

CÁLCULO DE PONTOS FIXOS POR ANÁLISE INTERVALAR PARA MAPAS DISCRETOS

HEITOR M. RODRIGUES JUNIOR*, MÁRCIA L. C. PEIXOTO†, LUCAS G. NARDO†, THALITA E. NAZARÉ†, ERIVELTON G. NEPOMUCENO*

*Programa de Pós-Graduação em Engenharia Elétrica (UFSJ/CEFET)
Universidade Federal de São João del-Rei
Pça. Frei Orlando, 170 - Centro - 36307-352 - São João del-Rei, MG, Brasil

†Grupo de Pesquisa em Controle e Modelagem (GCOM)
Departamento de Engenharia Elétrica
Universidade Federal de São João del-Rei
Pça. Frei Orlando, 170 - Centro - 36307-352 - São João del-Rei, MG, Brasil

Emails: heitormrjunior@gmail.com, marciapeixoto93@hotmail.com, gnlucas@gmail.com, thalitaenazare@hotmail.com, nepomuceno@ufs.br

Abstract— Two methods of detection of fixed points with interval arithmetic are discussed and compared with *Intlab*, which is a computational tool that works with this arithmetic. The results are compared through the computational cost and size of intervals obtained by each method. Although all methods reach the expected fixed points, there are differences in the accuracy of results obtained by each method.

Keywords— Interval Analysis, Fixed Point, *Intlab*, Logistic Map.

Resumo— Dois métodos de detecção de pontos fixos em funções recursivas que utilizam a aritmética intervalar são abordados e comparados com o *Intlab*, ferramenta computacional que trabalha com essa aritmética. A comparação dos cálculos realizados é feita através do custo computacional e do tamanho dos intervalos obtidos por cada método. Verifica-se que apesar de todos os métodos atingirem os pontos fixos esperados, existem diferenças na precisão dos resultados obtidos por cada método.

Palavras-chave— Análise Intervalar, Ponto Fixo, *Intlab*, Mapa Logístico.

1 Introdução

A computação numérica desempenha um papel fundamental na análise de soluções de sistemas dinâmicos não-lineares, sendo usada para resolver equações que modelam os mais diferentes sistemas (Overton, 2001; Lozi, 2013). Porém, resultados obtidos a partir do uso de técnicas computacionais quase nunca são exatos. Isso ocorre graças à limitação de representação dos números reais pelos computadores (Galias, 2013; Kulisch, 2012).

O fato de a computação numérica geralmente fazer uso da norma IEEE 754 para aritmética de ponto flutuante faz com que mesmo para um dado de entrada que pode ser precisamente representado pela máquina, o resultado de uma simples operação matemática envolvendo esse número possa não ser representável, gerando uma aproximação do resultado exato devida aos erros de arredondamento (Overton, 2001; Galias, 2013).

Enquanto a aritmética de ponto flutuante é afetada por erros de arredondamento, podendo produzir resultados imprecisos, a aritmética intervalar tem a vantagem de limitar a solução exata (Ruetsch, 2005). A técnica de computação por intervalos (Moore, 1979) tem o intuito de controlar estes erros em cálculos envolvendo ponto flutuante e vem sendo usada em diversas áreas, como análise de circuitos elétricos (Grigoletti et al., 2006), confiabilidade de sistemas com processo exponen-

cial de falhas (Mendonça e Campos, 2013), análise de sistemas não-lineares (Galias, 2013), processamento de sinais e sistemas elétricos de potência (Pereira e Costa, 2012) e simulação do polinômio NARMAX (Nepomuceno e Martins, 2016).

Em sistemas dinâmicos discretos, os pontos fixos tem importância fundamental como elementos para determinação de pontos de equilíbrio destes sistemas (Devaney et al., 1993).

O objetivo deste trabalho é comparar métodos de detecção de pontos fixos em funções recursivas, que fazem uso da aritmética intervalar: Nepomuceno (2014) faz a análise da convergência dessas funções por meio da propagação do erro em cada operação aritmética, enquanto Rodrigues Junior e Nepomuceno (2015) desenvolvem um método que leva em conta os intervalos de representação dos números no computador, na intenção de solucionar o problema da falta de precisão dos dados da máquina.

Ambos os métodos estudados apresentaram resultados similares alcançando os pontos fixos nas regiões estipuladas. No entanto, o método desenvolvido por Rodrigues Junior e Nepomuceno (2015) apresenta solução apenas para um caso de detecção de pontos fixos. Portanto, o que se pretende é, além de aprimorar o método de Rodrigues Junior e Nepomuceno (2015) de modo que abranja mais casos no cálculo de pontos fixos, testar ambos os métodos para outras condições e

comparar seus resultados com os obtidos pelo *Interval*, ferramenta computacional que trabalha com a aritmética intervalar na forma tradicional.

2 Conceitos Preliminares

2.1 Análise Intervalar

A técnica de computação por intervalos foi proposta em sua forma moderna por Moore (1979) como uma ferramenta para limitar os erros de arredondamentos na computação numérica e é um campo que engloba a matemática rigorosa com a computação científica. Essa ligação entre computação e matemática intervalar, torna possível resolver problemas que não podem ser resolvidos de forma eficiente usando a aritmética tradicional de ponto flutuante (Moore, 1979).

Portanto, a análise intervalar lida com dados numéricos na forma de intervalos compactos com o objetivo de codificar os erros computacionais no próprio intervalo (Hickey et al., 2001).

Por um intervalo, entende-se um conjunto fechado e limitado de números reais $x \in \mathbb{R}$ geralmente denotado por uma letra maiúscula X com os limites $x^- \leq x^+$ tal que $X = [x^-, x^+] = \{x : x^- \leq x \leq x^+\}$ (Nepomuceno e Martins, 2016). Dados dois intervalos reais $X = [x^-, x^+]$ e $Y = [y^-, y^+]$, a interseção $X \cap Y$ é o conjunto de números reais que pertence a ambos. A união $X \cup Y$ é o conjunto de números reais que pertencem a um ou outro (ou ambos) (Rothwell e Cloud, 2012).

O tamanho de um intervalo é dado por $\omega(X) = [x^+ - x^-]$. É importante destacar que a precisão da informação contida em um intervalo é inversamente proporcional ao tamanho do intervalo. Sendo assim, quanto menor o intervalo, mais precisa é a informação nele contida (Moore, 1979).

2.2 Funções Recursivas

O problema da propagação de erros deve ser especialmente levado em conta em funções recursivas que caracterizam sistemas caóticos, uma vez que pequenos erros obtidos em cada passo computacional crescem devido ao fato de que a contração não pode ser aplicada ao sistema (Galias, 2013; Hammel et al., 1987; Nepomuceno, 2014).

Uma função recursiva pode ser definida da seguinte forma: seja $\mathbb{I} \subseteq \mathbb{R}$ um espaço métrico com $f : \mathbb{I} \rightarrow \mathbb{R}$, tem-se que:

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

Uma série de tempo discreto pode ser gerada por um procedimento iterativo simples de (1). A sequência de valores de um mapa, representada por $\{x_0, x_1, \dots, x_n\}$ é uma órbita. Dependendo da escolha de f , a série pode apresentar tanto ponto fixo de período 1 ou maior, quanto comportamento caótico.

Segundo Feigenbaum (1978), se $f(x^*) = x^*$, então diz-se que x^* é um ponto fixo de $f(x)$. Expandindo esse conceito para os casos de pontos fixos de períodos superiores, tem-se:

Definição 1 Se $f(x_n) = x_{n-p}$, então o conjunto $\{x_{n-p}, \dots, x_{n-1}\}$ representa os pontos fixos de $f(x)$ de período p , $\forall n, p \in \mathbb{N}$.

2.3 Métodos Estudados

O método apresentado por Nepomuceno (2014) consiste na determinação de um erro $\delta_n > 0$ relacionado a cada operação aritmética realizada pelo computador a cada saída x_n de um mapa recursivo. Seja $D \subset \mathbb{Q}$ o conjunto dos números que podem ser armazenados no computador, para o intervalo $I \subseteq \mathbb{R}$, com $x \in I$ e $f_n : I \rightarrow \mathbb{R}$ sendo uma função iterativa, existe um outro intervalo $J \subseteq D$, com $\hat{x} \in J$ sendo uma representação de x e $\hat{f}_n$ sendo uma aproximação de f_n . Com isso, tem-se que

$$d(f_n(x), \hat{f}_n(\hat{x})) \leq \delta_n. \quad (2)$$

A partir de (2), pode-se afirmar que $f_m(x) \neq f_n(x)$ somente se

$$d(\hat{f}_m(\hat{x}), \hat{f}_n(\hat{x})) > \delta_m + \delta_n. \quad (3)$$

Portanto, com base em (Nepomuceno, 2014), pode-se fazer a seguinte definição a respeito de ponto fixo:

Definição 2 Seja $m - n = p$, $\forall m, n, p \in \mathbb{N}$. Se $d(\hat{f}_m(\hat{x}^*), \hat{f}_n(\hat{x}^*)) \leq \delta_m + \delta_n$, então f apresenta ponto fixo de período p .

Na metodologia adotada por Rodrigues Junior e Nepomuceno (2015), a solução encontrada para resolver o problema da influência da propagação do erro computacional foi trabalhar com intervalos gerados a partir do arredondamento da condição inicial x_0 para cima e para baixo, ou seja, os limites x_0^- e x_0^+ são os pontos flutuantes mais próximos de x_0 . Após a definição do intervalo inicial, são realizadas operações com cada limite. Essas operações são também arredondadas de modo que o tamanho do intervalo nunca fique nulo.

A definição de ponto fixo para esse trabalho pode ser abordada de uma forma mais simples em relação ao outro método.

Definição 3 Seja a sequência $[\hat{X}_0, \hat{X}_1, \dots, \hat{X}_n]$ a representação dos intervalos obtidos a cada iteração n como resposta de um mapa recursivo. Se $X_n \cap X_{n-p} \neq \emptyset$, então $x^* \in X_n$ é ponto fixo de período p .

A determinação do valor de x^* pode ser dada considerando o ponto médio do intervalo X_n com um erro associado dado pela metade do tamanho de X_n .

Tabela 1: As 10 primeiras iterações obtidas como solução de $x_{n+1} = rx_n(1 - x_n)$, com $r = 3,6$ e $x_0 = 0,6$.

	<i>wxMaxima</i>	Nepomuceno (2014)		Rodrigues Junior e Nepomuceno (2015)	
n	x_n	$(x_n - \hat{x}_n^-)$	$(\hat{x}_n^+ - x_n)$	$(x_n - \hat{x}_n^-)$	$(\hat{x}_n^+ - x_n)$
0	0,6000000000000000	0	0	0	0
1	0,8640000000000000	0	0	0	0
2	0,4230144000000000	$9,8657 \times 10^{-16}$	$1,0118 \times 10^{-15}$	$9,8657 \times 10^{-16}$	$1,0118 \times 10^{-15}$
3	0,8786635826135040	$3,0292 \times 10^{-15}$	$2,9661 \times 10^{-15}$	$1,0308 \times 10^{-15}$	$9,6761 \times 10^{-16}$
4	0,3838100083283017	$1,0726 \times 10^{-14}$	$1,1311 \times 10^{-14}$	$1,7337 \times 10^{-15}$	$1,2687 \times 10^{-15}$
5	0,8513995890071905	$1,9403 \times 10^{-14}$	$2,0564 \times 10^{-14}$	$1,4178 \times 10^{-15}$	$1,5797 \times 10^{-15}$
6	0,4554659838440795	$7,3478 \times 10^{-14}$	$7,1516 \times 10^{-14}$	$3,4793 \times 10^{-15}$	$3,5151 \times 10^{-15}$
7	0,8928601970580872	$9,6314 \times 10^{-14}$	$9,5826 \times 10^{-14}$	$1,2073 \times 10^{-15}$	$7,9103 \times 10^{-16}$
8	0,3443791160429314	$3,6635 \times 10^{-13}$	$3,6867 \times 10^{-13}$	$3,3653 \times 10^{-15}$	$3,6291 \times 10^{-15}$
9	0,8128157057151141	$7,7902 \times 10^{-13}$	$7,9095 \times 10^{-13}$	$4,0874 \times 10^{-15}$	$4,9054 \times 10^{-15}$
10	0,5477268033286385	$2,5375 \times 10^{-12}$	$2,5334 \times 10^{-12}$	$1,0571 \times 10^{-14}$	$8,4140 \times 10^{-15}$

2.4 Intlab

Neste trabalho, os métodos descritos em Nepomuceno (2014) e Rodrigues Junior e Nepomuceno (2015) são comparados com os resultados do *Intlab* pelos tamanhos dos intervalos finais obtidos e o número de iterações para atingirem diferentes pontos fixos.

O *Intlab* é um *toolbox* do *Matlab* designado para realizar operações com intervalos reais e complexos. O conceito do *Intlab* se divide em um biblioteca rápida da aritmética intervalar e em um ambiente interativo de programação para operações intervalares de fácil acesso (Rump, 1999).

A definição de um valor intervalar no *toolbox* pode ser realizada através das funções *intval* ou *infsup*. A função *infsup*(x^-, x^+) permite que o usuário entre com os limites inferior x^- e superior x^+ do intervalo que desejar. Já a função *intval*(x_0), utilizada neste trabalho, gera um intervalo baseado nos intervalos de representação da máquina, ou seja, a partir de um ponto x_0 , a função usa o maior e menor pontos flutuantes mais próximos a x_0 para formar um intervalo de modo que $x_0 \in \text{intval}(x_0)$.

3 Metodologia

A fim de verificar a coerência dos métodos estudados, calculam-se, através do *Matlab*, as trajetórias das órbitas do mapa logístico, dado por $x_{n+1} = rx_n(1 - x_n)$, com $r = 3,6$ e $x_0 = 0,6$ para cada método. Os resultados obtidos pelas 10 primeiras iterações com cálculos de cerca de 16 dígitos de precisão são comparados com cálculos simbólicos com precisão de 100 dígitos, obtidos pelo software *wxMaxima*.

A Tabela 1 apresenta os resultados x_n obtidos pelo *wxMaxima* além dos intervalos obtidos pelos dois métodos a cada iteração n . O termo $(x_n - \hat{x}_n^-)$ representa a diferença entre o resultado encontrado através do cálculo simbólico e o limite inferior do intervalo. Já $(\hat{x}_n^+ - x_n)$ é a diferença entre o limite superior do intervalo dos métodos e

o resultado com precisão de 100 dígitos. Se algum desses valores for negativo, significa que o termo x_n não está contido no intervalo $[\hat{x}_n^-, \hat{x}_n^+]$.

Como nenhum valor apresentado na Tabela 1 é negativo, pode-se confirmar que a órbita obtida pelo cálculo com maior precisão está contida dentro dos intervalos das órbitas de cada método. Isso é uma indicação de que ambas as estratégias adotadas por Nepomuceno (2014) e Rodrigues Junior e Nepomuceno (2015) são coerentes.

Tanto em Nepomuceno (2014) quanto em Rodrigues Junior e Nepomuceno (2015), a função recursiva usada para ilustrar a técnica de computação por intervalos na eliminação do erro computacional no cálculo do ponto fixo foi a função do mapa logístico, desenvolvida por (May, 1976) como um modelo populacional, dada por:

$$x_{n+1} = rx_n(1 - x_n). \quad (4)$$

Na equação (4), x_n é um número entre zero e um que representa a razão entre a população existente na n -ésima iteração e o maior número possível de indivíduos; e r é a taxa de crescimento da população. Para que x_n esteja sempre no intervalo indicado, o maior valor admissível para a taxa de crescimento r é 4.

À medida em que o parâmetro r se aproxima de 4, a trajetória da órbita do mapa logístico se torna mais instável, levando o sistema a apresentar comportamento caótico (Feigenbaum, 1978; Wang et al., 2004).

Quando se trabalha com intervalo em mapas discretos, é necessário que seu tamanho nunca seja nulo, de modo a evitar que o intervalo seja tratado como um ponto, tornando inútil a análise intervalar. Para que isso não ocorra, deve se observar a posição dos pontos limites dos intervalos no *cobweb plot*.

O *cobweb plot* é um procedimento gráfico usado para analisar o comportamento qualitativo de funções recursivas unidimensionais. Esse tipo de gráfico, apresentado na Figura 1, permite determinar a evolução de uma condição inicial sobre várias iterações (Tomida, 2008).

Tabela 2: Comparação dos cálculos para os três casos entre *Intlab*, Nepomuceno (2014) e Rodrigues Junior e Nepomuceno (2015).

Caso	<i>Intlab</i>			Nepomuceno (2014)			R. Junior e Nepomuceno (2015)		
	Período	Tamanho	n	Período	Tamanho	n	Período	Tamanho	n
1	2	$1,83 \times 10^{-6}$	20	2	$1,46 \times 10^{-8}$	27	2	$6,66 \times 10^{-16}$	55
2	4	$3,22 \times 10^{-4}$	23	4	$7,49 \times 10^{-5}$	33	4	$1,11 \times 10^{-15}$	207
3	8	$4,57 \times 10^{-4}$	23	8	$2,15 \times 10^{-3}$	35	8	$1,15 \times 10^{-14}$	302

Na intenção de que um intervalo sempre seja assegurado no mapa logístico para o método adaptado de Rodrigues Junior e Nepomuceno (2015), os arredondamentos realizados em cada iteração devem seguir uma das três situações:

- Limite superior x_n^+ menor do que 0,5: x_{n+1}^- deve ser arredondado para baixo e x_{n+1}^+ deve ser arredondado para cima;
- Limite inferior x_n^- maior do que 0,5: x_{n+1}^- deve ser arredondado para cima e x_{n+1}^+ deve ser arredondado para baixo;
- Limites inferior x_n^- e superior x_n^+ entre 0,5: ambos devem ser arredondados para baixo.

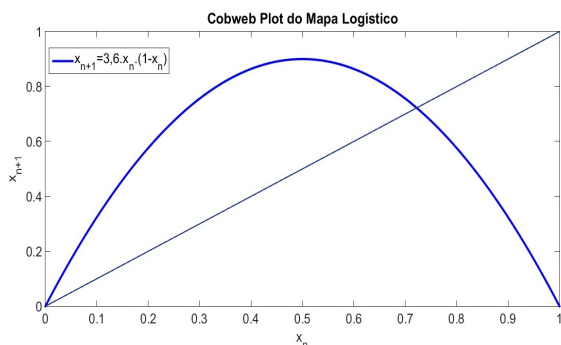


Figura 1: *Cobweb plot* do mapa logístico para o parâmetro $r = 3,6$.

Baseado em Feigenbaum (1978), três casos de pontos fixos de (4) serão analisados:

- Caso 1: $r = 330/100$ e $x_0 = 0,6$, cuja órbita deve convergir para ponto fixo de período 2;
- Caso 2: $r = 347/100$ e $x_0 = 0,6$, cuja órbita deve convergir para ponto fixo de período 4;
- Caso 3: $r = 355/100$ e $x_0 = 0,6$, cuja órbita deve convergir para ponto fixo de período 8;

4 Resultados

O resultado da aplicação dos métodos abordados em Nepomuceno (2014) e em Rodrigues Junior e Nepomuceno (2015) estão apresentados na Tabela 2, onde são descritos o período do ponto fixo calculado, o tamanho do intervalo final obtido e

o número de iterações necessárias por cada método para alcançar o ponto fixo. Para efeito de comparação, a Tabela 2 também apresenta os resultados obtidos por meio da aritmética intervalar tradicional encontrados através do toolbox *Intlab*.

Pela análise da Tabela 2 pode-se observar que, apesar de alcançarem pontos fixos de mesmo período, existem diferenças consideráveis com relação ao número de iterações necessárias para atingi-los e ao tamanho dos intervalos finais obtidos em cada caso pelos métodos analisados.

Outra observação que pode ser feita é que à medida em que se aumentou o parâmetro r da equação (4), o tamanho dos intervalos finais também aumentou em todos os métodos.

O tamanho dos intervalos cresce devido ao fato de que quanto maior o valor do parâmetro r , maior a inclinação da curva da Figura 1. Consequentemente, a distância entre as saídas obtidas a partir de dois pontos muito próximos, também será maior.

Pela análise das simulações, todos os métodos atingiram os pontos fixos estipulados de acordo com Feigenbaum (1978).

5 Conclusão

De acordo com os resultados obtidos, todos os métodos são eficazes para atingir os pontos fixos dos casos abordados no trabalho, porém cada um possui características distintas.

Pela análise da Tabela 2, pode-se observar que apesar do método desenvolvido por Nepomuceno (2014) calcular os pontos fixos mais rapidamente em relação ao de Rodrigues Junior e Nepomuceno (2015), seus resultados apresentam menor precisão, uma vez que o tamanho do intervalo final obtido chega a cerca de 10^{-11} maior. Além disso, pela análise da Tabela 1, pode-se observar que os intervalos em Nepomuceno (2014) apresentam taxa de crescimento maior em relação a Rodrigues Junior e Nepomuceno (2015).

Fazendo a comparação do método de Nepomuceno (2014) com os resultados obtidos pelo *Intlab*, pode ser observado que o número de iterações necessárias para alcançar o ponto fixo com Nepomuceno (2014) é maior para todos os casos. Porém, nos casos 1 e 2, os intervalos obtidos para esse método são mais precisos em relação aos resultados do *Intlab*.

A metodologia desenvolvida por Rodrigues Junior e Nepomuceno (2015) apresentou os melhores resultados em relação aos demais métodos. Mesmo necessitando de números de iterações muito maiores para todos os casos, os intervalos finais obtidos com esse método foram muito menores em relação ao método de Nepomuceno (2014) e à aritmética intervalar do *Intlab*.

Por exemplo, no caso 3, o número de iterações para alcançar o ponto fixo de período 8 pelo método de Rodrigues Junior e Nepomuceno (2015) foi aproximadamente 13 vezes o número com o uso do *Intlab*. Porém, o intervalo final encontrado para esse método foi cerca de 4×10^{10} vezes menor do que o correspondente obtido pelo *Intlab*.

Por fim, pretende-se em trabalhos futuros investigar esses métodos na construção de diagramas de bifurcação para o mapa logístico e trabalhar com outros mapas.

Agradecimentos

Agradecemos à CAPES, CNPq/INERGE, FAPEMIG e à Universidade Federal de São João del-Rei pelo apoio.

Referências

- Devaney, R. L., Siegel, P. B., Mallinckrodt, A. J. et al. (1993). A first course in chaotic dynamical systems: theory and experiment, *Computers in Physics* **7**(4): 416–417.
- Feigenbaum, M. J. (1978). Quantitative Universality for a Class of Non-linear Transformations, *Journal Of Statistical Physics* **19**(1): 25–52.
- 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.
- Grigoletti, P. S., Dimuro, G. P., Barboza, L. V. e Reiser, R. H. S. (2006). Análise intervalar de circuitos elétricos.
- Hammel, S., Yorke, J. e Grebogi, C. (1987). Do numerical orbits of chaotic dynamical processes represent true orbits?, *Journal of Complexity* **3**(2): 136–145.
- Hickey, T., Ju, Q. e Van Emden, M. H. (2001). Interval arithmetic: From principles to implementation, *Journal of the ACM (JACM)* **48**(5): 1038–1068.
- Kulisch, U. (2012). *Advanced arithmetic for the digital computer: design of arithmetic units*, Springer Science & Business Media.
- Lozi, R. (2013). Can we trust in numerical computations of chaotic solutions of dynamical systems?, *Topology and Dynamics of Chaos: In Celebration of Robert Gilmore's 70th Birthday. Edited by Letellier Christophe. Published by World Scientific Publishing Co. Pte. Ltd., 2013. pp. 63–98.*, Vol. 1.
- May, R. M. (1976). Simple mathematical models with very complicated dynamics, *Nature* **261**: 459–467.
- Mendonça, A. F. e Campos, M. A. (2013). Confiabilidade autovalidável de sistemas com processo exponencial de falhas, *TEMA (São Carlos)* **14**(3): 383–398.
- Moore, R. E. (1979). *Methods and applications of interval analysis*, Vol. 2, SIAM.
- Nepomuceno, E. G. (2014). Convergence of recursive functions on computers, *The Journal of Engineering* pp. 1–3.
- Nepomuceno, E. G. e 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.
- Overton, M. L. (2001). *Numerical Computing with IEEE floating point arithmetic*, SIAM.
- Pereira, L. E. S. e Costa, V. M. d. (2012). Análise de incertezas em sistemas de energia elétrica via fluxo de potência intervalar, *Anais do XIX CBA - Congresso Brasileiro de Autômática*.
- Rodrigues Junior, H. M. e Nepomuceno, E. G. (2015). Uso da computação por intervalos para cálculo de ponto fixo de um mapa discreto., *Anais do DINCON 2015 - Conferência Brasileira de Dinâmica, Controle e Aplicações*.
- Rothwell, E. J. e Cloud, M. J. (2012). Automatic error analysis using intervals, *Education, IEEE Transactions on* **55**(1): 9–15.
- Ruetsch, G. (2005). An interval algorithm for multi-objective optimization, *Structural and Multidisciplinary Optimization* **30**(1): 27–37.
- Rump, S. (1999). INTLAB - INTerval LABoratory, in T. Csendes (ed.), *Developments in Reliable Computing*, Kluwer Academic Publishers, Dordrecht, pp. 77–104.
- Tomida, A. G. (2008). Matlab toolbox and GUI for analyzing one-dimensional chaotic maps, *Proceedings - The International Conference on Computational Sciences and its Applications, ICCSA 2008* pp. 321–330.

Wang, S., Liu, W., Lu, H., Kuang, J. e Hu, G. (2004). Periodicity of chaotic trajectories in realizations of finite computer precisions and its implication in chaos communications, *International journal of modern physics B* **18**(17n19): 2617–2622.