



INPE – National Institute for Space Research
São José dos Campos – SP – Brazil – May 16-20, 2016

SIMULATION OF RECURSIVE FUNCTIONS BY MEANS OF INTERVAL ANALYSIS AND PSEUDO-ORBITS

Rodrigues Junior, H. M.¹, Peixoto, M. L. C.², Nepomuceno, E. G.³

¹Modelling and Control Research Group - GCOM, Federal University of São João del-Rei - UFSJ ,
São João del-Rei - MG, Brazil ,
heitormrjunior@gmail.com

²marciapexoto93@hotmail.com

³nepomuceno@ufsjeu.br

Abstract: In this work we use intersection of different pseudo-orbits obtained by interval extensions to reduce the bounds of the exact solution provided by the toolbox Intlab. The method is applied on the logistic map.

keywords: Modeling, Numerical Simulation and Optimization; Chaos and Global Nonlinear Dynamics; Bifurcation Analysis and Applications; Interval Analysis; Multiple Pseudo-Orbits.

1. INTRODUCTION

Numerical computational techniques are almost never exact. This fact occurs due to limitations in representing real numbers by computers. Instead of a true result, its approximation is returned [1]. Different software, machines and levels of precision may lead us to different results. Even for some mathematical equivalent functions, small rounding errors can be propagated, and after some iterations the result is no longer significant [1–3].

The error propagation problem is especially important for recursive functions that characterize chaotic systems, since small errors introduced in each computational step grow due the fact that the system is not contractive mapping [1, 2, 4]. According to [2], a recursive function can be defined as

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

A discrete-time series can be generated from a map by

choosing some values for its parameters, an initial condition and iterating it recursively. The sequence of values of a map, represented by $\{x_0, x_1, \dots, x_n\}$ is an orbit.

However, when a computer is used to iterate a map f , we have a pseudo-orbit [4]. In other words, there is an approximation of a true orbit due to inherent properties of digital computers and we represent a specific pseudo-orbit $i \in \mathbb{N}$ as $\{\hat{x}_{i,0}, \hat{x}_{i,1}, \dots, \hat{x}_{i,n}\}$, such that

$$|x_n - \hat{x}_{i,n}| \leq \delta_{i,n}, \quad (2)$$

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

The logistic map [5] is a recursive function given by

$$x_{n+1} = rx_n(1 - x_n). \quad (3)$$

This function describes a biological model in which x_n is a number between zero and one that represents the ratio of existing population to the maximum possible and r is the growth rate of the population. The values of interest for the parameter r are those between zero and four. May [5] was the first to describe a rich set of dynamical properties subject to the variation to r , varying from a fixed point to a chaotic behaviour.

To illustrate the error propagation, let us compute two pseudo-orbits, the first is given by $f(x) = rx(1 - x)$, and the second by its natural extension $F(x) = rx - rx^2$, with $r = 3.9$ and $x_0 = 0.6$. Any mathematical equivalent func-

tion of f is a natural extension. . The results obtained for the iterations 50 to 70 is shown in Figure 1.

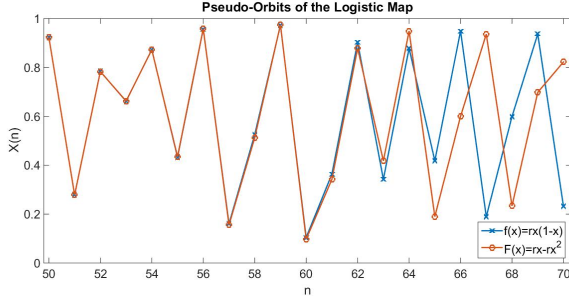


Figure 1 – Simulation of (3), with $r = 3.9$ and $x_0 = 0.6$ for two different natural extensions. The simulation was performed in Matlab, which follows the IEEE 754-2008 standard for float point.

Clearly, the trajectory of these two pseudo-orbits is completely different after 60 iterations. For example, the distance between these pseudo-orbits at $n = 67$ is greater than 0.74. Thus, both pseudo-orbits can not be accurate, and this lead us to a conclusion that even for very simple systems, propagation of rounding errors may lead to completely wrong results.

One way to try to reduce this problem is by means interval analysis [6]. While floating-point arithmetic is affected by rounding errors and can produce inaccurate results, interval arithmetic has the advantage of giving rigorous bounds for the exact solution. If the lower and upper bounds of the interval can be rounded down and rounded up, respectively, then finite precision calculations can be performed using intervals, to give an enclosure of the exact solution [7].

2. PURPOSE

In interval analysis, real numbers are replaced by intervals. An interval is a closed set of real numbers $x \in \mathbb{R}$ often denoted by an upper case letter X with endpoints $\underline{x} \leq \bar{x}$ such that $X = [\underline{x}, \bar{x}] = \{x : \underline{x} \leq x \leq \bar{x}\}$.

Let f be a function of a real variable x . An interval extension of f is an interval valued function F of an interval variable X with property

$$F(x) = f(x), \quad x \in \mathbb{R}. \quad (4)$$

Although mathematically equivalent, the interval extensions may have different sequences of arithmetic operations. Therefore it is important to mention that simple mathematical properties are not assured in interval arithmetic [3].

For a given interval $X = [\underline{x}, \bar{x}]$, its width is defined by $w(X) = \bar{x} - \underline{x}$ and its centre is $m(X) = 0.5(\underline{x} + \bar{x})$ [8]. The intersection of two intervals $X \cap Y$ is a set of real numbers which belongs to both. The union $X \cup Y$ is a set of real numbers which belongs to X or Y (or both). If $X \cap Y$ is not empty, then

$$w(X \cap Y) \leq \min\{w(X), w(Y)\}. \quad (5)$$

The accuracy of a numerical computation in an interval is inversely proportional to the size of the interval. Therefore, the smaller the interval, the more accurate the information. Besides that, if $X \cap Y \neq \emptyset$, the exact solution may belong to both intervals [6].

The aim of this paper is to use the Intlab toolbox to show that the intersection of different pseudo-orbits obtained from interval extensions of the logistic map, generate a smaller interval where the real orbit is contained, increasing the accuracy of solution.

The Intlab is a toolbox for Matlab supporting real and complex intervals designed to be very fast. The Intlab concept splits in a fast interval library and an interactive programming environment for easy use of interval operations [9].

3. METHODS

Let $\{x_0, x_1, \dots, x_n\}$ be a true orbit and let $[\hat{X}_{i,0}, \hat{X}_{i,1}, \dots, \hat{X}_{i,n}]$ be a pseudo-orbit obtained by some interval extension $i \in \mathbb{N}$ such that

$$\hat{X}_{i,n} = [\underline{\hat{x}}_{i,n}, \bar{\hat{x}}_{i,n}] = [x_n - \delta_{i,n}, x_n + \delta_{i,n}]. \quad (6)$$

From (2) and (6) it is clear that

$$x_n \in \hat{X}_{i,n}. \quad (7)$$

From this we may establish the following for two interval extensions F_1 and F_2 of the same function.

Lemma 1 $x_n \in [\hat{X}_{1,n} \cap \hat{X}_{2,n}]$, $n \in \mathbb{N}$.

Proof. Let us assume $x_n \notin [\hat{X}_{1,n} \cap \hat{X}_{2,n}]$, $n \in \mathbb{N}$. Hence $x_n \notin \hat{X}_{1,n}$ or $x_n \notin \hat{X}_{2,n}$ or both. According to (7), it means that $\hat{X}_{1,n}$ or $\hat{X}_{2,n}$ is not a pseudo-orbit, which is a contradiction. That completes the proof. $\square$

Besides that, according to (5) the intersection of these interval extensions can be used to limit the bounds of the true orbit x_n . To illustrate this fact we will use different interval extensions to calculate three fixed points of distinct periods of the logistic map. If $f(x_n) = x_{n-p}$ then x_n is a fixed point of period p . Based on [10], in interval arithmetic we have the following.

Lemma 2 If $X_{i,n} \cap X_{i,n-p} \neq \emptyset$, then $x^* \in X_{i,n}$ is a fixed point of period p .

Proof. Let us assume that x^* is a fixed point of period p . It means that $x^* \in X_{i,n}$ and $x^* \in X_{i,n-p}$, $i, n, p \in \mathbb{R}$. If $X_{i,n} \cap X_{i,n-p} = \emptyset$, it means that $x^* \notin X_{i,n}$ or $x^* \notin X_{i,n-p}$ or both, which is a contradiction. $\square$

In this work, first we will use the traditional approach to calculate the fixed points by an interval extension F_1 of (3). Then we will compare these calculations with a method

which the calculation is made by the intersection at each iteration of two interval extensions F_1 and F_2 of (3). We will use $F_1(X) = rX(1 - X)$ and $F_2(X) = r(X(1 - X))$.

These methods will be compared by the width of the intervals and the number of iterations to get to some fixed point. Based on [11], three cases of fixed points of (3) will be analyzed:

- Case 1: $r = 3.3$ and $x_0 = 0.6$, which orbit may converge to fixed point of period 2.
- Case 2: $r = 3.47$ and $x_0 = 0.6$, which orbit may converge to fixed point of period 4.
- Case 3: $r = 3.55$ and $x_0 = 0.6$, which orbit may converge to fixed point of period 8.

We used Matlab with double precision to compute the calculations. The first interval of each case is obtained rounding x_0 up and down. In other words, its endpoints $\underline{x}_0$ and $\overline{x}_0$ are the nearest floating points of x_0 . The algorithm in Matlab can be found in Section 7, Appendix.

4. RESULTS

The results obtained for each case with the calculations of F_1 and the intersection $F_1 \cap F_2$ are presented in Tables 1, 2 and 3. The fixed point, the width of the interval and the number of iterations reached by each method can be analysed at these tables.

Table 1 – Solution of case 1 for F_1 and $F_1 \cap F_2$.

	F_1	$F_1 \cap F_2$
Period of Fixed Point	2	2
Width of Interval	3.65900×10^{-6}	3.65833×10^{-6}
Number of Iterations	20	20

Table 2 – Solution of case 2 for F_1 and $F_1 \cap F_2$.

	F_1	$F_1 \cap F_2$
Period of Fixed Point	4	4
Width of Interval	6.44715×10^{-4}	4.72274×10^{-4}
Number of Iterations	23	23

Table 3 – Solution of case 3 for F_1 and $F_1 \cap F_2$.

	F_1	$F_1 \cap F_2$
Period of Fixed Point	8	8
Width of Interval	9.14991×10^{-4}	7.91852×10^{-4}
Number of Iterations	23	23

In Tables 1, 2 and 3 we note that, despite the same number of iterations to converge to the same fixed point, there is a difference between the final sizes of intervals obtained by each method in each case.

5. DISCUSSION

By analysing the results, we note that the two calculations converge to the results expected according to [11]. It shows that both were effective for these cases. The difference between the size of intervals obtained by these methods tends to grow when we approximate of unstable zones of (3) with r approaching 4. Table 4 shows the sizes of intervals obtained for the first 10 iterations for each method to equation (3) with $r = 3.6$ and $x_0 = 0.6$. We can see that $F_1 \cap F_2$ has a lower growth rate compared to F_1 .

Table 4 – Size of intervals obtained by F_1 and $F_1 \cap F_2$.

Iteration	F_1	$F_1 \cap F_2$	Difference
0	2.22×10^{-16}	2.22×10^{-16}	0
1	6.66×10^{-16}	6.66×10^{-16}	0
2	2.55×10^{-15}	2.11×10^{-15}	4.44×10^{-16}
3	9.77×10^{-15}	7.77×10^{-15}	2.00×10^{-15}
4	3.48×10^{-14}	2.77×10^{-14}	7.10×10^{-15}
5	1.25×10^{-13}	9.99×10^{-14}	2.60×10^{-14}
6	4.53×10^{-13}	3.60×10^{-13}	9.36×10^{-14}
7	1.63×10^{-12}	1.29×10^{-12}	3.37×10^{-13}
8	5.87×10^{-12}	4.66×10^{-12}	1.21×10^{-12}
9	2.11×10^{-11}	1.67×10^{-11}	4.37×10^{-12}
10	7.62×10^{-11}	6.04×10^{-11}	1.57×10^{-11}

6. CONCLUSIONS

Both methods are effective to estimate the values of fixed points. Our proposed method gets some slight better results as the size of intervals obtained by the method based on the intersections of the interval extensions is smaller or equal than that obtained by traditional approach. It means that the proposed method can be used to increase the accuracy of the solutions. For example, in Table 2, there is a reduction of approximately 26% and in Table 3, this reductions is 13%.

We expect that using more interval extensions the results may be improved, which is as task for future investigations. We also intend to analyse the behaviour of the method in chaotic regime and examine how its use is feasible to help in the bifurcation diagram building for the logistic map and other maps.

ACKNOWLEDGEMENTS

We would like to thank CAPES, CNPq/INERGE, FAPEMIG and Federal University of São João del-Rei by support.

References

- [1] Z. Galias, “The dangers of rounding errors for simulations and analysis of nonlinear circuits and systems? and how to avoid them,” *Circuits and Systems Magazine, IEEE*, vol. 13, no. 3, pp. 35–52, 2013.
- [2] E. G. Nepomuceno, “Convergence of recursive func-

tions on computers,” *The Journal of Engineering*, vol. 1, no. 1, 2014.

- [3] E. G. Nepomuceno and S. A. M. Martins, “A lower-bound error for free-run simulation of the polynomial narmax,” *Systems Science & Control Engineering*, no. ja, pp. 1–14, 2016. Doi: 10.1080/21642583.2016.1163296.
- [4] S. Hammel, J. 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.
- [5] R. M. May, “Simple mathematical models with very complicated dynamics,” *Nature*, vol. 261, pp. 459–467, 1976.
- [6] R. E. Moore, *Methods and applications of interval analysis*, vol. 2. SIAM, 1979.
- [7] G. Ruetsch, “An interval algorithm for multi-objective optimization,” *Structural and Multidisciplinary Optimization*, vol. 30, no. 1, pp. 27–37, 2005.
- [8] G. Alefeld and J. Herzberger, *Introduction to interval computation*. Academic press, 2012.
- [9] S. M. Rump, *INTLAB Interval Laboratory*. Springer, 1999.
- [10] H. M. Rodrigues Junior and E. G. Nepomuceno, “Uso da computação por intervalos para cálculo de ponto fixo de um mapa discreto,” in *Proceedings of DINCON 2015 - Conferência Brasileira de Dinâmica, Controle e Aplicações*, 2015.
- [11] M. J. Feigenbaum, “Quantitative Universality for a Class of Non-linear Transformations,” *journal Of Statistical Physics*, vol. 19, no. 1, pp. 25–52, 1978.

7. APPENDIX

7.1. Matlab Code of Proposed Method

```
clear all; clc; close all;
r=347/100;
N=40;
setround(-1);
x_inf= 0.6;
setround(1);
x_sup= 0.6;
setround(0);
F(1) = infsup(x_inf(1),x_sup(1));
F_1(1) = infsup(x_inf,x_sup);
F_2(1) = infsup(x_inf,x_sup);
s = 0;
for n=1:N
    F_1(n+1) = r*(F(n)*(1-F(n)));
    F_2(n+1) = r*(F(n)*(1-F(n)));
    F(n+1) = intersect(F_1(n+1),F_2(n+1));
    %Fixed Point Verification
    for i=1:n
        if n+1 >= 2*i
            int_F = intersect(F(n+1),F(n+1-i));
            int_aux = infsup(-100,100);
            f = in(int_F,int_aux);
            if f == 1
                disp(['Period: ' num2str(i)]);
                disp(['Iteration: ' num2str(n)]);
                s = 1;
                break;
            end
        end
    end
    if s == 1
        break;
    end
end
```