Published in *Industrial mathematics and complex systems,*
Emerging Mathematical models, Methods and Algorithms
P. Manchanda, R. Lozi, A.-H. Siddiqi (eds.)
Springer Nature Singapore, 2017, pp. 43-76.
Industrial and Applied Mathematics
DOI 10.1007/978-981-10-3758-0_4

The Challenging Problem of Industrial Applications of Multicore-Generated Iterates of Nonlinear Mappings

Jean-Pierre Lozi, Oleg Garasym and Ren®Lozi

Abstract The study of nonlinear dynamics is relatively recent with respect to the long historical development of early mathematics since the Egyptian and the Greek civilization, even if one includes in this field of research the pioneer works of Gaston Julia and Pierre Fatou related to one-dimensional maps with a complex variable, nearly a century ago. In France, Igor Gumosky and Christian Mira began their mathematical researches in 1958; in Japan, the Hayashi' School (with disciples such as Yoshisuke Ueda and Hiroshi Kawakami), a few years later, was motivated by applications to electric and electronic circuits. In Ukraine, Alexander Sharkovsky found the intriguing Sharkovsky's order, giving the periods of periodic orbits of such nonlinear maps in 1962, although these results were only published in 1964. In 1983, Leon O. Chua invented a famous electronic circuit that generates chaos, built with only two capacitors, one inductor and one nonlinear negative resistance. Since then, thousands of papers have been published on the general topic of chaos. However, the pace of mathematics is slow, because any progress is based on strictly rigorous proof. Therefore, numerous problems still remain unsolved. For example, the long-term dynamics of the Hénon map, the first example of a strange attractor for mappings, remain unknown close to the classical parameter values from a strictly mathematical point of view, 40 years after its original publication. In spite of this lack of rigorous mathematical proofs, nowadays, engineers are actively working on applications of chaos for several purposes: global optimization, genetic algorithms, CPRNG (Chaotic Pseudorandom Number Generators), cryptography, and so on. They use nonlinear maps for practical applications without the need of sophisticated theorems. In this chapter, after giving some prototypical examples of the industrial

J.-P. Lozi
I3S laboratory, UMR 7271, Université Côte d'Azur, CNRS, Euclide B,
Les Algorithmes, 2000 Route des Lucioles, 06900 Sophia Antipolis, France

O. Garasym
SOC, IBM, Wroclaw, Poland

R. Lozi
J. A. Dieudonné laboratory, UMR 7351, Université Côte d'Azur, CNRS,
28 Avenue Valrose, 06108 Nice Cedex 02, France
e-mail: rlozi@unice.fr ,

applications of iterations of nonlinear maps, we focus on the exploration of topologies of coupled nonlinear maps that have a very rich potential of complex behavior. Very long computations on modern multicore machines are used: they generate up to one hundred trillion iterates in order to assess such topologies. We show the emergence of randomness from chaos and discuss the promising future of chaos theory for cryptographic security.

Keywords Chaos · Cryptography · Mappings · Chaotic pseudorandom numbers Attractors

AMS Subject Classification 37N30 · 37D45 · 65C10 · 94A60

Chapter 4

4.1 Introduction

The last few decades have seen the tremendous development of new IT technologies that incessantly increase the need for new and more secure cryptosystems.

For instance, the recently invented Bitcoin cryptocurrency is based on the secure Blockchain system that involves hash functions [1]. This technology, used for information encryption, is pushing forward the demand for more efficient and secure pseudorandom number generators [2] which, in the scope of chaos-based cryptography, were first introduced by Matthews in the 1990s [3]. Contrarily to most algorithms that are used nowadays and based on a limited number of arithmetic or algebraic methods (like elliptic curves), networks of coupled chaotic maps offer quasi-infinite possibilities to generate parallel streams of pseudorandom numbers (PRN) at a rapid pace when they are executed on modern multicore processors. Chaotic maps are able to generate independent and secure pseudorandom sequences (used as information carriers or directly involved in the process of encryption/decryption [4]). However, the majority of well-known chaotic maps are not naturally suitable for encryption [5] and most of them do not exhibit even satisfactory properties for such a purpose.

In this chapter, we explore the novel idea of coupling a symmetric tent map with a logistic map, following several network topologies. We add a specific injection mechanism to capture the escaping orbits. In the goal of extending our results to industrial mathematics, we implement these networks on multicore machines and we test up to 100 trillion iterates of such mappings, in order to make sure that the obtained results are firmly grounded and able to be used in industrial contexts such as e-banking, e-purchasing, or the Internet of Things (IoT).

The chaotic maps, when used in the sterling way, could generate not only chaotic numbers, but also pseudorandom numbers as shown in [6] and as we show in this chapter with more sophisticated numerical experiments.

Various choices of PNR Generators (PRNGs) and crypto-algorithms are currently necessary to implement continuous, reliable security systems. We use a software approach because it is easy to change a cryptosystem to support protection, whereas

replacing hardware used for True Random Number Generators would be costly and time-consuming. For instance, after the secure software protocol Wi-Fi Protected Access (WPA) was broken, it was simply updated and no expensive hardware had to be replaced.

It is a very challenging task to design CPRNGs (Chaotic Pseudo Random Number Generators) that are applicable to cryptography: numerous numerical tests must ensure that their properties are satisfactory. We mainly focus on two- to five-dimension maps, although upper dimensions can be very easily explored with modern multicore machines. Nevertheless, in four and five dimensions, the studied CRPNGs are efficient enough for cryptography.

In Sect. 4.2, we briefly recall the dawn and the maturity of researches on chaos. In Sect. 4.3, we explore two-dimensional topologies of networks of coupled chaotic maps. In Sect. 4.4, we study more thoroughly a mapping in higher dimensions (up to 5) far beyond the NIST tests which are limited to a few millions of iterates and which seem not robust enough for industrial applications, although they are routinely used worldwide. In order to check the portability of the computations on multicore architectures, we have implemented all our numerical experiments on several different multicore machines. We conclude this chapter in Sect. 4.5.

4.2 The Dawn and the Maturity of Researches on Chaos

The study of nonlinear dynamics is relatively recent with respect to the long historical development of early mathematics since the Egyptian and the Greek civilizations (and even before). The first alleged artifact of mankind's mathematical thinking goes back to the Upper Paleolithic era. Dating as far back as 22,000 years ago, the Ishango bone is a dark brown bone which happens to be the fibula of a baboon, with a sharp piece of quartz affixed to one end for engraving. It was first thought to be a tally stick, as it has a series of what has been interpreted as tally marks carved in three columns running the length of the tool [7].

Twenty thousand years later, the Rhind Mathematical Papyrus is the best example of Egyptian mathematics. It dates back to around 1650 BC. Its author is the scribe Ahmes who indicated that he copied it from an earlier document dating from the 12th dynasty, around 1800 BC. It is a practical handbook, whose the first part consists of reference tables and a collection of 20 arithmetic and 20 algebraic problems and linear equations. Problem 32 for instance corresponds (in modern notation) to solving $x + \frac{x}{3} + \frac{x}{4} = 2$ for x [8].

Since those early times, mathematics have known great improvements, flourishing in many different fields such as geometry, algebra (both linked, thanks to the invention of Cartesian coordinates by René Descartes [9]), analysis, probability, number and set theory, and so on.

However, nonlinear problems are very difficult to handle, because, as shown by Galois' theory of algebraic equations which provides a connection between field theory and group theory, it is impossible to solve any polynomial equation

of degree equal or greater than 5 using only the usual algebraic operations (addition, subtraction, multiplication, division) and the application of radicals (square roots, cube roots, etc.) [10].

The beginning of the study of nonlinear equation systems goes back to the original works of Gaston Julia and Pierre Fatou regarding to one-dimensional maps with a complex variable, nearly a century ago [11, 12]. Compared to thousands of years of mathematical development, a century is a very short period. In France, 30 years later, Igor Gumosky and Christian Mira began their mathematical researches with the help of a computer in 1958 [13]. They developed very elaborate studies of iterations. One of the best-known formulas they published is

$$\begin{cases} x_{n+1} = f(x_n) + by_n \\ y_{n+1} = f(x_{n+1}) - x_n, \end{cases} \quad \text{with} \quad f(x) = ax + 2(1-a)\frac{x^2}{1+x^2} \tag{4.1}$$

which can be considered as a non-autonomous mapping from the plane $\mathbb{R}^2$ onto itself that exhibits esthetic chaos. Surprisingly, slight variations of the parameter value lead to very different shapes of the attractor (Fig. 4.1).

In Ukraine, Alexander Sharkovsky found the intriguing Sharkovsky's order, giving the periods of periodic orbits of such nonlinear maps in 1962, although these results were only published in 1964 [14]. In Japan the Hayashi' School (with disciples like Yoshisuke Ueda and Hiroshi Kawakami), a few years later, was motivated by applications to electric and electronic circuits. Ikeda proposed the Ikeda attractor [15, 16] which is a chaotic attractor for $u \geq 0.6$ (Fig. 4.2).

$$\begin{cases} x_{n+1} = 1 + u(x_n \cos t_n - y_n \sin t_n) \\ y_{n+1} = u(x_n \sin t_n + y_n \cos t_n), \end{cases} \quad \text{with} \quad t_n = 0.4 - \frac{6}{1 + x_n^2 + y_n^2} \tag{4.2}$$

In 1983, Leon O. Chua invented a famous electronic circuit that generates chaos built with only two capacitors, one inductor and one nonlinear negative resistance [17]. Since then, thousands of papers have been published on the general

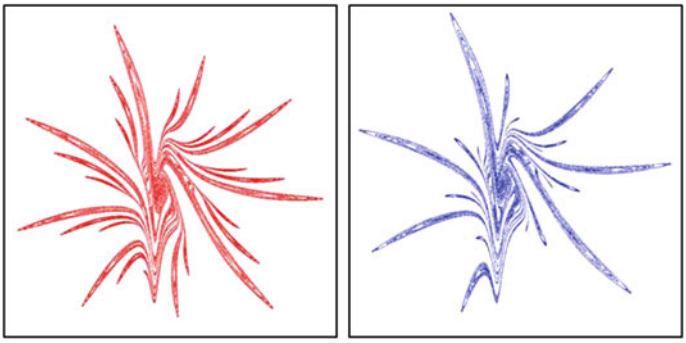

Fig. 4.1 Gumowski-Mira attractor for parameter values $a = 0.92768$ and $a = 0.93333$

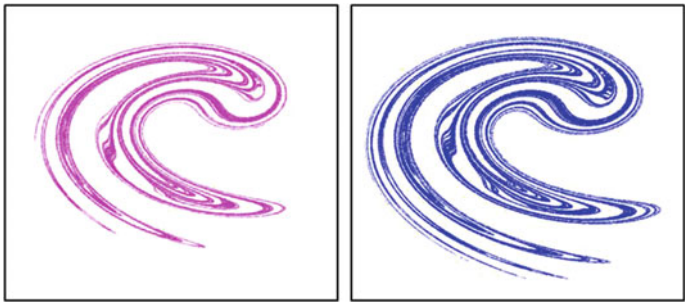

Fig. 4.2 Ikeda attractor for $u = 8.6$ and $u = 8.9$

topic of chaos. However the pace of mathematics is slow, because any progress is based on strictly rigorous proof. Therefore numerous problems still remain unsolved. For example, the long-term dynamics of the Hénon map [18], the first example of a strange attractor for mappings, remains unknown close to the classical parameter values from a strictly mathematical point of view, 40 years after its original publication.

Nevertheless, in spite of this lack of rigorous mathematical results, nowadays, engineers are actively working on applications of chaos for several purposes: global optimization, genetic algorithms, CPRNG, cryptography, and so on. They use nonlinear maps for practical applications without the need of sophisticated theorems. During the last 20 years, several chaotic image encryption methods have been proposed in the literature.

Dynamical systems which present a mixing behavior and that are highly sensitive to initial conditions are called chaotic. Small differences in initial conditions (such as those due to rounding errors in numerical computation) yield widely diverging outcomes for chaotic systems. This effect, popularly known as the butterfly effect, renders long-term predictions impossible in general [19]. This happens even though these systems are deterministic, meaning that their future behavior is fully determined by their initial conditions, with no random elements involved. In other words, the deterministic nature of these systems does not make them predictable. Mastering the global properties of those dynamical systems is a challenging issue nowadays that we try to fix by exploring several network topologies of coupled maps.

In this chapter, after giving some prototypical examples of industrial applications of iterations of nonlinear maps, we focus on the exploration of topologies of coupled nonlinear maps that have a very rich potential of complex behavior. Very long computations on multicore machines are used, generating up to one hundred trillion iterates, in order to assess such topologies. We show the emergence of randomness from chaos and discuss the promising future of chaos theory for cryptographic security.

4.3 Miscellaneous Network Topologies of Coupled Chaotic Maps

4.3.1 Tent-Logistic Entangled Map

In this section we consider only two 1-D maps: the logistic map

$$f_\mu(x) \equiv L_\mu(x) = 1 - \mu x^2 \tag{4.3}$$

and the symmetric tent map

$$f_\mu(x) \equiv T_\mu(x) = 1 - \mu|x| \tag{4.4}$$

both associated to the dynamical system

$$x_{n+1} = f_\mu(x_n), \tag{4.5}$$

where μ is a control parameter which impacts the chaotic degree. Both mappings are sending the one-dimensional interval $[-1,1]$ onto itself.

Since the first study by R. May [20, 21] of the logistic map in the frame of nonlinear dynamical systems, both the logistic (4.3) and the symmetric tent map (4.4) have been fully explored with the aim to easily generate pseudorandom numbers [22].

However, the collapse of iterates of dynamical systems [23] or at least the existence of very short periodic orbits, their non-constant invariant measure, and the easily-recognized shape of the function in the phase space, could lead to avoid the use of such one-dimensional maps (logistic, baker, tent, etc.) or two-dimensional maps (Hénon, Standard, Belykh, etc.) as PRNGs (see [24] for a survey). Yet, the very simple implementation as computer programs of chaotic dynamical systems led some authors to use them as a base for cryptosystems [25, 26]. Even if the logistic and tent maps are topologically conjugates (i.e., they have similar topological properties: distribution, chaoticity, etc.), their numerical behavior differs drastically due to the structure of numbers in computer realization [27].

As said above, both logistic and tent maps are never used in serious cryptography articles because they have weak security properties (collapsing effect) if applied alone. Thus, these maps are often used in modified form to construct CPRNGs [28–30].

Recently, Lozi et al. proposed innovative methods in order to increase randomness properties of the tent and logistic maps over their coupling and sub-sampling [31–33]. Nowadays, hundreds of publications on industrial applications of chaos-based cryptography are available [34–37].

In this chapter, we explore more thoroughly the original idea of combining features of tent (T_μ) and logistic (L_μ) maps to produce a new map with improved properties, through combination in several network topologies. This idea was recently introduced [38, 39] in order to improve previous CPRNGs.

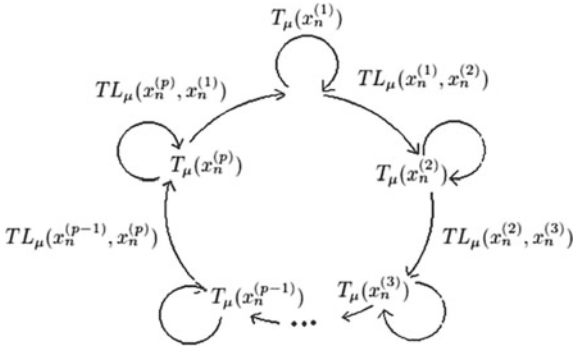

Fig. 4.3 Auto and ring-coupling of the TL_μ and T_μ maps (from [38])

Looking at both Eqs. (4.3) and (4.4), it is possible to reverse the shape of the graph of the tent map T and to entangle it with the graph of the logistic map L. We obtain the combined map

$$f_\mu(x) \equiv TL_\mu(x) = \mu|x| - \mu x^2 = \mu(|x| - x^2) \tag{4.6}$$

When used in more than one dimension, the TL_μ map can be considered as a two-variable map

$$TL_\mu(x^{(i)}, x^{(j)}) = \mu(|x^{(i)}| - (x^{(j)})^2), \quad i \neq j \tag{4.7}$$

Moreover, we can combine again the TL_μ map with T_μ in various ways. If with choose, for instance, a network with a ring shape (Fig. 4.3).

It is possible to define a mapping $M_{\mu,p} : J^p \to J^p$ where $J_p = [-1, 1]^p \subset R^p$:

$$M_{\mu,p} \begin{pmatrix} x_n^{(1)} \\ x_n^{(2)} \\ \cdot \\ \cdot \\ \cdot \\ x_n^{(p)} \end{pmatrix} = \begin{pmatrix} x_{n+1}^{(1)} \\ x_{n+1}^{(2)} \\ \cdot \\ \cdot \\ \cdot \\ x_{n+1}^{(p)} \end{pmatrix} = \begin{pmatrix} T_\mu(x_n^{(1)}) + TL_\mu(x_n^{(1)}, x_n^{(2)}) \\ T_\mu(x_n^{(2)}) + TL_\mu(x_n^{(2)}, x_n^{(3)}) \\ \cdot \\ \cdot \\ \cdot \\ T_\mu(x_n^{(p)}) + TL_\mu(x_n^{(p)}, x_n^{(1)}) \end{pmatrix} \tag{4.8}$$

However, if used in this form, system (4.8) has unstable dynamics and iterated points $x_n^{(1)}, x_n^{(2)}, \ldots, x_n^{(p)}$ quickly spread out. Therefore, to solve the problem of keeping dynamics in the torus $J^p = [-1, 1]^p \subset R^p$, the following injection mechanism has to be used in conjunction with (4.8)

$$\begin{cases} \text{if } (x_{n+1}^{(i)} < -1) \text{ then add } 2 \\ \text{if } (x_{n+1}^{(i)} > 1) \text{ then subtract } 2 \end{cases}, \quad i = 1, 2, \ldots, p. \tag{4.9}$$

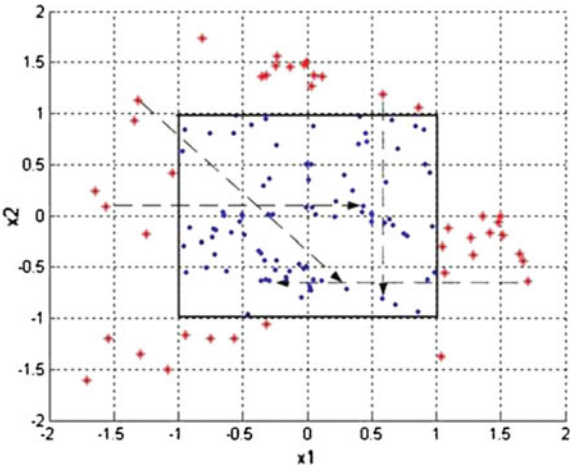

Fig. 4.4 Return mechanism from the $[-2, 2]^p$ torus to $[-1, 1]^p$ (from [38])

Under this injection mechanism, for $1 \leq i \leq p$, points come back from $[-2, 2]^p$ to $[-1, 1]^p$ (Fig. 4.4).

The TL_μ function is a powerful tool to change dynamics. Used in conjunction with T_μ, the map TL_μ makes it possible to establish mutual influence between system components $x_n^{(i)}$ in $M_{\mu,p}$. This multidimensional coupled mapping is interesting because it performs contraction and distance stretching between components, improving chaotic distribution.

The coupling of components has an excellent effect in achieving chaos, because they interact with global system dynamics, being a part of them. Component interaction has a global effect. In order to study this new mapping, we use a graphical approach, however other theoretical assessing functions are also involved.

Note that system (4.8) can be made more generic by introducing constants k^i which generalize considered topologies. Let $\underline{k} = (k^1, k^2, \ldots, k^p)$, we define

$$M_{\mu,p}^{\underline{k}} \begin{pmatrix} x_n^{(1)} \\ x_n^{(2)} \\ \cdot \\ \cdot \\ \cdot \\ x_n^{(p)} \end{pmatrix} = \begin{pmatrix} x_{n+1}^{(1)} \\ x_{n+1}^{(2)} \\ \cdot \\ \cdot \\ \cdot \\ x_{n+1}^{(p)} \end{pmatrix} = \begin{pmatrix} T_\mu(x_n^{(1)}) + k^1 \times TL_\mu(x_n^{(i)}, x_n^{(j)}), & i, j = (1,2) \text{ or } (2,1) \\ T_\mu(x_n^{(2)}) + k^2 \times TL_\mu(x_n^{(i)}, x_n^{(j)}) & i, j = (2,3) \text{ or } (3,2) \\ \cdot \\ \cdot \\ \cdot \\ T_\mu(x_n^{(p)}) + k^p \times TL_\mu(x_n^{(i)}, x_n^{(j)}) & i, j = (p,1) \text{ or } (1,p) \end{pmatrix} \tag{4.10}$$

System (4.10) is called alternate if $k^i = (-1)^i$ or $k^i = (-1)^{i+1}$, $1 \leq i \leq p$, or non-alternate if $k^i = +1$, or $k^i = -1$. It can be a mix of alternate and non-alternate if $k^i = +1$ or -1 randomly.

Table 4.1 The sixteen maps defined by Eq. (4.11)

Case	k^1	k^2	i	j	i'	j'
#1	+1	+1	1	2	1	2
#2	+1	−1	1	2	1	2
#3	−1	+1	1	2	1	2
#4	−1	−1	1	2	1	2
#5	+1	+1	2	1	2	1
#6	+1	−1	2	1	2	1
#7	−1	+1	2	1	2	1
#8	−1	−1	2	1	2	1
#9	+1	+1	1	2	2	1
#10	+1	−1	1	2	2	1
#11	−1	+1	1	2	2	1
#12	−1	−1	1	2	2	1
#13	+1	+1	2	1	1	2
#14	+1	−1	2	1	1	2
#15	−1	+1	2	1	1	2
#16	−1	−1	2	1	1	2

4.3.2 Two-Dimensional Network Topologies

We first consider the simplest coupling case, in which only two equations are coupled. The first condition needed to obtain a multidimensional mapping, in the aim of building a new CPRNG, is to obtain excellent uniform distribution of the iterated points. The second condition is that the CPRNG must be assessed positively by the NIST tests [40]. In [38, 39] this two-dimensional case is studied in detail. Using a bifurcation diagram and computation of Lyapunov exponents, it is shown that the best value for the parameter is $\mu = 2$. Therefore, in the rest of this chapter we use this parameter value and we only briefly recall the results found with this value in both of those articles. The general form of $M_{2,2}^{k}$ is then

$$M_{2,2}^{k}\begin{pmatrix} x_n^{(1)} \\ x_n^{(2)} \end{pmatrix} = \begin{pmatrix} x_{n+1}^{(1)} \\ x_{n+1}^{(2)} \end{pmatrix} = \begin{pmatrix} T_2(x_n^{(1)}) + k^1 \times TL_2(x_n^{(i)}, x_n^{(j)}) \\ T_2(x_n^{(2)}) + k^2 \times TL_2(x_n^{(i')}, x_n^{(j')}) \end{pmatrix} \tag{4.11}$$

with $i,\ j,\ i',\ j' = 1$ or 2, $i \neq j$, and $i' \neq j'$.

Considering this general form, it is possible to define 16 different maps (Table 4.1).

Among this set of maps, we study case #3 and case #13. The map of case #3 is called Single-Coupled alternate due to the shape of the corresponding network and denoted TTL_2^{SC},

$$TTL_2^{SC} = \begin{cases} x_{n+1}^{(1)} = 1 - 2|x_n^{(1)}| - 2(|x_n^{(1)}| - (x_n^{(2)})^2) = T_2(x_n^{(1)}) - TL_2((x_n^{(1)}), (x_n^{(2)})) \\ x_{n+1}^{(2)} = 1 - 2|x_n^{(2)}| + 2(|x_n^{(1)}| - (x_n^{(2)})^2) = T_2(x_n^{(2)}) + TL_2((x_n^{(1)}), (x_n^{(2)})) \end{cases} \tag{4.12}$$

and case #13 is called Ring-Coupled non-alternate and denoted TTL_2^{RC},

$$TTL_2^{RC} = \begin{cases} x_{n+1}^{(1)} = 1 - 2|x_n^{(1)}| + 2(|x_n^{(2)}| - (x_n^{(1)})^2) = T_2(x_n^{(1)}) + TL_2((x_n^{(2)}), (x_n^{(1)})) \\ x_{n+1}^{(2)} = 1 - 2|x_n^{(2)}| + 2(|x_n^{(1)}| - (x_n^{(2)})^2) = T_2(x_n^{(2)}) + TL_2((x_n^{(1)}), (x_n^{(2)})) \end{cases} \tag{4.13}$$

Both systems were selected because they have balanced contraction and stretching processes between components. They allow achieving uniform distribution of the chaotic dynamics. Equations (4.12) and (4.13) are used, of course, in conjunction with injection mechanism (4.9).

The largest torus where points mapped by (4.12) and (4.13) are sent is $[-2, 2]^2$. The confinement from torus $[-2, 2]^2$ to torus $[-1, 1]^2$ of the dynamics obtained by this mechanism is shown in Figs. 4.5 and 4.6: dynamics cross from the negative region (in blue) to the positive one, and conversely to the negative region, if the points stand in the positive regions (in red). Through this operation, the system's dynamics are trapped inside $[-1, 1]^2$. In addition, after this operation is done, the resulting system exhibits more complex dynamics with additional nonlinearity, which is advantageous for chaotic encryption (since it improves security).

A careful distribution analysis of both TTL_2^{SC} and TTL_2^{RC} has been performed using approximated invariant measures.

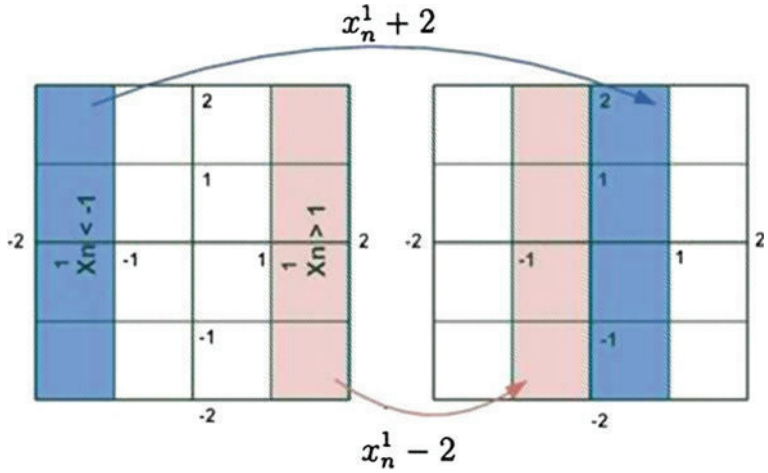

Fig. 4.5 Injection mechanism of the iterates from torus $[-2, 2]^2$ to torus $[-1, 1]^2$. If $x_n^{(1)} > 1$ then $x_n^{(1)} \equiv x_n^{(1)} - 2$; if $x_n^{(1)} < -1$ then $x_n^{(1)} \equiv x_n^{(1)} + 2$ (from [38])

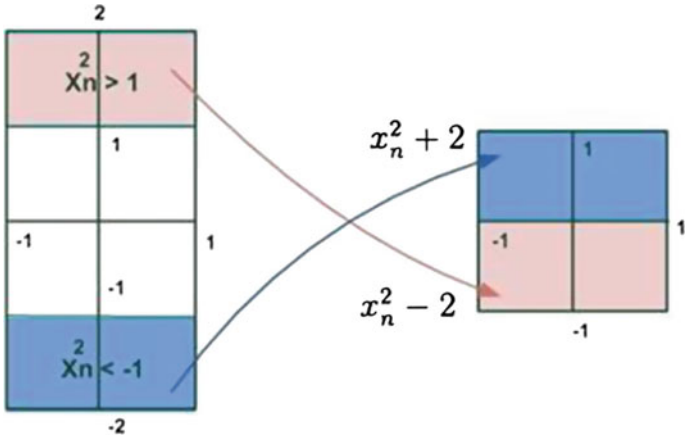

Fig. 4.6 If $x_n^{(2)} > 1$ then $x_n^{(2)} \equiv x_n^{(2)} - 2$; if $x_n^{(2)} < -1$ then $x_n^{(2)} \equiv x_n^{(2)} + 2$ (from [38])

4.3.3 Approximated Invariant Measures

We recall in this section the definition of approximated invariant measures which are important tools for assessing the uniform distribution of iterates. We have previously introduced them for the first studies of the weakly coupled symmetric tent map [22].

We first define an approximation $P_{M,N}(x)$ of the invariant measure, also called the probability distribution function linked to the one-dimensional map f (Eq. (4.5)) when computed with floating numbers (or numbers in double precision). To this goal, we consider a regular partition of M small intervals (boxes) r_i of $J = [-1, 1]$ defined by

$$s_i = -1 + \frac{2i}{M}, \quad i = 0, M, \tag{4.14}$$

$$r_i = [s_i, s_{i+1}[, \quad i = 0, M-2, \tag{4.15}$$

$$r_{M-1} = [s_{M-1}, 1], \tag{4.16}$$

$$J = \bigcup_{0}^{M-1} r_i. \tag{4.17}$$

The length of each box r_i is equal to

$$s_{i+1} - s_i = \frac{2}{M} \tag{4.18}$$

All iterates $f^{(n)}(x)$ belonging to these boxes are collected (after a transient regime of Q iterations decided a priori, i.e., the first Q iterates are discarded). Once the

computation of $N + Q$ iterates is completed, the relative number of iterates with respect to N/M in each box r_i represents the value $P_N(s_i)$. The approximated $P_N(x)$ defined in this article is therefore a step function, with M steps. Since M may vary, we define

$$P_{M,N}(s_i) = \frac{1}{2}\frac{M}{N}(\#r_i) \tag{4.19}$$

where $\#r_i$ is the number of iterates belonging to the interval r_i and the constant $1/2$ allows the normalisation of $P_{M,N}(x)$ on the interval J.

$$P_{M,N}(x) = P_{M,N}(s_i), \quad \forall\, x \in r_i \tag{4.20}$$

In the case of p-coupled maps, we are more interested by the distribution of each component $x^{(1)}, x^{(2)}, \ldots, x^{(p)}$ of the vector $X = \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ . \\ . \\ . \\ x^{(p)} \end{pmatrix}$ rather than by the distribution of the variable X itself in J^p. We then consider the approximated probability distribution function $P_{M,N}(x^{(j)})$ associated to one component of X. In this chapter, we use either N_{disc} for M or N_{iter} for N, depending on which is more explicit. The discrepancies E_1 (in norm L_1), E_2 (in norm L_2), and E_∞ (in norm L_∞) between $P_{N_{disc},N_{iter}}(x)$ and the Lebesgue measure, which is the invariant measure associated to the symmetric tent map, are defined by

$$E_{1,N_{disc},Niter}(x) = \| P_{N_{disc},Niter}(x) - 0.5 \|_{L_1} \tag{4.21}$$

$$E_{2,N_{disc},Niter}(x) = \| P_{N_{disc},Niter}(x) - 0.5 \|_{L_2} \tag{4.22}$$

$$E_{\infty,N_{disc},Niter}(x) = \| P_{N_{disc},Niter}(x) - 0.5 \|_{L_\infty} \tag{4.23}$$

In the same way, an approximation of the correlation distribution function $C_{M,N}(x, y)$ is obtained by numerically building a regular partition of M^2 small squares (boxes) of J^2, embedded in the phase subspace (x^l, x^m)

$$s_i = -1 + \frac{2i}{M}, \quad t_j = -1 + \frac{2j}{M}, \quad i, j = 0, M \tag{4.24}$$

$$r_{i,j} = [s_i, s_{i+1}[\times[t_j, t_{j+1}[, \quad i, j = 0, M-2 \tag{4.25}$$

$$r_{M-1,j} = [s_{M-1}, 1] \times [t_j, t_{j+1}], \quad j = 0, M-2 \tag{4.26}$$

$$r_{i,M-1} = [s_i, s_{i+1}[\times[t_{M-1}, 1], \quad j = 0, M-2 \tag{4.27}$$

$$r_{M-1,M-1} = [s_{M-1}, 1] \times [t_{M-1}, 1] \tag{4.28}$$

The measure of the area of each box is

$$(s_{i+1} - s_i).(t_{i+1} - t_i) = \left(\frac{2}{M}\right)^2 \tag{4.29}$$

Once $N + Q$ iterated points (x_n^l, x_n^m) belonging to these boxes are collected, the relative number of iterates with respect to N/M^2 in each box $r_{i,j}$ represents the value $C_N(s_i, t_j)$. The approximated probability distribution function $C_N(x, y)$ defined here is then a two-dimensional step function, with M^2 steps. Since M can take several values in the next sections, we define

$$C_{M,N}(s_i, t_j) = \frac{1}{4}\frac{M^2}{N}(\#r_{i,j}) \tag{4.30}$$

where $\#r_{i,j}$ is the number of iterates belonging to the square $r_{i,j}$ and the constant $1/4$ allows the normalisation of $C_{M,N}(x, y)$ on the square J^2.

$$C_{M,N}(x, y) = C_{M,N}(s_i, t_j) \quad \forall (x, y) \in r_{i,j} \tag{4.31}$$

The discrepancies E_{C_1} (in norm L_1), E_{C_2} (in norm L_2) and E_{C_∞} (in norm L_∞) between $C_{N_{disc},N_{iter}}(x, y)$ and the uniform distribution on the square are defined by

$$E_{C_1,N_{disc},N_{iter}}(x, y) = \|C_{N_{disc},N_{iter}}(x, y) - 0.25\|_{L_1} \tag{4.32}$$

$$E_{C_2,N_{disc},N_{iter}}(x, y) = \|C_{N_{disc},N_{iter}}(x, y) - 0.25\|_{L_2} \tag{4.33}$$

$$E_{C_\infty,N_{disc},N_{iter}}(x, y) = \|C_{N_{disc},N_{iter}}(x, y) - 0.25\|_{L_\infty} \tag{4.34}$$

Finally, let $AC_{N_{disc},N_{iter}}$ be the autocorrelation distribution function which is the correlation function $C_{N_{disc},N_{iter}}$ of (4.31), defined in the delay space $(x_n^{(i)}, x_{n+1}^{(i)})$ instead of the phase (x^l, x^m) space. We define in the same manner than (4.32), (4.33), and (4.34) $E_{C_1,N_{disc},N_{iter}}(x, y)$, $E_{C_2,N_{disc},N_{iter}}(x, y)$, and $E_{C_\infty,N_{disc},N_{iter}}(x, y)$.

4.3.4 *Study of Randomness of TTL_2^{SC} and TTL_2^{RC}, and Other Topologies*

Using numerical computations, we assess the randomness properties of the two-dimensional maps TTL_2^{SC} and TTL_2^{RC}. If all requirements 1–8 of Fig. 4.7 are verified, the dynamical systems associated to those maps can be considered as pseudorandom and their application to cryptosystems is possible.

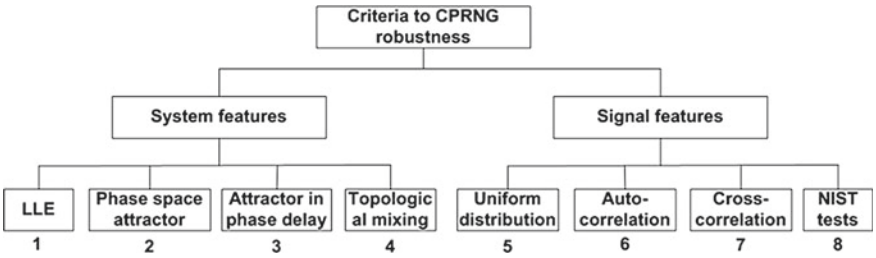

Fig. 4.7 The main criteria for assessing CPRNG (from [34])

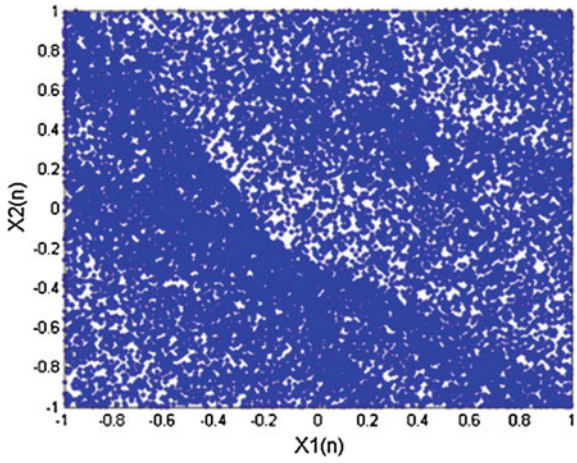

Fig. 4.8 Phase space behavior of TTL_2^{RC} non alternative (4.17), plot of 20, 000 points

Whenever one among the eight criteria is not satisfied for a given map, one cannot consider that the associated dynamical system is a good CPRNG candidate. As said above, when $\mu = 2$, the Lyapunov exponents of both considered maps are positive.

In the phase space, we plot the iterates in the system of coordinates $x_n^{(1)}$ versus $x_n^{(2)}$ in order to analyze the density of the points' distribution. Based on such an analysis, it is possible to assess the complexity of the behavior of dynamics, noticing any weakness or inferring on the nature of randomness. We also use the approximate invariant measures to assess more precisely the distribution of iterates.

The graphs of the attractor in phase space for the TTL_2^{RC} non-alternate (Fig. 4.8) and TTL_2^{SC} alternate (Fig. 4.9) maps are different. The TTL_2^{SC} map has well-scattered points in the whole pattern, but there are some more "concentrated" regions forming curves on the graph. Instead, the map TTL_2^{RC} has good repartition.

Some other numerical results we do not report in this chapter show that even if those maps have good random properties, it is possible to improve mapping randomness by modifying slightly network topologies.

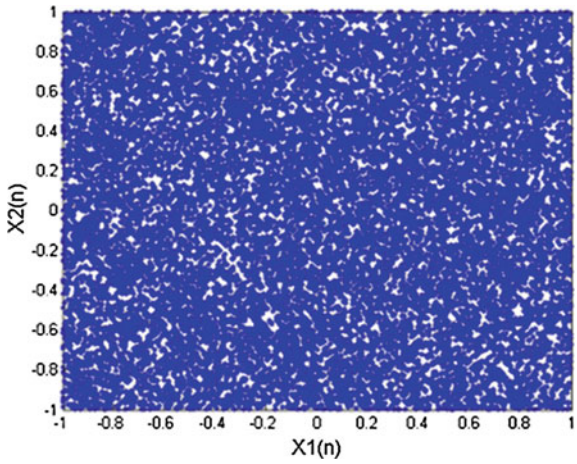

Fig. 4.9 Phase space behavior of TTL_2^{SC} alternative (4.18), plot of 20, 000 points

Equation (4.12) can be rewritten as

$$TTL_2^{SC}(x_n^{(1)}, x_n^{(2)}) = \begin{cases} x_{n+1}^{(1)} = 1 + 2(x_n^{(2)})^2 - 4|x_n^{(1)}| \\ x_{n+1}^{(2)} = 1 - 2(x_n^{(2)})^2 + 2(|x_n^{(1)}| - |x_n^{(2)}|) \end{cases} \tag{4.35}$$

In [38], it is shown that if the impact of component $x_n^{(1)}$ is reduced, randomness is improved. Hence, the following $MTTL_2^{SC}$ map is introduced

$$MTTL_2^{SC}(x_n^{(1)}, x_n^{(2)}) = \begin{cases} x_{n+1}^{(1)} = 1 + 2(x_n^{(2)})^2 - 2|x_n^{(1)}| \\ x_{n+1}^{(2)} = 1 - 2(x_n^{(2)})^2 + 2(|x_n^{(1)}| - |x_n^{(2)}|) \end{cases} \tag{4.36}$$

and the injection mechanism (4.9) is used as well, but it is restricted to three phases:

$$\begin{cases} \text{if } (x_{n+1}^{(1)} > 1) \text{ then subtract 2} \\ \text{if } (x_{n+1}^{(2)} < -1) \text{ then add 2} \\ \text{if } (x_{n+1}^{(2)} > 1) \text{ then subtract 2} \end{cases} \tag{4.37}$$

This injection mechanism allows the regions containing iterates to match excellently (Fig. 4.10).

The change of topology leading to $MTTL_2^{SC}$ greatly improves the density of iterates in the phase space (Fig. 4.11) where 10^9 points are plotted. The point distribution of iterates in phase delay for the variable $x^{(2)}$ is quite good as well (Fig. 4.12). On both pictures, a grid of 200×200 boxes is generated to use the box counting method defined in Sect. 4.3.3. Moreover, the largest Lyapunov exponent is equal to 0.5905, indicating a strong chaotic behavior.

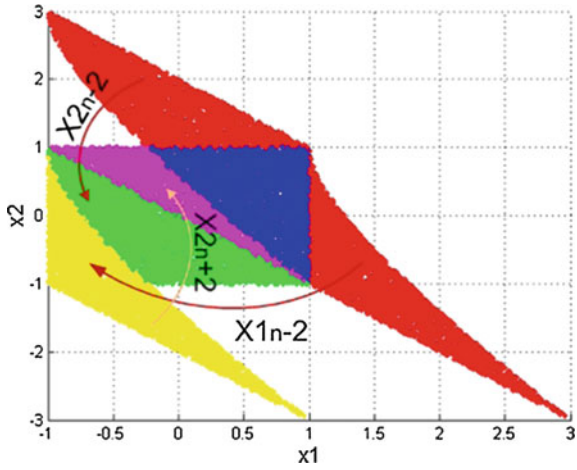

Fig. 4.10 Injection mechanism (4.21) of the $MTTL_2^{SC}$ alternative map (From [38])

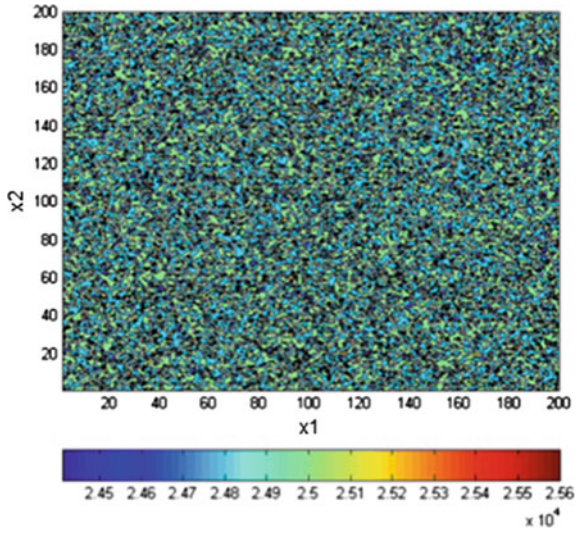

Fig. 4.11 Approximate density function of the $MTTL_2^{SC}$ alternative map, on the $(x^{(1)}, x^{(2)})$ plane (from [38])

However, regarding the phase delay for the variable $x^{(1)}$, results are not satisfactory. We have plotted in Fig. 4.13 10^9 iterates of $MTTL_2^{SC}$ in the delay plane, and in Fig. 4.14 the same iterates using the counting box method.

When such a great number of iterates is computed, one has to be cautious with raw graphical methods because irregularities of the density repartition are masked due to the huge number of plotted points. Therefore, these figures highlight the necessity of using the tools we have defined in Sect. 4.3.3.

Nevertheless, NIST tests were used to check randomness properties of $MTTL_2^{SC}$. Since they only require binary sequences, we generated 4×10^6 iterates whose 5×10^5 first ones were cut off. The rest of the sequence was converted to binary form according to the IEEE-754 standard (32-bit single-precision floating point).

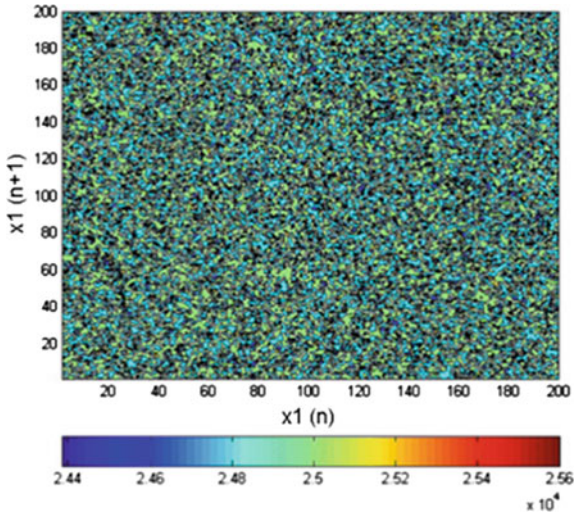

Fig. 4.12 Approximate density function of the $MTTL_2^{SC}$ alternative map, on the $(x_n^{(1)}, x_{n+1}^{(1)})$ plane (from [38])

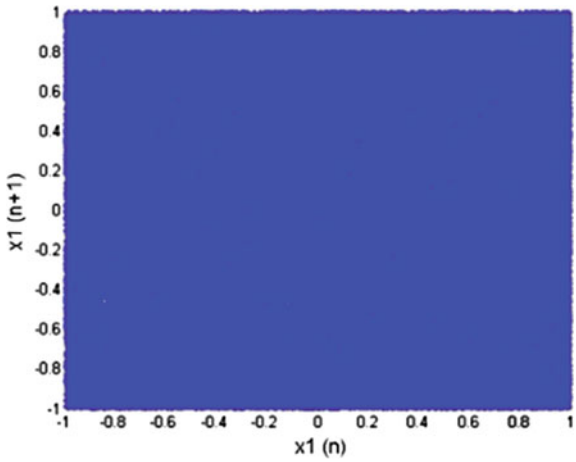

Fig. 4.13 Plot of one billion iterates of $MTTL_2^{SC}$ in the delay plane

Both variables of the generator successfully passed NIST tests, demonstrating strong randomness and robustness against numerous statistical attacks with respect to these tests (Figs. 4.15 and 4.16).

As said in the introduction, networks of coupled chaotic maps offer quasi-infinite possibilities to generate parallel streams of pseudorandom numbers. For example, in [39], the following modification of $MTTL_2^{SC}$ is also studied and shows good randomness properties

$$NTTL_2^{SC}(x_n^{(1)}, x_n^{(2)}) = \begin{cases} x_{n+1}^{(1)} = 1 - 2|x_n^{(2)}| = T_2(x_n^{(2)}) \\ x_{n+1}^{(2)} = 1 - (2x_n^{(2)})^2 - 2(|x_n^{(2)}| - |x_n^{(1)}|) \\ \qquad\quad = L_2(x_n^{(2)}) + T_2(x_n^{(2)}) - T_2(x_n^{(1)}) \end{cases} \tag{4.38}$$

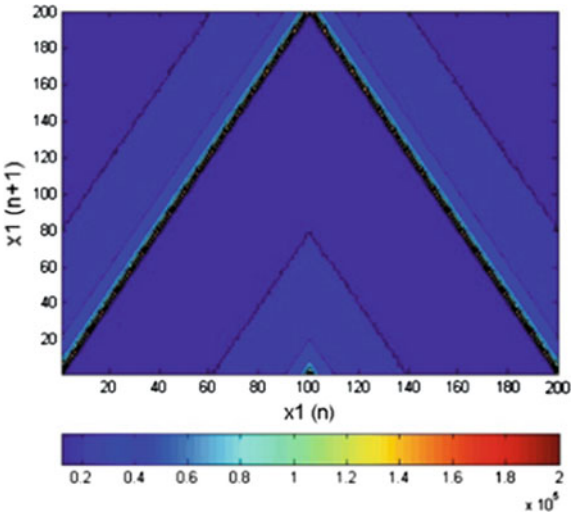

Fig. 4.14 Plot of one billion iterates of $MTTL_2^{SC}$ using the counting box method

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES

generator is <data/Modified TL_{\mu}^{SC} alternative map_x1.txt>

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
8	8	11	9	10	8	11	15	11	9	0.897763	100/100	Frequency
13	13	12	7	11	10	12	9	5	8	0.678686	99/100	BlockFrequency
6	7	5	12	16	12	12	9	14	7	0.191687	100/100	CumulativeSums
8	10	12	6	14	12	9	6	12	11	0.678686	100/100	Runs
14	11	12	10	15	5	6	13	8	6	0.236810	99/100	LongestRun
9	6	13	10	7	10	11	11	12	11	0.897763	97/100	Rank
11	12	6	19	4	11	11	13	8	5	0.037566	97/100	FFT
7	9	13	14	12	9	9	11	7	9	0.816537	100/100	NonOverlappingTemplate
10	11	15	10	11	9	12	6	11	5	0.595549	98/100	OverlappingTemplate
11	10	5	7	5	13	16	5	13	15	0.058984	100/100	Universal
14	6	11	10	7	9	13	12	8	10	0.739918	98/100	ApproximateEntropy
2	9	7	8	5	7	5	5	8	7	0.689019	63/63	RandomExcursions
5	8	4	4	6	4	4	11	6	11	0.222869	63/63	RandomExcursionsVariant
12	10	12	13	7	8	7	7	6	18	0.171867	99/100	Serial
9	13	11	12	7	9	7	16	7	9	0.534146	99/100	LinearComplexity

Fig. 4.15 Successful results of $NIST$ tests for the $MTTL_2^{SC}$ alternate map for the variable $x^{(1)}$ (from [38])

4.4 Numerical Study of a Particular Realisation of the $M_{\mu,p}^{k}$ Map in Higher Dimension

4.4.1 Mapping in Higher Dimension

Higher dimensional systems make it possible to achieve better randomness and uniform point distribution, because more perturbations and nonlinear mixing are involved. In this section, we focus on a particular realization of the $M_{\mu,p}^{k}$ map (4.10) from dimension two to dimension five.

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES

generator is <data/Modified TL_{\mu}^{SC} alternative map_x2.txt>

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
18	6	8	12	9	6	7	10	11	13	0.191687	98/100	Frequency
12	7	12	7	3	11	13	10	13	12	0.366918	98/100	BlockFrequency
15	14	8	6	8	13	7	10	9	10	0.494392	98/100	CumulativeSums
12	15	11	8	7	12	9	5	8	13	0.474986	98/100	Runs
9	12	13	13	9	14	9	6	8	7	0.637119	100/100	LongestRun
8	12	8	10	13	15	10	6	7	11	0.616305	98/100	Rank
8	12	9	15	9	8	17	9	9	4	0.181557	99/100	FFT
7	12	7	12	6	9	15	12	7	13	0.437274	100/100	NonOverlappingTemplate
9	12	11	3	16	8	10	13	10	8	0.289667	99/100	OverlappingTemplate
9	13	10	6	8	8	11	10	11	14	0.816537	99/100	Universal
7	24	9	7	7	8	8	17	7	6	0.000347	98/100	ApproximateEntropy
2	4	2	5	5	7	2	13	4	8	0.011791	52/52	RandomExcursions
5	4	8	5	2	1	8	6	4	9	0.191687	52/52	RandomExcursionsVariant
6	10	8	7	15	15	15	8	8	8	0.236810	100/100	Serial
7	9	11	11	6	15	7	11	8	15	0.419021	99/100	LinearComplexity

Fig. 4.16 *NIST* tests for the variable $x^{(2)}$ (from [38])

Usually, three or four dimensions are complex enough to create robust random sequences as we show here. Thus, it is advantageous if the system can increase its dimension. Since the $MTTL_2^{SC}$ alternative map cannot be nested in higher dimensions, we describe how to improve randomness and to obtain the best distribution of points, and how to produce more complex dynamics than the $TTL_2^{SC}(x^{(2)}, x^{(1)})$ alternative map in dimension greater than 2. Let

$$TTL_2^{RC,pD} = \begin{cases} x_{n+1}^{(1)} = 1 - 2|x_n^{(1)}| + 2(|x_n^{(2)}| - (x_n^{(1)})^2) \\ x_{n+1}^{(2)} = 1 - 2|x_n^{(2)}| + 2(|x_n^{(3)}| - (x_n^{(2)})^2) \\ \vdots \\ x_{n+1}^{(p)} = 1 - 2|x_n^{(p)}| + 2(|x_n^{(1)}| - (x_n^{(p)})^2) \end{cases} \tag{4.39}$$

be this realization.

We show in Figs. 4.17 and 4.18 successful NIST tests for $TTL_2^{RC,pD}$ in 3-D and 4-D, for the variable $x^{(1)}$.

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
8	14	8	9	10	9	11	12	6	13	0.779188	100/100	Frequency
11	9	9	8	6	15	7	13	9	13	0.574903	100/100	BlockFrequency
14	6	13	7	11	5	10	11	9	14	0.401199	100/100	CumulativeSums
12	10	7	7	16	8	13	7	13	7	0.366918	99/100	CumulativeSums
16	9	7	11	14	12	6	13	7	5	0.181557	100/100	Runs
13	9	14	11	11	8	9	12	5	8	0.678686	100/100	LongestRun
14	9	7	8	9	16	9	12	6	10	0.455937	100/100	Rank
13	4	9	11	7	4	10	12	19	11	0.037566	100/100	FFT
14	8	8	9	8	15	11	11	8	8	0.699313	100/100	NonOverlappingTemplate
14	15	12	10	6	9	13	7	3	11	0.162606	99/100	OverlappingTemplate
8	7	11	16	9	12	10	9	7	11	0.678686	100/100	Universal
13	11	10	12	6	12	12	14	6	4	0.304126	97/100	ApproximateEntropy
5	5	6	9	2	7	5	8	9	6	0.637119	62/62	RandomExcursions
6	2	4	9	6	11	6	5	6	7	0.407091	62/62	RandomExcursionsVariant
13	8	15	8	12	9	7	15	8	5	0.275709	99/100	Serial
13	6	15	12	11	6	15	8	8	6	0.213309	99/100	Serial
9	6	8	13	8	11	10	11	12	12	0.883171	99/100	LinearComplexity

Fig. 4.17 *NIST* test for $TTL_2^{RC,3D}$ for $x^{(1)}$ (from [38])

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
7	5	12	14	10	9	12	16	8	7	0.289667	99/100	Frequency
7	7	9	10	6	10	14	8	10	19	0.137282	99/100	BlockFrequency
8	2	9	16	13	9	13	9	7	14	0.090936	99/100	CumulativeSums
5	8	14	11	11	11	14	5	10	11	0.437274	99/100	CumulativeSums
6	16	13	11	9	10	8	7	11	9	0.554420	100/100	Runs
9	13	6	9	14	10	8	11	12	8	0.779188	99/100	LongestRun
9	8	14	6	12	12	8	10	8	13	0.719747	100/100	Rank
10	10	17	5	9	13	14	10	6	6	0.153763	99/100	FFT
9	7	9	13	9	10	10	14	6	13	0.719747	100/100	NonOverlappingTemplate
5	9	12	7	7	12	12	13	12	11	0.637119	99/100	OverlappingTemplate
12	16	8	7	9	10	7	12	8	11	0.616305	99/100	Universal
8	16	6	12	11	13	5	7	13	9	0.249284	99/100	ApproximateEntropy
4	8	4	6	8	5	7	8	9	7	0.804337	66/66	RandomExcursions
4	7	7	8	2	8	6	8	7	9	0.602458	66/66	RandomExcursionsVariant
11	10	10	18	6	5	11	12	10	7	0.213309	100/100	Serial
8	11	10	10	12	11	10	9	9	10	0.998821	98/100	Serial
10	7	13	11	8	7	11	14	11	8	0.798139	99/100	LinearComplexity

Fig. 4.18 $NIST$ test for $TTL_2^{RC,4D}$ for $x^{(1)}$ (from [38])

4.4.2 Numerical Experiments

All NIST tests for dimensions three to five for every variable are successful, showing that these realizations in 3-D up to 5-D are good CPRNGs. In addition to those tests, we study the mapping more thoroughly, far beyond the NIST tests which are limited to a few million iterates and which seem not robust enough for industrial mathematics, although they are routinely used worldwide.

In order to check the portability of the computations on multicore architectures, we have implemented all our numerical experiments on several different multicore machines.

4.4.2.1 Checking the Uniform Repartition of Iterated Points

We first compute the discrepancies E_1 (in norm L_1), E_2 (in norm L_2)n and E_∞ (in norm E_∞) between $P_{N_{disc},N_{iter}}(x)$ and the Lebesgue measure which is the uniform measure on the interval $J = [-1, 1]$. We set $M = N_{iter} = 200$, and vary the number N_{iter} of iterated points in the range 10^4 to 10^{14}. From our knowledge, this article is the first one that checks such a huge number of iterates (in conjunction with [39]). We compare $E_{1,200,N_{iter}}(x^{(1)})$ for $TTL_2^{RC,pD}$ with $p = 2$ to 5 (Table 4.2, Fig. 4.19).

As shown in Fig. 4.19, $E_{1,200,N_{iter}(x^{(1)})}$ decreases steadily when N_{iter} increases. However, the decreasing process is promptly (with respect to N_{iter}) bounded below for $p = 2$. This is also the case for other values of p, however, the boundary decreases with p, therefore showing better randomness properties for higher dimensional mappings.

Table 4.3 compares $x^{(1)}, x^{(2)}, \ldots, x^{(p)}$ for $TTL_2^{RC,5D}$, for different values of N_{iter}. It is obvious that the same quality of randomness is obtained for each one of them, contrarily to the results obtained for $MTTL_2^{SC}$.

Table 4.2 $E_{1,200,N_{iter}}(x^{(1)})$ for $TTL_2^{RC,pD}$ with $p = 2$ to 5

N_{iter}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
10^4	1.5631	1.5553	1.5587	1.5574
10^5	0.55475	0.5166	0.51315	0.5154
10^6	0.269016	0.159306	0.158548	0.158436
10^7	0.224189	0.050509	0.0501934	0.0505558
10^8	0.219427	0.0164173	0.0159175	0.0160018
10^9	0.218957	0.00640196	0.00505021	0.00509754
10^{10}	0.218912	0.00420266	0.00160505	0.00160396
10^{11}	0.218913	0.00392507	0.000513833	0.000505591
10^{12}	0.218913	0.00389001	0.000189371	0.000160547
10^{13}	0.218914	0.00388778	0.000112764	5.04473e-05
10^{14}	0.218914	0.003887	0.000101139	1.59929e-05

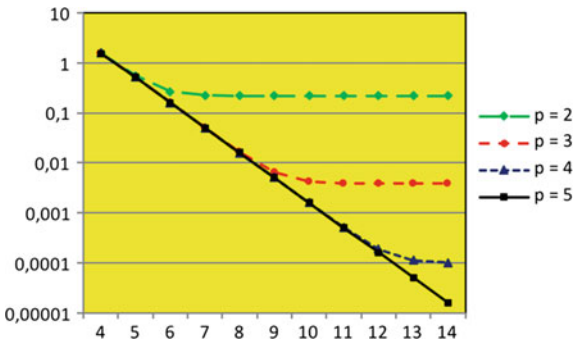

Fig. 4.19 Graph of $E_{1,200,N_{iter}}(x^{(1)})$ for $TTL_2^{RC,pD}$ with $p = 2$ to 5, with respect to N_{iter} (horizontal axis, logarithmic value)

Table 4.3 $E_{1,200,N_{iter}}(x^{(i)})$ for $TTL_2^{RC,5D}$ for $i = 1$ to 5

N_{iter}	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(5)}$
10^4	1.5574	1.55725	1.556	1.5585	1.55925
10^5	0.5154	0.51061	0.5098	0.51494	0.51293
10^6	0.158436	0.159162	0.159564	0.159864	0.159926
10^7	0.0505558	0.0504866	0.0503746	0.0505688	0.0505268
10^8	0.0160018	0.0158328	0.0158498	0.0160336	0.01591
10^9	0.00509754	0.0050514	0.00505756	0.00501442	0.00503467
10^{10}	0.00160396	0.00159738	0.00160099	0.00159454	0.00159916
10^{11}	0.000505591	0.000506327	0.000507006	0.000504258	0.000507526
10^{12}	0.000160547	0.000159192	0.000160014	0.000159213	0.000159159
10^{13}	5.04473e-05	5.03574e-05	5.05868e-05	5.04694e-05	5.01681e-05
10^{14}	1.59929e-05	1.60291e-05	1.59282e-05	1.59832e-05	1.60775e-05

Table 4.4 Comparison between $E_{1,200,N_{iter}}(x^{(1)})$, $E_{2,200,N_{iter}}(x^{(1)})$, and $E_{\infty,200,N_{iter}}(x^{(1)})$ for $TTL_2^{RC,5D}$

N_{iter}	Norm L_1	Norm L_2	Norm L_∞
10^4	1.5574	2.0038	19
10^5	0.5154	0.635522	3.4
10^6	0.158436	0.199731	0.96
10^7	0.0505558	0.0633486	0.256
10^8	0.0160018	0.02007	0.0896
10^9	0.00509754	0.00638219	0.02688
10^{10}	0.00160396	0.00200966	0.008672
10^{11}	0.000505591	0.000631963	0.0027444
10^{12}	0.000160547	0.000201102	0.0008602
10^{13}	5.04473e-05	6.32233e-05	0.00026894
10^{14}	1.59929e-05	2.00533e-05	9.89792e-05

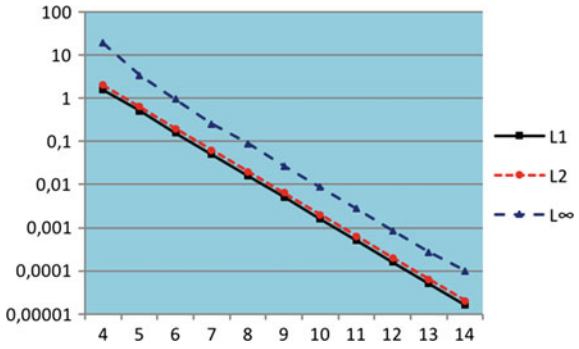

Fig. 4.20 Comparison between $E_{1,200,N_{iter}}(x^{(1)})$, $E_{2,200,N_{iter}}(x^{(1)})$, and $E_{\infty,200,N_{iter}}(x^{(1)})$ (vertical axis) for $TTL_2^{RC,5D}$ with respect to N_{iter} (horizontal axis, logarithmic value)

The comparisons between $E_{1,200,N_{iter}}(x^{(1)})$, $E_{2,200,N_{iter}}(x^{(1)})$, and $E_{\infty,N_{iter}}(x^{(1)})$ for $TTL_2^{RC,5D}$ in Table 4.4 and Fig. 4.20 show that

$$E_{1,200,N_{iter}}(x^{(1)}) < E_{2,200,N_{iter}}(x^{(1)}) < E_{\infty,N_{iter}}(x^{(1)}) \tag{4.40}$$

for every value of N_{iter}.

4.4.2.2 Autocorrelation Study in the Delay Space

In this section, we assess autocorrelation errors $E_{AC_1,N_{disc},N_{iter}}(x, y)$, $E_{AC_2,N_{disc},N_{iter}}(x, y)$, and $E_{AC_\infty,disc,N_{iter}}(x, y)$, defined by Equations (4.32), (4.33), and (4.34), in the delay space. As in Sect. 4.4.2.1, we have performed the experi-

Table 4.5 Comparison between $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+1}^{(1)})$, $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+2}^{(1)})$, and $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+3}^{(1)})$ for $TTL_2^{RC,2D}$

N_{iter}	$(x_n^{(1)}, x_{n+1}^{(1)})$	$(x_n^{(1)}, x_{n+2}^{(1)})$	$(x_n^{(1)}, x_{n+3}^{(1)})$
10^4	1.55955	1.57265	1.5515
10^5	0.55199	0.699355	0.547539
10^6	0.269654	0.519675	0.250936
10^7	0.224104	0.49941	0.198634
10^8	0.21938	0.497011	0.193007
10^9	0.218949	0.496766	0.192309
10^{10}	0.218914	0.496808	0.192253
10^{11}	0.218915	0.496793	0.192247
10^{12}	0.218913	0.496797	0.192245
10^{13}	0.218914		
10^{14}	0.218914		

Table 4.6 Comparison between $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+1}^{(1)})$, $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+2}^{(1)})$, and $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+3}^{(1)})$ for $TTL_2^{RC,3D}$

N_{iter}	$(x_n^{(1)}, x_{n+1}^{(1)})$	$(x_n^{(1)}, x_{n+2}^{(1)})$	$(x_n^{(1)}, x_{n+3}^{(1)})$
10^4	1.55575	1.5528	1.5489
10^5	0.51516	0.512514	0.514889
10^6	0.160148	0.158843	0.159728
10^7	0.0505148	0.0515855	0.0550998
10^8	0.0164343	0.0190644	0.0269715
10^9	0.00640451	0.0113919	0.0221408
10^{10}	0.00420824	0.0103092	0.0216388
10^{11}	0.003926197	0.0102078	0.0215621
10^{12}	0.00388937	0.0101965	0.0215576
10^{13}	0.00388768		
10^{14}	0.003887		

ments for $M = 20$ to $20,000$, however, in this chapter, we only present the results for $M = 200$. We first compare $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+1}^{(1)})$ with $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+2}^{(1)})$ and $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+3}^{(1)})$ for $TTL_2^{RC,pD}$ when the dimension of the system is within the range $p = 2$ to 5 (Tables 4.5, 4.6, 4.7 and 4.8). It is possible to see that better randomness properties are obtained for higher dimensional mappings.

The comparison between $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+1}^{(1)})$, $E_{AC_2,200,N_{iter}}(x_n^{(1)}, x_{n+1}^{(1)})$, and $E_{AC_\infty,200,N_{iter}}(x_n^{(1)}, x_{n+1}^{(1)})$ for $TTL_2^{RC,5D}$ in Table 4.9 shows that numerically

Table 4.7 Comparison between $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+1}^{(1)})$, $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+2}^{(1)})$, and $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+3}^{(1)})$ for $TTL_2^{RC,4D}$

N_{iter}	$(x_n^{(1)}, x_{n+1}^{(1)})$	$(x_n^{(1)}, x_{n+2}^{(1)})$	$(x_n^{(1)}, x_{n+3}^{(1)})$
10^4	1.5571	1.5518	1.54985
10^5	0.51115	0.510784	0.511188
10^6	0.158472	0.159263	0.159292
10^7	0.0503522	0.0506053	0.0506126
10^8	0.0159245	0.0159484	0.015918
10^9	0.00502109	0.00502642	0.00502197
10^{10}	0.00159193	0.00161135	0.00162232
10^{11}	0.00051438	0.000532052	0.0005489
10^{12}	0.000189418	0.000217634	0.000276982
10^{13}	0.000112771		
10^{14}	0.000101139		

Table 4.8 Comparison between $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+1}^{(1)})$, $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+2}^{(1)})$, and $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+3}^{(1)})$ for $TTL_2^{RC,5D}$

N_{iter}	$(x_n^{(1)}, x_{n+1}^{(1)})$	$(x_n^{(1)}, x_{n+2}^{(1)})$	$(x_n^{(1)}, x_{n+3}^{(1)})$
10^4	1.5577	1.5531	1.54975
10^5	0.51372	0.511144	0.513918
10^6	0.15872	0.158775	0.158022
10^7	0.0503658	0.0504011	0.0501632
10^8	0.0159765	0.0159229	0.0159837
10^9	0.00509015	0.00502869	0.00503495
10^{10}	0.00159581	0.00159398	0.00158143
10^{11}	0.000505068	0.000506309	0.000502137
10^{12}	0.000160547	0.000159144	0.000159246
10^{13}	5.0394e-05		
10^{14}	1.59929e-05		

$$E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+1}^{(1)}) < E_{AC_2,200,N_{iter}}(x_n^{(1)}, x_{n+1}^{(1)}) < E_{AC_\infty,200,N_{iter}}(x_n^{(1)}, x_{n+1}^{(1)}) \tag{4.41}$$

Equation (4.41) is not only valid for $M = 200$, but also for other values of M and every component of X.

In order to illustrate the numerical results displayed in these tables, we plot in Fig. 4.21 the repartition of iterates of $TTL_2^{RC,5D}$ in the delay plane $(x_n^{(1)}, x_{n+1}^{(1)})$, using the box counting method. On a grid of 200×200 boxes ($N_{iter} = M = 200$),

Table 4.9 Comparison between $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_{n+1}^{(1)})$, $E_{AC_2,200,N_{iter}}(x_n^{(1)}, x_{n+1}^{(1)})$, and $E_{AC_\infty,200,N_{iter}}(x_n^{(1)}, x_{n+1}^{(1)})$ for $TTL_2^{RC,5D}$

N_{iter}	Norm L_1	Norm L_2	Norm L_∞
10^4	1.5577	2.0012	19
10^5	0.51372	0.633959	3.8
10^6	0.15872	0.199793	0.88
10^7	0.0503658	0.0631425	0.26
10^8	0.0159765	0.0200503	0.084
10^9	0.00509015	0.00636626	0.02528
10^{10}	0.00159581	0.00199936	0.008604
10^{11}	0.000505068	0.000633088	0.0025432
10^{12}	0.000160547	0.000201102	0.0008602
10^{13}	5.0394e-05	6.31756e-05	0.000280168
10^{14}	1.59929e-05	2.00533e-05	9.89792e-05

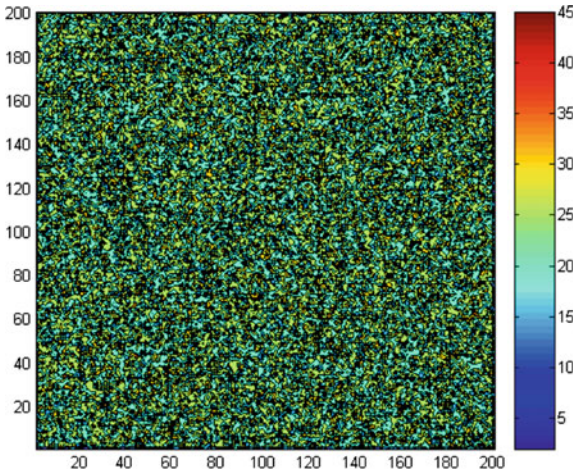

Fig. 4.21 Repartition of iterates in the delay plane $(x_n^{(1)}, x_{n+1}^{(1)})$ of $TTL_2^{RC,5D}$ with the box counting method; 10^6 points are generated on a grid of 200×200 boxes, the horizontal axis is $x_n^{(1)}$, and the vertical axis is $x_{n+1}^{(1)}$

we have generated 10^6 points. The horizontal axis is $x_n^{(1)}$, and the vertical axis is $x_{n+1}^{(1)}$. In order to check very carefully the repartition of the iterates of $TTL_2^{RC,5D}$, we have also plotted the repartition in the delay planes $(x_n^{(1)}, x_{n+2}^{(1)})$, $(x_n^{(1)}, x_{n+3}^{(1)})$, and $(x_n^{(1)}, x_{n+4}^{(1)})$ (Figs. 4.22, 4.23, and 4.24). This repartition is uniform everywhere as shown also in Table 4.8.

We find the same regularity for every component $x^{(2)}, x^{(3)}, x^{(4)}$, and $x^{(5)}$, as shown in Figs. 4.25, 4.26, 4.27, 4.28, and in Table 4.10.

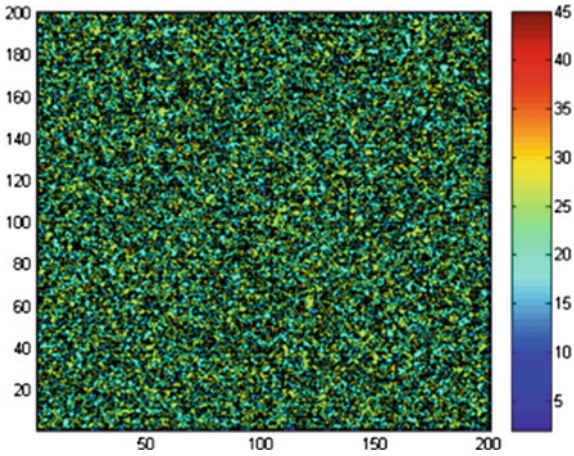

Fig. 4.22 Repartition of iterates in the delay plane $(x_n^{(1)}, x_{n+2}^{(1)})$ of $TTL_2^{RC,5D}$, as in Fig. 4.21

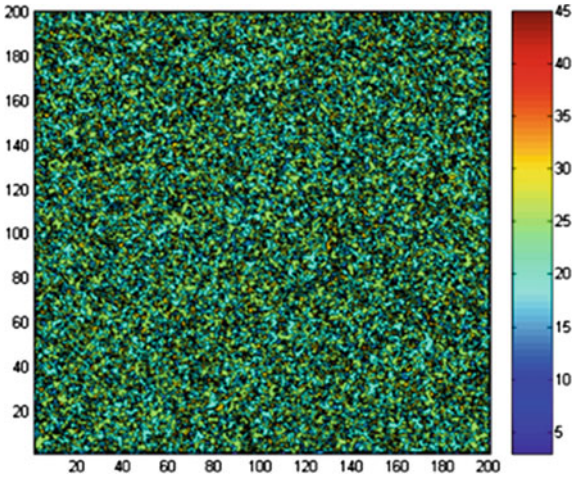

Fig. 4.23 Repartition of iterates in the delay plane $(x_n^{(1)}, x_{n+3}^{(1)})$ of $TTL_2^{RC,5D}$, as in Fig. 4.21

4.4.2.3 Autocorrelation Study in the Phase Space

Finally, in this section, we assess the autocorrelation errors $E_{C_1,N_{disc},N_{iter}}(x, y)$, $E_{C_2,N_{disc},N_{iter}}(x, y)$, and $E_{C_\infty,N_{disc},N_{iter}}(x, y)$, defined by Eqs. (4.32), (4.33), and (4.34), in the phase space. We checked all combinations of the components. Due to space limitations, we only provide part of the numerical computations we have performed to carefully check the randomness of $TTL_2^{RC,pD}$ for $p = 2, 5$ and $i = 1, p$. Like in the previous section, we only provide the results for $M = 200$. We first compare $E_{C_1,200,N_{iter}}(x_n^{(1)}, x_n^{(2)})$, $E_{C_2,200,N_{iter}}(x_n^{(1)}, x_n^{(2)})$, and $E_{C_\infty,200,N_{iter}}(x_n^{(1)}, x_n^{(2)})$ (Table 4.11), and our other results verified that

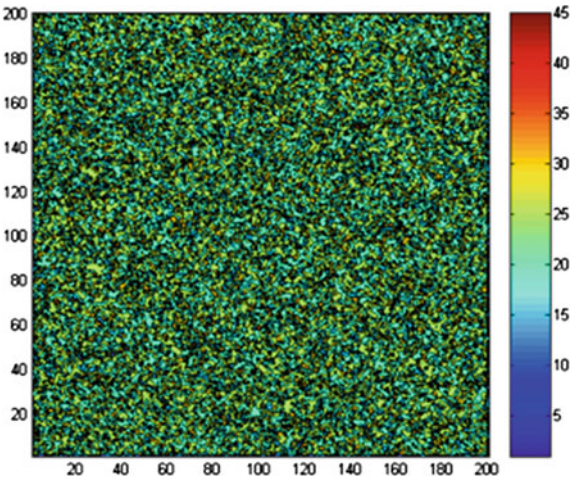

Fig. 4.24 Repartition of iterates in the delay plane $(x_n^{(1)}, x_{n+4}^{(1)})$ of $TTL_2^{RC,5D}$, as in Fig. 4.21

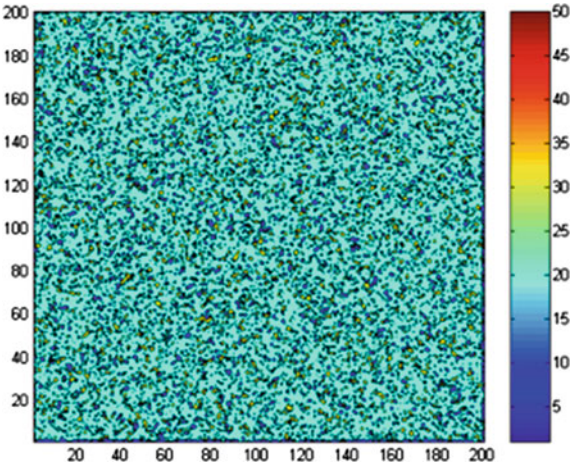

Fig. 4.25 Repartition of iterates in the delay plane $(x_n^{(2)}, x_{n+1}^{(2)})$ of $TTL_2^{RC,5D}$; box counting method, 10^6 points are generated on a grid of 200×200 boxes, the horizontal axis is $x_n^{(2)}$, and the vertical axis is $x_{n+1}^{(2)}$

$$E_{C_1,N_{disc},N_{iter}}(x_n^{(1)}, x_n^{(2)}) < E_{C_2,N_{disc},N_{iter}}(x_n^{(1)}, x_n^{(2)}) < E_{C_\infty,N_{disc},N_{iter}}(x_n^{(1)}, x_n^{(2)}) \tag{4.42}$$

We have also assessed the autocorrelation errors $E_{C_1,N_{disc},N_{iter}}(x_n^{(i)}, x_n^{(j)})$ for $i, j = 1, 5, i \neq j$, and various values of the number of iterates for $TTL_2^{RC,5D}$ (Table 4.12). We have performed the same experiments for $E_{C_1,N_{disc},N_{iter}}(x_n^{(1)}, x_n^{(2)})$ for $p = 1, 5$ (Table 4.13).

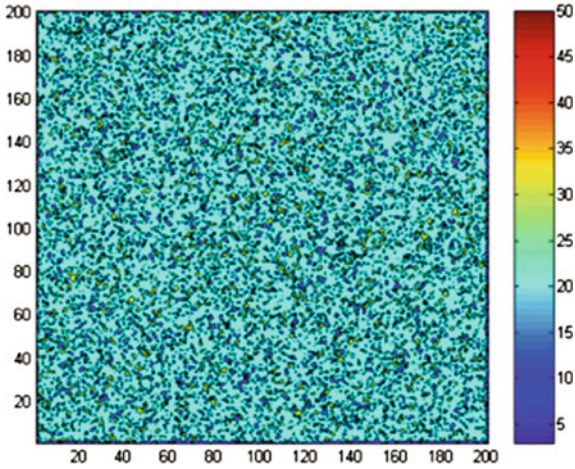

Fig. 4.26 Repartition of iterates in the delay plane $(x_n^{(3)}, x_{n+1}^{(3)})$ of $TTL_2^{RC,5D}$, as in Fig. 4.25

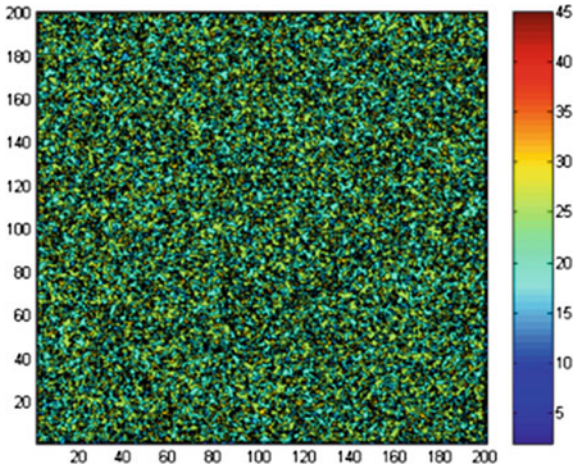

Fig. 4.27 Repartition of iterates in the delay plane $(x_n^{(4)}, x_{n+1}^{(4)})$ of $TTL_2^{RC,5D}$, as in Fig. 4.25

Our numerical experiments all show a similar trend: $TTL_2^{RC,pD}$ is a good candidate for a CPRNG, and the randomness performance of such mappings increases in higher dimensions.

4.4.2.4 Checking the Influence of Discretization in Computation of Approximated Invariant Measures

In order to verify that the computations we have performed using the discretization $M = N_{disc} = 200$ of the phase space and the delay space in the numerical experi-

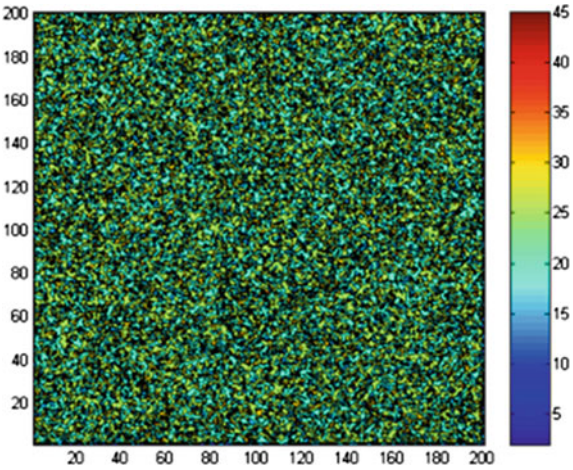

Fig. 4.28 Repartition of iterates in the delay plane $(x_n^{(5)}, x_{n+1}^{(5)})$ of $TTL_2^{RC,5D}$, as in Fig. 4.25

Table 4.10 Comparison between $E_{AC_1,200,N_{iter}}(x_n^{(i)}, x_{n+1}^{(i)})$, $E_{AC_1,200,N_{iter}}(x_n^{(i)}, x_{n+2}^{(i)})$, and $E_{AC_1,200,N_{iter}}(x_n^{(i)}, x_{n+3}^{(i)})$ for $TTL_2^{RC,5D}$ for $i = 1$ to 5

N_{iter}	i	$(x_n^{(i)}, x_{n+1}^{(i)})$	$(x_n^{(i)}, x_{n+2}^{(i)})$	$(x_n^{(i)}, x_{n+3}^{(i)})$
10^4	1	1.5577	1.5531	1.54975
	2	1.5577	1.5526	1.5508
	3	1.5577	1.5542	1.54445
	4	1.5577	1.5533	1.5468
	5	1.5577	1.5541	1.5504
10^8	1	0.0159765	0.0159229	0.0159837
	2	0.0159765	0.0159999	0.0158293
	3	0.0159765	0.0159047	0.0159605
	4	0.0159765	0.0159269	0.0159282
	5	0.0159765	0.0160591	0.0159274
10^{12}	1	0.000160547	0.000159144	0.000159246
	2	0.000159192	0.000159635	0.000159064
	3	0.000160014	0.00015892	0.000160555
	4	0.000159213	0.000159696	0.000159215
	5	0.000159159	0.000158831	0.000160007

ments do not introduce artifacts, we have performed the same computations varying also the value of $M = N_{disc} = 20, 200, 2000, 20000$, for $TTL_2^{RC,4D}$ (Table 4.14 and Fig. 4.29). The results show a normal regularity following the increasing value of N_{disc}.

Table 4.11 Comparison between $E_{AC_1,200,N_{iter}}(x_n^{(1)}, x_n^{(2)})$, $E_{AC_2,200,N_{iter}}(x_n^{(1)}, x_n^{(2)})$, and $E_{AC_\infty,200,N_{iter}}(x_n^{(1)}, x_n^{(2)})$ for $TTL_2^{RC,5D}$

N_{iter}	Norm L_1	Norm L_2	Norm L_∞
10^4	1.55915	2.00818	15
10^5	0.514	0.633448	3.4
10^6	0.158058	0.198943	0.96
10^7	0.0505508	0.0634574	0.308
10^8	0.0160114	0.0200538	0.0852
10^9	0. 00507915	0.0063595	0.02716
10^{10}	0.0015927	0.00199644	0.008128
10^{11}	0.000506086	0.000633916	0.0025712
10^{12}	0.000158795	0.000199203	0.00089288
10^{13}	5.03666e-05	6.30356e-05	0.000270156
10^{14}	1.60489e-05	2.00692e-05	8.53124e-05

Table 4.12 Comparison between $E_{C_1,200,N_{iter}}(x_n^{(i)}, x_n^{(j)})$, for $i, j = 1$ to 5, $i \neq j$, and for various values of number of iterates for $TTL_2^{RC,5D}$

N_{iter}	10^6	10^8	10^{10}	10^{12}	10^{14}
$x(1), x(2)$	0.158058	0.0160114	0.0015927	0.000158795	1.60489e-05
$x(1), x(3)$	0.158956	0.0159261	0.00159456	0.000159326	1.73852e-05
$x(1), x(4)$	0.15943	0.0160321	0.00160091	0.000160038	1.74599e-05
$x(1), x(5)$	0.159074	0.0158962	0.00160204	0.000159048	1.59133e-05
$x(2), x(3)$	0.15825	0.0159754	0.00159442	0.000160659	1.60419e-05
$x(2), x(4)$	0.159248	0.0159668	0.00159961	0.000160313	1.73507e-05
$x(2), x(5)$	0.15889	0.0160116	0.0015934	0.000160462	1.73496e-05
$x(3), x(4)$	0.159136	0.0158826	0.00158123	0.000158758	1.59451e-05
$x(3), x(5)$	0.159216	0.0159341	0.00161268	0.000159079	1.75013e-05
$x(4), x(5)$	0.158918	0.0160516	0.0016008	0.000159907	1.59445e-05

4.4.2.5 Computation Time of PRNs

The numerical experiments performed in this section have involved several multicore machines. We show in Table 4.15 different computation times (in seconds) for the generation of N_{iter} PRNs for $TTL_2^{RC,pD}$ with $p = 2$ to 5, and various values of the number of iterates (N_{iter}). The machine used is a laptop computer with a Core i7 4980HQ processor with eight logical cores.

Table 4.16 shows the computation time of only one PRN in the same experiment. Time is expressed in 10^{-10} s.

Table 4.13 Comparison between $E_{C_1,200,N_{iter}}(x_n^{(i)}, x_n^{(j)})$, for $TTL_2^{RC,pD}$ for $p = 2, \ldots, 5$, and various values of the number of iterates

N_{iter}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
10^4	1.5624	1.5568	1.55725	1.55915
10^5	0.57955	0.5163	0.51083	0.514
10^6	0.330084	0.160282	0.158256	0.158058
10^7	0.294918	0.0509584	0.0504002	0.0505508
10^8	0.291428	0.0176344	0.0157924	0.0160114
10^9	0.291012	0.00911485	0.00506758	0.00507915
10^{10}	0.291025	0.00783204	0.00159046	0.0015927
10^{11}	0.291033	0.00771201	0.000521561	0.000506086
10^{12}	0.291036	0.00769998	0.000209109	0.000158795
10^{13}		0.00769867	0.000150031	5.03666e-05
10^{14}		0.00769874	0.000144162	1.60489e-05

Table 4.14 Comparison between $E_{C_1,N_{disc},N_{iter}}(x_n^{(1)}, x_n^{(2)})$,for $TTL_2^{RC,4D}$ $M = N_{disc} = 20, 200, 2000, 20000$, and various values of the number of iterates

N_{iter}	$N_{disc} = 20$	$N_{disc} = 200$	$N_{disc} = 2000$	$N_{disc} = 20000$
10^4	0.1508	1.55725	1.99501	1.99995
10^5	0.04894	0.51083	1.95066	1.9995
10^6	0.015544	0.158256	1.55759	1.99501
10^7	0.005487	0.0504002	0.512542	1.95062
10^8	0.00159524	0.0157924	0.158971	1.55763
10^9	0.000517392	0.00506758	0.0504555	0.513028
10^{10}	0.000205706	0.00159046	0.0159528	0.159054
10^{11}	0.000147202	0.000521561	0.0050481	0.0504422
10^{12}		0.000209109		
10^{13}		0.000150031		
10^{14}		0.000144162		

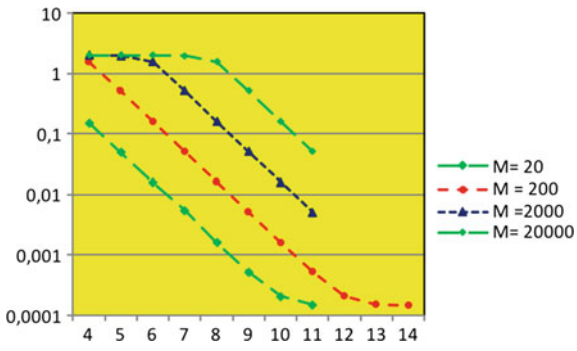

Fig. 4.29 Comparison between $E_{C_1,N_{disc},N_{iter}}(x_n^{(1)}, y_n^{(2)})$, for $TTL_2^{RC,4D}$, $M = N_{disc} = 20, 200, 2000, 20, 000$, and various values of the number of iterates

Table 4.15 Comparison of computation times (in second) for the generation of N_{iter} PRNs for $TTL_2^{RC,pD}$ with $p = 2$ to 5, and various values of N_{iter} iterates

N_{iter}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
10^4	0.000146	0.000216	0.000161	0.000142
10^5	0.000216	0.000277	0.000262	0.000339
10^6	0.001176	0.002403	0.001681	0.002467
10^7	0.011006	0.016195	0.018968	0.022351
10^8	0.113093	0.161776	0.166701	0.227638
10^9	1.09998	1.58949	1.60441	2.29003
10^{10}	11.4901	18.0142	18.537	26.1946
10^{11}	123.765	183.563	185.449	257.244

Table 4.16 Comparison of computation times (in 10^{-10} s) for the generation of only one PRN for $TTL_2^{RC,pD}$ with $p = 2$ to 5, and various values of the number of iterates

N_{iter}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
10^4	73.0	72.0	40.25	28.4
10^5	10.8	9.233	6.55	6.78
10^6	5.88	8.01	4.2025	4.934
10^7	5.503	5.39833	4.742	4.702
10^8	5.65465	4.0444	4.16753	4.55276
10^9	5.4999	5.2983	4.01103	4.58006
10^{10}	5.74505	4.50335	4.63425	5.23892
10^{11}	6.18825	6.11877	4.63622	5.14488

These results show that the pace of computation is very high. When $TTL_2^{RC,5D}$ is the mapping tested, and the machine used is a laptop computer with a Core i7 4980HQ processor with 8 logical cores, computing 10^{11} iterates with five parallel streams of PRNs leads to around 2 billion PRNs being produced per second. Since these PRNs are computed in the standard double precision format, it is possible to extract from each 50 random bits (the size of the mantissa being 52 bits for a double precision floating-point number in standard IEEE-754). Therefore, $TTL_2^{RC,5D}$ can produce 100 billion random bits per second, an incredible pace! With a machine with 4 Intel Xeon E7–4870 processors having a total of 80 logical cores, the computation is twice as fast, producing 2×10^{11} random bits per second.

4.5 Conclusion

In this chapter, we thoroughly explored the novel idea of combining features of a tent map (T_μ) and a logistic map (L_μ) to produce a new map with improved properties, through combination in several network topologies. This idea was recently introduced [38, 39] in order to improve previous CPRNGs. We have summarized the previously explored topologies in dimension two. We have presented new results of numerical experiments in higher dimensions (up to five) for the mapping $TTL_2^{RC,pD}$ on multicore machines and shown that $TTL_2^{RC,5D}$ is a very good CPRNG which is fit for industrial applications. The pace of generation of random bits can be incredibly high (up to 200 billion random bits per second).

References

1. Delahaye, J.-P.: Cryptocurrencies and blockchains. Inferences **2**, 4 (2016)
2. Menezes, A.J., Van Oorschot, P.C.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1996)
3. Matthews, R.: On the derivation of chaotic encryption algorithm. Cryptologia **13**(1), 29–42 (1989)
4. Lozi, R., Cherrier, E.: Noise-resisting ciphering based on a chaotic multi-stream pseudorandom number generator. In: Proceedings of the 2011 International Conference for Internet Technology and Secured Transactions (ICITST), Abu Dhabi, pp. 91–96 (2011)
5. Li, C.-Y., Chen, Y.-H., Chang, T.-Y., Deng, L.-Y., Kiwing, T.: Period extension and randomness enhancement using high-throughput reseeding-mixing PRNG. IEEE Trans. Very Large Scale Integr. VLSI Syst. **20**(2), 385–389 (2012)
6. Noura, H., Assad, S.E., Vladeanu, C.: Design of a fast and robust chaos-based cryptosystem for image encryption. In: 8th International Conference on Communications (COMM 2010), pp. 423–426 (2010)
7. Bogoshi, J., Naidoo, K., Webb, J.: The oldest mathematical artifact. Math. Gazette **71**(458), 294 (1987)
8. Smith, D.E.: History of Mathematics, vol. I, pp. 47–49. Dover Publication Inc., New-York (1923)
9. Descartes, R.: Discours de la méthode. La géométrie (1637)
10. Galois, E.: Mémoire sur les conditions de résolubilité des équations par radicaux (mémoire manuscrit de 1830). J. Math Pure et Appl. **10**, 471–433 (1845)
11. Julia, G.: Mémoire sur l'itération des fonctions rationnelles. Journal de mathématiques pures et appliquées **8**(1), 47–246 (1918)
12. Fatou, P.: Sur l'itération des fonctions transcendantes entières. Acta Math. **47**, 337–370 (1926)
13. Gumowski, I., Mira, C.: Recurrence and Discrete Dynamics systems. Lecture Notes in Mathematics. Springer, Berlin (1980)
14. Sharkovskiĭ, A.N.: Coexistence of cycles of a continuous map of the line into itself. Intern. J. Bifurc. Chaos, **5**(5), 1263–1273 (1995). Ukrainian Math. J. **16**, 61–71 (1964). [in Russian]
15. Ikeda, K.: Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system. Opt. Commun. **30**, 257–261 (1979)
16. Ikeda, K., Daido, H., Akimoto, O.: Optical turbulence: chaotic behavior of transmitted light from a ring cavity. Phys. Rev. Lett. **45**, 709–712 (1980)
17. Chua, L.O., Kumoro, M., Matsumoto, T.: The double scroll family. IEEE Trans. Circuit Syst. **32**(11), 1055–1058 (1984)

18. Hénon, M.A.: Two-dimensional mapping with a strange attractor. Commun. Math. Phys. **50**, 69–77 (1976)
19. Lorenz, E.N.: Deterministic nonperiodic flow. J. Atmos. Sci. **20**, 130–141 (1963)
20. May, R.M.: Stability and Complexity of Models Ecosystems. Princeton University Press, Princeton (1973)
21. May, R.: Biological populations with nonoverlapping generations: stable points, stable cycles, and chaos. Sci. New Ser. **186**(4164), 645–647 (1974)
22. Lozi, R.: Giga-periodic orbits for weakly coupled tent and logistic discretized maps. In: Siddiqi, A.H., Duff, I.S., Christensen, O. (eds.) Modern Mathematical Models Methods and Algorithms for Real-World Systems, pp. 80–124. Anamaya Publishers, New Delhi, India (2006)
23. Yuan, G., Yorke, J.A.: Collapsing of chaos in one dimensional maps. Phys. D **136**, 18–30 (2000)
24. Lozi, R.: Can we trust in numerical computations of chaotic solutions of dynamical systems? Topol. Dyn. Chaos, **A**(84), 63–98 (2013). Letellier, C., Gilmore, R. (eds.) World Scientific Series on Nonlinear Sciences
25. Baptista, M.S.: Cryptography with chaos. Phys. Lett. A **240**, 50–54 (1998)
26. Ariffin, M.R.K., Noorani, M.S.M.: Modified baptista type chaotic cryptosystem via matrix secret key. Phys. Lett. A **372**, 5427–5430 (2008)
27. Lanford III, O.E.: Informal remarks on the orbit structure of discrete approximations to chaotic maps. Exp. Math. **7**, 317–324 (1998)
28. Wong, W.K., Lee, L.P., Wong, K.W.: A modified chaotic cryptographic method. In: Communications and Multimedia Security Issues of the New Century, pp. 123–126 (2001)
29. Nejati, H., Beirami, A., Massoud, Y.: A realizable modified tent map for true random number generation. Circuits Syst. MWSCAS **10**, 621–624 (2008)
30. Lozi, R.: Mathematical chaotic circuits: an efficient tool for shaping numerous architectures of mixed chaotic/pseudo random number generator. In: Matoušek, M. (ed.) Proceedings of the Mendel 2014, pp. 163–176 (2014)
31. Lozi, R.: Emergence of randomness from chaos. Int. J. Bifurc. Chaos, **22**(2), 1250021–1/1250021–15 (2012)
32. Rojas, A., Taralova, I., Lozi, R.: New alternate ring-coupled map for multirandom number generation. J. Nonlinear Syst. Appl. **4**(1), 64–69 (2013)
33. Garasym, O., Lozi, R., Taralova, I.: Robust PRNG based on homogeneously distributed chaotic dynamics. J. Phys: Conf. Ser. **692**, 012011 (2016)
34. Jallaouli, O., Assad, S.E., Chetto, M., Lozi, R.: Design and analyses of two stream ciphers based on chaotic coupling and multiplexing techniques. Multimedia tools and applications, 27 pp, 29 June 2017. published online
35. Garasym, O., Taralova, I., Lozi, R.: Application of observer-based chaotic synchronization and identifiability to the original CSK model for secure information transmission. Indian J. Ind. Appl. Math. **6**(1), 1–26 (2015)
36. Farajallah, M., Assad, S.E., Deforges, O.: Fast and secure chaos-based cryptosystem for images. Int. J. Bifurc. Chaos **26**(2), 1650021 (2016)
37. Taralova, I., Lozi, R., Assad, S.E.: Chaotic generator synthesis: dynamical and statistical analysis. In: International IEEE Conference for Internet Technology And Secured Transactions, pp. 56–59 (2012)
38. Garasym, O., Taralova, I., Lozi, R.: New nonlinear CPRNG based on tent and logistic map. In: Jinhu Lü, G.C., Yu, X.Y.W. (eds.) Complex Systems and Networks, Dynamics, Controls and Application, pp. 131–162. Springer, Berlin (2016). Springer: Understanding Complex Systems
39. Garasym, O., Lozi, J.-P., Lozi, R.: How useful randomness for cryptography can emerge from multicore-implemented complex networks of chaotic maps? J. Differ. Equ. Appl., 1–39, February 2017. published online
40. Rukhin, A., Soto, J., Nechvatal, J., Barker, E., Leigh, S., Levenson, M., Banks, D., Heckert, A., Dray, J., Vo, S.: Statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST special publication (2010)