Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Physics Communications 160 (2004) 187–203
Computer Physics Communications
www.elsevier.com/locate/cpc

Evidence of the correlation between positive Lyapunov exponents and good chaotic random number sequences

Po-Han Lee [a], Yi Chen [b], Soo-Chang Pei [c], Yih-Yuh Chen [a,d,*]

[a] *Physics Department, National Taiwan University, Taipei, Taiwan, ROC*
[b] *Municipal Chien Kuo High School, Taipei, Taiwan, ROC*
[c] *Electrical Engineering Department, Taipei, Taiwan, ROC*
[d] *Institute of Astrophysics, National Taiwan University, Taipei, Taiwan, ROC*

Received 13 October 2003; accepted 31 March 2004

Available online 1 June 2004

Abstract

Using a new method to extract the data from various one-dimensional chaotic maps, we show that there is a nice correlation between the sign of the Lyapunov exponent of the maps and whether the extracted data form a good set of pseudo-random numbers using various well-known criteria.

PACS: 05.45.+b; 05.40.+j; 02.50.sk; 02.60.-x

Keywords: Chaos; FIPS PUB 140-2; SP 800-22 tests; Diehard battery of tests; Height correlation test; Logit map

1. Introduction

The need for random and pseudo-random numbers arises in many different kinds of applications, such as simulation, image encryption, transmission of e-mail on the web, numerical analysis, decision making, etc. A pseudo-random number generator (PRNG) is a cryptographic algorithm used to generate numbers that must appear random but are necessarily predetermined. Besides being of a high-quality, a pseudo-random number generator must also possess the following desirable properties: good distribution, long period, repeatability, portability and efficiency. For a description of well-studied uniform random number generators, the readers are referred to L'Ecuyer [1]. Briefly, a PRNG is a deterministic algorithm producing a sequence $(x_i)_{i\geqslant 0}$ of numbers in $[0,1)$ which, for virtually all generators used for computer simulations, is purely periodic. The most famous PRNG widely used today are linear congruential generator LCG(M,a,b,u_0), in which the parameters M, a, b, and u_0 are all integers. The LCG produces a sequence $(u_i)_{i\geqslant 0}$ of integers by $u_{i+1} = au_i + b \pmod M$, that is, u_{i+1} is the integer remainder of dividing $au_i + b$ by M. Using the modular method, the inversive congruential generator ICG(p,a,b,u_0) and the explicit inversive generator EICG(p,a,b) are studied in Leeb and Wegenkittl [2]. The parameters of the two gener-

* Corresponding author.
E-mail addresses: leepohan@ms17.hinet.net (P.-H. Lee), huei7777@ms31.hinet.net (Y. Chen), pei@cc.ee.ntu.edu.tw (S.-C. Pei), yychen@phys.ntu.edu.tw (Y.-Y. Chen).

0010-4655/$ – see front matter
doi:10.1016/j.cpc.2004.04.001

ators are defined by a (large) prime p and an integer u with $0 \leqslant u < p$, and one lets $\bar{u} = u^{p-2} \pmod{p}$ if $u \neq 0 \pmod{\text{p}}$ and $\bar{u} = 0$, otherwise. With suitable parameters p, a, b, and u_0, the ICG produces integers $(u_i)_{i\geqslant 0}$ by $u_{i+1} = a\bar{u}_i + b \pmod{p}$, and pseudorandom numbers $(x_i)_{i\geqslant 0}$ by $x_i = u_i/p$. A similar process is used in the EICG to produce a sequence $(u_i)_{i\geqslant 0}$ of integers by $u_i = \overline{ai + b}$, and pseudorandom numbers $(x_i)_{i\geqslant 0}$ by $x_i = u_i/p$. As far as hyperplane structures and long-range correlations are concerned, the EICG has similar favorable properties as the ICG, and these points are discussed in Eichenauer-Herrmann [3]. In regard to finding good random number generators, one is referred to the detailed discussions in Hellekalek [4].

As an alternative to the more traditional PRNGs described above, one may also consider using chaotic systems. Indeed, the potential of incorporating chaos into cryptography has been under intensive investigation since Pecora and Carroll demonstrated the possibility of synchronization in chaotic systems [5]. For instance, a very primitive way of utilizing it is to directly hide the data in the chaotic signals generated by a chaotic system [6,7] or to combine standard cryptographic operations with chaos, as was done in He and Vaidya [8]. However, approaches along these lines are more appropriate for covert communications because the driving signals are put in the public channel. As it turns out, methods such as those just mentioned are more akin to the generation of pseudo-random numbers, since the only aspect of chaos that is put into use in these systems is the generation of a sequence of presumably random numbers associated with the system variables. This then brings up one important question: In what respect is chaos tied to the randomness of a PRNG? For instance, a measure of the chaotic characteristics of a dynamical system is the Lyapunov exponent, and how is it correlated to the industry-standard criteria for PRNGs? One purpose of the present work is to address this question.

In fact, the idea of applying chaos theory to generate random numbers has produced interesting works in recent years [9–15]. For instance, Collins et al. [12] have applied the logit transformation to the logistic map to produce random numbers of uniform distribution. In this work, one starts with the simple and famous logistic map defined by

$$X_{n+1} = \mu X_n(1 - X_n), \quad X_n \in (0, 1) \tag{1}$$

where $0 < \mu \leqslant 4$, and then convert the nonuniform data of the logistic map to a more uniform set by the logit transformation assuming the form

$$Z_n = \ln \frac{X_n}{1 - X_n}. \tag{2}$$

On the other hand, the above and its variants all suffer from the same shortcoming of displaying a nice relationship between X_{n+1} and X_n, which is readily revealed when one does a simple return map. This has prompted González and Pino [13] to propose the following "z-logistic map" as a modification:

$$X_n = \sin^2(\pi\theta z^n), \tag{3}$$

where z is a parameter whose choice significantly determines how smooth the return map will look.

The simple algorithm of Eq. (3) is motivated by the observation that

$$X_n = \sin^2(\pi\theta 2^n) \tag{4}$$

is a general solution to the map (1) for $\mu = 4$. In contrast, for the z-logistic map we have the nice expression

$$X_{n+1} = \sin^2\left(z \sin^{-1}\sqrt{X_n}\right) \tag{5}$$

only when z is an integer. In a sense, these authors were able to overcome the aforementioned weakness because they have incorporated into the conventional recursion formula $X_{n+1} = f(X_n)$ an extra n-dependence so that it now becomes $X_{n+1} = f(X_n, n)$. In the following, however, we will restrict ourselves to the modified form

$$X_{n+1} = 0.25\mu \sin^2\left(z \sin^{-1}\sqrt{X_n}\right) \tag{6}$$

of Eq. (5) and refer to it as the restricted z-logistic map. Later we will also compound it with one extra transformation

$$Y_n = \frac{2}{\pi} \sin^{-1}\sqrt{X_n} \tag{7}$$

to see if this might change the statistical tests on the system.

Other chaotic maps of interest include the tent map defined by

$$X_{n+1} = \begin{cases} 2\mu(1 - X_n), & 0.5 < X_n \leqslant 1, \\ 2\mu X_n, & 0 < X_n \leqslant 0.5, \end{cases} \tag{8}$$

and the sine map defined by

$$X_{n+1} = \frac{\mu}{4}\sin(\pi X_n), \quad 0 < X_n < 1. \tag{9}$$

Once again, these two maps have a simple return map and therefore must be subject to further modification if one would like to use their variables for the purpose of generating random numbers. To achieve this, we can try extracting the lower bits of the variable of a chaotic map. Here in this paper we will call it the chaotic stream cipher and abbreviate it as CSC. The extraction of the lower bits is done by chopping off the leading bits after multiplying each variable X_n of the chaotic map by some constant A. Thus, the resulting number R_n can be succinctly expressed as

$$R_n \equiv \lfloor AX_n \rfloor \pmod{S}, \tag{10}$$

where $R(n) \in Z^+$, $\lfloor\ \rfloor$ is the Gaussian symbol which returns the highest integer that is smaller than the number inside the symbol, and S is yet another constant. The tests for the randomness of the numbers generated by a chaotic system are then performed and compared with some of the most famous generators summarized in Table 1. As a brief background information, we note that the LCGs cover a wide range of quality delivered by linear generators, ranging from RANDU (worst) to FISH (best). RANDU, formerly part of IBM's Scientific Subroutine Package, exhibits an infamous devastating defect in three dimensions: its points (x_i, x_{i+1}, x_{i+2}) all lie in just fifteen parallel planes. ANSIC is the generator employed by the ANSI C rand( ) function, BSD version. MINSTD, introduced by Lewis et al. as the random number generator for IBM's System/360 [16], was later proposed as a "minimal standard" generator by Park and Miller [17]. Finally, FISH is one of the best found by Fishman and Moore [18] in an exhaustive search among all maximum period LCGs with $M = 2^{31} - 1$ and $b = 0$. ICG, EICG1, and EICG7 have been chosen arbitrarily among the maximal period inversive generators with modulus $p = 2^{31} - 1$. For convenience, we have used the "CSCLOGISTIC(μ, t_0, x_0, A, S)" to represent our chaotic pseudorandom number generator based on logistic map. In this algorithm, the free parameters one can "tune" include μ, the evolution time t_0 after which we will extract data to get random numbers, the initial condition x_0, and the numbers A and S as described in Eq. (10). For example "CSCLOGISTIC($4.0, 10, 0.25, 10^7, 256$)" means that we will use the logistic map to generate the random numbers for the parameter $\mu = 4.0$ and extract the data after 10 steps of iteration and use Eq. (10) to truncate the data to get the 8 bits random numbers.

Table 1
The various random number generators defined in Section 1

Index	Generator	Parameter
1	RANAU	LCG($2^{31}, 65539, 0, 1$)
2	ANSIC	LCG($2^{31}, 1103515245, 12345, 12345$)
3	MINSTD	LCG($2^{31} - 1, 16807, 0, 1$)
4	FISH	LCG($2^{31} - 1, 950706376, 0, 1$)
5	ICG	ICG($2^{31} - 1, 65539, 0, 1$)
6	EICG1	EICG($2^{31} - 1, 1, 0$)
7	EICG7	EICG($2^{31} - 1, 7, 0$)
8	CSCLOGISTIC7	CSCLOGISTIC($\mu, 10, 0.25, 10^7, 256$)
9	CSCLOGISTIC20	CSCLOGISTIC($\mu, 10, 0.13, 10^{20}, 256$)
10	CSCLOGIT7	CSCLOGIT($\mu, 10, 0.25, 10^7, 256$)
11	CSCYN7	CSCYN($\mu, 10, 0.25, 10^7, 256$)
12	CSCZLOGISTIC2	CSCZLOGISTIC($\mu, 2, 10, 0.25, 10^7, 256$)
13	CSCZLOGISTIC3	CSCZLOGISTIC($\mu, 3, 10, 0.25, 10^7, 256$)
14	CSCZLOGISTIC5	CSCZLOGISTIC($\mu, 5, 10, 0.25, 10^7, 256$)
15	CSCSINE7	CSCSINE($\mu, 10, 0.25, 10^7, 256$)
16	CSCTENT7	CSCTENT($\mu, 10, 0.25, 10^7, 256$)
17	CSCTENT20	CSCTENT($\mu, 10, 0.13, 10^{20}, 256$)

The parameter μ of the generators 8 through 15 is 3.97; and of generators 16 and 17 is 0.9. The transformed data of generators studied are still based on Eq. (10).

The main purpose of the present work is twofold: Firstly, to show that CSC system can pass the established criteria for random numbers proposed in NIST [19,20], the Diehard battery of tests by Marsaglia [21], and the random walk tests of Vattulainen [22]. Secondly, to find whether there is a perfect correlation between the Lyapunov exponent and the passing criteria mentioned above.

2. Some famous statistical tests of random number generators

In order to test the CSC method described in Section 1, we have performed certain statistical tests for various chaotic maps we proposed. These tests include FIPS PUB 140-2 tests [19], SP 800-22 [20], Diehard battery of tests [21], and the random walk test [22]. For completeness and for reference, we give in the following a brief description of each of the aforementioned tests.

2.1. FIPS PUB 140-2 tests

This set of tests is meant to be a general purpose test suite which can be conveniently grouped into four, totaling 16 items, as specified in the FIPS PUB 140-2 tests. First, one considers a single bit stream of 20,000 consecutive bits output from the generator. The bits are then subjected to each of the following tests below. Failure to meet any of the specified criteria means that the sequence must be rejected. The four tests termed *monobit test*, *poker test*, *runs test*, and *long run test* are:

Monobit Test counts the number $\mathbf{X}$ of ones in the 20,000 bit stream. The test is passed if $9{,}725 < \mathbf{X} < 10{,}275$.

Poker Test starts by dividing the 20,000 bit stream into 5,000 contiguous 4-bit segments. One then counts and stores the number of occurrences of each of the 16 possible 4-bit values. Denoting $f(i)$ as the number of each 4-bit value i where $0 \leqslant i \leqslant 15$, one then evaluate the following:

$$\mathbf{X} = \frac{16}{5000}\left(\sum_{i=0}^{15}\left[f(i)\right]^2\right) - 5000. \tag{11}$$

The test is passed if $2.16 < \mathbf{X} < 46.17$.

Table 2
The required interval for runs test

Length of run	Required interval
1	2315–2685
2	1114–1386
3	527–723
4	240–384
5	103–209
> 6	103–209

In Runs Test one considers a run defined as the maximal sequence of consecutive bits of either all ones or all zeros, which is part of a 20,000 bit sample stream. The incidences of runs (for both consecutive zeros and consecutive ones) of all lengths ($\geqslant 1$) in the sample stream should be counted and stored. The test passes if the number of runs that occur (of lengths 1 through 6) is each within the corresponding interval specified in Table 2. This must hold for both zeros and ones; that is, all 12 counts must lie in the specified interval. For the purpose of this test, runs of greater than 6 are considered to be of length 6.

For Long Run Test one considers a run of length 26 or more (of either zeros or ones). On the sample of 20,000 bits, the test passes if there are *no* long runs.

To test the quality of the random bits generated, we will have to check a total of sixteen items (one for the monobit test, one for the poker test, twelve for the runs test, and two for the long run test). The test results will be discussed below.

2.2. SP 800-22 Test

The NIST test suite, SP 800-22, is a statistical package consisting of 16 tests, as listed in Table 3, that were developed to test the randomness of (arbitrarily long) binary sequences produced by either hardware or software based cryptographic random or pseudorandom number generators. These tests focus on a variety of different types of non-randomness that could exist in a sequence. Some tests are decomposable into a variety of subtests. For example, considering the Frequency Test, the focus of the test is the proportion of zeroes and ones for the entire sequence. The purpose of this test is to determine whether the number of ones and zeros in a sequence are approximately the same as would be expected for a truly random sequence. Listed below are some excerpts from the test suite.

Table 3
SP 800-22 tests

Index	The items of SP 800-22 tests	The number of subitems
1	The Frequency (Monobit) Test	1
2	Frequency Test within a Block	1
3	The Cumulative Sums (Cusums) Test	2
4	The Runs Test	1
5	Test for the Longest-Run-of-Ones in a Block	1
6	The Binary Matrix Rank Test	1
7	The Discrete Fourier Transform (Spectral) Test	1
8	The Non-overlapping Template Matching Test	148
9	The Overlapping Template Matching Test	1
10	Maurer's "Universal Statistical" Test	1
11	The Approximate Entropy Test	1
12	The Random Excursions Test	8
13	The Random Excursions Variant Test	18
14	The Serial Test	2
15	The Lempel–Ziv Compression Test	1
16	The Linear Complexity Test	1
Sum	16	189

For each subitem of SP 800-22 Test, a set of P-values (corresponding to the set of sequences) is produced. For a fixed significance level, a certain percentage of P-values are expected to indicate failure. For example, if the significance level is chosen to be 0.01 (i.e. $\alpha = 0.01$), then about 1% of the sequences are expected to fail. A sequence passes a statistical test whenever the P-value is greater than or equal to α and fails otherwise. For each statistical test, the proportion of sequences that passes is computed and analyzed accordingly. The interpretation of empirical results is threefold. Case 1: The analysis of the P-values does not indicate a deviation from randomness. Case 2: The analysis clearly indicates a deviation from randomness. Case 3: The analysis is inconclusive. The interpretation of empirical results can be conducted in any number of ways. Two approaches NIST has adopted include (1) the examination of the proportion of sequences that pass a statistical test and (2) the distribution of P-values to check for uniformity. In the event that either of these approaches fails (i.e. the corresponding null hypothesis must be rejected), additional numerical experiments should be conducted on different samples of the generator to determine whether the phenomenon was a statistical anomaly or a clear evidence of non-randomness. For example, if 1000 binary sequences were tested (i.e. $m = 1000$), $\alpha = 0.01$ (the significance level), and 996 binary sequences had P-values $\geqslant 0.01$, then the proportion is $996/1000 = 0.9960$. The range of acceptable proportions is determined using the confidence interval defined as

$$\hat{p} \pm 3\sqrt{\frac{\hat{p}(1-\hat{p})}{m}}, \tag{12}$$

where $\hat{p} = 1 - \alpha$ and m is the counting times of test for a certain size of sequence n, and the whole size of random number is M. As a modest test, we set up in this paper the parameter $n = 10^6$ and $m = 15$, and the size of random number is $M = 1.5 \times 10^7$.

If the proportion falls outside of this interval, then there is evidence that the data are nonrandom. Note that other standard deviation values could be used. For the example above, if $m = 15$, the confidence interval is 0.99 ± 0.077071 (i.e. the proportion should lie above 0.912929). The confidence interval was calculated using a normal distribution as an approximation to the binomial distribution, which is reasonably accurate for large sample sizes (e.g., $n \geqslant 1000$). The distribution of P-values is examined to ensure uniformity. Uniformity may also be determined via an application of a χ^2 test and the determination of a P-values corresponding to the Goodness-of-Fit Distributional Test on the P-values obtained for an arbitrary statistical test (i.e.

a P-value of the P-values). This is accomplished by computing

$$\chi^2 = \sum_{i=1}^{m} \frac{F_i - s/m}{s/m}, \tag{13}$$

where F_i is the number of P-values in sub-interval i, and s is the sample size. A P-value is calculated such that P-value$_T$ = igamc$((m-1)/2, \chi^2/2)$, where the igamc( ) is the incomplete gamma function, which, together with the gamma function, are defined, respectively, by

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, dt \quad \text{(gamma)}, \tag{14}$$

$$Q(a,x) = \frac{1}{\Gamma(a)} \int_x^\infty e^{-t} t^{a-1}\, dt \quad \text{(incomplete gamma)}, \tag{15}$$

with $Q(a,0) = 1$ and $Q(a,\infty) = 0$.

If P-value$_T \geqslant 0.0001$ and the portion of passing random numbers is higher than the criteria by Eq. (12), then the sequences can be considered to be uniformly distributed. See Table 3. We would also like to mention that the Non-overlapping Template Matching Test is so stringent that it is not at all easy to pass all the 148 subitems in the test suite. In our test, the criteria about each test was set up so that a pass is granted when the statistics of the random number sequence satisfies the criteria of proportion and P-value$_T$.

2.3. Diehard battery of tests

Diehard battery of tests, a set of powerful statistical tests for testing randomness of sequences of numbers, is proposed by Marsaglia, and the Diehard program written by B. Narasimhan can be found on the Web [23]. The Diehard test suite is important because, quoting the original author, it seems to be one of the most powerful general tests of randomness. This belief comes from the observation that many software and hardware generators which claim "perfect randomness" actually fail one or more sections of Diehard. In testing longer and longer sequences of random bits, the Diehard battery of tests are reported to have the ability of eventually detecting these defects. An updated version, the PowerTest, has also been proposed which contains many other interesting tests besides those in the original Diehard battery of tests. However, because of the stringent requirements in the Diehard test suite, a generator which passes Diehard battery of tests can be considered good as a rule of thumb.

The Diehard battery of tests consist of 18 different, independent statistical tests, as listed in Table 4. Results of tests are so called "p-values." In the PowerTest version of Diehard, these values are of Kolmogorov–Smirnov type, which means their values are real, between 0 and 1. An individual test is considered to be failed if p value approaches 1 closely, for example $p > 0.9999$. See the PowerTest description for further details. As a result, the Diehard battery of tests consist of 18 statistical tests and 215 subitems in the tests.

As a reminder from the original author, we should note that most of the tests in Diehard return a p-value, which should be uniform on $[0, 1)$ if the input file contains truly independent random bits. Those p-values are obtained by $p = F(X)$, where F is the assumed distribution of the sample random variable X, which is often normal. But that assumed F is just an asymptotic approximation, for which the fit will be worst in the tails. Thus one is reminded not to be alarmed by the occasional occurrences of having p-values near 0 or 1, such as 0.0012 or 0.9983. When a bit stream really fails, one will get p's of 0 or 1 to six or more places.

2.4. Random walk test

The Random walk test proposed by Vattulainen [22] is a framework for testing the quality of random numbers in parallel calculations. The key idea is to study the cross-correlations between distinct sequences of random numbers via correlations between various diffusing random walkers, each of which is governed by a distinct random number sequence. Such method aims at the property of two types of correlations, that is, correlations within a single random number sequence $\{r_i\}^{(k)}$ and correlations between *distinct*[1] ran-

[1] There are many ways to construct the sequences $\{r_i\}^{(1)}, \ldots, \{r_i\}^{(m)}$ for the processors one through m. We used random numbers $\{r_i\} = r_1, \ldots, r_\Omega, r_{\Omega+1}, \ldots, r_{2\Omega}, \ldots$ generated by a single pseudo-random number generator to make non-overlapping sequences $\{r_i\}^{(1)} = r_1, \ldots, r_\Omega, \{r_i\}^{(2)} = r_{\Omega+1}, \ldots, r_{2\Omega}$ and so forth. Other possibilities for constructing the sequences are given in, e.g., [24].

Table 4
The Diehard battery of tests

Index	The items of Diehard battery of tests	The number of subitems
1	The BIRTHDAY SPACINGS TEST	9
2	The OVERLAPPING 5-PERMUTATION TEST	2
3	The BINARY RANK TEST for 31×31 matrices	1
4	The BINARY RANK TEST for 32×32 matrices	1
5	The BINARY RANK TEST for 6×8 matrices	25
6	The BITSTREAM TEST	20
7	The OPSO TEST(Overlapping-Pairs-Sparse-Occupancy)	23
8	The OQSO TEST(Overlapping-Quadruples-Sparse-Occupancy)	28
9	The DNA TEST	31
10	The COUNT-THE-1's TEST on a stream of bytes	12
11	The COUNT-THE-1's TEST for specific bytes	25
12	The PARKING LOT TEST	10
13	The MINIMUM DISTANCE TEST	1
14	The 3DSPHERES TEST	20
15	The SQEEZE TEST	1
16	The OVERLAPPING SUMS TEST	10
17	The RUNS TEST	4
18	The CRAPS TEST	2
Sum	18	215

dom number sequences $\{r_i\}^{(k)}, \ldots, \{r_i\}^{(m)}$ generated by CSC systems and other famous PRNGs. Here we consider the case where the sizes Ω_k of sequences $\{r_i\}^{(k)}$ are equal for all k. Random numbers r_i are uniformly distributed between zero and one. In the height correlation test of this method, we consider the position x_i of a one-dimensional (1D) random walker versus the number of jumps made, i. The position $x_t = \sum_{i=1}^{t} \delta x_i$ is a sum of displacements δx_i, which are random variables

$$\delta x_i = \begin{cases} +1 & \text{if } r_i \leqslant 1/3, \\ 0 & \text{if } 1/3 < r_i \leqslant 2/3, \\ -1 & \text{otherwise.} \end{cases} \tag{16}$$

In this fashion, we construct the paths $x_i^{(1)}$ and $x_i^{(2)}$ from the sequences $\{r_i\}^{(1)}$ and $\{r_i\}^{(2)}$, respectively. The height between the two random walkers is then defined as $h_t = x_t^{(1)} - x_t^{(2)}$, whose correlation function $H_t \equiv \langle |h_t - h_0| \rangle \sim t^{\phi}$ is known to decay asymptotically as a power law with an exponent $\phi = 1/2$ [25]. Deviations from $\phi = 1/2$ are expected, if H_t does not correspond to a random process. In this work, the height correlation function H_t was investigated up to $\Omega = 2000$ with $M = 10^7$ independent runs.

The other test we have adopted is the so called S_N test, which is more general in the sense that it can be applied to study any number of random walks. In one dimension, N random walkers move simultaneously without any interaction such that, at any jump attempt, they can make a jump to the left or to the right with equal probability. After $t \gg 1$ jumps by all random walkers, the mean number of sites visited, $S_{N,t}$, has an asymptotic form $S_{N,t} \sim f(N)t^{\gamma}$, with the scaling function $f(N) = (\ln N)^{1/2}$ and $\gamma = 1/2$ [26]. The value of γ observed serves as a measure of correlations. For the similar process, the S_N test was set up to $\Omega = 4000$ with $M = 10^8$ independent runs.

3. Test results and the clear correlation with Lyapunov exponents

Trying to make use of the chaotic nature of simple maps, many researchers have discussed the possibility of using the logistic map to generate random numbers [12–15]. One distinct feature of chaotic maps is that at least one Lyapunov exponent of the systems is

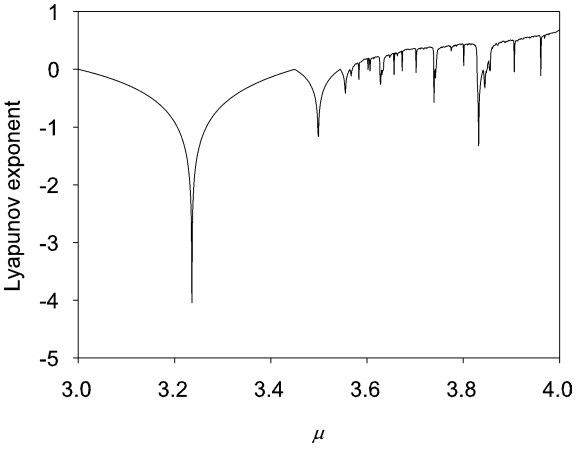

Fig. 1. The Lyapunov exponents of the logistic map.

positive for certain parameter regimes. But since the randomness one would like to see on a random number generator clearly must be correlated to the diverging nature of the trajectories of a chaotic map, which is tied to the existence of a positive Lyapunov exponent, it is natural to investigate just how good the correlation is. The computation of the Lyapunov exponent λ for the logistic map can be facilitated by noting

$$\mathrm{d}X_{n+1} = \mu(1-2X_n)\,\mathrm{d}X_n \tag{17}$$

and using the well-known formula

$$\lambda = \lim_{N\to\infty} \frac{1}{N}\sum_{n=0}^{N} \ln\left|\frac{\mathrm{d}f(X_n)}{\mathrm{d}X_n}\right|. \tag{18}$$

This is shown in Fig. 1 as a function of the tuning parameter μ.

To exhibit the correlation between the Lyapunov exponent and the success rate (of passing the random number generator criteria), we have taken CSCLOGISTIC$(3.97, 0, 0.13, 10^{20}, 256)$ in Eq. (10) as a concrete example. (Not surprisingly, we get more qualified random number sequences when we chop off more higher bits by adopting a larger value for A.) Thus, we have

$$\begin{aligned}
x_1 &= 3.97x_0(1-x_0) = 0.44900700000000000000,\\
x_2 &= 3.97x_1(1-x_1) = 0.98217686438547000000,\\
x_3 &= 3.97x_2(1-x_2) = 0.69496721662042407100,\\
x_4 &= 3.97x_3(1-x_3) = 0.25672770154087592061,\\
x_5 &= 3.97x_4(1-x_4) = 0.75754979754558697859,\\
x_6 &= 3.97x_5(1-x_5) = 0.72916236408338182336,\\
&\ldots,\\
x_{98} &= 3.97x_{97}(1-x_{97}) = 0.35779450320384782020,\\
x_{99} &= 3.97x_{98}(1-x_{98}) = 0.13696214658296892037,\\
&\ldots
\end{aligned} \tag{19}$$

so that, using the CSC method,

$$\begin{aligned}
R_1 &\equiv 10^{20}x_1 \pmod{256} \equiv 0,\\
R_2 &\equiv 10^{20}x_2 \pmod{256} \equiv 192,\\
R_3 &\equiv 10^{20}x_3 \pmod{256} \equiv 70,\\
R_4 &\equiv 10^{20}x_4 \pmod{256} \equiv 125,\\
R_5 &\equiv 10^{20}x_5 \pmod{256} \equiv 131,\\
R_6 &\equiv 10^{20}x_6 \pmod{256} \equiv 192,\\
&\ldots,\\
R_{98} &\equiv 10^{20}x_{98} \pmod{256} \equiv 122,\\
R_{99} &\equiv 10^{20}x_{99} \pmod{256} \equiv 133,\\
&\ldots.
\end{aligned} \tag{20}$$

In the calculation, some variables in the program are of type *ZZ* of Shoup,[2] which, though being very large positive numbers, happen to be quite convenient as far as programming task is concerned. For related programming code, please refer to Shoup [27] for details.

We are now ready to discuss our test results. We begin with the relatively simpler test of FIPS PUB 140-2. In Fig. 2 we have plotted the statistical tests of FIPS PUB 140-2 on the generated sequences by CSCLOGISTIC$(\mu, 31426, 0.25, 10^7, 256)$. Here, the ordinate is the number of successful passes a sequence goes through. Thus, a qualified sequence must have a passing value of 16 for most of the time, with possible occasional failures. (We should note in passing that one can not expect a true random sequence to pass it all the time even in principle.) Visually one can already see from Figs. 1 and 2 that whenever the Lyapunov exponent becomes positive for the parameter μ, the

[2] ZZ, built in the NTL library of source code, is a special class in C++ proposed by V. Shoup. NTL is a high-performance, portable C++ library providing data structures and algorithms for manipulating signed, arbitrary length integers, and for vectors, matrices, and polynomials over the integers and over finite fields. Please see [27] for details.

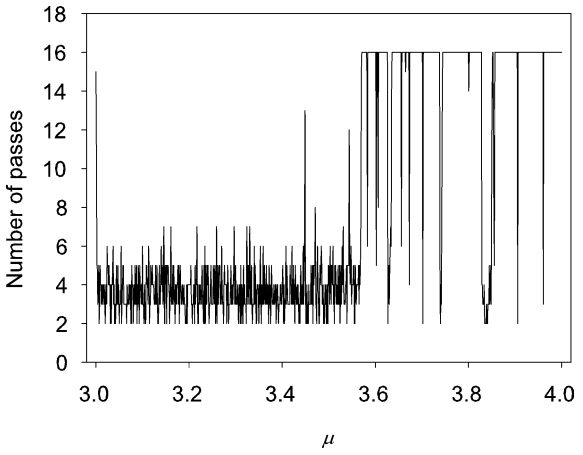

Fig. 2. The statistical test of FIPS PUB 140-2 for CSCLOGISTIC(μ, 31426, 0.25, 10^7, 256): it is passed when the total number of passes is 16.

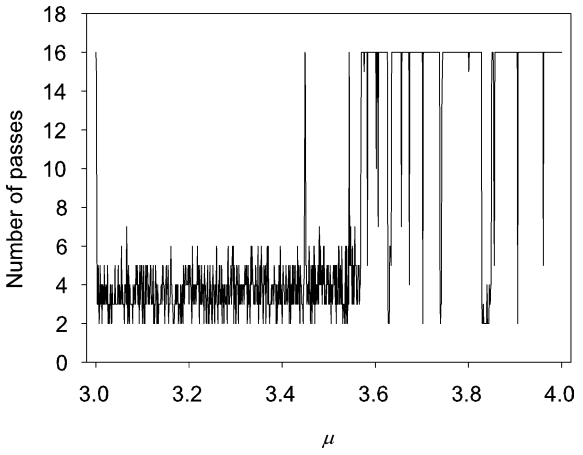

Fig. 3. The statistical test of FIPS PUB 140-2 for CSCLOGIT(μ, 31426, 0.25, 10^7, 256): it is passed when the total number of passes is 16.

sequence also tends to pass the FIPS PUB 140-2 test. To compare the two figures in a more quantitative manner, we have computed their correlation to be $C = 0.719727$ using the Pearson correlation, $C(x, y, N)$, in which x is taken from the data of the Lyapunov exponents and y is taken from the result of the statistical test using FIPS PUB 140-2, and $N = 1001$. The Pearson correlation is defined by

$$C(x, y, N) = \frac{N\sum x_i y_i - \sum x_i \sum y}{\sqrt{(N\sum x_i^2)(N\sum y_i^2)}} \tag{21}$$

where N is the size of the input data, and x, y are the sequences to be compared.

To further verify this observation, we have repeated the same analysis for other systems as well. But before we proceed, it should be recalled that the Lyapunov exponent of a one-variable system $X_{n+1} = f(X_n)$ remains the same if one invokes the change of variables $Y = g(X)$ to obtain a "new" dynamical system $Y_{n+1} = g \circ f \circ g^{-1}(Y_n)$, provided that g is monotonic and smooth. Thus, for instance, the logit transformation of Eq. (2) and the Y_n transformation of Eq. (7) will not change the Lyapunov exponent of the derived systems. (We have also explicitly verified this fact in our numerical calculations as an independent check of our codes.) Thus, in the following we will show only the results of the statistical tests on the generated random numbers without repeatedly displaying the Lyapunov exponent.

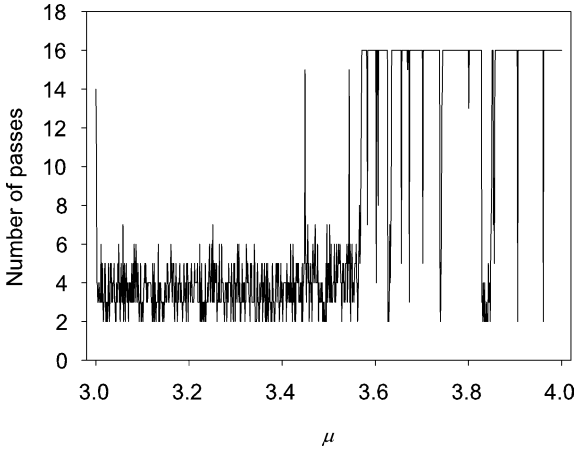

Fig. 4. The statistical test of FIPS PUB 140-2 for CSCYN(μ, 31426, 0.25, 10^7, 256): it is passed when the total number of passes is 16.

The statistical tests for CSCLOGIT(μ, 31426, 0.25, 10^7, 256) is shown in Fig. 3. Similarly, the results for the map modified by the CSCYN(μ, 31426, 0.25, 10^7, 256) are shown in Fig. 4, with the same parameters and initial value. As can be seen, the places where a sequence successfully passes all the tests correspond nicely to the parameters for which the Lyapunov exponent is positive. The correlation C is 0.720672 between Figs. 3 and 4.

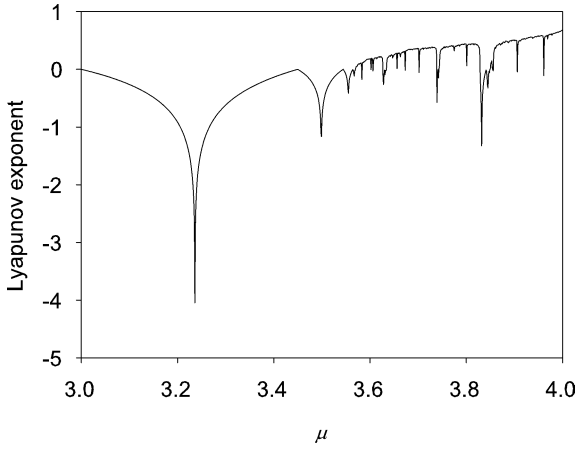

Fig. 5. The Lyapunov exponents of the z-logistic map, with $z = 2$ and initial value $X_0 = 0.25$.

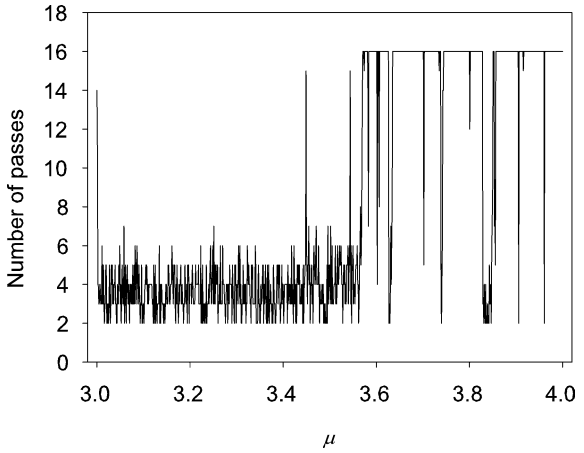

Fig. 6. The statistical test of FIPS PUB 140-2 for CSCZLOGISTIC2(μ, 31426, 0.25, 10^7, 256): it is passed when the total number of passes is 16.

Turning to the restricted z-logistic equation (6), CSCZLOGISTIC2(μ, 31426, 0.25, 10^7, 256), we show in Fig. 5 the Lyapunov exponents and in Fig. 6 the associated statistical tests for $z = 2$. The correlation C is 0.723478 between Figs. 5 and 6. As we increase z from 2 to 3 (Figs. 7 and 8) and onward to 5 (Figs. 9 and 10) we see a corresponding widening of the regions of successful passes. The correlation C is calculated to be 0.723478, 0.869878, and 0.676233 between Figs. 5 and 6, Figs. 7 and 8, and Figs. 9 and 10, respectively. But once again, the correlation between regions of successful passes and regions of positive

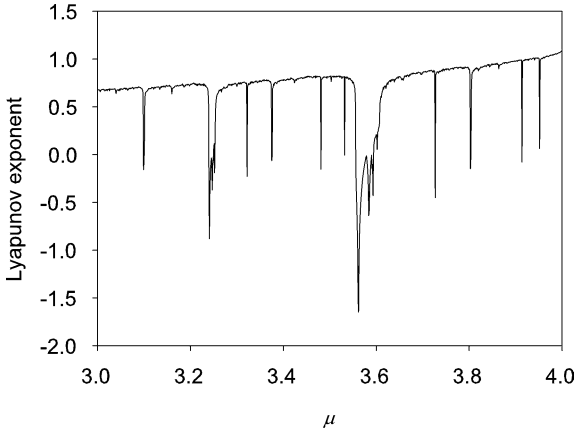

Fig. 7. The Lyapunov exponents of the z-logistic map, with $z = 3$ and initial value $X_0 = 0.25$.

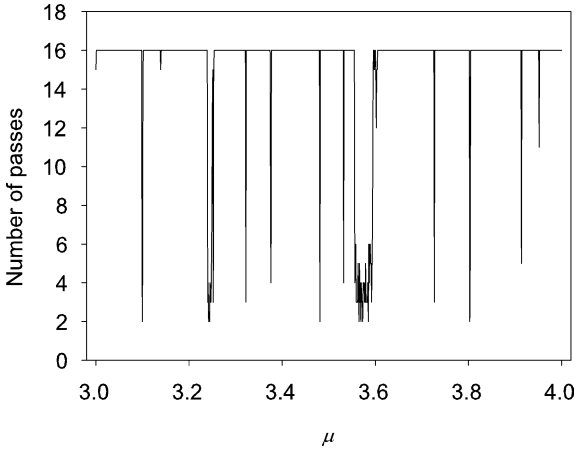

Fig. 8. The statistical test of FIPS PUB 140-2 for CSCZLOGISTIC3(μ, 31426, 0.25, 10^7, 256): it is passed when the total number of passes is 16.

Lyapunov exponents still correspond almost perfectly, as is obvious from the corresponding spikes in the figures. In the three examples, we find that C is the highest while $z = 3$ in CSCLOGIT. For the tent map, the defining equation implies

$$\begin{aligned} \mathrm{d}X_{n+1} &= -2\mu\,\mathrm{d}X_n, \quad 0.5 < X_n \leqslant 1, \\ \mathrm{d}X_{n+1} &= 2\mu\,\mathrm{d}X_n, \quad 0 < X_n \leqslant 0.5 \end{aligned} \tag{22}$$

which admits the exact calculation of the Lyapunov exponent, which is

$$\lambda = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N} \ln \left| \frac{\mathrm{d}f(X_n)}{\mathrm{d}X_n} \right| = \ln(2\mu). \tag{23}$$

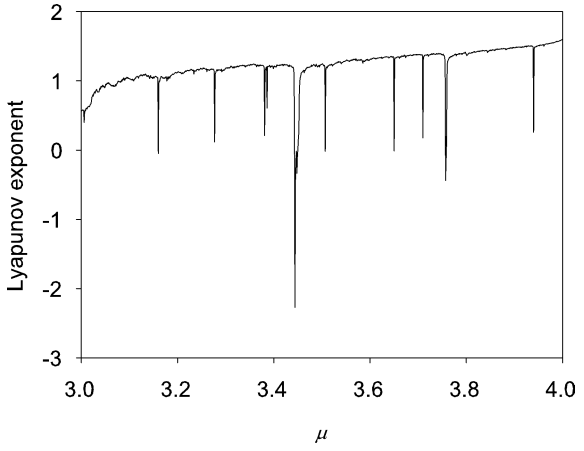

Fig. 9. The Lyapunov exponents of the z-logistic map, with $z = 5$ and initial value $X_0 = 0.25$.

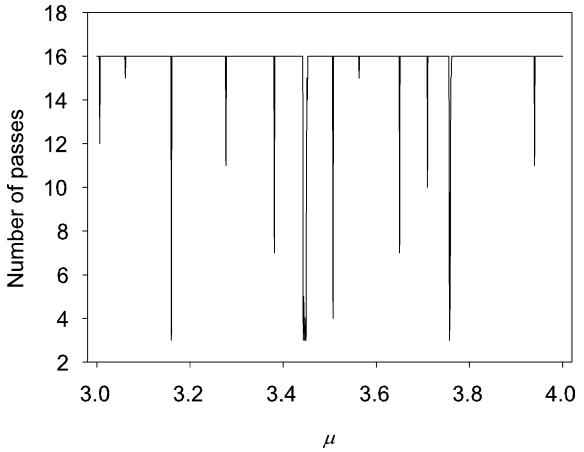

Fig. 10. The statistical test of FIPS PUB 140-2 for CSCZLOGISTIC5(μ, 31426, 0.25, 10^7, 256): it is passed when the total number of passes is 16.

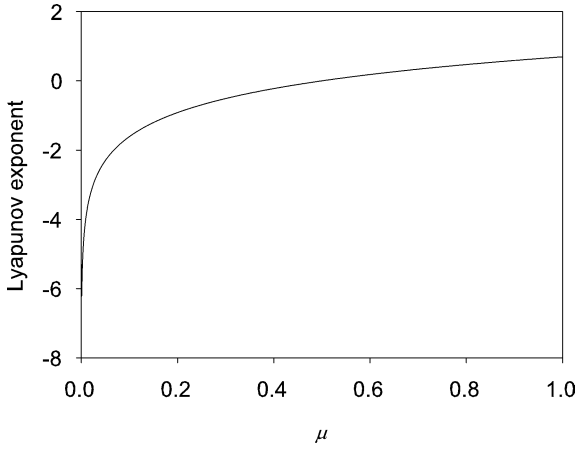

Fig. 11. The Lyapunov exponents of the Tent map, with initial value $X_0 = 0.25$.

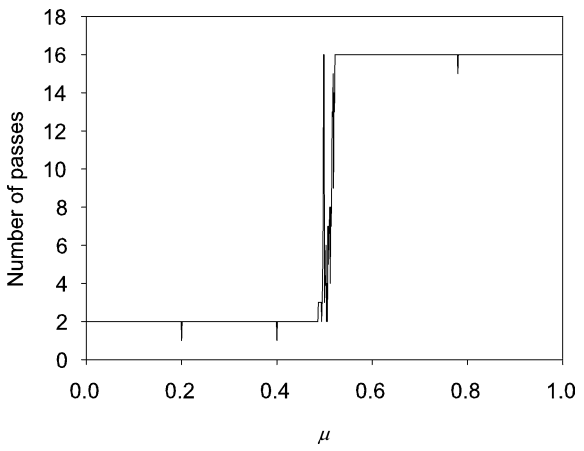

Fig. 12. The statistical test of FIPS PUB 140-2 for CSCTENT7(μ, 31426, 0.25, 10^7, 256): it is passed when the total number of passes is 16.

This is plotted in Fig. 11, together with the associated passing tests plotted in Fig. 12, with a correlation $C = 0.685491$. Figs. 13 and 14 present similar plots for the sine map, with a corresponding $C = 0.767776$. For both cases, we still observe the correspondence mentioned above.

To better understand how sequences of random numbers might fail an individual test, we have divided the entire parameter regime into 1000 equal intervals and computed the total number of failures for each individual test when a specific mapping function is given. This is shown in Table 5. Please note that the parameter regime is μ ($3 \leqslant \mu \leqslant 4$) for all mappings but the tent map, which has $0 \leqslant \mu \leqslant 1$. This table suggests that the Poker Test appears to be the most stringent test among all the sixteen criteria of the FIPS PUB 140-2 test. Also evident from this table is that the z-logistic map (with $z = 3$ or 5) is indeed a superior candidate for random number generator among all those having been tested. In Table 5, as a comparison, we list all the test results using the FIPS PUB 140-2 and compare the data with the number of occurrences of negative Lyapunov exponent for the same parameter regimes. The difference between the two data is rather

Table 5
Total number of failures in the statistical tests for various mapping functions described in Table 1

Subitem	Generator Index							
	8	10	11	12	13	14	15	16
monobit test	499	492	482	478	52	14	423	538
poker test	623	616	621	620	72	22	533	565
run test(0)	443	444	436	434	49	16	384	534
run test(1)	431	450	454	453	49	14	384	533
run test(00)	397	391	396	396	43	9	349	532
run test(11)	407	382	391	389	45	15	344	534
run test(000)	564	558	560	558	51	11	452	521
run test(111)	556	551	557	556	52	13	466	530
run test(0000)	601	588	597	596	59	15	496	521
run test(1111)	598	598	596	595	59	14	490	520
run test(00000)	604	596	599	597	58	12	508	524
run test(11111)	600	598	605	604	56	12	511	527
run test(000000)	603	602	603	601	61	15	512	532
run test(111111)	604	601	600	598	61	14	512	534
long run test(0)	0	0	0	0	2	1	2	0
long run test(1)	0	0	0	0	0	0	0	0
number of failures	623	616	622	621	74	25	536	565
Number of negative Lyapunov exponent	614	614	614	612	62	12	517	500

The random number generators studied in order include CSCLOGISTIC7($\mu = 3.97$), CSCLOGIT7($\mu = 3.97$), CSCYN7($\mu = 3.97$), CSCZLOGISTIC2($\mu = 3.97$), CSCZLOGISTIC3($\mu = 3.97$), CSCZLOGISTIC5($\mu = 3.97$), CSCSINE7($\mu = 3.97$), CSCTENT7($\mu = 0.9$). The parameter regime is equally divided into 1000 intervals, resulting in a set of 1001 test data. The bottom of the table presents the number of occurrences of negative Lyapunov exponent for various generators. The specifications of the test interval are the same as that for the statistical test in the text.

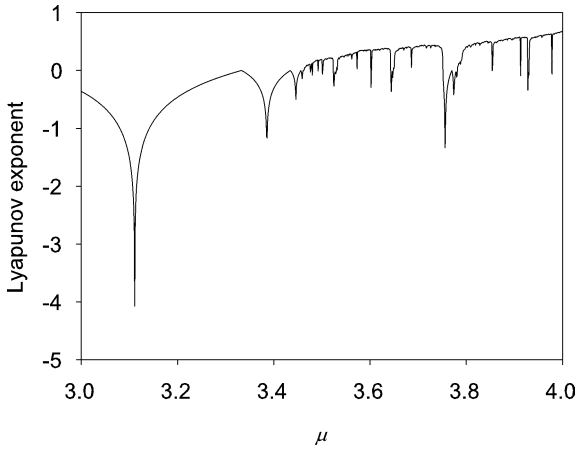

Fig. 13. The Lyapunov exponents of the Sine map, with initial value $X_0 = 0.25$.

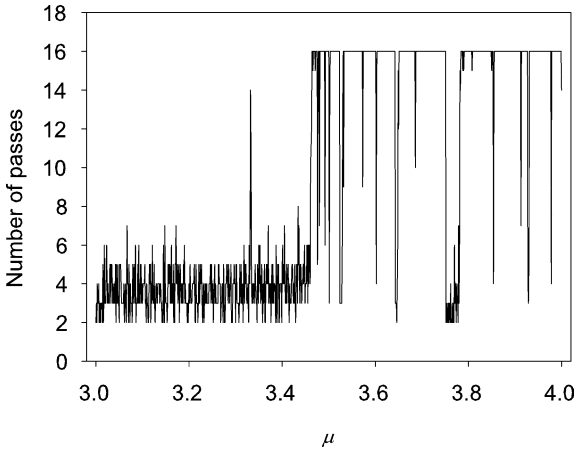

Fig. 14. The statistical test of FIPS PUB 140-2 for CSCSINE7($\mu, 31426, 0.25, 10^7, 256$): it is passed when the total number of passes is 16.

small, once reiterating the fact that positive Lyapunov exponent is nicely correlated with the criteria of a good random number generator.

But to get a better understanding on how CSCLOGISTIC performs under practical conditions, we have to check it against the various standard and more strin-

Table 6
The parameters of SP 800-22 tests

The parameters of SP 800-22 tests	Value
The Block Frequency Test: Block length	10
The Non-overlapping Template Test: Block length	10
The Overlapping Template Test: Block length	10
Maurer's "Universal" Test: Block length	6
"Universal" Test: Number of Initialization Steps	640
The Approximate Entropy Test: Block length	5
The Serial Test: Block length	5
The Linear Complexity Test: Sequence length	5000

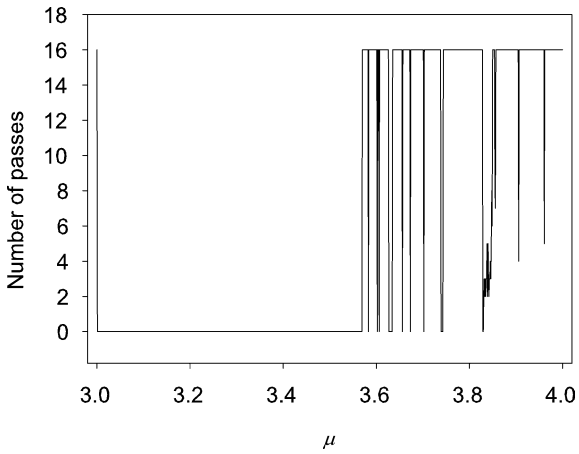

Fig. 15. The statistical test of FIPS PUB 140-2 for CSCLOGISTIC20(μ, 10, 0.13, 10^{20}, 256): it is passed when the total number of passes is 16.

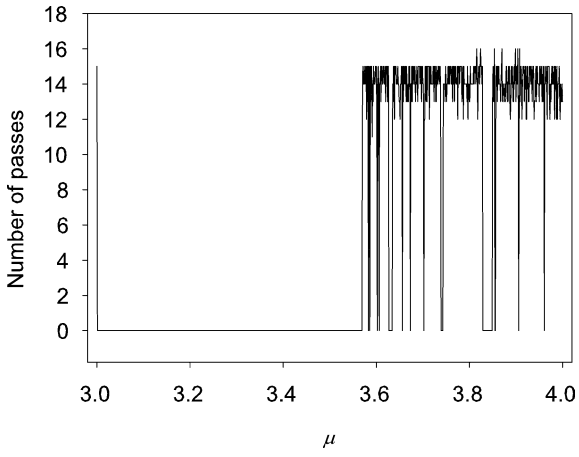

Fig. 16. The statistical test of SP 800-22 for CSCLOGISTIC20(μ, 10, 0.13, 10^{20}, 256): it is passed when the total number of passes is 16.

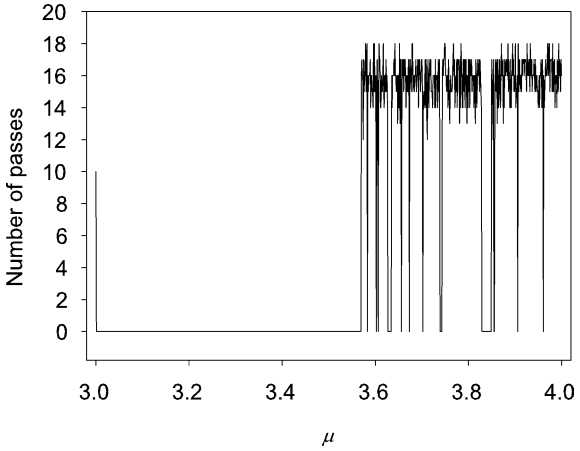

Fig. 17. Diehard battery of tests for CSCLOGISTIC20(μ, 10, 0.13, 10^{20}, 256): it is passed when the total number of passes is 18.

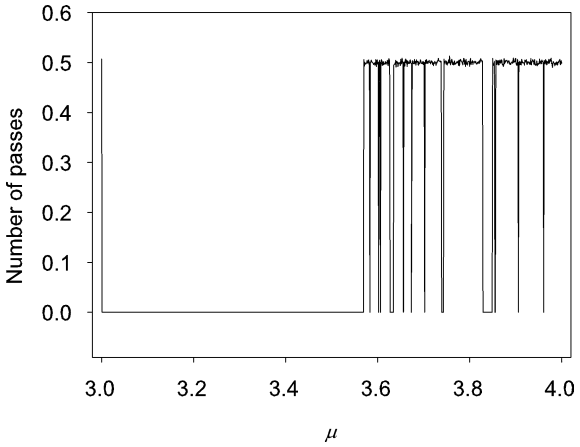

Fig. 18. The height correlation tests for CSCLOGISTIC20(μ, 10, 0.13, 10^{20}, 256): it is passed when the H_t is about 0.5 ± 0.0125.

gent tests. As discussed before, these include SP 800-22 tests, Diehard battery of tests, the height correlation test and S_N tests. But before we go on, we should mention that, due to the nature of the original dynamical system, there are "windows" in the parameter regime inside which the iteration will settle down to a fixed point. Thus, we will exclude these sets since they are not interesting. As a reference, we list below what have been excluded in this report: $3.001 \sim 3.569, 3.583, 3.602, 3.606, 3.627 \sim 3.634, 3.656, 3.673, 3.702, 3.739 \sim 3.743, 3.829 \sim 3.849, 3.855, 3.856, 3.906$ and 3.961. However, we do keep $\mu = 3.000$ in the test, simply because the phe-

Table 7
The various statistical tests for CSCLOGISTIC20 system described in Table 1

μ	FIPS PUB 140-2	SP 800-22 tests	Diehard battery of tests	Height correlation test (ϕ)
3.579	16	**12**	18	0.49966
3.582	16	**15**	18	0.50122
3.596	16	**14**	18	0.50607
3.598	16	**15**	18	0.50189
3.608	16	**14**	18	0.50578
3.617	16	**14**	18	0.50178
3.642	16	**15**	18	0.50170
3.652	16	**14**	18	0.49746
3.679	16	**14**	18	0.49717
3.680	16	**15**	18	0.50620
3.749	16	**15**	18	0.49670
3.750	16	**14**	18	0.50293
3.768	16	**13**	18	0.49608
3.784	16	**13**	18	0.49796
3.816	16	16	**17**	0.49729
3.824	16	16	**15**	0.50790
3.854	16	16	**17**	0.50249
3.870	16	16	**16**	0.50360
3.871	16	**13**	18	0.49633
3.885	16	**14**	18	0.50389
3.895	16	**13**	**16**	0.50907
3.899	16	16	**15**	0.50807
3.901	16	**15**	18	0.49770
3.904	16	16	**15**	0.50623
3.907	16	**12**	18	0.50099
3.908	16	16	**15**	0.49561
3.926	16	**14**	18	0.50109
3.933	16	**15**	18	0.50814
3.970	16	**15**	18	0.50115
3.971	16	**15**	18	0.49848
3.974	16	**14**	18	0.50164
3.981	16	**14**	18	0.50128
3.994	16	**13**	18	0.50303
Exact	16	16	18	1/2

The bold font means that it is fail to test. The CSCLOGISTIC20 studied based on the logistic map, the parameters as CSCLOGISTIC($\mu, 10, 0.13, 10^{20}, 256$) for each μ. The criteria of Height correlation test is set up 0.5 ± 0.0125.

nomenon of critical slowing down has prevented the system from settling down after an iteration of $n = 1.5 \times 10^7$ steps.

To get the correlation, we have set up all the statistical tests so that a 0 is returned when the parameter μ fails a test. Table 6 describes the setting of the parameters in the NIST programming code and we use such condition to test all the random number

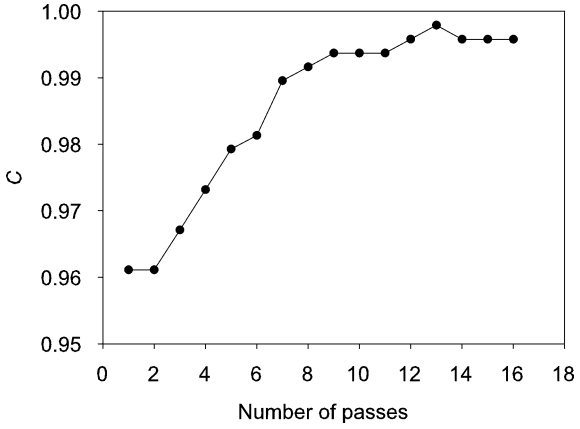

Fig. 19. The correlation between Lyapunov exponents and tests of FIPS PUB 140-2 for CSCLOGISTIC20($\mu, 10, 0.13, 10^{20}, 256$).

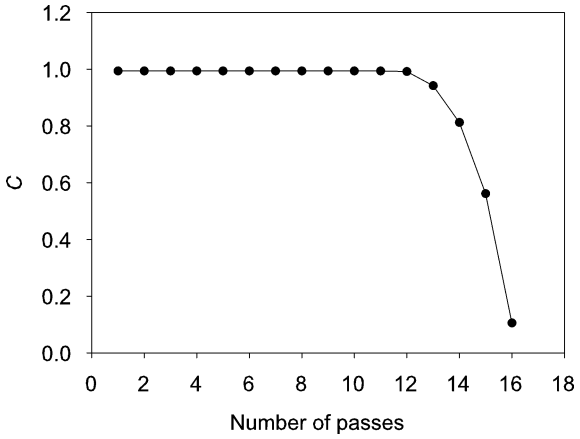

Fig. 20. The correlation between Lyapunov exponents and tests of SP 800-22 for CSCLOGISTIC20($\mu, 10, 0.13, 10^{20}, 256$).

generators defined in Table 1. The size of the random bits is 1.5×10^7, and the same process is run 15 times, i.e. $m = 15$, as has been said before. For the Diehard test suite, we set up the size of data as $n = 1.2 \times 10^7$ bytes for each μ. As to the height correlation tests, the function H_t was investigated up to $\Omega = 2000$ with size $n = 10^7$ independent runs. For the S_N tests, we set up $\Omega = 4000$ with $n = 10^8$ independent runs. For CSCLOGISTIC20 system defined above, we plot the FIPS PUB 140-2 tests in Fig. 15, the SP 800-22 tests in Fig. 16, Diehard battery of tests in Fig. 17 and height correlation tests in Fig. 18. In Tables 7 and 8, we list our thorough study of the strength of CSCLOGISTIC and com-

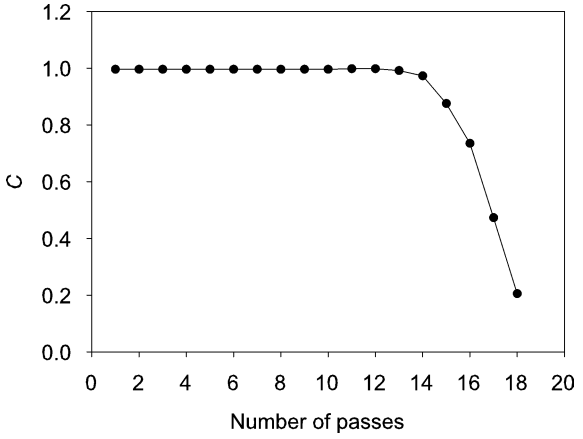

Fig. 21. The correlation between Lyapunov exponents and Diehard battery of tests for CSCLOGISTIC20(μ, 10, 0.13, 10^{20}, 256).

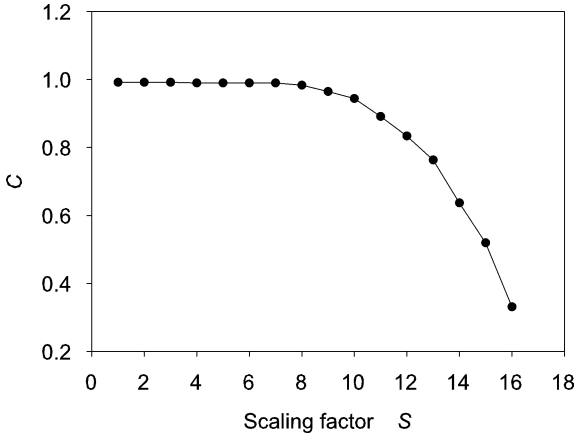

Fig. 22. The correlation between Lyapunov exponents and tests of height correlation for CSCLOGISTIC20(μ, 10, 0.13, 10^{20}, 256), where the scaling factor S checks the number of passes in height correlation test, and the criteria is $0.5 \pm 0.001 \times (17 - S)$.

pare it with other famous random number generators using various statistical tests. In theses tables, a failed test result is marked in bold font for convenience. In Table 7, for the special parameter μ, we find that the listed CSCLOGISTIC20(μ) behave well in these tests. Though it is hard to satisfy the whole wide range of tests for a given μ, CSCLOGISTIC20 can still be seen to possess the potential of being a good random number generator. For instance, referring to Table 8, we see that LCG can hardly pass most standard tests when CSCLOGISTIC20(μ) stands quite strong in this respect. In fact, the results suggest that it is better than LCG, ICG, and EICG using this test. As a side remark, we can also notice that the strength of FIPS PUB 140-2 is weak compared to other tests, whereas SP 800-22 tests is much more powerful as regards to checking for the robustness of a PRNG. In Figs. 19–21, we plot the correlation C versus the number of passing items for various tests. In Fig. 22, we use the scaling factor S to check the number of passes in height correlation test, the criteria being $0.5 \pm 0.001 \times (17 - S)$. Note that in plotting the four figures we have normalized things so that $a + 1$ is returned when the Lyapunov exponent is positive, while $a - 1$ is returned when it is negative. Similarly, we have assigned $a + 1$ if the data successfully pass the tests, and $a - 1$ is obtained if otherwise. When compared with Fig. 1, the correlation C for Figs. 19–22 read 0.726202, 0.717896, 0.720198, and 0.720785, respectively. This is shown in Table 9, which summarizes our assertion that, indeed, a positive Lyapunov exponent is nicely correlated with the industry criteria for a good random number generator. Finally, we show in Table 10 the number of steps it takes for our system to settle down to a stable period-doubled state for various scaling factor A. This table explains why CSCLOGISTIC($\mu = 3.0$) can occasionally pass the statistical tests: As the scaling factor A gets larger, the time to settle down also gets longer. This is the reason why we always kept the parameter $\mu = 3.0$ in our discussions. Despite all these successes, we must quickly add, however, that our conclusion is drawn on a relatively small set of data (of order 10^7 to 10^8) as compared to the huge number of data one would like to use in the simulation of actual physical systems (which could be as large as 10^{14}). This means our success is modest at the best, and further investigation in the future is still needed to see how things go in this line of approach.

4. Conclusion

Although it is tempting to use simple chaotic maps as an efficient method of generating useful random numbers, the correlation between adjacent numbers generated by the same map must first be processed before they can be used in an application. In this paper we have suggested a simple modulo operation,

Table 8
The various statistical tests for random number generators described in the text

Index	Generator	FIPS PUB 140-2	SP 800-22 tests	Diehard battery of tests	Height correlation test (ϕ)	S_N test
1	RANAU	**11**	**0**	**0**	**1.00730**	**0.01670**
2	ANSIC	**15**	**0**	**0**	**1.00842**	**0.03607**
3	MINSTD	16	**15**	**14**	**0.98954**	**0.04689**
4	FISH	16	**14**	**16**	**1.03299**	0.49587
5	ICG	16	**15**	**15**	**1.02911**	0.49826
6	EICG1	16	**15**	**16**	**0.96355**	**0.53779**
7	EICG7	16	**14**	**16**	**0.92437**	0.50474
8	CSCLOGISTIC7	16	**15**	**11**	**0.49604**	**0.48088**
9	CSCLOGISTIC20	16	**15**	18	0.50115	0.50171
10	CSCLOGIT7	16	**15**	**12**	0.49765	**0.51723**
11	CSCYN7	16	**13**	**9**	0.49701	0.50972
12	CSCZLOGISTIC2	16	**12**	**11**	0.50054	**0.52163**
13	CSCZLOGISTIC3	16	**11**	**13**	0.49746	**0.51702**
14	CSCZLOGISTIC5	16	**13**	**15**	0.50268	0.49785
15	CSCSINE7	16	**14**	**10**	0.49884	0.50533
16	CSCTENT7	16	**11**	**8**	0.50604	**0.52056**
17	CSCTENT20	16	**11**	**12**	0.49452	**0.52534**
	Exact	16	16	18	1/2	1/2

Numbers in bold signify a failure in the tests. For the tests of generators 1 through 7, Eq. (10) is also implemented when applying the tests of FIPS PUB 140-2, SP 800-22, and Diehard tests. In height correlation test and S_N test, we use instead only the original random data. The criteria of height correlation test and S_N test are both set to 0.5 ± 0.0125.

Table 9
The correlation calculation between Lyapunov exponents and various statistical tests of CSC systems, where X_1, X_2 represent the data of input sequences and N as the size of input data, $N = 1001$

Correlation function $C(X_1, X_2, N)$	Correlation C
C(Fig(1), Fig(2), N)	0.719727
C(Fig(1), Fig(3), N)	0.720672
C(Fig(1), Fig(4), N)	0.723787
C(Fig(5), Fig(6), N)	0.723478
C(Fig(7), Fig(8), N)	0.869878
C(Fig(9), Fig(10), N)	0.676233
C(Fig(11), Fig(12), N)	0.685491
C(Fig(13), Fig(14), N)	0.767776
C(Fig(1), Fig(15), N)	0.726202
C(Fig(1), Fig(16), N)	0.717896
C(Fig(1), Fig(17), N)	0.720198
C(Fig(1), Fig(18), N)	0.720785

the chaotic stream cipher (CSC), for this purpose. We have shown that, using standard criteria for random number generators, the scheme is quite robust with respect to these tests. Furthermore, we show that there is a close correlation between the Lyapunov exponent of the underlying chaotic map and the "randomness" defined by the industry standard tests for the generated sequences. Although the correlation is not perfect, our investigation suggests convincingly that the standard tests might be closely related to only a very few number of characteristics inherent in a chaotic system. However, we must emphasize that our guess still needs a lot of polishing because, up to this point, the known useful measures for chaos are not too many in number, and exactly which measure is more relevant remains to be investigated.

Acknowledgements

We thank the anonymous referee for providing very valuable suggestions and useful references to make this work more complete. This work was supported by the National Science Council of the Republic of China under grant numbers NSC90-2112-M-002-057 and NSC91-2112-M-002-021.

Table 10
The number of time steps for the CSCLOGISTIC($\mu, 0, 0.13, A, 256$) system to evolve into one of the periodic states for various scale parameter A. μ_j is the jth period doubling bifurcation parameter

μ	Period	The scale parameter A					
		10^5	10^6	10^7	10^8	10^9	10^{10}
$\mu_1 = 3.0$	2^1	20	2484	16910	84156	394138	1834660
$\mu_2 = 3.449489$	2^2	460	1876	9276	39220	186444	837172
$\mu_3 = 3.544090$	2^3	128	1104	4880	22040	105944	529136
$\mu_4 = 3.564407$	2^4	160	704	3104	13200	53136	299536
$\mu_5 = 3.568759$	2^5	96	416	1984	9120	32512	95968
$\mu_6 = 3.569692$	2^6	64	128	704	4544	15488	46784
$\mu_7 = 3.569891$	2^7	128	256	384	2944	15488	44544
$\mu_8 = 3.569934$	2^8	256	256	256	2304	7424	29184

References

[1] P. L'Ecuyer, Ann. Oper. Res. 53 (1994) 77.
[2] H. Leeb, S. Wegenkittl, ACM Trans. Model. Comput. Simul. 7 (1997) 272.
[3] J. Eichenauer-Herrmann, Internat. Statist. Rev. 63 (1995) 247.
[4] P. Hellekalek, Math. Comput. Simul. 46 (1998) 485.
[5] L.M. Pecora, T.L. Carroll, Phys. Rev. Lett. 64 (1990) 821.
[6] G. Perez, H.A. Cerdeira, Phys. Rev. Lett. 74 (1995) 1970.
[7] Y.-Y. Chen, Europhys. Lett. 34 (1996) 245.
[8] R. He, P.G. Vaidya, Phys. Rev. E 57 (1998) 1532.
[9] F. James, Comput. Phys. Commun. 60 (1990) 329.
[10] F. James, Chaos Solitons Fractals 6 (1995) 221.
[11] R. Brown, L.O. Chua, Internat. J. Bifur. Chaos Appl. Sci. Engrg. 6 (1996) 6564.
[12] J.J. Collins, M. Fanciulli, R.G. Hohlfeld, D.C. Finch, G.V.H. Sandri, E.S. Shtatland, Comput. Phys. 6 (1992) 630.
[13] J.A. González, R. Pino, Comput. Phys. Commun. 120 (1999) 109.
[14] M. Andrecut, Internat. J. Modern Phys. B 12 (1998) 921.
[15] S.C. Phatak, S.S. Rao, Phys. Rev. E 51 (1995) 3670.
[16] P.A.W. Lewis, A.S. Goodman, J.M. Miller, IBM Syst. J. 2 (1969) 136.
[17] S.K. Park, K.W. Miller, Commun. ACM 31 (1988) 1192.
[18] G.S. Fishman, L.R. Moore, SIAM J. Sci. Statist. Comput. 7 (1986) 24.
[19] National Institute of Standard and Technology and Communication Security Establishment, Derived Test Requirement (DTR) for FIPS PUB 140-2, Security Requirements for Cryptographic Modules, available at URL: http://www.nist.gov/cmvp.
[20] A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, proposed by National Institute of Standard and Technology, October 2000, available at URL: http://csrc.nist.gov/publications/nistpubs/.
[21] G. Marsaglia, in: L. Billard (Ed.), Computer Science and Statistics: The Interface, Elsevier, Amsterdam, 1985, p. 3.
[22] I. Vattulainen, Phys. Rev. E 59 (1999) 7200.
[23] G. Marsaglia, Diehard statistical tests, http://stat.fsu.edu/~geo/diehard.html.
[24] S.L. Anderson, SIAM Rev. 32 (1990) 221.
[25] J. Krug, Adv. Phys. 46 (1997) 139.
[26] H. Larralde, P. Trunfio, S. Havlin, H.E. Stanley, G.H. Weiss, Phys. Rev. A 45 (1992) 7128.
[27] URL: http://www.shoup.net/index.html.