Monte Carlo Methods and Appl., Vol. 2, No. 1, pp. 1–14 (1996)

Quasi-Monte Carlo Methods for Numerical Integration: Comparison of Different Low Discrepancy Sequences

Igor Radović, Ilya M. Sobol′ and Robert F. Tichy *

Abstract — Further numerical experience on quasi-Monte Carlo integration is established. Different kinds of low discrepancy point sequences including Halton, Sobol′, Faure, and Niederreiter sequences are considered and a new family of test functions is used.

Keywords — Monte Carlo method, Quasi-Monte Carlo method. Uniformly distributed sequences, Low discrepancy sequences, Numerical integration

1. Introduction

Simply speaking, quasi-Monte Carlo methods are deterministic versions of Monte Carlo methods. According to [1], any crude Monte Carlo algorithm with constructive dimension s designed for estimating of an expectation $a = M\theta$, can be interpreted as s-dimensional integration:

$$\frac{1}{N}\sum_{n=1}^{N} f(\xi_n) \xrightarrow{P} \int_{I^s} f(x)dx \tag{1}$$

where $\xi_1, \ldots, \xi_N$ are independent random points uniformly distributed in I^s, and $f(x)$ an absolutely integrable modelling function: $\theta = f(\xi)$, $a = Mf(\xi) = \int_{I^s} f(x)dx$.

*The first and the third authors were supported by the Austrian Science Foundation (Project P — 10223 — PHY). The second author was supported by the scientific cooperation program of the Austrian and Russian Academies of Sciences and an ISF grant N5Z000.
Authors+ addresses: I.Radović and R.F.Tichy, Institut für Mathematik, Technische Universität Graz, Steyrergasse 30, 8010 Graz, Austria. Fax: (+43) 316 827685 e-mail: radovic@ftug.tu-graz.ac.at, tichy@ftug.dnet.tu-graz.ac.at; I.M.Sobol′, Institute for Mathematical Modelling of the Russian Academy of Sciences, 4 Miusskaya Square, Moscow 125047, Russia. e-mail: sobol#3@imamod.msk.su

The corresponding quasi-Monte Carlo algorithm uses a non-random uniformly distributed in I^s sequence of points $x_1, \ldots, x_n, \ldots$. From a famous theorem by H. Weyl it follows that if $f(x)$ is bounded and Riemann-integrable in I^s then

$$\lim_{N-\infty} \frac{1}{N} \sum_{n=1}^{N} f(x_n) = \int_{I^s} f(x)dx \tag{2}$$

The main advantage of the quasi-Monte Carlo approach is a possible increase of the convergence rate: the stochastic convergence rate in (1) is $N^{-1/2}$ (assuming the variance $Df(\xi)$ is finite) while the approximation errors of (2)

$$\delta_N(f) = \frac{1}{N} \sum_{n=1}^{N} f(x_n) - \int_{I^s} f(x)dx \tag{3}$$

in some cases behave as $N^{-\alpha}$ with $0.5 < \alpha \leq 1$.

Theoreticans often mention as an advantage of quasi-Monte Carlo methods the possibility of deterministic error bounds. For example, if the total variation (in the sense of Hardy and Krause) $V(f)$ is finite, the Koksma-Hlawka inequality holds:

$$|\delta_N(f)| \leq V(f) D_N^*(x_n) \tag{4}$$

where $D_N^*(x_n)$ is the discrepancy [2] of the sequence of points $x_1, \ldots, x_N$. However, practitioners know that numerically (4) is of little use: in dimensions $s > 2$, the available discrepancy estimates are valid for large classes of sequences but are rather loose for individual sequences or specific values of N. Even more, inequality (4) is loose for every function $f(x)$ encountered in practical problems that is surely not the worst function with bounded variation.

Therefore, numerical experiments involving "good" uniformly distributed sequences and "reasonable" integrands $f(x)$ are of practical interest.

In [3] and [4] test functions $f(x) = \prod_{k=1}^{s} |4x^k - 2|$ were used that had been introduced in [5]. We shall consider a generalization of these integrands:

$$f(x) = \prod_{k=1}^{s} \frac{|4x^k - 2| + a_k}{1 + a_k} \tag{5}$$

If $a_1 = \cdots = a_s$, all the variables $x^1, \ldots, x^s$ are equivalent. But if $0 < a_1 < \cdots < a_s$, the influence of x^k will decrease as k is increased.

We regarded functions (5) as mathematical models illustrating the difference between various Monte Carlo approaches to particle tracking: simulation algorithms and algorithms that include statistical weights. The history of a particle is defined by a sequence of random numbers that play the role of variables $x^1, x^2, \ldots$. As a rule, the weight decreases after each collision, and therefore the influence of subsequent random numbers is diminished. Besides the weight is a product of factors that depend on different random numbers.

For the functions (5), the exact value of the integral $a = 1$. Our aim is to investigate the approximation errors $\delta_N(f)$ as N is increased. To avoid loss of accuracy, the computations are organized in the following way:

$$\eta_0 = 0 \; ; \quad \eta_n = \eta_{n-1} + f(x_n) - 1 \tag{6}$$

for $n = 1, 2, \ldots$. At prescribed values $n = N$ we have $\delta_N(f) = \eta_N / N$.

2. Numerical Experiments

2.1. Sequences

The Halton sequences [6] were the first explicitely defined low-discrepancy sequences: their discrepancies satisfy the inequality

$$D_N^* \leq K_s \frac{(\log N)^s}{N} \tag{7}$$

The individual sequence H is generated with the aid of the first s primes $b_1 < b_2 < \ldots < b_s$. In [7] a class of sequences called LP_τ-sequences was introduced which satisfy the relation (7). Points belonging to such sequences have dyadic rational coordinates and can be computed in a "superfast" way using logical operations rather than arithmetical ones. One of the LP_τ-sequences that satisfies additional uniformity properties [8] is often called the Sobol sequence, S.

The construction of LP_τ-sequences was generalized by H. Faure [9] who considered points with rational coordinates in arbitrary rational bases $b \geq 2$. The sequence F is obtained when the base b is the smallest prime $\geq s$. For F, estimate (7) holds, the parameter τ is the smallest

possible, $\tau = 0$, and $K_s \to 0$ as $s \to \infty$, while for H and S the available estimates of K_s were unbounded as $s \to \infty$.

H. Niederreiter investigated the general case: arbitrary bases $b \geq 2$ and arbitrary $\tau \geq 0$. In [10], (t, s)-sequences in base b are defined. For $b = 2$, this definition is identical with the definition of LP_τ-sequences ($t \equiv \tau$). And for $t = 0$, one obtaines the definiton used by Faure. Thus, F is a $(0, s)$-sequence in base b, $b \geq s$. A vast theoretical investigation of (t, s)-sequences can be found in [11].

In [12] (t, s)-sequences were implemented, called Niederreiter sequences. They included sequences *Ni.opt* with "optimal" bases and $Ni2$ with base 2. For practical purposes the authors of [12] recommended base 2.

In fact, both sequences S and $Ni2$ are LP_τ-sequences and can be generated by one subroutine. Only the tables of direction points (that contain precomputed points with indices $n = 2^m$) are different.

2.2. Programs

For generating sequences H and F we used programs from [3], for S - the fastest programs published in [13] and for $Ni2$ and *Ni.opt*-programs from [12]. We remark that recently a very fast algorithm for implementing H sequences has been constructed by P.Schatte. However, all these programs were rewritten by the first author of the present paper in C++ language. For F we selected the same bases as for *Ni.opt*.

The programs for generating F, $Ni2$ and *Ni.opt* contain an additional parameter L ("skipping length") that allows to omit an initial section of the sequence and to generate points $x_{L+1}, x_{L+2}, \ldots$ instead of $x_1, x_2, \ldots$. The authors of [14] and [15] regard this approach as unreasonable because good parameters L were selected using the exact values of the integrals that had to be computed. Besides, the parameter L strongly depends on the integrand.

In any case, it was somewhat unfair to compare performances of skipped sequences with unskipped ones. Therefore we considered not only sequences H and S but also sequences H^+ and S^+ whose initial point is x_{L+1}, and selected the same parameter $L = 4096$ that was recommended in [12] for $Ni2$.

Note that the generation of the sequences H, S, and $Ni2$ can be carried out simultaneously for all dimensions $1 \leq s \leq \tilde{s}$ because they are

s-dimensional projections of the $\tilde{s}$-dimensional sequence. However, the sequences F and $Ni.opt$ must be generated in each dimension independently.

2.3. Number of points

A popular strategy in Monte Carlo computations is to monitor the convergence of results while the number of trials is doubled. The same strategy can be recommended for quasi-Monte Carlo computations with LP_τ-sequences. Indeed, for initial sections of LP_τ-sequences containing $N = 2^m$ points the discrepancy estimate can be improved:

$$D_N^* = O((\log N)^{s-1}/N) \tag{8}$$

In [16] the same strategy was successfully used in computations with H. For (t, s)-sequences in base $b \geq 2$ the improved discrepancy estimate (8) holds for initial sections that contain $N = b^m$ points. However for skipped sequences with $L = b^p$, inital sections containing $N = b^m$ points obey (8) only at $m \leq p$.

Attempts were made to fix "best" values of N for a given sequence but for obvious reasons they failed: "best" values depend strongly on the integrand $f(x)$. From (6), one can see that the difference $\eta_n - \eta_{n-1}$ is equal to $f(x_n) - a$, and since the points x_n are uniformly distributed in I^s it will vary from $\min f - a$ to $\max f - a$. In [17], the authors numerically demonstrated the irregular behavior of η_n and called it "high frequency oscillations".

Clearly, one should not expect that the least integration errors will be at $N = 2^m$. Nevertheless, the strategy of considering several successive values $N = 2^m$ often provides stable approximations and a possibility to make judgments concerning the convergence. On the contrary, considering three or four arbitrarily selected values of N may lead to false conclusions.

Figure 1, where $x = \log_2 N, y = \log_2 |\eta_N|$, illustrates our assertion: the line that connects 120 points (x, y) reminds of a random process, while 13 points with abscissas $8 \leq m \leq 20$ yield more or less regular approximations.

2.4. Test Functions

We consider four different choices for the parameters $a_1, \ldots, a_s$ in the integrand (5):

(i) $a_1 = \cdots = a_s = 0.01$;
(ii) $a_1 = \cdots = a_s = 1$;
(iii) $a_k = k$;
(iv) $a_k = k^2$. .

All four functions are polynomials of the same degree and they are equally "good" for small values of s. But increasing s, the properties will be rather different. In all cases

$$\sup \ f(x) = A \equiv \prod_{k=1}^{s} \frac{2 + a_k}{1 + a_k}$$

For the function (i) the supremum $A = (2.01/1.01)^s$ increases catastrophically. For the function (ii) the supremum $A = (3/2)^s$ increases not so fast but still exponentially, whereas for (iii) the value $A = 1 + s/2$ is linear, and for (iv) A is bounded.

3. Computational results

We have computed approximate values of the test integrals in dimensions $3 \leq s \leq 50$.

3.1. First, we roughly summarize results obtained at $N \approx 33000$ comparing the obtained values of the integral itself.

For (i), "reasonable" values for H, H^+, and $Ni.opt$ are at $s \leq 6 - 12$, for F - at $s \leq 12 - 15$, for $Ni2$ at $s \leq 15 - 20$, for S at $s \leq 22 - 25$, and for S^+ at $s \leq 25 - 30$.

For (ii), "reasonable" values for $Ni.opt$ are at $s \leq 15 - 20$, for H - at $s \leq 20 - 25$, for H^+ and F at $s \leq 22 - 25$, for S, S^+, and $Ni2$ - at all s.

For (iii) and (iv) - for all sequences at all dimensions the obtained values are "reasonable".

By "reasonable" we mean that its absolute error does not exceed 0.1 . Of course, this paragraph is not very informative but the improvement of results from (i) to (iv) is evident.

3.2. We have studied values $\eta_N = N\delta_N$ and found that in most of cases the skipped sequences H^+ and S^+ yield more accurate results than the original sequences H and S. In fact, for the function (i) we got better results with H^+ for $s \leq 8$; for higher dimensions, H^+ was better at $\log_2 N \leq 15 - 17$ but afterwards there was no clear advantage of H^+ or H. For the function (ii) the situation was the same, just the dimension below which the performance of H^+ was better than that of H was $s = 10$. For the functions (iii) and (iv) H^+ was better than H in all cases.

The comparison of S and S^+ showed that for the worst function (i) the performance of both sequences is roughly speaking equal in all dimensions. For the function (ii) we observed that S^+ was better than S for dimensions $s \leq 25$, but for higher dimensions and $\log_2 N \geq 15 - 17$ the difference was minute. However for the functions (iii) and (iv) S^+ is always better than S.

Table 1 contains typical results: values of $|\eta_N|$ obtained at $s = 25$.

3.3. Tables 2-5 contain values of $|\eta_N|$ obtained for all the four test functions in computations with the dyadic sequences S, S^+, and $Ni2$. If the values of $|\eta_N|$are more or less stable when N increases we conclude that the errors δ_N decrease as N^{-1}. For the functions (iii) and (iv) this is true in all dimensions.

For the "bad" functions (i) and (ii) the situation is more complicated. More or less stable values are the values of $|\eta_N|$ in table 2. In tables 3 and 4, the $|\eta_N|$ are increasing and this can be interpreted as errors δ_N behaving like $N^{-\alpha}$ with some efficient $\alpha < 1$. In the worst case (table 5, $a_k = 0.01$) the efficient α may be even less than $\alpha = 0.5$that characterizes the stochastic convergence rate of the corresponding crude Monte Carlo estimate.

3.4. Table 6 contains similar results for sequences F and $Ni.opt$. Here the base is $b = 17$ and the favourable values $N = b^m$ are necessarily very sparse. Clearly there is no advantage of $Ni.opt$ over F in table 6.

3.5. Figures 2 and 3 demonstrate computational errors (more exactly, values of $|\eta_N|$) obtained in computations with S^+ and $Ni2$. At large N, for "good" functions (iii) and (iv) the $|\eta_N|$ are approximately constant and hence the errors $|\delta_N|$ are $O(N^{-1})$. But for

the "worst function (i) the $|\eta_N|$ are increasing like $N^{1-\alpha}$ with $\alpha \approx 0.7 - 0.5$.

We recall for comparison that probable errors r_N of crude Monte Carlo methods would yield straight lines in Figures 2 and 3 at

$$y = \log_2(Nr_N) = \frac{1}{2}\log_2 N + \log_2\left(0.6745\sqrt{Df(\xi)}\right)$$

corresponding to $\alpha = 0.50$.

3.6. In tables 2-6, one can easily notice extremely small values of $|\eta_N|$ corresponding to isolated values of N. As a rule, they are caused by changes of sign of the error $\delta_N(f)$ and have no relation to the efficient convergence rate.

3.7. The excellent performance of S^+ for the function (iv) is enigmatic. From tables 2-5 one can conclude that the errors $\delta_N = O(N^{-1})$ with a very small constant inside O. Maybe the skipping length $L = 4096$ was a lucky choice?

3.8. We aren't sure that the use of skipped sequences for solving real problems is expedient because there is no rule for selecting L. In example, the deviation of the computed value from the exact answer was minimized, however in real problems the exact answer is unknown. Besides, thus defined L strongly depends on the integrand.
Therefore we remind that in all our experiments the natural sequence S provided for the same efficient order of convergence as the skipped sequences.

3.9 Our results confirm the suggestion of [16] that quasi-random sequences are more promising for sophisticated Monte Carlo algorithms that include statistical weights, rather than for simulation algorithms.

Note in conclusion that our paper can be considered as a complement to [16]. Recently, numerical experiments on quasirandom integrations were carried out by A.B. Owen, F.J. Hickernell, W.J. Morokoff and R.E. Caffish. In [18], global sensitivity indices for the function (5) were computed. The decomposition of (5) into summands of different dimensions can be found there.

s =25	$a_k = 1$				$a_k = k^2$			
$\log_2 N$	H	H+	S	S+	H	H+	S	S+
7	1863	2.4	37.2	4.8	0.95	0.35	0.69	0.037
8	1800	53.5	55.9	25.6	0.86	0.55	1.12	0.055
9	1728	5.6	81.1	16.6	0.91	0.32	0.83	0.015
10	1669	24.8	119	27.5	1.07	0.44	1.19	0.035
11	1561	3.9	162	34.6	1.34	0.52	0.55	0.005
12	1845	41.4	206	6.8	0.95	0.06	1.37	0.007
13	1802	31.1	196	38.7	0.92	0.21	0.9	0.013
14	1760	203	323	86.6	1.15	0.41	1.12	0.021
15	1713	52.6	369	129	0.94	0.1	0.79	0.005
16	1390	566	410	312	1.04	0.22	1.36	0.008
17	627	1071	283	135	1.25	0.4	0.87	0.007
18	381	1385	51.2	342	0.76	0.55	1.31	0.008
19	188	1812	232	409	1.14	0.3	0.82	0.004
20	200	1530	806	457	0.86	0.83	1.19	0.011

Table 1: Comparison of Halton and Sobol sequences for dimension 25

$s = 6$	$a_k = 0.01$			$a_k = 1$			$a_k = k$			$a_k = k^2$		
$\log_2 N$	S	S+	N2	S	S+	N2	S	S+	N2	S	S+	N2
7	23.8	2.5	5.2	6.3	0.6	1.7	2	0.02	0.3	0.5	0.02	0.06
8	23.8	3.4	4.3	5.7	2.5	0.8	2.2	0.5	0.5	0.9	0.07	0.3
9	34.8	9.5	9.6	7.7	0.6	2.4	2.1	0.03	0.5	0.7	0.001	0.2
10	23.8	17.7	0.7	8.4	1.1	0.3	2.9	0.06	0.4	1	0.002	0.3
11	36.6	9.4	2.9	7.2	0.6	1.3	1.6	0.05	0.2	0.3	0.002	0.07
12	44.2	9.5	4.8	10	0.6	1.6	3.2	0.03	0.4	1.1	0.0002	0.03
13	33.5	17	7.3	8.2	1.1	0.08	2.3	0.06	0.01	0.7	0.001	0.04
14	21.6	32.2	6.4	8.5	2.1	0.7	2.9	0.1	0.3	0.9	0.003	0.3
15	19	54.1	10.1	5	3.5	2.2	2	0.2	0.4	0.5	0.007	0.07
16	29.2	25.9	12.8	8.7	1.7	2.2	2.9	0.2	0.9	1.1	0.01	0.4
17	53.7	3.2	19.7	8.8	0.2	0.5	2.1	0.02	0.3	0.6	0.001	0.04
18	53.7	0.4	40.2	10.4	0.03	1.5	3.1	0.04	0.07	1	0.003	0.1
19	58.9	1.4	106	9.7	0.03	5.7	2.3	0.0003	0.4	0.6	0.00001	0.1
20	59.4	27.1	34.2	9.7	1.7	5.2	2.6	0.1	1.1	0.9	0.004	0.04

Table 2: Binary sequences, dimension s=6

$s = 15$	$a_k = 0.01$			$a_k = 1$			$a_k = k$			$a_k = k^2$		
$\log_2 N$	S	S+	N2	S	S+	N2	S	S+	N2	S	S+	N2
7	85.8	102	35.7	32.6	8.1	6.5	4.2	0.09	0.4	0.7	0.04	0.05
8	156	45.2	52.1	40.8	0.7	6.8	5	0.5	0.2	1.1	0.08	0.3
9	165	132	174	38.3	7	21.8	4	0.2	1.1	0.8	0.01	0.3
10	165	67	162	43.7	6	25.6	5	0.4	0.5	1.1	0.03	0.3
11	186	63	59.2	43.5	6.6	2.4	3.7	0.3	0.2	0.5	0.009	0.06
12	234	7.5	265	55.3	16.6	30.5	6.4	0.1	0.5	1.3	0.001	0.009
13	238	78.9	274	36.3	0.4	28.2	5.3	0.07	0.6	0.9	0.00001	0.1
14	407	718	454	76.7	41.8	10.2	6.4	0.4	0.7	1.1	0.003	0.4
15	214	498	203	41.9	16.7	24.9	5.4	0.2	1.5	0.7	0.006	0.04
16	516	365	601	88.2	18.6	1.7	6.6	0.01	0.5	1.3	0.009	0.4
17	2025	1504	1038	135	54	95.7	5.6	0.5	1.9	0.8	0.006	0.04
18	750	621	51.9	121	64.5	48.5	7.1	0.5	0.2	1.3	0.007	0.2
19	4602	4690	901	212	172	40.5	6.4	0.9	0.5	0.8	0.002	0.1
20	3493	2629	2762	195	106	147	7.3	1	1.5	1.1	0.01	0.1

Table 3: Binary sequences, dimension s=15

$s = 25$	$a_k = 0.01$			$a_k = 1$			$a_k = k$			$a_k = k^2$		
$\log_2 N$	S	S+	N2	S	S+	N2	S	S+	N2	S	S+	N2
7	109	70.1	112	37.2	4.8	41.8	4.4	0.05	0.2	0.7	0.04	0.07
8	192	180	2.5	55.9	25.6	0.3	5.5	0.2	0.3	1.1	0.06	0.3
9	120	160	59.4	81.1	16.6	15	5.9	0.5	1.1	0.8	0.02	0.3
10	432	513	162	117	27.5	9.8	7	0.6	0.9	1.2	0.04	0.4
11	607	646	721	162	34.6	15.8	6.4	0.2	0.6	0.6	0.005	0.09
12	1595	440	552	206	6.8	54.1	8.9	0.2	0.2	1.4	0.007	0.01
13	1151	687	1288	196	38.7	159	7.7	0.1	1.9	0.9	0.01	0.1
14	3530	102	221	323	86.6	186	8.6	0.6	2.4	1.1	0.02	0.4
15	2492	417	4256	369	129	605	8.9	0.2	3	0.8	0.005	0.04
16	6206	6077	11838	410	312	827	10.2	0.7	2.1	1.4	0.008	0.4
17	8617	8452	9952	283	135	774	8.8	1.1	4.2	0.9	0.007	0.03
18	4202	1078	13380	51.2	342	905	10.6	1	2.4	1.3	0.008	0.2
19	3120	1725	8402	233	409	856	9.7	1.4	3.4	0.8	0.004	0.1
20	23685	17304	35636	806	457	1247	11.4	2.3	2.9	1.2	0.01	0.1

Table 4: Dyadic sequences, dimension $s = 25$

$s = 50$	$a_k = 0.01$			$a_k = 1$			$a_k = k$			$a_k = k^2$		
$\log_2 N$	S	S+	N2	S	S+	N2	S	S+	N2	S	S+	N2
7	123	120	93.5	35.8	26.2	40	4.7	0.07	0.2	0.7	0.04	0.06
8	184	247	219	11.9	36.2	17.5	5.2	1.2	1.1	1.1	0.04	0.3
9	435	471	400	39.3	20.7	38.4	6.4	0.4	1.6	0.9	0.004	0.3
10	931	786	646	131	9	31.8	7.8	0.6	0.08	1.2	0.04	0.3
11	1879	1530	1481	336	35.9	198	8.3	0.4	0.2	0.6	0.007	0.1
12	3401	3404	3462	551	393	840	12.6	0.04	2.5	1.4	0.01	0.008
13	6805	1997	6868	944	326	1619	11.6	0.2	2.2	0.9	0.02	0.1
14	7159	4891	12426	736	469	10.5	12.4	0.2	2.3	1.2	0.03	0.4
15	8231	5697	1246	253	507	974	13.4	1.2	7.2	0.8	0.001	0.05
16	4785	4157	18966	110	371	5795	14.6	1.8	2.2	1.4	0.03	0.4
17	22112	22449	$1.6 \cdot 10^6$	548	504	16396	13.6	2.4	5.8	0.9	0.01	0.03
18	$2 \cdot 10^5$	$2 \cdot 10^5$	$1.6 \cdot 10^5$	1705	2141	20861	16	2.4	6.3	1.4	0.01	0.2

Table 5: Dyadic sequences, dimension $s = 50$

$s = 15$	$a_k = 0.01$		$a_k = 1$		$a_k = k$		$a_k = k^2$	
N	F	N17	F	N17	F	N17	F	N17
2^7	59.2	51.2	13.5	10.5	0.8	1.3	0.1	0.1
2^8	58.3	31.1	2.1	8.7	0.7	0.08	0.8	0.4
2^9	62.9	85.2	7.3	5.6	0.5	3.4	1.3	1.9
2^{10}	159.4	246	18.1	21.3	6.1	2.7	3.8	3.1
2^{11}	39.2	596	0.9	22.2	4	2.7	2.1	0.9
2^{12}	264	789	5.5	29	5.1	0.01	2.5	0.1
2^{13}	557	785	3.9	9.7	7.3	0.7	3.3	1.1
2^{14}	1273	1554	27.7	39.1	4.4	0.003	2.5	0.3
2^{15}	2519	3705	41.2	51	6.8	14	2.7	7.8
17^1	16.1	16.5	6.6	8.1	0.8	1.1	0.02	0.03
17^2	65	54.1	4.8	0.1	0.2	1.3	0.04	0.5
17^3	386	645	11.2	13.1	0.07	0.07	0.007	0.02

Table 6: Comparison of Faure- and N.opt-sequences for dimension $s = 15$

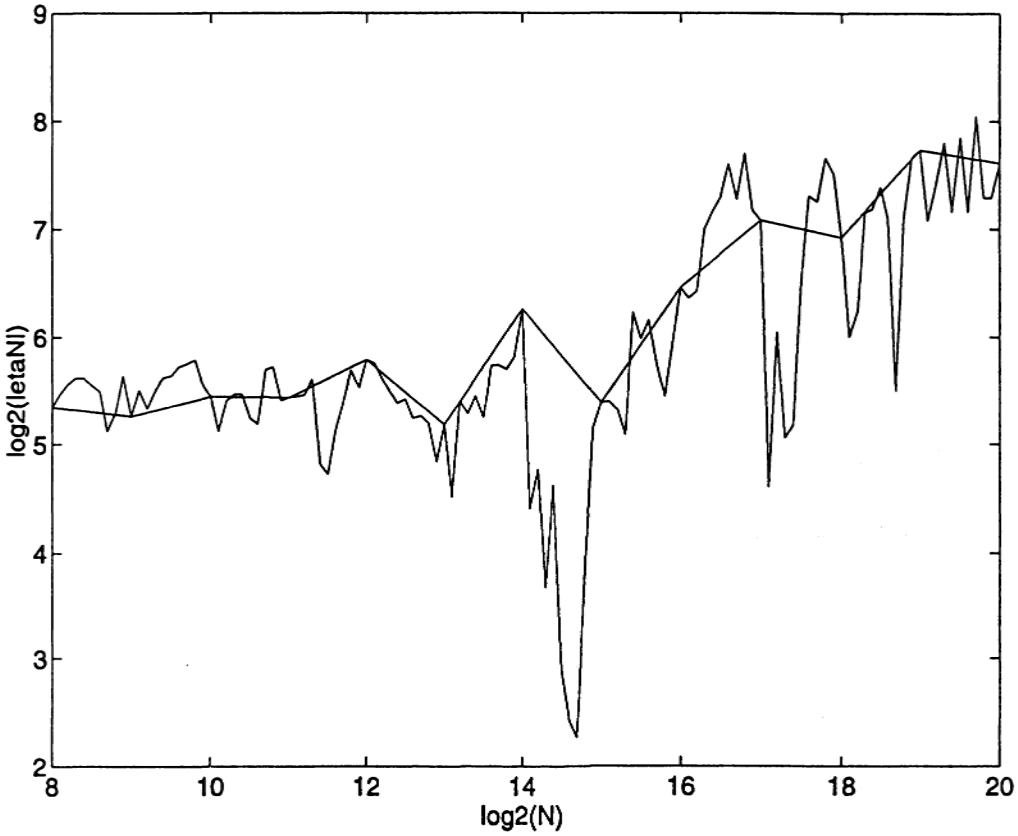

Figure 1. Sequence S, function (ii), dimension $s = 15$

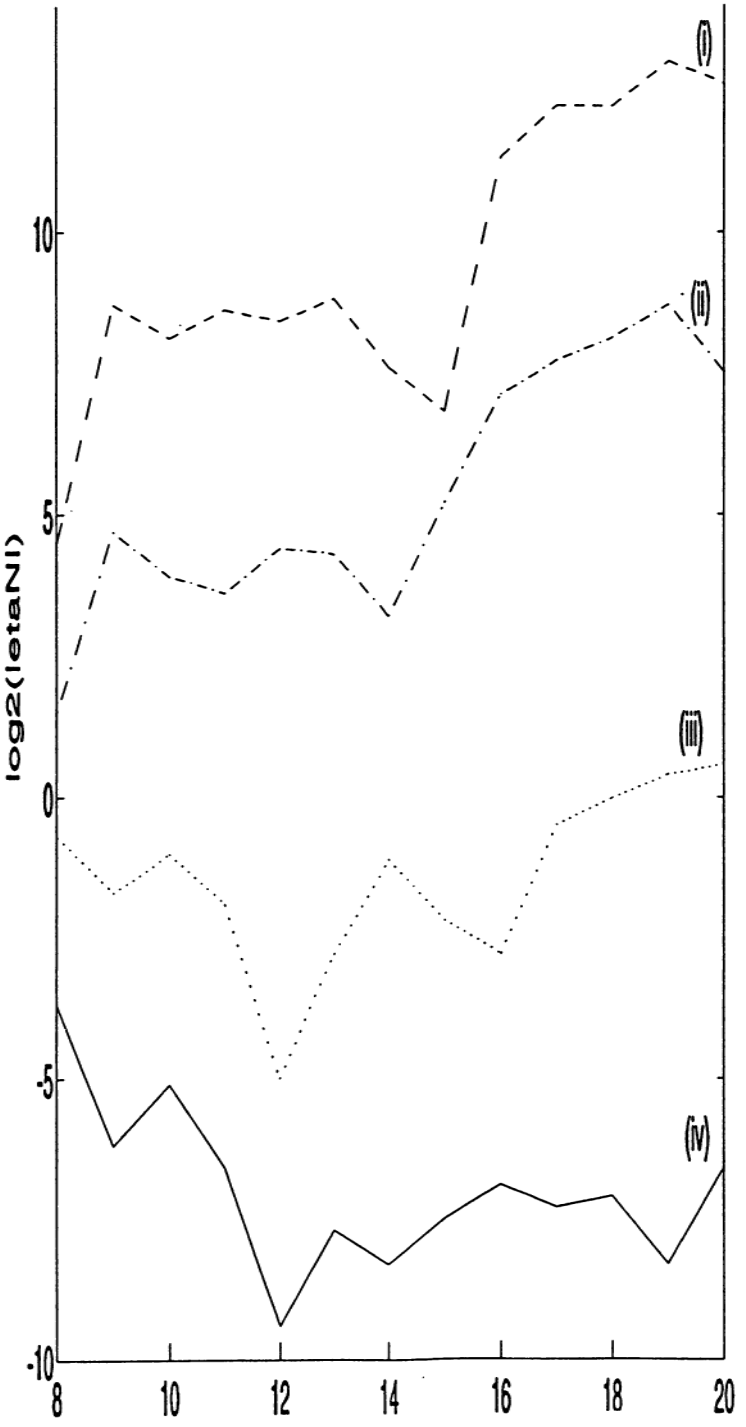

Figure 2. Sequence $S+$, dimension $s = 18$

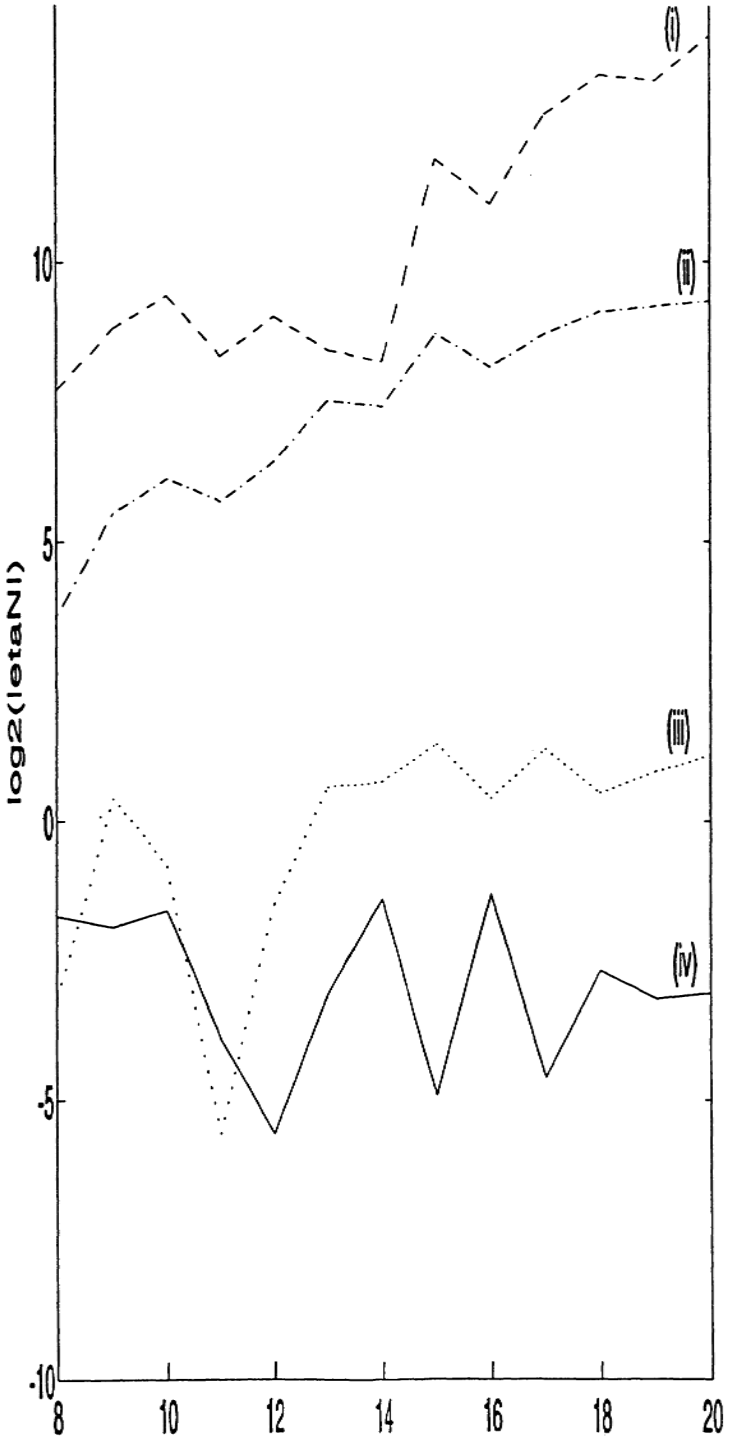

Figure 3. Sequence $Ni2$, dimension $s = 18$

References

[1] I.M.Sobol'. Quasi-Monte Carlo Methods. *Progress in Nuclear Energy* (1990). **24**. 55-61.

[2] L.Kuipers and H.Niederreiter. *Uniform Distribution of sequences*, J.Wiley, New York, 1974.

[3] B.L.Fox. Implementation and relative efficiency of quasirandom sequence generators. *ACM Trans. Math. Software* (1986), **12**,362-376.

[4] P.Bratley and B.L.Fox. Implementing Sobol's quasirandom sequence generator. *ACM Trans.Math.Software* (1988), **14**, 88-100.

[5] P.J.Davis and P.Rabinowitz. *Methods of Numerical Integration.* Academic Press, New York, 1984.

[6] J.H.Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numer.Math.* (1960), **2**, 84-90.

[7] I.M.Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *U.S.S.R. Comput.Math.Math.Phys.* (1967), **7**, 86-112.

[8] I.M.Sobol'. Uniformly distributed sequences with additional uniformity properties. *U.S.S.R. Comput.Math.Math.Phys.* (1976), **16**, 236-242.

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

[10] H.Niederreiter. Point sets and sequences with small discrepancy. *Monatsh. Math.* (1987), **104**, 273-337.

[11] H.Niederreiter. Random Number Generation and Quasi-Monte Carlo Methods. *SIAM*, (1992).

[12] P.Bratley, B.L.Fox and H.Niederreiter. Implementation and tests of low-discrepancy sequences. *ACM Trans.Model.Comp.Simul.* (1993), **2**, 195-213.

[13] I.M.Sobol',V.I.Turchaninov, Yu.L.Levitan and B.V.Shukhman. *Quasirandom sequence generators.* Keldysh Inst. Appl. Math., Rus. Acad. Sci., 1992 (in Russian).

[14] Yu.Levitan, N.I.Markovich, S.G.Rozin and I.M.Sobol'. On quasirandom sequences for numerical computations. *U.S.S.R. Comput. Math. Math. Phys.* (1988), **28**, 88-92.

[15] I.M.Sobol' and B.V.Shukhman. Random and quasirandom sequences: numerical estimates of uniformity of distribution. *J.Math. Computer Modelling.* (1993), **18**, No.8, 39-45.

[16] I.M.Sobol' and B.V.Shukhman. Integration with quasirandom sequences: numerical experience. *Internat. J.Modern Phys.* (1995). Pt.C., **6**, 263-275.

[17] Zakharov V.V. and Smirnova V.I. Experimental comparison of some pseudorandom sequences, In: Problems of Random Search, Zinatne, Riga,5 (1976) 185-190 (in Russian)

[18] A.Saltelli and I.M.Sobol'. Sensitivity analysis for nonlinear mathematical models: numerical experience. *Math. Modelirovanie* (1995), **7**, No.11, 16-28 (in Russian)