slighly better than n^{-1}. Both are much better than $O(n^{-1/2})$. Putting in the predicted logarithmic factor improves the match between asymptotic prediction and empirical outcome

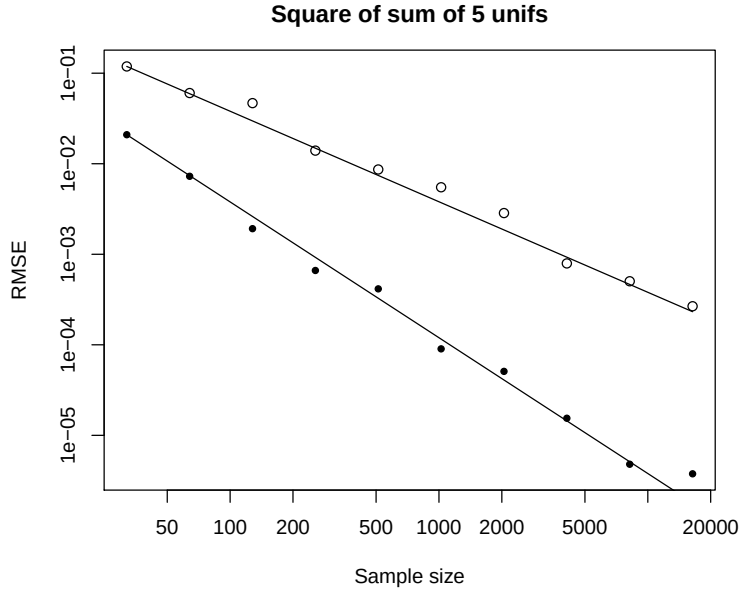

Figure 3: Solid points show RMSE for scrambled Sobol' estimate $\hat{\mu}_{\boldsymbol{x},1}$ versus n from $R = 10$ replicates. A reference line parallel to $n^{-3/2}$ goes through the first solid point. Open points show RMSE for scrambled Sobol' estimates $\hat{\mu}_{\boldsymbol{x},2}$ which drop the initial zero. A reference line parallel to n^{-1} goes through the first open point.

for $\hat{\mu}_{\boldsymbol{x},1}$. It is not clear what can explain $\hat{\mu}_{\boldsymbol{x},2}$ doing better here than the aysmptotic prediction. Perhaps the asymptotics become descriptive of actual errors at much larger n for this function than for the others. Judging by eye it is possible that the convergence rate is worse when the first point is dropped, but the evidence is not as clear as in the other figures where the computed values so closely follow theoretical predictions. There is an evident benefit to retaining the initial point that at a minimum manifests as a constant factor of improvement.

4 Wing weight function

The web site [36] includes a 10 dimensional function that computes the weight of a wing based on a physical model of the way the wing is manufactured. While one does not ordinarily want to know the average weight of a randomly manufactured wing, this function is interesting in that it has a real physical world origin instead of being completely synthetic. It is easily integrated by several QMC methods [28] and so it is very likely that it equals $g + \varepsilon$ for a favorable g and a small ε.

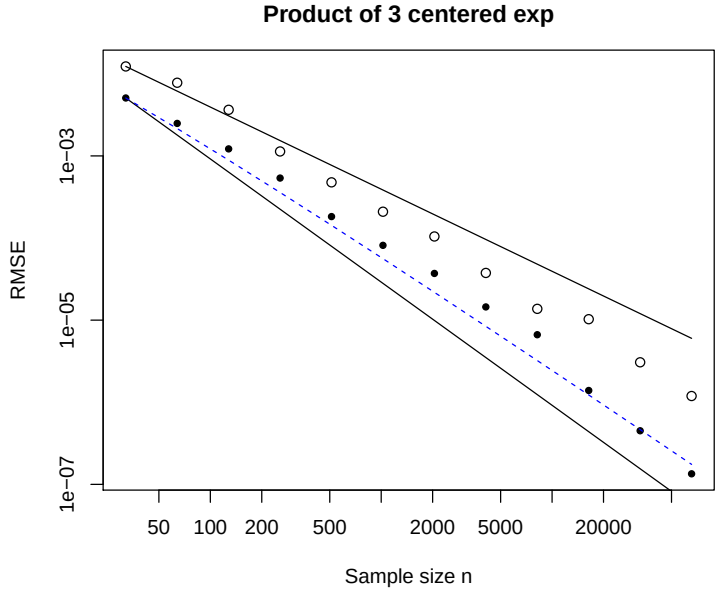

Figure 4: The integrand is a product of 3 centered exponentials. Solid points show RMSE for scrambled Sobol' estimate $\hat{\mu}_{\boldsymbol{x},1}$ versus n from $R = 10$ replicates. A reference line parallel to $n^{-3/2}$ goes through the first solid point. Open points show RMSE for scrambled Sobol' estimates $\hat{\mu}_{\boldsymbol{x},2}$ which drop the initial zero. A reference line parallel to n^{-1} goes through the first open point. A dashed reference line through the first solid point decays as $\log(n)/n^{3/2}$.

The wing weight function is

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

The definition and uniform ranges of these variables are given in Table 1.

For this function the standard deviation among 10 independent replicates is used instead of the root mean squared error. The results are in Figure 5. Once again there is a disadvantage to dropping the first Sobol' point. Even though this function has 10 inputs we still see a roughly $O(n^{-3/2})$ decay of the RMSE for scrambled Sobol' points and something more like $O(n^{-1})$ when the first point is omitted.

5 Discussion

QMC and RQMC points come as an $n \times d$ matrix of numbers in $[0, 1]$ that we can then pipe through several functions to change the support set and distribution

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 ($\mathrm{lb}/\mathrm{ft}^2$)
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 ($\mathrm{lb}/\mathrm{ft}^2$)

Table 1: Variables and their ranges for the wing weight function.

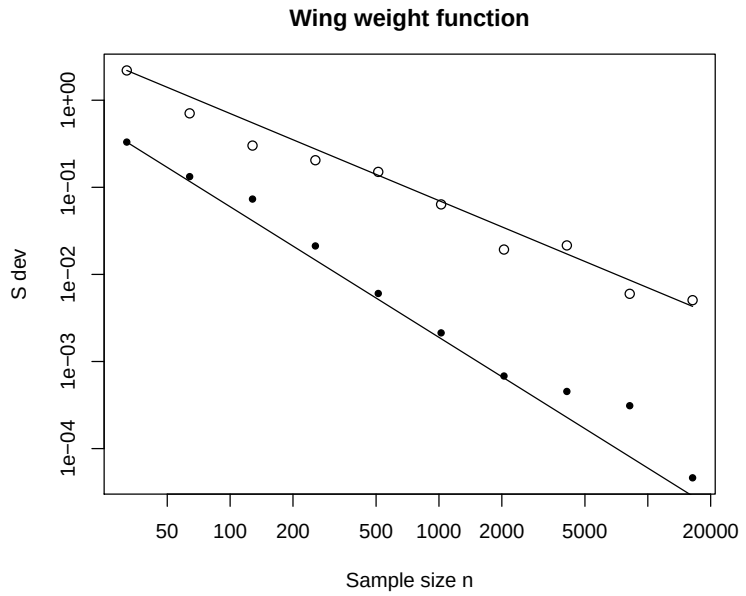

Figure 5: Solid points show standard deviation for scrambled Sobol' estimate $\hat{\mu}_{\boldsymbol{x},1}$ versus n from $R = 10$ replicates. A reference line parallel to $n^{-3/2}$ goes through the first solid point. Open points show standard deviation for scrambled Sobol' estimates $\hat{\mu}_{\boldsymbol{x},2}$ which drop the initial zero. A reference line parallel to n^{-1} goes through the first open point.

and finally evaluate a desired integrand. They are only slightly different from plain Monte Carlo points but the difference makes it easy to misapply them.

This paper has focussed on a small burn-in, dropping just one of the points and picking up the next n. Burn-in makes no difference to plain Monte Carlo

apart from doing some unneeded function evaluations, and it can bring large benefits to MCMC. See the comment by Neal in the discussion [15]. Burn-in typically spoils the digital net property. It is safer to just scramble the points which removes the problematic point near a singularity while also increasing accuracy on very favorable functions like those in the examples and also on some unfavorable ones having singularities or other sources of infinite variation in the sense of Hardy and Krause. See [29].

There are some exceptional cases where burn-in of (R)QMC may be harmless. For $d = 1$, any consecutive 2^m points of the van der Corput sequence [37] are a $(0, m, 1)$-net in base 2. It is possible that dropping the first $N = 2^M$ points of a Sobol' sequence for $M = 1000$ or some other large number will cause no problems (beyond numerical representation). Of course, that needs to be tested empirically. If one then takes the next $2^m \ll 2^M$ points they will be a digital net. Burning in many more points than one could ever use preserves the net property and may well avoid the boundary of $[0,1]^d$ sufficiently well. The Halton sequence [10] has few if any especially good sample sizes N and large burn-ins have been used

For plain Monte Carlo points it is natural to use a round number like 1000 or 10^6 of sample points. That can be very damaging in (R)QMC if the points were defined for some other sample size. Using 1000 points of a Sobol' sequence may well be less accurate than using 512. Typical sample sizes are powers of 2 for digital nets and large prime numbers for lattice rules [31, 17]. The Faure sequences [8] use $b = p \geqslant d$ where p is a prime number. With (R)QMC as with antibiotics, one should take the whole sequence.

Another practice that works well in MCMC, but should not be used in (R)QMC is 'thinning'. In MCMC, thinning can save storage space and in some cases can improve efficiency despite increasing variance [27]. One takes every k'th point, $\boldsymbol{x}_{k\times i}$ for some integer $k > 1$, or in combination with burn-in $\boldsymbol{x}_{B+k\times i}$ for some integer $B \geqslant 1$. To see the problem, consider the very basic van der Corput sequence $x_i \in [0,1]$. If $x_i \in [0, 1/2)$ then $x_{i+1} \in [1/2, 1)$. For instance [4] use that observation to point out that simulating a Markov chain with van der Corput points can be problematic. Now suppose that one thins the van der Corput sequence to every second point using $k = 2$. All of the retained points are then in either $[0, 1/2)$ or in $[1/2, 1)$. One will estimate either $2\int_0^{1/2} f(x)\,\mathrm{d}x$ or $2\int_{1/2}^2 f(x)\,\mathrm{d}x$ by using that sequence. The first component of a Sobol' sequence is usually a van der Corput sequence.

The Matlab R2020a `sobolset` function `https://www.mathworks.com/help/stats/sobolset.html` as of August 11, 2020 includes a thinning option through a parameter Leap which is an interval between points, corresponding to $k - 1$ in the discussion above. It also has a parameter Skip, corresponding to burn-in, which is a number of initial points to omit. Fortunately both Leap and Skip default to zero. However even having them present is problematic. It is not clear how one should use them safely. The left panel of Figure 6 shows a histogram of the values $\boldsymbol{x}_{10i,1}$ for $1 \leqslant i \leqslant \lfloor 2^{20}/10 \rfloor$. The right panel shows a histogram of the values $\boldsymbol{x}_{10i,2}$.

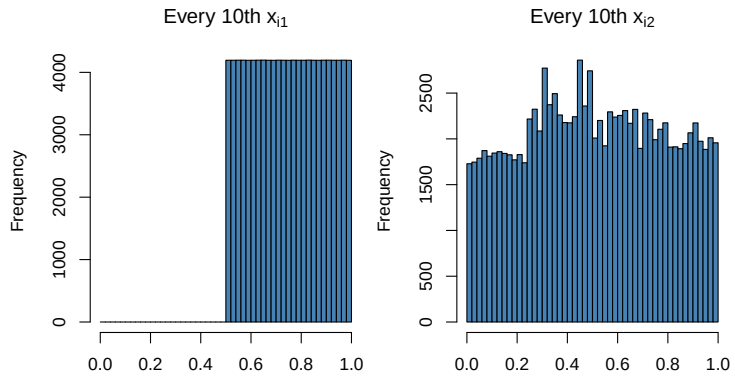

Figure 6: The left panel shows a histogram of every 10'th $\boldsymbol{x}_{i1}$ from the first 2^{20} Sobol' points. The right panel shows a histogram of every 10'th $\boldsymbol{x}_{i2}$ from the first 2^{20} Sobol' points.

Acknowledgments

This work was supported by the NSF under grant IIS-1837931, and a grant from Hitachi, Ltd. I thank Fred Hickernell, Pierre L'Ecuyer, Alex Keller, Max Balandat, Michael McCourt, Pamphile Roy and Sergei Kucherenko for stimulating discussions. Thanks to Mike Giles, Arnaud Doucet, Alex Keller and the whole team at ICMS for making MCQMC 2020 happen despite all the pandemic disruption.

References

[1] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. BoTorch: Bayesian optimization in PyTorch. Technical Report arXiv:1910.06403, Facebook Research, 2019.

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

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

[4] R. E. Caflisch and B. Moskowitz. Modified Monte Carlo methods using quasi-random sequences. In H. Niederreiter and P. J.-S. Shiue, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 1–16, New York, 1995. Springer-Verlag.

[5] S.-C. T. Choi, F. J. Hickernell, M. McCourt, and A. Sorokin. QMCPy: A quasi-Monte Carlo Python library, 2020.

[6] P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Academic Press, San Diego, 2nd edition, 1984.

[7] J. Dick and F. Pillichshammer. *Digital sequences, discrepancy and quasi-Monte Carlo integration*. Cambridge University Press, Cambridge, 2010.

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

[9] P. I. Frazier. A tutorial on Bayesian optimization. Technical Report arXiv:1807.02811, 2018.

[10] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2(1):84–90, 1960.

[11] F. J. Hickernell. Koksma-Hlawka inequality. *Wiley StatsRef: Statistics Reference Online*, 2014.

[12] F. J. Hickernell, C. Lemieux, and A. B. Owen. Control variates for quasi-Monte Carlo (with discussion). *Statistical Science*, 20(1):1–31, 2005.

[13] W. Hoeffding. A class of statistics with asymptotically normal distribution. *Annals of Mathematical Statistics*, 19(3):293–325, 1948.

[14] S. Joe and F. Y. Kuo. Constructing Sobol' sequences with better two-dimensional projections. *SIAM Journal on Scientific Computing*, 30(5):2635–2654, 2008.

[15] R. E. Kass, B. P. Carlin, A. Gelman, and R. M. Neal. Markov chain Monte Carlo in practice: a roundtable discussion. *The American Statistician*, 52(2):93–100, 1998.

[16] F. Y. Kuo and D. Nuyens. 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, 2016.

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

[18] W.-L. Loh. On the asymptotic distribution of scrambled net quadrature. *Annals of Statistics*, 31(4):1282–1324, 2003.

[19] H. Niederreiter. Point sets and sequences with small discrepancy. *Monatshefte fur mathematik*, 104(4):273–337, 1987.

[20] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, Philadelphia, PA, 1992.

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

[22] A. B. Owen. Monte Carlo variance of scrambled net quadrature. *SIAM Journal of Numerical Analysis*, 34(5):1884–1910, 1997.

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

[24] A. B. Owen. Scrambling Sobol' and Niederreiter-Xing points. *Journal of Complexity*, 14(4):466–489, December 1998.

[25] A. B. Owen. Local antithetic sampling with scrambled nets. *Annals of Statistics*, 36(5):2319–2343, 2008.

[26] A. B. Owen. A constraint on extensible quadrature rules. *Numerische Mathematik*, pages 1–8, 2015.

[27] A. B. Owen. Statistically efficient thinning of a Markov chain sampler. *Journal of Computational and Graphical Statistics*, 26(3):738–744, 2017.

[28] A. B. Owen. Monte carlo book: the quasi-monte carlo parts. `https://statweb.stanford.edu/~owen/mc/`, 2019.

[29] A. B. Owen and D. Rudolf. A strong law of large numbers for scrambled net integration. *SIAM Review*, 2020. to appear.

[30] R. Schürer and W. C. Schmid. MinT–new features and new results. In P. L'Ecuyer and A. B. Owen, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2008*, pages 501–512, Berlin, 2009. Springer-Verlag.

[31] I. H. Sloan and S. Joe. *Lattice Methods for Multiple Integration*. Oxford Science Publications, Oxford, 1994.

[32] I. M. Sobol'. The distribution of points in a cube and the accurate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967.

[33] I. M. Sobol'. *Multidimensional Quadrature Formulas and Haar Functions*. Nauka, Moscow, 1969. (In Russian).

[34] I. M. Sobol'. Asymmetric convergence of approximations of the Monte Carlo method. *Computational Mathematics and Mathematical Physics*, 33(10):1391–1396, 1993.

[35] I. M. Sobol', D. Asotsky, A. Kreinin, and S. Kucherenko. Construction and comparison of high-dimensional Sobol' generators. *Wilmott magazine*, 2011(56):64–79, 2011.

[36] S. Surjanovic and D. Bingham. Virtual library of simulation experiments: test functions and datasets. `https://www.sfu.ca/~ssurjano/`, 2013.

[37] J. G. van der Corput. Verteilungsfunktionen I. *Nederl. Akad. Wetensch. Proc.*, 38:813–821, 1935.