# RECENT ADVANCES IN UNIFORM RANDOM NUMBER GENERATION

Pierre L'Ecuyer

Département d'IRO
Université de Montréal, C.P. 6128, Succ. A
Montréal, H3C 3J7, CANADA

## ABSTRACT

This paper discusses certain classes of uniform random number generators which have been studied and better understood in the recent few years. Most of the attention is devoted to combined generators. We also mention others and point out some pitfalls. Combination is a good way to obtain fast and reliable generators, but the structural properties of the combined generator should be carefully examined before it could be recommended. Nonlinear generators offer some promise, but still require deeper investigation before specific instances can be safely recommended.

## 1. INTRODUCTION

Random numbers are the nuts and bolts of all stochastic simulations. Simple linear congruential generators (LCGs) (Bratley, Fox, and Schrage 1987; Knuth 1981) are still in widespread use for generating uniform random numbers, mainly because of their simplicity and ease of implementation. However, LCGs have several well-known defects and no longer satisfy the requirements of today's computer-intensive simulations, especially when their modulus fits into a 32-bit computer word (L'Ecuyer 1992; L'Ecuyer 1994c; Marsaglia 1985; Niederreiter 1992b).

Practically all random number generators (RNGs) used for simulation are deterministic automata with a finite state space, and so have a periodic behavior. Quality requirements for a general purpose RNG include a huge period length, good statistical properties, high speed, low memory usage, repeatability, portability, ease of implementation, and availability of jumping ahead and splitting facilities.

We will discuss the important questions of period length and statistical behavior in a moment. For certain simulation applications (e.g., in particle physics), billions of random numbers are required, and the generator's speed remains a critical factor, regardless of the available computing power. Memory utilization could become important when many "virtual" generators (many substreams) must be maintained in parallel. This is required, for example, for proper implementation of certain variance reduction techniques; see Bratley, Fox, and Schrage (1987), L'Ecuyer and Côté (1991), and L'Ecuyer (1994c). *Portability* means that the generator can be implemented efficiently in a standard high-level language, to produce exactly the same sequence (at least up to machine accuracy) with all "standard" compilers and on all "reasonable" computers. Being able to reproduce the same sequence of random numbers on a given computer or on different computers (*repeatability*) is important for program verification and for variance reduction (Bratley, Fox, and Schrage 1987; Ripley 1990). Repeatability is a major advantage of pseudorandom sequences with respect to sequences generated by physical devices. Of course, for the latter, one could store an extremely long sequence on a disk and reuse it as needed. But this is not as convenient as a good pseudorandom number generator which stands in a few lines of code. *Jumping ahead* means the ability to quickly compute, given the current state $s_n$, the state $s_{n+\nu}$ for any large $\nu$. This is useful for breaking up the sequence into long disjoint substreams and jump ahead quickly from one substream to the other. The package described by L'Ecuyer and Côté (1991) implements such facilities.

In terms of understanding the theoretical properties of a given generator, knowing the period length is not enough, even if it is astronomical. Suppose that our generator is to produce iid (independent and identically distributed) uniform variates over the interval $[0, 1]$ and let $u_n$ be the value generated at step $n$. Consider for example the set of all $t$-dimensional vectors of successive observations:

$$\Omega_t = \{\boldsymbol{u}_n = (u_n, \ldots, u_{n+t-1});\ n \geq 0\},$$

over the full period of the generator, for a given $t$.

Common practice is to demand that these points are very evenly distributed in the $t$-dimensional unit hypercube $[0,1]^t$. One may argue that points that are too evenly distributed do not look random and fail to imitate iid uniform variates as well as points whose distribution is too far from even. However, if the period length is astronomical and the starting point (or generator's *seed*) is selected at random, then one may view $\Omega_t$ as a sample space from which points are taken at random (one by one) by the generator, *without replacement*. If the points are very evenly distributed over $[0,1]^t$, then a good way to generate an iid sample from (approximately) the uniform distribution over $[0,1]$ would be to pick up points randomly from $\Omega_t$, *with replacement*, and use all the components of each vector. Picking with or without replacement will make practically no difference if the cardinality of $\Omega_t$ is several orders of magnitude larger than the number of values that we need. This (heuristic) argument suggests that (1) $\Omega_t$ be as evenly distributed as possible over $[0,1]^t$ and (2) the RNG should have a period length several orders of magnitude larger than whatever can be exhausted in practice.

"Superuniform" multidimensional distributions over the entire period, as just described, improve our confidence in the statistical behavior of the RNG over the fraction of the period that we use. That may be complemented with additional empirical statistical tests. However, empirical tests do not easily discriminate between good and mediocre generators. One should first select a generator on the basis of its theoretical properties, and then submit it to appropriate empirical tests. Some "standard" tests are described in Knuth (1981) and Marsaglia (1985). Ideally, the tests should be selected in relation with the target application. So, before using a general purpose generator, it may be wise to submit it to additional "specialized" empirical testing. In principle, for any RNG whose output sequence is periodic, it is possible to build a statistical test that the generator will fail miserably, if enough time is allowed. The idea of empirical statistical testing may then seem meaningless. However, from a pragmatic point of view, people usually feel good if the RNG passes a certain set of statistical tests which can be run in "reasonable" time. Further discussion of statistical testing can be found in L'Ecuyer (1992).

In the next section, we describe several popular classes of RNG based on linear recurrences. That includes the LGC, MRG, Tausworthe, GFSR, TGFSR, and AWC/SWB generators. In Section 3, we survey some recent developments regarding the combination of such linear-type generators. Section 4 discusses the lattice structure associated with those generators and the equidistribution properties over the entire period. In Section 5, we give a quick assessment of certain classes of nonlinear generators proposed in the last few years. For more extensive recent surveys and deeper treatments of RNGs, see Eichenauer-Herrmann (1992), James (1990), Knuth (1981), L'Ecuyer (1990), L'Ecuyer (1992), L'Ecuyer (1994c), Niederreiter (1991), Niederreiter (1992b), and Tezuka (1992). This paper is based largely on L'Ecuyer (1994c).

## 2. GENERATORS BASED ON LINEAR RECURRENCES

Multiple recursive generators (MRGs) (L'Ecuyer 1990; L'Ecuyer, Blouin, and Couture 1993; Niederreiter 1992b), defined as follows, generalize the LCGs:

$$x_n = (a_1 x_{n-1} + \cdots + a_k x_{n-k}) \bmod m; \quad (1)$$

$$u_n = x_n/m. \quad (2)$$

Here, the *modulus* $m$ and *order* $k$ are positive integers, while each $a_i$ belongs to $\mathbb{Z}_m = \{0, 1, \ldots, m-1\}$. For prime $m$ and properly chosen coefficients $a_i$, the MRG has a (maximal) period length $\rho = m^k - 1$. This can be achieved with only two non-zero coefficients $a_i$; e.g.,

$$x_n = (a_r x_{n-r} + a_k x_{n-k}) \bmod m. \quad (3)$$

Few non-zero coefficients makes the implementation faster, but also yields unfavorable limitations on the quality of the generator. Indeed, a necessary condition for an MRG to have "acceptable" behavior is that (1) have several non-zero coefficients $a_i$, whose sum of squares is "large enough" (L'Ecuyer 1994a). However, the generator then runs slower.

Taking $m = 2^e$ for $e > 1$ makes the implementation fast and easy, because the modulo operation just amounts to discarding the higher-order bits. However, the maximal period is then bounded above (for $k > 1$) by $(2^k - 1)m/2$, which is much smaller than $m^k$ for large $k$. Also, the maximal period for the $d$th least significant bit is at most $(2^k - 1)2^{d-1}$, and for $i = 2^{e-d-2} > 0$ and $d \geq 2$, all the points $(x_n, x_{n+i})$ lie on at most $2^{d-1}$ parallel lines. Furthermore, if the period is split into $2^d$ equal segments, all those segments are identical except for their $d$ most significant bits. For these and other similar reasons, we recommend that power-of-two moduli be avoided.

Division by $m$ as in (2) is not the only way of producing the output. A slightly more general way is to use $s$ terms of the recurrence (1) at each stage:

$$u_n = \sum_{j=1}^{L} x_{ns+j-1} m^{-j}, \tag{4}$$

where $s$ and $L \leq k$ are positive integers. The sequence $\{u_n\}$ is then called a *digital multistep sequence* (L'Ecuyer 1994c; Niederreiter 1992b). If (1) has period $\rho$ and $\gcd(\rho, s) = 1$, then (4) also has period $\rho$. The digital expansion (4) yields a better resolution than just $u_n = x_n/m$, and permits one to take smaller values of $m$. An important special case of (4) is when $m = 2$: each $u_n$ is then constructed by taking blocks of $L$ successive bits from the binary sequence (1), with spacings of $s-L \geq 0$ bits between the blocks. This results in the so-called *Tausworthe* generator (Knuth 1981; Niederreiter 1992b; Tausworthe 1965), whose implementation is discussed in Bratley, Fox, and Schrage (1987), L'Ecuyer (1994c), and Tezuka and L'Ecuyer (1991).

Another way of producing the output is to have $L$ copies of the recurrence (1) running in parallel, with different initial values, and to use one of those copies for each digit in the fractional expansion of $u_n$. Let $\{x_{j,n}\}$ denote the $j$th copy and suppose that it was started $d_j$ values ahead of the sequence $\{x_n\}$; that is, $x_{j,n} = x_{n+d_j}$ for all $j$ and $n$. One then has

$$u_n = \sum_{j=1}^{L} x_{j,n} m^{-j} = \sum_{j=1}^{L} x_{n+d_j} m^{-j}. \tag{5}$$

If the lags between the successive copies are identical, say $d_j = (j-1)d$ for some integer $d$, and if $\gcd(d, \rho) = 1$, then $n+d_j = n+(j-1)d = (ns+j-1)d$, so (5) becomes equivalent to (4) if we replace the sequence $\{x_n\}$ by $\{y_n = x_{nd}\}$; which can be accomplished by changing the coefficients of the recurrence (1) appropriately. If $m = 2$ and if there are only two non-zero coefficients as in (3), then the generator (5) is called a *generalized feedback shift register* (GFSR) generator (Fushimi and Tezuka (1983), Fushimi (1989)). If $X_n$ denotes the ($k$-bit) vector $(x_{1,n}, \ldots, x_{k,n})$, then the GFSR lends itself to the speedy implementation:

$$X_n = X_{n-r} \oplus X_{n-k},$$

where $\oplus$ denotes the bitwise exclusive-or.

A modification of the GFSR is the so-called *lagged-Fibonacci* generator, for which $\oplus$ can be replaced by any arithmetic or logical operation. One example is the *additive* generator (Knuth 1981):

$$X_n = (X_{n-r} + X_{n-k}) \bmod m, \tag{6}$$

where $m = 2^L$. It is called *subtractive* if $+$ is replaced by $-$. This is a special case of the MRG, but with a power-of-two modulus. Its maximal period length, for suitable choices of $r$ and $k$, is $(2^k - 1)2^{L-1} \approx 2^{k+L-1}$, which is $2^{L-1}$ times larger than that of a GFSR with the same values of $L$ and $k$, but falls way short of $2^{kL}$. Marsaglia (1985) and Marsaglia and Tsay (1985) give more details and specific examples with the operators $+$, $-$, and $\times$, in arithmetic modulo $2^L$. However, these additive generators turn out to have bad structural properties: all triples of the form $(u_n, u_{n+k-r}, u_{n+k})$, $n \geq 0$, lie in only two planes in the three-dimensional unit cube (see L'Ecuyer 1994a); so this author believes that they should be avoided.

Slight variations of additive and subtractive generators, called *add-with-carry* (AWC) and *subtract-with-borrow* (SWB), were proposed recently by Marsaglia and Zaman (1991). The modification is that a carry (or borrow) bit is maintained with the recurrence (6). This permits a period length of up to $M - 1$, where $M = m^k \pm m^r \pm 1$ (depending on the variant). It is a tremendous increase. Unfortunately, as shown by Tezuka, L'Ecuyer, and Couture (1994) and L'Ecuyer (1994a), these generators have the same bad structural properties as the additive and subtractive generators.

In a similar vein, Matsumoto and Kurita (1992) proposed a modification of GFSR generators maintaining the speed but increasing the period from $2^k - 1$ to $2^{kL} - 1$. They called them *twisted GFSR*. Again, those generators turned out to have bad structural and statistical properties (L'Ecuyer 1992; Tezuka 1992). Matsumoto and Kurita (1994) recognize that problem and propose an improved version.

## 3. COMBINED GENERATORS

Combination has long been advocated as a way of increasing the period length and improving the statistical properties of generators (Knuth 1981; L'Ecuyer and Côté 1991; L'Ecuyer 1994c; Marsaglia 1985; Tezuka and L'Ecuyer 1991; Wang and Compagner 1993). Unfortunately, many combined generators were not so well understood when they were designed, and this gave rise to not so good proposals in the literature. Some classes of combined generators have been successfully analyzed theoretically only very recently. We now discuss two of those classes: (a) combined MRGs and (b) combined Tausworthe/GFSR generators.

Consider $J$ MRGs running in parallel ($J \geq 2$):

$$x_{j,n} = (a_{j,1} x_{j,n-1} + \cdots + a_{j,k_j} x_{j,n-k_j}) \bmod m_j. \tag{7}$$

Suppose the $m_j$'s are pairwise relatively prime and that each recurrence $j$ is purely periodic with period length $\rho_j$; so, $x_{n+\rho_j} = x_n$ for all $n \geq 0$.

Let $\delta_1, \ldots, \delta_J$ be arbitrary integers such that $\gcd(\delta_j, m_j) = 1$ for each $j$ and define the two combinations:

$$z_n = \left(\sum_{j=1}^{J} \delta_j x_{j,n}\right) \bmod m_1; \qquad \tilde{u}_n = z_n / m_1 \quad (8)$$

and

$$w_n = \left(\sum_{j=1}^{J} \frac{\delta_j x_{j,n}}{m_j}\right) \bmod 1. \quad (9)$$

Let $k = \max(k_1, \ldots, k_J)$; $m = \prod_{j=1}^{J} m_j$; $n_j$ be the inverse of $m/m_j$ modulo $m_j$; and

$$a_i = \left(\sum_{j=1}^{J} \frac{a_{j,i} n_j m}{m_j}\right) \bmod m$$

for $i = 1, \ldots, k$, where $a_{j,i} = 0$ for $i > k_j$. Consider now the MRG (1–2). One can prove (see L'Ecuyer and Tezuka 1991 for $k = 1$ and L'Ecuyer 1994b for $k > 1$) that (9) is equivalent to (1–2), and has period length $\rho = \mathrm{lcm}(\rho_1, \ldots, \rho_J)$. These references also give tight bounds on $|u_n - \tilde{u}_n|$, which are close to zero when the $m_j$'s are close to each other. In other words, (9) is just a practical way of implementing an MRG with a large composite modulus, while (8) is a (slightly more efficient) way of implementing an approximation of that same MRG.

Advantages of the above combinations are (a) the increased period length; (b) the fact that (1) can have many non-zero coefficients even if the recurrence (7) of each component has only two non-zero coefficients; (c) addition of noise to the lattice structure in the case of the combination (8) (see the next section).

In terms of period length, the best one can achieve is $\rho_j = m_j^{k_j} - 1$, when each $m_j$ is prime. In that case, each $\rho_j$ is even, so $\rho \leq (m_1^{k_1} - 1) \cdots (m_J^{k_J} - 1)/2^{J-1}$. The total number of states for the combined generator (including the trivial states) is equal to $\prod_{j=1}^{J} m_j^{k_j}$, since each component has $m_j^{k_j}$ possible states. If the $k_j$'s are not all equal, this could be much less than $m^k$, in which case not all values of $(x_0, \ldots, x_{n+k-1})$ in (1) can be obtained as combinations of values of $(x_{j,0}, \ldots, x_{j,n+k-1})$ through (9); see Couture and L'Ecuyer (1994b). The states $(x_0, \ldots, x_{n+k-1})$ that can be obtained as a combination are *recurrent* states for (1), whereas those states that are not the result of a combination turn out to be *transient*.

Tausworthe and GFSR generators based on the recurrence (3) have important statistical defects (Matsumoto and Kurita 1988; Matsumoto and Kurita 1992; Compagner 1991) but may again be improved by combination. Tezuka and L'Ecuyer (1991) and Wang and Compagner (1993) propose to run $J$ "easy-to-implement" Tausworthe generators in parallel, the $j$th producing a sequence $\{u_{j,n},\ n \geq 0\}$, and to combine them by taking the bitwise exclusive-or of the $u_{j,n}$'s at each step $n$: $\{u_n = u_{1,n} \oplus \cdots \oplus u_{J,n},\ n \geq 0\}$. This combination is equivalent to a Tausworthe generator whose recurrence has a (reducible) characteristic polynomial which is the product of the characteristic polynomials of the individual components. If the latter polynomials are pairwise relatively prime, then the period is the least common multiple of the periods of the components, and could reach $\prod_{j=1}^{J}(2^{k_j} - 1)$, where $k_j$ is the order of the $j$th recurrence (the degree of the characteristic polynomial of component $j$). GFSR and twisted GFSR generators can also be combined in a similar way. Such combinations can be viewed as efficient implementations of recurrences with "good" characteristic polynomials.

## 4. LATTICES AND EQUIDISTRIBUTION

For a given dimension $t$, the set of all overlapping $t$-tuples of successive values produced by (1–2), from all possible initial states, is $T_t = \{\boldsymbol{u}_n = (u_n, \ldots, u_{n+t-1}) \mid n \geq 0,\ s_0 = (x_0, \ldots, x_{k-1}) \in \mathbb{Z}_m^k\}$. Let $L_t$ be the integer *lattice* generated by $T_t$ and $\mathbb{Z}_m^k$, that is, the set of all linear combinations of elements of $T_t$ and $\mathbb{Z}_m^k$, with integer coefficients. The points of $L_t$ lie in a set of equidistant parallel hyperplanes (Knuth 1981). For the points to be evenly distributed over the entire period, the distance $d_t$ between those successive hyperplanes should be small. Computing $d_t$ is often called the *spectral test*. Another quality measure for $L_t$ is the Beyer quotient $q_t$ (L'Ecuyer 1990; L'Ecuyer and Couture 1994), defined as the ratio of lengths of a shortest and longest vectors in a Minkowski reduced basis for the lattice, and which should be close to one. There now exist computer programs that permit one to compute $d_t$ and $q_t$ in dimensions up to around 40 or more (L'Ecuyer and Couture 1994), whatever be the size of the modulus $m$. The results of searches for good parameter values for simple and combined MRGs, with regards to the spectral test, are reported in L'Ecuyer, Blouin, and Couture (1993) and L'Ecuyer (1994b). Specific generator implementations are also given there.

The recurrence (1) can be generalized to a non-homogeneous recurrence, where a constant $b$ (say) is added to the right-hand-side of (1). The corresponding lattice is then simply shifted with respect to the origin (it becomes a *grid*), so its fundamental structure is unchanged.

When there are both transient and recurrent states, it is more appropriate to analyze the set $T_{r,t}$ of $t$-tuples which are recurrent, since only those states are obtained by the combination. One has $T_{r,t} \subseteq T_t$, and the inclusion is strict when the $k_j$'s are not all equal. Couture and L'Ecuyer (1994b) explain how to construct a lattice basis for the lattice $L_{r,t}$ associated with $T_{r,t}$ (or its shifting) and give several results and special techniques for computing $d_t$ efficiently in large dimensions for combined generators. They also give an illustration using the "RANMAR" generator proposed by Marsaglia, Zaman, and Tsang (1990), which combines a LCG with an MRG of order 97. That generator turns out to have a relatively bad high-dimensional structure.

The points $\tilde{\boldsymbol{u}}_n = (\tilde{u}_n, \ldots, \tilde{u}_{n+t-1})$, $n \geq 0$, produced by the combined generator (8) no longer belong to the lattice described above, because of the "noise" $\epsilon_n$. If we equate (or join) the opposite faces of the $t$-dimensional unit hypercube $[0,1]^t$, we obtain the $t$-dimensional unit torus. Computing the Euclidean distances in that torus is equivalent to "neglecting" the modulo 1 operation in (9). Then, the Euclidean distance between $\tilde{\boldsymbol{u}}_n$ and $\boldsymbol{u}_n$ in the unit torus is bounded by $\Delta\sqrt{t}$, where $\Delta = \max(|\Delta^+|, |\Delta^-|)$. Typically, the values of $\epsilon_n$ are also evenly distributed between $\Delta^-$ and $\Delta^+$ and, when $\Delta\sqrt{t}$ is larger than $d_t$, the hyperplane structure usually becomes unrecognizable.

Instead of $T_t$, one may want to consider vectors of *non-successive* values produced by the generator: fix a set of non-negative integers (called *lacunary indices*) $I = \{i_1, i_2, \cdots, i_t\}$, put $T_t(I) = \{(u_{i_1+n}, \ldots, u_{i_t+n}) \mid n \geq 0,\ s_0 = (x_0, \ldots, x_{k-1}) \in \mathbb{Z}_m^k\}$ and let $L_t(I)$ be the integer lattice generated by $T_t(I)$ and $\frac{1}{m}\mathbb{Z}_m^k$. The points of $L_t(I)$ again lie in equidistant parallel hyperplanes spaced, say, $d_t(I)$ apart. L'Ecuyer (1994c) explain how to construct a lattice basis and compute $d_t(I)$ for this more general case. Building on the results of Couture and L'Ecuyer (1994a) and Couture and L'Ecuyer (1994b), L'Ecuyer (1994a) examines the behavior of $d_t(I)$ for certain types of MRGs. He obtains large lower bounds on $d_t(I)$ for specific sets $I$, in small dimensions $t$, for some classes of generators. Bad behavior occurs in particular when (1) has few of small non-zero coefficients, or when $k = 1$ and the modulus $m$ can be expressed as a linear combination of powers of the multiplier $a_1$, with small coefficients. More specifically, if $I$ contains the indices $i$ such that $a_{k-i+1} \neq 0$, then

$$d_t(I) \ \geq \ \left(1 + \sum_{i=1}^{k} a_i^2\right)^{-1/2}.$$

If $k = 1$ and $m = \sum_{j=1}^{t} c_{i_j} a_1^{i_j}$ for some integers $c_\ell$, $\ell \in I$, then

$$d_t(I) \geq \left(\sum_{j=1}^{t} c_{i_j}^2\right)^{-1/2}.$$

It follows that for the AWC/SWB and additive or subtractive lagged-Fibonacci generators, the set $T_3(I)$, for a certain set $I$, is contained in only two planes in the three-dimensional space. This is obviously a serious defect. The two combined generators proposed in Marsaglia, Narasimhan, and Zaman (1990) and Marsaglia, Zaman, and Tsang (1990) can also be approximated by linear congruential generators for which $d_6(I) \geq 1/\sqrt{6} \approx 0.408$ for certain sets $I$. For simulation applications dealing specifically with random points in high dimensional space, those bad structures could have a dramatic effect (Ferrenberg, Landau, and Wong 1992; L'Ecuyer 1992).

Tausworthe generators (simple or combined) can be viewed as LCGs in a space of formal series and so have a lattice structure in that space, which can be used to analyze their equidistribution properties (Couture, L'Ecuyer, and Tezuka 1993; L'Ecuyer 1994c; Tezuka and L'Ecuyer 1991). Suppose we partition the unit hypercube $[0,1]^t$ into $2^{t\ell}$ cubic cells of equal size. If each cell contains the same number of points of $\Omega_t$ (assuming that we take the union of all subcycles of the generator to define $\Omega_t$, including the zero vector, so $\Omega_t = T_t$), we say that the sequence is $(t,\ell)$-*equidistributed*. It is possible only for $\ell \leq \lfloor k/t \rfloor$, since this $\Omega_t$ has cardinality $2^k$. When the sequence is $(t, \lfloor k/t \rfloor)$-equidistributed for $t = 1, \ldots, k$, it is called *maximally equidistributed*, or *asymptotically random*. Generators that are almost maximally equidistributed are proposed by Tezuka and L'Ecuyer (1991), while several other numerical illustrations are given in Couture, L'Ecuyer, and Tezuka (1993). Tezuka (1994) shows how to analyze the equidistribution of combined GFSR and twisted GFSR generators in a similar way.

A different way of measuring the uniformity of the distribution of a set of points in $t$ dimensions is through the notion of *discrepancy*. Consider the set $\Omega_t(N) = \{\boldsymbol{u}_n = (u_n, \ldots, u_{n+t-1}),\ 0 \leq n \leq N-1\}$. For any box of the form $R = \prod_{j=1}^{t}[\alpha_j,\ \beta_j)$, with $0 \leq \alpha_j < \beta_j \leq 1$, let $I(R)$ be the cardinality of $\Omega_t(N) \cap R$, and $V(R) = \prod_{j=1}^{t}(\beta_j - \alpha_j)$ be the volume of $R$. If $\mathcal{R}$ is the set of all such regions $R$, then

$$D_N^{(t)} = \max_{R \in \mathcal{R}} |V(R) - I(R)/N|$$

is the $t$-dimensional *(extreme) discrepancy* for the set $\Omega_t(N)$. With the additional condition $\alpha_j = 0$ for all $j$, we obtain a variant called the *star discrepancy*.

The discrepancy of a "truly random" sequence should be approximately in $O(N^{-1/2})$ (Niederreiter 1992b). If true randomness is to be imitated, the set of points that are used during a simulation should then have a discrepancy approximately of that order. Note that a too low discrepancy is no better than a too high discrepancy. However, following our heuristic argument given in the introduction, we might seek a very low discrepancy for $N$ equal to the period length of the generator, and use only a negligible fraction of the period during the simulation, hoping that the discrepancy will be in the "right order" over that portion. Niederreiter (1992b) gives general discrepancy bounds for several classes of generators, mostly for $N = \rho$. However, no efficient algorithm is available for computing the discrepancy exactly, except for a few special cases. This is a strong limitation for its practical utilization.

## 5. NONLINEAR GENERATORS

Many authors argue that since the structure of linear sequences is too regular, *nonlinear* generators should be used instead (Eichenauer-Herrmann 1992; Niederreiter 1992a; Niederreiter 1992b). Nonlinearity can be introduced by either (a) using a linear-type generator but transforming the state nonlinearly to produce the output, or (b) constructing a generator based on a nonlinear recurrence.

A simple example of (a) is the *explicit inversive generator* of Eichenauer-Herrmann (1993): take $x_n = an + c$, for $n \geq 0$, where $a \neq 0$ and $c$ are in $\mathbb{Z}_m$, $m$ prime, $z_n = x_n^{-1} = (an+c)^{m-2} \bmod m$, and $u_n = z_n/m$. The period is $\rho = m$ and it can be shown that in all dimensions $t \leq m-2$, the set $\Omega_t$ generates the complete lattice $\mathbb{Z}^t/m$. Niederreiter (1992a) and Niederreiter (1994a) shows that every hyperplane in $\mathbb{R}^t$ contains at most $t$ points from the set $\Omega_t$. He also obtains discrepancy bounds which have the same asymptotic orders as the discrepancy of truly random sequences.

Other variants of inversive nonlinear generators have been proposed and studied by Eichenauer et al. (1987), Eichenauer, Lehn, and Topuzŏglu (1988), Eichenauer-Herrmann and Niederreiter (1992), Eichenauer-Herrmann (1992), Eichenauer-Herrmann and Grothe (1992), Eichenauer-Herrmann (1994), Eichenauer-Herrmann and Ickstadt (1994), Niederreiter (1992b), Niederreiter (1994a), Niederreiter (1994b), and the references given there. See also L'Ecuyer (1994c). Most of them enjoy similar nice theoretical properties: they avoid the planes and have the right asymptotic orders of magnitude for their discrepancies.

Several nonlinear generators have also been proposed in the field of cryptology. The best known is perhaps the BBS generator, proposed by Blum, Blum, and Schub (1986). It evolves according to:

$$x_n = x_{n-1}^2 \bmod m,$$

where $m$ is the product of two distinct $k$-bit primes, both congruent to 3 modulo 4, and $\gcd(x_0, m) = 1$. At each step, the generator outputs the last $\nu$ bits of $x_n$, where $\nu$ is in the order of $\log(k)$. Under the assumption that factoring is hard, and that $m$ and $x_0$ are chosen somewhat "randomly", it is proven that no polynomial-time (in $k$) statistical test can distinguish (in some specific sense) a BBS generator from a truly random one. This means that for large enough $k$, the generator should behave very nicely from a statistical point of view. This and other cryptographic generators have been studied empirically by L'Ecuyer and Proulx (1989) and Boucher (1994). The results are that BBS performs much better than its competitors but that $k$ should be taken relatively large (say over 500), which makes (a software implementation of) the generator too slow for many practical simulation applications.

A common property of several classes of proposed nonlinear generators is that they behave rather well (in general) in terms of asymptotic discrepancy. However, specific well-tested parameter values with fast implementations are currently not available. The fact that the points do not lie in hyperplanes does not preclude the presence of another (nonlinear, perhaps more sneaky) structure. Moreover, it seems that the discrepancy bounds that are available are sometimes rather wide (the upper bound is in some cases larger than 1) when computed for specific parameter values of reasonable sizes. Therefore, practically speaking, the arguments about the right order of discrepancy and that the points avoid the planes are perhaps not as definitive as they may appear. This question is still open and nonlinear generators certainly deserve much further investigation.

## ACKNOWLEDGMENTS

This work has been supported by NSERC-Canada grant # OGP0110050 and FCAR-Québec grant # 93-ER-1654.

## REFERENCES


Blum, L., M. Blum., and M. Schub. 1986. A simple unpredictable pseudo-random number generator. *SIAM J. Computing*, 15(2):364–383. Preliminary version published in *Proceedings of CRYPTO'82, 61–78*.

Boucher, M. 1994. La génération pseudo-aléatoire cryptographiquement sécuritaire et ses considérations pratiques. Master's thesis, Département d'I.R.O., Université de Montréal.

Bratley, P., B. L. Fox., and L. E. Schrage. 1987. *A Guide to Simulation*. second ed. New York: Springer-Verlag.

Compagner, A. 1991. The hierarchy of correlations in random binary sequences. *Journal of Statistical Physics*, 63:883–896.

Couture, R., and P. L'Ecuyer. 1994a. On the lattice structure of certain linear congruential sequences related to AWC/SWB generators. *Mathematics of Computation*, 62(206):798–808.

Couture, R., and P. L'Ecuyer. 1994b. Orbits and lattices for linear random number generators with composite moduli. *Mathematics of Computation*. Submitted.

Couture, R., P. L'Ecuyer., and S. Tezuka. 1993. On the distribution of $k$-dimensional vectors for simple and combined Tausworthe sequences. *Mathematics of Computation*, 60(202):749–761, S11–S16.

Eichenauer, J., H. Grothe., J. Lehn., and A. Topuzŏglu. 1987. A multiple recursive nonlinear congruential pseudorandom number generator. *Manuscripta Mathematica*, 59:331–346.

Eichenauer, J., J. Lehn., and A. Topuzŏglu. 1988. A nonlinear congruential pseudorandom number generator with power of two modulus. *Mathematics of Computation*, 51(184):757–759.

Eichenauer-Herrmann, J. 1992. Inversive congruential pseudorandom numbers: A tutorial. *International Statistical Reviews*, 60:167–176.

Eichenauer-Herrmann, J. 1993. Statistical independence of a new class of inversive congruential pseudorandom numbers. *Mathematics of Computation*, 60:375–384.

Eichenauer-Herrmann, J. 1994. Improved lower bounds for the discrepancy of inversive congruential pseudorandom numbers. *Mathematics of Computation*, 62(206):783–786.

Eichenauer-Herrmann, J., and H. Grothe. 1992. A new inversive congruential pseudorandom number generator with power of two modulus. *ACM Transactions on Modeling and Computer Simulation*, 2(1):1–11.

Eichenauer-Herrmann, J., and K. Ickstadt. 1994. Explicit inversive congruential pseudorandom numbers with power of two modulus. *Mathematics of Computation*, 62(206):787–797.

Eichenauer-Herrmann, J., and H. Niederreiter. 1992. Lower bounds for the discrepancy of inversive congruential pseudorandom numbers with power-of-two modulus. *Mathematics of Computation*, 58:775–779.

Ferrenberg, A. M., D. P. Landau., and Y. J. Wong. 1992. Monte Carlo simulations: Hidden errors from "good" random number generators. *Physical Review Letters*, 69(23):3382–3384.

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

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

James, F. 1990. A review of pseudorandom number generators. *Computer Physics Communications*, 60:329–344.

Knuth, D. E. 1981. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. second ed., volume 2. Addison-Wesley.

L'Ecuyer, P. 1990. Random numbers for simulation. *Communications of the ACM*, 33(10):85–97.

L'Ecuyer, P. 1992. Testing random number generators. In *Proceedings of the 1992 Winter Simulation Conference*, 305–313. IEEE Press.

L'Ecuyer, P. 1994a. Bad lattice structures for vectors of non-successive values produced by some linear recurrences. Submitted.

L'Ecuyer, P. 1994b. Combined multiple recursive generators. Submitted.

L'Ecuyer, P. 1994c. Uniform random number generation. *Annals of Operation Research*. To appear.

L'Ecuyer, P., F. Blouin., and R. Couture. 1993. A search for good multiple recursive random number generators. *ACM Transactions on Modeling and Computer Simulation*, 3(2):87–98.

L'Ecuyer, P., and S. Côté. 1991. Implementing a random number package with splitting facilities. *ACM Transactions on Mathematical Software*, 17(1):98–111.

L'Ecuyer, P., and R. Couture. 1994. An implementation of the lattice and spectral tests for linear congruential and multiple recursive generators. Submitted.

L'Ecuyer, P., and R. Proulx. 1989. About polynomial-time "unpredictable" generators. In *Proceedings of the 1989 Winter Simulation Conference*, 467–476. IEEE Press.

L'Ecuyer, P., and S. Tezuka. 1991. Structural properties for two classes of combined random number generators. *Mathematics of Computation*, 57(196):735–746.

Marsaglia, G. 1985. A current view of random number generators. In *in Computer Science and Statistics, Sixteenth Symposium on the Interface*, 3–10, North-Holland, Amsterdam. Elsevier Science Publishers.

Marsaglia, G., B. Narasimhan., and A. Zaman. 1990. A random number generator for PC's. *Computer Physics Communications*, 60:345–349.

Marsaglia, G., and L.-H. Tsay. 1985. Matrices and the structure of random number sequences. *Linear Algebra and its Applications*, 67:147–156.

Marsaglia, G., and A. Zaman. 1991. A new class of random number generators. *The Annals of Applied Probability*, 1:462–480.

Marsaglia, G., A. Zaman., and W. W. Tsang. 1990. Towards a universal random number generator. *Statistics and Probability Letters*, 8:35–39.

Matsumoto, M., and Y. Kurita. 1988. The fixed point of an $m$-sequence and local non-randomness. Technical Report 88-027, Department of Information Science, University of Tokyo.

Matsumoto, M., and Y. Kurita. 1992. Twisted GFSR generators. *ACM Transactions on Modeling and Computer Simulation*, 2(3):179–194.

Matsumoto, M., and Y. Kurita. 1994. Twisted GFSR generators II. *ACM Transactions on Modeling and Computer Simulation*. To appear.

Niederreiter, H. 1991. Recent trends in random number and random vector generation. *Annals of Operations Research*, 31:323–345.

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

Niederreiter, H. 1992b. *Random Number Generation and Quasi-Monte Carlo Methods.* volume 63 of *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics.* Philadelphia: SIAM.

Niederreiter, H. 1994a. On a new class of pseudorandom numbers for simulation methods. *Journal of Computational and Applied Mathematics.* To appear.

Niederreiter, H. 1994b. Pseudorandom vector generation by the inversive method. *ACM Transactions on Modeling and Computer Simulation*, 4(2):191–212.

Ripley, B. D. 1990. Thoughts on pseudorandom number generators. *Journal of Computational and Applied Mathematics*, 31:153–163.

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

Tezuka, S. 1992. A unified view of long-period random number generators. Submitted for publication.

Tezuka, S. 1994. The $k$-dimensional distribution of combined gfsr sequences. *Mathematics of Computation*, 62(206):809–817.

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

Tezuka, S., P. L'Ecuyer., and R. Couture. 1994. On the add-with-carry and subtract-with-borrow random number generators. *ACM Transactions of Modeling and Computer Simulation*, 3(4):315–331.

Wang, D., and A. Compagner. 1993. On the use of reducible polynomials as random number generators. *Mathematics of Computation*, 60:363–374.

## AUTHOR BIOGRAPHY

**PIERRE L'ECUYER** is a professor in the department of "Informatique et Recherche Opérationnelle" (IRO), at the University of Montreal. He received a Ph.D. in operations research in 1983, from the University of Montreal. From 1983 to 1990, he was with the computer science department, at Laval University, Québec. His research interests are in Markov renewal decision processes, sensitivity analysis and optimization of discrete-event stochastic systems, random number generation, and discrete-event simulation in general. He is the Departmental Editor for the Simulation Department of *Management Science* and an Area Editor for the *ACM Transactions on Modeling and Computer Simulation*.