Fractal Fundamental Domains of Canonical Number Systems. Some Applications to Monte Carlo and Randomized Quasi-Monte Carlo Methods in Realistic Image Synthesis

Alexander N. Kalouguine
Department of Geoinformatics, Samara State Aerospace University, Russia
Image processing systems institute of RAS, Russia
alexklg@nm.ru

Abstract

This paper describes the algorithm for construction of the 'naturally' multi-dimensional pseudorandom point generator based on the theory of canonical number systems in multidimensional algebraic structures. Applications of the generator to the tasks of computer graphics are considered. The method for using dual LFSR-CNS generators for scrambling existing point sets is described. Numerical results are provided.

Keywords: *Image synthesis, Monte Carlo and Quasi-Monte Carlo methods, multidimensional pseudorandom point generation*

1. INTRODUCTION

Many tasks of the computer graphics may be solved using Monte Carlo (MC) and Quasi-Monte Carlo (QMC) methods. Halftoning [1], global illumination problem [2],[4], form factor calculations [3], and many other tasks may be reduced to multi-dimensional integration over the multidimensional unit cube; contrast enhancement [5], edge preserving image smoothing [6] problems may be approached via random walks.

To estimate the value of the multidimensional definite integral both Monte Carlo and Quasi-Monte Carlo methods may be used. Even though the Quasi-Monte Carlo approach is more efficient (with respect to the number of points necessary to reach the given error), it imposes more strict limitations onto the integrands, there exist certain problems with projections of the QMC point sets to the integration domains of lower dimensionality (e.g., 2D projections of 7D/8D Halton and Hammersley point sets), and efficiency of the method decreases with the growth of the task dimensionality [7]. Application of the Randomized Quasi-Monte Carlo (RQMC) approach implies that the QMC point set/sequence may be modified using either a multi-dimensional random sequence (e.g., using Cranley-Patterson rotations [8]), a certain scrambling algorithm.

Generating a multidimensional pseudorandom sequence is not an 'easy' task. Usually these sequences (with very rare exceptions) are 'made of' the 1D sequence produced by the 1D PRNG via parallelization so to increase the number of dimensions of the original sequence. These methods may result in correlation between the coordinate sequences of the multi-dimensional sequence and eventually in incorrect calculation results [9].

An illustrative example of the correlated 3D sequence obtained from the 1D sequence is the 3D Randu [13] sequence. An illustration of the 'non-random' distribution of Randu 3D points is provided in the Figure 1.

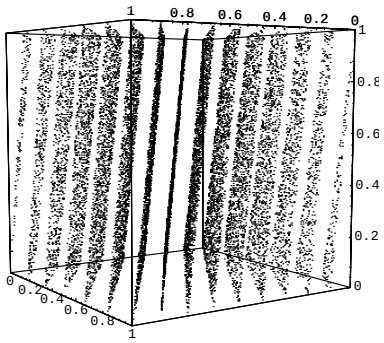

Figure 1: Randu 3D sequence, [13].

This should be mentioned, however, that for certain 1D generators, the effects of the increasing dimensionality may take place in much higher dimensions than that for the Randu sequence (e.g. Mersenne Twister is 623-equidistributed) however these effects are still observable.

This paper describes the algorithm for construction of the 'naturally' multi-dimensional pseudorandom point generator based on the theory of canonical number systems (LFSR-CNS generator) in the multidimensional algebraic structures. Also the method for using the dual LFSR-CNS generators for scrambling existing sequences is described.

2. MAIN IDEA

Many existing 1D pseudorandom number generators (PRNG) exploit number representation in conventional number systems with different bases. For example, the well known Tausworthe PRNG and multiple-recursive generator [23], [24] consist of 2 stages: (1) generation of the $s-$element vectors (usually bit vectors); (2) interpretation of the coordinates of these vectors as digits of the number expansion (of a fractional number) in a certain number system.

If a concept similar to the conventional number system may be defined in the multi-dimensional space, then the similar scheme may be used to generate not the 1D numbers but multidimensional points.

Such concept is canonical number systems (CNS), introduced by I. Kátai in [12] and further investigated by many authors [10], [14], [22].

In addition to the 'conventional number system properties', CNS in the multidimensional space exhibit unique properties, which may be successfully used both for random point generation and developing the scrambling techniques: in particular, the

fundamental domain [10] of the CNS usually represents the fractal set with the complex, irregular boundary. Properties of this set make this possible to construct an algorithm for 'random-like bijection' (see Section 7).

Below we provide description of the CNS-based pseudorandom point generation algorithm, the CNS-based scrambling technique and their applications to the MC and RQMC-based algorithms.

3. BACKGROUND

Let us recall the notation and basic results of the recurrence relations theory and theory of canonical number systems (CNS) necessary for the sequel.

Definition 1. A function defined in the finite field $\mathbf{GF}(q)$ of $q = p^s$ elements (p is prime), which satisfies the following linear recurrence relations with constant coefficients

$$y(n) + b_{s-1}y(n-1) + ... + b_0 y(n-s) = 0; \qquad (1)$$

$$b_0,...,b_{s-1} \in \mathbf{GF}(q), \quad b_0 \neq 0, \quad \vec{Y}(0) = (y(0),...,y(s-1)),$$

is said to be a linear recurrent sequence of degree s with the initial conditions (seed values) $\vec{Y}(0) = (y(0),...,y(s-1))$.

Linear recurrent sequence (1) of maximal period $T = q^s - 1$ is said to be the m-sequence.

Definition 2. The sequence $\{\vec{Y}(n)\} = \{\vec{Y}(0), \vec{Y}(1),...\}$, where

$$\vec{Y}(i) = (y(i), y(i+1),..., y(i+s-1))^T, \qquad (2)$$

is said to be '*caterpillar sequence of the sequence* (1)' The caterpillar sequence of the m-sequence also has the period of $T = q^s - 1$ elements [11].

Let us define the canonical number system in the k-dimensional lattice $\mathbb{Z}^k$. Let all the eigenvalues of the matrix $\mathbf{M} \in \mathbb{Z}^{k\times k}$ have moduli greater than 1. Let the set $D \subseteq \mathbb{Z}^k$ form the complete residue system (mod $\mathbf{M}$), containing the zero, and $D = \{v\vec{e}, v = 0,1,...,|\det \mathbf{M}| - 1\}$, where $\vec{e} = (1,0,...,0) \in \mathbb{Z}^k$.

Definition 3. The pair $(\mathbf{M}, D)$ is said to form [10] a canonical number system (CNS) in $\mathbb{Z}^k$, if for every element $\vec{\zeta} \in \mathbb{Z}^k$ there exists a unique expansion of the form:

$$\vec{\zeta} = \sum_{i=0}^{l} \mathbf{M}^i \zeta_i \vec{e}, \ \zeta_i \vec{e} \in D, \ \vec{e} = (1,0,...,0), \ \#D = q. \qquad (3)$$

For CNS $(\mathbf{M}, D)$, the matrix $\mathbf{M}$ is said to be the 'base' of the number system and the set D to be the 'digit set'. The companion matrices of the following polynomials may be used as the bases of the CNS [22].

$$f_1 = x^k + c_1 x + q,$$

$$\text{iff } -1 \le c_1 \le q-2; \ q \ge 2; \ |\det \mathbf{M}_{f_1}| = q = p;$$

$$f_2 = x^k + px^{k-1} + px^{k-2} + ... + px + p,$$

$$2 \le p \in \mathbb{N}, \ |\det \mathbf{M}_{f_2}| = q = p \ \ q = p;$$

$$f_4 = x^k + px^{k-1} + p^2 x^{k-2} + ... + p^{k-1}x + p^k$$

$$2 \le p \in \mathbb{N}, \ |\det \mathbf{M}_{f_4}| = q = p^k.$$

4. LFSR-CNS GENERATION ALGORITHM

The linear feedback shift register generator based on data representation in the canonical number system (LFSR-CNS generator [20]) generalizes (reinterpretes) the generation scheme by Tausworthe [23], [24] (so called Tausworthe generator, LFSR generator) and consists of two following stages:

Stage 1. Generate the caterpillar-sequence $\vec{Y}(i)$ of the m-sequence (1) with non-zero seed values $\vec{Y}(0)$. The degree of the sequence (1) should be $s = tk$, where k is dimensionality of the produced sequence and $t \in \mathbb{N}$ is a parameter controlling the period of the generator. The element of the caterpillar sequence $\{\vec{Y}(i)\}$ is said to be the *state of the generator*, the vector $\vec{Y}(0)$ is called the *initial state of the generator*.

Stage 2. The state of the generator is interpreted as a vector of digits in the expansion (3) of a certain element $\tilde{u}_i \in \mathbb{Z}^k$ in a q-nary canonical number system:

$$\tilde{u}_i = \sum_{j=1}^{s} \vec{Y}(i)_{j-1} (\mathbf{M}^{j-1}\vec{e}). \qquad (4)$$

Remark 1. In the lattice $\mathbb{Z}^k$ there exist several $q-$nary canonical number systems with different bases $\mathbf{M}$ and with the same digit set D. Thus, the single recurrence relation (1) used at the first generation stage induces a *family* of LFSR-CNS generators. The LFSR-CNS generators from one family will be called *dual* (see Section 7).

5. PROPERTIES OF THE LFSR-CNS GENERATOR

Theorem 1. At the output of the LFSR-CNS generator with non-zero initial state of generator, using the m-sequence of the order $s = tk$; $t, k \in \mathbb{N}$ in the finite field $\mathbf{GF}(q)$, there will be $q^s - 1$ different points of $\mathbb{Z}^k$ lattice generated.

Proof. As representation of the element $\tilde{u}_i \in \mathbb{Z}^k$ in the form (3) is unique, output points of the LFSR-CNS generator corresponding to different generator states will be different. Thus, if i runs the complete period of the m-sequence (1), there will be $T = q^s - 1$ different points $\tilde{u}_i \in \mathbb{Z}^k$ produced by the LFSR-CNS generator. ■

Corollary. For *any* number of dimensions, there exist parameters of the LFSR-CNS generator, which result in k-equidistribution [21] of the points at the output of the generator.

The set of points at the output of the generator (generator fundamental domain, $\tilde{F}$) may for certain CNS have an intricate shape and certain fractal properties (the fundamental domain of the LFSR-CNS generator is closely related to the fundamental domain of the CNS [10]).

In the Figure 2 there are illustrations provided of the fundamental domains of 3D LFSR-CNS generators.

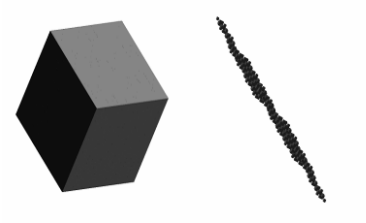

Figure 2. Fundamental domains of dual LFSR-CNS generators associated with the $\mathbf{M}_{f_1}$ (left) and $\mathbf{M}_{f_2}$ (right) CNS-bases; ($q = 2$)

This may be shown that for certain CNS, the complex fundamental domain of the associated generator may be efficiently (in terms of the computational complexity) converted into the multidimensional cube.

Theorem 2. Let the multidimensional cube be given by:

$$C(q,k,t) = \left([0;q^t)\right)^k \cap \mathbb{Z}^k,\ \#C(q,k,t) = \#\tilde{F},$$

then for the CNS associated with the M_{f_1}, M_{f_2}, M_{f_4} there exist parameters k,t of the generator such that the bijection $\mathrm{P}:\tilde{F} \to C(q,k,t)$ may be calculated in the following way:

$$\mathrm{Pr}_j\, P(u) = [\mathrm{Pr}_j\, u]_{\mathrm{mod}\, q^t},$$

where $[\cdot]_{\mathrm{mod}\, q^t}$ is a minimal non-negative residue of the class $\left(\mathrm{mod}\, q^t\right)$.

Remark 2. The bijection $\mathrm{P}:\tilde{F} \to C(q,k,t)$ is called *unification of the generator fractal domain*. Steps necessary to compute the unification constitute the 3rd stage of the LFSR-CNS generation scheme. An illustration for unification is provided in the Figure 3.

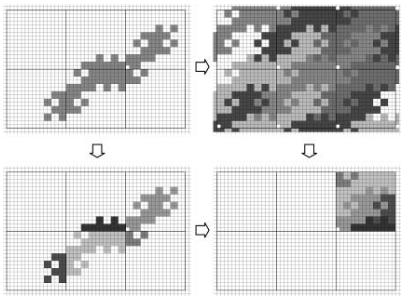

Figure 3. Unification of the fundamental domain

The additive shifts of the generator fundamental domain (top-left image) tessellate the multidimensional lattice (top-right image). However, if points at the output of the generator are considered (bottom-left image) not in the lattice but in the torus $\left(\mathrm{mod}\, q^t\right)$, the generated points fills all the torus points (except for zero point), or, if only the minimal non-negative residues are used, the points of the multidimensional cube (bottom-right image). The lattice tesselation induced by the fundamental domain represents the involute of the torus $\left(\mathrm{mod}\, q^t\right)$.

Below the pseudocode for the LFSR-CNS generation scheme (including unification) is provided (Pseudocode 1)

```
const q = 2;     //generator is binary
const int k = …; //desired generator dimensionality
const int t = …; //parameter controlling the generator period
                 //2^kt-1
const int s = k*t;
var int[0..s-1] A = ( ... ); //coefficients ai of the m-sequence
var int[0..s-1] Y; //state of the generator
typedef int[0..k-1] TBase; //Last column in the CNS Base.
                           //For comments, see below.
var TBase[0..s-1] MM;
function initializeGenerator(int[0..s-1] Y0) {
    for i from 0 to s-1 {
        Y[i] = Y0[i];
    }
}
function initializeCNSBase() {
// As the Set D of digits contains vectors with only 1st non-zero
// coordinate, only the first column of the matrices Mi in (3) and
// (4) is significant. Thus, only the first column is calculated
// using the properties of the matrix M.
    MM[0][0] = 1;
    for i from 1 to k {
        MM[0][i] = 0;
    }
    for i from 1 to s-1 {
        foo = MM[i-1];
        tmp = foo[k-1];
        for j from k-1 downto 1 {
             foo[j] = foo[j-1];
        }
        //for illustration, we use the CNS polynomial f2
        if (tmp != 0) {
             for j from 0 to k-1 {
                  foo[j] = foo[j] + (-q)*tmp;
             }
        }
        MM[i] = foo;
    }
}

function generateNextPoint()
{
    foo = 0;
    for i from 0 to s-1 { //generate the next element of the
                          //recurrent sequence
        foo = foo + (Y[i] * A[i]) mod q;
    }
    for i from 0 to s-2 {
        Y[i] = Y[i+1];
    }
    Y[s-1] = foo; // new state of the generator is calculated
    for j from 0 to k-1 {
      coord[j] = 0;
    }
    for i from 0 to s-1 { //calculating the multidimensional point
                          //with unification
        for j from 0 to k-1 {
             coord[j] = (coord[j] + Y[i]*MM[i][j]) mod q**t;
        }
    }
    for j from 0 to k-1 { //scaling
        coord[j] = coord[j] / (q**t);
    }
    //now the point is in the unit cube.
    return coord;
}
```

Pseudocode 1. LFSR-CNS generation scheme including unification

6. TESTING RESULTS

Properties of the sequence produced by the LFSR-CNS generator were tested applying various numerical tests including Knuth tests for the coordinate sequences [21], weighted spectral test (diaphony) [16], physical tests [15] and real-life tests via multidimensional MC integration [20]. Results of the numerical experiments verify good random properties of the generated multidimensional distribution.

As many of the modern testing libraries (e.g. Diehard, NIST, TestU01 [17], [18], [19]) are designed for testing 1D sequences, this is complicated to use these libraries to analyze multidimensional properties of the LFSR-CNS generator.

Also, as the LFSR-CNS generator doesn't exhibit lattice structure of the generated points this is impossible to use the conventional

spectral test [21], which appears to be the most powerful tool to assess the generator quality. However, values of the diaphony or the weighted spectral test proposed by Niederreiter [16] as an option for the generators, which cannot be analyzed using the spectral test, may be used instead.

As an illustration, the Figure 4 displays the results of the diaphony testing of the LFSR-CNS generator following the approach proposed in [16] .

For the single recurrence of degree $s = 48$, LFSR-CNS generators using the CNS with the base $\mathbf{M}_{f_2}$ $(q = 2)$ of 2D, 3D, 4D, 6D were created. The block of $K = 20$ point sets $\omega_i^{(N)}$, $i = 1, 2, ..., K$ that consist of $N = 2^2, 2^3, ..., 2^{15}$ successive points produced by the generator were used as the source data for test. For each of the blocks $\{\omega_i^{(N)}\}$, $i = 1, 2, ..., K$ the value $\hat{E} = \left\langle N \cdot F_n^2(\omega_i^{(N)}) \right\rangle$ were measured (see [16]) , where $F_n(\omega)$ is the diaphony of the set ω. It may be proved [16] that $E(N \cdot F_N^2(\omega)) = 1$ for the set of uniformly distributed independent points. The values provided in the Figure 4 shows that the value of the estimator $\hat{E}$ (vertical axis) are close to the best value 1 for any selected number of dimensions, the value of $\hat{E}$ doesn't grow with the growth of $\log_2 N$ (horizontal axis), which may be considered as an evidence of good random properties [16] of the generated sequence.

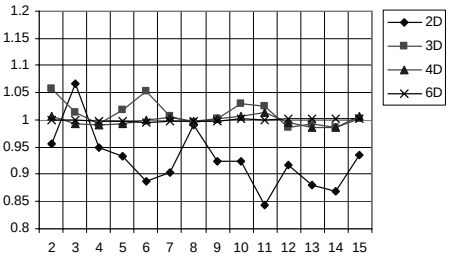

Figure 4. Diaphony test results. Value of the estimator $\hat{E}$ for the sets of 2D, 3D, 4D and 6D successive points at the output of the LFSR-CNS generator. The logarithm at the base 2 of the number of sample points is along the horizontal axis.

7. CNS SCRAMBLING

Regular (cubic) shape of the fundamental domain of the LFSR-CNS generator with the base $\mathbf{M}_{f_1}$ and complex fractal form of fundamental domain for CNS with the base $\mathbf{M}_{f_2}$ (see Figure 1) enables effective use of the dual LFSR-CNS generators for improving random quality of the multidimensional sequences. In the Figure 5, the flowchart of this algorithm is provided.

Step (1): Using the regularity of the cubic fundamental domain, given the multidimensional point, the state of the generator #1 is efficiently (in terms of computational complexity) reconstructed (this step is the inverse to the Stage 3 and Stage 2 of the LFSR-CNS scheme).

Step (2): The state of the generator #1 is interpreted as the state of the dual generator #2.

Step (3): The modified multidimensional point is constructed using 2nd and 3rd stages of the LFSR-CNS scheme.

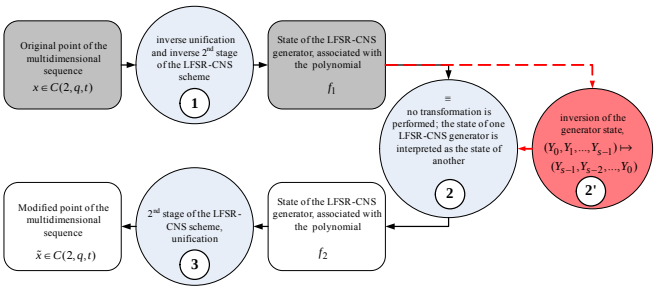

Figure 5. Multidimensional point set modification algorithm. Flowchart

The proposed method may be used in the arbitrary number of dimensions, application of the proposed scrambling technique (without the step 2') to the 3D *Randu* sequence removes the observed correlation. This result was verified using the numerical experiments with the weighted spectral test.

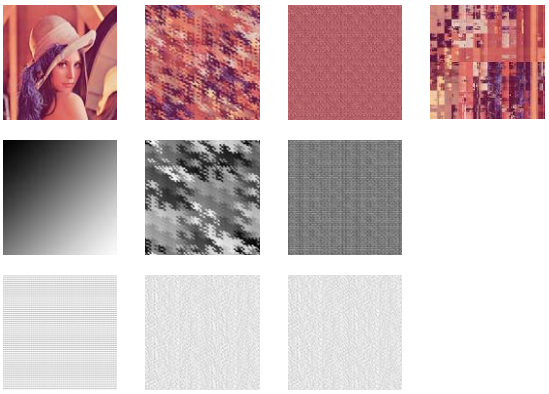

Figure 6. Use of dual generators to improve the properties of the multi-dimensional sequences. Column #1: Original image. Column #2: Scrambling without step 2'. Column #3: Scrambling with step 2'. Column #4: Owen scrambling.

If before the 'Step (2)', the vector, representing the state of the generator 1 is inverted, the scrambling result will be even more noticeable. In the Figure 6, the results of application of this CNS scrambling to reordering the pixels of 3 standard images are presented. The second column corresponds to the modification without the 'Step (2')' and the third column represents the results if the 'Step (2')' is used. In the first row, the forth column contains the Lenna image passed through the Owen scrambling [25] scheme.

8. CONCLUSION

The LFSR-CNS generator produces the naturally multidimensional sequence, which is free from drawbacks typical for the other multi-dimensional sequences obtained as a result of 'parallelizing' the 1D sequence. In contrast to the conventional PRNG, existence of the CNS in the lattices of any number of dimensions enables generation of sequences perfectly satisfying the requirements of a particular application.

Optimizations with respect to computational efficiency (CPU time) of the LFSR-CNS scheme is planned as a next research stage. Currently, computations according to the vanilla scheme

(see Pseudocode 1) may be computationally expensive especially if the number of dimensions k is high.

Families of the LFSR-CNS generators, different nature of the generator fundamental domains allows to use a pair of LFSR-CNS generator for scrambling of the multidimensional sequences. The method, being applied multidimensional sequence constructed from several 1D sequences, improves its multidimensional random properties.

These two results provide the grounds for application of the LFSR-CNS sequences to the tasks of machine graphics that may be solved via MC and RQMC methods.

9. ACKNOWLEDGEMENTS

The work was partially supported by U.S. Civilian Research & Development Foundation and by Russian Foundation for Basic Research (RFBR Project # 06-01-00722).

10. REFERENCES

[1] K. Hanson, *Quasi-Monte Carlo: halftoning in high dimensions*, SPIE Conf. Computational Imaging, January 20-24, 2003

[2] A. Keller, *Quasi-Monte Carlo Methods in Computer Graphics: The Global Illumination Problem*, Proc. of the SIAM Conference in Park City, 1995.

[3] A.Keller, *Quasi-Monte Carlo Methods in Computer Graphics*, in O. Mahrenholtz, K. Marti, and R. Mennicken (eds.), ICIAM / GAMM 95, Special Issue of ZAMM, Issue 3: Applied Stochastics and Optimization, pp.109-112, 1996.

[4] L. Szirmay-Kalos, W. Purgathofer, *Analysis of the Quasi-Monte Carlo Integration of the Rendering Equation*, TR-186-2-98-24, August, 1998.

[5] B. Smolka, K. Wojciechowski, *Contrast Enhancement of Gray Scale Images Based on the Random Walk Model,* CAIP 1999, pp. 411-418.

[6] B. Smolka, K. Wojciechowski, *Edge Preserving Image Smoothing Based on Self Avoiding Random Walk,* ICPR 2000, pp. 3668-3671.

[7] W.Morokoff, R.Caflisch, *Quasi-Monte Carlo integration*, J. Comput. Phys. 122, 1995, pp. 218-230.

[8] R. Cranley, T. Patterson. *Randomization of number theoretic methods for multiple integration,* SIAM J. Num. Anal., 1976, pp. 13:904-914.

[9] P. Hellekalek, *Don't Trust Parallel Monte Carlo,* In 12th Workshop on Parallel and Distributed Simultation, PADS'98, May 1998, pp. 82-89.

[10] W. Muller, J. Thuswaldner, R. Tichy, *Fractal properties of number systems,* Periodica Math. Hungar., to appear.

[11] R. Lidl , H. Niederreiter , *Finite Fields*, Addison-Wesley, Reading, Massachussets, 1983.

[12] I. Kátai , B. Kovács, *Canonical number systems in imaginary quadratic fields*, Acta Mathematica Academiae Scientarium Hungaricae, 37 (1-3), 1981, pp. 159-164.

[13] G. Fishman, L. Moore III, *An exhaustive analysis of multiplicative congruential random number generators with modulus* 2^{31}*- 1,* SIAM J. Sci. Statist. Comput. 7, 1986, pp, 24-45; erratum, ibid. 7, p. 1058.

[14] S. Akiyama, J. Thuswaldner. *A survey on topological properties of tiles related to number systems*, Geometriae Dedicata. vol. 109, 2004, pp. 89-105

[15] I. Vattulainen, *Framework for testing random numbers in parallel calculations*, Phys. Rev. E, 59, 6, 7200, 1999.

[16] P. Hellekalk, H.Niedderreiter, *The Weighted Spectral Test: Diaphony*, ACM Trans. on Model. and Comp. Simul., Vol 8., No. 1, 1998, pp. 43-60.

[17] Diehard tests, http://www.stat.fsu.edu/~geo/diehard.html

[18] NIST Test Suite, http://csrc.nist.gov/rng/

[19] TestU01, Empirical Testing of Random Number Generators, http://www.iro.umontreal.ca/~simardr/testu01/tu01.html

[20] A. Kalouguine, V. Chernov, *3D generalization for LSFR random point generator*, Proceedings of 2nd International Conference IASTED: Automation, Control, and Information Technology (ACIT`2005), Russia, Novosibirsk, June 20-24, 2005.

[21] D.E. Knuth, *The Art of Computer Programming,* Vol 2. Seminumerical Algorithms. Second Edition. Addison-Wesley. Reading. Massachusetts, 1981.

[22] A. Kovács, *Generalized binary number systems*, Annales Univ. Sci. Budapest, Sect. Comp. 20, 2001, pp, 195-206.

[23] R.Tausworthe, *Random Numbers Generated by Linear Recurrence Modulo Two*, Mathematics of Computation, 19, 1965, pp. 201-209.

[24] P. L'Ecuyer , *Maximally equidistributed combined Tausworthe generators,* Mathematics of Computation, 65, 213, 1996, pp. 203-213.

[25] A. Owen, *Randomly Permuted (t; m; s)-Nets and (t; s)-Sequences,* in Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, H. Niederreiter and P. Jau-Shyong Shiue Eds., Springer-Verlag, New York, 1995, pp. 299-317.

About the author

Alexander Kalouguine is a Ph.D. student at Samara State Aerospace University, Department of Geoinformatics. His contact email is alexklg@nm.ru