The dangers of rounding errors for simulations and analysis of nonlinear circuits and systems — and how to avoid them

Zbigniew Galias, *Member, IEEE,*

***Abstract*—Due to rounding errors results obtained by numerical simulations of nonlinear systems may be unreliable. It is however possible to carry out the computations in such a way that the results obtained are rigorous. In this paper several tools based on interval analysis are presented. Methods for computing trajectories, techniques for finding accurate enclosures of attractors, interval operators for proving the existence of fixed points and periodic orbits, and methods for proving the existence of nontrivial symbolic dynamics, are described. Computational techniques are illustrated by performing validated numerical analysis of the Hénon map and the Chua's circuit.**

***Index Terms*—nonlinear system, chaos, interval analysis, Chua's circuit, Hénon map.**

I. Introduction

Numerical simulations play a fundamental role in studying dynamics of nonlinear systems. Due to inherent properties of digital computers results found by numerical simulations are almost never exact. The limitation comes from finite representations of real numbers. Even if the input data are machine representable numbers, the result of a simple mathematical operation may be not representable. Instead of a true result its approximation is returned by a computer. Small rounding errors accumulate and are propagated in further computations. For more complex problems this may result in totally wrong answers. The problem of error propagation is especially important for chaotic systems since errors introduced in each computation step grow exponentially (on average) due to sensitivity to initial conditions (positive Lyapunov exponent). Apart from rounding, there are other sources of computation errors when analysing nonlinear systems. For example, errors are introduced during integration of continuous time systems by using numerical methods, which are constructed by skipping higher-order terms in the Taylor expansion of the solution. This error is sometimes called the truncation error. Nevertheless, computer generated solutions are often accepted as true solutions.

To see how rounding errors may influence the results, let us compute a trajectory of the Hénon map $h(x,y) = (1+y-ax^2, bx)$, with $a = 1.4$, $b = 0.3$. We want to compute $(x_k, y_k) = h(x_{k-1}, y_{k-1})$ for $k = 1, 2, \ldots, 200$ for the initial point $(x_0, y_0) = (0, 0)$. During computations, the IEEE 754 double precision format to represent real numbers is used. The results obtained are compared with the accurate solution found using multiple precision interval computations (this technique will be described in detail in Section IV-A). The distance e_k between the double precision solution and the true solution versus the iteration number k is shown in Fig. 1. One can see that the error e_k, which is initially at the level of machine error grows exponentially fast and after approximately 80 iterations gets close to 1. From then on, the error does not increase any more because both trajectories remain in the same region of the state space (the Hénon attractor is bounded).

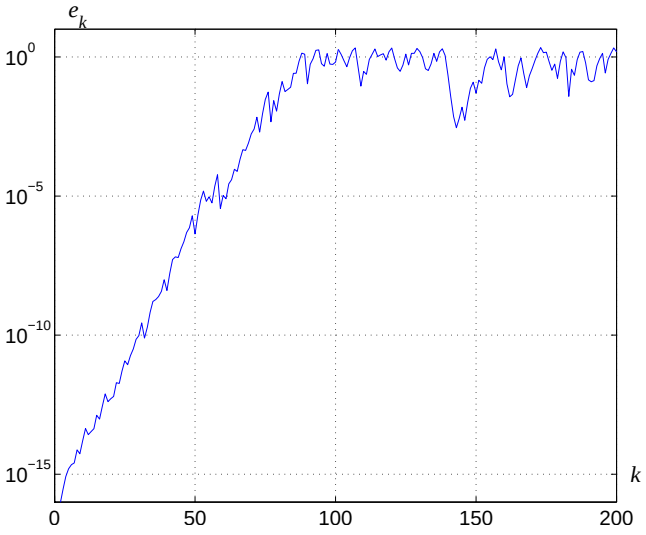

Fig. 1. Propagation of rounding errors during computations of the trajectory of the initial point $(0,0)$ for the Hénon map; the Euclidean distance e_k between the double precision solution and the true solution

This example shows that indeed, even for very simple systems propagation of rounding errors may lead to completely wrong results. The trajectory found using double precision computations after a certain number of iterations ($k = 80$ in this particular example) becomes uncorrelated with the true trajectory.

In this context, it is worth to mention the shadowing lemma which says that under certain assumptions a pseudo-trajectory (for example a trajectory with rounding errors at every iteration) is shadowed by a true one, i.e.

This work was supported in part by the AGH University of Science and Technology, grant no. 11.11.120.611.

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

stays close to it. The statement is valid also for trajectories of infinite length. For a precise formulation see [1]. It follows that for certain systems computer generated trajectories may serve as good approximations of true trajectories. However, the statement is valid only for trajectories near to an invariant hyperbolic set, the existence of which might be difficult or impossible to prove, as is the case for the Hénon map. Also, this approach is useless when the goal is to compute an enclosure of the trajectory based at a given initial point.

Sometimes, from observations of long trajectories one draws conclusions about the steady state behaviour of the system. Here, we show an example that such claims might be wrong. Fig. 2 shows two parts of the trajectory of the Hénon map with $a = 1.399999486944$, $b = 0.3$ based at the initial condition $(x_0, y_0) = (0.1, 0.1)$. The first part of the plot is obtained by skipping $5 \cdot 10^9$ iterations and plotting the next 10000 iterations. The plot looks like the classical Hénon attractor observed for $a = 1.4$, $b = 0.3$. The second part, also composed of 10000 points is obtained by skipping $6 \cdot 10^9$ iterations. One can see that the second part of the plot is composed of 33 dots.

Just by looking at the plot, we cannot say what is the true steady state solution (the observed period–33 orbit might be an artefact of the numerical procedure). We can however answer this question using techniques presented in this work. The existence of a period–33 stable periodic orbit for $a = 1.399999486944$, $b = 0.3$ is proved in Section IV-B (see also [2]).

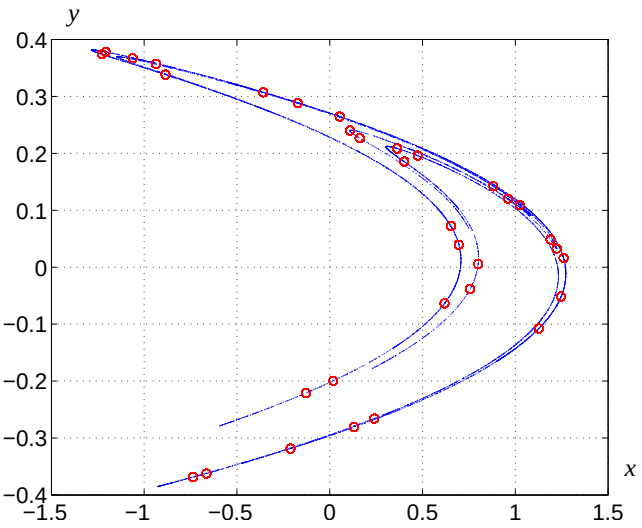

Fig. 2. Trajectory of the Hénon map with $a = 1.399999486944$, $b = 0.3$ for the initial condition $(x_0, y_0) = (0.1, 0.1)$, 10000 points after skipping $5 \cdot 10^9$ iterations are plotted using blue dots, 10000 points after skipping $6 \cdot 10^9$ iterations are plotted using red circles, chaotic transient is observed for more than $5 \cdot 10^9$ iterations; eventually, the trajectory converges to the period–33 stable periodic orbit

This example shows that it might be necessary to wait very long until the steady state is observed. Even if a trajectory looks chaotic it might be a transient to a simpler steady state behavior, like a periodic orbit. Sometimes, one computes certain characteristics of a trajectory like the Lyapunov exponents or the power spectrum to support statements about the existence of chaos. If one of the Lyapunov exponents evaluated over a long trajectory is positive and the power spectrum is continuous, it is often claimed that the system under study is chaotic. However, note that Lyapunov exponents and the power spectrum evaluated over the chaotic transient would not help in predicting what is the steady state behavior. Lyapunov exponents become negative and the power spectrum becomes discrete only after the trajectory converges to the periodic steady state.

Examples presented above show that standard computer simulations may lead to false conclusions. Fortunately, there are computational methods, which can be used to obtain results rigorous in the mathematical sense. Such methods have to cope with the problem of rounding errors. Interval analysis [3] implemented on a computer provides a solution. In interval analysis, instead of performing mathematical operations on uncertain real quantities, one works with intervals that contain the quantities of interest. All calculations are performed in such a way that the true result is always enclosed within the interval produced by the computations.

In this paper, several computational methods for the rigorous study of nonlinear systems are presented. In Section II, a brief introduction to interval analysis is presented. We describe the wrapping effect and show methods how to reduce it. Automatic differentiation is also discussed. In Section III, several interval analysis based methods for rigorous studies of nonlinear systems are presented. This includes methods for computing trajectories and representing the dynamics of a nonlinear system in the form of a directed graph. It is explained how this representation can be used to obtain accurate enclosures of the invariant part of a set. Interval operators for proving the existence of zeros of nonlinear maps are also described. It is shown how to apply these operators to find all short cycles. In Sections IV and V two nonlinear systems are analysed numerically: the Hénon map — a two-dimensional discrete system, and the Chua's circuit — a three-dimensional continuous system. We show how to compute enclosures of trajectories, find a trapping region, find enclosures of the invariant part of a given set, prove the existence of periodic orbits and prove the existence of chaos in the topological sense.

II. Interval analysis

In interval analysis [3], [4] intervals are used instead of real numbers. The closed interval with endpoints $\underline{a} \leq \overline{a}$ is denoted by $\mathbf{a} = [\underline{a}, \overline{a}] = \{a \colon \underline{a} \leq a \leq \overline{a}\}$. An interval vector is a Cartesian product of m intervals $\mathbf{v} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m) = \{(a_1, a_2, \ldots, a_m) \colon a_i \in \mathbf{a}_i \text{ for } i = 1, \ldots, m\}$. It corresponds to a box in $\mathbb{R}^m$.

We will use bold letters to denote intervals, interval vectors and matrices, and usual math italic to denote real quantities. For a given interval $\mathbf{a} = [\underline{a}, \overline{a}]$ its diameter and center are defined as $\mathrm{diam}(\mathbf{a}) = \overline{a} - \underline{a}$ and $\mathrm{mid}(\mathbf{a}) = 0.5(\overline{a} + \underline{a})$, respectively.

On the set of intervals basic arithmetic operators ($+$, $-$, $\times$, $\div$) are defined in such a way that the result of an operation on intervals is the smallest interval containing results of the corresponding real operation for all combinations of values from these intervals with the exception that $\mathbf{a} \div \mathbf{b}$ is undefined if $0 \in \mathbf{b}$. For example the sum of intervals $\mathbf{a}$ and $\mathbf{b}$ is defined as $\mathbf{a} + \mathbf{b} = \{a + b : a \in \mathbf{a}, b \in \mathbf{b}\}$. The result of arithmetic operations on intervals can be found by performing "real" arithmetic operations on their endpoints. For example formulas for addition and multiplication of intervals are: $[\underline{a}, \overline{a}] + [\underline{b}, \overline{b}] = [\underline{a} + \underline{b}, \overline{a} + \underline{b}]$, $[\underline{a}, \overline{a}] \times [\underline{b}, \overline{b}] = [\min\{\underline{a}\underline{b}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{a}\overline{b}\}, \max\{\underline{a}\underline{b}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{a}\overline{b}\}]$.

In practice, it is impossible to perform arithmetic operations (real of interval) with infinite precision. We are limited by representations of finite precision. When interval arithmetic is implemented on a computer one has to make sure that the result of an operation on intervals contains the results of the corresponding real operation for all combinations of values from these intervals. In the best implementation of interval arithmetic the left (right) endpoint found by a computer is the largest (smallest) representable real number not larger (not smaller) than the correct value. This is obtained by using directed rounding when performing "real" computations on interval endpoints.

Interval extensions of elementary functions are implemented using Taylor series of finite length and controlling the error caused by skipping higher order terms. For monotonic functions like $\exp$ and $\log$ it is sufficient to evaluate the functions at interval endpoints controlling the rounding modes. For non-monotonic functions like $\sin$ and $\cos$ one additionally has to verify conditions for existence of local extrema within the considered interval.

There are a number of interval algorithms, which can be used for solving various computational problems [3], [4]. Some of them are simple extensions of algorithms for real arithmetic, for example computation of the enclosure of a trajectory of a continuous system using Taylor expansion of the solution. There are also algorithms specific for interval analysis. They are based on the fact that an interval is a set of real numbers. Hence, one can compute the intersection of two intervals, or verify whether one interval is enclosed in another. Existence theorems for zeros of nonlinear maps based on evaluation of an interval operator belong to this class.

A. The wrapping effect and variable dependency

When computations are carried out in interval arithmetic, one often observes overestimation of the result. This effect is clearly visible when computations are based on recursive formulas, like finding trajectories of discrete or continuous dynamical systems [3], [5]. This overestimation can be related to the wrapping effect caused by enclosing intermediate results by intervals or to the variable dependency problem. The overestimation propagates, when intermediate results are used in further computations. In case of computing trajectories the overestimation eventually prevents long-term integration. First, let us show a simple example of the wrapping effect.

Let us consider a linear discrete dynamical system $z_{k+1} = f(z_k)$, where f defines a rotation by the angle φ, i.e. $f(x, y) = (x\cos(\varphi) - y\sin(\varphi), x\sin(\varphi) + y\cos(\varphi))$. Let us compute the image of a square $P = [0.9, 1.1] \times [-0.1, 0.1]$ after $n = 8$ iterations for $\varphi = \pi/8$. Since f describes the rotation, it is clear that the image of P is a square of the same size as P. For $\varphi = \pi/8$ after $n = 8$ iterations we obtain the square $f^n(P) = [-1.1, -0.9] \times [-0.1, 0.1]$ (compare Fig. 3).

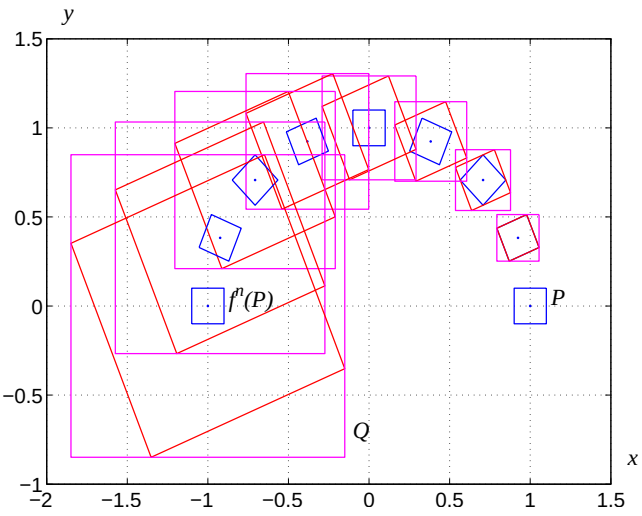

Fig. 3. Wrapping effect when computing iterations of $f(x, y) = (x\cos(\varphi) - y\sin(\varphi), x\sin(\varphi) + y\cos(\varphi))$ for $\varphi = \pi/8$

On the other hand when the trajectory $f^n(P)$ is computed using interval arithmetic, after each iteration the result is enclosed in an interval vector. The area of the enclosure grows by the factor $(\cos\varphi + \sin\varphi)^2$ in each iteration. The result Q obtained after $n = 8$ iterations is shown in Fig 3. The area of the square Q is $(\cos(\varphi) + \sin(\varphi))^{2n} \approx 72$ times larger that the area of the true result $f^n(P)$.

The second factor which may cause the overestimation is the variable dependency problem. If a certain variable is used many times in computations, the result may be overestimated. This can be easily seen by computing $2x - x$ for $x \in \mathbf{x}$, using interval arithmetic methods. For example, if $\mathbf{x} = [0, 1]$ then $2\mathbf{x} - \mathbf{x} = [0, 2] - [0, 1] = [-1, 2]$, while the true result is $[0, 1]$.

In the example presented above it is easy to rearrange the computations, so that the result obtained is exact. To see the variable dependency problem in a more complex situation, when there is no simple reduction of components let us compute

$$f(x_1, x_2) = \frac{-x_1 + 2x_2}{2x_1 + x_2} - 0.2x_2 \tag{1}$$

for $x_1 \in \mathbf{x}_1 = [0, 1]$, $x_2 \in \mathbf{x}_2 = [1, 3]$. Computations in interval arithmetic give $f(\mathbf{x}_1, \mathbf{x}_2) = (-[0, 1] + 2 \times [1, 3]) \div (2 \times [0, 1] + [1, 3]) - 0.2 \times [1, 3] = [1, 6] \div [1, 5] - [0.2, 0.6] = [0.2, 6] - [0.2, 0.6] = [-0.4, 5.8]$. The diameter of the result

is 6.2. It can be shown that the true result is $[-0.2 + 1/3, 1.8]$ with the diameter $5/3$. We observe a significant overestimation. This is a result of a multiple presence of the variables x_1 and x_2 in the evaluated expression.

Let us see, that the result may depend on the formula used. If we evaluate f using a mathematically equivalent formula $f(x_1, x_2) = (-x_1 + 2x_2 - 0.4x_1x_2 - 0.2x_2^2)/(2x_1+x_2)$, the result is $[-2, 5.8]$ with the diameter 7.8.

B. Reducing the overestimation

There are a number of methods to reduce or fully eliminate the overestimation. The most general technique is the division method. In order to evaluate the map f on the interval vector $\mathbf{x}$, we split $\mathbf{x}$ into smaller interval vectors $\mathbf{x}_i$, compute enclosures $\mathbf{f}_i$ of images of $\mathbf{x}_i$ under f and as a result take the interval $\mathbf{f} = \bigcup \mathbf{f}_i$. The main disadvantage of this method is that the computation time grows linearly with the number of boxes over which the map is evaluated. Results obtained by applying this method for the evaluation of (1) over $(\mathbf{x}_1, \mathbf{x}_2) = ([0, 1], [1, 3])$ using different number of boxes are reported in Table I. Even when the number of boxes is small (4 or 16) the improvement is significant. Also note that it is possible to obtain an arbitrarily good approximation but the number of boxes to improve the approximation grows very fast.

TABLE I
EVALUATION OF (1) USING THE DIVISION METHOD WITH DIFFERENT NUMBER OF BOXES

n	result	diameter
1	$[-0.4, 5.8]$	6.2
2×2	$[-0.15, 3.8]$	3.95
4×4	$[-0.015, 2.8]$	2.815
10×10	$[0.0725, 2.2]$	2.1275
100×100	$[0.1271, 1.84]$	1.7129
1000×1000	$[0.1327, 1.804]$	1.6713
10000×10000	$[0.13327, 1.8004]$	1.66713
true result	$[0.13333, 1.8]$	1.66667

Another approach to reduce the wrapping effect is based on the mean value form

$$\{f(x) \colon x \in \mathbf{x}\} \subset f(\hat{x}) + f'(\mathbf{x})(\mathbf{x} - \hat{x}), \tag{2}$$

where $\hat{x} \in \mathbf{x}$. Instead of computing $f(\mathbf{x})$ in interval arithmetic we choose a single point $\hat{x} \in \mathbf{x}$, find its image under f, compute the Jacobian matrix of f over $\mathbf{x}$, and finally use the formula (2) to find an enclosure. Usually, $\hat{x}$ is chosen to be the center of $\mathbf{x}$. This method works fine, when the input intervals are narrow. For example for $\mathbf{x} = [0.9, 1]$, $\mathbf{y} = [2.9, 3]$ the mean value form produces an interval with the diameter 0.0706, while the standard evaluation gives a result with the diameter 0.1451. It is interesting to note that one needs to divide this interval vector into more than 100 boxes to achieve the same accuracy using the division method. On the other hand, for large intervals the method does not improve the result. When $(\mathbf{x}_1, \mathbf{x}_2) = ([0, 1], [1, 3])$ the mean value form gives the result with the diameter 24.6, which is much worse than the standard interval arithmetic evaluation.

There are a number of methods which are aimed at reducing the wrapping effect for finding trajectories of dynamical systems. They are based on using different representations of the set enclosing the solution. Taylor expansions of first order are used in affine arithmetic [6]. The Lohner method described in Section III uses different types of parallelograms [5]. Special classes of polytopes have also been proposed [7]. One of the most promising techniques is based on Taylor models [8]. In this approach a function is represented on a given interval by the Taylor polynomial of a given length and an interval reminder term, which encloses the approximation error.

C. Automatic differentiation

In many applications it is necessary to compute derivatives of a map. In the context of finding enclosures of trajectories of dynamical systems derivatives are needed for example to evaluate the mean value form and to find coefficients of the Taylor expansion of the solution in case of continuous systems. Classical methods of finding derivatives include numerical and symbolic differentiation. In numerical differentiation, derivatives are calculated as difference quotients. This method does not provide enclosures of true derivatives, and therefore cannot be used if the goal is to compute exact results. Moreover, computing higher order derivatives using difference quotients leads to significant accuracy loss. The symbolic differentiation method works by first deriving symbolic formulas for derivatives, and then calculating derivatives using these formulas. The main disadvantages of this approach are increasing complexity when calculating higher order derivatives and low computation speed.

An alternative method to compute derivatives is based on automatic differentiation [9]. The evaluation of a map is decomposed into a sequence of elementary operations including the four basic elementary operations and elementary functions. Derivatives of elementary operations can be easily calculated by a computer program. These values are combined using the chain rule of differentiation. In order to see how this idea works in practice let us compute $\partial f/\partial x_1$ for f defined in (1). Let us assume that $x_1 = 1$ and $x_2 = 2$. The sequence of elementary operations to find $f(x_1, x_2)$ and to calculate $\partial f/\partial x_1$ at $(x_1, x_2) = (1, 2)$ is:

$$\begin{aligned}
z_1 &= x_1 = 1, & z_1' &= 1,\\
z_2 &= x_2 = 2, & z_2' &= 0,\\
z_3 &= -z_1 = -1, & z_3' &= -z_1' = -1,\\
z_4 &= 2z_2 = 4, & z_4' &= 2z_2' = 0,\\
z_5 &= z_3 + z_4 = 3, & z_5' &= z_3' + z_4' = -1,\\
z_6 &= 2z_1 = 2, & z_6' &= 2z_1' = 2,\\
z_7 &= z_6 + z_2 = 4, & z_7' &= z_6' + z_2' = 2,\\
z_8 &= z_5/z_7 = 0.75, & z_8' &= (z_5'z_7 - z_5z_7')/z_7^2 = 0.625,\\
z_9 &= 0.2z_2 = 0.4, & z_9' &= 0.2z_2' = 0,\\
z_{10} &= z_8 + z_9 = 1.15, & z_{10}' &= z_8' + z_9' = 0.625.
\end{aligned}$$

The initial values $z_1' = 1$ and $z_2' = 0$ are chosen so that a correct derivative is computed. For the computation of $\partial f/\partial x_2$ the initial values have to be exchanged. Observe that a computer stores the numerical values of the derivatives — no symbolic formula is created. Also note that the only information which is needed to compute derivatives is the formula how a function is computed.

There are a number of freely available software packages for interval computations. One of the first C and C++ packages available was the BIAS/Profil interval package [10]. This package can be easily combined with the FADBAD++ package for automatic differentiation [11]. The INTLAB [12] package provides easy access to interval arithmetic from the MATLAB environment.

One of the very well developed tools for interval analysis is the CAPD library. It is a collection of C++ modules designed for nonrigorous and validated computations for dynamical systems and to computation of homology of sets and maps [13], [14]. The CAPD library handles automatic differentiation internally (it also provides access to the FADBAD++ package), and the only input the user has to provide to study a dynamical system is the equation defining the map or the flow.

III. Validated studies of nonlinear systems

In this section, we present various interval analysis based methods, which can be used for rigorous studies of nonlinear systems. First, we discuss the problem how to find enclosures of trajectories.

A. Enclosing trajectories

Let us assume that $f\colon \mathbb{R}^m \mapsto \mathbb{R}^m$ is the map defining a discrete dynamical system. The problem of finding a trajectory based at x_0 is equivalent to computing subsequent iterations $x_{k+1} = f(x_k)$ for $k = 0, 1, \ldots$. Each set can be approximated with arbitrary precision by a union of interval vectors. Hence, without loss of generality, we may assume that the set of initial conditions for which we want to compute enclosures is a single interval vector. Let us denote it by $\mathbf{x}_0$. Below, we show how to find enclosures of sets $S_k = \{f^k(x_0)\colon x_0 \in \mathbf{x}_0\}$.

The simplest method is to evaluate the formula $\mathbf{x}_{k+1} = f(\mathbf{x}_k)$ in interval arithmetic for $k = 0, 1, \ldots$, where $f(\mathbf{x}_k)$ denotes the evaluation of the interval extension of f over the interval vector $\mathbf{x}_k$. Note that in this method, the enclosure of each of the sets S_k is an interval vector. As shown in the previous section, this may lead to huge overestimation (the wrapping effect).

Another option is to use the mean value form $\mathbf{x}_{k+1} = f(\hat{x}_k) + f'(\mathbf{x}_k)(\mathbf{x}_k - \hat{x}_k)$, where $\hat{x}_k \in \mathbf{x}_k$ is usually chosen as the center of $\mathbf{x}_k$. Note that the wrapping effect is still present — in each iteration the enclosure of the solution is represented by an interval vector.

Better results can be obtained by using different representations of the set S_k. This idea was proposed in [5] for the computation of trajectories of continuous systems and is called the Lohner algorithm. The set S_k is represented as an unevaluated sum $S_k \subset \hat{x}_k + \mathbf{r}_k$, where $\hat{x}_k$ is the center of the solution set. To initiate this representation, we compute $\hat{x}_0 = \mathrm{mid}(\mathbf{x}_0)$ and $\mathbf{r}_0 = \mathbf{x}_0 - \hat{x}_0$. Next, in each step we use the mean value form to find enclosures of S_{k+1}, i.e. we compute $\hat{x}_{k+1} = \mathrm{mid}(f(\hat{x}_k))$, $\mathbf{A}_k = f'(\hat{x}_k + \mathbf{r}_k)$, $\mathbf{r}_{k+1} = f(\hat{x}_k) - \hat{x}_{k+1} + \mathbf{A}_k\mathbf{r}_k$.

Different versions of the Lohner algorithm are obtained by using different representations of $\mathbf{r}_k$. When $\mathbf{r}_k$ is represented as an interval vector we obtain just the standard mean value form. This version will be referred to as the IV (interval vector) version. In other methods $\mathbf{r}_k$ is represented as a parallelogram, i.e. $\mathbf{r}_k = B_k\hat{\mathbf{r}}_k$, where B_k is an invertible matrix and $\hat{\mathbf{r}}_k$ is an interval vector. To start the computations, we choose the identity matrix $B_0 = \mathrm{I}$. Then, in each step we choose B_{k+1}, compute rigorously its inverse B_{k+1}^{-1}, and use the following update formula: $\hat{\mathbf{r}}_{k+1} = B_{k+1}^{-1}(f(\hat{x}_k) - \hat{x}_{k+1}) + (B_{k+1}^{-1}\mathbf{A}_k B_k)\hat{\mathbf{r}}_k$. In the first version (referred to as PAR, for parallelogram), we choose $B_{k+1} = \mathrm{mid}(\mathbf{A}_k B_k)$. This method often fails due to the necessity of computing B_{k+1}^{-1}, especially when the matrix B_k becomes ill conditioned. In the second version (referred in the following as the QR version), the matrix $\mathrm{mid}(\mathbf{A}_k B_k)$ is decomposed as $Q_{k+1}R_{k+1}$, where Q_{k+1} is orthogonal and R_{k+1} is upper triangular, and we choose $B_{k+1} = Q_{k+1}$. In this version $B_{k+1}^{-1} = Q_{k+1}^T$, so there is no need to compute the inverse of B_{k+1}. The last version is useful when the initial set has nonzero diameter. In this version, referred to as the IE version (internal enclosure), $\mathbf{r}_k$ is represented as the sum $\mathbf{r}_k = \mathbf{E}_k\mathbf{r}_0 + \tilde{\mathbf{r}}_k$. The first component is updated using the formula $\mathbf{E}_{k+1} = \mathbf{A}_k\mathbf{E}_k$ and $\tilde{\mathbf{r}}_k$ is handled using for example the QR version.

When we want to compute a trajectory of a single point a good alternative is to use multiple precision interval arithmetic. An example is shown in Section IV-A.

Let us now consider the initial value problem $x'(t) = f(x(t))$, $x(0) = x_0$, where $x(t) \in \mathbb{R}^m$ and $f\colon \mathbb{R}^m \mapsto \mathbb{R}^m$ is a smooth map. Let $\varphi(t, x_0)$ denote the solution of the initial value problem. Below, we briefly present rigorous numerical methods to compute enclosures of the set $\{\varphi(T, x_0)\colon x_0 \in \mathbf{x}_0\}$ for given $T > 0$ and $\mathbf{x}_0$.

To construct a rigorous integration method, we need a numerical method $x_{k+1} = \Phi(x_k, h_k)$, where h_k is the integration step, for which there exist formulas for the difference $\varepsilon(x_k, h_k)$ between the true solution $\varphi(h_k, x_k)$ and $\Phi(x_k, h_k)$, i.e. $\varphi(h_k, x_k) = \Phi(x_k, h_k) + \varepsilon(x_k, h_k)$. Usually, the error term $\varepsilon(h_k, x_k)$ depends on some unknown parameters but we can compute its enclosure. The most commonly used integration method in he context of validated computations is the Taylor method which in 1D can be written as $\Phi(x, h) = x + hx' + h^2x^{(2)}/2 + \ldots + h^n x^{(n)}/n!$ with the error term $\varepsilon(x, h) = h^{n+1}x^{(n+1)}(\theta h)/(n+1)!$, where $\theta \in [0, 1]$. Although in general the value of θ is unknown, we can easily compute bounds for the error term $\varepsilon(x_k, h_k)$ having an enclosure of the set $\{\varphi(t, x_k)\colon t \in [0, h_k], x_k \in \mathbf{x}_k\}$.

Each integration step starts with finding a *rough enclosure* $\mathbf{y}_k$, containing trajectories $\{\varphi(t, x_k)\colon t \in [0, h_k], x_k \in \mathbf{x}_k\}$. As a candidate for this set one can

choose $\mathbf{y}_k = \mathbf{x}_k + [0, h_k] f(\mathbf{x}_k)$. If $\mathbf{x}_k + [0, h_k] f(\mathbf{y}_k) \subset \mathbf{y}_k$ then $\{\varphi(t, x_k) \colon t \in [0, h], x_k \in \mathbf{x}_k\} \subset \mathbf{y}_k$, and $\mathbf{y}_k$ is a rough enclosure. If not, we may inflate $\mathbf{y}_k$ or choose a smaller integration step h_k and try to verify the condition $\mathbf{x}_k + [0, h_k] f(\mathbf{y}_k) \subset \mathbf{y}_k$ again.

Once the rough enclosure $\mathbf{y}_k$ is known, the continuous system may be integrated rigorously using the formula $\mathbf{x}_{k+1} = \Phi(\mathbf{x}_k, h_k) + \varepsilon(\mathbf{y}_k, h_k)$. To reduce the wrapping effect, the Lohner method may by used. In this method $\Phi(\mathbf{x}_k, h_k)$ is computed using the mean value form $\Phi(\hat{x}_k, h_k) + \partial\Phi/\partial x(\mathbf{x}_k, h_k)(\mathbf{x}_k - \hat{x}_k)$ and different representations of $\mathbf{x}_k$ are used. Note, that in order to use the Taylor method of order p we need formulas for the derivatives $x^{(k)}$ of the solution up to order $p+1$ and also formulas for derivatives $\partial\Phi/\partial x$ of the Taylor method. These derivatives can be found using automatic differentiation. The only formula which needs to be supplied to the computation procedure is the formula for the right hand side f of the differential equation.

Let us note that in order to use an integration method of order p, the vector field has to be of class C^{p+1}. It follows, that integration methods presented above cannot be used directly for the integration of piecewise linear (PWL) systems (and more generally for piecewise smooth systems). Now, we briefly discuss how to handle this case. Let us assume that the vector field f is a PWL continuous map and that linear regions are separated by the planes $\Sigma_1, \Sigma_2, \ldots$ called in the following the C^0-hyperplanes.

Clearly, when trajectories remain in a single linear region, then general integration methods developed for smooth systems can be used without any modifications. When intersections of trajectories with the C^0-hyperplanes are transversal it is possible to extend methods developed for smooth systems to integration of PWL systems. This is achieved by using the C^0-hyperplanes as transversal sections. When a trajectory intersects a C^0-hyperplane, its intersection with the plane is computed and the result is used as a set of initial conditions for further computations. Another option is to treat the PWL system as a perturbed linear system and use theory of differential inclusions to obtain estimates for solutions of the nonlinear system. This method allows one to handle trajectories which are tangent to the C^0-hyperplanes (for details see [15], [16]).

Sometimes, it is necessary to integrate trajectories passing arbitrarily close to an unstable equilibrium. Since the integration time to pass a neighborhood of the unstable equilibrium is not bounded, this case has to be treated in a different way. First, a cube around the equilibrium is selected. If the trajectory hits the cube the computations are interrupted. We then change to the normal form coordinates and explicitly compute the exit set from the cube. There are two ways in which a box can pass through the cube. If the box intersects the stable manifold of the equilibrium then it is split along the line of intersection, and exits the cube in two pieces. Otherwise, the box flows out in one piece. After leaving the cube, we switch back to the original coordinates, and resume numeric computations. For a detailed description of this procedure see [17]. An application to PWL systems is presented in [18].

B. Graph representation of the dynamics

In this section we present methods which allow us to study global dynamics of the system. The region of the state space where the dynamics is studied is covered by boxes. Usually as the region of interest one selects a trapping region enclosing the numerically observed attractor. Using methods presented in the previous section, for each box we find a set of boxes containing its image. This information is represented as a directed graph, where boxes are graph vertices and non-forbidden transitions are graph edges. This method is in some sense similar to the nonrigorous method of generalized cell mappings [19], where with each cell (box) probabilities of going to other cells are associated. For a similar concept of analysing structural features of complex systems see [20].

Based on the graph structure we can gather a lot of information on the global dynamics of the original system. One can find accurate enclosures of the region where interesting dynamics takes place, find the approximate structure of the attractor, locate homoclinic and heteroclinic orbits. One can also find an enclosure of the set containing all periodic orbits of a given length or the invariant part of a given set.

The algorithms described here operate on objects called ε–boxes. Let $\varepsilon = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m) \in \mathbb{R}^m$, where m is the dimension of the state space. An *ε–box* is a an interval vector with corners lying on a regular grid: $\mathbf{v} = ([k_1\varepsilon_1, (k_1+1)\varepsilon_1], [k_2\varepsilon_2, (k_2+1)\varepsilon_2], \ldots, [k_m\varepsilon_m, (k_m+1)\varepsilon_m])$, where k_i are integer numbers. ε–boxes are well suited for rigorous computations. Using different ε it is possible to represent a given set by ε–boxes with arbitrary precision. When ε is fixed an ε–box is defined uniquely by a sequence of integer numbers. Such a representation makes it possible to use different combinatorial algorithms for operations on sets of ε–boxes.

Finding the invariant part of a set containing the attractor is an important step in studies of dynamical systems. It allows one to find a region, where interesting dynamics can take place. We say that x belongs to the *invariant part* of a set Ω under f if there exist a trajectory $(x_k)_{k=-\infty}^{\infty}$, enclosed in Ω such that $x = x_0$.

Algorithms for finding the invariant part of a given set are relatively simple, and have been described in many papers. A generalized bisection technique used for the computation of invariant sets, invariant measures and unstable manifolds was described in [21]. A combinatorial procedure for finding invariant parts, isolating neighbourhoods, and index pairs is described in [22].

Now, we present a simple algorithm which for a given set Ω computes an enclosure of its invariant part (compare also [23]). In order to find an invariant part of Ω we choose $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_m)$ and cover Ω by ε–boxes. The set $V = \{\mathbf{v}_i\}$ of ε–boxes is the vertex set of the graph. Non-forbidden transitions between boxes constitute the edge

set $E = \{(\mathbf{v}_i, \mathbf{v}_j)\colon f(\mathbf{v}_i) \cap \mathbf{v}_j \neq \emptyset\}$. The algorithm to improve the enclosure is following. The box $\mathbf{v}_i$ is removed from the graph if it is not the beginning of any edge ($\{(\mathbf{v}_i, \mathbf{v}_j)\colon \mathbf{v}_j \in V\} = \emptyset$) or if it is not the ending of any edge ($\{(\mathbf{v}_j, \mathbf{v}_i)\colon \mathbf{v}_j \in V\} = \emptyset$). Removing boxes is continued until no more boxes can be removed from the graph.

The process of finding the invariant part is often combined with the subdivision algorithm. Each box is split into several smaller boxes and the computations (finding connections and removing boxes not belonging to the invariant part) are repeated. This process is continued until the required accuracy of the covering is achieved.

C. Periodic orbits

In this section, we discuss methods to study the existence of fixed points and periodic orbits using interval analysis tools. Fixed points and periodic orbits represent long term behavior of dynamical systems and are the simplest examples of limit sets. Periodic orbits are especially important in the analysis of chaotic system, which under certain assumptions are characterized by the existence of infinitely many periodic orbits embedded within the attractor. The structure of a chaotic attractor is built on the set of unstable periodic orbits which are ordered hierarchically. Shorter orbits provide a coarse description of an attractor, while longer orbits reveal its finer structure [24], [25]. Short periodic orbits can be used to characterize the attractor. Using the number and lengths of short periodic orbits one can compute good approximations of the topological entropy and the dimension of the attractor.

The basic numerical technique for locating periodic orbits is based on the Newton method for searching for zeros of nonlinear maps. If $f\colon \mathbb{R}^m \mapsto \mathbb{R}^m$ is a differentiable map then the Newton iteration is defined by

$$x_{k+1} = \mathrm{N}(x_k) = x_k - f'(x_k)^{-1} f(x_k), \tag{3}$$

where $f'(x)$ is the Jacobian matrix of f at x, and x_0 is the initial point.

Fig. 4 shows an example of the Newton iteration process for the map $f(x) = x^2 - 2$. The initial point is selected as $x_0 = 0.6$. One can see that the convergence is very fast. After three iterations $x_3 \approx 1.41623$, and the relative error in finding the position of the true zero of f (which is $\sqrt{2}$) is less than 0.0015. After six iterations the error is of the order of double precision machine error. Generally, if the initial point is sufficiently close to the unknown zero, the Newton method converges quadratically, i.e. the number of correct digits is doubled in each iteration.

In order to find a period–p orbit of f one can apply the Newton method for the map $g(x) = x - f^p(x)$. In order to find many (hopefully all) short periodic orbits one can use the Newton method with different initial conditions.

Due to rounding errors which are inevitable in numerical simulations it is not sure that there is a real periodic orbit in a neighborhood of the computer generated one.

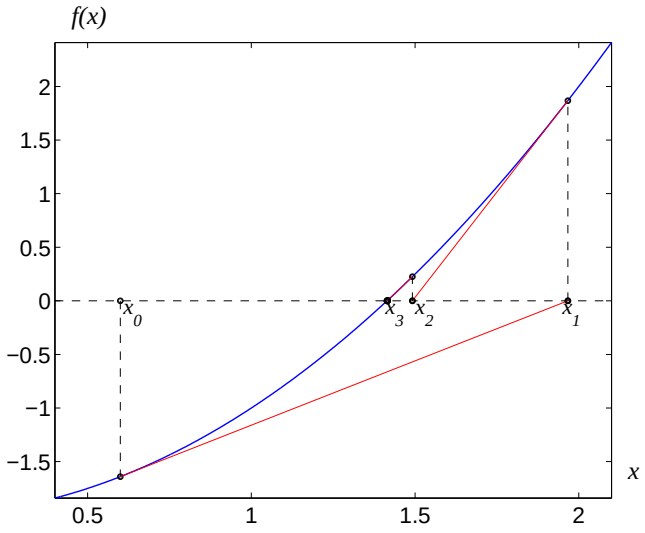

Fig. 4. Newton operator to find a zero of $f(x) = x^2 - 2$, $x_0 = 0.6$

Proving the existence of periodic orbits is usually a non-trivial task. This can be done analytically only for some simple dynamical systems.

D. Interval operators for proving the existence of zeros

In this section, we present interval operators for proving the existence of zeros of nonlinear maps, and we show how to use them to prove the existence of periodic orbits. Let us assume that $f : \mathbb{R}^m \mapsto \mathbb{R}^m$ is a twice continuously differentiable map. The *interval Newton operator* evaluated at the interval vector $\mathbf{x}$ is defined by

$$\mathrm{N}(\mathbf{x}) = \hat{x} - f'(\mathbf{x})^{-1} f(\hat{x}), \tag{4}$$

where $f'(\mathbf{x})$ is an interval matrix containing the Jacobian matrices $f'(x)$ for $x \in \mathbf{x}$ and $\hat{x}$ is an arbitrary point from $\mathbf{x}$. Usually, one chooses $\hat{x}$ to be the center of $\mathbf{x}$.

The following theorem [26], [27] can be used to study the existence and uniqueness of zeros:

Theorem 1: Let $\mathbf{x}$ be an interval vector, $\hat{x} \in \mathbf{x}$ and let us assume that $f'(\mathbf{x})^{-1}$ exists.

(a) If $\mathrm{N}(\mathbf{x}) \cap \mathbf{x} = \emptyset$, then $\mathbf{x}$ contains no zeros of f,
(b) If $\mathrm{N}(\mathbf{x}) \subset \mathbf{x}$, then $\mathbf{x}$ contains exactly one zero of f.

As an example to see how the interval Newton method works in practice, let us use it to prove the existence of a single zero of a function $f(x) = x^2 - 2$ in the interval $\mathbf{x} = [0.5, 2]$. In this case we have

$$\begin{aligned}
&\hat{x} = 1.25, \quad f(\hat{x}) = -0.4375,\\
&f'(\mathbf{x}) = 2\mathbf{x} = [1, 4], \quad f'(\mathbf{x})^{-1} = [0.25, 1],\\
&\mathrm{N}(\mathbf{x}) = \hat{x} - f'(\mathbf{x})^{-1} f(\hat{x})\\
&\qquad = 1.25 - [0.25, 1] \cdot (-0.4375)\\
&\qquad = 1.25 - [-0.4375, -0.109375]\\
&\qquad = [1.359375, 1.6875].
\end{aligned}$$

Since $\mathrm{N}(\mathbf{x}) = [1.359375, 1.6875] \subset \mathbf{x} = [0.5, 2]$, it follows from Theorem 1 that $\mathbf{x}$ contains exactly one zero of f. A plot of the function f is shown in Fig. 5.

Derivatives of f at the endpoints of $\mathbf{x}$ define slopes of straight lines which pass through the point $(\hat{x}, f(\hat{x}))$ and intersect the horizontal axis at the endpoints of the interval $\mathrm{N}(\mathbf{x})$.

Once the existence condition is satisfied, one may iterate the interval Newton operator to obtain a sequence of nested intervals containing the solution: $\mathbf{x}_0 = \mathbf{x}$, $\mathbf{x}_{k+1} = \mathrm{N}(\mathbf{x}_k)$, for $k \geq 0$. In the example considered the convergence is very fast. The diameter $d_k = \mathrm{diam}(\mathbf{x}_k)$ of the interval $\mathbf{x}_k$ enclosing zero after k iterations is $d_0 = 1.5$, $d_1 \approx 0.328$, $d_2 \approx 0.0229$, $d_3 \approx 4.34 \cdot 10^{-5}$, and $d_4 \approx 7.23 \cdot 10^{-11}$.

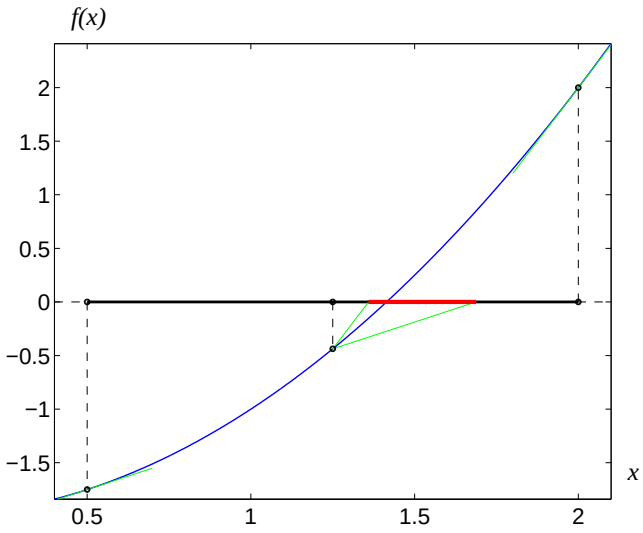

Fig. 5. Interval Newton operator to prove the existence of a single zero of $f(x) = x^2 - 2$ in the interval $\mathbf{x} = [0.5, 2]$. $\hat{x} = 1.25$, $\mathrm{N}(\mathbf{x}) = [1.359375, 1.6875]$

There are other interval operators, which may be used to study the existence of zeros of nonlinear maps, for example the Krawczyk operator and the Hansen–Sengupta operator. The Krawczyk operator $\mathrm{K}(\mathbf{x}) = \hat{x} - Cf(\hat{x}) + (I - Cf'(\mathbf{x}))(\mathbf{x} - \hat{x})$ is obtained by using the mean value form for modified Newton operator $\mathrm{N}(x) = x - Cf(x)$, where C is an invertible preconditioning matrix. C is usually chosen as the inverse of the Jacobian matrix $f'(\hat{x})$. The Hansen–Sengupta operator is based on the Gauss–Seidel iterative method for solving linear systems of equations.

These two operators have similar properties as the Newton operator (compare [26]). Their advantage is that they do not involve inversion of interval matrices, and in certain cases produce narrower enclosures of zeros.

E. Existence of periodic orbits

To study the existence of period–p orbits of f one can apply an interval operator to the map $g = \mathrm{id} - f^p$, where id denotes the identity map.

The second possibility is to apply an interval operator to the map $G : (\mathbb{R}^m)^p \mapsto (\mathbb{R}^m)^p$ defined by $[G(z)]_k = x_{(k+1) \bmod p} - f(x_k)$ for $k = 0, 1, \ldots, p-1$, where $z = (x_0, x_1, \ldots, x_{p-1})^{\mathrm{T}}$. Let us note that z is a zero of G if and only if x_0 is a fixed point of f^p. The problem of the existence of periodic orbits is converted to the problem of the existence of zeros of a higher dimensional function.

Once an interval vector containing a periodic orbit is known, a better approximation of the position of the orbit can be found by iterating the interval operator. It is also possible to find the Jacobian matrix and verify stability of the orbit.

F. Finding all short periodic orbits

All period–p cycles of f enclosed in a given set Ω can be found using a combination of the method described above and the generalized bisection technique (compare also [28]).

First, the region Ω is covered by a finite number of interval vectors. Then for each interval vector $\mathbf{x}$ the interval operator $\mathrm{N}(\mathbf{x})$ for the map $g = \mathrm{id} - f^p$ is evaluated. Finally, we use the Theorem 1 to prove that there is exactly one fixed point of f^n in $\mathbf{x}$ or that there are no fixed points of f^n in $\mathbf{x}$. If none of the two conditions hold, the interval vector $\mathbf{x}$ is split into smaller parts and the computations are repeated.

Note that the graph representation of the dynamics over the set Ω can help us in locating short periodic orbits (see also [29]). To find all period-p orbits we first find all period–p cycles in the graph (each cycle may correspond to a periodic orbit of the dynamical system). Then, for each cycle, we evaluate an interval operator on the interval vector corresponding to the cycle under study. This method is very useful for finding all short periodic orbits in case of flows, when the time needed for the evaluation of the Poincaré map is significant (compare [30]).

G. Symbolic dynamics and other tools

Symbolic dynamics approach is one of common methods for the characterization of systems trajectories [31]. In this approach, the state space is divided into disjoint sets $N_1, N_2 \ldots, N_p$ and with each trajectory (x_k) a sequence of symbols (s_k) is associated in such a way that $x_k \in N_{s_k}$.

Let us now define what is understood by the existence of symbolic dynamics of a certain type for the map $f \colon \mathbb{R}^m \mapsto \mathbb{R}^m$. Let $A = (a_{ij})_{i,j=1}^p$ be a matrix with entries 0 or 1. The *subshift of finite type* with the transition matrix A is the map $\sigma_A = \sigma|\Sigma_A$, where $\sigma \colon \Sigma_p \mapsto \Sigma_p$ is the shift operator (i.e. $(\sigma(s))_i = s_{i+1}$) defined on the set $\Sigma_p = \{(\ldots, s_{-1}, s_0, s_1, s_2 \ldots) \colon s_k \in \{1, 2, \ldots, p\}\}$ of bi-infinite sequences, and $\Sigma_A = \{s \in \Sigma_p \colon a_{s_k s_{k+1}} = 1 \text{ for all } k\}$, The map $\sigma_A = \sigma|\Sigma_A$ is the shift operator restricted to the set Σ_A.

We say that the map f is *semiconjugate with a subshift of finite type* σ_A if there exists a homeomorphism $h \colon \Omega \mapsto \Sigma_A$, with $\Omega \subset \mathbb{R}^m$ such that $h \circ f = \sigma_A \circ h$. The existence of h means that there is a one-to-one relation between trajectories in Ω and a sequence of symbols in Σ_A. The existence of symbolic dynamics means that the dynamics of f is at least as complicated as the dynamics of σ_A.

From the existence of nontrivial symbolic dynamics one can conclude that the map is chaotic in the topological sense, i.e. that its topological entropy is positive. More precisely if f is semiconjugate with a subshift with the

transition matrix A then the topological entropy $\mathrm{H}(f)$ of f is not less than the dominant eigenvalue of A, i.e. $\mathrm{H}(f) \geq \lambda$, where λ is an eigenvalue of A such that $\lambda \geq |\lambda_j|$ for all eigenvalues λ_j of A.

There are a number of methods which can be used for proving the existence of symbolic dynamics. A simple connectivity based method was described in [32]. A topological method, which can be used to prove the existence of symbolic dynamics is based on the notion of a covering relation [33], [34]. In this method one has to prove that sets supporting the symbolic dynamics cover topologically each other. Each covering corresponds to a non-zero entry in the transition matrix. Some examples are given in the following sections.

Other rigorous computational methods for studying nonlinear systems include techniques to find enclosures of stable and unstable invariant manifolds, proving the existence of chaos in the sense of Shilnikov [35], proving the existence of homoclinic tangencies, proving the existence of certain types of bifurcations [36]. In certain cases it is also possible to prove the existence of a chaotic attractor, which is generally a much more complicated problem than proving the existence of nontrivial symbolic dynamics. In [17] the existence of the Lorenz attractor was proved. The method used requires hyperbolicity of the considered dynamical system, and can be applied only to a narrow class of nonlinear systems.

H. Application of the methods for continuous time systems

All the methods described in the previous sections can be applied to the analysis of continuous systems using the Poincaré map technique. This is a general method which reduces problems concerning continuous dynamical systems to the corresponding problems for discrete systems.

Let Σ be the union of hyperplanes Σ_1, Σ_2,..., Σ_l. The *Poincaré map* $P : \Sigma \mapsto \Sigma$ is defined as $P(x) = \varphi(\tau(x), x)$, where $\varphi(t, x)$ is the trajectory of the system based at x, and $\tau(x)$ is the time needed for the trajectory $\varphi(t, x)$ to reach Σ. The Poincaré map and its derivatives can be evaluated using rigorous integration methods described in previous sections.

IV. The Hénon map

As the first example let us consider the Hénon map [37]:

$$h(x, y) = (1 + y - ax^2, bx). \tag{5}$$

For the classical parameter values $a = 1.4$, $b = 0.3$ the Hénon attractor is observed (see Fig. 6). The quadrangle Ω, with corners $(-1.33, 0.42)$, $(1.32, 0.133)$, $(1.245, -0.14)$, and $(-1.06, -0.5)$, is a trapping region for h, i.e., $h(\Omega) \subset \Omega$. Ω contains the numerically observed attractor. The Hénon map has two fixed points. One of them belongs to Ω, while the second one lies outside Ω (compare Fig. 6).

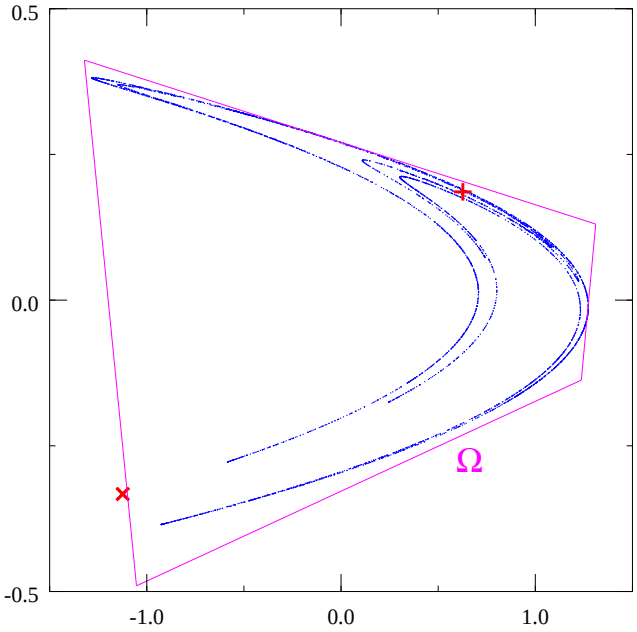

Fig. 6. A trajectory of the Hénon map composed of 10000 points, the trapping region Ω, and two unstable fixed points: ("+", "×")

A. Rigorous computations of trajectories

Let us first show how to compute very accurate enclosures of true trajectories. Let us assume that the initial point is $(x_0, y_0) = (0, 0)$ and standard parameter values $a = 1.4$, $b = 0.3$ are used. Fig. 7(a) shows the first coordinate of the trajectory found using double precision nonrigorous computations (red star symbols) and validated enclosure of the first coordinate found using interval analysis (blue intervals). Computations involve representing parameter values 0.3 and 1.4 as intervals with nonzero diameter (the numbers 1.4 and 0.3 are not machine representable) and evaluating the Hénon map formula in interval arithmetic. Around iteration number $k = 57$ the diameter of the interval becomes visible, and for $k \geq 66$ it is larger than the size of the attractor. This is a classical example of the wrapping effect. Theoretically, the result $h^k(x_0, y_0)$ is a point. However, due to rounding errors and their propagation, the diameter of the enclosure grows very fast and after a certain number of iterations the results obtained using interval analysis become useless.

Some reduction of the wrapping effect can be achieved by using methods presented in Sec. II-B. However, when the set of initial conditions is small (a point in our case) much more accurate results can be achieved by using multiple precision interval arithmetic. Most of the modern interval packages provide implementations or interface to multiple precision computations, for example the GNU Multiple Precision Arithmetic Library. Fig. 7(b) shows how the diameter of the enclosure of the first coordinate changes with the iteration number k for various precisions. Results for representations of length 64, 128, and 256 bits are plotted. For comparison, results for the standard interval arithmetic based on the double precision computations

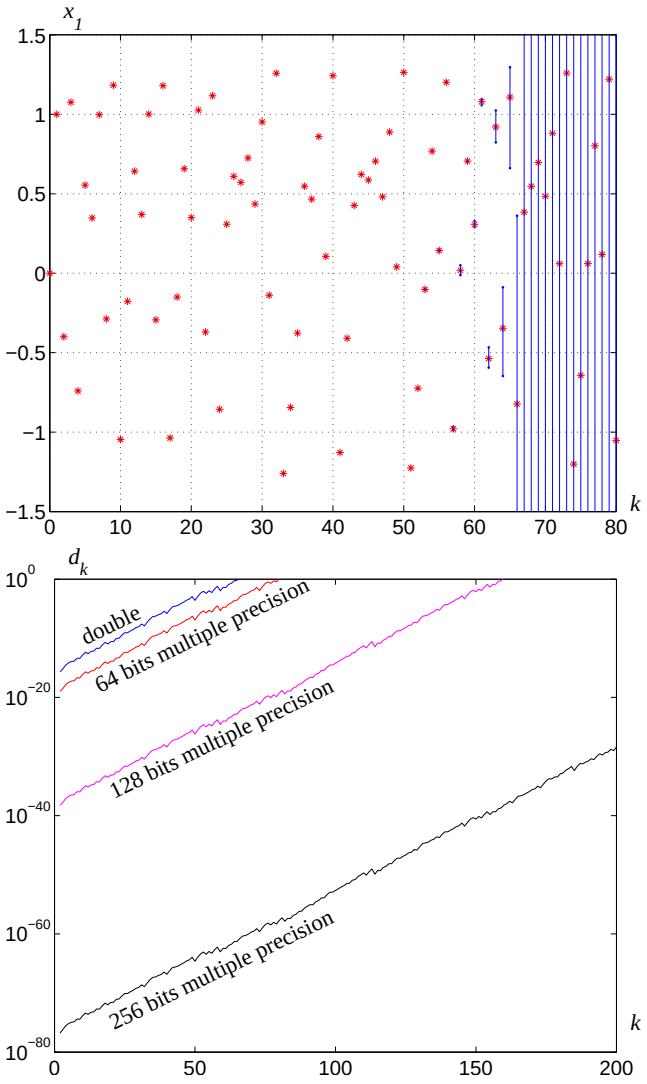

Fig. 7. (a) the first coordinate of the trajectory of the Hénon map with $a = 1.4$, $b = 0.3$ based at $(0, 0)$ computed nonrigorously in double precision (red stars) and rigorous enclosures of the first coordinate found using interval analysis (blue intervals), (b) the diameter of the validated enclosure of the first coordinate found using double and multiple precision interval arithmetic

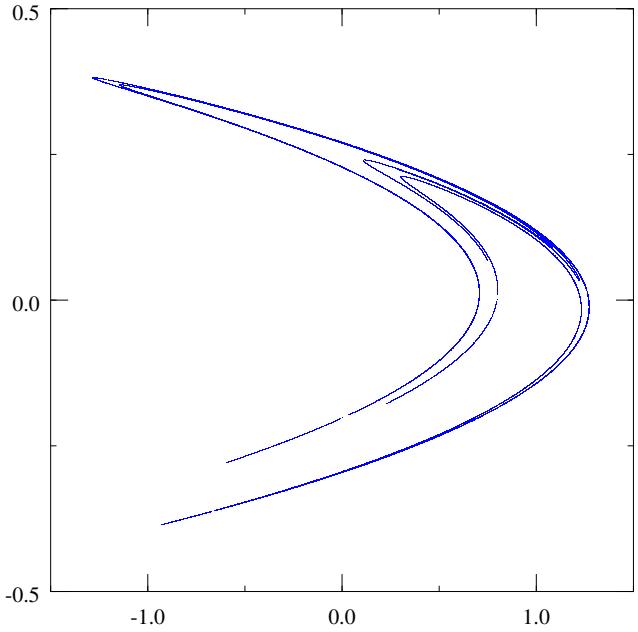

Fig. 8. Periodic orbits with period $p \leq 30$ within the trapping region Ω

are shown. One can see that the diameter of the representation grows almost linearly in the logarithmic scale. Based on this observation one can easily decide which representation should be used to achieve required accuracy after a specified number of iterations. In the example shown, the diameter of the enclosure after 200 iterations is close to 10^{-30}, when using representations composed of 256 bits. This representation was used to compute the distance between the double precision trajectory and the true trajectory reported in the Introduction.

B. Short periodic orbits

In this section, we present results on the existence of short periodic orbits.

First, let us show that for $a = 1.399999486944$, $b = 0.3$ the period-33 sink presented in Fig. 2 exists. The proof is carried out in the following steps. First, we take a pseudo-periodic orbit $(x_0, y_0), (x_1, y_1), \ldots (x_{p-1}, y_{p-1})$ of length $p = 33$ found by iterating the Hénon map. This pseudo-periodic orbit with the initial point $(x_0, y_0) = (0.653621903129, 0.0720266134686)$ serves as an initial guess of the position of the true orbit. Then, by applying the standard (non-interval) Newton operator the approximation is improved. Finally, we define the interval vector $(\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_{p-1})$ around this position by using the formula $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{y}_i) = ([x_i - r, x_i + r], [y_i - r, y_i + r])$ with $r = 10^{-9}$ and prove the existence of a single periodic orbit within this interval vector by applying the interval Newton operator. To show that the orbit is stable, we evaluate the Jacobian matrix $h'(\mathbf{z}_{p-1}) \cdots h'(\mathbf{z}_1) \cdot h'(\mathbf{z}_0)$ and verify that its eigenvalues are within the unit circle.

Now, let us consider the case $(a, b) = (1.4, 0.3)$. The problem of existence of short periodic orbits was studied using the Krawczyk operator combined with the generalized bisection procedure. For larger periods this operator is faster than other operators (compare [38]). All periodic orbits with period $p \leq 30$ have been found (see Fig. 8). There are exactly 109033 periodic orbits with period $p \leq 30$ and 3065317 points belonging to these orbits. In particular, it was confirmed that there are no period–3 and period–5 orbits within the trapping region and that there are periodic orbits with all other periods $p \leq 30$. The results obtained are summarized in Table II, where Q_p is the number of periodic orbits with period p, and P_p is the number of fixed points of h^p.

Short periodic orbits give a good approximation of the attractor. One can see small gaps in the plot when compared to the numerically observed attractor (see Fig. 8).

Results on short periodic orbits can be used to obtain estimates for the topological entropy for the Hénon map. $\mathrm{H}_p = p^{-1} \log(\mathrm{P}_p)$ is an estimate of the topological

TABLE II
THE NUMBER Q_p OF PERIOD–p ORBITS FOR THE HÉNON MAP

p	Q_p	P_p	H_p	p	Q_p	P_p	H_p
1	1	1	0.00000	16	102	1695	0.46471
2	1	3	0.54931	17	166	2823	0.46739
3	0	1	0.00000	18	233	4263	0.46432
4	1	7	0.48648	19	364	6917	0.46535
5	0	1	0.00000	20	535	10807	0.46440
6	2	15	0.45134	21	834	17543	0.46535
7	4	29	0.48104	22	1225	27107	0.46398
8	7	63	0.51789	23	1930	44391	0.46525
9	6	55	0.44526	24	2902	69951	0.46481
10	10	103	0.46347	25	4498	112451	0.46521
11	14	155	0.45849	26	6806	177375	0.46485
12	19	247	0.45912	27	10518	284041	0.46507
13	32	417	0.46408	28	16031	449519	0.46485
14	44	647	0.46231	29	24740	717461	0.46495
15	72	1081	0.46571	30	37936	1139275	0.46486

entropy based on the number of periodic orbits with period p. One can see that the values $H_p(h)$ are almost constant for $p \geq 10$. This allows us to state the hypothesis that the topological entropy of the Hénon map is close to $H(h) \approx 0.465$.

C. The invariant part

Let us now compute the invariant part of the set $B = [-1.5, 1.5] \times [-0.5, 0.5]$. This set contains the trapping region Ω and both unstable fixed point (compare Fig. 6).

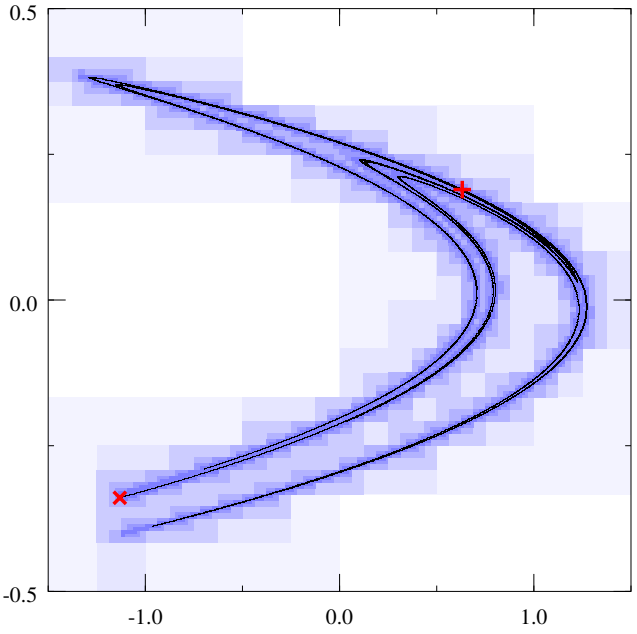

Fig. 9. The invariant part of $B = [-1.5, 1.5] \times [-0.5, 0.5]$ for the Hénon map, unstable fixed points ($+$, $\times$)

We have found ε–boxes covering the invariant part of B for $\varepsilon = (1/2^n, 1/(3 \cdot 2^n))$, with $n = 1, 2, \ldots, 11$. Fig. 9 shows results obtained for different ε (darker color corresponds to larger n — more accurate covering). The covering of the invariant part contains the chaotic attractor, the fixed point lying outside of Ω and one branch of the unstable manifold of this fixed point connecting it to the attractor.

The area of the set B is 3, while the area of the covering containing the invariant part is smaller than 0.01. With such a fine representation of the attractor the area of the region containing complicated dynamics is significantly reduced.

D. Symbolic dynamics and bounds for topological entropy

The problems of the existence of symbolic dynamics and obtaining bounds for the topological entropy for the Hénon map were studied in many research papers [39], [40], [41], [34], [42].

In [34], using the method of covering relations the existence of the full shift dynamics on two symbols for h^7 was proved. The full shift corresponds to a transition matrix with no zero entries, i.e. $a_{ij} = 1$ for $i, j = 1, 2$. The dominant eigenvalue of the transition matrix is 2, and hence a lower bound of the topological entropy of h is $H(h) \geq (\log 2)/7 > 0.099$.

Using the same method, one can prove the existence of the golden mean shift for h^2. The sets supporting the symbolic dynamics and their images under h^2 are shown in Fig 10(a). One can see that the image of the first set covers both sets (i.e. its image intersects vertical edges of both sets, it has empty intersection with horizontal edges, and images of vertical edges of the first set lie to the left and to the right of both sets), while the second one covers the first one. Based on this information the transition matrix A has the following entries $a_{11} = a_{12} = a_{21} = 1$, $a_{22} = 0$. Its dominant eigenvalue is $0.5(\sqrt{5} + 1)$, which leads to the following bound for the topological entropy: $H(h) \geq 0.5 \log(0.5(\sqrt{5} + 1)) > 0.24$.

In [42] the existence of symbolic dynamics on five symbols for different iterates of h was proved. This symbolic dynamics gives the bound $H(h) > 0.338$ for the topological entropy.

It is possible to further increase the lower bound of the topological entropy by constructing more complicated symbolic dynamics. Fig 10(b) shows sets supporting symbolic dynamics on 29 symbols involving 46 covering relations. The resulting bound of the topological entropy is $H(h) > 0.43$. Other methods can also be used to obtain bounds for the topological entropy. In [43], rigorously found enclosures of stable and unstable manifolds of unstable periodic points were used to show that $H(h) > 0.46469$.

V. THE CHUA'S CIRCUIT

The dynamics of the Chua's circuit [44] shown in Fig. 11 is governed by a third order ordinary differential equation:

$$\begin{aligned} C_1 \dot{x}_1 &= (x_2 - x_1)/R - g(x_1), \\ C_2 \dot{x}_2 &= (x_1 - x_2)/R + x_3, \\ L \dot{x}_3 &= -x_2 - R_0 x_3, \end{aligned} \tag{6}$$

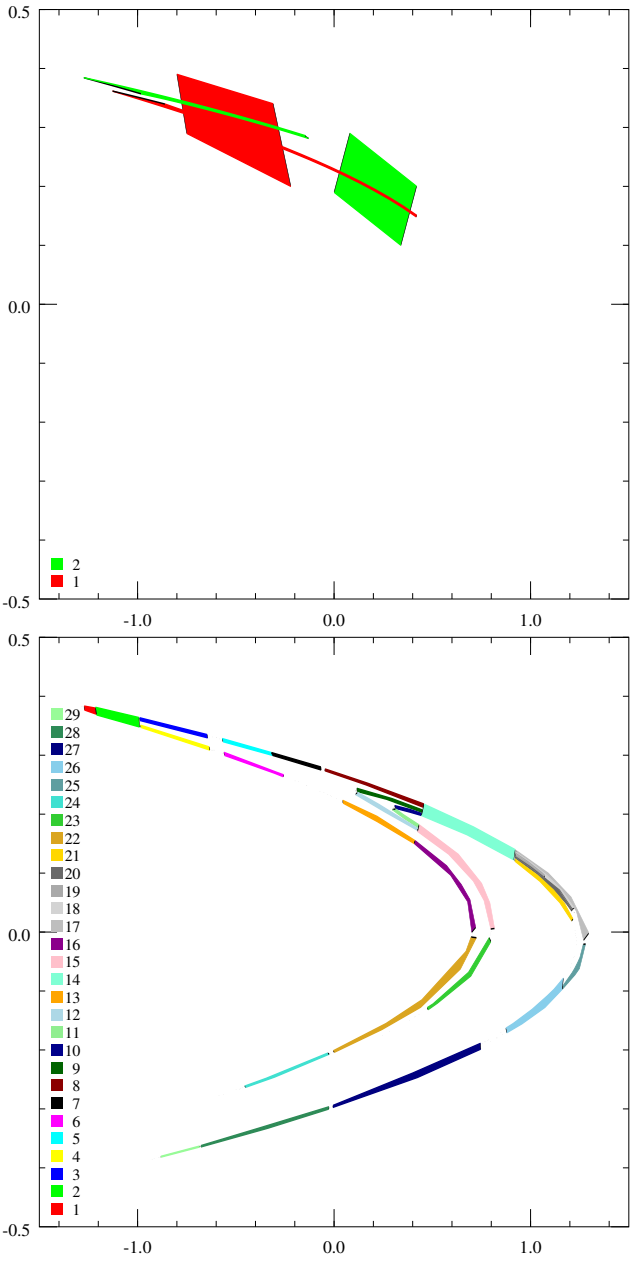

Fig. 10. (a) symbolic dynamics for h^2, quadrangles N_1, N_2 and their images, (b) symbolic dynamics on 29 symbols

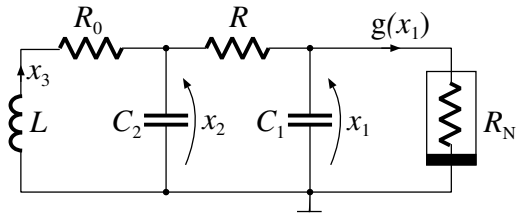

Fig. 11. The Chua's circuit

(see Fig. 12(b)).

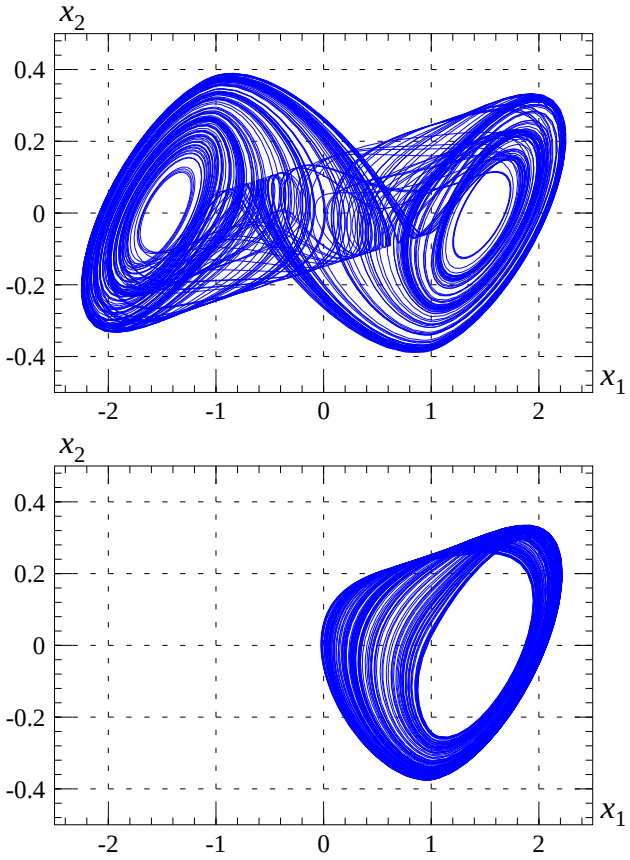

Fig. 12. Trajectories of the Chua's circuit with the PWL nonlinearity; (a) $C_2 = 9.3515$, the double-scroll attractor; (b) $C_2 = 7.65$, the spiral attractor

where $g(\cdot)$ is a three-segment piecewise linear characteristic $g(z) = G_b z + 0.5(G_a - G_b)(|z+1| - |z-1|)$. There are a number of interesting dynamical phenomena observed for this system. For different values of the parameters one can observe stable fixed points, stable periodic orbits, period-doubling bifurcations, and different types of attractors including the spiral attractor, the double scroll attractor and the double hook attractor [44].

In this work, the system (6) is considered with the following parameter values (after appropriate parameter rescaling): $C_1 = 1$, $G_a = -3.4429$, $G_b = -2.1849$, $L = 0.06913$, $R = 0.33065$, $R_0 = 0.00036$. For $C_2 = 9.3515$ the double-scroll attractor is observed (see Fig. 12(a)) and for $C_2 = 7.65$ the spiral attractor exists

When the nonlinearity is of a cubic type $g(z) = g_1 z + g_2 z^3$, the double scroll attractor is observed for $C_1 = 0.7$, $C_2 = 7.8$, $L = 1.891$, $R = 2$, $R_0 = 0.01499$, $g_1 = -0.59$, $g_2 = 0.02$ (compare Fig. 13).

A. Comparison of methods for rigorous integration

Let us first consider the Chua's circuit with the cubic nonlinearity. This is an example of a smooth dynamical system, and hence general integration methods can be used to compute trajectories. Let $\mathbf{x} = [-2.301, -2.3] \times [-0.141, -0.14] \times [1.23, 1.231]$ be the set of initial conditions. Fig. 14 shows results of rigorous numerical integration using four methods described in Section III-A. One can see that the standard Taylor method is able to integrate

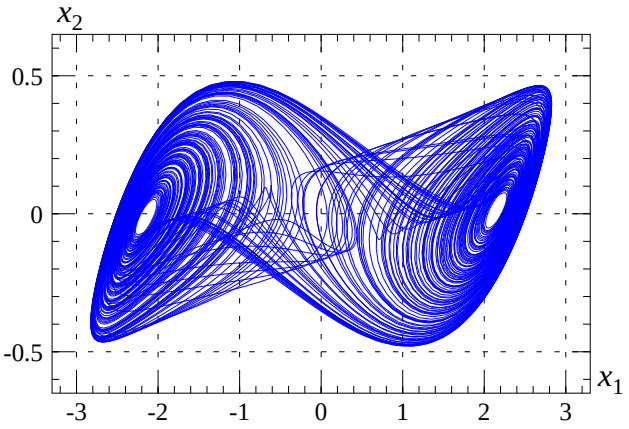

Fig. 13. The double-scroll attractor for the Chua's circuit with the cubic nonlinearity

the system for a very short time only ($t < 4$). Using the mean value form (Lohner IV) increases the integration time to $t \approx 23$. The QR version of the Lohner method which uses parallelograms to represent the solution set allows us to integrate up to $t \approx 85$. The internal enclosure (IE) version of the Lohner method which uses doubleton representation outperforms other methods and carries out integration for $t < 175$. From this example one can see that the Lohner method significantly reduces the wrapping effect. Without using this method the enclosures obtained are overly pessimistic and very often useless.

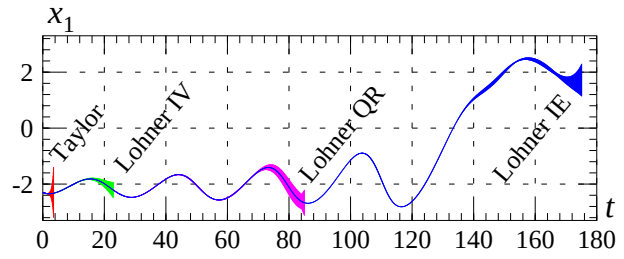

Fig. 14. Rigorous computation of a trajectory for the Chua's circuit with the cubic nonlinearity

Let us now consider the PWL nonlinearity. The state space $\mathbb{R}^3$ can be divided into three regions $U_1 = \{x \in \mathbb{R}^3 \colon x_1 < -1\}$, $U_2 = \{x \colon |x_1| < 1\}$ and $U_3 = \{x \colon x_1 > 1\}$ separated by planes $\Sigma_1 = \{x \colon x_1 = -1\}$ and $\Sigma_2 = \{x \colon x_1 = 1\}$. In the region R_i, the system is linear, the state equation can be written as: $\dot{x} = A_i(x - p_i)$, and the solution has the form $\varphi(t, x) = \exp(A_i t)(x - p_i) + p_i$.

In Section III-A, two methods to compute trajectories of PWL systems were described. The first method uses planes Σ_i as transversal sections. Each time a trajectory hits Σ_i, the intersection is found and is used as a set of initial conditions for further computations. The second method treats the PWL system as a perturbed linear system and obtains enclosures for solutions of the nonlinear system using the theory of differential inclusions. As an example let us compute a trajectory for the initial point $x = (1, -0.2212, -6.8978)$. The diameter d of the enclosure obtained using the two method versus the integration time is shown in Fig 15. One can see that for the first method there are jumps in the size of the enclosure. These jumps correspond to intersections with the planes Σ_i. The second method handles intersections via integration of perturbed dynamical systems and as one can see this approach significantly reduces the overestimation.

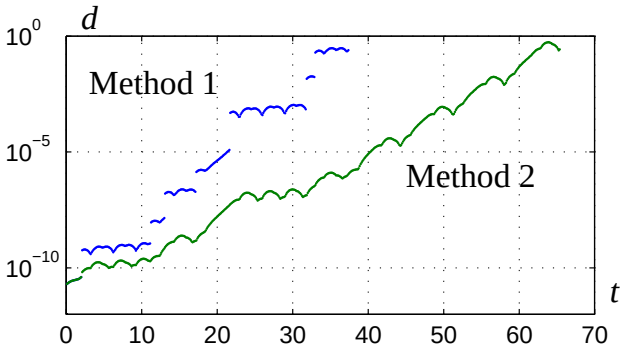

Fig. 15. Comparison of integration methods for PWL systems, the diameter d of the enclosure as a function of the integration time

B. Analysis of the spiral attractor

In this section, the dynamics of the spiral attractor is studied. A computer generated trajectory of the Poincaré map P defined by the plane Σ_2 is shown in Fig. 16(a).

In the first step of the rigorous analysis, a trapping region containing the numerically observed attractor is found. Two polygons enclosing a long trajectory are constructed and modified by hand to satisfy the condition for the trapping region. The fact that the selected set is a trapping region is proved rigorously using techniques presented in the previous sections.

In the second step, the graph representation of the dynamics of the system in the trapping region is found. The trapping region is covered by 6067 ε–boxes, with $\varepsilon = (0.001, 0.0025)$. Using the algorithm presented in Section III-B an accurate enclosure of the invariant part is found. After four subdivisions, we obtain the covering of the invariant part composed of 24482 ε–boxes of size $(0.001, 0.0025)/2^4$. There are 139553 non-forbidden transitions between boxes.

Once the graph is generated, we can find bounds for the return time for all points belonging to the attractor. Using the information on admissible connections between boxes and bounds for the return time for individual boxes one can obtain enclosures for the return time $\mathbf{T}_n$ of P^n, for example $\mathbf{T}_1 \subset [1.1986, 4.3658]$ and $\mathbf{T}_{1000} \subset [3270.4, 3314.1]$. It follows that the average return time for every trajectory belonging to the attractor belongs to the interval $[3.2704, 3.3141]$ and that the period of an orbit having p intersections with Σ_2 belongs to the interval $[3.2704 \cdot p, 3.3141 \cdot p]$.

The method described in Section III-F is applied to find all low-period cycles of P. All periodic orbits with no more than $p = 16$ intersection with Σ_2 are found. The

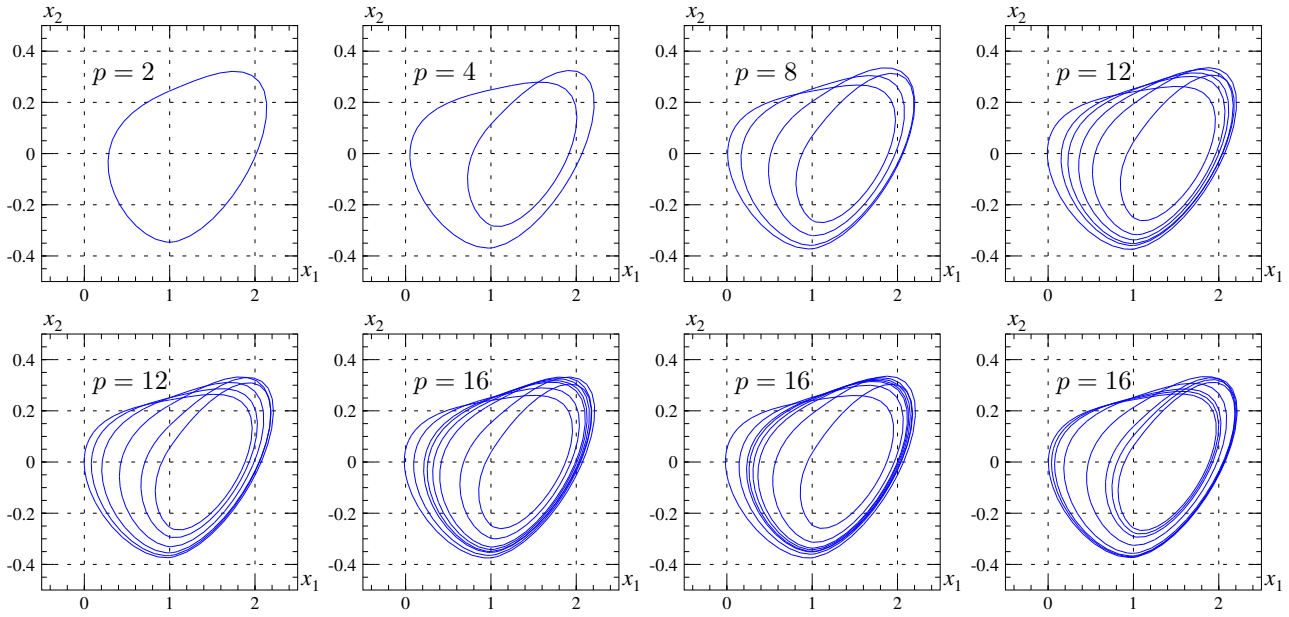

Fig. 17. Eight shortest periodic orbits embedded in the Chua's circuit spiral attractor, p is the number of intersections with Σ_2

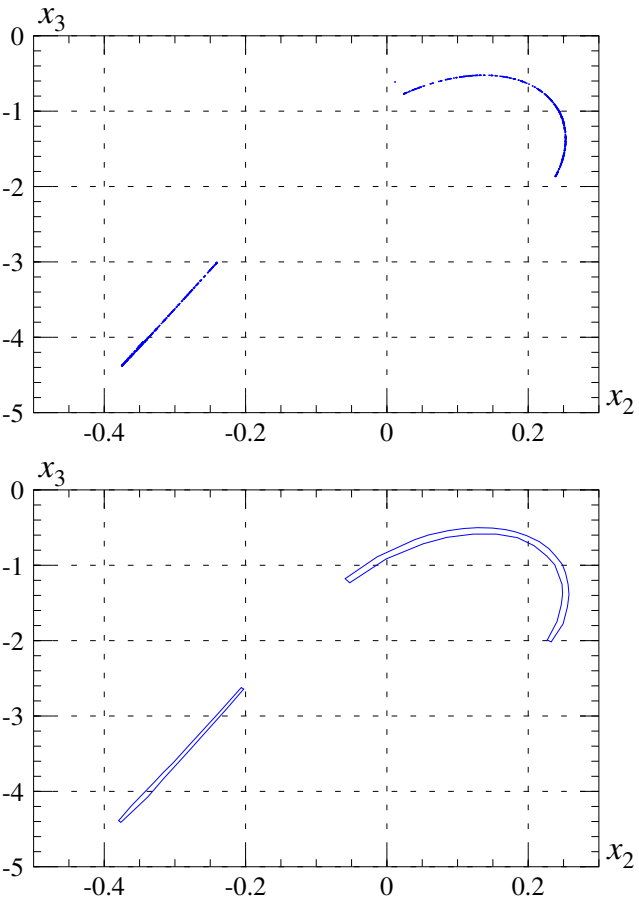

Fig. 16. Chua's circuit, the spiral attractor, (a) computer generated trajectory of the Poincaré map P, (b) a trapping region composed of two polygons

orbits are plotted in Fig. 17. There is one period-2 orbit, one period–4 orbit, one period–8 orbit, two period–12 orbits, and three period-16 orbits. One should note that there are no periodic orbits with periods 6, 10 and 14. It was shown that all of the orbits found are unstable. Since the average return time belongs to the interval $[3.2704, 3.3141]$ it follows that all periodic orbits with the flow time shorter than 58 have been found ($3.2704 \cdot 18 > 58$).

C. Analysis of the double-scroll attractor

There are a number of results concerning chaotic behaviour for the Chua's circuit double-scroll attractor. The geometric structure of the attractor was described in [45]. The existence of chaos in the sense of Shilnikov was shown in [35]. It was proved that for some parameter value belonging to a certain small interval there exist a homoclinic orbit. From the existence of this homoclinic orbit, it follows the existence of a symbolic dynamics on some sets located close to the homoclinic orbit.

In [46], it was shown that the double scroll attractor is chaotic in the topological sense. Let P be the Poincaré map defined by the plane $\Sigma_2 = \{x\colon x_1 = 1\}$. A trajectory of P is shown in Fig. 18. Intersections with Σ_2 with different directions are plotted using different colors. Fig. 18 also shows two quadrangles supporting nontrivial symbolic dynamics. It can be shown that the image of S_1 covers both S_1 and S_2, and that the image of S_2 covers S_1. It follows that the golden mean subshift is embedded in P^2. Hence, the topological entropy of P is positive and the system is chaotic in the topological sense.

The trapping region for the double scroll attractor was constructed in [18]. In order to carry out the proof, a combination of methods for integration of trajectories tangent to the C^0-hyperplanes and methods for integration of trajectories passing arbitrarily close to an unstable

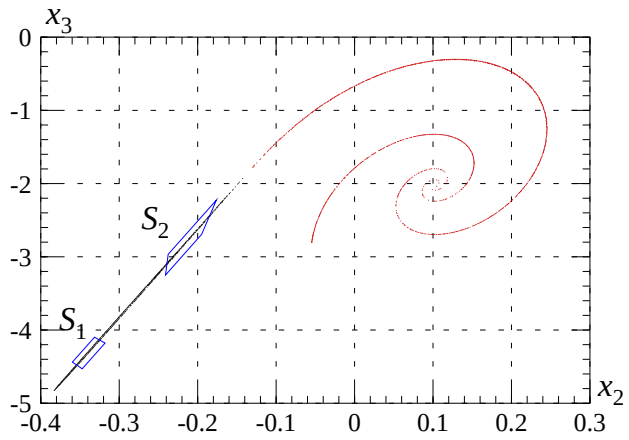

Fig. 18. Chua's circuit, the double-scroll attractor, computer generated trajectory of the Poincaré map P and quadrangles S_1, S_2 supporting nontrivial symbolic dynamics

equilibrium was used. The existence of a trapping region makes it possible to study rigorously the global dynamics of the double scroll attractor.

VI. Conclusions

In this work, several interval analysis based methods for studying the dynamics of nonlinear systems have been described. These methods make it possible to prove with the aid of a computer mathematically precise statements concerning discrete and continuous dynamical systems. Such statements may involve the existence of symbolic dynamics, the existence and stability of periodic orbits, the existence of trapping regions for the system, graph representation of the dynamics, and so on. Examples of validated computations concerning the Hénon map and the Chua's circuit are presented.

This work shows that many interval analysis tools are available to study dynamical systems and formulate statements and prove theorems about such systems in a precise mathematical sense. Several interval arithmetic based software libraries are easily available. They are capable of computing accurate enclosures of trajectories, proving existence of periodic orbits, and performing other validated computations, with very little input which has to be provided by the user. Usually, it is sufficient to provide the equation (a map or a vector field) defining the dynamical system.

In spite of these developments, there are still many open problems in the area of rigorous numerical analysis of nonlinear systems. One of the most challenging problems is long term integration of dynamical systems. In this context, a promising approach in offered by Taylor models. Unfortunately, there are very few software packages implementing this idea and access to them is limited.

Another important problem is the development of methods making it possible to prove the existence of chaotic attractors. Such a proof was carried out successfully for the Lorenz system, but for other systems, like the Chua's circuit, the problem of existence of a chaotic attractor is still open.

Another great challenge is rigorous study of higher dimensional systems and dynamical systems of infinite dimensions, for example partial differential equations and dynamical systems with delays. A necessary step to analyse such systems is to reduce the dimensionality to a finite number and at the same time ensure that the results obtained are valid for the original infinite dimensional system.

Zbigniew Galias (M'91) received the M.S. degree in electronics from the AGH University of Science and Technology, Krakow, Poland in 1990, the M.A. degree in mathematics from the Jagiellonian University, Krakow in 1992, and the Ph.D. degree and the senior doctorate degree in electrical engineering from the AGH University of Science and Technology, Krakow in 1996 and 2004, respectively.

He became an Assistant Professor at the AGH University of Science and Technology, Krakow in 1992, and was promoted to Associate Professor in 1996. Since 2005, he is a Professor of Electrical Engineering at the AGH University of Science and Technology, Krakow. His research interests include simulation, analysis, and design of nonlinear systems, chaos, interval arithmetic and computer assisted proofs. He has published more than 100 research papers. He has held visiting research positions at the Technical University Munich, the University of California at Berkeley, the University of California at San Diego, and the RMIT University, Melbourne.

Dr. Galias was awarded the Polish Research Foundation scholarship in 1994, received the Polish Prime Minister award for the Ph.D. theses in 1997, and was awarded the Fulbright fellowship in 1999.

REFERENCES

[1] A. Katok and B. Hasselblatt, *Introduction to the Modern Theory of Dynamical Systems*. Cambridge University Press, 1997.

[2] Z. Galias and W. Tucker, "Combination of exhaustive search and continuation method for the study of sinks in the Hénon map," in *Proc. IEEE Int. Symposium on Circuits and Systems, ISCAS'13*, Beijing, May 2013, pp. 2571–2574.

[3] R. Moore, *Methods and applications of interval analysis*. Philadelphia: SIAM, 1979.

[4] G. Alefeld and J. Herzberger, *Introduction to interval computations*. New York: Academic Press, 1983.

[5] R. Lohner, "Enclosing the solutions of ordinary initial and boundary value problems," in *Computerarithmetic, Scientific Computation and Programming Languages*. Stuttgart: Teubner, 1987, pp. 225–286.

[6] L. Figueiredo and J. Stolfi, "Adaptive enumeration of implicit surfaces with affine arithmetic," *Computer Graphic Forum*, vol. 15, pp. 287–296, 1996.

[7] W. Kühn, "Rigorously computed orbits of dynamical systems without the wrapping effect," *Computing*, vol. 61, no. 1, pp. 47–67, 1998.

[8] M. Berz and K. Makino, "Verified integration of ODEs and flows using differential algebraic methods on high order Taylor models," *Reliable Computing*, vol. 4, pp. 361–369, 1998.

[9] L. Rall, *Automatic Differentiation: Techniques and Applications*, ser. Lecture Notes in Computer Science. Berlin: Springer Verlag, 1981, vol. 120.

[10] O. Knüppel, "PROFIL/BIAS—a fast interval library," *Computing*, vol. 53, no. 3–4, pp. 277–287, 1994, source code available at http://www.ti3.tuhh.de/keil/profil/index_e.html.

[11] C. Bendtsen and O. Stauning, "FADBAD, a flexible C++ package for automatic differentiation," Technical University of Denmark, IMM, Department of Mathematical Modelling, Tech. Rep. IMM–REP–1996–17, 1996, source code available at http://www.fadbad.com/fadbad.html.

[12] S. Rump, "INTLAB — INTerval LABoratory," in *Developments in Reliable Computing*, T. Csendes, Ed. Dodrecht: Kluver Axademic Publishers, 1999, pp. 77–104.

[13] "CAPD library," source code available at http://capd.ii.uj.edu.pl/.

[14] D. Wilczak and P. Zgliczynski, "C^r-Lohner algorithm," *Scheda informaticae*, vol. 20, pp. 9–46, 2011.

[15] P. Zgliczyński and T. Kapela, "A Lohner-type algorithm for control systems and ordinary differential inclusions," *Discrete and Continuous Dynamical Systems B*, vol. 11, pp. 365–385, 2009.

[16] Z. Galias, "Rigorous study of the Chua's circuit spiral attractor," *IEEE Trans. Circ. Syst. I*, vol. 59, no. 10, pp. 2374–2382, 2012.

[17] W. Tucker, "The Lorenz attractor exists," *C. R. Acad. Sci. Paris*, vol. 328, pp. 1197–1202, 1999.

[18] Z. Galias, "Trapping region for the double scroll attractor," in *Proc. IEEE Int. Symposium on Circuits and Systems, ISCAS'12*, Seoul, May 2012, pp. 401–404.

[19] C. Hsu, "Global analysis by cell mapping," *Int. J. Bifurcation and Chaos*, vol. 4, no. 2, pp. 727–771, 1992.

[20] R. Donner, M. Small, J. Donges, N. Marwan, Y. Zou, R. Xiang, and J. Kurths, "Recurrence-based time series analysis by means of complex network methods," *Int. J. Bifurcation and Chaos*, vol. 21, no. 4, pp. 1019–1046, 2011.

[21] M. Dellnitz, A. Hohmann, O. Junge, and M. Rumpf, "Exploring invariant sets and invariant measures," *Chaos: an Interdisciplinary Journal of Nonlinear Science*, vol. 7, no. 2, pp. 221–228, 1997.

[22] A. Szymczak, "A combinatorial procedure for finding isolating neighborhoods and index pairs," *Proc. Royal Society of Edinburgh*, vol. 127A, pp. 1075–1088, 1997.

[23] Z. Galias, "Rigorous investigations of Ikeda map by means of interval arithmetic," *Nonlinearity*, vol. 15, pp. 1759–1779, 2002.

[24] D. Auerbach, P. Cvitanović, J. Eckmann, G. Gunaratne, and I. Procaccia, "Exploring chaotic motion through periodic orbits," *Phys. Rev. Lett.*, vol. 58, no. 23, pp. 2387–2389, 1987.

[25] P. Cvitanović, "Invariant measurement of strange sets in terms of cycles," *Phys. Rev. Lett.*, vol. 61, no. 24, pp. 2729–2732, 1988.

[26] A. Neumaier, *Interval methods for systems of equations*. Cambridge University Press, 1990.

[27] G. Alefeld, "Inclusion Methods for Systems of Nonlinear Equations – The Interval Newton Method and Modifications," in *Topics in Validated Computations, Proceedings of the IMACS-GAMM International Workshop on Validated Computation*, J. Herzberger, Ed. Elsevier, 1994, pp. 7–26.

[28] R. Kearfott and M. Novoa, "Algorithm 681: INTBIS, a portable interval Newton/bisection package," *ACM Trans. Math. Software*, vol. 16, no. 2, pp. 152–157, 1990.

[29] M. Dellnitz, O. Schütze, and S. Sertl, "Finding zeros by multilevel subdivision techniques," *IMA Journal of Numerical Analysis*, vol. 22, no. 2, pp. 167–185, 2002.

[30] Z. Galias, "Counting low-period cycles for flows," *Int. J. Bifurcation and Chaos*, vol. 16, no. 10, pp. 2873–2886, 2006.

[31] C. Robinson, *Dynamical Systems: Stability, Symbolic Dynamics, and Chaos*. USA: CRC Press, 1995.

[32] B. Hassard, S. Hastings, W. Troy, and J. Zhangk, "A computer proof that the Lorenz equations have "chaotic" solutions," *Appl. Math. Letters*, vol. 7, pp. 79–83, 1994.

[33] P. Zgliczyński, "Fixed point index for iterations of maps, topological horseshoe and chaos," *Topological Methods in Nonlinear Analysis*, vol. 8, no. 1, pp. 169–177, 1996.

[34] P. Zgliczyński, "Computer assisted proof of chaos in the Rössler equations and the Hénon map," *Nonlinearity*, vol. 10, no. 1, pp. 243–252, 1997.

[35] T. Matsumoto, L. Chua, and K. Ayaki, "Reality of chaos in the double scroll circuit: a computer-assisted proof," *IEEE Trans. Circ. Syst.*, vol. CAS–35, no. 7, pp. 909–925, July 1988.

[36] H. Kokobu, D. Wilczak, and P. Zgliczynski, "Rigorous verification of cocoon bifurcations in the Michelson system," *Nonlinearity*, vol. 20, pp. 2147–2174, 2007.

[37] M. Hénon, "A two dimensional map with a strange attractor," *Commun. Math. Phys.*, vol. 50, pp. 69–77, 1976.

[38] Z. Galias, "Interval methods for rigorous investigations of periodic orbits," *Int. J. Bifurcation and Chaos*, vol. 11, no. 9, pp. 2427–2450, 2001.

[39] P. Grassberger and H. Kantz, "Generating partitions for the dissipative Hénon map," *Physica*, vol. 17D, pp. 235–238, 1985.

[40] P. Grassberger, H. Kantz, and U. Moenig, "On the symbolic dynamics of the Hénon map," *J. Phys. A*, vol. 22, pp. 5217–5230, 1989.

[41] S. Newhouse and T. Pignataro, "On the estimation of topological entropy," *Journal of Statistical Physics*, vol. 72, pp. 1331–1351, 1993.

[42] Z. Galias and P. Zgliczyński, "Abundance of homoclinic and heteroclinic orbits and rigorous bounds for the topological entropy for the Hénon map," *Nonlinearity*, vol. 14, pp. 909–932, 2001.

[43] S. Newhouse, M. Berz, J. Grote, and K. Makino, "On the estimation of topological entropy on surfaces," *Contemporary Mathematics*, vol. 469, pp. 243–270, 2008.

[44] L. Chua and G. Lin, "Canonical realisation of Chua's circuit family," *IEEE Trans. Circ. Syst.*, vol. CAS–37, no. 7, pp. 885–902, July 1990.

[45] T. Matsumoto, L. Chua, and M. Komuro, "The double scroll," *IEEE Trans. Circ. Syst.*, vol. CAS–32, no. 8, pp. 798–817, Aug. 1985.

[46] Z. Galias, "Positive topological entropy of Chua's circuit: A computer assisted proof," *Int. J. Bifurcation and Chaos*, vol. 7, no. 2, pp. 331–349, 1997.