Linear Congruence Generators

Matthew Moreno

December 2, 2015

1 Introduction

1.1 Motivation

In addition to the lottery [1], sequences of random numbers are necessary ingredients for Monte Carlo methods, digital cryptography, and computer simulations of phenomena with random aspects, and other applications. Over the years, several schemes to use physical phenomena that emulate the function of a mathematical random variable have been put forward. Notably, these include use of a "sonic roulette wheel" by the Rand Corporation to generate data for its book of "a million random digits"(Rand Corporation 2001), the extraction of random digits from images of lava lamps in motion (Noll, Mende, and Sisodiya 1998), the use of human input (e.g. on the keyboard or mouse) to generate random digits (Cole 2011), and the parsing of random streams of data from quantum effects such as the fluctuations in the magnetic field of a vacuum (*ANU Quantum Random Number Server* 2015). While these operations give good results in practice, they are expensive to implement and provide a slow and highly constrained bandwith of values. Thus, deterministic mathematical operations to generate sequences of numbers that emulate the distribution of numbers from an ideal random variable have been developed. These methods are commonly referred to as "pseudorandom number generation." In general, these algorithms are launched with a "seed" value from which an initial internal state is generated. Then, a sequence is generated by repeatedly performing a deterministic computational operation on the state to transition to a new internal state and yield a new "random" value. It is important to note several important distinctions beetween pseudorandom number generation and true random generation. First, future values produced by a pseudorandom generator, unlike the future value of a true random variable, can be deduced from information on the current state of the

[1] if it is being conducted fairly, at least!

pseudorandom generator—pseudorandom generation is a completely deterministic process. Also, unlike ideal random variables, sequences of numbers generated by pseudorandom methods are periodic. There are a finite number of internal states that a pseudorandom generator can be in so periodicity arises in the output of the generator when it eventually returns to a previously encountered internal state. John Von Neumann, who worked with early computing devices such as the ENIAC, commented on these important distinctions, noting that that "any one who considers arithmetical methods of producing random digits is, of course, in a state of sin" (Neumann 1950). The unique properties that differentiate pseudorandom generation from true random generation—determinism and periodicity—can be overcome or even employed gainfully in applied settings, however. The periodicity of sequences generated via pseudorandom generation can easily be made so large[2] that it is of no practical concern (Matsumoto and Nishimura 1998). The determinism of pseudorandom generation allows, so long as the initial seed value is known, for a pseudorandom sequence to be recreated so the computations performed using that particular sequence can be readily repeated. This is particularly useful in the debugging process (Hull and Dobell 1962). Further, the use of pesuedorandom generation frees computer scientists from expensive, specialized hardware required to perform true random number generation and the physically-limited throughput capacity of true random number generation. It should be no surprise, therefore, that pseudorandom generation is widely employed today in applications ranging from financial simulation to biological simulation to digital playlist shuffling (Schwartz 2008; Dunbar 2015).

1.2 The Linear Congruence Method

The Linear Congruence Method (LCM) is a well-studied approach to pseudorandom number generation. It was developed by Lehmer in 1949. Using a multiplier of 23 and a modulus of $10^8 + 1$, he successfully generated sequences of more than five million eight decimal digit numbers using an ENIAC computing machine (Hull and Dobell 1962). This method has a solid theoretical framework showing that, under certain special circumstances, the sequence possesses the very similar moments to the uniform distribution over (0,1]. Additionally, by strategically choosing the modulus as a power of two, the calculations required to perform the linear congruence method can be performed rather efficiently with binary computing machines. m-tuples derived from sequences generated using Linear Congruence method have been shown to lie on relatively few hyperplanes in $\mathbb{R}^m$ (Marsaglia 1968). Thus, because the values of a member of a LCM-generated sequence are not completely statistically independent of the other values in the sequence, the LCM approach is not appropriate for applications highly sensitive to the quality of pseudorandom sequences that are provided

[2] The Mersenne twister, for example, exhibits a $2^{19937} - 1$ element long periodic in its output.

(such as Monte Carlo methods). Although this approach has been largely superseded by a new generation of pseudorandom generators such as the Mersenne Twister, it is still not infrequently employed today, and is of theoretical interest and historical significance.

2 The Linear Congruence Method

2.1 Algorithm

The Linear Congruence Method algorithm is presented as shown in (Hull and Dobell 1962). Begin by choosing "magic values" as follows:

- **m**: modulo; $m > 0, m \in \mathbb{Z}$
- **a**: multiplier; $a > 0, a \in \mathbb{Z}$
- **c**: increment; $c \geq 0, a \in \mathbb{Z}$

The particular choice of "magic values" determines important characteristics of the sequence $\{\mathbb{X}_i\}$ that will be generated using LCG. This will be discussed in greater detail later, but suffice it to say that a poor choice of "magic values" may lead to relatively short periodicity in $\{\mathbb{X}_i\}$ while choosing "magic values" that fulfill certain criteria guarantees a periodicity of exactly m, the modulo value chosen.

Next, a seed value $\mathbb{X}_0$ is chosen such that $\mathbb{X}_0 < m$ and $\mathbb{X}_0 \in \mathbb{Z}$. The sequence $\{\mathbb{X}_i\}$ is then generated recursively using the relationship

$$\mathbb{X}_{n+1} = (a \cdot \mathbb{X}_n + c) \mod m \tag{1}$$

In this way, the sequence $\{\mathbb{X}_i\}$ can be built up term after term as desired; this relationship is typically employed in applications using the LCM method. However, a closed-form expression can also be used to determine the nth value of a sequence $\{\mathbb{X}_i\}$ with seed $\mathbb{X}_0$

$$\mathbb{X}_n = \left(a^n \mathbb{X}_0 + \frac{(a^n - 1)c}{a - 1}\right) \mod m \tag{2}$$

This type of "shortcut" is useful in applications where distinct, finite subsequences of a single pseudorandom sequence with a certain seed are utilized asynchronously, such as GPU computing(*MWC64X - Uniform random number gener* 2015), as well as in formal mathematical analysis of the properties of the Linear Congruence Method.

2.2 Computational Examples

Table 1 provides a first example of a sequence generated by the Linear Congruence Method with $m = 5$, $a = 3$, $c = 2$, $\mathbb{X}_0 = 1$. Note that after only four steps in this sequence, we have returned to a value already encountered in the sequence. Because—by definition—the value of each item $\mathbb{X}_n$ where $n > 0$ in the sequence $\{\mathbb{X}_i\}$ depends only on the value of the item in the sequence that directly precedes it, $\mathbb{X}_{n-1}$, the sequence will exhibit a periodicity with period $p = 4$. Observe further that $p < m$ and, relatedly, that there does not exist n such that $\mathbb{X}_n = 4$.

n	$\mathbb{X}_n$	$(\mathbb{X}_n \cdot a + c) \mod m$
0	1	$(1 \cdot 3 + 2) \mod 5 = 0$
1	0	$(0 \cdot 3 + 2) \mod 5 = 2$
2	2	$(2 \cdot 3 + 2) \mod 5 = 3$
3	3	$(3 \cdot 3 + 2) \mod 5 = 1$
4	1	

Table 1: An annotated sequence of numbers generated using the linear congruence method with period $p < m$.

Table 2 provides a second, and final, example of a sequence generated by the Linear Congruence Method with $m = 9$, $a = 4$, $c = 2$ $\mathbb{X}_0 = 2$. After nine steps in this sequence, we have to returned to a value already encountered in the sequence. Thus, this sequence will exhibit periodicity with period $p = 9$. Take special note that, for this particular set of "magic values" we have $p = m$ and with $0 \leq j, k < m$ for every k there exists a unique j such that $\mathbb{X}_k = j$ and vice versa.[3] As discussed in Section 3, these observations hold true for any possible choice of seed value $\mathbb{X}_0$ and result from fulfillment of particular conditions on the "magic values" chosen for the Linear Congruence Generator.

n	$\mathbb{X}_n$	$(\mathbb{X}_n \cdot a + c) \mod m$
0	4	$(4 \cdot 4 + 2) \mod 9 = 0$
1	0	$(0 \cdot 4 + 2) \mod 9 = 2$
2	2	$(2 \cdot 4 + 2) \mod 9 = 1$
3	1	$(1 \cdot 4 + 2) \mod 9 = 6$
4	6	$(6 \cdot 4 + 2) \mod 9 = 8$
5	8	$(8 \cdot 4 + 2) \mod 9 = 7$
6	7	$(7 \cdot 4 + 2) \mod 9 = 3$
7	3	$(3 \cdot 4 + 2) \mod 9 = 5$
8	5	$(5 \cdot 4 + 2) \mod 9 = 4$
9	4	

Table 2: An annotated sequence of numbers generated using the linear congruence method with period $p = m$.

[3] Equivalently, a one-to-one bijective relation exists between $\{\mathbb{X}_i\}$ where $0 \leq i < p$ and $\{n\}$ where $0 \leq i < n$.

3 Distribution of Random Variables Simulated Using the Linear Congruence Method

We begin by introducing a theorem from (Hull and Dobell 1962) in order to facilitate our investigation into the distribution of sequences $\{\mathbb{X}_i\}$ generated via LCM.

Theorem 3.1 (Linear Congruence Generator Full Period Theorem)
The sequence generated by the recursive relationship shown in Equation 1 has period length $p = m$ if and only if

1. *c is relatively prime to m;*
2. *for all prime factors f of m, $a \mod f = 1$;*
3. *if 4 is a factor of m, $a \mod 4 = 1$.*

This theorem will allow us to reckon out the kth raw moment of a scaled form of a sequence $\{\mathbb{X}_i\}$ produced from a Linear Congruence Generator with "magic numbers" that fulfill the conditions of Theorem 3.1 and thus have period length $p = m$. With $\{\mathbb{X}_i\}$ defined as a sequence generated from a Linear Congruence Generator with modulo m and "magic numbers" satisfying the stipulations of Theorem 3.1, define the sequence $\{\mathbb{Y}_i\}$ such that

$$\mathbb{Y}_n = \frac{\mathbb{X}_n}{m} \ \forall n \tag{3}$$

A value $\mathbb{X}_n$ in a sequence $\{\mathbb{X}_i\}$ from a Linear Congruence Generator with modulo m is inherently restricted $0 \leq \mathbb{X}_n < m$ so $\mathbb{Y}_n \in [0, 1) \ \forall n$. Theorem 3.2 gives us $\lim_{m\to\infty} E(\{\mathbb{Y}_i\}^k) = 1/(k+1)$ if we assume special conditions on the "magic numbers" governing the LCM generator behind $\{\mathbb{Y}_i\}$. First, though, we develop Lemmas 3.1 and 3.2.

Lemma 3.1 (Uniqueness of Values of Sequences Generated Via LCM)
Let p represent the smallest periodicity of a sequence $\{\mathbb{X}_i\}$ from a Linear Congruence Generator. All values $\mathbb{X}_0\}, ..., \mathbb{X}_{p-1}$ are unique.

Proof. Suppose there exists $0 \leq j, k < p$ such that $i \neq j$ and $\mathbb{X}_j = \mathbb{X}_k$. For convenience, we assume without loss of generality that $j < k$ By the recursive definition 1 of the sequence $\{\mathbb{X}_i\}$, for elements s and t of the sequence $\mathbb{X}_s = \mathbb{X}_t$ implies $\mathbb{X}_{s+1} = \mathbb{X}_{t+1}$. Thus, by induction with base case $\mathbb{X}_j = \mathbb{X}_k$, for all $n > 0 \in \mathbb{Z}$ we have $\mathbb{X}_{j+n} = \mathbb{X}_{k+n}$. Rewriting, we have $\mathbb{X}_{j+n} = \mathbb{X}_{j+(k-j)+n}$ for all $n > 0 \in \mathbb{Z}$. It follows by further inductive analysis that $\mathbb{X}_{j+n} = \mathbb{X}_{j+\alpha(k-j)+n}$ for any $\alpha > 0 \in \mathbb{Z}$. Take note that with $j, k < p$ we have $k - j < p$. The existence of $0 \leq j, k < p$ such that $\mathbb{X}_j = \mathbb{X}_k$ therefore implies the existence of periodicity $k - j < p$ in the sequence $\mathbb{X}_i$. Our initial supposition therefore violates the status of p as the smallest periodicity of a sequence $\{\mathbb{X}_i\}$, so it cannot be true. □

Lemma 3.2 (Composition of Full Period Sequences Generated Via LCM)
Let p represent the smallest periodicity of a sequence $\{\mathbb{X}_i\}$ from a Linear Congruence Generator with modulo m. If the sequence $\{\mathbb{X}_i\}$ achieves maximal period-length, that is if $p = m$, the subsequence $\{\mathbb{X}_i\}$ with $i \in \mathbb{Z}$ and $0 \leq i < p$ is a re-arrangement of the sequence $0, ..., p - 1$.

Proof. We want to show that for every $0 \leq n < p$ with $n \in \mathbb{Z}$ there exists a unique $i \in \mathbb{Z}$ with $0 \leq i < p$ such that $\mathbb{X}_i = n$. Lemma 3.1 gives us uniqueness; there cannot exist $0 \leq j, k < p$ such that $i \neq j$ and $\mathbb{X}_j = \mathbb{X}_k$. Existence, however, remains untreated. We will first show $q \in \{\mathbb{X}_i\} \Rightarrow q \in \{0, ..., p-1\}$ then verify existence by showing $q \in \{0, ..., p - 1\} \Rightarrow q \in \{\mathbb{X}_i\}$.

- $q \in \{\mathbb{X}_i\} \Rightarrow q \in \{0, ..., p - 1\}$

  Recall that, by the definition of the modulo operation, a value $\mathbb{X}_n$ in a sequence $\{\mathbb{X}_i\}$ from a Linear Congruence Generator with modulo m is restricted $0 \leq \mathbb{X}_n < m$. As an assumption, we have $p = m$ so the restriction on $\mathbb{X}_n$ can be written as $0 \leq \mathbb{X}_n < p$. With $\mathbb{X}_n \in \mathbb{Z}$ we have $q \in \{0, ..., p - 1\}$.

- $q \in \{0, ..., p - 1\} \Rightarrow q \in \{\mathbb{X}_i\}$

  Suppose there exists $q \in \{0, ..., p-1\}$ such that $q \notin \{\mathbb{X}_i\}$. We have $g \in \{\mathbb{X}_i\} \Rightarrow g \in \{0, ..., p-1\}$ so we would have $g \in \{\mathbb{X}_i\} \Rightarrow g \in \{\{0, ..., p-1\} \setminus \{q\}\}$. Thus, we would have $\{\mathbb{X}_i\} \subset \{0, ..., p - 1\} \setminus \{q\}$. Note that $|\{\mathbb{X}_i\}| = |\{0, ..., p - 1\}| = p$. Because $q \in \{0, ..., p - 1\}$, $|\{0, ..., p - 1\} \setminus \{q\}| = p - 1$. Uniqueness of $\mathbb{X}_n \in \{\mathbb{X}_i\}$ together with $g \in \{\mathbb{X}_i\} \Rightarrow g \in \{\{0, ..., p - 1\} \setminus \{q\}\}$ would force $|\{\mathbb{X}_i\}| \leq p - 1$. However, we know $|\{\mathbb{X}_i\}| = p$. Thus, it must be true that $q \in \{0, ..., p - 1\} \Rightarrow q \in \{\mathbb{X}_i\}$.

□

Theorem 3.2 (Linear Congruence Method Sequence kth Raw Moment)
With $\{\mathbb{Y}_i\}$ defined via Equation 3 with a Linear Congruence Generator with "magic numbers" satisfying Theorem 3.1, $\lim_{m\to\infty} E(\{\mathbb{Y}_i\}^k) = 1/(k + 1)$.[4]

Proof. From a frequentist perspective it makes sense to calculate the kth moment of a random variable simulated by a sequence $\{\mathbb{S}_i\}$ as $\lim_{a\to\infty} \sum_{n=0}^{a} \frac{\mathbb{S}_n^k}{n}$. If the sequence $\{\mathbb{S}_i\}$ has periodicity p, we calculate the

[4] Indirectly inspired by (Schruben 2007).

expected value of a random variable simulated a sequence $\{\mathbb{S}_i\}$ as

$$E(\{\mathbb{S}_i\}) = \frac{1}{p}\sum_{n=0}^{p-1}\mathbb{S}_n^k$$

Note that we can choose an arbitrarily large m and find values of c and a that fulfill Theorem 3.1. Consider, for example, defining $m = 3^z$ with $z > 1, z \subset \mathbb{Z}$, c as 2, and a as 2. We have $\lim_{z\to\infty} m = \lim_{z\to\infty} 3^z = \infty$ so m is unbounded. With the unbounded nature of m in hand define $\{\mathbb{X}_i\}$ as the sequence generated by a linear congruence generator with "magic values" satisfying Theorem 3.1 such that $\{\mathbb{X}_i\}$ takes on its full periodicity m. Define sequence $\{\mathbb{Y}_i\}$ such that each term $\mathbb{Y}_n = \mathbb{X}_n/m$.

$$\lim_{m\to\infty} E(\{\mathbb{Y}_i\}^k) = \lim_{m\to\infty}\frac{1}{m}\sum_{n=0}^{m-1}\mathbb{Y}_n^k = \lim_{m\to\infty}\frac{1}{m}\sum_{n=0}^{m-1}(\mathbb{X}_n/m)^k = \frac{1}{m^{k+1}}\sum_{n=0}^{m-1}\mathbb{X}_n^k$$

Lemma 3.2 gives us that the subsequence $\{\mathbb{X}_i\}$ with $i \in \mathbb{Z}$ and $0 \leq i < p$ is a re-ordering of the sequence $0, ..., p-1$ so we rearrange to find

$$\lim_{m\to\infty} E(\{\mathbb{Y}_i\}^k) = \lim_{m\to\infty}\frac{1}{m^{k+1}}\sum_{n=0}^{m-1}n^k$$

At this point, we bring in Faulhaber's formula to do some heavy lifting

$$\sum_{n=1}^{N} n^k = \frac{1}{k+1}\sum_{j=0}^{k}(-1)^j\binom{k+1}{j}B_jN^{k+1-j}$$

where B_j is the jth Bernoulli number. Rewriting our expression for the kth raw moment of $\{\mathbb{Y}_i\}$ with Faulhaber's formula yields

$$\lim_{m\to\infty} E(\{\mathbb{Y}_i\}^k) = \lim_{m\to\infty}\frac{1}{m^{k+1}}\frac{1}{k+1}\sum_{j=0}^{k}(-1)^j\binom{k+1}{j}B_j(m-1)^{k+1-j} + \frac{1}{m^{k+1}}$$

Removing elements that disappear as $m \to \infty$, our expression cleans up to

$$\lim_{m\to\infty} E(\{\mathbb{Y}_i\}^k) = \lim_{m\to\infty}\frac{1}{m^{k+1}}\frac{1}{k+1}(-1)^0\binom{k+1}{0}B_0(m-1)^{k+1}$$

Further evaluation and simplification, including use of the identity $B_0 = 1$, yields

$$\lim_{m\to\infty} E(\{\mathbb{Y}_i\}^k) = \frac{1}{k+1}$$

□

Remark that our results from Theorem 3.2 give, with $\{\mathbb{Y}_i\}$ defined via Equation 3

$$\lim_{m\to\infty} E(\{\mathbb{Y}_i\}) = 1/2$$

and

$$\lim_{m\to\infty} var(\{\mathbb{Y}_i\}) = E(\{\mathbb{Y}_i\}^2) - (E(\{\mathbb{Y}_i\}))^2 = 1/3 - (1/2)^2 = 1/12$$

These values match exactly the expected value and the variance of the uniform random distribution $U[0,1)$.Next, we will obtain an expression for the kth raw moment of the uniform random distribution $U[0,1)$ in order to perform a more rigorous comparison between a random variable simulated via a sequence $\{\mathbb{Y}_i\}$ and $U[0,1)$.

Theorem 3.3 (Uniform Random Distribution Kth Moment)
For $\mathbb{U} \sim U[0,1)$, $E(\mathbb{U}^k) = 1/(1+k)$.

Proof. Take $\mathbb{U} \sim U[0,1)$ and let $f(x)$ represent the probability distribution of $\mathbb{U}$. By definition,

$$E(\mathbb{U}^k) = \int_{-\infty}^{\infty} f(x) \cdot x^k dx$$

For a uniform random distribution over $[0,1)$, we have $f(x) = 1$ for x on $[0,1)$ and $f(x) = 0$ otherwise. Thus,

$$E(\mathbb{U}^k) = \int_0^1 x^k dx = \left.\frac{x^{k+1}}{k+1}\right|_0^1 = \frac{1}{k+1}$$

□

Theorems 3.2 and 3.3 reveal that the kth moments of a random variable distributed as $U[0,1)$ and a random variable simulated via $\{\mathbb{Y}\}_i$ (as defined in Equation 3) for large m are both given as $1/(1+k)$. Moment generating functions can be used as a unique identifying feature of a probability distribution (Dunbar 2015). The identical results of Theorems 3.2 and 3.3 thus draw strong similarities between the behavior of distributions of a random variable $U[0,1)$ and a random variable simulated via $\{\mathbb{Y}\}_i$. Our analysis leads us to conclude that, for large m and selection of "magic numbers" that satisfy Theorem 3.1 the distribution of a random variable simulated using the Linear Generation Method very well resembles the unit uniform distribution $U[0, 1)$.

4 Independence and Random Variables Simulated Using the Linear Congruence Method

Linear Congruence Generators suffer from inherent correlations between consecutive elements of the sequences they generate, as shown by (Marsaglia 1968). Thus, sequences generated through the Linear Congruence Method do not truly display independence and should not be used for sensitive applications such as Monte Carlo methods. However, with the right choice of "magic values," Linear Congruence generators can still perform "well enough" to pass many, except the most rigorous, statistical tests for randomness. We will begin this section by briefly discussing Theorem 4.1, a well known result from (Marsaglia 1968), before moving along to touch on results from statistical tests for randomness.

Theorem 4.1 (Random Numbers Fall Mainly In the Planes)
If $c_1, c_2, ..., c_n$ is any choice of integers such that

$$c_1 + c_2k + c_3k^2 + ... + c_nk^{n-1} \equiv 0 \mod m$$

then all of the points $\pi_1, \pi_2, ...$ will lie i the set of parallel hyperplanes defined by the equations

$$c_1x_1 + c_2x_2 + ... + c_nx_n = 0, \pm 1, \pm 2, ...$$

There are at most

$$|c_1| + |c_2| + ... + |c_n|$$

of these hyperplanes which intersect the unit n-cube, and there is always a choice of $c_1, c_2, ..., c_n$ such that all of the points fall in fewer than $(n!m)^{1/n}$ hyperplanes.

Theorem 4.1 essentially tells us that plots of n-tuples of consecutive values in a LCM derived in a n-dimensional space are arranged in a highly ordered fashion. Specifically, they are restricted to a bounded number of distinct hyperplanes. The result of this theorem can readily be appreciated visually. Consider Figure 1; it is apparent that significant correlation exists between consecutive values (in the form of diagonal "streaks") in a sequence generated by the LCG3 generator (a LCM generator with specific "magic values").

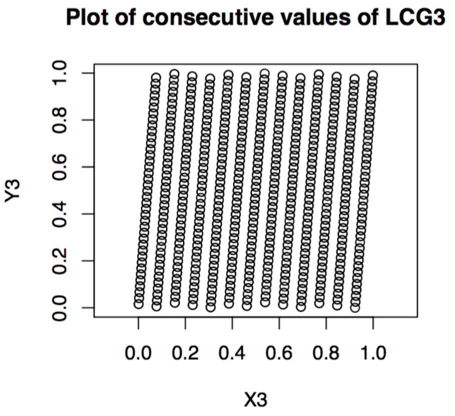

Figure 1: A plot of consecutive values showing egregious correlations in a sequence generated via the Linear Congruence Method (Burgoine 2013).

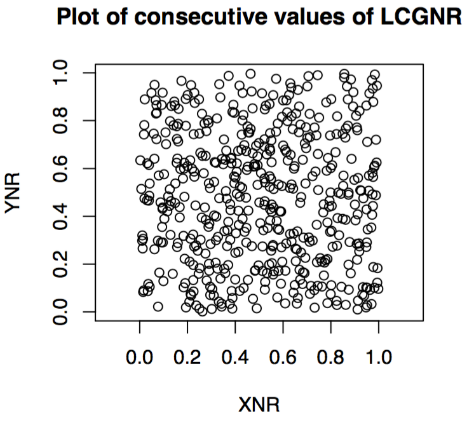

Figure 2: A plot of consecutive values from a sequence generated using the Linear Congruence Method with little visually apparent correlations (Burgoine 2013).

In Figure 2, on the other hand, statistical tests confirm that, as we would expect from a cursory visual inspection, the correlations are not as egregious. Recall that Theorem 4.1 gives us the upper bound on the number of planes containing all n-tuples generated by as $(n!m)^{1/n}$. This bound decreases as m, the modulo component of the Linear Congruence Generator, decreases. As we would expect from visual comparison of Figures 1 and 2, m for the LCG3 generator associated with Figure 1 is much smaller than the m for the LCGNR generator associated with Figure 2 (Burgoine 2013).

The question of how to evaluate the statistical performance of a pseudorandom generator is nebulous. The practical consensus, though, seems to be subjecting it to a battery of tests where each tests a specific statistical property of a random variable that would be expected to manifest in the series of values generated by a pseudorandom generator (L'Ecuyer and Simard 2007). Testing a somewhat slaphazard series of null hypotheses, does not definitively affirm the quality of pseudorandom number generator. Instead this approach of trial by statistical battery simply looks for evidence that a sequence of numbers fails to fulfill a specific statistical property we desire. Examples of statistical tests that might be considered include comparison of the empirical distribution of the maximum streak lengths to the theoretical distribution of maximum streak lengths by a chi-square test or comparison of actual outcomes of a series of random walks (i.e. number of steps to the right, maximum distance reached, fraction of time spent to the right of the origin, number of returns to zero, and number of sign changes) to the expected theoretical distributions, also using a chi-square test (L'Ecuyer and Simard 2007). In (Burgoine 2013), Linear Congruence Generators with differing "magic values" were subjected to a battery of statistical probes such as Kolmogorov-Smirnov tests and the Spearman's Rank Correlation Coefficient test. The performance of these generators fell on a wide spectrum, one passing all the tests and others failing differing numbers of tests. Although LCG statistical performance can be enhanced by an appropriate choice of "magic numbers," it is still vastly outperformed by new, more sophisticated generators such as the Mersenne Twister (Burgoine 2013). Thus, more sophisticated generators should be preferred in settings where the very high statistical quality of pseudorandom sequences is essential.

5 References

ANU Quantum Random Number Server (2015). URL: https://qrng.anu.edu.au/ (visited on 11/19/2015).

Burgoine, Paul (2013). *The Testing of Random Number Generators*. URL: http://www1.maths.leeds.ac.uk/~voss/projects
(visited on 12/01/2015).

Cole, Eric (2011). *Network Security Bible*. en. John Wiley & Sons. ISBN: 9780470570005.

Dunbar, Steven (2015). *Stochastic Processes and Advanced Mathematical Finance: Moment Generating Functions*. URL: https://www.math.unl.edu/~sdunbar1/MathematicalFinance/Lessons/LimitTheoremsCoinTossing/Mome
(visited on 12/01/2015).

Hull, Thomas E. and Alan R. Dobell (1962). "Random number generators". In: *SIAM review* 4.3, pp. 230–254. URL: http://epubs.siam.org/doi/pdf/10.1137/1004061 (visited on 12/01/2015).

L'Ecuyer, Pierre and Richard Simard (2007). "TestU01: AC library for empirical testing of random number generators". In: *ACM Transactions on Mathematical Software (TOMS)* 33.4, p. 22. URL: http://dl.acm.org/citation.c
(visited on 11/19/2015).

Marsaglia, George (1968). "Random numbers fall mainly in the planes". In: *Proceedings of the National Academy of Sciences of the United States of America* 61.1, p. 25. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC
(visited on 12/01/2015).

Matsumoto, Makoto and Takuji Nishimura (1998). "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator". In: *ACM transactions on modeling and computer simulation : a publication of the Association for Computing Machinery.* 8.1, pp. 3–30. ISSN: 1049-3301. DOI: 10.1145/272991.272995.

MWC64X - Uniform random number generator for OpenCL. (2015). URL: http://cas.ee.ic.ac.uk/people/dt10/research
(visited on 12/01/2015).

Neumann, Von J. (1950). “Various Techniques Used in Connection with Random Digits, Collected Works”. In: *National Bureau of Standards, Applied Math Series* 12, pp. 36–38.

Noll, Landon Curt, Robert G. Mende, and Sanjeev Sisodiya (1998). “Method for seeding a pseudo-random number generator with a cryptographic hash of a digitization of a chaotic system”. US5732138 A. U.S. Classification 380/28, 708/254, 380/46; International Classification H04L9/00, H04L9/22, G06F7/58; Cooperative Classification H04L9/0662, H04L9/001, G06F7/582, H04L9/0869; European Classification H04L9/00C, G06F7/58P, H04L9/22. URL: http://www.google.com/patents/US5732138 (visited on 11/19/2015).

Rand Corporation (2001). *A Million Random Digits with 100,000 Normal Deviates*. English. Santa Monica, CA: American Book Publishers. ISBN: 9780833030474.

Schruben, Lee (2007). *IEOR 261 Experimenting with Simulated Systems Lecture Notes*.

Schwartz, Russel (2008). *Biological Modeling and Simulation: A Survey of Practical Models, Algorithms, and Numerical Methods*. English. 1 edition. Cambridge, Mass: The MIT Press. ISBN: 9780262195843.