

Lista 1 - Cálculo Numérico

Cristian Diamantaras Vilela - 118109047

8 de maio de 2022

1 Questão 1

Link: <https://github.com/Nyon0k/CalculoNumerico/tree/main/Lista1>

2 Questão 2

A modificação já se encontra nos códigos do link acima.

3 Questão 3

No braço: Figura 1.

No código: Figura 2.

Resultado: Os dois convergem, pois $x_0 = 1$ é a raiz da função dada. No braço, por ser o caso da parada por iteração, para apenas depois das 4 iterações e em todas as iterações os x novos são iguais, justamente por já ser a raiz. No código, por ser o caso da tolerância, para por $x_0 = 1$ respeitar a condição $f(x_{\text{novo}}) \leq \text{tolerancia}$, já que o $f(x_{\text{novo}}) = 0$ e $0 \leq 10^{-7}$.

4 Questão 4

4.1

Sim, é possível. $f(x) = (x^3) - 2(x^2) - 11x + 12$ (converge). $f(x) = \sqrt{3}|x|$ (diverge).

Obs.: O código correspondente é o MNRMod1.py

4.2

O código correspondente é o MNRMod2.py

$$x_0 = 1 \quad f(x) = x^3 - 2x + 2 = 0$$

$$f'(x) = 3x^2 - 2$$

$$x_{m+1} = x_m - \frac{f(x_m)}{f'(x_m)}$$

1ª iter

$$x_0 = 1$$

$$x_1 = 1 - \frac{0}{1} = 1 \Rightarrow x_1 = 1$$

na próxima iteração $f(x_1)$ já é \leq ao erro 10^{-7}

2ª iter

$$x_1 = 1$$

$$x_2 = 1 - \frac{0}{1} = 1 \Rightarrow x_2 = 1$$

3ª iter

$$x_2 = 1$$

$$x_3 = 1 - \frac{0}{1} = 1 \Rightarrow x_3 = 1$$

4ª iter

$$x_3 = 1$$

$$x_4 = 1 - \frac{0}{1} = 1 \Rightarrow x_4 = 1$$

Figura 1: Questão 3 no papel.

```
1 import math as m
2 import numpy as np
3
4 dados = [] #salvar iterações e |f(Raiz)|
5
6 #Executa o cálculo da função de entrada
7 def f(x):
8     # f_result = (x*m.log10(x))-1
9     f_result = (x**3)-(2*x)+1 #código questão 3
10    return f_result
11
12 #Executa o cálculo da função derivada
13 def derivada(x):
14     # d_result = m.log10(x)+1/np.log(10)
15     d_result = 3*(x**2)-2 #código questão 3
16    return d_result
17
18 #Executa o método da bissecção
19 def newtonRaphson(x_inicial, err):
20    x = x_inicial
21    cont = 0
22    while True:
23        cont += 1
24        x_novo = x-(f(x)/derivada(x))
25        if abs(f(x_novo)) <= err:
26            dados.append(cont)
27            dados.append(abs(f(x_novo)))
28            return x_novo
29        x = x_novo
30
31 ##### variaveis pré-definidas #####
32 a, b = 2, 3 #intervalo [a,b]
33 x_inicial = 1
34 err = 1e-7 #tolerancia do erro
35 raiz = newtonRaphson(x_inicial, err)
36 print('Iterações: ' + str(dados[0]) + '\nRaiz: ' + str(raiz) + "\n|f(Raiz)|: " + str(dados[1]))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Python

```
PS C:\Users\crist> & python c:/Cristian/Computacao/CalcNum/20221/Lista1/MetodoNewtonRaphson.py
Iterações: 1
Raiz: 1.0
|f(Raiz)|: 0.0
PS C:\Users\crist>
```

Figura 2: Questão 3 no código.