Journal scientific and applied research, vol. 13, 2018
Association Scientific and Applied Research
International Journal

Original Contribution ISSN 1314-6289

A NUMERICAL STUDY ON QUASI MONTE CARLO METHODS BASED ON RANDOMLY SHIFTED LATTICE RULE FOR COMPUTATION OF MULTIPLE INTEGRALS

Venelin Todorov[1,2], Valentin Dimitrov, Iliyn Tsvetkov

[1] DEPARTMENT OF INFORMATION MODELING, INSTITUTE OF MATHEMATICS AND INFORMATICS, BULGARIAN ACADEMY OF SCIENCES

[2] DEPARTMENT OF PARALLEL ALGORITHMS, INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGIES, BULGARIAN ACADEMY OF SCIENCES

[3] ROUSSE UNIVERSITY "ANGEL KANCHEV"

E-mail: vtodorov@math.bas.bg, venelin@parallel.bas.bg dvd@gmail.com, iliyan_ts@yahoo.com;

ABSTRACT: *A comprehensive numerical study between randomly shifted lattice rules and Fibonacci based lattice rule for computing multidimensional integrals has been done. The methods have not been compared before and both are recommended in case of smooth integrands. The two stochastic methods are completely different thus it is not trivial of which one of them outperforms the other. We consider a case study with smooth integrand functions of different dimensions.*

KEYWORDS: *Quasi-Monte Carlo sequences, multidimensional integrals, randomly shifted lattice rules, Fibonacci lattice rule*

Introduction

Nowadays Monte Carlo (MC) and quasi-Monte Carlo (QMC) methods have become a popular computational device for many problems. Multidimensional integrals are usually solved with MC and QMC algorithms [11]. New approaches have been developed that outperform standard MC algorithm in terms of numerical efficiency [1]. It has been found that there can be efficiency gains in using deterministic sequences rather than the random sequences which are a feature of standard Monte Carlo [5]. The crude Monte Carlo method has rate of convergence $O(N^{-1/2})$ which is independent of

the dimension of the integral, and that is why Monte Carlo integration is the only practical method for many high-dimensional problems [2]. Much of the efforts to improve MC methods are in construction of variance reduction methods which speed up the computation or to use quasi-random sequences [4]. QMC methods use deterministic sequences that have better uniform properties measured by discrepancy [12]. They are usually superior to the MC methods as they have a convergence rate of $((\log N)^s/N)$, where N is the number of samples and s is the dimensionality of the problem under consideration.

Basic Definitions

Let Gs denote the unit cube in s-dimensional space [23]:

$$G_s = [0,1)^s = \{x = (x_1,\ldots,x_s)\} \mid 0 \le x_j < 1, j = 1,\ldots,s\}.$$

Let n1 < n2 < ... be a sequence of positive integers, and let Pnl be any set of nl points in Gs. (Here a set may have multiple copies of the same point.) For any r = (r1,..., rs) note that r1...rs is the volume of the box [0; r). Let Nnl(r) denote the number of points in Pnl lying inside the box [0; r). The discrepancy of the set Pnl is defined as the largest difference between the proportion of points in the box and the volume of the box:

$$D(n_l) := \sup_{r \in G_s} \left| \frac{N_{n_l}(r)}{n_l} - r_1 \ldots r_s \right|$$

This notion was introduced byWeyl (1916). If D(nl) = o(1) as nl goes to infinity, then the sequence of sets Pnl ,
n1 < n2 < ... is said to be uniformly distributed on Gs with discrepancy D(nl). The subscript l is often omitted for simplicity. Not only is the discrepancy a geometric method for measuring uniformity of a set, the discrepancy of a set measures its quality for use in numerical quadrature. The error of this approximation is bounded by the Koksma-Hlawka inequality [14]:

$$\left| \int_{G_s} f(x)dx - \frac{1}{n} \sum_{k=1}^{n} f(x_k) \right| \le D(n).V(f),$$

where D(n) is the discrepancy of the set

$$P(n) = \{x_1, \ldots, x_n\},$$

and V(f) is the bounded variation of f in the sense of Hardy and Krause. If the integrand is smoother and also periodic, then better error bounds may be obtained, in particular for quadrature rules using lattice point sets.
The lattice S is an infinite set of points with the following three properties [16,17]:

1. If x and x' belong to S , then x + x' and x - x' also belongs to S .
2. S contains s linearly independent points.
3. There exists a sphere centered at 0 that contains only 0 itself.
By a ”lattice rule” then, we shall mean a rule of the form

$$I_N(f) = \frac{1}{N}\sum_{j=0}^{N-1} f(x_j),$$

in which $x_0,...,x_{N-1}$ are all the points of a multiple-integration lattice that lie in Gs. The cubic lattice is

$$\left\{\left(\frac{j_1}{n},\ldots,\frac{j_s}{n} : j_i \in \mathbb{Z}, 1 \leqq i \leqq s\right)\right\},$$

where n is a positive integer. The corresponding lattice rule is the ”rectangle rule”

$$I_N(f) = \frac{1}{N}\sum_{j_1=0}^{n-1}\cdots\sum_{j_s=0}^{n-1} f\left(\frac{j_1}{n},\ldots,\frac{j_s}{n}\right),$$

where $N = n^s$. Because N rises very rapidly with s, the rectangle rule suffers in a very obvious way from the ”curse of dimensionality.” Note that this rule is equivalent, because of the assumed periodicity, to a product-trapezoidal rule.

Lattice rules are based on the use of deterministic sequences rather than random sequences. They are a special type of so-called low discrepancy sequences. It is known that as long as the integrand is sufficiently regular, lattice rules outperform the basic Monte Carlo method and most of the other types of low discrepancy sequences [8,12].

Fibonacci based lattice rule for numerical integration

There are constructions of sequences known such that for their discrepancy:

$$D_N^*(x_1,\ldots,x_N) \leq C\frac{(\ln N)^s}{N}.$$

Here C is a certain constant, depending on the sequence. These sequences are believed to have the best possible order of convergence.

The monographs of Sloan and Kachoyan [18], Niederreiter [15], Hua and Wang [9], Wang and Hickernell [23] and Sloan and Joe [17] provide comprehensive expositions of the theory of integration lattices.

Let n be an integer, and $a = (a_1,...,a_s)$ be an integer vector modulo n. A set of the form [19]

$$P_n = \left\{ \left\{ \frac{ak}{n} \right\} = \left(\left\{ \frac{a_1 k}{n} \right\}, \ldots, \left\{ \frac{a_s k}{n} \right\} \right) \mid k = 1, \ldots, n \right\}$$

is called a lattice point set, where {x} denotes the fractional part of x. The vector *a* is called a lattice point or generator of the set. As one can see, the formula for the lattice point set is simple to program. The difficulty lies in finding a good value of a, such that the points in the set are evenly spread over the unit cube. The choice of good
generating vector, which leads to small errors, is not trivial [9,10]. Complicated methods from theory of numbers are widely used, for example Zaremba's index or error of the worst function. We consider the following generating vector based on generalized Fibonacci numbers of corresponding dimensionality [9,23]:

$$a = (1, F_{l+1}^{(s)}, \ldots, F_{l+s-1}^{(s)})), \quad n_l = F_l^{(s)},$$

where

$$F_{l+s}^{(s)} = F_l^{(s)} + F_{l+1}^{(s)} + \ldots + F_{l+s-1}^{(s)}, l = 0, 1, \ldots$$

with initial conditions

$$F_0^{(s)} = F_1^{(s)} = \ldots = F_{s-2}^{(s)} = 0, F_{s-1}^{(s)} = 1,$$

for l=0,1,…
The discrepancy of the set obtained by using the vector described above is asymptotically estimated in [9].
The number of calculation required to obtain the generating vector is O(ln n_l). The generation of a new point requires constant number of operations, thus to obtain a lattice set of the described kind consisting of n_l points, O(ln n_l) number of operations are necessary. However the discrepancies of the lattice point sets obtained by these two methods have larger upper bounds than those obtained by Korobov's method [17,18].

Randomly shifted lattice rules

The method is developed by Dirk
Nuyens in [12,13]. Given a generating vector **z** of integers, the *k*-th point of the sequence is given by

$$\mathbf{x}_k := \varphi(k)\ \mathbf{z} \bmod l, \text{ for } k=0,1,2,\ldots,$$

where φ is typically the radical inverse or the gray coded radical inverse function in the base of the lattice sequence. The radical inverse of an integer k with m digit base b expansion

$$k = (k_{m-1}\ k_{m-2}\ \cdots\ k_0)_b$$

is obtained by mirroring the digits at the fractional point, i.e.,

$$\varphi(k) = (0.k_0\ k_1\ \cdots\ k_{m-1})_b\ .$$

Obviously the result is a rational in b^m and an alternative view is thus to look at this mapping as a permutation of the integers in $\{0,\ldots,b^m-1\}$. In other words we are reversing the digits. Luckily, in base 2 reversing digits can be done efficiently. There is a Matlab/Octave and a C++ version to reverse bits developed by Dirk Nuyens in [13]. To obtain the radical inverse in base 2 of an unsigned 32 bit integer then just requires scaling the result of the bit reversion by 2^{-32}.

Working in gray code ordering has speed advantages for digital nets (and sequences) but less so for lattice sequences. The comparison with the digital sequences will be a future study.

Sobol Sequences

Sobol sequences (also called LP_τ sequences or (t, s) sequences in base 2) are an example of quasi-random low discrepancy sequences. The Sobol quasi-random sequences was first introduced by the Russian mathematician Ilya M. Sobol in 1967 [19] and later described in [20]. We use an adaptation of the INSOBL and GOSOBL routines in ACM TOMS Algorithm 647 [4] and ACM TOMS Algorithm 659 [3]. The original code can only compute the "next" element of the sequence. The revised code allows the user to specify the index of the desired element [21]. These sequences use a base of two to form successively finer uniform partitions of the unit interval and then reorder the coordinates in each dimension. In his article, Sobol described Π_τ-meshes and LP_τ sequences, which are (t,m,s)-nets and (t,s)-sequences in base 2 respectively. The terms (t,m,s)-nets and (t,s)-sequences in base b (also called Niederreiter sequences) were coined in 1988 by Niederreiter [15]. The term Sobol sequences was introduced in late English-speaking papers in comparison with Halton, Faure and other low-discrepancy sequences. A more efficient gray code was proposed by Antonov and Saleev in [16].

Numerical example and results

We will test the performance of the randomly shifted lattice rule (RSLR) based on a special choice of the generating vector obtained with the fast component by component construction developed by Dirk Nuyens [12,13] and a particular lattice rule with generating vector, based on the generalized Fibonacci numbers of the corresponding dimensionality (FIBO) on multidimensional integrals of smooth functions of different dimensions. A comparison with Sobol quasi-random sequence (SOBOL) for a preliminary given computational time will be given. We will be interested which of the methods gives lowest relative errors for 1 minute. We consider examples of 4, 10 and 25 dimensional integrals of smooth integrands. The computational time is given in seconds. We have used CPU Intel i5 2410M and 8GB of ram for running the numerical experiments and the computations have been done with Matlab.

Example 1.

$$\int_{[0,1]^4} x_1 x_2^2 e^{x_1 x_2} \sin(x_3)\cos(x_4) \approx 0.1089748630.$$

Exampe 2.

$$\int_{[0,1]^{10}} \frac{4x_1 x_3^2 e^{2x_1 x_3}}{(1+x_2+x_4)^2} e^{x_5+\cdots+x_{10}} \approx 14.808435.$$

Example 3.

$$\int_{[0,1]^{25}} \frac{4x_1 x_3^2 e^{2x_1 x_3}}{(1+x_2+x_4)^2} e^{x_5+\cdots+x_{20}} x_{21}\ldots x_{25} \approx 103.808.$$

Table 1: The relative error for 4 dimensional integral

N	FIBO	Time	RSLR	Time
100	1.39e-1	0.001	1.31e-1	0.002
1000	9.27e-3	0.01	3.83e-3	0.02
10000	7.90e-4	0.09	5.76e-3	0.22
100000	3.40e-4	1.10	4.29e-4	2.24
1000000	2.68e-5	5.79	1.38e-5	16.5

Table 2: The computational time for 4 dimensional integral

time in seconds	FIBO	SOBOL	RSLR
0.1	9.27e-3	5.17e-3	3.44e-3
1	3.26e-4	5.37e-5	7.21e-4
10	7.21e-6	1.43e-5	9.21e-5
60	9.10e-8	2.68e-7	5.51e-6

Table 3: The relative error for 10 dimensional integral

N	FIBO	Time	RSLR	Time
100	8.35e-1	0.001	5.77e-1	0.002
1000	1.47e-1	0.08	6.72e-2	0.02
10000	4.21e-2	0.12	8.71e-3	0.25
100000	1.02e-2	0.91	9.57e-4	2.37
1000000	1.08e-3	6.27	1.95e-4	14.01

Table 4: The computational time for 10 dimensional integral

time in seconds	FIBO	SOBOL	RSLR
0.1	9.82e-2	1.06e-2	5.94e-2
1	4.58e-2	9.15e-2	8.12e-3
10	1.37e-2	9.93e-4	3.16e-4
60	1.28e-3	1.38e-4	1.65e-5

Table 5: The relative error for 25 dimensional integral

N	FIBO	Time	RSLR	Time
1000	9.84e-1	0.03	2.77e-1	0.04
10000	7.10e-1	0.11	9.70e-2	0.31
100000	1.97e-1	0.81	1.42e-3	2.59
1000000	9.09e-2	6.40	7.04e-4	11.1

Table 6: The computational time for 25 dimensional integral

time in seconds	FIBO	SOBOL	RSLR
0.1	7.10e-1	2.50e-1	4.22e-1
1	1.21e-1	1.09e-1	1.13e-1
10	8.86e-2	1.85e-2	4.43e-3
60	7.13e-2	9.21e-3	1.15e-4

In the Table 1,3 and 5 are presented the relative error for the 4,10 and 25 dimensional integrals with Fibonacci lattice sequence (FIBO) and randomly shifted lattice rule (RSLR) for a fixed number of points. In Table 2,4,6 are presented the relative errors for 4,10 and 25 dimensional integrals with Sobol quasi-random sequence (SOBOL), FIBO and RSLR for a fixed computational time which is a measure of the computational complexity. Obviously FIBO has the lowest computational complexity and is the fastest algorithm, while RSLR and SOBOL are slower, because they need an additional time for generating the corresponding low discrepancy sequences. As can been seen from the results for 4 dimensional integral, the randomly shifted lattice rule produces more rapid convergence, and lower errors, than the Fibonacci lattice sequence for a given number of realizations of the random variable-see Table 1, but for a fixed computational time- Fibonacci sequence gives better results- see Table 2. It can be seen that for lower dimensions FIBO is better than Sobol for a fixed computational time. Therefore Fibonacci lattice rule is the best choice for low dimensional integrals. For the 10-dimensional integral RSLR gives lower relative errors than the Fibonacci algorithm- see Table 3. For a preliminary given time in seconds Sobol and RSLR gives better results than Fibonacci-see Table 4. We can conclude that for mid and high dimensions RSLR sequence gives more reliable results than FIBO and it can be successfully compete with one of the best quasi-random sequences of Sobol. For 25- dimensional integral as expected FIBO produces the worst results, while randomly shifted lattice rule is more appropriate is clearly better - see Table 5. For a fixed computational time RSLR again gives lower relative errors than - see Table 6. For higher dimensions the errors can not be small. However, RSLR gives sufficient accuracy. This multidimensional integral can be applied to various problems [2] where data is taken in randomized way [6]. They are often used in physical problems [7] and are most useful when it is difficult or impossible to use other mathematical methods. This multidimensional integral can be applied to various problems like stochastic tomography connected with people migration. International migration is a topic that is attracting a significant level of interest in current political debate and is high on the agenda for policy makers in central and local government. In our previous paper [22] we described the Hammersley sequence and the comparison with the randomly shifted lattice rule and other types of low discrepancy sequences will be an object of a future study.

Conclusion

In this paper we analyze the performance of different quasi-Monte Carlo methods for multidimensional integrals. Stochastic methods under consideration are an efficient way to solve the problem under consideration. Clearly the progress in this area is closely connected with developing fast and reliable algorithm for multidimensional integrals.

Acknowledgement

This work was supported by the Bulgarian Academy of Sciences through the, Program for Career Development of Young Scientists", Grant DFNP-17-88/28.07.2017, Project „Efficient Numerical Methods with an Improved Rate of Convergence for Applied Computational Problems" and by the Bulgarian National Science Fund under Project DN 12/5-2017, Project "Efficient Stochastic Methods and Algorithms for Large-Scale Problems".

References

[1]. N. Bakhvalov, On the approximate calculation of multiple integrals, Journal of Complexity, Volume 31, Issue 4, 2015, pp 502-516.

[2]. P. Boyle, Y. Lai, K. Tan, (2001) Using Lattice Rules to Value Low-Dimensional Derivative Contracts.

[3]. Paul Bratley, Bennett Fox, Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator, ACM Transactions on Mathematical Software, Volume 14, Number 1, March 1988, pages 88-100.

[4]. Paul Bratley, Bennett Fox, Harald Niederreiter, Implementation and Tests of Low Discrepancy Sequences, ACM Transactions on Modeling and Computer Simulation, Volume 2, Number 3, July 1992, pages 195-213.

[5]. I. Dimov, Monte Carlo Methods for Applied Scientists, New Jersey, London, Singapore, World Scientific, 2008, 291p.

[6]. Dzurov V. APPLICATION OF SUMMARIZED FUNCTIONS FOR INFORMATION SOURCE PROTECTION.// Journal Scientific and Applied Research, 2013, No 3, pp. 51-56, ISSN 1314-6289.

[7]. Dzurov V, M.Kostova. I.Georgiev. A MATHEMATICAL MODEL SYSTEM FOR RADIOLOCATIONAL IMAGE RECONSTRUCTION OF DYNAMIC OBJECT WITH LOW RADIOLOCATIONAL VISIBILITY.// Proceedings of the Union of Scientists – Ruse, "Mathematics, Informatics and Physics", 2012, No 9, pp. 35-40, ISSN 1314-3077.

[8]. Hickernell, F. J., Hong, H. S., L'Ecuyer, P. and Lemieux, C. (2001), Extensible lattice sequences for quasi-Monte Carlo quadrature, SIAM J. Sci. Comput. 22, 1117-1138.

[9]. Hua, L.K., Wang, Y.: Applications of Number Theory to Numerical analysis, 1981

[10]. Ladislav Kocis, William Whiten, Computational Investigations of Low-Discrepancy Sequences, ACM Transactions on Mathematical Software, Volume 23, Number 2, 1997, pages 266-294.

[11]. Kroese, D.P., Taimre, T., Botev, Z.: Handbook of Monte Carlo Methods,Wiley Series in Probability and Statistics, (2011)
[12]. F.Y. Kuo and D. Nuyens, Application of quasi-Monte Carlo methods to elliptic PDEs with random diffusion coefficients - a survey of analysis and implementation, Foundations of Computational Mathematics, 16(6):1631-1696, 2016.
[13]. Dirk Nuyens, The “Magic Point Shop” of QMC point generators and generating vectors,
[14]. https://people.cs.kuleuven.be/~dirk.nuyens/qmc-generators/ A. Owen: Monte Carlo theory, methods and examples, (2013)
[15]. H. Niederreiter, Random Number Generation and quasi-Monte Carlo Methods, SIAM, 1992, ISBN13: 978-0-898712-95-7, LC: QA298.N54.
[16]. V. M. Saleev, I. A. Antonov An Economic Method of Computing LP Tau-Sequences, USSR Computational Mathematics and Mathematical Physics, Volume 19, 1980, pages 252-256.
[17]. I.H. Sloan and S. Joe: Lattice Methods for Multiple Integration, Oxford Uni Press, Oxford, 1994.
[18]. I.H. Sloan and P.J. Kachoyan, Lattice methods for multiple integration: Theory, error analysis and examples, (1987), J. Numer. Anal., 24, 116-128.
[19]. I.M. Sobol (1967), "Distribution of points in a cube and approximate evaluation of integrals". *Zh. Vych. Mat. Mat. Fiz.* **7**: 784–802 (in Russian); *U.S.S.R Comput. Maths. Math. Phys.* **7**: 86–112 (in English).
[20]. Sobol, I. M. (1976) "Uniformly distributed sequences with an additional uniform property". *Zh. Vych. Mat. Mat. Fiz.* **16**: 1332–1337 (in Russian); *U.S.S.R. Comput. Maths. Math. Phys.* **16**: 236–242 (in English).
[21]. I. Sobol’, D. Asotsky, A. Kreinin, S. Kucherenko, "Construction and Comparison of High-Dimensional Sobol’ Generators" , Wilmott Journal. ***Nov****: 64–79,* 2011.
[22]. V. Todorov, I. Dimov, V. Dzurov, T. Stanchev, I. Tsvetkov, V. Dimitrov: A Numerical study on Hammersley sequence and Fibonacci based lattice rule for computation of multidimensional integrals, Journal Scientific and Applied Research, Vol. 12, 18-26, Konstantin Preslavsky Publishing House, ISSN 1314-6289, 2017.
[23]. Y. Wang and F.J. Hickernell, An historical overview of lattice point sets, (2002).