LATIN SUPERCUBE SAMPLING FOR VERY HIGH
DIMENSIONAL SIMULATIONS

BY

ART B. OWEN

TECHNICAL REPORT NO. 508
JANUARY 1997

PREPARED UNDER THE AUSPICES
OF
NATIONAL SCIENCE FOUNDATION GRANT
DMS94-04594 and THE ALCOA FOUNDATION

DEPARTMENT OF STATISTICS
STANFORD UNIVERSITY
STANFORD, CALIFORNIA
Latin Supercube Sampling for Very High Dimensional Simulations

By
Art B. Owen

Technical Report No. 508
January 1997

Prepared Under the Auspices
Of
National Science Foundation Grant DMS94-04594 and The Alcoa Foundation

Department of Statistics
Stanford University
Stanford, California
Latin Supercube Sampling for Very High Dimensional Simulations

Art B. Owen
Stanford University
January 1997

Abstract

This paper considers two extensions of Latin hypercube sampling, for very high dimensional problems. The first extension is to Latin hypercube sampling of the infinite dimensional unit cube. The second extension is to Latin supercube sampling (LSS), in which groups of input variables receive identical run order permutations, independent of the permutations applied to other groups. LSS is motivated by a desire to extend Quasi-Monte Carlo (QMC) methods into ever higher dimensions. In high dimensions, QMC can lose effectiveness and in high enough dimensions, it can be difficult even to construct QMC points. One solution is to use QMC on a few variables and ordinary Monte Carlo (MC) on the others. LSS allows one to use several sets of QMC sampled variables in a given problem, thus helping to extend the reach of QMC methods into higher dimensional problems. When used with randomized QMC (RQMC) methods, LSS allows data based estimates of error.
1 Introduction

The fundamental problem we consider here is to compute the value of

\[ I = \int_{[0,1]^d} f(X) dX. \]

This problem can include, via change of variable, integration over nonrectangular regions, integration with respect to non-uniform probability distributions, and reformulations designed to improve accuracy, such as importance sampling or periodization. Some simulations described below can be cast in this form with \( d = \infty \).

For small dimensions \( d \) and smooth integrands \( f \), classical techniques such as those of Davis and Rabinowitz [6] provide an excellent solution. For moderate dimensions, iterated versions of the classical techniques may work well. These are justified by Fubini's theorem and are typically constructed as tensor products of univariate or low dimensional rules.

A working definition of a "high dimensional" problem, is one in which such iterated integration methods are computationally infeasible. This can depend somewhat on \( f \) and on the computer at hand, but large changes in computing power are required to make even small increases in the point at which dimension becomes large. For high dimensional problems, simulation methods such as Monte Carlo (MC) and equidistribution or Quasi-Monte Carlo (QMC) are used.

Theory has it that QMC is more accurate than MC at least for enormous simulations, and numerical experiments often find the advantage appears at practical sample sizes. But, the advantages of QMC have often been observed to diminish as \( d \) increases. See Morokoff and Caflisch [20, 21, 22]
for a discussion of this point.

Randomized Quasi-Monte Carlo (RQMC) methods are hybrids that combine the strengths of MC and QMC. RQMC methods inherit the accuracy of the underlying QMC methods, and sometimes improve it due to a random cancellation of errors. The primary motivation for RQMC is to get the accuracy of QMC with sample based error bounds constructed by MC replication.

For a large enough dimension $d$, it becomes difficult or impossible to even construct a QMC point set, having meaningful equidistribution properties. A working definition of a "very high dimensional" problem, is one for which QMC constructions either are not practically computable or are not especially equidistributed for practical sample sizes.

Some examples of very high dimensional simulations are given next. This section then concludes with an outline of the article.

1.1 Examples of very high dimensional problems

1.1.1 Transport Simulations

Particle transport simulations are used in the design of radiation shields for nuclear reactors. See Spanier [39]. The trajectories of a large number of particles are simulated as they transit from a source towards a target possibly undergoing a sequence of collisions. Each collision may take 6 or 7 random numbers to describe changes in the particles' positions, velocities and energies and whether they are absorbed. To simulate $k$ collisions thus takes $d = 6k$ or $7k$. Since there is no a priori upper bound on $k$ these problems can be considered infinite dimensional, even though any given particle only undergoes a finite number of collisions.
Light transport problems, arising in computer graphics, have a similar flavor. One can follow a photon from a light source until it splats on the imaging plane, possibly after a number of reflections. See Keller [18] and Guibas and Veach [45].

Solving Laplace's equation with a boundary constraint arises in the design of semiconductors [35]. One approach due to Kakutani [17] is to simulate Brownian paths from a point in the region until they reach the boundary. The solution is the average of the boundary values reached by the paths. The simulation proceeds in a series of steps each of which takes a particle to the edge of a bounding box within the boundary region. There is no a priori limit on the number of steps one particle might require, so again $d = \infty$.

1.1.2 Financial valuation

The value of some financial derivatives depends on a whole time series of random fluctuations, in a way that cannot be replaced by a closed form expression. For example, collateralized mortgage obligations [4, 32, 33] depend on the interest rate at 360 future time periods (for monthly payments on 30 year mortgages) and on the fraction of mortgage holders prepaying in each of those 360 time periods. Thus $d = 360$ if the prepayment levels are modelled as a deterministic function of interest rates and otherwise $d > 360$.

Similarly, Asian options depend on the average value of a security over a number of time points and some options involve multiple correlated securities at a number of future time points. See Boyle, Broadie and Glasserman [2] and Joy, Boyle and Tan [16] for examples and further references.
1.1.3 Ergodic simulations and transients

Some simulations are conducted by following only a single sample path for a very long run. For such simulations to work, the problem must have an ergodic property such that one long run simulation converges to the ensemble average of many runs. The methods described here are not aimed at this problem per se, but can be of use in studying transient phenomena.

For example, suppose that a queue starts off empty and we want to know the average amount of time required to reach half of its capacity for the first time. It may be better here to simulate a large number of initially empty queues than to sample a single queue (and wait for the initial conditions to recur). Once again \( d = \infty \).

1.2 Outline of this article

Section 2 reviews MC and QMC integration methods and some hybrids combining the best features of both. Section 3 presents an ANOVA (analysis of variance) decomposition of square integrable functions over \([0,1]^d\) for \( d < \infty \). This decomposition helps to explain for which integration problems QMC might be expected to improve on MC. Section 4 reviews Latin hypercube sampling for \( d < \infty \). Latin hypercube sampling may be thought of as a hybrid of MC and QMC methods. Section 5 describes the QMC technique known as lattice sampling and some randomized hybrids of it. Section 6 describes the QMC methods known as \((t,m,s)\)-nets and \((t,s)\)-sequences and some randomized hybrids of them.

One way to handle very high dimensional integrals is to select a subset \( s \) of the input variables, use QMC or RQMC on them and "pad out" the other \( d - s \) columns of the input matrix by some other method. This technique is
described in Section 7. The difficult issues are deciding how many variables to embed in the QMC method and which ones. Techniques for reformulating the integration problem to enhance the importance of a small number of input variables are considered.

Section 8 proposes a method in which several different sets of s input variables can be handled by (R)QMC. The method is called Latin supercube sampling (LSS) because it randomizes the run order in blocks of input variables in the way that Latin hypercube sampling randomizes one input variable at a time.

Section 9 considers the case where \( d = \infty \). Using martingale truncations, an ANOVA decomposition is developed for square integrable functions on \([0, 1)^\infty\), and Latin hypercube sampling for \( d = \infty \) is studied.

Section 10 considers the accuracy of Latin supercube sampling. When used with effective (R)QMC methods it can be expected to be at least as good as MC and possibly much better when the input variables within (R)QMC block are well chosen. Some conclusions are given in Section 11.

2 Monte Carlo, Quasi-Monte Carlo and hybrids

All of the methods we consider in this paper estimate the integral \( I \) by

\[
\hat{I} = \hat{I}_n = \frac{1}{n} \sum_{i=1}^{n} f(X_i)
\]

for carefully chosen points \( X_i \in [0, 1)^d \). That is, we do not consider here the effects of weighting the observations unequally. Some importance sampling and periodization techniques appear to weight the observations, but can be written as in (1) by subsuming the weight into \( f \).
2.1 Monte Carlo

The simplest Monte Carlo method for estimating $I$ takes $n$ points $X_i$ independently drawn from the uniform distribution on $[0, 1)^d$. In practice this is almost always approximated by deterministic points taken from a pseudorandom number generator, but we will analyze the pseudorandom points as though they were genuinely random.

Under Monte Carlo sampling, and mild conditions on $f$, the estimator $\hat{I}$ is a random variable with expectation $I$ and variance $\sigma^2/n$ where $\sigma^2 = \int (f(X) - I)^2 dX$. Thus the error in $\hat{I}$ is of order $n^{-1/2}$ in probability. Classical methods can achieve rates much better than $n^{-1/2}$, for small $d$ and well behaved $f$, but the MC rate is remarkable in that it holds for all dimensions with only weak conditions on $f$.

2.2 Quasi-Monte Carlo

The accuracy of MC is adversely affected by gaps and clusters that arise by chance among the $X_i$. Equidistribution, or Quasi-Monte Carlo methods use deterministic lists of points $X_1, \ldots, X_n$ that are constructed to avoid, to the extent possible, gaps and clusters. For details on QMC methods see the monograph by Niederreiter [25].

To quantify the uniformity of a list of points, one uses a distance between the continuous uniform distribution on $[0, 1)^d$ and the discrete uniform distribution taking $X_i$ with probability $1/n$ for $i = 1, \ldots, n$. The most widely studied distance measure is the star discrepancy

$$D_n^* = D_n^*(X_1, \ldots, X_n) = \sup_{0 \leq c_j < 1} \left| \prod_{j=1}^d c_j - \frac{1}{n} \sum_{i=1}^n \prod_{j=1}^d 1_{0 \leq X_i^j < c_j} \right|$$ (2)
which compares the continuous and discrete uniform distributions on hyper-rectangles with one corner at the origin. Some other discrepancy measures appear in [25] and Hickernell [13] has further generalized discrepancy.

Star discrepancy is related to integration accuracy by the Koksma-Hlawka inequality

\[
|\hat{I} - I| \leq D_n^*(X_1, \ldots, X_n)V_{HK}(f)
\]

where \(V_{HK}(f)\) is the total variation of \(f\) in the sense of Hardy and Krause. See [25] for the definition of \(V_{HK}\), and [13] for analogues of the Koksma-Hlawka inequality with other notions of discrepancy.

It is possible to construct an infinite sequence of points \(X_1, X_2, \ldots\) along which \(D_n^* = O(n^{-1} (\log n)^d)\). This proves that one can achieve an asymptotic rate better than that of MC, at least for integrands with \(V_{HK}(f) < \infty\). Even for \(d = 10\), it requires impractically large \(n\) to make \((\log n)^d\) negligible compared to \(n\), suggesting that the benefit of QMC may be an illusion. Empirical studies suggest that QMC is more accurate than MC on some real problems with practical sample sizes. The superiority of QMC over MC appears to take greater \(n\) to set in when \(d\) is large. See Morokoff and Caflisch [21, 22], Sarkar and Prasad [36] and van Rensburg and Torrie [44] on these issues.

2.3 Randomized quasi-Monte Carlo

A serious drawback with QMC methods is that there is no practical way to estimate the size of \(\hat{I} - I\) from the function evaluations \(f(X_1), \ldots, f(X_n)\). Estimating \(V_{HK}\) from data appears to be extremely difficult, and in any event, the inequality (3) can be quite conservative. In fact the error bound in (3) increases with \(n\) until \(n \geq \exp(d)\).
By comparison, in the basic MC method above, the estimate

\[ s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (f(X_i) - \hat{f})^2 \]

has expected value \( \sigma^2 \) and the central limit theorem allows one to construct approximate confidence intervals for \( I \). For more complicated MC methods, one can replicate the whole procedure independently a number of times.

Hybrid methods have been developed to combine the best of MC and QMC. A survey of some of these methods is given in [29]. The general approach is to introduce some randomness into a QMC procedure, while retaining the equidistribution properties of the QMC method. If the resulting estimate \( \hat{f} \) is unbiased, then by using several independently randomized estimates one can use standard statistical methods to combine the estimates and estimate the variance of the result.

3 Functional ANOVAs

This section describes some ways to decompose the integrand \( f \) into a sum of simpler functions. We begin with notation. The set \( \mathcal{A} = \{1, 2, \ldots, d\} \) denotes the coordinate axes of \([0, 1]^d\). We assume here that \( d < \infty \), treating the infinite dimensional case in Section 9. The letter \( u \) denotes a subset of \( \mathcal{A} \), \( |u| \) is the cardinality of \( u \) and \( -u \) is the complement of \( u \) with respect to \( \mathcal{A} \). We use these subsets as superscripts: \([0, 1]^u\) denotes the space of values for components of \( X^j \) with \( j \in u \), \( X^u \) denotes the coordinate projection of a point \( X \) onto \([0, 1]^u\) and, in integrals, \( dX^u = \prod_{j \in u} dX^j \). The case \( u = \emptyset \) can require special attention, either by specifying a convention, or by restricting some operations to \(|u| > 0\).
Under the mild conditions that \( \int f(X)^2 dX < \infty \) and \( f \) is measurable, we can write \( f(X) \) as a sum of \( 2^d \) orthogonal functions, one for each subset of the input axes

\[
f(X) = \sum_{u \subseteq \{1,2,\ldots,d\}} f_u(X^u). \tag{4}
\]

Here \( f_u \) only depends on the components of \( X \) in \( X^u \), though for convenience we may write it as a function \( f_u(X) \) that is constant over values of \( X^j \) with \( j \neq u \).

The ANOVA decomposition is orthogonal in that \( \int f_u(X) f_v(X) dX = 0 \) whenever \( u \neq v \). The functions \( f_u \) are defined recursively by

\[
f_u(X^u) = \int_{Z^u = X^u} \left( f(Z) - \sum_{v \subsetneq u} f_v(X^u) \right) dZ^{-u} \tag{5}
\]

where the sum is over strict subsets \( v \neq u \). Examples make equation (5) clearer. First of all \( f_\emptyset = \int f(Z) dZ = I \). Then \( f_{\{i,j\}}(X^j) = \int_{Z^j = X^j} (f(Z) - I) \prod_{k \neq j} dZ^k \) and so on. The function \( f_{\{i\}}(X^j) \), called the “main effect of \( X^j \)” can be thought of as the average effect of the \( j \)th variable on the response function \( f(X) \). The function \( f_{\{i,j,k\}}(X^{\{i,j,k\}}) \) describes the joint effect, or interaction, of variables \( X^j \) and \( X^k \) on the response, beyond their individual contributions \( f_{\{i\}} \) and \( f_{\{k\}} \). If these variables have a strong synergistic (or antagonistic) effect on \( f \) then their interaction will be large. More generally \( f_u \) is the \(|u|\)-factor interaction among variables \( X^u \).

The term ANOVA is an acronym for “analysis of variance”. The decomposition is completely analogous to the one used in experimental statistics, for functions over Cartesian products of finite sets. The variance being an-
alyzed is \( \sigma^2 = \int (f(X) - I)^2 dX \) and one easily finds that

\[
\sigma^2 = \sum_{|u|>0} \sigma_u^2
\]

where \( \sigma_u^2 = \int f_u(X^u)^2 dX^u \). This ANOVA decomposition for functions is due to Hoeffding. Takemura [41] gives a history. Notable contributions were made by Efron and Stein [8], Stein [40] and Wahba [46]. The notation used here is based on Owen [27]. Hickernell [13] has proposed some generalizations.

The best (in mean square) constant approximation to \( f \) is \( I = f_\emptyset \). Stein [40] notes that the best additive approximation to \( f \) is

\[
f_{\text{add}} = I + \sum_{j=1}^d f_{\{j\}}(X^j).
\]

Furthermore the best approximation to \( f \) using sums of functions depending on \( m \) or fewer components is \( \sum_{|u| \leq m} f_u(X^u) \), and the best approximation to \( f \) using variables contained in the set \( u \) is \( \sum_{u \subseteq u} f_u \). We note for later use that

\[
\int_{[0,1]^u} f(X) dX^u = \sum_{u \subseteq u} f_u(X).
\]

For the integration rule given by (1) we have \( \hat{I} = \sum_u \hat{f}_u \) where

\[
\hat{f}_u = \frac{1}{n} \sum_{i=1}^n f_u(X_i).
\]

Easily \( \hat{f}^\emptyset = I \) and so \( \hat{I} - I = \sum_{|u|>0} \hat{f}_u \). For QMC methods Caflisch, Morokoff
and Owen [4] note that

$$|\hat{I} - I| \leq \sum_{|u| > 0} |\hat{I}^u| \leq \sum_{|u| > 0} D_n^u V_{HK}(f_u),$$

(8)

where $D_n^u$ is the $|u|$ dimensional star discrepancy of $X_1^u, \ldots, X_n^u$ and $V_{HK}(f_u)$ is the Hardy-Krause variation of $f_u$ over $[0,1]^u$. QMC methods that match large components $f_u$ with small discrepancies $D_n^u$ might be expected to work well. For example, QMC methods often have small $D_n^u$ for small $|u|$ and moderate $n$. They should therefore work well on integrands dominated by low dimensional structure, even if $D_n^*$ itself never becomes much smaller than it would under MC for feasible $n$.

3.1 Effective dimension


**Definition 1** The effective dimension of $f$, in the superposition sense, is the smallest integer $d_S$ such that $\sum_{0 < |u| \leq d_S} \sigma_u^2 \geq 0.99 \sigma^2$.

A method with good uniformity in every $d_S$ dimensional projection of $X_1, \ldots, X_n$ can be expected to work well with functions of effective dimension $D_S$, in the superposition sense.

**Definition 2** The effective dimension of $f$, in the truncation sense, is the smallest integer $d_T$ such that $\sum_{u \subseteq \{1,2,\ldots,d_T\}} \sigma_u^2 \geq 0.99 \sigma^2$.

A method with good uniformity in the first $d_T$ input variables of $X_1, \ldots, X_n$ can be expected to work well with functions of effective dimension $D_T$, in the truncation sense.
To illustrate the difference, a linear function has $d_S = 1$ but can have $d_T = d$. Clearly the cutoff threshold 0.99 is arbitrary and could be replaced by another.

4 Latin hypercube sampling

For one dimensional integration, the midpoint rule takes

$$A_i = \frac{i - 0.5}{n}, \quad i = 1, \ldots, n.$$ (9)

If we view this as a one dimensional quasi-Monte Carlo sampling scheme, the following stratified sampling method

$$A_i = \frac{i - V_i}{n}, \quad i = 1, \ldots, n$$ (10)

where the $V_i$ are independent $U(0,1]$ random variables, can be thought of as a randomized QMC method. In (10) each interval $[(k - 1)/n, k/n)$ for $k = 1, \ldots, n$ has exactly one uniformly randomly located point in it.

A variant of Latin hypercube sampling can be constructed from $d$ midpoint rules, by randomizing their run order

$$X_i^j = A_{\pi_j(i)} = \frac{\pi_j(i) - 0.5}{n}. \quad (11)$$

The $\pi_j$ are independent uniform random permutations of the integers $1, \ldots, n$. That is, every one of $n!$ possible permutations has an equal chance of being used.

The original definition of Latin hypercube sampling of McKay, Conover and Beckman [19], was based on randomizing the run order of $d$ indepen-
dently stratified samples $A^j$

$$X^j_i = A^j_{\pi_j(i)} = \frac{\pi_j(i) - U^j_i}{n} = \frac{\pi_j(i) - U^j_i}{n}. \quad (12)$$

Here $U^j_1, \ldots, U^j_n$ are $V^j_1, \ldots, V^j_n$ in random order, and so the $U^j_i$ are also independent $U(0,1)$ random variables. The centered version in equation (11) was originally due to Patterson [34] in the setting of agricultural experiments, while the version in equation (12) was motivated by computer experiments.

In Latin hypercube sampling each coordinate projection of the $X_i$ is nearly uniform. It should therefore be no surprise that when $f$ is nearly a sum of univariate functions of the coordinates of $X$, that Latin hypercube sampling does well.

Under Latin hypercube sampling (12), $\hat{I}$ is a random variable with $E(\hat{I}) = I$ and

$$V_{LHS}(\hat{I}) = \frac{1}{n} \int (f(X) - f_{\text{add}}(X))^2 dX + o\left(\frac{1}{n}\right)$$

$$= \frac{1}{n} \left(\sigma^2 - \sum_{j=1}^d \sigma^2_j\right) + o\left(\frac{1}{n}\right) \quad (13)$$

where $f_{\text{add}}$ is the additive function closest to $f$ in mean square. See Stein [40] for details.

If the integrand is not nearly a sum of univariate functions, then Latin hypercube sampling cannot be expected to improve upon Monte Carlo. But it never does much worse than Monte Carlo either. For any $f$ with $\int f^2 dX < \infty$, we have

$$V_{LHS}(\hat{I}) \leq \frac{n}{n-1} V_{MC}(\hat{I}). \quad (14)$$

Latin hypercube sampling with $n > 1$ observations is never worse than
Monte Carlo sampling with \( n - 1 \) observations. See Owen [30] for details.

5 Lattice Methods and Randomizations

The definitive reference on Lattice methods is the book by Sloan and Joe [37]. The simplest form of Lattice method is the method of good lattice points, or the number theoretic method. Another good reference for these points is Hua and Wang [14] and Fang and Wang [9] discuss applications in statistics.

In the method of good lattice points,

\[
X_i^j = \left\{ \frac{i \cdot g_j}{n} \right\}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, s \tag{15}
\]

where \( \{z\} = z - [z] \) is the fractional part of \( z \) (or "\( z \) modulo 1") and \( g_j \) are integers. The points \( X_i \) belong to a lattice. A judicious choice for \( n \) and the \( g_j \) can produce a sequence of points with good equidistribution properties in the unit cube, hence the term "good" in "good lattice points".

Good lattice points and their generalizations are extremely well suited to integrating periodic functions with rapidly decaying Fourier coefficients. Combined with techniques for replacing non-periodic integrands by periodic ones having the same integral they are one of the leading families of Quasi-Monte Carlo integration methods.

5.1 Cranley-Patterson Rotations

Cranley and Patterson [5] randomize the good lattice points as follows. Let \( A_i^j \) be the points of a good lattice point for \( 1 \leq i \leq n \) and \( 1 \leq j \leq s \). Let
$U^j$ be independent $U[0, 1)$ random variables. Then let

$$X^j_i = \{A^j_i + U^j\}. \tag{16}$$

If one maps the unit interval $[0, 1)$ onto the unit circle by identifying the endpoints, the result of the Cranley-Patterson randomization is to rotate the $n$ points $X^j_1, \ldots, X^j_n$ to the right by a distance of $U^j$, with wrap-around. Taking $r$ replicates of these $n$ points and letting $r \to \infty$ produces errors at the Monte Carlo error rate $O((nr)^{-1/2})$ but perhaps with a much more favorable constant than has simple Monte Carlo.

Joe [15] uses the same randomization on more general lattice rules.

5.2 Orthogonal rotations

The Cranley-Patterson rotations are natural for periodic functions in the unit cube. However sometimes the points $X^j_i$ are transformed from independent uniform random variables to independent $N(0, 1)$ random variables $A^j_i = \Phi^{-1}(X^j_i)$, where $\Phi$ is the normal distribution function, prior to use. At the level of the $A_i$, there is no special reason to prefer randomizations that use coordinate rotations. One might instead produce a random orthogonal matrix $Q$ (see Devroye [7]) and apply it to the rows of $A$ to produce row vectors $Z_i = A_iQ$.

This technique gets expensive if $s$ is large because computing $Q$ takes $O(s^3)$ work.
6 Nets and Randomizations

We begin with more notation. We describe nets as sequences of points in the cube \([0,1)^s\), where usually \(s = d\). When \(s > d\) one simply selects \(d\) of the dimensions to use. When \(s < d\) one can apply the compromise methods of Sections 7 and 8.

The integer \(b \geq 2\) is used throughout as a base for representing points in \([0,1)\). Thus \(X_i^j = \sum_{k=1}^{\infty} x_{ijk} b^{-k}\) where \(x_{ijk}\) are integers with \(0 \leq x_{ijk} < b\).

6.1 \((t,m,s)\)-nets, \((t,s)\)-sequences and \((\lambda,t,m,s)\)-nets.

Here we describe equidistribution methods known as \((t,m,s)\)-nets and \((t,s)\)-sequences. These have been developed by Sobol', Faure, and Niederreiter and a comprehensive discussion of them appears in the monograph by Niederreiter [25] on which this subsection is based. We also make two minor variations on Niederreiter’s terminology in Definitions 3 and 4.

An elementary interval of \([0,1)^s\) in base \(b\) is defined as a set of the form

\[
E = \prod_{j=1}^{s} \left[ \frac{t_j}{b^{k_j}}, \frac{t_j + 1}{b^{k_j}} \right]
\]

for nonnegative integers \(k_j\) and \(t_j < b^{k_j}\). Tezuka [42] uses the term “\(b\)-ary box” for “elementary interval in base \(b\)”, and we abbreviate this further to “\(b\)-box”.

The \(b\)-box \(E\) in (17) is a hyperrectangle of volume \(b^{-m}\) where \(m = \sum_{j=1}^{s} k_j\). In addition to the volume of a \(b\)-box, the “effective dimension” of the \(b\)-box is a useful quantity.
Definition 3 The effective dimension of the $b$-box $E$ defined in (17) is

$$\delta = \delta(E) = \sum_{j=1}^{s} k_j > 0.$$ 

For a point $X$ to belong to $E$ requires nontrivial constraints on exactly $\delta(E)$ of the coordinates $X^j$. The effective dimension of the $b$-box is the same as the effective dimension of the function equal to one in that $b$-box and zero outside of it (in the superposition sense, possibly requiring a threshold higher than 0.99). The unit cube $[0,1)^s$ is the unique $b$-box of effective dimension 0. No $b$-box of effective dimension $\delta$ can have volume larger than $b^{-\delta}$.

Ideally each $b$-box of volume $V$ should have $nV$ points of the integration rule. Let $t \geq 0$ and $m \geq 0$ be integers. A finite sequence $X_1, \ldots, X_n \in [0,1)^s$ with $n = b^m$ is a $(t,m,s)$-net in base $b$ if every $b$-box of volume $b^{t-m}$ contains exactly $b^t$ points of the sequence. The net property starts to become relevant at $n = b^{t+1}$ where it constrains the equidistribution over some $b$-boxes of effective dimension 1. It takes at least $n = b^{t+s}$ points before the net property applies to any $b$-boxes of effective dimension $s$.

The infinite sequence $X_1, X_2, \ldots \in [0,1)^s$ is a $(t,s)$-sequence in base $b$ if for all $m \geq 0$ and all $k \geq 0$ the finite sequence $X_{kb^m+1}, \ldots, X_{(k+1)b^m}$ is a $(t,m,s)$-net in base $b$. Niederreiter [25] discusses existence and construction of $(t,m,s)$-nets and $(t,s)$-sequences.

Smaller values of $t$ imply better equidistribution properties for both $(t,m,s)$-nets and $(t,s)$-sequences. An advantage of using nets taken as the first $n = b^m$ points of a $(t,s)$-sequences is that one can later increase $n$ through a sequence of values $n = \lambda b^m$, $1 \leq \lambda < b$, and find that all of the points used in $I_{\lambda b^m}$ are also used in $I_{(\lambda+1)b^m}$. As $n$ increases through this sequence of values, any $b$-box of volume $V$ eventually contains $nV$ of the
points, and once such a $b$-box is balanced this way, it remains balanced as $n$ increases.

The initial $\lambda b^m$ points of a $(t, s)$-sequence are well equidistributed but are not a $(t, m, s)$-net, unless $\lambda$ is a power of $b$. Owen [30] introduces the following definition to describe such point sets.

**Definition 4** Let $m, t, \lambda$ be integers with $m \geq 0$, $0 \leq t \leq m$, and $1 \leq \lambda < b$. A sequence of $\lambda b^m$ points in $[0, 1]^s$ is called a $(\lambda, t, m, s)$-net in base $b$ if every $b$-box of volume $b^{t-m}$ contains $\lambda b^t$ points of the sequence and no $b$-box of volume $b^{t-m-1}$ contains more than $b^t$ points of the sequence.

Numerical integration by averaging over the points of a $(t, m, s)$-net can have an error of order $n^{-1}(\log n)^{s-1}$, for functions of bounded variation in the sense of Hardy and Krause. See Niederreiter [25] for this result and some related ones. The rate attained along a fixed $(t, s)$-sequence is $n^{-1}(\log n)^s$.

### 6.2 Base $b$ scrambling of the unit cube

The upper bound from the Koksma-Hlawka inequality is very hard to estimate, and is usually quite conservative, since it applies to the worst possible integrand $f$ for the given points $X_1, \ldots, X_n$. For this reason Owen [29] suggested randomizing the points of a $(t, m, s)$-net or $(t, s)$-sequence. By independently repeating the randomization and noting how the resulting answers differ, it is possible to judge statistically the accuracy of an answer.

For this to work it is necessary that the randomization preserve the $(t, m, s)$-net or $(t, s)$-sequence structure. Below is a geometric description of the scrambling method. For an algebraic description see [29, 30].

Begin by partitioning the unit cube $[0, 1]^s$ along the $X^1$ axis into $b$ parallel $b$-boxes of the form $[a/b, (a+1)/b) \times [0, 1]^{s-1}$ for $a = 0, \ldots, b-1$. Then
randomly shuffle those $b$-boxes replacing them in one of the $b!$ possible orders, each such order having probability $1/b!$. Next take each such $b$-box in turn, partition it into $b$ congruent $b$-boxes of volume $b^{-2}$ along the $X^1$ axis, and randomly shuffle those boxes. Then repeat this process on $b^2$ $b$-boxes of volume $b^{-3}$, $b^3$ $b$-boxes of volume $b^{-4}$ and so, ad infinitum. In practice this can stop when the $b$-boxes are narrow compared to machine precision. The full scrambling involves applying the above operations along the other $s - 1$ axes $X^2, \ldots, X^s$ as well. All of the many permutations used are to be statistically independent, and in practice they are generated through pseudo-random numbers.

If $A$ or $A_i$ is a point in $[0,1)^s$, let $X$ or $X_i$ be that point's location after the scrambling. A sequence $(X_i)$ thereby inherits certain equidistribution properties of a sequence $(A_i)$ and the individual points in it are uniformly distributed on $[0,1)^s$. Owen [29, 30] proves the following two propositions.

**Proposition 1** If $(A_i)$ is a $(\lambda, t, m, s)$-net in base $b$ then $(X_i)$ is a $(\lambda, t, m, s)$-net in base $b$ with probability 1.

**Proposition 2** Let $A$ be a point in $[0,1)^s$ and let $X$ be the scrambled version of $A$ as described above. Then $X$ has the uniform distribution on $[0,1)^s$.

A consequence of Proposition 2 is that $\hat{I}$ is a random variable with expectation $I$. This holds even if the underlying $A_i$ are not points of a net. Because of Proposition 1, any theorems describing the accuracy of $\hat{I}$ based on a $(\lambda, t, m, s)$-net with points $A_i$ also holds for the scrambled points $X_i$.

The randomization in scrambled nets can lead to improved accuracy due to some error cancellation. This appeared to be the case in the simulated
examples in [29] and in [31] it is shown that

\[ V_{PNET}(\hat{I}) = O(n^{-3}(\log n)^{s-1}) \]  \hspace{1cm} (18)

under mild smoothness conditions on \( f \). Thus the typical error is of order \( n^{-3/2}(\log n)^{(s-1)/2} \), an improvement on the rate attained by unrandomized nets.

The variance in scrambled net simulation satisfies

\[ V_{PNET}(\hat{I}) = \sum_{|u| > 0} V_{PNET}(\hat{I}^u). \]  \hspace{1cm} (19)

The results in [31] suggest that \( V_{PNET}(\hat{I}^u) \) may become appreciably smaller than \( V_{MC}(\hat{I}^u) \) at around \( n = b^{+|u|} \). Moreover \( V_{PNET}(\hat{I}^u) \leq 2.7183 V_{MC}(\hat{I}^u) \) for any scrambled net with \( t = 0 \) in any dimension \( s \geq 1 \) with any integrand \( f \), so least favorable integrands can’t make randomized nets much worse than simple Monte Carlo.

### 6.3 Cranley-Patterson Randomization

It is also possible to apply the Cranley-Patterson randomization of Section 5.1 to points from \((t, m, s)\)-nets and \((t, s)\)-sequences. See Tuffin [43] for details. This randomization does not preserve equidistribution over \( b \)-boxes, but it does produce unbiased estimates \( \hat{I} \) whose variance can be estimated by replication.

### 6.4 Nets in very high dimensions

There are limitations to the applicability of nets in very high dimensions. Consider \( d = 50 \). For a \((0, s)\)-sequence to be usable requires \( s \geq 50 \), which in
turn requires \( b \geq 50 \). Niederreiter [24] gives constructions of \((0,s)\)-sequences for prime powers \( b \geq s \). The natural choice for \( d = 50 \), is the smallest one, \( b = 53 \). Because 53 is a prime, the resulting sequence is a Faure sequence [10]. Natural sample sizes with such a base are powers of 53. The superior asymptotic rate of convergence of these may be expected to set in after \( n = 53^{50} \approx 1.64e86 \). Unless \( n \geq 53^{2} = 2809 \) the net does not balance any \( b \)-box not balanced by Latin hypercube sampling.

Another approach is to use a \((t,s)\)-sequence with smaller \( b \), and in the tradeoff take a larger \( t \). The most widely studied version of this strategy takes \( b = 2 \), the smallest possible base. For \( b = 2 \) and \( s = 50 \), the presently best possible value of \( t \) is 77, by a construction of Niederreiter and Xing [23]. The superior asymptotic rate of convergence with this sequence may be expected to set in after \( n = b^{t+s} = 2^{127} \approx 1.70e38 \). This is the smallest value of \( n \) for which we can partition the unit cube into \( b \)-boxes of effective dimension 50 and be sure that each such box has at least one point in it.

Widely used software for generating nets has a limit on the dimension. For instance, the code of Bratley, Fox and Niederreiter [3] has by default a limit of \( d = 12 \).

The asymptotic advantage of nets, scrambled or otherwise, in dimension 50 appears to take an impractically large number of observations to set in, and the matter is worse for \( d = 1000 \) or \( d = \infty \). Yet as described in Section 11 and seen in [4, 32, 33] it is possible for high dimensional functions to be integrated accurately. This may be due to low effective dimension of the high dimensional functions.
7 Padding Techniques

The full input sequence for the simulation is a \( n \) by \( d \) matrix \( X_i^j \) of numbers between 0 and 1. An (R)QMC method may be used for \( s \) of the columns of this matrix. Some simple techniques below may be used to "pad out" the matrix.

7.1 Padding by Monte Carlo

A natural solution is to pad out the input matrix with independent \( U[0, 1] \) random variables. Suppose for example, that \( A_i \) are the points of an (R)QMC method in \( s \) dimensions. Then one might take \( X_i^j = A_i^j \) for \( 1 \leq j \leq s \) and \( X_i^j = U_i^j \) for \( j > s \) where \( U_i^j \) are independent \( U[0, 1] \) random variables, independent of the randomization, if any, used in constructing \( A \). This approach is taken by Spanier [39] and Okten [26] studies the resulting discrepancy.

The success of this method will depend in part on making a good choice of which \( s \) variables to use (R)QMC on. One might try to identify the most important subset \( u \) of \( s \) variables, perhaps by their ANOVA contribution \( \sum_{0 < |v|, u \leq v} \sigma_v^2 \). These are not necessarily the first \( s \) variables in any natural order, and it may take subject matter knowledge, guesswork and experimentation to make a good choice. A more realistic goal may be to pick those variables with a large value of \( \sum_{0 < |v| \leq m, u \leq v} \sigma_v^2 \), where \( m \) is the largest dimensionality in which good equidistribution may be expected of the projected (R)QMC points.

7.2 Padding by Latin hypercube sampling

It is possible to pad out (R)QMC in a way that is better than simple Monte Carlo. For any choice \( n \), one can pad with a Latin hypercube sample of \( n \)
rows and $d - s$ columns. This is even true if $d > n$. In simulations with large enough $d$ ($d$ can be infinite) it may take a “sampling without replacement” trick to generate only those rows of the column $X^j$ that the simulation really uses. Owen [28] gives an example in which a randomized orthogonal array sample is padded out with a Latin hypercube sample.

Padding with Latin hypercube samples instead of independent samples improves the accuracy of integration for the main effects in the $d - s$ variables not sampled by a net. In principle, this may make one want to change which subset $u$ of $s$ variables to apply QMC to. Consider a variable that has an enormous but purely additive effect on $f$. That variable is handled poorly by Monte Carlo but Latin hypercube sampling handles it as well as most (R)QMC methods do. If one is padding by MC then this variable should be among the QMC variables, but if one is padding by LHS then it should not be among the QMC variables. A natural choice would be the subset $u$ for which $\sum_{1 < |v|, v \subseteq u} \sigma^2_v$ or perhaps $\sum_{1 < |v| \leq m, v \subseteq u} \sigma^2_v$ is maximized.

A disadvantage of Latin hypercube padding is that a simulation with a given value of $n$ might not extend easily to a larger value of $n$. One remedy, used by Caflisch, Morokoff and Owen [4] is to use $r$ independent Latin hypercube samples of $n'$ rows and $d - s$ columns where $n = rn'$. Using blocks of Latin hypercube samples allows one to select sample sizes that are multiples of the block size $n'$. It makes sense to choose a block size which is a divisor of the natural sample sizes for the (R)QMC method being used. For nets, a power of $b$ seems reasonable.
7.3 Padding with (R)QMC

For simplicity suppose that $d = ks$. It would clearly not work to repeat an $s$ dimensional (R)QMC method $k$ times taking $X_i = (A_i, A_i, \ldots, A_i)$, where $A_i$ are the $s$ dimensional points of an (R)QMC method. For example, the points $X_i^j$ and $X_i^k$ where $j = k \mod s$ would then lie on a diagonal in the $X^{(j,k)}$ plane.

Similarly, using $k$ independent randomizations of a single underlying QMC point set cannot usually be expected to work well in practice either. To see this, consider what becomes of the diagonal line mentioned in the previous paragraph. Unless the randomization manages to smear that line into points uniformly distributed in the unit square, something will go wrong. The randomization of Cranley-Patterson (Section 5.1) shifts the line to a new random location with "wraparound" and the line still has unit slope. The scrambling of Section 6.2 replaces the diagonal by a randomly generated one dimensional point set, which for any $m > 0$ belongs to the union of $b^m$ squares, each of area $b^{-2m}$.

Another tempting trick that fails is to apply various nets in relatively prime bases. One might consider using Faure sequences, $(0, p)$-sequences in base $p$ for primes $p = 2, 3, 5, \ldots$ until the sum of primes used is greater than or equal to $d$. This does with Faure sequences what the Halton sequence does with generalized van Der Corput sequences. But by trying it and plotting the points, it becomes clear that something goes wrong. Faure [11] (Section 6) suggests something similar but reports in a personal communication that he thinks it's not a good idea. As explained by Niederreiter (personal communication), there is no reason to expect this method to work, because unlike the Halton setting, there is no analogue of the Chinese remainder
7.4 Reducing Effective Dimension

When one plans to use (R)QMC points on some inputs and padding for the others, one may be able to do more than simply choose the important variables for QMC. In some cases one can rewrite the integrand in a way that puts a greater amount of the variation into a small number of inputs. For example, when the inputs are used to construct Brownian paths a Brownian bridge encoding as in [4] makes the first few input variables more important than the regular encoding.

Acworth, Broadie and Glasserman [1] use an encoding based on principal components for Brownian motion. The first 5 principal components of Brownian motion explain about 96% of the variation in the Brownian path. Thus it may pay to use an (R)QMC method on these variables but not on the others.

Fox [12] describes some additional settings where hybrids may be applied. For discrete event simulation, Fox suggests defining the arrival process by first drawing the number of arrivals, then the median arrival time, then a sequence of intermediate times. Fox recommends using QMC on the earlier variables and MC for the remainder. Another possibility he considers is recursively splitting the time interval and using binomial sampling to determine how many observations to put in each subinterval.

Yet another recommendation is for simulations requiring finite independent samples $Z_1, \ldots, Z_d$ from some distribution. There Fox suggests using QMC to generate some or all of the order statistics and then MC to generate the remaining order statistics, if any, as well as the random allocation.
of order statistics to sample values.

8 Latin Supercube Sampling

8.1 Introduction

In Latin supercube sampling (LSS) one takes a list of QMC point sets or RQMC point sets and randomizes their run order the same way that Latin hypercube sampling randomizes the run order of the midpoint and stratified techniques.

Let $\mathcal{A} = \bigcup_{r=1}^{k} \mathcal{A}_r$ with $\mathcal{A}_r \cap \mathcal{A}_{r'} = \emptyset$ for $r \neq r'$ be a partition of the simulation variables into $k$ nonempty subsets. Letting $s_r = |\mathcal{A}_r|$ we have $s_r \geq 1$ and $\sum_{r=1}^{k} s_r = d$. Let $\mathcal{X}_i^r \in [0,1)^{A_r}$ for $i = 1, \ldots, n$ and $r = 1, \ldots, k$. In practice these $\mathcal{X}_i^r$ will ordinarily be points of an $s_r$ dimensional (R)QMC method. For $r = 1, \ldots, k$ let $\pi_r(i)$ be independent uniform random permutations of the integers $1, \ldots, n$. Then a Latin supercube sample (LSS) is formed by taking

$$X_i = (\mathcal{X}_{\pi_1(i)}^1, \mathcal{X}_{\pi_2(i)}^2, \ldots, \mathcal{X}_{\pi_k(i)}^k).$$

(20)

Equation (20) assumes that the variables in $\mathcal{A}_r$ come before those in $\mathcal{A}_{r'}$ whenever $r < r'$. This is not at all necessary. The variables can be interleaved in any order, taking $X_i^{A_r} = \mathcal{X}_{\pi_r(i)}$. The ordering assumption is made solely to simplify notation. In cases where some of the $s_r$ are infinite, it becomes problematic to use the notation in (20), a difficulty we ignore.

In words, the first $s_1$ columns in the LSS are obtained by randomly permuting the run order of the QMC points $\mathcal{X}_1^1, \ldots, \mathcal{X}_n^1$, the next $s_2$ columns come from an independent permutation of the run order of $\mathcal{X}_1^2$ and so on.
The best results may be expected if one can arrange that variables that
interact most strongly are grouped into a subset $A_r$. In the extreme case,
suppose that $\sigma_u^2 = 0$ unless $u = \emptyset$ or $u \subseteq A_r$ for some $r$. For simplicity
suppose also that each $s_r = s$. In this case the error in LSS is the sum of $k$
(R)QMC errors, one from each subset of axes. One then has an integration
rule for a $d$ dimensional function, that converges at a rate usually seen in $s$
dimensional problems. More realistically, one might be able to arrange for
most of the variance in the function $f$ to take place within subsets $A_r$ of
axes, that the residual

$$f(X) - I - \sum_{r=1}^{k} \sum_{0 < |u| \subseteq A_r} f_u(X)$$

is small, and that this residual contributes to the integration error at essen-
tially the Monte Carlo rate.

### 8.2 Partitions

The best way to partition axes is problem specific and may take skill to
guess. In addition to the variation considerations above, software engineer-
ing complications arise. It may be much more convenient to group together
variables used within a software module than to group together variables
that span several modules.

For example, in particle transport simulations, it may be best for the
variables that determine the $r$'th collision to be drawn from the same ran-
domization of the (R)QMC points. Or, it might be better to use one
(R)QMC sequence for the change in $x$ component direction for $s$ consecu-
tive collisions, another for the change in $y$ component direction and so
forth. Which is better depends on which input variables "interact most" and this can be expected to depend on what response is being measured.

In a financial simulation with \( k \) Brownian paths, it may make sense to select \( 5 \) principal components of each path, apply an (R)QMC method to each of them with LSS and then pad out the other variables by LHS. Alternatively, it may be better to group the \( k \) first principal components together then the \( k \) second components and so on.

In a discrete event simulation each source of arrivals could receive its own set of (R)QMC variables. Similarly the variables describing service times could be bundled within one or more (R)QMC sets.

9 Infinite Dimensional versions

9.1 Issues when \( d = \infty \)

The infinite dimensional case needs to be treated with some extra care. For example, in finite dimensional Latin hypercube sampling, one may proceed by showing that

\[
V_{LHS}(\hat{I}) = \frac{1}{n} V(f(X_1)) + \frac{n-1}{n} \text{Cov}(f(X_1), f(X_2)).
\]

Then \( \text{Cov}(f(X_1), f(X_2)) = E((f(X_1) - I)E(f(X_2) - I|X_1)) \), and given \( X_1 \) the location of \( X_2 \) is uniformly distributed over a set of volume \((1 - 1/n)^d\). But if \( d = \infty \) this set has volume zero, calling into question averages over it.

An ANOVA on infinitely many dimensions also requires some care, because there are uncountably many subsets of \( \{1, 2, \ldots \} \) among which to partition the variance. Also if \( |u| = \infty \) then the definition of \( f_u \) in equation
(5) involves an uncountable sum over all proper subsets of \( u \). What would we make of an ANOVA effect \( f_u \) for \( |u| = \infty \), such as an interaction among all components \( X^j \) for which \( j \) is prime? Fortunately, we only need to use finite subsets \( u \), and there are only countably many of these. For \( |u| < \infty \) we define \( f_u \) by equation (5) as before.

9.2 Martingale truncation

In the infinite dimensional examples of Section 1.1, one expects that \( f \) should be "almost finite dimensional", in that the first \( s \) dimensions for some possibly large \( s \) should capture virtually all of the important variation in \( f \). This expectation is borne out, whenever \( \int_{[0,1]^{\infty}} f(X)^2 dX < \infty \).

Let \( 1:s \) denote the set \( \{1,2,\ldots,s\} \) of leading variables. There are a number of ways to approximate an infinite dimensional \( f \) by a function of \( X^{1:s} \). One can replace \( X^{s+1},X^{s+2},\ldots \) by their central values 0.5, by the \( f \) minimizing or maximizing values, or by the expectation of \( f(X) \) over \( X^{s+1},X^{s+2},\ldots \) with \( X^{1:s} \) held fixed. This latter approximation is most convenient theoretically because it allows the use of martingale methods. See Williams [47]. We do not give a rigorous treatment of martingales here.

Now define

\[
f^s(x^1,\ldots,x^s) = E(f(X)|X^1 = x^1,\ldots,X^s = x^s),
\]

where expectation is taken over independent \( X^j, j \geq s + 1 \) having a \( U[0,1] \) distribution. When \( 1:s \subseteq u \) the function \( f^s(X^u) \) is taken to mean \( f^s(X^{1:s}) \) ignoring any coordinates in \( 1:s - u \). For \( s = 0 \) we take \( f^0(X^u) = I \) for any \( u \). The sequence \( f^s(X), s \geq 0 \) is a martingale, by Levy's upward theorem (Chapter 14.2 of [47]), when the \( X^j \) are independent \( U[0,1] \) random
variables. The key martingale property is that $E(f^{s+1}(X) | X^{1:s}) = f^s(X)$.

Because we assume $E(f(X)^2) < \infty$ the martingale is bounded in $L^2$. Thus as $s \to \infty$, we have $f^s(X) \to f(X)$ in $L^2$ and pointwise. (The pointwise convergence holds "almost surely", a distinction that we ignore.) For any $\epsilon > 0$ there is a dimension $s < \infty$ for which $E((f^s(X) - f(X))^2) < \epsilon$.

9.3 ANOVA with $d = \infty$

A convenient way to handle infinite dimensional domains is to first truncate the dimension, replacing $f$ by some $f^s$, and then apply a finite dimensional ANOVA to $f^s$. For $u \subseteq 1:s$ let $f^s_u$ be the ANOVA term for $u$ obtained by replacing $f$ by $f^s$ in equation (5). If $u$ is not a subset of $1:s$, take $f^s_u = 0$.

Consider first $u = \emptyset$. For $s \geq 1$,

$$f^s_\emptyset = \int_{[0,1]^s} f^s(X^{1:s}) dX^{1:s} = E(f(X)) = f_\emptyset$$

by the definition (21) of $f^s$. By induction on $|u|$ it becomes clear that $s < \infty$ and $u \subseteq 1:s$ implies $f^s_u = f_u$. Thus leading terms in the ANOVA decomposition of the martingale random variables match those of the original $f(X)$.

For any $\epsilon > 0$ let $s_0$ be such that $E((f^s - f)^2) < \epsilon$ when $s \geq s_0$. It follows that $E((f^s_u - f_u)^2) < \epsilon$ uniformly in $|u| < \infty$ whenever $s \geq s_0$. If $u \subseteq 1:s$ then $f^s_u = f_u$ and otherwise $E((f^s_u - f_u)^2) = E(f_u(X)^2) \leq E(f(X)^2) - E(f^s(X)^2) < \epsilon$.

The ANOVA decomposition of $f$ is now

$$f(X) = \sum_{0 \leq |u| < \infty} f_u(X) \quad (22)$$
with
\[ \sigma^2 = \int (f(X) - I)^2 dX = \sum_{1 \leq |u| < \infty} \int f_u(X)^2 dX. \] (23)

9.4 LHS with \( d = \infty \)

Using the ANOVA decomposition of Section 9.3 we write
\[ \hat{I} - I = \frac{1}{n} \sum_{i=1}^{n} \sum_{1 \leq |u| < \infty} f_u(X_i) = \sum_{1 \leq |u| < \infty} \hat{I}_u. \]

Because each \( X_i \) is individually \( U[0,1)^\infty \) it follows that \( E_{LHS}(\hat{I}) = I \).

Let \( u \neq v \) be finite subsets. Without loss of generality there is some \( j \in u \) with \( j \notin v \). Then \( E(f_u(X_j)f_v(X_{i'})) = 0 \) upon averaging over \( X_i^j \), regardless of whether \( i = i' \). It follows that
\[ V_{LHS}(\hat{I}) = \sum_{1 \leq |u| < \infty} V_{LHS}(\hat{I}_u). \]

Now applying equation (14) term by term we get for \( n > 1 \)
\[ V_{LHS}(\hat{I}) \leq \frac{n}{n-1} V_{MC}(\hat{I}), \quad 1 \leq d \leq \infty \] (24)

Thus even in the infinite dimensional case, the variance under Latin hypercube sampling can never be much worse than under ordinary Monte Carlo. How much better can it be?

Write \( f(X) = f^s(X) + (f(X) - f^s(X)) \) and recall that \( f(X) - f^s(X) \) has mean zero given \( X^{1:s} \).

\[ V_{LHS}(\hat{I}) = V_{LHS}(\hat{I}^s - \hat{I}) + \sum_{1 \leq |u|, u \subseteq 1:s} V(\hat{I}_u). \] (25)
Now \( V_{LHS}(\hat{\mathcal{I}}^s - \hat{\mathcal{I}}) \leq (n - 1)^{-1}E((f - f^s)^2) \) by (24) and from (13)

\[
\sum_{1 \leq |u|, u \subseteq \mathcal{I}_s} V(\hat{\mathcal{I}}_u) = \frac{1}{n} \left( E((f^s - I)^2) - \sum_{j=1}^{s} E(f_j(X)^2) \right) + o\left(\frac{1}{n}\right).
\]

Putting these together, we find that

\[
V_{LHS}(\hat{\mathcal{I}}) \leq \frac{1}{n} \left( \sigma^2 - \sum_{j=1}^{s} \sigma_j^2 \right) + o(1/n)
\]

holds for any \( s < \infty \).

For any \( \epsilon > 0 \), we can find \( s \) with \( \sum_{j=s+1}^{\infty} \sigma_j^2 < \epsilon \) so that

\[
V_{LHS}(\hat{\mathcal{I}}) \leq \frac{1}{n} \left( \sigma^2 - \sum_{j=1}^{\infty} \sigma_j^2 + \epsilon \right) + o(1/n).
\]

Thus we conclude that

\[
V_{LHS}(\hat{\mathcal{I}}) \equiv \frac{1}{n} \left( \sigma^2 - \sum_{j=1}^{\infty} \sigma_j^2 \right).
\]

Latin hypercube sampling of the infinite dimensional cube removes an additive component, just as in the finite dimensional case.

10 LSS Accuracy

10.1 Finite \( k \), QMC

We begin with Latin supercube sampling of a finite number of finite sets of variables. Let \( \mathcal{X}^r \in [0,1)^{s_r} \), where \( 1 \leq s_r < \infty \) for \( r = 1, \ldots, k < \infty \). Suppose that the points \( \mathcal{X}_i^r \) for \( i = 1, \ldots, n \) are a nonrandom \( s_r \) dimensional integration scheme and that \( d = \sum_{r=1}^{k} s_r \). The motivating case is where one
has an $s_r$ dimensional QMC method for each set of points $X_i^r$.

For independent random permutations $\pi_r$ of 1 through $n$, let

$$X_i = (X_{\pi_1(i)}^1, X_{\pi_2(i)}^2, \ldots, X_{\pi_k(i)}^k)$$

define an integration point in $[0,1]^s$. We are interested in the error $\hat{I} - I$
where $\hat{I} = n^{-1} \sum_{i=1}^n f(X_i)$ as always.

The integration error may be decomposed as

$$\hat{I} - I = (\hat{I} - I_G) + (I_G - I)$$ (26)

where

$$I_G = \frac{1}{n^k} \sum_{i_1=1}^n \sum_{i_2=1}^n \cdots \sum_{i_k=1}^n f(X_{i_1}^1, X_{i_2}^2, \ldots, X_{i_k}^k)$$ (27)

is the average of $f$ over a perhaps very large grid containing all the integration points that could possibly be sampled by LSS.

The fact that the $n$ points $X_i^r$ are embedded in $[0,1)^{s_r}$ does not affect the
distribution of $\hat{I} - I_G$. We could associate point $X_i^r$ with $(i - 0.5)/n \in [0,1)$,
and then find that $\hat{I} - I_G$ is the error in a $k$ dimensional Latin hypercube
sample of the centered type given by equation (11). The variance of centered
Latin hypercube sampling is studied in [28]. Alternatively one can construct
a step function constant within $n^k$ cubic cells of volume $n^{-k}$ and consider $\hat{I}$ to
be a uniform Latin hypercube sample (equation 12) from this $k$ dimensional
step function, noting that the $U_i$ are in this case irrelevant. Either way, for
$n > 1$, the random variable $\hat{I}$ has expectation $I_G$ and variance no more than
\[(n - 1)^{-1} \sigma_G^2 \text{ where} \]

\[
\sigma_G^2 = \frac{1}{n^k} \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \cdots \sum_{i_k=1}^{n} \left( f(X_1^{i_1}, X_2^{i_2}, \ldots, X_k^{i_k}) - I_G \right)^2.
\]

The error in \( I_G \) is that of a \( k \) fold iterated method. The rate of error is thus that of the worst of the individual integration rules \( \mathcal{X}_r^r \). More precisely it is the sum of the \( k \) error rates, and thus the fact that \( k < \infty \) is significant. See Davis and Rabinowitz [6].

If QMC methods are used for \( \mathcal{X}_r^r \) then \( I_G \) is within QMC error of \( I \). Similarly \( \sigma_G^2 \) is within QMC error of \( V(f(X)) \). (The latter condition requires \( f^2 \) to be of finite variation in the sense of Hardy and Krause.) Assuming that the QMC methods are better than MC for the sample size \( n \) and dimensionalities \( s_r \) used, the Monte Carlo error from the Latin hypercube sampling of \( \hat{I} - I_G \) will dominate the error.

This shows that LSS will not be much worse than ordinary Monte Carlo, provided that the QMC methods used as building blocks are themselves not worse than ordinary Monte Carlo. If the QMC methods are superior to MC for the sample sizes used then the error is dominated by \( \hat{I} - I_G \). The LSS variance of this error may be studied in terms of a \( k \) dimensional ANOVA decomposition of the \( n^k \) values \( f(X_1^{i_1}, X_2^{i_2}, \ldots, X_k^{i_k}) \). The result is that

\[
V_{LSS}(\hat{I}) = \frac{1}{n} \left( \sigma_G^2 - \sum_{r=1}^{k} \sigma_{G(r)}^2 \right)
\]

where

\[
\sigma_{G(r)}^2 = \frac{1}{n} \sum_{i_r=1}^{n} \left( \frac{1}{n_{k-1}} \sum_{i_1=1}^{n} \cdots \sum_{i_{r-1}=1}^{n} \sum_{i_{r+1}=1}^{n} \cdots \sum_{i_k=1}^{n} f(X_1^{i_1}, \ldots, X_k^{i_k}) - I_G \right)^2.
\]
Each $\sigma^2_{G(r)}$ is within QMC errors of

$$\int_{[0,1)^{A_r}} \left( \int_{[0,1)^{-A_r}} (f(X) - I) dX^{-A_r} \right)^2 dX^{A_r}$$

$$= \int_{[0,1)^{A_r}} \left( \sum_{0 < |u|, u \subseteq A_r} f_u(X) \right)^2 dX^{A_r}$$

$$= \sum_{0 < |u|, u \subseteq A_r} \sigma_u^2$$

using equation (7) with $\nu = -A_r$. Therefore

$$V_{LSS} (\hat{I}) \simeq \frac{1}{n} \left( \sigma^2 - \sum_{r=1}^{k} \sum_{0 < |u|, u \subseteq A_r} \sigma_u^2 \right). \quad (29)$$

10.2 Finite $k$, RQMC

Latin supercube sampling with QMC point sets removes the ANOVA components corresponding to subsets of input variables that appear together in a QMC list, from the asymptotic variance. Asymptotically, the bias $I_G - I$ can be neglected.

It may take very large $n$ before we are confident that the variance of $\hat{I}$ is well approximated by its asymptotic representation (29). This is not a critical flaw, because one can replicate the LSS randomization to get a direct variance estimate. In fact replication may be the only practical way to estimate (29).

What can be a critical flaw is that the error $I_G - I$ may not be negligible for practical $n$. Although this is the error in a rule of $n^k$ points, it may be no more accurate than the least accurate of the $k$ QMC methods used. This bias $I_G - I$ persists over independent replication of LSS randomization.

This concern motivates using RQMC point sets instead of QMC ones.
If the RQMC point sets satisfy \( X_i^r \sim U[0,1]^d \) then \( I_G \) becomes a random variable with expectation \( I \). Then the difference \( I_G - I \) contributes to the variance of \( \hat{I} \) but does not cause the mean of \( \hat{I} \) to be in error. The total variance \( V_{RQMC,LSS}(\hat{I}) \) may be estimated by replication. Assuming that the RQMC methods are asymptotically better than MC methods, then the asymptotic variance of LSS with RQMC sampling is also given by equation \((29)\). That is if \(|I_G - I| = o(n^{-1/2})\) then \( V_{RQMC}(I_G) = o(1/n) \) and is asymptotically negligible. This would be the case for example, if the QMC methods were superior to MC methods, and the randomizations used preserved or enhanced the QMC properties.

10.3 Infinite \( k \), or infinite \( s_r \)

Suppose that \( k = \infty \). One can mimic the martingale construction used in Section 9 taking \( f^r = E(f(X)|X^1, \ldots, X^r) \). Then as in LHS with \( d = \infty \) one can truncate the integrand at some large \( r \) capturing virtually all of the variance. Thus taking \( k = \infty \) should not be especially problematic for LSS.

The problem with having some \( s_r = \infty \) is that it may be impossible to find an (R)QMC method in infinite dimensions that has better performance than MC. Without better performance than MC one cannot conclude that the error \( I_G - I \) is eventually negligible.

11 Conclusions

In view of results like Bahvalov's theorem (given in [6]) numerical integration in high dimensions is known to be intractable. This means that whatever method we're using, there are integrands, perhaps even smooth ones, on which we'll get bad results. Sloan and Wozniakowski [38] give another
intractability result for high dimensions, where smoothness means rapidly decaying Fourier coefficients. Of course, intractability does not mean that we'll always get bad results in practice.

When good results are obtained in integrating a high dimensional function, we should conclude first of all that an especially tractable integrand was tried and not that a generally successful method has been found. A secondary conclusion is that we might have made a very good choice in selecting an integration method to exploit whatever features of \( f \) made it tractable. For example, even if \( f \) is virtually linear, simple Monte Carlo will get a bad result if \( f \) has a large variance, while some other methods will do very well.

In this paper we've considered methods that can exploit integrands of low effective dimension. Latin hypercube sampling works well on integrands that are largely one dimensional in the superposition sense. Methods based on \((R)QMC\) with padding work well on integrands with low dimensional structure among the variables treated by \((R)QMC\) and, if padded by Latin hypercube sampling, one dimensional structure among the other variables.

Latin supercube sampling allows the practitioner to exploit still more structure in the integrand. If much of the variation is concentrated within groups of a few variables, especially if it is concentrated within low dimensional subsets of those variables, then LSS allows us to exploit that structure by grouping those variables within \((R)QMC\) point sets.

It is not reasonable to expect that these methods will be able to turn high dimensional integration into a tractable problem. But it may turn out that some broad classes of integration problems have their variation concentrated among several small subsets of input variables. It may also be possible for practitioners to engineer their integrands in order to concentrate
the variation in such subsets.

References


