Trends in Mathematics
Information Center for Mathematical Sciences
Volume 6, Number 2, December, 2003, Pages 155–162

QUASI-RANDOM NUMBER GENERATION IN THE VIEW OF PARALLEL IMPLEMENTATION

CHI-OK HWANG

Abstract. Pseudo-random number sequences have been used in Monte Carlo methods in wide areas of science and engineer. Recently, quasi-random number sequences having uniform distribution property are used in numerical integrations, particle transport problems, financial problems and so on. Here, I survey on the quasi-random number generation and its applications. At the end, I discuss about the possible implementation of parallel quasi-random number generation.

1. Introduction

Since the early application of Monte Carlo methods to the direct simulation of random neutron diffusion in fissile material during the world war II, Monte Carlo methods have been used in wide areas [1] [2] of science and engineer such as percolation theory [3], diffusion problems, statistical mechanics, polymer chain simulation, radiation transfer, particle transport problem [4], financial problem [5], etc. For their stochastic property, pseudo-random and quasi-random (low-discrepancy) numbers are used. Pseudo-random number sequences try to mimic 'real' random numbers in theoretical and empirical tests. In contrast, quasi-random numbers give uniformly distributed sequences of numbers. In some Monte Carlo methods so called quasi-Monte Carlo methods, uniformity is more essential than randomness like in numerical evaluation of integrals. While the convergence of the standard Monte Carlo methods is only $O(N^{-1/2})$, that of the quasi-Monte Carlo methods can be approaching $O(N^{-1})$ in optimal cases.

However, the quasi-Monte Carlo methods have limitations. First, classically the methods are valid for integration problems. Secondly, the quasi-Monte Carlo methods generally lose their improved accuracy in high dimensional problems or non-smooth integrands. In many cases, the first problem can be overcome by rewriting the desired problem in the form of integration. To overcome the second problem, techniques like the generalized Brownian bridge and particle reordering were introduced, by which high dimensional problems are changed into rather moderate

1991 *Mathematics Subject Classification.* 11K36, 11B99.

Key words and phrases. quasirandom, parallel.

We wish to acknowledge the financial support of Computational Electronics Center, Inha University, Incheon, Korea for this work.

dimensional ones [6]. Also, hybrid (mixed) Monte Carlo methods which use mixed random number sequences appeared recently [4] [7].

For the uniformity test of a sequence of quasi-random numbers, star discrepancy defined as follows is used. At first, for a one-dimensional sequence of numbers, $\{x_n\}_{n=1}^N$, the star discrepancy [6] is

$$D_N^* = D_N^*(x_1, ..., x_n) = \sup_{0 \leq u \leq 1} |\frac{1}{N} \sum_{n=1}^{N} \chi_{[0,u)}(x_n) - u|, \tag{1.1}$$

where $\chi_{[0,u)}$ is the characteristic function of the half open interval $[0, u)$, which makes the term $\sum_{n=1}^N \chi_{[0,u)}(x_n)$ count the number of x_n's in the interval $[0, u)$. So, the term $|\frac{1}{N} \sum_{n=1}^N \chi_{[0,u)}(x_n) - u|$ measures the uniformity of the sequence in the interval $[0, u)$. The star discrepancy takes the supremum, maximal deviation from uniformity in the interval $[0, 1)$ to characterize the uniformity of the sequence.

More generally, for a set $\{x_i\}$ of N points in the d-dimensional unit cube, the star discrepancy [6] is

$$D_N^* = \sup_E |\frac{\# \text{ of } x_i \in E}{N} - m(E)|, \tag{1.2}$$

where E is a sub-rectangle of the unit cube, $m(E)$ is the volume of E and the sup is taken over all sub-rectangles. This definition is based on the observation that for a given rectangle, the fraction of points in the rectangle is close to the volume of the rectangle. Unfortunately, for moderate and high dimensions the theoretical star discrepancy bound is useless and numerical evaluation of the star discrepancy in high dimensions requires an enormous computational cost.

The motivation for developing good quasi-random number generation can be found in the following Koksma-Hlwaka inequality theorem [6],

$$|\frac{1}{N} \sum_{n=1}^{N} f(x_n) - \int_0^1 f(x)dx| \leq V(f) D_N^*, \tag{1.3}$$

where $f(x)$ has bounded variation $V(f)$ on $[0, 1)$ and $\{x_1, ...x_N\} \in [0, 1)$ has star discrepancy D_N^*. The first term on the right side is related to variance reduction techniques in (quasi)-Monte Carlo methods. Reducing the other term, the star discrepancy on the right side is the research area of quasi-random number generation.

A quasi-random number sequence satisfies the condition

$$D_N^* \leq C_d \frac{\log^d N}{N} \tag{1.4}$$

, where C_d is a N-independent but maybe dimension-dependent constant. Reducing the constant is one of the goals of developing a new quasi-random sequence.

Van der Corput sequence is the first quasi-random number sequence using the coefficients of the digit expansion of an increasing integer n in base 2 [8]. Many later quasi-random numbers are based on the Van der Corput sequence or generalizations of the Van der Corput sequence.

2. Quasi-random Number Sequences

Every integer $n \geq 0$ has a unique digit expansion

$$n = \sum_{j=0}^{\infty} a_j(n) b^j, \tag{2.1}$$

in base b, where $a_j(n) \in Z_b = \{0, 1, ..., b-1\}$ for all $j \geq 0$ and $a_j(n) = 0$ for all sufficiently large j.

Definition 2.1) For an integer $b \geq 2$, the radical-inverse function ϕ_b in base b is defined by

$$\phi_b(n) = \sum_{j=0}^{\infty} a_j(n) b^{-(j+1)} \tag{2.2}$$

for 0 and all positive integers, where n is given by its digit expansion in base b.

The function gives the symmetric reflection of the digit expansion in the decimal point. For an integer $b \geq 2$, the van der Corput sequence in base b is $\{x_0, x_1, ...\}$ with $x_n = \phi_b(n)$ for all $n \geq 0$ [9].

The original van der Corput sequence is in base 2 [8]. Halton used s van der Corput sequences with relatively prime bases $b_1, b_2, ..., b_s$ for different dimensions to construct an s-dimensional quasi-random number sequence.

$$x_n = (\phi_{b_1}(n), ..., \phi_{b_j}(n), ..., \phi_{b_s}(n)). \tag{2.3}$$

The Halton sequence gives a uniform distribution in lower dimensions ($s \approx 1-10$), as the dimension increases the sequence loses uniformity rapidly because of the correlations between the radical inverse functions for different dimensions and in high dimensions ($s > 30$) the sample points projected onto two dimensions are ordered into lines. To improve the discrepancy of the Halton sequence, Braaten and Weller [10] used permutations of the digits in the digit expansion of each van der Corput sequence to break the correlations between the inverse radical functions of different dimensions. The sequence

$$\phi_{b_j}(n) = \sum_i \sigma(a_i(j, n)) b^{-(i+1)}, \tag{2.4}$$

where σ is the operator of permutations on $a_i(j, n)$'s, is called a scrambled or generalized Halton sequence. Braaten and Weller presented their results for dimensions less than 17. The generalized Halton sequences remove the cycles of the original Halton sequence but introduces a certain degree of local nonuniformity. One advantage of the generalized Halton sequences is that we can construct different sequences only by changing the starting point [11]. But the problem is that in high-dimensional problems the sequences lose their uniformity property. Also, it seems to be needed to investigate the correlation of such sequences with different starting points. Anyhow, The Braaten-Weller reconstruction of the Halton sequence opened up a promising direction for improvement of the original Halton sequence in higher dimensions (Braaten-Weller algorithm is up to $s = 16$). Also, Hallekalek investigated the permutations of the Halton sequence [12]. Later, a modification of the Halton sequence (Halton sequence leaped, HaltonRR2) and a new construction of the generalized Halton sequence were developed by Kocis and Whiten [13]. The modification showed considerable improvements on the original Halton sequence in very high dimensions (observed up to 10^6).

To motivate the following definitions for the understanding of new sequences [9], let's observe the following property of the van der Corput sequence in a base $b \geq 2$. For fixed integers $k \geq 0$ and $m \geq 1$, considering the b^m points x_n with $kb^m \leq n < (k+1)b^m$, every b-adic interval $[ab^{-m}, (a+1)b^{-m})$, where $a \in Z$ and $0 \leq b < b^m$, contains exactly one point x_n with $kb^m \leq n < (k+1)b^m$. To understand this, note that for $kb^m \leq n < (k+1)b^m$ the m least significant digits in the digit expansion of n in base b can range freely while the remaining leading digits are fixed ; this means that for $x_n = \phi_b(n)$ its m leading digits after the decimal point can range freely while the remaining digits are fixed.

Definition 2.2 [9]. For a point set $P = \{x_1, x_2, ...x_N\}$and an arbitrary $B \subset \overline{I}^s := [0,1]^s$, let's define

$$A(B; P) = \sum_{n=1}^{N} \chi_B(x_n). \tag{2.5}$$

Definition 2.3) [9] For all subintervals J of $\overline{I}^s := [0,1]^s$, the s-dimensional Lebesgue measure is defined by

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \chi_J((x)_n) = \lambda_s(J), \tag{2.6}$$

when $x_1, x_2, ...$ is uniformly distributed.

Definition 2.4) [9] For the dimension $s \geq 1$ and an integer $b \geq 2$, let's define a subinterval E of $I^s := [0,1)^s$

$$E = \prod_{i=1}^{s} [a_i b^{-d_i}, (a_i+1)b^{-d_i}), \tag{2.7}$$

as an elementary interval in base b, where $a_i, d_i \in Z$ and $d_i \geq 0, 0 \leq a_i < b^{d_i}$.

Definition 2.5) [9] For a point set P of b^m points in $I^s := [0,1)^s$, and every elementary interval E in base b with $\lambda_s(E) = b^{t-m}$, a (t,m,s)-net in base b is defined such that $A(E;P) = b^t$, where $0 \leq t \leq m$ are integers.

Definition 2.6) [9] For an integer $t \geq 0$ and all integers $k \geq 0$ and $m > t$, a sequence $x_0, x_1, ...$ in $I^s := [0,1)^s$ is called a (t,s)-sequence in base b if the point set consisting of the x_n with $kb^m \leq n < (k+1)b^m$ is a (t,m,s)-net in base b.

As an another option to improve the Halton sequence besides the permutation methods such as the generalized Halton sequence and the HaltonRR2, quasi-random number sequences which use (t,s)-sequences showed up. A Sobol sequence [14] is a multidimensional (t,s)-sequence using base 2. This idea was developed to alternative multidimensional (t,s)-sequences with base $b \geq s$ by Faure [15]. However, the Faure sequences have very poor quasi-random number properties for large s [13]. In contrast, the Sobol sequence seems to maintain good properties for large s even though the size of patterns projected on two-dimensional plane by the Sobol sequence is larger than that of the generalized Halton sequence and the HaltonRR2.

The Sobol sequence was implemented by Antonov and Saleev [16]. To improve the efficiency, they modified the original Sobol sequence. Later, Bratley and Fox [17] and Press et al. [18] implemented the Sobol sequence again. In Bratley and Fox [17] implementation of the Sobol sequence and empirical comparison with the Faure sequence, it is shown that the Faure and Sobol sequences seem to be comparable to each other in accuracy but the Sobol generator with exclusive-or function is faster.

The Faure sequence was implemented and experimentally investigated with dimensions $s \leq 40$, and base $b = 41$ by Fox [19]. He found that in high dimensions the properties are very bad for the first $b^4 - 2$ points. So, his implementation starts from $n = b^4 - 2$. A serious problem of the Faure sequence is that as N increases the points projected onto two dimensions do not cover the unit square uniformly but part of the square in diagonal strips.

Niederreiter [9] [20] found a class of (t,s)-sequences defined using coefficients of the Laurent series based on his general construction principles for (t,s)-sequences. He generalized and improved earlier quasi-random number generation by Sobol and Faure. Note that the van der Corput sequence in base b is a $(0,1)$-sequence in base b. In 1993, Tezuka [21] developed a subclass of the Niederreiter sequences using polynomial arithmetic over finite fields.

The Niederreiter sequences were implemented in two ways (sequences in any prime-power base and sequences in base 2) and tested by Bratley et al. [22]. In their report, the Niederreiter sequence with base 2 was recommended due to the order-of-magnitude fastness and a slight loss of portability. The fastness of the base-2 Niederreiter sequence results from the binary nature of computers. Between the base-2 Niederreiter sequence and the Sobol sequence, the base-2 Niederreiter sequence was recommended theoretically even though empirical results did not show any guidance. Theoretically, the base-2 Niederreiter sequence is strictly better in dimensions $s > 7$ and comparable to the Sobol sequence in dimensions less than 7.

Contradicting to the original purpose of the introduction of the quasi-random numbers, in moderate and high dimensions($s \approx 40 - 400$), with any reasonable size N the integration theoretic error bound proportional to the discrepancy

$$D_N^* = O((\log N)^s/N), \tag{2.8}$$

is very much larger than the expected error of usual Monte Carlo integrations $(O(1/\sqrt{N})$ [13]. Also, in high dimensions the constant factors of the above error bound are unrealistically big [10].

Experimental studies were done for a limited number of dimensions using test functions [10] [17] [21] [22]. These computational studies show that the quasi-Monte Carlo method is better only when the sample size is sufficiently large than the usual Monte Carlo method. The sample size increases proportional to the dimension s.

Based on the existing (quasi-)random number generation methods, hybrid-Monte Carlo sequences have been recently used since Spanier [7] used first in his particle transport problems [4]. The hybrid (mixed or scrambled) s-dimensional sequence elements consists of a d-dimensional quasi-random sequence vectors followed by a $(s-d)$-dimensional pseudo-random sequence vectors.

Definition 2.7) [4] For a given d-dimensional quasi-random number sequence, $x_1, x_2, ...$ and $(s-d)$-dimensional pseudo-random number sequence, $y_1, y_2, ...$, a mixed (s, d) sequence is an s-dimensional sequence($s > d$) defined by

$$m_n = (x_n, y_n), \tag{2.9}$$

where $n = 1, 2, ...$

In high dimensions, compared to quasi-random sequences the mixed sequences give significant error reduction even though additional numerical work is necessary to quantify the improvement. For the low-dimensional part, a quasi-random sequence is used to take advantages of the superior error reduction property of the quasi-random sequence in low dimensions and a pseudo-random sequence for the remaining high dimensional part.

Recently, it is realized that the most commonly-used linear congruential generators (LCG's) for pseudo-random number generation could be used for quasi-random number generation via the method of "good lattice points" (GLP) [23]. This new method is based on the observation that LCG s-tuples lie on lattices composed of a family of hyper-planes. There are some tasks to be done to implement this method. First, it is necessary to identify multiplier-modulus pairs for LCG's which provides good lattices in the desired dimension. Secondly, the ways how to extend the point set size should be investigated more. One possible way is to use the additive constant in the LCG. The other way is to increase the modulus in the LCG. But the relationship of the lattices produced by LCG's of different moduli with related multipliers is not known.

Also, there is another possible alternative to generate quasi-random numbers using other higher order recursions besides LCG's.

3. Discussions and Conclusions

In low dimensions ($s < 30$ or 40), quasi-Monte Carlo methods in numerical integrations are better than usual Monte Carlo methods. In high dimensions, hybrid-Monte Carlo methods are preferable.

There are serially efficient implementations of the Sobol, Halton, Faure, and Niederreiter sequences [17][19][22]. Each of these sequences requires s one-dimensional quasi-random stream initializations and produces s-tuple quasi-random numbers. But there are only a few parallel implementations of the quasi-random sequences [24]. In Shukhman's implementation [24], he used basically Sobol sequences. When we consider the natural parallelism of quasi-Monte Carlo applications, implementation of parallel quasi-random number generation is very plausible. For the implementation of parallel quasi-random number generation for parallel and distributed computing systems, several considerations following the model case of pseudo-random number generation library (scalable parallel random number generators (SPRNG) library [25]) are necessary: (1) explicit parameterization of all quasi-random streams, (2) enough number of streams for a largest possible problem, (3) efficient and tractable initialization of the i-th stream, and (4) small and uniform computational cost of the generation of any quasi-random number stream. The second property eliminates some possible parallel quasi-random number generation methods. The minimal required number of available streams for possibly big problems is 2^{80} (compare to 2^{64} in SPRNG [25] for pseudo-random numbers). The Halton and Faure sequences require prime numbers. Because of the limited possible calculation of prime numbers (much less than 2^{80}), the two methods are not eligible.

Fortunately, the Sobol and Niederreiter sequences do not require primes but a distinct primitive polynomial modulo 2 for each dimension for the base-2 sequences. Using the efficient serial implementations of the base-2 sequences [22], parallel implementation is expected to be possible.

Also, thanks to the pseudo-random number library (SPRNG) and the recent development of the mixed sequences, there is another possible implementation for the parallel and distributed computing systems.

References

[1] K. Binder, *Monte Carlo Methods in Statistical Physics*, Springer, Berlin, (1996).

[2] H. Gould and J. Tobochnik, *An Introduction to Computer Simulation Methods*, Addison-Wesley, (1996).

[3] D. Stauffer and A. Aharony, *Introduction to Percolation Theory*, Taylor & Francis, (1994).

[4] G. Okten, Monte Carlo Methods and Applications **2**, 255(1996)

[5] S. Ninomiya and S. Tezuka, Appl. Math. Finance **3**, 1(1996)

[6] W. Morokoff, Siam Review **40(4)**, 765(1998)

[7] J. Spanier, Proc. Conf. on Monte Carlo Methods in Scientific Computing, Univ. of Las Vegas, 1994

[8] J. G. van der Corput, Nederl. Akad. Wetensch. Proc. Ser. B **38**, 813,1058(1935)

[9] H. Niederreiter, *Random Number Generation and Quasi-Monte C arlo Methods*(Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania 1992)

[10] E. Braaten and G. Weller, J. of Comp. Physics **33**, 249(1979)

[11] J. Struckmeier, J. of Comp. Appl. Math. **61**, 29(1995)

[12] P. Hallekalek, J. of Number Theory **18**, 41(1984)

[13] L. Kocis and W. J. Whiten, ACM Trans. Math. Software **23(2)**, 2 66(1997)

[14] I. M. Sobol, Zh. Vychisl. Mat. Mat. Fiz. **7**, 784(1967)

[15] H. Faure, Acta Arith. **41**, 337(1982)

[16] I. A. Antonov and V. M. Saleev, USSR Comput. Math. Math. Phys. **19**, 252(1979)

[17] P. Bratley and B. L. Fox, ACM Trans. Math. Softw. **14(1)**, 88(1988)

[18] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C. 2nd Ed. The Art of Scientific Computing*(Cambridge University Pre ss, New York, N.Y. 1992)

[19] B. L. Fox, ACM Trans. Math. Softw. **12(4)**, 88(1986)

[20] H. Niederreiter, J. of Number Theory **30**,51(1988)

[21] S. Tezuka, ACM Trans. Model. Comput. Simul. **3(2)**, 99(1993)

[22] P. Bratley, B. L. Fox and H. Niederreiter, ACM Trans. Model. Comput. Simul. **2(3)**, 195(1992)

[23] I. H. Sloan and S. Joe, *Lattice Methods for Multiple Integration* (Oxford University Press, New York, N.Y.(1994)

[24] B. Shukhman, Computer Physics Communications **78**, 279(1994)

[25] M. Mascagni, D. Ceperley, L. Mitas, F. Saied and A. Spinivasan, "SPRNG:Scalable Parallel Random Number Generators library", **http://sprng.cs.fsu.edu or http://www.ncsa.uiuc.edu/Apps/SPRNG**, 1998

Computational Electronics Center, Inha University, Incheon 402-751, Korea
E-mail address: chwang@hsel.inha.ac.kr