

Solução Intervalar para Ajuste de Curvas Usando o Módulo Python-XSC

M. H. F. MARCONE¹, F. T. SANTANA^{2*}, C. V. M. PEREIRA³ e A. B. MOREIRA³

Recebido em 10 de dezembro de 2018 / Aceito em 6 e fevereiro de 2021

RESUMO. Este trabalho propõem o uso da Matemática Intervalar em conjunto com o Método dos Mínimos Quadrados e a linguagem Python para obter a melhor solução aproximada para sistemas intervalares obtidos de experimentos físicos. Este método de resolução será aplicado para obter a melhor função aproximada que ajusta um conjunto de dados oriundos de um experimento físico, no qual um carro se desloca com aceleração constante sob um trilho de ar inclinado. Ao fazer esse tipo de abordagem usando a Matemática Intervalar, busca-se inferir como as incertezas provenientes do experimento, assim como os erros gerados pelas representações e operações numéricas em computadores, interferem no resultado obtido. Para isso, se fez necessário a utilização da biblioteca Python for Extended Scientific Computing (Python-XSC), a qual é baseada na estrutura da aritmética intervalar e fornece funções para a resolução de sistemas lineares intervalares. A aplicação do estudo feito se mostrou bastante eficiente e de fácil utilização, o que motiva sua utilização.

Palavras-chave: aproximação de funções, matemática intervalar, mínimos quadrados, Python, PYXSC.

1 INTRODUÇÃO

As ciências experimentais, como a Física e a Engenharia, lidam constantemente com dados numéricos oriundos de práticas experimentais que trazem uma incerteza inerente do próprio processo de medição. Esses dados, geralmente associados a grandezas físicas, devem ser claramente interpretados e, muitas vezes, essas medidas ficam sujeitas a erros que não podem ser eliminados [13]. Por isso, neste contexto, é comum o aparecimento de sistemas lineares inconsistentes e a necessidade de se buscar a melhor solução aproximada através do método dos mínimos quadrados.

*Autor correspondente: Fabiana Tristão Santana – E-mail: fabianatsantana@gmail.com

¹Escola de Ciências e Tecnologia, Universidade Federal do Rio Grande do Norte, 59072-970, Natal, RN, Brasil – E-mail: marcosmarcone48@gmail.com <https://orcid.org/0000-0002-3467-472X>

²Escola de Ciências e Tecnologia, Universidade Federal do Rio Grande do Norte, 59072-970, Natal, RN, Brasil – E-mail: fabianatsantana@gmail.com <https://orcid.org/0000-0002-9204-4769>

³Escola de Ciências e Tecnologia, Universidade Federal do Rio Grande do Norte, 59072-970, Natal, RN, Brasil – E-mail: caiomp2010@hotmail.com <https://orcid.org/0000-0003-0668-5234>

³Escola de Ciências e Tecnologia, Universidade Federal do Rio Grande do Norte, 59072-970, Natal, RN, Brasil – E-mail: abessa@ect.ufrn.br <https://orcid.org/0000-0003-1211-3697>

Segundo [15], a incerteza no valor de uma grandeza x indica o quanto ela pode ser diferente do valor verdadeiro. Neste estudo, fundamentado pela Teoria de Erros, é comum utilizar probabilidades e a função gaussiana, também conhecida por função normal de erros, para descrever erros experimentais. Alguns softwares, como o Pasco Capstone, utilizados para lidar com dados experimentais já são implementados levando em conta esse contexto e utilizam uma aritmética apropriada.

Neste trabalho a Matemática Intervalar, introduzida por Moore [11], é utilizada para tratar a incerteza numérica na obtenção e processamento dos dados experimentais e, com isso, os coeficientes e termos independentes do sistema linear que modela o problema serão todos intervalares.

A melhor solução intervalar para o sistema será obtida com o módulo computacional de resolução de sistemas lineares com coeficientes intervalares, apresentado em [6], denominado Python for Extended Scientific Computing (Python-XSC). Este módulo se mostrou eficiente para encontrar a melhor função intervalar que ajusta dados obtidos em práticas experimentais.

Atualmente, existem várias bibliotecas para Matemática Intervalar implementadas nas mais diversas linguagens. Entretanto, segundo [3], estas bibliotecas, em geral, não possuem como objetivo serem utilizadas em práticas de ensino, muito menos serem de fácil aprendizado para o desenvolvimento de aplicações. Por exemplo, os módulos mais conhecidos nessa área são pagos, como o Intlab [14] (biblioteca de Matemática Intervalar para o Matlab), o que dificulta o acesso.

Por outro lado, o módulo PyInterval é licenciado pela GPL, ou seja, tem seu código aberto e é distribuído como software livre. Este módulo, apresentado em [6], garante excelentes resultados por ter sido desenvolvido com arredondamentos direcionados, isto é, o resultado exato sempre está no intervalo obtido.

Como é ressaltado em [6], o PyInterval foi desenvolvido permitindo expansões futuras de acordo com a necessidade dos usuários. Por exemplo, em [5], o módulo foi usado na análise intervalar de circuitos elétricos e aqui será utilizado para obter a melhor solução aproximada, pelo processo de mínimos quadrados, para um problema físico de estimar a função horária de um carro que se movimenta em um tubo de ar com a aceleração constante.

Este artigo está organizado da seguinte forma. Na Seção 2 é apresentada o método dos Mínimos Quadrados utilizando a Álgebra Linear. Os principais conceitos de vetores e matrizes intervalares são apresentados na Seção 3. Além disso, nessa seção será definido o sistema intervalar e além da solução utilizando o módulo PyInterval, também será exposto um método de solução composto pelas soluções dos sistemas formados pelas extremidades de cada intervalo. Na Seção 4, serão apresentados dados de alguns experimentos e as soluções serão obtidas e analisadas. As considerações finais são apresentadas na Seção 5.

2 A BIBLIOTECA PYTHON FOR EXTENDED SCIENTIFIC COMPUTING (PYTHON-XSC)

Atualmente, existem várias bibliotecas para Matemática Intervalar implementadas nas mais diversas linguagens. Entretanto, segundo [3], muitas não tem por objetivo serem utilizadas em práticas de ensino e não são de fácil aprendizado, o que dificulta sua utilização nas aplicações [14].

Já o módulo Python for Extended Scientific Computing (Python-XSC) [6] foi desenvolvido para ser mais acessível. Ele é um conjunto desenvolvido em Python, utilizando o paradigma da Programação Orientada a Objetos (POO), que implementa classes básicas para a manipulação de intervalos utilizando a Matemática Intervalar [1, 11, 12]

A arquitetura do módulo Python-XSC é composta pelas seguintes classes: (i) xscGlobals, (ii) eFloat, (iii) interval, (iv) iMatrix, (v) iEquationSolve e (vi) erro (Figura 1).

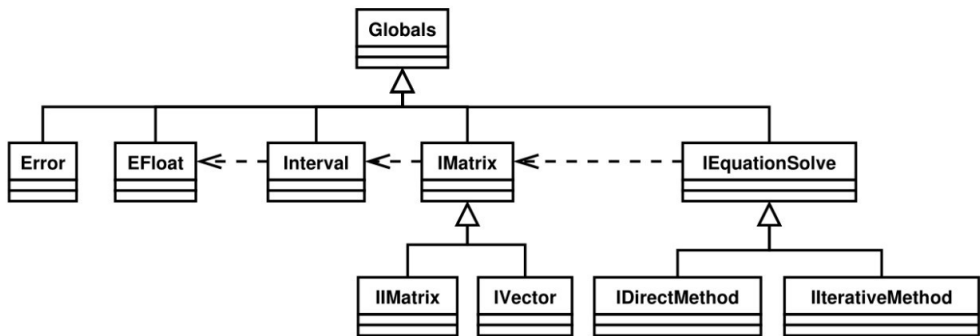


Figura 1: Diagrama de classes da biblioteca Python-XSC. Fonte [6].

É importante ressaltar que o módulo Python-XSC é licenciado pela GPL, ou seja, tem o código aberto e é distribuído como software livre. Assim como, o Python-XSC é portátil e multiplataforma, características estas herdadas da linguagem Python.

Neste trabalho o Python-XSC é utilizado para representar os intervalos, as matrizes intervalares e para resolver um sistema linear intervalar proveniente da aplicação do método dos mínimos quadrados com o objetivo de obter o melhor ajuste intervalar para a curva que descreve o movimento retilíneo uniforme. Os arquivos necessários para executar o método discutido aqui podem ser acessados em <https://url.gratis/X1zGk>.

3 ABORDAGEM MATRICIAL PARA O MÉTODO DE MÍNIMOS QUADRADOS

O processo de Mínimos Quadrados fornece a melhor solução aproximada para sistemas lineares inconsistentes [2]. Existem algumas formas diferentes de abordar e desenvolver esse método, por exemplo, neste trabalho será utilizada a Álgebra Linear.

Pela Álgebra Linear, um sistema $A \vec{x} = \vec{b}$ onde \vec{x} é o vetor coluna constituído pelas incógnitas reais x_1, x_2, \dots, x_n é consistente se, e somente se, o vetor coluna \vec{b} de termos independentes é combinação linear dos vetores coluna de A , por sua vez, constituídos pelos coeficientes das incógnitas [2]. Supondo que W é o espaço-coluna de A , o sistema $A \vec{x} = \vec{b}$ é inconsistente quando $\vec{b} \notin W$. Neste caso, a melhor solução para o problema, conhecida como solução por Mínimos Quadrados, é obtida substituindo \vec{b} pelo vetor de W mais próximo de \vec{b} . Este vetor é exatamente a projeção de \vec{b} em W aqui denotada por $proj_W \vec{b}$.

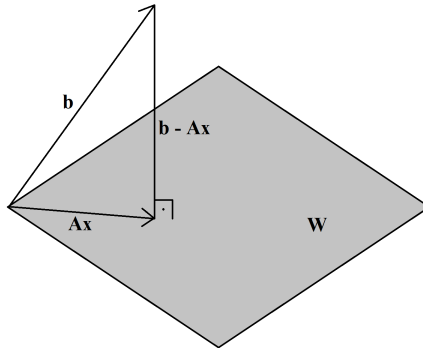


Figura 2: $A \vec{x} = proj_W \vec{b}$

O espaço-coluna de A e o espaço-nulo de A^T são complementos ortogonais e como $\vec{b} - A \vec{x}$ é ortogonal à W (Figura 2), segue que $\vec{b} - A \vec{x}$ satisfaz $A^T (\vec{b} - A \vec{x}) = \vec{0}$, isto é

$$A^T (\vec{b} - A \vec{x}) = \vec{0} \Leftrightarrow A^T \vec{x} = A^T \vec{b} . \tag{3.1}$$

O sistema (3.1), conhecido como sistema auxiliar, é consistente e sua única solução $\vec{x} = (A^T A)^{-1} A^T \vec{b}$ é também a melhor solução aproximada para $A \vec{x} = \vec{b}$.

Considerando a incerteza nos dados numéricos, [7] define o vetor intervalar

$$X = \left[[x_1, \bar{x}_1] \quad \dots \quad [x_n, \bar{x}_n] \right]^T , \tag{3.2}$$

cujos termos são incógnitas, a matriz intervalar

$$A = (A_{ij})_{m \times n} = \begin{bmatrix} [A_{11}, \bar{A}_{11}] & \dots & [A_{1n}, \bar{A}_{1n}] \\ \vdots & & \vdots \\ [A_{m1}, \bar{A}_{m1}] & \dots & [A_{mn}, \bar{A}_{mn}] \end{bmatrix} , \tag{3.3}$$

onde os termos das m linhas e n colunas são os coeficientes das incógnitas e o vetor intervalar

$$B = \left[[b_1, \bar{b}_1] \quad \dots \quad [b_m, \bar{b}_m] \right]^T , \tag{3.4}$$



Figura 3: Trilho de ar inclinado.

formado pelos termos independentes do sistema.

Utilizando tais termos, [6] desenvolve uma biblioteca livre com a linguagem de programação Python, denominada biblioteca Python-XSC, com a qual é possível obter a solução do sistema intervalar $AX = B$, pelo método dos mínimos quadrados.

Para habilitar a biblioteca PyInterval, primeiramente deve-se utilizar o comando *from pyxsc import* seguido de asterisco. As funções *imatrix()* e *ivector()* definem os termos do sistema intervalar $AX = B$ e a transposta da matriz A é obtida com a função *getTranspose()*. Esta função é utilizada para definir os termos do novo sistema $A^TAX = A^TB$, que fornece a solução por Mínimos Quadrados com a função *iterativeMethod()* para A^TA e A^TB .

4 AJUSTES DE FUNÇÕES COM BIBLIOTECA PYINTERVAL

Para aplicar o método de solução com a biblioteca Python-XSC foi realizado um experimento no Laboratório de Física da Escola de Ciências e Tecnologia da Universidade Federal do Rio Grande do Norte que consistiu em registrar o tempo de percurso e o deslocamento, denotados por t e $x(t)$, de um carro que se movia livremente sobre um trilho de ar inclinado (Figura 3). A partir dos dados numéricos coletados, o objetivo era determinar o ajuste da função horária do espaço $s(t) = s_0 + v_0t + at^2/2$, onde s_0 , v_0 e a são, respectivamente, a posição inicial, a velocidade inicial e a aceleração do carro.

Foram realizados três experimentos correspondentes a inclinações diferentes para do trilho. Em cada um deles foram registrados dados do tipo $(t, x(t))$, correspondentes aos eventos “deslocamento do carro sobre o trilho”. Os registros das informações foram feitos considerando uma incerteza de 0.001 em ambas as coordenadas, o que resultou nos dados intervalares $([\underline{t}, \bar{t}], [\underline{x}, \bar{x}])$.

EXPERIMENTO I: $([0.080, 0.082], [0.359, 0.361])$, $([0.157, 0.159], [0.409, 0.411])$, $([0.229, 0.231], [0.459, 0.461])$, $([0.302, 0.304], [0.509, 0.511])$, $([0.425, 0.427], [0.609, 0.611])$, $([0.537, 0.539], [0.709, 0.711])$, $([0.726, 0.728], [0.909, 0.911])$, $([0.870, 0.872], [1.109, 1.111])$, $([1.04, 1.042], [1.309, 1.311])$ e $([1.174, 1.176], [1.509, 1.511])$.

Na Figura 4 pode-se observar a biblioteca intervalar sendo habilitada com o comando `from pyxsc import*`. Em seguida as matrizes intervalares do sistema $AX = B$ são definidas. Tais termos, por sua vez, são usados na definição do sistema auxiliar $A^T AX = A^T B$ e, por fim, a função `iterativeMethod(.)` fornece a solução intervalar:

$$X(T) = [-0.965469, 1.572287]t^2 + [-1.273477, 1.572287]t + [-0.066763, 0.699266]. \tag{4.1}$$

A título de validação, utilizou-se os pontos médios dos dados experimentais com o objetivo de verificar se os coeficientes da função real de ajuste para esses dados estão nos respectivos intervalos da solução intervalar. A implementação para este caso real foi feita no software GeoGebra, utilizando o método dos mínimos quadrados, de forma semelhante ao que foi desenvolvido em [9, 10]. O sistema $A_m x_m = B_m$, obtido substituindo os valores numéricos na função $x(t) = bt^2 + v_0t + s_0$, onde $b = \frac{a}{2}$ é:

$$\left\{ \begin{array}{l} 0.006561b + 0.081v_0 + s_0 = 0.36 \\ 0.024964b + 0.158v_0 + s_0 = 0.41 \\ 0.0529 + 0.23v_0 + s_0 = 0.46 \\ 0.091809b + 0.303v_0 + s_0 = 0.51 \\ 0.181476b + 0.426v_0 + s_0 = 0.61 \\ 0.289444b + 0.538v_0 + s_0 = 0.71 \\ 0.528529b + 0.727v_0 + s_0 = 0.91 \\ 0.758641b + 0.871v_0 + s_0 = 1.11 \\ 1.083681b + 1.041v_0 + s_0 = 1.31 \\ 1.380625b + 1.175v_0 + s_0 = 1.51 \end{array} \right. \Rightarrow \underbrace{\begin{bmatrix} 0.006561 & 0.081 & 1 \\ 0.024964 & 0.158 & 1 \\ 0.0529 & 0.23 & 1 \\ 0.091809 & 0.303 & 1 \\ 0.181476 & 0.426 & 1 \\ 0.289444 & 0.538 & 1 \\ 0.528529 & 0.727 & 1 \\ 0.758641 & 0.871 & 1 \\ 1.083681 & 1.041 & 1 \\ 1.380625 & 1.175 & 1 \end{bmatrix}}_{A_m} \underbrace{\begin{bmatrix} b \\ v_0 \\ s_0 \end{bmatrix}}_{x_m} = \underbrace{\begin{bmatrix} 0.36 \\ 0.41 \\ 0.46 \\ 0.51 \\ 0.61 \\ 0.71 \\ 0.91 \\ 1.11 \\ 1.31 \\ 1.51 \end{bmatrix}}_{B_m} \tag{4.2}$$

Para implementar o sistema (4.2) na Janela CAS do GeoGebra, primeiramente definiu-se os pontos $P_i := (t_i, s(t_i))$ e os vetores $p_i := (t_i, s(t_i))$, para $i = 1, \dots, 10$. Em seguida $A := \{\{\text{Elemento}(p1,1,1)^2, \text{Elemento}(p1,1,1), 1\}, \{\text{Elemento}(p2,1,1)^2, \text{Elemento}(p2,1,1), 1\}, \dots, \{\text{Elemento}(p10,1,1)^2, \text{Elemento}(p10,1,1), 1\}\}$ foi definida como a matriz de coeficientes, $B := \{\{\text{Elemento}(p1,2,1)\}, \{\text{Elemento}(p2,1,1)\}, \dots, \{\text{Elemento}(p10,1,1)\}\}$ como a matriz de termos independentes e A^T foi definida por $M := \text{MatrizTransposta}(A)$.

```

1 from pyxsc import*
2 #Matriz A
3
4 A=imatriz([[ [0.0064,0.006724], [0.080,0.082], [1,1]],
5             [ [0.02465,0.02528], [0.157,0.159], [1,1]],
6             [ [0.05244,0.05336], [0.229,0.231], [1,1]],
7             [ [0.09120,0.09242], [0.302,0.304], [1,1]],
8             [ [0.18062,0.18233], [0.425,0.427], [1,1]],
9             [ [0.28837,0.29052], [0.537,0.539], [1,1]],
10            [ [0.52708,0.52998], [0.726,0.728], [1,1]],
11            [ [0.75690,0.76038], [0.870,0.872], [1,1]],
12            [ [1.08160,1.08576], [1.040,1.042], [1,1]],
13            [ [1.37828,1.38298], [1.174,1.176], [1,1]])]
14
15 # Matriz b
16
17 b = ivector([ [0.359,0.361],
18              [0.409,0.411],
19              [0.459,0.461],
20              [0.509,0.511],
21              [0.609,0.611],
22              [0.709,0.711],
23              [0.909,0.911],
24              [1.109,1.111],
25              [1.309,1.311],
26              [1.509,1.511]])
27
28 #Transposta de A
29 At = A.getTranspose()
30 #Aplicando o processo de mínimos quadrados
31 M = At*A
32 N = At*b
33 #Solucionando o sistema Mx=N
34 X = iIterativeMethod(M,N)
35 x.getXVector()

```

Figura 4: Resolução do sistema no software Python com método PyInterval.

Por fim, o comando $p:=\text{MatrizInversa}(M*A)*M*B$ fornece o vetor com os valores de b, v_0 e s_0 , com os quais define-se a solução:

$$x_m = 0.428744t^2 + 0.514575t + 0.315822, \quad (4.3)$$

cujos coeficientes estão contidos nos respectivos coeficientes da solução intervalar fornecida com a biblioteca Python.

EXPERIMENTO 2: $([0.210, 0.212], [0.249, 0.251]), ([0.390, 0.392], [0.299, 0.301]), ([0.526, 0.528], [0.349, 0.351]), ([0.709, 0.711], [0.399, 0.401]), ([0.917, 0.919], [0.499, 0.501]), ([1.079, 1.081], [0.599, 0.601]), ([1.392, 1.394], [0.799, 0.801]), ([1.638, 1.640], [0.999, 1.001]), ([1.877, 1.879], [1.199, 1.201])$ e $([2.070, 2.072], [1.399, 1.401])$.

A função intervalar de ajuste, obtida com a biblioteca Python, é:

$$X(T)=[-0.111972;0.560323]t^2+[-0.691658;0.741903]t+[-0.121523;0.566551], \quad (4.4)$$

e a função real de ajuste para os valores médios dos dados intervalares, obtida com o software GeoGebra, é:

$$x_m(t) = 0.216454t^2 + 0.120832t + 0.204502. \quad (4.5)$$

EXPERIMENTO 3: $([0.071, 0.073], [0.359, 0.361]), ([0.136, 0.138], [0.409, 0.411]), ([0.0.208, 0.210], [0.459, 0.461]), ([0.291, 0.210], [0.509, 0.511]), ([0.386, 0.388], [0.609, 0.611]), ([0.669, 0.671], [0.909, 0.911]), ([0.825, 0.827], [1.109, 1.111]), ([0.878, 0.880], [1.309, 1.311])$ e $([1.107, 1.109], [1.509, 1.511])$.

A função intervalar de ajuste, obtida com a biblioteca Python, é:

$$X(T)=[-1.197102;1.779465]t^2+[-1.178821;1.779465]t+[-0.076206;0.668844], \quad (4.6)$$

e a função real de ajuste para os valores médios dos dados intervalares, obtida com o software GeoGebra, é:

$$x_m(t) = 0.054117t^2 + 1.100512t + 0.208027. \quad (4.7)$$

Os coeficientes da solução intervalar, obtidos com a biblioteca PyInterval possuem uma amplitude relativamente pequena e contém os coeficientes das soluções dos sistemas lineares reais obtidos com o ponto médio dos intervalos amostrais de cada experimento.

5 CONCLUSÃO

Este trabalho chamou a atenção para a incerteza numérica presente em experimentos físicos e para a importante ferramenta computacional baseada em linguagem de programação Python, denominado módulo Python-XSC. Esta biblioteca usa a Matemática Intervalar, assim como toda a teoria de vetores e matrizes intervalares. Com o Python-XSC o método dos mínimos quadrados pôde ser aplicado para obter a melhor solução intervalar para o sistema inconsistente, obtido no processo de ajuste dos dados experimentais. A ferramenta foi aplicada em três experimentos, cujo objetivo era obter a função de ajuste de dados. Os resultados obtidos foram satisfatórios, pois os coeficientes intervalares obtidos tinham pequena amplitude e continham os coeficientes da função real de ajuste obtida com os valores médios dos dados intervalares. Problemas como este, de ajuste de dados, são muito frequentes em experimentos que lidam com dados numéricos e, por isso, o estudo de ferramentas como a biblioteca Python-XSC trazem significativas contribuições para a comunidade acadêmica e industrial.

ABSTRACT. In this work, we used the Interval Mathematics, Method of Least Squares and Python language to obtain the best approximate solution for interval systems obtained

from physical experiments. This resolution method will be applied to obtain the best approximate function that fits a data set from a physical experiment, in which a car travels with constant acceleration under an inclined air trail. When doing this type of approach using Interval Mathematics, we seek to infer how the uncertainties arising from the experiment, as well as the errors generated by the representations and operations on computer, interfere with the obtained result. For this, it was necessary to use the Python for Extended Scientific Computing (Python-XSC) library, which is based on the interval arithmetic structure and provides functions for the solving of interval linear systems. The application of the study was very efficient and easy to use, which motivates its use.

Keywords: function approximation, interval mathematics, least squares, Python, PYXSC.

REFERÊNCIAS

- [1] B. M. Acióly. “Fundamentação Computacional da Matemática Intervalar”. Tese (Doutorado em Ciência da Computação). Universidade Federal do Rio Grande do Sul, Porto Alegre, RS (1991).
- [2] H. Anton & C. Rorres. “Álgebra Linear com Aplicações”, 8. ed. Bookman, Porto Alegre (2001).
- [3] A. M. Dias. “Ambiente de Técnicas Intervalares” (ATI) Versão 2.0. Dissertação (Mestrado em Ciência da Computação). Universidade Católica de Pelotas, Pelotas, RS (2002).
- [4] GeoGebra. “O que é o GeoGebra” (2020). URL <https://www.geogebra.org/about?lang=pt-PT>. Acesso em: 29 jul. 2020.
- [5] P. S. Grigoletti et al. Análise intervalar de circuitos elétricos, *Trends in Applied and Computational Mathematics*, **7**(2) (2006), 287–296 .
- [6] P. S. Grigoletti, G. P. Dimuro & L. V. Barboza. Módulo python para matemática intervalar. *Trends in Applied and Computational Mathematics*, **8**(1) (2007), 73–82 .
- [7] E. Hansen & G. W. Walster. “Global Optimization Using Interval Analysis: revised and expanded”, Vol. 264, CRC Press (2003).
- [8] M. Hohenwarter & J. Hohenwarter. “Ajuda GeoGebra:manual oficial da versão 3.2” (2020). URL <https://wiki.geogebra.org/pt/Manual>. Acesso em: 13 out. 2020.
- [9] D. B. S. Lima, T. A. S. Almeida, E. C. Ribeiro & F. T. Santana. Soluções por Mínimos Quadrados Utilizando a Janela CAS do GeoGebra. In “Proceeding Series of the Brazilian Society of Computational and Applied Mathematics”, SBMAC (2018).
- [10] D. B. S. Lima, M. H. F. Marcone & F. T. Santana. Estudo de Sistemas Intervalares Aplicados em Problemas Físicos com Imprecisões Numéricas. In “Proceeding Series of the Brazilian Society of Computational and Applied Mathematics”, SBMAC (2018).
- [11] R. E. Moore. “Interval Analysis”. Prentice Hall, New Jersey (1966).
- [12] R. E. Moore. “Methods and Applications of Interval Analysis”. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (1979).

- [13] H. N. Nagashima. “Laboratório de Física I” (2020). URL <https://www.feis.unesp.br/Home/departamentos/fisicaequimica/relacaodedocentes973/apostila-lab-fisica-i----agosto-2011.pdf>. Acesso em: 13 de out. de 2020.
- [14] S. M. Rump. Interval computations with INTLAB. *Brazilian Electronic Journal on Mathematics of Computation* (BEJMC), [S.l.] (1999).
- [15] J. H. Vuolo. “Fundamentos da Teoria de Erros”, 2ed. Edgard Blücher, São Paulo (1996).

