The Lower Bound Error as an Auxiliary Technique to Select the Integration Step-size in the Simulation of Chaotic Systems

Article · June 2018

DOI: 10.21528/LNLM-vol16-no1-art4

3 authors:

Wilson Rocha Lacerda Junior
Federal University of São João del-Rei
15 PUBLICATIONS **25** CITATIONS
SEE PROFILE

Samir A M Martins
Federal University of São João del-Rei
70 PUBLICATIONS **321** CITATIONS
SEE PROFILE

Erivelton Nepomuceno
Maynooth University
254 PUBLICATIONS **1,269** CITATIONS
SEE PROFILE

Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 16, Iss. 1, pp. 56-67, 2018

THE LOWER BOUND ERROR AS AN AUXILIARY TECHNIQUE TO SELECT THE INTEGRATION STEP-SIZE IN THE SIMULATION OF CHAOTIC SYSTEMS

Wilson Rocha Lacerda Junior
GCOM Modeling and Control Group
Federal University of São João del-Rei
São João del-Rei, Minas Gerais, Brasil
wilsonrljr@outlook.com

Samir Angelo Milani Martins
GCOM Modeling and Control Group, Department of Electrical Engineering
Federal University of São João del-Rei
São João del-Rei, Minas Gerais, Brasil
martins@ufsj.edu.br

Erivelton Geraldo Nepomuceno
GCOM Modeling and Control Group, Department of Electrical Engineering
Federal University of São João del-Rei
São João del-Rei, Minas Gerais, Brasil
nepomuceno@ufsj.edu.br

Abstract − This work presents a method to choose the integration step-size h for discretization of nonlinear and chaotic dynamic systems, in order to obtain a simulation with numerical reliability. In this context, the Lower Bound Error is used as an auxiliary technique in the search for the optimal value of h, considering the Fourth Order Runge Kutta as the discretization method. The Lorenz equations, Rössler equations and Duffing-Ueda oscillator were used as case studies. This work, besides investigating the most adequate step-size h for each case, shows that the choice of very small values of h results in significantly inferior solutions, despite the consensus that the smaller the step-size, the higher the accuracy.

Keywords − dynamical system, discrete time systems, chaos, numerical simulation, lower bound error.

1. INTRODUCTION

Numerical calculation assumes an essential part in examining the solutions of nonlinear dynamic systems [1–4]. Since the work of Lorenz in 1963 [5], numerical experiments have been used in order to understand complex nonlinear dynamical systems that exhibit chaotic behaviour [6]. However, due to inherent properties of the digital computers the results found by numerical simulations are, in many cases, imprecise [1]. Moreover, [3] expresses that there are many published papers wherein the reliability of numerical outcomes is not cautiously verified. In addition, the same author say that "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 ? ".

One of the causes of computational errors is related to the rounding method used by the machines. One of the first papers to deal with the problem of rounding off errors for the linear case in a digital computer is [7], where the author studies the effects of round-off in the floating point realization of a general linear discrete filter governed by a stable difference equation. Besides this, there are also recent works aiming to understand and minimize the round-off error [8–12]. To investigate the round-off errors in recursive functions, [13] introduced a technique to evaluate the lower bound error based on the on the fact that the interval extensions [14] are mathematically equivalent but may exhibit different computer simulation results.

There are other sources of computation errors when analyzing nonlinear systems, such as errors introduced during integration of continuous time systems by using numerical methods, which are constructed by skipping higher-order terms in the Taylor expansion of the solution [1]. In order to obtain results for numerical problems, researchers often use discretization methods available in popular tools on accessible computers. In that regard, [10] reported the existence of multiple pseudo-orbits for nonlinear dynamics systems when discretization schemes are used in a set-up where the step-size is not varied and the initial conditions are kept unchanged.

Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 16, Iss. 1, pp. 56-67, 2018

In this context, one of the most used discretization methods in the simulation of nonlinear dynamic systems is the fourth order Runge Kutta [15–17]. Regarding this method, the choice of the integration step-size h is fundamental to the accuracy and stability of the model to be simulated [18–20]. Therefore, with the knowledge that chaotic dynamical systems are sensitive to the initial conditions, any error in a single step of numerical integration is amplified by the inherent divergence of trajectories. Hence, it is unreasonable to expect that a numerical method will accurately track the trajectory of its initial condition for long periods of time, since trajectories of nearby initial conditions do not remain close to the chosen one [21]. In this sense, in order to obtain numerical solutions with high precision, a common procedure is to decrease h to very small values [22]. However, given the recursivity of the process, the canceling result [23] and the propagation of the error due to this recursivity, this can result in even worse solutions. Thus, this work shows an alternative method to choose the most suitable step-size h for simulation and analysis of nonlinear dynamic systems, which uses the Lower Bound Error as an auxiliary technique to evaluate which h guarantees the highest numerical reliability in the simulation, considering the error propagation in a iterative process simulated in a computer.

The rest of the paper is laid out as follows: In Section 2, the preliminary concepts are briefly reviewed. The proposed method based on the lower bound error is presented Section 3. The results as well as the discussion are presented in Section 4, while concluding remarks and perspectives for future research are shown in Section 5.

2. Definitions

Definition 2.1 *(Interval extension- R. E. Moore. [14]) Let f be a function of real variable x. An interval extension of f is an interval valued function F of an interval variable x, with the property*

$$F(x) \quad = \quad f(x) \; for \; real \; arguments,$$

where an interval means a closed set of real numbers $x \in \mathbb{R}$ such that $X = [\underline{X}, \overline{X}] = \{x : \underline{X} \leq x \leq \overline{X}\}$.

To illustrate Definition 2.1, consider the example in the sequel.

Example 2.1 *Let $x_{n+1} = f(x_n) = rx_n(1 - x_n)$. A few examples of interval extension of f are:*

$$\begin{aligned} X_{n+1} = F(X_n) &= r(X_n(1 - X_n)), & (1)\\ &= rX_n - rX_n^2, & (2)\\ &= r(X_n - X_n^2), & (3)\\ &= rX_n - (rX_n)X_n. & (4) \end{aligned}$$

In Example 2.1, Equations 1-4 are mathematically equivalent, but they have different sequences of its basic arithmetic operations [24, 25]. Floating point standard does not have commutative and distributive operations, so they may exhibit different pseudo-orbits [23, 26, 27].

Definition 2.2 *G and H are equivalent interval extensions if*

$$G(X) = H(X) \; for \; interval \; arguments.$$

Consider the following example of Definition 2.2.

Example 2.2 *Let the following extension intervals:*

$$\begin{aligned} G(X) &= rX(1 - X)), \\ H(X) &= r(X(1 - X)), \\ L(X) &= rX - rX^2. \end{aligned}$$

Considering $r = 5$ e $X = [0.4, 0.5]$:

$$\begin{aligned} G([0.4, 0.5]) &= 5[0.4, 0.5](1 - [0.4, 0.5])) &= [1, 1.5], \\ H([0.4, 0.5]) &= 5([0.4, 0.5](1 - [0.4, 0.5])) &= [1, 1.5], \\ L([0.4, 0.5]) &= 5[0.4, 0.5] - 5([0.4, 0.5])^2 &= [0.75, 1.7]. \end{aligned}$$

In this example, only $G(X)$ and $H(X)$ are equivalent interval extensions.

Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 16, Iss. 1, pp. 56-67, 2018

Definition 2.3 *Let $n \in \mathbb{N}$, a metric space $M \subset \mathbb{R}$, the relation*

$$x_{(n+1)} = f(x_n), \tag{5}$$

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 xn obtained by iterating Equation 5 starting from an initial condition x_0 is called the orbit of x_0 [28].

Definition 2.4 *An orbit is a sequence of values of a map, represented by $\{x_n\} = [x_0, x_1, \ldots, x_n]$.*

Definition 2.5 *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 discretization scheme. A pseudo-orbit is an approximation of an orbit and is represented as*

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

such that

$$|x_n - \hat{x}_{i,n}| \leq \delta_{i,n}, \tag{6}$$

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

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

$$I_{i,n} = [\hat{x}_{i,n} - \delta_{i,n}, \hat{x}_{i,n} + \delta_{i,n}]. \tag{7}$$

From Equations 6 and 7, it is easy to realize that:

$$x_n \in I_{i,n}, \forall i \in \mathbb{N}. \tag{8}$$

There is no unique pseudo-orbit, as there are different hardware, software, numerical precision standard and discretization schemes, which may yield different output for each extension interval.

Definition 2.6 *(Lower Bound Error, [13]). Let $\hat{x}_{a,n}$ and $\hat{x}_{b,n}$ two pseudo-orbits derived from two interval extensions. Let $\delta_{\alpha,n} = |\hat{x}_{a,n} - \hat{x}_{b,n}|/2$ be the lower bound error of a map $f(x)$, then $\delta_{a,n} \geq \delta_{\alpha,n}$ or $\delta_{b,n} \geq \delta_{\alpha,n}$.*

The proof of Definition 2.6 can be obtained directly from [13] and will not be reproduced here. Definition 2.6 establishes that as a minimum one of the two pseudo-orbits ought to have an error greater or equal to the lower-bound error. If this lower-bound error is greater than the required precision, the simulation ought to no longer be carried on without in addition evaluation.

3. Method

This section presents the methodology applied to obtain the results of this work. The criteria used to choose the models, the discretization scheme, the integration steps and the stopping criterion, as well as the algorithms used in the work are established.

3.1 Case studies

As case studies, three models systems were chosen, which exhibit nonlinear dynamics, such as chaos. The chosen models are for the Lorenz equations, the Rössler equations and the forced Duffing Ueda oscillator, all of them extensively studied in the literature. For each model, the purpose is to use two equivalent interval extensions to produce two different pseudo-orbits.

3.2 Discretization scheme and step-sizes

The Fourth Order Runge Kutta (RK4) is defined as the discretization scheme to be used in order to simulate the models in this work. This choice is due the fact that RK4 is a well known discretization scheme and widely used by the researchers. Function 1 describes the algorithm used to solve the proposed systems of differential equations.

The step-size h are adjusted from $h = 0.09 : -0.01 : 0.003$, i.e. the values of h are increased in intervals of 0.01 from 0.003 to 0.09. Hence, this range cover values considered high to very small values of h, often used in temporal discretization to obtain solutions of ordinary differential equations. Due to the technical limitations of the computer

Function Fourth Order Runge Kutta(RK4)()

```
Result: yout = y0 + (dt/6)*(f1+2*f2+2*f3+f4)
initialization;
define initial conditions and integration step-size;
function yout = rk4(fun,dt,t0,y0);
/*compute f1, f2, f3 and f4;
f1 = fun(t0,y0);
f2 = fun(t0+dt/2,y0+(dt/2)*f1);
f3 = fun(t0+dt/2,y0+(dt/2)*f2);
f4 = fun(t0+dt,y0+dt*f3);
yout = y0 + (dt/6)*(f1+2*f2+2*f3+f4);
```

available for the study, solutions of the proposed models for $h < 0.003$ become unfeasible, since the process becomes computationally complex as a result of the number of iterations required.

To avoid any influence concerning different software, operating systems and hardware in the simulation of the models, all routines are performed in a computer with a processor I5 5200 @ 2.20 GHz and Windows 10 Build OS 14393.693 operation system. To produce the result of this paper, all routines were performed in Matlab, version R2016a.

Procedure 1 shows how the parameters are defined in order to obtain the results for each interval extension.

Procedure Parameters definition and solution of the systems equations

```
for g = 0.09 : −0.001 : 0.003 do
    define the parameters of the system;
    Let y0 the initial conditions;
    Let dt the integration step-size ;
    dt ← g;
    define the simulation interval int ;
    int=[0:dt:final];
    yin = y0;
    for i = 1 to length (int) do
        time = i*dt;
        /*firstly choose one interval extension, then execute this. Later, run the routine again with other interval
          extension. */ ;
        youtput = rk4((t,y)system_eq(t,y,parameters),dt,time,yin);
        Y = [Y youtput];
        yin = youtput;
    end
    data=Y ;
    save the data for all values of g ;
    clear all data from work space ;
end
```

3.3 Stop simulation criterion

To stop the simulation when a required numerical precision is no longer satisfied, a criterion based on the relative error of a sequence of values is established. Let $\varepsilon_{\alpha,n}$ be the relative precision at iteration n:

$$\varepsilon_{\alpha,n} = \frac{|\hat{x}_{a,n} - \hat{x}_{b,n}|}{|\hat{x}_{a,n} + \hat{x}_{b,n}|} \tag{9}$$

where $n \in \mathbb{N}$. In this work, $\varepsilon = 0.001$. Thus, the integration step-size h that guarantees the highest numerical reliability considering the chosen stop criterion may be verified. Procedure 2 show how this can be easily implemented.

It is worth emphasizing that any other chaotic system can be used to illustrate the main ideas of this paper. Therefore the main point is not show that the proposed method can be applied to the three case studies, but to any system defined by a set of differential equations and more than that to any discretization scheme with fixed step-size. Furthermore, this method can also be applied in a scenario related to the simulation errors due to the discretization schemes, complementing the results achieved by [10].

Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 16, Iss. 1, pp. 56-67, 2018

Procedure Evolution of relative error and choice of step-size

```
n=1;
time_max=0;
for g = 0.09 : −0.001 : 0.003 do
    system_eq1=load data of the first interval extension, given value of g;
    system_eq2=load data of the second interval extension, given value of g;
    Let er = 0.001 the stop criteria ;
    for k=1:length(system_eq1) do
        ere = |system_eq1 − system_eq2| / |system_eq1 + system_eq2|
    end
    maxm = iteration of simulation within the established criteria;
    time_max(n)= g × maxm ;
    if time_max(n) > time_max(n − 1) then
        return h_selected = time_max(n) / maxm(n).
    else
        return h_selected = time_max(n−1) / maxm(n−1).
    end
    Plot the evolution of relative error;
end
```

4. Results

In this section, the results of the three case studies are presented. Two different set of parameters and initial conditions are used for each case study. The results show which are the integration step-sizes that guarantee, considering the proposed method, the greatest numerical reliability in the simulations using interval extensions and the lower error limit as an auxiliary technique for this purpose.

4.1 Lorenz equations

Consider the Lorenz equations [5] defined by the following set of differential equations:

$$\begin{aligned}\frac{dx_l}{dt} &= \underline{\sigma(y_l - x_l)}\\ \frac{dy_l}{dt} &= \underbrace{x_l(\rho - z_l) - y_l}\\ \frac{dz_l}{dt} &= x_l y_l - \beta z_l\end{aligned} \tag{10}$$

Consider now one equivalent interval extensions of Equation 10 given by:

$$\begin{aligned}\frac{dx_{l1}}{dt} &= \underline{\sigma y_{l1} - \sigma x_{l1}}\\ \frac{dy_{l1}}{dt} &= \underbrace{x_{l1}\rho - x_{l1}z_{l1} - y_{l1}}\\ \frac{dz_{l1}}{dt} &= x_{l1}y_{l1} - \beta z_{l1}.\end{aligned} \tag{11}$$

Note that Equations 10 and 11 are mathematical equivalent. However, they are written slightly different, as indicated by the underline terms. Consider, then, the solutions of Lorenz Equations with parameters and initial conditions shown in Table 1. The step-sizes h are kept the same for both interval extensions. Adjusting the step-size with a factor $D = 0.001$ between values $h = 0.09$ and $h = 0.003$, a simulation time that satisfies $\varepsilon = 0.001$ was obtained for each value of h. Hence, the h that guarantees the highest numeric reliability in the simulation can be chosen.

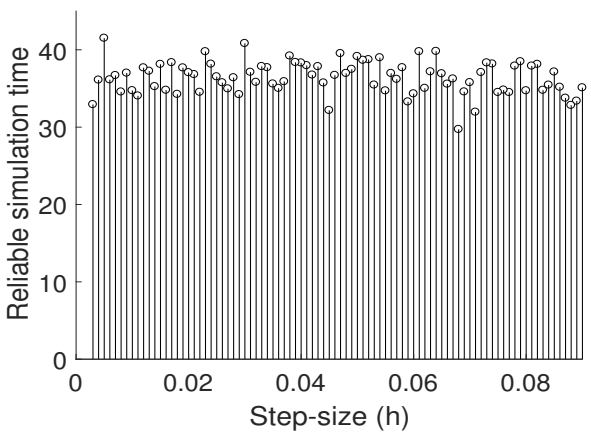
Figure 1 shows the results of each simulation in LTU (Lorenz time units) considering the proposed step-sizes of h. The values obtained are related to the variables x_l and x_{l1} of the Lorenz equations. In the first case, the most appropriate step-size is $h = 0.005$, which guaranteed 41.51 LTU of simulation within the established criteria. Moreover, as the integration step-sizes decreases, the reliable simulation time varies until reaching a maximum value, from which the decrease of h reduces the reliable simulation time. Given that $h = 0.005$ is the most adequate size-step, the lowest value of h used ($h = 0.003$) shows a significant worsening of 20.66% when compared to it, considering the simulation

Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 16, Iss. 1, pp. 56-67, 2018

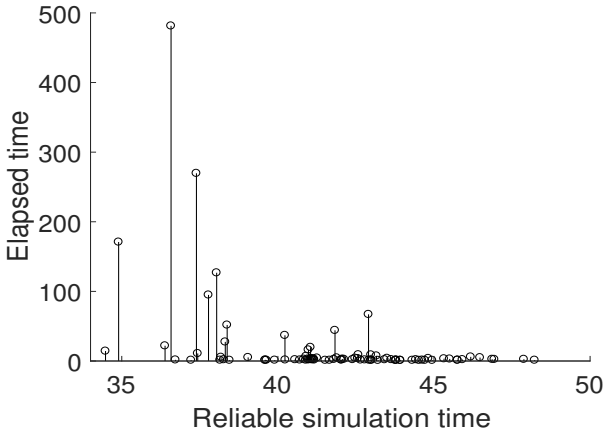
Table 1: Lorenz system parameters and initial conditions.

Parameters	rho	beta	Initial Contidions		
sigma	rho	beta	x_0	y_0	z_0
First case	10	28	8/3	0	0	0
Second case	8	24.08	7/3	1	0.1	2

time that satisfies $\varepsilon = 0.001$. In the second case, the most appropriate step-size is $h = 0.075$, which guaranteed 48.23 LTU of simulation within the established criteria. When compared to $h = 0.003$, $h = 0.075$ shows a superiority of 24.15%. In addition, Figure 1b shows the reliable simulation time given the elapsed time of each simulation. In this sense, it is possible to establish a balance between quality of results and computational performance, allowing the choice of the best step-size to be more versatile.



(a) Accomplished time of reliable simulation given each value of h. Parameters: $sigma = 10$, $rho = 28$ and $beta = 8/3$



(b) Reliable simulation time related to elapsed time given each value of h. Parameters: $sigma = 10$, $rho = 28$ and $beta = 8/3$

Figure 1: Lorenz system - Accomplished time of reliable simulation given each value of h and its relation with elapsed time simulation.

Figures 2a shows a specific window for the performed simulations, which uses the criterion presented in Section 3, for $h = 0.005$ and $h = 0.075$, respectively. It is worth emphasizing that no modifications are made to the performed routines and, despite the mathematical equivalence, the simulation results in two different pseudo-orbits in both cases. Figures 2b shows the evolution of relative error between the two pseudo-orbits x_l and x_{l1} given $h = 0.005$, which values are plotted using Logarithmic scale. As seen, the error grows exponentially and quickly becomes large enough to cause very different outputs.

4.2 Rössler equations

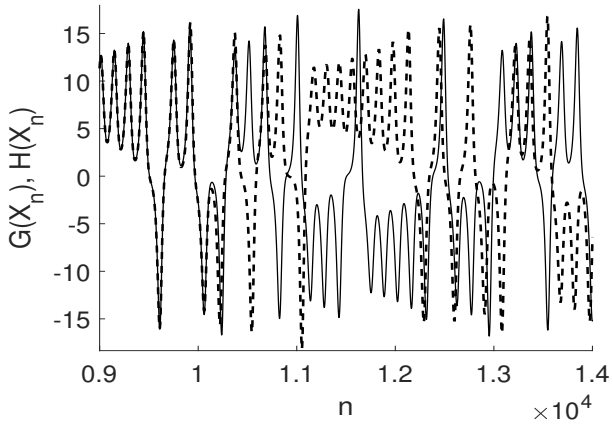
Consider the Rössler equations [29] defined by the following set of differential equations:

$$\begin{aligned} \frac{dx_r}{dt} &= \underline{-(y_r + z_r)} \\ \frac{dy_r}{dt} &= x_r + \alpha * y_r \\ \frac{dz_r}{dt} &= \underbrace{\beta + z_r x_r - z_r \zeta} \end{aligned} \tag{12}$$

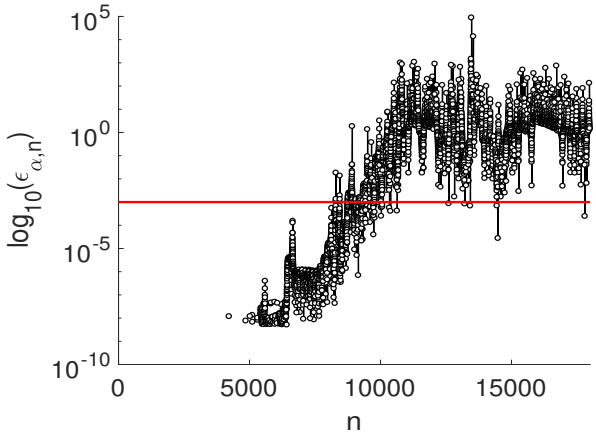
Consider now one equivalent interval extensions of Equation 12 given by:

$$\begin{aligned} \frac{dx_{r1}}{dt} &= \underline{-y_{r1} - z_{r1}} \\ \frac{dy_{r1}}{dt} &= x_{r1} + \alpha * y_{r1} \\ \frac{dz_{r1}}{dt} &= \underbrace{\beta + z_{r1}(x_{r1} - \zeta)} \end{aligned} \tag{13}$$

In a similar way to what was done in Subsection 4.1, Equations 12 and 13 are mathematical equivalent. Consider the solutions of the Rössler equations given the parameters shown in Table 2. The step-sizes h for both interval extensions



(a) Free-run simulations of Equations 10 and 11. x_l (-) and x_{l1} (–) were obtained with same initial conditions ($(x_0, y_0, z_0) = (1, 1, 1)$), parameters ($\sigma = 10$, $\rho = 28$ and $\beta = 8/3$) and step-size ($h = 0.005$). n stands for the number of iterations.



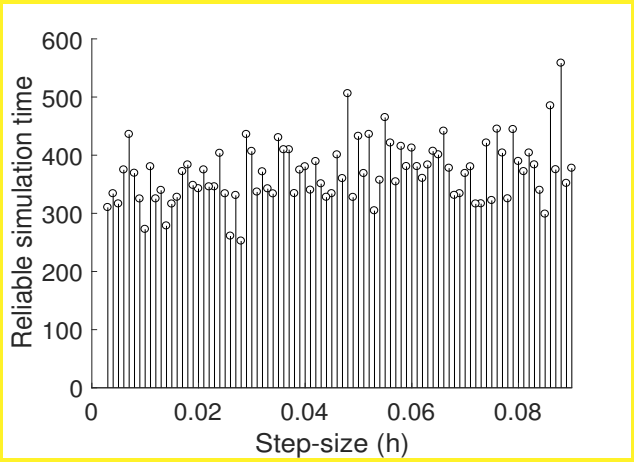
(b) Evolution of relative error $\delta_{\alpha,n}$ of x_l and x_{l1}, as seen in Figure 2a given $\varepsilon = 0.001$ as adopted precision.

Figure 2: Free-run simulations of Equations 10 and 11 and its respective evolution of relative error $\delta_{\alpha,n}$

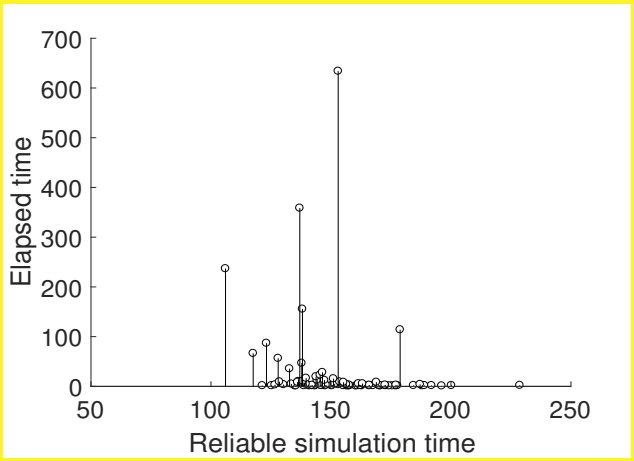
Table 2: Rössler system parameters and initial conditions.

	Parameters	rho	beta	Initial Contidions		
	α	β	ζ	x_0	y_0	z_0
First case	0.2	0.2	5	0	0	0
Second case	0.38	0.3	4.82	1	−10	1

are kept unchanged. Adjusting the step-size with a factor $D = 0.001$ between values $h = 0.09$ e $h = 0.003$, considering the first case, the most appropriate step-size is $h = 0.088$, shown in Figure 3a. $h < 0.088$ result in a lower reliability in the obtained data, where $h = 0.003$ shows a decrease of 44.45 % when compared to $h = 0.088$, which reached 558.4 Rössler time units. In the second case, the most appropriate step-size is $h = 0.049$, which guaranteed 228.7 Rössler time units of simulation within the established criteria. When compared to $h = 0.003$, $h = 0.049$ shows a superiority of 33.10%. In addition, Figure 3b and shows the reliable simulation time given the elapsed time of each simulation. Figures 4 and shows the result of simulations for x_r e x_{r1} and the evolution of relative error.



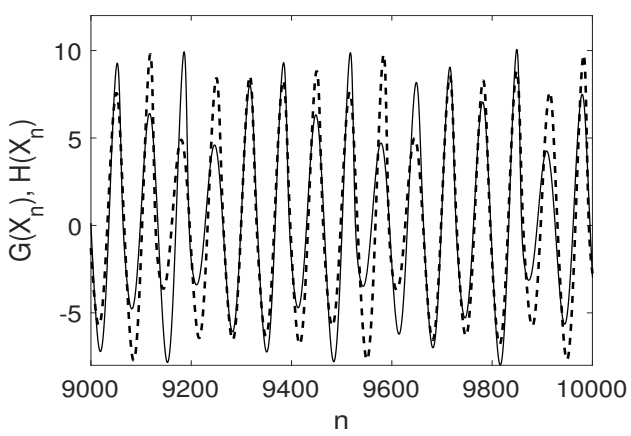
(a) Accomplished time of reliable simulation given each value of h. Parameters: $\alpha = 0.2$, $\beta = 0.2$ and $\zeta = 5$



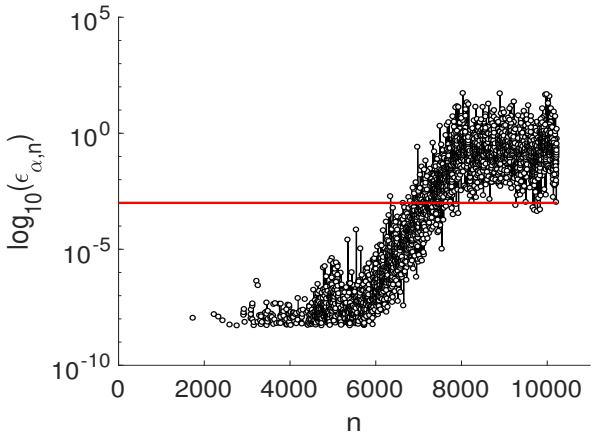
(b) Reliable simulation time related to elapsed time given each value of h. Parameters: $\alpha = 0.2$, $\beta = 0.2$ and $\zeta = 5$

Figure 3: Rössler system -Accomplished time of reliable simulation given each value of h and its relation with elapsed time simulation.

Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 16, Iss. 1, pp. 56-67, 2018



(a) Free-run simulations of Equations 12 and 13. x_r (-) and x_{r1} (–) were obtained with the same initial conditions ($(x_0, y_0, z_0) = (0, 0, 0)$), parameters ($\alpha = 0.2$, $\beta = 0.2$ and $\zeta = 5$) and step-size ($h = 0.088$).



(b) Evolution of relative error $\delta_{\alpha,n}$ of x_r and x_{r1}, as seen in Figure 4a, given $\varepsilon = 0.001$ as adopted precision.

Figure 4: Free-run simulations of Equations 12 and 13 and its respective evolution of relative error $\delta_{\alpha,n}$

4.3 Duffing-Ueda oscillator

Consider a periodically forced nonlinear Duffing-Ueda oscillator [30, 31]:

$$\frac{d^2y}{dt^2} + k\frac{dy}{dt} + \mu y^3 = Acos(t). \tag{14}$$

Equation 14 can be written as an autonomous system of two equations:

$$\begin{aligned} \frac{dx_d}{dt} &= y_d \\ \frac{dy_d}{dt} &= \delta cos(t) - \lambda y_d - \underline{x_d^3} \end{aligned} \tag{15}$$

Consider now one interval extension of Equation 15 written as:

$$\begin{aligned} \frac{dx_{d1}}{dt} &= y_{d1} \\ \frac{dy_{d1}}{dt} &= \delta cos(t) - \lambda y_{d1} - \underline{x_{d1} * x_{d1} * x_{d1}} \end{aligned} \tag{16}$$

Following the same procedures described in Subsections 3, 4.1 and 4.2, the most appropriate step-size for solving Duffing-Ueda oscillator, considering the first case, is $h = 0.059$, as shown in Figure 5a. This result was obtained solving the set of Equations 15 and 16 given the parameters presented in Table 3. $h < 0.059$ decrease the reliability in the obtained data, in accordance with the results obtained with the Lorenz and Rössler Equations. $h = 0.003$ shows a expressive decrease of 47.13% when compared to $h = 0.059$, which reached 335.1 Duffing time units of reliable simulation, given the proposed criteria.

The second case clearly shows the importance of the careful choice of the step-size in the simulation of chaotic systems, since the proposed method allowed the choice of an step-size that ensured that the simulation occurred within the established criteria, as shown by Figure 6c and 6d. In the second case, given that $h = 0.087$ is the most adequate size-step, the lowest value of h used ($h = 0.003$) shows a expressive worsening of 84.84% when compared to it, considering the simulation time that satisfies $\varepsilon = 0.001$. Moreover, it is important to observe the disparity regarding the computational cost involved, since the best solution obtained also presents by far computationally superior performance. In addition, Figure 5b and 5d shows the reliable simulation time given the elapsed time of each simulation.

All results presented are summarized in Table 4.

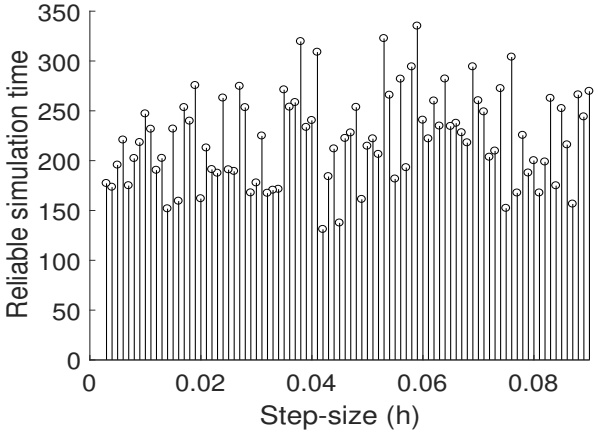
5 Conclusions

This work presented a method to choose the appropriate integration step-size h for a given range, which guarantees rigorous numerical reliability in the simulation of nonlinear and chaotic dynamic systems. In this context, the Lower Bound Error was used as an auxiliary technique in the choice of such value.

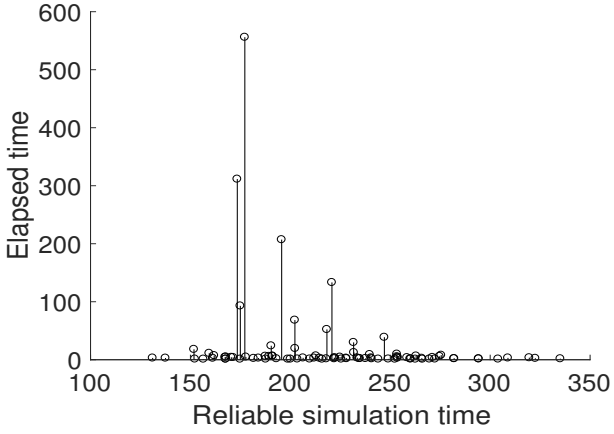
Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 16, Iss. 1, pp. 56-67, 2018

Table 3: Duffing Oscillator system parameters and initial conditions.

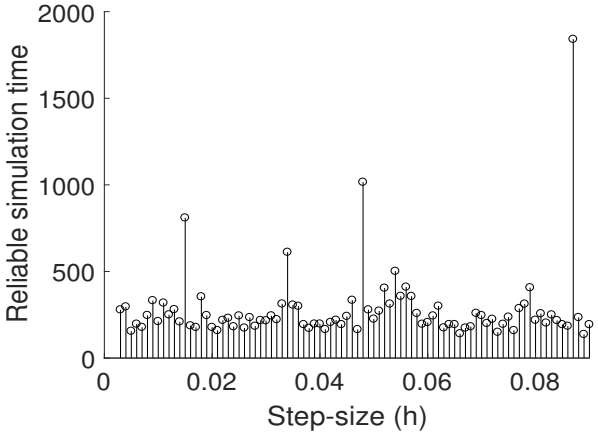
	Parameters		Initial Contidions	
	δ	λ	x_0	y_0
First case	7.5	0.05	3	4
Second case	8	0.01	1	0



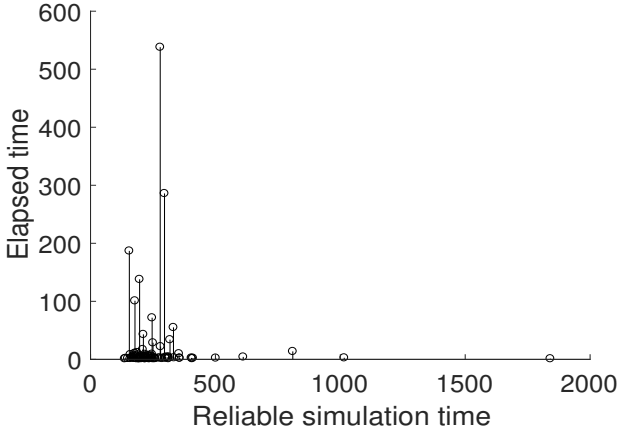
(a) Accomplished time of reliable simulation given each value of h. Parameters: $\delta = 7.5$ and $\lambda = 0.05$



(b) Reliable simulation time related to elapsed time given each value of h. Parameters: $\delta = 7.5$ and $\lambda = 0.05$



(c) Accomplished time of reliable simulation given each value of h. Parameters: $\delta = 8.0$ and $\lambda = 0.01$



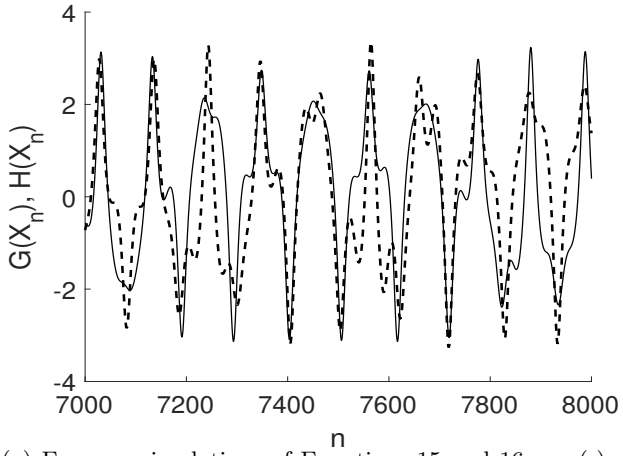
(d) Reliable simualtion time related to elapsed time given each value of h. Parameters: $\delta = 8.0$ and $\lambda = 0.01$

Figure 5: Accomplished time of reliable simulation given each value of h and its relation with elapsed time simulation.

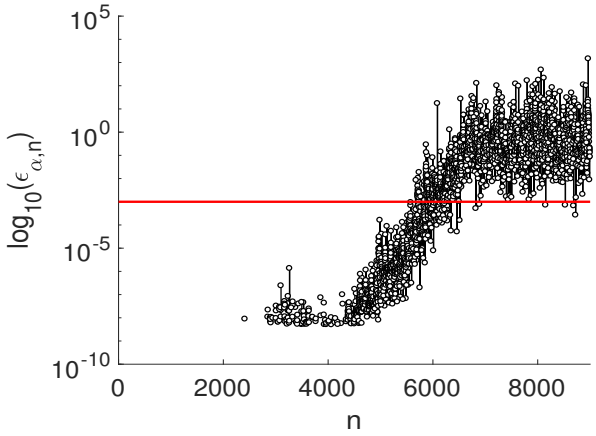
Three case studies were presented: the analysis of the Lorenz equations, Rössler equations and a periodically forced nonlinear Duffing-Ueda oscillator. For each case, two set of different parameters and initial conditions were considered. In all cases a similar behavior can be seen by adjusting the values of h, so that, as the value of h decreases, the reliability in the simulation increases to a optimal value. However, by decreasing the value of h beyond this optimal value, the solutions presented a significant loss in the simulation reliability of the analyzed models, contradicting what is pointed-out in specialized literature. This is a consequence on how the error propagates when simulating continuous nonlinear dynamic systems, as the ways computers store numbers lead to limited precision. As the integration step decreases, more computations are required to obtain the system outcome, and consequently, the error propagation increase during the simulation.

Hence, were established the most adequate values of h for each analyzed system, guaranteeing a longer simulation time that satisfies the predefined precision. Thus, it can be seen that very small values of h do not guarantee a greater precision in the simulation of non-linear dynamic systems, as would be expected. In addition, choosing h using this method avoids the choice of very small values that can have, in addition to the problems presented in this study, a highest computational cost than what is really necessary.

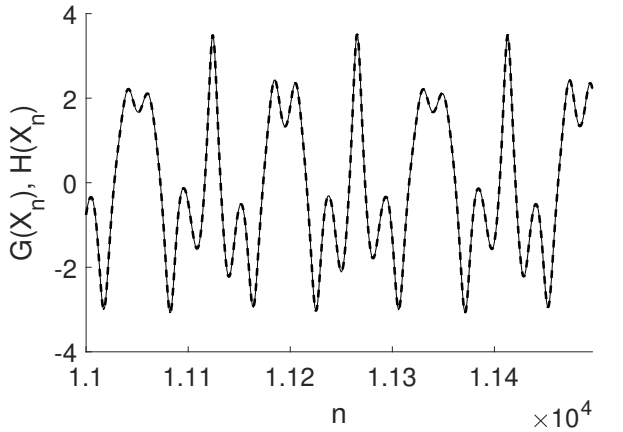
Future works intend to apply this method using others discretization schemes, in order to verify the influence of each method on the reliability of the simulations of complex and chaotic systems. Moreover, forthcoming studies pretend to find better ways to, computationally, simulate the model, and not only to detect failures in the simulation.



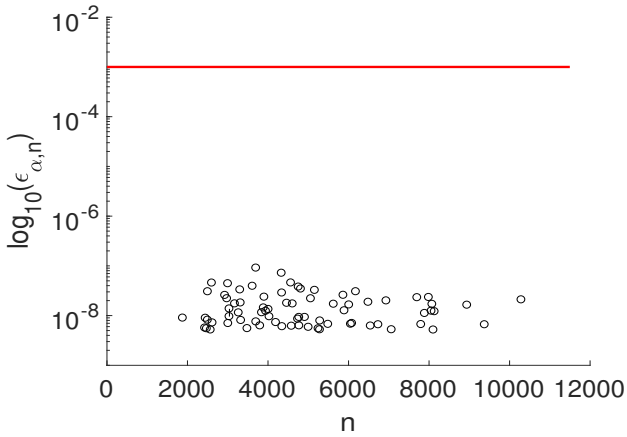
(a) Free-run simulations of Equations 15 and 16. x_r (-) and x_{r1} (–) were obtained with the same initial conditions. Parameters ($\delta = 7.5$, $\lambda = 0.05$) and step-size ($h = 0.059$).



(b) Evolution of relative error $\delta_{\alpha,n}$ of x_r and x_{r1}, as seen in Figure 6a, given $\varepsilon = 0.001$ as adopted precision.



(c) Free-run simulations of Equations 15 and 16. x_r (-) and x_{r1} (–) were obtained with the same initial conditions. Parameters ($\delta = 8$, $\lambda = 0.01$) and step-size ($h = 0.087$).



(d) Evolution of relative error $\delta_{\alpha,n}$ of x_l and x_{l1}, as seen in Figure 6c given $\varepsilon = 0.001$ as adopted precision.

Figure 6: Free-run simulations of Equations 15 and 16 and its respective evolution of relative error $\delta_{\alpha,n}$

6. Acknowledgments

This work has been supported by CNPq/INERGE, Fapemig and Capes. The authors also would like to thank professor Márcio Júnior Lacerda for his valuable comments about the text and method.

References

[1] Z. Galias. "The dangers of rounding errors for simulations and analysis of nonlinear circuits and systems? and how to avoid them". *IEEE Circuits and Systems Magazine*, vol. 13, no. 3, pp. 35–52, 2013.

[2] S. M. Hammel, J. A. Yorke and C. Grebogi. "Do numerical orbits of chaotic dynamical processes represent true orbits?" *Journal of Complexity*, vol. 3, no. 2, pp. 136–145, 1987.

[3] R. Lozi. *Can we trust in numerical computations of chaotic solutions of dynamical systems*, volume 84. World Scientific Series in Nonlinear Science Series A, 2013.

[4] T. Sauer, C. Grebogi and J. A. Yorke. "How long do numerical chaotic solutions remain valid?" *Physical Review Letters*, vol. 79, no. 1, pp. 59, 1997.

[5] E. N. Lorenz. "Deterministic nonperiodic flow". *Journal of the atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[6] E. Ott. *Chaos in dynamical systems*. Cambridge university press, 2002.

Table 4: Most adequate step-sizes for each model.

	Case studies	Most adequate step-size	Improvement in relation to $h = 0.003$ (%)
First case	Lorenz equations	0.005	20.66
Second case		0.075	24.15
First case	Rössler equations	0.088	44.45
Second case		0.049	33.10
First case	Duffing oscillator	0.059	47.13
Second Case		0.087	84.84

[7] I. Sandberg. "Floating-Point-Roundoff Accumulation in Digital-Filter Realizations". *Bell system technical journal*, vol. 46, no. 8, pp. 1775–1791, 1967.

[8] M. Antoñana, J. Makazaga and A. Murua. "Reducing and monitoring round-off error propagation for symplectic implicit Runge-Kutta schemes". *Numerical Algorithms*, pp. 1–20, 2017.

[9] R. Lozi, V. A. Pogonin and A. N. Pchelintsev. "A new accurate numerical method of approximation of chaotic solutions of dynamical model equations with quadratic nonlinearities". *Chaos, Solitons & Fractals*, vol. 91, pp. 108–114, 2016.

[10] E. G. Nepomuceno and 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*, vol. 95, pp. 21–32, 2017.

[11] K. J Hanley and C. O'Sullivan. "Analytical study of the accuracy of discrete element simulations". *International Journal for Numerical Methods in Engineering*, vol. 109, no. 1, pp. 29–51, 2017.

[12] X. Li and S. Liao. "The influence of numerical noises on statistics computation of chaotic dynamic systems". *arXiv preprint arXiv:1609.09354*, 2016.

[13] E. Nepomuceno and S. Martins. "A lower bound error for free-run simulation of the polynomial NARMAX". *Systems Science & Control Engineering*, vol. 4, no. 1, pp. 50–58, 2016.

[14] R. E. Moore. *Methods and applications of interval analysis*. SIAM, 1979.

[15] J. Guckenheimer. "Numerical analysis of dynamical systems". *Handbook of dynamical systems*, vol. 2, pp. 345–390, 2002.

[16] E. Hairer, C. Lubich and M. Roche. "The numerical solution of differential-algebraic systems by Runge-Kutta methods", 2006.

[17] E. Hairer, S. Nørsett and G. Wanner. "Solving Ordinary Differential Equations I", 1987.

[18] H. Gavin. "Numerical integration for structural dynamics". *Department of Civil and Environmental Engineering, Duke University, Durham, NC*, 2001.

[19] J. C. Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

[20] J.-W. Hu and H.-M. Tang. "Numerical methods for differential equations". *City University, Hong Kong*, 2003.

[21] B. Fiedler. *Handbook of dynamical systems*, volume 2. Gulf Professional Publishing, 2002.

[22] R. Hamming. *Numerical methods for scientists and engineers*. Courier Corporation, 2012.

[23] M. L. Overton. *Numerical computing with IEEE floating point arithmetic*. SIAM, 2001.

[24] M. Rudolph-Lilith and L. E. Muller. "On a representation of the Verhulst logistic map". *Discrete Mathematics*, vol. 324, pp. 19–27, 2014.

[25] M. Yabuki and T. Tsuchiya. "Double precision computation of the logistic map depends on computational modes of the floating-point processing unit". *arXiv preprint arXiv:1305.3128*, 2013.

[26] D. Goldberg. "What every computer scientist should know about floating-point arithmetic". *ACM Computing Surveys (CSUR)*, vol. 23, no. 1, pp. 5–48, 1991.

[27] I. of Electrical and E. E. (IEEE). "IEEE standard for floating-point arithmetic". pp. 1–58, 2008.

[28] R. Gilmore and M. Lefranc. *The topology of chaos: Alice in stretch and squeezeland*. John Wiley & Sons, 2012.

[29] O. E. Rössler. "An equation for continuous chaos". *Physics Letters A*, vol. 57, no. 5, pp. 397–398, 1976.

[30] G. Duffing. *Erzwungene Schwingungen bei veränderlicher Eigenfrequenz und ihre technische Bedeutung*. Number 41-42. R, Vieweg & Sohn, 1918.

[31] Y. Ueda. "Randomly transitional phenomena in the system governed by Duffing's equation". *Journal of Statistical Physics*, vol. 20, no. 2, pp. 181–196, 1979.