Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

ANALYSIS OF ADD-WITH-CARRY AND SUBTRACT-WITH-BORROW GENERATORS

Shu Tezuka

IBM Research
Tokyo Research Laboratory
5-19 Sanbancho, Chiyoda-ku
Tokyo 102, JAPAN

Pierre L'Ecuyer

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

ABSTRACT

Marsaglia and Zaman recently proposed a new class of random number generators, *add-with-carry* and *subtract-with-borrow*, 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. 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.

1. THE AWC AND SWB GENERATORS

Marsaglia and Zaman (1991) recently proposed two new types of random number generators, called *add-with-carry* (AWC) and *subtract-with-borrow* (SWB). The AWC generator is based on the recursion

$$x_i = (x_{i-s} + x_{i-r} + c_i) \bmod b, \qquad (1)$$

$$c_{i+1} = I(x_{i-s} + x_{i-r} + c_i \geq b), \qquad (2)$$

where $r > s$ are positive integers called the lags, 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 $x_{i-s} + x_{i-r} + c_i \geq b$. 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. 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 is much beyond what could be required by any application.

When b is large enough (see James 1990), one can produce a $U(0,1)$ variate at each step by

$$u_i = x_i/b. \qquad (3)$$

More generally, one can use L successive values of x_j to produce one u_i as follows:

$$u_i = \sum_{j=0}^{L-1} x_{Li+j} b^{j-L}. \qquad (4)$$

If b is small, or if more precision is desired, take a larger L. Here, it is important to note that the digits of u_i are filled up from the least significant to the most significant one. The sequence (4) is an analogue of the Tausworthe sequence (Tausworthe 1965). For the latter, the digits of u_i are filled up by a linear feedback shift register sequence modulo two (so, $b = 2$).

The SWB is based on a similar recursion:

$$x_i = (x_{i-s} - x_{i-r} - c_i) \bmod b, \qquad (5)$$

$$c_{i+1} = I(x_{i-s} - x_{i-r} - c_i < 0), \qquad (6)$$

where $r > s$. Here, c_i is called the *borrow*. The maximum possible period is $b^r - b^s$, and is achieved when $M = b^r - b^s + 1$ is prime and b is a primitive root modulo M. The u_i's can be produced from the x_j's in the same way. For a full period AWC or 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 (Marsaglia and Zaman 1991).

The AWC and SWB methods can be viewed as variants of the so-called *additive* or *subtractive* methods recommended in Knuth (1981). Marsaglia and Zaman (1991) have given a list of parameter sets for AWC and SWB generators, for which the order of b modulo M is very large or near to 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 with a very large prime modulo is a difficult task in practice.

In this paper, we analyze the structure of the sequence u_i, $i = 1, 2, \ldots$, produced by an AWC or an SWB generator. That sequence turns out to be essentially the same as the sequence produced by a linear congruential generator (LCG). More precisely, we have the following. Let $M = b^r + b^s - 1$ for AWC, and $M = b^r - b^s + 1$ for SWB. Suppose that M 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 $A = (b^*)^L \bmod M = b^{(M-2)L} \bmod M$:

$$\begin{aligned} X_i &= AX_{i-1} \bmod M, & (7) \\ w_i &= X_i/M. & (8) \end{aligned}$$

Our main result is:

THEOREM 1. *Let $\{u_i,\ i \geq 0\}$ be the sequence (4) produced by an AWC or SWB generator, while $\{w_i,\ i \geq 0\}$ is the sequence produced by (8). Then, if X_0 is chosen appropriately, one has*

$$u_i = b^{-L} \lfloor b^L w_i \rfloor \tag{9}$$

for all $i > r$.

The condition "X_0 is chosen appropriately" means that the two sequences must have corresponding initial seeds. Otherwise, (9) will hold after an appropriate shift of one of the two sequences. Equation (9) 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, $|u_i - w_i| \leq b^{-L}$. In other words, the sequences (4) and (8) 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. So, for all practical purposes, considering the limited precision of floating point numbers on computers, one can safely assume that $u_i = w_i$.

We call (7-8) the *LCG representation* of the corresponding AWC or SWB generator. In Section 2, we sketch briefly a proof of our main result and show how to obtain the state of the LCG representation from the state of an AWC generator, and vice-versa. For a theoretical evaluation of the structural properties of an AWC or SWB generator, one can study the lattice structure of its LCG representation. In Section 3, we do that on some examples, including a small family of generators taken from Marsaglia and Zaman (1991), to which we apply the spectral and Beyer tests. To apply these tests, we used the software package described in L'Ecuyer and Couture (1992), which is still under development, and based in part on the algorithms given in Afflerbach and Grothe (1985) and Fincke and Pohst (1985). Observe that if the multiplier A in (7) is replaced by its inverse $A^* = b^L \bmod M$, then it will produce the same sequence, 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 A^* or A will yield the same results.

A more detailed version of this paper, with all the proofs and further results, is currently in preparation. For a survey of random number generation for simulation, see L'Ecuyer (1990).

2. SWITCHING BETWEEN THE AWC AND LCG REPRESENTATIONS

In this section, we sketch the proof of the main result and show how one can switch from the AWC (or SWB) sequence to its LCG representation, and vice-versa. For simplification, we will assume here that $L = 1$ and will consider the AWC generator. The more general case ($L > 1$) and the SWB generator can be analyzed in a similar way. Consider the LCG (7). Since b^* is the multiplicative inverse of b, one also has $X_{i-1} = bX_i \bmod M$. From that, X_i is now defined for all integers $i \in \mathbb{Z}$. Now, let

$$w_i = X_i/M = .x_i x_{i-1} x_{i-2} \ldots, \tag{10}$$

where the right-hand-side denotes the digital expansion of X_i/M in base b. For each i, define

$$\begin{aligned} c_i &= \sum_{j=1}^{\infty} b^{-j}(-x_{i-j} + x_{i-j+r} + x_{i-j+s}), \\ A_i &= \sum_{j=1}^{r} b^{r-j} x_{i-j} + \sum_{j=1}^{s} b^{s-j} x_{i-j}. \end{aligned}$$

Then, one can show that

$$X_i = A_{i+1} + c_{i+1}, \tag{11}$$

that c_{i+1} must be 0 or 1, and that the sequence $\{(x_i, c_{i+1}),\ i \in \mathbb{Z}\}$ satisfies the recursion (1-2). Further, $u_i = x_i/b = \lfloor bw_i \rfloor / b$, which is (9).

From (11), one can transform the state $(x_{i-r+1}, \ldots, x_i, c_{i+1})$ of the AWC generator into the state X_i of its LCG representation. Reciprocally, by expanding $w_i = X_i/m$ in base b, one recovers $x_i, x_{i-1}, \ldots, x_{i-r}$, and then compute c_i using the fact that $c_i = 1$ if and only if $(x_{i-r} + x_{i-s} + 1 - x_i) \bmod b = 0$.

It is important to note that the transformation from the LCG state to the AWC state is not onto. Indeed, if the LCG has full period, it has $M - 1$ (non trivial) possible states. On the other hand, the AWC generator has $2b^r$ possible states, which is more than twice $M - 1$. So, many different AWC states can be mapped to the same X_i. It turns out, however, that for all initial states of the AWC except $(0, \ldots, 0, 0)$ and $(b-1, \ldots, b-1, 1)$ (which must be avoided), the period is equal to the order of b modulo M (which is $M - 1$ if the LCG has full period). Further, the length of the initial transient is at most r. This explains the "$i > r$" at the end of the statement of Theorem 1.

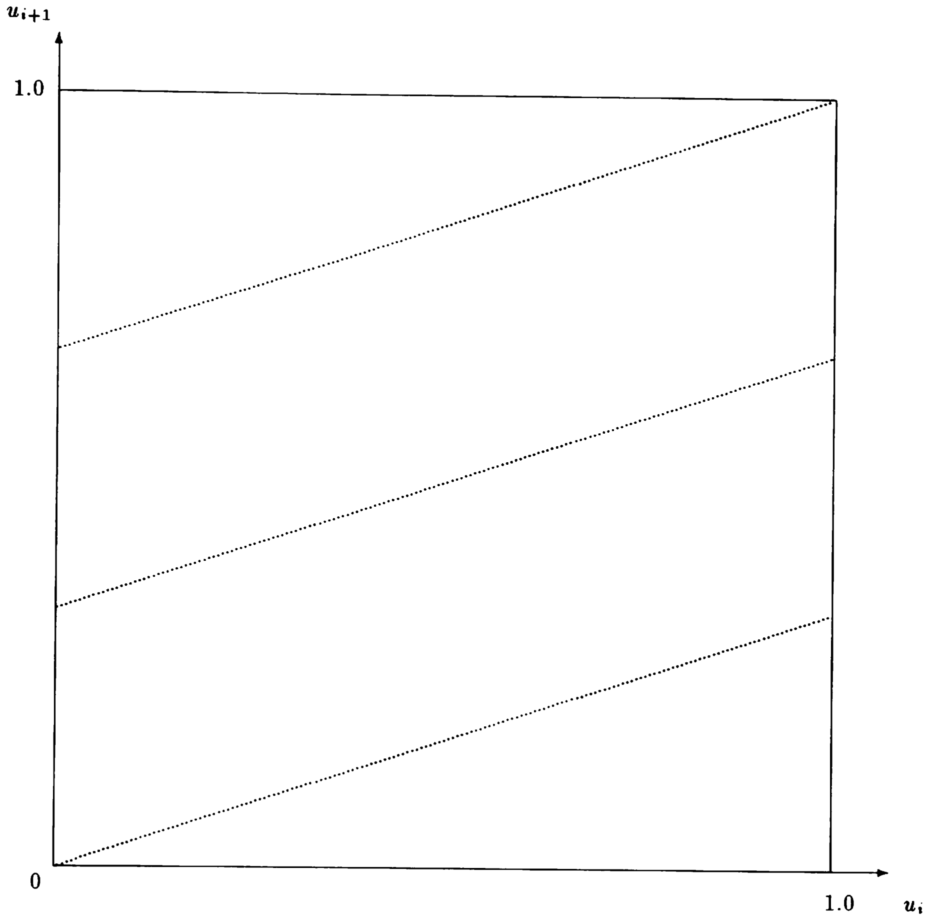

Figure 1: All Pairs of Successive Points for the SWB Generator of Example 1

3. EXAMPLES, LATTICE STRUCTURE, AND SPECTRAL TEST

3.1. Example 1

As a first example, we consider the a SWB generator with $(b, s, r, L) = (2, 2, 9, 9)$. Here, $x_i = (x_{i-2} - x_{i-9} - c_i) \bmod 2$,

$$u_i = \frac{1}{2^9} \sum_{j=0}^{8} x_{9i+j} 2^j,$$

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 $(x_1, ..., x_9, c_{10}) = (1, 0, ..., 0)$. This looks like a typical lattice structure of a (bad) LCG.

The LCG representation of that SWB generator is

$$X_i = 170 X_{i-1} \bmod 509; \qquad w_i = X_i / 509.$$

Note that 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: those with sharp eyes can see that the points in Figure 1 are not exactly aligned on the three lines. However, everybody will agree that the approximation is quite good.

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.

3.2. Example 2: A "Classroom" AWC Generator

We now examine the "classroom" AWC generator given in Section 7 of Marsaglia and Zaman (1991), for which $(b, s, r, L) = (6, 2, 21, L)$. The sequence is defined by

$$u_i = \frac{1}{6^L} \sum_{j=0}^{L-1} x_{Li+j} 6^j,$$

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

Table 1: Beyer and Spectral Tests for Example 2

L	7	9	11	17	19
A	3760617870802950	3760620047585286	3760620108051462	3760620109779030	3760620109779066
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

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 (1991), the x_i's, if used directly, could provide and excellent simulation of independent throws of a dice.

The LCG representation is given by

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

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 applied the spectral and Beyer tests to the corresponding LCG's for the other values of L. The results are given in Table 1. The values d_t and q_t are respectively the distance between hyperplanes in the unit hypercube, and the Beyer quotient, in dimension t. For more details on the Beyer and spectral tests, see L'Ecuyer (1990).

It turns out that for all values of L, the lattice structure is 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 $L = 19$).

ACKNOWLEDGMENTS

This work was supported by NSERC-Canada grant # OGP0110050 and FCAR-Québec grant # 93ER1654 to the second author. Raymond Couture and Josée Turgeon helped doing the computations for the numerical examples.

REFERENCES

Afflerbach, L. and H. Grothe. 1985. Calculation of Minkowski-Reduced Lattice Bases. *Computing*, **35**: 269–276.

Fincke, U. and M. Pohst. 1985. "Improved Methods for Calculating Vectors of Short Length in a Lattice, Including a Complexity Analysis", *Mathematics of Computation*, **44**, 170:463–471.

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

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

L'Ecuyer, P. 1990. Random Numbers for Simulation. *Communications of the ACM*, **33**, 10:85–97.

L'Ecuyer, P. and R. Couture. 1992. An Implementation of the Lattice and Spectral Tests for Linear Congruential and Multiple Recursive Generators. In preparation.

Marsaglia, G. and A. Zaman. 1991. A new class of random number generators. *The Annals of Applied Probability*, **1**, 3:462–480.

Tausworthe, R. C. 1965. Random numbers generated by linear recurrence modulo two. *Math. Comp.*, **19**:201-209.

AUTHOR BIOGRAPHIES

SHU TEZUKA is an advisory researcher at the IBM Tokyo Research Laboratory, Tokyo, Japan. His research interests are in random number generation, global optimization, and stochastic simulation. He is an Editor for the *Journal of Information Processing Society of Japan*, and a councillor of the *Society of Information Theory and its Applications*. He is also a member of ACM, IEEE, SIAM, and IPSJ.

PIERRE L'ECUYER is a full 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 a professor in the computer science department, at Laval University, Ste-Foy, 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 an Associate Editor for *Management Science* and for the *ACM Transactions on Modeling and Computer Simulation*. He is also a member of ACM, IEEE, ORSA and SCS.