Resolution-Stationary Random Number Generators

Francois Panneton

Caisse Centrale Desjardins, 1 Complexe Desjardins, bureau 2822
Montral (Québec), H5B 1B3, Canada

Pierre L'Ecuyer

Département d'Informatique et de Recherche Opérationnelle
Université de Montréal, C.P. 6128, Succ. Centre-Ville
Montréal (Québec), H3C 3J7, Canada

Abstract

Besides speed and period length, the quality of uniform random number generators is usually assessed by measuring the uniformity of their point sets, formed by taking vectors of successive output values over their entire period length. For $\mathbb{F}_2$-linear generators, the commonly adopted measures of uniformity are based on the equidistribution of the most significant bits of the output. In this paper, we point out weaknesses of these measures and introduce generalizations that also give importance to the low-order (less significant) bits. These measures look at the equidistribution obtained when we permute the bits of each output value in a certain way. In a parameter search for good generators, a quality criterion based on these new measures of equidistribution helps avoiding generators that fail statistical tests targeting their low-order bits. We also introduce the notion of resolution-stationary generators, whose point sets are invariant under a multiplication by certain powers of 2, modulo 1. For such generators, less significant bits have the same equidistribution properties as the most significant ones. Tausworthe generators have this property. We finally show how an arbitrary $\mathbb{F}_2$-linear generator can be made resolution-stationary by adding an appropriate linear transformation to the output. This provides new efficient ways of implementing high-quality and long-period Tausworthe generators.

Key words: random number generation, linear recurrence modulo 2, uniformity, quasi-Monte Carlo, Tausworthe generator

1 Introduction

A broad family of random number generators (RNGs) are based on a linear recurrence modulo 2 (i.e., a linear recurrence in the finite field $\mathbb{F}_2$ whose two elements are represented by 0 and 1), sometimes with another linear transformation at the output. We call them $\mathbb{F}_2$-*linear generators*. They can be defined by the general equations

$$\mathbf{x}_n = (x_n^{(0)}, \ldots, x_n^{(k-1)})^{\mathsf{t}} = \mathbf{A}\mathbf{x}_{n-1} \tag{1}$$

$$\mathbf{y}_n = (y_n^{(0)}, \ldots, y_n^{(L-1)})^{\mathsf{t}} = \mathbf{B}\mathbf{x}_n \tag{2}$$

$$u_n = \sum_{\ell=0}^{L-1} 2^{-\ell-1} y_n^{(\ell)} \in [0,1) \tag{3}$$

where k and L are positive integers, $\mathbf{x}_n$ is the *state* at step n, $\mathbf{A}$ is a $k \times k$ *transition matrix*, $\mathbf{B}$ is a $L \times k$ *tempering matrix*, $\mathbf{y}_n$ is the L-bit *output vector* at step n, and all elements of these vectors and matrices are elements of $\mathbb{F}_2$. That is, all the arithmetic in (1) and (2) is done "modulo 2." The real number $u_n \in [0,1)$ is the *output* of the generator at step n and the number of bits in this output, L, is called the *resolution* of the generator.

It is well-known [3,10] that each bit of $\mathbf{x}_n$ or $\mathbf{y}_n$, i.e., each sequence $\{x_n^{(\ell)},\ n \geq 0\}$ or $\{y_n^{(\ell)},\ n \geq 0\}$ in $\mathbb{F}_2$ obeys a linear recurrence of order k whose characteristic polynomial is $\det(\mathbf{A} - z\mathbf{I})$, the characteristic polynomial of the matrix $\mathbf{A}$, where $\mathbf{I}$ denotes the identity matrix. For this reason, we call k the *order* of the generator. The maximal period length of such a generator is $2^k - 1$.

A traditional criterion to assess the quality of these RNGs is the equidistribution of the point set

$$\Psi_t = \{(u_0, \ldots, u_{t-1}) : \mathbf{x}_0 \in \mathbb{F}_2^k\}$$

(viewed as a multiset) that contains the vectors of t successive output values produced by the generator from all possible initial states [2,4,15,17]. To define this notion, for any given positive integers t and ℓ, we partition $[0,1)^t$ along each axis into 2^ℓ equal subintervals. This determines $2^{t\ell}$ cubic cells. If each cell

⋆ This work has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Grant Number ODGP0110050, NATEQ-Québec grant Number 02ER3218, and a Canada Research Chair to the second author. The first author benefited from NSERC and NATEQ scholarships.

Email address: panneton@iro.umontreal.ca (Francois Panneton).

URLs: http://www.iro.umontreal.ca/~panneton (Francois Panneton), http://www.iro.umontreal.ca/~lecuyer (Pierre L'Ecuyer).

contains exactly $2^{k-t\ell}$ points from Ψ_t, we say that Ψ_t and the generator are (t,ℓ)-equidistributed. For any given $\ell > 0$, let

$$t_\ell = \max\{t \geq 0 : \Psi_t \text{ is } (t,\ell)\text{-equidistributed}\}.$$

We have the upper bound $t_\ell \leq \lfloor k/\ell \rfloor$. We call

$$\Delta_\ell \stackrel{\text{def}}{=} \lfloor k/\ell \rfloor - t_\ell$$

the *dimension gap* in resolution ℓ. By combining the resolution gaps for different values of ℓ, we can define various uniformity criteria. One example is $\Delta = \sum_{\ell=1}^{L} \Delta_\ell$ [4,5,8,13,**?**]. A smaller Δ_ℓ means better uniformity when we consider only the ℓ most significant bits of the output values. A small Δ indicates a good overall uniformity of the most significant bits.

One limitation of this type of criterion is that it puts emphasis on the most significant bits and does not care much about the least significant ones. Because of this, when making computer searches for good $\mathbb{F}_2$-linear generators based on the optimization of criterion Δ, we may end up with generators whose low-order bits have bad equidistribution. These generators may then fail statistical tests aimed at these least significant bits.

Example 1 When searching for good parameters for the WELL generators of [**?**], with the REGPOLY software package [11], we found several instances with $k = 128$ and $L = 32$ having full period length $2^{128} - 1$ and $\Delta = 5$, a fairly good value. We picked two of them that differed significantly by the quality of their least significant bits, the first one fairing much better than the second one in this regards (we will return to this in Example 2). We tested them with the batteries of statistical tests SmallCrush and Crush from TestU01 [6]. These batteries run in about one minute and 90 minutes of CPU time, respectively. As it turned out, the first generator passed all the tests except those that test the linear complexity (or linearity) of the output sequence (all $\mathbb{F}_2$-linear generators with such a small value of k fail these types of tests), whereas the second one failed several other tests oriented toward the least significant bits.

The primary goal of this paper is to examine uniformity criteria for $\mathbb{F}_2$-linear generators and see how these criteria can take the least significant bits into account. For this, in the next section, we introduce a new criterion that generalizes Δ. This criterion looks not only at the uniformity of Ψ_t, but at a larger class of point sets constructed by permuting the bits of each output $\mathbf{y}_n$. In Section 3, we introduce a notion of resolution-stationary generators. For generators having this property, the new criterion that generalizes Δ typically requires less effort to compute. We look at different classes of generators and see to what extent they are (partially or totally) resolution-stationary. We

prove an important result saying that the class of full-period $\mathbb{F}_2$-linear resolution stationary generators contains the class of Tausworthe generators and is not much larger. We then show how an arbitrary $\mathbb{F}_2$-linear generator can be turned into a resolution-stationary generator by applying an appropriate linear output transformation. This technique effectively turns an arbitrary $\mathbb{F}_2$-linear RNG into a Tausworthe RNG. From another viewpoint, it provides a variety of efficient implementation methods for Tausworthe generators.

2 A Generalization of Δ

For an arbitrary vector of bit indexes $J = (j_0, \ldots, j_{L'-1})$, where $L' > 0$ and $0 \le j_\ell < L$ for each ℓ, we can construct the output as

$$u_n = \sum_{\ell=0}^{L'-1} 2^{-\ell-1} y_n^{(j_\ell)}.$$

We denote by $\Psi_t(J)$ the set of all t-dimensional vectors $(u_0, \ldots, u_{t-1})$ (the counterpart of Ψ_t) when u_n is redefined in this way. Thus, $\Psi_t(J)$ depends on L' bits of each $\mathbf{y}_n$.

We do not assume that the j_ℓ's are in increasing order. For example, the vector J can be a permutation of $(0, \ldots, L-1)$, in which case $\Psi_t(J)$ is obtained by applying the corresponding permutation to the bits of each output vector $\mathbf{y}_n$.

We define

$$\begin{aligned} t(J) &= \max\{t \ge 0 : \Psi_t(J) \text{ is } (t,\ell)\text{-equidistributed}\}, \\ \Delta_\ell(J) &= \lfloor k/\ell \rfloor - t(j_0, \ldots, j_{\ell-1}) \qquad \text{for } 1 \le \ell \le L', \text{ and} \\ \Delta(J) &= \sum_{\ell=1,\ldots,L'} \Delta_\ell(J). \end{aligned}$$

The latter is the value of Δ for the point set $\Psi_t(J)$. With $J = (0, \ldots, L-1)$ we recover the standard criterion mentioned in the introduction.

If we want to consider all bits on equal footing, one of the simplest choices is to take the vectors J of the form $J_p = (p, \ldots, L-1, 0, \ldots, p-1)$ for $0 \le p < L$. This corresponds to applying a p-bit left rotation to each output vector $\mathbf{y}_n$. The p most significant bits become the p least significant ones and the $L-p$ least significant bits become the $L-p$ most significant ones. One criterion that considers all these point sets at once is

$$\tilde{\Delta} = \max_{0 \le p < L} \Delta(J_p).$$

Example 2 We return to the two WELL generators mentioned in Example 1, which we now call WELL-A and WELL-B. They have $k = 128$ and $L = 32$. We computed $\Delta(J_0), \ldots, \Delta(J_{31})$ for these RNGs. The values are given in Tables 1 and 2. We get $\tilde{\Delta} = 9$ for WELL-A and $\tilde{\Delta} = 255$ for WELL-B; a huge difference! The worst possible value of $\Delta(J_p)$ for a generator with $L = 32$ and $k = 128$ is 255, and WELL-B reaches this value for five different values of p. WELL-A is definitely much better with respect to this criterion.

Table 1
Values of $\Delta(J_p)$ for $p = 0, \ldots, 31$ (reading row by row from left to right and from top to bottom) for the WELL-A generator.

5	4	4	9	6	8	7	7
8	8	7	5	5	2	6	5
4	7	7	1	5	4	8	2
8	6	5	8	6	5	7	2

Table 2
Values of $\Delta(J_p)$, $p = 0, \ldots, 31$, for WELL-B.

5	7	10	13	18	27	37	44
54	65	79	106	104	100	112	125
144	163	205	213	215	215	213	215
235	**255**	**255**	**255**	**255**	251	**255**	4

A similar type of behavior can be observed with the xorshift generators proposed by [7] and studied further in [13].

3 Resolution-Stationary Generators

Certain point sets $\Psi_t(J)$ have the interesting property that if we shift the bits of all coordinates of all points by j positions to the left and then discard the first j bits (this is equivalent to multiplying all output values u_n by 2^j, modulo 1, or to adding j to each coordinate of the vector J), then for certain values of j the point set $\Psi_t(J)$ remains unchanged. Of course, this is possible only if enough bits remain after the shift, i.e., only if $j < L - j_\ell$ for all ℓ when $J = (j_0, \ldots, j_{L'-1})$. We call *resolution-stationary* a generator for which this property holds whenever the condition $0 < j < L - j_\ell$ is satisfied for all ℓ. If this holds only when j is a multiple of some positive integer v, we use the term *v-wise resolution stationary*. These properties are useful because they permit one to obtain the values of $\Delta(J)$ for several J's at once by computing the value for a single J only. This can speed up the search for good parameters. The remainder of this section is devoted to studying these properties.

To make the above definitions more formal, for $J = (j_0, \ldots, j_{L'-1})$ and an integer $j \geq 0$, denote by $J + j = (j_0 + j, \ldots, j_{L'-1} + j)$ the vector obtained by

adding j to each coordinate of J and let $j_{\max} = \max_{0 \le \ell < L'} j_\ell$.

Definition 1 *For a given positive integer v, a generator is called* v-wise resolution-stationary *if for all t, whenever j is a multiple of v and $0 \le j < L - j_{\max}$, we have*

$$\Psi_t(J) = \Psi_t(J + j).$$

When this holds for $v = 1$, we simply say resolution-stationary.

Lemma 1 *For a v-wise resolution-stationary generator, we have $t(J) = t(J + j)$, $\Delta_\ell(J) = \Delta_\ell(J + j)$, and $\Delta(J) = \Delta(J + j)$ whenever j is a multiple of v and $0 \le j < L - j_{\max}$.*

The proof follows directly from the definitions. This result implies that when v is small, we do not need to compute all $\Delta(J)$'s: several ones can be deduced from the other ones.

Example 3 Suppose we want to compute $\tilde{\Delta}$ with $L = 32$. The values of $t(J)$ required in this case include (among others) $t(i, i + 1)$ for $i = 0, \ldots, 30$. But if the generator is resolution-stationary, then these 31 values are all equal to $t(0, 1)$, so we only need to compute the latter. If the generator is 2-wise resolution-stationary instead, then $t(0, 1) = t(2, 3) = \cdots = t(30, 31)$ and $t(1, 2) = t(3, 4) = \cdots = t(29, 30)$, so in this case we only need to compute $t(0, 1)$ and $t(1, 2)$, and we save 29 computations. In general, if no resolution-stationarity is assumed, we need to compute $L \times L$ values $t(J)$ to obtain $\tilde{\Delta}$, whereas if it is resolution-stationary, that number is reduced to $L(L + 1)/2$, counting in both cases all the one-dimensional vectors J (for these, it is often known by construction that $t(J) = k$).

Lemma 2 *If the output of a given RNG satisfies*

$$y_n^{(\ell)} = y_{n+d}^{(\ell+v)} \tag{4}$$

for $0 \le \ell < L - v$ and all $n > 0$, for a given integer $d \ge 0$, then this RNG is v-wise resolution-stationary. Conversely, if an RNG is v-wise resolution-stationary and has full period length $2^k - 1$, then it satisfies (4) for $0 \le \ell < L - v$ and all $n > 0$, for some integer $d \ge 0$

Proof: For the first part, observe first that (4), together with the fact that $\{y_n^{(\ell)}, n \ge 0\}$ follows the same recurrence for all ℓ, implies that these bit sequences are purely periodic (i.e., they have no transient part). For a given

integer $i \geq 0$ such that $j = iv < L - j_{\max}$, denote

$$u_n = \sum_{\ell=0}^{L'-1} 2^{-\ell-1} y_n^{(j_\ell)}$$

and

$$\tilde{u}_n = \sum_{\ell=0}^{L'-1} 2^{-\ell-1} y_n^{(j_\ell+iv)}.$$

Then, for any integer $n \geq 0$, we have

$$\begin{aligned}\Psi_t(J) &= \{(u_n, \ldots, u_{n+t-1}) : \mathbf{x}_0 \in \mathbb{F}_2^k\} \\ &= \{(\tilde{u}_{n+id}, \ldots, \tilde{u}_{n+id+t-1}) : \mathbf{x}_0 \in \mathbb{F}_2^k\} \\ &= \Psi_t(J + iv),\end{aligned}$$

which proves that the RNG is v-wise resolution-stationary.

Conversely, suppose $\Psi_t(J) = \Psi_t(J + iv)$ for all t whenever $j = iv < L - j_{\max}$. Take $J = (0, \ldots, L - v - 1)$ and $i = 1$. We know that the sequences of $\{y_n^{(\ell)}, n \geq 0\}$ and $\{y_n^{(\ell+v)}, n \geq 0\}$ are the same bit sequence, with different starting points, for all ℓ. So we must have $y_n^{(0)} = y_{n+d}^{(v)}$ for some integer $d \geq 0$. But then, since $\Psi_t(J) = \Psi_t(J + v)$ for all t, the lag between these two bit sequences for $\ell > 0$ must be the same as for $\ell = 0$, i.e., it must also be d. That is, we must have $y_n^{(\ell)} = y_{n+d}^{(\ell+v)}$ for the same d, for $\ell = 1, \ldots, L' - 1$. ∎

The primary example of resolution-stationary generators is the class of Tausworthe generators, to be discussed in the next section. Here we give examples of v-wise resolution-stationary generators.

Example 4 For a large class of RNGs such as the TGFSR [8], xorshift generators [7,13], and the LFSR over $\mathbb{F}_{2^w}$ [11,12], the recurrence (1)–(3) has the special form

$$\mathbf{z}_n = \sum_{i=1}^{r} \mathbf{D}_i \mathbf{z}_{n-i} \tag{5}$$

$$\mathbf{y}_n = (\mathbf{T}\mathbf{z}_n^{\mathsf{t}}, \mathbf{T}\mathbf{z}_{n-1}^{\mathsf{t}}, \mathbf{T}\mathbf{z}_{n-2}^{\mathsf{t}}, \ldots)^{\mathsf{t}} \tag{6}$$

$$u_n = \sum_{\ell=0}^{L-1} 2^{-\ell-1} y_n^{(\ell)} \tag{7}$$

where r and w are positive integer such that $k = rw$, $\mathbf{z}_n$ is a w-bit vector, $\mathbf{y}_n$ is a L-bit vector, and $\mathbf{T}$ and the $\mathbf{D}_i$'s are $w \times w$ matrices. Here, we have $\mathbf{x}_n =$

$(\mathbf{z}_n^{\mathsf{t}}, \ldots, \mathbf{z}_{n-r+1}^{\mathsf{t}})^{\mathsf{t}}$. From Equation (6), it is easy to see that these generators are w-wise resolution-stationary with $d = 1$. Of course, this resolution-stationarity is meaningless if $L = w$, which is often the case for these types of generators. In practice, these generators are typically not resolution-stationary.

Example 5 In [12], the authors use LFSR generators over $\mathbb{F}_{2^w}$ (a special case of Example 4) to construct point sets Ψ_t of small cardinality for quasi-Monte Carlo integration. For that, they use values of w and r ranging from 2 to 9 and for which $k = rw \leq 18$, so that the number of points does not exceed 2^{18}. They take $L = 32$. These small generators are w-wise resolution-stationary.

4 Links with Tausworthe Generators

A Tausworthe generator [14,4,5] is defined by taking $\mathbf{A} = \mathbf{C}^s$, where

$$\mathbf{C} = \begin{pmatrix} & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ a_k & a_{k-1} & a_{k-2} & \ldots & a_1 \end{pmatrix}$$

with $a_k = 1$ and $\mathbf{C}$ has a primitive characteristic polynomial $P(z) = z^k - a_1 z^{k-1} - \cdots - a_k$, s is a positive integer such that $\mathsf{gcd}\ (s, 2^k - 1) = 1$, and $\mathbf{y}_n = (\mathbf{x}_n, \mathbf{C}^k \mathbf{x}_n, \mathbf{C}^{2k} \mathbf{x}_n, \ldots)$. Here, L can be viewed as infinite. The period length of this RNG is $\rho = 2^k - 1$ and each $\mathbf{y}_n$ is a bit sequence with period length ρ.

Proposition 1 *Any Tausworthe RNG is resolution-stationary.*

Proof: As pointed out in [1], for the Tausworthe generator, the sequences $\{y_n^{(\ell)},\ n \geq 0\}$ for $\ell = 0, 1, 2, \ldots$, are the same sequence with starting points that are spaced at equal distance d from each other, where d is the multiplicative inverse of s modulo ρ (which implies that $\mathsf{gcd}\ (d, \rho) = 1$). That is, $y_n^{(\ell)} = y_{n+d}^{(\ell+1)}$ for all ℓ. The result then follows from Lemma 2. (A related result is given in [5], at the end of Section 2). ∎

Proposition 2 *An $\mathbb{F}_2$-linear generator defined via (1)–(3) and with full period $\rho = 2^k - 1$ can be represented as a Tausworthe generator if and only if it is resolution-stationary with* $\mathsf{gcd}\ (d, \rho) = 1$.

Proof: If it can be represented as a Tausworthe generator, then the property follows from the proof of the preceding proposition. Suppose now that it is resolution-stationary for some integer d with $\mathsf{gcd}\ (d, \rho) = 1$. Then, from

Lemma 2, it satisfies $y_n^{(\ell)} = y_{n+d}^{(\ell+1)}$ for all $\ell < L-1$. The fact that the RNG is equivalent to a Tausworthe generator then follows from Fushimi's results [1]. ∎

The next example illustrates what happens when the condition $\mathsf{gcd}\ (d, \rho) = 1$ is not satisfied.

Example 6 Let $k = 4$, $L = \infty$,

$$\mathbf{A} = \begin{pmatrix} 1\,0\,0\,1 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 1\,1\,0\,1 \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} \mathbf{I} \\ \mathbf{A} \\ \mathbf{A}^2 \\ \vdots \end{pmatrix},$$

and $\mathbf{x}_0 = (0, 1, 0, 0)^{\mathsf{t}}$. This RNG has full period length $\rho = 2^k - 1 = 15$ and is easily seen to be resolution-stationary with $d = 3$. The bit sequences $y_n^{(\ell)}$ for $\ell = 0, \ldots, 5$ and $n \geq 0$ are

$$\begin{array}{l} 011110001001101011110001001101\ldots \\ 101011110001001101011110001001\ldots \\ 001101011110001001101011110001\ldots \\ 001001101011110001001101011110\ldots \\ 110001001101011110001001101011\ldots \\ 011110001001101011110001001101\ldots \end{array}.$$

Here, the bits in any given row have period length 15 and the next row is obtained by shifting the current row to the right by three bits. After five 3-bit shifts, we obtain the same row again. This means that the bits in any column have period length $\rho/d = 5$. But since column n is the bit sequence $\mathbf{y}_n$, the period length in each column would have to be $\rho = 15$ for the RNG to be representable as a Tausworthe RNG.

5 Achieving Resolution-Stationarity by Transforming the Output

Many interesting classes of $\mathbb{F}_2$-linear generators, for instance the Mersenne Twister [9] and the WELL [**?**], are not resolution-stationary. However, any $\mathbb{F}_2$-linear RNG can be made resolution-stationary by applying an appropriate linear output transformation as follows. This is equivalent to redefining the matrix $\mathbf{B}$.

The idea is to replace the output vector $\mathbf{y}_n$ at step n by

$$\tilde{\mathbf{y}}_n = (y_n^{(0)}, y_{n-d}^{(0)}, y_{n-2d}^{(0)}, \ldots, y_{n-(L-1)d}^{(0)})^{\mathsf{t}}$$

It then follows from Lemma 2 that this redefined generator is resolution-stationary. If the original generator has full period $\rho = 2^k - 1$ and $\gcd(d, \rho) = 1$, this output transformation turns it into a Tausworthe generator. Thus, this technique can be viewed as a new way of implementing a Tausworthe generator from the output of *any* $\mathbb{F}_2$-linear generator. By applying this transformation to a fast and long-period underlying (original) generator, with good uniformity for its high-order bits and possibly many non-zero coefficients in its characteristic polynomial, we can obtain a fast resolution-stationary generator which is more likely to have good uniformity for all its bits.

An easy way of implementing this output transformation is by saving the relevant bits into L-bit vectors $\mathbf{s}_0, \ldots, \mathbf{s}_{d-1}$ so that $y_{n-d}^{(0)}, y_{n-2d}^{(0)}, \ldots, y_{n-(L-1)d}^{(0)}$ can be recovered from the bit vector $\mathbf{s}_{n \bmod d}$ at step n.

Before we start using the generator, we must initialize the vectors $\mathbf{s}_j$ appropriately. Let $\mathbf{e}_1 = (1, 0, \ldots, 0)^{\mathsf{t}}$ denote the L-bit unit vector, $\oplus$ the xor operator, $\gg$ the right shift operator, and & the bitwise-and operator. Suppose that $L > 1$. The vectors $\mathbf{s}_j$ can be initialized by the following algorithm:

Initialize $\mathbf{x}_0$ to a random bit vector;
Initialize $\mathbf{s}_j$ to the zero bit vector for $j = 0, \ldots, d-1$;
For $i = 0, \ldots, L-1$ do
    For $j = 0, \ldots, d-1$ do
        $\mathbf{s}_j \leftarrow (\mathbf{y}_{id+j} \,\&\, \mathbf{e}_1) \oplus (\mathbf{s}_j \gg 1)$.

The role of the bit mask $\mathbf{e}_1$ is to pick up the first bit. At the end of this initialization, $\mathbf{s}_j$ contains

$$\tilde{\mathbf{y}}_{(L-1)d+j} = (y_{(L-1)d+j}^{(0)}, y_{(L-2)d+j}^{(0)}, \ldots, y_j^{(0)})^{\mathsf{t}}.$$

Once this initialization is done, the vector $\tilde{\mathbf{y}}_n$, for $n \geq Ld$, is obtained by

$$\mathbf{s}_{n \bmod d} \leftarrow (\mathbf{s}_{n \bmod d} \gg 1) \oplus (\mathbf{y}_n \& \mathbf{e}_1)$$
$$\tilde{\mathbf{y}}_n \leftarrow \mathbf{s}_{n \bmod d}.$$

For this to work, the output vectors $\tilde{\mathbf{y}}_n$ must be generated in sequence, so that the vectors $\mathbf{s}_j$ are updated appropriately. At each step n, we have $\mathbf{s}_{n \bmod d} = (y_n^{(0)}, y_{n-d}^{(0)}, y_{n-2d}^{(0)}, \ldots, y_{n-(L-1)d}^{(0)})^{\mathsf{t}}$.

A drawback of this method is that it requires additional memory for the

$\mathbf{s}_j$'s. For a high-quality generator, the value of d must be large, otherwise the bits will follow a common recurrence with a small lag. But a larger d means more memory. This trap can be avoided by combining two (or more) different Tausworthe generators that can be implemented in this way. Each one can have a very small d and yet the combined Tausworthe generator will generally have a very large d [16,4]. In other words, the output transformation technique just introduced can be useful to implement the components of an efficient large-period combined Tausworthe generator.

6 Conclusion

$\mathbb{F}_2$-linear RNGs are usually selected on the basis of the good uniformity for their most significant bits; the least significant bits are neglected. To remedy this situation, we have introduced a new uniformity criterion $\tilde{\Delta}$ that gives equal importance to all the bits. When searching for good RNGs with respect to $\tilde{\Delta}$, we can still use the old criterion Δ as a filter, and compute $\tilde{\Delta}$ only for the RNGs whose value of Δ exceeds a given threshold, to save computing time.

The criterion $\tilde{\Delta}$ considered in this paper is based on the permutations J_p, but other types of index vectors than J_p could be considered as well, perhaps depending on the specific classes of applications that we have in mind. For example, if the RNG is to be used as a source of random bits by taking all bits of $\mathbf{y}_0, \mathbf{y}_1, \ldots$ in succession, then we may want to consider the uniformity of the point set defined by all blocks of t successive bits that can occur in the bit sequence $(\mathbf{y}_0^{\mathsf{t}}, \mathbf{y}_1^{\mathsf{t}}, \ldots)$.

Finally, we introduced a notion of resolution-stationarity, studied the links between this notion and Tausworthe RNGs, and proposed a way of transforming an arbitrary full-period $\mathbb{F}_2$-linear RNG into a Tausworthe RNG, thus recovering the nice resolution-stationary properties of these RNGs.

References

[1] M. Fushimi. An equivalence relation between Tausworthe and GFSR sequences and applications. *Applied Mathematics Letters*, 2(2):135–137, 1989.

[2] M. Fushimi and S. Tezuka. The k-distribution of generalized feedback shift register pseudorandom numbers. *Communications of the ACM*, 26(7):516–523, 1983.

[3] P. L'Ecuyer. Uniform random number generation. *Annals of Operations Research*, 53:77–120, 1994.

[4] P. L'Ecuyer. Maximally equidistributed combined Tausworthe generators. *Mathematics of Computation*, 65(213):203–213, 1996.

[5] P. L'Ecuyer. Tables of maximally equidistributed combined LFSR generators. *Mathematics of Computation*, 68(225):261–269, 1999.

[6] P. L'Ecuyer and R. Simard. *TestU01: A Software Library in ANSI C for Empirical Testing of Random Number Generators*, 2002. Software user's guide. Available at `http://www.iro.umontreal.ca/~lecuyer`.

[7] G. Marsaglia. Xorshift RNGs. *Journal of Statistical Software*, 8(14):1–6, 2003. See `http://www.jstatsoft.org/v08/i14/xorshift.pdf`.

[8] M. Matsumoto and Y. Kurita. Twisted GFSR generators II. *ACM Transactions on Modeling and Computer Simulation*, 4(3):254–266, 1994.

[9] M. Matsumoto and T. Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30, 1998.

[10] H. Niederreiter. New methods for pseudorandom number and pseudorandom vector generation. In *Proceedings of the 1992 Winter Simulation Conference*, pages 264–269. IEEE Press, 1992.

[11] F. Panneton and P. L'Ecuyer. Random number generators based on linear recurrences in F_{2^w}. In H. Niederreiter, editor, *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 367–378, Berlin, 2004. Springer-Verlag.

[12] F. Panneton and P. L'Ecuyer. Infinite-dimensional point sets based on linear recurrences over GF(2^w). In H. Niederreiter and D. Talay, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, Berlin, 2005. Springer-Verlag. to appear.

[13] F. Panneton and P. L'Ecuyer. On the xorshift random number generators. *ACM Transactions on Modeling and Computer Simulation*, 15(4), 2005. to appear.

[14] R. C. Tausworthe. Random numbers generated by linear recurrence modulo two. *Mathematics of Computation*, 19:201–209, 1965.

[15] S. Tezuka. *Uniform Random Numbers: Theory and Practice*. Kluwer Academic Publishers, Norwell, Mass., 1995.

[16] S. Tezuka and P. L'Ecuyer. Efficient and portable combined Tausworthe random number generators. *ACM Transactions on Modeling and Computer Simulation*, 1(2):99–112, 1991.

[17] J. P. R. Tootill, W. D. Robinson, and D. J. Eagle. An asymptotically random Tausworthe sequence. *Journal of the ACM*, 20:469–481, 1973.