CUBIC TRIGONOMETRIC CHAOTIC SYSTEMS FOR HIGH-QUALITY PSEUDO-RANDOM NUMBER GENERATION

Mokhtar Ouamri
Department of Physics
Faculty of Matter Sciences
University of Tiaret
Algeria
mokhtar.ouamri@univ-tiaret.dz

Article history:
Received 23.06.2025, Accepted 19.09.2025

Abstract

A new two-dimensional chaotic system, called the Cubic Trigonometric Coupled Map (CTCM), is proposed in this article. It is characterised by strong nonlinear interactions and complex dynamic behaviour. A thorough review of the Lyapunov exponents and bifurcation structures shows that CTCM has a strange attractor with fractal geometry and is highly sensitive to initial conditions. In addition, a new pseudo-random number generator (PRNG) based on the chaotic characteristics of the CTCM has been implemented. Experimental results show that the generator achieves near-optimal entropy levels up to 7.999 with high sensitivity to initial conditions, making this PRNG a promising solution for applications in information security, numerical modeling, stochastic simulations, data encryption, and secure communication systems.

Key words

Cubic Trigonometric Coupled Map, Chaos, Lyapunov Exponent, Bifurcation, Entropy, Pseudo-Random Number Generator .

1 Introduction

In Chaos [Broer and Takens, 2011], far from being a simple disorder, embodies a form of dynamic complexity of fascinating richness, mixing mathematical rigor and seemingly unpredictable behavior. This duality makes chaos a central field of research in applied mathematics, physics, biology, cybernetical neuroscience [Babich et al., 2025], cryptography, and artificial intelligence [Strogatz, 2024]. Chaotic maps, such as the logistic map [May, 1976], the Ikeda map [Ikeda, 1979], or the Clifford map [Sprott, 1993], perfectly illustrate this underlying beauty: from simple deterministic equations, they generate trajectories highly sensitive to initial conditions, leading to behavior that is difficult to predict in the long term. This property of exponential sensitivity, combined with a dense distribution in the state space, gives chaotic systems a natural pseudo-randomness exploitable to simulate randomness. Recent advances even extend this connection to modern machine learning approaches, where chaotic dynamics are used to enhance prediction of extreme events in complex time series [Gromov et al., 2024]. Thus, chaos becomes a conceptual bridge between determinism and uncertainty, providing a valuable source for the generation of complex sequences, while revealing the order hidden behind the apparent unpredictability of the real world.

A pseudo-random number generator (PRNG) [Gutbrod, 1999] is an essential element in many areas of scientific, technical and industrial research. It allows to produce sequences of numbers that simulate chance, while being generated in a deterministic way. Several PRNG families [Bhattacharjee and Das, 2022] are distinguished according to their generation principle: the PRNG based on chaotic maps, which exploit the dynamic properties of deterministic chaos; the PRNG based on polynomial equations, which are based on non-linear algebraic relations; material PRNGs, which use physical phenomena as a source of entropy; nature-inspired PRNGs, which imitate biological or evolutionary mechanisms; and PRNG based on cryptographic algorithms, designed to resist attacks and ensure unpredictability. Among these approaches, chaotic PRNGs attract particular attention because of their ability to produce complex sequences, sensitive to initial conditions, and difficult to predict, what makes them interesting for security or nonlinear modeling applications.

The integration of strange attractors and fractal structures [Ruelle, 2006] in the design of PRNGs represents a significant step forward in the quest for highly complex and unpredictable numerical sequences. The strange attractors, typical products of chaotic systems, are distinguished by their non-linear dynamic behavior, unpredictable but nevertheless deterministic. Their fractal geometry (infinitely complex at all scales) allows to generate sequences that have a very high entropy and low internal correlation, which is essential for applications requiring a high level of pseudo-randomness.

We present in this article a new two-dimensional chaotic system called the Cubic Trigonometric Coupled Map (CTCM), inspired by the famous Clifford attractor. The proposed model is then used in the design of a pseudo-random number generator that presents a complex, difficult to predict and weakly correlated behavior.

The remainder of the article is organized as follows: Section 2 presents the Cubic Trigonometric Coupled Map (CTCM) model and discusses its chaotic properties. Section 3 introduces the proposed Pseudorandom Number Generator (PRNG) based on the CTCM. Section 4 provides a comprehensive performance analysis of the proposed PRNG. Finally, Section 5 concludes the paper by summarizing the key findings and outlining potential directions for future research.

2 Cubic Trigonometric Coupled Map (CTCM)

2.1 Definition of the CTCM Model

The two-dimensionnal Cubic Trigonometric Coupled Map (CTCM) is defined by the following equations:

$$\begin{cases} x_{n+1} = a \cdot \cos^3(r \cdot y_n) + b \cdot \cos^3(r \cdot x_n), \\ y_{n+1} = b \cdot (\sin(r \cdot y_n) + a \cdot \sin(r \cdot x_n)) \, . \end{cases} \tag{1}$$

The use of cubic cosine terms in this model plays a central role in shaping its nonlinear behavior. These terms amplify the oscillatory dynamics and contribute to the emergence of intricate trajectories, making the system particularly rich in complex patterns. In equation (1), x_n and y_n correspond to the system states controlled by the three parameters a, b and r respectively. The first two control the shape of the attractor, while r determines the frequency of trajectory oscillations generated by the system.

2.2 Sensitivity Analysis of CTCM model via Lyapunov Exponents

In this section, we analyze the sensitivity of CTCM model to initial conditions with respect to the parameters r, a, and b by studying the spectrum of Lyapunov exponents [Wolf et al., 1985]. The Lyapunov exponent quantifies the average exponential rate of divergence of neighboring trajectories. A positive value indicates the likely presence of chaos. The maximal Lyapunov exponent is given by:

$$\lambda = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \ln \|J_n \cdot \vec{v}_n\| \, , \tag{2}$$

where J_n is the Jacobian matrix of the map evaluated at the point (x_n, y_n),

$$J_n = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{pmatrix}_{(x_n, y_n)} . \tag{3}$$

The vector $\vec{v}_n$ is a unit vector representing the direction of an infinitesimal perturbation, and it evolves according to

$$\vec{v}_{n+1} = \frac{J_n \cdot \vec{v}_n}{\|J_n \cdot \vec{v}_n\|}. \tag{4}$$

Figure 1 shows the Lyapunov exponent as a function of r for fixed parameters $a = 1.9$ and $b = 4$. For values of r less than 0.6, the Lyapunov exponent is negative, indicating a stable and predictable behavior of the system. Between $r = 1$ and approximately $r = 1.3$, the Lyapunov exponent oscillates between positive and negative values, revealing an alternating behavior between stability and chaos. Starting from $r = 1.2$, the Lyapunov exponent becomes clearly positive, indicating a pronounced chaotic regime in which small initial variations significantly influence the long-term evolution of the system.

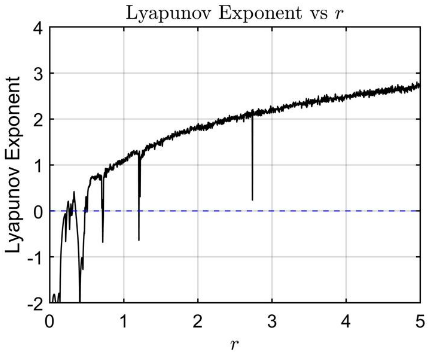

Figure 1. Largest Lyapunov exponent of the CTCM as a function of the control parameter r for fixed parameters $a = 1$ and $b = 2$.

In a second experiment, we fix $r = 4$. Figure 2 displays the Lyapunov exponent as a function of parameters a and b. It is observed that the parameter b strongly influences the system's dynamics. Furthermore, a large chaotic region appears for values of a between 0 and 4 and b between 0.8 and 4. When $b < 0.8$, stability zones are present near $a \approx 2$ and b close to zero. Transition zones between stability and chaos, characterized by rapid variations in the Lyapunov exponent, are also observed.

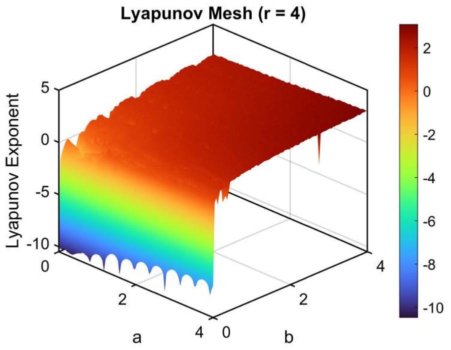

Figure 2. The largest Lyapunov exponent of the CTCM as a function of the parameters a and b with the control parameter fixed at $r = 4$.

2.3 Bifurcation Diagram of CTCM model

We explored in this section how the system's dynamics evolve as we vary the parameter r, keeping $a = 1.9$ and $b = 4$ fixed (See figure 3 and 4).

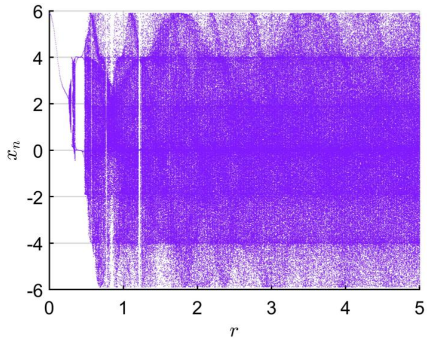

Figure 3. Bifurcation diagram of the x–component of the CTCM as a function of the control parameter r; the parameters a and b are kept fixed.

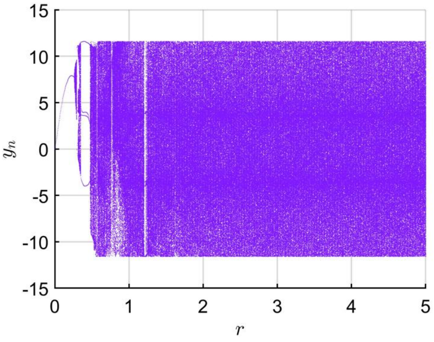

Figure 4. Bifurcation diagram of the y–component of the CTCM as a function of the control parameter r; the parameters a and b are kept fixed.

At low values of r, the system behaves periodically, showing a few stable points that indicate regular and predictable motion. As r increases past about 0.6, the system starts experiencing period-doubling bifurcations, signaling a transition to more complex and less predictable behavior. Beyond about $r = 1.2$, the system enters a chaotic regime. Here, the values of y_n spread out widely, reflecting a strong sensitivity to initial conditions and irregular oscillations. We also notice vertical gaps in the diagram, which likely correspond to parameter ranges where the system's trajectories do not settle down, possibly due to transient effects or different types of bifurcations.

2.4 Phase diagram

In this section, system dynamics and complexity will be analyzed in terms of phase diagrams. For this purpose, we have generated three invariant sets $\{(x_n, y_n)\}$ using CTCM model from the initial conditions $x_0 = 0.1$ and $y_0 = 0.5$ as is illustrated in the figures 5, 6, 7 and 8 respectively.

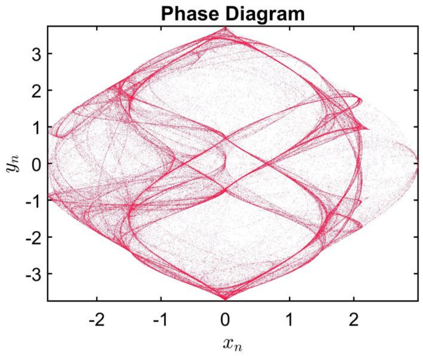

Figure 5. Phase diagram for $a = b = r = 1.5$.

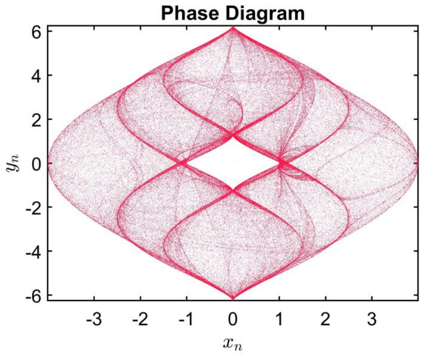

Figure 6. Phase diagram for $a = 1.5, b = 2.5, r = 2.5$.

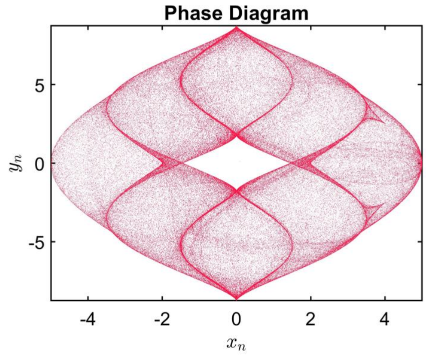

Figure 7. Phase diagram for $a = 1.5, b = 3.5, r = 3.5$.

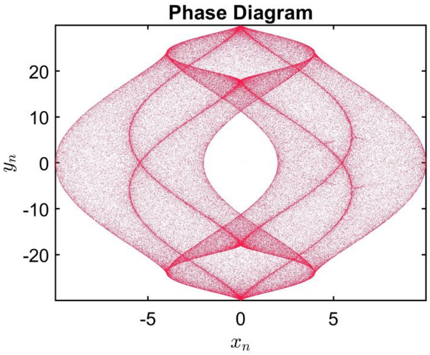

Figure 8. Phase diagram for $a = 4, b = 6, r = 6$.

We can clearly see that the point cloud forms a strange fractal attractor, whose shape resembles a double diamond-eye structure. All initial conditions tested within a large rectangle are irreversibly attracted to this set, suggesting that it constitutes the system's global attractor. Also, we notice that the orbits remain confined in a bounded region of the phase space, which allows us to say that the attractor acts as a basin of attraction where all the trajectories converge asymptotically.

2.5 Complexity of CTCM

The geometric complexity of the attractor generated by our dynamic system was better comprehended by us by using the box-counting method to calculate its fractal dimension [Falconer, 2013]. A two-dimensional strange attractor typically exhibits a non-integer fractal dimension between 1 and 2. Box-counting consists in overlapping the normalized attractor with square grids of decreasing size, denoted ε, and counting the number of boxes $N(\varepsilon)$ that contain at least one point. By plotting $\log N(\varepsilon)$ against $\log(1/\varepsilon)$, we observe an almost linear trend. consequently, the fractal dimension becomes the slope of the regression line Figure 9 shows the relationship between the number of boxes $N(\varepsilon)$ and the size of boxes $1/\varepsilon$ in a fractal analysis by counting boxes (box-counting) from the slope of the linear regression on the log-log graph. In this case, we got a slope ≈ 1.7962 which means obviously that the attractor has a fractal structure with a Hausdorff dimension close to 1.8.

3 Pseudo-Random Number Generator (PRNG) Based on the CTCM Chaotic Map

The complexity and unpredictability of the strange attractor of CTCM model offers a great chance for the conception of a robust pseudo random number generator. For this purpose, our proposed approach relies on a simple yet effective comparison-based method, as illustrated in Algorithm 1. First, each state is multiplied by 10^3. Then, the resulting value is reduced modulo 2 in order to extract its decimal component within the interval $[0, 2)$. This operation emphasizes the chaotic fluctuations found in the least significant digits of the original sequences. The resulting variables, `temp1` and `temp2`, are then compared: if `temp1` $<$ `temp2`, the corresponding output bit is set to 1; otherwise, it is set to 0. The binary sequence `binary_sequence` thus generated inherits the unpredictability and sensitivity to initial conditions that characterize CTCM chaotic system.

4 The performance analysis of the proposed PRNG

4.1 Irreversibility of the Proposed PRNG

The section discusses the concept of reversibility in pseudorandom number generators (PRNGs) and emphasises the importance of being non-reversible to prevent reconstructing previous states from current states. The presence of non-linear trigonometric functions like sin and cos^3 means that multiple values are possible for the next states, which destroys the uniqueness needed for reversibility. Moreover, numerical implementations of finite precision arithmetic introduce errors that build up over time. This makes it practically impossible to reliably recover past states, even in systems that are theoretically invertible.

4.2 Key space

The security strength of the chaotic CTCM-based PRNG is closely linked to the size of its key space. In our case, the secret key is composed of five parameters:

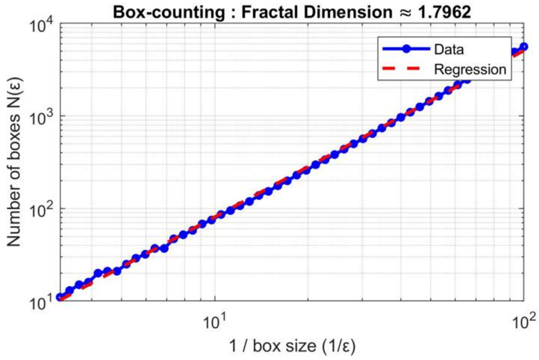

Figure 9. Box-counting method applied to the attractor showing fractal structure.

Algorithm 1: Generation of a Binary Sequence from Chaotic Sequences x and y

```
Input: Two real-valued sequences
       $x = \{x_1, x_2, \ldots, x_N\}$,
       $y = \{y_1, y_2, \ldots, y_N\}$
Output: A binary sequence binary_sequence
        of length $N$
Initialize counter $c \leftarrow 0$;
for $i \leftarrow 1$ to $N$ do
    Compute temp1 $\leftarrow \mathrm{mod}(x_i \cdot 10^3, 2)$;
    Compute temp2 $\leftarrow \mathrm{mod}(y_i \cdot 10^3, 2)$;
    if temp1 $<$ temp2 then
        Set binary_sequence$[c + 1] \leftarrow 1$;
    else
        Set binary_sequence$[c + 1] \leftarrow 0$;
    Increment $c \leftarrow c + 1$;
```

the initial conditions x_0 and y_0, and the control parameters r, a and b. The values x_0 and y_0 are real numbers in the interval $[0, 1]$, while r, a and b are real values chosen from $[0, 8]$. Assuming a floating-point precision of 10^{-15}, each parameter contributes around 50 bits of entropy. The result is a total key space comprising over 2^{250} possible combinations, or more than 2^{100}, which is generally considered sufficient to resist brute-force attacks [Knudsen and Robshaw, 2011]. Its large key space, combined with its high sensitivity to initial conditions, makes it a promising candidate for cryptographic and security applications.

4.3 NIST statistical test analysis

The NIST Statistical Test Suite [Bassham et al., 2010] offers a complete set of empirical tests that evaluate the randomness of binary sequences produced by cryptographic or pseudo-random number generators. This suite consists of fifteen statistical tests [Zaman and Ghosh, 2012], such as the frequency test, runs test, approximate entropy, and linear complexity test, each targeting a specific characteristic expected from a truly random sequence. When applied to a binary sequence, the tests yield p-values that indicate whether the sequence behaves similarly to an ideal random process. A generator is considered statistically sound if the p-values meet the NIST-recommended significance level (commonly 0.01) across all tests. The NIST test suite is widely regarded as a standard benchmark to assess the quality, unpredictability, and security of random number generators, including those based on chaotic systems. To carry out the NIST assessment, we first generated two binary sequences of length 100×10^6 bits with the CTCM-based PRNG. Each original stream was subsequently divided into 100 subsequences of length 10^6 bits in order to match the input requirements of the test suite. The generation parameters were set to $(a = 1.9), (b = 4)$ and $(r = 4)$. For the first sequence, the initial condition was $(x_0, y_0) = (0.9, 0.2)$. For the second sequence, we used the slightly disturbed initial condition $(x_0, y_0) = (0.9 + 10^{-15}, 0.2)$. Table 1 displays the results of the statistical tests conducted on both sequences, highlighting noticeable similarities and differences. Nevertheless, both sequences successfully pass the tests. Moreover, the proposed PRNG exhibits high sensitivity to even small key changes, which is an important property to consider.

4.4 Entropy of PRNG analysis

In this section, we evaluated the randomness and unpredictability of the PRNG sequence $x = \{x_1, x_2, \ldots, x_n\}$ using the Shannon entropy, given by:

$$H(x) = -\sum_{a \in \mathcal{A}} P(a) \log_2 P(a) \tag{5}$$

where $P(a)$ denotes the empirical probability of the symbol $a \in \mathcal{A}$ in the sequence x, and is calculated as follows:

$$P(a) = \frac{1}{n} \sum_{i=1}^{n} \delta(x_i, a) \tag{6}$$

Here, $\delta(x_i, a)$ is the Kronecker delta function, which is equal to 1 if $x_i = a$, and 0 otherwise. Since our PRNG is binary and generates only the symbols 0 and 1 then the closer the entropy result is to 1, the more random and unpredictable the result.

The table 2 shows the computed entropy values for a pseudo-random number generator (PRNG) based on the sequence length. We notice that the entropy is of order 0.9985 when length $>$ 1000, however, when the length increases, the entropy approaches 1. For example, at 3501 bits, the entropy is almost maximum (0,99999999994703), which strongly indicates that the PRNG produces an almost perfectly balanced and unpredictable binary sequence. In another experiment, we converted the binary sequence into a sequence of bytes by grouping each block of 8 consecutive bits into a byte. This transformation allows for a more relevant analysis of entropy. A pseudorandom number generator (PRNG) is considered of good quality if the entropy of the resulting sequence is close to 8, indicating a high degree of unpredictability. The experiment was conducted on a binary sequence of size 4×2^{20} bits to evaluate the statistical quality of the proposed PRNG. As can be seen in the table 3, the analysis of the entropy values of the binary sequences discussed above shows very good results for long binary sequences of any length: for a sequence of 131,073 bytes, the applied entropy is 7.9986, that is to say the unpredictability is minimal and appreciated that it is relatively close to the maximum entropy of 8. In this way as the sequence becomes longer, the entropy increases slightly: at 262,145 bytes, it is a 7.9992 and at 393,217 bytes, it measures 7.9995 So in increasing with the long sequence to perform the two previous

Table 1. NIST statistical test results for two binary sequences.

Test Name	**Sequence 1**		**Sequence 2**	
	p**-value**	**Result**	p**-value**	**Result**
Frequency	0.474986	Success	0.455937	Success
Block-Frequency	0.075719	Success	0.759756	Success
Cumulative Sums (1)	0.554420	Success	0.798139	Success
Cumulative Sums (2)	0.759756	Success	0.616305	Success
Runs	0.514124	Success	0.455937	Success
Longest Run	0.955835	Success	0.678686	Success
Rank	0.096578	Success	0.304126	Success
FFT	0.798139	Success	0.514124	Success
Non-Overlapping	0.494392	Success	0.883171	Success
Overlapping	0.102526	Success	0.955835	Success
Universal	0.051942	Success	0.494392	Success
Approximate Entropy	0.334538	Success	0.010237	Success
Random Excursions	0.202268	Success	0.574903	Success
Random Excursions Variant	0.719747	Success	0.455937	Success
Serial (1)	0.616305	Success	0.249284	Success
Serial (2)	0.304126	Success	0.401199	Success
Linear Complexity	0.102526	Success	0.924076	Success

Table 2. Entropy values as a function of the binary sequence length

Length of sequence (in bits)	**Entropy**
501	0.998962279385191
1001	0.998541696925956
1501	0.999351554826180
2001	0.999142103991909
2501	0.999766457662374
3001	0.999996075276857
3501	0.999999470333038

Table 3. Binary entropy computed for different sequence lengths.

Length of Sequence (in bytes)	**Entropy**
131,073	7.99862540382325
262,145	7.99920045931387
393,217	7.99946709076136

cases, a pseudo-generatorrandom, created leads to the generation of more random behavioural sequences from application-specific entropies. Therefore it is sensible to increase the high-values of entropy with the sequence length for better be-verie the statistical quality way of the PRNG and know that it is reliable for generation in a random information.

4.5 Time processing of the proposed PRNG

As part of evaluating the computational performance of our PRNG, we used MATLAB 2020 on hardware consisting of a RYZEN 7 5700X processor with 16 GB of RAM. In this experiment, we generated 2×10^6 bits several times to determine the average computational time. This latter was found to be approximately 0.25 s, which means that the average rate to generate 1 megabit is about 0.25 s.

4.6 PRNG sensitivity Analysis using Correlation and NBCR metrics

The purpose of this section is to test the sensitivity to initial conditions between the two sequences x and y. We have selected the correlation coefficient and the number of changed bits rate (NBCR) as our two metrics for this purpose. The correlation coefficient quantifies the linear relationship between the two sequences and is calculated as follows:

$$\rho_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\,\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \qquad (7)$$

where $\bar{x}$ and $\bar{y}$ represent the mean values of sequences x and y, respectively. A correlation coefficient ρ_{xy} close to zero indicates that the two sequences are statistically uncorrelated and largely independent in terms of linear dependency.

The Normalized Bit Change Rate (NBCR) between two binary sequences $x = \{x_1, x_2, \ldots, x_n\}$ and $y = \{y_1, y_2, \ldots, y_n\}$ is defined as:

$$\text{NBCR}(x, y) = \frac{1}{n}\sum_{i=1}^{n} |x_i - y_i| \tag{8}$$

Where n represents the length of the sequences, and x_i and y_i are the i-th bits of sequences x and y, respectively. Since the bits are binary ($x_i, y_i \in \{0, 1\}$), the absolute difference $|x_i - y_i|$ equals 1 if the bits differ and 0 if they are equal. The closer NBCR is to 50%, the better we can qualify the PRNG. In order to make this experiment successful, we produced two binary sequences with an initial length of 5×10^6 bits that differ solely in their initial conditions: (x_0, y_0) for the first sequence, and $(x_0, y_0 + 10^{-15})$ for the second sequence. The table 4 shows the correlation coefficient ρ and the Normalized Bit Change Rate (NBCR) between two binary sequences generated with slightly different initial conditions. The first sequence starts with $x_0 = 0.9$ and $y_0 = 0.2$, while the second sequence has a small variation in x_0, specifically $x_0 = 0.9 + 10 - 15$.

Table 4. Correlation and NBCR between two binary sequences with slightly different initial conditions. First sequence: $x_0 = 0.9$, $y_0 = 0.2$; second sequence: $x_0 = 0.9 + 10^{-15}$, $y_0 = 0.2$.

Parameters (a; b; r)	ρ	NBCR
$a = 1.9;\ b = 4;\ r = 4$	-0.00067	50.03%
$a = 2.5;\ b = 1;\ r = 3$	-0.00018	50.01%
$a = 1.5;\ b = 1;\ r = 2$	-0.00017	50.00%
$a = 4;\ b = 4;\ r = 1.6$	-0.00051	50.03%

We observe that for the four sets of parameters, the correlation coefficients remain very close to zero, ranging from (-0.00017) to (-0.00067). This indicates a strong decorrelation, meaning there is no significant relationship between the two sequences. In addition, the NBCR values are remarkably close to the ideal value of 50%, which is a strong indicator of the sensitivity of the proposed PRNG to small variations in initial conditions.

4.7 Autocorrelation Analysis

In this section, we have evaluated the autocorrelation of the proposed PRNG defined by the following formula:

$$R(k) = \frac{\sum_{t=1}^{N-k}(x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^{N}(x_t - \bar{x})^2}, \tag{9}$$

it qualifies the linear dependence between the values of a time series x_t separated by a lag k, where $\bar{x}$ represents the mean of the series. The figure 10 shows autocorrelation results as a function of k lag values. We observe a rapid decrease in the autocorrelation value as a function of time lag. This behavior indicates that the successive values produced by this PRNG are very weakly correlated with each other. This property of statistical independence of the elements of the generated sequence is a desirable feature for a random generator, as it allows a good quality of randomness to be achieved. and security.

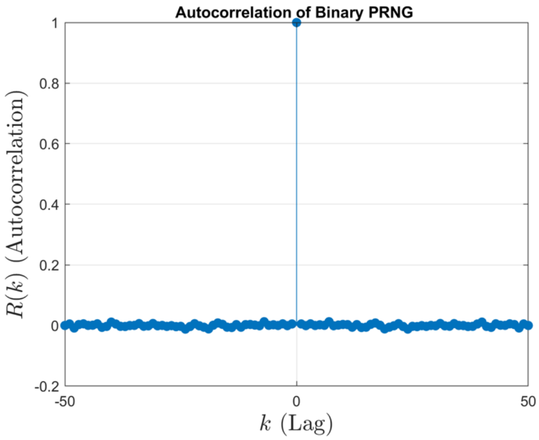

Figure 10. Autocorrelation of the binary sequence of the proposed PRNG.

4.8 Comparaison study

We compared our proposed approach with some existing PRNG algorithms from the literature that reported common evaluation metrics such as entropy, Normalized Bit Change Rate (NBCR), and correlation coefficient. This allowed us to perform a fair and consistent comparison based on standard statistical and security indicators.

The results presented in the table 5 show that our approach stands out advantageously on several key aspects. With a high entropy of 7.9994 bits per byte and a very low correlation coefficient of -0.00018, our PRNG algorithm appears to offer a highly competitive level of statistical quality. This objective comparative analysis allows us to position our solution as an interesting alternative to the existing algorithms, with promising performance in terms of robustness

5 Conclusion

In this study, we demonstrated that the CTCM model possesses rich chaotic dynamics, characterized by high sensitivity to initial conditions and the presence of fractal attractors. Taking advantage of these properties, we have consequently designed a PRNG capable of generating highly unpredictable and weakly correlated sequences, with an NBCR value close to 50%. These promising results confirm the relevance of our approach for cryptographic applications, in particular for key generation and the application of secure protocols. Furthermore, although the potential of the proposed model for image encryption and secure visualization is promising, further

Table 5. Comparison of pseudo-random number generator (PRNG) algorithms based on key space, entropy, NBCR, and correlation coefficient.

PRNG Algorithm	Key Space	Entropy	NBCR	Correlation Coefficient
Proposed	2^{250}	7.9994	50.01	-0.00018
Ref. ([Agarwal, 2021])	2^{320}	7.9864	49.97	0.0016
Ref. ([Zhao et al., 2019])	2^{70}	7.9896	49.74	None
Ref. ([Barani et al., 2020])	2^{588}	7.9937	50.13	0.0003
Ref. ([Wang and Cheng, 2019])	$\sim 2^{82}$	7.9692	51.92	None

dedicated experiments and evaluations are still needed. These orientations will form the core of our future work to fully validate these applications.

References

Agarwal, S. (2021). Designing a pseudo-random bit generator using generalized cascade fractal function. *Chaos Theory and Applications*, **3** (1), pp. 11–19.

Babich, N., Chen, O., Chulkin, V., Marzel, E., Rybalko, A., and Fradkov, A. (2025). Outline of cybernetical neuroscience. *Cybernetics and Physics*, **14** (1), pp. 13–18.

Barani, M. J., Ayubi, P., Valandar, M. Y., and Irani, B. Y. (2020). A new pseudo-random number generator based on generalized newton complex map with dynamic key. *Journal of Information Security and Applications*, **53**, pp. 102509.

Bassham, L. E., Rukhin, A. L., Soto, J., Nechvatal, J. R., Smid, M. E., Leigh, S. D., Levenson, M., Vangel, M., Heckert, N. A., and Banks, D. L. (2010). A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, NIST.

Bhattacharjee, K. and Das, S. (2022). A search for good pseudo-random number generators: Survey and empirical studies. *Computer Science Review*, **45**, pp. 100471.

Broer, H. W. and Takens, F. (2011). *Dynamical Systems and Chaos*, vol. 172. Springer.

Falconer, K. (2013). *Fractal geometry: mathematical foundations and applications*. John Wiley & Sons.

Gromov, N., Smirnov, L., and Levanova, T. (2024). Prediction of extreme events and chaotic dynamics using wavenet. *Cybernetics and Physics*, **13** (1), pp. 20–31.

Gutbrod, F. (1999). New trends in pseudo-random number generation. *Annual Reviews of Computational Physics VI*, pp. 203–257.

Ikeda, K. (1979). Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system. *Optics Communications*, **30** (2), pp. 257–261.

Knudsen, L. R. and Robshaw, M. J. B. (2011). Brute force attacks. In *The Block Cipher Companion*, pp. 95–108. Springer.

May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, **261** (5560), pp. 459–467.

Ruelle, D. (2006). What is a strange attractor. *Notices of the AMS*, **53** (7), pp. 764–765.

Sprott, J. C. (1993). *Strange Attractors: Creating Patterns in Chaos*. M & T Books, Div. of MIS: Press Inc., 115 West 18th Street, New York, NY, United States.

Strogatz, S. H. (2024). *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Chapman and Hall/CRC.

Wang, L. and Cheng, H. (2019). Pseudo-random number generator based on logistic chaotic system. *Entropy*, **21** (10), pp. 960.

Wolf, A., Swift, J. B., Swinney, H. L., and Vastano, J. A. (1985). Determining lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*, **16** (3), pp. 285–317.

Zaman, J. K. M. S. and Ghosh, R. (2012). Review on fifteen statistical tests proposed by nist. *Journal of Theoretical Physics and Cryptography*, **1**, pp. 18–31.

Zhao, Y., Gao, C., Liu, J., and Dong, S. (2019). A self-perturbed pseudo-random sequence generator based on hyperchaos. *Chaos, Solitons & Fractals: X*, **4**, pp. 100023.