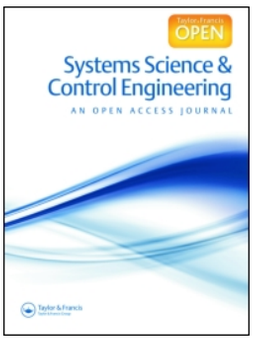

Systems Science & Control Engineering

An Open Access Journal

ISSN: (Print) 2164-2583 (Online) Journal homepage: https://www.tandfonline.com/loi/tssc20

On the lower bound error for discrete maps using associative property

E.G. Nepomuceno, S.A.M. Martins, G.F.V. Amaral & R. Riveret

To cite this article: E.G. Nepomuceno, S.A.M. Martins, G.F.V. Amaral & R. Riveret (2017) On the lower bound error for discrete maps using associative property, Systems Science & Control Engineering, 5:1, 462-473, DOI: 10.1080/21642583.2017.1387874

To link to this article: https://doi.org/10.1080/21642583.2017.1387874

Published online: 19 Oct 2017.

SYSTEMS SCIENCE & CONTROL ENGINEERING: AN OPEN ACCESS JOURNAL, 2017
VOL. 5, NO. 1, 462–473
https://doi.org/10.1080/21642583.2017.1387874

Taylor & Francis
Taylor & Francis Group

On the lower bound error for discrete maps using associative property

E.G. Nepomuceno[a], S.A.M. Martins[a], G.F.V. Amaral[a] and R. Riveret[b]

[a]Control and Modelling Group (GCOM), Department of Electrical Engineering, Federal University of São João Del-Rei, São João Del-Rei, Brazil; [b]DATA61, CSIRO, Brisbane, Australia

ABSTRACT

This paper introduces a class of pseudo-orbits which guarantees the same lower bound error (LBE) for two different natural interval extensions of discrete maps. In previous work, the LBE was investigated along with a simple technique to evaluate numerical accuracy of free-run simulations of polynomial NARMAX or similar discrete maps. Here we prove that it is possible to calculate the LBE for two pseudo-orbits, extending so the results of previous work in which the LBE is valid for only one of the two pseudo-orbits. The main application of this technique is to provide a simple estimation of the LBE. We illustrate our approach with the Logistic Map and Hénon Map. Using double precision, our results show that we ought simulate the Logistic Map and Hénon Map with less than 100 iterations, which is, for instance, far less than the number usually considered as transient to build bifurcation diagrams.

ARTICLE HISTORY
Received 11 July 2017
Accepted 1 October 2017

KEYWORDS
Nonlinear dynamics and chaos; Numerical simulation; Lower bound error

1. Introduction

Discrete maps, also called recursive functions, are used to account for many types of systems (Feigenbaum, 1978), in particular nonlinear dynamics systems (Ferreira, Nepomuceno, & Cerqueira, 2006; Hammel, Yorke, & Grebogi, 1987; Lozi, 2013; Pereira, Kurcbart, & Nepomuceno, 2005). To investigate these functions in nonlinear dynamical systems, numerical computation plays a key role (Galias, 2013; Lozi, 2013), and a myriad of numerical experiments (Peck, 2004) have been performed since the work of Lorenz (1963) to understand the behaviour of such nonlinear dynamical systems (Hammel et al., 1987). They are used as direct tool to model many types of dynamical systems (Feigenbaum, 1978) and many types of discretization schemes for continuous nonlinear dynamical systems make use of discrete maps (Letellier, Mendes, & Mickens, 2007).

Numerous researchers are confident in numerical solutions of nonlinear dynamical systems. All sort of dynamics have been classified, such as chaotic (Ott, 2002) or intermittent (Hirsch, Huberman, & Scalapino, 1982) dynamics, whose conclusions are based on numerical simulations. However, there exist doubts about these dynamics. The work of Ford (1986), with his intriguing title: 'Chaos: Solving the unsolvable, predicting the unpredictable', is one of the first attempts to articulate some discussions on the discovered evidences by computer simulations. Later Lozi (2013) 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?'. Similarly, Galias (2013) expresses the importance of developing methods to prove the existence of chaotic attractors; the existence for Lorenz system have been carried successfully (Tucker, 1999a, 1999b), while existence for Chua's circuit is still open. Nevertheless, it is important to state, that the proof in Tucker (1999a, 1999b) is based on 'combination of normal form theory and rigorous numerics'. As any discretization scheme may be seen as a recursive function, difficulties have been reported to keep reliability on the results of numerical simulation (Amigó, Kocarev, & Szczepanski, 2007; Berthé, 2012; Elabbasy, Elsadany, & Zhang, 2014; Letellier & Mendes, 2005; Lima, Claro, Ribeiro, Xavier, & López-Castillo, 2013; Mendes & Letellier, 2004; Nepomuceno & Mendes, 2017; Rodríguez & Barrio, 2012; Teixeira, Reynolds, & Judd, 2007). Hence, the error propagation in computer simulations requires a careful analysis (Lozi, 2013).

Based on a preliminary work about convergence of recursive functions on a computer (Nepomuceno, 2014), Nepomuceno and Martins (2016) proposed a method to calculate the lower bound error (LBE) for free-run simulations of recursive functions, with particular attention to polynomial NARMAX (Billings, 2013). In that work, a simulation is performed by means of two natural interval extensions, derived from mathematical properties, such

CONTACT E.G. Nepomuceno nepomuceno@ufsj.edu.br

as commutativity, distributivity or associativity. These two natural interval extensions produce two different pseudo-orbits. The LBE has already been shown useful in works such as Mendes and Nepomuceno (2016), where the authors have developed a fast and robust method to calculate the positive largest Lyapunov exponent, and Nepomuceno and Mendes (2017), where the authors analyse the consequences of error propagation in discretization schemes to solve nonlinear differential equations.

In Nepomuceno and Martins (2016), the results on the LBE have been proved to be valid for only one of the two pseudo-orbits. In this work, we present a step further in the consolidation of the LBE. We found a class of natural interval extensions, which derives pseudo-orbits with equal error bound when performing the rules of float point based on the IEEE 754-2008 standard (Goldberg, 1991; Institute of Electrical and Electronics Engineers (IEEE), 2008; Overton, 2001). The investigation of different behaviours from *different expressions* for the Logistic Map has also been studied in Yabuki and Tsuchiya (2013). We may think that an arithmetical operation in computer is not exact but *exact within a factor* of $(1+\varepsilon)$, where ε is the machine epsilon (Overton, 2001). The associative property of multiplication allows us to produce different pseudo-orbits, while keeping the same error bound. In this way, we prove a theorem that states the lower bound error for this kind of natural interval extension, called arithmetic interval extension, and for two pseudo-orbits. We also focus our attention on discrete maps, using the Logistic Map and Hénon Map as examples of application. Another contribution regards the notation of the LBE to clarify a connection with absolute error and relative error in floating-point arithmetic.

The remainder is organized as follows. First, preliminary concepts are presented, then we propose a new theorem on the lower bound of two pseudo-orbits. We apply the methodology for the Logistic Map and Hénon Map, and the results are validated by means of a comparison with an arbitrary precision.

2. Preliminary concepts

In this section, we present the concepts of natural interval extensions, orbits and pseudo-orbits and a brief overview of floating point arithmetic as specified by IEEE floating point standard 754-2008 Institute of Electrical and Electronics Engineers (IEEE) (2008). These concepts are used to establish the main contribution of this paper, as the lower bound error is derived from two different natural interval extensions and their pseudo-orbits under the arithmetic carried out by IEEE 754-2008.

2.1. Natural interval extension

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 $[\underline{X},\bar{X}]$, i.e. $X=\{x:\underline{X}\leq x\leq\bar{X}\}$. In an degenerated interval, we have $\underline{X}=\bar{X}$ and such an interval amounts to a real number $x=\underline{X}=\bar{X}$. In this context, Moore, Kearfott, and Cloud (2009) give the definition of natural interval extensions of a function.

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

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

An interval extension of f is thus an interval valued function which has real values when the arguments are real (degenerate intervals) and coincides with f in this case. The natural interval extension is achieved by combining the function rule $f(x)$ with the equivalents of the basic arithmetic and elementary functions.

There are natural interval extensions that are equivalent in terms of intervals. This leads us to the following definition.

Definition 2.2 (Equivalent interval extension): Two natural interval extensions G and H of a function f are *equivalent* if

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

Example 2.3: Let us consider the following extension intervals:

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

$$H(X)=r(X(1-X)), \tag{2}$$

$$L(X)=rX-rX^2. \tag{3}$$

If $r=3$ and $X=[0.3,0.4]$, then we have

$$G([0.3,0.4])=3[0.3,0.4](1-[0.3,0.4])=[0.54,0.84],$$
$$H([0.3,0.4])=3([0.3,0.4](1-[0.3,0.4]))=[0.54,0.84],$$
$$L([0.3,0.4])=3[0.3,0.4]-3([0.3,0.4]^2)=[0.42,0.93].$$

Here, only $G(X)$ and $H(X)$ are equivalent interval extensions.

It is imperative to stress that from the perspective of classic mathematics on real numbers, diverse syntactic

partners of two equivalent natural interval extensions yield a similar outcome. However, the outcome might be distinctive in interval arithmetic, specifically when the arithmetic operations are performed on computers using float point operations.

2.2. IEEE floating point representation

Finite IEEE floating point numbers can be expressed as in the following form Overton (2001):

$$\pm (b_0.b_1b_2b_3\cdots b_{p-1}) \times 2^E, \tag{4}$$

where p is the precision of the floating point system; b_0 to b_{p-1} is the binary expansion of significand S; E is the exponent.

Example 2.4: Let us consider the single precision, that is, 32 bits, where 1 bit is for the signal, 8 is for the exponent (represented in a bias format) and 23 is reserved for the significand. The decimal 0.1 is approximated by a binary string

$$00111101110011001100110011001101$$

where the first 0 represents the positive signal of Equation (4), the following binary string 01111011 represents the biased exponential, $E = 123 - 127 = -4$, and finally the significand is given by 10011001100110011001101. Recall that $b_0 = 1$ is not represented and it is denominated hidden bit. Using Equation (4), we have:

$$0.1 \approx +(1.10011001100110011001101) \times 2^{-4}. \tag{5}$$

Let $x \in \mathbb{R}$ be a real in a computer using the IEEE standard. We define x_- to be the floating point number nearest to x that is less than or equal to x, and define x_+ to be the floating point number nearest to x that is greater than or equal to x. The IEEE standard defines the *correctly rounded value of x*, denoted here as round(x). If x is a floating point number, then round$(x) = x$. Otherwise, round(x) depends on which rounding modes is in effect, as follows:

- Round down: round$(x) = x_-$.
- Round up: round$(x) = x_+$.
- Round towards zero: round$(x) = x_-$ if $x > 0$; round $(x) = x_+$ if $x < 0$.
- Round to nearest: round(x) is either x_- or x_+, whichever is nearer to x.

The absolute rounding error associated with x is defined as

$$\text{abserr}(x) = |\kappa| \tag{6}$$

where

$$\kappa = \text{round}(x) - x, \tag{7}$$

and the relative rounding error is

$$\text{relerr}(x) = |\delta|, \tag{8}$$

where

$$\delta = \frac{\text{round}(x) - x}{x}. \tag{9}$$

Therefore, for a binary floating point system with precision p, we have:

$$\text{round}(x) = x(1 + \delta) \tag{10}$$

$$= x + \kappa \tag{11}$$

for some δ and κ satisfying

$$|\delta| \leq \varepsilon, \tag{12}$$

$$|\kappa| \leq \gamma. \tag{13}$$

where $\varepsilon = 2^{-p}$ and $\gamma = \varepsilon \times 2^E$ if round mode in effect is rounding to nearest and considering x within the normalized range (Overton, 2001).

A key feature of the IEEE standard is that it requires correctly rounded arithmetic operations. Let x and y denote float point numbers, let $+, -, \times$ and $\div$ the standard arithmetic operations, and let $\oplus, \ominus, \otimes$ and $\oslash$ the equivalent operations as they are actually implemented on computers. The rule is as follows: if x and y are floating point numbers, then (Overton, 2001)

$$x \oplus y = \text{round}(x + y) = (x + y)(1 + \delta), \tag{14}$$

$$x \ominus y = \text{round}(x - y) = (x - y)(1 + \delta), \tag{15}$$

$$x \otimes y = \text{round}(x \times y) = (x \times y)(1 + \delta), \tag{16}$$

$$x \oslash y = \text{round}(x \div y) = (x \div y)(1 + \delta). \tag{17}$$

We can 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

$$(x + y)(1 - \varepsilon) \leq x \oplus y \leq (x + y)(1 + \varepsilon) \tag{18}$$

which is equivalent to say

$$x \oplus y \in [(x + y)(1 - \varepsilon), (x + y)(1 + \varepsilon)]. \tag{19}$$

We may continue likewise for subtraction, multiplication and division. Henceforth we may revise Equations (14)

to (17) using a description of intervals as follows.

$$x \oplus y \in Z = [\underline{Z}, \bar{Z}] = [(x+y)(1-\varepsilon), (x+y)(1+\varepsilon)], \tag{20}$$

$$x \ominus y \in Z = [\underline{Z}, \bar{Z}] = [(x-y)(1-\varepsilon), (x-y)(1+\varepsilon)], \tag{21}$$

$$x \otimes y \in Z = [\underline{Z}, \bar{Z}] = [(x \times y)(1-\varepsilon), (x \times y)(1+\varepsilon)], \tag{22}$$

$$x \oslash y \in Z = [\underline{Z}, \bar{Z}] = [(x \div y)(1-\varepsilon), (x \div y)(1+\varepsilon)]. \tag{23}$$

It is known that certain properties of classic arithmetic, such as associativity of addition, do not hold in the context of floating point arithmetic (Institute of Electrical and Electronics Engineers (IEEE), 2008; Overton, 2001). Consequently, two mathematical equivalent sequences of arithmetic operations can yield two different results, as in the case of orbits and pseudo-orbits of maps.

2.3. Orbits and pseudo-orbits

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

$$x_{n+1} = f(x_n), \tag{24}$$

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 Equation (24) starting from an initial condition x_0 is called the orbit of x_0 (Gilmore, Lefranc, & Tufillaro, 2012).

Definition 2.5 (Orbit): Given a map $x_{n+1} = f(x_n)$, an orbit of the map is a sequence of values of the map, represented by $\{x_n\} = [x_0, x_1, \ldots, x_n]$.

The calculation of an orbit is usually carried out by a finite-precision computer, leading to a pseudo-orbit. A *pseudo-orbit* of a map is an approximation of a mathematical orbit in a specific hardware or software. There is no unique pseudo-orbit, as there are different hardware, software and numerical precision standards, such as IEEE 754-2008, which may yield different outputs for each extension interval. Here it is important to emphasize that some pseudo-orbits may produce better results than others. For this reason, there has been different investigations to obtain the most precise pseudo-orbits; for example by manipulating natural interval extensions, see e.g. Horner's method (Lambers & Sumner, 2016; Muller et al., 2010; Rodríguez & Barrio, 2012; Stahl, 1997), or by using specific interval extensions, such as mean value forms, which usually produce narrower widths (Caprani & Madsen, 1980; Dymowa, 2011; Rall, 1983). In this paper we do not focus on the reduction of the propagation error or the width of the error bounds.

Definition 2.6 (Pseudo-orbit): Given a map $x_{n+1} = f(x_n)$, an ith pseudo-orbit $\{\hat{x}_{i,n}\}$ is an approximation of an orbit such that

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

with the absolute error given by

$$\text{abserr}_{i,n} = |\kappa_{i,n}| \tag{25}$$

where

$$\kappa_{i,n} = \hat{x}_{i,n} - x_n. \tag{26}$$

The relative error is given by

$$\text{relerr}_{i,n} = |\delta_{i,n}| \tag{27}$$

where

$$\delta_{i,n} = \frac{x_n - \hat{x}_{i,n}}{x_n}. \tag{28}$$

Thus, for a floating point system, we have:

$$\hat{x}_{i,n} = x_n(1+\delta_{i,n}) \tag{29}$$

$$= x_n + \kappa_{i,n}, \tag{30}$$

for some $\delta_{i,n}$ and $\kappa_{i,n}$ satisfying

$$|\delta_{i,n}| \leq \varepsilon_{i,n}, \tag{31}$$

$$|\kappa_{i,n}| \leq \gamma_{i,n}. \tag{32}$$

It is clear that for $n = 0$, Equations (29) and (32) are equivalent to Equations (10) and (13), that is, $\delta_{i,0} = \delta$ and $\varepsilon_{i,0} = \varepsilon$, wherein the index i could be dropped as there is no pseudo-orbit in action, but the process of rounding. It is important to stress out that the IEEE standard does not mention any general approach to calculate $\varepsilon_{i,n}$ or $\gamma_{i,n}$ for $n \geq 1$. This has been the focus of many works, particularly those related to interval arithmetic (Moore et al., 2009). Here we may define an interval associated with each value of a pseudo-orbit, $I_{i,n}$ such that

$$x_{n+1} \in I_{i,n} = [f(x_n)(1-\varepsilon_{i,n}), f(x_n)(1+\varepsilon_{i,n})] \tag{33}$$

$$= [f(x_n) - \gamma_{i,n}, f(x_n) + \gamma_{i,n}]. \tag{34}$$

Let $\{\hat{x}_{a,n}\}$ and $\{\hat{x}_{b,n}\}$ be two pseudo-orbits, here we denote the LBE as $\ell_{\Omega,n}$, where ℓ is the LBE calculated at a point

n using a set of pseudo-orbits indicated by Ω. In this paper, Ω is defined with just two interval natural extensions, denoted by $\Omega = [\{\hat{x}_{a,n}\}, \{\hat{x}_{b,n}\}]$. A preliminary result in Nepomuceno and Martins (2016) indicates that $\gamma_{a,n} \geq \ell_{\Omega,n}$ or $\gamma_{b,n} \geq \ell_{\Omega,n}$. In this paper we provide a class of pseudo-orbit which allows to state that $\gamma_{a,n} = \gamma_{b,n} \geq \ell_{\Omega,n}$. It is important to stress that $\gamma_{a,n} = \gamma_{b,n}$ does not imply $\kappa_{a,n} = \kappa_{b,n}$. In other words, the class of natural interval extensions we are going to introduce here presents the equivalent bounds for the absolute error, but the absolute error for a specific instant n may be different (see Figure 1(a)).

3. Lower bound error

The key point of this paper is the analysis of a specific class of natural interval extensions based on the associative property of multiplication. This leads us to the following definition of equivalent arithmetic intervals, cf. Definition 2.2.

Definition 3.1 (Arithmetic interval extension): Two equivalent natural interval extensions G and H of a function f are *arithmetic interval equivalent* if they produce the same resulting interval following Equation (20) to (23).

Manipulations on terms of a function by means of associative property of multiplication will keep the same bounds of error. We indicate this key aspect in the following Lemma.

Lemma 3.2: *Associative property of multiplication does not change the error bounds in float point arithmetical operation using IEEE 754-2008.*

Proof: Let us consider a generic mathematical operation $a \times b \times c$, its float point equivalent $a \otimes b \otimes c$, and the machine epsilon ε. We have

$$a \otimes b \otimes c = a \otimes (b \otimes c)$$
$$((a \times b)(1+\varepsilon)) \otimes c = a \otimes ((b \times c)(1+\varepsilon))$$
$$(((a \times b)(1+\varepsilon)) \times c)(1+\varepsilon) = (a \times ((b \times c)(1+\varepsilon)))(1+\varepsilon)$$
$$(a \times b \times c)(1+\varepsilon)^2 = (a \times b \times c)(1+\varepsilon)^2$$

■

Example 3.3: Let us consider the map of the Logistic Equation $f(x) = rx(1-x)$. 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

$$\begin{aligned} Z_G &= r \otimes x \otimes (1 \ominus x) \\ &= r \otimes x \otimes [(1-x)(1-\varepsilon), (1-x)(1+\varepsilon)] \\ &\in [rx(1-\varepsilon), rx(1+\varepsilon)] \otimes [(1-x)(1-\varepsilon), \\ &\quad \times (1-x)(1+\varepsilon)] \\ &= [rx(1-x)(1-\varepsilon)^2(1-\varepsilon), rx(1-x)(1+\varepsilon)^2(1+\varepsilon)] \\ &= [rx(1-x)(1-\varepsilon)^3, rx(1-x)(1+\varepsilon)^3] \end{aligned} \tag{35}$$

whereas for $H(x)$ we have

$$\begin{aligned} Z_H &= r \otimes (x \otimes (1 \ominus x)) \\ &= r \otimes (x \otimes [(1-x)(1-\varepsilon), (1-x)(1+\varepsilon)] \\ &= r \otimes ([x(1-x)(1-\varepsilon)(1-\varepsilon), x(1-x)(1+\varepsilon) \\ &\quad \times (1+\varepsilon)] \\ &= [rx(1-x)(1-\varepsilon)^3, rx(1-x)(1+\varepsilon)^3]. \end{aligned} \tag{36}$$

Thus

$$f(x) \in [rx(1-x)(1-\varepsilon)^3, rx(1-x)(1+\varepsilon)^3] = Z_G = Z_H \tag{37}$$

As we can see, the intervals in (35) and (36) are the same. Although, the computational outcome for $G(x)$ and $H(x)$ is likely unique, they present the same error bounds due to round off.

Another important point to emphasize in this case is that $f(\cdot)$ is a map, and the error produced by round off is the same for $G(x)$ and $H(x)$ for any iteration of a map. This is the key aspect that we are going to apply to develop a stronger result on lower bound error.

Arithmetic equivalent extensions can be used to refine the following theorem proved in Nepomuceno and Martins (2016).

Theorem 3.4: *Let $\{\hat{x}_{a,n}\}$ and $\{\hat{x}_{b,n}\}$ be two pseudo-orbits derived from two natural interval extensions. Let $\ell_{\Omega,n} = |\hat{x}_{a,n} - \hat{x}_{b,n}|/2$ be the lower bound error associated with the set of pseudo-orbits $\Omega = [\{\hat{x}_{a,n}\}, \{\hat{x}_{b,n}\}]$ of a map, then $\gamma_{a,n} \geq \ell_{\Omega,n}$ or $\gamma_{b,n} \geq \ell_{\Omega,n}$.*

Theorem 3.4 is limited in that at least one of the two pseudo-orbits has an error bound equal or greater than the lower bound error $\ell_{\Omega,n}$. We refine here this result for a class of natural interval extensions, namely the class of *equivalent arithmetic interval extensions* (Definition 3.1), for which error bound are the same. So a stronger result for the lower bound error is given by the following theorem.

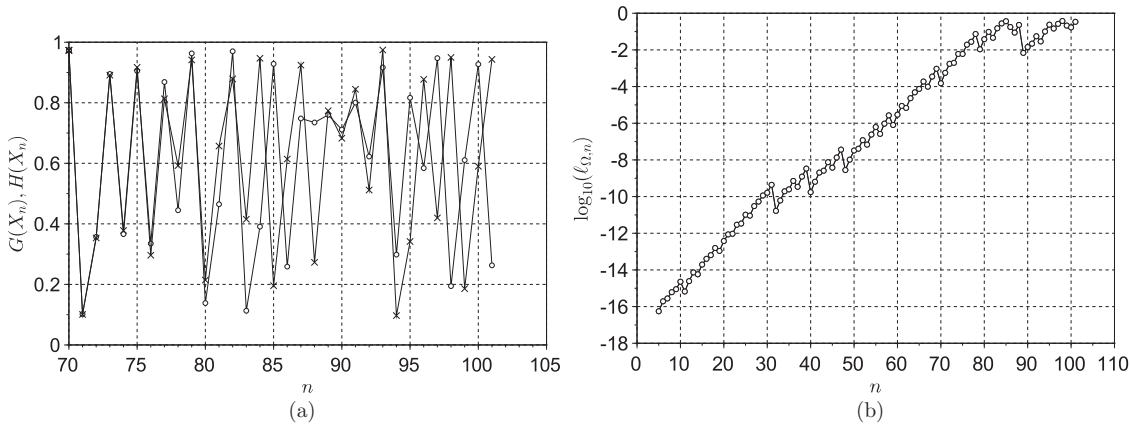

Figure 1. (a) Logistic Map for two pseudo-orbits and (b) evolution of lower bound error. (a) Simulation of Equations (39) and (40), with results for $G(X_n)$ $(-\times-)$ and $H(X_n)$ $(-\text{o}-)$ and the same initial condition $X_0 = 0.1$. n stands for the number of iterations and (b) Evolution of lower bound error $\ell_{\Omega,n}$ of Equation (38). The values are plotted using $\log_{10}$. When $n = 83$, $\ell_{\Omega,n} > 0.1$, and thus all digits have lost significance.

Theorem 3.5: *Let* $\{\hat{x}_{a,n}\}$ *and* $\{\hat{x}_{b,n}\}$ *be two pseudo-orbits derived from two arithmetic interval extensions. Let* $\ell_{\Omega,n} = |\hat{x}_{a,n} - \hat{x}_{b,n}|/2$ *be the lower bound error associated to the set of pseudo-orbits* $\Omega = [\{\hat{x}_{a,n}\}, \{\hat{x}_{b,n}\}]$ *of a map, then* $\gamma_{a,n} = \gamma_{b,n} \geq \ell_{\Omega,n}$.

Proof: The two interval natural extensions are arithmetic interval extensions. By Lemma 3.2, it holds that $\varepsilon_{a,0} = \varepsilon_{b,0} = \varepsilon$. As $\gamma = \varepsilon \times 2^E$, we also have $\gamma_{a,0} = \gamma_{b,0}$. By induction, this result can be further extended to $\gamma_{a,n} = \gamma_{b,n}$. Now we need to show the definition of $\ell_{\Omega,n}$ implies $\gamma_{a,n} = \gamma_{b,n} \geq \ell_{\Omega,n}$. It is known that $I_{a,n} \cap I_{b,n} \neq \emptyset$ if and only if

$$\hat{x}_{b,n} - \gamma_{b,n} \leq \hat{x}_{a,n} + \gamma_{a,n}$$

and

$$\hat{x}_{a,n} - \gamma_{a,n} \leq \hat{x}_{b,n} + \gamma_{b,n}.$$

Thus, by Equation (34), $|\hat{x}_{a,n} - \hat{x}_{b,n}| \leq \gamma_{a,n} + \gamma_{b,n}$ (see Theorem 3.2 in Nepomuceno & Martins, 2016). Suppose that $\gamma_{a,n} = \gamma_{b,n} < \ell_{\Omega,n}$, it implies that $\gamma_{a,n} + \gamma_{b,n} < 2\ell_{\Omega,n} = |\hat{x}_{a,n} - \hat{x}_{b,n}|$, which is a contradiction. Therefore, $\gamma_{a,n} = \gamma_{b,n} \geq \ell_{\Omega,n}$. ■

Theorem 3.4 establishes that at least one of the two pseudo-orbits must have a bound error greater or equal to the lower bound error. Theorem 3.5 extends this result and states that two pseudo-orbits, which are derived from two arithmetic interval extensions, have their bound error greater than the lower bound error. This difference has a practical meaning: if this lower bound error is greater than the required precision, we must interrupt our simulation, as *both* pseudo-orbits are no longer reliable.

Table 1. Summary of the main notations for error of floating point representation and computation of a pseudo-orbit. For $n = 0$, we have $\kappa_{i,0} = \kappa$, $\gamma_{i,0} = \gamma$, $\delta_{i,0} = \delta$ and $\varepsilon_{i,0} = \varepsilon$, where the index i can be just dropped as we have only the round mode to the nearest in effect.

Definition	Floating Point	Pseudo-orbit
Absolute error	$\kappa = \text{round}(x) - x$	$\kappa_{i,n} = \hat{x}_{i,n} - x_n$
Limitant of absolute error	$\lvert\kappa\rvert \leq \gamma$	$\lvert\kappa_{i,n}\rvert \leq \gamma_{i,n}$
Relative error	$\delta = \dfrac{\text{round}(x) - x}{x}$	$\delta_{i,n} = \dfrac{\hat{x}_{i,n} - x_n}{x_n x}$
Limitant of relative error	$\lvert\delta\rvert \leq \varepsilon$	$\lvert\delta_{i,n}\rvert \leq \varepsilon_{i,n}$

3.1. Summary of notation

Notation is recapitulated in Table 1. We can notice that for $n = 0$, the floating point is the initial condition of the pseudo-orbit, that is, $\kappa_{i,0} = \kappa$, $\gamma_{i,0} = \gamma$, $\delta_{i,0} = \delta$ and $\varepsilon_{i,0} = \varepsilon$, where the index i can be dropped without any damage. The main contribution of this paper is to state that $\gamma_{a,n} = \gamma_{b,n} \geq \ell_{\Omega,n}$.

4. Numerical experiments

Numerical experiments were conducted by means of an arbitrary precision technique. We used the software Maxima to compute a pseudo-orbit with a precision much higher than common computations. We used 1000 digits in Maxima and compare our results with 16 digits precision in Scilab 5.5.2. We are interested in noticing whether the error calculated by the absolute difference between Scilab and Maxima results are equal or greater than the lower bound error. We illustrate our results with the Logistic Map and Hénon Map.

4.1. Logistic map

Let the logistic map (May, 1976) defined by

$$x_{n+1} = rx(1-x). \tag{38}$$

with $r = 3.9$ and $x_0 = 0.1$. Let two arithmetic interval extensions be:

$$G(x_n) = rx_n(1-x_n) \tag{39}$$

and

$$H(x_n) = r(x_n(1-x_n)). \tag{40}$$

We compared the first ten iterations using double precision, i.e around 16 digits of precision, with another simulation with 1000 digits of precision. In both cases, our lower bound is correct, that is, the module of the difference between the Maxima result and the two natural interval extensions $G(X_n)$ and $H(X_n)$ are equal or greater than the lower bound error. Table 2 shows these ten first values. Appendix presents the code executed by Maxima. The code line with *true* as result indicates that the error of each pseudo-orbit is greater than the lower bound. The value obtained by Maxima after the 10th iteration is given by

$$x_{10} = \frac{\begin{array}{l}53794864568829880906628838541442125245790409075273217060045554271534954655967618274950084078873860808556737363221567274417019281674291943374299653712990570974676612703889858876409243719222336812721221352769459740793165221223804914804851134430878834288393542284568758770784131518928767057745321865323776428371386366501741301084253703705450434842136646461684682240303774029013670798540312949405555537323311470868961221729324113564335632471373441172980248227865152128855073888354030438191468855321654844143696339039775077332922365706670082828739847841101068166292252346988590635995273657036146042926096491375716900588584618283532443981575950163422711852751273033769658893367785709103990604330588291836140309309836169821552331120913994104122944400565694423682481200753457281261892938631041024254174669708927956079891276424428536575600453001485033117739590719825785882461032177853519352884146130495368481611726543705995384938408688385619128058400475989050259605191972286114265363240522784304177485010464235529818810328486520740 \\ \hline 10^{1023}\end{array}$$

Since $\ell_{\Omega,10} = 2.275957200481570908 \cdot 10^{-15}$, we can see that the lower bound error is valid by proceeding with the following basic operations:

$$\begin{aligned}\kappa_{G,10} &= |x_{10} - G(X_{10})| \\ &= 1.110223024625157 \cdot 10^{-14} > \ell_{\Omega,10}\end{aligned} \tag{41}$$

and

$$\begin{aligned}\kappa_{G,10} &= |x_{10} - H(X_{10})| \\ &= 6.550315845288424 \cdot 10^{-15} > \ell_{\Omega,10}.\end{aligned} \tag{42}$$

Figure 1(a) shows the simulation of the two pseudo-orbits (39) and (40) from iteration 70–101. Figure 1(b) shows the evolution of lower bound error in logarithm scale. When $n = 83$, the lower bound error is superior than 0.1. It means that all the digits have been lost significance. From this results, it is clear that we cannot trust the simulation of Logistic Map with $r = 3.9$ and $x_0 = 0.1$ using Scilab and double precision beyond $n = 83$, which is a contrary view of works, such as Ott (2002), that suggests a transient of 500 points to build the bifurcation diagram.

4.2. Hénon map

Let the Hénon map (Hénon, 1976) be defined by

$$x_{n+1} = y_n + 1 - ax_n^2 \tag{43}$$

$$y_{n+1} = bx_n, \tag{44}$$

where $a = 1.4$, $b = 0.3$ with $x_0 = y_0 = 0.1$.

Let two arithmetic interval extensions be:

$$G(x_n) = y_n + 1 - ax_n^2 \tag{45}$$

and

$$H(x_n) = y_n + 1 - (ax_n)x_n. \tag{46}$$

Observe that in this case only associative property is applied, and Equation (44) is kept the same for both extensions. Following Lemma 3.2 it is clear that $Z_G = Z_H$, in a similar way as Equation (37).

Figure 2(a) shows the simulation of the two pseudo-orbits (45) and (46) from iteration 70–101. Figure 2(b) shows the evolution of lower bound error in logarithm scale. When $n = 92$, the lower bound error is superior than 0.1. It means that all the digits have lost significance.

To check the validity of the lower bound error, we proceed in a similar way that the logistic map. The value

Table 2. Simulation of the first ten values of the algorithm presented in Section A.1, where x_n and y_n are arithmetic interval extension extension $G(x_n)$ and $H(x_n)$, respectively, and presented in Equation (39) and (40) for the logistic map. The third column is the lower bound error of these two pseudo-orbits.

n	$G(X_n)$	$H(X_n)$	LBE ($\ell_{\Omega,n}$)
1	1.0000000000000000D-01	1.0000000000000000D-01	0.0000000000000000D+00
2	3.5100000000000342D-01	3.5100000000000342D-01	0.0000000000000000D+00
3	8.8841610000000419D-01	8.8841610000000419D-01	0.0000000000000000D+00
4	3.8661843971708196D-01	3.8661843971708751D-01	2.775557561562891351D-17
5	9.2486402497246189 56D-01	9.2486402497246200 66D-01	5.551115123125782702D-17
6	2.7101318510837718 59D-01	2.7101318510837679 73D-01	1.942890293094023946D-16
7	7.7050365056257963 39D-01	7.7050365056257907 88D-01	2.775557561562891351D-16
8	6.8962832262603945 83D-01	6.8962832262604067 96D-01	6.106226635438360972D-16
9	8.3476028710633520 81D-01	8.3476028710633343 18D-01	8.881784197001252323D-16
10	5.3794864568828770 77D-01	5.3794864568829225 96D-01	2.275957200481570908D-15

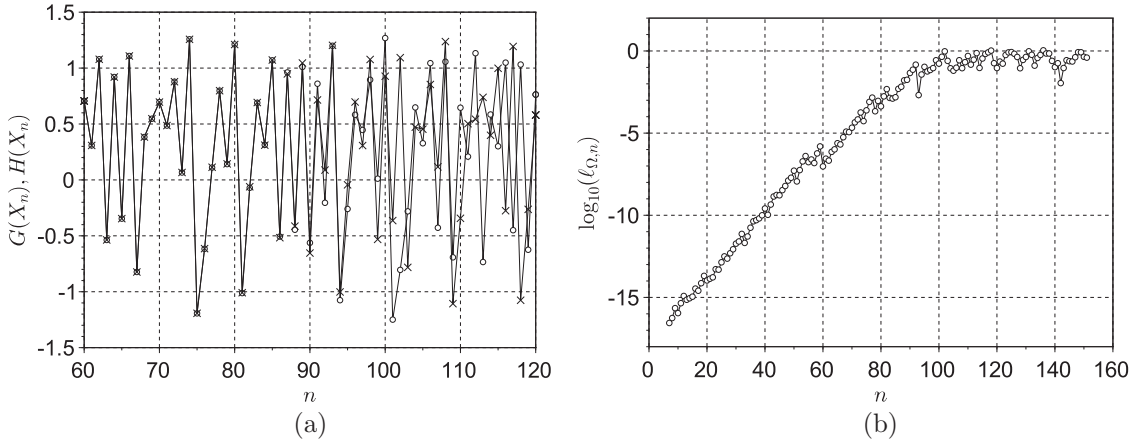

Figure 2. (a) Hénon Map Map for two pseudo-orbits and (b) evolution of lower bound error. (a) Simulation of Equations (45) and (46), with results for $G(x_n)$ $(-\times-)$ and $H(x_n)$ $(-\,\mathrm{o}\,-)$ and the same initial condition $x_0 = 0$. n stands for the number of iterations and (b) Evolution of lower bound error $\ell_{\Omega,n}$ of Equations (43) and (44). The values are plotted using $\log_{10}$. When $n \geq 92$, $\ell_{\Omega,n} > 0.1$, and thus all digits have lost significance.

obtained by Maxima after 10 iterates is given below.

$$x_{10} = -\frac{\begin{array}{l}5309907282872740139933927160476594448\\3913106737780297402886306091412947126\\9098529892917600878645273468905127265\\1484954547918653873808672953665890461\\6120212133039045534881737240250885469\\9269794390560573884800164217991883163\\6127469867650289654447644572337371770\\3623417135499471848270350256366057276\\2274446556000481099814630715310703825\\3069995224536458395653318629512059434\\35166928774884653262411063\end{array}}{\begin{array}{l}5075883674631298446548049111661087093\\6472376994021911632121206424789533957\\7859894786481485811156857245932918211\\5501791693204142684887253405561412715\\3523595568774329994713901337653176413\\3648761368190565436156142210276559442\\7495506046899009788830881007015705108\\6425781250000000000000000000000000000\\0000000000000000000000000000000000000\\0000000000000000000000000000000000000\\00000000000000000000000\end{array}}$$

Since $\ell_{\Omega,10} = 4.440892098500626162 \cdot 10^{-16}$, we can see that the lower bound error is valid by proceeding with the following operations:

$$\begin{aligned}\kappa_{G,10} &= |x_{10} - G(X_{10})| \\ &= 1.554312234475219 \cdot 10^{-15} > \ell_{\Omega,10}\end{aligned} \tag{47}$$

and

$$\begin{aligned}\kappa_{H,10} &= |x_{10} - H(X_{10})| \\ &= 6.661338147750939 \cdot 10^{-16} > \ell_{\Omega,10}.\end{aligned} \tag{48}$$

4.3. Symbolic validation

We checked our results using ten iterations of symbolic computations by means of Maxima. In both cases, when $n = 10$ the error of the used pseudo-orbits appeared greater than the lower bound error. The code is presented in Section A.3 and A.4. A close look on Table 2 and 3 indicates some values equal to zero. One of the reasons is that the difference caused by two different pseudo-orbit may not be observed in the first few iterations, as the rounding mode may produce equal float point numbers for both.

Table 3. Simulation of the first ten values of the algorithm presented in Section A.2, where x_n and y_n are arithmetic interval extensions $G(x_n)$ and $H(x_n)$ of Equations (45) and (46) for the Hénon map. The third column is the lower bound error.

n	$G(X_n)$	$H(X_n)$	LBE ($\ell_{\Omega,n}$)
1	1.0000000000000000D+00	1.0000000000000000D+00	0.0000000000000000D+00
2	-3.9999999999999112D-01	-3.9999999999999112D-01	0.0000000000000000D+00
3	1.0760000000000068D+00	1.0760000000000068D+00	0.0000000000000000D+00
4	-7.4088640000000560D-01	-7.4088640000000560D-01	0.0000000000000000D+00
5	5.5432227921305587 96D-01	5.5432227921305587 96D-01	0.0000000000000000D+00
6	3.4755161507526011 19D-01	3.4755161507526005 64D-01	2.7755575615628913 51D-17
7	9.9718770856592631 18D-01	9.9718770856592642 28D-01	5.5511151231257827 02D-17
8	-2.8787117203836976 03D-01	-2.8787117203837020 44D-01	2.2204460492503130 81D-16
9	1.1831385762027353 26D+00	1.1831385762027351 04D+00	1.1102230246251565 40D-16
10	-1.0461049983101609 05D+00	-1.0461049983101600 17D+00	4.4408920985006261 62D-16

We checked for different initial conditions and parameters: this pattern continues to appear. This point is left to future investigations.

4.4. Conservativeness of the result

Although it seems that the LBE presents a deep conservative position, that is, the LBE produces an overestimation of the error, there are some evidences that show the opposite. First, the numerical examples show the LBE quite close to a very rigorous computation of the precise error (see eqs. (41)–(47) and (42)–(48)). Besides, the fact that it has been possible to calculate the largest positive Lyapunov Exponent from the LBE in other work (see Mendes & Nepomuceno, 2016) is an evidence that computation is very close to the error propagation. As explained in Mendes and Nepomuceno (2016), starting two chaotic maps from the same initial condition, two different pseudo-orbits usually produce the same divergence of those observed due the sensitivity of initial conditions which are often observed for chaotic systems.

5. Conclusion

This paper investigates a technique to evaluate the LBE of discrete maps. We found a class of natural interval extensions that produce an equivalent bound error in recursive functions or discrete maps. This step allows us to compute the LBE for two pseudo-orbits, while previous results were applied for only one of the two pseudo-orbits. The focus of this work is on discrete maps. The relevance of the method relies on its simplicity to calculate the LBE and its multiple applications, for example in discretization schemes of differential equations, optimization techniques, polynomial NARMAX, Lyapunov Exponent computation to cite a few computational techniques.

The propagation of the error is performed by means of a simple code. The results agree in general with the literature that shows an exponential growing of the error for chaotic systems (Adler, Kneusel, & Younger, 2001). For instance, Figure 2(b) is very similar with results presented in Figure 1 in Galias (2013). Both results indicate a maximum of around 100 iterations for Hénon Map using double precision before losing the significance of simulation, that is, when the error reaches the magnitude of the variable. The comparison of the proposed method and the technique described in Galias (2013) seems an interesting direction for further research.

Much of efforts and results in the the study of nonlinear dynamics relies on the use of numerical simulations, to validate (Billings, 2013), to verify accuracy (Hammel et al., 1987) and even to make mathematical proofs of existence of strange attractors (Tucker, 1999a, 1999b). Therefore our method may be useful as an additional tool to falsify non-reliable simulations.

We verified our results for the first ten iterates using an arbitrary precision by means of the software Maxima. As mentioned by Adler et al. (2001), a mathematical package cannot be used as a black box. Accordingly we would like to stress the importance of having access to the code and full details of numerical simulations, otherwise, it may be impossible to reproduce results from only given equations.

We hope that in a near future we can investigate other classes of pseudo-orbits with similar results. Another interesting research is the identification of classes of pseudo-orbits for which the LBE grows or tends to zero. In the same line, we intend to investigate LBE methods to reduce the propagation of error in recursive functions, as well as to narrow the[1] width of interval arithmetic methods.

Note

1. Scilab is a free software which may be downloaded from www.scilab.org.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The authors are grateful for financial support of Fapemig, CNPq/INERGE and Capes.

References

Adler, C.,Kneusel, R., & Younger, W. (2001). Chaos, number theory, and computers. *Journal of Computational Physics*, *166*(1), 165–172.

Amigó, J. M., Kocarev, L., & Szczepanski, J. (2007). Discrete Lyapunov exponent and resistance to differential cryptanalysis. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *54*(10), 882–886.

Berthé, V. (2012). Numeration and discrete dynamical systems. *Computing*, *94*(2–4), 369–387.

Billings, S. A. (2013). *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. Chichester: John Wiley & Sons.

Caprani, O., & Madsen, K. (1980). Mean value forms in interval analysis. *Computing*, *25*(2), 147–154.

Dymowa, L. (2011). The methods for uncertainty modeling. In L. Dymowa (Ed.), *Soft computing in Economics and Finance* (pp. 41–105). Berlin: Springer. doi:10.1007/978-3-642-17719-4_3

Elabbasy, E., Elsadany, A., & Zhang, Y. (2014). Bifurcation analysis and chaos in a discrete reduced Lorenz system. *Applied Mathematics and Computation*, *228*, 184–194.

Feigenbaum, M. J. M. J. (1978). Quantitative universality for a class of nonlinear transformations. *Journal of Statistical Physics*, *19*(1), 25–52.

Ferreira, D. D., Nepomuceno, E. G., & Cerqueira, A. S. (2006). Nonlinear model validation using coherence. In *Proceedings of the XII Latin-American congress on automatic control* (pp. 1–5). Salvador, BA, Brazil.

Ford, J. (1986). Chaos: Solving the unsolvable, predicting the unpredictable. In M. F. Barnsley & S. G. Demko (Eds.), *Chaotic dynamics and fractals*. London: Academic Press.

Galias, Z. (2013). The dangers of rounding errors for simulations and analysis of nonlinear circuits and systems? and how to avoid them. *IEEE Circuits and Systems Magazine*, *13*(3), 35–52.

Gilmore, R., Lefranc, M., & Tufillaro, N. B. (2012). The topology of chaos: Alice in stretch and squeezeland. *American Journal of Physics*, *71*. doi:10.1119/1.1564612

Goldberg, D. (1991). What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, *23*(1), 5–48.

Hammel, S. M., Yorke, J. A., & Grebogi, C. (1987). Do numerical orbits of chaotic dynamical processes represent true orbits?. *Journal of Complexity*, *3*(2), 136–145.

Hénon, M. (1976). A two-dimensional mapping with a strange attractor. *Communications in Mathematical Physics*, *50*(1), 69–77.

Hirsch, J. E., Huberman, B. A., & Scalapino, D. J. (1982). Theory of intermittency. *Physical Review A*, *25*(1), 519–532.

Institute of Electrical and Electronics Engineers (IEEE) (2008). *754-2008 – IEEE standard for floating-point arithmetic*. IEEE.

Lambers, J. V., & Sumner, A. C. (2016). *Explorations in numerical analysis*. Hattiesburg: The University of Southern Mississippi.

Letellier, C., & Mendes, E. M. A. M. (2005). Robust discretizations versus increase of the time step for the Lorenz system. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *15*(1), 013110.

Letellier, C., Mendes, E. M. A. M., & Mickens, R. E. (2007). Non-standard discretization schemes applied to the conservative Hénon-Heiles system. *International Journal of Bifurcation and Chaos*, *17*(03), 891–902.

Lima, M. R., Claro, F., Ribeiro, W., Xavier, S., & López-Castillo, A. (2013). The numerical connection between map and its differential equation: Logistic and other systems. *International Journal of Nonlinear Sciences and Numerical Simulation*, *14*(1). doi:10.1515/ijnsns-2011-0032

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, *20*(2), 130–141.

Lozi, R. (2013). Can we trust in numerical computations of chaotic solutions of dynamical systems?. In C. Letellier & R. Gilmore (Eds.), *Topology and dynamics of Chaos: In celebration of Robert Gilmore's 70th birthday* (pp. 63–98). London: World Scientific. doi:10.1142/9789814434867_0004

May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, *261*(5560), 459–467. doi:10.1038/261459a0

Mendes, E., & Letellier, C. (2004). Displacement in the parameter space versus spurious solution of discretization with large time step. *Journal of Physics A: Mathematical and General*, *37*(4), 1203–1218.

Mendes, E. M. A. M., & Nepomuceno, E. G. (2016). A very simple method to calculate the (positive) largest Lyapunov exponent using interval extensions. *International Journal of Bifurcation and Chaos*, *26*(13), 1650226.

Moore, R. E., Kearfott, R. B., & Cloud, M. J. (2009). *Introduction to interval analysis*. Philadelphia: SIAM.

Muller, J.-M., Brisebarre, N., Dinechin, F., Jeannerod, C.-P., Lefèvre, V., Melquiond, G., & Torres, S. (2010). *Handbook of floating-point arithmetic, Vol. 7*. Boston: Birkhäuser Boston.

Nepomuceno, E. G. (2014). Convergence of recursive functions on computers. *The Journal of Engineering*, 1–3. doi:10.1049/joe.2014.0228

Nepomuceno, E. G., & Martins, S. A. M. (2016). A lower bound error for free-run simulation of the polynomial NARMAX. *Systems Science & Control Engineering*, *4*(1), 50–58.

Nepomuceno, E. G., & Mendes, E. M. A. M. (2017). On the analysis of pseudo-orbits of continuous chaotic nonlinear systems simulated using discretization schemes in a digital computer. *Chaos, Solitons & Fractals*, *95*, 21–32.

Ott, E. (2002). *Chaos in dynamical systems* (2nd ed.). Cambridge: Cambridge University Press.

Overton, M. L. (2001). *Numerical computing with IEEE floating point arithmetic*. Philadelphia: Society for Industrial and Applied Mathematics.

Peck, S. L. (2004). Simulation as experiment: A philosophical reassessment for biological modeling. *Trends in Ecology & Evolution*, *19*(10), 530–534.

Pereira, D. S. P., Kurcbart, S., & Nepomuceno, E. G. (2005). Using Scilab for nonlinear dynamic systems. In *8th International Congress of mechanical engineering - COBEM 2005* (pp. 1–8). Ouro Preto.

Rall, L. B. (1983). Mean value and Taylor forms in interval analysis. *SIAM Journal on Mathematical Analysis*, *14*(2), 223–238.

Rodríguez, M., & Barrio, R. (2012). Reducing rounding errors and achieving Brouwer's law with Taylor series method. *Applied Numerical Mathematics*, *62*(8), 1014–1024.

Stahl, V. (1997). A sufficient condition for non-overestimation in interval arithmetic. *Computing*, *59*(4), 349–363.

Teixeira, J. J., Reynolds, C. A., & Judd, K. (2007). Time step sensitivity of nonlinear atmospheric models: Numerical convergence, truncation error growth, and ensemble design. *Journal of the Atmospheric Sciences*, *64*(1), 175–189.

Tucker, W. (1999a). *The Lorenz attractor exists* (PhD thesis). Uppsala University, Sweden.

Tucker, W. (1999b). The Lorenz attractor exists. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, *328*(12), 1197–1202.

Yabuki, M., & Tsuchiya, T. (2013). Double precision computation of the logistic map depends on computational modes of the floating-point processing unit. *ArXiv e-prints*, pp. 1–10.

Appendix

Appendices present the code used in this paper. As already mentioned, it seems crucial that researchers have access to the code of a paper.

Two routines are given for each studied case. All routines are performed in Scilab 5.5.2 in a double precision. The first routine is `ex01.sci` and presents the result for Logistic Map. The second routine `ex02.sci` is devoted to Hénon Map. Observe that we present the results using `format(25)`, which exhibits all decimal digits in Scilab for double precision. In order to improve the screen exhibition, one should set the display number of columns properly.

To validate the results symbolic computation was performed by Maxima 13.04.2. The routines `ex01val.wxm` and `ex01val.wxm` are used to validate with higher precision the results obtained for the Logistic Map and Hénon Map, respectively.

A.1. ex01.sci

```
//Logistic Map - Scilab
r = 3.9;
x(1,1) = 0.1;
y(1,1) = x(1);
for k = 1:100
//G(x)
x(1,k+1) = (r*x(k))*(1-x(k));
//H(x)
y(1,k+1) = r*(y(k)*(1-y(k)));
end;
format('e',25)
[x(1:10)' y(1:10)' abs(x(1:10)'-y(1:10)')/2]

//Lower bound error
lbe=abs((x-y)/2);

//Figure 1 (a)
scf(1)
clf
plot(70:101,x(70:101),'o-k')
plot(70:101,y(70:101),'x-k')
xlabel("$n$",'fontsize',5)
ylabel("$G(X_n), H(X_n)$",'fontsize',5)
//subplot(212),plot(log10(abs(y-z)))
a=get("current_axes")//get the handle of the
                      newly created axes
a.axes_visible="on"; //makes the axes
                      visible
a.font_size=5; //set the tics label font
                size
a.box="off";
f=gcf(); f.background=-2;
xs2pdf(1,'logistic.pdf');

//Figure 1 (b)
scf(2)
plot(5:101,log10(lbe(5:101)),'o-k')
xlabel("$n$",'fontsize',5)
ylabel("$\log_{10}(\ell_{\Omega,n})$",
    'fontsize',5)
a=get("current_axes")//get the handle of
    the newly created axes
a.axes_visible="on"; //makes the axes
                      visible
a.font_size=5;//set the tics label font size
a.box="off";
//set(gca(),"data_bounds",matrix([5,2000,
    -16,4],2,-1));
f=gcf(); f.background=-2;
xs2pdf(2,'lbe.pdf');
```

A.2. ex02.sci

```
//Henon Map - Scilab
x=0;y=0;a=1.4;b=0.3;
w=x;z=y;
for k=1:150
//G(x)
x(k+1)=y(k)+1-a*x(k)^2;
y(k+1)=b*x(k);
//H(x)
w(k+1)=z(k)+1-(a*w(k))*w(k);
z(k+1)=b*w(k);
end
format('e',25)
//Table
[x(2:11) w(2:11) abs(x(2:11)-w(2:11))/2]

//Lower bound error
lbe=abs((x-w)/2);

//Figure 2 (a)
scf(1)
clf
plot(60:120,x(60:120),'o-k')
plot(60:120,w(60:120),'x-k')
xlabel("$n$",'fontsize',5)
ylabel("$G(X_n), H(X_n)$",'fontsize',5)
//subplot(212),plot(log10(abs(y-z)))
a=get("current_axes")//get the handle of the
    newly created axes
a.axes_visible="on"; //makes the axes
                      visible
a.font_size=5; //set the tics label font
    size
a.box="off";
f=gcf();
f.background=-2;
xs2pdf(1,'henon.pdf');
```

```
//Figure 2 (b)
scf(2)
clf
plot(7:151,log10(lbe(7:151)),'o-k')
xlabel("$n$",'fontsize',5)
ylabel("$\log_{10}(\ell_{\Omega,n})$",
    'fontsize',5)
a=get("current_axes")//get the handle of
    the newly created axes
a.axes_visible="on"; //makes the axes
                      visible
a.font_size=5; //set the tics label font
    size
a.box="off";
//set(gca(),"data_bounds",matrix([5,2000,
    -16,4],2,-1));
f=gcf(); f.background=-2;
xs2pdf(2,'lbehenon.pdf');
```

A.3. ex01val.wxm

```
(%i1) fpprec:1000;
(%o1) 1000
(%i2) x:1/10;
(%o2) 1/10
(%i3) for i: 1 thru 9 do x:39/10*x*(1-x);x;
(%o3) done
(%o4) 53794864568829880900662883854l4[963
    %digits]104642355298188103284865207401/
1000000000000000000000000000000[964 digits]00
    0000000000000000000000000000
(%i5) is(abs(x-5.379486456882877077D-01)
    >2.275957200481570908D-15);
(%o5) true
(%i6) is(abs(x-5.379486456882922596D-01)
    >2.275957200481570908D-15);
(%o6) true
```

A.4. ex02val.wxm

```
(%i1) fpprec:1000$
(%i2) x:0$
(%i3) y:0$
(%i4) a:14/10$
(%i5) b:3/10$
(%i6) for i: 1 thru 10 do
(aux:x,
x:y+1-a*x^2,
y:b*aux);
(%o6) done
(%i7) is(abs(x-(-1.046104998310160905D+00))
    >4.440892098500626162D-16);
(%o7) true
(%i8) is(abs(x-(-1.0461049983101
    60017D+00))>4.440892098500626162D-16);
(%o8) true
```