Version: November 27, 1995

ON THE LATTICE STRUCTURE OF THE ADD-WITH-CARRY AND SUBTRACT-WITH-BORROW RANDOM NUMBER GENERATORS

SHU TEZUKA
IBM Research, Tokyo Research Laboratory

PIERRE L'ECUYER
Université de Montréal

RAYMOND COUTURE
Université Laval

ABSTRACT: Marsaglia and Zaman recently proposed new classes of random number generators, called *add-with-carry* (AWC) and *subtract-with-borrow* (SWB), which are capable of quickly generating very long period (pseudo)-random number sequences using very little memory. We show that these sequences are essentially equivalent to linear congruential sequences with very large prime moduli. So, the AWC/SWB generators can be viewed as efficient ways of implementing such large linear congruential generators. As a consequence, the theoretical properties of such generators can be studied in the same way as for linear congruential generators, namely via the spectral and lattice tests. We also show how the equivalence can be exploited to implement efficient jumping ahead facilities for the AWC and SWB sequences. Our numerical examples illustrate the fact that AWC/SWB generators have extremely bad lattice structure in high dimensions.

CR Categories and Subject Descriptors: G.3 [**Probability and Statistics**]: *Random Number Generation*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Lattice structure, add-with-carry, subtract-with-borrow, linear congruential generators, spectral test

---

Author's Addresses:
S. Tezuka, 5-19 Sanbancho, Chiyodaku, Tokyo 102, Japan;
P. L'Ecuyer, Département d'IRO, U. de Montréal, C.P. 6128, Succ.A, Montréal, H3C 3J7, Canada;
R. Couture, Département d'informatique, Université Laval, Ste-Foy, G1K 7P4, Canada.

1. THE AWC AND SWB GENERATORS

Marsaglia and Zaman [12] proposed the following types of random number generators, called *add-with-carry* (AWC) and *subtract-with-borrow* (SWB). Let b, r, and s be positive integers, where b is called the *base* and $r > s$ are called the *lags*. The AWC generator is based on the recurrence

$$\begin{aligned} x_i &= (x_{i-s} + x_{i-r} + c_i) \bmod b, && (1) \\ c_{i+1} &= I(x_{i-s} + x_{i-r} + c_i \geq b), && (2) \end{aligned}$$

where c_i is called the *carry*, and I is the indicator function, whose value is 1 if its argument is true, and 0 otherwise. That generator is extremely fast, since it requires no multiplication, and the modulo operation can be performed by just subtracting b if and only if $x_{i-s} + x_{i-r} + c_i \geq b$. The maximum possible (or full) period is $b^r + b^s - 2$. It is attained when $M = b^r + b^s - 1$ is prime and b is a primitive root modulo M (see [12]). For example, one can take b around 2^{31} and r around 20, yielding a period of approximately 2^{620} if the full period conditions are satisfied. This goes much beyond the requirements of most applications.

To produce values $\{u_i\}$ whose distribution (hopefully) approximates the $U(0,1)$ distribution, one can use $L \leq r$ successive values of x_j to produce one u_i as follows [3]:

$$u_i = \sum_{j=1}^{L} x_{Li-j+1} b^{-j}. \tag{3}$$

Assuming that L is relatively prime to $M - 1$, the sequences $\{u_i\}$ and $\{x_i\}$ have the same periods. If b is small, or if more precision is desired, take a larger L. If b is large enough (e.g., a large power of two), one can just take $L = 1$. Here, the digits of u_i are filled up from the least significant to the most significant one. The sequence $\{u_i\}$ defined by (3) is an analogue of the Tausworthe sequence [13, 16]. For the latter, the digits of u_i are filled up by a linear feedback shift register sequence modulo two (i.e., $b = 2$). The difference with (1) is the presence of the carry and the fact that b is not necessarily equal to two.

The AWC has a variant called *complementary* AWC, or AWC-c, based on:

$$\begin{aligned} x_i &= (2b - 1 - x_{i-s} - x_{i-r} - c_i) \bmod b && (4) \\ &= (-x_{i-s} - x_{i-r} - c_i - 1) \bmod b, \\ c_{i+1} &= I(x_{i-s} + x_{i-r} + c_i \geq b). && (5) \end{aligned}$$

The SWB also comes in two flavors, which we will call SWB-I and SWB-II, based on the recurrences:

$$\begin{aligned} x_i &= (x_{i-s} - x_{i-r} - c_i) \bmod b, && (6) \\ c_{i+1} &= I(x_{i-s} - x_{i-r} - c_i < 0), && (7) \end{aligned}$$

and

$$
\begin{aligned}
x_i &= (x_{i-r} - x_{i-s} - c_i) \bmod b, && (8)\\
c_{i+1} &= I(x_{i-r} - x_{i-s} - c_i < 0), && (9)
\end{aligned}
$$

respectively. Here, c_i is called the *borrow*.

We will use the general notation AWC/SWB to refer to any of those four variants. For each of them, the maximum possible period is $M - 1$, achieved when M is prime and b is a primitive root modulo M, where the value of M depends on the variant, as shown in Table 1.

Table 1: Values of M for the AWC/SWB variants.

	M
AWC	$b^r + b^s - 1$
AWC-c	$b^r + b^s + 1$
SWB-I	$b^r - b^s + 1$
SWB-II	$b^r - b^s - 1$

In all cases, the u_i's can be produced from the x_j's as in (3). For a full period AWC/SWB generator, the x_i's are provably almost equidistributed in up to r dimensions, i.e., among all (overlapping) r-dimensional vectors of successive values of x_i's, over the whole period, every r-dimensional vector with components in $\{0, \ldots, b - 1\}$ appears exactly once, except for a tiny percentage of exceptions [12].

The AWC/SWB methods can be viewed as slight modifications to the so-called *additive* or *subtractive* methods discussed in Knuth [4]. The only difference in implementation is that for the latter, there is no carry or borrow ($c_i = 0$ for all i). But in terms of period length, this makes an enormous difference: for example, if $b = 2^e$ (a power of two), the maximal period lengths for the additive and subtractive generators are only $(2^r - 1)2^{e-1} \approx 2^{r+e-1}$, which falls way short of $b^r + b^s - 2 \approx 2^{re}$, unless $e = 1$. The additive and subtractive generators belong to the more general class of *lagged-Fibonacci* generators. See [5, 11] for more details.

Marsaglia and Zaman [12] give a list of parameter sets for SWB-I generators, for which the order of b modulo M is very large or near the maximum. Those generators do not have full period, but a large period anyway. Finding full period generators with a very large period is hard, because checking the primitivity requires the factorization of $M - 1$, which is a difficult task in practice when M is large. For example, for M around 2^{1000}, the best factorization programs currently available typically cannot factorize $M - 1$ in reasonable time.

In this paper, we analyze the structure of the sequence $u_i,\ i = 1, 2, ...$, produced by an AWC/SWB generator. That sequence turns out to be practically the same as the sequence produced by a linear congruential generator (LCG). More precisely, we have the following. Let $s_i = (x_{i-r+1}, \ldots, x_i, c_{i+1})$ be the *state* of the AWC/SWB generator at step i. Equation (3) transforms the state s_{Li} into the uniform variate u_i. Suppose that M (given in Table 1) is prime and let b^* be the multiplicative inverse of b modulo M, i.e., such that $b^*b \bmod M = 1$. That inverse can be computed easily as $b^* = b^{M-2} \bmod M$. Consider the following LCG with modulo M and multiplier $A = b^*$:

$$
\begin{aligned}
X_i &= AX_{i-1} \bmod M, && (10)\\
v_i &= X_i/M, && (11)\\
w_i &= v_{Li} \;=\; X_{Li}/M. && (12)
\end{aligned}
$$

Our main result is:

THEOREM **1.** *Let $\{u_i,\ i \geq 0\}$ be the sequence (3) produced by an AWC/SWB generator, while $\{w_i,\ i \geq 0\}$ is the sequence produced by (12). If s_0 and X_0 correspond, then, for all $i \geq r$, the (fractional) digital expansions in base b of u_i and w_i have the same first L digits. In other words, one has*

$$u_i = b^{-L}\lfloor b^L w_i \rfloor. \tag{13}$$

The condition "s_0 and X_0 correspond" means that the two sequences must have corresponding initial seeds. Otherwise, (13) will hold after an appropriate shift of one of the two sequences. To be more precise, the theorem holds under the condition that $X_0 = \varphi(s_0) \neq 0$, where φ is defined by equation (21) for AWC, and in equations (24), (22), and (23) for the AWC-c, SWB-I, and SWB-II generators, respectively. Equation (13) means that u_i is a truncated version of w_i: only the first L fractional digits in base b are kept, the others are chopped off. As a consequence, $|u_i - w_i| \leq b^{-L}$. So, the sequences (3) and (12) are the same, if they have corresponding initial seeds, up to a precision of b^{-L}. For example, it could be reasonable to take $b > 2^{30}$ and $L = 2$, in which case the first 60 bits of u_i and w_i will be the same. For all practical purposes, considering the limited precision of floating point numbers on computers, one can then safely assume that $u_i = w_i$.

We call (10–12) the *LCG representation* of the corresponding AWC/SWB generator. In Section 2, we explain how one can switch between the AWC (or SWB) and LCG representations, and we prove our main theorem. Along the way, we prove some properties which were stated and illustrated by examples, but not proved, in the paper of Marsaglia and Zaman [12]. In Section 3, we explain how switching between these representations allows one to jump ahead (efficiently) in the sequence of the AWC/SWB generator. For a theoretical evaluation of the structural properties of an AWC/SWB generator, one can study the lattice structure of its LCG representation. We discuss that in Section 4. In Section 5, we illustrate those properties with numerical examples. Some of them are generators taken from Marsaglia and Zaman [12]. It turns out that all the generators examined perform extremely badly, in the spectral test, in dimensions $r+1$ and higher. At the end of Section 4, we show that this holds in general: for all AWC/SWB generators with $L = 1$, the distance between the hyperplanes in the lattice of the associated LCG is at least $1/\sqrt{3}$ in all dimensions larger than r. The latter result is from Couture and L'Ecuyer [2]. The present paper is a detailed and expanded version of [15] and the main idea of Theorem 1 was first stated in [14].

2. MAPPINGS BETWEEN THE DIFFERENT REPRESENTATIONS AND PROOF OF THE MAIN THEOREM

In this section, we examine the state spaces for both the AWC (or SWB) and LCG representations, and define mappings between those spaces. We then use those mappings to prove our main result. We will make the development with the AWC generator. The same development, with minor changes that we will explain afterwards, can be made for the three other types of generators. We use $\mathbb{Z}$ to represent the ring of integers and, for any positive integer m, $\mathbb{Z}_m$ represents the ring of integers modulo m.

Since b^* is the multiplicative inverse of b, one has $X_{i-1} = bX_i \bmod M$. ¿From that, X_i is now defined for all integers $i \in \mathbb{Z}$. Dividing by M, we obtain

$$v_{i-1} \;=\; bv_i \bmod 1 \tag{14}$$

for all $i \in \mathbb{Z}$. This corresponds to a left shift of the digital expansion in base b by one digit and dropping off the non-fractional part. Let x_i be the coefficient of b^{-1} in the digital expansion in base b of v_i. Then, $bv_i = v_{i-1} + x_i$ and one can write

$$v_i \;=\; X_i/M \;=\; \sum_{j=1}^{\infty} x_{i-j+1} b^{-j} \;=\; .x_i x_{i-1} x_{i-2} \cdots . \tag{15}$$

Now, for all $i \in \mathbb{Z}$, define

$$\alpha_i \;=\; -x_i + x_{i-r} + x_{i-s}; \tag{16}$$

$$c_i \;=\; \sum_{j=1}^{\infty} \alpha_{i-j} b^{-j}; \tag{17}$$

$$A_i \;=\; \sum_{j=1}^{r} x_{i-j} b^{r-j} + \sum_{j=1}^{s} x_{i-j} b^{s-j}.$$

LEMMA 1. *The sequence $\{(x_i, c_{i+1})\}$ obeys the recurrence (1–2).*

PROOF. ¿From (15), one has

$$
\begin{aligned}
X_i &= Mv_i = (b^r + b^s - 1)\left(\sum_{j=1}^{\infty} x_{i-j+1} b^{-j}\right) \\
&= \sum_{j=1}^{r} x_{i-j+1} b^{r-j} + \sum_{j=1}^{s} x_{i-j+1} b^{s-j} + \sum_{j=1}^{\infty} \alpha_{i-j+1} b^{-j} \\
&= A_{i+1} + c_{i+1}.
\end{aligned}
$$

Since X_i and A_{i+1} are integers, c_{i+1} must be an integer. ¿From (16–17) and the fact that $0 \le x_i \le b-1$ for each i, it is easily seen that $-1 < c_i < 2$. So, each c_i is 0 or 1. ¿From the definition of c_i, one has

$$
bc_{i+1} = \sum_{j=1}^{\infty} \alpha_{i-j+1} b^{-j+1} = \alpha_i + c_i = -x_i + x_{i-r} + x_{i-s} + c_i. \tag{18}
$$

It then follows that $c_{i+1} = 0$ if and only if $x_i = x_{i-r} + x_{i-s} + c_i$. Otherwise, $c_{i+1} = 1$ and $x_i = x_{i-r} + x_{i-s} + c_i - b$. Since $0 \le x_i < b$, we have $c_{i+1} = 1$ if and only if $x_{i-r} + x_{i-s} + c_i > b$. In summary,

$$
\begin{aligned}
x_i &= (x_{i-r} + x_{i-s} + c_i) \bmod b; \\
c_{i+1} &= (x_{i-r} + x_{i-s} + c_i) \operatorname{div} b,
\end{aligned}
$$

for each i, which corresponds to the AWC recurrence (1–2). ∎

So, for X_0 fixed, with $v_0 = X_0/M = .x_0 x_{-1} \ldots$ and c_0 defined by (17), (1–2) is an equivalent way of defining (uniquely) $c_1, x_1, c_2, x_2, \ldots$. The sequence $\{s_i\}$ is then defined. In other words, reading the fractional expansion in base b of X_0/M (for any integer X_0 in $\{1, \ldots, M\}$) gives the AWC sequence $\{x_i\}$ in reverse order.

The LCG (10) is defined over the state space $\mathbb{Z}_M$. The AWC generator, on the other hand, has state space $S = \mathbb{Z}_b^r \times \mathbb{Z}_2$, the state at step i being $s_i = (x_{i-r+1}, \ldots, x_i, c_{i+1})$. We now define two mappings $\psi : \mathbb{Z}_M \to S$ and $\varphi : S \to \mathbb{Z}_M$ such that $\psi(X_i) = s_i$ and $\varphi(s_i) = X_i$. These mappings will make the correspondence between the LCG and AWC generators. Clearly, they will not be one-to-one, because $\mathbb{Z}_M$ has cardinality $M = b^r + b^s - 1$, while S has cardinality $2b^r$. However, we will show that the restriction of φ to $S_0 = \psi(\mathbb{Z}_M)$ is one-to-one and has ψ as an inverse. The set S_0 corresponds to the set of "periodic seed vectors" defined in Section 4.5 of Marsaglia and Zaman [12].

For any $X = X_0 \in \mathbb{Z}_M$, again let $x_0, x_{-1}, \ldots,$ be the digits of the fractional expansion in base b of X_0/M:

$$
X_0/M = \sum_{j=1}^{\infty} x_{1-j} b^{-j}. \tag{19}
$$

¿From (18) with $i = 0$ and (2), c_0 and c_1 can be recovered as follows: $c_0 = I[(x_{-r} + x_{-s} - x_0 + 1) \bmod b = 0]$ and $c_1 = I[x_{-r} + x_{-s} + c_0 \geq b]$. Define

$$\psi(X_0) = (x_{1-r}, \ldots, x_0, c_1). \tag{20}$$

This defines a mapping $\psi : Z_M \to S$. Applying that mapping to X_i one obtains $\psi(X_i) = (x_{i-r+1}, \ldots, x_i, c_{i+1}) = s_i$ for all i.

Conversely, for any $\xi = (x_{1-r}, \ldots, x_0, c_1) \in S$, define

$$\varphi(\xi) = \left(\sum_{j=1}^{r} x_{1-j} b^{r-j} + \sum_{j=1}^{s} x_{1-j} b^{s-j} \ + c_1 \right) \bmod M. \tag{21}$$

If $\psi(X_0) = \xi$, then,

$$\begin{aligned} X_0 &= Mv_0 = (b^r + b^s - 1) \left(\sum_{j=1}^{\infty} x_{1-j} b^{-j} \right) \\ &= \sum_{j=1}^{r} x_{1-j} b^{r-j} + \sum_{j=1}^{s} x_{1-j} b^{s-j} \ + c_1 \\ &= \varphi(\xi). \end{aligned}$$

We just proved the following:

Lemma 2. *For all $X \in \mathbb{Z}_M$, one has $\varphi(\psi(X)) = X$. This also implies that $\psi(\varphi(\xi)) = \xi$ for all $\xi \in S_0$.*

So, if $s_0 = \psi(X_0)$, then $s_i = \psi(X_i) \in S_0$ and $X_i = \varphi(s_i)$ for all $i \geq 0$, where the sequences $\{s_i\}$ and $\{X_i\}$ follow the recurrences (1–2) and (10) respectively. This means that these two sequences are in fact just two representations of the same sequence.

Observe that the largest possible value for the expression inside the parentheses in (21) is M (when $\xi = (b-1, \ldots, b-1, 1)$), and this is the only case where the "mod M" operation does something. This corresponds to the case where $X/M = 1$ in (19). Therefore, two values of $\xi \in S$ have $\varphi(\xi) = 0$, namely $\xi = (0, \ldots, 0, 0)$ and $\xi = (b-1, \ldots, b-1, 1)$. These two states must clearly be avoided. All other states in S are mapped to non-zero values in $\mathbb{Z}_M$.

The following facts are well known [8]. If b is primitive modulo M, the LCG has two cycles: one of length 1 (the state $X = 0$) and the other of length $M - 1$. Of course, only the latter is interesting for practical use. When b is not primitive modulo M, then the order of b modulo M is a divisor ρ of $M - 1$ (that is, ρ is the smallest positive integer i such that $b^i \bmod M = 1$), and the LCG has $(M-1)/\rho$ subcycles of length ρ, in addition to the trivial cycle containing state 0.

If $s_0 \in S_0$, then $\{x_i\}$ is purely periodic with period ρ. What happens when s_0 is not in S_0 ? The next lemma shows that if $s_0 \neq (b-1, \ldots, b-1, 1)$, the sequence $\{s_i\}$ will reach S_0 after at most r steps. ¿From then on, the previous lemmas and the above discussion will apply. Note that if $s_0 = (b-1, \ldots, b-1, 1)$, then $s_i = s_0 \notin S_0$ for all i.

Lemma **3.** *For any $s_0 \in S$, $s_0 \neq (b-1, \ldots, b-1, 1)$, one has $s_i \in S_0$ for all $i \geq r$.*

Proof. Let $s_0 = (x_{-r+1}, \ldots, x_0, c_1)$ and

$$\begin{aligned} X_0 &= \varphi(s_0) \\ &= c_1 + \sum_{j=1}^{r} x_{1-j} b^{r-j} + \sum_{j=1}^{s} x_{1-j} b^{s-j} \\ &= \sum_{j=0}^{r} y_{1-j} b^{r-j}. \end{aligned}$$

Here, we could get rid of the "mod M" in (21) because $s_0 \neq (b-1, \ldots, b-1, 1)$. Now, it is easily seen that $x_1, \ldots, x_r, c_{r+1}$ can be expressed as functions of $y_{1-r}, \ldots, y_0, y_1$ only:

$$\begin{aligned} x_1 &= (x_{1-r} + x_{1-s} + c_1) \bmod b \ = \ y_{1-r}, \\ &\vdots \\ x_s &= (x_{s-r} + x_0 + c_s) \bmod b \ = \ y_{s-r}, \\ x_{s+1} &= (x_{s-r+1} + x_1 + c_{s+1}) \bmod b \ = \ (y_{s-r+1} + x_1) \bmod b, \\ c'_{s+2} &= I[y_{s-r+1} + x_1 \geq b], \\ x_{s+2} &= (x_{s-r+2} + x_2 + c_{s+2}) \bmod b \ = \ (y_{s-r+2} + x_2 + c'_{s+2}) \bmod b, \\ &\vdots \\ x_r &= (x_0 + x_{r-s} + c_r) \bmod b \ = \ (y_0 + x_{r-s} + c'_r) \bmod b, \\ c'_{r+1} &= I[y_0 + x_{r-s} + c'_r \geq b], \\ c_{r+1} &= y_1 + c'_{r+1}. \end{aligned}$$

Then, $s_r = (x_1, \ldots, x_r, c_{r+1})$ is uniquely determined by X_0, and must therefore be equal to $s_r = \psi(X_r) \in S_0$. ∎

Proof of Theorem 1. Assuming that $X_0 = \varphi(s_0) \neq 0$, the result follows directly from Lemmas 1–3 and the fact that (3) is a truncation of the digital expansion in base b of $w_i = v_{Li}$. ∎

Similar proofs of the three lemmas and of the main theorem can be made for the other AWC/SWB variants, with the following minor changes.

For SWB-I, define $M = b^r - b^s + 1$, let $c_0 = I[(y_{-r} - y_{-s} + y_0 + 1) \bmod b = 0]$ and $c_1 = I[y_{-s} - y_{-r} - c_0 < 0]$ in the definition of ψ, and

$$\varphi(\xi) = \left(c_1 + \sum_{j=1}^{r} y_{1-j} b^{r-j} - \sum_{j=1}^{s} y_{1-j} b^{s-j} \right) \bmod M. \tag{22}$$

For SWB-II, define $M = b^r - b^s - 1$, let $c_0 = I[(y_{-s} - y_{-r} + y_0 + 1) \bmod b = 0]$ and $c_1 = I[y_{-r} - y_{-s} - c_0 < 0]$ in the definition of ψ, and

$$\varphi(\xi) = \left(-c_1 + \sum_{j=1}^{r} y_{1-j} b^{r-j} - \sum_{j=1}^{s} y_{1-j} b^{s-j} \right) \bmod M. \tag{23}$$

Here, the parenthesized expression in (23) can be larger than M, namely when $\xi = (b-1, \ldots, b-1, 0)$.

For AWC-c, define $M = b^r + b^s + 1$, let $c_0 = I[(y_{-r} + y_{-s} + y_0 + 2) \bmod b = 0]$ and $c_1 = I[y_{-r} + y_{-s} + c_0 \geq b]$ in the definition of ψ, and

$$\varphi(\xi) = 1 + c_1 + \sum_{j=1}^{r} y_{1-j} b^{r-j} + \sum_{j=1}^{s} y_{1-j} b^{s-j}. \tag{24}$$

Here, the results are slightly different. No state ξ in S has $\varphi(\xi) = 0$. One has $\varphi(0, \ldots, 0, 0) = 1$ and $\varphi(b-1, \ldots, b-1, 1) = M - 1$. On the other hand, one has $\psi(0) = (0, \ldots, 0, 0)$ if $b \neq 2$, and $\psi(0) = (0, \ldots, 0, 1)$ if $b = 2$. Therefore, Lemma 2 does not apply for $X = 0$. However, for $X \neq 0$, the proof goes through.

3. JUMPING AHEAD

To jump ahead efficiently in the sequence $\{s_i\}$, one can exploit the one-to-one mapping between $\mathbb{Z}_M$ and S_0 defined in the previous section. Given s_i, one can precompute $s_{i+\nu}$, for any (large) positive integer ν, by the following three steps:

$$\begin{aligned} X_i &:= \varphi(s_i); \\ X_{i+\nu} &:= (A^\nu \bmod M) X_i \bmod M; \\ s_{i+\nu} &:= \psi(X_{i+\nu}). \end{aligned}$$

The integer $(A^\nu \bmod M)$ can be computed efficiently by a divide-to-conquer algorithm as explained in [5].

Such jumping-ahead facilities are necessary for breaking up the sequence into long disjoint subsequences. This is useful, for example, to obtain multiple streams of random numbers in a simulation (which is required by some variance reduction techniques) or to generate subsequences in parallel on multiple processor systems.

4. LATTICE STRUCTURE AND SPECTRAL TEST

It is well known that linear congruential generators have a lattice structure which can be analyzed through the Beyer and spectral tests [4, 5, 7, 9]. More precisely, suppose we construct points in $[0,1)^t$ by taking t successive values produced by the generator:

$$\boldsymbol{w}_{t,i} = (w_i, \ldots, w_{i+t-1}).$$

Let T_t be the set of all such points, for all possible initial states $X_0 \in \mathbb{Z}_M$:

$$T_t = \{\boldsymbol{w}_{t,i} = (w_i, \ldots, w_{i+t-1}) \mid i \geq 0,\ X_0 \in \mathbb{Z}_M\}.$$

Then T_t is the intersection of a lattice L_t with the unit hypercube $[0,1)^t$. The *Beyer quotient* is defined as the ratio q_t of the lengths of the shortest and longest vectors in a *Minkowski-Reduced Basis* of that lattice. A value of q_t close to one indicates that the points of L_t are rather "uniformly" distributed, while a very small value indicates the opposite (a "bad" lattice structure). The lattice structure also means that the points lie in a set of equidistant parallel hyperplanes. Let d_t be the distance between those hyperplanes in dimension t. Generally speaking, we would like d_t to be as small as possible, because larger values of d_t (close to 1) mean thicker slices of space containing no points.

The LCG that produces the points T_t is in fact equivalent to

$$Y_i = \tilde{A} Y_{i-1} \bmod M, \tag{25}$$

$$w_i = Y_i / M, \tag{26}$$

where $Y_0 = X_0$ and $\tilde{A} = A^L \bmod M = b^{M-L-1} \bmod M$. If the multiplier $\tilde{A}$ above is replaced by its inverse $\tilde{A}^* = b^L \bmod M$, then it will produce the same sequence $\{w_i\}$, but in reverse order. Since the reverse sequence has the same lattice structure as the original one, applying the spectral or Beyer test with the multiplier b^L or A^L will yield the same results.

Consider now the points produced by an AWC or SWB generator:

$$\boldsymbol{u}_{t,i} = (u_i, \ldots, u_{i+t-1}),$$

assuming that $s_0 = \psi(X_0)$. It follows from Theorem 1 that $|u_i - w_i| < b^{-L}$. Therefore, the Euclidean distance between $\boldsymbol{w}_{t,i}$ and $\boldsymbol{u}_{t,i}$ is bounded by $b^{-L}\sqrt{t}$. If that bound is small with respect to the Euclidean distance d_t between hyperplanes, then the AWC or SWB generator inherits the lattice structure of the associated LCG, with some small (often negligible) added "noise" due to the truncation. We will examine specific numerical examples in the next section. For those examples, the values of q_t and d_t were computed using the software package described in L'Ecuyer and Couture [7].

The following result, which follows from the more general results given in [2], shows that AWC/SWB generators with $L = 1$ always have a bad lattice structure in dimensions larger than r. We give a simple proof here for completeness.

LEMMA 4. *For the LCG (10–11), one has* $d_t \geq 1/\sqrt{3}$ *for all* $t \geq r+1$.

PROOF. Consider the AWC generator (the proof is similar for the other variants). One has

$$X_{i-r} + X_{i-s} - X_i \;\equiv\; (b^r + b^s - 1)X_i \;\equiv\; MX_i \;\equiv\; 0 \pmod{M}.$$

So, by following the same reasoning as in Section xx of Knuth [4], it follows that the dual lattice has a vector of square length equal to 3, and the conclusion follows. ∎

5. NUMERICAL EXAMPLES

5.1 Example 1: A Small SWB Generator

Consider the SWB-I generator with $(b, s, r, L) = (2, 2, 9, 9)$. Here, $x_i = (x_{i-2} - x_{i-9} - c_i) \bmod 2$,

$$u_i = \sum_{j=1}^{9} x_{9i-j+1} 2^{-j},$$

and the period is $2^9 - 2^2 = 508$. Figure 1 shows a two-dimensional plot of the pairs of successive points (u_i, u_{i+1}) produced by this generator over its entire period. The starting values were $s_0 = (x_{-8}, ..., x_0, c_1) = (1, 0, ..., 0)$. This looks like a typical lattice structure of a (bad) LCG.

The LCG representation of that SWB generator is

$$Y_i = 170 Y_{i-1} \bmod 509; \qquad w_i = Y_i/509,$$

where $Y_i = X_{Li}$ and 170 is the inverse of $2^9 (= 3)$ modulo 509. Since u_i is just the truncated version of w_i, the points produced by the SWB generator do not form exactly a lattice, but it really takes sharp eyes see that the points in Figure 1 are not exactly aligned on the three lines. The approximation is quite good indeed.

If the multiplier 170 was replaced by 3, we would get the same graphic, but reflected with respect to the diagonal $u_i = u_{i+1}$. Hence, the points of the LCG representation will be on three lines of slope 3 instead of slope 1/3.

5.2 Example 2: A "Classroom" AWC Generator

We now examine the "classroom" AWC generator given in Section 7 of Marsaglia and Zaman [12], for which $(b, s, r, L) = (6, 2, 21, L)$. The sequence is defined by

$$u_i = \sum_{j=1}^{L} x_{Li-j+1} 6^{-j},$$

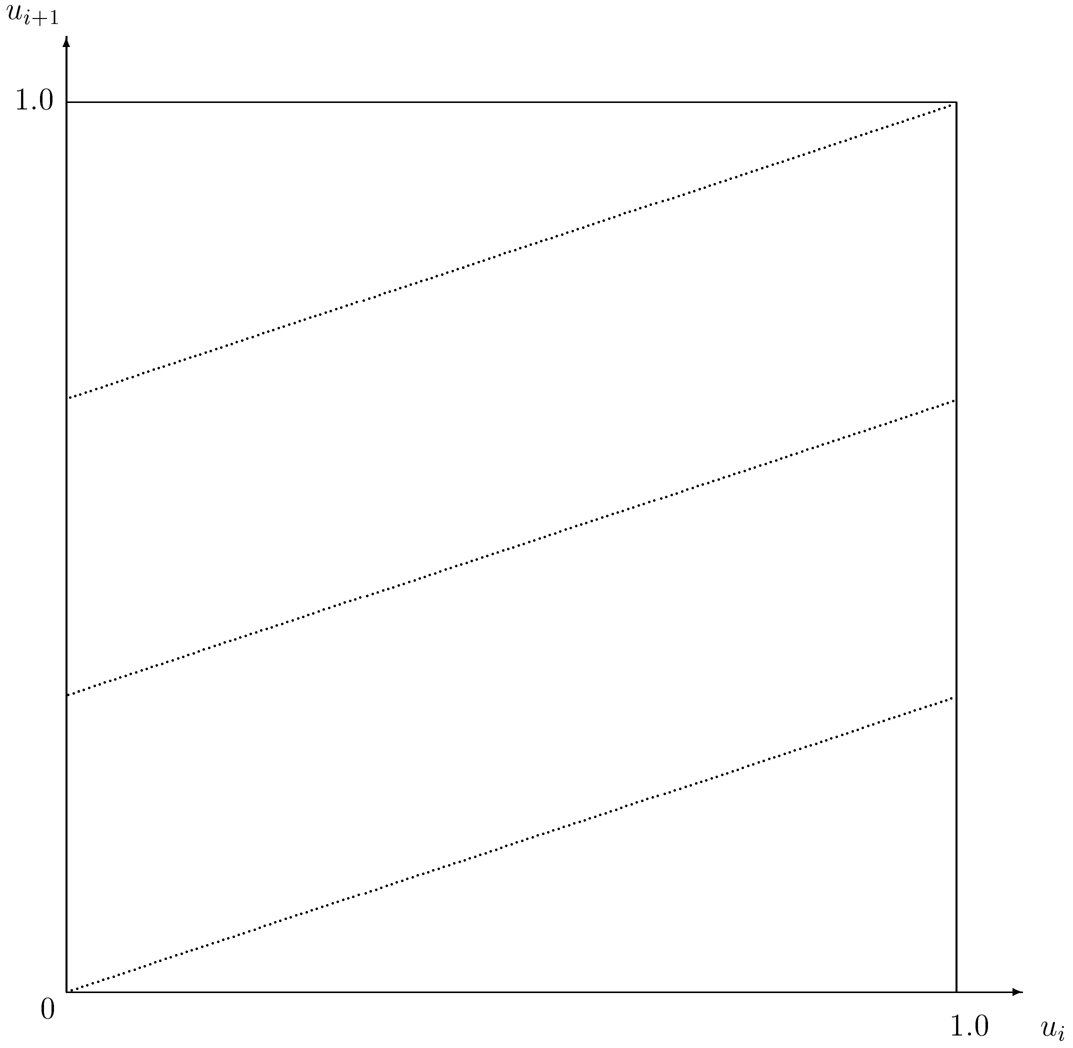

Figure 1: All pairs of successive points for the SWB generator of Example 1.

where x_i is generated by $x_i = (x_{i-21} + x_{i-2} + c_i) \bmod 6$. We will look at different values of L. Since $M = 6^{21} + 6^2 - 1 = 21,936,950,640,377,891$ is prime and $b = 6$ is a primitive root modulo M, the sequence of x_i's have period $M - 1$. When L is relatively prime to $M - 1$, the u_i's also have that same period. According to Marsaglia and Zaman [12], the x_i's, if used directly, could provide an excellent simulation of independent throws of a dice.

The LCG representation is given by

$$X_{Li} = Y_i = (6^*)^L Y_{i-1} \bmod M; \qquad w_i = Y_i/M.$$

The following values of L are relatively prime to $M - 1$: $L = 1, 3, 7, 9, 11, 17, 19$. For small L, like 1 or 3, the resolution is much too low and as a result, the LCG is not a good approximation of the AWC sequence. We have computed the values of q_t and d_t for the corresponding LCG's for the other values of L. The results are given in Table 2. For all those values of L, the lattice structure turns out to be quite bad in low dimensions. In fact, it is amazing to see how terrible are some of those multipliers in lower dimensions (e.g., for $L = 17$ and $L = 19$). The upper bound $6^{-L}\sqrt{t}$ on the noise is much smaller than the distance between hyperplanes, except for $L = 7, 9, 11$ in dimension 2 and $L = 7$ in dimension 3.

5.3 Example 3: A Larger SWB Generator

One SWB-I generator recommended by Marsaglia and Zaman (1991) has parameters $(b, s, r, L) = (2^{32}, 6, 21, 1)$. That generator does not have full period, it has 192 subcycles of period $(2^{666} - 2^{186})/3$ each (besides the two trivial cycles of period 1). The LCG representation has modulus $M = 2^{672} - 2^{192} + 1$ and multiplier $A = (2^{32})^* \bmod M = 2^{160} - 2^{640} \bmod M$.

We can study the lattice structure formed by the vectors of successive points in the union of all the subcycles (for a single cycle, the points do not necessarily form a lattice, but for the union of all cycles, they do). Table 3 gives the values of d_t and q_t for t up to 30. The bad behavior in dimensions larger than 21 is in accordance with Lemma 4. We recall that for dimensions smaller or equal to 21, the lattice structure of the associated LCG provides only limited information on the behavior of the AWC/SWB generator, because the truncation error is as large as the distance between the successive hyperplanes. But the small values of d_t for $t \leq 21$ agree with the fact that over the full period, the points are very evenly distributed over the unit hypercube.

5.4 Example 4: The RANMAR SWB Generator

James [3] recommends the SWB-I generator with parameters $(b, s, r, L) = (2^{32} - 5, 22, 43, 1)$. That generator is also given in [12] and used as a component of the combined generator proposed in [10]. Since b is primitive modulo $M = b^{43} - b^{22} + 1$, the (full) period length

Table 2: Beyer and spectral tests for Example 2.

L	7	9	11	17	19
q_2	3.572E-6	4.630E-3	0.167	7.662E-11	1.149E-13
q_3	1.000	2.171E-5	3.473E-6	9.926E-8	4.329E-12
q_4	1.251E-4	2.200E-5	1.216E-4	1.286E-4	1.673E-10
q_5	1.251E-4	4.692E-3	7.293E-4	0.167	6.434E-9
q_6	4.380E-3	4.440E-3	2.552E-2	0.205	2.453E-7
q_7	4.372E-3	0.959	6.143E-2	0.669	9.282E-6
q_8	4.372E-3	0.103	0.473	0.567	3.490E-4
q_9	0.153	0.103	0.550	0.750	1.305E-2
q_{10}	7.088E-2	0.222	0.740	0.477	0.476
q_{11}	7.070E-2	0.229	0.589	0.634	0.562
q_{12}	0.627	0.521	0.861	0.703	0.653
q_{13}	0.358	0.513	0.646	0.870	0.639
q_{14}	0.358	0.536	0.658	0.778	0.729
q_{15}	0.551	0.844	0.613	0.724	0.697
q_{16}	0.439	0.733	0.777	0.663	0.867
q_{17}	0.533	0.761	0.769	0.645	0.800
q_{18}	0.777	0.772	0.854	0.737	0.819
q_{19}	0.700	0.853	0.835	0.778	0.909
q_{20}	0.847	0.816	0.864	0.797	0.829
$1/m$					
d_2	3.572E-6	9.923E-8	1.654E-8	7.713E-4	1.992E-2
d_3	3.572E-6	4.570E-3	4.762E-3	7.713E-4	1.992E-2
d_4	2.856E-2	4.570E-3	4.762E-3	7.713E-4	1.992E-2
d_5	2.856E-2	4.570E-3	4.762E-3	7.713E-4	1.992E-2
d_6	2.856E-2	4.570E-3	4.762E-3	3.532E-3	1.992E-2
d_7	2.856E-2	4.570E-3	1.182E-2	4.998E-3	1.992E-2
d_8	2.856E-2	4.486E-2	1.182E-2	1.342E-2	1.992E-2
d_9	2.856E-2	4.486E-2	1.839E-2	1.526E-2	1.992E-2
d_{10}	5.573E-2	4.486E-2	2.243E-2	3.542E-2	1.992E-2
d_{11}	5.573E-2	4.486E-2	3.742E-2	3.542E-2	3.475E-2
d_{12}	5.573E-2	4.486E-2	3.904E-2	4.657E-2	4.608E-2
d_{13}	9.713E-2	6.428E-2	7.715E-2	5.185E-2	5.463E-2
d_{14}	9.713E-2	6.496E-2	7.715E-2	7.727E-2	6.441E-2
d_{15}	9.713E-2	6.652E-2	7.715E-2	7.981E-2	7.125E-2
d_{16}	0.100	9.129E-2	8.220E-2	0.104	8.138E-2
d_{17}	0.100	9.853E-2	9.245E-2	0.104	0.103
d_{18}	0.100	9.853E-2	0.102	0.106	0.103
d_{19}	0.120	0.104	0.109	0.114	0.105
d_{20}	0.120	0.114	0.115	0.123	0.117
6^{-L}	3.572E-6	9.923E-8	2.756E-9	5.908E-14	1.641E-15

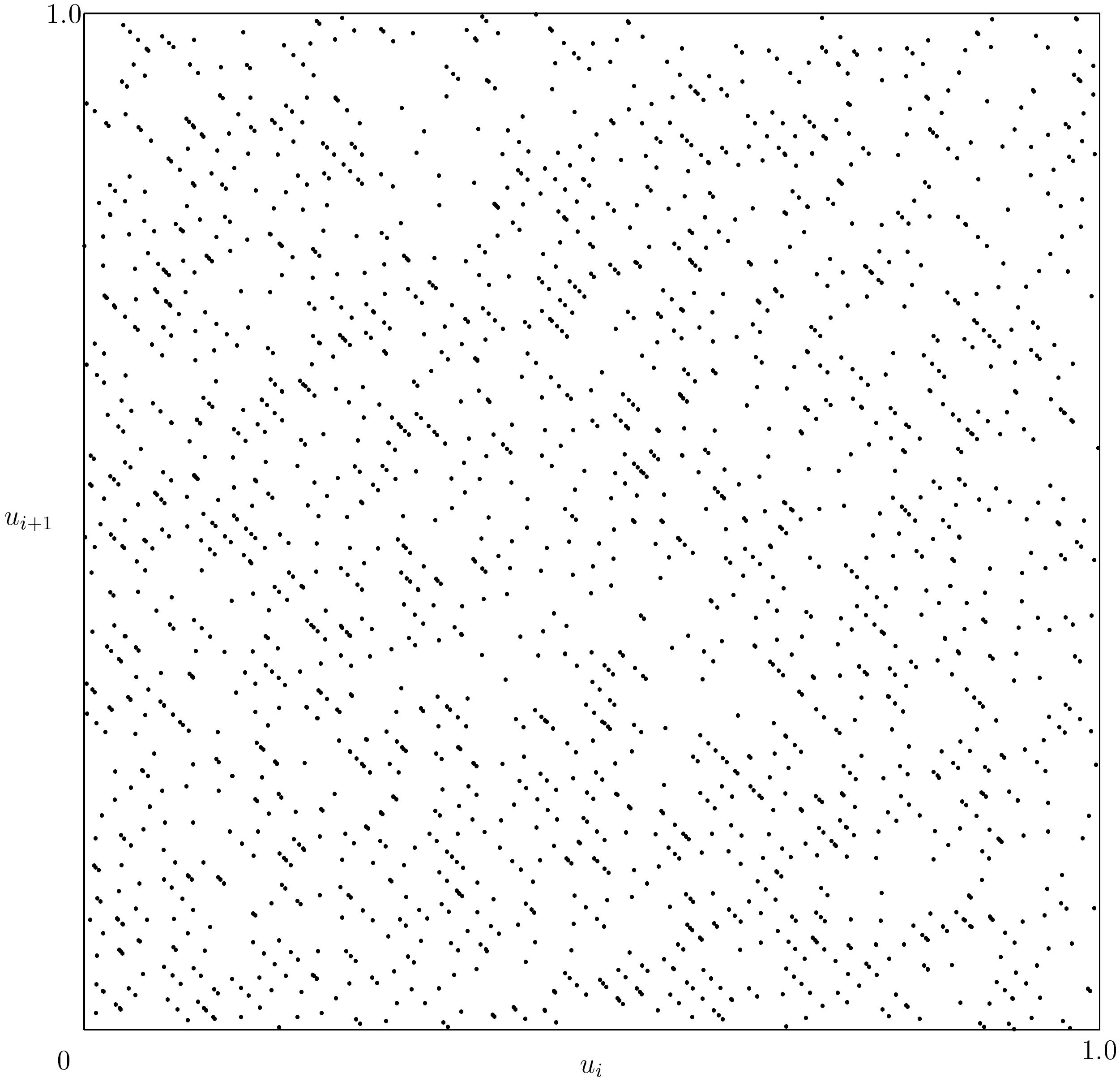

Figure 2: 2000 pairs of successive points for the SWB of Example 2, with $L = 19$.

Table 3: The Values of d_t and q_t for Example 3.

t	d_t	q_t
2	2.328 E-10	9.414 E-184
3	2.328 E-10	4.041 E-174
4	2.328 E-10	1.696 E-164
5	2.328 E-10	7.457 E-155
6	2.328 E-10	3.203 E-145
7	2.328 E-10	1.376 E-135
8	2.328 E-10	5.909 E-126
9	2.328 E-10	2.538 E-116
10	2.328 E-10	1.090 E-106
11	2.328 E-10	4.682 E-97
12	2.328 E-10	2.012 E-87
13	2.328 E-10	8.636 E-78
14	2.328 E-10	3.709 E-68
15	2.328 E-10	1.593 E-58
16	2.328 E-10	6.842 E-49
17	2.328 E-10	2.939 E-39
18	2.328 E-10	1.262 E-29
19	2.328 E-10	5.421 E-20
20	2.328 E-10	2.328 E-10
21	2.328 E-10	.9999
22	.5773	4.033 E-10
24	.5773	4.033 E-10
25	.5773	4.033 E-10
26	.5773	4.033 E-10
27	.5773	4.033 E-10
28	.5773	3.802 E-10
29	.5773	3.802 E-10
30	.5773	3.802 E-10

is $M - 1 = 2^{1376} - 2^{704} + 1$. The LCG representation has modulus M and multiplier $A = (2^{32} - 5)^* \bmod M = (2^{32} - 5)^{21} - (2^{32} - 5)^{42} \bmod M$.

Table 4 gives the values of d_t and q_t for that LCG generator, for up to $t = 50$. In all dimensions $t \leq 43$, one has $d_t \leq b^{-1}$, while for $t \geq 44$, we have $d_t = 1/\sqrt{3} \approx 0.577$, in accordance with Lemma 4. So, using that generator for applications which require points in large dimensional spaces could lead to problems. L'Ecuyer [6] has applied a few statistical tests to this generator and found that it fails (rather spectacularly) the "birthday spacing" test proposed by Marsaglia [9].

Table 4: The Values of d_t and q_t for Example 4.

t	d_t	q_t
2	2.328 E-10	1.118 E-395
3	2.328 E-10	4.803 E-386
4	2.328 E-10	2.064 E-376
5	2.328 E-10	8.864 E-367
6	2.328 E-10	3.810 E-357
7	2.328 E-10	1.635 E-347
8	2.328 E-10	7.023 E-338
9	2.328 E-10	3.015 E-328
10	2.328 E-10	1.295 E-318
11	2.328 E-10	5.562 E-309
12	2.328 E-10	2.389 E-299
13	2.328 E-10	1.026 E-289
14	2.328 E-10	4.405 E-280
15	2.328 E-10	1.893 E-270
16	2.328 E-10	8.133 E-261
17	2.328 E-10	3.495 E-251
18	2.328 E-10	1.500 E-241
19	2.328 E-10	6.444 E-232
20	2.328 E-10	2.769 E-222
21	2.328 E-10	1.188 E-212
22	2.328 E-10	5.104 E-203
23	2.328 E-10	2.192 E-193
24	2.328 E-10	9.413 E-184
25	2.328 E-10	4.044 E-174
26	2.328 E-10	1.737 E-164

t	d_t	q_t
27	2.328 E-10	7.459 E-155
28	2.328 E-10	3.203 E-145
29	2.328 E-10	1.376 E-135
30	2.328 E-10	5.909 E-126
31	2.328 E-10	2.538 E-116
32	2.328 E-10	1.090 E-106
33	2.328 E-10	4.682 E-97
34	2.328 E-10	2.011 E-87
35	2.328 E-10	8.636 E-78
36	2.328 E-10	3.709 E-68
37	2.328 E-10	1.593 E-58
38	2.328 E-10	6.842 E-49
39	2.328 E-10	2.939 E-39
40	2.328 E-10	1.262 E-29
41	2.328 E-10	5.421 E-20
42	2.328 E-10	2.328 E-10
43	2.328 E-10	.9999
44	.5773	2.328 E-10
45	.5773	4.033 E-10
46	.5773	4.033 E-10
47	.5773	4.033 E-10
48	.5773	4.033 E-10
49	.5773	4.033 E-10
50	.5773	4.033 E-10

6. Conclusion

We have shown in this paper that the AWC/SWB generators are essentially equivalent to LCGs with large moduli. So, they can be viewed as (extremely) efficient ways of implementing LCGs with "huge" moduli. The difference is a "truncation error" of size at most b^{-L}. When the associated LCG has a lattice structure with distance between hyperplanes significantly larger than $b^{-L}\sqrt{t}$ in dimension t, the AWC/SBW generator also inherits that lattice structure. Our examples illustrate how bad could be that lattice structure for the generators proposed in [12]. In fact, in turns out that all AWC/SWB generators have a very bad lattice structure in dimensions larger than r [2]. Therefore, AWC/SWB generators should not be used directly by themselves. To make those generators useful, one would have to find appropriate combinations with other types of generators, with good theoretical properties. This could be a subject for further research.

Acknowledgments

The second author's work was supported by NSERC-Canada grant # OGP0110050 and FCAR-Québec grant # 93ER1654. Part of the work was accomplished while the second author was holding the Toshiba Chair at Waseda University, in Tokyo. Josée Turgeon helped doing the computations for the second numerical example.

References

[1] R. Couture, P. L'Ecuyer, and S. Tezuka, On the Distribution of k-Dimensional Vectors for Simple and Combined Tausworthe Sequences. To appear in *Mathematics of Computation.*

[2] R. Couture and P. L'Ecuyer, On the Lattice Structure of Certain Linear Congruential Sequences Related to AWC/SWB Generators. In preparation.

[3] F. James, A review of pseudorandom number generators. *Computer Physics Communications*, **60** (1990) 329-344.

[4] D. E. Knuth, *The Art of Computer Programming : Seminumerical Algorithms*, vol. 2, second edition. Addison-Wesley, 1981.

[5] P. L'Ecuyer, Random Numbers for Simulation. *Communications of the ACM* **33**, 10 (1990) 85–97.

[6] P. L'Ecuyer, Testing Random Number Generators. *Proceedings of the 1992 Winter Simulation Conference*, IEEE Press, 305–313.

[7] P. L'Ecuyer and R. Couture, An Implementation of the Lattice and Spectral Tests for Linear Congruential and Multiple Recursive Generators. In preparation.

[8] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications*. Cambridge University Press, Cambridge, 1986.

[9] G. Marsaglia, A Current View of Random Number Generation. *Computer Science and Statistics, Proceedings of the Sixteenth Symposium on the Interface*, Atlanta, march 1984. Elsevier Science Publ. (North-Holland), 1985, 3–10.

[10] G. Marsaglia, B. Narasimhan, and A. Zaman, A Random Number Generator for PC's, *Computer Physics Communications* **60** (1990) 345–349.

[11] G. Marsaglia and L.-H. Tsay, Matrices and the Structure of Random Number Sequences, *Linear Algebra and its Applications* **67** (1985) 147–156.

[12] G. Marsaglia and A. Zaman, A New Class of Random Number Generators, *The Annals of Applied Probability* **1** (1991) 462–480.

[13] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM CBMS-NFS Regional Conference Series in Applied Mathematics, vol. 63, SIAM, Philadelphia, 1992.

[14] S. Tezuka, Analysis of Marsaglia's New Random Number Generators, IBM TRL Research Report, RT-5018, (Feb. 1991)

[15] S. Tezuka and P. L'Ecuyer, Analysis of Add-with-carry and Subtract-with-borrow generators, *Proceedings of the 1992 Winter Simulation Conference*, IEEE Press. (1992) 443–447.

[16] S. Tezuka, A Unified View of Long-Period Random Number Generators, Submitted for publication.