Applied Mathematics and Computation 329 (2018) 408–419

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Detecting unreliable computer simulations of recursive functions with interval extensions

Erivelton G. Nepomuceno[a,*], Samir A.M. Martins[a], Bruno C. Silva[a], Gleison F.V. Amaral[a], Matjaž Perc[b,c,d]

[a] Control and Modelling Group (GCOM), Department of Electrical Engineering, Federal University of São João del-Rei, São João del-Rei, MG 36307-352, Brazil
[b] Faculty of Natural Sciences and Mathematics, University of Maribor, Koroška cesta 160, Maribor SI-2000, Slovenia
[c] CAMTP – Center for Applied Mathematics and Theoretical Physics, University of Maribor, Mladinska 3, Maribor SI-2000, Slovenia
[d] Complexity Science Hub, Josefstädterstraße 39, Vienna A-1080, Austria

ARTICLE INFO

Keywords:
Nonlinear dynamics
Chaos
Numerical simulation
Lower bound error

ABSTRACT

This paper presents a procedure to detect unreliable computer simulations of recursive functions. The proposed method calculates a lower bound error which is derived from two different pseudo-orbits based on interval extensions. The interval extensions are generated by taking into account the associative property of multiplication, which keeps the same error bound. We have tested our approach on the logistic map using many different programming languages and simulation packages, including Matlab, Scilab, Octave, Fortran and C. In all cases, the number of iterates is significantly lower than that considered reliable in the existing literature. We have also used the lower bound error on the logistic map and on the polynomial NARMAX for the Rössler equations to estimate the largest Lyapunov exponent, which determines the critical simulation time that guarantees the reliability of the simulation.

© 2018 Elsevier Inc. All rights reserved.

1. Introduction

Numerical computing and dynamical systems go hand in hand and this has been a long-term relation. In the 60s, it had already been observed the use of computers dedicated to efforts in understanding climate phenomena [22]. In fact, many numerical problems in dynamical systems, such as finding sinks in the Hénon Map or iterating the Lorenz attractor, have motivated research on arithmetic algorithms for extended precision [20]. As stated in [25], computational techniques are applied to different topics in nonlinear dynamics, such as synchronisation, bifurcation and chaos, complex networks, conservative systems and nonlinear partial differential equations.

In fact, many works have been published delivering an idea of reliability in their numerical solutions of nonlinear dynamical systems. This is not the precise idea that Lozi states in [24]. In fact, he asks if "In the simple case of a dynamic discrete system (of Hénon map) there are doubts as to the nature of the computational results: long unstable pseudo-orbits or strange attractors?" [24]. Other earlier works, such as [6,10,37], have already raised some suspicions on the conclusions of nonlinear dynamics upon numerical simulations. Recently, Galias [11] expresses the importance of developing methods

* Corresponding author.
E-mail addresses: nepomuceno@ufsj.edu.br (E.G. Nepomuceno), martins@ufsj.edu.br (S.A.M. Martins), amaral@ufsj.edu.br (G.F.V. Amaral), matjaz.perc@uni-mb.si (M. Perc).

https://doi.org/10.1016/j.amc.2018.02.020
0096-3003/© 2018 Elsevier Inc. All rights reserved.

to prove the existence of chaotic attractors and warns its audience to the dangers of rounding errors for simulations and analysis of nonlinear circuits. Similar results can be found in [33], but using a method based on interval extension.

Different from the Lorenz system, the proof for the chaotic attractor of the Chua's circuit needs to be further investigated [11]. The case of Chua's circuit is an outstanding example of nonlinear dynamical system built upon on inequalities, which has been seen as important source of inconsistent results according to [3]. A "good agreement" between simulation and experimental data cannot be seen as the final answer, which has been considered as paradoxical in [6]. In such direction, it has been reported that a simulation of the Chua's circuit using the same set of parameters and initial conditions presents chaotic and periodical oscillation depending of the numerical method applied [41]. This kind of chaos suppression according numerical methods is not a new issue and it has been already studied in works such as [7]. Another fact that points out for the importance of the reliability [14,34] and reproducibility [40,43] of numerical simulation is the fact that there are theoretical models with no possible real experiment. Take an example of logistic map [26], where x is considered the population density and a continuous variable in the interval of 0 to 1. It means that a population has a infinite number of individuals, which is impossible to make an equivalent experiment.

The term "Computational Chaos" has been defined during investigations on chaotic behaviour of difference equations used to approximate a continuous system represented by a set of differential equations as the step-size is increased [30]. Further results on the same issue can be found in [11,21,23,45,46]. For instance, in [21], the author has introduced the concept of critical predictable time to provide a more precise description of computed chaotic solutions of nonlinear differential equations.

According to [17], one of the first attempts to examine the relation between numerical experiments and the true dynamics of a system is found in [15]. By means of the Cray X-MP, a computer costing multi-million dollars at that time, they have reported the shadowing property as valid for a considerable number of iterates when simulating the logistic map for the parameter $r = 3.8$ and the initial condition $x_0 = 0.4$. When this case is considered, according to [15], a pseudo-orbit of the logistic map is shadowed by a true orbit within a distance of 10^{-8} for 10^7 iterates. Since the theorem is proved for specific conditions, "it can be raised an issue if the computer test constitutes a sufficient condition, and therefore the theorem is proved for all cases, or whether it is a necessary condition, the result may not be valid for all cases", as states [32].

Many researchers have applied the result in [15] on studies concerning the dynamical systems theory since then. More than 150 citations of [15] have been analysed through the Scopus online platform and it was observed that, in some cases, the shadowing theorem is seen as a property possible to be generalised for non-hyperbolic systems. The paper of [15] is mentioned as the one that proves the shadowing property for systems with non-hyperbolic behaviour by [39]. In the same line, [15] is considered as "first proof of the existence of a shadow for a two-dimensional non-hyperbolic system over a non-trivial length of time" by [17]. In addition, remarks that the shadowing property had been shown valid for some initial condition and parameter values of the logistic map, without mentioning the cases for which the property fails are presented in [4].

Thus, beyond using the computer to study nonlinear systems, it also becomes important verifying the reliability of its numerical results. Some recent works have explored the issue of checking computer results [11,31,32]. An interesting approach to analyse the forward error of iterative numerical algorithm is proposed in [16]. The authors state that "many iterative numerical algorithms can be considered as dynamical systems, and therefore can be studied using control systems theory", and although the proposed forward error analysis schemes may also be applied to algorithms that can be represented by a nonlinear dynamical system, it will be necessary to show that the dynamical system used to represent the algorithm is Lyapunov stable [16]. One of the key steps towards a control-theory approach to analyse error may require a way to proper measure this error. In this sense, based on the preliminary work about convergence of recursive functions on a computer [31], a method has been proposed to calculate a lower bound error for free-run simulation of polynomial NARMAX [2] by Nepomuceno and Martins [32]. In that work, a simulation is performed by means of two interval extensions, derived from mathematical properties, such as commutative, distributive or associative [29]. These two interval extensions produce two different pseudo-orbits. Applying interval analysis, the authors show a way to calculate an inferior limit, that is, a lower bound, for the error. This method can be extended to other types of recursive functions, as done in [33]. More recently, the lower bound error has been used to develop a fast and robust method to calculate the positive largest Lyapunov exponent by Mendes and Nepomuceno [27,33]. It has been also used to estimate a critical simulation time, similar to that presented in [21], but calculated by means of interval extensions.

In [32], three identified models from literature have been investigated and it has been showed that it is not possible to undertake an arbitrary number of iterations for some cases of simulation of polynomial NARMAX without losing all significant digits of the simulation. In fact, there are cases that less than one hundred of iterations, all digits lose their significance. This is a serious matter, as for many types of validation techniques, such as the bifurcation diagrams, hundreds of iterations have been considered as transient. For instance, in [35, p. 40], the author described a procedure to build a bifurcation diagram of the logistic map considering the first 500 iterations as transient. This suggested procedure does not make any requirement of computer precision, on the contrary, what we have seen in the literature, most of works have been using only double precision. For instance, in [1] the authors use 1000 points in their algorithm to detect nonlinear dynamics without any description of software or precision. Regarding the studies on Hénon Map, it is noticed a need of an extremely large number of points to observe a sink starting from random initial conditions by [12]. The author in [12] concludes that the most numerical studies do not display anything but transient behaviour. This emphasises the need to a very careful analysis of reliability of a numerical simulation.

In this paper we propose a method to detect unreliable computer simulation of recursive functions based on interval extensions. In particular, we are interested in verifying whether the results regarding the confidence of numerical simulation of logistic map, as enunciated by [15], is also valid for today's programming languages. In some sense, our main motivation comes from [9], in which a discussion on the using of computer experimentation in mathematics proof is presented. Our method is based in the lower bound error proposed in [32]. First, we check in detail the theorem proposed by [15] using the lower bound error and also symbolic computation. Here we applied associative property of multiplication to produce two interval extensions in order to keep the numerical simulations with the same error interval. After that, we verify the results for all combination of parameters and initial conditions indicated as valid in [15] for three interpreted languages: Matlab, Scilab, Octave and in two compiled languages: Fortran and C. All tests indicate that the theorem is not valid for these programming languages in modern computers. We also present a simple technique to estimate a critical simulation time for systems based on the largest Lyapunov exponent and interval extensions. This is a supplementary way to verify the reliability of the simulation. In a similar way showed by Palmore and Hering [37], our results show that floating-point errors can propagate in a simulation of chaotic systems, and within less than 100 steps, destroy all accuracy of a result.

The outline of the paper is the following: in Section 2 we recall some preliminary concepts on the polynomial NARMAX, IEEE standards on floating point and interval arithmetic [18,19]. Still in Section 2, we present the definitions of orbits and pseudo-orbits and the lower bound error. Then, in Section 3, we present our method to to detect unreliable computer simulation of recursive functions. Section 4 is devoted to present the results,then the final remarks are given in Section 5.

A preliminary version of our work was presented in [44].

2. Preliminary concepts

Essential concepts for understanding following sections appear here. Some of the definitions and the lower bound theorem presented in this section are given by [32].

2.1. Floating point and interval arithmetic

Given the combination of applying rounding methods and the how the computer stores numbers, mathematical properties such as commutative, associative and distributive, cannot be guaranteed for floating-point operations [18,36]. It can be said that the floating-point simulations represent an abstraction of the reality and results from reality may not be easily transferred to the computer [6].

A key feature of the IEEE 754–2008 standard [18] is that it requires correctly rounded arithmetic operations. Let x and y be floating point numbers, let $+, -, \times, \div$ denote the standard arithmetic operations, and let $\oplus$, $\ominus$, $\otimes$, $\oslash$, the equivalent operations as they are actually implemented on computers. The rule is as follows: if x and y are floating point numbers, then [36]

$$x \oplus y = \text{round}(x+y) = (x+y)(1+\varepsilon) \tag{1}$$

$$x \ominus y = \text{round}(x-y) = (x-y)(1+\varepsilon) \tag{2}$$

$$x \otimes y = \text{round}(x \times y) = (x \times y)(1+\varepsilon) \tag{3}$$

$$x \oslash y = \text{round}(x \div y) = (x \div y)(1+\varepsilon) \tag{4}$$

where

$$|\varepsilon| \leq 2^{-p} \tag{5}$$

for double precision and rounding mode in effect being round to nearest we have the machine epsilon ($p = 53$). As ε is taken as absolute value in Eq. (5), we may write that the result of an arithmetic operation relies in the interval given by the multiplication of $(1-\varepsilon)$ and $(1+\varepsilon)$. Taking the addition as example, we have

$$(x+y)(1-\varepsilon) \leq x \oplus y \leq (x+y)(1+\varepsilon) \tag{6}$$

which is equivalent to say

$$x \oplus y \in [(x+y)(1-\varepsilon), (x+y)(1+\varepsilon)] \tag{7}$$

and we may proceed in the same for subtraction, multiplication and division. Hence we may rewrite Eqs. (1) to (4) using a description of interval such as Eqs. (8) to (11).

$$x \oplus y \in Z = [\underline{Z}, \overline{Z}] = [(x+y)(1-\varepsilon), (x+y)(1+\varepsilon)] \tag{8}$$

$$x \ominus y \in Z = [\underline{Z}, \overline{Z}] = [(x-y)(1-\varepsilon), (x-y)(1+\varepsilon)] \tag{9}$$

$$x \otimes y \in Z = [\underline{Z}, \overline{Z}] = [(x \times y)(1-\varepsilon), (x \times y)(1+\varepsilon)] \tag{10}$$

$$x \oslash y \in Z = [\underline{Z}, \overline{Z}] = [(x \div y)(1-\varepsilon), (x \div y)(1+\varepsilon)] \tag{11}$$

2.2. Interval extension

It is known that we do not have certain properties of real arithmetic, such as associative property of addition in floating-point arithmetic [18,36,38]. It means that two mathematical equivalent sequence of arithmetic operations can lead to two different results. However, we may define a set of operations with the same resulting interval.

An interval is set of real numbers such that any number that lies between two numbers in the set is also included in the set. As for notation, an interval X is denoted $[\underline{X}, \overline{X}]$, i.e. $X = \{x : \underline{X} \leq x \leq \overline{X}\}$. In a degenerated interval, we have $\underline{X} = \overline{X}$ and such an interval amounts to a real number $x = \underline{X} = \overline{X}$. Let the following definition [28]:

Definition 2.1. (*Interval extension*) - Let f be a function of real variable x. A function F is an *interval extension* of f if F agrees with f for degenerate interval arguments:

$$F([x, x]) = f(x).$$

An interval extension of f is thus an interval valued function which has real values when the arguments all are real (degenerate intervals) and coincides with f in this case. There are interval extensions that are equivalent from the point of view of intervals. This leads us to the following definition:

Definition 2.2. (*Equivalent interval extension*) - The interval extensions G and H are equivalent if

$$G(X) = H(X) \quad \text{for all interval arguments.}$$

Example 2.3. Let us consider the following extension intervals:

$$G(X) = rX(1-X) \tag{12}$$

$$H(X) = r(X(1-X)) \tag{13}$$

$$L(X) = rX - rX^2. \tag{14}$$

If $r = 3.9$ and $X = [0.2, 0.25]$, then we have

$$\begin{aligned} G([0.2, 0.25]) &= 3[0.2, 0.25](1 - [0.2, 0.25]) \\ &\quad [0.58500, 0.78000], \\ H([0.2, 0.25]) &= 3([0.2, 0.25](1 - [0.2, 0.25])) \\ &\quad [0.58500, 0.78000], \\ L([0.2, 0.25]) &= 3[0.2, 0.25] - 3([0.2, 0.25]^2) \\ &\quad [0.53625, 0.81900]. \end{aligned}$$

In Example 2.3 only $G(X)$ and $H(X)$ are equivalent interval extensions. It is important to stress that from the point of view of mathematics based on real numbers, the different syntactic counterparts of $G(X)$ and $H(X)$ yield the same final result. However, in the interval arithmetic the final result is different, and similar process is undertaken when any arithmetic operations is being doing on computers.

Consider the following lemma:

Lemma 2.4. *Two interval extensions generated by Associative property of multiplication presents the same error interval due to round off using IEEE 754–2008 standard.*

Proof. Let these two generic interval extensions $a \times b \times c$ and $a \times (b \times c)$ derived from associative property of multiplication. Let the machine epsilon ε. We have

$$\begin{aligned} a \otimes b \otimes c &= a \otimes (b \otimes c) \\ ((a \times b)(1+\varepsilon)) \otimes c &= a \otimes ((b \times c)(1+\varepsilon)) \\ (((a \times b)(1+\varepsilon)) \times c)(1+\varepsilon) &= (a \times ((b \times c)(1+\varepsilon)))(1+\varepsilon) \\ (a \times b \times c)(1+\varepsilon)^2 &= (a \times b \times c)(1+\varepsilon)^2 \end{aligned}$$

It shows that both interval extensions present the same level of error, that is $(1+\varepsilon)^2$ and that completes the proof. □

2.3. Orbits and pseudo-orbits

Let $n \in \mathbb{N}$, a metric space $M \subset \mathbb{R}$, the relation

$$x_{n+1} = f(x_n), \tag{15}$$

where f: $M \to M$, is a recursive function or a map of a state space into itself and x_n denotes the state at the discrete time n. The sequence $\{x_n\}$ obtained by iterating Eq. (15) starting from an initial condition x_0 is called the orbit of x_0 [13].

Consider the following definitions for orbits and pseudo-orbits:

Definition 2.5. An orbit is a sequence of values of a map, represented by $\{x_n\} = [x_0, x_1, \ldots, x_n]$.

Definition 2.6. Let $i \in \mathbb{N}$ represents a pseudo-orbit, which is defined by an initial condition, an interval extension of f, some specific hardware, software and numerical precision standard. A pseudo-orbit is an approximation of an orbit and we represent as

$$\{\hat{x}_{i,n}\} = [\hat{x}_{i,0}, \hat{x}_{i,1}, \ldots, \hat{x}_{i,n},]$$

such that,

$$|x_n - \hat{x}_{i,n}| \leq \delta_{i,n}, \tag{16}$$

where $\delta_{i,n} \in \mathbb{R}$ is the error and $\delta_{i,n} \geq 0$.

Considering the definitions of orbit and pseudo-orbit, [15] enunciate a theorem for the logistic map [26],

$$x_{n+1} = f(x_n) = rx_n(1 - x_n), \tag{17}$$

$n \in \mathbb{N}$ and $r \in \mathbb{R}$.

2.4. The lower bound error

A pseudo-orbit defines an interval where the true orbit sits. Hence an interval associated with each value of a pseudo-orbit may be defined as

$$I_{i,n} = [\hat{x}_{i,n} - \delta_{i.n}, \hat{x}_{i,n} + \delta_{i.n}]. \tag{18}$$

From (17) and (18) it is clear that

$$x_n \in I_{i,n}, \text{ for all } i \in \mathbb{N}. \tag{19}$$

Theorem 2.7 (Lower bound error *[32]*)**.** *Let two pseudo-orbits $\{\hat{x}_{a,n}\}$ and $\{\hat{x}_{b,n}\}$ derived from two interval extensions. Let*

$$\delta_{\alpha,n} = \frac{|\hat{x}_{a,n} - \hat{x}_{b,n}|}{2}$$

be the lower-bound error of a map f(x), then $\delta_{a,n} \geq \delta_{\alpha,n}$ or $\delta_{b,n} \geq \delta_{\alpha,n}$.

Theorem 2.7 establishes that at least one of the two pseudo-orbits must have an error greater or equal to the lower-bound error. This has a practical meaning. If this lower-bound error is greater than the required precision, the simulation should not be carried on without further analysis. It should kept in mind that simulation of a continuous nonlinear system using a discretization scheme presents multiple pseudo-orbits, to which Theorem 2.7 can be directly applied.

2.5. Theorem proposed by Hammel and colleagues

The following theorem using the same notation as the original work is presented by Hammel and colleagues [15]:

Theorem 2.8 (Hammel et al.)**.** *There is a true orbit $\{x_n\}$ of the logistic map for which p_n is within a distance of 10^{-8} of x_n for 10^7 iterates (i.e., for each $n = 0, 1, 2, \ldots, 10^7$).*

To make an equivalence to [15], the pseudo-orbit in this work is called $\{\hat{x}_{i,n}\}$, instead of p_n. So in other words, they state that a pseudo-orbit $\{\hat{x}_{i,n}\}$ is shadowed by a true orbit $\{x_n\}$ within shadowing distance $\delta_x = 10^{-8}$, with $x_0 = 0.4$ and $r = 3.8$, for $n = 0, 1, \ldots, 10^7$ iterates. The proof has been undertaken by means of computer simulation as described in [15].

2.6. Critical simulation time

For chaotic systems, two interval extensions diverge and the critical simulation time when the orbits reach the maximum distance can be established as shown in the following lemma. A preliminary version of this lemma has been published in [33].

Lemma 2.9. *Let a chaotic system be described by function $f(x_n)$. Let $F(X_n)$ and $G(X_n)$ be interval extensions of $f(x_n)$. Then $\delta_{\alpha,n} \to D/2$ as $n \to T_s$, where*

$$T_s = \frac{\log_{10}(D/2) - P}{\lambda},$$

D is the diameter of the phase space, P is the precision of the machine and λ is the largest Lyapunov exponent.

Proof. Lemma 2.9 establishes that $F(X_n)$ and $G(X_n)$ will eventually produce different results, even from the same initial condition. For chaotic system, this means that the results diverge from each other. The maximum distance between these two pseudo-orbits is the diameter of the phase space and therefore the lower bound error is limited to the half of this value. The critical simulation time T_s may be calculated using the value of the largest Lyapunov exponent, λ and the precision P of the simulation. Thus,

$$\log_{10}(D/2) = \lambda T_s + P,$$

which is the maximum error that the lower bound error may reach and this completes the proof. □

This almost naive method has been shown quite interesting to proceed a set of tests. Its main feature is the possibility to present the evolution of error.

2.7. The polynomial NARMAX

A NARMAX (Nonlinear AutoRegressive Moving Average model with eXogenous input) model can be written as [5]:

$$y_{n+1} = \mathcal{F}^{\ell}[y_n, \ldots, y_{n-1-k_y}, u_n, \ldots, u_{n-1-k_u}, \quad e_n, \ldots, e_{n-1-k_e}], \tag{20}$$

where y_n, u_n and e_n are respectively the output, input and noise terms at the discrete time $n \in \mathbb{N}$. The parameters k_y, k_u and k_e are the maximum lag considered for output, input and noise. Terms of e_n are frequently included in the parameter estimation process, to avoid bias of an estimator. In this work $\mathcal{F}^{\ell}[\cdot]$ is assumed to be a polynomial with non-linearity degree $\ell \in \mathbb{Z}^+$. Nonlinear systems can be modelled using particular cases of the polynomial NARMAX, such as NAR, NARX and NARMA. In this paper, we name them all as polynomial NARMAX.

3. Methodology

In this section, we present the method proposed in this article to detect unreliable computer simulation.

First we further develop the Theorem 2.7 in order to use two interval extensions that present the same error interval according to IEEE 754–2008. Applying the Lemma 2.4, we can write the following Corollary:

Corollary 3.1. *(Lower bound error with equivalent interval extensions) - If two pseudo-orbits $\{\hat{x}_{a,n}\}$ and $\{\hat{x}_{b,n}\}$ are derived from two interval equivalent extensions by means of associative property of multiplication, then $\delta_{a,n} \geq \delta_{\alpha,n}$ and $\delta_{b,n} \geq \delta_{\alpha,n}$.*

Proof. This corollary is a simple inference of Theorem 2.7 and Lemma 2.4. We know by Theorem 2.7 that two pseudo-orbits present $\delta_{a,n} \geq \delta_{\alpha,n}$ or $\delta_{b,n} \geq \delta_{\alpha,n}$. But, by the Lemma 2.4, we show that associative property of multiplication keeps the same bounds of error, that is, $\delta_{a,n} = \delta_{b,n}$. Thus, $\delta_{a,n} = \delta_{b,n} \geq \delta_{\alpha,n}$, what is equivalent to say that $\delta_{a,n} \geq \delta_{\alpha,n}$ and $\delta_{b,n} \geq \delta_{\alpha,n}$. And that completes the proof. □

Example 3.2. Let us consider the logistic map $f(x) = rx(1-x)$. The equivalent interval extensions $G(x) = rx(1-x)$ and $H(x) = r(x(1-x))$ are arithmetic interval equivalent, since we just apply the associative property $abc = a(bc)$. Indeed the resulting interval for $G(x)$ is as follows:

$$\begin{aligned}
Z_G &= r \otimes x \otimes (1 \ominus x) \\
&= r \otimes x \otimes [(1-x)(1-\varepsilon), (1-x)(1+\varepsilon)] \\
&\in [rx(1-\varepsilon), rx(1+\varepsilon)] \otimes [(1-x)(1-\varepsilon), (1-x)(1+\varepsilon)] \\
&= [rx(1-x)(1-\varepsilon)^2(1-\varepsilon), rx(1-x)(1+\varepsilon)^2(1+\varepsilon)] \\
&= [rx(1-x)(1-\varepsilon)^3, rx(1-x)(1+\varepsilon)^3]
\end{aligned} \tag{21}$$

whereas for $H(x)$ we have

$$\begin{aligned}
Z_H &= r \otimes (x \otimes (1 \ominus x)) \\
&= r \otimes (x \otimes [(1-x)(1-\varepsilon), (1-x)(1+\varepsilon)] \\
&= r \otimes ([x(1-x)(1-\varepsilon)(1-\varepsilon), x(1-x)(1+\varepsilon)(1+\varepsilon)] \\
&= [rx(1-x)(1-\varepsilon)^3, rx(1-x)(1+\varepsilon)^3].
\end{aligned} \tag{22}$$

Thus

$$f(x) \in [rx(1-x)(1-\varepsilon)^3, rx(1-x)(1+\varepsilon)^3] = Z_G = Z_H \tag{23}$$

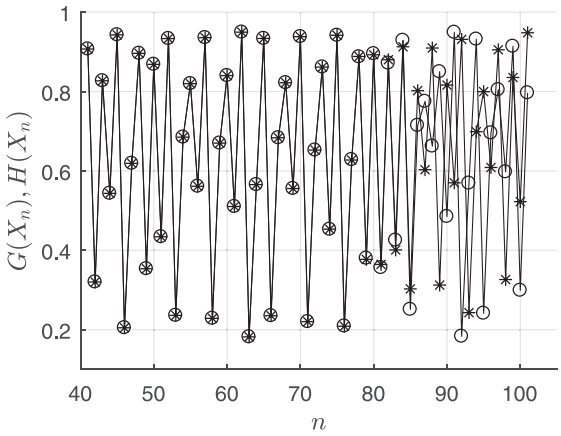

Fig. 1. Simulation of $G(X)$ and $H(X)$. Results of n iterates for $G(X_n)$ (–o–) and $H(X_n)$ (–*–), considering $X_0 = 0.4$ and $r = 3.8$.

As we can see, the intervals in (21) and (22) are the same. □

The importance of Lemma 2.4 is that, although the computational result for $G(x)$ and $H(x)$ would be probably different due to different sequence of arithmetic operations, they have the same error interval due to round off.

Consider the following interval extensions of the logistic map (Eq. 17) (See [47] to further discussions):

$$G(X) = rX(1 - X) \tag{24}$$

$$H(X) = r(X(1 - X)). \tag{25}$$

The computationally generated pseudo-orbits for 100 iterations of (24) and (25), $\{\hat{x}_{G,n}\}$ and $\{\hat{x}_{H,n}\}$, respectively, are considered for analysis, and, as already mentioned, for 0.4 as initial condition and $r = 3.8$. A third pseudo-orbit $\{\hat{x}_{P,n}\}$, from an interval extension $P(X)$ of (17), was obtained through the use of the variable-precision arithmetic (VPA) Matlab function.

At first, the computational tests were performed using the Matlab R2016a, on Windows 8 (64 bits) operating system. Then, in order to assure the consistence of the results, the simulations were performed using Octave 4.0.0, Scilab 5.5.0 and Matlab R2016a, considering the logistic map (Eq. (17)), for values of r highlighted in [15]. We also tested in Fortran 90 and C, with the compiler GCC 4.9.2. All the simulation were performed using Windows 8 (64 bits) and Ubuntu 14.04 (64 bits) operating systems. For the specific case of Fortran, the compiler has been setup in such way that 64 bits has been replaced by 32 bits in Windows.

Our main test consist in calculating the number of iterates in which the lower bound error reaches the precision of 10^{-8}. Thus we are interested in finding the n such that $\delta_{\alpha,n} > 10^{-8}$.

One important aspect of this method, it should be used to falsify nor to validate, that is, the fact that the lower bound error presents a desirable value does not mean that the simulation is reliable. On other hand, when the lower bound error presents a value above an expected value, we can say that simulation is not reliable.

Finally, we calculate the critical simulation time based on the largest Lyapunov Exponent and equivalent interval extensions.

4. Results

This section illustrates the two main results of this paper. First, the shadowing property is verified according to the same results presented in [15]. Second, the critical simulation time is calculated for the logistic map and for the polynomial NARMAX of the Rössler equations.

4.1. Unreliable computer simulation test

Fig. 1 shows the results for the simulation of interval extensions from Eqs. (24) and (25) after 40 iterates. In the $51th$ iterate it may be already observed a lower bound error $\delta_{\alpha,51} = 10^{-7.638}$, greater than 10^{-8} (Fig. 2). Thus, by Corollary 3.1, $\delta_{G,51} \geq 10^{-7.638}$ and $\delta_{H,51} \geq 10^{-7.638}$.

In Fig. 3, results for the simulation of $G(X)$ e $H(X)$ are represented, and also, results produced by the iteration process of the logistic map using 1000-digit precision through VPA, $P(X_n)$.

Let $\delta_{GP,n}$ represents $|\hat{x}_{G,n} - \hat{x}_{P,n}|$ and $\delta_{HP,n}$ represents $|\hat{x}_{H,n} - \hat{x}_{P,n}|$. The evolution of $\delta_{GP,n}$ and $\delta_{HP,n}$ is shown in Fig. 4.

As $\{\hat{x}_{P,n}\}$ is produced through symbolic computing of high precision, $\{\hat{x}_{P,n}\}$ is expected to be very close to a true orbit, at least for a number of iterates much lower than the number of iterates, which is the case. Then it is expected that $\delta_{P,n} \approx 0$, $\delta_{G,n} \approx \delta_{GP,n}$, and $\delta_{H,n} \approx \delta_{HP,n}$. Considering that $\delta_{P,43} \approx 0$, $\delta_{G,43} \approx 10^{-7.921}$ and $\delta_{H,43} \approx 10^{-7.954}$, for example.

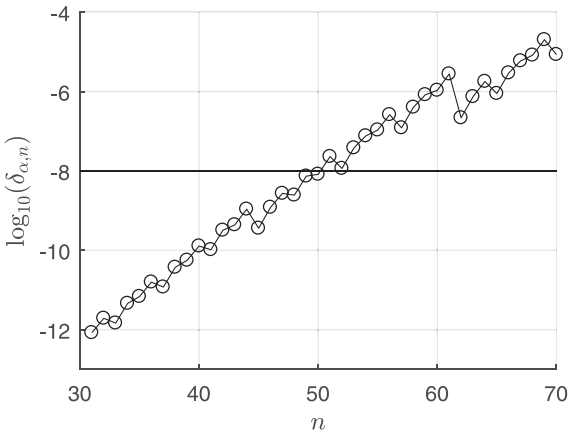

Fig. 2. Evolution of the lower bound error $\delta_{\alpha,n}$. Values plotted using the log_{10} for $\delta_{\alpha,n}$. The line at -8 indicates the value used in the Theorem 2.8.

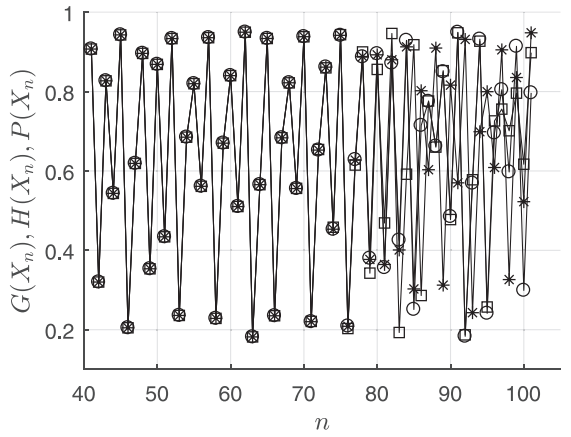

Fig. 3. Simulation of $G(X)$, $H(X)$ and $P(X)$. Results of n iterates for $G(X_n)$ (–o–), $H(X_n)$ (–*–) and $P(X_n)$ (–□–), considering $X_0 = 0.4$ and $r = 3.8$.

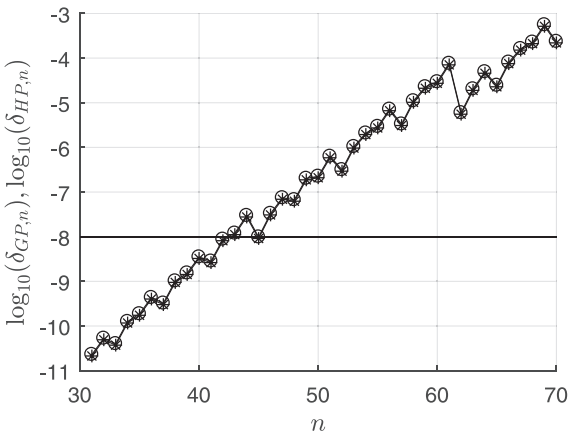

Fig. 4. Evolution of $\delta_{GP,n}$ and $\delta_{HP,n}$ over an interval of n. Depicted are values using the log_{10} for $\delta_{GP,n}$ (–o–) and $\delta_{HP,n}$ (–*–).

Table 1 shows the value of parameter r used for simulation of Eq. (17) and the iteration from which the error is greater than 10^{-8}. The same results were obtained when the simulation was performed either on languages Matlab, Scilab or Octave. Operating systems Windows 8 and Ubuntu 14.04 were considered in this simulation and the results were the same presented in Table 1.

To check if there is a lack of precision, we proceed the same test presented in Table 1 first column, but now using 72 digits of precision, which is approximately equivalent to 256 bits. This presents a higher precision than that was used in

Table 1
Simulation of the model (17) for different values of the parameter r, considering $X_0 = 0.4$. The second column shows the iteration from which the error is greater than 10^{-8} using double precision for the following languages: Matlab, Scilab, Octave, C and Fortran. The third column, performed only in Matlab, presents the result for 72 digits, or approximately 256 bits.

r	Iteration (64 bits)	Iteration ($\approx$ 256 bits)
3.600	84	888
3.625	78	840
3.650	72	628
3.700	55	473
3.750	61	451
3.800	50	389
3.860	45	400
3.910	40	342

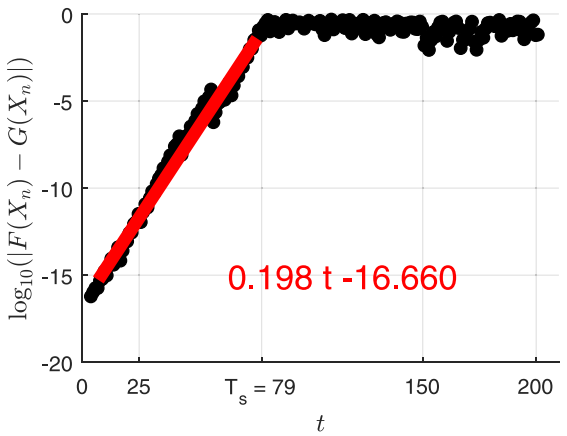

Fig. 5. Illustration of Lemma 2.9 for the Logistic map. The largest Lyapunov exponent is given by the inclination of the fitted line, that is, $\lambda = 0.198$ (here measured in base 10). The critical simulation time $T_s = 79$ is also indicated in the figure. The parameter $r = 3.9$ and initial condition is $x_0 = 0.1$.

[15]. The results are shown in the second column of Table 1. As it was expected, the number of iterates increases, but it is still far away from the mentioned 10^7 iterates in [15].

4.2. Critical simulation time test

4.2.1. Logistic map

To illustrate Lemma 2.9 consider the following example. Let the logistic map with the following two interval extensions $F(X_n) = rX_n(1 - X_n)$ and $G(X_n) = rX_n - rX_nX_n$, where $r = 3.9$ and $X_0 = 0.1$. The lower bound error and Lyapunov exponent ($\lambda = 0.198$) are shown in Fig. 5 (for the sake of simplicity logarithm base 10 is used). Let $D = 1$ and $P \approx 16$, since the logistic equation was simulated on a 64-bit Matlab R2016a environment, then

$$T_s = \frac{\log_{10}(1/2) + 16}{0.198} \approx 79. \tag{26}$$

Note that *i*) $T_s \approx 79$ is in very good agreement with the value shown in Fig. 5, which the lower bound error reaches the same magnitude of the variable x of the logistic map, *ii*) the line fitted to the curve gives the Lyapunov exponent and *iii*) the independent term, 16.6, gives a good estimation of the precision P.

4.2.2. Polynomial NARMAX

The Rössler equations [42] are given by:

$$\begin{cases} \dot{x} = -y - z, \\ \dot{y} = x + ay, \\ \dot{z} = b + z(x - c). \end{cases}$$

This system has been identified by [8] which polynomial NARMAX are given by Eq. (27).

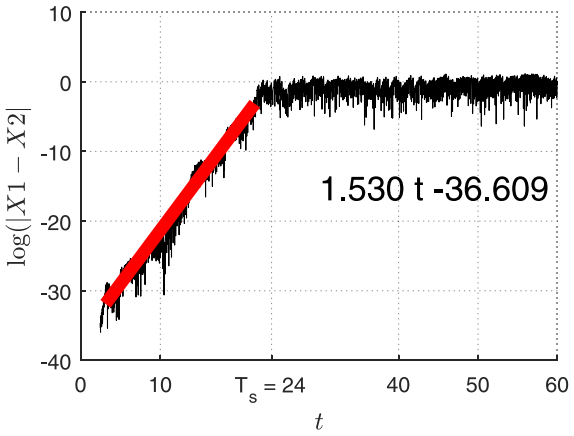

Fig. 6. LLE calculation for the polynomial NARMAX (27) identified Rössler equations [42]. The largest Lyapunov exponent is 1.530, and it is the inclination of the logarithm of the lower bound error. The critical simulation time $T_s = 24$ is indicated and it constitutes an estimation of the maximum time of simulation for this case.

Table 2
Largest positive Lyapunov exponent for the Rössler equations studied in [8].

	Literature (Data)	Literature (Model)	Our method
LLE	1.242	1.566	1.530

$$\begin{aligned}
x(k) = {} & +0.1972 \times 10x(k-1) - 0.104 \times 10x(k-2) \\
& +0.7456 \times 10^{-4}x(k-4)x(k-2)^3x(k-1) \\
& -0.2053 \times 10^{-4}x(k-5)x(k-4)^4 - 0.285 \times 10^{-4}x(k-5)x(k-1)^4 \\
& +0.2484 \times 10^{-4}x(k-3)^2x(k-2)^3 \\
& +0.1238 \times 10^{-2}x(k-2)x(k-1)^2 + 0.4353 \times 10^{-4}x(k-5)^4 \\
& +0.2258 \times 10^{-2}x(k-5)x(k-2)x(k-1)^2 \\
& +0.3123 \times 10^{-4}x(k-4)^5 + 0.7531 \times 10^{-2}x(k-1)^4 \\
& -0.2703 \times 10^{-2}x(k-3)^2x(k-1)^2 - 0.7807 \times 10^{-2}x(k-1)^3 \\
& -0.7077 \times 10^{-4}x(k-3)^2x(k-2)^2x(k-1) \\
& -0.3304 \times 10^{-2}x(k-3)x(k-2)^3 - 0.8847 \times 10^{-2}x(k-5)x(k-1) \\
& +0.7631 \times 10^{-2}x(k-4)x(k-1) - 0.387 \times 10^{-4}x(k-5)^3x(k-1)^2 \\
& +0.4676 \times 10^{-2}x(k-3)^3x(k-1).
\end{aligned} \tag{27}$$

For this case, the two interval extensions are yielded by the change of the term

$$+0.7456 \times 10^{-4}x(k-4)x(k-2)^3x(k-1)$$

into

$$+0.7456 \times 10^{-4}x(k-4)x(k-2)x(k-2)^2x(k-1).$$

As one can see, the interval extensions used to simulate the systems are mathematically equivalent, but they differ in the sequence of arithmetical operations, using associative property of multiplication. Fig. 6 shows the logarithm of the lower bound error of the simulation using the two specified interval extensions. Table 2 summarises the result of LLE for Rössler equations. The model used has been identified in [8]. The literature suggests a value of 1.242 and model identified gives 1.566. The result seems slight different, but if we take into account the confidence interval to calculation of LLE using data, it is more appropriated given by 1.242 ± 0.399. Thus, the value of LLE calculated in [8] is in good agreement with the original system. Similar reasoning can be applied to method presented in this paper, in which the LLE is calculated as 1.530.

5. Conclusions

In summary, we have demonstrated a technique to detect unreliable computer simulation of recursive functions. We show that, on the contrary to what is shown in many papers in literature, the logistic map simulations using the tested languages Matlab, Scilab, Octave, C and Fortran and double precision is not reliable for millions of iterates for the initial

condition and parameters suggested in [15]. In fact, after less than one hundred of iterates, the simulation is no longer reliable in double precision. Even for a higher precision of approximately 256 bits on Matlab, the results are very distant from those presented in [15].

Our proposed method can be seen as test to falsify reliability of simulation based on the shadowing property for recursive functions, such as discrete maps. It is clear that a true orbit cannot agree with certain accuracy simultaneously to two pseudo-orbits that diverge from each other. This is the very basic idea that underlines our approach. Our method is compared with the principal result obtained by Hammel and colleagues [15] for the case of logistic map. After almost 30 years, and with more than 150 citations according to Scopus, that paper has been still receiving citations, considered by many as a corner stone to prove that the logistic map, for some combinations of parameters and initial conditions, present the shadowing property. On the contrary, we show that this is not a general true. One important aspect is that, following the principles of IEEE 754–2008 standard on floating point, we are able to find the same results for the five languages used in two different operating systems, as can be seen in Table 1. The number of iterates that the lower bound error is greater than 10^{-8} are the same for all tests. This seems to be an evidence of robustness of our method, and a clear evidence that the proved theorem by Hammel and colleagues is not valid for such cases. We believe that a diffusion of this result should be an important contribution to nonlinear science community.

This paper also used the largest Lyapunov exponent to indicate a critical simulation time (T_s) in order to stop the simulation of chaotic systems for a given precision P. Different from other methods present in the literature, as in [21], our method uses the lower bound error to calculate the largest Lyapunov exponent. Using this method, we make a clear connection between error propagation and the meaning of the Lyapunov exponent, offering in such way, a simple way to determine the maximum number of iterates. In our case, the logistic map with $r = 3.9$ and $x_0 = 0.1$ presents a largest Lyapunov exponent $\lambda = 0.198$, which means that using double precision it would reliable to simulate around 79 iterates. This has direct implications, for instance, to build diagram bifurcation of the logistic map, for which, double precision allows to iterate a maximum of 79 iterates, considering the transient and stationary regime. This value is significant lower that number presented in the literature (See [35] to see a step-by-step method to build a bifurcation diagram, in which 500 iterates are considered as transient). For the polynomial NARMAX identified to Rössler equation, we found a $\lambda = 1.530$ which gives a maximum simulation time of $T_s = 24$. This is another important aspect that must be accompanied with the identified model to the user be aware of the limits on the simulation of such model.

The present paper emphasises the requirement of rigorous care to simulate the chaotic systems for long time without a serious care about the precision. This is important in order to build bifurcation diagram and free-run simulation of polynomial NARMAX, for instance. We believe that the result presented in this paper is important for the rigorous development of the nonlinear dynamics field, which is highly dependent on computer arithmetic [38]. The method presented is also an alternative tool to check non-reliable computer simulations based on recursive functions, such as maps or discretization schemes of continuous systems.

Acknowledgments

The authors are thankful to the Brazilian Research Agencies CNPq/INERGE, FAPEMIG and CAPES, and to the Slovenian Research Agency (Grants P5-0029 and J1-7009). We would also like to express our deepest gratitude to Prof. Dr. Eduardo M. A. M. Mendes for his continuous encouragement, reading of the manuscript and valuable follow up.

References

[1] M. Barahona, C. Poon, Detection of nonlinear dynamics in short, noisy time series, Nature 381 (6579) (1996) 215–217.
[2] S.A. Billings, Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains, John Wiley & Sons, West Sussex, 2013.
[3] N. Brisebarre, C. Lauter, M. Mezzarobba, J. Muller, Comparison between binary and decimal floating-point numbers, IEEE Trans. Comput. 65 (7) (2016) 2032–2044.
[4] F. Chaitin-Chatelin, Is finite precision arithmetic useful for physics? J. Univers. Comput. Sci. 2 (5) (1996) 380–395.
[5] S. Chen, S.A. Billings, Representations of non-linear systems: the {Narmax} model, Int. J. Control 49 (3) (1989) 1013–1032.
[6] R.M. Corless, What good are numerical simulations of chaotic dynamical systems? Comput. Math. Appl. 28 (10) (1994) 107–121.
[7] R.M. Corless, C. Essex, M.A.H. Nerenberg, Numerical methods can suppress chaos, Phys. Lett. A 157 (1) (1991) 27–36.
[8] M.V. Correa, L.A. Aguirre, E.M.A.M. Mendes, Modeling chaotic dynamics with discrete nonlinear rational models, Int. J. Bifurc. Chaos 10 (05) (2000) 1019–1032.
[9] D. Epstein, S. Levy, Experimentation and proof in mathematics, Not. Am. Math. Soc. 42 (6) (1995) 670–674.
[10] J. Ford, Chaos: solving the unsolvable, predicting the unpredictable, Chaotic Dynamics and Fractals, Academic Press, 1986.
[11] 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. 13 (3) (2013) 35–52.
[12] Z. Galias, W. Tucker, Is the hénon attractor chaotic? Chaos: Interdiscip. J. Nonlinear Sci. 25 (3) (2015) 033102.
[13] R. Gilmore, M. Lefranc, N.B. Tufillaro, The Topology of Chaos: Alice in Stretch and Squeezeland, Wiley-Interscience, 2012.
[14] E. Guralnik, M. Aharoni, A.J. Birnbaum, A. Koyfman, Simulation-based verification of floating-point division, IEEE Trans. Comput. 60 (2) (2011) 176–188.
[15] S.M. Hammel, J.A. Yorke, C. Grebogi, Do numerical orbits of chaotic dynamical processes represent true orbits? J. Complex. 3 (2) (1987) 136–145.
[16] A. Hasan, E.C. Kerrigan, G.A. Constantinides, Control-theoretic forward error analysis of iterative numerical algorithms, IEEE Trans. Autom. Control 58 (6) (2013) 1524–1529.
[17] W.B. Hayes, K.R. Jackson, C. Young, Rigorous high-dimensional shadowing using containment: the general case, Dyn. Syst. 14 (2) (2006) 329–342.
[18] Institute of electrical and electronics engineers (IEEE), 2008, IEEE standard for floating-point arithmetic IEEE. 754–2008.
[19] Institute of electrical and electronics engineers (IEEE), 2015, IEEE Standard for Interval Arithmetic IEEE Std 1788–2015. 1–97.

[20] M. Joldes, O. Marty, J. Muller, V. Popescu, Arithmetic algorithms for extended precision using floating-point expansions, IEEE Trans. Comput. 65 (4) (2016) 1197–1210.
[21] S. Lao, On the reliability of computed chaotic solutions of non-linear differential equations, Tellus A 61 (2009) 550–564.
[22] E.N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. 20 (2) (1963) 130–141.
[23] E.N. Lorenz, Computacional periodicity as observed in a simple system, Tellus 58A (2006) 549–557.
[24] R. Lozi, Can we trust in numerical computations of chaotic solutions of dynamical systems? in: Topology and dynamics of Chaos, World Scientific Series in Nonlinear Science Series A, volume 84, World Scientific, 2013, pp. 63–98.
[25] E.E.N. Macau, C.L.P. Lambruschini, Advanced computational and experimental techniques in nonlinear dynamics, Eur. Phys. J. Spec. Top. 223 (2014) 2645–2648.
[26] R.M. May, Simple mathematical models with very complicated dynamics, Nature 261 (5560) (1976) 459–467.
[27] E.M.A.M. Mendes, E.G. Nepomuceno, A very simple method to calculate the (positive) largest Lyapunov exponent using interval extensions, Int. J. Bifurc. Chaos 26 (13) (2016) 1650226.
[28] R.E. Moore, R.B. Kearfott, M.J. Cloud, Introduction to Interval Analysis, Society for Industrial and Applied Mathematics (2009) 1–49.
[29] R.E. Moore, R.E. Moore, Methods and applications of interval analysis, Vol. 2, SIAM, 1979, pp. 1–31.
[30] E.N. Lorenz, Computational chaos - a prelude to computational instability, Phys. D 35 (1989) 299–317.
[31] E.G. Nepomuceno, Convergence of recursive functions on computers, The Journal of Engineering 10 (2014) 560–562 https://doi.org/10.1049/joe.2014.0228.
[32] E.G. Nepomuceno, S.A.M. Martins, A lower bound error for free-run simulation of the polynomial NARMAX, Syst. Sci. Control Eng. 4 (1) (2016) 50–58.
[33] E.G. Nepomuceno, E.M. Mendes, On the analysis of pseudo-orbits of continuous chaotic nonlinear systems simulated using discretization schemes in a digital computer, Chaos Solitons Fractals 95 (2017) 21–32.
[34] N. Oh, S. Mitra, E.J. McCluskey, ED/sup 4/i: error detection by diverse data and duplicated instructions, IEEE Trans. Comput. 51 (2) (2002) 180–199.
[35] E. Ott, Chaos in Dynamical Systems (Second edition), Cambridge University Press, Cambridge, 2002.
[36] M.L. Overton, Numerical Computing with IEEE Floating Point Arithmetic, SIAM, 2001, pp. 1–39, doi:10.1137/1.9780898718072.
[37] J. Palmore, C. Herring, Computer arithmetic, chaos and fractals, Phys. D: Nonlinear Phenom. 42 (1–3) (1990) 99–110.
[38] B. Parhami, Computer Arithmetic algorithms and Hardware Architectures, Oxford University Press, New York, 2010.
[39] A.A. Petrov, S.Y. Pilyugin, Lyapunov functions, shadowing and topological stability, Topological Methods in Nonlinear Analysis 43 (1) (2014) 231–240.
[40] N. Revol, P. Theveny, Numerical reproducibility and parallel computations: issues for interval algorithms, IEEE Trans. Comput. 63 (8) (2014) 1915–1924.
[41] M.S. Rodrigues, E.G. Nepomuceno, G.F.V. Amaral, V.V.R. Silva, Simulation of Chua's circuit by means of interval analysis, in: Proceedings of Anals of 6th International Conference on Nonlinear Science and Complexity, São José dos Campos - Brazil, 2016, pp. 1–4.
[42] O.E. Rössler, An equation for continuous chaos, Phys. Lett. A 57 (5) (1976) 397–398.
[43] M. Šalamon, T. Dogša, Problem of non-repeatability of the circuits simulation [problem neponovljivosti simulacij električnih vezij], Inf. MIDEM - J. Microelectron. Electron. Compon. Mater. 34 (1) (2004) 11–17.
[44] B.C. Silva, F.L. Milani, E.G. Nepomuceno, S.A.M. Martins, G.F.V. Amaral, Revisiting Hammel et al. (1987): does the shadowing property hold for modern computers? in: Proceedings of the 6th International Conference on Nonlinear Science and Complexity, 2016, pp. 1–4.
[45] J. Teixera, C.A. Renyolds, K. Judd, Time step sensivity of nonlinear atmospheric models: numerical convergnce, truncation error growth, and ensemble design, J. Atmosp. Sci. 64 (1) (2007) 175–189.
[46] C. Varsakelis, P. Anagnostidis, On the susceptibility of numerical methods to computational chaos and superstability, Commun. Nonlinear Sci. Numer. Simul. 33 (2016) 118–132.
[47] M. Yabuki, T. Tsuchiya, Double Precision Computation of the Logistic Map Depends on Computational Modes of the Floating-point Processing Unit. (2013). arXiv Preprint arXiv:1305.3128, 1–10. Retrieved from http://arxiv.org/abs/1305.3128v1f