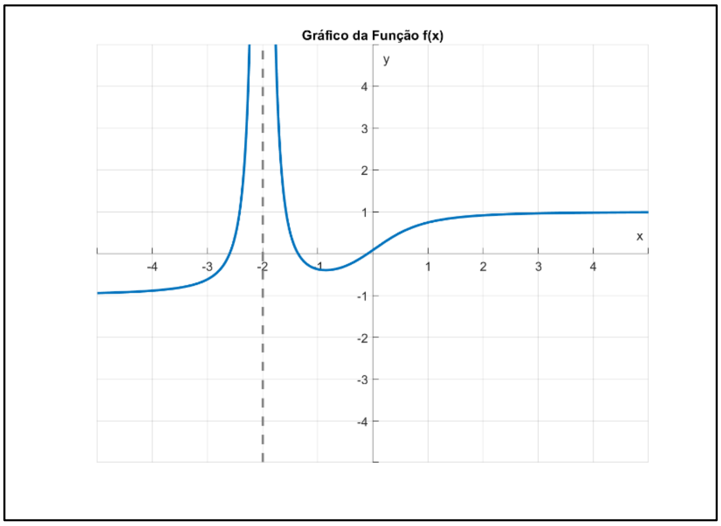
Grupo I

<u>Alínea a)</u>

Análise ("estudo tão completo quanto possível") da função: $f(x) = \frac{x}{\sqrt{x^2+1}} + \frac{1}{3(x+2)^2}$

1. **Obter o gráfico da função**



2. **Identificar o domínio**

   $D_f = \mathbb{R} \setminus \{-2\}$

3. **Estudar a continuidade**

   f é contínua em $\mathbb{R} \setminus \{-2\}$ (f é definida pela soma de duas funções racionais contínuas nos seus domínios)

4. **Determinar as coordenadas dos pontos de interseção do gráfico com os eixos coordenados**

   - <u>Interseção do gráfico com o eixo Ox e o eixo Oy</u> ($f(x) = 0$ e $f(0)$):

```
Zeros de f(x):
   -2.5976

   -1.3565

   -0.0919

Ordenada na origem:
    0.0833
```

5. Estudar a paridade da função, ou seja, a simetria do gráfico

$f(-x) \neq f(x)$ e $f(-x) \neq -f(x)$, $\forall\, x \in D_f$ $\therefore$ a função não é par nem ímpar.

Graficamente também podemos constatar que não é simétrica em relação ao eixo Oy (ou seja, não é par) e que não é simétrica em relação à Origem (ou seja, não é ímpar).

6. Determinar a existência de assíntotas e as equações que as definem

- Verticais:

$$\lim_{x \to -2} f(x) = +\infty$$

```
>> f

f =

  function_handle with value:

    @(x)x./(sqrt(x.^2+1))+1./(3*(x+2).^2)

>> limit(f, x, -2)

ans =

Inf
```

A reta de equação $x = -2$ é uma assíntota vertical do gráfico de f.
Como f é contínua em $\mathbb{R} \setminus \{-2\}$, não existem outras assíntotas verticais.

- Não verticais:

Quando $x \to -\infty$:

$$\lim_{x \to -\infty} f(x) = -1$$

```
>> f

f =

  function_handle with value:

    @(x)x./(sqrt(x.^2+1))+1./(3*(x+2).^2)

>> limit(f/x, x, -Inf)

ans =

0

>> limit(f-(0*x), x, -Inf)

ans =

-1
```

A reta de equação $y = -1$ é uma assíntota horizontal do gráfico de f, quando $x \to -\infty$.

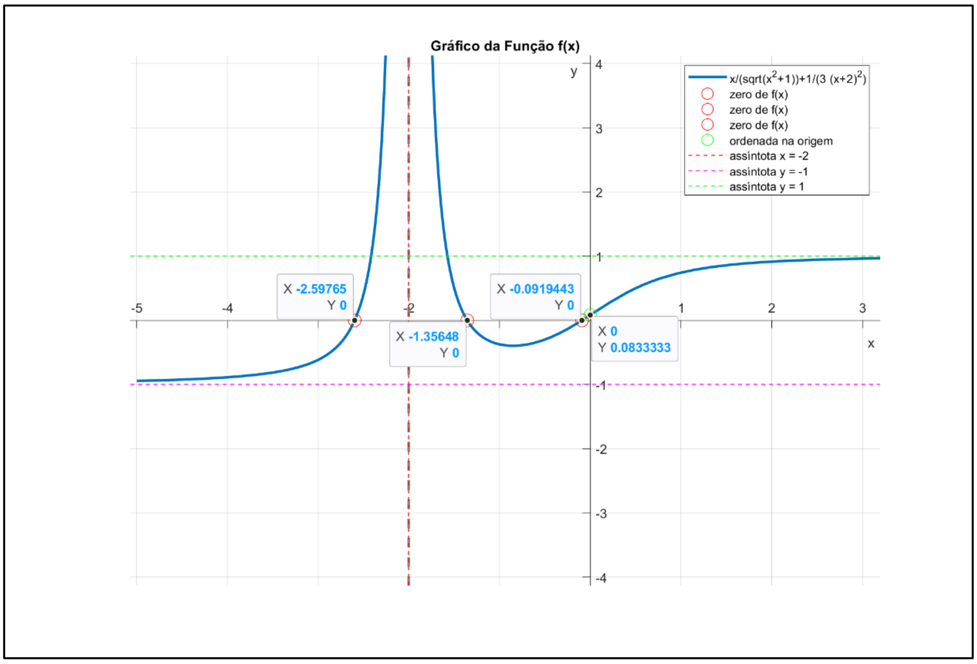
Quando $x \to +\infty$:

$$\lim_{x \to +\infty} f(x) = 1$$

```
>> f

f =

  function_handle with value:

    @(x)x./(sqrt(x.^2+1))+1./(3*(x+2).^2)

>> limit(f/x, x, +Inf)

ans =

0

>> limit(f-(0*x), x, +Inf)

ans =

1
```

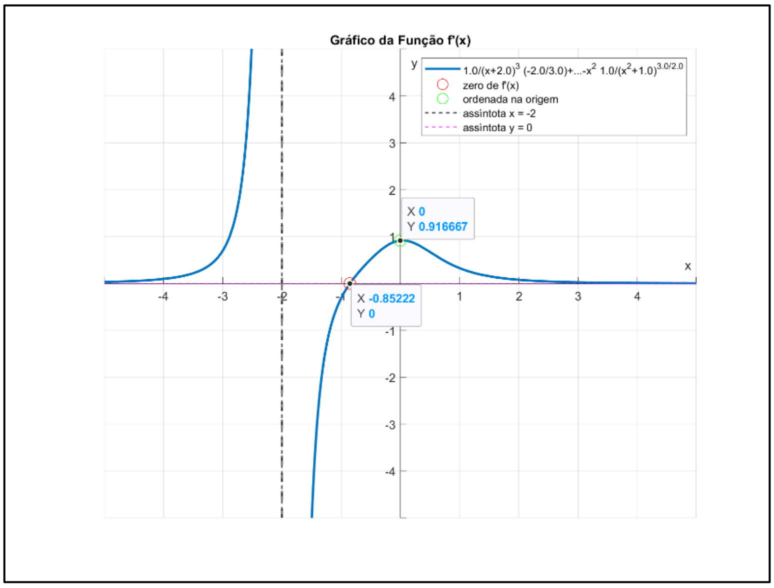
A reta de equação $y = 1$ é uma assíntota horizontal do gráfico de f, quando $x \to +\infty$.

7. Gráfico com as informações anteriores



[link para o script em MATLAB](#)

8. Estudar a monotonia e determinar os extremos



[link para o script em MATLAB]

$f'(x) = 0$

```
% Definir a função-----------------------------------
syms x
f = @(x) x./(sqrt(x.^2+1)) + 1./(3*(x+2).^2);
% ---------------------------------------------------

% Calcular a derivada -------------------------------
df = diff(f, x);
% ---------------------------------------------------

% Converter a derivada para uma função MATLAB --------
df_func = matlabFunction(df);
% ---------------------------------------------------

% Encontrar a raizes usando a função fzero -----------
disp('A raiz da função derivada de f é:');
disp(fzero(df_func, -1));
% ---------------------------------------------------
```

Command Window

```
>> T1_f2_alinea_a__monotonia_e_extremos
A raiz da função derivada de f é:
   -0.8522
```

$f(-0.8522)$

```

% Encontrar as imagens dos pontos críticos -----------
disp('As imagens dos pontos críticos são:')
f = x./(sqrt(x.^2+1)) + 1./(3*(x+2).^2);
disp(eval(subs(f, x, fzero(df_func, -1))))
% ---------------------------------------------------

```

Command Window

```
>> T1_f2_alinea_a__monotonia_e_extremos
As imagens dos pontos críticos são:
   -0.3956
```

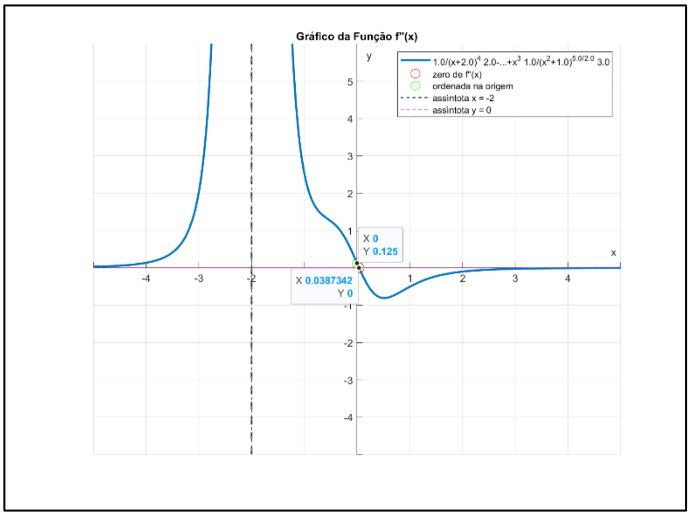
[link para o script em MATLAB]

x	$-\infty$	-2		-0.8522	$+\infty$
sinal de f'	$+$	n.d.	$-$	0	$+$
variação de f	↗	n.d.	↘	-0.3956	↗
				Min.	

f é estritamente crescente em $]-\infty; -2[$ e em $[-0.8522; +\infty[$.
f é estritamente decrescente em $]-2; -0.8522]$.

$f(-0.8522) = -0.3956$ é mínimo relativo e $x = -0.8522$ o minimizante.

9. Estudar o sentido da concavidade e a existência de pontos de inflexão



$$f''(x) = 0$$

```
% Encontrar a raizes usando a função fzero -----------
disp('A raiz da função segunda derivada de f é:');
disp(fzero(df_func, 0));
% --------------------------------------------------

% Encontrar as imagens dos possíveis pontos de inflexão
disp('As imagens dos possíveis pontos de inflexão:')
f = x./(sqrt(x.^2+1)) + 1./(3*(x+2).^2);
disp(eval(subs(f, x, fzero(df_func, 0))))
% --------------------------------------------------
```

Command Window

```
>> T1_grupo_1__alinea_a__estudo_da_2a_derivada
A raiz da função segunda derivada de f é:
    0.0387

As imagens dos possíveis pontos de inflexão:
    0.1189
```

[link para o script em MATLAB](#) [link para o script em MATLAB](#)

x	$-\infty$	-2		0.0387	$+\infty$
sinal de f''	$+$	n.d.	$+$	0	$-$
Sentido da concavidade do gráfico de f		n.d.		0.1189	
				P.I.	

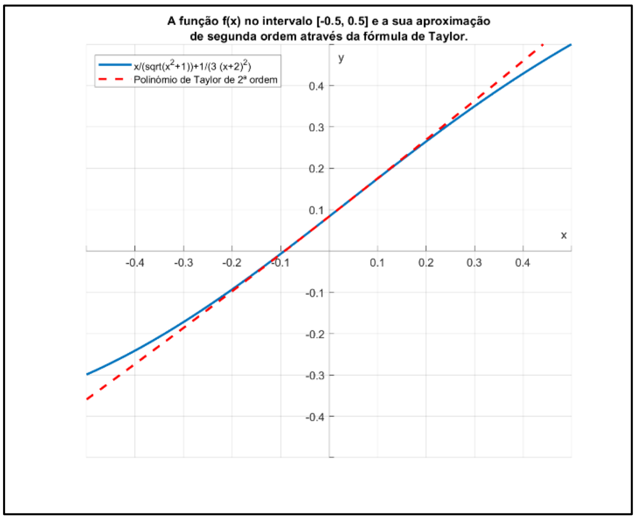
O ponto de coordenadas $(0.0387, 0.1189)$ é um ponto de inflexão do gráfico da função f.

O gráfico de f tem a concavidade voltada para cima em $]-\infty; -2[$ e em $]-2; 0.0387]$ e voltada para baixo em $[0.0387, +\infty[$.

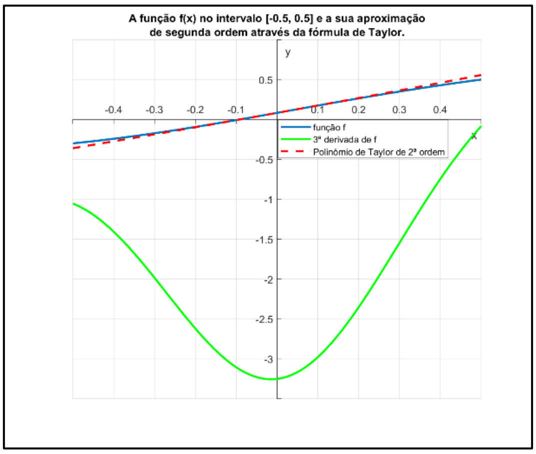
10. Indicar o contradomínio

$$D'_f =]-1; +\infty[$$

Alínea b)



Incluindo a 3ª derivada de f para o cálculo do erro cometido:



[link para o script em MATLAB]

O polinómio de Taylor de 2ª ordem, em potências de x (centrado em $a = 0$), para a função de $f(x)$, é dado por:

$$P_2(x) = f(0) + \frac{f'(0) \cdot x}{1!} + \frac{f''(0) \cdot x^2}{2!} + R_2(x)$$

Onde $R_2(x)$ é o resto de Lagrange e é dado por:

$$R_2(x) = f^{(3)}(\xi) \cdot \frac{x^3}{3!} \quad \text{, com } \xi \in [-0.5; 0.5]$$

$$\Leftrightarrow$$

$$|\, R_2(x)| = |f^{(3)}(\xi)| \cdot |\frac{x^3}{3!}| \quad \text{, com } \xi \in [0; x]$$

Através dos cálculos em MATLAB, temos que os maiores valores em módulo são:

$$f^{(3)}(-0.0145) \approx -3.255 \;\Leftrightarrow\; |\, f^{(3)}(-0.0145)\,| \approx 3.255 \qquad \text{e} \qquad |\, \frac{(0.5)^3}{6}\,| \approx 0{,}021$$

Portanto,

$$|\, f^{(3)}(x)\,| \leq 3.255 \qquad \text{e} \qquad |\, \frac{(x)^3}{6}\,| \leq 0{,}021$$

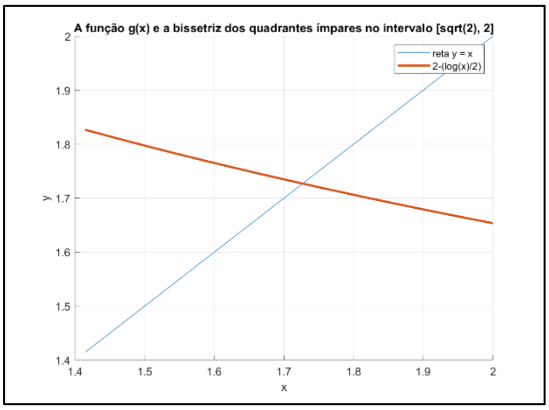
Ou seja, um majorante do erro cometido é:

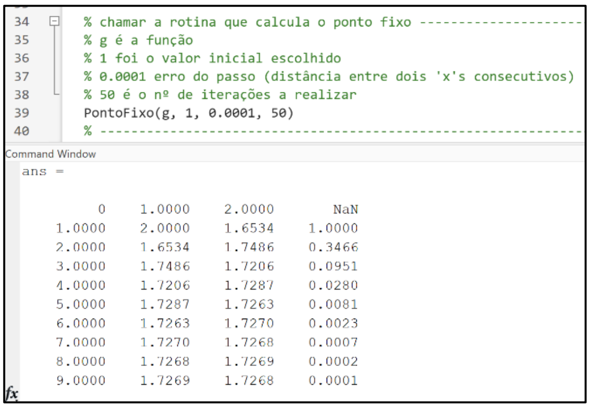
$$|\, R_2(x)\,| < 0{,}068$$

Grupo II

$$g(x) = 2 - \frac{\ln(x)}{2}$$

a) Mostre que $g(x)$ tem um ponto fixo no intervalo $[\sqrt{2}; 2]$:





```
34  % chamar a rotina que calcula o ponto fixo --------------------
35  % g é a função
36  % 1 foi o valor inicial escolhido
37  % 0.0001 erro do passo (distância entre dois 'x's consecutivos)
38  % 50 é o nº de iterações a realizar
39  PontoFixo(g, 1, 0.0001, 50)
40  % ------------------------------------------------------------
```

Command Window

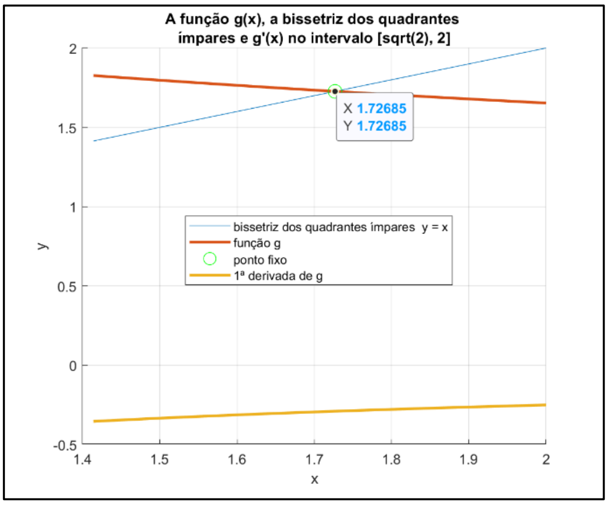
ans =

0	1.0000	2.0000	NaN
1.0000	2.0000	1.6534	1.0000
2.0000	1.6534	1.7486	0.3466
3.0000	1.7486	1.7206	0.0951
4.0000	1.7206	1.7287	0.0280
5.0000	1.7287	1.7263	0.0081
6.0000	1.7263	1.7270	0.0023
7.0000	1.7270	1.7268	0.0007
8.0000	1.7268	1.7269	0.0002
9.0000	1.7269	1.7268	0.0001

[link para o script em MATLAB]

Sabemos que g (com domínio $\mathbb{R}^+$) é contínua no intervalo $[\sqrt{2}, 2]$ e constatamos graficamente que $g([\sqrt{2}, 2]) \subset [\sqrt{2}, 2]$.

Analiticamente, temos: $\quad x \in [\sqrt{2}, 2] \;\Rightarrow\; \sqrt{2} < \; 2 - \frac{\ln(x)}{2} < 2$



Calculando $g'(x)$ no intervalo verificamos que,

$$max_{x \in [a,b]} \mid g'(x) \mid < 1$$

portanto, pelo corolário **1.1** do teorema **1.1** **(**da sebenta de cálculo numérico) concluímos que g tem um, e só um, ponto fixo c no intervalo $[\sqrt{2}, 2]$.

Pelos cálculos no MATLAB, após 9 iterações, temos que o ponto fixo é $c \approx 1.727$ com um erro de ≈ 0.0001.

b) Determine-o pelo método da bissecção:

Para encontrar o ponto fixo pelo método da bisseção precisamos de transformar o problema da pesquisa do ponto fixo, que resolvemos na alínea anterior, num problema de pesquisa de zeros.

Para tal, como o ponto fixo é o ponto de interseção entre a função g e a bissetriz dos quadrantes ímpares ($y = x$), temos que: $\quad ponto\ fixo \iff g(x) = x \iff g(x) - x = 0$

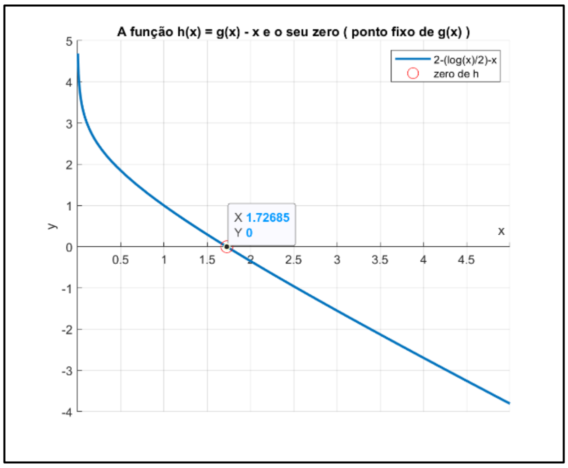
Usando o MATLAB, com a rotina do *método da bisseção*, encontramos o zero desta equação, e que corresponde ao ponto fixo:

```
% econtrar o zero através do método da bissecção --------------------
% h é a função
% sqrt(2) é o limite inferior do intervalo
% 2 é o limite superior do intervalo
% 0.0001 é o valor dos erros
% 50 é o nº de iterações
MetBiss(h, sqrt(2), 2, 0.0001, 0.0001, 50)
% ---------------------------------------------------------------
```

Command Window

ans =

0	1.4142	2.0000	0.4125	-0.3466	1.7071	0.0255	0.2929
1.0000	1.7071	2.0000	0.0255	-0.3466	1.8536	-0.1621	0.1464
2.0000	1.7071	1.8536	0.0255	-0.1621	1.7803	-0.0687	0.0732
3.0000	1.7071	1.7803	0.0255	-0.0687	1.7437	-0.0217	0.0366
4.0000	1.7071	1.7437	0.0255	-0.0217	1.7254	0.0019	0.0183
5.0000	1.7254	1.7437	0.0019	-0.0217	1.7346	-0.0099	0.0092
6.0000	1.7254	1.7346	0.0019	-0.0099	1.7300	-0.0040	0.0046
7.0000	1.7254	1.7300	0.0019	-0.0040	1.7277	-0.0011	0.0023
8.0000	1.7254	1.7277	0.0019	-0.0011	1.7266	0.0004	0.0011
9.0000	1.7266	1.7277	0.0004	-0.0011	1.7271	-0.0004	0.0006
10.0000	1.7266	1.7271	0.0004	-0.0004	1.7268	0.0000	0.0003
11.0000	1.7268	1.7271	0.0000	-0.0004	1.7270	-0.0002	0.0001
12.0000	1.7268	1.7270	0.0000	-0.0002	1.7269	-0.0001	0.0001



[link para o script em MATLAB]

c) Fazer o mesmo que na alínea anterior, usando agora o método de Newton-Raphson:

```
% econtrar o zero através do método Newton-Raphson ---
% h é a função
% dh é a 1ª derivada de h
% 2 é o ponto inicial escolhido
% 0.0001 é o valor dos erros
% 50 é o nº de iterações
MetNR(h, dh, 2, 0.0001, 0.0001, 50)
% ---------------------------------------------------
```

Command Window

ans =

0	2.0000	-0.3466	-1.2500	NaN
1.0000	1.7227	0.0053	-1.2902	0.2773
2.0000	1.7268	0.0000	-1.2895	0.0041
3.0000	1.7269	0.0000	-1.2895	0.0000

[link para o script em MATLAB]

d) Comentar os resultados obtidos nas duas últimas alíneas:

Em relação aos resultados obtidos, constatamos que o *método de Newton-Raphson* é bem mais eficiente do que o *método da bisseção* (tendo em conta, como foi o caso, que o ponto inicial escolhido está relativamente próximo da raiz a se encontrar).
Foi definido, em ambos, os mesmos valores para os erros, e verificamos que ambos convergem para ≈ 1,7269 porém o erro cometido pelo *método Newton-Raphson* foi menor, o que significa que este método retornou um valor ainda mais aproximado do valor real.
Ou seja, ainda que o *método da bisseção* seja um algoritmo robusto, eficaz e não tão sensível às condições iniciais, o *método de Newton-Raphson* não só convergiu mais depressa (pois o número de iterações necessárias para convergir à raiz foram menores) como também o erro cometido foi menor.