Periodic orbits of the ensemble of cat maps and pseudorandom number generation

L. Barash[1,a] and L.N. Shchur[1,2,b]

[1] *Landau Institute for Theoretical Physics, 142432 Chernogolovka, Russia*
[2] *Materials Science Division, Argonne National Laboratory, Argonne, Illinois 60439, USA*
e-mail: [a]`barash@itp.ac.ru`, [b]`lev@landau.ac.ru`

We propose methods to construct high-quality pseudorandom number generators (RNG) based on an ensemble of hyperbolic automorphisms of the unit two-dimensional torus (Sinai-Arnold map, or cat map) while keeping a part of the information hidden. The single cat map provides the random properties expected from a good RNG and is hence an appropriate building block for an RNG, although unnecessary correlations are always present in practice. We show that introducing hidden variables and introducing rotation in the RNG output, accompanied with the proper initialization, suppress these correlations dramatically and complicate deciphering. We analyze the mechanisms of the single-cat-map correlations analytically and show how to diminish them. We generalize Percival-Vivaldi theory in the case of the ensemble of maps, find the period of the proposed RNG analytically, and also analyze its properties. We present efficient practical realizations for the RNGs and check our predictions numerically. We also test our RNGs using the known stringent batteries of statistical tests and found that the statistical properties of our best generators are not worse than that of other best modern generators.

PACS numbers: 02.50.Ng, 02.70.Uu, 05.45.-a

I. INTRODUCTION

Molecular dynamics and Monte Carlo simulations are important computational techniques in many areas of science: in quantum physics [1], statistical physics [2], nuclear physics [3], quantum chemistry [4], material science [5], among many others. The simulations rely heavily on the use of random numbers, which are generated by deterministic recursive rules. Such rules produce pseudo random numbers, and it is a great challenge to design random number generators (RNG) that behave as realizations of independent uniformly distributed random variables and approximate "true randomness" [6].

There are several requirements for a good RNG and its implementation in a subroutine library. Among them are statistical robustness (uniform distribution of values at the output with no correlations), unpredictability, long period length, efficiency, theoretical support (precise prediction of the important properties), portability and others [6, 7, 8].

A number of RNGs introduced last five decades fulfill most of the requirements and are successfully employed in simulations. Nevertheless, each of them have some weak properties which may (or may not) influence the results.

The most widely used RNGs can be divided into two classes. The first class is represented by the Linear Congruential Generator (LCG), and the second, by Shift Register (SR) generator.

Linear Congruential Generators (LCG) are the best-known and (still) most widely available RNGs in use today. An example of the realization of an LCG generator is the UNIX `rand` generator $y_n = (1103515245 \, y_{n-1} + 12345) \pmod{2^{31}}$. The practical recommendation is that LCGs should be avoided for applications dealing with the geometric behavior of random vectors in high dimensions because of the bad geometric structure of the vectors that they produce [6, 9].

Generalized Feedback Shift Register (GFSR) sequences are widely used in many areas of computational and simulational physics. These RNGs are quite fast and possess huge periods given a proper choice of the underlying primitive trinomials [10]. This makes them particularly well-suited for applications that require many pseudorandom numbers. But several flaws have been observed in the statistical properties of these generators, which can result in systematic errors in Monte Carlo simulations. Typical examples include the Wolff single cluster algorithm for the 2D Ising model simulation [11], random and self-avoiding walks [12], 3D Blume-Capel model using local Metropolis updating [13].

Modern modifications and generalizations to the LCG and GFSR methods have much better periodic and statistical properties. Some examples are the Mersenne twister [14] (this generator employs the modified and generalized GFSR scheme), combined LCGs generators [15] and combined Tausworthe generators [16, 17].

Most RNGs used today can be easily deciphered. Perhaps the generator with the best properties of unpredictability known today is the BBS generator [18, 19], which is proved to be polynomial-time perfect under certain reasonable assumptions [7, 18] if the size s of the generator is sufficiently large. This generator is rather slow for practical use because its speed decreases rapidly as s increases.

We propose to use an ensemble of simple nonlinear dynamical systems to construct an RNG. Of course, not all dynamical systems are useful. For instance, the Baker's transformation is a simple example of a chaotic system: it is area preserving and deterministic, its state is maintained in a bounded domain. The base of the Baker's

transformation is the Bernoulli Shift $x_{n+1} = 2x_n \pmod 1$, it yields a sequence of random numbers provided we have a random irrational seed. But in real computation, the seed number has finite complexity, and the number of available bits reduces at each step. Obviously, there is no practical use of this scheme for an RNG.

The logistic map [20, 21] also does not help to construct an RNG. First, manipulation with real values of fixed accuracy leads to significant errors during long orbits. Second, the sequence of numbers generated by a logistic map does not have a uniform distribution [21]. Also, the logistic map represents a chaotic dynamical system only for isolated values of a parameter. Even small deviations from these isolated values lead to creating subregions in the phase space, i.e., the orbit of the point does not span the whole phase space.

The next class of dynamical systems is Anosov diffeomorphisms of the two-dimensional torus, which have attracted much attention in the context of ergodic theory. Anosov systems have the following stochastic properties: ergodicity, mixing, sensitive dependence on initial conditions (which follows from positivity of the Lyapunov exponent), and local divergence of all trajectories (which follows from positivity of the Kolmogorov-Sinai entropy). These properties resemble certain properties of randomness. Every Anosov diffeomorphism of the torus is topologically conjugate to a hyperbolic automorphism, which can be viewed as a completely chaotic Hamiltonian dynamical system. Hyperbolic automorphisms are represented by 2×2 matrixes with integer entries, a unit determinant, and real eigenvalues, and are known as *cat maps* (there are two reasons for this terminology: first, CAT is an acronym for Continuous Automorphism of the Torus, second, the chaotic behavior of these maps is traditionally described by showing the result of their action on the face of the cat [22]). We note that cat maps are Hamiltonian systems. Indeed, if $k = \mathrm{Tr} M > 2$, then the action of the map (1) on the vector $\binom{p}{q}$ can be described as the motion in the phase space specified by the Hamiltonian [23] $H(p,q) = (k^2 - 4)^{-1/2} \sinh^{-1}((k^2 - 4)^{1/2}/2)(m_{12}p^2 - m_{21}q^2 + (m_{11} - m_{22})pq)$. Here, p and q are taken modulo 1 at each observation (i.e., we preserve only the fractional part of p and q; the integer part is ignored), and observations occur at integer points of time.

In this paper, we present RNGs based on ensemble of cat maps, and analyze the requirements for a good RNG with respect to our scheme. The basic idea is applying the cat map to the discrete set of points (there are two modifications: for $g = 2^m$ and for prime g, where $g \times g$ is the lattice) in such a way that all the points belong to periodic trajectories.

The similar utilization of cat maps for an RNG is called matrix generator for pseudorandom numbers. It was introduced in [24, 25] and discussed for the prime values of g. However, the single matrix generator being the generalization of the linear congruential method, suffers from both the defects of LCG [26] and the defects of GFSR (see Section IV). The periodic and statistical properties of the matrix generators and of the equivalent multiple recursive generators have been studied [27, 28], however the single 2×2-matrix generator still has significant correlations between values at the output.

Also, there is an impressive theoretical basis for relating properties of the periodic orbits of cat maps and properties of algebraic numbers [29], which to the best of our knowledge has never been directly applied to RNG theory. Applying the ensemble of matrix transformations of the two-dimensional torus while using only a single bit from the point of each map, and utilizing rotation in the RNG output are the distinctive features of our generator. Also, as for other generators, the proper initialization of the initial state is important. As will be seen, proposed scheme has several advantages. First, it can essentially reduce correlations and lead to creating an RNG not worse than other modern RNGs. Second, both the properties of periodic orbits and the statistical properties of such generator can be analyzed both theoretically and empirically. Several examples of the RNGs made by this method, as well as the effective realizations, are presented.

The generator is introduced in Sec. II. In Sec. III, we present the results for stringent statistical tests and conclude that the RNGs show statistical behaviour not worse than other modern generators. Correlations for a single cat map are also analyzed thoroughly. In particular, the one-dimensional random walks test is very important because it can reveal single-map correlations from the output of the RNG made from the ensemble of cat maps. Indeed, some correlations are found by the random walks test. We analyze the mechanism of these correlations in Sec. IV; they appear to be associated with the geometric properties of the cat map. We find these correlations analytically (Sec. IV). We provide a method for obtaining quantities such as the periods of cat maps, the number of orbits with a given period, and the area in the phase space swept by the orbits with a given period (Appendix A). We also provide a method for obtaining periods of the generator for arbitrary parameters of the map and lattice (Appendix B). This gives the primary theoretical support of the generator. In particular, we found that the typical period of the generator for the $2^m \times 2^m$ lattice is $T_m = 3 \cdot 2^{m-2}$. The method is based on the work of Percival and Vivaldi, who transformed the study of the periodic orbits of cat maps into the modular arithmetic in domains of quadratic integers and showed how to classify periodic orbits of these systems [29]. The key ideas in [29] needed for our consideration are briefly reviewed in Appendix A. Appendix C gives the method to analyze correlations between orbits of different points and to choose the proper initial conditions in order to minimize the correlations. Appendixes D and E support the other sections, giving detailed proofs of the underlying results. Appendix F presents the efficient realizations for several versions of RNG, the initialization techniques and the analysis for the speed of the RNGs.

II. THE GENERATOR

A. Description of the method

We consider hyperbolic automorphisms of the unit two-dimensional torus (the square $(0,1] \times (0,1]$ with the opposite sides identified). The action of a given cat map R is defined as follows: first, we transform the phase space by the matrix

$$M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \in SL_2(\mathbb{Z}); \tag{1}$$

second, we take the fractional parts in $(0,1)$ of both coordinates. The elements of M are integers, $\det M = 1$, and the eigenvalues of M are $\lambda = (k \pm \sqrt{k^2 - 4})/2$, where $k = \text{Tr}(M)$ is the trace of the matrix M. The eigenvalues should be real because complex values of λ lead to a nonergodic dynamical process, and hyperbolicity condition is $|k| > 2$.

It is easy to prove that the periodic orbits of the hyperbolic toral automorphism R consist precisely of those points that have rational coordinates [22, 23, 29]. Hence, it is natural to consider the dynamics of the map defined on the set of points with rational coordinates that share a given denominator g. The lattice of such points is invariant under the action of the cat maps. In practice, we construct generators with $g = 2^m$, where m is a positive integer, and generators with $g = p = 2^m - 1$, where m is a Mersenne exponent, i.e. $p = 2^m - 1$ is a prime integer.

The notion of an RNG can be formalized as follows: a generator is a structure $\mathcal{G} = (S, s_0, T, U, G)$, where S is a finite set of *states*, $s_0 \in S$ is the *initial state* (or *seed*), the map $T : S \to S$ is the *transition function*, U is a finite set of *output* symbols, and $G : S \to U$ is the *output function* [7]. Thus, the state of the generator is initially s_0, and the generator changes its state at each step, calculating $s_n = T(s_{n-1})$, $u_n = G(s_n)$ at step n. The values u_n at the output of the generator are called the *observations* or the *random numbers* produced by the generator. The output function G may use only a small part of the state information to calculate the random number, the majority of the information being ignored. In this case, there exist *hidden variables*, i.e., some part of the state information is "hidden" and cannot be restored using only the sequence of RNG observations.

We consider the generator with $S = L^s$, where $L = \{0, 1, \ldots, g-1\} \times \{0, 1, \ldots, g-1\}$ is the lattice on the torus and s is a positive integer. In other words, the state consists of coordinates of s points of the $g \times g$ lattice on the torus. For instance, the initial state consists of points $\begin{pmatrix} x_i^{(0)} \\ y_i^{(0)} \end{pmatrix}$, where $x_i^{(0)}, y_i^{(0)} \in \{0, 1, \ldots, g-1\}$ and $i = 0, 1, \ldots, (s-1)$. We note that these are points of the integer lattice, i.e., $x_i^{(0)}$ and $y_i^{(0)}$ are positive integers. The actual initial points on the unit two-dimensional torus $(0,1] \times (0,1]$ are

$$\begin{pmatrix} x_i^{(0)}/g \\ y_i^{(0)}/g \end{pmatrix}, i = 0, 1, \ldots, (s-1). \tag{2}$$

The transition function of the generator is defined by the action of the cat map R, i.e., these s points are affected at every step by the cat map:

$$\begin{pmatrix} x_i^{(n)}/g \\ y_i^{(n)}/g \end{pmatrix} = M \begin{pmatrix} x_i^{(n-1)}/g \\ y_i^{(n-1)}/g \end{pmatrix} \pmod{1}, i = 0, 1, \ldots, (s-1). \tag{3}$$

Here the mod 1 operation means taking the fractional part in $(0,1)$ of the real number. An equivalent description of the transition function is

$$\begin{pmatrix} x_i^{(n)} \\ y_i^{(n)} \end{pmatrix} = M \begin{pmatrix} x_i^{(n-1)} \\ y_i^{(n-1)} \end{pmatrix} \pmod{g}, i = 0, 1, \ldots, (s-1). \tag{4}$$

We let $\alpha_i^{(n)}$ denote 0 or 1 depending on whether $x_i^{(n)} < (g/2)$ or $x_i^{(n)} \geq (g/2)$, i.e. $\alpha_i^{(n)} = \lfloor 2x_i^{(n)}/g \rfloor$. The output function of the generator $G : L^s \to \{0, 1, \ldots, 2^s - 1\}$ is defined as $a^{(n)} = \sum_{i=0}^{s-1} \alpha_i^{(n)} \cdot 2^i$. In other words, $a^{(n)}$ is the s-bit integer, it consists of the bits $\alpha_0^{(n)}, \alpha_1^{(n)}, \ldots, \alpha_{s-1}^{(n)}$. In the case $g = 2^m$, $a^{(n)}$ contains precisely the first bits of the integers $x_0^{(n)}, x_1^{(n)}, \ldots, x_{s-1}^{(n)}$. The sequence of random numbers produced by the generator is $\{a^{(n)}\}$.

We see that the constructed RNG has much hidden information. For example, if $g = 2^m$, then $s(m-1)$ bits of $\begin{pmatrix} x_i^{(n)} \\ y_i^{(n)} \end{pmatrix}$ are the hidden variables; these are the bits that are not involved in constructing the value of output function $a^{(n)}$.

Thus, applying the chaotic behavior of Anosov motion and introducing an ensemble of systems while keeping part of the information hidden are the main ingredients of the proposed method. Cat maps provide ergodicity and hyperbolicity properties, which are obviously necessary for good generators. Introducing hidden variables reduces correlations (as is shown in Sec. III) and strongly complicates deciphering the RNG.

The calculation of the period of the RNG is presented in Appendix B. The typical period length is $T_m = 3 \cdot 2^{m-2}$ for the $2^m \times 2^m$ lattice. The proper initializations for the generators are presented in Appendix F. The proper initialization guarantees that the actual period is not smaller than T_m, and that the points $\binom{x_i^{(0)}}{y_i^{(0)}}$, $i = 0, 1, \ldots, (s-1)$ belong to different orbits of the cat map.

B. Connection With Other Generators

There are several known connections between Anosov dynamical systems and pseudorandom number generation.

First, the concept of the Shift Register Sequence, which is widely used to construct high-quality RNGs, is connected to dynamical systems (see, e.g., discussion in [31]). Let the state of the shift register be $\mathbf{v_{n-1}} = (a_{n-r}, a_{n-r+1}, \ldots, a_{n-1})$. At the next iteration, the state of the shift register is $\mathbf{v_n} = (a_{n-r+1}, a_{n-r+2}, \ldots, a_n)$, where $a_n = c_r a_{n-r} + c_s a_{n-s} \pmod 2$. In other words, $\mathbf{v_{n+1}} = A\mathbf{v_n} \pmod 2$, where A is an $r \times r$ matrix.

Second, Linear Congruential Generators in some cases can be described by the action of the hyperbolic toral automorphism [39].

Last, it can be shown that, for each i, the sequence $\{x_i^{(n)}\}$, defined above, follows a linear recurrence modulo g, as well as the sequence $\{y_i^{(n)}\}$:

$$x^{(n)} = kx^{(n-1)} - qx^{(n-2)} \pmod g \tag{5}$$

$$y^{(n)} = ky^{(n-1)} - qy^{(n-2)} \pmod g, \tag{6}$$

where $k = \text{Tr} M$, $q = \det M = 1$. The characteristic polynomial of the last linear recurrence is $f(x) = q - kx + x^2$, which is exactly the same as that of the matrix M [24, 30]. The period properties of the sequence (5) follow from the finite field theory [6].

C. Generators for prime g: modifications for $\det M = 1$ and for $\det M \neq 1$

The matrix generator of pseudorandom numbers, that is equivalent to the sequence (5), has been studied in [6, 24, 32] in the case when $g = p$ is a prime number. The sequence (5) yields maximum possible period $p^2 - 1$ if and only if the characteristic polynomial $f(x)$ is primitive over $\mathbb{Z}_p$. However, for $q = 1$, the polynomial $f(x) = x^2 - kx + 1$ is not primitive over $\mathbb{Z}_p$ for $p > 2$. Therefore, if $\det M = 1$, the period is always smaller than $p^2 - 1$. The even stronger result follows from [29]: the period can not be larger than $(p + 1)$ when g is a prime number and $\det M = 1$.

Matrix generators with $q \neq 1$ are not immediately connected with Hamiltonian dynamical systems. Indeed, the transformation with $\det M \neq 1$ does not preserve the volume in phase space, and does not immediately represent a cat map. However, whatever q is, we have $\det M^p \equiv 1 \pmod p$. This means that the action of the matrix M^p on a lattice $p \times p$ is exactly the same as the action of a unimodular matrix. Therefore, any orbit of a "non-Hamiltonian" transformation M contains exactly p cat-map orbits.

Also, transformations with $q = 1$ preserve the norm on the orbit modulo g (see Appendix C), in contrast to transformations with $q \neq 1$. It is shown in Appendix C that some of the correlations between the orbits are inherent in the case $q = 1$ and are suppressed for $q \neq 1$.

D. Revolving the RNG Output

It will be seen that in the scheme of the Generator, there are correlations between the first bits of $a^{(n)}$, correlations between the second bits of $a^{(n)}$, and so on. In order to suppress these correlations, we modify the algorithm as follows. At each step, we renumber the points of the generator: $1 \to 2, 2 \to 3, \ldots, s \to 1$. In other words, the bits inside $a^{(n)}$ are rotated, and the RNG output function is defined as $b^{(n)} = \sum_{i=0}^{s-1} \alpha_i^{(n)} \cdot 2^{(i+n) \pmod s}$ instead of $a^{(n)} = \sum_{i=0}^{s-1} \alpha_i^{(n)} \cdot 2^i$, where $\alpha_i^{(n)} = \lfloor 2x_i^{(n)}/g \rfloor$.

The main advantage of the modified algorithm is that it leads to decreasing the correlations of the values $a^{(n)}$ between each other. For example, we will see in Sec. IV that the rotation strongly reduces the specific correlations found by random walks test.

We note that the rotation of bits in the RNG output does not deteriorate any properties of the RNG provided that s divides the period of free orbits T_m (in practice, this is very realistic condition). In particular, neither the period of the generator becomes smaller (see Appendix B), nor the statistical properties become worse.

The rotation of bits in the RNG output is thus a practically useful modification. In addition, the rotation makes deciphering an even more complicated problem.

In the next section most promising generators are thoroughly tested. In particular, it is shown that using the ensemble of transformations while taking only an elder bit from the point of each cat map suppresses the major correlations of a simple cat-map generator, based on single cat map (see Sec. III A). It was also obtained empirically, that even small deviations from the RNG scheme presented in this section (e.g., constructing $a^{(n)}$ from different and non-first bits of the integers $x_0^{(n)}, x_1^{(n)}, \ldots, x_{s-1}^{(n)}$) result in correlations appearing and the RNG properties deteriorating.

III. STATISTICAL TESTS

A. Simple Knuth Tests

In this section, we present the results of a number of standard statistical tests [6], revealing the correlation properties of the generator described in Sec. II. Namely, the frequency test, serial test, maximum-of-t test, test for monotonic subsequences ("run test"), and collision test were applied for an RNG with $M = \begin{pmatrix} 2 & 3 \\ 3 & 5 \end{pmatrix}$, $g = 2^m = 2^{28}$ and $s = 28$ points in the state. All the statistical tests were passed. All empirical tests except the collision test (CT) are based on either the chi-square test (χ^2) or the Kolmogorov-Smirnov test (KS). We follow Knuth's notation [6].

The results of the tests are presented in Table I, where n is number of values of $a^{(n)}$ for each test (for the serial test, n is the number of pairs $\{a^{(2n)}, a^{(2n+1)}\}$) and ν is the number of degrees of freedom. For the serial test $d = 8$, i.e., we used exactly 3 bits of each $a^{(n)}$ number; hence, $\nu = d^2 - 1 = 63$. For the run test, $\nu = 5$ means that we sought monotonic subsequences of lengths 1,2,3,4,5 and of length ≥ 6.

For all of the KS tests, the empirical distributions of $P(K^+)$ and $P(K^-)$ were calculated, where $P(x)$ is the theoretical Kolmogorov-Smirnov distribution [6]. Figure 1 shows these empirical distributions for the frequency test. These distributions lead to their own values of K^+ and K^-: the values $P(K'^+)$ and $P(K'^-)$ characterizing the empirical distribution of K^+ and the values $P(K'^+)$ and $P(K'^-)$ characterizing the empirical distribution of K^-. These values are presented in Table I. Our RNG passes all the KS tests because the values K^+ and K^- are distributed in accordance with the theory prediction. For each chi-square test, the empirical distribution of 20 values of $P(V)$ was calculated. In our tests, it looks similar to those shown in Fig. 1, where $P(x)$ is the theoretical chi-square distribution and V is the output of the chi-square test. For each collision test, the number of collisions c and the theoretical probability $P(c)$ that the number of collisions is not larger than c were calculated. The empirical distribution of $P(c)$ was analyzed, and the results are also presented in Table I.

TABLE I: Results of the statistical tests for the RNG based on the ensemble of cat maps (see Sec. II) with parameters $M = \begin{pmatrix} 2 & 3 \\ 3 & 5 \end{pmatrix}$, $g = 2^m = 2^{28}$, $s = 28$.

Test	Parameters	Number and type of tests		Tests output values V_1 and V_2	Distribution of V_1		Distribution of V_2		Conclusion
					$P(K'^+)$	$P(K'^-)$	$P(K'^+)$	$P(K'^-)$	
Frequency test	$n = 10^6$	20	KS	$V_1 = K^+, V_2 = K^-$	0.592392	0.174451	0.457758	0.473379	PASSED
Serial test	$n = 10^6, d = 8$	20	χ^2	$V_1 = V$	0.837112	0.17128	n/a	n/a	PASSED
Run test	$n = 10^6, \nu = 5$	20	χ^2	$V_1 = V$	0.383601	0.805434	n/a	n/a	PASSED
Maximum-of-t test	$n = 10^6, t = 5$	20	KS	$V_1 = K^+, V_2 = K^-$	0.120912	0.765201	0.704026	0.589702	PASSED
Collision test	$m = 2^{20}, n = 2^{14}$	20	CT	$V_1 = c$	0.150537	0.858955	n/a	n/a	PASSED

We note that all the empirical tests here except the collision test are essentially multibit. This means that the whole ensemble of cat maps influences the test result, and one can guess that hidden variables inside the generator is one reason for the successful test results. Therefore, this is also useful to analyze the properties of a single-bit cat-map generator, i.e., properties of a generator from Sec. II with $s = 1$. Tests for a single-bit cat-map generator are also successfully passed. Namely, the frequency test and the serial test, which were modified for a one-bit generator, and the collision test (which is a single-bit test even in Table I) are passed. But there are correlations in the single-bit cat-map generator (discussed later in this paper), and the most convenient method for observing them is the random
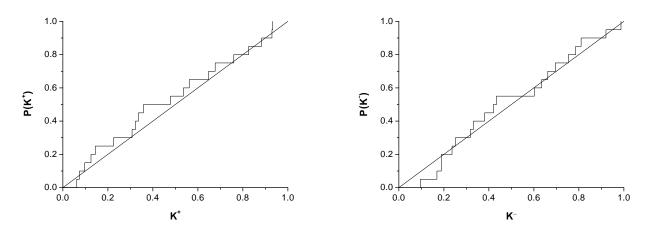
FIG. 1: Distribution of $P(K^+)$ and $P(K^-)$ for the frequency test. The test was performed for the RNG from Sec. II with the parameters $M = \left(\begin{smallmatrix} 2 & 3 \\ 3 & 5 \end{smallmatrix}\right)$, $g = 2^m = 2^{28}$, $s = 28$.

walks test with $\mu = 1/2$ (see Section IV). The random walks test is not the only test that can reveal the single-bit cat-map correlations. The same correlations are also observed by improved versions of some of the standard tests, e.g., the serial test for subsequences of length 5. Of course, many tests in Sec. III B would not be passed for a single-bit cat-map generator.

For comparison, we analyzed a simple generator based on the single cat map. Table II shows that such generator, with the transition function defined as $\left(\begin{smallmatrix} x^{(n)} \\ y^{(n)} \end{smallmatrix}\right) = M\left(\begin{smallmatrix} x^{(n-1)} \\ y^{(n-1)} \end{smallmatrix}\right)$ (mod 1), and the output function defined as $u_n = x^{(n)}$, has very bad properties. Of course, the frequency test is passed, since the trajectories of a cat map uniformly fill the phase space. However, all other tests are not passed. Therefore, the simple generator based on the single cat map does have strong correlations in the output and practically is not useful.

TABLE II: Results of the statistical tests for a simple RNG based on the single cat map, with parameters $M = \left(\begin{smallmatrix} 2 & 3 \\ 3 & 5 \end{smallmatrix}\right)$, $g = 2^m = 2^{28}$.

Test	Parameters	Number and type of tests		Tests output values V_1, V_2	Distribution of V_1		Distribution of V_2		Conclusion
					$P(K'^+)$	$P(K'^-)$	$P(K'^+)$	$P(K'^-)$	
Frequency test	$n = 10^7$	20	KS	$K^+; K^-$	0.96235	0.170581	0.787296	0.067341	PASSED
Serial test	$n = 5 \cdot 10^6, d = 8$	20	χ^2	V	0.989499	0.006013	n/a	n/a	NOT PASSED
Run test	$n = 10^6, \nu = 5$	20	χ^2	V	0	1	n/a	n/a	NOT PASSED
Maximum-of-t test	$n = 10^6, t = 5$	20	KS	$K^+; K^-$	0	1	0	1	NOT PASSED
Collision test	$d = 4, m = 2^{10}, n = 2^{14}$	20	CT	c	0	1	n/a	n/a	NOT PASSED

B. Batteries of Stringent Statistical Tests

Knuth tests are very important, but still not sufficient for the present-day sound analysis of the RNG statistical properties. The hundreds of known statistical tests and algorithms are available in software packages, for example, widely used packages DieHard [35], NIST [36] and TestU01 [34]. All of them include tests, described by Knuth [6], as well as many other tests.

Table III shows the summary results for the SmallCrush, PseudoDiehard, Crush and Bigcrush batteries of tests from [34]. SmallCrush contains 14 tests, PseudoDiehard – 126 tests, Crush – 93 tests, Bigcrush – 65 tests. The detailed parameters and initializations for generators GS, GR, GSI, GRI, GM19 and GM31, based on the scheme proposed in Section II, are given in Appendix F.

For comparison, we also test several other generators. Namely, the standard generators RAND, RAND48, RANDOM, and modern generators MT19937 and LFSR113. RAND is the simple LCG generator based on the recurrence $x_n = (1103515245\ x_{n-1} + 12345)$ (mod 2^{31}). RAND48 is the 64-bit LCG based on the recurrence

$x_n = 25214903917\, x_{n-1} + 11 \pmod{2^{48}}$. RANDOM provides an interface to the set of five additive feedback random number generators. RAND, RAND48 and RANDOM are implemented in the functions `rand()`, `rand48()` and `random()` in the standard Unix or Linux C library `stdlib` (see the documentation to `rand()`, `rand48()` and `random()`). MT19937 is the 2002 year version of Mersenne Twister generator of Matsumoto and Nishimira [14], which is based on the recent generalizations to the GFSR method. LFSR113 is the Combined Tausworthe generator of L'Ecuyer [17].

The detailed statistics for the batteries of tests and the explicit results for every single test from the batteries can be found in [49].

TABLE III: Number of unpassed tests for the batteries of tests SmallCrush, Crush, Bigcrush [34], and DieHard [35]. Here $k = \text{Tr}\,M$ and $q = \det M$ are the RNG parameters (see Sec. II and Appendix F), and a statistical test is considered to be unpassed if the p-value is outside the interval $[1/100, 1 - 1/100]$.

Generator	k	q	SmallCrush	Diehard	Crush	Bigcrush
GS	3	1	0	44	20	22
GR	3	1	0	5	5	15
GSI	11	1	0	1	10	13
GRI	11	1	1	6	5	13
GM19	15	28	0	2	2	3
GM31	7	11	0	2	3	1
RAND	–	–	13	88	102	85
RAND48	–	–	5	27	22	27
RANDOM	–	–	3	17	13	21
LFSR113	–	–	0	3	8	8
MT19937	–	–	0	2	1	4

Most of the p-values for the unpassed tests for the cat map generators are of the order of 10^{-3} to 10^{-5}, however, several are very small. We believe that the reason for small p-values is connected with the the small period of the generators GS, GR, GSI, GRI and UNIX RAND. The period of the order of $3 \cdot 10^9$ being sufficient for many applications, is not sufficient for many of the tests from Crush and Bigcrush. Therefore, the generators GS, GR, GSI and GRI demonstrate smaller p-values and larger number of unpassed tests from Crush and Bigcrush.

The existence of linear congruential dependencies between orbits is another reason for small p-values for GS, GR, GSI and GRI. These correlations are analytically described in Appendix C. The GM19 and GM31 generators, having period sufficient for Crush and Bigcrush batteries, are at the same time free from the linear congruential dependencies. Therefore, they demonstrate much better statistical properties in Table III.

Of course, the number of unpassed tests is liable to random statistical flukes, especially when many p-values lie in the suspect region $[10^{-2}, 10^{-5}] \cup [1 - 10^{-2}, 1 - 10^{-5}]$. Table IV illustrates the flukes by showing the results of all batteries of tests for GM31 generator. The batteries were executed in the following order: SmallCrush, SmallCrush, PseudoDiehard, PseudoDiehard, Crush, Crush, BigCrush and BigCrush, i.e. each battery was executed twice. For tests in Tables III and IV generator GM31 was initialized with identical parameters, in accordance with Appendix F. The numbers of unpassed tests themselves in Table III and Table IV approximately indicate the statistical robustness of generators. However, if the p-value lie in the suspect region, one doesn't know exactly whether the systematic correlations in the RNG were found or the statistical fluke happened.

We conclude that the best of the generators based on cat maps are competitive with other good modern generators. In particular, we recommend the RNG realizations for GRI and GM31 based on the SSE2 command set for the practical use. In Appendix F we present the effective realizations of generators GRI-SSE and GM31-SSE and recipes for the proper initialization. These are the best realizations, among generators examined here with $g = 2^m$ and with prime g correspondingly.

IV.   THE RANDOM WALKS TEST

Analyzing theoretically the statistical properties of the generator is another important challenge. Traditionally such analysis is performed by the discussion of lattice structure [26, 28, 30] and the discussion of discrepancy [33]. The discrepancy of matrix generator was analyzed by Niederreiter [27] who, in particular, proved that the behaviour of the discrepancy is strongly connected to the behaviour of integer called *figure of merit*. Although the calculation of the exact values of the figure of merit would give excellent basis for practical selection of matrices for matrix generators,

TABLE IV: Applying each battery of tests twice for the GM31 generator.

	Number of	Unpassed tests, testing first time			Unpassed tests, testing over again		
	unpassed Tests	No	Name	p-value	No	Name	p-value
SmallCrush	0/0	–	–	–	–	–	–
PseudoDiehard	3/2	1	BirthdaySpacings	0.0071	6	CollisionOver	0.0014
		6	CollisionOver	0.999	7	CollisionOver	0.0018
		14	Run of U01	0.0093			
Crush	2/2	70	Fourier1, $r = 0$	0.0095	14	BirthdaySpacings, $t = 7$	0.0024
		71	Fourier1, $r = 20$	0.0093	70	Fourier1, $r = 0$	0.0033
BigCrush	4/3	6	MultinomialBitsOver	0.0014	32	SumCollector	0.0067
		23	Gap, $r = 0$	0.9970	36	RandomWalk1 J(L=90)	0.9952
		27	CollisionPermut	0.0052	41	RandomWalk1 J(L=10000)	0.9965
		39	RandomWalk1 H(L=1000)	0.9982			

these values are still very hard to compute. To the best of our knowledge, this calculation was never done for matrix generators of pseudorandom numbers.

Because of the hidden variables, the lattice structure of matrix generator does not directly influence the statistical properties of the RNG introduced in Sec. II. Instead, we perform the search for correlations of other nature using the random walks test. The random walks test proved sensitive and powerful for revealing correlations in RNG. In particular, correlations in the shift register RNG were found [40] and explained [41, 42] by means of the random walks tests. In addition, random walks test is useful tool for analysis and usually provides a clear view of the correlation mechanism in pseudorandom numbers.

There are several variations of the random walks test in different dimensions [43]. We consider the one-dimensional directed random walk model [41]: a walker starts at some site of an one-dimensional lattice, and at discrete times i, he either takes a step in a fixed direction with probability μ or stops with probability $1 - \mu$. In the latter case, a *new* walk begins. The probability of a walk of length n is $P(n) = \mu^{n-1}(1 - \mu)$, and the mean walk length is $\langle n \rangle = 1/(1 - \mu)$. We note that the Ising simulations using cluster updates with the Wolff method are closely related to the random walk problem [42]. Namely, the mean cluster size in the Wolff method equals the mean walk length for $\mu = \tanh(J/k_B T)$, where J is the strength of the spin coupling and T is the temperature.

Figure 2 shows the correlations in the RNG found by the random walks test. We applied 100 chi-square tests. Each test performed $n = 10^7$ random walks with $\mu = 1/2$ for a generator with $M = \begin{pmatrix} 2 & 3 \\ 3 & 5 \end{pmatrix}$, $m = 32$, $s = 1$. The result of the test with $\mu = 1/2$ does not depend on s, because only the first bit of the RNG is taken into account. For each test, the value $\delta P_l = (Y_l - np_l)/(np_l)$ was calculated for all walk lengths $l \leq 7$. Here, p_l is the theoretical probability of walk length l for uncorrelated random numbers, Y_l is the simulated number of walks with length l. We note that correlations can be found only for large number of random walks (see Table VI), and no correlations are found even for $n = 6 \cdot 10^4$ random walks.

These correlations can be explained as follows. There are 32 five-bit sequences, and they do not have the same frequency of appearing in the RNG output. We consider one of them, for example, 10011. Let $X = (0, \frac{1}{2}] \times (0, 1]$ and $Y = (\frac{1}{2}, 1] \times (0, 1]$, i.e., X and Y are the left and the right halves of the torus. Let x be the initial point $\begin{pmatrix} x_0^{(0)} \\ y_0^{(0)} \end{pmatrix}$ of the generator. For the first bits of the first five outputs of the generator to be 10011, it is necessary and sufficient to have $x \in Z_{10011} = Y \cap R^{-1}(X) \cap R^{-2}(X) \cap R^{-3}(Y) \cap R^{-4}(Y)$. Here, R is the action of the cat map. The set Z_{10011} consists of polygons. Each polygon can be calculated exactly. The area $S(Z_{10011})$ equals the probability for the first five outputs of the generator to be 10011. This shows that the nature of the correlations is found in the geometric properties of the cat map.

Figure 3 (the left picture) represents the polygons corresponding to the subsequences of length three for the cat map with $M = \begin{pmatrix} 2 & 3 \\ 3 & 5 \end{pmatrix}$. Each set of polygons, e.g., $Z_{010} = X \cap R^{-1}(Y) \cap R^{-2}(X)$, represents the region on the torus for the first initial point of the RNG and is drawn with its own color. The right picture represents the subsequences of length five for the cat map with $M = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$. Here, each set of polygons represents the regions on the torus for the third point of the generator, e.g., $\tilde{Z}_{01001} = R^{-2}(X) \cap R^{-1}(Y) \cap X \cap R(X) \cap R^2(Y)$, and is drawn with its own color. Of course, $S(\tilde{Z}_{01001}) = S(Z_{01001}) = P(01001)$ because the cat maps are area preserving. Therefore, the choice of pictures of Z_i or pictures of $\tilde{Z}_i$ is unimportant if we only want to calculate the areas. Thus, the geometric structures in Fig. 3 show the regions of $Z_{000}, \ldots, Z_{111}$ (the left picture) or $\tilde{Z}_{00000}, \ldots, \tilde{Z}_{11111}$ (the right picture) and illustrate the geometric approach to calculating the probabilities.

The exact areas $S(Z_{00000}), \ldots, S(Z_{11111})$ can be easily calculated for various toral automorphisms. We prove the
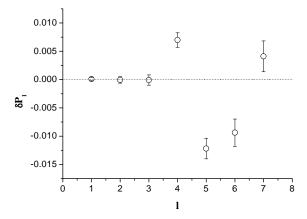
FIG. 2: The deviation δP_l of the probability of a walk length l from the value for uncorrelated random numbers versus walk length l. The mean value and the variance for δP_l are represented for 100 chi-square random walk simulations. See text for details.
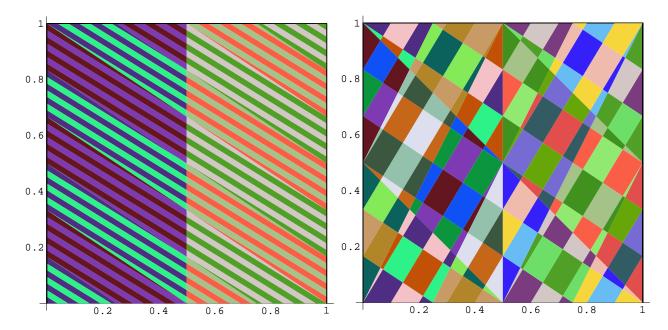

FIG. 3: (Color online) Left: the regions on the torus for the first initial point of the RNG described in Sec. II with $M = \left(\begin{smallmatrix} 2 & 3 \\ 3 & 5 \end{smallmatrix} \right)$. These regions correspond to the sequences $000, 001, 010, 011, 100, 101, 110, 111$ of first bits generated by the RNG. Each region is drawn with its own color. Right: the regions on the torus for the third point of the RNG, described in Sec. II with $M = \left(\begin{smallmatrix} 1 & 1 \\ 1 & 2 \end{smallmatrix} \right)$. These regions correspond to the sequences of length five of first bits generated by the RNG. Each region is drawn with its own color.

following geometric propositions:

1. In any case, every subsequence of length 3, 2, or 1 respectively has the same probability $1/8$, $1/4$, or $1/2$.

2. If $k = \mathrm{Tr}(M)$ is an odd number, then every subsequence of length 4 has the same probability $P_0 = 1/16$.

3. If k is even, then the probability of the subsequence 0000 depends only on the trace k of matrix M of the cat map. It equals $P = P_0 \cdot k^2/(k^2 - 1)$, where $P_0 = 1/16$.

The line of reasoning is presented in Appendix E and in [46]. Of course, the probability of subsequence 0000 automatically gives the probabilities of all other subsequences of length four. We note that if k is odd, then the ideal $\langle 2 \rangle$ is inert (see Appendix A 3), and the inert case is the easiest for exact analysis of the RNG period (see Appendix A, B).

The probabilities of the subsequences of length five for maps with odd traces and of the subsequences of length four for maps with even traces are calculated exactly and shown in Table V. It can be conjectured from Table V that if k is odd, then the probability of the subsequence 00000 of length five equals $P_0 \cdot (1 + 1/(3k^2 - 6))$, where $P_0 = 1/32$.

TABLE V: The probabilities of subsequences for different cat maps, characterized by the trace k.

k	$P(0000)/P_0$	k	$P(0000)/P_0$	k	$P(00000)/P_0$
4	16/15	30	900/899	3	22/21
6	36/35	32	1024/1023	5	70/69
8	64/63	34	1156/1155	7	142/141
10	100/99	36	1296/1295	9	238/237
12	144/143	38	1444/1443	11	358/357
14	196/195	40	1600/1599	13	502/501
16	256/255	42	1764/1763	15	670/669
18	324/323	44	1936/1935	17	862/861
20	400/399	46	2116/2115	19	1078/1077
22	484/483	48	2304/2303	21	1318/1317
24	576/575	50	2500/2499	23	1582/1581
26	676/675	52	2704/2703	25	1870/1869
28	784/783	54	2916/2915	27	2182/2181

The probabilities can thus be approximated as $P/P_0 = 1 + Bk^{-2}$ for large k, where $P_0 = 2^{-n}$ for subsequences of length $n = 4, 5$. Here, $B = 1$ when k is even and $n = 4$, $B = 1/3$ when k is odd and $n = 5$. We conclude that the deviations found by the random walks test will vanish as the trace k increases.

Table VI shows that using rotation in the RNG output (see Sec. II D) results in suppressing correlations found by the random walks test. This is not surprising, because even the one-bit random walks test with $\mu = 1/2$ deals with the ensemble of cat maps, when the rotation is used.

V.   DISCUSSION

In this paper, we have proposed a scheme for constructing a good RNG. The distinctive features of this approach are applying the ensemble of cat maps while taking only a single bit from the point of each cat map and applying methods that allow to analyze both the properties of the periodic orbits and the statistical properties of such a generator both theoretically and empirically. We have seen that the algorithm in Sec. II can generate sequences with very large period lengths. Although the essential correlations are always present and the important statistical deficiencies are found, good algorithm with proper initialization is able to minimize them. The best generators created by this method have statistical properties not worse, and the speed rate slightly slower than that of the good modern RNGs. The techniques used allow calculating the period lengths and correlation properties for a wide class of sequences based on cat maps.

Future modifications and enhancements are possible, and at present time we recommend generators GRI-SSE and GM31-SSE for the practical use. One may find program codes for the generators and for the proper initialization

TABLE VI: Left: results of the random walks test (i.e. 100 chi-square random walks simulations) for different n and for $\mu = 1/2$, $m = 32$, $s = 1$. Right: results of the random walks test for different μ and s and for $n = 10^6$, $m = 32$. Here $s \neq 1$ means using rotation in the RNG output (see Sec. II D). Actually, the same one-bit random walks test is used because $\mu = 1/2$.

n	$P(K'^+)$	$P(K'^-)$	Result	μ	s	$P(K'^+)$	$P(K'^-)$	Result
10^4	0.746106	0.594428	PASSED	1/4	1	0.235776	0.882413	PASSED
$3 \cdot 10^4$	0.341899	0.675728	PASSED	3/4	1	0.299174	0.613382	PASSED
$6 \cdot 10^4$	0.307433	0.694282	PASSED	1/8	1	0.087016	0.770549	PASSED
10^5	0.018332	0.966717	UNCERTAIN	1/16	1	0.67967	0.920998	PASSED
$3 \cdot 10^5$	0.0001594	1	NOT PASSED	1/2	2	0.605407	0.344068	PASSED
$6 \cdot 10^5$	0.0001378	1	NOT PASSED	1/2	3	0.527088	0.645272	PASSED
10^6	0	1	NOT PASSED	1/2	4	0.558105	0.360828	PASSED

in [49] and discussion of generator details in Appendix F. Authors would appreciate any comments on the user experience.

VI.   ACKNOWLEDGMENTS

APPENDIX A: PERIODIC ORBITS OF THE CAT MAPS ON THE $2^n \times 2^n$ LATTICE

In this section, we review the key arithmetical methods for studying orbit periods that are described in detail in [29]. Some of the results are presented in this appendix in a more general form. Notations are discussed briefly, the details and proofs on the formalism of quadratic integers and quadratic ideals can be found in [44, 45].

1.   The dynamics of the cat map and rings of quadratic integers

We consider the unit two-dimensional torus (the square $(0,1] \times (0,1]$ with the opposite sides identified). We take a cat map $M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \in SL_2(\mathbb{Z})$, which acts on a lattice $g \times g$ on the torus, where $g = 2^n$. The elements of M are integers, $\det M = 1$, and $|k| > 2$, where $k = \mathrm{Tr}(M)$.

For any given trace $k > 2$, there exists a unique map $M \in SL_2(\mathbb{Z})$ such that the connection between the properties of periodic orbits of the automorphism and the arithmetic of quadratic integers is the most natural. Indeed, we consider a matrix M such that

$$\begin{cases} \lambda &= m_{11} + \tau m_{21} \\ \lambda\tau &= m_{12} + \tau m_{22} \end{cases}. \tag{A1}$$

Here, τ is the base element of the ring of quadratic integers $R_D = \{a + b\tau : a, b \in \mathbb{Z}\}$ that contains λ. This means that $\exists n \in \mathbb{Z} : k^2 - 4 = n^2 D$, where D is a squarefree integer and $\tau = \sqrt{D}$ for $D \not\equiv 1 \pmod 4$; $\tau = \frac{1}{2}(1 + \sqrt{D})$ for $D \equiv 1 \pmod 4$.

It easily follows from (A1) that $x' + y'\tau = \lambda(x + y\tau)$ is equivalent to $\begin{pmatrix} x' \\ y' \end{pmatrix} = M\begin{pmatrix} x \\ y \end{pmatrix}$ for any x, y, x', y'. Indeed, $\lambda(x + y\tau) = \lambda x + (\lambda\tau)y = (m_{11}x + m_{12}y) + (m_{21}x + m_{22}y)\tau = x' + y'\tau$. The action of the map M corresponds to multiplication by the quadratic integer λ, while the action of M^{-1} corresponds to multiplication by λ^{-1}. Hence, we can choose either of the two eigenvalues $\lambda = (k \pm \sqrt{k^2 - 4})/2$, e.g., the largest one, because the exact choice is unimportant for studying orbit periods.

Generally speaking, there are infinitely many maps in $SL_2(\mathbb{Z})$ that have identical eigenvalues, and not all the maps are related by a canonical transformation (share the same dynamics). But arguments presented in [29] strongly suggest that they still share the same orbit statistics.

2.   Invariant sublattices on the torus and the factorization of quadratic ideals

We note that each element of R_D represents some point of $\mathbb{Z}^2$. Let A be a quadratic ideal. We say that $\xi \equiv \eta \pmod A$ if $(\xi - \eta) \in A$. We consider the principal quadratic ideal generated by g: $\langle g \rangle = \{ag + bg\tau : a, b \in \mathbb{Z}\}$. It corresponds to the set of points of a square lattice with the side g. Then the period of an orbit containing the point $\begin{pmatrix} x/g \\ y/g \end{pmatrix}$ is the smallest integer T such that $\lambda^T z \equiv z \pmod{\langle g \rangle}$. Here, x and y are integers, and $z = x + y\tau$.

Each quadratic ideal A is associated with some sublattice of $\mathbb{Z}^2$. Because λ is a unit, the sublattice is invariant with respect to multiplication by λ: $\lambda A = A$. Since we are interested in invariant lattices on the unit two-dimensional torus, we consider only those sublattices of $\mathbb{Z}^2$ that are invariant under an arbitrary translation $\begin{pmatrix} ag \\ bg \end{pmatrix}$, where $a, b \in \mathbb{Z}$. These sublattices correspond to quadratic ideals that divide $\langle g \rangle$. The factorization of the ideal $\langle g \rangle$ thus yields invariant sublattices on the torus.

3.   The classification of prime ideals and the orbit periods for the $2^n \times 2^n$ lattice

We consider $2^n \times 2^n$ lattices on the torus. Because $\langle g \rangle = \langle 2^n \rangle = \langle 2 \rangle^n$, it is sufficient to have the ideal factorization of $\langle 2 \rangle$. We recall that the ideal $\langle 2 \rangle$ is said to be *inert* if $\langle 2 \rangle$ is already a prime ideal; it is said to be *split* if $\langle 2 \rangle = P_1 P_2$, where P_1 and P_2 are prime ideals; it is said to be *ramified* if $\langle 2 \rangle = P_1^2$, where P_1 is a prime ideal. The ideal $\langle 2 \rangle$ is inert for $D \equiv 5 \pmod 8$, split for $D \equiv 1 \pmod 8$, and ramified for $D \not\equiv 1 \pmod 4$.

It follows that if the trace k is odd, then $\langle 2 \rangle$ is inert; if $k \equiv 0 \pmod 4$, then $\langle 2 \rangle$ is ramified. Indeed, for odd k, we have $k^2 - 4 \equiv 5 \pmod 8 \Rightarrow D \equiv 5 \pmod 8$; for $k \equiv 0 \pmod 4$, we have $(k^2 - 4)/4 = n_1^2 D \equiv 3 \pmod 4 \Rightarrow D \equiv 3 \pmod 4$. For $k \equiv 2 \pmod 4$, we obtain $(k^2 - 4)/4 = n_1^2 D \equiv 0 \pmod 4$, i.e., all three possibilities (inert, split, or ramified ideal $\langle 2 \rangle$) can occur.

Let T_n denote the period of any of the free orbits for $g = 2^n$ and T_n' denote the period of those ideal orbits for $g = 2^n$ that do not belong to the sublattice $\frac{g}{2} \times \frac{g}{2}$. We recall that an orbit belonging to a given lattice $\mathbb{Z}^2 / g\mathbb{Z}^2$ is called an *ideal orbit* if it belongs to some ideal A such that $A | \langle g \rangle$ and $A \neq \langle 1 \rangle$. Otherwise, it is called a *free orbit*.

The behavior of periodic orbits on the 2×2 lattice follows from Propositions B1–B3 in [29]. Namely, we have the following:

- If $\langle 2 \rangle$ is inert, then either $T_1 = 3$ or $T_1 = 1$; all orbits are free.

- If $\langle 2 \rangle$ is split, then $T_1 = T_1' = 1$; there are two ideal orbits and one free orbit.

- If $\langle 2 \rangle$ is ramified, then $T_1 = 2$ and $T_1' = 1$; there is an ideal orbit and a free orbit (it is also possible that $T_1 = 1$ and $T_1' = 1$; there are two free orbits and an ideal orbit).

To determine the structure of periodic orbits on the $2^n \times 2^n$ lattice, we prove the following theorem.
Theorem.

1. For all n, either $T_{n+1} = 2T_n$ or $T_{n+1} = T_n$.

2. For all n, either $T_n' = T_n$ or $T_n' = T_{n-1}$.

3. For all $n \geq 3$, $T_n \neq T_{n-1} \Rightarrow T_{n+1} \neq T_n$.

4. If $n \geq 4$, $T_n \neq T_{n-1}$, and $T_n' = T_n/a$, where $a \in \{1, 2\}$, then $T_{n+1}' = T_{n+1}/a$.

This theorem generalizes Propositions C1 and C2 in [29]. The line of reasoning is presented in Appendix D.

Therefore, knowing T_n and T_n' for small n suffices for determining the orbit statistics for all n. There always exist n_1, n_2, and n_3 such that $T_n = T_1 2^{n - n_1}$ and $T_n' = T_1 2^{n - n_2}$ for all $n \geq n_3$.

If $\langle 2 \rangle$ is inert, then every ideal that divides $\langle g \rangle$ has the form $\langle 2 \rangle^r$. Therefore, each ideal orbit belongs to the $2^{n-1} \times 2^{n-1}$ sublattice and coincides with a free orbit for some sublattice $2^r \times 2^r$, where $r < n$. We now find the number of free orbits in the inert case. There are $2^{2n} - 1$ points on a lattice. The ideal orbits contain $2^{2n-2} - 1$ points. Consequently, there are $(2^{2n} - 2^{2n-2})/T_n = 3 \cdot 2^{2n-2}/T_n$ free orbits.

We suppose that the typical inert case occurs, i.e., $T_n = 3 \cdot 2^{n-2}$. Then the phase space is divided into the following regions:

- $3/4$ of the phase space is swept by 2^n trajectories of period T_n,

- $3/16$ of the phase space is swept by 2^{n-1} trajectories of period $T_{n-1} = T_n/2$,

- $3/64$ of the phase space is swept by 2^{n-2} trajectories of period $T_{n-2} = T_n/4$,

- and so on. All such statements hold as long as the trajectory length exceeds just a few points.

Therefore, on one hand, cat map orbits have huge periods; on the other hand, the number of orbits is sufficiently large (see Theorem 2 in Appendix B). Both these properties are important for our construction of the RNG.

APPENDIX B: THE RNG PERIOD

In this section, we find the periods of the generators in Sec. II and Sec. II D. As a result of Appendix A and Appendix B, the RNG period can be obtained for arbitrary parameters of the map and lattice.

Theorem 1. If $g = 2^m$, then the period T of the sequence $\{a^{(n)}\}$ in Sec. II equals the period T_m of free orbits of the cat map for the overwhelming majority of RNG initial conditions.
Proof.

1. At least one of the initial points $\binom{x_i^{(0)}}{y_i^{(0)}}$ belongs to a free orbit. Indeed, the probability of this in the inert case equals $(1 - 4^{-s})$.

2. Therefore, T is not less than T_m. Indeed, T is not less than the period of the sequence of first bits of $x_i^{(0)}, x_i^{(1)}, \ldots$ for each i. But the period of the sequence of first bits of points of the cat map orbit is equal to the orbit period for the vast majority of orbits. The probability of the opposite is tiny provided that the orbit is not too short.

3. Finally, T is not larger than T_m. Indeed, the period of each cat map orbit divides T_m.

Example. In the typical example of the inert case, where $M = \binom{4\ 9}{3\ 7}$, we obtain $T_m = 3 \cdot 2^{m-2}$. This fact was also tested numerically as follows. First, the initial conditions were set randomly. Second, the period of $\{a^{(n)}\}$ was accurately found numerically. This operation was repeated 1000 times for $m = s = 14$. Each time the period of $\{a^{(i)}\}$ turned out to be $6144 = 3 \cdot 2^{11}$. To check the period numerically, we first check whether the whole state of the RNG (not only the output) coincides at the moments 0 and T and then verify that a smaller period (which could possibly divide T) does not exist.

Theorem 2. The probability that two arbitrary points of the $2^m \times 2^m$ lattice on the torus belong to the same orbit of the cat map equals $9/(7 \cdot 2^{m+2})$. The probability that s arbitrary points of the lattice do not belong to s different orbits of the cat map (i.e., two of the points belong to the same orbit) is $9s(s-1)/(7 \cdot 2^{m+3})$.

The proof of Theorem 2 is straightforward. Of course, both these probabilities are tiny if m is sufficiently large.

Theorem 3. If $g = 2^m$ and $s|T_m$, then the period T of the sequence $\{b^{(n)}\}$ in Sec. II D equals T_m for the overwhelming majority of RNG initial conditions.

Proof.

1. Because $s|T_m$, we have $b_{i+T_m} = b_i$ for all i. Therefore, $T|T_m$.

2. If s does not divide T, then $\forall i \in \{0, 1, \ldots, s-1\} \exists j \in \{0, 1, \ldots, s-1\}, j \neq i$, such that $\binom{x_i^{(0)}}{y_i^{(0)}}$ and $\binom{x_j^{(0)}}{y_j^{(0)}}$ belong to the same orbit of the cat map. It follows from Theorem 2 that this event is highly improbable. Therefore, $s|T$.

3. Because T is a period of $\{b^{(n)}\}$ and $s|T$, we have $b_{i+T} = b_i \Rightarrow a_{i+T} = a_i$ for all i. Therefore, $T_m|T$.

The above theorems show that the period calculations for sequences $\{a^{(n)}\}$ and $\{b^{(n)}\}$ are reliable in general case, because the chance of the period dependence on the initial state is exponentially small. However, it is desirable property that the period does not depend on any conditions at all. The proper initialization (see Appendix F) guarantees that i) at least one of the initial points belong to a free orbit of the cat map; ii) all initial points belong to different orbits of the cat map. Therefore, both the periods of $\{a^{(n)}\}$ and $\{b^{(n)}\}$ are guaranteed to equal T_m provided the initialization in Appendix F is applied.

The above theorems and considerations hold for $g = 2^m$. In the other case, when $g = p$ is a prime integer, it follows from the finite field theory that the period of any orbit of the matrix transformation is equal to $p^2 - 1$ provided the polynomial $f(x) = x^2 - kx + q$ is primitive modulo p. The methods for good parameter and initialization choice for such generators are also presented in Appendix F for generators GM19 and GM31. The similar argumentation as in Theorems 1 and 3 show that in this case i) the period of the sequence $\{a^{(n)}\}$ equals $p^2 - 1$; ii) the period of the sequence $\{b^{(n)}\}$ is divisible by $p^2 - 1$, i.e. rotation cannot decrease the period of such generator.

APPENDIX C: ORBITS, NORM AND CORRELATIONS BETWEEN ORBITS

In this section i) we show that the norm modulo g is the characteristic of the whole orbit; ii) we find the number of orbits of each norm modulo g and discuss how symmetries affect the norm; iii) we find the linear congruential dependencies between orbits. The consideration holds for maps with $q = 1$ on a $2^m \times 2^m$ lattice.

1. Orbits and Norm

We recall that the norm of a quadratic integer $\alpha = a + b\sqrt{D}$ is simply an integer $N(\alpha) = \alpha\alpha^* = a^2 - b^2 D$. If $\langle 2 \rangle$ is inert, then the quadratic integer $x + y\tau$, where $\tau = \frac{1+\sqrt{D}}{2}$ represents the point $\binom{x}{y}$, i.e. $N\binom{x}{y} = x^2 + xy - \frac{D-1}{4}y^2$. A cat map preserves the value of $N\binom{x}{y}$ (mod 2^m), because the action of a cat map can be described as $x' + y'\tau = \lambda(x + y\tau)$ (mod $\langle 2^m \rangle$), where λ is a matrix eigenvalue and $N(\lambda) = 1$. Therefore, norm modulo g is a characteristic of

the whole orbit. We note that for a point on a free orbit, either x or y is odd, consequently the norm is also an odd number.

Let us prove that if the period of free orbits is $T = 3 \cdot 2^{m-2}$, then for each $N = 1, 3, \ldots, 2^m - 1$ there are exactly two orbits that have the norm N (these two orbits are symmetrical, i.e. the second one contains the points $\binom{2^m - x_n}{2^m - y_n}$, where $\binom{x_n}{y_n}$ are the points of the first one). Indeed, there are exactly 2^m free orbits that occupy $T \cdot 2^m = 2^{2m} - 2^{2m-2}$ points. On the other hand, there is the method to obtain two symmetrical orbits having any odd norm. We note that possible other symmetries (e.g. symmetries considered in Appendix E) preserve the norm modulo 256. Moreover, in most cases they preserve the norm modulo 2^{m-1} or modulo 2^{m-2}.

2. Correlations between orbits

Consider a pair of free orbits with norms N_1 and N_2. The set $A = \{1, 3, \ldots, 2^m - 1\}$ is a group under multiplication (it is called modulo multiplication group), hence there exists $t \in A$ such that $N_1 \equiv tN_2 \pmod{2^m}$. It is known that for the equation $k^2 \equiv t \pmod{2^m}$ to have a solution $k \in A$ it is necessary and sufficient to have $t \equiv 1 \pmod 8$. Thus, if $N_1 \equiv N_2 \pmod 8$, there exists k such that $N_1 \equiv k^2 N_2 \pmod{2^m}$. If $\binom{x_n}{y_n}$ are the points of the orbit of norm N_2, then $\binom{kx_n}{ky_n} \pmod{\langle 2^m \rangle}$ are the points of the orbit of norm $N_1 \equiv k^2 N_2 \pmod{2^m}$, in the same order. However, there may be large shift between the values of different orbits.

Thus, the case $N_1 \equiv N_2 \pmod 8$ is dangerous, because there may be correlations between orbits. The points of the first orbit are connected to the points of the second orbit with linear congruential dependence. The parameter k may be interpreted as a random odd number.

APPENDIX D: PROOF OF THE THEOREM IN APPENDIX A 3.

Proposition 1. T_n is the least integer such that $\lambda^{T_n} \equiv 1 \pmod{\langle 2^n \rangle}$. In particular, any free orbit has the same period.

Proof. We suppose that the period of an orbit containing the point z is T. Then $\lambda^T z \equiv z \pmod{\langle 2^n \rangle} \Rightarrow z(\lambda^T - 1) \in \langle 2^n \rangle$. If the orbit is free, then $z \notin P$ for any ideal P such that $P | \langle 2^n \rangle$, $P \neq \langle 1 \rangle$. Therefore, $(\lambda^T - 1) \in \langle 2^n \rangle$.

Proposition 2. $T_n | T_{n+1}$.

Proof. Indeed, $\lambda^{T_{n+1}} \equiv 1 \pmod{\langle 2^{n+1} \rangle} \Rightarrow \lambda^{T_{n+1}} \equiv 1 \pmod{\langle 2^n \rangle} \Rightarrow T_{n+1} = mT_n$, where $m \in \mathbb{N}$.

Proposition 3. For all n, either $T_{n+1} = 2T_n$ or $T_{n+1} = T_n$.

Proof. Because $(\lambda^{T_n} - 1) \in \langle 2^n \rangle$, we have $(\lambda^{T_n} + 1) = (\lambda^{T_n} - 1) + 2 \in \langle 2 \rangle$. Consequently, $\lambda^{2T_n} - 1 = (\lambda^{T_n} - 1)(\lambda^{T_n} + 1) \in \langle 2^{n+1} \rangle$, i.e., either $T_{n+1} = 2T_n$ or $T_{n+1} = T_n$.

Proposition 4. If $n \geq 3$ and $T_n \neq T_{n-1}$, then $T_{n+1} \neq T_n$.

Proof. It follows from $T_n = 2T_{n-1}$ that $\begin{cases} \lambda^{T_{n-1}} \equiv 1 \pmod{\langle 2^{n-1} \rangle} \\ \lambda^{T_{n-1}} \not\equiv 1 \pmod{\langle 2^n \rangle} \end{cases} \Rightarrow \lambda^{T_{n-1}} = 1 + z \cdot 2^{n-1}$, where $z \notin \langle 2 \rangle$.

Squaring the last equation, we obtain $\lambda^{2T_{n-1}} = 1 + z \cdot 2^n + z^2 \cdot 2^{2n-2} \equiv 1 + z \cdot 2^n \pmod{\langle 2^{n+1} \rangle}$ for $n \geq 3$. Hence, $\lambda^{2T_{n-1}} \not\equiv 1 \pmod{\langle 2^{n+1} \rangle} \Rightarrow T_{n+1} \neq T_n$.

Proposition 5. If $\langle 2 \rangle$ is split, then for all n, $T'_n = T_n$. In particular, T'_n is the same for all ideal orbits that do not belong to the sublattice $2^{n-1} \times 2^{n-1}$, no matter what the ideal is.

Proof. Because $\langle 2 \rangle$ is split, we have $\langle 2 \rangle = P_1 P_2$. Let T and S be the smallest integers such that $\lambda^T \equiv 1 \pmod{P_1^n}$ and $\lambda^S \equiv 1 \pmod{P_2^n}$. We prove that $T = S$. First, we note that P_1 and P_2 are conjugate ideals, i.e., $P_1 = P_2^*$. Assume $T = S + R$ and $R \geq 0$. Taking the conjugate of the congruence $\lambda^S \equiv 1 \pmod{P_2^n}$, we obtain $\lambda^{*S} \equiv 1 \pmod{P_1^n}$, where $\lambda^* = \lambda^{-1}$. Therefore, $\lambda^S \lambda^{*S} \lambda^R \equiv \lambda^T \equiv 1 \pmod{P_1^n} \Rightarrow \lambda^R \equiv 1 \pmod{P_1^n}$, i.e., there exists an integer $l \geq 0$ such that $R = lT$. Because $T = S + lT$, we have $l = 0 \Rightarrow T = S$.

Let z belong to an ideal orbit of length T'_n and $z \in P_2^k$, $z \notin P_2^{k+1}$, where $k \in \{1, 2, \ldots, n\}$. Then T'_n and T_n are the smallest integers such that $\lambda^{T'_n} \equiv 1 \pmod{P_1^n P_2^{n-k}}$ and $\lambda^{T_n} \equiv 1 \pmod{P_1^n P_2^n}$. Therefore, $T'_n | T_n$. On the other hand, $\lambda^{T'_n} \equiv 1 \pmod{P_1^n} \Rightarrow \lambda^{T'_n} \equiv 1 \pmod{P_2^n} \Rightarrow \lambda^{T'_n} \equiv 1 \pmod{P_1^n P_2^n}$, i.e., $T_n | T'_n$. Therefore, $T'_n = T_n$.

Proposition 6. If $\langle 2 \rangle$ is ramified, then for all n, either $T'_n = T_n$ or $T'_n = T_{n-1}$.

Proof. We have $\langle 2 \rangle = P^2$. We consider an orbit belonging to P. We now show that the orbit period is either T_n or T_{n-1}.

$$\begin{cases} \lambda^{T_{n-1}} \equiv 1 \pmod{\langle 2^{n-1} \rangle} \\ \lambda^{T'_n} \equiv 1 \pmod{\langle 2^{n-1} \rangle P} \\ \lambda^{T_n} \equiv 1 \pmod{\langle 2^n \rangle} \end{cases} \Rightarrow T_{n-1} | T'_n | T_n.$$

Using Proposition 3, we complete the proof.

Proposition 7. If $\langle 2 \rangle$ is ramified, $n \geq 3$, and $T_n = 2T_{n-1}$, then $T'_{n+1} = 2T'_n$.

Proof. Let $T = T_{n-1}$. Then we have

$$\begin{cases} \lambda^T \equiv 1 \pmod{A} \\ \lambda^T \not\equiv 1 \pmod{AP} \end{cases}, \tag{D1}$$

where $A = \langle 2^{n-1} \rangle$ for $T'_n = T_n$ and $A = \langle 2^{n-1} \rangle P$ for $T'_n = T_{n-1}$. In any case, $\langle 2^{n-1} \rangle | A$, $AP | \langle 2^n \rangle$. It follows from (D1) that $\lambda^T = 1 + z$, where $z \in A$, $z \notin AP$. Hence, $\lambda^{2T} = 1 + 2z + z^2$. We note that $2z \in (\langle 2 \rangle A)$, $2z \notin (\langle 2 \rangle AP)$, and $z^2 \in \langle 2^{n+1} \rangle \Rightarrow z^2 \in (\langle 2 \rangle AP)$ for $n \geq 3$. Therefore,

$$\begin{cases} \lambda^{2T} \equiv 1 \pmod{\langle 2 \rangle A} \\ \lambda^{2T} \not\equiv 1 \pmod{\langle 2 \rangle AP} \end{cases}. \tag{D2}$$

If $T'_n = T_n$, this means that $T'_{n+1} \neq T_n \Rightarrow T'_{n+1} = T_{n+1}$. In the case where $T'_n = T_{n-1}$, we have $T'_{n+1} = T_n$. In any case, $T'_{n+1} = 2T'_n$.

APPENDIX E: PROOF OF PROPOSITIONS IN SEC. IV

We here prove the first two of the propositions in Sec. IV. The detailed proof of the last one can be found elsewhere [46]. Here, we follow the notation in Sec. IV. The construction of the subsequences is also presented in Sec. IV.

Proposition 1. Every subsequence of length two has the same probability $1/4$.

Proof. Let $X = (0, \frac{1}{2}] \times (0, 1]$ and $Y = (\frac{1}{2}, 1] \times (0, 1]$, i.e., X and Y are the left and the right halves of the torus. Let R be the cat map (the action of R is defined as transforming the phase space with matrix (1) and taking the fractional parts of both coordinates in $(0, 1)$). We consider the set $R^{-1}(X)$. Figure 4 (left) illustrates the structure of this set. We show that this set is similar for any matrix $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. Indeed, the frontier of the set X passes into the frontier of the set $R^{-1}(X)$ with the transformation R^{-1}. We have

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} 0 \\ y \end{pmatrix} \equiv \begin{pmatrix} 1/2 \\ t \end{pmatrix} \pmod 1, \text{ where } 0 \leqslant t \leqslant 1 \quad \Leftrightarrow \quad y \in \left\{ \frac{1}{2b}, \frac{3}{2b}, \ldots, \frac{2b-1}{2b} \right\} \tag{E1}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} 0 \\ y \end{pmatrix} \equiv \begin{pmatrix} 0 \\ t \end{pmatrix} \pmod 1, \text{ where } 0 \leqslant t \leqslant 1 \quad \Leftrightarrow \quad y \in \left\{ \frac{2}{2b}, \frac{4}{2b}, \ldots, \frac{2b}{2b} \right\} \tag{E2}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ 0 \end{pmatrix} \equiv \begin{pmatrix} 1/2 \\ t \end{pmatrix} \pmod 1, \text{ where } 0 \leqslant t \leqslant 1 \quad \Leftrightarrow \quad x \in \left\{ \frac{1}{2a}, \frac{3}{2a}, \ldots, \frac{2a-1}{2a} \right\} \tag{E3}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ 0 \end{pmatrix} \equiv \begin{pmatrix} 0 \\ t \end{pmatrix} \pmod 1, \text{ where } 0 \leqslant t \leqslant 1 \quad \Leftrightarrow \quad x \in \left\{ \frac{2}{2a}, \frac{4}{2a}, \ldots, \frac{2a}{2a} \right\}. \tag{E4}$$

The lines of the slope $-a/b$ pass into vertical lines with the transformation M. Therefore, the slope of the lines in Fig. 4 is $-a/b$. If a/b is negative, then all pictures are slightly different, but the analogous argumentation holds.

The number of stripes in $R^{-1}(X) \cap X$ is $(a + 2b + 1)/2$ for odd a and $(a + 2b)/2$ for even a. If a is even, then there is a symmetry between $R^{-1}(X) \cap X$ and $R^{-1}(Y) \cap X$. This symmetry is the 180-degree turn with respect to the point $\begin{pmatrix} 1/4 \\ 1/2 \end{pmatrix}$. In this case, $S(R^{-1}(X) \cap X) = S(R^{-1}(Y) \cap X) = 1/4$, and the proposition is proved.

We now show that the areas are also equal for odd a. Let $a \leq 2b$ (the analogous argument holds for $a > 2b$). Let S_i and S_e be the areas of stripes in Fig. 4 (right). Then $S_0 = 1/(8ab)$ is the area of the very first "stripe", which has a triangular form. Obviously, $S_i = (2i + 1)S_0$ and $S_e = 1/(4b) = 2aS_0$. Therefore, the set $R^{-1}(X) \cap X$ occupies the area $S = 2(S_0 + S_2 + \ldots + S_{a-1}) + \frac{2b-a-1}{2}S_e = 2S_0 \sum_{i=0}^{(a-1)/2}(4i+1) + (2b - a - 1)aS_0 = 2baS_0 = \frac{1}{4}$.

Proposition 2. Every subsequence of length three has the same probability $1/8$.

Proof. Let $P(000) = \alpha$ and $P(001) = \beta$. Then $\alpha + \beta = P(00) = 1/4$. Consequently, $P(011) = 1/4 - P(001) = \alpha$ and $P(111) = 1/4 - P(011) = \beta$. On the other hand, $P(111) = P(000) = \alpha$ because $R^{-2}(Y) \cap R^{-1}(Y) \cap Y$ passes into $R^{-2}(X) \cap R^{-1}(X) \cap X$ with the 180-degree rotation with respect to the point $\begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$. Therefore, $\alpha = \beta = 1/8$.

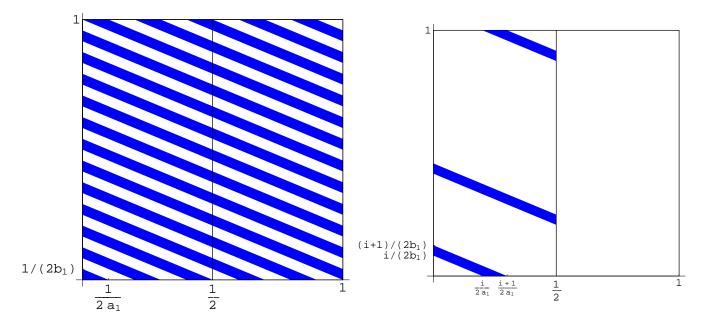**Proposition 3.** If k is odd, then every subsequence of length four has the same probability $1/16$.

FIG. 4: Left: The set $R^{-1}(X)$ for the matrix $M = \begin{pmatrix} 5 & 12 \\ 2 & 5 \end{pmatrix}$. This set is similar for arbitrary cat map with positive entries. The torus is divided into two halves for convenience. Right: Bottom: the stripe S_i. Top: the symmetrical stripe of the same area. Middle: the stripe S_e.

Proof. Let $M = \begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix}$, $M^2 = \begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix}$, and $M^3 = \begin{pmatrix} a_3 & b_3 \\ c_3 & d_3 \end{pmatrix}$. Here $k = \text{Tr}M = a_1 + d_1$.

Let $P(0000) = \alpha$ and $P(0001) = \beta$. Then $\alpha + \beta = P(000) = 1/8$. Because $M \in SL_2(\mathbb{Z})$, we have $a_2 = ka_1 - 1$ and $a_3 = ka_2 - a_1 = a_1(k^2 - 1) - k$. Hence, there are two possibilities.

1. If a_1 is even, then a_2 and a_3 are odd. Taking the 180-degree rotations with respect to the points $\begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$, $\begin{pmatrix} 1/4 \\ 1/2 \end{pmatrix}$, and Proposition 2 into account, we have $P(0100) = P(0000) = P(1111) = P(1011) = \alpha$, $P(0101) = P(0001) = P(1110) = P(1010) = \beta$, $P(1010) = P(1000) = 1/8 - P(0000) = \beta$, $P(0110) = P(0010) = 1/8 - P(1010) = \alpha$. Therefore, $1/4 = P(00) = P(0000) + P(0010) + P(0100) + P(0110) = 4\alpha$, i.e., $\alpha = \beta = 1/16$.

2. If a_1 is odd, then a_2 is even and a_3 is odd. Taking the 180-degree rotations with respect to the points $\begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$, $\begin{pmatrix} 1/4 \\ 1/2 \end{pmatrix}$ and Proposition 2 into account, we have $P(0010) = P(0000) = P(1111) = P(1101) = \alpha$, $P(0011) = P(0001) = P(1110) = P(1100) = \beta$, $P(0100) = 1/8 - P(1100) = \alpha$, $P(0110) = 1/8 - P(1110) = \alpha$. Therefore, $1/4 = P(00) = P(0000) + P(0010) + P(0100) + P(0110) = 4\alpha$, i.e., $\alpha = \beta = 1/16$.

The 180-degree rotation with respect to the point $\begin{pmatrix} 0 \\ 1/4 \end{pmatrix}$ can also be useful in the analysis (the result of this transformation depends on whether b_i are odd or even), although it was not used here.

APPENDIX F: REALIZATIONS AND ALGORITHMS

1. RNG realizations in C language and in inline assembler, speed of realizations

In this section, we present efficient algorithms for several versions of the RNG introduced in Sec. II. In particular, GS (cat map Generator, Simple version), GR (cat map Generator, with Rotation), GRI (cat map Generator, with Rotation, with Increased trace), GM (cat map Generator, Modified version). The parameters and characteristics for these generators can be found in Table VII, the results of stringent statistical tests – in Section III. For comparison, both in Table VII and in Section III B, we also test the standard UNIX generators `rand()`, `rand48()`, `random()`, and modern generators MT19937 [14] and LFSR113 [17] (see Section III B for details on them).

Most of our generators are speeded up with the help of Eq. (5). Also, the Streaming SIMD Extensions 2 (SSE2) technology, introduced in Intel Pentium 4 processors [47], allows to use 128-bit XMM-registers to accelerate compu-

TABLE VII: Characteristics and parameters for several versions of the RNG based on the ensemble of cat maps (see Sec. II) and for other generators (last five entries). Here "CPU-time" means the CPU time (in seconds) needed to generate 10^8 uniform random numbers on a 3.0 GHz Pentium 4 PC running Linux. This parameter characterizes the speed of the generator. The effective periods for generators GM19, GM19-SSE and GM31-SSE are 32 times smaller than the formal periods, presented in this Table.

Generator	g	s	k	q	Rotation	SSE2	Period	CPU-time
GS	2^{32}	32	3	1	−	−	$3.2 \cdot 10^9$	55.4
GS-SSE	2^{32}	32	3	1	−	+	$3.2 \cdot 10^9$	2.49
GR-SSE	2^{32}	32	3	1	+	+	$3.2 \cdot 10^9$	2.79
GSI-SSE	2^{32}	32	11	1	−	+	$3.2 \cdot 10^9$	3.66
GRI	2^{32}	32	11	1	+	−	$3.2 \cdot 10^9$	78.2
GRI-SSE	2^{32}	32	11	1	+	+	$3.2 \cdot 10^9$	4.03
GM19	$2^{19}-1$	32	6	3	+	−	$2.7 \cdot 10^{11}$	120.5
GM19-SSE	$2^{19}-1$	32	6	3	+	+	$2.7 \cdot 10^{11}$	6.11
GM31-SSE	$2^{31}-1$	32	7	11	+	+	$4.6 \cdot 10^{18}$	8.86
RAND	−	−	−	−	−	−	$2.1 \cdot 10^9$	2.48
RAND48	−	−	−	−	−	−	$2.8 \cdot 10^{14}$	4.64
RANDOM	−	−	−	−	−	−	$3.4 \cdot 10^{10}$	1.88
MT19937	−	−	−	−	−	−	$4.3 \cdot 10^{6001}$	2.45
LFSR113	−	−	−	−	−	−	$1.0 \cdot 10^{34}$	2.98

tations. The similar technique was used before for other generators [48]. The SSE2 algorithms for our generator are able to increase performance up to 23 times as compared with usual algorithms (see Table VII).

The algorithms for GS, GRI and GM19 are shown in Table VIII. Table IX illustrates the key ideas for the speed up of cat-map algorithms by means of SSE2. We use the GCC inline assembler syntax for the SSE2 algorithms. The action of the fast SSE2 algorithms shown in the left column are equivalent to the action of the slow algorithms shown in the right column.

The complete realizations for all RNGs can be found in [49]. We note that GM31-SSE is the only algorithm here that exploits 64-bit SSE-arithmetic for calculating Eq. (5).

2. Initialization of generators

The proper initialization is very important for a good generator.

For generators GS, GS-SSE, GR-SSE, GRI, GSI-SSE and GRI-SSE we use the following method of initialization.

- Norms of all points should be different modulo 256. In particular, this guarantees that the initial points $\binom{x_i^{(0)}}{y_i^{(0)}}$, $i = 0, 1, \ldots, (s-1)$ belong to different orbits of the cat map, and that none of the symmetries may convert one orbit to another (see Appendix C).

- At least one point should belong to a free orbit, i.e. at least one of the coordinates x or y should be an odd number. This guarantees that the period length is not smaller than T_m (see Appendix B).

We choose the parameters k and q for generators GM19 and GM31 such that the polynomial $f(x) = x^2 - kx + q$ is primitive modulo p, where $p = 2^{19} - 1$ for GM19 and $p = 2^{31} - 1$ for GM31. Therefore, the actual period of the generator is $p^2 - 1$.

In order to construct the initialization method for GM19 and GM31, we use the "jumping ahead" property, the possibility to skip over terms of the generator. In other words, we utilize an easy algorithm to calculate x_n quickly from x_0 and x_1, for any large n. We choose the following initial conditions: $x_i^{(0)} = x_{i*A}$, $x_i^{(1)} = x_{i*A+1}, i = 0, 1, \ldots, 31$. Here A is the value of the order of $(p^2 - 1)/32$. We recommend to choose A randomly; at least A should not be chosen very close to the divisor of $p^2 - 1$ or to the large power of 2. The meaning of A is the effective period of the generator. For GM19 and GM31 the effective periods are approximately 32 times smaller than the periods in Table VII.

TABLE VIII: Codes in ANSI C language for generators GS, GRI and GM19.

```
const unsigned long halfg=2147483648;
unsigned long x[32],y[32]; char rotate;

//----------- Generator GS ----------------

unsigned long GS(){
  unsigned long i,output=0,bit=1;
  for(i=0;i<32;i++){
    x[i]=x[i]+y[i];
    y[i]=x[i]+y[i];
  }
  for(i=0;i<32;i++){
    output+=((x[i]<halfg)?0:bit); bit*=2;}
  return output;
}

//----------- Generator GRI ----------------

unsigned long GRI(){
  unsigned long i,oldx,oldy,output=0,bit=1;
  oldx=x[31]; oldy=y[31];
  for(i=31;i>0;i--){
    x[i]=4*x[i-1]+9*y[i-1];
    y[i]=3*x[i-1]+7*y[i-1];
  };
  x[0]=4*oldx+9*oldy; y[0]=3*oldx+7*oldy;
  for(i=0;i<32;i++){
    output+=((x[i]<halfg)?0:bit); bit*=2;}
  rotate++; return output;
}
```

```
//------------------ Generator GM19 ---------------------

const unsigned long k=14;
const unsigned long q=15;
const unsigned long g=524287;
const unsigned long qg=7864305;
const unsigned long halfg=262143;
unsigned long x[2][32]; char new,rotate;

unsigned long GM19(){
  unsigned long i,output=0,bit=1;
  char old=1-new;
  for(i=0;i<32;i++)
    x[old][i]=(qg+k*x[new][i]-q*x[old][i])%g;
  for(i=0;i<32;i++){
    output+=((state->x[old][(256+i-rotate)%32]<halfg)?0:bit);
    bit*=2;
  }
  new=old; rotate++; return output;
}
```

The initialization routines for all generators can also be found in [49].

[1] K.S.D. Beach, P.A. Lee, P. Monthoux, Phys. Rev. Lett. **92** (2004) 026401.
[2] D.P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics* (Cambridge University Press, Cambridge, 2000).
[3] S.C. Pieper and R.B. Wiring, Ann. Rev. Nucl. Part. Sci., **51** (2001) 53.
[4] A. Lüchow, Ann. Rev. Phys. Chem., **51** (2000) 501.
[5] A.R. Bizzarri, J. Phys.: Cond. Mat. **16** (2004) R83.
[6] D.E. Knuth, *The art of Computer Programming*, Vol. 2, (Addison-Wesley, Cambridge, 1981).
[7] P. L'Ecuyer, Ann. of Oper. Res., **53** (1994), 77.
[8] R.P. Brent and P. Zimmermann, *Random Number Generators with period divisible by a Mersenne prime* in *Lecture Notes in Computer Science*, Comp. Sc. and its Appl. - ICCSA 2003, **2667** (Springer-Verlag, Berlin, 2003), 1.
[9] R.R. Coveyou and R.D. MacPherson, J. ACM **14** (1967) 100; G. Marsaglia, Proc. Nat. Acad. Sci. USA, **61** (1968) 25.
[10] S.W. Golomb, *Shift Register Sequences*, (Holden-Day, San Francisco, 1967).
[11] A.M. Ferrenberg, D.P. Landau, Y.J. Wong, Phys. Rev. Lett., **69** (1992) 3382.
[12] P. Grassberger, Phys.Lett. 181 (1993) 43.
[13] F. Schmid, N. B. Wilding, Int.J.Mod.Phys. C 6 (1995) 781.
[14] M. Matsumoto and T. Nishimura, ACM Trans. on Mod. and Comp. Sim., **8(1)** (1998) 3.
[15] P. L'Ecuyer, Oper. Res., **47(1)** (1999) 159.
[16] P. L'Ecuyer, Math. of Comp., **65(213)** (1996), 203.
[17] P. L'Ecuyer, Math. of Comp., **68(225)** (1999), 261.
[18] L. Blum, M. Blum, M. Shub, SIAM J. of Comp., **15** (1986) 364.
[19] T. Moreau, http://www.connotech.com/bbsindex.htm (1996).
[20] A.J. Lichtenberg, M.A. Lieberman, *Regular and Stochastic Motion*, (Springer-Verlag, New York, 1983).
[21] H. G. Schuster, *Deterministic Chaos, An Introduction*, (Physik Verlag, Weinheim, 1984).

TABLE IX: Equivalent realizations for several algorithms with inline assembler code for Pentium 4 processor (left column) and ANSI C language (right column). First row presents the main part of the GRI algorithm. Second row presents the packing 16 high bits of 16 integers into one integer. These or similar equivalences are used in constructing the SSE2-algorithms for any of the discussed RNGs [49].

```
unsigned long x[4],y[4];

[.......]

asm("movaps (%0),%%xmm0\n" \
    "movaps (%1),%%xmm1\n" \
    "paddd  %%xmm1,%%xmm0\n" \
    "paddd  %%xmm1,%%xmm0\n" \
    "movaps %%xmm0,%%xmm2\n" \
    "pslld  $2,%%xmm0\n" \
    "paddd  %%xmm1,%%xmm0\n" \
    "movaps %%xmm0,(%0)\n" \
    "psubd  %%xmm2,%%xmm0\n" \
    "movaps %%xmm0,(%1)\n" \
    ""::"r"(x),"r"(y));
```

```
unsigned long i,newx[4],x[4],y[4];

[.......]

for(i=0;i<4;i++){
  newx[i]=4*x[i]+9*y[i];
  y[i]=3*x[i]+7*y[i];
  x[i]=newx[i];
}
```

```
unsigned long x[16],output;

[.......]

asm("movaps (%1),%%xmm0\n" \
    "movaps 16(%1),%%xmm1\n" \
    "movaps 32(%1),%%xmm2\n" \
    "movaps 48(%1),%%xmm3\n" \
    "psrld  $31,%%xmm0\n" \
    "psrld  $31,%%xmm1\n" \
    "psrld  $31,%%xmm2\n" \
    "psrld  $31,%%xmm3\n" \
    "packssdw %%xmm1,%%xmm0\n" \
    "packssdw %%xmm3,%%xmm2\n" \
    "packsswb %%xmm2,%%xmm0\n" \
    "psllw  $7,%%xmm0\n" \
    "pmovmskb %%xmm0,%0\n" \
    ""::"=r"(output):"r"(x));
```

```
const unsigned long halfg=2147483648;
unsigned long x[16],i,output=0,bit=1;

  [.......]

for(i=0;i<16;i++){
  output+=((x[i]<halfg)?0:bit;
  bit*=2;
}
```

[22] V.I. Arnol'd, A. Avez, *Ergodic Problems of Classical Mechanics*, (Nenjamin, New York, 1968).
[23] J.P. Keating, Nonlinearity, **4** (1991) 277.
[24] H. Grothe, Statistische Hefte, **28** (1987), 233.
[25] H. Niederreiter, Math.Japonica, **31(5)** (1986), 759.
[26] L. Afferbach, H. Grothe, J. of Comput. and Applied Math., **23** (1988), 127.
[27] H. Niederreiter, J. of Comput. and Applied Math., **31** (1990), 139.
[28] P. L'Ecuyer, P. Hellekalek, *Random Number Generators: Selection Criteria and Testing* in *Random and Quasi-Random Point Sets*, number 138 in Lectures Notes In Statistics (Springer, 1998), 223.
[29] I.C. Percival, F. Vivaldi, Physica D, **25** (1987) 105.
[30] P. L'ecuyer, Comm. of the ACM, **33(10)** (1990), 85.
[31] L.N. Shchur and P. Butera, Int. J. Mod. Phys. C **9** (1998) 607.
[32] H. Niederreiter, *New developments in uniform pseudorandom number and vector generation* in *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, ed. H. Niederreiter and P. J.-S. Shiue, number 106 in Lecture Notes in Statistics, (Springer-Verlag, 1995), 87.
[33] H. Niederreiter, Annals of Operations Research, **31** (1991), 323.
[34] P. L'Ecuyer, R. Simard, TestU01: A Software Library in ANSI C for Empirical Testing of Random Number Generators (2002). Software user's guide. Available at http://www.iro.umontreal.ca/∼lecuyer.
[35] G. Marsaglia, Die Hard: A battery of tests for random number generators: http://stat.fsu.edu/pub/diehard.
[36] A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications, http://csrc.nist.gov/rng/SP800-22b.pdf.
[37] V.I. Arnol'd, *Mathematical Methods of Classical Mechanics*, (Springer, New York, 1978).

[38] J.H. Hannay, M.V. Berry, Physica D **1** (1980) 267.

[39] A. Bonelli, S. Ruffo, Int. J. Mod. Phys. C, **9** (1998) 987.

[40] W. Selke, A.L. Talapov and L.N. Shchur, Pis'ma ZhETF, 58 (1993) 684; JETP Lett., 58 (1993) 665; I. Vattulainen, T. Ala-Nissila, K. Kankaala, Phys. Rev. Lett., **73** (1994) 2513; F. Schmid, N.B. Wilding, Int. J. Mod. Phys. C **6** (1995) 781.

[41] L.N. Shchur, J.R. Heringa, H.W.J. Blöte, Physica A **241** (1997) 579.

[42] L.N. Shchur, H.W.J. Blöte, Phys. Rev. E **55** (1997) R4905.

[43] K. Binder, D.W. Heermann, *Monte Carlo Simulation in Statistical Physics*, (Springer-Verlag, Berlin, 1992).

[44] H. Cohn, *A Second Course in Number Theory*, (Wiley, New York, 1962); reprinted with the title *Advanced Number Theory* (Dover, New York, 1980).

[45] R. Chapman, *Notes on Algebraic Numbers*, http://www.maths.ex.ac.uk/∼rjc/notes/algn.ps (1995, 2002).

[46] The detailed proof of the third proposition can be found at http://www.comphys.ru/barash/cat-map-details.ps.

[47] http://developer.intel.com/design/pentium4/manuals/index_new.htm.

[48] L.N. Shchur and T.A. Rostunov, JETP Lett. **76** (2002) 475.

[49] The complete gcc-compatible algorithms for generators in Table III and Table VII and the detailed results for the batteries of tests can be found at http://www.comphys.ru/barash/cat-map-algorithms.zip.