Solving Systems of Linear Algebraic Equations Using Quasirandom Numbers[⋆]

Aneta Karaivanova and Rayna Georgieva

CLPP - Bulgarian Academy of Sciences,
Acad. G. Bonchev St., Bl.25A, 1113 Sofia, Bulgaria
anet@copern.bas.bg, rayna@copern.bas.bg

Abstract. In this paper we analyze a quasi-Monte Carlo method for solving systems of linear algebraic equations. It is well known that the convergence of Monte Carlo methods for numerical integration can often be improved by replacing pseudorandom numbers with more uniformly distributed numbers known as quasirandom numbers. Here the convergence of a Monte Carlo method for solving systems of linear algebraic equations is studied when quasirandom sequences are used. An error bound is established and numerical experiments with large sparse matrices are performed using Soboĺ, Halton and Faure sequences. The results indicate that an improvement in both the magnitude of the error and the convergence rate are achieved.

1 Introduction

Monte **C**arlo **m**ethods (MCMs) for solving **s**ystems of **l**inear **a**lgebraic **e**quations (SLAE) have been used for many years [7,8,6,4]. They give statistical estimates for the components of the solution vector by performing random sampling of a certain random variable whose mathematical expectation is the desired solution. However, MCMs require pseudorandom number generators of high quality, high speed and long period and the results of simulation are very sensitive to the generator. Even using "good" generator the convergence rate of a MCM is $O(\frac{1}{\sqrt{N}})$, where N is the number of performed realizations.

On the other hand, the convergence of MCMs for numerical integration can often be improved by replacing pseudorandom numbers (PRNs) with more uniformly distributed numbers known as quasi-random numbers (QRNs) [2]. Quasi-Monte Carlo methods often include standard approaches of variance reduction, although such techniques do not necessarily directly translate. The fundamental feature underlying all quasi-MCMs, however, is the use of a quasi-random sequence. In this paper the convergence of a Monte Carlo method for estimating the solution of SLAE is studied when quasirandom sequences are used. An error bound is established and numerical experiments with large sparse matrices are performed using three different QRN sequences: Soboĺ, Halton and Faure. The

[⋆] Supported by the Ministry of Education and Science of Bulgaria under Grant # MM 902/99 and by Center of Excellence BIS-21 grant ICA1-2000-70016

S. Margenov, J. Wasniewski, and P. Yalamov (Eds.): ICLSSC 2001, LNCS 2179, pp. 166–174, 2001.

results indicate that an improvement in both the magnitude of the error and the convergence rate can be achieved using QRNs in place of PRNs.

2 Background

2.1 Monte Carlo Method for SLAE

Assume that the system of linear algebraic equations (SLAE) is presented in the form:

$$x = Ax + \varphi \tag{1}$$

where A is a real square $n \times n$ matrix, $x = (x_1, x_2, ..., x_n)^t$ is a $1 \times n$ solution vector and $\varphi = (\varphi_1, \varphi_2, ..., \varphi_n)^t$ is a given vector.[1] Assume that all the eigenvalues of A lie in the unit circle. The matrix and vector norms are determined as follows: $\|A\| = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|$, $\|\varphi\| = \max_{1 \le i \le n} |\varphi_i|$.

Now consider the sequence $x^{(1)}, x^{(2)}, \ldots$ defined by the following recursion:

$$x^{(k)} = Ax^{(k-1)} + \varphi, \;\; k = 1, 2, \ldots.$$

Given initial vector $x^{(0)}$, the approximate solution to the system $x = Ax + \varphi$ can be developed via a truncated Neumann series:

$$x^{(k)} = \varphi + A\varphi + A^2\varphi + \ldots + A^{(k-1)}\varphi + A^k x^{(0)}, \;\; k > 0 \tag{2}$$

with a truncation error of $x^{(k)} - x = A^k(x^{(0)} - x)$.

Consider the problem of evaluating the inner product of a given vector g with the vector solution of (1)

$$(g, x) = \textstyle\sum_{\alpha=1}^{n} g_\alpha x_\alpha. \tag{3}$$

To solve this problem via a MCM (see, for example, [12]) one has to construct a random process with mean equal to the solution of the desired problem. First, we construct a random trajectory (Markov chain) T_i of length i starting in state k_0

$$k_0 \to k_1 \to \cdots \to k_j \to \cdots \to k_i,$$

with the following rules

$$P(k_0 = \alpha) = p_\alpha, \;\; P(k_j = \beta | k_{j-1} = \alpha) = p_{\alpha\beta},$$

where p_α is the probability that the chain starts in state α and $p_{\alpha\beta}$ is the transition probability to state β from state α. Probabilities $p_{\alpha\beta}$ define a transition matrix P. The natural requirements are $\sum_{\alpha=1}^{n} p_\alpha = 1$, $\sum_{\beta=1}^{n} p_{\alpha\beta} = 1$ for any $\alpha = 1, 2, ..., n$, the distribution $(p_1, ..., p_n)^t$ is acceptable to vector g and similarly the distribution $p_{\alpha\beta}$ is acceptable to A [12].

[1] If we consider a given system $Lx = b$, then it is possible to choose a non-singular matrix M such that $ML = I - A$ and $Mb = \varphi$, and so $Lx = b$ can be presented as $x = Ax + \varphi$.

It is known [12] that the mathematical expectation $E\Theta^*[g]$ of the random variable $\Theta^*[g]$ is:

$$\begin{aligned} E\Theta^*[g] &= (g,x) \\ \text{where } \Theta^*[g] &= \tfrac{g_{k_0}}{p_{k_0}} \textstyle\sum_{j=0}^{\infty} W_j \varphi_{k_j} \\ \text{and} \quad W_0 = 1\ , \quad W_j &= W_{j-1} \tfrac{a_{k_{j-1}k_j}}{p_{k_{j-1}k_j}} . \end{aligned} \tag{4}$$

We use the following notation for a partial sum (4) $\theta_i[g] = \frac{g_{k_0}}{p_{k_0}} \sum_{j=0}^{i} W_j \varphi_{k_j}$. According to the above conditions on the matrix A, the series $\sum_{j=0}^{\infty} W_j \varphi_{k_j}$ converges for any given vector φ and $E\theta_i[g]$ tends to (g,x) as $i \longrightarrow \infty$. Thus $\theta_i[g]$ can be considered an estimate of (g,x) for i sufficiently large.

To find one component of the solution, for example the r-th component of x, we choose $g = e(r) = (0,...,0,1,0,...,0)$ where the one is in the r-th place. It follows that $(g,x) = \sum_{\alpha=1}^{n} e_\alpha(r) x_\alpha = x_r$ and the corresponding Monte Carlo method is given by

$$x_r \approx \frac{1}{N} \sum_{s=1}^{N} \theta_i[e(r)]_s, \tag{5}$$

where N is the number of chains and $\theta_i[e(r)]_s$ is the value of $\theta_i[e(r)]$ in the s-th chain.

Thus the Monte Carlo estimate for (g,x) is $(g,x) \approx \frac{1}{N} \sum_{s=1}^{N} \theta_i[g]_s$, where N is the number of chains and $\theta_i[g]_s$ is the value of $\theta_i[g]$ taken over the s-th chain, and a statistical error of size $O(Var(\theta_i)^{1/2} N^{-1/2})$.

2.2 Quasirandom Numbers and Integration

Quasi-Monte Carlo methods are based on the idea that random Monte Carlo techniques can often be improved by replacing the underlying source of random numbers with a more uniformly distributed deterministic sequence.

QRNs are constructed to minimize a measure of their deviation from uniformity called discrepancy. Consider a set $\{x_n\}$ of N points in the d-dimensional unit cube I^d. The discrepancy of this set is

$$D_N^* = D_N^*(x_1, \ldots, x_N) = \sup_{E \subset I^d} \left| \frac{\#\{x_n \in E\}}{N} - m(E) \right| , \tag{6}$$

where E is a subrectangle of I^d, $m(E)$ is the volume of E, and the sup is taken over all subrectangles. When the sup is taken only over all subrectangles with one vertex at 0, the discrepancy is called star discrepancy D^*.

The mathematical motivation for QRNs can be found in the classic Monte Carlo application of numerical integration. Let us assume that we are interested in the numerical value of $I = \int_{I^d} f(x)\, dx$, and we seek to optimize approximations of the form $I \approx \frac{1}{N} \sum_{n=1}^{N} f(x_n)$. A solution to the optimization of the integration nodes, $\{x_n\}_{n=1}^{N}$, comes from the famous Koksma-Hlawka inequality:

Theorem (Koksma-Hlawka): if $f(x)$ has bounded variation, $V(f)$, on I^d, and $x_1, \dots, x_N \in I^d$ have star discrepancy D_N^*, then:

$$\left| \frac{1}{N} \sum_{n=1}^{N} f(x_n) - \int_{I^d} f(x)\, dx \right| \leq V(f) D_N^*. \tag{7}$$

This simple bound on the integration error is a product of $V(f)$, the total variation of the integrand in the sense of Hardy and Krause, and D_N^*, the star discrepancy of the integration points. A major area of research in Monte Carlo is variance reduction, which indirectly deals with minimizing $V(f)$. Quasirandom number generation deals with minimization of the other factor.

The star discrepancy of a point set of N truly random numbers in one dimension is $O(N^{-1/2}(\log\log N)^{1/2})$, while the discrepancy of N QRNs can be as low as N^{-1}.[2] In $s > 3$ dimensions it is rigorously known that the discrepancy of a point set with N elements can be no smaller than a constant depending only on s times $N^{-1}(\log N)^{(s-1)/2}$. This remarkable result of Roth, has motivated mathematicians to seek point sets and sequences with discrepancies as close to this lower bound as possible. Since Roth's remarkable results, there have been many constructions of low discrepancy point sets that have achieved star discrepancies as small as $O(N^{-1}(\log N)^{s-1})$. Most notably there are the constructions of Hammersley, Halton, [7], Soboĺ, [13], Faure, [5], and Niederreiter, [7,1].

While QRNs do improve the convergence of applications like numerical integration, it is by no means trivial to enhance the convergence of all MCMs. In fact, even with numerical integration, enhanced convergence is by no means assured in all situations with the naïve use of QRNs. This fact was born out by careful work of Caflisch, Morokoff and Moskowitz(see, for example, [2]). In a nutshell, their results showed that at high dimensions, $s \approx> 40$, quasi-Monte Carlo integration ceases to be an improvement over regular Monte Carlo integration. Perhaps more startling was that they showed that a considerable fraction of the enhanced convergence is lost in quasi-Monte Carlo integration when the integrand is discontinuous. In fact, even in two dimensions one can lose the approximately $O(N^{-1})$ quasi-Monte Carlo convergence for an integrand that is discontinuous on a curve such as a circle. In the best cases the convergence drops to $O(N^{-2/3})$, which is only slightly better than regular Monte Carlo integration.

3 Quasi-Monte Carlo Method for SLAE

We consider the presented Monte Carlo method for solving systems of linear algebraic equations by generating the "random" walks with deterministic, quasirandom sequences. The goal is to generate walks that are in fact not random, but have in some sense better distribution properties in the space in all walks on the

[2] Of course, the N optimal quasirandom points in $[0,1)$ are the obvious: $\frac{1}{(N+1)}, \frac{2}{(N+1)}, \dots \frac{N}{(N+1)}$.

matrix elements. Each walk is constructed according to the following initial and transition densities:

$$p_\alpha = \frac{|g_\alpha|}{\sum_{\alpha=1}^{n} |g_\alpha|}, \quad p_{\alpha\beta} = \frac{|a_{\alpha\beta}|}{\sum_{\beta=1}^{n} |a_{\alpha\beta}|} \qquad \alpha, \beta = 1, \ldots, n.$$

This quasi-Monte Carlo method is faster and has a lower error bound. Let us analyze the error.

We evaluate the inner product (3) of a given vector and the unknown solution vector. Let $x^{(0)} = 0$, then substituting x with $x^{(k)}$ from (2) will give

$$(g, x) \approx (g, x^{(k)}) = g^T\varphi + g^T A\varphi + g^T A^2\varphi + \ldots + g^T A^{(k-1)}\varphi, \;\; k > 0.$$

We can directly consider $g^T A^i \varphi$ as an integral if we define the sets $G = [0, n)$ and $G_i = [i-1, i), \;\; i = 1, \ldots, n$, and likewize define the piecewise continuous functions $g(x) = g_i, \;\; x \in G_i, i = 1, \ldots, n$, $a(x, y) = a_{ij}, x \in G_i, y \in G_j, \;\; i, j = 1, \ldots, n$ and $\varphi(x) = \varphi_i, \;\; x \in G_i, i = 1, \ldots, n$. Then computing $g^T A^i \varphi$ is equivalent to computing an $(i+1)$-dimensional integral and we may analyze using QRNs in this case with bounds from numerical integration. We do not know A^i explicitly, but we do know A and can use quasirandom walks on the elements of the matrix to compute approximately $g^T A^i \varphi$. Using $(i+1)$-dimensional quasirandom sequence to form N walks $[k_0, k_1, \ldots, k_i]_s$, $s = 1, \ldots, N$ we have the following error bound [9]:

$$\left| g^T A^i \varphi - \frac{1}{N} \sum_{s=1}^{N} \left[\frac{g_{k_0}}{p_{k_0}} W_i \varphi_{k_i} \right]_s \right| \leq C_1(A, g, \varphi)\, D_N^*,$$

where W_i is defined in (4), and $[f]_s$ means the value of f on the s-th walk.

Then we have

$$\left| (g, x) - (g, x^{(k)}) \right| \leq C_2(A, g, \varphi)\, k\, D_N^*.$$

Here D_N^* has order $O((log^k N)/N)$. Remember that the order of the mean square error for the analogous Monte Carlo method is $O(N^{-1/2})$.

4 Computational Results

We now present the numerical results for the accuracy of the described MCM and quasi-MCM for computing the scalar product (g, x) for a given vector g (x is the unknown solution vector of the given SLAE), and for computing components of the solution vector. The results are presented as a function of N, the number of walks, and as a function of the length of the walks. For each case the error is computed with respect to the exact solution.

A large number of numerical tests were performed for solving systems of linear algebraic equations with general sparse matrices of size 128, 1024, 2000. The method is realized using PRNs and Soboĺ, Halton and Faure QRNs. The

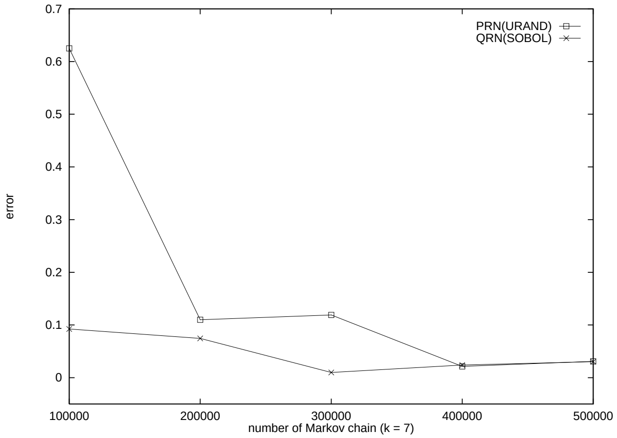

Fig. 1. Accuracy in computing (g, x) with matrix of order 2000.

sequence of pseudorandom numbers is generated with the generator URAND. In the given figures and table, the following notation is used: n - dimension of the tested matrix, k - length of the walk, N — number of the walks. The length of the walks is chosen having in mind the spectral radius of matrices. Because the projections of the n-dimensional quasirandom sequence over $I^i = [0, 1)^i$, $i = 1, \ldots, n-1$, are very well uniformly distributed, we use the same chain for computing all the members in the Neumann series (2). The accuracy when compute a scalar product of a given vector g and the solution of system of size 2000 is presented on Figure 1, where $g_i = 0, i = 1, \ldots, 1000, g_i = 1, i = 1001, \ldots, 2000$. On this figure, the tendency is obvious: the use of QRNs gives better accuracy. For the three tested linear systems of equations we present the results for the 64-th component of the solution whose exact value is 1. The absolute errors for the quasi-MCM and root mean square error for the random MCM with respect to the length of the walks are plotted on Figure 2, and with respect to the number of walks are plotted on Figure 3. The results confirm that using QRNs we obtain much higher accuracy than using PRNs. The magnitude of the error in computing x_{64} is presented on Table 1. Moreover, another important feature of quasi-Monte Carlo methods is the increased smoothness of convergence as the number of samples increases. The best results are obtained using Sobol´ sequence.

5 Conclusion

This paper continues studying the application of quasi-Monte Carlo approach in Linear Algebra problems (see also [9,10]). Our theoretical estimation and numerical experiments for solving SLAE with general sparse matrices confirm that using QRNs we achieve an improvement of the magnitude of error and the convergence rate. The use of Sobol´ sequence gives the best results.

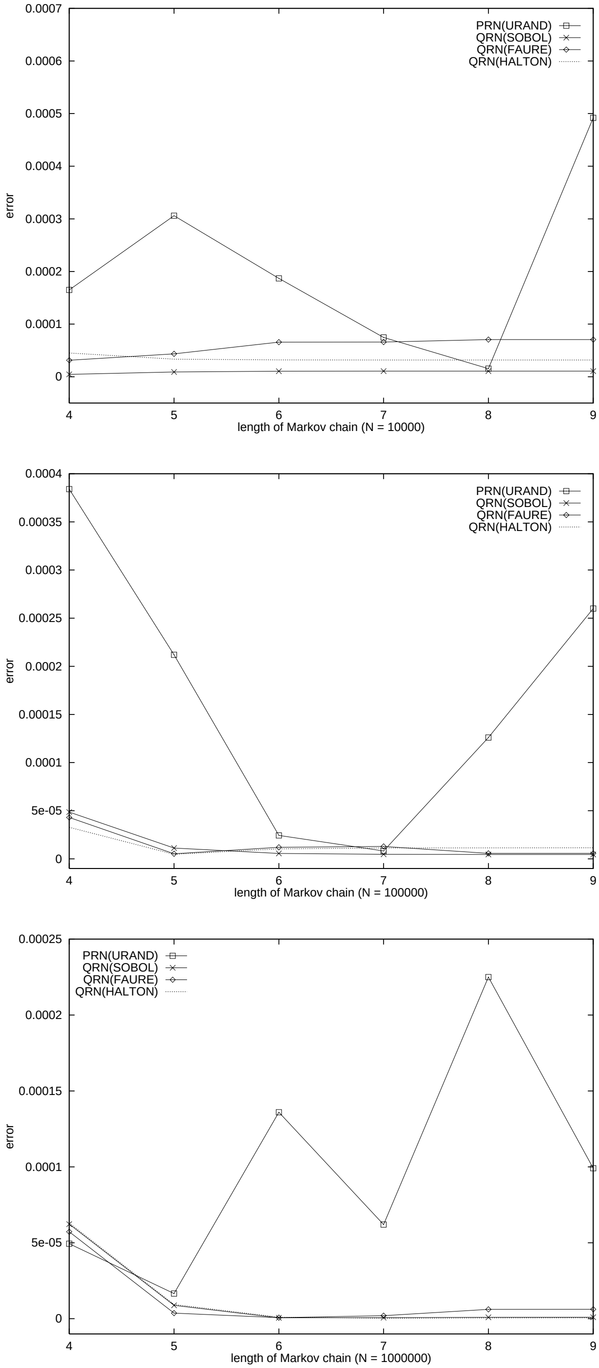

Fig. 2. Accuracy in computing x_{64} with matrix of order 128, 1024 and 2000 with respect to the length of the walks.

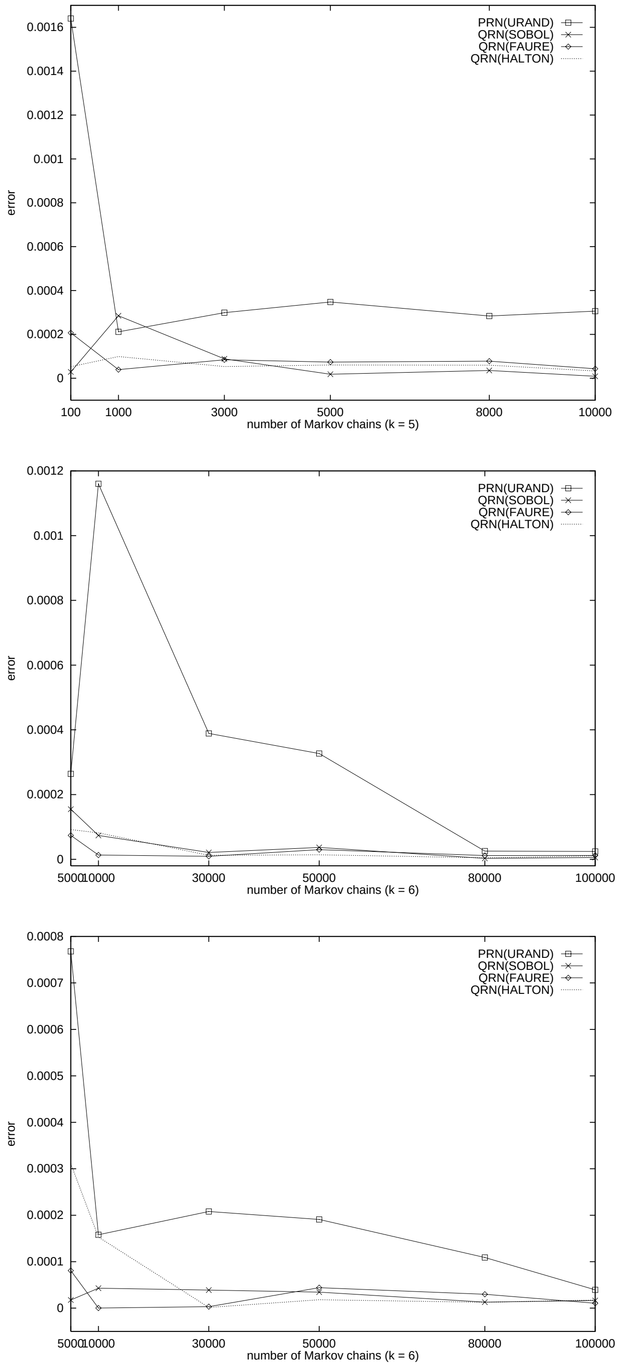

Fig. 3. Accuracy in computing x_{64} with matrix of order 128, 1024 and 2000 with respect to the number of the walks.

Table 1. Accuracy in computing x_{64} for a sparse matrices of order n.

n	N	k	URAND	SOBOL	FAURE	HALTON
128	10000	5	0.306e-03	0.917e-05	0.435e-04	0.334e-04
1024	100000	5	0.212e-03	0.112e-04	0.531e-05	0.471e-05
2000	1000000	6	0.136e-03	0.573e-06	0.707e-06	0.104e-05

References

1. P. Bratley, B. L. Fox, and H. Niederreiter. Implementation and tests of low-discrepancy point sets, *ACM Trans. on Modeling and Comp. Simul.*, 2, 195–213, 1992.
2. R. E. Caflisch. Monte Carlo and quasi-Monte Carlo methods, *Acta Numerica*, 7, 1–49, 1998.
3. I. Dimov and A. Karaivanova. Iterative Monte Carlo algorithms for linear algebra problems, *Lecture Notes in Computer Science*, 1196, Springer, 66–77, 1996.
4. I. Dimov, V. Alexandrov, and A. Karaivanova. Resolvent Monte Carlo Methods for linear algebra problems, in *The second IMACS Seminar on Monte Carlo Methods, Mathematics and Computers in Simulation*, 55(1-3), 25–35, 2001.
5. H. Faure. Discrépance de suites associées à un système de numération (en dimension s), *Acta Arithmetica*, XLI, 337-351, 1992.
6. J. H. Halton. Sequential Monte Carlo techniques for the solution of linear systems, TR 92-033, University of North Carolina at Chapel Hill, Department of Computer Science, 1992.
7. J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, *Numer. Math.*, 2, 84–90, 1960.
8. J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*, Chapman and Hall, London, New York, 1964.
9. M. Mascagni and A. Karaivanova. Are quasirandom numbers good for anything besides integration?, in *Proc. of Advances in Reactor Physics and Mathematics and Computation into the Next Millennium (PHYSOR2000)*, 2000, (to appear).
10. M. Mascagni and A. Karaivanova. Matrix computations using quasirandom sequences, in *Numerical Analysis and its Applications*, *Lecture Notes in Computer Science*, 1988, Springer, 552-559, 2001.
11. H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, 1992.
12. I. M. Soboĺ. *Monte Carlo Numerical Methods*, Nauka, Moscow, 1973, (in Russian).
13. I. M. Soboĺ. The distribution of points in a cube and approximate evaluation of integrals, *Zh. Vychisl. Mat. Mat. Fiz.*, 7, 784–802, 1967, (in Russian).