E-fólio A | Folha de resolução para E-fólio

UNIDADE CURRICULAR: Computação Numérica

CÓDIGO: 21180

DOCENTES: Paulo Shirley (professor) e Filipe Pais (tutor)

A preencher pelo estudante

NOME: João Nunes

N.º DE ESTUDANTE: 2300321

CURSO: Licenciatura em Engenharia Informática

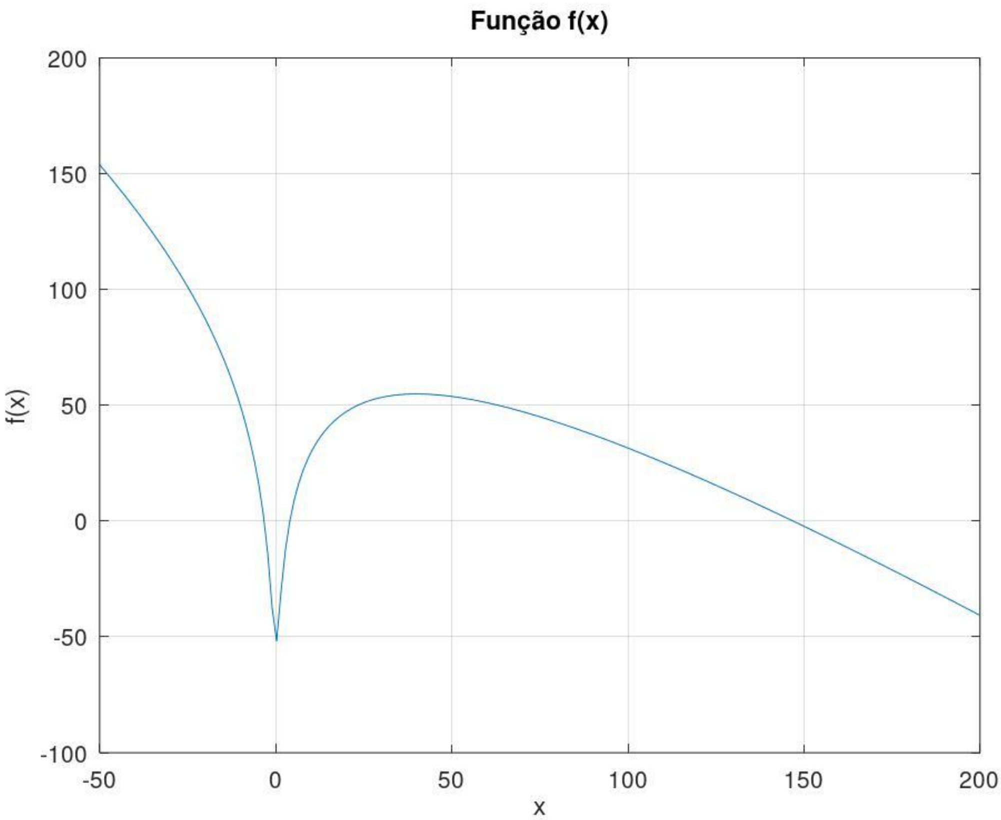
DATA DE ENTREGA: 25/11/2024

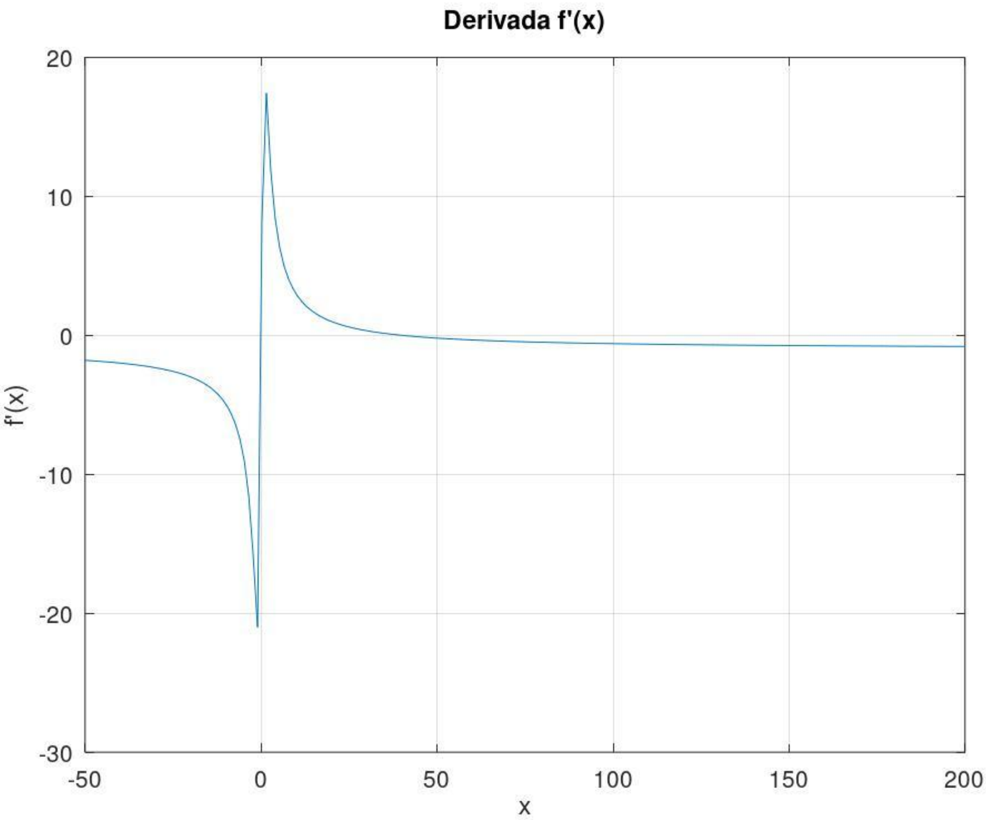
TRABALHO / RESOLUÇÃO:

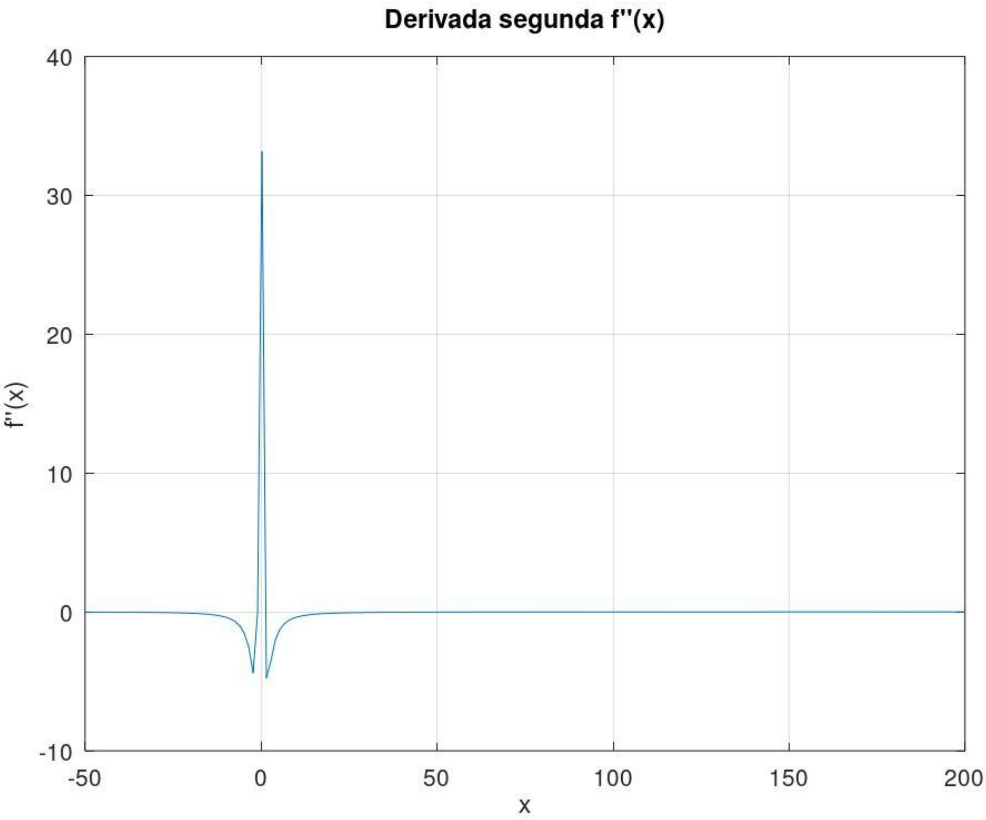
1. Tal como pedido, implementei as funções $f(x), f'(x), e\ f''(x)$ em ficheiros separados (**fx.m**, **dfx.m** e **d2fx.m**, respetivamente). Para a função $f(x)$, defini-a como:

   $$f(x) = -x + 20ln(1 + x^2) - a, com\ a = -4 + 20ln(17)$$

2. Utilizando o comando plot, o script **efa1.m** cria os gráficos das funções criados na alínea anterior no intervalo $[-50,200]$, com 200 pontos, título, etiqueta nas abcissas e grelha, sendo o resultado o seguinte:




Derivada f'(x)
f'(x)
x
20
10
0
-10
-20
-30
-50
0
50
100
150
200


Derivada segunda f''(x)
f''(x)
x
40
30
20
10
0
-10
-50
0
50
100
150
200

3. Para implementar o método de Newton com os argumentos de entrada e saída fornecidos no enunciado do efólioA, desenvolvi uma função que implementa o método para calcular raízes, retornando:
    a. A estimativa da raiz de r
    b. O erro absuluto e
    c. O número de iterações n
    d. O vetor com as iterações x

4. Para testar o funcionamento da função alg_newton(), usei dois valores iniciais ($x^0 = 1\ e\ x^0 = 10$), com erro máximo de 10^{-9}, e comparei os resultados obtidos com a teoria.
   Para os resultados obtidos com o script **efa2.m**:
    a. Para $\boldsymbol{x^0 = 1}$**:**

```
x0 = 1.0, Raiz: 4.000000000, Erro: 5.77e-15, Intervalo: [4.000000000, 4.000000000]
C teórico: 0.12341
k        xk                  Ck
0        1.000000000     0.10058
1        3.094806488     0.12593
2        3.896813084     0.12372
3        3.998682642     0.12341
4        3.999999786     0.12586
```

    b. Para $\boldsymbol{x^0 = 10}$

```
x0 = 10.0, Raiz: 146.936747689, Erro: 1.67e-11, Intervalo: [146.936747689, 146.936747689]
C teórico: 0.00127
k        xk                  Ck
0        10.000000000    0.00784
1        -0.011546786    0.00847
2        -36.028579380   0.00367
3        24.061356563    0.01321
4        -52.556280702   0.00275
5        37.387739189    0.07523
6        -755.844381769  0.00002
7        163.781786598   0.00106
8        147.237447206   0.00127
9        146.936862360   0.00127
```

Assim, para o x_0 = 1, o método convergiu para a raiz $\boldsymbol{r} = \mathbf{4.000000000}$, com erro absoluto muito pequeno ($\boldsymbol{\epsilon = 5.77 \times 10^{-15}}$). Inicialmente, C_k apresenta uma flutuação devia à inicial longe da raiz ($\boldsymbol{C_0 = 0.10058}$). A partir da terceira iteração (k = 2), e C_k estabiliza e converge para o valor teórico C, com apenas pequenas variações numéricas normais devido ao arredondamento.

Para o x_0 = 10, o método convergiu para a raiz $\boldsymbol{r = 146.936747689}$, com um erro absoluto pequeno ($\boldsymbol{\epsilon = 1.67} \times 10^{-11}$). Inicialmente, os valores C_k, são instáveis e apresentam variações significativas devido à estimativa inicial $x_0 = 10$, que está muito longe da raiz. Após várias oscilações, a partir da sétima iteração (k=7), C_k estabiliza e converge para o valor teórico $\boldsymbol{C = 0.00127}$.