Fast Generation of Low–Discrepancy Sequences

J. Struckmeier
Department of Mathematics
University of Kaiserslautern
Germany

Abstract

The paper presents a fast implementation of a constructive method to generate a special class of low–discrepancy sequences which are based on Van Neumann–Kakutani transformations. Such sequences can be used in various simulation codes where it is necessary to generate a certain number of uniformly distributed random numbers on the unit intervall.

From a theoretical point of view the uniformity of a sequence is measured in terms of the discrepancy which is a special distance between a finite set of points and the uniform distribution on the unit intervall.

Numerical results are given on the cost efficiency of different generators on different hardware architectures as well as on the corresponding uniformity of the sequences. As an example for the efficient use of low–discrepancy sequences in a complex simulation code results are presented for the simulation of a hypersonic rarefied gas flow.

1 Introduction

The generation of pseudo–random numbers is one of the central parts in standard simulation methods. Typical fields of application are Monte–Carlo methods used for example in gas–surface scattering theory. Typically, pseudo–random numbers are used like ideal random numbers without looking in detail on the generators. A well know consequence of this negligence is the random number generator RANDU implemented in the IBM scientific Subroutine Package in the seventies which is described as "inaccurate, obsolete and downright dangerous to use" ([12]).
A theoretical concept to quantify the quality of pseudo–random numbers is the so–called discrepancy which was introduced by H. Weyl in his famous paper "Über die Gleichverteilung von Zahlen mod Eins" ([11]). From a theoretical point of view a lot of effort is spent by mathematicians to construct well–distributed sequences in multidimensions. Unfortunately this research does not influence the research on simulation methods used for applications.
One attempt in this direction was given by Pages and coworker who presented a comparison of different methods to generate random numbers.
The aim of the current paper is to present a fast algorithm to generate a special class of low–discrepancy sequences. From a practical point of view the efficiency to generate low–discrepancy sequences is a central aspect – it is not useful to use well–distributed sequences which have suitable uniformity properties, but are too expensive to generate. Chapter 2 recalls first the concept of linear congruential methods which are mainly implemented in subroutine packages. Furthermore a special class of low–discrepancy sequences, the so–called genralized Halton–sequences, is described. The algorithm to generate such sequences is described. Chapter 3 presents numerical results on the cost efficiency of such sequences as well as on the quality of the random numbers.

2 Random numbers and Low–Discrepancy Sequences

The classical approach to generate random numbers, which should be uniformly distributed on the unit intervall, is the linear congruential generator. This type of algorithm is the standard way to produce random numbers on a computer and most of the internal subroutine packages work with this kind of generator.
The second part of the chapter describes a special class of deterministic sequences, which have much better uniformity properties. Nevertheless they are not very often used in realistic applications.
One central aspect in realistic applications is typically the computational effort to perform the calculation. It is not often recognized that the generation of random numbers is a important part for the computational costs.
The aim of the chapter is to show that there exist very efficient algorithms to generate deterministic sequences which can even be faster than standard generator.

2.1 Linear congruential methods

Linear congruential methods for the generation of random numbers are given by the following recurrence relation

$$x_{n+1} = (a \cdot x_n + b) \bmod m$$

where a, b and m are some given constants.
It is obvious that the maximal period of the generator is given by the modulus m. Nevertheless the parameter a, b and m must be chosen carefully to achieve a reasonable period. The necessary condition for a maximal period m is given by the following well–known theorem.

Theorem 1
The linear congruential sequence has period m if and only if the following holds:

(1) b is relatively prime to m

(2) $a-1$ is a multiple of p, p being a prime divisor of m.

(3) $a-1$ is a multiple of 4 if m is a multiple of 4.

The study of Marsaglia ([3]) gives a method to determine the period for a given set $\{a, b, m\}$.
Normally the user of computer codes does not take care on the generation of pseudo–random numbers. The standard task is to take the generators, which are implemented directly on the computer. For example the generator implemented on UNIX–machines is provided with the parameters

$$\begin{aligned} a &= 69069 \\ b &= 1 \\ m &= 4294967296 \end{aligned}$$

This generator is used as an example for the computations given in the next chapter.
Using linear congruential methods problems may occur in the construction of multidimensional sequences. To generate uniformly distributed sequences in $[0,1]^k$ k consecutive numbers are choosen to produce a random k-tuple. The uniformity of those sequences may be very poor.
In general one may say that the confidence in the quality of generators can be dangerous; as given in the introduction see for example the popular IBM random number generator RANDU, which was implemented in the IBM scientific Subroutine Package in the seventies. This generator produces completely correlated consecutive triples ([4]).

2.2 Low discrepancy sequences

The theory of uniformly distributed sequences is based on the number theoretical definition of the uniform distribution mod 1 and the definition of the discrepancy.
This concept was introduced by H.Weyl in 1916 in his famous paper 'Über die Gleichverteilung von Zahlen mod. Eins' ([11]). It is out of the scope of the paper to discuss the number theoretical concept of the discrepancy and the estimates, which can be given for

some special sequences. The authors refer to the book of H.Niederreiter ([6]), where the reader may find all references necessary for a detailed study of this approach.
The low discrepancy sequence used in the current comparison is a generalization of the classical Van der Corput sequence with base 2 ([10]). In the current paper we will focus on the generation of this special class of low–discrepancy sequences.
A theoretical investigation together with numerical experiments can be found in [7] and [8], where the reader also find the proofs of the theorems given below.

Definition 1
Let z_1 and z_2 be two numbers in $[0,1]$ with p-adic expansions, where p is an arbitrary integer:

$$\begin{aligned} z_1 &= \sum_{k=0}^{\infty} \frac{z_1^k}{p^{k+1}} \\ z_2 &= \sum_{k=0}^{\infty} \frac{z_2^k}{p^{k+1}} \end{aligned}$$

with $\quad z_i^k \in \{0,...,p-1\} \quad \forall \quad i = 1,2; k \in I\!N.$
Then the the 'left addition' $\oplus$ is defined by

$$z_1 \oplus z_2 := \sum_{k=0}^{\infty} \frac{z^k}{p^{k+1}}$$

with coefficients z^k, $k \in I\!N$ given by the recurrence relation

$$\begin{aligned} z^0 &= (z_1^0 + z_2^0) \bmod p \\ z^k &= (z_1^k + z_2^k) \bmod p \quad + \quad \frac{1}{p}(z_1^{k-1} + z_2^{k-1} - z^{k-1}) \end{aligned}$$

The following example should illustrate the 'left addition' $\oplus$.

Example 1
Let $p = 2, z_1 = 0.10111$ and $z_2 = 0.100101$ then

$$z_1 \oplus z_2 = 0.10111 \oplus 0.100101 = 0.0110001$$

With the help of the left addition $\oplus$ one can define a sequence on $[0,1]$ by the following recurrence relation

Definition 2
Let p be an integer and $x_0 \in [0,1]$ arbitary.
Then

$$T_p(x) := x \oplus \frac{1}{p}$$

is called p–adic Van Neumann Kakutani transformation.

We use T_p for defining a sequence $(x_n)_{n \in N}$ recursively:

$$x_{n+1} = T_p(x_n)$$

Theorem 2
$\forall \quad x \in [0,1]$ the sequence $(T_p^k(x))_{k\in N}$ is uniformly distributed on $[0,1]$.

The quality of a given sequence in $[0,1]$ is measured using the so called "discrepancy" of the sequence, which gives an estimate on the uniformity of a finite section. Furthermore the sequence is called a low discrepancy sequence if the discrepancy of the points converge with the best possible order. It can be shown that the construction given above produces low–discrepancy sequences.

Definition 3
Let $(x_n)_{n=1,...,N}$ be a finite pointset in $[0,1]$.
Then the discrepancy $D((x_n)_{n=1,...,N})$ of the pointset is defined by:

$$D((x_n)_{n=1,...,N}) := \sup_{a\in[0,1]} |a - \frac{1}{N}\sum_{n=1}^{N} \mathcal{X}_{[0,a)}(x_n)|$$

The connection between the discrepancy of a sequence and the uniform distribution on $[0,1]$ is given by the following theorem.

Theorem 3
A sequence $(x_n)_{n\in N}$ is uniformly distributed in the unit intervall $[0,1]$ if and only if

$$\lim_{N\to\infty} D((x_n)_{n=1,...,N}) = 0$$

Theorem 4
The following estimate for the discrepancy of the sequence $(T_p^k(x_0))_{k=1,...,N}$ holds:

$$D((T_p^k(x_0))_{k=1,...,N}) \leq C \cdot \frac{\ln(N)}{N}$$

where the absolute constant C only depends on p and $\overset{o}{x}$.
Furthermore the estimate given above is the best possible order of convergence for a infinite sequence of points in $[0,1]$.

The main advantage of these sequences is the easy way to construct multidimensional uniformly distributed sequences.

Theorem 5
Let $p_1, ..., p_k$ be relatively prime.
Then the sequence $(x_n^1, ..., x_n^k)_{n\in N}$ in $[0,1]^k$ given by

$$x_{n+1}^i = T_{p_i}(x_n^i) \quad \forall i = 1, ..., k$$

and starting point $\overset{o}{x} = (\overset{o}{x}{}^1, ..., \overset{o}{x}{}^k)$ in $[0,1]^k$ is a multidimensional low discrepancy sequence.

In the following we will call sequences as defined above 'generalized Halton sequences'.
One advantage of the generalized Halton sequences is the way to construct different sequences only by changing the starting point $\overset{o}{x} \in [0,1]^k$.
This fact is important in simulation methods, where it is necessary to average over 'independent' samples in order to reduce the statistical scattering. It is clear that sequences with different starting point are in some sense correlated. This correlation is not investigated in the current paper, but will be topic of a forthcoming paper.

2.3 Fast Generation of generalized Halton sequences

From a computational point of view it is necessary to find a fast algorithm to generate the generalized Halton sequences in arbitrary dimensions.
In order to illustrate the algorithm used in the following figure 1 shows the graph of T_2

$$\begin{aligned} T_2 : \quad [0,1] &\longrightarrow [0,1] \\ x &\longrightarrow x \oplus \frac{1}{2} \end{aligned}$$

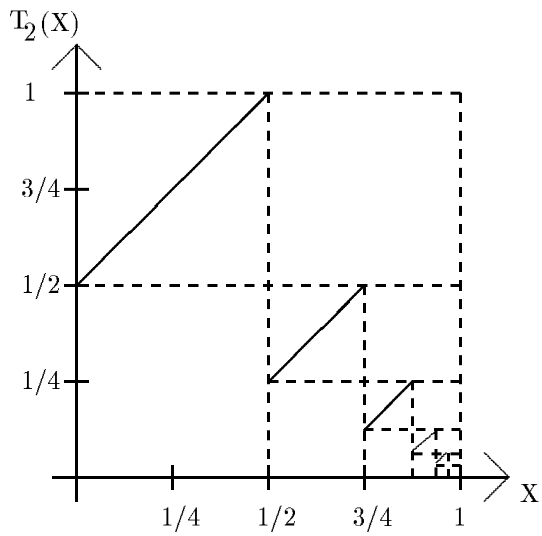

Fig. 1 : Function $T_2 : [0,1] \longrightarrow [0,1]$

The graph of T_2 is given by infinitely many parallel lines with slope 1.
The following representation of the Van–Neumann–Kakutani transformation T_p leads to an efficient algorithm to generate generalized Halton sequences.

Lemma 1
Let p be an integer and $x \in [0,1]$ arbitrary.
Define the sequence $(b_k^p)_{k \in N}$ by

$$b_k^p := \frac{1}{p^k} \cdot (p + 1 - p^k) \quad \forall k \in I\!N \tag{1}$$

Then the p–adic Van Neumann–Kakutani transformation T_p is given by

$$T_p(x) = x + b_k^p \tag{2}$$

where

$$k = [-\frac{\ln(1-x)}{\ln(p)}] + 1 \tag{3}$$

Using the above lemma the following relations are obvious:

$$(1) \quad b_1^p = \frac{1}{p}$$

$$(2) \quad \lim_{k\to\infty} b_k^p = -1 \quad \forall p \in I\!N$$

Consequently, in the limit $p \to \infty$ the graph of the function T_p degenerates to the line $T_p(x) = x$, which gives the identity on the intervall $[0, 1]$.
It is obvious that for the limit $p \to \infty$ the sequence can not be uniformly distributed on the unit intervall: the sequence is given by a constant point $\overset{o}{x}$.
Nevertheless the estimation for the discrepancy still holds which means that the absolute constant $C(p, \overset{o}{x})$ in front of the estimate must tend to infinity if p tends to infinity.
This result was already obtained by Faure, who gives the asymptotic behaviour for the absolute constant $C(p, \overset{o}{x})$ ([1]).
For practical applications it means that the generalized Halton sequences in high dimensions will be not very useful, because the quality of this components which corresponds to large values of p_k will be not sufficient.
This statement is validated by the numerical results given in ([8]).

Using the representations (1),(2) and (3) it is possible to developed a fast generator:

Algorithm (LD)
(to generate generalized Halton sequences to base p)

0) Initialization step:
   Generate the sequence (b_k^p) according to (3)

1) Choose an arbitrary starting point $\overset{o}{x} \in [0, 1]$

2) Suppose the random number x_n is given.
   Calculate the integer k according to (1) with $x = x_n$,
   then x_{n+1} is given by

$$x_{n+1} = x_n + b_k^p$$

Due to the finite number of digits used on a computer to represent a real number the generalized Halton–sequences, which have no period from a theoretical point of view, will only produce a finite set of numbers on a computer.
This can be used to implement the algorithm (LD):

Define an integer $M > 1$.
Generate the first M points of the sequence (b_k^p) and the corresponding partition of the unit intervall $[0, 1]$ into subintervalls of the form

$$[l_k^p, l_{k+1}^p],\ k = 0, ..M$$

where

$$l_0^p = 0 \qquad l_{M+1}^p = 1 \qquad l_k^p = 1 - p^{-k} \tag{4}$$

Implementation of Algorithm (LD)
(to generate generalized Halton sequences to base p)

0) Initialization step:
Generate the first M points of the sequence (b_k^p)
Generate the corresponding partition on the unit intervall according to (4)

1) Choose an arbitrary starting point $\overset{o}{x} \in [0, 1]$

2) Suppose the random number x_n is given.
If x_n is less then l_M^p,
determine the unique integer k such that $x_n \in [l_{k-1}^p, l_k^p]$
else determine the integer k according to (3),
then x_{n+1} is given by

$$x_{n+1} = x_n + b_k^p$$

For the numerical results presented in the next chapter the first 32 points of the sequence (b_k^p) respectively (l_k^p) were calculated a priori and used to generate the generalized Halton-sequence.
The implementation given above is very fast using for example FORTRAN 77 language, because more then 95% of the points are less then l_{32} and can be determined in a BLOCK–IF structure by a simple addition.

3 Numerical Results

The numerical results presented in this chapter give first an overview on the cost efficiency to generate the generalized Halton sequences. Furthermore we illustrate the uniformity properties in comparison with the classical linear congruential pseudo–random numbers. Because the main aspect of the paper is the fast generation of low–discrepancy sequences, we restrict ourselve to a small number of statistical tests, which is of course not enough to get a qualified comparison between low–discrepancy sequences and pseudo–random numbers. A detailled and much more elaborated investigation the reader may find in reference ([8]). Finally we present results on the efficiency in a complex simulation for the description of hypersonic rarefied gas flows.

3.1 Efficiency

Besides the uniformity of a sequence the computional costs which are necessary to generate a uniform sequence in $[0,1]$ play a central role in practical applications. It is obvious that the efficiency strongly depends on the hardware architecture. One may expect differences between standard and RISC architectures. Hence, in the current investigation four different machines were used to compare the efficiency:

1) IBM 6000/530

2) HP 9000/835 SRX (produced in 1988)

3) HP 9000/710

4) nCUBE 2S (1 node)

The generation of the generalized Halton–sequences is based on the algoithm presented in the last chapter. The program was written in FORTRAN 77 using a BLOCK–IF structure and 32 points of the sequence $(b_k^p)_{k\in N}$ (see chapter 2).

Hardware	g.H. (p=2)	LC (F77)	rand() (UNIX)
IBM 6000/530	1.9	2.8	1.6
HP 9000/835 SRX	4.8	25.8	12.9
HP 9000/710	1.0	3.1	2.0
nCUBE 2S 1 node	6.3	5.4	-

Tab.1: CPU–Time in seconds to generate 10^6 random numbers

Remark:
The CPU–time to generate Halton sequences to base $p > 2$ decreases, because the probability that the number will be generated by a simple addition in the first IF–structure of the algorithm (LD) is given by $\frac{p-1}{p}$.

The results show, that the generation of generalized Halton–sequences using the algorithm (LD) presented in the last chapter is very efficient.

3.2 Uniformity Properties

We restrict the comparison on the uniformity on a small number of statistical tests:

1) the discrepancy

2) the Weyl sum

3) the uniformity in higher dimensions

1) **The Discrepancy**
The following table gives the uniformity of the sequences in $[0, 1]$ – the measure used to quantify this property is the discrepancy as defined in chapter 2.

Sequence	D_N	V_M	D_N	V_M	D_N	V_M
Optimal	$1.72 \cdot 10^{-2}$		$5.15 \cdot 10^{-3}$		$2.89 \cdot 10^{-3}$	
rand()	$1.30 \cdot 10^{-1}$	$1.6 \cdot 10^{-3}$	$7.76 \cdot 10^{-2}$	$6.7 \cdot 10^{-4}$	$6.40 \cdot 10^{-2}$	$3.0 \cdot 10^{-4}$
g.H. (b=2)	$3.97 \cdot 10^{-2}$	$7.1 \cdot 10^{-5}$	$1.25 \cdot 10^{-2}$	$5.7 \cdot 10^{-6}$	$9.71 \cdot 10^{-3}$	$8.1 \cdot 10^{-7}$
g.H. (b=3)	$3.50 \cdot 10^{-2}$	$6.1 \cdot 10^{-5}$	$1.64 \cdot 10^{-2}$	$8.1 \cdot 10^{-6}$	$8.99 \cdot 10^{-3}$	$3.6 \cdot 10^{-6}$
g.H. (b=5)	$3.43 \cdot 10^{-2}$	$6.1 \cdot 10^{-5}$	$1.57 \cdot 10^{-2}$	$1.1 \cdot 10^{-5}$	$9.63 \cdot 10^{-3}$	$2.3 \cdot 10^{-6}$
	N = 29	M = 20	N = 97	M = 20	N = 173	M = 20

Tab.2: Averaged Discrepancy of different sequences

Remark:
D_N is the discrepancy of a finite set of N points
V_M is the variation of the discrepancy for M independent samplings

2) **The Weyl sum**
The quality of sequences can also be tested using the so–called Weyl sum defined by

$$\Phi = \int_0^1 \cos(2\pi x) \mathrm{d}x$$

Sequence	$\|\Phi\|$	$\|\Phi\|$	$\|\Phi\|$	$\|\Phi\|$
rand()	$2.11 \cdot 10^{-2}$	$1.45 \cdot 10^{-2}$	$1.19 \cdot 10^{-2}$	$2.91 \cdot 10^{-3}$
g.H. (p=2)	$1.43 \cdot 10^{-3}$	$1.48 \cdot 10^{-3}$	$1.44 \cdot 10^{-4}$	$5.59 \cdot 10^{-5}$
g.H. (p=3)	$2.20 \cdot 10^{-3}$	$1.20 \cdot 10^{-3}$	$1.56 \cdot 10^{-3}$	$2.12 \cdot 10^{-5}$
g.H. (p=5)	$5.70 \cdot 10^{-3}$	$2.74 \cdot 10^{-3}$	$1.05 \cdot 10^{-3}$	$4.69 \cdot 10^{-5}$
	N = 29	N = 97	N = 173	N = 307

Tab.3: Approximation Errors in the Weyl sum

Remark:
The results are obtained by taking 20 independent samples.

3) **Uniformity in higher dimensions**
An important aspect for practical applications is the uniformity of a sequences in higher dimensions.
The generic way to generate multidimensional sequences when using pseudo–random numbers is generate suceeding random numbers in $[0, 1]$.
The following table illustrate the expectation value $\mathbb{E}(xyz)$, which should be equal to 0.125 if the random number (x, y, z) is uniformly distributed in $[0, 1]^3$ and the corresponding variance.

Sequence	$\mathbb{E}_N$	V_M	$\mathbb{E}_N$	V_M	$\mathbb{E}_N$	V_M
rand()	$1.226 \cdot 10^{-1}$	$5.8 \cdot 10^{-4}$	$1.204 \cdot 10^{-1}$	$1.3 \cdot 10^{-4}$	$1.222 \cdot 10^{-1}$	$1.7 \cdot 10^{-5}$
g.H.	$1.275 \cdot 10^{-1}$	$4.1 \cdot 10^{-5}$	$1.254 \cdot 10^{-1}$	$1.3 \cdot 10^{-5}$	$1.252 \cdot 10^{-1}$	$6.2 \cdot 10^{-7}$
	N = 29	M = 20	N = 97	M = 20	N = 307	M = 20

Tab.4: Expectation value $\mathbb{E}(xyz)$ and variance

Remark:
V_M is the variation of $\mathbb{E}(xyz)$ for M independent samplings. The generalized Halton–sequence in $[0, 1]^3$ is based on the prime numbers 2,3 and 5. The threedimensional pseudo–random sequence is generated using the UNIX–$rand()$ subroutine and suceeding triples.

3.3 Example: Hypersonic Rarefied Gas Flows

One example for a simulation method where generalized Halton sequences can be efficiently used is the Finite–Pointset–Method (FPM) for the description of hypersonic rarefied gas flows ([5]). This particle method is used to solve the so–called Boltzmann equation which describes rarefied gas flows.
In this approach the velocity distribution function $f(t, x, v)$ of a rarefied gas ensemble is approximated by a finite particle set and the dynamic behaviour of the distribution function $f(t, x, v)$ which is described by the Boltzmann equation is transfered to a dynamical behaviour of the finite particle set.
The dynamical process consists of a free transport of the particle set over a small time increment Δt according to the given velocity of the particle and a binary interactions between between the particles – the so–called collision process.
The FPM code is used to describe the aerodynamic characteristics of a space vehicle during the reentry phase at high altitudes. The computational costs for this application are quite high because it is necessary to work with a reasonable number of discrete particles in order to obtain accurate results – for threedimensional computations with complex body geometry several millions of particles are used.
Pseudo random numbers are used at various steps of the calculation:

1) Aproximation of the initial velocity distribution by a finite set of particles

2) Approximation of the boundary conditions

3) Description of the collisions between the particles

Code improvement is always an important task in this research and one way is to improve the accuracy of the results, i.e. to reduce the statistical scattering. This approach was first investigated by Lecot ([2]) in a spatially homogenous relaxation problem. Within the framework of the European Space Project HERMES a detailed comparison of different simulation methods for rarefied gas flows as well as the behaviour of low–discrepancy sequences in the simulation approach was investigated by the author ([9]).
The efficiency of low–discrepancy sequences is demonstrated by the following two numerical experiments:
The first result concerns the approximation of the initial distribution function $\overset{o}{f}(x,v)$ by a finite set of particles. In typical applications the gas ensemble is assumed to be in an equilibrium state which is given by an Maxwellian distribution of the form

$$\overset{o}{f} = \frac{\rho}{(2\pi RT)^{3/2}} \exp(-\frac{(v-u)^2}{2RT})$$

The quantities which have to be approximated with a high accuracy are integral values over the velocity space – this integrals determine the macroscopic quantities of the gas flow, i.e. density ρ, flow velocity u and temperature T.
A typical result is given in figure 1:
The plotted curves show the statistical scattering using independent samples to determine the correct pressure in x–direction at the beginning of the computation. The curves are plotted versus the number of particles used for one independent sampling.
During the instationary calculation several sets of random numbers are used to determine the solution. The results given in figure 2 present the statistical scattering of the same quantity at the end of the computation. In this calculation, generalized Halton sequences are used to determine the postcollisional velocities of two particles when they undergo a collision.
The statistical scattering can be reduced by about 20% – in terms of the computational costs it means a reduction by a factor 2.

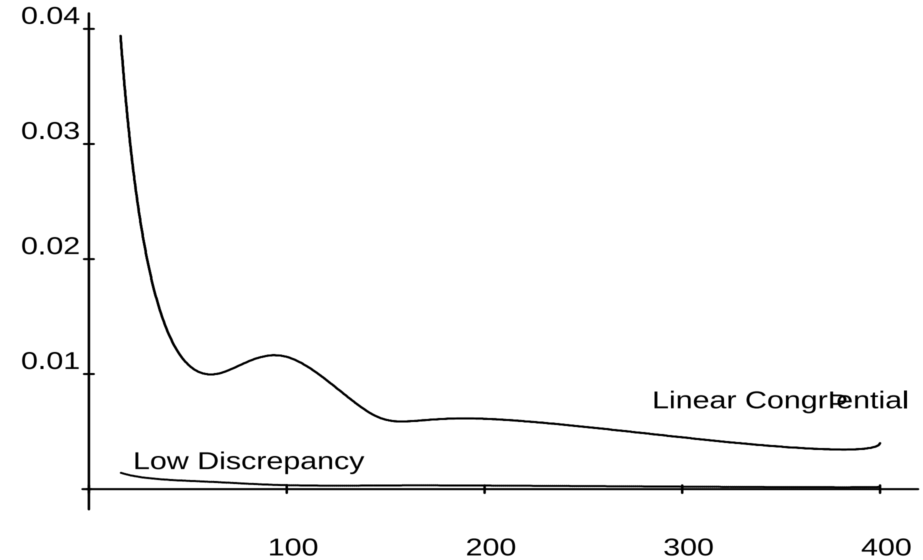

Fig. 1 : Statistical scattering for the inital pressure in x–direction

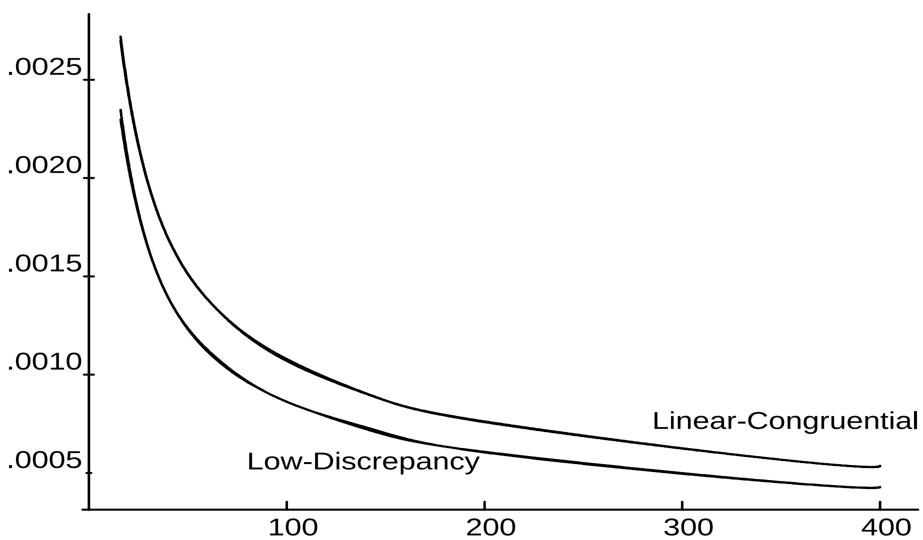

Fig. 2 : Statistical scattering for the pressure in x–direction

4 Conclusion

Generalized Halton sequences based on Van Neumann–Kakutani transformations can be generated with the same cost efficiency as classical linear congruential random numbers. Depending on the hard architecture the computational costs can even be lower. In addition to that, deterministic sequences stand out for much better uniformity properties than classical random numbers.
The common problem in simulation codes – the combination of deterministic sequences and independent sampling – can be eliminated by using generalized Halton sequences with arbitrary starting points; an investigation on the influence of the correlation of such sequences seems to be necessary. In high dimensional problems generalized Halton sequences loose a lot of their uniformity property – because of the 'singular limit' when the corresponding basis tends to infinity.
Nevertheless, in low dimensions they can be extremely useful to improve the accuracy of various simulation methods. This fact directly influences the computational costs of complex simulation and hopefully increases the fields of applications of such methods

Acknowledgement

The research presented in this paper was partly supported by the European Space Project HERMES under contract DPH 6473/91 of the HERMES Reserach on Qualification Phase.

References

[1] Faure, H.: Discrépances des suites associées à un systéme de numération (en dimension un), Bull. Soc. Math. France, 109 (1981), pp. 143–182.

[2] Lecot, C.: A Direct Simulation Monte Carlo Scheme and Uniformly Distributed Sequences for Solving the Boltzmann Equation, Computing 41 (1989), pp. 41–57.

[3] Marsaglia, G.: The Structure of Linear Congruential Sequences, in: Zaremba, S. (Ed.), Applications of Number Theory to Numerical Analysis, Academic Press, New York (1972), pp. 249–286.

[4] Marsaglia, G.: Random numbers fall mainly in the planes, Proc. Natl. Acad. Sci. 61 (1968), pp. 25–28.

[5] Neunzert, H.; Gropengießer, F. and Struckmeier, J.: Computational method for the Boltzmann equation, in: Spigler, R.(Ed.), Applied and Ind. Mathematics, Venice–1, 1989, Kluwer Acvad. Publ., Dordrecht (1991), pp. 111–140.

[6] Niederreiter, H.: Random Number Generation and Quasi–Monte Carlo Methods, SIAM, Philadelphia (1992).

[7] Pagès, G.: Van der Corput sequences, Kakutani transform and one–dimensional numerical integration, to appear in J. Comp. and Appl. Math.

[8] Pagès, G. and Xiao, Y.J.: Sequences with low discrepancy and pseudo–random numbers: theoretical remarks and numerical tests, to appear in J. Comp. and Appl. Math.

[9] Struckmeier, J. and Steiner, K.: A Comparison of Simulation Methods for Rarefied Gas Flows, Preprint, Berichte der Arbeitsgruppe Technomathematik, No. 92, June 1993, submitted to Physics of Fluids, Part A.

[10] Van der Corput, J.G.: Verteilungsfunktionen I,II, Nederl. Akad. Wetensch. Proc. Ser. B, 38 (1935), pp. 813–821, pp. 1058–1066.

[11] Weyl, H.: Über die Gleichverteilung von Zahlen mod. Eins, Math. Ann. 77 (1916), pp. 313–352.

[12] Yakowitz, S.J.: Computational Probability and Simulation, Addison Wesley, Massachusetts (1977).

Jens Struckmeier
Department of Mathematics
University of Kaiserslautern
P.O. Box 3049
67653 Kaiserslautern
Germany
FAX: (49) 631 205 3052
E-Mail: struckm@mathematik.uni-kl.de