J Nonlinear Sci
DOI 10.1007/s00332-016-9306-9

Journal of Nonlinear Science

Random Numbers from a Delay Equation

Julian Self[1] · **Michael C. Mackey**[1]

Received: 16 December 2015 / Accepted: 21 April 2016

Abstract Delay differential equations can have "chaotic" solutions that can be used to mimic Brownian motion. Since a Brownian motion is random in its velocity, it is reasonable to think that a random number generator might be constructed from such a model. In this preliminary study, we consider one specific example of this and show that it satisfies criteria commonly employed in the testing of random number generators (from TestU01's very stringent "Big Crush" battery of tests). A technique termed digit discarding, commonly used in both this generator and physical RNGs using laser feedback systems, is discussed with regard to the maximal Lyapunov exponent. Also, we benchmark the generator to a contemporary common method: the multiple recursive generator, MRG32k3a. Although our method is about 7 times slower than MRG32k3a, there is in principle no apparent limit on the number of possible values that can be generated from the scheme we present here.

Communicated by Govind Menon.

Deterministic differential delay equations are well known to sometimes have chaotic solutions that are unpredictable in spite of the fact that they approach either ensemble or trajectory limiting densities that are independent of initial conditions (functions). We show that this characteristic may be used effectively for producing a random number generator.

Electronic supplementary material The online version of this article (doi:10.1007/s00332-016-9306-9) contains supplementary material, which is available to authorized users.

✉ Julian Self
julian.self@mail.mcgill.ca

[1] Departments of Physiology, Physics, and Mathematics, and Centre for Applied Mathematics in Bioscience and Medicine (CAMBAM), McGill University, 3655 Promenade Sir William Osler, Montréal, QC H3G 1Y6, Canada

Published online: 23 May 2016

Keywords Pseudorandom number generator (PRNG) · Random number generator (RNG) · Differential delay equation (DDE) · Deterministic chaos

Mathematics Subject Classification 65C10 (primary) · 34K99 (secondary)

1 Introduction

From Monte Carlo simulators to student selection in American charter schools to financial transactions, random number generators (RNG) are widely employed. It is difficult to articulate what constitutes numbers that are truly random, but often, if generators pass a defined battery of tests, they are said to be random.

In this paper, we show how a first-order differential equation with a delayed argument (differential delay equation, DDE) that has been recently studied can be used as an effective random number generator. In Sect. 2, a very brief history of popular RNGs is given. In Sect. 3, a previously studied DDE producing a Brownian motion is introduced. Section 4 introduces a straightforward scheme for generating numbers from a DDE. Section 5 discusses how to increase generation speed by borrowing a technique from comparable and experimentally realized feedback laser systems. This technique, termed digit discarding, and its potential relationship to the Lyapunov exponent are discussed. Section 5 also reports the results of the statistical tests applied to the RNG. Section 6 discusses the period of the generator and contains a comparison of the DDE as a RNG with a standard generator method, the multiple recursive generator.

This is, as far as the authors know, the first random number generator to employ a differential delay equation, while producing high-quality random numbers. The quality of the numbers is matched by but a few documented generators, with a period no shorter than any other. It is important to remember that when employing random numbers, one does not a priori know the result of, say, a given simulation, so it is impossible to say in which way it would be acceptable for a generator to be systematically flawed. Furthermore, while periods exceeding currently used generators cannot be readily shown to be needed for a stream, longer periods are typically seen as being tied to higher-quality generated numbers L'Ecuyer (1999).

2 A Very Short History of RNGs

Although there also exist non-deterministic, physically implemented, RNGs L'Ecuyer (2012), this section focuses specifically on deterministic software RNGs. Two of the most currently used general-purpose RNGs have a rich history that can be traced back to the earlier RNG counterparts from which they were derived. The Mersenne twister is heavily inspired from the linear feedback shift register (LFSR), while the combined multiple recursive algorithm (CMRG) has its origins in the linear congruential generator. A very brief overview is presented here, and both Knuth and L'Ecuyer have given complete and detailed histories of these generators Knuth (1997), L'Ecuyer (2012).

2.1 Linear Congruential Generators

The linear congruential generator (LCG) was introduced in 1949 by D.H. Lehmer Knuth (1997), in which, for integers X_n, the following sequence can be expressed:

$$X_n = (X_{n-1}a + c)\ mod\ m \tag{2.1}$$

The modulus is denoted m, the multiplier a, the increment c, and the starting value X_0 Knuth (1997). The random number output U_n can be obtained by dividing X_n by m. Much work has been done on studying what values the multiplier, increment, and modulus must have for better distributed output sequences and longer periods. For example, the period length can only be of length m if the increment is relatively prime to the modulus. These generators are still used today, for example they are the default RNG in Java. The output sequences do possess serious flaws in their structure, and so are not suggested.

LCGs were later generalized to multiple recursive generators (MRG), where X_n is a function of not only X_{n-1}, but of linear combinations of $(X_{n-1}, \ldots, X_{n-k})$. So-called lagged Fibonacci generators are of this type.

The MRG algorithm was further improved by employing different MRGs in parallel to form the input of a new modular recurrence relation for the aptly called combined multiple recursive generator (CMRG). This latter generator provides sequences much better distributed than its antecedent, the MRG. The details of the CMRG can be found in L'Ecuyer, and one widely used implementation is the MRG3k32a L'Ecuyer (1999).

2.2 Linear Feedback Shift Registers

In 1965, Tausworthe introduced a binary representation RNG utilizing a recurrence relation modulo 2 Tausworthe (1965). It can be expressed by the following relation L'Ecuyer (2012):

$$\begin{aligned} X_i &= (c_1 X_{i-1} + \cdots + c_k X_{i-k})\ mod\ 2 \\ U_i &= \sum_{l=1}^{w} X_{is+l-1} 2^{-l} \end{aligned} \tag{2.2}$$

In this equation, $\mathbf{c}$ and s are characteristic for a given generator, w is the size of the output vector, and U_i is a final output of this generator which is called the linear feedback shift register (LFSR).

$$\mathbf{X}_{l+n} = \mathbf{X}_{l+m}\ xor\ \mathbf{X}_l\ \mathbf{A}\ (l = 0, 1, \ldots) \tag{2.3}$$

For $\mathbf{A}$ as the identity matrix, the above equation describes the generalized feedback shift register (GFSR). Lewis and Payne (1973), Matsumoto and Kurita (1992). In this case, $\mathbf{X}_l$ is a word of size w with components 0 or 1 while xor refers to the bitwise exclusive or operation. The word, considered as real number between 0 and 1 in binary representation, is the pseudorandom output of the GFSR Matsumoto and Kurita (1992).

The GFSR was further generalized, or twisted, by picking a non-identity matrix **A**. This finally gives rise to the twisted generalized feedback shift register (TGFSR). A variation in the TGFSR is the Mersenne twister, one implementation being MT19937 Matsumoto and Nishimura (1998), which is perhaps the most widely used generator today. For example, it is the default generator in the applied mathematics software package MATLAB.

2.3 Other Generators

There are a wide variety of other RNG algorithms that have been suggested. For example, the LCG can be generalized to a nonlinear recurrence relation Knuth (1997), Eichenauer-Herrmann (1995). Some cryptographic cyphers may also be used as RNGs and in some cases have been thoroughly tested L'Ecuyer and Simard (2007). However, the tests commonly applied to pass cryptographic standards, e.g., the NIST tests, are weak L'Ecuyer and Simard (2007) and so each algorithm would have to be tested and considered separately before it could be recommended as a robust "general-purpose" RNG.

3 Chaotic Solutions to a Delay Differential Equation

Several investigators Beck (1991), Chew and Ting (2002), Mackey and Tyran-Kamińska (2006) have shown that a Brownian-like motion can arise when a particle is subjected to impulsive kicks $f(t)$ derived from a discrete time dynamical system, and whose dynamics are modeled by the following equations where x is the position, v is the velocity, m is the mass, and γ is the friction coefficient:

$$\begin{cases} \dfrac{dx}{dt} = v \\ m\dfrac{dv}{dt} = -\gamma v + f(t). \end{cases}$$

Lei and Mackey (2011) sought an alternative continuous time description of the "random force" $f(t)$, which was assumed to depend on the state (velocity) of a particle, but with a lag time τ, i.e.,

$$f(t) = F(v(t-\tau)),$$

where F has the appropriate properties to generate chaotic solutions. They considered the following differential delay equation

$$\begin{cases} \dfrac{dx}{dt} = v \\ m\dfrac{dv}{dt} = -\gamma v + F(v(t-\tau)), \\ v(t) = \phi(t), \ -\tau \le t \le 0, \end{cases} \tag{3.1}$$

where $\phi(t)$ denotes the initial (or history) function which must always be specified for a differential delay equation.

First, some observations about the second equation in 3.1, which determines the dynamics of the velocity. A simple form of the "random" force is binary and fluctuates between $\pm f_0$, for instance given by

$$F(v) = 2f_0\left[H(\sin(2\pi\beta v)) - \frac{1}{2}\right], \tag{3.2}$$

where H is the Heavyside step function

$$H(v) = \begin{cases} 0 \text{ for } v < 0 \\ 1 \text{ for } v \geq 0. \end{cases}$$

Then, we have the following equation

$$\frac{dv}{dt} = -\gamma v + 2\left[H\left(\sin(2\pi\beta v(t-1)) - \frac{1}{2}\right)\right]. \tag{3.3}$$

(Here and later, we always assume the mass $m = 1$ and $f_0 = 1$ which can be achieved through the appropriate scaling.) The delay differential Eq. 3.3 with a binary "random force" can be solved iteratively by the method of steps[1]. Despite its simplicity, it can display behaviors similar to a random process. An example solution of Eq. 3.3 is shown in Fig. 1. The "random force" in Eq. 3.3 is discontinuous and gives a continuous zigzag velocity curve (c.f. Fig. 1b)

3.1 Deterministic Brownian Motion

Lei and Mackey (2011) focused on an analogous differential delay equation

$$\begin{aligned} &\frac{dv}{dt} = -\gamma v + \sin(2\pi\beta v(t-1)), \\ &v(t) = \phi(t), \ -1 \leq t \leq 0. \end{aligned} \tag{3.4}$$

In Eq. 3.4, β measures the "frequency" of the dependence of the nonlinear function on $v(t-1)$, and this turned out to be an essential parameter in their study. Thus, they studied the dynamical properties of the solutions of Eq. 3.4, both analytically and numerically, but really focused on the probabilistic properties of the chaotic solutions of

[1] A solution of Eq. 3.3 is associated with a time sequence $t_0 < t_1 < \cdots < t_n < \cdots$, which is defined such that $\sin(2\pi\beta v(t)) \geq 0$ when $t \in [t_{2k}, t_{2k+1})$, and $\sin(2\pi\beta v(t)) < 0$ when $t \in [t_{2k-1}, t_{2k})$. Furthermore, if the sequence $(t_0, \cdots, t_n)$ is known, then the solution $v(t)$ when $t \in (t_n, t_n + 1)$ can be obtained explicitly, and therefore, t_{n+1}, which is defined as $\sin(2\beta v(t_{n+1})) = 0$, is determined by $(t_0, \cdots, t_n)$. Once we obtain the entire sequence $\{t_n\}$, the solution of Eq. 3.3 consists of exponentially increasing or decreasing segments on each interval $[t_n, t_{n+1}]$. Nevertheless, the nature and properties of the map $t_{n+1} = F_n(t_0, t_1, \cdots, t_n)$ is still not characterized and has defied analysis to date.

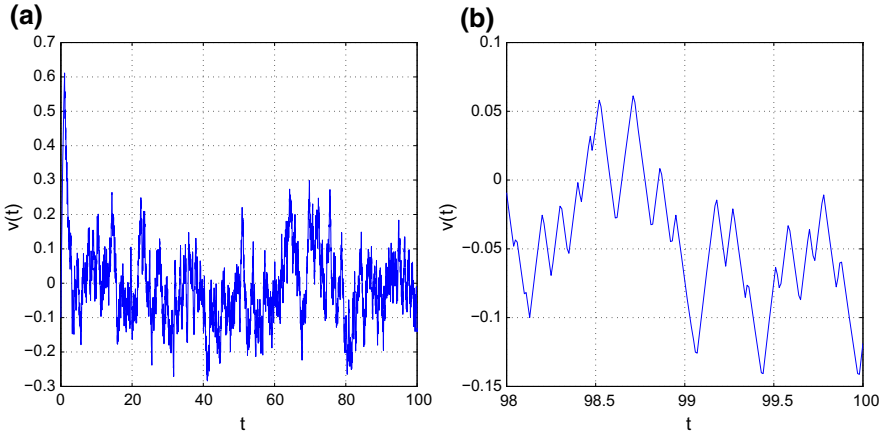

Fig. 1 **a** A sample solution of Eq. 3.3 with $\beta = 10, \gamma = 1, f_0 = 1$, and an initial function $\phi(t) \equiv -0.1, t \in [-1, 0]$. **b** the solution segment for $98 \leq t \leq 100$

$$\begin{cases} \dfrac{dx}{dt} = v \\ \dfrac{dv}{dt} = -\gamma v + \sin(2\pi\beta v(t-1)), \\ v(t) = \phi(t), \ -1 \leq t \leq 0, \end{cases} \tag{3.5}$$

and characterized the statistical solution properties. Their main result was to show that Eq. 3.5 can reproduce experimentally observed Brownian motion data over a wide range of timescales, in spite of the fact that the evolution equation is deterministic. Therefore, the chaotic solutions of 3.5 are a deterministic Brownian motion.

Throughout Lei and Mackey (2011), the probabilistic properties of solutions of Eqs. 3.3 and 3.4 were studied numerically. In their simulations, for a given set of parameters, they solved one of the equations with a randomly selected constant initial function

$$v(t) = v_0 \in (-1, 1), \quad (-1 \leq t \leq 0),$$

where v_0 is drawn from a uniformly distributed density. The solution $v(t)$ was obtained using Euler's method (with a time step $\Delta t = 0.001$) up to $t = 10^5$, and was sampled every 10^3 steps to generate a time series $\{v_n\}$, where $v_n = v(n \times 10^3 \Delta t)$, and $n = 1, 2, \cdots$. The resulting time series of values $\{v_n\}$ was used to characterize the statistical properties of the solution.

In particular, Lei and Mackey (2011) focused on the mean value μ, the upper bound K, the standard deviation σ, and the excess kurtosis γ_2 of the time series, defined by

$$\mu = \frac{1}{N}\sum_{n=1}^{N} v_n, \ K = \max_n |v_n|, \ \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(v_n - \mu)^2,$$

$$\gamma_2 = \frac{\mu_4}{\sigma^4} - 3, \ \text{where } \mu_4 = \frac{1}{N}\sum_{n=1}^{N}(v_n - \mu)^4.$$

The excess kurtosis γ_2 measures the sharpness of the density of the sequence, and a value of $\gamma_2 = 0$ is characteristic of a normal Gaussian distribution.

Lei and Mackey (2011) found that for Eq. 3.4, their numerical results could be approximately fit by the functions

$$K(\beta, \gamma) = \frac{1}{\sqrt{\gamma}(0.68\sqrt{\beta} + 0.60\sqrt{\gamma})} \tag{3.6}$$

$$\sigma(\beta, \gamma) = \frac{0.32}{\sqrt{\beta\gamma}} \tag{3.7}$$

$$\gamma_2(\beta, \gamma) = -\frac{\gamma}{\beta}. \tag{3.8}$$

Additionally, they examined the behavior of the normalized correlation function of a solution defined as

$$C(r) = \lim_{T\to\infty} \frac{\int_0^T v(t)v(t+r)dt}{\int_0^T v(t)^2 dt}.$$

Figure 2a shows $C(r)$ for different values of β (with $\gamma = 1$) for Eq. (3.4). From Figure 2, the correlation function can be approximated as an exponential function of the form

$$C(r) \simeq e^{-r/t_0}, \tag{3.9}$$

where t_0 is the correlation time. Figure 2b–c shows that the correlation time is largely independent of β, and that it is approximately given by $1/\gamma$. Identical results were found for Eq. 3.3, but they did not show these results.

From their numerical results, it was clear that the excess kurtosis γ_2 of the irregular solutions of 3.3 and 3.4 varied with β and γ according to $\gamma_2 \simeq -\gamma/\beta$. Thus, the corresponding distributions approached Gaussian-like distributions when β is large (and γ

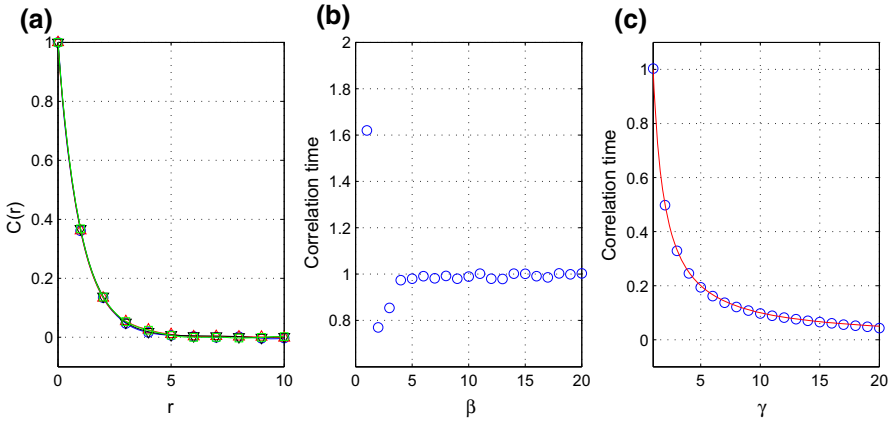

Fig. 2 **a** Correlation function $C(r)$ computed for the solutions of Eq. 3.4. Here, $\gamma = 1$, and $\beta = 5$ (*blue circles*), 10 (*red up triangles*), 15 (*black down triangles*), 20 (*green squares*), respectively. **b** Correlation time as a function of β (with $\gamma = 1$). **c** Correlation time as a function of γ (with $\beta = 20$), *solid curve* is the fit with $t_0 = 1/\gamma$ (Color figure online)

is fixed), but one with a truncated tail so that it is supported on a set of finite measure. They called such truncated Gaussian distributions *quasi-Gaussian distributions*.

Let μ and σ be the mean and standard deviation of a quasi-Gaussian noise, and assume that the noise signal is supported on an interval $[\mu - K, \mu + K]$. Then, the density function is given by

$$p(v; \mu, \sigma, K) = \begin{cases} \mathcal{C}_0 e^{-\frac{(v-\mu)^2}{2\sigma^2}} & \text{if } |v - \mu| \leq K \\ 0 & \text{other wise,} \end{cases} \tag{3.10}$$

where

$$\mathcal{C}_0 = \frac{1}{(\Phi(K/\sigma) - \Phi(-K/\sigma))}$$

and

$$\Phi(z) = \int_{-\infty}^{z} e^{-s^2/2} ds = \sqrt{\frac{\pi}{2}} \left[1 + \text{erf}\left(\frac{z}{\sqrt{2}} \right) \right].$$

4 DDE-RNG

4.1 Mapping to Random Numbers

Knowing the density, or distribution, of solutions from Eq. 3.4, it is possible to generate random numbers. One way to do this is the following. First, γ can be scaled to 1, while the parameter β should be chosen to be larger than 20 to assure a non-periodic time series solution Lei and Mackey (2011). The history function ϕ can be taken as any constant in the interval $(-1, 1)$, as it was in Lei and Mackey (2011). Finally, the Euler method can be used with a time step of $\Delta t = 0.001$, and the time series can be sampled with an appropriate interval for a sufficiently small correlation coefficient. The sampled time series can be mapped to a uniform distribution the interval $[0, 1)$ by using Eq. 3.4 with Eq. 3.10 where $\mu = 0$:

$$\zeta(v) = \frac{\text{erf}\left(\frac{|v|}{\sqrt{2}\sigma} \right)}{\text{erf}\left(\frac{K}{\sqrt{2}\sigma} \right)} \tag{4.1}$$

$\zeta(v)$, defined in Eq. 4.1, produces a set of random numbers between 0 and 1 when applied to a finite set of $v(t)$'s chosen at equally spaced times and solving Eq. 3.4.

4.2 Sampling Interval

Assuming the correlation function expressed in Eq. 3.9 holds for large enough time series, picking a sampling interval of $\triangle t = 10$ allows sampling for a series up to $t = 10^{10}$, or equivalently, 10^9 generated random values. However, for a larger time

series and number of generated values N, the following requirement can be derived Knuth (1997):

$$\frac{1}{2}\ln(N) < \triangle t \tag{4.2}$$

4.3 History Function Restriction

Using the map appearing in Eq. 4.1, negative and positive history functions ϕ will generate the same numbers. Although the mapped time series is itself random, it would be useful from a RNG perspective to know that two different ϕs produce a different set of random numbers. Thus, ϕ can be picked as either always positive or always negative to avoid two same sets of generated numbers for two different ϕs. Furthermore, the sine function symmetry and shift properties also restrict ϕ, since $|\sin(v)| = |\sin(n\pi \pm v)|$ for any integer n. Thus, sets of numbers generated from different time series with different ϕ's should satisfy the following restriction:

$$\begin{aligned} &\phi_i = (0,1) \setminus \left(\phi_j = \left(\phi_i \pm \frac{n}{2\beta}\right) \cup \left(-\phi_i \pm \frac{n}{2\beta}\right)\right), \\ &n \in [1,2,3,...),\ j \neq i \end{aligned} \tag{4.3}$$

4.4 Problems with Generation

Generating random numbers with the scheme presented in this section is problematic for two reasons.

1. Generating numbers this way is slow. Sampling at every $\triangle t = 10$ requires on the order 10^4 computations for a single randomly generated number.
2. The map 4.1 is hard to apply for random number generation because $v(t)$ may take the value of the maximum K. More precisely, when a truncated Gaussian is mapped with the $\zeta(v)$ map appearing in Eq. 4.1, if a value where $v(t) = K$ happens to be sampled, it is mapped to exactly 1, which is not in the desired interval [0,1). This value can be individually removed from the set of generated random numbers, but such a procedure may be inconvenient.

5 LSF-DDE-RNG

5.1 LSF Scheme

Although Eq. 3.4 used as explained in Section 4 has no theoretical limit on the number of possible generated values, it is slow. In the last 10 years, different schemes have been presented to generate random numbers that use feedback laser mechanisms Reidler et al. (2009), Hirano et al. (2010), Oliver et al. (2013). In such work, the measurements taken from a laser system yield Gaussian-distributed values which are used to generate random numbers. Two steps are employed: digit discarding and post-processing. The

digit discarding involves only considering a certain amount of least significant bits (or digits) while completely discarding the others. The reasoning of Reidler et al. states that this allows fast random number generation, provided that the sampling rate is much slower than the chaos affecting the given least significant bits Reidler et al. (2009). Also, Oliver et al. (2013) state that "the autocorrelation function of the captured time series data is also affected by bit truncation, in such a way that residual correlations in the original dynamics are destroyed, and thus allowing for an increase in the rate of random bit generation."

For the post-processing, different schemes have been employed. Reidler et al. suggest taking differences between measured values to generate each random number as well as using an xor operation on the least significant bits Reidler et al. (2009). Oliver et al. suggest using an appropriate sampling rate after digit discarding Oliver et al. (2013).

Since the solution to Eq. 3.4 is analogous to the measurement from these systems as the time series yields Gaussian-distributed values, a similar scheme can be applied here, which will henceforth be referred to as LSF-DDE-RNG (Least Significant Figures).

Although bit truncation can flatten a Gaussian distribution into an approximately uniform distribution Oliver et al. (2013), this approximation breaks down as the number of generated values goes to infinity. This can be shown by calculating the expected probability for each set of possible bits resulting from the Gaussian distribution after digits (or bits) are discarded. Thus, the two-step digit discarding scheme can be applied with Eq. 3.4, but a mapping function, equivalent to Eq. 4.1, which maps values to a uniform distribution, should be used before discarding digits. The post-processing for the LSF-DDE-RNG, presented here, is the use of a sufficient sampling rate.

5.2 Revised Mapping Interval

Discarding a certain number of decimal digits from samples of numbers taken from a uniform distribution in [0,1) also yields a uniform distribution in the interval [0,1). A similar map to 4.1 can be used to map samples from the solution to Eq. 3.4 to a uniform interval, which does not involve the maximum K:

$$\xi(v) = erf\left(\frac{|v|}{\sqrt{2}\sigma}\right) \tag{5.1}$$

This equation allows mapping the solution from Eq. 3.4 to a uniform distribution in the interval $[0,erf(\frac{K}{\sqrt{2}\sigma})]$ where $erf(\frac{K}{\sqrt{2}\sigma})$ is close to one. If more than zero decimal digits are discarded from samples taken from this interval, the samples will then be uniformly distributed in the interval (0,1]. The value for σ computed as from equation 3.7 has been revised and better follows the following relationship:

$$\sigma = \frac{1}{\sqrt{(12.62677\beta - 11.00613)}} \tag{5.2}$$

Equation 5.2 has negligible error if β is picked between 20 and 50 and if digit discarding is used. In other words, Eqs. 3.4 and 5.1 with 5.2 can be used together for random number generation employing digit discarding.

5.3 Digits Discarded

After mapping to a uniform distribution in the interval [0, 1) and discarding digits, the values produced are strictly positive. The appropriate empirical autocorrelation function can be expressed as

$$\rho = \frac{1}{N-1} \sum_{n=1}^{N-1} (v_n v_{n+1} - 0.25) \tag{5.3}$$

Using the theoretical distribution of ρ L'Ecuyer and Simard (2007), it is possible to test whether values behave as they should if they were truly drawn from a random sequence. This can be done by calculating the p values from the autocorrelation for samples generated after discarding m decimal digits. This also indicates whether or not there is correlation between successive values.

The p values are shown in Table 1. P values are rounded to 10^{-2}. Values between 0.01 and 0.99 are considered here to indicate negligible correlation between successive values. Here, the number of generated values tested was $N = 10^8$, and the values were sampled at every $\triangle t = 0.001$. The time series solution was obtained as described in Sect. 3. The tabulated p values suggest that discarding between 4 to 12 digits destroys correlation between successive values in the time series. Furthermore, digits 14 and above should not be used in random number generation. Double float data type precision was used.

Table 1 Autocorrelation p value after discarding m decimal digits for $N = 10^7$

m no. of discarded digits	p value
1	0
2	0
3	1.00
4	0.22
5	0.10
6	0.54
7	0.32
8	0.71
9	0.06
10	0.86
11	0.38
12	0.87
13	1.00

Formally, digit discarding for a sample point $v(t)$ can be written as the function $DD(v)$ where

$$DD(v) = \frac{10^m v - floor(10^m v)}{10^m} \tag{5.4}$$

In the above equation, DD is the digit discarding function, $v(t)$ is a sample point from the time series, m is the number of discarded digits, and $floor$ represents the integer floor function.

5.4 Lyapunov Exponent

Wolf et al. describe Lyapunov exponents as "the average exponential rates of divergence or convergence of nearby orbits in phase space Wolf et al. (1984)." In the case of chaotic trajectories, if initially separated (e.g., with a small difference in the initial function), the orbits diverge on average exponentially, at least until the limit provided by the volume of the phase space. And so, chaotic dynamical systems are characterized by positive Lyapunov exponents (LE's) Hand and Finch (1998). Using the methods provided by Breda and Van Vleck (2013), it was found that the maximal Lyapunov exponent λ was 2.4496, as averaged over t = 100 for Eq. 3.4 and β=32.1357941. The computed λ is shown in Fig. 3. Reidler et al. stipulate that a requirement for random number generation should be that the "sampling rate (clock period), is slow enough in comparison with the strength of the chaos, controlled by the spectrum of the Lyapunov exponents Reidler et al. (2009)." It remains an open question as to the direct relationship between the so-called clock period and the Lyapunov exponent, as implied by Reidler et al. However, we here speculate on a possible requirement between the amount of digits discarded m and the maximal Lyapunov exponent λ:

$$e^{\lambda \triangle t} - 1 > 10^{-m} \tag{5.5}$$

Equation 5.5 holds in the case studied here, where Eq. 3.4 is used with digit discarding, $-12 \leq m \leq -4$ and $\triangle t$=0.001. It is intuitive that if λ were larger, less discarding of digits may be required (i.e., smaller m), and conversely, if λ was negative, no random number generation could be achieved. The exponential functional form is intuitively

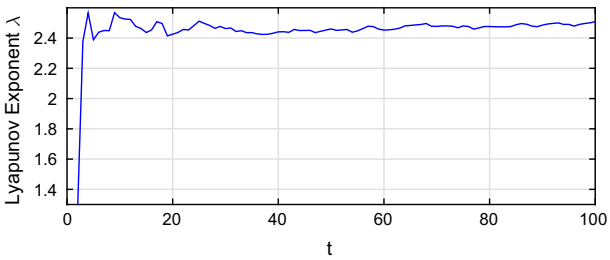

Fig. 3 Maximal and averaged Lyapunov exponent for Eq. 3.4 as computed by the methods of Breda and Van Vleck (2013)

suggested as it can quantify the divergence of initially separated trajectories. Anyhow, further work should clarify whether Eq. 5.5 holds for other systems (e.g., different β, the experimentally realized laser systems, different DDE's). Also, besides Eq. 5.5, it remains unknown whether there exists a quantitative statement of the claim from Reidler et al. quoted above. Although outside the scope of this paper, further study could determine to what extent the RNG laser feedback systems Reidler et al. (2009), Hirano et al. (2010), Oliver et al. (2013) are analogous to RNG generators using DDEs, such as the one presented in this work.

5.5 Sampling Rate

Although the autocorrelation function indicates no correlation between successive outputted values in the time series at every $\triangle t = 0.001$ with $m = 4$ to $m = 12$ discarded digits, values generated with this minimal sampling rate fail certain statistical tests of randomness. The values of the sampling rate had to be increased to $\triangle t = 0.002$ for the values to pass all required statistical tests. Figure 4 shows all the steps needed to produce random numbers for the LSF-DDE-RNG. First, in Fig. 4 (a), Eq. 3.4 is solved with the Euler method as explained in Sect. 4. (b), the mapping function 5.1 is used, as explained in Sect. 5.2. It is used for every other (discrete) time series point, since $\triangle t = 0.002$ was picked. (c), Every two successive values from the mapped time series (red circles) yield a random number (black $\times$) as shown in d), after m=8 digits are discarded. In this case, two mapped values (red circles) must be used for one 10-digit random number (2^{31} bits of resolution is standard), since keeping more than 9 digits from one number has been shown in Table 1 to be undesirable (i.e., digits above the 4th and below the 14th are preferred). In other words, in this scheme, exactly four time series points yield one random number.

5.6 Statistical Tests of Randomness

Testing randomness of sets of numbers is quite involved. It requires checking both global randomness and local randomness. While testing these generated numbers, the null hypothesis is that all the generated numbers are truly random. Many different tests have been proposed, for which, incidentally, the question of interdependency remains an open problem Soto (1999). L'Ecuyer and Simard L'Ecuyer and Simard (2007) have compiled batteries of tests judged to be adequate in the testing of randomness. The most stringent of these batteries is TestU01's "BigCrush." The tests involved, among others, the collision test, run test. and the poker test. For example, the run test checks whether there are too few or too many monotonically increasing and decreasing subsequences. In the collision test, equally spaced intervals and the number of repeated values for a same bin, or collisions, are compared to the expected amount. In the poker test, subsequences of values are treated as poker hands and are studied against expected hands.

Here, battery BigCrush is used from the TestU01 library, which tests random numbers with up to 2^{31} bits of resolution. This means a 10 decimal random number is sufficient. Here, the digits from 9 to 13 are used from sampled points in the time

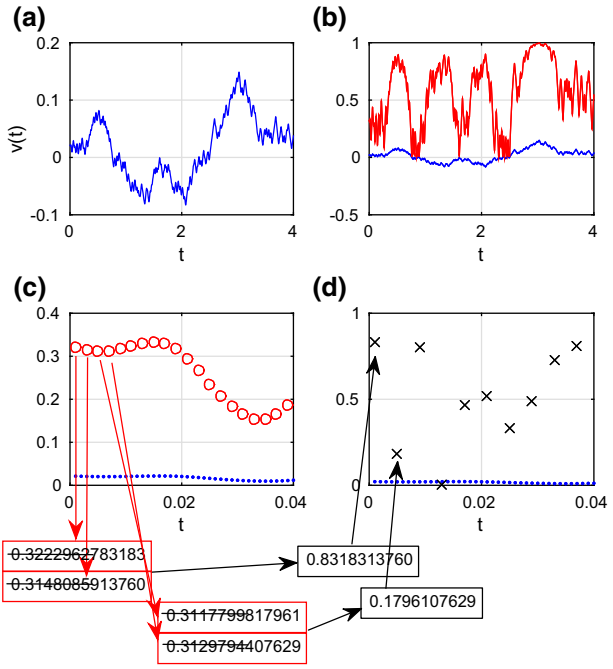

Fig. 4 Time series solution for Eq. 3.4 is shown in *blue*, where $\beta = 32.1357941$ and $\phi = 0.8876641$. **b** Equation 5.1 used on time series values (*solid blue*) to yield mapped values (*solid red*). **c** Same as **b**, but with a shorter timescale. The first four time series values (*red circles*) to be used in **d** as random numbers (*black* ×), after m = 8 digits are discarded, are explicitly shown in *red boxes*. The digit discarding is shown with a *black strikethrough* (Color figure online)

series, as shown in Fig. 4. In other words, $m = 8$ digits are discarded from every sampled point. A set of two values from the time series is needed for each 10-digit random number as using digits 9 to 13 yields five decimal digits. The sampling interval is picked to be $\triangle t = 0.002$.

All tests from the BigCrush battery were passed when applied. 2.7×10^{11} numbers generated from a single time series were verified for randomness using 160 statistical tests, including collision tests, run tests, and poker tests. The parameters used were $\beta = 32.1357941$ and $\phi = 0.8876641$, and the results of BigCrush have here been omitted due to their length. The criteria and specificity of the tests can be found in L'Ecuyer and Simard (2007). 10^8 numbers generated from the LSF-DDE-RNG are illustrated in Fig. 5a by employing two consecutive numbers as x- and y-coordinates for 10^4 points.

The same battery of stringent tests has not been applied for different β and ϕ, but from Lei and Mackey it is expected that other values of (β, ϕ), following the prescriptions of Sects. 4.1 and 4.3, could provide different, but also sufficiently random, sets. The highest precision for which β and ϕ yield significantly different time series is unknown.

Finally, we note that though some of the proposed feedback laser RNGs use the DIEHARD or NIST tests, Reidler et al. (2009), Hirano et al. (2010) the LSF-DDE-

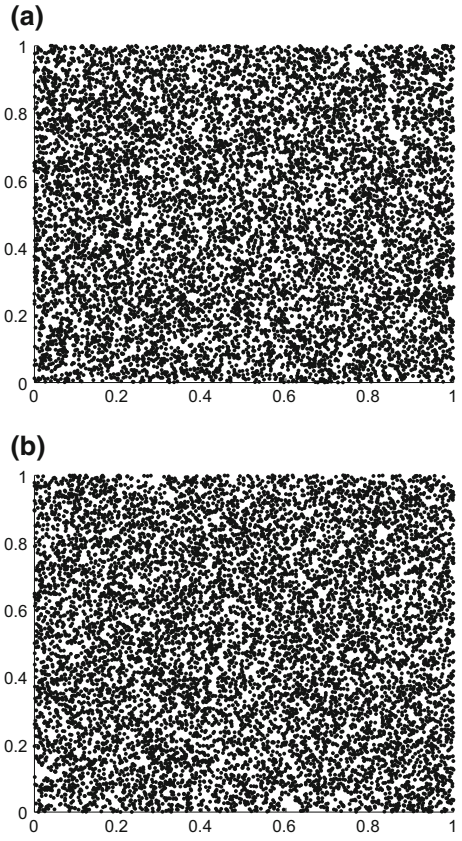

Fig. 5 **a** 10000×10000 points for which each two successive random numbers generated by the LSF-DDE-RNG are assigned to a set of (x, y)-coordinates. **b** 10000×10000 points for which each two successive random numbers generated by MT19937 are assigned to a set of (x, y) coordinates

RNG with the above parameter β and initial function ϕ is likely at least as random since the tests that were passed were much more stringent L'Ecuyer and Simard (2007).

6 Discussion and Benchmarking

In 2005, Falcioni et al. discussed three requirements for using a deterministic chaotic system as a RNG: a sufficiently high Kolmogorov–Sinai (KS) entropy, small time correlation, and large period Falcioni et al. (2005). Although a discussion of the KS entropy is outside the scope of this paper, it is notable that a sufficiently large maximal Lyapunov exponent, as discussed in Sect. 5.4, is a different but comparable metric Falcioni et al. (2005) to use for RNG suitability. The sufficient weakness of the time correlation for the LSF-DDE-RNG was studied in Sect. 5.3. What remains to discuss is the question of the period length, which here cannot be feasibly solved numerically. However, previous scholarship has argued statistically that the period length for randomly evolving discrete dynamical systems should be on the order of the square root of the number of discrete states of the given system Falcioni et al. (2005), Coste and Henon (1986). Thus, since for the LSF-DDE the initial function and subsequent

states involve up to 10^{16000} discrete states, the period is likely on the order of 10^{8000}. Furthermore, increasing the precision (e.g., long float instead of double float) should allow even longer periods and prevent breakdown from computer precision should it ever be necessary to reach sequences of 10^{8000} random numbers or more.

For benchmarking, a MRG was considered L'Ecuyer (1999). A specific implementation already tested for good speed and randomness was used, MRG32k3a L'Ecuyer (1999). Using C, $N = 10^8$ numbers were generated with the LSF-DDE-RNG and MRG32k3a. MRG32k3a took about 4 seconds for 10^8 values. The LSF-DDE-RNG generated 10^8 random numbers in 26 seconds, using a computer running Linux with a 2.50 GHz Intel i5-2520M CPU. The LSF-DDE-RNG's generation time scales as N.

Although the LSF-DDE-RNG generator is about 7 times slower than MRG32k3a, its period is likely on the order of 10^{8000}, significantly more than MRG32k3a, which has a period of 2^{191}. Even though both these are very high, it is advisable to use many fewer than all the possible numbers generated from a given generator with a period, so the number of usable numbers is much less L'Ecuyer (1999).Also, L'Ecuyer and Simard, in 2007, showed that many widely employed RNGs failed their "Big Crush" battery of tests, and the generation time for 10^8 random numbers for different generators was also reported, including MRG32k3a L'Ecuyer and Simard (2007).

Figure 5b shows 10^8 numbers generated by the very widely used MT19937 "Mersenne twister" (using MATLAB software). Two successive numbers are used for a set of x and y coordinates of $10^4 \times 10^4$ points. One may look at the random numbers produced by LSF-DDE-RNG in Fig. 5a and compare them to Fig. 5b. Although the Mersenne twister failed two tests from "Big Crush," L'Ecuyer and Simard (2007), it is impossible to tell the quality of random number generators from visual inspection alone.

7 Conclusion and Further Work

It is intuitive that chaotic time series from a DDE could produce random numbers, and the work detailed here proposes one such method. The digit discarding technique borrowed from feedback laser systems raises questions about the technique's relationship to the maximal Lyapunov exponent λ, for which a possible relation is speculated in Eq. 5.5.

Although it is the first of its kind, our proposed RNG, the LSF-DDE, produces random numbers on the same scale of quality, albeit slower, than its currently widely used counterparts MT19937 and MRG32k3a. It is not unimaginable that, like MRGs with LCGs, the ratio of speed to quality of our presented algorithm can be, in the future, dramatically increased due to improvements, perhaps in the underlying algorithm. It does nonetheless feature a fundamental difference from other popular software generators as it does not have a practical period.

Equation 3.4 seems to be able to serve as a RNG, and similar equations may also be useful for RNGs. Equations 3.1 and 3.2 have solutions with similar behavior to solutions from Eq. 3.4. And so, tests could be carried out to examine their usefulness as RNGs. Finally, the output quality of the proposed generator could be further checked

by running tests for larger sequences of numbers. Different values of β and ϕ could also be used to verify for similar randomness as these quantities are varied.

Acknowledgments This work was supported by the Natural Sciences and Engineering Research Council (NSERC, Canada). We would like to thank Dimitri Breda for providing the scripts that were used to calculate the Lyapunov exponents and Tony Humphries, Erik Van Vleck, Joshua Lackman, and Serhiy Yanchuk for their help.

References

Beck, C.: Higher correlation functions of chaotic dynamical systems-a graph theoretical approach. Nonlinearity **4**, 1131–1158 (1991)

Breda, D., Van Vleck, E.: Approximating Lyapunov exponents and sacker-sell spectrum for retarded functional differential equations. Numer. Math. **126**, 225–257 (2013)

Chew, L.Y., Ting, C.: Microscopic chaos and Gaussian diffusion processes. Physica A **307**, 275–296 (2002)

Coste, J., Henon, M.: Disordered Systems and Biological Organization. Springer, Berlin (1986)

Eichenauer-Herrmann, J.: Pseudorandom number generation by nonlinear methods. Int. Stat. Rev. **63**, 247–255 (1995)

Falcioni, M., Palatella, L., Pigolotti, S., Vulpiani, A.: Properties making a chaotic system a good pseudo random number generator. Phys. Rev. **E 72**, 016220 (2005)

Hand, L.N., Finch, J.D.: Analytical Mechanics. Cambridge University Press, Cambridge (1998)

Hirano, K., Yamazaki, T., Morikatsu, S., Okumura, H., Aida, H., Uchida, A., Yoshimori, S., Yoshimura, K., Harayama, T., Davis, P.: Fast random bit generation with bandwidth enhanced chaos in semiconductor lasers. Opt. Express **18**, 5512–5524 (2010)

Knuth, D.E.: Seminumerical Algorithms, 3rd ed., The Art of Computer Programming, Vol. 2 (Addison-Wesley, 1997)

L'Ecuyer, P.: Good parameters and implementations for combined multiple recursive random number generators. Oper. Res. **47**, 159–164 (1999)

L'Ecuyer, P.: Handbook of Computational Statistics. Springer, Berlin (2012). Chap. 3

L'Ecuyer, P., Simard, R.: Testu01: A C library for empirical testing of random number generators. ACM T. Math. Software **33**, 22 (2007)

Lei, J., Mackey, M.C.: Deterministic Brownian motion generated from differential delay equations. Phys. Rev. E **84**, 041105 (2011)

Lewis, T.G., Payne, W.H.: Generalized feedback shift register pseudorandom number algorithm. J. Assoc. Comput. Mach. **20**, 456–468 (1973)

Mackey, M.C., Tyran-Kamińska, M.: Deterministic Brownian motion: the effects of perturbing a dynamical system by a chaotic semi-dynamical system. Phys. Rep. **422**, 167–222 (2006)

Matsumoto, M., Kurita, Y.: Twisted GFSR generators. ACM T. Model. Comput. S **2**, 179–194 (1992)

Matsumoto, M., Nishimura, T.: Mersenne Twister. ACM T. Model. Comput. S. **8**, 3–30 (1998)

Oliver, N., Soriano, M.C., Sukow, D.W., Fischer, I.: Fast random bit generation using a chaotic laser: approaching the information theoretic limit. IEEE J. Quantum Elect. **49**, 910–918 (2013)

Reidler, I., Aviad, Y., Rosenbluh, M., Kanter, I.: Ultrahigh-speed random number generation based on a chaotic semi-conductor laser. Phys. Rev. Lett. **103**, 024102 (2009)

Soto, J.: Statistical Testing of Random Number Generators. National Institute of Standards and Technology, (1999)

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

Wolf, A., Swift, J.B., Swinney, H.L., Vastano, J.A.: Determining Lyapunov exponents from a time series. Physica D **16**, 285–317 (1984)