Computing 75, 281–295 (2005)
Digital Object Identifier (DOI) 10.1007/s00607-004-0105-z

Computing
Printed in Austria

Bad Lattice Points

K. Entacher, T. Schell and **A. Uhl,** Salzburg

Received November 8, 2002; revised March 9, 2004
Published online: May 18, 2005

Abstract

We introduce and discuss the term "bad lattice points" which can be seen as a counterpart to the method of good lattice points for Monte Carlo and quasi–Monte Carlo integration. We show several examples of the occurrence of bad lattice points in the latter fields and perform a computer search for such point sets.

Mathematics Subject Classifications: Primary 11Y40, 11-04; secondary 11K45, 68W40.

Keywords: Monte Carlo and quasi–Monte-Carlo methods, lattice rules, random number generation, spectral test, good lattice points.

1. Introduction

For an explanation or motivation of the term "bad lattice points", the prior introduction of a basic term from the fields of Monte Carlo (MC) and quasi–Monte Carlo (QMC) methods is required, the so-called method of *good lattice points* or *Korobov lattice rule*. Good lattice points are classical node sets for QMC integration, defined by the Russian mathematician Korobov [35]. Consider a parameter a with $1 < a < N$, $N \in \mathbb{N}$, and the s-dimensional vector $\mathbf{v} := (1, a, a^2, \ldots, a^{s-1})$, $s \geq 2$. A Korobov lattice rule is defined by the set

$$P_N := \{(n/N)\mathbf{v} \pmod 1) : 0 \leq n < N\}. \tag{1}$$

The set P_N can be seen as the intersection of the s-dimensional unit cube $I^s := [0, 1)^s$ with the integer lattice

$$L_s(a, N) := \left\{ \sum_{i=1}^{s} k_i \cdot \boldsymbol{b}_i : k_i \in \mathbb{Z} \right\}, \tag{2}$$

with lattice basis $\boldsymbol{b}_1 = (1, a, \ldots, a^{s-1})/N$, $\boldsymbol{b}_2 = (0, 1, 0, \ldots, 0)$, $\ldots$,$\boldsymbol{b}_s = (0, 0, \ldots, 0, 1)$.

The classical application of such Korobov lattice rules is the approximate calculation of integrals over I^s, by the (quasi–)Monte-Carlo quadrature rule

$$\int_{I^s} f(\mathbf{x})\, d\mathbf{x} \approx \frac{1}{N}\sum_{n=0}^{N-1} f(\mathbf{x}_n), \quad \mathbf{x}_n \in P_N. \tag{3}$$

More recent lattice rules, so-called rank-r lattice rules are constructed by modular summation over multiples of different vectors $\mathbf{v}_i$, $1 \le i \le r$. Korobov lattice rules are a special case of rank-1 rules. For more details on the theory of integration lattices, see [53], [54], [60].

Figure 1 shows examples of simple lattices P_N with $N = 2^7 - 1$, $a = 3$ (left) and $a = 53$ (right). Intuitively, one would call the left lattice a "bad" lattice and the right one a "good" lattice, since for QMC integration of an arbitrary 2-dimensional function one would choose the node set as evenly distributed as possible.

But how to distinguish between good and bad lattices? For this task, several equidistribution measures for an assessment of the lattice quality have been constructed, see [17], [27], [53], [60]. For our purposes we use the spectral test which can be computed very efficiently and provides a reliable measure for lattice assessment [17]. This test has extensively been applied to find good lattices for several MC and QMC applications, e.g., see [2], [22], [32], [41]–[43], [45]. For further development on the spectral test not only for lattices, see [25], [26], [54].

The spectral test uses the dual[1] lattice $L_s^*(a,N)$ of $L_s(a,N)$ which for Korobov lattice rules is simply given by a dual basis B^* where $\boldsymbol{b}_1^* = (N, 0, \ldots, 0)$, $\boldsymbol{b}_2^* = (-a, 1, 0, \ldots, 0)$, …, $\boldsymbol{b}_s^* = (-a^{s-1}, 0, \ldots, 0, 1)$.

From the latter basis, by means of the Fincke-Pohst algorithm [21], the shortest vector $\mathbf{v}$ of the dual lattice can be computed. One over the euclidean length of this shortest vector yields the spectral test d_s which determines the maximum distance between adjacent hyper-planes, taken over all families of parallel hyper-planes which contain all points of the lattice. Furthermore, the L_1-norm $|\mathbf{v}|_1$ of the shortest vector minus one gives an upper bound n_s of the smallest number of

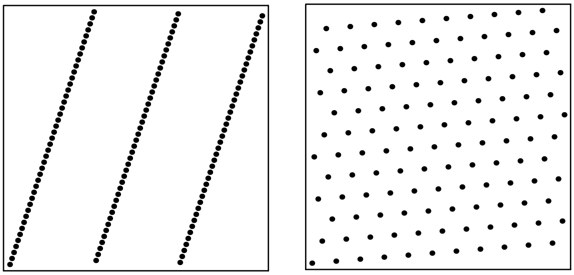

Fig. 1. Lattice rules P_N with $N = 2^7 - 1$ and $a = 3$ (left) and $a = 53$ (right)

[1] The *dual* of a lattice L_s is defined as $L_s^* := \{\boldsymbol{w} \in \mathbb{R}^s : \boldsymbol{w} \cdot \boldsymbol{v} \in \mathbb{Z} \text{ for all } \boldsymbol{v} \in L_s\}$. The dual basis of a given lattice basis $B = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_s\}$ is provided by the set of vectors $B^* = \{\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_s^*\}$ such that $\boldsymbol{b}_i \cdot \boldsymbol{b}_j^* = \delta_{i,j}$, with $\delta_{i,j} = 1$, if $i = j$ and $\delta_{i,j} = 0$ otherwise.

hyper-planes on which all the points of the lattice lie, see [12] and also [22], [34], [44]. Widely used is also a normalized spectral test $S_s := d_s^*/d_s$, $2 \leq s \leq 8$, for which $0 \leq S_s \leq 1$ (values near 1 imply a good lattice structure). The constants d_s^* are absolute lower bounds on d_s, see [22], [34]. L'Ecuyer [42] used also certain lower bounds d_s^* for dimensions $s > 8$ in order to compute S_s for arbitrary dimensions. For our examples in Fig. 1 above we have $S_2 = 0.26$ and $n_2 = 3$ for the left graphics and $S_2 = 0.99$ and $n_2 = 12$ for the right one.

In the present paper, we will not consider good lattices[2]. As the title already suggests, we will investigate the counterparts, called "bad lattice points" (BLPs). But what do we mean by BLPs? For example, it is easy to construct Korobov lattice rules with very bad lattice structures. If the parameter a is small such as $a = 2, 3, 4, \ldots$, the vectors of the corresponding lattices are placed on a small number of hyperplanes in *all* dimensions $s \geq 2$. This can be seen from the dual basis vector $\mathbf{b}_2^*$ above. For small parameters a the latter vector is already a short vector in the dual lattice and therefore the number n_s of hyperplanes containing all lattice points is lower than $a = |\mathbf{b}_2^*|_1 - 1$. Therefore, one might call Korobov lattice rules with very small parameters "worst lattice points". Moreover, the parameters $a_i := a^i \pmod{N}$ are also small for small powers i. Therefore, lattices rules $L_s(a_i, N)$ from such powers are of poor quality as well. Following the idea of lattices providing poor quality lattice points across a wide range of dimensions, we give the following definition.

Definition 1: *A lattice rule is defined to deliver "worst lattice points" with respect to a threshold-vector* $\mathbf{t} = (t_1, \ldots, t_s)$ *iff* $S_i < t_i$, $\forall i = 1, \ldots, s$.

Since this is a very strict demand it is hardly suited as the only means to characterize poor quality lattices. For a more general definition, low quality is only required in one dimension.

Definition 2: A lattice rule is defined to deliver "bad lattice points" with respect to a threshold-vector $\mathbf{t} = S_i < t_i$, $1 \leq i \leq s$, $\forall i = 1, \ldots, s$.

How should we set $\mathbf{t}$ to result in poor quality lattice rules ? A common approach is to use a fixed value across dimensions, e.g., $t_1 = \ldots = t_s = 0.1$. As an example we consider multiplicative prime modulus LCGs with $m = 467421271$ and randomly generate 1000000 primitive root multipliers which are evaluated with respect to $\mathbf{t} = 0.1$. Table 1 displays the number of multipliers which are rated as "bad" according to Definition 2.

Table 1. Number of "bad" generators

$s = 2$	3	4	5	6	7	8	16	24
11323	2484	436	44	2	0	0	0	0

[2] On selection of good lattices see [18], [23], [41]–[43], [45], [53], [61].

We observe a steady decrease of the number of multipliers which are rated as bad with increasing dimension s and no more such multipliers for dimensions $s \geq 7$. This phenomenon may be explained in two ways: either there are no poor quality lattice rules in higher dimensions or our criterion is not suited to identify them. No matter what is true, a pragmatic approach is to rate a lattice rule as of poor quality in case it is contained in a small quantile of the empirical distribution of the quality of all multipliers, e.g., in the 1% quantile. Table 2 shows the thresholds (depending on dimension) which need to be applied to result in 1% bad lattice points. These estimates have been obtained by averaging values across a wide range of moduli.

It is very interesting to note that the thresholds increase with increasing dimensions (this fact may also be derived from numerical results in [1], [2]). The values in Tables 1 and 2 are our principal motivation to define bad lattice points with respect to a threshold vector instead of a fixed threshold value. These results also suggest that the normalization factors within the normalized spectral test in high dimensions [42] require refinement in order to use this measure for comparisons across dimensions.

In this work we concentrate on BLPs which occur in the field of random number generation. Certain vectors of linear random numbers produce lattice structures if all numbers of the generator are consumed. In Sect. 2, we discuss BLPs of the latter type which mainly occurred if the performance of the generator's implementation dominated the design decisions. We denote LCGs of this type "unreasonable". We discuss such generators which have been implemented in commercial software.

We also use the term BLP to refer to lattices which have been selected due to their high quality in certain applications or dimensions, but it turned out that the same lattices used in different applications or different dimensions show poor quality (or also a sub-lattice or a projection to certain indices behaves badly). Such scenarios frequently occur in different applications of MC and QMC methods. Lemieux and L'Ecuyer [45], for example, recently showed examples of high quality Korobov lattice rules with poor projections, i.e., lattices obtained by projections over sub-spaces showed poor quality. In Sect. 3, we analyze further examples of BLPs in the meaning as outlined above. The quality of the BLPs in Sects. 2 and 3 is analyzed theoretically and empirically by means of the spectral test. Section 4 contains results of an extensive computer search for BLPs.

2. BLPs: Unreasonable LCGs

Linear congruential random number generators (LCGs) have extensively been applied as a source for randomness in computer simulation for a long time and they are until now the most common random number generators. Although we

Table 2. Thresholds resulting in 1% : "bad" generators

t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_{16}	t_{24}
0.10	0.16	0.22	0.27	0.31	0.35	0.37	0.51	0.57

have to mention that recent versions use implementations based on a combination of LCGs, or multiple recursive generators (MRGs) to get improved quality and huge periods. The definitions and basic properties of linear random number generators are contained in [22], [34], [38], [53]. LCGs are generated by means of the recursion $y_{n+1} \equiv ay_n + b \pmod{m}$, $n \geq 0$, and by an initial seed y_0, $a \neq 1$, b, $y_0 \in \mathbb{Z}_m$ (we abbreviate by $LCG(m, a, b)$). Normalized PRNs in $[0, 1[$ are obtained by the transformation $x_n := y_n/m$. An important property of linear congruential generators in general (this also holds for combined LCGs and for MRGs), is that arbitrary s-dimensional vectors

$$\boldsymbol{x}_i := (x_i, x_{i+j_1}, \ldots, x_{i+j_{s-1}}), \tag{4}$$

with fixed lags $j_1, \ldots, j_{s-1}$, are contained in certain grid structures (shifted lattices) [44]. For $j_1 = 1, \ldots, j_{s-1} = s-1$, the case which has been studied in detail, these vectors are called overlapping s-tuples. For multiplicative LCGs with prime moduli, the latter s-tuples produce Korobov lattice rules, i.e., intersections of a lattice $L_s(a, m)$ with the s-dimensional unit cube I^s. LCGs with power-of-two moduli produce shifted versions of the latter lattice, see [22], [34], [38], [47], [53], [56]. Essentially different lattices $L_s(a', m')$ are obtained for vectors (4) with arbitrary lags which occur if, for example, lagged subsequences from the output of an LCG are used [16], [38], [44].

For different reasons, LCGs have been implemented in commercial software where certain vectors (4) above showed extremely bad lattice structures. The most famous of these LCGs is the well known "IBM" generator RANDU $LCG(2^{31}, 2^{16} + 3 = 65539, 0)$ [22], [24], [25], [34], [36], [55]. The bad behavior of generators like RANDU should be well known in the scientific community for long years. Nevertheless, related generators of poor quality found ways to be recommended or implemented in recent literature or software, respectively.

In the following section we consider such generators. The first examples are[3] $UG1 := LCG(2^{32}, 3141592653, 1)$ implemented in the mathematical software DERIVE (*www.derive.com*) and the generator[4] $UG2 := LCG(2^{35} - 31, 5^5 = 3125, 0)$ described in [58].

Both LCGs show bad spectral test results in dimension two. For $UG1$ the spectral test in dimension $s = 2$ and 3 equal $S_2 = 0.09718$ and $S_3 = 0.5552$ and for $UG2$ we get $S_2 = 0.01569$ and $S_3 = 0.8564$. Figure 2 shows zooms into the lattice structure

[3] The multiplier probably stems from Knuth [34, pp. 32; 44; 102], who studied $LCG(2^{35}, 3141592653, 2718281829)$ (note that the digits of the multiplier equal the first digits of π in its decimal representation).

[4] Note, that $UG2$ was already used in the sixties. Quote from [46]: *The generator was described by Hutchinson* [29] *and ascribed to Professor D. H. Lehmer. Hutchinson discussed a particular form of the generator for the IBM 7094, in which $p = 2^{35} - 31$ is the largest prime less than 2^{35} and $A = 5^5$. Unfortunately, his tests on this generator were not published; our own tests and use of the generator confirmed that it is an exceptionally good pseudorandom number generator.*

A similar generator `Mgen()` $= LCG(2^{26}, 5^5, 0)$, used for shuffling purposes, is implemented in the simulation software C++SIM (*cxxsim.ncl.ac.uk*). Note, that *Mgen()* shows satisfying spectral test results.

of $UG2$ which exhibit the coarseness of the lattice in dimension two and the fine lattice structure in dimension three.

Generators with extreme defects are $UG3 = LCG(2^{47}, 2^9 + 1 = 513, 297410973)$, used for the numerical calculations in [28] (first empirical results of this LCG with different additive constant are given in [31]), and $UG4 = LCG(2^{48}, 2^{24} + 3, 0)$, which corresponds to the NETLIB Module RANDNUM-CRAY (a vectorized random number generator for the Cray X-MP, see [3] and *www.netlib.org*). Similar as for RANDU the multipliers for these LCGs were chosen to obtain fast implementations (e.g., a multiplication with $2^9 + 1$ is equal to a shift of 9 bits and an addition of 1). The spectral tests of both LCGs show spectacularly bad results, see Table 3. LCGs with power of two moduli $m = 2^\alpha$ and multipliers close to a power of two are well known to produce bad lattice structures, since such multipliers are solutions of certain polynomial equations in $\mathbb{Z}_m$ with very small coefficients which is equivalent to the existence of short dual vectors and therefore bad spectral tests, see below and [52, p. 1026], [34, p. 104], [22], [33], [50]. One of the first (mixed) LCGs of this type for example is the generator $UG5 = LCG(2^{35}, 2^7 + 1, 1)$, already proposed by Rotenberg in 1960 [59] (citation from [31], [50]). The related $LCG(2^{32}, 2^7 + 1, 907633385, 0)$ was implemented in Version 3.0 of Turbo-Pascal (Borland International), for its bad behavior see [14], [55].

In the Introduction, we already demonstrated that small multipliers result in bad lattice structures. Generator $UG5$ and also $UG3$ still belong to this case. We

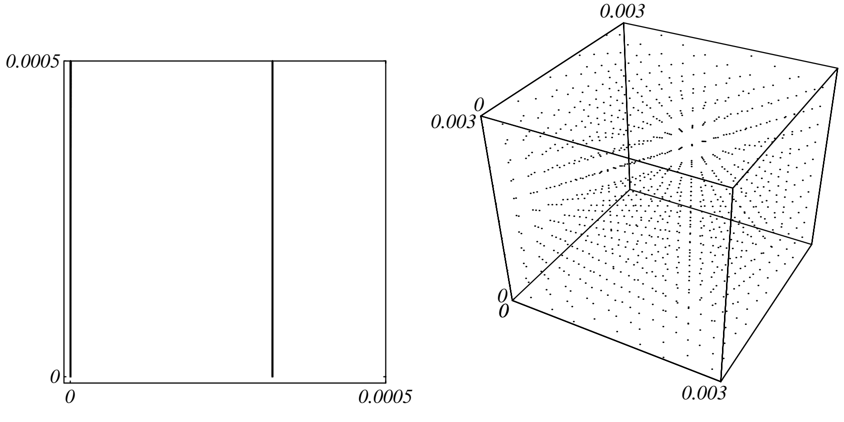

Fig. 2. Zooms into the lattice structure of overlapping vectors from $UG2$

Table 3. Spectral tests S_s and n_s for $UG3$, $UG4$ and $UG5$

		$s = 2$	3	4	5	6	7	8
$UG3$	S_s	**0.00004**	0.0088	0.1252	0.6168	0.2157	0.1297	0.1022
	n_s	513	513	513	513	127	39	19
$UG4$	S_s	0.6580	**0.00023**	0.0031	0.01487	0.0411	0.0841	0.1415
	n_s	8388609	15	17	17	17	17	17
$UG5$	S_s	**0.00065**	0.0353	0.2520	0.7926	0.2157	0.2128	0.2892
	n_s	129	129	129	129	31	19	19

recall that for small parameters a and certain dimensions s, the dual basis vectors $\mathbf{b}_2^* = (-a, 1, 0, \ldots, 0)$ are already short vectors in the corresponding dual lattice. Therefore, for $UG5$ we get $n_s = 129$, $s = 2, 3, 4$ and for $UG3$, $n_s = 513$ for $s = 2, 3, 4, 5$. For larger dimensions some shorter dual vectors occur (compare Table 3). For these dimensions the form of the multiplier, in our case $a = \beta + 1$ with $\beta = 2^\alpha$, causes these short vectors. For such a multiplier a the modulus m can easily be expressed as $m = \sum_{j=0}^{s-1} c_j a^j$, $c_j \in \mathbb{Z}$ and therefore $n_s \leq \sum_{j=0}^{s-1} |c_j| - 1$. The latter property is a special case of the more general Proposition 3 in [40].

As an example, consider $UG3$ in dimension $s = 6$. We have $m = 4(a-1)^5 = 4(a^5 - 5a^4 + 10a^3 - 10a^2 + 5a - 1)$, which yields a dual lattice vector $\boldsymbol{w} = (-4, +20, -40, 40, -20, 4)$. For this vector we get $n_6 \leq |\boldsymbol{w}|_1 - 1 = 127$ which equals the spectral test in Table 3 above. As a final example consider $UG4$ for which $a = \beta + 3$, $\beta = 2^{24}$. In the same way as above, for dimension $s = 3$ we get $m = (a-3)^2$, hence $\boldsymbol{w} = (-9, 6, -1)$ as a dual vector, which yields $n_3 \leq |\boldsymbol{w}|_1 - 1 = 15$, which again equals the corresponding value in Table 3.

LCGs with multipliers near a power of two, in particular RANDU, have widely been used for a long time and implementations are still available. Some further examples and references concerning the latter LCGs are contained in [7], [31], [47], [51], [52].

Not only LCGs may produce bad lattice structures. L'Ecuyer et al. [5], [6], [40], [62], [63] identified bad lattice structures for overlapping vectors and vectors of non-successive values produced by several linear methods (multiple recursive generators, lagged-Fibonacci generators). Especially the *add-with-carry* (AWC) and *subtract with borrow* (SWB) pseudorandom number generator proposed by Marsaglia and Zaman [49] exhibited extremely bad lattice structures in high dimensions. This is due to the fact that AWC and SWB generators are almost equivalent to special LCGs with large moduli.

An example is $LCG(m = a^{48} - a^8 + 1, a = 2^{31}, 0)$ which closely approximates the *subtract with borrow* (SWB) [49] pseudorandom number generator which is implemented in *Mathematica*[5] (SWB version 7 with period $\approx 10^{445}$ given in Table 2 of the latter paper). In the case of the *Mathematica* generator the (non-normalized) spectral test $d_s = 1/\sqrt{3}$ for dimensions $s \geq 49$ (4.6×10^{-10} for $s \leq 49$). Moreover, certain three-dimensional vectors of non-successive values of such generators lie in parallel planes that are at least $1/\sqrt{3}$ apart [40].

Another similar SWB generator given in [49, Vers. 3, Table 2] and used as a component of the combined generator proposed in [48] is almost equivalent to $LCG(m = a^{43} - a^{22} + 1, a = 2^{32} - 5, 0)$. Spectral tests and Beyer quotients for this generator are given in [5], [63], empirical results in [37]. The lattice structure of the combined generator from [48] is examined in [5]. Further empirical results which exhibit defects of AWC and SWB generators are given in [20].

[5] *www.wri.com*; *www.wolfram.com*

3. BLPs: Long Range Correlations

In contrast to the previous section we now consider the situation where the parameters of an LCG are well chosen with respect to lattice structures of overlapping vectors. But if we change certain lags in (4), then bad lattices occur. In the following we consider the case of prime LCGs only.

As an example, we examine special lags of the form $j_1 = l$, $j_2 = 2l, \ldots, j_{s-1} = (s-1) \cdot l$ with $l \geq 2$. An analysis of the corresponding vectors $\mathbf{x}_i = (x_i, x_{i+l}, x_{i+2 \cdot l}, \ldots, x_{i+(s-1) \cdot l})$ can be seen as an analysis of correlations between consecutive blocks of random numbers with block-length l. Elementary calculations (see [16], [53, p.172] or [56] for related concepts) show that for certain LCGs the vectors $\mathbf{x}_i$ are contained in the intersection of I^s with a shifted lattice $L_s(c, m)$ with $c = a^l \pmod{m}$, m the modulus of the LCG.

Especially when the lags l are large we obtain the well known analysis of *long-range correlations*[6] of random numbers [9], [8], [10], [11], [13]. The latter papers treat the case $s = 2$ which can be seen as an analysis of correlation between pairs of large consecutive blocks of random numbers. We can also study long-range correlations between larger numbers of blocks which is equal to a lattice analysis for larger dimension s. An empirical analysis of such long-range correlations for higher dimension is contained in [19].

As an example, we consider the widely used prime LCG $G1 := LCG(m = 2^{31} - 1$, $a = 16807$, $0)$. This particular generator has widely been used and actual implementations are available from the Internet. See [3], [8], [22], [30], [36], [38], [39], [47], [55], [56] for references, empirical tests and implementations in free and commercial software. The following online resources contain related material: Resampling Stats (*www.resample.com*), Numerical Recipes (*www.nr.com*), the mathematical software MATLAB (*www.mathworks.com*), the IMSL Libraries, or the simulation software ACSL (*www.acslsim.com*), SIMAN/Arena, Slam II, AweSim (*www.pritsker.com*) and the network simulation tools ns-2 (*www.isi. edu/ nsnam/*) and OMNeT++ (*www.hit.bme.hu/phd/vargaa/*).

Interesting BLPs occur if we use lags $l = (m-1)/i$ with small divisors i of the period $m - 1$ (note that m is prime). Table 4 shows spectral test results n_s for such cases. These results can be verified theoretically: For these lags the order of $c := a^l \pmod{m}$ in the multiplicative group $\mathbb{Z}_m^*$ equals $ord(c) = i$. Hence, for prime numbers i we get $c^i - 1 \equiv 0 \pmod{m}$, which implies $1 + c + c^2 + \ldots + c^{i-1} \equiv 0 \pmod{m}$, and therefore the low quality spectral test $d_i = 1/\sqrt{i}$. Now consider non-prime divisors i:

(1) Let $i = 6$. Thus we get

$$c^6 - 1 \equiv (c^3 - 1)(c^3 + 1) \equiv (c^2 - 1)(c^4 + c^2 + 1) \equiv 0 \text{ (mod m)}.$$

[6] The term long-range correlations may be a bit misleading since it also appears in the theory of stochastic processes. In our context it refers to a geometric property of linear random number generators.

Table 4. Long-range correlations n_s among large consecutive blocks from $G1$

$i\backslash s$	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	1														
3		2													
6		2	1	1	1										
7						6									
9						2	2	2							
11										10					
14						6	1	1	1	1	1	1	1		
18						2	2	2	1	1	1	1	1	1	1
31		8	8	8	8	8	8	8	4	4	2	2	2	2	2

Let $4 \le s \le 6$. From $(1, c, c^2, c^3, c^4, c^5) \cdot (1, 0, 0, 1, 0, 0) \equiv c^3 + 1 \equiv 0 \pmod{\mathrm{m}}$ we obtain $d_s = 1/\sqrt{2}$. For dimension $s = 3$ we use $1 + c^2 + c^4 \equiv 0 \pmod{\mathrm{m}}$ and $c \equiv -c^4$ (mod m), and get $d_3 = 1/\sqrt{3}$.

(2) For $i = 9$ we similarly use $c^9 - 1 \equiv (c^3 - 1) \cdot (c^6 + c^3 + 1) \equiv 0 \pmod{\mathrm{m}}$ which yields $d_s = 1/\sqrt{3}$ for $7 \le s \le 9$.

(3) Let $i = 14$. In the same way we use $c^{14} - 1 \equiv (c^7 - 1) \cdot (c^7 + 1) \equiv (c^2 - 1) \cdot (1 + c^2 + c^4 + c^6 + c^8 + c^{10} + c^{12}) \equiv 0 \pmod{m}$, and obtain $d_s = 1/\sqrt{2}$ for dimensions $8 \le s \le 14$. For dimension $s = 7$ we get $d_7 = 1/\sqrt{7}$ since $(1, c, c^2, c^3, c^4, c^5, c^6) \equiv (1, -c^8, c^2, -c^{10}, c^4, -c^{12}, c^6) \pmod{m}$.

(4) Similar calculations yield the entries for $i = 18$. Note, that the latter calculations are valid for arbitrary primitive roots a modulo m, and hence for all full period multiplicative LCGs with prime moduli.

(5) For $i = 31$ the situation slightly changes. In this case, the (bad) spectral test values in Table 4 depend on the specific multiplier $a = 16807$ and on the modulus $m = 2^{31} - 1$. There are exactly 31 solutions of the equation $x^{31} - 1 \equiv 0$ in $\mathbb{Z}_m^*$. These solutions are $x = 2^\alpha$, $\alpha = 0, \ldots 30$. The number $c = a^{(m-1)/31}$ is also a solution of $x^{31} - 1 \equiv 0$. For our example we get $c \equiv 2^{14} \pmod{m}$ and therefore $8c^2 - 1 \equiv 0 \pmod{\mathrm{m}}$, which yields the bad spectral tests in dimensions $s \ge 3$.

4. BLPs: Results of a Computer Search

For several applications of random numbers it is common practice to split the output of a PRNG into interleaved subsequences (this procedure is also denoted "leap-frog" technique). In the case of multiplicative LCGs with modulus m and multiplier a, leaped substreams can be initialized as LCGs by a simple parameter substitution [38], [57], i.e., for leap factor l we obtain a generator with modulus m and multiplier $c := a^l \pmod{m}$, for details see [16], [40]. Therefore, the spectral test for $L_s(c, m)$ can be used as a quality measure for leaped sequences, since all overlapping vectors from a single substream are contained in $L_s(c, m)$. This setting is related to the situation in the previous section when considering small lags l instead of large ones.

In the case of LCGs it easily may happen that the distribution quality of these subsequences is very bad [15], [40]. Several well known LCGs have been revealed to exhibit low quality substreams due to the corresponding bad spectral test results [15], [16]. Since the quality of LCGs in general depends in a very sensitive way on the parameters, choosing a "good" initial generator might not necessarily assure good quality of the leaped substreams. Below, we give examples of such "good" initial generators resulting in extremely low quality leaped substreams as a further class of BLPs. Additionally we consider "bad" initial generators and compare the results of these two cases.

First we conduct a computer search for BLPs of the first type: The normalized spectral test values S_s of an initial prime LCG in dimensions $s = 2, \dots, 8$ are required to exceed threshold t (e.g., $S_s \geq t$ for $s = 2, \dots, 8$). This is a very common criterion often used in literature when searching for good lattice points (however, the results in the Introduction would suggest to use dimension dependent thresholds as well in this case since the fixed threshold does hardly demand any quality restrictions in high dimensions). We randomly generate 650000 multipliers a satisfying this requirement (where each a needs to be a primitive root additionally). This assures initial generators with adjustable quality. Subsequently, the quality of certain subsequences of these generators with leap factor l is evaluated by applying the spectral test in dimensions $s = 2, \dots, 8, 16, 24$ (where $1 \leq l \leq 16$, and $l \in \{2^j \,:\, 5 \leq j \leq 9\}$). In Tables 6 and 7 given below, each entry corresponds to the amount of bad lattice points found in per mille (‰) for a specific dimension s and leap factor l. Table 5 shows the thresholds t_i which are used to rate bad lattice points in this case. These thresholds have been selected to obtain 1.4‰ bad lattice points in each dimension (the values have been found again by averaging over a wide range of moduli).

Table 5. Thresholds used to identify extremely poor quality "bad" generators

t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_{16}	t_{24}
0.036	0.084	0.134	0.185	0.227	0.264	0.292	0.455	0.530

Table 6. Results for modulus $m = 2^{61} - 1$ and "good" initial generators: amount of bad lattice points in ‰

l	$s = 2$	3	4	5	6	7	8	16	24
1	0	0	0	0	0	0	0	2.52	2.21
2	1.42	1.44	1.49	1.51	1.57	1.68	1.70	2.53	2.27
3	1.49	1.39	1.48	1.61	1.68	1.63	1.64	2.40	2.23
4	1.51	1.41	1.49	1.49	1.61	1.59	1.58	2.40	2.28
5	1.37	1.42	1.39	1.46	1.58	1.64	1.63	2.54	2.15
6	1.41	1.39	1.49	1.50	1.64	1.68	1.67	2.52	2.33
7	1.45	1.40	1.45	1.58	1.53	1.72	1.61	2.49	2.27
2^3	1.43	1.38	1.55	1.56	1.59	1.71	1.60	2.53	2.23
2^4	1.44	1.45	1.4	1.46	1.68	1.74	1.71	2.36	2.19
2^5	1.42	1.52	1.49	1.49	1.58	1.55	1.55	2.52	2.23
2^6	1.31	1.43	1.54	1.52	1.54	1.53	1.75	2.51	2.21
2^7	1.40	1.40	1.44	1.56	1.65	1.58	1.67	2.49	2.20
2^8	1.43	1.47	1.43	1.54	1.66	1.66	1.69	2.39	2.20
2^9	1.37	1.48	1.42	1.50	1.66	1.67	1.71	2.46	2.23

In order to facilitate the huge amount of computations, LLL [4] is used as an approximation to the spectral test [18]. Normalization constants for the spectral test for $s > 8$ are taken from [42].

Table 6 displays the results for $m = 2^{61} - 1$ and threshold $t = 0.6$ which are considered as initial generators with good quality. In order to generate 650000 multipliers satisfying this condition, 37604048 primitive roots out of 213323609 possible candidates had to be evaluated (this means that 1.73% of the primitive roots are rated as good). This took about 645 hours on a 1 GHz AMD system. The lowest spectral test value is found for leap factor $l = 32$ and dimension $s = 2$ where the normalized spectral test equals 0.0006 ($a = 2295210864777300077$)! Considering the use of high quality initial generators one might expect a lower amount of bad lattice points as compared to random sampling (i.e., 1.4‰). Obviously this is not at all the case. Hence a good initial generator does not assure good quality of its leaped substreams. In increasing dimensions we see an increase of the amount of bad lattice points. This phenomenon might originate from the fact that the thresholds in Table 5 have been estimated using mean values of percentiles considering a wide range of moduli. The results in Table 6 correspond to a single Mersenne-prime modulus. The threshold estimates may not be optimal for this modul for large dimensions. This is subject to further investigations, the increasing behavior in the tables does not influence the central conclusions of this section. Recall that the zero entries in the table for $l = 1$ and $s = 2, \ldots, 8$ originate from our condition to rate an initial generator as being of high quality iff $S_s \geq 0.6 \; \forall s, s = 2, \ldots, 8$. Initial generators of that type can of course never satisfy the condition to be rated as bad lattice point with respect to the thresholds t_i given in Table 5.

Table 7 displays the results for $m = 2^{61} - 1$ and poor quality initial generators. In particular, for being rated as poor quality initial generator, a multiplier need to be a bad lattice point with respect to the thresholds in Table 5 for dimensions $s = 2, \ldots, 8$. Assuming independence of the involved random variables, an overall amount of 1% poor quality generators is expected. In order to generate 650000

Table 7. Results for modulus $m = 2^{61} - 1$ and "bad" initial generators: amount of BLPs in ‰

l	$s = 2$	3	4	5	6	7	8	16	24
1	131.43	133.84	136.89	144.76	151.03	154.89	156.82	2.48	2.23
2	1.36	1.40	1.42	1.53	1.59	1.62	1.66	2.52	2.33
3	1.4	1.48	1.41	1.55	1.68	1.66	1.71	2.53	2.12
4	1.43	1.49	1.42	1.60	1.59	1.64	1.60	2.47	2.11
5	1.32	1.44	1.45	1.57	1.62	1.57	1.65	2.48	2.18
6	1.43	1.41	1.50	1.51	1.68	1.71	1.67	2.48	2.25
7	1.39	1.46	1.48	1.58	1.58	1.60	1.67	2.36	2.26
2^3	1.41	1.46	1.52	1.57	1.64	1.76	1.65	2.56	2.31
2^4	1.42	1.39	1.49	1.50	1.64	1.70	1.63	2.44	2.21
2^5	1.35	1.48	1.33	1.51	1.60	1.74	1.67	2.63	2.22
2^6	1.39	1.50	1.47	1.55	1.67	1.65	1.72	2.49	2.27
2^7	1.42	1.48	1.40	1.53	1.6	1.61	1.69	2.57	2.33
2^8	1.40	1.47	1.50	1.60	1.62	1.62	1.73	2.42	2.26
2^9	1.48	1.46	1.46	1.64	1.61	1.60	1.66	2.43	2.30

multipliers satisfying these conditions, 60858324 primitive roots out of 345242561 possible candidates had to be evaluated (this means that 1.07% of the primitive roots are rated as bad which supports the above mentioned assumption of independency). This took about 744 hours on a 1 GHz AMD system. One could expect significantly higher values in the table in this case (a higher amount of bad lattice points). Again, this is not true at all. The overall impression is that both tables have entries of roughly the same magnitude, no matter if poor or high quality initial generators have been used. When comparing the tables in more detail we observe that about 41% of the entries of Table 6 are even larger (which means a larger amount of bad lattice points) as compared to the corresponding entries of Table 7. This means that the quality of the initial generator has no impact for the quality of its leaped substreams. Therefore, for applications requiring leaped substreams it makes no sense to test only the initial generators for their quality. All substreams required for an application need to be tested to guarantee sufficient quality. Worst lattice points have to be avoided in any case.

However, in contrast to the results with respect to initial generators with high quality, extremely low spectral test values occur for $l = 1$ and for $l = 2, 3, 4$. As a matter of fact, all these values result from the *small* multiplier $a = 37$ (in Sect. 1, we called lattices from such small multipliers "worst lattice points"). Recall that overlapping vectors generated from an LCG with this multiplier are placed on at most 37 hyperplanes only. Furthermore the numbers $a_l := a^l (\text{mod m}), l = 2, 3, 4$ are in comparison to the magnitude of m small as well ($a_2 = 1369$, $a_3 = 50653$ and $a_4 = 1874161$). Hence, overlapping vectors from subsequences with step size $l = 2, 3, 4$ are also placed on a few hyperplanes in certain dimensions.

Table 8 finally shows results of computer search involving 200000 multipliers a where we used the minimum of an averaged spectral test over dimensions $s = 8, 16, 24$ as a search criterion. We compare high quality initial generators and initial generators without any quality restriction (i.e., 200000 randomly chosen primitive roots). Again there are no significant differences in the results no matter if threshold $t = 0.0$ (no quality restriction) or $t = 0.6$ (high quality initial generators, compare also Table 6) is used.

5. Conclusion

In the present article we introduced and discussed the term "bad lattice points" (BLPs) which should be seen as a counterpart to the method of good lattice points for Monte Carlo and quasi–Monte-Carlo integration. We recommend a definition of BLPs with respect to certain threshold-vectors for normalized spectral tests. We further study different possibilities of BLP occurances in the field of random

Table 8. Average spectral test results for $m = 2^{61} - 1$, $t = 0.0$ and $t = 0.6$

t	$l = 2$	3	4	5	6	7	2^3	2^4	2^5	2^6	2^7	2^8	2^9
0.0	0.43	0.43	0.45	0.45	0.43	0.44	0.45	0.45	0.45	0.44	0.45	0.46	0.45
0.6	0.46	0.45	0.43	0.45	0.43	0.46	0.46	0.44	0.45	0.44	0.46	0.43	0.45

number generation. In this context, we investigate historical and recent BLP implementations in form of linear random number generators, and studied possibilities of BLP appearance as parallel streams of random numbers. A computer search for the amount of BLPs in parallel streams completes the paper.

Acknowledgements

This work was partially supported by the Austrian Science Fund, projects no. 13732, S8303, and S8310. The authors would further like to thank the anonymous referees for their valuable comments.

References

[1] Afflerbach, L., Gruber, G.: Assessment of random number generators in high accuracy. In: New directions in simulation for manufacturing and communications (Morito, S., Sakasegawa, H., Fushimi, M., Nakano, K., eds.), pp. 128–133. OR Society of Japan, 1994.

[2] Anderson, S. L.: Random number generators on vector supercomputers and other advanced architectures. SIAM Rev. *32*, 221–251 (1990).

[3] Boisvert, R. F., McClain, M., Miller, B.: GAMS: The guide to available mathematical software. National Institute of Standards and Technology, Gaithersburg, MD, USA, 1998. http://gams.nist.gov/ (February 2004).

[4] Cohen, H.: A course in computational algebraic number theory. Graduate Texts in Mathematics, vol. 38. Springer 1993.

[5] Couture R., L'Ecuyer, P.: On the lattice structure of certain linear congruential sequences related to AWC/SWB generators. Math. Comp. *62*, 799–808 (1994).

[6] Couture, R., L'Ecuyer, P.: Linear recurrences with carry as uniform random number generators. In: Proc. 1995 Winter Simulation Conf. (Alexopoulos, C., Goldsman, D., Kang, K., Lilegdon, W.R., eds.), pp. 263–267 (1995).

[7] Coveyou, R. R., MacPherson, R. D.: Fourier analysis of uniform random number generators. J. Assoc. Comput. Mach. *14*, 100–119 (1967).

[8] DeMatteis, A., Eichenauer-Herrmann, J., Grothe, H.: Computation of critical distances within multiplicative congruential pseudorandom number sequences. J. Comp. Appl. Math. *39*, 49–55 (1992).

[9] DeMatteis, A., Pagnutti, S.: Parallelization of random number generators and long-range correlations. Numer. Math. *53*, 595–608 (1988).

[10] DeMatteis, A., Pagnutti, S.: Long-range correlation analysis of the Wichmann-Hill random number generator. Stat. Comput. *3*, 67–70 (1993).

[11] DeMatteis, A., Pagnutti, S.: Controlling correlations in parallel Monte Carlo. Parallel Comput. *21*, 73–84 (1995).

[12] Dieter, U.: How to calculate shortest vectors in a lattice. Math. Comp. *29(131)*, 827–833 (1975).

[13] Eichenauer-Herrmann, J., Grothe, H.: A remark on long-range correlations in multiplicative congruential pseudo random number generators. Numer. Math. *56*, 609–611 (1989).

[14] Eichenauer-Herrmann, J., Grothe, H.: Upper bounds for the beyer ratios of linear congruential generators. J. Comput. Appl. Math. *31(1)*, 73–80 (1990).

[15] Entacher, K.: Bad subsequences of well-known linear congruential pseudorandom number generators. ACM Trans. on Modeling and Computer Simulation *7(1)*, 61–70 (1998).

[16] Entacher, K.: Parallel streams of linear random numbers in the spectral test. ACM Trans. on Modeling and Computer Simulation *9(1)*, 31–44 (1999).

[17] Entacher, K., Hellekalek, P., L'Ecuyer, P.: Quasi–Monte Carlo node sets from linear congruential generators. In: Monte Carlo and quasi–Monte Carlo methods 1998. Berlin: Springer 2000, pp 188–198.

[18] Entacher, K., Schell, Th., Uhl A.: Efficient lattice assessment for LCG and GLP parameter searches. Math. Comp. *71*, 1231–1242 (2002).

[19] Entacher, K., Uhl, A., Wegenkittl, S.: Parallel random number generation: long-range correlations among multiple processors. In: Parallel computation (Zinterhof, P., Vajteršic, M., Uhl, A., eds.). Proc. 4th Int. Conf. of the ACPC (ACPC99), Lecture Notes in Computer Science, pp. 107–116. Springer 1999.

[20] Ferrenberg, A. M., Landau, D. P., Wong, Y. J.: Monte Carlo simulations: hidden errors from "good" random number generators. Phys. Rev. Lett. *69*, 3382–3384, 1992.

[21] Fincke, U., Pohst, M.: Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. Math. Comp. *44*, 463–471, 1985.

[22] Fishman, G. S.: Monte Carlo: Concepts, algorithms, and applications. Springer Series in Operations Research, vol. 1. New York: Springer 1996.

[23] Haber, S.: Parameters for integrating periodic functions of several variables. Math. Comp. *41(163)*, 115–129, 1983.

[24] Hellekalek, P.: Good random number generators are (not so) easy to find. Math. Comp. Simulation *46*, 485–505, 1998.

[25] Hellekalek, P., Larcher, G. (eds.): Random and quasi–random point sets. Lecture Notes in Statistics, vol. 138. Berlin: Springer 1998.

[26] Hellekalek, P., Niederreiter H.: The weighted spectral test: Diaphony. ACM Trans. on Modeling and Computer Simulation *8*, 43–60, 1998.

[27] Hickernell, F. J.: Lattice rules: How well do they measure up? In [25], pp. 109–166.

[28] Honerkamp, J.: Stochastische dynamische Systeme. Weinheim: VCH Verlagsgesellschaft 1990.

[29] Hutchinson, D. W.: A new uniform pseudo-random number generator. Commun. ACM *9(6)*, 432–433, 1966.

[30] Jain, R.: The art of computer systems performance analysis. New York: Wiley 1991.

[31] Jansson, B.: Random number generators. PhD thesis, University of Stockholm, 1966. Victor Pettersons Bokindustri AB (also published by Almqvist and Wiksell).

[32] Kao, C., Wong, J. Y.: Random number generators with long period and sound statistical properties. Comp. Math. Appl. *36(3)*, 113–121, 1998.

[33] Kennedy, W. J., Gentle, J. E.: Statistical computing. New York: Dekker 1980.

[34] Knuth, D. E.: The art of computer programming, vol. 2: Seminumerical algorithms, 2nd ed. Reading, MA: Addison-Wesley 1981.

[35] Korobov, N. M.: Number-theoretic methods in approximate analysis. Moscow: Fizmatgiz 1963 (in Russian).

[36] Law, A. M., Kelton, W. D.: Simulation modeling and analysis, 2nd ed. New York: McGraw-Hill 1991.

[37] L'Ecuyer, P.: Testing random number generators. In: Proc. 1992 Winter Simulation Conf., pp. 305–313. IEEE Press 1992.

[38] L'Ecuyer, P.: Uniform random number generation. Ann. Oper. Res. *53*, 77–120, 1994.

[39] L'Ecuyer, P.: Software for uniform random number generation: Distinguishing the good and the bad. In: Proc. 2001 Winter Simulation Conf. 2001.

[40] L'Ecuyer, P.: Bad lattice structures for vectors of non-successive values produced by some linear recurrences. INFORMS J. Comp. *9*, 57–60, 1997.

[41] L'Ecuyer, P.: Good parameter sets for combined multiple recursive random number generators. Operations Res. *47*, 159–164, 1999.

[42] L'Ecuyer, P.: Tables of linear congruential generators of different sizes and good lattice structure. Math. Comp. *68(225)*, 249–260, 1999.

[43] L'Ecuyer, P., Blouin, F., and Couture, R.: A search for good multiple recursive generators. ACM Trans. on Modeling and Computer Simulation *3*, 87–98, 1993.

[44] L'Ecuyer, P., Couture, R.: An implementation of the lattice and spectral tests for multiple recursive linear random number generators. INFORMS J. Comp. *9*(2), 209–217, (1997).

[45] Lemieux, C., and L'Ecuyer, P.: On selection criteria for lattice rules and other quasi–Monte Carlo point sets. Math. Comp. Simulation *55*, 139–148, 2001.

[46] Lewis, P. A., Goodman, A. S., Miller, J. M.: A pseudo-random number generator for the system/360. IBM Syst. J. *8*, 136–146, 1969.

[47] Marsaglia, G.: The structure of linear congruential sequences. In: Applications of number theory to numerical analysis (Zaremba, S. K., ed.), pp. 248–285. New York: Academic Press 1972.

[48] Marsaglia, G., Narasimhan, B., Zaman A.: A random number generator for PC's. Comp. Phys. Comm. *60*, 345–349, 1990.

[49] Marsaglia, G., Zaman, A.: A new class of random number generators. Ann. Appl. Probability *1*, 462–480, 1991.

[50] Morgan, B. J.T.: Elements of simulation. London New York: Chapman and Hall 1986.

[51] Neave, H. R.: On using the Box-Müller transformation with multiplicative congruential pseudorandom number generators. Appl. Statist. *22*, 92–97, 1973.

[52] Niederreiter, H.: Quasi–Monte Carlo methods and pseudo-random numbers. Bull. Amer. Math. Soc. *84*, 957–1041, 1978.

[53] Niederreiter, H.: Random number generation and quasi–Monte Carlo methods. Philadelphia: SIAM 1992.

[54] Niederreiter, H. et al. (eds.): Monte Carlo and quasi–Monte Carlo methods 1996, 1998, 2000, 2002. Proc. Conf. MCQMC 1996–2002. Springer.

[55] Park, S. K., Miller, K. W.: Random number generators: good ones are hard to find. Comm. ACM *31*, 1192–1201, 1988.

[56] Ripley, B. D.: The lattice structure of pseudo-random number generators. Proc. Roy. Soc. London Ser. A *389*, 197–204, 1983.

[57] Ripley, B. D.: Thoughts on pseudorandom number generators. J. Comput. Appl. Math. *31*, 153–163, 1990.

[58] Ross, S. M.: A course in simulation. New York: Macmillan, 1990.

[59] Rotenberg, A.: A new pseudo-random number generator. J. Assoc. Comp. Mach. *7*, 75–77, 1960.

[60] Sloan, I. H., Joe, S.: Lattice methods for multiple integration. New York: Oxford University Press 1994.

[61] Sloan, I. H., Kachoyan, P. J.: Lattice methods for multiple integration: Theory, error analysis and examples. SIAM J. Numer. Anal. *24*, 116–128, 1987.

[62] Tezuka, S.: Uniform random numbers: Theory and practice. Kluwer Academic Publishers 1995.

[63] Tezuka, S., L'Ecuyer, P., Couture, R.: On add-with-carry and subtract-with-borrow random number generators. ACM Trans. on Modeling and Computer Simulation *3*, 315–331, 1993.

K. Entacher
School of Telecommunications Engineering
University of Applied
Sciences and Technologies
Schillerstr. 30
5020 Salzburg
Austria
e-mail: karl.entacher@fh-sbg.ac.at

T. Schell
Department of Scientific Computing
University of Salzburg
Jakob-Haringer-Str. 2
5020 Salzburg
Austria
e-mail: tschell@cosy.sbg.ac.at

A. Uhl
Department of Scientific Computing
University of Salzburg
Jakob-Haringer-Str. 2
5020 Salzburg
Austria
e-mail: uhl@cosy.sbg.ac.at