

Dynamics of the Hénon Map in the Digital Domain

Zbigniew Galias¹, Senior Member, IEEE

Abstract—Nonlinear maps are usually implemented using finite-precision floating-point formats both in practical applications and in theoretical investigations. In the digital domain, the size of the state space is finite and every trajectory after a finite number of iterations reaches a cycle. It is therefore important to study the influence of rounding errors and the finiteness of the state space on properties of nonlinear maps such as the number of cycles, their periods, sizes of basins of attraction of cycles, and average convergence times. In this work, a thorough analysis of the dynamics of finite-precision implementations of the Hénon map is carried out. Six computational formulas and three popular finite-precision floating-point formats are considered. An exhaustive search is performed to find all cycles existing for single-precision floating-point implementations. Interval methods are used to reduce the number of initial conditions that must be considered. An efficient graph-based algorithm is designed to find basins of attraction. For the double-precision and extended-precision implementations, statistical methods are utilized to find cycles and to estimate sizes of their basins of attraction. Properties of observed cycles and corresponding dynamical phenomena are thoroughly discussed.

Index Terms—Hénon map, finite-precision computations, cycle, basin of attraction.

I. INTRODUCTION

NUMERICAL simulations are one of the main tools used to study nonlinear maps. Nonlinear maps implemented in the digital domain are used in various applications, including pseudo-random number generators [1], [2], [3], [4], [5], secure communication [6], image encryption [7], [8], and chaotic cryptography [9], [10], [11]. Finite-precision implementations change dynamical properties of nonlinear maps and may promote various phenomena including the existence of short periodic attractors which are undesirable in practical applications. Hence, it is important to investigate the influence of computational accuracy and the computational formula used on properties of nonlinear maps.

Various methods to improve properties of chaotic maps implemented in the digital domain have been proposed. This includes methods to increase the length of generated trajectories before repeating itself and methods to improve randomness of generated trajectories. A varying parameter compensation method is introduced in [1]. A reseeding-mixing method to extend periods of observed cycles and to improve

the statistical properties of generated trajectories is carried out in [12]. Dynamic parameter-control chaotic system framework in which output of one chaotic map is used to control the parameter of another chaotic map, is introduced in [3]. Cascade chaotic systems are used to improve the randomness and security level of pseudo-random number generators in [13]. A congruential generator based method is proposed to counteract the performance degradation of digital chaos in [14].

When a nonlinear map is implemented in the digital domain the state space is finite. As a consequence, each trajectory, after a finite number of steps, must visit a point in the state space that was visited before. From then on, the trajectory remains periodic. The part of a trajectory before reaching the steady-state periodic solution is called a transient. Assuming that the size of the state space is M , the maximum length of a periodic trajectory that may be observed is M . In practice, observed trajectories have much shorter periods. The eventual periodicity of all trajectories has important consequences from both the theoretical and practical points of view. The problem is especially important for non-linear maps that exhibit chaotic behavior. Under certain assumptions, a chaotic attractor contains an infinite number of unstable periodic orbits. The measure of the set of unstable periodic orbits is zero, which means that there is a zero probability of selecting an initial point with periodic behavior. An arbitrarily small deviation from the position of an unstable periodic orbit results in the divergence of the trajectory from this orbit. However, when calculations are carried out in finite precision, it may happen that some of these unstable periodic orbits could actually be observed in simulations. This is caused by a nonzero probability that a trajectory initiated close to the position of an unstable period- p orbit returns to the initial point after a number of iterations, which is equal to p or its low multiple.

The key properties of chaotic map implementations are the lengths of observed cycles and the statistical properties of generated trajectories. The degradation of dynamical properties of nonlinear maps caused by finite-precision implementations has been studied by many researchers. The influence of rounding errors on dynamical behaviors of the logistic map is studied in [15]. The authors study the length of the longest limit cycle and the average transient length as a function of the size of the state space. It is shown that for the logistic map, in simulations, one observes cycles of length of the order \sqrt{M} , where M is the size of the state space. It is argued that although periodic trajectories are relatively short, different types of behavior are correctly recognized in numerical simulations, and that certain statistical properties of observed trajectories, such as Lyapunov exponents, are preserved. The relation between the expected period of observed orbits and the maximum rounding error for the Ikeda map is studied in [16]. Dynamical degradation of

Manuscript received 23 May 2022; revised 31 August 2022, 20 September 2022, and 18 October 2022; accepted 22 October 2022. Date of publication 31 October 2022; date of current version 25 January 2023. This work was supported by the AGH University of Science and Technology. This article was recommended by Associate Editor W. X. Zheng.

The author is with the Department of Electrical Engineering, AGH University of Science and Technology, 30-059 Kraków, Poland (e-mail: galias@agh.edu.pl).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TCSI.2022.3217139>.

Digital Object Identifier 10.1109/TCSI.2022.3217139

piecewise linear chaotic maps caused by finite-precision implementations is studied in [17]. Statistical study of the influence of double-precision errors on the behavior of the logistic map is carried out in [18]. The influence of the rounding method on properties of trajectories generated by the logistic map is studied in [19]. Five rounding modes are considered, and it is shown that periods of cycles and transient lengths depend on the rounding mode. In [9], fixed-point arithmetic is used to study cycle lengths of finite-precision implementation of the logistic map and the influence of correlation properties on the performance of chaos-based cryptosystems is discussed.

Properties of finite-precision implementations of nonlinear maps may be studied using the idea of state transition diagrams [15]. A similar approach under the name of state-mapping networks is used in [20] to analyze dynamical behaviors of the logistic map and the tent map implemented using fixed-point arithmetic. Analysis of dynamical properties of the logistic map implemented with up to 16 bits of precision is carried out in [14]. Dynamical properties of the logistic map implemented using single-precision and double-precision computations are analyzed in [21]. The graph-based approach is used in [22] to analyze the dynamics of the logistic map.

In principle, to find all cycles existing in a finite-precision implementation of a nonlinear map one should compute trajectories initiated in all points belonging to the state space. When the size of the state space is small and all states can be stored in a computer memory, then a complete characterization of cycles and their basins of attraction can be carried out using the graph-based approach [14], [15], [20], [21]. In this approach, one selects a point in the state space and computes its trajectory until a cycle is found or a point visited before is reached. After all points are visited, the graph structure is known and cycles and their basins of attraction can be easily identified. This approach has been used to study the logistic map and the tent map implemented using fixed-point arithmetic and low precision floating-point arithmetic. Similar calculations have been carried out for the single-precision floating-point implementation of the logistic map in [21]. To find all cycles in larger state spaces one may use a trajectory-based approach, where the exhaustive search is carried out in a reduced search space. This method has been applied to the logistic map implemented in double-precision (compare [21]). Most of the results available in the literature involve nonlinear maps of dimension one and low-precision accuracy. For higher dimensional maps, a complete characterization of the dynamics of the map implemented in the digital domain is a challenging research problem even for the single-precision accuracy.

In this work, we develop methods which may be used to study higher-dimensional maps implemented using floating-point formats. As an example, we study dynamical properties of Hénon map [23]—a well-known example of a chaotic two-dimensional map—implemented using three popular binary floating-point formats: the single-precision (binary32) format, the double-precision (binary64) format, and the extended-precision format. Two selections of parameter values are considered: the classical parameter values for which the famous chaotic attractor is observed in simulations and

parameter values for which a stable period-28 orbit exists. Due to the size of the state space, one cannot use methods developed for one-dimensional systems to analyse the Hénon map for any of the considered floating point formats. For the single-precision format, a new method is proposed to find all cycles existing in the system and to construct basins of attraction in case of small basins. For the double-precision and the extended-precision format, statistical approach is used to find cycles and sizes of their basins of attraction. Properties of observed trajectories are discussed and it is verified, whether observations on the number of cycles and their average length reported in the literature [15], [16], [20] are valid for the Hénon map.

The main contribution and the novelty of this work is the development of methods to study the dynamics of nonlinear maps in the digital domain and carrying out analysis of the Hénon map implemented using different computational formulas in most popular floating point formats.

All computer programs are developed in C++ language and compiled using the g++ compiler, version 9.4.0. For the interval arithmetic support the CAPD library [24] is used. Computation times are reported for a single core 3.4 GHz processor. Parallel computations are used to reduce the wall computation time. The C++ code implementing algorithms presented in this work and the data necessary to carry out the computations are available at <http://www.zet.agh.edu.pl/henon/tcasi2023>.

The layout of the remaining part of the paper is as follows. In Section II, the definition of the Hénon map is recalled and its finite precision implementations are described. Computational tools for the analysis of finite-precision implementations of nonlinear maps are presented in Section III. Analysis of finite-precision implementations of the Hénon map for two sets of parameter values is carried out in Section IV. Conclusions are presented in Section V.

II. THE HÉNON MAP IN FINITE PRECISION

The *Hénon map* [23] is a two-parameter map of a plane defined by

$$h(x, y) = (1 + y - ax^2, bx). \quad (1)$$

The linear change of coordinates $(x, y) \mapsto (\bar{y}, b\bar{x})$ converts (1) to

$$h(x, y) = (y, 1 + bx - ay^2). \quad (2)$$

Digital implementations of maps (1) and (2) are equivalent, provided that the order of elementary arithmetic operations is the same. In the following, we use the latter version because it permits a more efficient search algorithm for cycles.

We consider six computational formulas for equation (2):

$$h(x, y) = (y, 1 + bx - yya), \quad (3a)$$

$$h(x, y) = (y, 1 + bx - ayy), \quad (3b)$$

$$h(x, y) = (y, bx - ayy + 1), \quad (3c)$$

$$h(x, y) = (y, bx - yya + 1), \quad (3d)$$

$$h(x, y) = (y, 1 - yya + bx), \quad (3e)$$

$$h(x, y) = (y, 1 - ayy + bx). \quad (3f)$$

TABLE I
PROPERTIES OF FINITE-PRECISION DATA TYPES

	float	double	long double
m_{10}	6	15	18
m_2	24	53	64
ε	2^{-23}	2^{-52}	2^{-63}
t [s]	1.39	1.39	2.91

These formulas are mathematically equivalent. They differ in the order of elementary arithmetic operations in which the final result is evaluated. This may lead (and usually leads) to different final results when computations are performed in finite precision.

In the following, we consider three finite-precision floating-point formats: the single-precision binary floating-point format (binary32, `float` in C++), the double-precision binary floating-point format (binary64, `double` in C++), and the extended-precision format (`long double` in C++). The first two types are defined in the IEEE 754 floating-point computation standard [25] and are commonly used in microprocessors. Some important properties of these representations are collected in Table I: m_{10} is the number of decimal digits of precision for a given data type, m_2 specifies the number of base 2 digits in the mantissa part, and ε is the smallest positive number such that the arithmetic operation $1 + \varepsilon$ carried out using a given data type returns a value larger than 1.

In the last row of Table I computation speeds are compared. For each data type, the time t needed to compute 10^8 iterations of the Hénon map is reported. Computations carried out using the first two data types take the same time. The single-precision format is used primarily to save memory space and not to gain speed. Note that careful implementation of computational formulas in C++ is needed when using the `float` type. In C++ the constants are of `double` type by default and therefore the constant `1.0F` should be used instead of `1.0` in the formulas (3). If the constant `1.0` is used, then all variables are cast to the `double` type, which takes extra time, and the computations are slower. Computations using the `long double` format are more than two times slower than those for the first two formats.

III. ANALYSIS OF NONLINEAR MAPS IN THE DIGITAL DOMAIN

In this section, a set of tools for studying nonlinear maps implemented using finite-precision arithmetic is presented. We describe methods to estimate the size of the state space, compute the inverse of a map, find all cycles, their basins of attraction, and average convergence times (transient lengths). The methods are illustrated using the Hénon map $h: \mathbb{R}^2 \mapsto \mathbb{R}^2$ as an example. After minor adjustments, the methods can be applied to other nonlinear maps.

In the digital domain, the number of representable points in the state space is finite. For an n -dimensional map a representable point is a point in \mathbb{R}^n which can be exactly represented using a selected finite-precision arithmetic. The dynamics of a finite precision implementation of a map can be represented as a graph with the vertices being representable

points and the edges being transitions between these points as defined by the action of the map. The graph has a structure of c connected components, where c is the number of cycles. Each connected component is the basin of attraction of a single cycle. In the characterization of finite-precision implementations of maps, the most important problems are the number of cycles and their properties such as the period, the size of the basin of attraction, and the average convergence time.

The basin of attraction of a given cycle is the set of initial points such that trajectories based at these points after a finite number of iterations reach this cycle. The probability c_{prob} of reaching a given cycle starting from a random initial condition belonging to the state space can be computed as the ratio of the basin size s and the size of the state space M .

The convergence time for a given representable point is defined as the number of iterations needed to reach the steady state. For a point that belongs to a cycle, the convergence time is zero. The average convergence time t_{aver} and the maximum convergence time t_{max} for a given cycle are defined as the average and maximum convergence times calculated over the whole basin of attraction of this cycle. The average (maximum) length of a trajectory before repeating itself can be computed as the sum of the period p of the steady-state cycle and the average (maximum) convergence time t_{aver} (t_{max}).

A. Size of the State Space

Let us estimate the size M of the state space for finite-precision implementations of the Hénon map. Observed trajectories are enclosed in the rectangle $[-1.5, 1.5] \times [-1.5, 1.5]$ (see Figs. 1 and 2). The number of representable points (x, y) belonging to this rectangle is approximately $4.6 \cdot 10^{18}$ and $8.5 \cdot 10^{37}$ when single- and double-precision representations are used.

The size of the state space can be reduced using interval-arithmetic-based tools. First, the region of interest $[-1.5, 1.5] \times [-1.5, 1.5]$ is covered by boxes (two-dimensional interval vectors) of the form $\mathbf{v}_k = [k_x \delta, (k_x + 1) \delta] \times [k_y \delta, (k_y + 1) \delta]$, where k_x and k_y are integers and $\delta > 0$ is a real number. For a fixed δ each box \mathbf{v}_k is uniquely defined by integers k_x and k_y . Next, for each box \mathbf{v}_k an enclosure of its image is found using interval arithmetic computations. The result obtained is an interval vector containing images of all points belonging to the box \mathbf{v}_k . Boxes \mathbf{v}_j which have non-empty intersection with the enclosure are identified. This process creates a graph representation of possible transitions between boxes. In this representation, boxes are graph vertices and non-forbidden transitions are graph edges. The reduction of the size of the state space is obtained by removing boxes containing transient dynamics only, which is equivalent to finding an enclosure of the invariant part of the map. In this procedure, all boxes which are not a beginning of any edge are removed. Similarly, we remove all boxes which are not an end of any edge. A detailed description of this procedure is given in [26]. Decreasing the box size δ increases the number of boxes in the covering of the invariant part but decreases its area and hence decreases the number of representable

points belonging to these boxes. The value of δ should be selected as small as possible under the condition that the covering can be stored in the computer memory. Using this procedure with boxes of size $\delta = 2^{-17}$ yields the covering $E = \{\mathbf{v}_k\}_{k=1}^N$ of the invariant part composed of $N = 25736445$ boxes (compare also [27]) with the area below 0.0014 which is significantly smaller than the area of the rectangle $[-1.5, 1.5] \times [-1.5, 1.5]$. Let us denote by Ω the set of representable points belonging to the boxes in the covering E . In the following, the set Ω is considered as the state space of finite-precision implementations of the Hénon map for all computational formulas (3). The size of Ω is $M_f \approx 2.6 \cdot 10^{13}$ and $M_d \approx 5.9 \cdot 10^{31}$ for single- and double-precision representations, respectively. For a single box the number of representable points belonging to this box is calculated as a product of representable numbers in the x and y coordinates. M_f and M_d are computed as the sum of representable points in each box over all boxes belonging to E . Note that the number of representable points in a given box is not constant despite the fact that the boxes are of the same size. For example, for the single-precision representation, the number of representable points belonging to a box in the covering E varies from 4225 to $1.2 \cdot 10^{11}$. This huge difference is a consequence of using subnormal numbers defined in the IEEE 754 standard. With subnormal numbers, the density of representable numbers in a neighborhood of zero is significantly increased.

B. Finding All Cycles

As mentioned before, in order to find all cycles, one should, in general, consider all points belonging to the state space and for each initial point find the cycle to which its trajectory converges. In the case of the Hénon map, the size of the state space is very large even for the single-precision floating-point format ($M_f \approx 2.6 \cdot 10^{13}$). The number of initial points to be considered can be reduced by skipping certain boxes. The idea is to introduce an ordering of boxes and to skip a box if all trajectories passing through this box has to pass through boxes larger according to the selected ordering. An ordering should permit removing boxes containing many representable points (i.e., those that lie close to the axes $x = 0$ and $y = 0$). The following ordering is used during the computations: $\mathbf{v}_k > \mathbf{v}_l$ if and only if $k_x > l_x$ or $k_x = l_x$ and $k_y > l_y$, where (k_x, k_y) and (l_x, l_y) are pairs of integers defining boxes \mathbf{v}_k and \mathbf{v}_l , respectively. The box \mathbf{v}_k is skipped if its image or preimage under the m th iteration of the map is enclosed in the set of boxes $E_{>k} = \{\mathbf{v} \in E: \mathbf{v} > \mathbf{v}_k\}$. This elimination procedure does not require recalculation of images or preimages of boxes. It can be carried out based on the information on possible transitions between boxes which is stored in the graph representation of the dynamics of the map. Applying this procedure to the set of boxes E with $m \in \{1, 2, \dots, 10\}$ leads to the set E' composed of $N = 1183485$ boxes. Using $m > 10$ increases the computation time without significant improvement of the results. The number of representable points in a single box belonging to E' varies from 4225 to 33153. The number of initial points that must be considered

Algorithm 1 Find All Cycles.

```

1: function REDUCECOVERING( $E, M$ )
2:    $E' \leftarrow E$ 
3:   for all  $\mathbf{v}_k \in E'$  do ▷ process each box
4:     for all  $m \in \{\pm 1, \pm 2, \dots, \pm M\}$  do
5:       if  $h^m(\mathbf{v}_k) \subset E_{>k}$  then
6:          $E' \leftarrow E' \setminus \{\mathbf{v}_k\}$  ▷ remove box
7:       end if
8:     end for
9:   end for
10:  return  $E'$ 
11: end function
12: function FINDALLCYCLES
13:   $E \leftarrow$  the set of boxes covering the attractor
14:   $E' \leftarrow$  REDUCECOVERING( $E, 10$ )
15:   $C \leftarrow \emptyset$  ▷ initialize the set of cycles
16:  for all  $\mathbf{v}_k \in E'$  do ▷ all boxes in  $E'$ 
17:    for all  $(x, y) \in \mathbf{v}_k$  do ▷ all initial points in  $\mathbf{v}_k$ 
18:       $c \leftarrow$  the cycle for the initial condition  $(x, y)$ 
19:       $C \leftarrow C \cup \{c\}$ 
20:    end for
21:  end for
22:  return  $C$ 
23: end function

```

to find all cycles is reduced from $M_f \approx 2.6 \cdot 10^{13}$ representable points belonging to boxes in the covering E to $M'_f \approx 6.5 \cdot 10^9$ representable points in E' . In the last step for each initial point in E' the corresponding steady state cycle is computed. The complete procedure to find all cycles is presented as the Algorithm 1.

For the double-precision and extended-precision computations the presented method to find all cycles does not work due to very large sizes of the state space. Cycles can be found by computing steady state trajectories starting from random initial conditions. This method does not guarantee finding all cycles and will be referred to as the statistical approach. Cycles with a large basin of attraction (which is equivalent to a high convergence probability), are more likely to be found. The probability that a cycle with the convergence probability c_{prob} is detected starting from N random initial conditions is $1 - (1 - c_{\text{prob}})^N$. To detect such a cycle with the probability larger than 0.95 one should use N such that $1 - (1 - c_{\text{prob}})^N > 0.95$. It follows that with N random initial conditions a cycle with the convergence probability $c_{\text{prob}} \geq c_N = 1 - \sqrt[N]{0.05}$ is detected with the probability larger than 0.95. For example, for $N = 100$ and $N = 10000$ we have $c_N \approx 0.02951$ and $c_N \approx 0.0002995$, respectively.

C. Computing Preimages in the Digital Domain

The (infinite precision) Hénon map (2) is invertible. The preimage of $z = (x, y)$ is a single point $z' = (x', y') = ((y - 1 - ax^2)/b, x)$. Finite-precision implementations are not invertible—the preimage of a representable point $z = (x, y)$ is a (possibly empty) set of representable points, whose images are equal to z . The procedure to find the preimage $h^{-1}(z)$

Algorithm 2 Find the Preimage Set S for the Point (x, y) .

```

1: function  $h(x, y)$  ▷ evaluate the map
2:   return  $(y, 1+b*x-y*a)$ 
3: end function
4: function  $h^{-1}(x, y)$  ▷ evaluate the inverse
5:   return  $((y-1-a*x*x)/b, x)$ 
6: end function
7: function FINDPREIMAGE( $x, y$ )
8:    $S \leftarrow \emptyset$  ▷ initialize the preimage
9:    $(x', y') \leftarrow h^{-1}(x, y)$ 
10:  repeat
11:     $(x'', y'') \leftarrow h(x', y')$ 
12:    if  $(x'', y'') = (x, y)$  then
13:       $S \leftarrow S \cup \{(x', y')\}$  ▷ update the preimage
14:    end if
15:     $x' \leftarrow$  the smallest representable number above  $x'$ 
16:  until  $y'' \leq y$ 
17:   $(x', y') \leftarrow h^{-1}(x, y)$ 
18:  repeat
19:     $(x'', y'') \leftarrow h(x', y')$ 
20:    if  $(x'', y'') = (x, y)$  then
21:       $S \leftarrow S \cup \{(x', y')\}$  ▷ update the preimage
22:    end if
23:     $x' \leftarrow$  the largest representable number below  $x'$ 
24:  until  $y'' \geq y$ 
25:  return  $S$ 
26: end function

```

of $z = (x, y)$ when the map h is implemented in the digital domain using one of the formulas (3) is based on the weak monotonicity of the expression $1+bx-ay^2$ with respect to the variable x when calculations are carried out in finite precision. Initially, the result set $h^{-1}(z)$ is empty. After computing $z' = (x', y') = ((y-1-ax^2)/b, x)$, the image $z'' = h(z')$ is found using the selected computational formula. If $z'' = z$, then z' is added to the preimage $h^{-1}(z)$. Next, x' is increased to be the next representable number, and the point $z' = (x', y')$ is added to $h^{-1}(z)$ if $z'' = (x'', y'') = h(z') = z$. Increasing x' and adding new points to $h^{-1}(z)$ is continued as long as $y'' \leq y$ (the stopping condition is $y'' > y$). A similar procedure is carried out with decreasing x' starting from $x' = (y-1-ax^2)/b$ and using the stopping condition $y'' < y$. A pseudo code for the algorithm to compute the preimage of the version (3a) is presented as the Algorithm 2. Other versions are obtained by changing the line 2 in the algorithm.

The time needed to find the preimage using the Algorithm 2 depends linearly on the preimage size. For the Hénon map, the size of the preimage in case of single-precision computations varies from zero (empty set) to approximately $1.77 \cdot 10^9$. For example, the preimage of $z = (0, 1.25)$ contains 7 points and the preimage of $(1.25, -1.1875) = h(z) = h(0, 1.25)$ contains 1760908629 points. Extremely large preimage sets exist when the preimage of z contains a point (x', y') with the first coordinate being zero or close to zero. In this case, many points of the form (x', y') with x' being a subnormal number produce the image z .

D. Constructing Basins of Attraction

The basin of attraction of a cycle may be constructed in a recursive manner by finding preimages of points belonging to the cycle, their preimages, and so on until there are no preimages in the state space. The procedure to compute the basin of attraction is as follows. First, we compute the preimage P of the cycle, that is, the set of points outside of the cycle whose images belong to the cycle. For each point $z \in P$ its basin of attraction (the set of points whose trajectories reach z) has a directed rooted tree structure with z being the root. The basin of attraction of $z \in P$ can be found using the Depth First Search (DFS) algorithm [28]. The DFS algorithm is continued until a leaf (a point without preimages) is reached or a preimage is outside the state space Ω . The advantage of this procedure is that we do not need to store the basin of attraction in a computer memory. As a consequence, the procedure can be applied to relatively large basins. In Section IV, it is shown that the proposed approach can successfully handle basins of size up to $2 \cdot 10^{11}$.

The sizes of larger basins can be estimated using a statistical method. In this approach, a large number of initial points in the state space Ω is selected and for each initial point the steady state solution is identified. The probability c_{prob} of convergence to a given cycle is calculated as the ratio of the number of initial points whose trajectories converge to this cycle and the total number of initial points. The size of the basin can be estimated as $s = c_{\text{prob}} \cdot M$ where M is the number of representable points in Ω .

In infinite precision, to distinguish whether a cycle represents a stable orbit, one may compute the Lyapunov spectrum of a trajectory along this cycle. For a two-dimensional map, the Lyapunov exponents λ_1, λ_2 computed along a cycle characterize the rate of attracting or repelling nearby trajectories in different directions of the state space. We assume that the Lyapunov exponents are sorted in descending order, i.e., $\lambda_1 \geq \lambda_2$. Note that the Lyapunov spectrum computed for different cycles is usually different. If all Lyapunov exponents are negative, then the cycle is stable. This indicates that the cycle corresponds to a periodic attractor of the system. If the largest Lyapunov exponent is positive, then the orbit is unstable. In this case, we may expect that the cycle corresponds to a chaotic trajectory and the observed periodic trajectory is an artifact caused by rounding errors. For the calculation of Lyapunov exponents, the QR decomposition-based method is used. During the computations a trajectory (z_1, z_2, \dots, z_n) of the length n is considered. The matrix Q_0 is initialized to be the identity matrix. At the k th step the Jacobian matrix $J_k = h'(z_k)$ is computed and the matrix $J_k Q_{k-1}$ is decomposed as a product $Q_k R_k$ of an orthogonal matrix Q_k and an upper triangular matrix R_k . The i th Lyapunov exponent is computed as the sum of logarithms of the i th diagonal entries of matrices R_k divided by n (for details see [29]). When Lyapunov exponents are calculated along a cycle, it is sufficient to consider a single pass of a trajectory along this cycle. To obtain accurate results, the QR decomposition algorithm should be applied for several iterations before starting calculations of the Lyapunov exponents. This ensures that the matrices Q_k are

TABLE II

CHAOTIC CASE. THE float DATA TYPE. CYCLES EXISTING FOR $a = 1.399999976$, $b = 0.300000012$. v IS THE VERSION OF THE COMPUTATIONAL FORMULA, p IS THE PERIOD OF THE ORBIT, (x, y) IS THE POSITION OF THE PERIOD- p POINT, $\lambda_{1,2}$ ARE LYAPUNOV EXPONENTS, AND d IS THE LYAPUNOV DIMENSION COMPUTED ALONG THE ORBIT. t_{aver} AND t_{max} ARE THE AVERAGE AND MAXIMUM CONVERGENCE TIMES

v	i	p	x	y	λ_1	λ_2	d	c_{prob}	s	t_{aver}	t_{max}
(3a)	1	49157	-1.28466189	-0.928997874	0.4188	-1.623	1.2581	0.645		31049	105988
	2	24310	-1.28458834	-0.928808093	0.4195	-1.623	1.2584	0.093		5659	18970
	3	8837	-1.28465486	-0.92898953	0.4213	-1.625	1.2592	0.0843		16639	38477
	4	5148	-1.2842896	-0.927835107	0.4096	-1.614	1.2538	0.147		31458	58549
	5	4116	-1.2845577	-0.928635001	0.4181	-1.622	1.2578	0.00358	$6.54 \cdot 10^{10}$	1731	4427
	6	3082	-1.28349948	-0.925096393	0.4256	-1.63	1.2612	0.000532	$2.22 \cdot 10^{10}$	236	1466
	7	2702	-1.28465915	-0.928972602	0.423	-1.627	1.26	0.00697	$2.08 \cdot 10^{11}$	1983	6018
	8	2245	-1.28441238	-0.927927613	0.4098	-1.614	1.2539	0.00207	$4.45 \cdot 10^{10}$	1718	3614
	9	1479	-1.28465891	-0.928981423	0.4321	-1.636	1.2641	0.0163		5940	12046
	10	256	-1.28447223	-0.928208947	0.4042	-1.608	1.2513	0.000722	$8.36 \cdot 10^9$	1962	3737
	11	206	-1.27932668	-0.911034703	0.4196	-1.624	1.2585	0.000078	$1.87 \cdot 10^8$	742	1303
	12	7	-1.04667759	-0.178779602	0.41	-1.614	1.254		2697	3.4	6
	13	4	-0.70676684	0.638193965	0.5391	-1.743	1.3093		1241	3.2	5
	14	1	0.631354511	0.631354511	0.6543	-1.858	1.3521		84	2.5	3
	15	1	-1.13135445	-1.13135445	1.182	-2.386	1.4953		12	2.3	3
(3b)	1	55754	-1.28464162	-0.92885685	0.4188	-1.623	1.2581	0.96		26784	99990
	2	12209	-1.28462386	-0.928899765	0.4186	-1.623	1.258	0.0373		7311	20085
	3	1748	-1.28459668	-0.9288311	0.4176	-1.622	1.2575	0.00159	$1.54 \cdot 10^{10}$	1386	4412
	4	814	-1.28283966	-0.922817707	0.4129	-1.617	1.2554	0.00114	$9.69 \cdot 10^9$	1051	4887
	5	219	-1.28431988	-0.927983046	0.3825	-1.586	1.2411	0.000004	$2.34 \cdot 10^7$	110	328
	6	63	-1.28372622	-0.925351858	0.3871	-1.591	1.2433	0.000002	365685	13	47
	7	21	-1.00495303	-0.0504530668	0.4413	-1.645	1.2682		606703	33	60
	8	1	0.631354511	0.631354511	0.6543	-1.858	1.3521		43	2.2	3
(3c)	1	16538	-1.2846632	-0.928994417	0.4209	-1.625	1.259	0.0945		7060	17465
	2	3161	-1.28464437	-0.928908467	0.4104	-1.614	1.2542	0.00782	$1.89 \cdot 10^{11}$	4216	9632
	3	1732	-1.28463411	-0.928890705	0.4207	-1.625	1.2589	0.0755		11936	26898
	4	619	-1.28360343	-0.925424099	0.4107	-1.615	1.2544	0.822		54139	95362
	5	605	-1.28010464	-0.913319945	0.4418	-1.646	1.2684	0.00007	$1.54 \cdot 10^8$	464	1071
	6	32	-1.25456524	-0.822577953	0.3606	-1.565	1.2305	0.000082	$1.16 \cdot 10^6$	36	96
(3d)	1	55500	-1.28465104	-0.928904057	0.418	-1.622	1.2577	0.756		33647	99284
	2	30557	-1.28461313	-0.928730011	0.4192	-1.623	1.2583	0.19		23071	64185
	3	27681	-1.2846632	-0.928985	0.4216	-1.626	1.2594	0.0529		3623	12669
	4	287	-1.27696562	-0.901554942	0.4401	-1.644	1.2677	0.000588	$6.37 \cdot 10^9$	761	3848
	5	28	-1.24772	-0.798988581	0.4346	-1.639	1.2652		533338	23	52
	6	7	-1.04667759	-0.178779602	0.41	-1.614	1.254		64965	20	35
(3e)	1	14119	-1.28465605	-0.928986311	0.416	-1.62	1.2568	0.0607		8574	19736
	2	12268	-1.28375006	-0.925694764	0.4175	-1.621	1.2575	0.34		49903	105242
	3	2063	-1.28272998	-0.922461152	0.424	-1.628	1.2605	0.599		49600	128191
	4	705	-1.2840482	-0.927098632	0.4264	-1.63	1.2616	0.000148	$2.36 \cdot 10^9$	655	1745
	5	1	-1.13135445	-1.13135445	1.182	-2.386	1.4953		12	2.3	3
(3f)	1	27059	-1.28466356	-0.928991437	0.4156	-1.62	1.2566	0.723		32852	79417
	2	2184	-1.2836802	-0.92540741	0.4238	-1.628	1.2604	0.0823		14124	33374
	3	1112	-1.28084075	-0.915024519	0.4051	-1.609	1.2518	0.00308	$5.05 \cdot 10^{10}$	4847	8622
	4	292	-1.27995944	-0.913259268	0.4042	-1.608	1.2513	0.00001	$9.01 \cdot 10^7$	107	447
	5	178	-1.28062797	-0.91453135	0.4291	-1.633	1.2628	0.192		29459	53642
	6	33	-1.09031641	-0.294407964	0.4187	-1.623	1.258		443071	21	59
	7	18	-0.996454597	-0.0403779745	0.4013	-1.605	1.25		465305	22	87
	8	15	-1.26138532	-0.84638989	0.4461	-1.65	1.2703		711692	10	30
	9	13	-1.20857644	-0.672999203	0.1988	-1.403	1.1417	0.000046	$2.07 \cdot 10^8$	660	1321
	10	4	-0.706766784	0.638194025	0.5391	-1.743	1.3093		50310	22	34

smaller than the estimated state space size $M_f \approx 2.6 \cdot 10^{13}$. The discrepancy is likely to be caused by the fact that a large number of points in the state space (for example, points with a coordinate being a subnormal number) are not reachable (i.e., have an empty preimage). In this context, a more precise estimate of the size of the state space is based on the set E' which contains approximately $6.5 \cdot 10^9$ representable points.

For all cycles, the largest Lyapunov exponent and the Lyapunov dimension are positive. In most cases, the largest Lyapunov exponent is close to the value of the largest Lyapunov exponent $\lambda_1 \approx 0.419$ observed for higher precision computations. The largest deviations are seen for short

periodic orbits: $\lambda_1 \approx 1.182$ for the period-1 cycles existing for versions (3a) and (3e), $\lambda_1 \approx 0.654$ for the period-1 cycles existing for versions (3a) and (3b), $\lambda_1 \approx 0.5391$ for the period-4 cycle (version (3f)), and $\lambda_1 \approx 0.1988$ for the period-13 cycle (version (3f)). These cycles correspond to short unstable periodic orbits of the infinite-precision Hénon map. Their presence in finite-precision implementations is a result of rounding errors. The small maximum Lyapunov exponent $\lambda_1 \approx 0.1988$ observed for the period-13 cycle (version (3f)) means that this cycle is weakly repelling. This explains the large basin of attraction of this cycle when compared to other low period cycles.

TABLE III
CHAOTIC CASE. THE `double` DATA TYPE. CYCLES EXISTING FOR $a = 1.39999999999999911$, $b = 0.29999999999999989$

v	i	p	x	y	λ_1	λ_2	d	c_{prob}	t_{aver}	t_{max}
(3a)	1	2875179482	-1.2846637923573334	-0.92898795752924412	0.41922	-1.62319	1.25827	0.864	$1.2 \cdot 10^{10}$	$2.2 \cdot 10^{10}$
	2	1210581335	-1.2846637922954911	-0.92898790673907872	0.41921	-1.62318	1.25826	0.0909	$2.8 \cdot 10^9$	$5.0 \cdot 10^{10}$
	3	642615468	-1.2846637922348774	-0.92898809181774666	0.41922	-1.62319	1.25827	0.00162	$1.0 \cdot 10^9$	$1.0 \cdot 10^{10}$
	4	353107615	-1.2846637914947496	-0.92898824053195694	0.41921	-1.62318	1.25826	0.0409	$2.5 \cdot 10^9$	$5.0 \cdot 10^{10}$
	5	323659552	-1.2846637919996193	-0.92898785995540933	0.41923	-1.62321	1.25827	0.00162	$1.0 \cdot 10^9$	$1.0 \cdot 10^{10}$
	6	140522953	-1.2846637917838009	-0.92898776964827112	0.41923	-1.62320	1.25827	0.000974	$1.0 \cdot 10^9$	$1.0 \cdot 10^{10}$
(3f)	1	7516037528	-1.2846637922906359	-0.92898805896544401	0.41922	-1.62319	1.25827	0.988	$1.1 \cdot 10^{10}$	$2.6 \cdot 10^{10}$
	2	351284733	-1.2846637918188992	-0.92898771375469336	0.41924	-1.62321	1.25828	0.000334	$1.0 \cdot 10^9$	$1.0 \cdot 10^9$
	3	333039039	-1.2846637922066118	-0.92898782955173664	0.41922	-1.62319	1.25827	0.00936	$1.3 \cdot 10^9$	$2.0 \cdot 10^9$
	4	121055524	-1.284663791167618	-0.92898758584169372	0.41927	-1.62324	1.25829	0.00134	$1.0 \cdot 10^9$	$1.0 \cdot 10^9$
	5	96097140	-1.2846637903101052	-0.9289881993246264	0.41916	-1.62313	1.25824	0.000669	$1.0 \cdot 10^9$	$1.0 \cdot 10^9$

In all cases, the average length of a trajectory before it repeats itself belongs to the interval of $[5 \cdot 10^4, 8 \cdot 10^4]$. It follows that for the `float` type it does not make sense to consider trajectories with more than 10^5 iterations.

Let us now consider the double-precision floating-point format. In this case, the graph-based approach does not work due to extremely long computation time needed to consider all initial conditions. Therefore, we are limited to the statistical approach. For the sake of brevity, only the results for versions (3a) and (3f) are presented. For each version $N = 3000$ initial conditions are considered and steady-state behavior is found. The average computation time is 4 hours per initial point. Computation times are much longer than for the single-precision case due to much longer transients. The results are presented in Table III.

The number of cycles found is 6 and 5 for versions (3a) and (3f), respectively. The periods of the longest cycles are $p = 2875179482$ and $p = 7516037528$, which corresponds to estimates of the size of the state space $p^2 \approx 8.3 \cdot 10^{18}$ and $p^2 \approx 5.6 \cdot 10^{19}$. These estimates are significantly smaller than the number $M_d \approx 5.9 \cdot 10^{31}$ of representable points in the covering E . It is expected that the difference is related to the usage of subnormal numbers, similarly as for the single-precision floating-point format. For both versions, the longest cycle has the largest basin of attraction. The probability of convergence to the longest cycle is $c_{\text{prob}} = 0.864$ and $c_{\text{prob}} = 0.988$ for versions (3a) and (3f), respectively. The probability of convergence to the second longest cycle for the version (3a) is $c_{\text{prob}} \approx 9\%$. For the version (3f) the convergence probability for each cycle apart from the longest one is below 1%. In general, the probability of convergence is lower for shorter orbits. There are, however, some exceptions. For example, for the version (3a) the probability of convergence to the orbit #4 is more than twice that of orbit #3, despite the fact that the period is almost two times shorter.

Values of Lyapunov exponents are very close to each other for all cycles. This means that each cycle represents a chaotic trajectory and can be used to characterize behavior of the infinite-precision system in terms of Lyapunov exponents and the Lyapunov dimension. The average length of a trajectory before repeating itself is above 10^{10} .

One should keep in mind that although short cycles have not been found, it is very likely that there exist short cycles with periods below 100 as in the single-precision case. Such

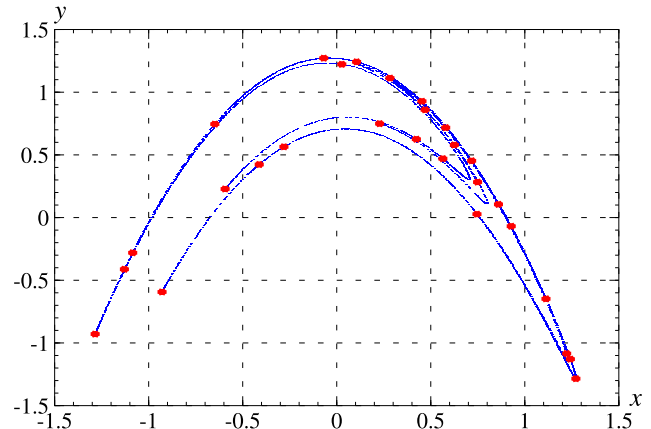


Fig. 2. Transient trajectory (blue dots) and stable period-28 orbit (red circles) for $a = 1.4$, $b = 0.2999999774905$ (periodic case). Computations are carried out in double precision.

orbits are difficult to find using the statistical approach due to small basins of attraction.

The results obtained for the extended-precision (`long double`) format are shown in Table IV. The number of initial conditions considered for each case is $N = 250$. The average computation time is 4 and 2 days per initial point for versions (3a) and (3f), respectively. The longer average computation time for version (3a) is the effect of longer convergence times. The lower number of cycles found (2 and 3 for versions (3a) and (3f), respectively) compared to the double-precision case is related to the lower number of initial conditions considered. The results regarding Lyapunov exponents and the Lyapunov dimension are practically the same for all orbits. The average convergence time is above 10^{12} and is 100 times longer than for the double-precision case. It follows that the extended-precision format is sufficient for most applications since longer trajectories are usually not needed.

B. Periodic Case

Let us now consider the parameter values $a = 1.3999999999999991118$, $b = 0.29999997749050000273$ for which a stable period-28 orbit exists (compare [32]). Fig. 2 shows a short part of a chaotic transient trajectory (blue dots) and the periodic attractor (red circles). The coordinates of the point

TABLE VI
PERIODIC CASE. THE `double` DATA TYPE. CYCLES EXISTING FOR $a = 1.4$, $b = 0.2999999774905$

v	i	p	x	y	λ_1	λ_2	d	c_{prob}	t_{aver}	t_{max}
(3a)	1	2932686468	-1.2846637110950554	-0.92898766231225172	0.41920	-1.62317	1.25826	0.0945	$2.8 \cdot 10^9$	$5.0 \cdot 10^9$
	2	2739066510	-1.2846637110576862	-0.92898770743596515	0.41922	-1.62319	1.25827	0.0774	$2.1 \cdot 10^9$	$4.0 \cdot 10^9$
	3	2639598963	-1.2846637110137289	-0.92898780198678121	0.41921	-1.62318	1.25827	0.0171	$1.1 \cdot 10^9$	$2.0 \cdot 10^9$
	4	507236098	-1.284663710857775	-0.92898789001902848	0.41921	-1.62319	1.25827	0.00302	$1.0 \cdot 10^9$	$1.0 \cdot 10^9$
	5	81490751	-1.2846637062458517	-0.92898823384952789	0.41926	-1.62323	1.25829	0.00101	$1.0 \cdot 10^9$	$1.0 \cdot 10^9$
	6	69959678	-1.2846637088812898	-0.9289871793035116	0.41926	-1.62323	1.25829	0.00101	$1.9 \cdot 10^9$	$1.0 \cdot 10^9$
	7	532	-1.2846430067651009	-0.92893246181716327	-0.07034	-1.13362	0	0.806	$7.6 \cdot 10^9$	$1.5 \cdot 10^{10}$
(3f)	1	3440173509	-1.2846637110921753	-0.92898766263602706	0.41921	-1.62319	1.25827	0.439	$4.9 \cdot 10^9$	$1.2 \cdot 10^{10}$
	2	2979320146	-1.284663711030581	-0.92898760859483165	0.41922	-1.62319	1.25827	0.128	$2.8 \cdot 10^9$	$6.0 \cdot 10^9$
	3	224	-1.284643006765241	-0.92893246181714739	-0.07046	-1.13351	0	0.0734	$3.0 \cdot 10^9$	$6.0 \cdot 10^9$
	4	196	-1.2846430067644505	-0.92893246181722922	-0.07020	-1.13377	0	0.36	$5.1 \cdot 10^9$	$1.2 \cdot 10^{10}$

TABLE VII
PERIODIC CASE. THE `Long double` DATA TYPE. CYCLES EXISTING FOR $a = 1.3999999999999991118$, $b = 0.29999997749050000273$

v	i	p	x	y	λ_1	λ_2	d	c_{prob}	t_{aver}	t_{max}
(3a)	1	672	-1.2846430067625941951	-0.92893246181742686828	-0.07042	-1.134	0	0.82	$2.1 \cdot 10^{10}$	$2.0 \cdot 10^{11}$
	2	252	-1.2846430067625941006	-0.92893246181742687869	-0.07042	-1.134	0	0.105	$1.9 \cdot 10^{10}$	$8.0 \cdot 10^{10}$
	3	196	-1.2846430067625932401	-0.92893246181742696933	-0.07042	-1.134	0	0.0614	$2.0 \cdot 10^{10}$	$1.0 \cdot 10^{11}$
	4	28	-1.2846430067625934601	-0.92893246181742694472	-0.07042	-1.134	0	0.0128	$3.4 \cdot 10^{10}$	$1.0 \cdot 10^{11}$
	5	28	-1.2846430067625926228	-0.92893246181742703395	-0.07042	-1.134	0	0.000916	$5.0 \cdot 10^9$	$5.0 \cdot 10^9$
(3f)	1	616	-1.284643006762594939	-0.92893246181742683624	-0.07042	-1.134	0	0.572	$2.1 \cdot 10^{10}$	$1.6 \cdot 10^{11}$
	2	252	-1.2846430067625937126	-0.92893246181742691848	-0.07042	-1.134	0	0.298	$2.1 \cdot 10^{10}$	$1.3 \cdot 10^{11}$
	3	56	-1.284643006762593032	-0.92893246181742699069	-0.07042	-1.134	0	0.00376	$1.5 \cdot 10^{10}$	$3.6 \cdot 10^{10}$
	4	28	-1.2846430067625933041	-0.92893246181742696136	-0.07042	-1.134	0	0.11	$2.5 \cdot 10^{10}$	$1.1 \cdot 10^{11}$
	5	28	-1.2846430067625919449	-0.92893246181742710615	-0.07042	-1.134	0	0.0132	$1.8 \cdot 10^{10}$	$5.5 \cdot 10^{10}$
	6	28	-1.2846430067625914648	-0.92893246181742715733	-0.07042	-1.134	0	0.00282	$4.2 \cdot 10^{10}$	$5.7 \cdot 10^{10}$

indeed chaotic. The reason is that the transient times needed to converge to a stable periodic orbit may be very long. For long transients there is a non-negligible probability that before convergence a transient trajectory hits a point in the state space which was visited before, thus forming a long periodic trajectory not related to the existing periodic attractor.

The results obtained for the extended-precision implementation based on the calculations of trajectories started in $N = 1000$ randomly selected initial points are reported in Table VII. The average computation time is 5 hours per initial point and is significantly smaller than for the chaotic case, which permits testing more initial points. All tested trajectories are attracted to a cycle located very close to the true period-28 orbit. All periods are equal to 28 or are low multiples of 28. For the versions (3a) and (3f) there are two and three distinct period-28 orbits, respectively. Lyapunov exponents computed along the cycles are all negative and are equal within the reported accuracy (four significant decimal digits). Average convergence times are below 10^{10} for all cycles, which is approximately 100 times shorter than for the chaotic case.

We may conclude that the extended-precision implementation is better suited to study dynamical behaviors of the Hénon map for parameter values $a = 1.4$, $b = 0.2999999774905$. Due to the larger size of the state space, the probability that a transient trajectory hits a cycle before converging to the stable periodic orbit is lower than for the double-precision case.

On the other hand, one may say that the use of a lower precision introduces more chaos in the system (rounding errors are larger). For the double-precision implementation some of the observed cycles resemble chaotic trajectories with positive Lyapunov exponents, while for the extended-precision

all steady-state trajectories are low-period cycles with negative Lyapunov exponents.

V. CONCLUSION

A set of tools was presented for the analysis of finite-precision implementations of nonlinear maps. A method to reduce the search space to find all cycles was proposed. An efficient algorithm to compute preimages of a given representable point was described. A DFS based algorithm was designed to find basins of attraction of cycles.

The proposed methods were used to analyze finite-precision implementations of the Hénon map for chaotic and periodic cases. It was shown that different computational formulas lead to different dynamical properties of finite-precision implementations in spite of the fact that these formulas are equivalent from the mathematical point of view. It follows that when results of computer simulations of nonlinear maps are reported, it is necessary to precisely describe the way the map is implemented to permit reproducibility of the results. Different computational formulas and precisions should be tested to make sure that the observed phenomena are not artifacts caused by rounding errors. For the single-precision floating-point implementation, all cycles were found. The structure of the state space was investigated. Basins of attraction of cycles were calculated using the graph-based approach for basins with sizes below $2 \cdot 10^{11}$. Sizes of larger basins were estimated using the statistical approach. For double-precision and extended-precision implementations, cycles and their basins of attraction were studied using the statistical approach. Relations between cycle lengths, precision used, and the size of the state space were studied. It was shown that finite-precision

implementations of nonlinear maps may support cycles with very low periods. This may degrade properties of non-linear maps in practical applications.

The results show that dynamical properties of finite-precision implementations, such as the number of cycles, their lengths, and the probability of convergence to a given cycle, depend on the order of elementary operations used to evaluate the map. It follows that the selection of the computational formula used in finite-precision computations is important both for applications of nonlinear maps and for the results of theoretical studies. In applications of nonlinear maps one should test all possible evaluation methods and select the one with best dynamical properties.

From the results obtained for the double-precision implementations it follows that rounding errors may strongly disturb the simulation results. Chaotic-like trajectories observed in simulations do not necessarily imply that the dynamics of the infinite-precision system is chaotic. The results presented should alert researchers studying nonlinear maps to dangers of making conclusions about dynamical properties based purely on computer simulations.

The methods presented and the results obtained for the Hénon map can be used by other researchers to help design simulation studies and applications of nonlinear maps. The methods after minor modifications may be used to study other nonlinear maps implemented in the digital domain.

REFERENCES

- [1] H. Hu, Y. Xu, and Z. Zhu, "A method of improving the properties of digital chaotic system," *Chaos, Solitons Fractals*, vol. 38, no. 2, pp. 439–446, 2008.
- [2] S.-L. Chen, T. Hwang, and W.-W. Lin, "Randomness enhancement using digitalized modified logistic map," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 12, pp. 996–1000, Dec. 2010.
- [3] Z. Hua and Y. Zhou, "Dynamic parameter-control chaotic system," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3330–3341, Dec. 2016.
- [4] C. E. C. Souza, D. P. B. Chaves, and C. Pimentel, "One-dimensional pseudo-chaotic sequences based on the discrete Arnold's cat map over Z_{3^m} ," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 1, pp. 491–495, Jan. 2021.
- [5] C. Li, K. Tan, B. Feng, and J. Lu, "The graph structure of the generalized discrete Arnold's cat map," *IEEE Trans. Comput.*, vol. 71, no. 2, pp. 364–377, Feb. 2022.
- [6] M. Garcia-Bosque, C. Sanchez-Azqueta, and S. Celma, "Secure communication system based on a logistic map and a linear feedback shift register," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 1170–1173.
- [7] L. Chen, C. Li, and C. Li, "Security measurement of a medical communication scheme based on chaos and DNA coding," *J. Vis. Commun. Image Represent.*, vol. 83, Feb. 2022, Art. no. 103424.
- [8] H. Li, L. Deng, and Z. Gu, "A robust image encryption algorithm based on a 32-bit chaotic system," *IEEE Access*, vol. 8, pp. 30127–30151, 2020.
- [9] I. Öztürk and R. Kiliç, "Cycle lengths and correlation properties of finite precision chaotic maps," *Int. J. Bifurcation Chaos*, vol. 24, no. 9, 2014, Art. no. 1450107.
- [10] M. Farajallah, S. E. Assad, and O. Deforges, "Fast and secure chaos-based cryptosystem for images," *Int. J. Bifurcation Chaos*, vol. 26, no. 2, 2016, Art. no. 1650021.
- [11] S. Zhu, C. Zhu, H. Cui, and W. Wang, "A class of quadratic polynomial chaotic maps and its application in cryptography," *IEEE Access*, vol. 7, pp. 34141–34152, 2019.
- [12] C.-Y. Li, Y.-H. Chen, T.-Y. Chang, L.-Y. Deng, and K. To, "Period extension and randomness enhancement using high-throughput reseeding-mixing PRNG," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 2, pp. 385–389, Feb. 2012.
- [13] Y. Zhou, Z. Hua, C.-M. Pun, and C. L. P. Chen, "Cascade chaotic system with applications," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 2001–2012, Sep. 2015.
- [14] C. Fan and Q. Ding, "Analysis and resistance of dynamic degradation of digital chaos via functional graphs," *Nonlinear Dyn.*, vol. 103, no. 1, pp. 1081–1097, Jan. 2021.
- [15] P. M. Binder and R. V. Jensen, "Simulating chaotic behavior with finite-state machines," *Phys. Rev. A, Gen. Phys.*, vol. 34, no. 5, pp. 4460–4463, 1986.
- [16] C. Grebogi, E. Ott, and J. A. Yorke, "Roundoff-induced periodicity and the correlation dimension of chaotic attractors," *Phys. Rev. A, Gen. Phys.*, vol. 38, no. 7, pp. 3688–3692, Oct. 1988.
- [17] S. Li, G. Chen, and X. Mou, "On the dynamical degradation of digital piecewise linear chaotic maps," *Int. J. Bifurcation Chaos*, vol. 15, no. 10, pp. 3119–3151, 2005.
- [18] J. A. Oteo and J. Ros, "Double precision errors in the logistic map: Statistical study and dynamical interpretation," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, 2007, Art. no. 036214.
- [19] T. Miyazaki, S. Araki, and S. Uehara, "A study on differences in properties of the logistic maps over integers affected by rounding," in *Proc. Int. Symp. Inf. Theory Appl.*, Oct. 2010, pp. 830–835.
- [20] C. Li, B. Feng, S. Li, J. Kurths, and G. Chen, "Dynamic analysis of digital chaotic maps via state-mapping networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 6, pp. 2322–2335, Jun. 2019.
- [21] Z. Galias, "Periodic orbits of the logistic map in single and double precision implementations," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 11, pp. 3471–3475, Nov. 2021.
- [22] R. De Leo and J. A. Yorke, "The graph of the logistic map is a tower," *Discrete Continuous Dyn. Syst.*, vol. 41, no. 11, p. 5243, 2021.
- [23] M. Hénon, "A two dimensional map with a strange attractor," *Commun. Math. Phys.*, vol. 50, no. 1, pp. 69–77, 1976.
- [24] T. Kapela, M. Mrozek, D. Wilczak, and P. Zgliczyński, "CAPD::DynSys: A flexible C++ toolbox for rigorous numerical analysis of dynamical systems," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 101, Oct. 2021, Art. no. 105578.
- [25] *IEEE Standard for Floating-Point Arithmetic*, IEEE Std 754-2019 (Revision of IEEE 754-2008), 2019, pp. 1–84.
- [26] Z. Galias, "The dangers of rounding errors for simulations and analysis of nonlinear circuits and systems—And how to avoid them," *IEEE Circuits Syst. Mag.*, vol. 13, no. 3, pp. 35–52, 3rd Quart., 2013.
- [27] Z. Galias, "On topological entropy of finite representations of the Hénon map," *Int. J. Bifurcation Chaos*, vol. 29, no. 13, 2019, Art. no. 1950175.
- [28] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2009.
- [29] A. Wolf, J. Swift, H. Swinney, and J. A. Vastano, "Determining Lyapunov exponents from a time series," *Phys. D, Nonlinear Phenomena*, vol. 16, no. 3, pp. 285–317, Oct. 1984.
- [30] J. L. Kaplan and J. A. Yorke, "Chaotic behavior of multidimensional difference equations," in *Functional Differential Equations and Approximation of Fixed Points*, H.-O. Peitgen and H.-O. Walthers, Eds. Berlin, Germany: Springer, 1979, pp. 204–227.
- [31] Z. Galias and W. Tucker, "Is the Hénon attractor chaotic?" *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 25, no. 3, 2015, Art. no. 033102.
- [32] Z. Galias and W. Tucker, "On the structure of existence regions for sinks of the Hénon map," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 24, no. 1, Mar. 2014, Art. no. 013120.



Zbigniew Galias (Senior Member, IEEE) received the M.Sc. degree in electronics from the AGH University of Science and Technology, Kraków, Poland, in 1990, the M.Sc. degree in mathematics from Jagiellonian University, Kraków, in 1992, and the Ph.D. and Senior Doctorate degrees in electrical engineering from the AGH University of Science and Technology in 1996 and 2004, respectively.

In 2015, he received the title of a Professor. He is currently a Full Professor at the AGH University of Science and Technology. His research interests

include complex and nonlinear circuits and systems, chaos, computer assisted proofs, power networks, and sliding mode control.

Prof. Galias served/is serving as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, *IEEE Circuits and Systems Magazine*, and the *International Journal of Bifurcation and Chaos*.