



Note on improvement precision of recursive function simulation in floating point standard

Melanie Rodrigues e Silva¹

Group of Control and Modeling, UFSJ, São João del-Rei, MG, Brazil

Erivelton Geraldo Nepomuceno²

Department of Electrical Engineering, UFSJ, São João del-Rei, MG, Brazil

Samir Angelo Milani Martins³

Department of Electrical Engineering, UFSJ, São João del-Rei, MG, Brazil

Abstract. An improvement on precision of recursive function simulation in IEEE floating point standard is presented. It is shown that the average of rounding towards negative infinite and rounding towards positive infinite yields a better result than the usual standard rounding to the nearest in the simulation of recursive functions. In general, the method improves one digit of precision and it has also been useful to avoid divergence from a correct stationary regime in the logistic map. Numerical studies are presented to illustrate the method.

Keywords. IEEE floating point, rounding mode, numerical simulation, recursive functions and Logistic map.

1 Introduction

The mathematical implementation of functions and systems in computers enables the scientific investigation in many areas. According to [19], interferences and noises of real systems are not incorporated into the simulations, though it is known that it is hardware as a set of real physical systems then the noises are always present. However, limitations of software and hardware are obstacles to the reliability of results [14, 15, 18, 22, 24, 26]. Nevertheless, alternatives have been developed to circumvent hardware shortcomings and improve results, such as in [3, 6, 7, 17, 25]. These alternatives that restricts the effects of hardware limitations are usually called as rigorous computing, which is based on the development of refined methods for the implementation of algorithms [11, 14].

Chaotic systems such as discrete maps and problems involving recursive functions have a differentiated degree of complexity, since the current iteration depends on the previous

¹me_rodrigues_silva@hotmail.com

²nepomuceno@ufs.br

³martins@ufs.br

ones [8]. The study of chaotic systems still receive attention and many of scientific conclusions in this area relies upon computer simulation. Chaotic behaviour has been associated to many real applications, from electronic circuits [2] to ultrasonic cutting system, which dynamic behavior is analyzed using a two-degree-of freedom Duffing oscillator model [21]. Moreover, these iterations are usually evaluated as numerical ill-conditioned [5]. In general, the simulation process of these types of functions involves errors and its reliability has been questioned in many works [14, 15]. In this sense, it is well known that in a floating point environment, a computer simulation of a chaotic system may be limited to a short number of iterates [26]. To deal with this problem, many tools have been investigated. We may summarize these tools in two categories. The first is focused on hardware performance using parallel and cluster computation [16]. The second major categories is devoted to improve the algorithm, using interval analysis, significance arithmetic or noisy-mode computation [11]. In this paper, we focus our attention on the noisy-mode computation, but instead of adding pseudo-random digits, as suggested in [11], we exploit the random nature of rounding error [4, 20]. Considering that the rounding errors are uniformly distributed, we present the background necessary for the understanding of the guidelines and the algorithm of a method that improves the precision of the computational simulation of recursive functions. In order to prove the efficiency of the proposed strategy, we bring examples of the logistic map case.

2 Background

This section contains some definitions about recursive functions, orbits, and interval pseudo-orbits. Let $n \in \mathbb{N}$, be a metric space such that $I \subseteq \mathbb{R}$, and $f : I \rightarrow \mathbb{R}$. Thus, recursive function can be defined as [18]:

$$x_n = f(x_{n-1}), \quad (1)$$

or by composite functions:

$$x_n = f_1(x_{n-1}) = f_2(x_{n-2}) = \dots = f_n(x_0), \quad (2)$$

where $f : I \rightarrow I$ is a recursive function or a map of a state space into itself and x_n denotes the state at the discrete time n . As suggested by [13], the sequence x_n obtained by iterating Eq. (1) starting from an initial condition x_0 is called the orbit of x_0 .

Definition 1. *An orbit is a sequence of values of a map, represented by $x_n = [x_0, x_1, \dots, x_n]$.*

The *Definition 2.* is suggested by [24] and highlights the presence of the difference between real and computational pseudo-orbits.

Definition 2. *Let $i \in \mathbb{N}$ represents a pseudo-orbit, which is defined by an initial condition and a combination of software and hardware. A pseudo-orbit is an approximation of an orbit and can be represented as*

$$\{\hat{x}_{i,n}\} = [\hat{x}_{i,0}, \hat{x}_{i,1}, \dots, \hat{x}_{i,n}] \quad (3)$$

such that,

$$|x_n - \hat{x}_{i,n}| \leq \xi_{i,n} \quad (4)$$

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

There is not an unique pseudo-orbit, as there are different hardware, software, numerical precision standard and discretization schemes, which can produce different results and consequently different errors $\xi_{i,n}$.

3 Method for error reduction

It is reasonable to assume that round-off errors are uniformly distributed [4, 9, 20, 23]. According to IEEE 754-2008 [10], the arithmetic operations are also rounded. In this sense, we also assume here that a finite set of arithmetic function presents an error randomly distributed.

Rounding to the nearest is the most usual for computational arithmetic. However, there are software that enables the user to set the rounding mode towards positive infinite or negative infinite. With these considerations in mind, we establish the main contribution of this letter in the following Lemma.

Lemma 1. *Let $\hat{x}_{i,n}^-$ and $\hat{x}_{i,n}^+$ be the calculated value by round towards positive infinite and round towards negative infinite, respectively. The arithmetic average given by*

$$\hat{x}_{j,n} = \frac{\hat{x}_{i,n}^+ + \hat{x}_{i,n}^-}{2} \quad (5)$$

such as $\hat{x}_{j,n} = f(\hat{x}_{j,n-1}) + \delta_{j,n}$, presents an round-off error smaller than the round-off error due the round to nearest as $n \rightarrow \infty$, therefore $\delta_{j,n} < \xi_{i,n}$.

Proof. Assuming Eq. (5), considering that $\hat{x}_{i,n}^- = f(\hat{x}_{j,n-1}) + \delta_{i,n}^-$ and $\hat{x}_{i,n}^+ = f(\hat{x}_{j,n-1}) + \delta_{i,n}^+$, then:

$$\hat{x}_{j,n} = \frac{f(\hat{x}_{j,n-1}) + \delta_{i,n}^- + f(\hat{x}_{j,n-1}) + \delta_{i,n}^+}{2} \Rightarrow \hat{x}_{j,n} = \frac{2f(\hat{x}_{j,n-1}) + \delta_{i,n}^- + \delta_{i,n}^+}{2} \quad (6)$$

and

$$\hat{x}_{j,n} = f(\hat{x}_{j,n-1}) + \frac{\delta_{i,n}^- + \delta_{i,n}^+}{2} \quad (7)$$

But, as we have been considered $\delta_{i,n}^-$ and $\delta_{i,n}^+$ uniformly distributed and it is well known that averaging a random variable leads to a reduction of the noise power in n , which is in this case is 2. And that completes the proof. $\square$

The Algorithm 1 presents a pseudo-code based on the Lemma 1. The round to the nearest is the standard mode. The operators $round^-$ and $round^+$ stands for round towards negative infinite and round towards positive infinite. Where there is no indication of the round mode, one should consider as the round to the nearest. In this paper, we implemented this algorithm in *Matlab R2016a*. The comparison is made with the high precision values provided by the *VPA* toolbox of *Matlab R2016a*.

Algorithm 1: Simulation of recursive function based on Lemma 1.

```

1  $\hat{x}_{i,0}$ ; %Initial condition
2  $N$ ; %Number of iterates
3  $\hat{x}_{i,0}^- \leftarrow \text{round}^-(\hat{x}_{i,0})$ ;
4  $\hat{x}_{i,0}^+ \leftarrow \text{round}^+(\hat{x}_{i,0})$ ;
5  $\hat{x}_{j,0} \leftarrow (\hat{x}_{i,0}^- + \hat{x}_{i,0}^+)/2$ ; %Average of the two round modes
6  $\hat{x}_{i,0}^- \leftarrow \hat{x}_{j,0}$ ;
7  $\hat{x}_{i,0}^+ \leftarrow \hat{x}_{j,0}$ ;
8 For  $n \leftarrow 1$  to  $N$  %Main loop
9  $\hat{x}_{i,n}^- \leftarrow \text{round}^-(f(\hat{x}_{j,n-1}^-))$ ;
10  $\hat{x}_{i,n}^+ \leftarrow \text{round}^+(f(\hat{x}_{j,n-1}^+))$ ;
11  $\hat{x}_{j,n} \leftarrow (\hat{x}_{i,n}^- + \hat{x}_{i,n}^+)/2$ ;
12  $\hat{x}_{i,n}^- \leftarrow \hat{x}_{j,n}$ ;
13  $\hat{x}_{i,n}^+ \leftarrow \hat{x}_{j,n}$ ;
14 EndFor

```

4 Numerical Examples

This sections presents three numerical examples using the method proposed in this paper. All the examples are based on the logistic map [1]

$$x_{n+1} = rx_n(1 - x_n), \quad (8)$$

where the initial condition and the value of bifurcation parameter r are indicated in each example.

Example 4.1. *Logistic map with $x_0 = 1/3.9$ and $r = 3.9$.*

Only using the round to nearest, the pseudo-orbit presents a chaotic behaviour. However, using the Algorithm 1, it is observed that the logistic map function converges to a fixed point as shown in Table 1. As one can see, the $\hat{x}_{j,n}$ reaches a fixed point, which is the exact answer for this example, as verified in the following equations:

$$x_1 = 3.9(1/3.9)(1 - (1/3.9)) = (1 - 10/39) = 29/39 \quad (9)$$

$$x_2 = 3.9(29/39)(1 - 29/39) = 29/39 \quad (10)$$

and $29/39 \approx 0.743589743589744$ is a fixed point, which is the value that $\hat{x}_{j,n}$ converges to. On the other hand, the pseudo-orbit $\hat{x}_{i,n}$ diverges and presents a chaotic behaviour.

Example 4.2. *Logistic map with $r = 3.9$ and $x_0 = 0.01$.*

The Figure 1 shows the logarithm (base 10) of the error for the first 20 iterates of the logistic map, using the Algorithm 1 and a traditional method based on the round to the nearest. The Algorithm 1 starts with an error greater than the traditional method, but after few iterates the error becomes smaller. The error presented in Figure 1 has been calculated by means of VPA toolbox of Matlab with 1000-digit precision.

Table 1: Iterates of the logistic map with $x_0 = 1/3.9$ and $r = 3.9$. $\hat{x}_{i,n}$ is following the rounding to the nearest and $\hat{x}_{j,n}$ is following the Algorithm 1. The values are shown in decimal and hexadecimal notation. Notice that the third iterate is equal to the second for the Algorithm 1. This can be seen only in the hex format.

n	$\hat{x}_{i,n}$ (hex)	$\hat{x}_{i,n}$ (dec)	$\hat{x}_{j,n}$ (hex)	$\hat{x}_{j,n}$ (dec)
1	3fd0690690690691	0.256410256410256	3fd0690690690691	0.256410256410256
2	3fe7cb7cb7cb7cb8	0.743589743589744	3fe7cb7cb7cb7cb8	0.743589743589744
3	3fe7cb7cb7cb7cb7	0.743589743589744	3fe7cb7cb7cb7cb8	0.743589743589744
4	3fe7cb7cb7cb7cb9	0.743589743589744	3fe7cb7cb7cb7cb8	0.743589743589744
5	3fe7cb7cb7cb7cb5	0.743589743589743	3fe7cb7cb7cb7cb8	0.743589743589744
6	3fe7cb7cb7cb7cbd	0.743589743589744	3fe7cb7cb7cb7cb8	0.743589743589744
7	3fe7cb7cb7cb7cae	0.743589743589743	3fe7cb7cb7cb7cb8	0.743589743589744
8	3fe7cb7cb7cb7ccb	0.743589743589746	3fe7cb7cb7cb7cb8	0.743589743589744
9	3fe7cb7cb7cb7c93	0.743589743589740	3fe7cb7cb7cb7cb8	0.743589743589744
10	3fe7cb7cb7cb7cfd	0.743589743589751	3fe7cb7cb7cb7cb8	0.743589743589744

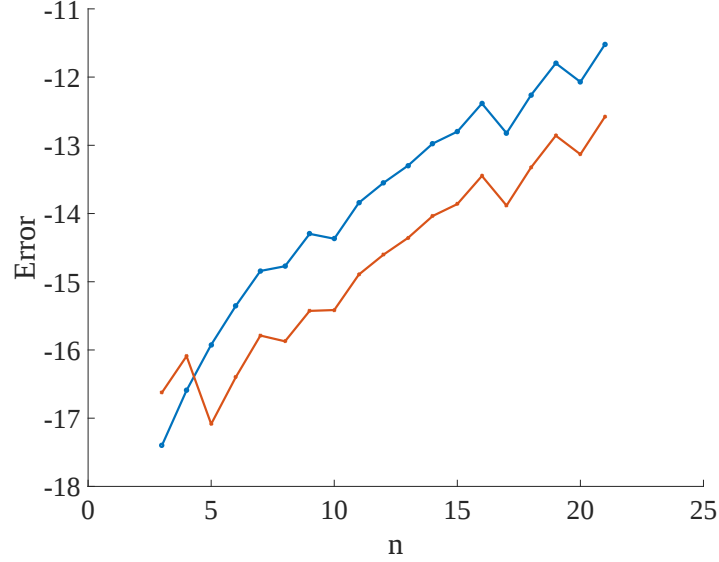


Figure 1: Propagation error for the logistic map with $r = 3.9$ and $x_0 = 0.01$. Traditional method (blue). Algorithm 1 (red)

Example 4.3. *Logistic map with three sets of initial conditions and bifurcation parameter, given by $(x_0, r) = [(0.1, 4.2); (0.2, 4); (0.41, 3.85)]$.*

Table 2 shows results of $\xi_{i,n}$ and $\delta_{j,n}$ for the first ten iterations of the logistic map case, considering these three different set of initial conditions and bifurcation parameter. It is possible to note a general decrease in the error that keeps along the iteration process. We arbitrary chosen these parameters, but we have performed our tests with many other sets with similar results.

Table 2: Results of $\xi_{i,n}$ and $\delta_{j,n}$ of the logistic map case, considering different initial conditions, x_0 and r values.

	$x_0 = 0.1$ and $r = 4.2$		$x_0 = 0.2$ and $r = 4$		$x_0 = 0.41$ and $r = 3.85$	
n	$\xi_{i,n}$	$\delta_{j,n}$	$\xi_{i,n}$	$\delta_{j,n}$	$\xi_{i,n}$	$\delta_{j,n}$
1	9.33254 e-18	9.33254 e-18	4.57088 e-16	1.20226 e-16	3.63078 e-16	2.69153 e-17
2	3.98107 e-17	3.31131 e-17	7.58577 e-16	9.12010 e-17	7.76247 e-16	6.76082 e-18
3	1.38038 e-16	8.31763 e-17	1.99526 e-15	2.08929 e-16	1.25893 e-15	4.67735 e-17
4	3.38844 e-16	1.17489 e-17	1.41253 e-15	6.76082 e-17	2.81838 e-15	1.28824 e-16
5	1.86208 e-15	7.58577 e-16	5.24807 e-15	2.69153 e-16	3.23593 e-15	2.51188 e-16
6	1.90546 e-15	6.91830 e-16	1.62181 e-14	8.51138 e-16	9.33254 e-15	7.07945 e-16
7	7.76247 e-15	2.81838 e-15	1.28824 e-14	6.76082 e-16	6.02559 e-15	4.67735 e-16
8	2.88403 e-14	1.04712 e-14	4.78630 e-14	2.51188 e-15	1.99526 e-14	1.54881 e-15
9	6.30957 e-14	2.29086 e-14	1.34896 e-13	7.07945 e-15	4.16869 e-14	3.23593 e-15
10	1.41253 e-13	5.12861 e-14	4.16869 e-15	1.94984 e-16	5.88843 e-14	4.46683 e-15

5 Conclusions

This paper presents a method based on the average of the round modes which promotes a better performance in the simulation of recursive function. The method exploits the stochastic nature of rounding. It consists on the average of two round mode which promotes a reduction of the power noise by a factor of 2. This is a good example of, although a simulation can be seen as a theoretical experiment, in fact, it presents some real world dimension, as stochasticity, and therefore, usual tools presented in Engineering can be applied.

We applied the method in three numerical examples. In the first case, we show that the method is able to keep the correct fixed point, while the rounding to the nearest diverge to chaotic behaviour. In the second and third we show the general behaviour of the method in reducing the error of the simulation.

It is important to say that there is no significant increase of computational time and this method can be easily extended to many other applications, such as numerical solution of differential equations, as seen applied in [21]. In the future, we intend to investigate the use of interval extensions and other rounding modes as further discussed in [24, 26].

Acknowledgements

This work has been supported by the CNPq and CAPES. A special thanks to Control and Modelling Group (GCOM).

References

- [1] R. May Simple mathematical models with very complicated dynamics, *Nature*, vol. 261, 459–467, (1976).
- [2] T. Matsumoto, A chaotic attractor from Chua’s circuit, *Circuits and Systems, IEEE Transactions on*, vol. 31, 1055–1058, (1984).

- [3] K. Sugihara and M. Iri, Two design principles of geometric algorithms in finite-precision arithmetic, *Applied Mathematics Letters*, vol. 2, 203–206, (1989).
- [4] K. Furuya, H. Date and K. Fukudome, Effects of finite register length in the fast hartley transform, *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, vol. 75, 76–86, (1992).
- [5] R. M. Corless, What good are numerical simulations of chaotic dynamical systems?, *Computers & Mathematics with Applications*, vol. 28, 107–121, (1994).
- [6] P. V. Mcdonough and J. P. Noonan and G. H. Hall, A new chaos detector, *Computers & Electrical Engineering*, vol. 21, 417–431, (1995).
- [7] G. S. Maharana and P. K. Meher, Algorithm for efficient interpolation of real-valued signals using discrete Hartley transform, *Computers & Electrical Engineering*, vol. 23, 129–134, (1997).
- [8] E. Ott, *Chaos in dynamical systems*, Cambridge: Cambridge University Press, (2002).
- [9] R. Landau and M. J. Paez and C. C. Bordeianu, *Computational physics : problem solving with computers*, Wiley-VCH John Wiley distributor, (2007).
- [10] IEEE. Institute of Electr. and Electr. Eng., 754-2008 - IEEE standard for floating-point arithmetic, IEEE, (2008).
- [11] B. Parhami, *Computer arithmetic algorithms and hardware architectures*, Oxford Univ. Press, New York, (2012).
- [12] R. Gilmore and M. Lefranc, *The topology of chaos: Alice in stretch and squeezeland*, John Wiley & Sons, (2012).
- [13] R. Gilmore and M. Lefranc, *The topology of chaos: Alice in stretch and squeezeland*, John Wiley & Sons, (2012).
- [14] 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, 35–52, (2013).
- [15] R. Lozi, Can we trust in numerical computations of chaotic solutions of dynamical systems?, *World Scientific Publishing Co. Pte. Ltd.*, vol. 1, 63–98, (2013).
- [16] S.J. Liao and P.F. Wang, On the mathematically reliable long-term simulation of chaotic solutions of Lorenz equation in the interval $[0, 10000]$, *Science China: Physics, Mechanics and Astronomy*, vol. 57, 330–335, (2014).
- [17] C. Wang and T. Tang, Recursive least squares estimation algorithm applied to a class of linear-in-parameters output error moving average systems, *Applied Mathematics Letters*, vol. 29, 35–41, (2014).

- [18] E. G. Nepomuceno, Convergence of recursive functions on computers, *The Journal of Engineering*, 1–3, (2014).
- [19] G. M. V. Ladeira and G. V. Lima and C. A. Araújo, Analysis of nonlinear dynamics using electronic circuits, 23rd ABCM International Congress of Mechanical Engineering - COBEM2015, (2015).
- [20] Y. A. Gadzhiev, On the variance of a centered random value roundoff error, *Signal Processing*, Elsevier, vol. 106, 30–40, (2015).
- [21] K. Yu and C. Wang and H. Yau and J. Chen, Numerical computation and nonlinear dynamic analysis of ultrasonic cutting system, *Computers & Electrical Engineering*, vol. 51, 270–283, (2016).
- [22] E. G. Nepomuceno, and S. A. M. Martins, A lower bound error for free-run simulation of the polynomial NARMAX, *Systems Science & Control Engineering*, vol. 4, 50–58, (2016).
- [23] R. Li and S. Nadarajah, Mean and variance of round off error, *Signal Processing*, vol. 127, 185–190, (2016).
- [24] E. M. A. Mendes and E. G. Nepomuceno, A Very Simple Method to Calculate the (Positive) Largest Lyapunov Exponent Using Interval Extensions, *International Journal of Bifurcation and Chaos*, vol. 26, 1650226, (2016).
- [25] W. Shi and J. Wu and S. Lam and T. Srikanthan, Algorithms for bi-objective multiple-choice hardware/software partitioning, *Computers & Electrical Engineering*, vol. 50, 127–142, (2016).
- [26] E. G. Nepomuceno and E. M. A. 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, 21–32, (2017).