

# A Parallel Search for Korobov Lattice Rules

Karl Entacher and Bernhard Hechenleitner\*

**Abstract.** We present results from an extensive parallel search for Korobov lattice rules using the LLL-spectral test with a new normalization strategy. The resulting lattice parameters are distributed via a web-server [9] which provides general information on the spectral test, a database for lattice rule parameters, software for spectral test calculations and related applications, efficient on-line parameter searches, scientists working in the field of MC&QMC Methods, and further links and references.

## 1. Introduction

The method of *good lattice points* (GLP) also called *Korobov lattice rules* is a central technique from the fields of *Monte Carlo* (MC) and *quasi-Monte Carlo* (QMC) methods. Good lattice points are classical node sets for QMC integration, defined by the Russian mathematician Korobov [15, 16, 17]. For  $y \in \mathbb{R}$  let  $\{y\} = y - \lfloor y \rfloor$  be the *fractional part* of  $y$ . Consider a vector  $\vec{a} \in \mathbb{Z}^s$ ,  $s \geq 2$ . A Korobov lattice rule is defined by the set

$$P_m := \{ \vec{x}_n : 0 \leq n < m \}, \quad \text{with} \quad \vec{x}_n := \left\{ \frac{n \cdot \vec{a}}{m} \right\}. \quad (1)$$

In the following we will use the term Korobov lattice rule  $P_m$  only for special vectors  $\vec{a}$  defined by a parameter  $a$  with  $1 < a < m$ ,  $m \in \mathbb{N}$ , and  $\vec{a} := (1, a, a^2, \dots, a^{s-1})$ ,  $s \geq 2$ , see [16]. The set  $P_m$  can be seen as the intersection of the  $s$ -dimensional *unit cube*  $I^s := [0, 1]^s$  with the lattice

$$L_s(a, m) = \left\{ \sum_{i=1}^s k_i \vec{b}_i : \vec{k} \in \mathbb{Z}^s \right\}, \quad (2)$$

with basis  $\vec{b}_1 = (1, a, \dots, a^{s-1})/m$ ,  $\vec{b}_2 = (0, 1, 0, \dots, 0)$ ,  $\dots$ ,  $\vec{b}_s = (0, 0, \dots, 0, 1)$ . The classical application of such Korobov lattice rules is the approximate calculation of integrals over  $I^s$ , by the (quasi-) Monte Carlo quadrature rule

$$\int_{I^s} f(\vec{x}) d\vec{x} \approx \frac{1}{m} \sum_{n=0}^{m-1} f(\vec{x}_n), \quad \vec{x}_n \in P_m. \quad (3)$$

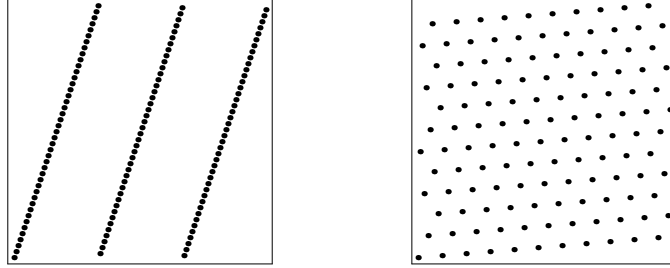
More recent lattice rules, so-called rank- $r$  lattice rules are constructed by modular summation over multiples of different vectors  $\vec{a}_i$ ,  $1 \leq i \leq r$ . Korobov lattice rules are a special case of rank-1 rules. For more details on the theory of integration lattices see [25, 27, 26].

---

\*Fachhochschule Salzburg, Schillerstraße 30, A-5020 Salzburg. Email: {karl.entacher, bernhard.hechenleitner}@fh-sbg.ac.at. Research supported by the Austrian Science Fund (FWF) Grant S8311-MAT.

## 1.1. Lattice Quality

The choice of the parameter  $a$  heavily determines the distribution quality of the lattice. Figure 1 below shows examples of simple lattices  $P_m$  with  $m = 2^7 - 1$ ,  $a = 3$  (left) and  $a = 53$  (right). The central goal for QMC integration is to find lattice parameters  $a$  with optimal distribution behavior in different dimensions. For this task, several equidistribution measures for an assessment of the lattice quality have been constructed, see [25, 27, 2, 11]. On computational selection of good lattice points see [7, 8, 12, 18, 19, 28, 29, 30]. For our purposes we use the *spectral test*, which can be computed very efficiently and provides a reliable measure for lattice assessment [2]. This test has extensively been applied to find good lattices for several MC and QMC applications, e.g. see [1, 4, 6, 13, 20, 21, 22, 24].



**Figure 1.** Lattice rules  $P_m$  with  $m = 2^7 - 1$  and  $a = 3$  (left) and  $a = 53$  (right)

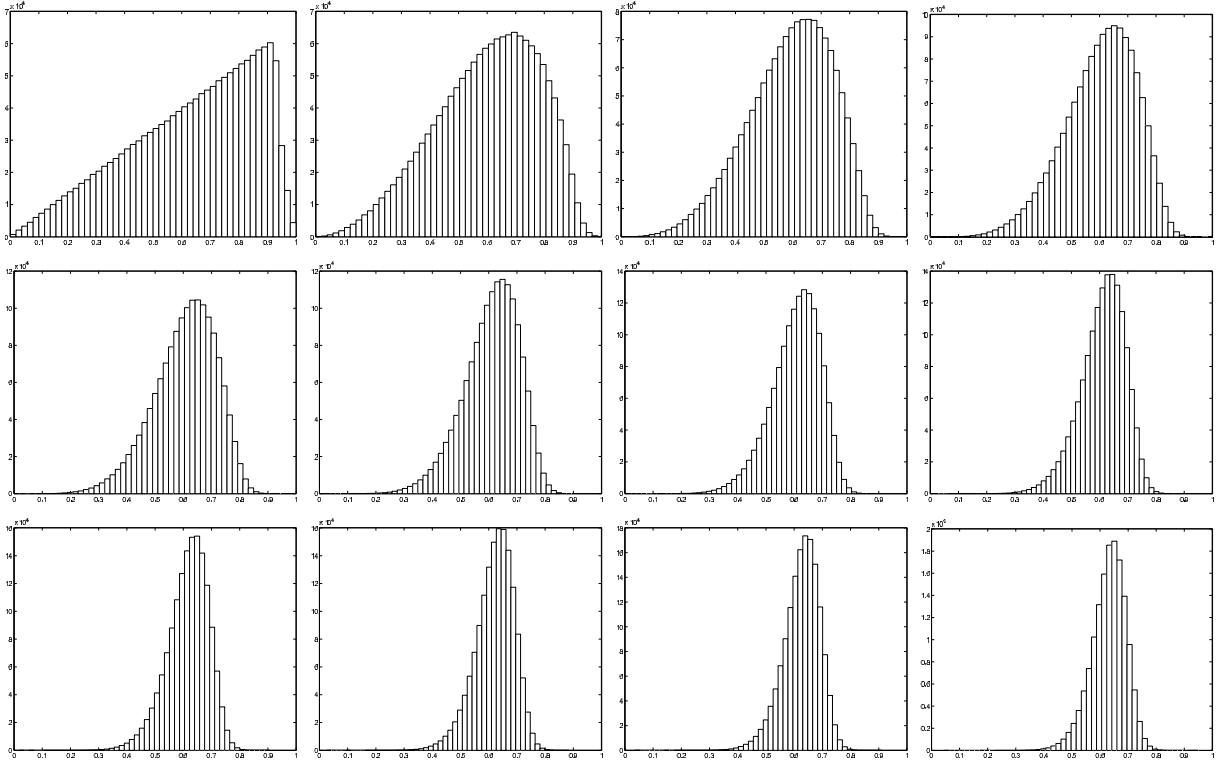
The spectral test uses the dual<sup>1</sup> lattice  $L_s^*(a, m)$  of  $L_s(a, m)$  which for Korobov lattice rules is given by a dual basis  $B^*$  where  $\vec{b}_1^* = (m, 0, \dots, 0)$ ,  $\vec{b}_2^* = (-a, 1, 0, \dots, 0)$ ,  $\dots$ ,  $\vec{b}_s^* = (-a^{s-1}, 0, \dots, 0, 1)$ . From the latter basis the shortest vector  $\vec{v}$  of the dual lattice can be computed by means of the Fincke-Pohst algorithm [5]. 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 [6, 14, 23].

To enable comparison of spectral test results obtained in different dimensions, a normalized spectral test  $S_s := d_s^*/d_s$  for which  $0 \leq S_s \leq 1$  was introduced [6]. Here, high values of  $S_s$  imply good lattice structures. For the constants  $d_s^*$  see [14, p. 105] if  $2 \leq s \leq 8$ , and [21] for arbitrary dimensions. A typical function measuring the “quality” of a lattice parameter  $a$  in terms of the spectral test across dimensions is:

$$M_k := \min_{1 \leq s \leq k} S_s. \quad (4)$$

Fishman, see [6], was one of the first who applied this measure to find optimal parameters  $a$  for  $m = 2^{31} - 1$ ,  $2^{32}$ ,  $2^{48}$  in order to get high quality linear congruential random number generators satisfying a fixed threshold  $M_6 \geq 0.8$ . Recently the measure  $M_k$  is maximized for dimensions up to  $k = 32$  in the context of large scale parameter searches [20, 21, 24].

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



**Figure 2. Distribution behavior of normalized spectral tests  $S_s$  for  $2 \leq s \leq 13$  (from top left to down right)**

## 1.2. A new Normalization Approach to the Spectral Test

One problem with the measure  $M_k$  (4) is that the magnitudes of the single normalized spectral tests  $S_s$  vary significantly for  $1 \leq s \leq k$  as can be seen from Figure 2. The figure shows the distribution behavior of normalized spectral tests  $S_s$  for  $2 \leq s \leq 13$  from top left to down right. For these histograms we used  $m = 2^{64} - 59$  and a random sample of 1300000 parameters  $a$  for each dimension  $s$ . Therefore a fixed threshold to find the best parameter may not be optimal since it is very unlikely to find a parameter  $a$  with  $S_8 > 0.8$  for example. The distributions also indicate that for increasing dimensions a randomly chosen lattice point will behave well with high probability, in analogy to Zinterhof's gratis lattice points approach [29]. The behavior shown in Figure 2 is for Korobov lattices almost independent of the size of the modulus  $m$ , see [3]. In the latter paper we empirically analyzed the distributions and proposed a normalization strategy based on distribution dilation. To cope with empirical outliers we determined estimates for 0.1% and 99.9% quantiles using regression. Normalized spectral test values between the inter-quantile ranges are linearly transformed to  $[0,1]$ , and values outside of the quantiles are mapped to zero and one respectively.

The regression functions for the lower quantile  $L_s$  and the upper quantile  $U_s$ ,  $s = 2, 3, \dots$  are:

$$L_s := 0.000042s^3 - 0.0027s^2 + 0.067s - 0.097$$

$$U_s := -0.000058s^3 + 0.0036s^2 - 0.059s + 1.09$$

Using these functions we linearly transformed  $S'_s := (S_s - L_s)/(U_s - L_s)$  the normalized spectral tests in dimension  $2 \leq s \leq 24$  and used the transformed measure  $S'_s$  for the computer experiments in the following section.

## 2. Parallel Search for GLP

Although the spectral test was chosen as the method for assessing the quality of a lattice, the search for GLPs can become computationally intensive. Therefore, the approach of a parallel application using a cluster was chosen. We restrict our parameter searches to *prime* moduli  $m$  in the range  $2^6 < m < 2^{256}$ . Parameters for "small" moduli, e.g.  $m \leq 2^{31} - 1$  may be applied as lattice rules for QMC-integration, and the parameters for larger  $m$  as multipliers for multiplicative LCGs with prime moduli. Therefore our main task is to find the best GLP or multiplier  $a \in A$  where  $A$  is the set of all primitive roots modulo  $m$  since this restriction provides parameters for multiplicative LCGs with full period [25]. For moduli  $m \leq 2^{31} - 1$  we performed exhaustive searches, i.e. the search space contained all multipliers in  $A$ . For larger moduli only a randomly chosen subset of  $A$  was considered as the search space.

As a search criterion we use the measure  $M_k$  (4). Concerning the normalization method for the spectral test, the common normalization method ("old") with  $S_s$  and our new strategy ("new") using  $S'_s$  in (4) instead of  $S_s$  are distinguished.

The basic sketch of the search method is

- Find a primitive element  $e$  modulo  $m$ . In case  $m$  is a prime number, then  $e$  is a generator of the cyclic group  $\mathbb{Z}_m$  (i.e. the multiplicative order of  $e$  modulo  $m$  is  $\phi(m) = m - 1$  where  $\phi$  is Euler's totient function).
- Take relevant powers  $a = e^\varepsilon \pmod{m}$  as multipliers for which  $\gcd(\varepsilon, \phi(m)) = 1$ . The latter constraint ensures that  $a$  is also a primitive element modulo  $m$ .
- For each  $a$  in the search space and all dimensions up to dimension  $k$ , calculate the LLL reduction, find the spectral test value  $d_s$ , calculate the defined normalization ("old" or "new"), and find the minimum  $M_k$  of the corresponding normalized spectral test values.
- Find the largest  $M_k$  value across all chosen multipliers in  $A$ .

For this purpose, a prototype of a distributed application for the parallel search of GLPs has been developed.

### 2.1. Components and Activities

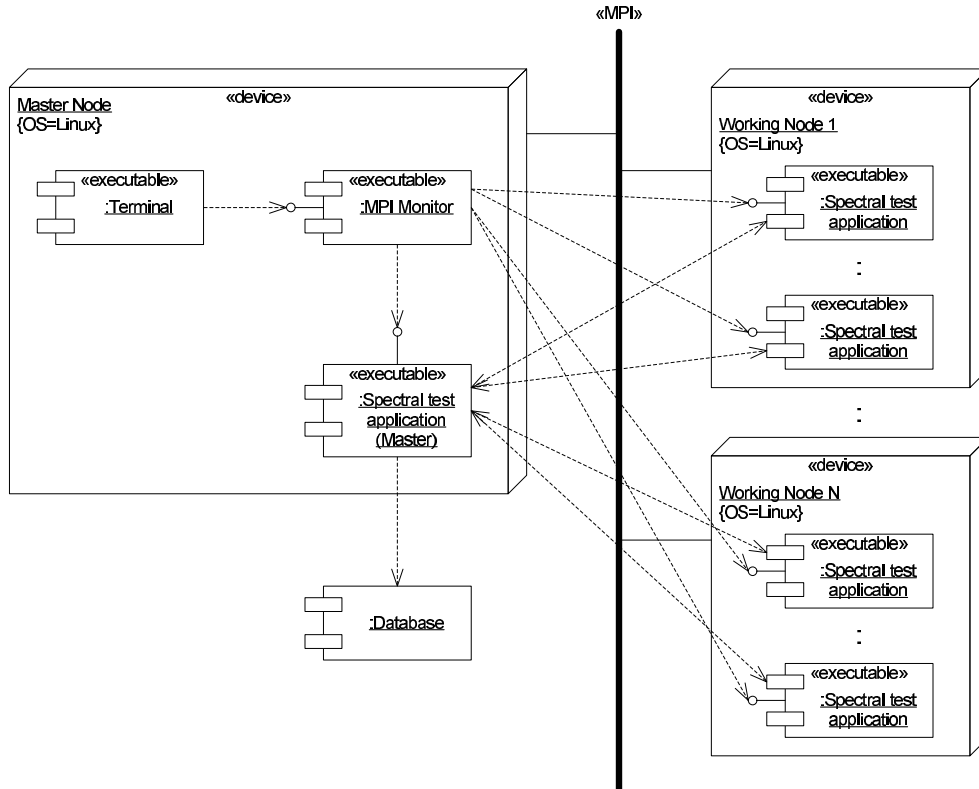
For increasing efficiency, the PC-cluster *Gaisberg* of the High Performance Computing Group<sup>2</sup> of the department of Scientific Computing at the University of Salzburg has been used to conduct parallel searches. The cluster consists of 25 identically equipped nodes as described in Table 1. The dependencies of the software components are shown in the UML component and deployment diagram in Figure 3. The `Master Node` controls the system setup. The distributed `Spectral test` application is started via the `MPI Monitor` application called `mpimon` at the `Master Node`, together with corresponding arguments in the form of command line options. A typical setup is to start two instances of the search application per `Working Node` for maximum efficiency, as each node provides two CPUs. The `Spectral test` application at the `Master Node` determines important parameters for the search and distributes their values to the search processes at the `Working`

---

<sup>2</sup>Home page: <http://hpc.sbg.ac.at> (23.11.2004)

Architecture	PC
CPU	2 AMD Athlon MP 2800+ (2.1 GHz)
Memory	2 GByte
Operating System	Red Hat Linux 7.3
Linux Kernel	2.4.20
Cluster Interconnect	<i>Scalable Coherent Interface (SCI)</i> from Scali
Programming Interface	<i>Message Passing Interface (MPI)</i> from Scali

**Table 1. Cluster nodes**



**Figure 3. Software components and their deployment within the MPI cluster setup**

Nodes utilizing MPI. When a search process finishes its partial search, it passes back its best result – consisting of the best  $M_k$  value, its corresponding multiplier  $a$ , and the number of executed search loops – to the master process, again by utilizing MPI. The master process sorts the received results and prints out the detailed values. Finally, the best result may be stored in a Database.

## 2.2. Search Algorithm

The search algorithm for each search process is as follows. Assume, the modulus  $m$ , the search dimension  $k$ , and the normalization method for the normalized spectral test value *method* have been defined. In the current version of our Spectral test application we applied prime numbers  $m < 2^{256}$  and dimensions  $k \in \{8, 16, 24\}$  and  $method \in \{old, new\}$ .

In a first step, the values for the best global minimum  $M_k$ , denoted as  $M_k^*$ , and the best global multiplier  $a^*$  are initialized to 0. Then the primitive root  $e$  and a starting exponent  $\alpha$ , which have been determined and distributed by the master process, are set. Next, the ending exponent  $\beta$  for  $e$ , which

---

**Algorithm 1. Search Loop**

---

```

WHILE  $\varepsilon < \beta$  DO
  IF  $\gcd(\varepsilon, m - 1) = 1$  THEN
    set multiplier  $a \leftarrow e^\varepsilon \pmod{m}$ ;
    set minimum of normalized spectral test value  $M_k \leftarrow 1$ ;
    FOR  $i \leftarrow 2$  to  $k$  DO
      do LLL reduction;
      find spectral test value  $d_s$ ;
      set  $S_s \leftarrow d_s^*/d_s$ ;
      IF method is new THEN
        set  $S_s \leftarrow (S_s - L_s)/(U_s - L_s)$ ;
      ENDIF
      IF  $S_s < M_k$  THEN
         $M_k \leftarrow S_s$ ;
      ENDIF
    ENDFOR
    IF  $M_k > M_k^*$  THEN
       $M_k^* \leftarrow M_k$ ;
       $a^* \leftarrow a$ ;
    ENDIF
  ENDIF
   $\varepsilon \leftarrow \varepsilon + \gamma$ ;
ENDWHILE

```

---

also represents the loop boundary, is determined and set by each search process. Only those exponents  $\varepsilon$  are relevant for consideration which fulfill the condition  $\gcd(\varepsilon, m - 1) = 1$ . Since  $m$  is a prime number, all even values of  $\varepsilon$  are irrelevant and therefore may additionally be skipped. For this purpose the exponent increment distance  $\delta$  is set to 2 by default. Applying a leapfrog method across the search processes, the exponent increment value  $\gamma$  for each search process is

$$\gamma = \delta \cdot n_p \quad (5)$$

where  $n_p$  is the total number of search processes. The starting exponent  $\varepsilon$  for each search process is defined by its rank<sup>3</sup>  $r_p$ :

$$\varepsilon = \alpha + \delta \cdot (r_p - 1) \quad (6)$$

After initialization of important search parameters, the search process enters the central search loop, which is described in Algorithm 1. As long as the exponent  $\varepsilon$  is smaller than the ending exponent  $\beta$ , in a first step it has to be checked if the new exponent yields another primitive element  $a$  when applied as  $a = e^\varepsilon \pmod{m}$ . If not,  $\varepsilon$  is incremented by  $\gamma$  and the next loop cycle is executed. However, if  $\varepsilon$  yields another primitive element,  $a$  is set as the new multiplier for the modulus  $m$ , and the minimum of the normalized spectral test values  $M_k$  is initialized to the value 1.

For all dimensions  $s$ ,  $2 \leq s \leq k$ , first of all the LLL reduction is calculated and the spectral test value  $d_s$  is determined. Next, the normalized spectral test value with regard to the common normalization  $S_s$  is calculated. If the improved normalization is desired,  $S_s$  is transformed to  $S'_s$  according to the methods of the improved normalization (see Sect. 1.2.). If this normalized or transformed value results in a new minimum,  $M_k$  is reassigned accordingly. Finally, if the specific multiplier  $a$  yields a new

---

<sup>3</sup>The rank of the first search process is 1.

highest global value for  $M_k$ , then this pair of values  $(a, M_k)$  is taken as the new best global pair of values  $(a^*, M_k^*)$ . Before entering the loop again, the value for  $\varepsilon$  is increased by  $\gamma$ .

### 3. Results

Results of parallel searches for GLPs using the prototype described in Section 2. are depicted in the following tables. The set of considered moduli consisted of the largest primes smaller than  $2^l$ , for different values of  $l$ . When several multipliers produced identical results for all parameters, the smallest found was chosen.

#### 3.1. Exhaustive Searches Regarding $M_{16}$ new

Exhaustive searches for  $l \leq 31$  have been conducted for dimension  $k = 16$  with regard to the improved normalization. The results are shown in Table 2.

$m$	$a$	$M_8$ new $M_8$ old	$M_{16}$ new $M_{16}$ old	$M_{24}$ new $M_{24}$ old
$2^7 - 1$	<b>12</b>	0.676401 0.644176	<b>0.676401</b> 0.644176	0.428414 0.644176
$2^8 - 5$	<b>97</b>	0.689814 0.658919	<b>0.689814</b> 0.658919	0.377866 0.658919
$2^9 - 3$	<b>35</b>	0.741633 0.682022	<b>0.741633</b> 0.682022	0.326905 0.661289
$2^{10} - 3$	<b>65</b>	0.726822 0.690694	<b>0.703401</b> 0.663168	0.534339 0.663168
$2^{11} - 9$	<b>328</b>	0.712951 0.695508	<b>0.712951</b> 0.695508	0.480021 0.695508
$2^{12} - 3$	<b>1495</b>	0.670084 0.64051	<b>0.670084</b> 0.64051	0.185094 0.594527
$2^{13} - 1$	<b>1716</b>	0.683929 0.648543	<b>0.683929</b> 0.648543	0.582373 0.648543
$2^{14} - 3$	<b>1543</b>	0.657914 0.633986	<b>0.657914</b> 0.633986	0.657914 0.633986
$2^{15} - 19$	<b>7912</b>	0.726093 0.673006	<b>0.726093</b> 0.673006	0.726093 0.673006
$2^{16} - 15$	<b>4623</b>	0.662787 0.637396	<b>0.662787</b> 0.637396	0.662787 0.637396
$2^{17} - 1$	<b>124189</b>	0.721084 0.668376	<b>0.713932</b> 0.668376	0.368518 0.647801
$2^{18} - 5$	<b>195669</b>	0.708398 0.691738	<b>0.708398</b> 0.691738	0.480731 0.691738
$2^{19} - 1$	<b>157781</b>	0.740903 0.695729	<b>0.718146</b> 0.674205	0.572488 0.674205
$2^{20} - 3$	<b>515951</b>	0.711515 0.664548	<b>0.702975</b> 0.662974	0.51697 0.662974
$2^{21} - 9$	<b>1043187</b>	0.728307 0.686079	<b>0.728307</b> 0.686079	0.586649 0.686079
$2^{22} - 3$	<b>1766427</b>	0.707595 0.662352	<b>0.707595</b> 0.662352	0.600572 0.662352
$2^{23} - 15$	<b>1874695</b>	0.721676 0.674617	<b>0.721676</b> 0.674617	0.622978 0.674617

(continued on next page)

**Table 2: Best multipliers  $a$  with regard to  $M_{16}$  new (exhaustive search)**

$m$	$a$	$M_8$ new $M_8$ old	$M_{16}$ new $M_{16}$ old	$M_{24}$ new $M_{24}$ old
$2^{24} - 3$	<b>7486275</b>	0.718868 0.667317	<b>0.718868</b> 0.667317	0.454087 0.631279
$2^{25} - 39$	<b>6332596</b>	0.731324 0.676041	<b>0.731324</b> 0.676041	0.457129 0.638841
$2^{26} - 5$	<b>52018955</b>	0.752829 0.685247	<b>0.752499</b> 0.685247	0.351023 0.605083
$2^{27} - 39$	<b>33298047</b>	0.741795 0.682116	<b>0.732958</b> 0.682116	0.719319 0.682116
$2^{28} - 57$	<b>157619099</b>	0.730058 0.673146	<b>0.730058</b> 0.673146	0.36788 0.633818
$2^{29} - 3$	<b>237848403</b>	0.757498 0.718127	<b>0.748349</b> 0.694721	0.565678 0.694721
$2^{30} - 35$	<b>283615328</b>	0.738809 0.680384	<b>0.726937</b> 0.680384	0.586649 0.680384
$2^{31} - 1$	<b>1624371841</b>	0.745905 0.681567	<b>0.740411</b> 0.679026	0.562823 0.671093

Table 2: Best multipliers  $a$  with regard to  $M_{16}$  new (exhaustive search)

### 3.2. Random Searches Regarding $M_{16}$ new

For selected values of  $l \geq 32$  random searches with 5 million search loops for each modulus have been conducted for dimension  $k = 16$  with regard to the improved normalization. The results are shown in Table 3.

$m$	$a$	$M_8$ new $M_8$ old	$M_{16}$ new $M_{16}$ old	$M_{24}$ new $M_{24}$ old
$2^{32} - 5$	<b>1293990095</b>	0.69116 0.655171	<b>0.690461</b> 0.655171	0.342604 0.64034
$2^{33} - 9$	<b>2158789984</b>	0.70189 0.660154	<b>0.70189</b> 0.660154	0.590126 0.660154
$2^{34} - 41$	<b>3776435258</b>	0.736812 0.684262	<b>0.710422</b> 0.667627	0.337038 0.610099
$2^{35} - 31$	<b>7181326119</b>	0.680919 0.646797	<b>0.680919</b> 0.646797	0.355042 0.629019
$2^{36} - 5$	<b>38891637263</b>	0.706486 0.664937	<b>0.706486</b> 0.664937	0.435935 0.647801
$2^{37} - 25$	<b>69039132776</b>	0.679882 0.646199	<b>0.679882</b> 0.646199	0.239454 0.616925
$2^{38} - 45$	<b>21031018811</b>	0.704478 0.669976	<b>0.704478</b> 0.669976	0.250654 0.568228
$2^{39} - 7$	<b>269619482669</b>	0.677809 0.644993	<b>0.672613</b> 0.644993	0.0425622 0.512227
$2^{40} - 87$	<b>69467474331</b>	0.69686 0.656045	<b>0.69686</b> 0.656045	0.69686 0.656045
$2^{41} - 21$	<b>218656971933</b>	0.698059 0.666838	<b>0.698059</b> 0.666838	0.0777805 0.493494
$2^{42} - 11$	<b>229783777925</b>	0.699892 0.660735	<b>0.69345</b> 0.660735	0.458055 0.644385
$2^{43} - 57$	<b>5263137617748</b>	0.692232 0.65336	<b>0.659653</b> 0.649122	0.367466 0.611121

(continued on next page)

Table 3: Best multipliers  $a$  with regard to  $M_{16}$  new (random search)

$m$	$a$	$M_8$ new $M_8$ old	$M_{16}$ new $M_{16}$ old	$M_{24}$ new $M_{24}$ old
$2^{44} - 17$	<b>12638685565254</b>	0.764379 0.695219	<b>0.694132</b> 0.674971	0.511164 0.674971
$2^{45} - 55$	<b>26149719913860</b>	0.692276 0.664496	<b>0.680095</b> 0.651628	0.214835 0.543676
$2^{46} - 21$	<b>5953835945761</b>	0.678082 0.651011	<b>0.668187</b> 0.651011	0.301874 0.641004
$2^{47} - 115$	<b>37640087416336</b>	0.701833 0.661972	<b>0.701833</b> 0.661972	0.398566 0.61095
$2^{48} - 59$	<b>48148998458978</b>	0.703679 0.659127	<b>0.684088</b> 0.654355	0.423169 0.654355
$2^{49} - 81$	<b>431837933206670</b>	0.695962 0.663241	<b>0.680293</b> 0.657839	0.350417 0.657839
$2^{50} - 27$	<b>376529839293650</b>	0.68521 0.649286	<b>0.684673</b> 0.649286	0.420929 0.649286
$2^{51} - 129$	<b>950293112394973</b>	0.684711 0.651062	<b>0.684711</b> 0.651062	0.197563 0.586637
$2^{52} - 47$	<b>4454220042005239</b>	0.666641 0.651115	<b>0.65843</b> 0.639048	0.419686 0.639048
$2^{53} - 111$	<b>1761160892805327</b>	0.696569 0.655348	<b>0.696569</b> 0.655348	0.224716 0.600923
$2^{54} - 33$	<b>1710829619865345</b>	0.661327 0.63543	<b>0.659802</b> 0.63543	0.217848 0.567123
$2^{55} - 55$	<b>22787272923226360</b>	0.673101 0.642876	<b>0.661759</b> 0.642876	0.589824 0.642876
$2^{56} - 5$	<b>61816584622201414</b>	0.695139 0.655047	<b>0.66547</b> 0.655047	0.218367 0.567316
$2^{57} - 13$	<b>3643855382186504</b>	0.657307 0.636062	<b>0.657307</b> 0.636062	0.486574 0.636062
$2^{58} - 27$	<b>227100874476387391</b>	0.703933 0.659262	<b>0.655933</b> 0.647527	0.234654 0.573341
$2^{59} - 55$	<b>391807813020384168</b>	0.655285 0.631924	<b>0.655285</b> 0.631924	0.351225 0.616464
$2^{60} - 93$	<b>76804917236851124</b>	0.675635 0.644223	<b>0.652078</b> 0.644223	0.152143 0.593491
$2^{61} - 1$	<b>1913316880966389191</b>	0.702841 0.676093	<b>0.646707</b> 0.659276	0.193176 0.588867
$2^{62} - 57$	<b>324674048276838515</b>	0.733944 0.702553	<b>0.672593</b> 0.660165	0.237245 0.616073
$2^{63} - 25$	<b>3752596661900542267</b>	0.651694 0.632115	<b>0.651694</b> 0.632115	0.198214 0.537591
$2^{64} - 59$	<b>13710539820831023230</b>	0.682999 0.652028	<b>0.658445</b> 0.648604	0.423735 0.648604
$2^{127} - 1$	<b>46063703935658972935\ 285977143409257637</b>	0.644344 0.627594	<b>0.641023</b> 0.627594	0.2536 0.55787
$2^{128} - 159$	<b>12258788848426212840\ 5468094914299633318</b>	0.65689 0.635768	<b>0.65689</b> 0.635768	0.22094 0.58924
$2^{256} - 189$	<b>11554497500795383204\ 74530780189899572159\ 11177005857939805177\ 817323448152360190</b>	0.68684 0.650178	<b>0.645854</b> 0.631879	0.179022 0.56323

Table 3: Best multipliers  $a$  with regard to  $M_{16}$  new (random search)

## 4. Conclusions

We used the spectral test with a new normalization strategy to perform a large scale parallel parameter search for Korobov lattice rules. A selection of the parameters from this experiment is given in the article. The collection of all results of the conducted searches is also available electronically at the *Spectral Test Server* [9]. This web-based application server offers interactive access to a database, which contains detailed calculation results for many lattice rules, information about scientists working in the field of MC&QMC as well as many publication references. The server further provides the possibility to execute GLP search tasks according to the search algorithm described in Section 2.2. directly via a web-browser in a single-threaded approach. In future work we will extend the parameter searches to rank-r rules and distribute the corresponding results at the Spectral Test Server as well.

## Acknowledgments

The authors would like to kindly thank the *High Performance Computing Group* at the Department of Scientific Computing of the University of Salzburg for support and access to their cluster.

## References

- [1] S.L. Anderson. Random number generators on vector supercomputers and other advanced architectures. *SIAM Rev.*, **32**:221–251, 1990.
- [2] K. Entacher, P. Hellekalek, and P. L’Ecuyer. Quasi-Monte Carlo node sets from linear congruential generators. In H. Niederreiter and J. Spanier, editors, *Monte Carlo and Quasi-Monte Carlo Methods 1998*, pages 188–198. Springer, 2000.
- [3] K. Entacher, G. Laimer, H. Röck, and A. Uhl. Normalization of the Spectral Test in High Dimensions. *Monte Carlo Methods and Applications*, to appear, 2004.
- [4] K. Entacher, Th. Schell, and A. Uhl. Efficient lattice assessment for LCG and GLP parameter searches. *Mathematics of Computation*, **71**(239):1231–1242, 2001.
- [5] U. Fincke and M. Pohst. Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Math. Comp.*, **44**:463–471, 1985.
- [6] G.S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*, volume 1 of *Springer Series in Operations Research*. Springer, New York, 1996.
- [7] A. Genz. Testing Multidimensional Integration Routines. In B. Ford, J.C. Rault, and F. Thomasset, editors, *Tools, methods and languages for scientific and engineering computation*. Elsevier Science Publishers B.V. (North-Holland), 1984.
- [8] S. Haber. Parameters for integrating periodic functions of several variables. *Math. Comp.*, **41**(163):115–129, 1983.
- [9] B. Hechenleitner and K. Entacher. Spectral Test Server, Salzburg University of Applied Sciences and Technologies, Austria. <http://spectral.fh-sbg.ac.at>, 2004.
- [10] P. Hellekalek and G. Larcher (eds.). *Random and Quasi-Random Point Sets*, volume **138** of *Lecture Notes in Statistics*. Springer, Berlin, 1998.
- [11] F.J. Hickernell. Lattice Rules: How Well Do They Measure Up. In [10] pp. 109–166.
- [12] L.K. Hua and Y. Wang. *Applications of Number Theory to Numerical Analysis*. Springer-Verlag, Berlin, 1981.

- [13] C. Kao and J.Y Wong. Random number generators with long period and sound statistical properties. *Computers Math. Applic.*, **36** (3):113–121, 1998.
- [14] D.E. Knuth. *The Art of Computer Programming*, volume 2: Seminumerical Algorithms. Addison-Wesley, Reading, MA, 2nd edition, 1981.
- [15] N.M. Korobov. The approximate calculation of multiple integrals. *Dokl. Akad. Nauk SSSR*, **124**:1207–1210, 1959. (in Russian).
- [16] N.M. Korobov. Properties and calculation of optimal coefficients. *Dokl. Akad. Nauk SSSR*, **132**:1009–1012, 1960. (in Russian).
- [17] N.M. Korobov. *Number-Theoretic Methods in Approximate Analysis*. Fizmatgiz, Moscow, 1963. (in Russian).
- [18] N.M. Korobov. On the computation of optimal coefficients. *Dokl. Akad. Nauk SSSR*, **267**:289–292, 1982. (in Russian).
- [19] A. Lauss, P. Zinterhof, and M. Feldbacher. Parallel implementation of fast algorithms for good lattice points - new and extensive tables of good lattice rules. Technical Report Deliverable D5Z-1a, Research Institute for Software Technology RIST++, University of Salzburg, 1994.
- [20] P. L'Ecuyer. Good Parameter Sets for Combined Multiple Recursive Random Number Generators. *Operations Research*, **47**:159–164, 1999.
- [21] P. L'Ecuyer. Tables of linear congruential generators of different sizes and good lattice structure. *Math. Comp.*, **68**(225):249–260, 1999.
- [22] P. L'Ecuyer, F. Blouin, and R. Couture. A search for good multiple recursive generators. *ACM Transactions on Modeling and Computer Simulation*, **3**:87–98, 1993.
- [23] P. L'Ecuyer and R. Couture. An implementation of the lattice and spectral tests for multiple recursive linear random number generators. *INFORMS Journal on Computing*, **9**(2):209–217, 1997.
- [24] C. Lemieux and P. L'Ecuyer. On selection criteria for lattice rules and other quasi-Monte Carlo point sets. *Mathematics and Computers in Simulation*, **55**:139–148, 2001.
- [25] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, Philadelphia, 1992.
- [26] H. Niederreiter et al. (editors). Monte Carlo and Quasi-Monte Carlo Methods 1996, 1998, 2000, 2002, 2004. The series of proceedings for the conferences MCQMC 1996 - 2004, Springer Verlag.
- [27] I.H. Sloan and S. Joe. *Lattice Methods for Multiple Integration*. Oxford Univ. Press, New York, 1994.
- [28] I.H. Sloan and P.J. Kachoyan. Lattice methods for multiple integration: theory, error analysis and examples. *SIAM J. Numer. Anal.*, **24**:116–128, 1987.
- [29] P. Zinterhof. Gratis Lattice Points for Multidimensional Integration. *Computing*, **38**:347–353, 1987.
- [30] P. Zinterhof and P. Zinterhof jun. Computing Good Lattice Points in parallel. Technical Report ACPC/TR 93-18, Austrian Center of Parallel Computation, 1993.