Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 11, n. 1, 2025.

Trabalho apresentado no XLIII CNMAC, Centro de Convenções do Armação Resort - Porto de Galinhas - PE, 2024

Comparação entre os Métodos de Newton-Raphson, Secante e Müller para determinação de raízes de funções.

Ingryd Carolly de Oliveira Medeiros[1] Anderson Carlos da Silva Morais[2] Lucas Perdigão Soares[3] Matheus da Silva Menezes[4]
DCME/UFERSA, Mossoró, RN

O desafio na modelagem matemática não se resume apenas à criação do modelo mais abrangente, mas também à elaboração de um modelo simples e eficiente que incorpore as principais características do fenômeno em estudo [1]. A busca de raízes de funções polinomiais são exemplos de problemas matemáticos que podem ser resolvidos por métodos numéricos, oferecendo uma alternativa aos métodos analíticos, principalmente nos casos em que o polinômio possui grau mais elevado [2].

De acordo com [3], o estudo das raízes baseia-se no teorema de Bolzano sobre existência e unicidade, no qual afirma que se $f(x)$ é uma função contínua em um intervalo $[a, b]$ e $f(a) \cdot f(b) < 0$, então há pelo menos um ponto $c \in [a, b]$ que é raiz da função f. Já a unicidade é obtida por meio do corolário deste teorema, onde, se $f'(x)$ preserva o sinal no intervalo, essa raiz será única.

Segundo [4], o Método de Newton, equação (1), baseia-se na ideia de aproximar a função por uma reta tangente já que se trata de uma função derivada em um ponto inicial e, em seguida, encontrar a interseção dessa reta com o eixo x, que será a nova aproximação para a raiz da função. Já o Método da Secante, equação (2), é uma aproximação para o Método de Newton, sem a necessidade do cálculo da derivada, no qual as condições para a convergência do método são praticamente as mesmas.

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{1}$$

$$x_{k+1} = \frac{x_{k-1} f(x_k) - x_k f(x_{k-1})}{f(x_k) - f(x_{k-1})} \tag{2}$$

O Método de Müller (3) surge como alternativa aos métodos (1) e (2) para polinômios que possuam raízes complexas [3]. Esse método é utilizado para resolver qualquer problema de determinação de raiz, mas é especialmente eficaz na aproximação das raízes de polinômios. Ele é bastante semelhante ao método da secante, porém utiliza uma parábola para conectar três pontos e determinar a aproximação, ao invés de uma reta para ligar dois pontos.

$$x_{k+1} = x_k - \frac{2c}{b \pm (b)\sqrt{b^2 - 4ac}} \tag{3}$$

O objetivo deste estudo é comparar as soluções encontradas pelos métodos definidos por (1), (2) e (3). A fim de avaliar a eficácia de cada método, serão analisadas a quantidade de iterações, o tempo de processamento e a convergência das raízes das funções. Foram utilizadas as funções (4), (5), (6) e (7). As funções, os intervalos e os valores iniciais (x_0) utilizados foram:

[1] ingryd.medeiros@alunos.ufersa.edu.br
[2] anderson.morais92395@alunos.ufersa.edu.br
[3] lucas.soares@alunos.ufersa.edu.br
[4] matheus@ufersa.edu.br

$$f_1(x) = 2x^4 + 4x^3 + 3x^2 - 10x - 15 = 0, \quad I = [-2, -1] \text{ e } x_0 = -1.5 \tag{4}$$

$$f_2(x) = x^5 - 2x^4 - 9x^3 + 22x^2 + 4x - 24 = 0, \quad I = [-3, 0] \text{ e } x_0 = -1.5 \tag{5}$$

$$f_3(x) = 5x^3 + x^2 - e^{(1-2x)} + \cos(x) + 20 = 0, \quad I = [-1, 0] \text{ e } x_0 = -1 \tag{6}$$

$$f_4(x) = x\sin(x) + 4 = 0, \quad I = [-1, -5] \text{ e } x_0 = -3 \tag{7}$$

As implementações foram desenvolvidas usando linguagem Python 3.10 e um computador com Processador Intel(R) Core(TM) i3-6006U CPU @ 2.00GHz, 4 GB RAM, Sistema Operacional Windows 10, com SSD de 256GB. Para a execução dos métodos numéricos foram considerados os critérios de parada $\varepsilon = 10^{-10}$, ou um limite de 300 iterações. Os resultados computacionais obtidos foram organizados na Tabela 1, apresentando, para cada método, a raiz aproximada ($\tilde{x}$), o número de iterações (It) e o tempo T (em segundos).

Tabela 1: Comparação entre as raízes calculadas pelos métodos numéricos

Funções	$\tilde{x}$ (Newton)	It	$T(s)$	$\tilde{x}$ (Secante)	It	$T(s)$	$\tilde{x}$ (Müller)	It	$T(s)$
$f_1(x)$	-1.300384	5	0.012	-1.300384	6	0.036	-1.300384	5	0.014
$f_2(x)$	-1.000000	6	0.022	-1.000000	8	0.025	Não convergiu	-	-
$f_3(x)$	-0.929561	6	0.016	-0.929561	7	0.100	-0.929561	8	0.054
$f_4(x)$	4.323240	4	0.014	4.323240	6	0.028	4.323240	6	0.064

Fonte: Autoria Própria.

Ao observar os resultados da Tabela 1, podemos verificar que, para todos os problemas considerados, o método de Newton-Raphson convergiu em um número menor iterações e com menor tempo, tornando-o mais eficiente quando comparado com os demais métodos. Com exceção ao método de Müller, que não convergiu para o problema f_2, os métodos apresentaram soluções convergentes para os problemas considerados, porém com desempenhos diferentes devido às suas abordagens distintas para o cálculo das raízes.

Referências

[1] L.C. Barroso e M. M. A. Barroso. **Cálculo Numérico (com aplicações)**. 2a. ed. São Paulo: Harbra, 1987.

[2] D. M. Flemming e M. B. Gonçalves. **Cálculo A: Funções, Limite, Derivação e Integração.** 6a Ed. São Paulo: Pearson Educación, 2007.

[3] N. B. Franco. **Cálculo Numérico.** 2a Ed. Pearson Prentice Hall, 2006.

[4] M.A.G. Ruggiero e V.L.R Lopes. **Cálculo Numérico: Aspectos Teóricos e Computacionais.** 2a Ed. São Paulo: Makron Books, 1997.