Quasi-Monte Carlo: An Empirical Study on Low-Discrepancy Sequences

Lukas Finschi

Eidgenössische Technische Hochschule Zürich

Institut für Operations Research
Clausiusstrasse 45
CH-8092 Zürich

September 23, 1996
Revised December 19, 1996

Abstract

For numerical integration and many other purposes, Monte Carlo methods have been used for a long time. Newer developments replace the pseudorandom sequences of Monte Carlo methods by deterministic sequences (*quasirandom* or *low-discrepancy sequences*), and the methods are then called *quasi-Monte Carlo methods*. In this paper we outline several theoretical aspects of such deterministic sequences, show two examples of low-discrepancy sequences, and present the results of an empirical study.

Contents

1 Monte Carlo and Quasi-Monte Carlo

In this section we consider (as an example) the problem of numerical integration in dimension d in order to introduce the concept of quasi-Monte Carlo and to present some of its most important properties.

For a more detailed introduction we refer to Niederreiter [7], where the mathematical theory of quasi-Monte Carlo is treated and where an extensive bibliography can be found.

First we recall how the integral of some given function f (with the d-dimensional unit cube I^d as integration domain) is approximated when we use a *classical integration rule* (e.g. trapezoidal rule):

$$\int_{I^d} f(x)dx \approx \sum_{k_1=0}^{n} \cdots \sum_{k_d=0}^{n} w_{k_1} \cdots w_{k_d} \cdot f\left(\frac{k_1}{n}, \ldots, \frac{k_d}{n}\right),$$

where w_{k_i} denote the weights of the integration rule. If $\frac{\partial^2 f}{\partial u_i^2}$ is continuous on I^d for $i = 1, \ldots, d$, then the error of the estimate is $O\left(\frac{1}{N^{2/d}}\right)$. So, for a prescribed level of accuracy, the required number of nodes increases exponentially with d (this property is also called *the curse of dimensionality*).

One possible way to overcome the curse of dimensionality is the use of *Monte Carlo* methods: Here we take N independent uniformly distributed random samples $x_1, \ldots, x_N$ in I^d and approximate our integral by

$$\int_{I^d} f(x)dx \approx \frac{1}{N} \sum_{k=1}^{N} f(x_k). \tag{1}$$

As the nature of Monte Carlo methods is stochastic we can only give probabilistic error bounds; since this bounds are $O\left(\frac{1}{N^{1/2}}\right)$, we might prefer Monte Carlo to classical integration rules for $d \geq 5$. Remark that this convergence rate does not reflect any regularity of f stronger than $f \in L^2$. When we use a Monte Carlo method we have to consider the problem of generating random numbers; in practice there is hardly another way than to use pseudorandom numbers which generate a deterministic sequence with certain stochastic properties.

Finally we arrive at *quasi-Monte Carlo* methods: The idea is to use deterministic points $x_1, \ldots, x_N$ instead of random nodes in the Monte Carlo integration rule (1). In fact, it is possible to construct point sets $x_1, \ldots, x_N$ (socalled *low-discrepancy point sets*) for every N such that we have a deterministic error bound for the estimate computed as in (1) which is $O\left(\frac{(\log N)^{d-1}}{N}\right)$, a significant improvement of the Monte Carlo convergence rate. Some of the quasi-Monte Carlo methods (the socalled *lattice rules* or *methods of good lattice points*) even reflect a higher degree of regularity of the integrand f in the error bound.

The following table shows a direct comparison of some properties of Monte Carlo and quasi-Monte Carlo:

	Monte Carlo	quasi-Monte Carlo
Character of sequence	random (?)	deterministic
Convergence rate for numerical integration in dimension d	$O\left(\frac{1}{N^{1/2}}\right)$	$O\left(\frac{(\log N)^{d-1}}{N}\right)$
Reflection of higher regularity of the integrand in the error bound	no	possible

While Monte Carlo methods were first used and described in the late 1940s and have been used since in a wide field of applications, the development of quasi-Monte Carlo was slower: Since the mathematical origin is the theory of numbers (e.g. diophantine approximation), the roots of quasi-Monte Carlo go back quite far. First applications of the theory of evenly distributed points in I^d were made by Koksma in the early 1940s, the first use of the term *quasi-Monte Carlo*

in an article seems to be from Richtmyer (1951), important theoretical developments followed especially by Hlawka in the 1960s and Niederreiter in the 1970s. Since this time the theoretical research has become more and more intensive and still holds on. Similarly, the constructions of low-discrepancy point sets and sequences have a long tradition and are still a research topic of great interest: Important sequences and point sets were introduced by van der Corput (1935), Halton (1960), Hammersley (1960), Sobol (1967) and Faure (1981/82).

2 Discrepancy and Low-Discrepancy Sequences

In the comparison of Monte Carlo and quasi-Monte Carlo in section 1 we have promised the existence of deterministic point sets $x_1, \ldots, x_N$ for numerical integration, such that the error of the approximation as in (1) is $O\left(\frac{(\log N)^{d-1}}{N}\right)$. In this section now we give a short introduction in the concepts of discrepancy and variation as used to establish the deterministic error bound for (1), then we discuss some general aspects of low-discrepancy point sets and point sequences before we present concrete constructions of low-discrepancy sequences.

2.1 Discrepancy, Variation, and the Koksma-Hlawka Error Bound

One of the very important concepts in the theory of quasi-Monte Carlo is the one of the discrepancy of a point set $x_1, \ldots, x_N$. The discrepancy measures how evenly a given set of points is distributed in a given area, let us say in the unit cube I^d in dimension d. For a general definition of discrepancy see [7], we concentrate on the two most common definitions of discrepancy:

Definition 1 (Discrepancy D_N and D_N^*)

$$D_N(x_1, \ldots, x_N) := \sup_{\substack{0 \le u_i < v_i \le 1 \\ i=1,\ldots,d}} \left| \frac{\left|\{x_1, \ldots, x_N\} \cap \prod_{i=1}^d [u_i, v_i)\right|}{N} - \prod_{i=1}^d (v_i - u_i) \right|,$$

i.e.: For every subinterval A of I^d of the form $[u_1, v_1) \times \cdots \times [u_d, v_d)$ we divide the number of points x_k in A by N and take the absolute difference of this quotient to the volume of A; the maximum difference is the discrepancy D_N.

$$D_N^*(x_1, \ldots, x_N) := \sup_{\substack{0 < v_i \le 1 \\ i=1,\ldots,d}} \left| \frac{\left|\{x_1, \ldots, x_N\} \cap \prod_{i=1}^d [0, v_i)\right|}{N} - \prod_{i=1}^d v_i \right|.$$

Since for every point set $P = \{x_1, \ldots, x_N\}$ we have

$$D_N^*(P) \le D_N(P) \le 2^d \cdot D_N^*(P),$$

the two measures D_N and D_N^* are equivalent in this sense.

For the relation between discrepancy and point sets (or sequences) see subsection 2.2.

For the following definitions let be f a function defined on I^d, $P(I^d)$ the set of all partitions of I^d into subintervals, $p \in P(I^d)$ such a partition, and $A \in p$ a subinterval of I^d. Furthermore let $\triangle(f, A)$ denote an alternating sum of the values of f at the vertices of A (i.e. if x_i and x_j are any two adjacent vertices: when $f(x_i)$ is added, then $f(x_j)$ is subtracted).

Definition 2 (Variation of f on I^d in the sense of Vitali)

$$V^{(d)}(f) := \sup_{p \in P(I^d)} \sum_{A \in p} |\triangle(f, A)|.$$

If $\frac{\partial^d f}{\partial x_1 \ldots \partial x_d}$ is continuous on I^d (not only in the inner of I^d !), then we have the formula

$$V^{(d)}(f) = \int_{I^d} \left| \frac{\partial^d f}{\partial x_1 \ldots \partial x_d} \right| dx.$$

Definition 3 (Variation of f on I^d in the sense of Hardy and Krause)

$$V(f) := \sum_{r=1}^{d} \sum_{1 \le i_1 < i_2 < \cdots < i_r \le d} V^{(r)}(f; i_1, \ldots, i_r),$$

where $V^{(r)}(f; i_1, \ldots, i_r)$ is the variation in the sense of Vitali of the restriction of f to the r-dimensional face $\{(x_1, \ldots, x_d) \in I^d \mid x_k = 1 \text{ if } k \notin \{i_1, \ldots, i_r\}\}$.

Now we are able to formulate a deterministic error bound for numerical integration:

Theorem 1 (Koksma-Hlawka inequality) *If the variation of f on I^d in the sense of Hardy and Krause is finite, i.e. $V(f) < \infty$, then the following inequality holds for any $x_1, \ldots, x_N \in I^d$:*

$$\left| \frac{1}{N} \sum_{k=1}^{N} f(x_k) - \int_{I^d} f(x) dx \right| \le V(f) \cdot D_N^*(x_1, \ldots, x_n).$$

In order to have small errors in the approximation of (1), we want to find point sets and sequences with small discrepancy; this is the topic of the following two subsections.

2.2 Low-Discrepancy Point Sets and Sequences

As we have seen so far, we would like to have point sets $P = \{x_1, \ldots, x_N\}$ with discrepancy $D_N(P)$ as small as possible, or, for practical reasons (i.e. when we do not know at the beginning of a computation what N is sufficient for our purpose), we need point sequences $x_1, x_2, \ldots$ with small discrepancy $D_N(x_1, \ldots, x_N)$ for every N which is large enough.

In the case of dimension $d = 1$ the situation is well understood:

- For every N there exists the optimal point set $P = \left\{\frac{2k-1}{N} | k = 1, \ldots, N\right\}$ with discrepancy $D_N(P) = \frac{1}{N}$ and $D_N^*(P) = \frac{1}{2N}$.
- There exists an absolute constant $c > 0$, such that for any sequence $x_1, x_2, \ldots$ in $I^1 \equiv [0, 1]$ we have $D_N(x_1, \ldots, x_N) \ge c \cdot \frac{\ln N}{N}$ for infinitely many N. In fact there exist several sequences with $D_N(x_1, \ldots, x_N) = O\left(\frac{\log N}{N}\right)$ (see subsection 2.3).
- In practice one should prefer classical integration rules to quasi-Monte Carlo when dimension $d = 1$.

For dimension $d \ge 2$ we can say the following:

- There exist constructions (e.g. Hammersley point sets) which give for any N a point set $P = \{x_1, \ldots, x_N\}$ with $D_N(P) = O\left(\frac{(\log N)^{d-1}}{N}\right)$.
- There exist several constructions of point sequences $S = (x_k)_{k=1}^{\infty}$ in I^d (e.g. Halton, Sobol, and Faure sequences) with $D_N(x_1, \ldots, x_N) = O\left(\frac{(\log N)^d}{N}\right)$.
- Except for the case of $d = 2$ we do not know whether there exist constants $B_d > 0$ (depending only on d) such that for every point set $P = \{x_1, \ldots, x_N\}$ in I^d the inequality $D_N(P) \ge B_d \cdot \frac{(\ln N)^{d-1}}{N}$ holds; if such B_d exist (and this is widely believed), then there also exist constants $\bar{B}_d > 0$ such that for any point sequence $S = (x_k)_{k=1}^{\infty}$ in I^d we have $D_N(S) \ge \bar{B}_d \cdot \frac{(\ln N)^d}{N}$ for infinitely many N.
- There exists a special approach to low-discrepancy point sets, namely the socalled *method of good lattice points*: The point set $P = \{x_1, \ldots, x_N\}$ is constructed by $x_k := \left\{\frac{k-1}{N}\vec{g}\right\}$ for well chosen $N \in \mathbb{N}$ and $\vec{g} \in \mathbb{Z}^d$, where $\{z\}$ denotes the fractional part of z. Applied to numerical integration this technique is particularly suited for periodic integrands; the error analysis is based on Fourier series and reflects the regularity of the integrand. In general it is to say that for any $d \ge 2$ and $N \ge 2$ there exists a $\vec{g} \in \mathbb{Z}^d$ with $D_N(P) = O\left(\frac{(\log N)^d}{N}\right)$, but finding $\vec{g}$ is a serious problem in practice. For further information see [7] or [9].

At this point let us fix the meaning of the term *low-discrepancy sequence*:

Definition 4 (Low-discrepancy sequence) *For given dimension d a point sequence $x_1, x_2, \ldots$ in I^d is called a low-discrepancy sequence when $D_N(x_1, \ldots, x_N) = O\left(\frac{(\log N)^d}{N}\right)$.*

For every low-discrepancy sequence $x_1, x_2, \ldots$ in dimension d there exists a constant c_d such that we can write

$$D_N^*(x_1, \ldots, x_N) \leq c_d \frac{(\ln N)^d}{N} + O\left(\frac{(\log N)^{d-1}}{N}\right).$$

For the sequences of Halton, Sobol, and Faure, we have the following table for the constants c_d (basing on [3] and [4]):

Dimension d	2	3	4	5	6	7	13	∞	order of $\log c_d$
c_d for Halton	0.65	0.81	1.25	2.62	6.13	17.3	90580	∞	$O(d \log d)$
c_d for Sobol	1.04	1.00	1.44	1.66	3.20	5.28	647	∞	$O(d \log \log d)$
c_d for Faure	0.39	0.12	0.099	0.024	0.018	0.0041	0.000010	0	$(c_d \to 0)$

This results suggest that the construction of Faure should be superior to the others. Emipirical comparisons of Fox [4] (for Halton and Faure) and of Bratley and Fox [2] (for Sobol and Faure) indicate that Sobol and Faure are of similar accuracy when used for numerical integration, but Halton works worse. Following this, for the rest of this paper we concentrate on the construction and application of Sobol and Faure sequences.

For recent results on the construction of low-discrepancy sequences we refer to [8].

2.3 Construction of Sobol and Faure Sequences

In this section we present the constructions of low-discrepancy sequences of Sobol and Faure. Both constructions are based on algebraic theory; they are easily implemented and fast to be computed in practice.

In order to describe the constructions, let us fix the dimension $d \geq 2$; we will construct points $x_0, x_1, x_2, \ldots$ in the unit cube I^d, and for every $k \in \mathbb{N}$ and $i \in \{1, \ldots, d\}$ let x_k^i denote the i-th coordinate of x_k, i.e. $x_k = (x_k^1, \ldots, x_k^d)$.

The Construction of Sobol [10]

We first need d different irreducible polynomials $p_i \in \mathbb{F}_2[x]$, where $\mathbb{F}_2$ is the finite field with two elements (such polynomials can be found in lists as e.g. in [6]). For convenience we set an a priori upper bound 2^m for the length of the computed part of the sequence (this bound is absolutely not necessary, but it makes the construction easier to be implemented). Now we construct for every $i \in \{1, \ldots, d\}$ the sequence $x_0^i, x_1^i, x_2^i, \ldots, x_k^i, \ldots$ for $k < 2^m$ in the same way:

Let be i fixed and $p_i(x) = x^r + a_1 x^{r-1} + \cdots + a_{r-1}x + 1$ the i-th of the given polynomials, then we choose arbitrary $v_1, \ldots, v_r$ with $1 \leq \frac{v_j}{2^{m-j}} \leq 2^j - 1$, $\frac{v_j}{2^{m-j}}$ an odd integer, e.g. $v_j := 2^{m-j} \cdot (2^j - 1)$. After this we compute for $j = r+1, \ldots, m$ (in this order)

$$v_j := a_1 v_{j-1} \oplus a_2 v_{j-2} \oplus \cdots \oplus a_{r-1} v_{j-r+1} \oplus v_{j-r} \oplus \frac{v_{j-r}}{2^r},$$

where $\oplus$ is the bitwise xor-operation (remark: for every j is $\frac{v_j}{2^{m-j}}$ an odd integer). For example we have $p_j(x) = x^3 + x + 1$ and choose $m = 5$, $v_1 = 16$, $v_2 = 24$, $v_3 = 28$, then

$$\begin{array}{lllllllll} v_j & := & v_{j-2} & \oplus & v_{j-3} & \oplus & \frac{v_{j-3}}{8}, \text{ i.e.} & & \\ v_4 & := & [11000]_2 & \oplus & [10000]_2 & \oplus & [10]_2 = [1010]_2 & = & 10, \\ v_5 & := & [11100]_2 & \oplus & [11000]_2 & \oplus & [11]_2 = [111]_2 & = & 7. \end{array}$$

Having once computed $v_1, \ldots, v_m$, we can compute x_k^i in two ways,

- either directly by $x_k^i := \frac{g_1 v_1 \oplus g_2 v_2 \oplus \cdots \oplus g_m v_m}{2^m}$, where $g_j := b_j \oplus b_{j+1}$ and $b_j \in \{0, 1\}$ the coefficients in the binary representation of $k \equiv \sum_{j \geq 1} b_j 2^{j-1}$,
- or, when x_{k-1}^i is given, by $x_k^i := \frac{(x_{k-1}^i \cdot 2^m) \oplus v_c}{2^m}$, where $c := \min\{j \mid b_j = 1\}$ and $b_j \in \{0, 1\}$ the coefficients in the binary representation of $k \equiv \sum_{j \geq 1} b_j 2^{j-1}$.

So we always have $x_0^i = 0$ for $i = 1, \ldots, d$; continuing our example, we obtain iteratively

$$\begin{aligned} x_1^i &:= \frac{(x_0^i \cdot 2^5) \oplus v_1}{2^5} = \frac{[0]_2 \oplus [10000]_2}{2^5} = \frac{[10000]_2}{2^5} = \frac{2^4}{2^5} = \frac{1}{2}, \\ x_2^i &:= \frac{(x_1^i \cdot 2^5) \oplus v_2}{2^5} = \frac{[10000]_2 \oplus [11000]_2}{2^5} = \frac{[1000]_2}{2^5} = \frac{2^3}{2^5} = \frac{1}{4}, \\ x_3^i &:= \frac{(x_2^i \cdot 2^5) \oplus v_1}{2^5} = \frac{[1000]_2 \oplus [10000]_2}{2^5} = \frac{[11000]_2}{2^5} = \frac{24}{32} = \frac{3}{4}, \end{aligned}$$

and so on; or directly

$$\begin{aligned} x_3^i &:= \frac{0 \cdot v_1 \oplus 1 \cdot v_2}{2^5} = \frac{v_2}{2^5} = \frac{24}{32} = \frac{3}{4}, \text{ or as a further example,} \\ x_{23}^i &:= \frac{v_3 \oplus v_4 \oplus v_5}{2^5} = \frac{[11100]_2 \oplus [1010]_2 \oplus [111]_2}{2^5} = \frac{[10001]_2}{2^5} = \frac{17}{32}. \end{aligned}$$

The time costs for computing a Sobol point x_k are $O(d \cdot (\log k)^2)$, or, with the a priori bound $k < 2^m$ and after the initialization of $v_1, \ldots, v_m$, $O(d \cdot \log k)$.

The Construction of Faure [3]

First let be p the smallest prime number with $p \geq d$. Then it would be convenient (but not necessary) to set an a priori upper bound p^m for the length of the computed part of the sequence and to precompute for all i, j with $0 \leq j \leq i \leq m$ the binary coefficients modulo p, i.e. $c_{i,j} := \binom{i}{j} \bmod p$ (efficiently, one uses the recurrence $c_{i,j} := c_{i-1,j} + c_{i-1,j-1} \bmod p$), $c_{i,j} := 0$ for $i < j$.

For $k < p^m$: With $k \equiv \sum_{j=0}^{\bar{m}-1} b_j p^j$ the p-adic representation of k, $b_j \in \{0, \ldots, p-1\}$ and $\bar{m} \in \{0, \ldots, m\}$, $b_{\bar{m}-1} \neq 0$, we first set

$$x_k^1 := \sum_{j=0}^{\bar{m}-1} \frac{b_j}{p^{j+1}};$$

then for $i = 2, \ldots, d$ (in this order of i):

$$\begin{aligned} \bar{b}_j &:= \sum_{\ell=j}^{\bar{m}-1} c_{\ell,j} b_\ell \bmod p, \quad j \in \{0, \ldots, \bar{m}-1\}, \\ b_j &:= \bar{b}_j, \quad j \in \{0, \ldots, \bar{m}-1\}, \\ x_k^i &:= \sum_{j=0}^{\bar{m}-1} \frac{b_j}{p^{j+1}}. \end{aligned}$$

For example $d = 3$, i.e. $p = 3$; then:

$$x_0 := (0,0,0), \; x_1 := \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right), \; x_2 := \left(\frac{2}{3}, \frac{2}{3}, \frac{2}{3}\right), \; x_3 := \left(\frac{1}{9}, \frac{4}{9}, \frac{7}{9}\right),$$

or, more detailed, for x_{11}: $11 = 1 \cdot 3^2 + 0 \cdot 3^1 + 2 \cdot 3^0 \equiv [1,0,2]_3$, so

$$\begin{aligned} x_{11}^1 &:= \frac{2}{3} + \frac{1}{27} = \frac{19}{27}, \\ x_{11}^2 &:= \frac{2c_{0,0} + c_{2,0} \bmod 3}{3} + \frac{c_{2,1}}{9} + \frac{c_{2,2}}{27} = 0 + \frac{2}{9} + \frac{1}{27} = \frac{7}{27}, \\ x_{11}^3 &:= \frac{2c_{1,0} + c_{2,0} \bmod 3}{3} + \frac{2c_{1,1} + c_{2,1} \bmod 3}{9} + \frac{c_{2,2}}{27} = 0 + \frac{1}{9} + \frac{1}{27} = \frac{4}{27}. \end{aligned}$$

The time costs for computing a Faure point x_k are $O\left(d \cdot (\frac{\log k}{\log p})^2\right)$.

Remark that we compute for Sobol in base $b = 2$, for Faure in base $b = p$ for a prime p; so the discrepancy $D_N(x_1, \ldots, x_N)$ may be especially small for $N = b^j$ for an integer j (e.g. see [2]).

3 Empirical Study

In this section we present experiences and results from an empirical study. After a short look at the practical behaviour of low-discrepancy sequences we compare antithetic Monte Carlo and quasi-Monte Carlo estimations of mean, variance and quantiles of a given portfolio. We conclude with some suggestions and questions.

3.1 Low-Discrepancy Sequences

Since we will consider a five-dimensional problem in subsection 3.2, we have computed point sequences in the five-dimensional unit cube I^5. A short look at a two-dimensional projection of a Monte Carlo and a quasi-Monte Carlo sequence (each of length 1000) demonstrates the fundamental difference between the two methods:

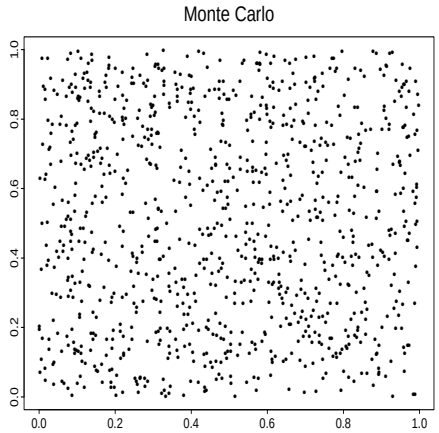

Figure 1: Monte Carlo

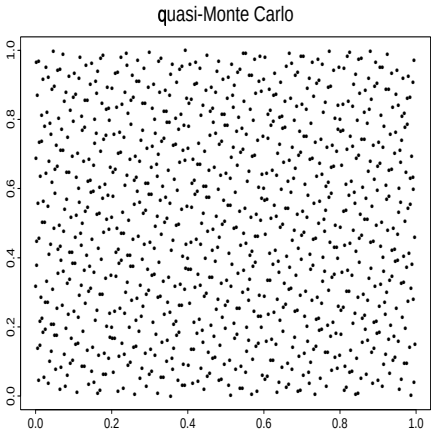

Figure 2: quasi-Monte Carlo

When examinating a low-discrepancy sequence one easily can see that in fact the whole space of the given cube is evenly filled with points, and the speed of the filling is the same in every region of the cube. Even a short extract of one hundred points $x_{k+1}, \ldots, x_{k+100}$ for large k (tested for $k \approx 100000$) is well distributed in the cube; nevertheless we recommend to run every quasi-Monte Carlo simulation with a sequence starting at x_0 (or x_1). Working with Sobol sequences (depending on the choice of the polynomials p_i and the starting values of v_i) one can see that too few points (< 50) may not fill the cube but only some hyperplane; in general it is to say that a low-discrepancy sequence may not be useful before a minimum length of points is generated.

3.2 Approximation of Polynomial Risk Profiles

In this section we consider a five-dimensional risk profile $f(\omega_1, \ldots, \omega_5)$ composed by summation of univariate polynomial risk profiles $f_i(\omega_i)$ of maximum degree three; the risk factors are assumed to be $\mathcal{N}(0, \Sigma)$-distributed with covariance matrix Σ. This portfolio is taken from Allen et al. [1].

For the *Monte Carlo simulation* the points $x_1, \ldots, x_N$ were generated as $\mathcal{N}(0, \Sigma)$-distributed (pseudo)-random numbers in $\mathbb{R}^5$; in fact we used the variance reduction technique of the antithetic variates, i.e. the points $x_1, x_3, x_5, \ldots, x_{N-1}$ (generated as random numbers) were completed with $x_2, x_4, x_6, \ldots, x_N$ by setting $x_{2\ell} := -x_{2\ell-1}$. Without variance reduction technique, the results of a Monte Carlo approximation are very bad; as an example we show a plot of two Monte Carlo approximations of the mean of f, one of them without variance reduction technique, the other with antithetic variates as described above (Figure 3). For an explanation of the lines in the plot, compare with the following plots. The approximations of other values than the mean (like variance, quantiles) show a similar or even stronger superiority of the variance reduced Monte Carlo.

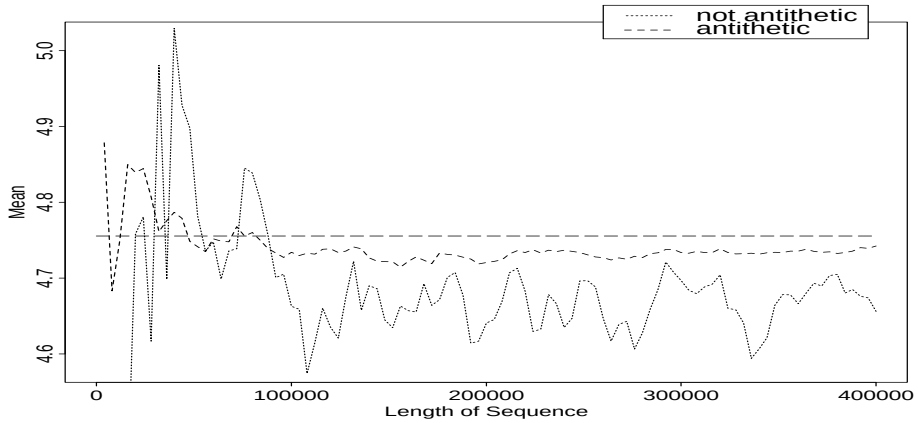

Figure 3: Monte Carlo with and without variance reduction technique

For both *quasi-Monte Carlo* simulations (Sobol and Faure) first a low-discrepancy sequence $\bar{x}_1, \ldots, \bar{x}_N$ in I^5 was generated (with Sobol's or Faure's technique, see subsection 2.3) which then was transformed to a $\mathcal{N}(0, \Sigma)$-distributed sequence $x_1, \ldots, x_N$ by applying the inverse normal function (i.e. $\tilde{x}_k^i := \Phi^{-1}(\bar{x}_k^i)$) and $x_k := V\tilde{x}_k$, where $VV^T \equiv \Sigma$ (for the computation of Φ^{-1} we refer to [5]).

Each of these three sequences (antithetic Monte Carlo, Sobol, Faure) was evaluated pointwise such that we arrived at three sequences of the form $f(x_1), f(x_2), \ldots, f(x_N)$ ($N = 400000$), for which mean, variance and quantiles were determined (as described in the sequel).

In particular we are interested in the question, whether one sequence is more suited than another for the approximation of a value like the mean, the variance or an α-quantile of the underlying risk profile f. We will describe our technique of the following comparisons for the *mean* of f: For each of the sequences $f(x_1), f(x_2), \ldots, f(x_N)$ we computed a sequence of *means* m_j, $j \in \{1, \ldots, N\}$, where

$$m_j := mean(\{f(x_1), \ldots, f(x_j)\}) \equiv \frac{1}{j} \sum_{k=1}^{j} f(x_k).$$

This gives three sequences which approximate the *mean* of f; they correspond to the three lines in the following plot (Figure 4). The horizontal line in the plot is an estimate for the *mean* of f (actually computed as the mean of m_{2N}^{Sobol} and m_{2N}^{Faure}).

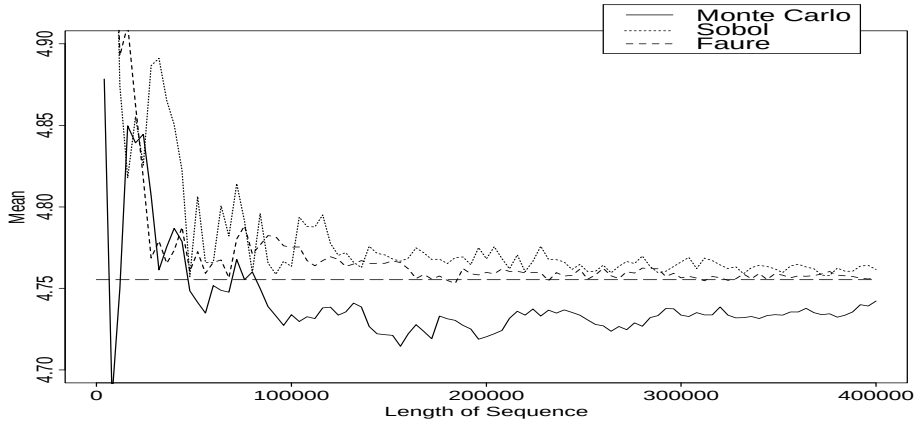

Figure 4: Mean of Monte Carlo, Sobol, and Faure

The approximation of the two quasi-Monte Carlo sequences is significantly better than the approximation of the Monte Carlo sequence. Here especially the sequence of Faure converges remarkably fast and stable.

The next plot shows the comparison for the *variance*:

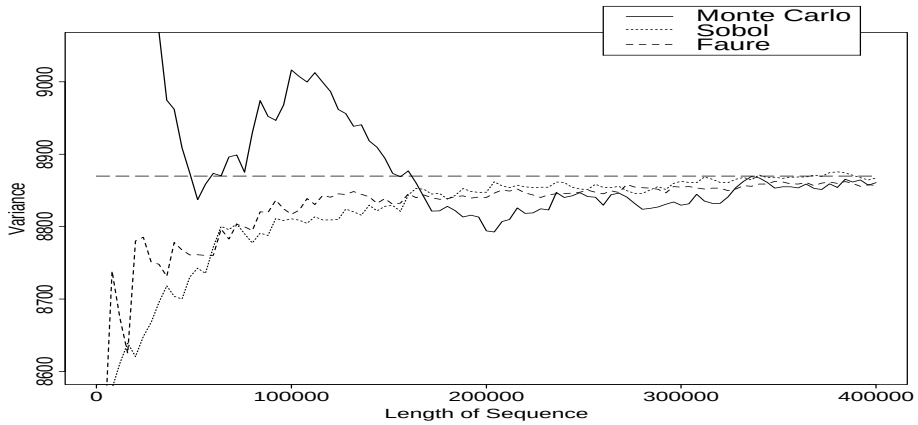

Figure 5: Variance of Monte Carlo, Sobol, and Faure

As we have seen in the comparison for the mean of f, also for the approximation of the variance the low-discrepancy sequences of Sobol and Faure converge faster and very stable compared to the Monte Carlo sequence.

The comparison of the approximation of α-*quantiles* (studied for $\alpha = 50\%, 10\%, 5\%, 1\%$) do not show as clear advantages of quasi-Monte Carlo as seen in the approximation of mean and variance. We just can say that low-discrepancy sequences are not worse than pseudorandom sequences and even more stable in approximation; this is true for all studied quantiles. Sobol and Faure are in general of similar accuracy. As an example we give the plot for the 5% - quantiles:

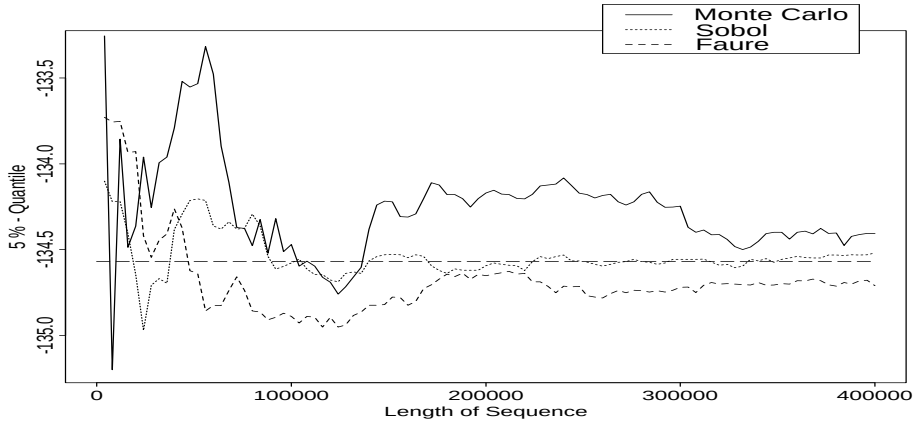

Figure 6: 5% - Quantile of Monte Carlo, Sobol, and Faure

3.3 Conclusions

As (empirical) results of the study that we have presented in this section, we suggest the following:

- Never use Monte Carlo without variance reduction technique.
- Quasi-Monte Carlo is significantly better for estimating the mean of a function than Monte Carlo.
- Quasi-Monte Carlo is more stable approximating the variance of a function than Monte Carlo.

- Quasi-Monte Carlo is never worse than Monte Carlo for the approximation of quantiles.
- Low-discrepancy sequences can be generated easily and fast.

$\Longrightarrow$ For estimating mean, variance, and quantiles, you better use quasi-Monte Carlo than Monte Carlo methods. However, since quasi-random sequences are constructed under the aspect of low discrepancy and not randomness, they can not be used if the relationship among points (e.g. difference between two consecutive points) matters.

We can not answer the following questions, but point them out as possible research topics:

- When the variation in the sense of Hardy and Krause $V(f)$ of a given function f is not finite (as in our study), what length should a low-discrepancy sequence have for attaining certain accuracy?
- What is the progress of quasi-Monte Carlo compared to Monte Carlo for numerical integration in higher dimensions? Quasi-Monte Carlo methods seem to be superior to Monte Carlo methods, but nevertheless the problem of numerical integration in higher dimensions remains hard; for studies on this question see e.g. [2] and [4].

References

[1] M. Allen, K. Frauendorfer, E. Königsperger, H.-J. Lüthi, G. Studer:
Profit & Loss Distribution: An Empirical Study on Polynomial Risk Profiles and their Approximations,
RiskLab-Report, May 1995

[2] Paul Bratley, Bennett L. Fox:
Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator,
ACM Transactions on Mathematical Software, Vol. 14, No. 1, March 1988, pp. 88 - 100

[3] Henri Faure:
Discrépance de suites associées à un système de numération (en dimension s),
Acta Arithmetica XLI, 1982, pp. 337 - 351

[4] Bennett L. Fox:
Algorithm 647: Implementation and Relative Efficiency of Quasirandom Sequence Generators,
ACM Transactions on Mathematical Software, Vol. 12, No. 4, Dezember 1986, pp. 362 - 376

[5] Corwin Joy, Phelim P. Boyle, Ken Seng Tan:
Quasi-Monte Carlo Methods in Numerical Finance,
Preprint, February 1995

[6] Rudolf Lidl, Harald Niederreiter:
Introduction to finite fields and their applications,
Cambrigde University Press, 1986
ISBN 0 521 30706 6

[7] Harald Niederreiter:
Random Number Generation and Quasi-Monte Carlo Methods,
Society for Industrial and Applied Mathematics, Philadelphia, 1992
ISBN 0 89871 295 5

[8] Harald Niederreiter, Chaoping Xing:
Quasirandom Points and Global Function Fields,
Preprint, July 1996

[9] Ian H. Sloan:
Numerical Integration in High Dimensions - The Lattice Rule Approach,
in: T. O. Espelid, A. Genz (eds.): Numerical Integration, Kluwer Academic Publishers, 1992, pp. 55 - 69

[10] I. M. Sobol:
On the Distribution of Points in a Cube and the Approximate Evaluation of Integrals,
U.S.S.R. Computational Mathematics and Mathematical Physics, Vol. 7, No. 4, 1967, pp. 86 - 112