Sum-discrepancy test on pseudorandom number generators

Makoto Matsumoto[a,*], Takuji Nishimura[b]

[a]*Faculty of Science, Hiroshima University, Hiroshima 739-8526, JAPAN*

[b]*Faculty of Science, Yamagata University, Yamagata 990-8560, JAPAN*

Abstract

We introduce a non-empirical test on pseudorandom number generators (prng), named *sum-discrepancy test*. We compute the distribution of the sum of consecutive m outputs of a prng to be tested, under the assumption that the initial state is uniformly randomly chosen. We measure its discrepancy from the ideal distribution, and then estimate the sample size which is necessary to reject the generator. These tests are effective to detect the structure of the outputs of multiple recursive generators with small coefficients, in particular that of lagged Fibonacci generators such as random() in BSD-C library, as well as add-with-carry and subtract-with-borrow generators like RCARRY. The tests show that these generators will be rejected if the sample size is of order 10^6.

We tailor the test to generators with a discarding procedure, such as ran_array and RANLUX, and exhibit empirical results. It is shown that ran_array with half of the output discarded is rejected if the sample size is of the order of 4×10^{10}. RANLUX with luxury level 1 (i.e. half of the output discarded) is rejected if the sample size is of the order of 2×10^8, and RANLUX with luxury level 2 (i.e. roughly 3/4 is discarded) will be rejected for the sample size of the order of 2.4×10^{18}.

In our previous work, we have dealt with the distribution of the Hamming weight function using discrete Fourier analysis. In this work we replace the Hamming weight with the continuous sum, using a classical Fourier analysis, i.e., Poisson's summation formula and Levy's inversion formula.

Key words: random number generation, statistical test, Fourier transform
PACS: 02.50.Ng, 05.10.Ln

[*] Corresponding author.
Email addresses: m-mat@math.sci.hiroshima-u.ac.jp (Makoto Matsumoto), nisimura@sci.kj.yamagata-u.ac.jp (Takuji Nishimura).
URL: http://www.math.keio.ac.jp/matumoto (Makoto Matsumoto).

1 Discrepancy test

A pseudorandom number generator (prng) generates a sequence of numbers in an interval I, mimicking a sequence of random variables having uniform identical independent distribution (i.i.d.). Usually I is one of: the unit interval $[0, 1)$ with uniform measure, a finite set $\{0, 1, 2, \ldots, M-1\}$, or the two-element field $\mathbb{F}_2 := \{0, 1\}$ when we consider a random bit.

A deterministic prng has a state space S. If we specify an initial state, then it produces a virtually infinite sequence of elements in I. Suppose that m outputs are used for every simulation. Then a prng may be considered as a function

$$G : S \to I^m,$$

where S is the state space (G for Generator). Consider the case where the simulation is iterated several times. We make a further assumption that S is a probability space, and that for every simulation the initial state is independently randomly chosen from S with respect to the probability measure (mostly S is I^n for some n, and the measure is the uniform measure). Usually we initialize the generator only once at the first simulation, so the above assumption is not satisfied. However, for most generators, the state after one simulation has almost no correlation with the initial state, and seems as if it was randomly chosen from S. Thus, the above assumption is a reasonable approximation.

Under these assumptions, the distribution of consecutive m-tuples is given by the m-dimensional random variable G defined over S. We want to see whether this m-dimensional random variable has a distribution close to the ideal one or not. To compare m-dimensional distributions is not very easy, so we make them one-dimensional. We fix a *test function* $t : I^m \to \mathbb{R}$, and study the discrepancy of the distribution of t. The ideal distribution of t, denoted by T, is the distribution of the random variable

$$t : I^m \to \mathbb{R}$$

when I^m is equipped with the uniform measure. This comes from the null hypothesis that the generated sequence is uniform i.i.d. The distribution of t generated by G, denoted by T_G, is the distribution of the random variable

$$t \circ G : S \to I^m \to \mathbb{R}$$

when S is considered as a probability space as assumed.

A *t-discrepancy test*[1] on G measures the discrepancy between T and T_G. We need to do the following. First, we choose a simple meaningful t for which the

[1] The term *discrepancy* is a little confusing with the notion of *low discrepancy*

ideal and actual distributions T and T_G are both explicitly computable. Then, we give a suitable measure on the discrepancy between two distributions. Here we adopt the χ^2-discrepancy defined in the next section.

We may obtain T_G by an exhaustive enumeration over S, if the cardinality of S is small. But for modern generators, the cardinality of S is often greater than 2^{100}, so such a brute-force approach is not possible.

We know two cases where S is large but T_G is computable. One is the case when G is an $\mathbb{F}_2$-linear generator and t is the Hamming weight function, which was treated in the previous paper[11]. The other case is when G is based on a linear recursion with modulo-N-arithmetic and t is the sum function, which we shall treat in this paper.

2 χ^2-discrepancy

Consider a set of events $X := \{0, 1, 2, 3, \ldots, \nu\}$. Let $(p_k)_{k=0,1,\ldots,\nu}$ be a probability distribution on X, i.e.,

$$0 < p_k \leq 1 \text{ and } \sum_{k=0}^{\nu} p_k = 1.$$

Let $(q_k)_{k=0,1,\ldots,\nu}$ be another probability distribution.

Definition 1 *We define the χ^2-discrepancy δ between the two distributions (p_k) and (q_k) by*

$$\delta := \sum_{k=0}^{\nu} (q_k - p_k)^2 / p_k.$$

This value is known as the *noncentrality parameter* ([14]) appearing in the χ^2-test when the null hypothesis is not true.

Suppose that we make a null hypothesis that one trial of a probabilistic event conforms to the distribution p_k, and the different trials are i.i.d. To test this null hypothesis, we perform N trials, and count the number Y_k of occurrences of each event $k \in \{0, 1, \ldots, \nu\}$. *The χ^2-value* $\mathcal{X}$ of this experiment is defined as

$$\mathcal{X} := \sum_{k=0}^{\nu} (Y_k - Np_k)^2 / Np_k.$$

It is known that $\mathcal{X}$ has approximately the *χ^2-distribution with ν degrees of freedom* under the null hypothesis, if Np_k is large enough for each k. Let X be a random variable having the χ^2-distribution with ν degrees of freedom.

sequences, but is standard in the model selection theory, cf. [7].

Recall that the (left) p-value p corresponding to the observed χ^2-value $\mathcal{X}$ is defined by

$$p = \mathrm{Prob}(X < \mathcal{X}).$$

If the p-value is too high like > 0.99, then we reject the null hypothesis with significance level > 0.99.

Suppose that the above null hypothesis is not correct, and the different trials have independent identical distribution $(q_k)_{k=0,\ldots,\nu}$. Then it is known [14] that $\mathcal{X}$ approximately has the *non-central χ^2-distribution with ν degrees of freedom with noncentrality parameter δ.* As a consequence, the expectation of $\mathcal{X}$ is approximated by

$$E(\mathcal{X}) \sim \nu + N\delta$$

(for an elementary proof and the error estimate, see [11]).

We shall define *the risky sample size with significance level* p to be the sample size N such that

$$\mathrm{Prob}(X < \nu + N\delta) = p.$$

Thus, if we apply the χ^2-test with the sample size N, then the average of the observed χ^2-value corresponds to the p-value p. When $p = 0.99$, we call the corresponding N the *risky sample size*, simply, and when $p = 0.75$, call the corresponding N the *safe sample size.* Approximation formulae for these values are given in [11].

Remark 2 The idea of finding general approximation formulae for the sample size where a class of generators start failing a given test was introduced and implemented in [3] and developed in [4].

3 Sum discrepancy test.

Assume that $I = [0, 1)$, and that the test function is the sum:

$$t : I^m \to \mathbb{R}, (w_1, w_2, \ldots, w_m) \mapsto t = \sum_{j=1}^{m} w_j.$$

Then T and T_G in §1 are random variables with values in $[0, m)$. We fix a suitable categorization of this interval $[0, m)$ into $\nu+1$ intervals. We discreticize T and T_G by letting p_k (q_k) be the probability that T (T_G) falls in the k-th interval, respectively. Then we compute the χ^2-discrepancy between the two discrete distributions, and then obtain the safe and risky sample sizes as in §2.

The only non-trivial step is to compute the distribution functions of T and T_G. For general generators this seems intractable, as we have stated at the end of §1, but it is possible for the following class of generators when $m-n$ is small (here n is the dimension of internal state, see below).

Assume that I is $[0,1)$ identified with the 1-dimensional torus $\mathbb{R}/\mathbb{Z}$, i.e., real numbers considered modulo 1 with the usual additive group structure. Assume moreover that $S=I^n$ and G is a continuous group homomorphism $S=I^n\to I^m$. Then, the distribution function F_T of the ideal distribution T is obtained by

$$F_T(\alpha) := \mathrm{Prob}(T<\alpha) = \int\limits_{0\le w_1,\ldots,w_m\le 1, t<\alpha} dw_1 dw_2\cdots dw_m,$$

whereas the distribution function F_{T_G} of T_G is obtained by

$$F_{T_G}(\alpha) := \mathrm{Prob}(T_G<\alpha) = \int\limits_{\{0\le w_1,\ldots,w_m\le 1, t<\alpha\}\cap G(S)} (dw_1 dw_2\cdots dw_m)|_{G(S)},$$

where $(dw_1 dw_2\cdots dw_m)|_{G(S)}$ is the restriction of the uniform measure on I^m to the subgroup $G(S)$ normalized so that $\mathrm{Vol}(G(S))=1$. (For the latter, we used the fact that the above measure on $G(S)$ is the image measure induced from that of S.)

A problem is that the dimension m is often as large as 30, so this integration seems intractable. A possible solution might be Monte Carlo integration, but this is nothing but the test of randomness of the underlying random number generators, and takes a lot of time to obtain the required precision. A more practical solution is to use the characteristic function and Fourier inversion. We recall the definition of characteristic functions, and Levy's inversion formula [1, Sect.26, Theorem 26.2].

Definition 3 *Let $F_T(\alpha)$ be the distribution function of a random variable T. Then, its* characteristic function *is the Fourier transformation*

$$\psi_T(\theta) := \int\limits_{\alpha=-\infty}^{\alpha=\infty} e^{2\pi i\theta\alpha} dF_T(\alpha).$$

Theorem 4 (Levy inversion)

$$\mathrm{Prob}(a\le T\le b) = F_T(b)-F_T(a) = \int\limits_{-\infty}^{\infty} \frac{e^{-2\pi i\theta b}-e^{-2\pi i\theta a}}{-2\pi i\theta}\psi_T(\theta)d\theta.$$

Thus, one can reconstruct the distribution function from its characteristic function. Next, we recall another Fourier inversion formula, known as Poisson's summation formula. Define the orthogonal group pairing

$$\begin{aligned} (\mathbb{R}/\mathbb{Z})^m \times \mathbb{Z}^m &\to \mathbb{C} \\ \mathbf{w} \;\;,\;\; \mathbf{n} \;\; &\mapsto e(\mathbf{w}|\mathbf{n}) := e^{2\pi i \sum_j w_j n_j}. \end{aligned}$$

For

$$f : (\mathbb{R}/\mathbb{Z})^m \to \mathbb{C},$$

we define its Fourier transformation as usual:

$$\begin{aligned} &\hat{f} : \mathbb{Z}^m \to \mathbb{C}, \\ &\hat{f}(\mathbf{n}) := \int\limits_{\mathbf{w}\in(\mathbb{R}/\mathbb{Z})^m} f(\mathbf{w}) e(\mathbf{w}|\mathbf{n}) d\mathbf{w}. \end{aligned}$$

Let $C \subset (\mathbb{R}/\mathbb{Z})^m$ be a Lie subgroup, and $C^\perp \subset \mathbb{Z}^m$ be its orthogonal compliment.

Theorem 5 (Poisson's summation formula) *It holds that*

$$\int\limits_{\mathbf{w}\in C} f(\mathbf{w}) d\mathbf{w}|_C = \sum_{\mathbf{n}\in C^\perp} \hat{f}(\mathbf{n})$$

if the right hand side is absolutely convergent.

For a proof, see [12, Theorem 5.5.2]. Use Remark 5.5.11 there for $G = \mathbb{Z}^m$. As an application of this theorem, put $C = G(S)$, $f(\mathbf{w}) = e^{2\pi i\theta \sum_j w_j}$. Then

$$\int\limits_{\mathbf{w}\in C} f(\mathbf{w}) d\mathbf{w}|_C = \int\limits_{-\infty}^{\infty} \left(\int\limits_{\{\mathbf{w}\in C\}\cap \sum_j w_j = \alpha} e^{2\pi i\theta\alpha} d\mathbf{w}' \right) d\alpha$$

$$= \int\limits_{-\infty}^{\infty} e^{2\pi i\theta\alpha} \left(\int\limits_{\{\mathbf{w}\in C\}\cap \sum_j w_j = \alpha} d\mathbf{w}' \right) d\alpha = \int\limits_{-\infty}^{\infty} e^{2\pi i\theta\alpha} dF_{T_G}(\alpha) = \psi_{T_G}(\theta).$$

Here, the restricted measure is the product of two measures $d\mathbf{w}|_C = d\mathbf{w}' d\alpha$, where $d\mathbf{w}'$ is the measure on the cut $C \cap \{\mathbf{w}| \sum_j w_j = \alpha\}$ for each fixed α, and $d\alpha$ is the standard measure of real line. The first equality follows from Fubini's Theorem.

Since

$$\hat{f}(\mathbf{n}) = \left(\frac{e^{2\pi i\theta} - 1}{2\pi i\theta}\right)^m \prod_{j=1}^m \frac{\theta}{\theta + n_j},$$

we have an explicit formula:

$$F_{T_G}(b) - F_{T_G}(a) = \sum_{\mathbf{n} \in G(S)^\perp} \int_{-\infty}^{\infty} \frac{e^{-2\pi i\theta b} - e^{-2\pi i\theta a}}{-2\pi i\theta} \left(\frac{e^{2\pi i\theta} - 1}{2\pi i\theta} \right)^m \prod_{j=1}^{m} \frac{\theta}{\theta + n_j} d\theta. \qquad (1)$$

Note that the pole $\frac{1}{\theta+n_j}$ is absorbed in $(e^{2\pi i\theta}-1)$. If $m \geq 2$ then the right hand side is absolutely convergent, and this gives F_{T_G}. As in the $\mathbb{F}_2$-case (see [11]), we choose m so that the rank of $G(S)^\perp$ is small. A difference from the $\mathbb{F}_2$-case is that we approximate $\sum_{\mathbf{n} \in G(S)^\perp}$ by a finite sum. Experimentally, summands decrease rapidly for $\mathbf{n} \to \infty$.

Remark 6 If we put $G(S)^\perp = \{\mathbf{0}\}$ in this formula, we obtain the ideal distribution function F_T. It is well known that F_T is approximated by a normal distribution, but in the following experiments, we experienced that the approximation error ruins the test result. We need a more precise value of F_T obtained from this integration.

The formula (1) implies that the deviation from the ideal distribution is given as

$$\int_{-\infty}^{\infty} \frac{e^{-2\pi i\theta b} - e^{-2\pi i\theta a}}{-2\pi i\theta} \left(\frac{e^{2\pi i\theta} - 1}{2\pi i\theta} \right)^m \{ \sum_{\mathbf{n} \in G(S)^\perp - \{\mathbf{0}\}} \prod_{j=1}^{m} \frac{\theta}{\theta + n_j} \} d\theta.$$

Thus, the existence of vectors in $G(S)^\perp$ whose components have small absolute values affects the distribution.

In this point, our test is similar to the spectral test,

in which the length of the shortest vector in the dual lattice L^* gives the criterion on the uniformity of the output. (We refer to [6] for the spectral test, including the definition of L^*.) The larger this length, the more uniformly the points are distributed in the m-dimensional cube I^m. It is easy to show that $L^* \supset G(S)^\perp$, but L^* is m-dimensional and the dimension of $G(S)^\perp$ may be much smaller. The existence of a short vector in $G(S)^\perp$ implies that in L^*, which shows that the spectral test detects a finer structure than our test does.

Conceptually, the difference between the two tests is "local versus global." In our test, we consider the distribution of a function $f(w_1, \ldots, w_m)$ $(= \sum_{j=1}^{m} w_j)$, which depends only on each of the m-consecutive outputs, and neglects the relation between them. On the contrary, the spectral test detects the structure of the set of points in the m-dimensional cube, hence considers the relation among the m-tuples.

4 Test Results

4.1 lagged Fibonacci generators

A multiple recursive generator is to generate a sequence $\{x_i\}$ of elements of $\mathbb{Z}/N := \{0, 1, 2, \ldots, N-1\}$ by a linear recursion

$$x_{n+i} = a_{n-1}x_{n-1+i} + a_{n-2}x_{n-2+i} + \cdots + a_1x_{1+i} + a_0x_i, \quad i = 0, 1, 2, \ldots (2)$$

where $a_0, a_1, \ldots, a_{n-1}$ are suitably chosen integers. If we look at the m consecutive outputs, this generator maps the initial seeds $(x_0, x_1, \ldots, x_{n-1})$ to $(\mathbb{Z}/N)^m$.

We identify $(\mathbb{Z}/N)$ with the N-torsion points in $I = \mathbb{R}/\mathbb{Z}$, i.e., rational numbers with denominator dividing N. If N is large, we may approximate $\mathbb{Z}/N$ by I, through the embedding

$$\mathbb{Z}/N \to I, \quad x \mapsto x/N,$$

where x is considered to be a real number in the computation of x/N.

Then $G : S = I^n \to I^m$ given by $(x_0, x_1, \ldots, x_{n-1}) \mapsto (x_0, x_1, \ldots, x_{m-1})$ under the recursion (2) will approximate the multiple recursive generators, under the assumption that $a_i \ll N$ for all i. Thus, we may apply the methodology stated in §3. In the case that $a_0 = \pm 1$ and exactly one of $a_1, a_2, \ldots, a_{n-1}$ is ± 1 and all the rest are zero, the generator is called *a lagged Fibonacci generator*. In this case, the approximation is fairly good.

Essentially same approximation is also valid for add-with-carry or subtract-with-borrow generators like RCARRY introduced in [9]. We refer to Lüscher's work [8] about the approximation of these discrete recursions by continuous recursions, and do not discuss here.

4.2 Results on a lagged Fibonacci generator `random()`*.*

A new standard random number generator `random()` is recommended in the manual on BSD-C. In Linux, the standard `rand()` function is replaced with this. If you type "`man random`" in Unix machine, it will tell "The `random()` function uses a nonlinear additive feedback random-number generator," but actually `random()` is a simple lagged Fibonacci generator, defined by the recursion

$$x_{i+31} = x_{i+28} + x_i \bmod 2^{32} \quad (i = 1, 2, \ldots).$$

We choose $m = 34$, so $G(S)^\perp$ is of rank 3. We categorize $[0, m)$ into 10 intervals, so that each p_k is almost same with each other.

The space $G(S)$ is the solution space

$$w_{i+31} = w_{i+28} + w_i \quad (i = 1, 2, \ldots, 34), \quad w_i \in \mathbb{R}/\mathbb{Z}, \tag{3}$$

and thus a basis $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \in \mathbb{Z}^{34}$ of $G(S)^\perp$ is given by

$$\begin{aligned}\mathbf{b}_1 &= (1, 0, 0, 0, \ldots, 0, 1, 0, 0, -1, 0, 0),\\ \mathbf{b}_2 &= (0, 1, 0, 0, \ldots, 0, 0, 1, 0, 0, -1, 0),\\ \mathbf{b}_3 &= (0, 0, 1, 0, \ldots, 0, 0, 0, 1, 0, 0, -1).\end{aligned}$$

It is easy to show that these vectors are shortest in $G(S)^\perp$. Consider the set

$$B_s := \{n_1\mathbf{b}_1 + n_2\mathbf{b}_2 + n_3\mathbf{b}_3 \mid |n_1| + |n_2| + |n_3| \le s\},$$

and let δ_s be the approximation of χ^2-distance obtained by replacing the infinite sum $\sum_{\mathbf{n}\in G(S)^\perp}$ in (1) with the finite sum $\sum_{\mathbf{n}\in B_s}$.

With the help of Mathematica, we obtain $\delta_1 = 1.37601 \times 10^{-6}$, $\delta_2 = 1.55475 \times 10^{-6}$, $\delta_3 = 1.59015 \times 10^{-6}$, $\delta_4 = 1.60127 \times 10^{-6}$, and $\delta_5 = 1.60581 \times 10^{-6}$. We did not estimate the error between δ_s and δ, but adopted δ_2 as an approximation of δ here and in the following tests (note that B_2 consists of 24 vectors). The choice $s = 2$ is not the best, but makes the tests faster and easier. The cardinality of $B_1, B_2, \ldots, B_5$ is 6,24,63,128,230, respectively.

Computation using δ_2 gives a safe sample size of the order of 1.6×10^6, and a risky sample size of the order of 8.3×10^6. The next table shows a confirmation of these forecast by five independent replications of the corresponding empirical χ^2-tests with sample sizes near the safe/risky ones.

N	1st	2nd	3rd	4th	5th
1.6×10^6	73.1%	77.9%	88.5%	75.3%	99.8%
8.3×10^6	100%	99.5%	69.5%	98.8%	99.1%

Table 1: `random()`: χ^2-Tests for five different initial values

4.3 Test on RCARRY

RCARRY, one of subtract-with-borrow generators proposed by Marsaglia and Zaman[9], can be tested in the same scheme. RCARRY generates an integer sequence x_j by the recursion

$$\begin{aligned}x_{j+24} &:= x_{j+14} - x_j - C_j \bmod 2^{24},\\ C_{j+1} &:= [x_{j+14} < x_j + C_j],\end{aligned}$$

where $C_j = 0, 1$ and $[\quad]$ is the indicator function, i.e., it is 1 or 0 according to the predicate inside is true or not. For the distribution of m-tuples with small m (e.g. < 1000), we may neglect C_j (see [2] and [13]), so this is approximated by the lagged Fibonacci generator with parameters $x_{j+24} := x_{j+14} - x_j \bmod 2^{24}$.

We choose $m = 27$. Then $G(S)$ is 24-dimensional and $G(S)^{\perp}$ is a 3-dimensional lattice. Similarly to the previous case, the infinite sum $\sum_{\mathbf{n}\in G(S)^{\perp}}$, is approximated by the sum over B_2. As a result, we have δ of the order of 4.0×10^{-6}, a safe sample size of the order of $620,000$, and a risky sample size $\sim 3,200,000$.

N	1st	2nd	3rd	4th	5th
$630,000$	20.5%	$\underline{100.0}\%$	40.8%	14.3%	94.3%
$3,300,000$	$\underline{99.6\%}$	$\underline{100.0\%}$	$\underline{99.0\%}$	$\underline{98.9\%}$	89.1%

Table 2: RCARRY: χ^2-Tests for five different initial values.

5 Generators with discarding procedures

Lüscher[8] proposed an improvement of low-quality pseudorandom number generators, by discarding a part of the output sequence. For simplicity, we assume that the original generator is given by (2). We fix an integer $p \geq n$ called the *luxury index*. We generate p pseudorandom real numbers using the recursion (2), use the first n, discard the next $p - n$, then generate p, use n, discard the next $p - n$, and iterate this procedure. If $p = n$, then the output is identical with the original one.

We shall design a model for such generators to apply the sum-discrepancy test. Let

$$G : I^n \to I^M, \quad b \mapsto G(b) = (G(b)_1, \ldots, G(b)_M)$$

be the original generator, where M is chosen to be $M = m + p - n$. Then, the improved generator G' with luxury index p is a function

$$G' : I^n \times \{0, 1, 2, \ldots, n-1\} \to I^m.$$

obtained from G as follows. The state space is $S' = I^n \times \{0, 1, 2, \ldots, n-1\}$. Here, the first I^n shows the state of the original generator, and the second

$\{0, 1, 2, \ldots, n-1\}$ shows the position of the present state with respect to the discarding procedure. Namely, the state $(b, j) \in S'$ shows that, in the generator G', the original generator has the state b and we are going to use the next $n-j$ outputs and then discard $p-n$ after.

At first we have $j = 0$, and using the recursion (2) we generate one pseudorandom number, and output it. Then we increment j to 1, which is a counter showing that after $n-j$ generations, we need to discard $p-n$ numbers. Again generate one number, output it, and increment j to 2. Iterate this until $j = n$. When $j = n$ holds, we have used the n consecutive output of the original generator, and hence we discard $p-n$ numbers from G, set $j = 0$, and return to the first step.

Thus, explicitly we have

$$G'(b, j) = (G(b)_1, G(b)_2, \ldots, G(b)_{n-j}, G(b)_{p-j+1}, G(b)_{p-j+2}, \ldots, G(b)_{p+m-n}).$$

From this we see that, for fixed j, $G'(b, j)$ is a linear function $I^n \rightarrow I^m$ and hence the technique of Fourier analysis is applicable.

Let $T_{G'}$ denote the random variable obtained as the summation of the m components in $G'(b, j)$ under the condition that both $b \in S$ and $j \in \{0, 1, 2, \ldots, n-1\}$ are uniformly distributed. Let $T_{G'(-,j)}$ denote the random variable obtained as the summation of the m components in $G'(b, j)$ under the condition that j is fixed and b is uniformly distributed in S. Then, we have

$$F_{T_{G'}}(\alpha) = \frac{1}{n} \sum_{j=0}^{n-1} F_{T_{G'(-,j)}}(\alpha).$$

Once $F_{T_{G'}}$ is obtained, the rest of the sum-discrepancy test goes in the same way, and gives the safe and risky sample sizes. To compute $F_{T_{G'(-,j)}}(\alpha)$, we need to compute a basis of $G'(S, j)^\perp$. For simplicity, we explain in the case of $j = 0$, since other cases are similar. We have

$$G'(b, 0) = (G(b)_1, G(b)_2, \ldots, G(b)_n, G(b)_{p+1}, G(b)_{p+2}, \ldots, G(b)_{p+m-n}).$$

This is the projection of $G(S) \subset (\mathbb{R}/\mathbb{Z})^M$ to $(\mathbb{R}/\mathbb{Z})^J$, where J denotes the set $\{1, 2, \ldots, n, p+1, p+2, \ldots, p+m-n\}$. By duality, $G'(b, 0)^\perp = G(S)^\perp \cap \mathbb{Z}^J \subset \mathbb{Z}^M$. A basis $\{b_1, b_2, \ldots, b_{M-n}\}$ of $G(S)^\perp$ is obtained in the previous way. Then a basis of $G(S)^\perp \cap \mathbb{Z}^J$ is obtained by the integer-version of Gaussian elimination, i.e. it is the basis of the solution of the system of equations

$$\mathrm{proj}_i(\sum_{j=1}^{M} n_j \mathbf{b}_j) = 0 \quad (\forall i \notin J),$$

where proj_i denotes the i-th component.

RANLUX is a pseudorandom number generator proposed by Lüscher[8] and implemented in Fortran by James[5]. RANLUX generator is obtained from RCARRY (see §4.3) by a discarding procedure described above, with $n = 24$ and $p \geq 24$. In [5], five standard luxury indices $p = 24, 48, 97, 223, 389$ are chosen, and each of them is called *RANLUX with luxury level* $0, 1, 2, 3, 4$, respectively. The level 3, i.e., $p = 223$ is defined as the default level. Let G be the RCARRY generator. As formulated in §5, G is a mapping from I^{24} to I^M, where $M = m + p - n$. Let G' be RANLUX generator. We consider G' as a function

$$G' : I^{24} \times \{0, 1, \dots, 23\} \to I^m.$$

We choose $m = 27$ and the luxury $p = 48$, i.e., what James called luxury level 1 . In this case, $G'(S, j)$ is 24-dimensional and its orthogonal complement $G'(S, j)^{\perp}$ is a 3-dimensional lattice. We consider the distribution of the sum of consecutive 27 outputs. We categorize the range $[0, 27)$ into 10 intervals, so that the probability that the random variable T obeying the ideal distribution falls into each category is almost the same. In this case, a computation gives a basis of $G'(S, 0)^{\perp}$:

$$\mathbf{b}_1 = (-1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, -1, 0, 0, 0, 0, 0, 1, 0, 0),$$
$$\mathbf{b}_2 = (0, -1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, -1, 0, 0, 0, 0, 0, 1, 0),$$
$$\mathbf{b}_3 = (0, 0, -1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, -1, 0, 0, 0, 0, 0, 1).$$

A hand computation shows that these are the shortest. We define B_2 as in §4, and take the sum over the 24 vectors in $G'(S, 0)^{\perp}$ as an approximation. We compute the basis of $G'(s, j)^{\perp}$ in the same way for $j = 1, 2, \dots, 24$. With the help of Mathematica, we obtained an approximation of χ^2-discrepancy δ of the order of 6.3×10^{-8}, a safe sample size of the order 3.9×10^7, and a risky sample size of the order 2.0×10^8. We carried out these χ^2-tests five times with different initial values for safe and sample sizes respectively. In Table 3, we list the p-values of these empirical χ^2-tests.

sample size	1st	2nd	3rd	4th	5th
4.0×10^7	39.0%	83.5%	86.3%	54.9%	70.6%
2.0×10^8	98.0%	71.8%	99.8%	91.0%	99.0%

Table 3: RANLUX: χ^2-Tests with $p = 48$, for five different initial values

Next we show the result for luxury $p = 97$, i.e., luxury level 2 according to James. We choose $m = 27$ and categorize the range $[0, 27)$ into 10 intervals as in the previous case. We compute a basis of $G'(S, j)^{\perp}$ in the previous way. For example, in the case $j = 0$, each vector in the basis has 10 nonzero components, with maximum absolute value 10. Again define B_2 as in the previous case, and approximate δ.

We have obtained a χ^2-discrepancy δ of the order 5.4×10^{-18}, a safe sample size of the order 4.6×10^{17}, and a risky sample size of the order 2.4×10^{18}. In this case, both safe and risky sample sizes are too large to carry out the χ^2-tests. For higher luxury levels, we could not compute δ, since it appears to be very small and the approximation error seems to exceed its value.

5.2 Sum-discrepancy test on ran_array

ran_array is a pseudorandom number generator proposed by Knuth[6]. It is an improvement, by discarding strategy, of the following lagged Fibonacci generator:

$$x_{j+100} := -x_{j+63} + x_j \bmod 2^{30} \quad (j = 1, 2, \ldots).$$

For a given luxury index $p \geq 100$, ran_array generates by iterating the following procedure: generate p numbers by the above lagged Fibonacci generator, use the first 100 numbers, discard the remaining $p - 100$ numbers.

We choose $m = 103$ and $p = 100$. In this case, the generator is identical to the original generator. The subgroup $G(S)$ is 100-dimensional and its orthogonal complement $G(S)^\perp$ is a 3-dimensional lattice. We categorize the range $[0, 103)$ into 10 intervals, so that the probability that the random variable T obeying the ideal distribution falls into each category is almost the same. To compute $F_{T_G}(\alpha)$, we used the dominating 24 vectors in $G(S)^\perp$ as in the previous cases. Then, we obtained a χ^2-discrepancy δ of the order of 1.74753×10^{-8}, a safe sample size of the order of 1.43×10^8 and a risky sample size of the order of 7.35×10^8. In Table 4, we show the results of the corresponding χ^2-tests.

sample size	1st	2nd	3rd	4th	5th
1.43×10^8	75.6%	52.4%	13.6%	73.9%	93.9%
7.35×10^8	98.5%	100%	92.1%	94.9%	99.7%

Table 4: ran_array: χ^2-Tests with $p = 100$, for five different initial values

We now study the luxury index to $p = 200$. The other conditions are the same as those of the previous case. We obtain a χ^2-discrepancy δ of the order of 3.07818×10^{-10}, a safe sample size of the order of 8.1×10^9 and a risky sample size of the order of 4.2×10^{10}. In Table 5, we show the results of the corresponding χ^2-tests.

sample size	1st	2nd	3rd	4th	5th
8.1×10^9	61.0%	50.8%	56.8%	30.4%	67.2%
4.2×10^{10}	96.3%	99.8%	95.0%	100%	99.8%

Table 5: ran_array: χ^2-Tests with $p = 200$, for five different initial values

We change the luxury index to $p = 300$. Other conditions are the same as those of the previous cases. Then, we obtained a χ^2-discrepancy δ of the order of 2.8×10^{-15}, a safe sample size of the order of 9.0×10^{14} and a risky sample size of the order of 4.6×10^{15}. In this case, safe and risky sample sizes are too large to carry out the χ^2-tests. For higher luxury indices, we could not compute δ by the same reason as for RANLUX.

6 Conclusion

We have shown by sum-discrepancy tests that RANLUX and ran_array generators have deviations from the ideal probability distribution if the discarded portion is 3/4 or less. If we discard more, then the generator would be safer, but is slower. We would like to point out that there are other generators which discard nothing but seem to have at least same quality in randomness, like Mersenne Twister MT19937 [10]. The table below gives a comparison of its speed with various luxury values of RANLUX. It exhibits the time required to generate 10^7 random numbers for MT19937 and RANLUX with luxury value $L = 24, 48, 97, 223, 389$. We measured the time on FreeBSD4.2R with PentiumIII 1GHz processor.

The generator ran_array shows similar tendency. For the recommended luxury index $p = 1000$, it is more than three times slower than MT19937.

MT19937	RANLUX($L = 24$)	$L = 48$	$L = 97$	$L = 223$	$L = 389$
0.311	0.273	0.638	1.169	2.618	4.316

(sec.)

Table 6: Speed Comparison between MT19937 and RANLUX

We conclude that in a serious simulation RANLUX should be used with luxury level at least 3, as recommended by Lüscher[8]. From the viewpoint of efficiency, there are better alternatives, like Mersenne Twister, available at the website `http://www.math.keio.ac.jp/matumoto/emt.html` .

References

[1] Billingsley, P.: Probability and Measure. John Wiley & Sons, New York, second edition, 1986.

[2] Couture, R. and L'Ecuyer, P.: On the lattice structure of certain linear congruential sequences related to AWC/SWB generators. Mathematics of Computation, 62 (1994) 798–808.

[3] L'Ecuyer, P. and Hellekalek, P.: Random number generators: Selection criteria and testing, In P. Hellekalek and G. Larcher, editors, Random and Quasi-Random Point Sets, volume 138 of Lecture Notes in Statistics, Springer, New York, (1998) 223–265.

[4] L'Ecuyer, P., Simard, R. and Wegenkittl, S.: Sparse serial tests of uniformity for random number generators. SIAM Journal on Scientific Computing, 2001. To appear.

[5] James, F.: RANLUX: A Fortran implementation of the high-quality pseudo-random number generator of Lüscher, Computer Physics Communications, 79 (1994) 111–114.

[6] Knuth, D. E.: The Art of Computer Programming. Vol. 2. Seminumerical Algorithms 3rd Ed. Addison-Wesley, Reading, Mass., 1997.

[7] Linhart, H. and Zucchini, W.: Model Selection. John Wiley & Sons, New York, 1986.

[8] Lüscher, M.: A portable high-quality random number generator for lattice field theory simulations, Computer Physics Communications, 79 (1994) 100–110.

[9] Marsaglia, G. and Zaman, A.: A new class of random number generators, The Annals of Applied Probability, 1 (1991) 462–480.

[10] Matsumoto, M. and Nishimura, T.: Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator, ACM Transactions on Modeling and Computer Simulation, 8 (1998) 3–30.

[11] Matsumoto, M. and Nishimura, T.: A nonempirical test on the weight of pseudorandom number generators, In K.T. Fang, F.J.Hickernel, and H. Niederreiter, editors, Monte Carlo and Quasi-Monte Carlo Methods 2000, Springer, (2002) 381–395.

[12] Reiter, S. and Stegeman, J.D.: Classical harmonic analysis and locally compact groups. Oxford Science Publications, Oxford, 2000.

[13] Tezuka, S., L'Ecuyer, P. and Couture, R.: On the add-with-carry and subtract-with-borrow random number generators. ACM Transactions of Modeling and Computer Simulation, 3 (1994) 315–331.

[14] Tiku, M.: Noncentral chi-square distribution, In S. Kots and N. L. Johnson, editors, Encyclopedia of Statistical Sciences, vol. 6, John Wiley, (1981) 276–278.