
AN INTRODUCTION TO QUASI MONTE CARLO METHODS

Kristian Stormark

Project — Fall 2005



NTNU
Norwegian University of
Science and Technology

Faculty of Information Technology,
Mathematics and Electrical Engineering

Preface

This paper has been written as a part of the course TMA4700 “Matematisk fordypning” (Eng.: *Mathematical specialization*) at the Department of Mathematical Sciences of the Norwegian University of Science and Technology (NTNU) during the autumn of 2005.

I have attempted to make the presentation as simple and self-contained as possible, so that a student with some background in mathematics at the university level should be able to follow it.

I would like to give thanks to my supervisors Professor Håvar Rue and Associate Professor Håkon Tjelmeland for their assistance on this project. Their guidance and constructive feedback have been most valuable. I would also like to thank my girlfriend for her patience, and the Lord for giving my new strength each morning.

This paper is inspired by the work of many authors. I have tried to give credit where credit is due. The book *Monte Carlo methods in Financial Engineering* by Professor Paul Glasserman has been an especially useful source.

Trondheim, December 2005
Kristian Stormark

Abstract

Quasi Monte Carlo (QMC) methods are numerical techniques for estimating expectation values, or equivalently, for integration. Contrasted to the random sampling of the classical Monte Carlo methods (MC), we investigate the improved convergence rate that often is attainable for the QMC methods by the use of certain deterministic sequences. Motivated by the Koksma-Hlawka inequality, we explore some of the properties of the favourable low-discrepancy sequences. In particular, we report methods of construction for the Halton sequences, and the (t, d) -sequences of Faure and Sobol'. We also mention the lattice rules, mainly by a quick reference to the Korobov rules. To explain the successful application of the Quasi Monte Carlo methods to many high-dimensional integrals, we refer to the notion of effective dimension. We also mention some of the possible modification procedures for the QMC methods, such as randomization and hybridization. A numerical examination of the Halton, Faure and Sobol' sequences is performed by the use of a simple test problem, and we find that the convergence is considerably faster for the QMC methods than for the MC methods.

Contents

Preface	iii
Abstract	v
1 Introduction	1
1.1 Classical Monte Carlo	1
1.2 Some general comments	2
1.3 Prelude to a deterministic approach	3
2 Quasi Monte Carlo methods	5
2.1 General principles	5
2.2 Choice of sequences	6
2.3 Effective dimension	9
3 Construction of point sets	12
3.1 Preliminary - Van der Corput sequences	12
3.2 A simple extension - Halton sequences	14
3.3 Digital nets	15
3.3.1 Faure sequences	18
3.3.2 Sobol' sequences	20
3.4 Lattice rules	25
3.4.1 Extensible lattice rules	25
3.4.2 Korobov rules	26
4 Possible extensions	27
4.1 Randomization	27
4.2 Hybrid Monte Carlo	28
5 A simple test	29
5.1 Test function	29
5.2 Comments on the test function	30
5.3 Some general comments	31
5.4 Testing procedure	31
5.5 Results	33
6 Summary	38
References	40

1 Introduction

The complexity of many real-life computational problems is of an extent that prohibits solution by analytical means, and very often the only feasible course of action is simulation. In line with such a strategy, the Monte Carlo methods have proven indispensable over the last decades. Nevertheless, the convergence of these methods is sometimes too slow for certain desirable utilizations. As a result, the continuative methodology referred to as *Quasi Monte Carlo* has been developed. While the convergence rate of classical Monte Carlo (MC) is $O(n^{-1/2})$, the convergence rate of Quasi Monte Carlo (QMC) can be made almost as high as $O(n^{-1})$. Correspondingly, the use of Quasi Monte Carlo is increasing, especially in the areas where it most readily can be employed.

1.1 Classical Monte Carlo

The term *Monte Carlo* is nowadays used in all sorts of scientific literature, and broadly speaking it denotes various approaches for solving computational problems by means of “random sampling”. Historically, the term was used as code word for the stochastic simulations conducted at Los Alamos during the development of the atomic bomb. The name was inspired by the famous casino in Monaco, thus reflecting the element of chance that enters the methods (Metropolis, 1987). In the absence of a generally recognized definition, we will adopt the one given by Anderson (1999).

Definition 1.1.1 (Monte Carlo) *Monte Carlo is the art of approximating an expectation by the sample mean of a function of simulated random variables.*

Many quantities of interest (e.g. probabilities, integrals and summations) can be formulated as expectations, and the Monte Carlo methods are indeed important tools in a wide range of applied fields, from computational physics, computer science and statistics to economics, medicine and sociology.

The basic idea of estimating an expectation by the corresponding sample mean may seem rather intuitive, and in addition, it has a sound mathematical foundation, which we will report the essentials of below.

Let X be a random variable with probability distribution $p_X(x)$ defined on $\Omega \subseteq \mathbb{R}^d$, and let $g(\cdot)$ be a mapping from Ω into \mathbb{R} . The expected value (or mean) of the random variable $g(X)$, denoted $E g(X)$, is then

$$\mu = E g(X) = \begin{cases} \int_{\Omega} g(x)p_X(x)dx & \text{in the continuous case,} \\ \sum_{x \in \Omega} g(x)p_X(x) & \text{in the discrete case.} \end{cases} \quad (1)$$

Given a sequence X_1, X_2, \dots of independent and identically distributed (i.i.d.) random variables with the same distribution function as X , we define the (classical) Monte Carlo estimator of μ as

$$\hat{\mu}_n = \frac{1}{n} \sum_{i=1}^n g(X_i). \quad (2)$$

By the strong law of large numbers $\hat{\mu}_n$ will converge *almost surely*¹ to μ , provided that also the latter quantity is finite. Further on, if both the expectation and the variance of $g(X)$ is finite, then by the central limit theorem, the asymptotic distribution of the approximation error $\hat{\mu}_n - \mu$ is the normal distribution with mean 0 and standard deviation $\sqrt{\sigma^2(g)/n}$, where $\sigma^2(g)$ is the variance of $g(X)$. The variance of $g(X)$ can be estimated by the sample variance,

$$\sigma^2(g) = \text{E} (g(X) - \mu)^2 \approx \frac{1}{n-1} \sum_{i=1}^n (g(X_i) - \hat{\mu}_n)^2 = s_n^2(g).$$

Due to the form of the limiting standard deviation of $\hat{\mu}_n - \mu$, the convergence rate of the MC estimator is said to be

$$O\left(\sqrt{\sigma^2(g)/n}\right) = O\left(n^{-1/2}\right). \quad (3)$$

Details on the asymptotic properties of the Monte Carlo estimator can be found in Casella and Berger (2002). Even for correlated random variables similar results may continue to hold, and extension to samples from Markov chains is possible.

1.2 Some general comments

In the remaining sections we will refer only to the continuous case in order to maintain a plain presentation. The discrete case is analogous. We will also follow the common practice of using a normalized version of (1). More specifically, by performing a suitable change of variables if necessary, the domain of integration can be contained in the unit hypercube $[0, 1]^d$. By introducing the characteristic function to the integrand, the domain of integration can be extended to the whole unit hypercube, so that the corresponding uniform distribution can be applied. That is,

$$\int_{\Omega} g(x)p_X(x)dx = \int_{[0,1]^d} f(u)du, \quad (4)$$

where U_1, \dots, U_n are random variables from the uniform distribution on $[0, 1]^d$ and $f(u)$ denotes the transformed version of the integrand $g(x)p_X(x)$ as outlined above. The *art* of Monte Carlo estimation is then essentially to perform the transformation in a clever way. One of the main reasons for using the normalized formulation is that the starting point of practically all simulation algorithms is a random number generator that produces number sequences that mimic the uniform distribution on $[0, 1]$. In some situations the integrand f may not be available in explicit form, but this need not cause any serious problems as long as there exist *some* way to evaluate it. In such a case the dimension d represents the maximum number of uniforms needed to carry out each evaluation.

As indicated above, we will follow the standard of taking all intervals to be closed to the left and open to the right. This is a convenient convention, but has no *practical* influence on the main results, since the volume (Lebesgue measure) of such a boundary is zero. For

¹A sequence of random variables X_1, X_2, \dots , converges *almost surely* to a random variable X if, for every $\epsilon > 0$, $P(\lim_{n \rightarrow \infty} |X_n - X| < \epsilon) = 1$.

instance, the normalized integral in (4) could just as well be taken over the closed unit hypercube $[0, 1]^d$ as over the half-open unit hypercube $[0, 1)^d$. On the other hand, in some parts of theory behind the Quasi Monte Carlo methods it is necessary to specify to what sets the boundary points of different rectangles belong. For this reason we will consistently employ the half-open sets.

Since the expectation of a random variable can be treated as an integral (and vice versa), we will in the following focus on the approximation of the normalized integral. By letting U_1, \dots, U_n constitute a random sample from the uniform distribution on $[0, 1)^d$, the classical Monte Carlo estimator can be formulated as

$$\hat{I}_n(f) = \frac{1}{n} \sum_{i=1}^n f(U_i) \approx \int_{[0,1)^d} f(u) du = I(f). \quad (5)$$

Beside the conceptual simplicity, the most lucrative aspect of the classical Monte Carlo methods is by far the independence of problem dimension to the probabilistic error bounds, which then is $O(n^{-1/2})$. For instance, in the case of numerical integration in high dimensions, this property very often renders Monte Carlo methods superior, even though regular integration rules usually have a better convergence rate in lower dimensions. By way of comparison, we can mention the product trapezoidal rule which is $O(n^{-2/d})$ in the d -dimensional case, at least for twice continuously differentiable integrands.

Despite the high applicability to high-dimensional problems, the convergence rate of the classical Monte Carlo methods may still appear rather slow in practice. Several variance reduction techniques have therefore been developed to speed things up as much as possible (see Glasserman (2003)). Though fundamentally inadequate to overcome the slow convergence rate itself, these techniques can at least reduce the incorporated coefficients and thus improve the convergence by a hopefully considerable factor.

The manifestations of the statistical approach taken by the classical Monte Carlo methods is also quite evident; as sampling-based procedures they have mere *probabilistic* error bounds. No guarantee can be given that the expected precision is obtained in a particular computation, and in some situations this can be very inconvenient, if not totally unacceptable. Neither is any regularity of the integrand reflected in the convergence rate, a position which must be considered unfavourable (Niederreiter, 1992). Of course, the relatively weak condition of square-integrability is generally needed for the asymptotic properties of the MC estimator to hold in the first place.

For that matter, the generation of random samples is in itself a difficult task, a fact we will address in brevity in the following. Niederreiter (1992) and Okten (1999) provide further discussions on the subject.

1.3 Prelude to a deterministic approach

The accessibility of random samples is essential for the use of classical Monte Carlo methods, at least from a theoretical point of view. Without engaging the rather philosophical problem of defining “randomness”, we can in passing mention that questions have been raised about the ability of the random number generators used in modern computers to produce numbers of such a character (Niederreiter, 1992; Okten, 1999). All the same, Monte Carlo methods have been widely implemented and put to practical use,

and the success is obvious. Lending the words of Okten (1999), this potential disparity can be overcome *«by the theory of uniform distribution, which claims that the reason our “random” methods work does not have anything to do with their “randomness”, but “uniformity”. This brings us to the notion of “quasi-Monte Carlo”»*. Indeed it does.

2 Quasi Monte Carlo methods

The quest for generating genuine *random* samples by arithmetical methods may be considered an odd undertaking. Still, the so-called pseudo-random numbers generated by ordinary computers will suffice for most practical purposes, and so the main motivation behind turning to Quasi Monte Carlo is that of improved convergence properties. As we shall see, the principal difference between classical Monte Carlo and Quasi Monte Carlo is the nature of the sampling procedure; MC methods rely on random samples, while QMC methods use a deterministic sequence of points to ensure a more even distribution over the unit hypercube. The disadvantage of using a random sample in this setting is namely the *independence* between the different sample points, which for finite sample can be seen to induce clusters and gaps to a noticeable extent. The phenomenon is illustrated in Figure 1. By the transition to the so-called *low-discrepancy sequences*, the convergence rate can be increased from $O(n^{-1/2})$ to nearly $O(n^{-1})$. However, this notation conceals a dependence on the problem dimension which may be of severe practical importance. We will return to this later. In addition to the improved convergence rate, the prospect of obtaining deterministic error bounds, which then is present for QMC, is also alluring.

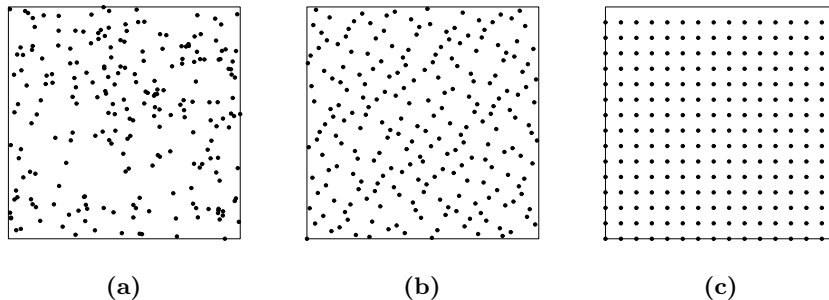


Figure 1: Plots of three different point sets in $[0, 1]^2$, each of size $n = 225$. (a): points of a (pseudo-) random sample. (b): points from a low-discrepancy sequence. (c): the points of a grid. The clusters and gaps in the left plot is a result of the mutual independence of the sample points.

2.1 General principles

As we might already have given away, the idea of using deterministic sampling points is based on the notion of uniform representation emerging from even spread of sample points. Before we give a closer account of this matter, let us present the Quasi Monte Carlo approximation of the normalized integral, which then for the point set $\{x_i\}_{i=1}^n$ is given by

$$I(f) = \int_{[0,1]^d} f(x) dx \approx \frac{1}{n} \sum_{i=1}^n f(x_i) = \hat{I}_n(f). \quad (6)$$

The resemblance to the MC estimator (5) is pronounced, but in this case the sampling points, x_1, \dots, x_n , constitute a deterministic point set and not a random sample.

Naturally, the particular choice of point set $\{x_i\}_{i=1}^n$ is essential for the accuracy of the QMC approximation (6). This is most strongly indicated by the *Koksma-Hlawka inequality*, which we will state in the next section. This inequality provides the theoretical basis for the QMC methods. In general, we refer to all methods that approximate integrals according to (6) as a QMC method, regardless of the specific sets of points that are being used. Still, only methods that employ “clever” point sets will be attractive for practical purposes, and for this reason the usage of such point sets is often implied when the term *Quasi Monte Carlo* is used.

While MC and QMC is quite similar in many aspects, the significance of problem dimension is one of the properties that most clearly distinguish them. For instance, the construction of random samples in arbitrary dimension d is no more difficult than in dimension 1, since k i.i.d. multivariate uniforms can be acquired by clustering kd i.i.d. univariate uniforms. More importantly, the rate of convergence for the MC estimator is independent of d . For QMC estimation on the other hand, the effect of increasing dimension can be far more corrupting. We will return to this weakness in the following sections. Moreover, it is not possible to construct appropriate sequences for QMC in some dimension d by combining lower dimensional sequences in the same way as for MC samples. While the dimension need not even be properly determined in the MC case, it is cardinal for QMC.

2.2 Choice of sequences

The presentation of this section is inspired by the excellent account of QMC given by Glasserman (2003), which again follows Niederreiter (1992) on some of the topics of concern.

In a broad sense, the QMC approximation (6) can be understood as the evaluation of an entire unit by considering only a selection of fragments. Consequently, it is reasonable to expect the accuracy of the approximation to increase with both the number of fragments *and* the evenness of their representation. This is indeed the case, and apart from the definite properties of the integrand, the above factors may solely confine the error, as the Koksma-Hlawka inequality will render evident.

Bearing this in mind, the points on a grid may initially appear as an advisable choice of point set, since it represents a collection of highly separated points. The unit hypercube, $[0, 1]^d$, would then be partitioned into blocks of equal representation, accommodating the aspired uniformity (see Figure 1). Still, the grid has a major drawback; the number of points to be used for an actual computation virtually *demand*s a priori specification. This is due to the fact that number of points needed to maintain the structure would grow exponentially if a constituted grid were to be refined later on. For instance, if the starting grid comprised b splits in each of the d dimensions (for a total of b^d points), then the grid would consist of $b^{(k+1)d}$ points after k successive refinements. For most practical purposes, this makes the grid a poor choice. Fortunately, there exist sequences that assure uniformity as the original sequence is upgraded with bounded-length segments.

To explore the properties of such sequences, we will start by specifying a measure of uniformity (or actually, a measure of deviation from uniformity) for arbitrary point sets $\{x_1, \dots, x_n\}$ in $[0, 1]^d$, which we will refer to as *the discrepancy*. In the below definition, the term $\#\{x_i \in A\}$ represents the number of points from the point set that is included in the set A . That is, $\#\{x_i \in A\} = \sum_{i=1}^n \chi_A(x_i)$, where $\chi_A(\cdot)$ is the characteristic function

(indicator function) on A .

Definition 2.2.1 (Discrepancy) *Let \mathcal{A} be a collection of (Lebesgue measurable) subsets of $[0, 1]^d$. The discrepancy of the point set $\{x_1, \dots, x_n\}$ relative to \mathcal{A} is*

$$D(x_1, \dots, x_n; \mathcal{A}) = \sup_{A \in \mathcal{A}} \left| \frac{\#\{x_i \in A\}}{n} - \text{vol}(A) \right|. \quad (7)$$

As can readily be seen from the above expression, both of the entering terms are positive, being a portion and a volume, respectively. The volume of the unit hypercube is 1, so both terms are also bound from above by this value, and thus, $0 \leq D(x_1, \dots, x_n; \mathcal{A}) \leq 1$. A far more essential feature of the discrepancy is that the measure will become large (i.e., close to 1) both when a small subset contains many of the points and when a large subset contains few (or none) of the points, and so $D(x_1, \dots, x_n; \mathcal{A})$ is indeed a measure of the non-uniformity of the point set relative to the collection \mathcal{A} .

Particularly, the *ordinary* (or *extreme*) *discrepancy* $D(x_1, \dots, x_n)$ is defined by taking \mathcal{A} to be the collection of all rectangles in $[0, 1]^d$ on the form

$$\prod_{i=1}^d [u_j, v_j), \quad 0 \leq u_j < v_j \leq 1,$$

that is, all rectangles in $[0, 1]^d$ with edges aligned with the coordinate axes. Somewhat less general, the *star discrepancy* $D^*(x_1, \dots, x_n)$ is defined by restricting the above collection to the rectangles anchored at the origin ($u_j = 0$ for each j). Illustrations of the star discrepancy for two point sets in $[0, 1]^2$ is given in Figure 2. As accounted for in Section 1.2, we take all rectangles to be closed at the bottom and open at the top. In particular, the points on the upper and left edges of the largest rectangle in subfigure (b) are not included in the rectangle.

Both the ordinary discrepancy and the star discrepancy make exclusive use of rectangles, and even though other relevant subsets are ignored, we will see that these are adequate for characterizing the approximation error in (6). For this reason we will now state the Koksma-Hlawka inequality (without proof). The marvel of this result is that it supplies the QMC approximation (6) with a highly illustrative error bound.

Result 2.2.1 (Koksma-Hlawka inequality) *If the function f has finite Hardy-Krause variation $V(f)$ on $[0, 1]^d$, then for any $x_1, \dots, x_n \in [0, 1]^d$*

$$\left| \frac{1}{n} \sum_{i=1}^n f(x_i) - \int_{[0,1]^d} f(x) dx \right| \leq V(f) D^*(x_1, \dots, x_n). \quad (8)$$

We will only make superficial use of the so-called Hardy-Krause variation here, so we refer to Glasserman (2003), p. 287-288, for a precise definition. The significance of the

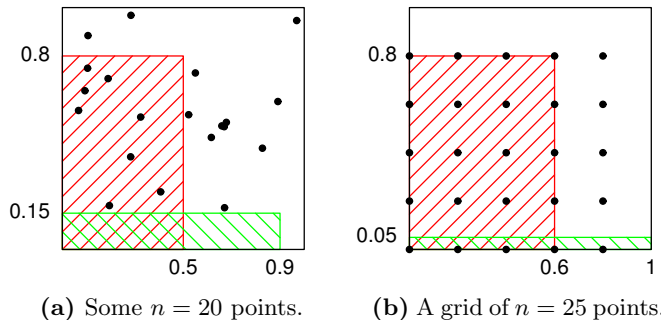


Figure 2: Some rectangles in $[0,1]^2$ anchored at the origin, comprising various proportions of sample points. (a): The largest rectangle contains a proportion of $8/20 = 0.4$ of the points and has volume 0.4, for a difference of 0. The smallest rectangle contains a proportion of $2/20 = 0.1$ of the points and has volume 0.1875, for a difference of -0.0875 . (b): The largest rectangle contains a proportion of $12/25 = 0.48$ of the points and has volume 0.48, for a difference of 0. The smallest rectangle contains a proportion of $5/25 = 0.2$ of the points and has volume 0.05, for a difference of 0.15. The star discrepancy of a point set is the supremum of the absolute value of the differences taken over all such origin-anchored rectangles.

above formulation (8) is that the error bound of the QMC approximation can be seen to depend on the point set *only through the deviation from uniformity* (here expressed as the star discrepancy), and for that matter, on the integrand only through the (Hardy-Krause) variation.

Generally, it is very hard to compute $V(f)$ and $D^*(x_1, \dots, x_n)$ - it may even be more intricate than the actual integral evaluation! In some cases where both of them are known, the above bound is found to grossly overestimate the true error (even though it can be proven sharp in the sense that for a given point set and for each $\epsilon > 0$ there exists an infinitely differentiable function for which the error comes within ϵ of the bound, cf. Niederreiter (1992), p. 20). The requirement of finite Hardy-Krause variation, $V(f)$, is also quite restrictive, and for instance, it means that f needs to be bounded.

Along with other substantial issues, these weaknesses render the Koksma-Hlawka inequality rather useless for practical error control. Generalized versions of the inequality exist, some of which are more suitable for computation (Hickernell, 1998). Nevertheless, the principal importance of uniformity is just as well articulated by (8). We will therefore treat the star discrepancy as a capable measure of the integration error.

According to Glasserman (2003), little is known about the best possible discrepancy in dimensions higher than 1. Niederreiter (1992) states that “it is widely believed” that in dimension $d \geq 2$, any point set x_1, \dots, x_n satisfies

$$D^*(x_1, \dots, x_n) \geq c_d \frac{(\log n)^{d-1}}{n}, \quad (9)$$

and that the first n elements of an arbitrary infinite sequence x_1, x_2, \dots satisfy

$$D^*(x_1, \dots, x_n) \geq c'_d \frac{(\log n)^d}{n}, \quad (10)$$

for constants c_d and c'_d depending only on the dimension d . As the above bounds (9) and (10) indicate, it is generally possible to achieve lower star discrepancy for point sets that

are customized to each particular sample size n , than for point sets taken from a single sequence. On the other hand, if the number of points needed to achieve some given degree of accuracy is unknown a priori, the consecutive points of an infinite sequence will usually provide higher computational efficiency. Sequences with star discrepancy of $O((\log n)^d/n)$ exist, and these are commonly known as *low-discrepancy sequences*.

Definition 2.2.2 (Low-discrepancy sequence) *A sequence x_1, x_2, \dots of points in $[0, 1]^d$ is a low-discrepancy sequence if for any $n > 1$*

$$D^*(x_1, \dots, x_n) \leq C_d \frac{(\log n)^d}{n}, \quad (11)$$

where the constant C_d depends only on the dimension d .

The construction of low-discrepancy sequences is the topic of Section 3. Points taken from such sequences are usually characterized as *quasi-random*, but the term is perhaps somewhat of a misnomer. In reality, there is nothing “random” about the points, even though they are distributed uniformly over the unit hypercube. For this reason, the term *sub-random* is held out by purists, but the designation of quasi-random still prevails. Of course, the prefix *quasi* should not be confused with the somewhat similar *pseudo* prefix. Whereas *pseudo-random* refers to the seemingly independent “random numbers” generated on a computer, *quasi-random* points are as such highly correlated by construction.

Since any power of n grows faster than a fixed power of the logarithm, we by definition have that the star discrepancy of any low-discrepancy sequence is

$$O\left(\frac{(\log n)^d}{n}\right) = O\left(\frac{1}{n^{1-\epsilon}}\right), \quad \epsilon > 0, \quad (12)$$

allowing for the more informal description of “almost $O(1/n)$ convergence” for the QMC approximation (6), as introduced earlier and justified by the Koksma-Hlawka inequality (8). Despite the *asymptotic* property expressed in (12), the number of points needed to experience this rate of convergence may be unattainable in practical situations for high values of d . For this reason, the QMC methods have traditionally been assumed applicable only for problems of moderate dimension (say, for $d \leq 15$). Still, QMC has proven successful for relatively high dimensions in a number of cases. For instance, Paskov and Traub (1995) demonstrates a successful application of QMC for $d = 360$ in a financial setting. We will comment further on the phenomenon in the next section.

2.3 Effective dimension

As already mentioned, the asymptotic convergence rate indicated by (12) for QMC methods that employ low-discrepancy sequences may not be relevant in high dimensions, since for large d , the factor $(\log n)^d/n$ is considerably larger than $n^{-1/2}$ unless n is huge. In particular, the apparent error bound given by the right hand side of (10) increases with n until n exceeds $\exp(d)$. So to account for the rather unexpected success of QMC in many high-dimensional applications, the concept of *effective dimension* has been introduced (Caffisch et al., 1997). We will present only the basic idea here, but a more extensive treatment can be found in Wang and Fang (2003) and Wang and Sloan (2005).

In the current setting, the effective dimension is linked to the so-called *anova decomposition*

of the integrand. More generally, the anova (analysis of variance) decomposition is a technique that decomposes a function $f(x)$ into a sum of simpler functions depending only on distinct groups of the coordinates of the full variable x . We will outline the essentials below.

Let $\mathcal{I} = \{1, \dots, d\}$ be the set of dimension indices, where d is the nominal dimension of f (i.e. $f : \mathbb{R}^d \mapsto \mathbb{R}$). For any subset $e \subseteq \mathcal{I}$, let $|e|$ denote its cardinality (number of elements), and let \bar{e} denote its complementary set in \mathcal{I} (i.e., the indices of \mathcal{I} not in e). Let x_e be the $|e|$ -dimensional vector containing the coordinates of x with indices in e , and let \mathcal{U}^e be the corresponding $|e|$ -dimensional unit hypercube. For instance, $\mathcal{U}^{\mathcal{I}}$ is then identical to $[0, 1]^d$, since $|\mathcal{I}| = d$.

Given that $f(x)$ is a square integrable function, it can be expressed as the sum of its 2^d anova terms². That is,

$$f(x) = \sum_{e \subseteq \mathcal{I}} f_e(x), \quad (13)$$

where $\int_0^1 f_e(x) dx_j = 0$ if $j \in e$. The anova terms $f_e(x)$ can be defined recursively by

$$f_e(x) = \int_{\mathcal{U}^{\bar{e}}} f(x) dx_{\bar{e}} - \sum_{k \subset e} f_k(x), \quad (14)$$

where the last sum is over strict subsets k of e ($k \neq e$). By convention, $\int_{\mathcal{U}^{\emptyset}} f(x) dx_{\emptyset} = f(x)$, and clearly $f_{\emptyset} = \int_{\mathcal{U}^{\mathcal{I}}} f(x) dx = I(f)$. It also follows from (14) that the anova decomposition is orthogonal in the sense that $\int_{\mathcal{U}^{\mathcal{I}}} f_e(x) f_k(x) dx = 0$ whenever $e \neq k$. Informally speaking, the anova terms f_e point out the joint contribution of the group of variables e to the function f .

As the name itself readily announces, the anova decomposition formulation (13) of f is customized for an analysis of the different components of the “variance” of the function f on $[0, 1]^d$. More precisely, we define the variance of f to be $\sigma^2(f) = \int_{\mathcal{U}^{\mathcal{I}}} [f(x) - I(f)]^2 dx$, which then in standard statistical terminology corresponds to the variance of the random variable $f(U)$, when U is a uniformly distributed random variable on $[0, 1]^d$. The following property holds:

$$\sigma^2(f) = \sum_{e \subseteq \mathcal{I}} \sigma_e^2(f),$$

where $\sigma_e^2(f) = \int_{\mathcal{U}^{\mathcal{I}}} [f_e(x)]^2 dx$ for $|e| > 0$ and $\sigma_{\emptyset}^2(f) = 0$. The component of the variance of f corresponding to $e \subseteq \mathcal{I}$ and all of its subsets can therefore be expressed as

$$V_e(f) = \sum_{k \subseteq e} \sigma_k^2(f). \quad (15)$$

Accordingly, we will now define two different notions of effective dimension; the *truncation dimension* $d_t(\rho)$ and the *superposition dimension* $d_s(\rho)$. The parameter ρ is usually chosen close to 1 (say, $\rho = 0.99$), specifying the level of significance.

Definition 2.3.1 (Truncation dimension) *The effective dimension of f in the truncation sense is the smallest integer $d_t(\rho)$ such that*

²Each index i is either included or excluded from a subset, hence the base 2.

$$V_{\{1, \dots, d_t\}} \geq \rho \sigma^2(f). \quad (16)$$

Definition 2.3.2 (Superposition dimension) *The effective dimension of f in the superposition sense is the smallest integer $d_s(\rho)$ such that*

$$\sum_{e \subseteq \mathcal{I}: |e| \leq d_s} \sigma_e^2(f) \geq \rho \sigma^2(f). \quad (17)$$

We will not attempt to give a proper interpretation of the above quantities, but qualitatively speaking, the truncation dimension, d_t , can be seen as the number of “important” variables if the variables are ordered by their importance. On the contrary, the superposition dimension, d_s , is independent of the assignment of indices, and can be seen as a measure of the order of significant variable interactions. The latter may therefore be the most useful when all the variables are equally important. To concretize somewhat, we can mention that a linear function has superposition dimension $d_s = 1$ and a quadratic function has superposition dimension $d_s \leq 2$, but either could have truncation dimension $d_t = d$.

We will not show this here, but by expressing the error of the QMC approximation (6) as a sum of errors introduced by the various anova components of f , the importance of the effective dimension for QMC can be apprehended. The essentials are the following (cf. Wang and Fang (2003) for details):

- (i) if the integrand has low truncation dimension, it is reasonable to expect an improvement of QMC over MC,
- (ii) if the integrand has low superposition dimension, even if the truncation dimension is high, it is still reasonable to expect that QMC will be more efficient than MC.

Nevertheless, the effective dimension provides only partial information of the difficulty of approximating integrals. Low effective dimension *does not suffice* to guarantee the effectiveness of QMC.

3 Construction of point sets

Like for Section 2.2, the presentation of this section is based on that of Glasserman (2003). In addition to supplementary theory, Glasserman also provide code snippets for implementation of each of the point sets we will describe here.

As assessed in Section 2.2, sample points with low discrepancy are highly suitable to use in QMC methods. To construct such samples, different approaches are possible. Still, the most natural settings for development and analysis of low-discrepancy methods are number theory and abstract algebra. To avoid unnecessary details, we will present only the main ideas of the construction strategies. A thorough treatment of the underlying theory can be found in Niederreiter (1992).

We should perhaps also point out that we in the following will make a clearer distinction between *low-discrepancy sequences* and *low-discrepancy point sets*. By a low-discrepancy sequence in dimension d we mean an infinite sequence with star discrepancy of order $O(\log^d n/n)$, while low-discrepancy point sets denote a series of *fixed-size* samples that achieve the same order of star discrepancy. Consequently, the use of low-discrepancy sequences is often convenient in practical situations, as they allow the sample size to be upgraded without abandoning the previously calculated function values (cf. the discussion of the grid in Section 2.2). Of course, the first n terms of any low-discrepancy sequence then constitute a low-discrepancy point set by definition.

The purpose of this section is to present the two main families of methods; digital nets and lattice rules. The digital nets are taken from low-discrepancy sequences, while the lattice rules primarily define low-discrepancy point sets (but certain extension procedures are available). Parts of the theory behind the lattice rules also suggest that they are especially beneficial when the integrand is smooth, even though they can be used for practically all integrands.

Before we actually introduce the digital nets and the lattice rules, we will describe a particular class of one-dimensional low-discrepancy sequence known as the *Van der Corput sequences*. As mentioned in Section 2.1, it is generally not possible to construct low-discrepancy sequences in some dimension d by clustering elements from low-discrepancy sequences in lower dimensions; the low-discrepancy sequences are confined to the dimension for which they are accommodated. Nevertheless, many of the sequences are based on *extensions* and *permutations* of the Van der Corput sequences, the simplest being the *Halton sequences* which we will present in Section 3.2.

3.1 Preliminary - Van der Corput sequences

To describe the Van der Corput sequences properly, we will first have to introduce the concept of base- b expansions of non-negative integers, which then simply is a representation of the integers by power series in an integer base $b \geq 2$. We will also define the so-called *radical inverse function*.

Each positive integer k has a unique representation (called its *base- b* or *b -ary* expansion) as a linear combination of non-negative powers of b with coefficients in $\{0, 1, \dots, b-1\}$. The perhaps most well-known example is the *binary representation* used by modern computers, which corresponds to a base-2 expansion. For instance, the base-2 (or binary) expansion of the integer 43 (or 101011 in binary form) is $1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$.

More generally, the base- b expansion of $k \in \mathbb{N}$ can be written as

$$k = \sum_{j=0}^{\infty} a_j(k)b^j, \quad (18)$$

where all but finitely many of the coefficients $a_j(k)$ are zero. In order to be precise, we say that *the base- b expansion of k has exactly r terms* if $a_{r-1}(k) \neq 0$ and $a_j(k) = 0$ for all $j \geq r$. Explicitly, r is given by $\lfloor \log_b(k) \rfloor + 1$, where the $\lfloor \cdot \rfloor$ denotes the floor function (or the *integer value function*). In particular, the base-2 expansion of the integer 43 has exactly 6 terms.

Closely connected to the base- b expansion is the *radical inverse function*, ψ_b , which can be characterized as the mapping from \mathbb{N} into $[0, 1)$ that operates by “flipping the coefficients about the base- b decimal point to get the base- b fraction $.a_0a_1a_2\dots$ ”. Broadly speaking, the coefficients of the expansion of k in base b is treated as coefficients of an expansion in the inverse base, b^{-1} . That way, the most rapidly changing digit in k is taken as the most significant digit of the fraction. Succeeding values of k thus result in a spread-out order of fractions from $[0, 1)$. Expressed in mathematical terms, we have that

$$\psi_b(k) = \sum_{j=0}^{\infty} \frac{a_j(k)}{b^{j+1}}, \quad (19)$$

where a_j is the j term of the b -ary expansion of k . The radical inverse function is thus nothing but a “conversion” of the base- b floating-point value $.a_0a_1a_2\dots$ back to decimal. The term *radical* is due to the fact that the base of a number system is alternatively referred to as *the radix* (and the coefficients $a_0a_1a_2\dots$ are then usually known as the *mantissa*).

In order to complete the example of the integer 43 (or 101011 in binary form), we mention that the above formula (19) in particular means that $\psi_2(43)$ is 0.110101 in binary form, which then equals 0.828125 in decimal form³.

Using the radical inverse function as defined above, we can present the Van der Corput sequences in the following way:

Definition 3.1.1 (Van der Corput sequence) *Let $\psi_b(k)$ denote the radical inverse function. The Van der Corput sequence in base b is then*

$$0 = \psi_b(0), \psi_b(1), \psi_b(2), \dots \quad (20)$$

The figure below shows the first 16 terms of the Van der Corput sequence in base $b = 2$. Listed above each point is the position of the term in the sequence. For instance, the point $1/8$ is the 8th term of the sequence (i.e., $\psi_2(8) = 1/8$). As the figure illustrates, the Van der Corput sequence in base 2 fills $[0, 1)$ in an alternating fashion; every other term is assigned respectively to the left and right part of the interval, as the most significant digit of the fraction is inverted at each step.

³Binary 0.110101 equals 0.820125 in decimal since $1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} + 0 \cdot 2^{-5} + 1 \cdot 2^{-6} = 53/64 = 0.828125$.

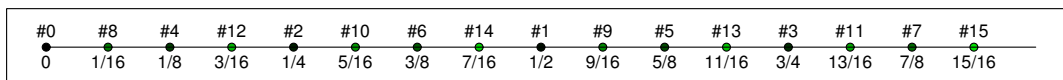


Figure 3: The first 16 terms of the Van der Corput sequence in base 2. The position of the term in the sequence is listed above each point.

In general, the “mixing” of the sequences becomes slower for increasing base b , as the most significant digit is altered less frequently, and a higher number of points is therefore needed to achieve uniformity. In particular, the first 16 terms of the Van der Corput sequence in base 16 is identical to the terms above, but in this case, the sequence appears in increasing order from left to right.

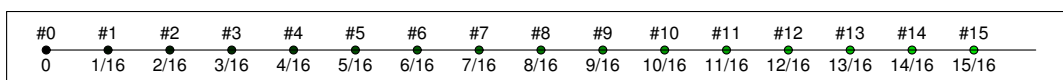


Figure 4: The first 16 terms of the Van der Corput sequence in base 16. The position of the term in the sequence is listed above each point.

The term *cycle length* is sometimes used to denote the length of a segment that covers the domain in a balanced way, and the length will thus increase with the base. Nevertheless, the Van der Corput sequences are low-discrepancy sequences in all bases, as the first n terms have a star discrepancy of $O(\log n/n)$, as shown by Niederreiter (1992). In fact, the Van der Corput sequences are digital $(0, 1)$ -sequences, and $(0, m, 2)$ -nets known as *Hammersley nets* can easily be derived from these. We will treat the concept of (t, d) -sequences and (t, m, d) -nets in Section 3.3.

3.2 A simple extension - Halton sequences

Before we proceed to the more popular digital nets and lattice rules, we will present a simple extension of the Van der Corput sequences to low-discrepancy sequences in arbitrary dimension d .

Definition 3.2.1 (Halton sequence) *Let b_1, \dots, b_d be relative prime integers⁴ greater than 1, and let $\psi_b(k)$ denote the radical inverse function in base b . The Halton sequence in bases b_1, \dots, b_d is then*

$$x_k = (\psi_{b_1}(k), \psi_{b_2}(k), \dots, \psi_{b_d}(k)), \quad k = 0, 1, 2, \dots \quad (21)$$

Smaller bases will provide more rapid mixing than larger bases, a property that is inherited from the Van der Corput sequences. Consequently, b_1, \dots, b_d will be taken to be the first d prime numbers. Of course, the Halton sequences will then become less efficient for increasing dimension d in the same way as the uniformity of the Van der Corput sequences are complicated by increasing base b . This is illustrated in the figure below, where three 2-dimensional projections of the first 1000 points of Halton sequence in dimension $d = 40$

⁴Two integers m and n are relative prime if they share no positive factors (divisors) other than 1.

are plotted. The projections are taken onto the first, middle and last pairs of coordinates, respectively.

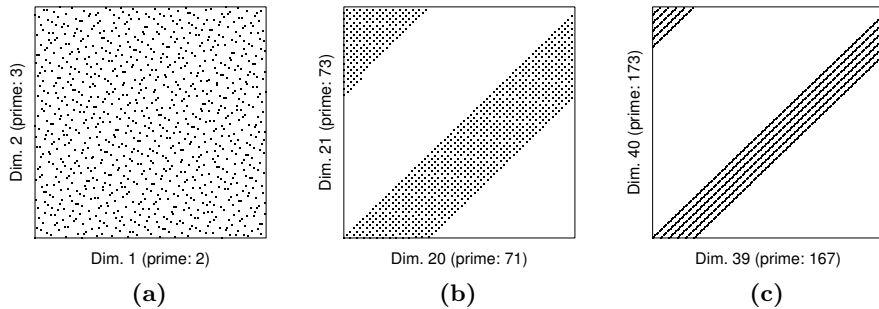


Figure 5: The first 1000 points of Halton sequence in dimension $d = 40$ projected onto coordinates (a): 1 and 2, (b): 20 and 21, (c): 39 and 40.

As the above plots show, the drift from uniformity is considerable for coordinates corresponding to large bases on relatively small samples. As a compensation, slightly altered sequences known as *leaped* Halton sequences have been proposed, but even so, the Halton sequences are often found to be inferior to other low-discrepancy sequences (see p. 295 of Glasserman (2003)).

3.3 Digital nets

As accounted for in the above section, the principle behind the Halton sequences is a direct combination of Van der Corput sequences in a set of (relative prime) bases b_1, \dots, b_d . A similar, but far more sensible approach is to construct low-discrepancy sequences from *permutations* of the Van der Corput sequence in a single base b . This is the fundamental idea behind most popular constructors of low-discrepancy sequences. Often, the permutations can be represented through so-called *generator matrices*, and for such cases, the constructed points are commonly referred to as *digital nets* or *digital sequences* (Glasserman, 2003; L'Écuyer, 2003). Without stressing their analytical importance, we will here give a simple presentation of the notions of (t, m, d) -nets and (t, d) -sequences. Niederreiter (1992) provides a more thorough formulation of the theory, and an extensive analysis of the properties of the constructs.

In order to give concise definitions, we will first introduce the b -ary boxes in dimension d (sometimes also called *elementary intervals in base b*). A b -ary box is a subset of $[0, 1)^d$ on the form

$$\prod_{i=1}^d \left[\frac{a_i}{b^{k_i}}, \frac{a_i + 1}{b^{k_i}} \right), \quad (22)$$

where a_i, k_i are non negative integers such that $a_i < b^{k_i}$. A b -ary box is thus a particular type of rectangles in $[0, 1)^d$, spanning unitary segments of multiples of powers of $1/b$ in each dimension, so in a sense, they represent “consecutive b -fold truncations” of the unit hypercube. It follows immediately from (22) that the volume of such a box is $1/b^{k_1 + \dots + k_k}$. Definitions of (t, m, d) -nets and (t, d) -sequences are given below.

Definition 3.3.1 ((t,m,d)-net) Let $m \geq t$ be non negative integers. A (t,m,d) -net in base b is a set of b^m points in $[0,1)^d$ such that each b -ary box of volume b^{t-m} contains exactly b^t of the points.

Definition 3.3.2 ((t,d)-sequence) Let $t \geq 0$ be an integer. A sequence $\{x_i\}_{i=0}^{\infty}$ of points in $[0,1)^d$ is a (t,d) -sequence in base b if, for all integers $q \geq 0$ and $m > t$, the points of the sub sequence $\{x_i\}_{i=qb^m}^{(q+1)b^m-1}$ is a (t,m,d) -net in base b .

Thus a (t,d) -sequence is essentially an infinite stream of (t,m,d) -nets, simultaneously for all $m > t$. As shown by Niederreiter (1992), all (t,d) -sequences are low-discrepancy sequences. To illustrate the above constructs, we will give an example of a $(0,4,2)$ -net in base 3. The example is inspired by a figure of Owen (2003).

Example 3.3.1:

Let us consider a $(0,4,2)$ -net in base 3 ($t = 0, m = 4, d = 2, b = 3$). The relevant 3-ary boxes are in this case those of volume $1/b^{m-t} = 1/3^4 = 1/81$, and each box should then contain exactly $b^t = b^0 = 1$ point. To satisfy the volume restriction, the sum of k_1 and k_2 has to equal 4, and the possible values of (k_1, k_2) for the 3-ary boxes are then $(0,4), (1,3), (2,2), (3,1)$ and $(4,0)$, each corresponding to a particular shape. Different values of a_i for the boxes simply represent shifts (“unit” translations), so for the purpose of visual conception, it suffice to consider the cases where $a_i = 0$.

The $b^m = 3^4 = 81$ points of a $(0,4,2)$ -net in base 3 is plotted in Figure 6. The corresponding 3-ary boxes are also indicated; one of each of the 5 different types is shaded (in particular, the anchored boxes where $a_1 = a_2 = 0$). Reference lines are included for ease of interpretation. In a way, the net is similar to the solution configuration of the modern *Su Doku* puzzles. \square

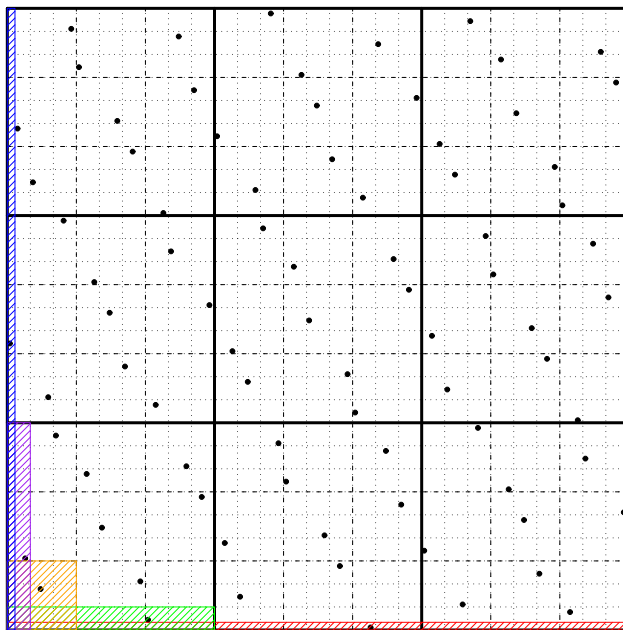


Figure 6: The points of a $(0,4,2)$ -net in base 3.

As Niederreiter (1992) points out, any (t, m, d) -net in base b is also a (u, m, d) -net in base b for integers $t \leq u \leq m$, and any (t, d) -sequence in base b is also a (u, d) -sequence in base b for integers $t \leq u$. For fixed m, d and b , smaller values of t are thus preferable, since this then indicate greater uniformity. The optimal would be to have $t = 0$, so that each b -ary box of size $1/b^m$ would contain exactly one point, but in practice, a t -value of zero may not be feasible. Similarly, if m, d and t are fixed, the smallest possible base b is preferred, since the size of the balanced point sets is given as b^m , so for larger b , a larger number of points is needed for the property of uniformity to hold.

The plots of Figure 7 illustrate four different (t, m, d) -nets. The points are taken from a class of low-discrepancy sequences known as the *Faure sequences*, which we will examine shortly. In fact, all Faure sequences are $(0, d)$ -sequences, and so the net property will hold for all (integer) values of $t \geq 0$. The position of the points in the sequences is included exclusively as a reference.

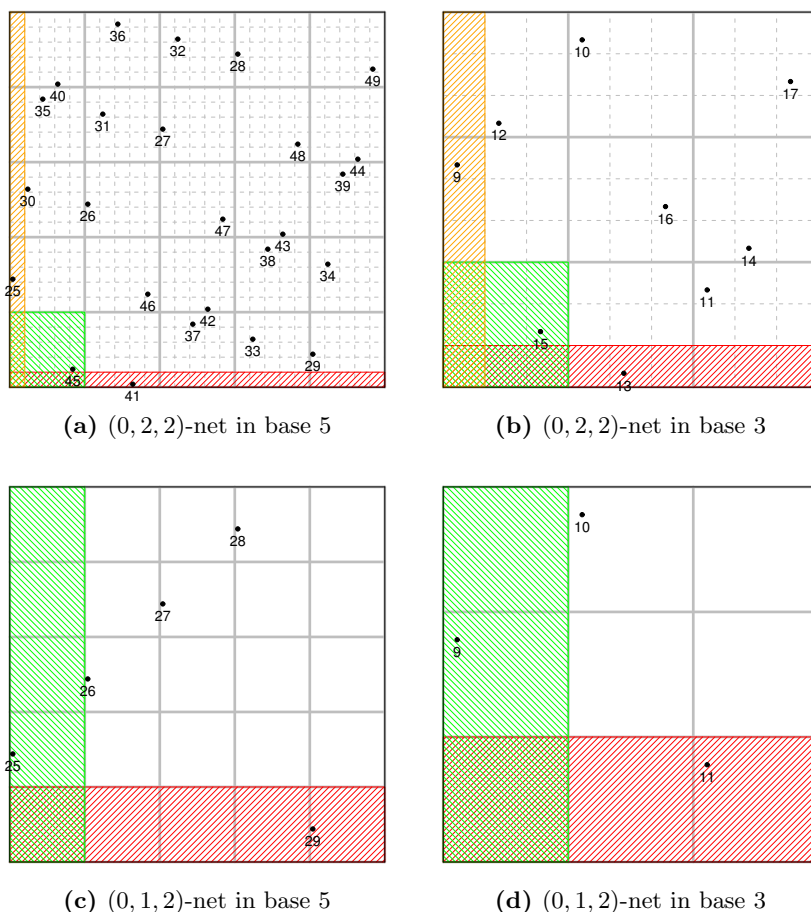


Figure 7: Some $(0, m, d)$ -nets. The nets of the left column of subfigure (i.e., subfigure (a) and subfigure (c)) are in base $b = 3$, while the nets of the right column are in base $b = 5$. The points are taken from two-dimensional Faure sequences in the corresponding bases, and the numbers associated with each point indicates the position in the sequence. The nets of the upper row have points for $m = 2$, while the nets of the lower row have points for $m = 1$.

The most popular constructions of (t, d) -sequences are due to Faure, Sobol' and Niederreiter. We will examine the former two in the following subsections. As each coordinate of these sequences follows the same construction, we will use (u) as superscript on the various quantities to indicate the particular coordinate $u \in \{1, 2, \dots, d\}$.

3.3.1 Faure sequences

The *Faure sequences* are low-discrepancy sequences constructed from permutations of the Van der Corput sequence in a single base b , where the base b is a prime at least as great as the dimension d . Since small bases are preferable (ref. Section 3.2), we will follow the common practice of taking b as *the smallest prime greater than or equal to d* . Without providing proper motivation, we will here present an explicit construction of the Faure sequences.

Let $a_j(k)$ denote the j th coefficient of the base- b expansion of the integer k , as given by (18), and define for each coordinate $u \in \{1, 2, \dots, d\}$

$$\phi_b^{(u)}(k) = \sum_{i=1}^{\infty} \frac{y_i^{(u)}(k)}{b^i}, \quad (23)$$

where

$$y_i^{(u)}(k) = \sum_{j=0}^{\infty} \binom{j}{i-1} (u-1)^{j-i+1} a_j(k) \pmod{b}, \quad (24)$$

under the following convention for binomial coefficients

$$\binom{m}{n} = \begin{cases} \frac{m!}{(m-n)!n!}, & m \geq n, \\ 0, & \text{otherwise.} \end{cases}$$

For each k , only a finite number of coefficients a_j are nonzero. Without loss of generality, we will assume that the base- b expansion of k has exactly r terms, and effectively, the sums in (23) and (24) are over a finite number of terms. As explained by Glasserman (2003), the expression (24) may therefore be seen as the result of the matrix-vector calculation

$$\begin{pmatrix} y_1^{(u)}(k) \\ y_2^{(u)}(k) \\ \vdots \\ y_r^{(u)}(k) \end{pmatrix} = C^{(u-1)} \begin{pmatrix} a_0(k) \\ a_1(k) \\ \vdots \\ a_{r-1}(k) \end{pmatrix} \pmod{b}, \quad (25)$$

where $C^{(u)}$ is the $r \times r$ matrix with elements

$$C_{m,n}^{(u)} = \begin{cases} \binom{n-1}{m-1} u^{n-m}, & n \geq m, \\ 0, & \text{otherwise.} \end{cases} \quad (26)$$

We use the convention that 0^j is 1 for $j = 0$, and zero otherwise, and so $C^{(0)}$ is the identity matrix. The matrices $C^{(u)}$ are usually referred to as *the generator matrices*. For

both (24) and (25), the modulus operator should be applied posteriorly, that is, after all summations and multiplications are conducted. In fact, it could also be applied to one or both of the operands of a summation/multiplication, as long as it is also used on the result. On vectors or matrices, the operator is applied element-wise.

From (23) and the definition given in (19) it follows that $\phi_b^{(u)}(k)$ is nothing but the radical inverse of the integer $y^{(u)}(k) = \sum_{i=1}^r y_i^{(u)}(k)b^i$, and the coefficients $y_i^{(u)}(k)$ are basically linear combinations of $a_0(k), \dots, a_{r-1}(k)$, as can be seen from (25).

Irrespective of how the calculations are interpreted, we can define the Faure sequences from (23) in the following way.

Definition 3.3.3 (Faure sequence) *Let $b \geq d$ be a prime, and let $\phi_b^{(u)}(k)$ be given by (23). The d -dimensional Faure sequence in base b is then*

$$x_k = (\phi_b^{(1)}(k), \phi_b^{(2)}(k), \dots, \phi_b^{(d)}(k)), \quad k = 0, 1, 2, \dots \quad (27)$$

It can be shown that the Faure sequences are $(0, d)$ -sequences, which then implies low-discrepancy. Thus, the Faure sequences optimize the uniformity parameter t in a base at least as large as the smallest prime greater than or equal to the dimension d . From Definition 3.3.2 it then follows that any set of Faure points of the form

$$\{x_k : qb^m \leq k < (q+1)b^m\}, \quad q \geq 0, m \geq 1, \quad (28)$$

is a $(0, m, d)$ -net. As mentioned in the previous section, the nets of Figure 7 are all taken from Faure sequences, and each column of plots have points from the same sequence since the bases are equal. For instance, the subfigures (a) and (c) can thus be seen to illustrate the property stated by (28); the points from the two-dimensional Faure sequence in base $b = 5$ for $k = 1(5^2), \dots, 2(5^2) - 1$ form a $(0, 2, 2)$ -net, and correspondingly, the points for $k = 5(5^1), \dots, 6(5^1) - 1$ form a $(0, 1, 2)$ -net.

Under the additional restriction of $q < b$, the nets given by (28) are called *Faure nets*, and all one-dimensional projections of such nets are identical, as each in fact is a permutation of the corresponding segment of the Van der Corput sequence in base b . Consequently, the $(0, m, d)$ -nets of subfigures (a) and (b) of Figure 7 classify as Faure nets, while the $(0, m, d)$ -nets of subfigures (c) and (d) do not. The discriminant of the classification is then the position q of the nets “in the stream of nets”, and since both the nets of subfigures (a) and (b) corresponds to $q < b$, the one-dimensional projections of these point sets are identical. In particular, the one-dimensional projections of the points of subfigures (a) is the segment of the Van der Corput sequence in base $b = 5$ given by $k = 25, \dots, 49$, while the points of subfigures (b) correspond to the segment of the Van der Corput sequence in base $b = 3$ for $k = 9, \dots, 17$.

As Glasserman (2003) points out, the above mentioned properties of the Faure nets follow from the so-called *cyclic properties* of the generator matrices $C^{(u)}$. Amongst other things, these properties simplify the construction of the Faure sequences, as the matrices $C^{(u)}$ then can be constructed recursively. We will not examine this matter here, but refer instead to the discussion in Glasserman (2003).

Figure 8 shows some two-dimensional projections of the 31-dimensional Faure net in base 31 for $k = 14(31)^2, \dots, 15(31)^2 - 1$, which then constitute a $(0, 2, 31)$ -net. As the plots indicate, the projection onto the i th and j th axis depends only on the distance $(i - j) \bmod b$. In particular, the first and last plots are identical, since modulo 31, 1 is the

successor of 31. This can also be explained by the cyclic properties of the generator matrices, and again, we refer to the discussion of Glasserman (2003) for further details.

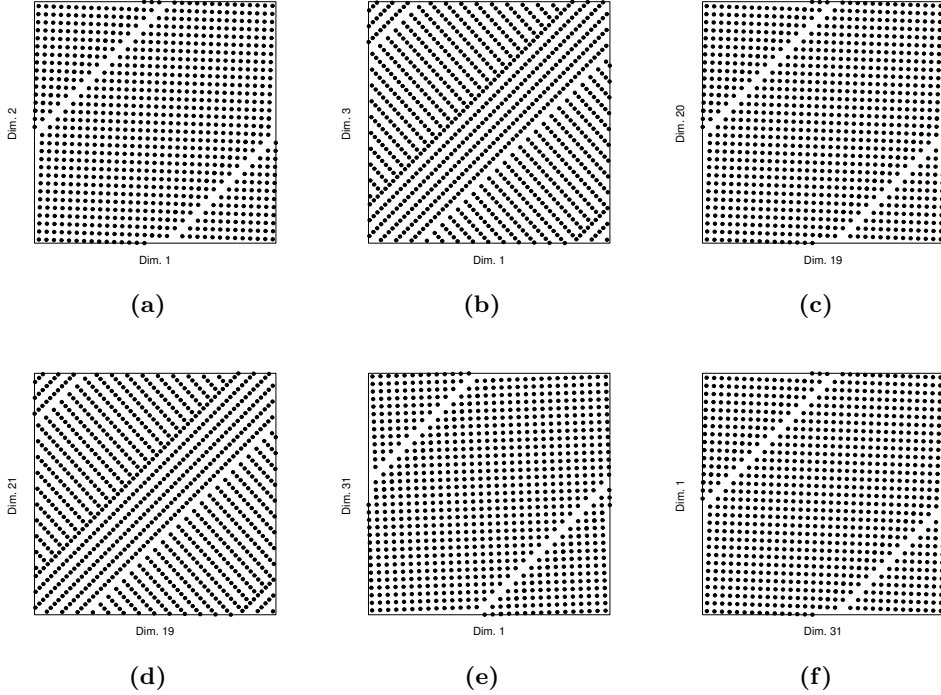


Figure 8: Projections of 961 Faure points in dimension 31 onto coordinates (a): 1 and 2, (b): 1 and 3, (c): 19 and 20, (d): 19 and 21, (e): 1 and 31, (f): 31 and 1. The points correspond to $k = 14(31)^2, \dots, 15(31)^2 - 1$.

3.3.2 Sobol' sequences

As described in the previous section, the Faure sequences are $(0, d)$ -sequences in a prime base as least as large as the dimension d . The *Sobol' sequences* are also (t, d) -sequences, but then in a fixed base $b = 2$, and with a value of t that depends on the dimensions d . While the Faure sequences have an optimal value of the uniformity parameter t , the Sobol' sequences have an optimal base b .

The construction of the Sobol' sequences can be represented in the same way as that of the Faure sequences, the “only” differences being other generator matrices and a fixated base.

Let $a_j(k)$ denote the j th coefficient of the binary expansion of the integer k , as given by (18) with $b = 2$, and define for each coordinate $u \in \{1, 2, \dots, d\}$

$$\phi_2^{(u)}(k) = \sum_{i=1}^r \frac{y_i^{(u)}(k)}{2^i} \quad (29)$$

where r is the number of terms in the binary expansion of k , and

$$\begin{pmatrix} y_1^{(u)}(k) \\ y_2^{(u)}(k) \\ \vdots \\ y_r^{(u)}(k) \end{pmatrix} = V^{(u)} \begin{pmatrix} a_0(k) \\ a_1(k) \\ \vdots \\ a_{r-1}(k) \end{pmatrix} \pmod{2}, \quad (30)$$

for generator matrices $V^{(u)}$ of dimension $r \times r$.

For each coordinate u , the columns of the corresponding generator matrix $V^{(u)}$ are the digits of a set of binary fractions $v_1^{(u)}, v_2^{(u)}, \dots, v_r^{(u)}$. These binary fractions are commonly referred to as *direction numbers*, and a different set of numbers is needed for each coordinate. We will illustrate this with an example given by Glasserman (2003). The first five direction numbers for the coordinate u are there taken as

$$v_1^{(u)} = 0.1, \quad v_2^{(u)} = 0.11, \quad v_3^{(u)} = 0.0111, \quad v_4^{(u)} = 0.1111, \quad v_5^{(u)} = 0.00101,$$

so that the corresponding generator matrix is

$$V^{(u)} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (31)$$

Since the binary representation of $k = 1, 2, \dots, 31$ all have $r \leq 5$, the generator matrix given by (31) can be used to calculate $\phi_2^{(u)}(k)$ for all of these.

The generator matrices is fully specified by the sets of direction numbers, and so is the Sobol' sequences. For suitable selections of direction numbers (or equivalently, generator matrices), the Sobol' sequences can be defined from (29) in the following way.

Definition 3.3.4 (Sobol' sequence) *Let $\phi_2^{(u)}(k)$ be given by (29) for an appropriate set of generator matrices $V^{(u)}$, $u = 1, 2, \dots, d$. The Sobol' sequence is then*

$$x_k = (\phi_2^{(1)}(k), \phi_2^{(2)}(k), \dots, \phi_2^{(d)}(k)), \quad k = 0, 1, 2, \dots \quad (32)$$

In the above definition, there is no limit on the value of k , and the generator matrices should consequently be of infinite dimension (since r goes to infinity with k). In practical computations, when generating a bulk of Sobol' points, say, for $k = k_a, k_a + 1, \dots, k_b$, the generator matrices would be accommodated for the greatest integer, k_b . As stated in Section 3.1, r grows rather slowly with k . For instance, generator matrices of dimension 20×20 can be used to construct the Sobol' points for $k = 0, 1, \dots, 1048575$.

As the above definition makes clear, the selection of the sets of direction numbers is the essential part of the generation of the Sobol' sequences. The remainder of this subsection will therefore be devoted to describing a general strategy for choosing proper sets. We will have to go by a detour, and even though some of the intermediate steps may seem slightly technical, the goal is simply to determine the necessary sets of direction numbers for the generator matrices.

Sobol's method for choosing the set of direction numbers starts by selecting a so-called *primitive polynomial over the binary arithmetic* for each coordinate. That is, a polynomial

$$P^{(u)}(x) = x^q + c_1^{(u)}x^{q-1} + \dots + c_{q-1}^{(u)}x + 1, \quad (33)$$

with coefficients $c_i^{(u)} \in \{0, 1\}$, that satisfies the following two properties with respect to binary arithmetic:

- (i) the polynomial is irreducible (i.e., it cannot be factored),
- (ii) the smallest power p for which the polynomial divides $x^p + 1$ is $p = 2^q - 1$.

Property (i) states that the constant term 1 of (33) must indeed be present. By definition, the degree of the polynomials is the largest power q with a nonzero coefficient. For instance, the polynomials

$$x + 1, \quad x^2 + x + 1, \quad x^3 + x + 1, \quad x^3 + x^2 + 1$$

are the primitive polynomials of degree one, two and three. A different primitive polynomial is needed for each coordinate $u \in \{1, 2, \dots, d\}$, and so the degree q of the polynomials will in general differ between the coordinates. That is, $q = q(u)$, but we will suppress this in the notation at this point. In fact, the primitive polynomials are usually assigned in order of increasing degree to the different coordinates. Under this convention, the above polynomials would from left to right be assigned to the 1st, 2nd, 3rd and 4th coordinates, respectively.

As an example, the polynomial $P(x) = x^3 + x^2 + 1$ is a primitive polynomial of degree $q = 3$. Property (ii) then states that $x^p - 1$ is divisible by $P(x)$ for $p = 7$, but not for $p < 7$. To substantiate somewhat (for the case of $p = 7$), we mention that

$$\frac{x^7 + 1}{x^3 + x^2 + 1} = x^4 + x^3 + x^2 + 1 \quad \text{mod } 2,$$

since modulo 2, we have that $2 = 0$, so

$$(x^4 + x^3 + x^2 + 1)(x^3 + x^2 + 1) = x^7 + 2(x^6 + x^5 + x^4 + x^3 + x^2) + 1 = x^7 + 1.$$

The primitive polynomials can be compactly be represented as integers, where the binary form then gives the coefficients. For instance, the integer 143 with the binary form 10001111 encodes the polynomial $x^7 + x^3 + x^2 + x + 1$. Table 1 lists the primitive polynomials up to degree 6. Following Glasserman, we have included a zero-degree polynomial.

Degree	Primitive polynomials modulo 2
0	1 (i.e., 1)
1	3 (i.e., $x + 1$)
2	7 (i.e., $x^2 + x + 1$)
3	11 (i.e., $x^3 + x + 1$), 13 (i.e., $x^3 + x^2 + 1$)
4	19 (i.e., $x^4 + x + 1$), 25 (i.e., $x^4 + x^3 + 1$)
5	37, 59, 47, 61, 55, 41
6	67, 97, 91, 109, 103, 115

Table 1: The primitive polynomials over the binary arithmetic of degree 6 or less. The *binary* representation of the integers gives the coefficients of the corresponding polynomial.

The number of primitive polynomials of degree q can be computed from the so-called *Euler totient function* (Bratley and Fox, 1988). Primitive polynomials are widely tabulated, and algorithms to identify them are available (Arndt, 2005).

Due to convenience, we will here introduce the binary addition operator \oplus . For non-negative integers r and s , let $a_j(r)$ and $a_j(s)$ denote the j th coefficient of their respective binary expansions, as given by (18) with $b = 2$. The binary addition of r and s , $r \oplus s$, is then defined as

$$r \oplus s = \sum_{j=0}^{\infty} t_j(r, s) 2^j, \quad (34)$$

where the coefficients are given as $t_j(r, s) = a_j(r) + a_j(s) \pmod{2}$. The operator \oplus can thus be implemented as a bitwise exclusive or (XOR) operation on the computer representation of the operands. For instance,

$$43 \oplus 143 = 164,$$

or in binary form, where \oplus then denotes the bit-wise XOR operation,

$$00101011 \oplus 10001111 = 10100100.$$

Since the Sobol' sequences in general are constructed in base $b = 2$, most implementations make additional use of bit-level operations to increase the efficiency (Glasserman, 2003). From the coefficients of the primitive polynomial $P^{(u)}(x)$ we can then define the recurrence relation

$$m_j^{(u)} = 2c_1^{(u)} m_{j-1}^{(u)} \oplus 2^2 c_2^{(u)} m_{j-2}^{(u)} \oplus \dots \oplus 2^{q-1} c_{q-1}^{(u)} m_{j-q+1}^{(u)} \oplus 2^q m_{j-q}^{(u)} \oplus m_j^{(u)}, \quad (35)$$

for integers $m_j^{(u)}$ with $j > q$. By specifying initial values $m_1^{(u)}, m_2^{(u)}, \dots, m_q^{(u)}$ for the recurrence equation (35), the direction numbers for dimension u are taken to be

$$v_j^{(u)} = \frac{m_j^{(u)}}{2^j}, \quad j = 1, 2, \dots \quad (36)$$

Hence the problem of selecting (infinite) sets of binary fractions has been reduced to selecting the sets of initial values $m_j^{(u)}$ for $j = 1, \dots, q(u)$ for each dimension u . A minimal requirement is that each of these initializing integers is taken odd and less than 2^j . Then all the subsequent integers $m_j^{(u)}$ given by the recurrence equation (35) will share this property, and the direction numbers given by (36) will be from the interval $(0, 1)$.

As explained in Glasserman (2003), if two different coordinates is initialized with the same values m_1, \dots, m_p for some value p , then the first $2^p - 1$ terms of the sequence would have the same values for these coordinates. Consequently, some care should be taken when the sets of initial values are chosen. Sobol' (1976) presented a condition for selecting the sets of initial numbers, and Chi et al. (2005) refers to other strategies.

We will not give any further account of the selection of good initializing values, but refer instead to the references of Glasserman (2003) and Chi et al. (2005). As explained there, most implementations make use of the so-called *gray code* to simplify the construction of the Sobol' sequences. We will wrap up this subsection with Table 2, listing the initial values for the primitive polynomials of Table 1 as given by Bratley and Fox (1988).

u	m_1	m_2	m_3	m_4	m_5	m_6	m_7	u	m_1	m_2	m_3	m_4	m_5	m_6	m_7
1	1	(1)	(1)	(1)	(1)	(1)	(1)	11	1	3	5	1	15	(19)	(113)
2	1	(3)	(5)	(15)	(17)	(51)	(85)	12	1	1	7	3	29	(51)	(47)
3	1	1	(7)	(11)	(13)	(61)	(67)	13	1	3	7	7	21	(61)	(55)
4	1	3	7	(5)	(7)	(43)	(49)	14	1	1	1	9	23	37	(97)
5	1	1	5	(3)	(15)	(51)	(125)	15	1	3	3	5	19	33	(3)
6	1	3	1	1	(9)	(59)	(25)	16	1	1	3	13	11	7	(101)
7	1	1	3	7	(31)	(47)	(109)	17	1	1	7	13	25	5	(255)
8	1	3	3	9	9	(57)	(43)	18	1	3	5	11	7	11	(103)
9	1	3	7	13	3	(35)	(89)	19	1	1	1	3	13	39	(65)
10	1	1	5	11	27	(53)	(69)								

Table 2: Initial values for the primitive polynomials of Table 1. The numbers in the parentheses need not be specified, since they can be determined from the recurrence equation.

Two-dimensional projections of a Sobol' sequence in dimension $d = 29$ are shown in Figure 9, and the non-uniformity of the projections corresponding to the later coordinates could be avoided by choosing better initializing values. As an alternative, the technique known as *scrambling* could have been employed (Chi et al., 2005).

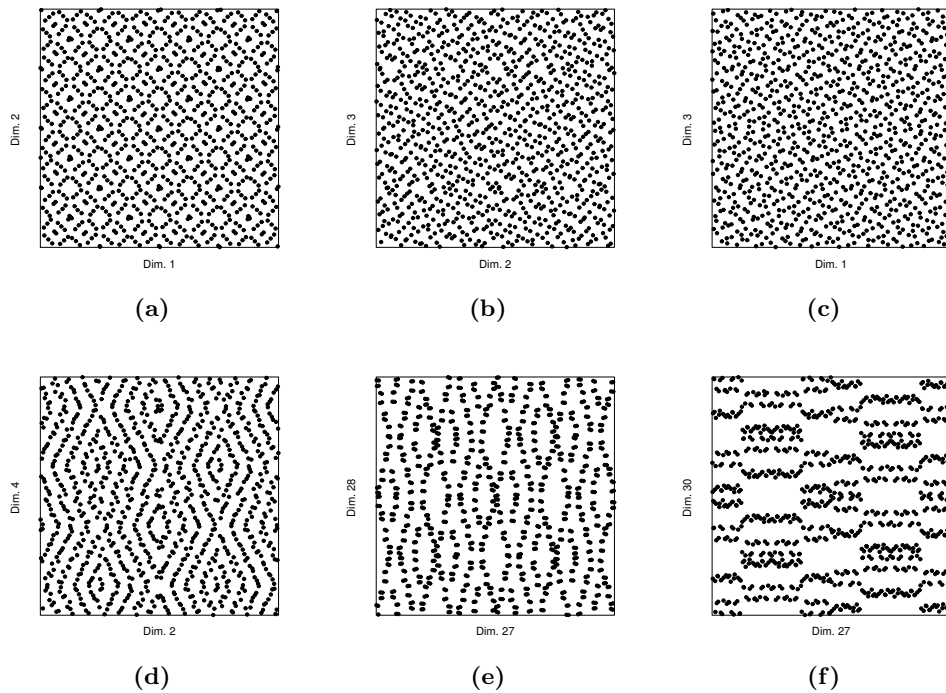


Figure 9: Projections of 1024 points of a Sobol' sequence onto coordinates (a): 1 and 2, (b): 2 and 3, (c): 1 and 3, (d): 2 and 4, (e): 27 and 29, (f): 27 and 30. The point corresponds to $k = 1, \dots, 1024$.

3.4 Lattice rules

A d -dimensional integration lattice, L , is a discrete subset of \mathbb{R}^d that is closed under addition and subtraction and which contains the integer vectors \mathbb{Z}^d as a subset. The so-called *node set* of the lattice is the set of points of L that fall inside the unit hypercube. For the QMC purpose, we will refer to the points given by the node set of a lattice as the point set of a *lattice rule*. Our presentation of the lattice rules here will be quite simple, and we refer to Glasserman (2003), Niederreiter (1992) and Hickernell et al. (2001) for a more comprehensive introduction to the use of such point sets for QMC.

In dimension d , a *rank-1* lattice rule of size n is on the form

$$\left\{ \frac{k}{n}v \pmod{1} \quad : \quad k = 0, 1, \dots, n-1 \right\}, \quad (37)$$

where v is a d -dimensional vector of integers. The multiplied vectors are taken modulo 1, so only the fractional part of each element is kept. To avoid repetition of points, n is required to be relative prime to each of the elements of the generating vector v .

More generally, an n -point lattice rule of rank r takes the form

$$\left\{ \sum_{i=1}^r \frac{k_i}{n_i} v_i \pmod{1} \quad : \quad k_i = 0, 1, \dots, n_i - 1, i = 1, 2, \dots, r \right\}, \quad (38)$$

for linearly independent vectors v_1, \dots, v_r and integers $n_1, \dots, n_r \geq 2$ where each n_{i+1} is a multiple of n_i such that $n_1 n_2 \dots n_r = n$. As for the rank-1 case, n_i must be relative prime to each of the elements of the corresponding generating vector v_i .

The use of a lattice rule does not solely guarantee that the points are well-distributed in the unit hypercube. Similar to generator matrices of the digital nets, the choice of generating vectors is essential for the quality of the lattice rule point sets. The strategy for finding good generating vectors is typically to optimize some discrepancy measure. An important example of such criteria is the so-called *spectral test*, which also is used for finding pseudo-random number generators (Hickernell et al., 2001; L'Ecuyer and Lemieux, 2000).

Due to simplicity, the rank-1 lattice rules are often preferred in practical situations. A popular class of such rules is the *Korobov rules* which we will introduce in Section 3.4.2. The Lattice rules are known to be particularly effective when the integrand f of (6) is periodic. On the other hand, less well-behaved functions may consider some special care. We refer to the discussion of Glasserman (2003) for more details.

3.4.1 Extensible lattice rules

The lattice rules primarily define low discrepancy point sets of fixed size, but extensions procedures are available. We will here present the popular method of Hickernell et al. (2001) for extending fixed-size rank-1 lattice rules to infinite sequences.

Let the size n of the rank-1 lattice rule (37) be given as $n = b^m$ for some base b and integer m . The segment $\psi_b(0), \psi_b(1), \dots, \psi_b(n-1)$ of the Van der Corput sequence in base b will then be a permutation of the coefficients k/n , $k = 0, 1, \dots, n-1$ for the d -dimensional generating vector v of (37). Consequently, the point set can just as well be formulated as

$$\{\psi_b(k)v \pmod{1} \quad : \quad k = 0, 1, \dots, n-1\},$$

and the extension to an infinite sequence is then possible by simply removing the upper limit on k . As Glasserman explains, the first b^m points in this sequence are then the original lattice point set, and each of the next $(b - 1)$ non-overlapping segments of length b^m will be so-called *shifted* versions of the original lattice. The first b^{m+1} points will again form a lattice rule on the form (37), but then with $n = b^{m+1}$. Thus the extension procedure extends and refines the original lattice.

3.4.2 Korobov rules

A particular class of the rank-1 lattice rules (37) is the *Korobov rules*, which take the generating vector v to be on the form $v = (1, \alpha, \alpha^2, \dots, \alpha^{d-1})$ for some integer α . Figure 10 illustrates the Korobov rules for four different values of α in dimension $d = 2$.

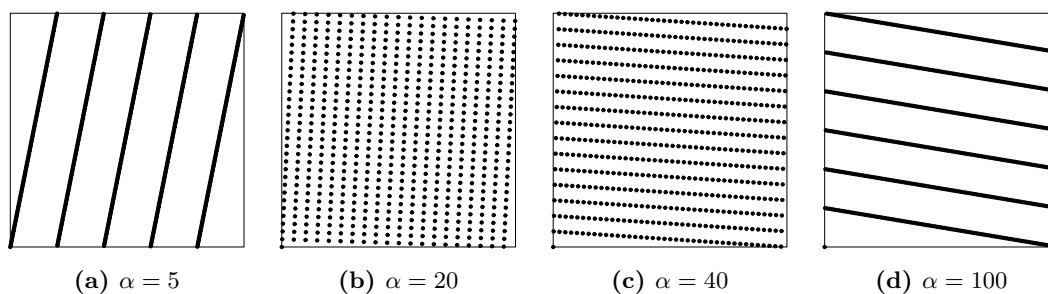


Figure 10: The $n = 601$ points of the Korobov rules in dimension $d = 2$ for different values of the integer parameter α .

4 Possible extensions

In the previous section we have introduced some popular techniques for generating sample points that are suitable for QMC. The essential property of such point sets is that of low (star) discrepancy; the points are intended to give a uniform representation of the unit hypercube. As indicated by the Koksma-Hlawka inequality (8), the error of the QMC approximation (6) should then be correspondingly small. This motivates the use of QMC rather than MC in a variety of situations. On the other hand, the estimation of the integration error is far more complex for QMC than for MC. In addition, the QMC methods are genuinely challenged by high dimension. Several remedying procedures have been proposed in order to enhance the robustness of the QMC methods. We will here report the two main concepts known as *randomization* and *hybridization*.

4.1 Randomization

The transition from MC to QMC is popularly referred to as *a transition from randomness to determinism*. In principle, this is indeed the case. It may therefore seem rather strange to propose a randomization of the QMC points, since they are so carefully selected in the first place. There are at least two good reasons for doing so. The most apparent reason is the possibility of measuring error through a confidence interval similarly as for the classical Monte Carlo methods. Even though it is possible to operate with fully deterministic error bounds for QMC, this can be very inconvenient in practice, as the computational effort needed for evaluation is usually very high. The other reason is that randomization may actually reduce the approximation error in some situations. For instance, there are particular settings where the error can be reduced by randomization due to cancellation. In other situations, randomization can reduce the error by suppressing the effects of sub-optimal construction. As an example, Chi et al. (2005) finds that randomization can improve the performance of a Sobol' sequence when the choice of initial values is improper. We will not examine these aspects here, but refer instead to p. 320 of Glasserman (2003), and the references given there.

It is important to keep in mind that the motivation behind the use of all QMC methods necessarily is to reduce the approximation error compared to that of MC. As a natural extension of pure QMC, randomized QMC methods (RQMC) generally offer to exchange *only some* accuracy for an applicable estimate of the error. To avoid getting lost in details, we will only introduce the concept of randomization by stating a generic recipe given by Owen (1998b):

Let x_1, \dots, x_n be a QMC point set, and let r_i be a randomized version of x_i . The randomization should be such that the following properties hold:

- (i) $r_i \sim \text{unif}([0, 1]^d)$, $i = 1, \dots, n$,
- (ii) r_1, \dots, r_n inherits the QMC properties of x_1, \dots, x_n with probability 1.

Property (i) ensures that the QMC estimator (6) is unbiased when employed to the randomized point set, r_1, \dots, r_n , so that the computation of an asymptotically valid confidence interval for $I(f)$ is straightforward. Property (ii) simply states that the randomization should preserve the specific properties of the generating QMC point set.

Of course, different randomization procedures have been developed for different point

sets. For instance, *random shifts* is customized for randomizing QMC sets generated with lattice rules (but can be operated on other low-discrepancy point sets as well), and *random permutation of digits* and *scrambling* are designed for randomizing QMC sets corresponding to digital nets (Glasserman, 2003).

4.2 Hybrid Monte Carlo

Unquestionably, the improved convergence rate offered by QMC in many situations is attractive, but still, some of the advantages of MC prevail. For instance, the successful construction of QMC point sets with relevant QMC properties for “reasonable” sample sizes can be difficult to accomplish in very high dimensions. On the other hand, the construction of proper MC samples of arbitrary sizes is uncomplicated for *all* dimensions. Consequently, several approaches to combine the best from QMC and MC (or possibly some other suitable sampling technique) have been proposed, and in general the collective term of *hybrid Monte Carlo* is used for such methods. In a sense, randomized QMC can also be considered as a crossing between QMC and MC, but the term is usually reserved for combinations of a more governed nature.

We will here restrict our treatment of the subject to illustrating the use in one of the situations where it perhaps most naturally can be employed. So let us consider an integrand f with truncation dimension d_t that is considerably lower than the nominal dimension d . It is then possible to employ a d_t -dimensional QMC (or RQMC) rule on the “important” variables and then do something else for the rest. This type of filling-technique is often referred to as *padding*. Okten (1999) suggests to use a so-called *mixed sequence*, i.e. to fill out the remaining “slots” with pseudo-random numbers, which then is equivalent to concatenating MC vectors and QMC vectors. As an alternative, Owen (1998b) suggests to use *Latin hypercube sampling* (or *Latin supercube sampling* if the superposition dimension implies an adequate reduction of dimension). For a proper description of the procedures involving hyper- and supercube sampling, see (Owen, 1998b) or (Owen, 1998a).

5 A simple test

In this section we will examine the behaviour of some of the low-discrepancy sequences of Section 3 on a simple test case. More precisely, we will try QMC integration out using the Halton, Faure and Sobol' sequences, and the pseudo-random sequences of classical Monte Carlo will be included as a reference. We will not utilize any randomization procedures, even though randomized sequences often are found advantageous, as mentioned in Section 4.1. References to other and more comprehensive numerical investigations of QMC is given by Glasserman (2003), p. 323-324.

5.1 Test function

Let X be a d -variate random variable with a normal distribution with expectation 0 and covariance matrix Σ ,

$$X \sim N_d(0, \Sigma).$$

We will take the covariance matrix of X to be on the form

$$\Sigma = \begin{pmatrix} 1 & & & \\ & 1 & \rho & \\ & \rho & \ddots & \\ & & & 1 \end{pmatrix}, \quad (39)$$

so that the variance of each component of X is 1 and the covariance between component i and component j is ρ for $i \neq j$. The covariance matrix will here coincide with the correlation matrix, as the variance of each component is 1. We will take the parameter ρ such that $0 \leq \rho < 1$. The above matrix (39) will then be symmetric and positive definite, and consequently, it can be represented in terms of the so-called *Cholesky decomposition*. That is, there exist a unique lower triangular matrix L such that

$$\Sigma = LL^T. \quad (40)$$

We shall assume that we want to estimate the mean of the random variable $f(X)$, where $f(\cdot)$ is the mapping from \mathbb{R}^d into \mathbb{R} given by

$$f(x) = \frac{1}{d^2} \left(\sum_{j=1}^d x_j \right)^2. \quad (41)$$

By definition, the expectation of $f(X)$ is then given as

$$\mathbb{E} f(X) = \int_{\mathbb{R}^d} f(x)p(x)dx, \quad (42)$$

where $p(\cdot)$ is the density function of X . In order to approximate it by QMC we shall proceed by reformulating it as an integral over the unit hypercube.

Let U_1, U_2, \dots, U_d be independent random variables from the uniform distribution on $[0, 1)$,

$$U_i \sim Unif(0, 1) \quad i = 1, 2, \dots, d,$$

and let $\Phi(\cdot)$ be the cumulative distribution function of the standard (univariate) normal distribution,

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-x^2/2} dx. \quad (43)$$

The random variables $\Phi^{-1}(U_1), \Phi^{-1}(U_2), \dots, \Phi^{-1}(U_d)$ will then be independent random variables from the standard (univariate) normal distribution. Since the variables are independent, they will also be *simultaneously* standard (d -variate) normal distributed. That is, for $Z = \Phi^{-1}(U) = (\Phi^{-1}(U_1), \Phi^{-1}(U_2), \dots, \Phi^{-1}(U_d))^T$, we have that

$$Z \sim N_d(0, I).$$

By basic theory of linear transformations of multivariate normal distribution it then follows that

$$LZ = L\Phi^{-1}(U) \sim N_d(0, LL^T),$$

so that

$$\mathbb{E} f(X) = \int_{\mathbb{R}^d} f(x)p(x)dx = \int_{[0,1]^d} f(L\Phi^{-1}(u)) du. \quad (44)$$

The value of the rightmost integral of (44) can then be estimated by a straightforward QMC approximation as given by (6). Unfortunately, the inverse of the normal cumulative distribution function, $\Phi^{-1}(\cdot)$, is a non-linear function for which no closed-form solution exists. We will therefore have to compute the function values $\Phi^{-1}(U_i)$ numerically.

By noting that $\mathbb{E} f(X)$ is equal to the variance of the random variable $h(X) = 1/d \sum_{i=1}^d X_i$, the exact solution can be found analytically as the sum of the elements of the covariance matrix Σ divided by d^2 , i.e.,

$$\int_{[0,1]^d} f(L\Phi^{-1}(u)) du = \mathbb{E} f(X) = \frac{1}{d^2} \sum_{j=1}^d 1 + (d-1)\rho = \frac{1-\rho}{d} + \rho.$$

We will use the exact value as reference when we consider the quality of the QMC approximations.

5.2 Comments on the test function

Informally speaking, it is not unlikely that the effective dimension (ref. Section 2.3) of the integrand of (44) will be influenced by the parameter ρ in some way, as it governs the correlation between the different components of X . In particular, we expect that for values of ρ close to 1, a large proportion of the “variance” of the integrand can be accounted for by a single coordinate, so that the *truncation dimension* d_t may be low compared to the nominal dimension d . On the other hand, for values of ρ close to 0, the matrix L will be similar to the identity matrix. Since $f(x)$ is a quadratic function in x , the *superposition dimension* d_s may in such a case be of a tractable size even if d is relatively large. For values of ρ significantly different from 0 or 1, we have no reason to expect that the effective dimension will be less than the nominal dimension. In a strict sense, these arguments are inadequate for describing the actual effect of ρ on the two notions of effective dimension.

Still, from a qualitative point of view they might give some guidance to the behaviour of the QMC methods. We will therefore experiment with different values of ρ . The above considerations can be summarized as follows:

- (i) the parameter ρ taken close to 0 promotes a small d_s relative to d ,
- (ii) the parameter ρ taken close to 1 promotes a small d_t relative to d .

5.3 Some general comments

For point sets taken from low-discrepancy sequences, not all sample sizes n are equally favourable (Fox, 1986). The approximation error will in general decrease with n , but not necessarily in a monotonic way (ref. the concepts of *mixing* and *cycle length* introduced in Section 3.1). For the (t, d) -sequences of Faure and Sobol' for instance, the sample should be increased in bulks of points corresponding to the digital nets of the respective sequence. That is, n should be taken as powers of the bases of the sequences. Of course, such considerations will be especially important for the Faure points, since the Faure base will be considerably higher than the base 2 of the Sobol' points.

The mixing of the Faure sequence will be affected by the dimension d irrespective of the effective dimension of the integrand, since the base necessarily is taken as the least prime greater than or equal to the *nominal* dimension. The Faure approximations will thus in general be inferior for any incomplete cycle (unfavourable sample size) regardless of any reduction of the effective dimension. In contrast, the mixing of the Halton sequence may be influenced by the effective dimension, or more specifically, by the truncation dimension d_t . The reason for this is that the first components of the Halton points always correspond to the smallest prime bases. The cycle length of the first components will therefore be short, and these will be well-distributed quite frequently for increasing n . While the mixing of the full sequence will be slow for a large nominal dimension, the mixing of the "effective" part of the sequence may still be relatively fast.

On the other hand, the Faure sequence uses *the* first prime greater than or equal to d , while the Halton sequence uses the d first primes. As a consequence, the mixing will be much slower for the Halton sequence than for the Faure if the truncation dimension d_t is of the same size as the nominal dimension d , and d is relatively large. This can be seen from the fact that the d th prime increases much faster with d than the first prime greater than or equal to d . For instance, for $d = 100$, the first prime greater than or equal to d is 101, while prime number d is 541.

The Sobol' sequence uses base $b = 2$ in all dimensions, and may therefore be competitive with MC for modest sample sizes n even for high dimensional problems if the effective dimension is of a tractable size.

5.4 Testing procedure

Before we undertake the actual numerical tests, we will make a few comments on how we plan to conduct them and how we will present the results. The results of the tests will be reported in the next section.

To construct the Faure and Halton sequences we will use functions based on the code

snippets of Glasserman (2003) implemented in C. We will call the functions from R (R Development Core Team, 2005) using the standard C-interface. To generate Sobol' sequences, we will use a function from the R package *fOptions* belonging to the open source library of Würtz (2004). The built-in random number generator of R will be used to generate the pseudo-random sequences.

To demonstrate the convergence of the QMC approximations, we will plot the error as a function of the sample size n for $n \in \{1, 2, \dots, N\}$. That is, we will approximate the value of (44) using the n first points of the sequences (to get the cumulative mean) and then subtract the exact value to get the error of the approximation. The value of N is of course arbitrary here since we will be taking all the points from infinite sequences. The value of N could be selected at any time, either before or during the computations, and does not in any way represent a fundamental fixation of the sample size. Still, a fixation of N is needed to generate the plots. We will take the values of N sufficiently large to indicate some degree of convergence for the approximations (in a visual sense).

In line with the discussion of Section 5.3, we will not plot the Faure sequence for all sample sizes n . More specifically, we will only use the points where n is a multiple of the base b to draw the curves of the approximation error. Figure 11 demonstrates the difference between the plots where the error is plotted for all $n \in \{1, 2, \dots, N\}$ and where the error is plotted only for the values of n that are multiples of the base b . The convergence of the QMC approximations using Faure sequences will then appear to be less locally fluctuating than the actual development would entail, but the main behaviour will be even more evident. The local fluctuations for values of n within the smallest digital nets can be seen from subfigure (a).

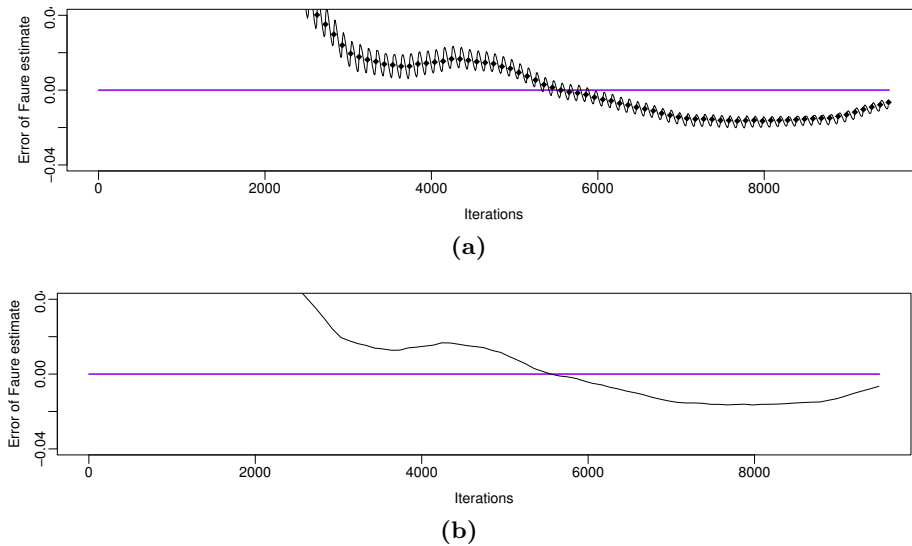


Figure 11: The behaviour of a QMC approximation using sample points from a Faure sequence for sample sizes $n \leq N = 9500$. (a): approximations plotted for all values of n . The points where n is a power of the base are indicated with small diamond. (b): approximations plotted only for the points where n is a power of the base.

The most favourable values of $n \leq N$ for a Faure sequence in base b are those where n is a multiple of b^m and m is such that $b^m \leq N < b^{m+1}$. As defined in Section 3.3.1, such points give the border points of the largest Faure nets of size less than N . We will indicate such

points with dots. The plots of Figure 11 correspond to a situation where the dimension is $d = 100$, and the maximal number of points used for the plot is $N = 8500$. For $d = 100$, the Faure base will be 101, and so the integer value of m that satisfies $b^m \leq N < b^{m+1}$ is $m = 1$. For such a case, the points that will be used to draw the Faure curves will be the same points that mark the largest Faure nets. The corresponding nets for the sequence of Figure 11 is indicated in subfigure (a) of Figure 12. In general, however, the points of the largest Faure nets will be a subset of the points used to draw the curve, and this is illustrated in the subfigure (b).

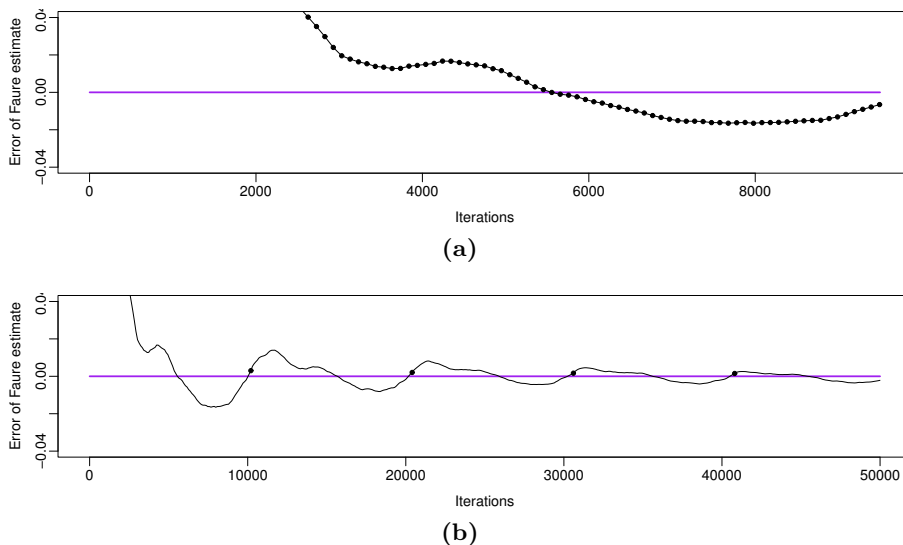


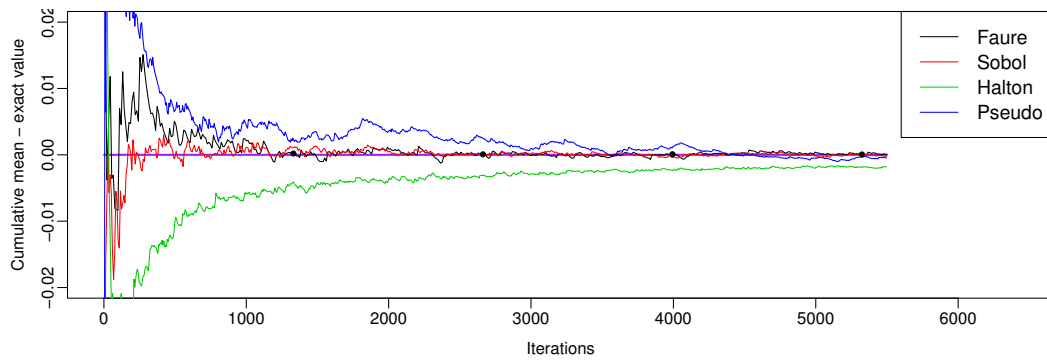
Figure 12: The most favourable points of QMC approximations using points from the Faure sequences for sample sizes $n \leq N$. (a): the largest Faure nets in base $b = 101$ for $N = 9500$. (b): the largest Faure nets in base $b = 101$ for $N = 50000$.

To avoid overloaded plots, we will use the presentation style of Figure 12 for the Faure sequences in the following. For the other sequences we will plot the approximation error for all values of $n \leq N$. By construction, the estimates of the pseudo-random sequences will be of a random nature, and consequently, each plotted curve will represent just one of all the possible realizations. These curves should accordingly be interpreted as a *possible* behaviour of the MC estimator.

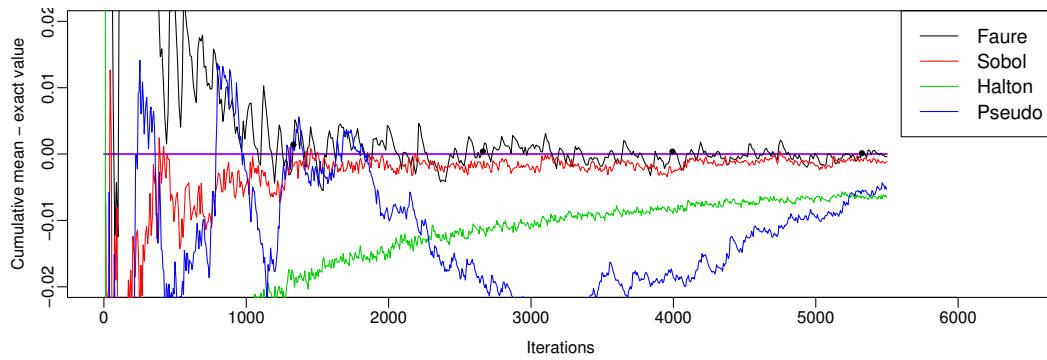
To improve the performance of the (t, s) -sequences, it is common practice to skip some of the initial points of the sequence. This is due to the so-called *leading-zeros phenomenon* (see Bratley et al. (1992); Fox (1986)). In this regard, we will start the Sobol' sequence at 2^{12} and the Faure sequence at b^4 , where b then denotes the base of the Faure sequence.

5.5 Results

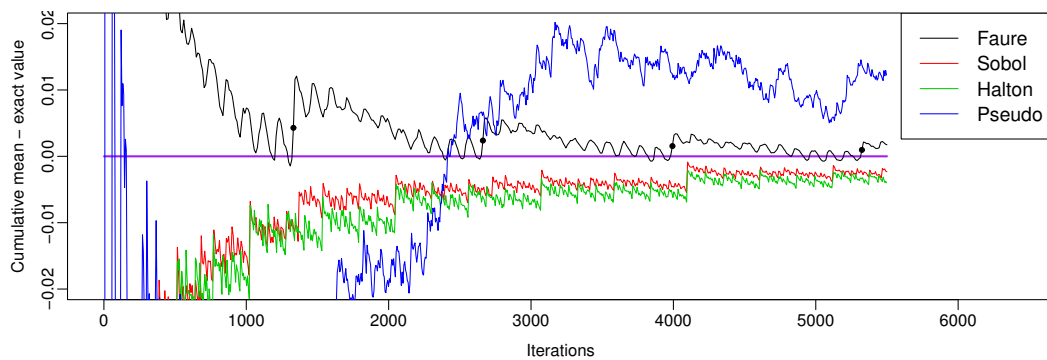
We will estimate the values of the test function as given by (44) for the situations where $d \in \{10, 50, 100\}$ and $\rho \in \{0.01, 0.50, 0.99\}$. Plots of the error of the estimates are shown on the next pages.



(a) $d = 10, \rho = 0.01, N = 5500$

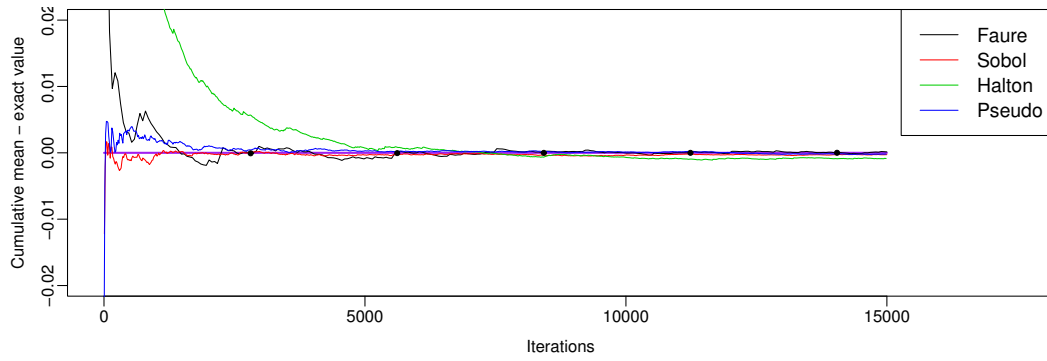


(b) $d = 10, \rho = 0.50, N = 5500$

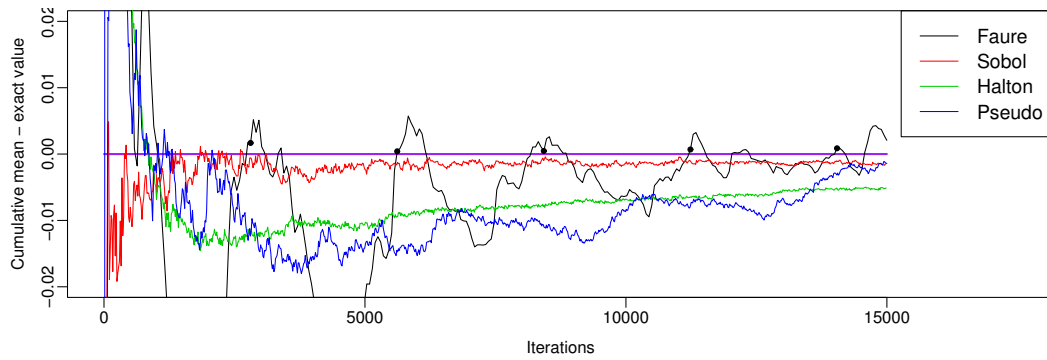


(c) $d = 10, \rho = 0.99, N = 5500$

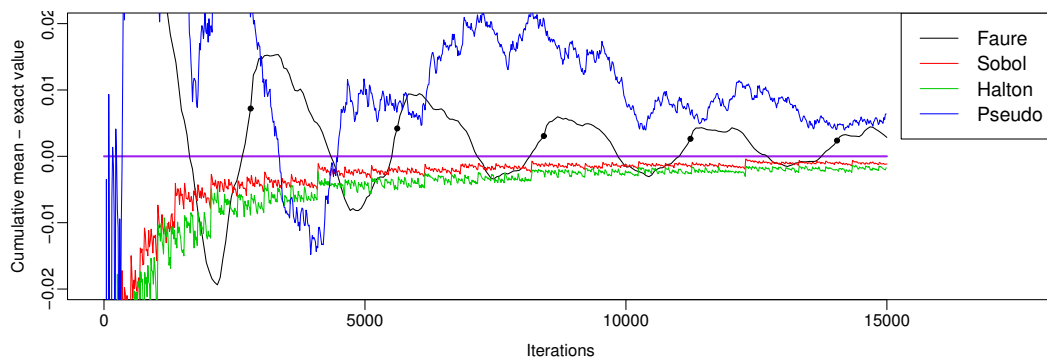
Figure 13: Estimated minus exact value of the test function for the different point sets. Plotted as a function of the sample size n for various values of the parameter ρ in dimension $d = 10$ for $N = 5500$.



(a) $d = 50$, $\rho = 0.01$, $N = 15000$

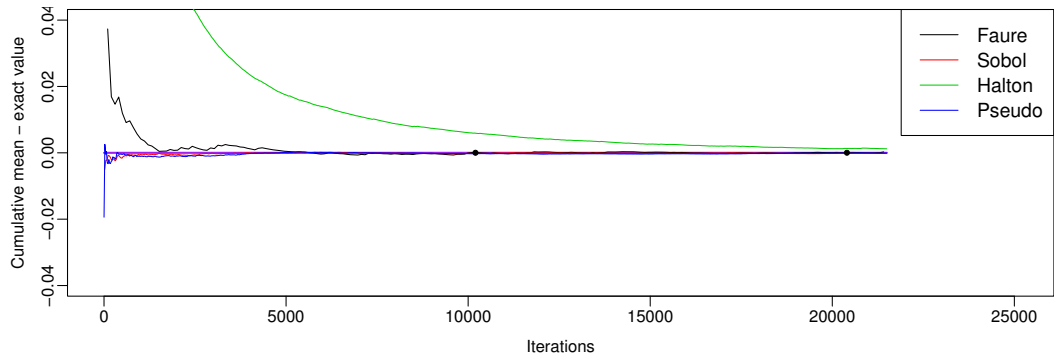


(b) $d = 50$, $\rho = 0.50$, $N = 15000$

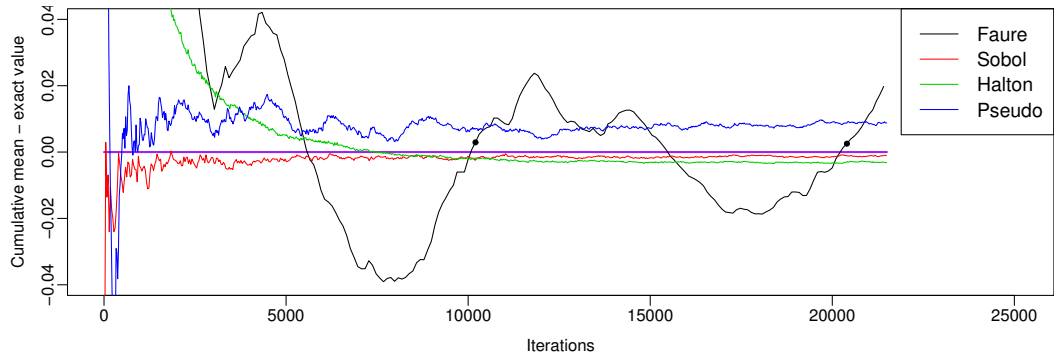


(c) $d = 50$, $\rho = 0.99$, $N = 15000$

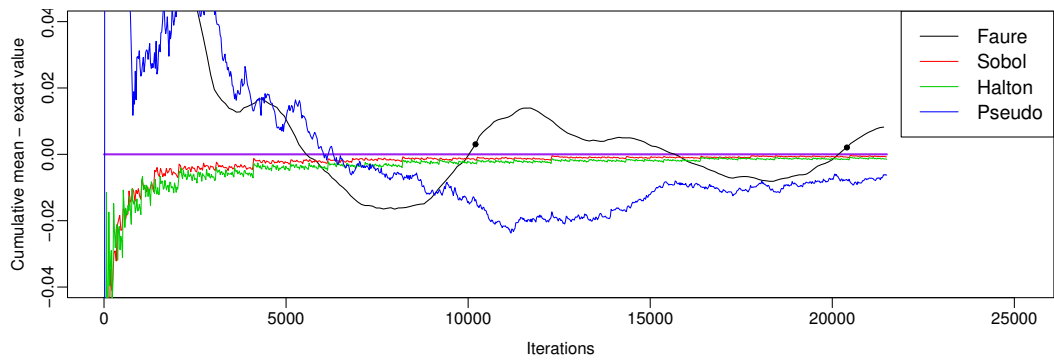
Figure 14: Estimated minus exact value of the test function for the different point sets. Plotted as a function of the sample size n for various values of the parameter ρ in dimension $d = 50$ for $N = 15000$.



(a) $d = 100, \rho = 0.01, N = 21500$



(b) $d = 100, \rho = 0.50, N = 21500$



(c) $d = 100, \rho = 0.99, N = 21500$

Figure 15: Estimated minus exact value of the test function for the different point sets. Plotted as a function of the sample size n for various values of the parameter ρ in dimension $d = 100$ for $N = 21500$.

In general, the observed behaviour of the QMC approximations in Figure 13-15 seem to be in agreement with the expected behaviour, as presented in Section 5.2-5.3. We will make a few additional comments on the performance of each of the sequences below.

- the pseudo-random sequences are apparently inferior to the low-discrepancy sequences in most of the cases considered here. The only clear exceptions are the situations for $\rho = 0.01$, where the approximation of the Halton sequences converges significantly slower than the others. This can be seen from the subfigures (a) of Figures 13-15. Still, the gain of QMC might be perhaps be seen to lessen some in the high-dimensional cases of Figure 15.
- the Halton sequences appear to be useful even in a dimension as high as $d = 100$ when the coordinates are sufficiently blended together. This can be seen from subfigures (b) and (c) of Figure 15, where the values taken for the correlation parameter ρ are 0.50 and 0.99, respectively.
- the Faure sequences seem to be quite good in general, even though some care should be taken regarding the sample size n . They provide estimates of high accuracy for all the given dimensions, but seem to be especially suitable for the low-dimensional cases, as shown by Figure 13.
- the Sobol' sequences appear to be the most flexible sequences for the given test problem. Relatively good estimates are attained fast, and the overall accuracy is satisfactory for each of the situations shown in Figure 13-15.

All the low-discrepancy sequences perform surprisingly well for the situation of subfigure (b) of Figure 15, where the dimension is $d = 100$ and the correlation parameter is $\rho = 0.50$. Presumably, this should then represent a rather difficult case, since the dimension is relatively high and the parameter ρ is taken with maximal distance to 0 and 1. The fact that the QMC methods seem to handle the situation with ease might indicate that the effective dimension of the integrand will be reduced also for the intermediate values of ρ , in addition to when it is taken close to 0 or 1. On the other hand, this does not seem all that unreasonable since a correlation of 0.5 will cause a considerable co-variation in the coordinates. Nevertheless, it is quite evident that the Faure, Sobol' and pseudo-random sequences give rise to a much faster convergence than the Halton sequence in the case of subfigure (a) of Figure 15 than in the case of the subfigure (b), and so the influence of effective dimension do appear to be present at some level.

For the given test problem, the convergence of the pseudo-random sequences are not competitive with that of the low-discrepancy sequences. The only situations where the pseudo-random sequences seem to manage are those of the subfigures (a) of Figure 13-15, where $\rho = 0.01$. This might then be partially due to the "averaging nature" of the function $f(x)$ of the integrand, as given by (41). When the coordinates is correlated, the effect of such a leveling operation will diminish, as the various terms of the sum typically will be less different. In all the other cases, the accuracy of the QMC methods are superior to that of the MC methods, as the subfigures (b) and (c) of Figure 13-15 show.

6 Summary

Both the necessary theoretical foundation and the underlying motivation of the QMC methods is expressed by the Koksma-Hlawka inequality (8), which then demonstrates how the approximation error of the normalized integral (5) can be reduced by the selection of point sets that cover the unit hypercube in a more uniform way than a typical realization of a pseudo-random sample. More specifically, the Koksma-Hlawka inequality formulates that the integration error can be bound by the so-called star discrepancy of the employed point set, which then is a measure of the non-uniformity of the point set with respect to origin-anchored, axis-parallel rectangles. This motivates the use of low-discrepancy sequences for the approximation, and the usage of such sequences is therefore usually implied when a method is characterized as a QMC method.

By definition, the low-discrepancy sequences have a star discrepancy of order $O(\log^d(n)/n)$, which then is “nearly $O(1/n)$ ” for any fixed dimension d . On the other hand, for a large d , the magnitude of such an upper bound on the star discrepancy will be relatively large unless n is huge. By giving some attention to the notions of effective dimension, as defined by (16) and (17), it is possible to gain a certain qualitative understanding of why the QMC methods can outperform MC methods even for high-dimensional problems. In particular, if either the truncation dimension d_t or the superposition dimension d_s of the integrand is tractable, then it is not unlikely that QMC may represent a significant improvement over MC in practice, even when the nominal dimension d is considerably larger.

The Van der Corput sequences (20) are simple one-dimensional low-discrepancy sequences defined for each integer base $b \geq 2$ by the so-called radical inverse function (19). The radical inverse function operates by flipping the base- b expansion of any non-negative integer about the base- b decimal point. The most rapid changing digit of the base- b expansions of these integers will therefore be taken as the most significant digit, and segments of succeeding values of the function will be scattered over the unit interval, $[0, 1)$. The distribution of the points over $[0, 1)$ will tend to be uniform quite often, but the “frequency” of the balanced configurations (the cycle length) will increase with the base. Extensions and permutations of the Van der Corput sequences are the basis of most constructions of low-discrepancy sequences.

One of the simplest ways of extending the Van der Corput sequences is that of the Halton sequences (21), which in dimension d takes the values of the Van der Corput sequences in the d lowest prime bases for the different coordinates. Due to the issue of slower mixing for higher bases, the efficiency of the Halton sequences will decrease rather rapidly for increasing dimension. For this reason, alternative procedures where low-discrepancy sequences are constructed from permutations of the Van der Corput sequence in a single base b is often preferred. When the permutations can be represented in terms of generator matrices, the sequences are commonly referred to as digital sequences. The digital sequences of Faure (27) and Sobol’ (32) are both (t, d) -sequences as defined by Definition 3.3.2. The (t, d) -sequences are essentially infinite streams of (t, m, d) -nets simultaneously for all $m > t$, and each such digital net gives a set of points that are distributed in accordance with net property of Definition 3.3.1. Broadly speaking, use of such nets ensures that the points are scattered over the unit hypercube in the “Su-Doku-solution” sense, as illustrated by Figure 6.

The Faure sequences optimize the uniformity parameter with $t = 0$, while the Sobol’ sequences optimize the base with $b = 2$. The construction of the Faure sequences is

uncomplicated for all dimensions d , as the generator matrices are fully specified, but the mixing will be reduced for increasing d . To construct good Sobol' sequences, the initializing values required to construct the generator matrices have to be selected with care. This makes the construction of the Sobol' sequences somewhat less straightforward than for the Faure sequences, but the base 2 practically removes any potential difficulty concerning cycle lengths. In addition, the binary base of the Sobol' sequences gives certain computational advantages. As a partial remedy for the problem of identifying good initializing values, some scrambling techniques are available.

The lattice rules represent an alternative strategy for generating point sets with low discrepancy, even though the extension procedure outlined in Section 3.4.1 indicates that there is a connection between the coefficients of particular rank-1 lattice rules and the Van der Corput sequences. In general, however, the point sets of the lattice rules are defined by the node set of different integration lattices, and the lattices are specified by the so-called generating vectors as formulated by (37). The Korbov rules is a popular family of rank-1 lattice rules, and these are known to be particularly effective for periodic integrands.

Naturally, there exist other low-discrepancy sequences than the selection presented in this paper. For instance, the so-called generalized Faure sequences and the Niederreiter sequences are being put to practical use. Further on, the treatment of the lattice rules as given in Section 3.4 is rather superficial.

From the results of the numerical test given in Section 5, we found that the convergence of the approximations when using low-discrepancy sequences was superior to that of the pseudo-random sequences. The effect of a reduction in the effective dimensions was also verified to some extent. Of course, no general conclusions can be drawn from the test results given here. Still, the use of QMC seem to be profitable for these particular situations. The computational effort needed to generate points from low-discrepancy sequences is usually no more than that needed to generate pseudo-random numbers, and thus the use of QMC may be advisable due to the improved rate of convergence. Unfortunately, the QMC methods provide in general no "black-box" solution to the problem of high-dimensional integration, and some considerations should always be made when they are to be applied.

References

- Anderson, E. C. “Monte Carlo Methods and Importance Sampling.” Published on http://ib.berkeley.edu/labs/slatkin/eriq/classes/guest_lect/mc_lecture_notes.pdf (1999). Lecture Notes for Stat 578C Statistical Genetics, Berkeley University of California.
- Arndt, J. “Algorithms for Programmers - ideas and source code.” (2005).
- Bratley, P. and Fox, B. L. “ALGORITHM 659: implementing Sobol’s quasirandom sequence generator.” *ACM Trans. Math. Softw.*, 14(1):88–100 (1988).
- Bratley, P., Fox, B. L., and Niederreiter, H. “Implementation and tests of low-discrepancy sequences.” *ACM Trans. Model. Comput. Simul.*, 2(3):195–213 (1992).
- Caffisch, R. E., Morokoff, W. J., and Owen, A. B. “Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension.” *J. Comp. Finance*, 1(1):27–46 (1997).
- Casella, G. and Berger, R. L. *Statistical Inference*. Duxbury, second edition (2002).
- Chi, H., Beerli, P., Evans, D. W., and Mascagni, M. “On the Scrambled Sobol sequences.” *Lecture Notes in Computer Science*, 3615:775–784 (2005).
- Fox, B. L. “Algorithm 647: Implementation and Relative Efficiency of Quasirandom Sequence Generators.” *ACM Trans. Math. Softw.*, 12(4):362–376 (1986).
- Glasserman, P. *Monte Carlo Methods in Financial Engineering*. Springer (2003).
- Hickernell, F. J. “A generalized discrepancy and quadrature error bound.” *Math. Comput.*, 67(221):299–322 (1998).
- Hickernell, F. J., Hong, H. S., L’Écuyer, P., and Lemieux, C. “Extensible Lattice Sequences for Quasi-Monte Carlo Quadrature.” *SIAM Journal on Scientific Computing*, 22(3):1117–1138 (2001).
- L’Écuyer, P. “Quasi-Monte Carlo methods for simulation.” Proceedings of the 2003 Winter Simulation Conference (2003). Available on <http://www.wintersim.org/prog03.htm>.
- L’Écuyer, P. and Lemieux, C. “Variance Reduction via Lattice Rules.” *Manage. Sci.*, 46(9):1214–1235 (2000).
- Metropolis, N. C. “The Beginning of the Monte Carlo Method.” *Los Alamos Science Special Issue*, 15 (1987).
- Niederreiter, H. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics (1992).
- Okten, G. *Contributions to the Theory of Monte Carlo and Quasi-Monte Carlo Methods*. Dissertation.com (1999). Phd dissertation.
- Owen, A. B. “Latin supercube sampling for very high-dimensional simulations.” *ACM Trans. Model. Comput. Simul.*, 8(1):71–102 (1998a).

- . “Monte Carlo extension of quasi-Monte Carlo.” In *WSC '98: Proceedings of the 30th conference on Winter simulation*, 571–578. Los Alamitos, CA, USA: IEEE Computer Society Press (1998b).
- . “Chapter 1: Quasi-Monte Carlo Sampling.” Lecture notes (2003). Available on <http://citeseer.ist.psu.edu/604470.html>.
- Paskov, S. H. and Traub, J. F. “Faster valuation of financial derivatives.” *J. Portfolio Management*, 22(1):113–120 (1995).
- R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria (2005). ISBN 3-900051-07-0. URL <http://www.R-project.org>
- Sobol', I. M. “Uniformly distributed sequences with an additional uniform property.” *USSR Comput. Math. Math. Phys.*, 16(5):236–242 (1976). Russian.
- Wang, X. and Fang, K.-T. “The effective dimension and quasi-Monte Carlo integration.” *J. Complex.*, 19(2):101–124 (2003).
- Wang, X. and Sloan, I. H. “Why Are High-Dimensional Finance Problems Often of Low Effective Dimension?” *SIAM Journal on Scientific Computing*, 27(1):159–183 (2005).
- Würtz, D. *Rmetrics: An Environment for Teaching Financial Engineering and Computational Finance with R*. Rmetrics, ITP, ETH Zürich, Zürich, Switzerland (2004). URL <http://www.rmetrics.org>