**Métodos de Aproximação de Séries**

FCUP

Análise Numérica (M2018) 2018/2019

Trabalho de Grupo 2

Ângelo Gomes – 201703990 – MIERSI

Eduardo Morgado – 201706894 – MIERSI

Simão Cardoso – 201604595 – MIERSI

Sónia Rocha – 201704679 – MIERSI

1. **Considerações Iniciais:**

Na realização deste trabalho, utilizamos como linguagem de implementação Python(3.7.2), onde os pontos de virgula flutuante são sempre de precisão dupla, uma das soluções para este problema seria, através de bibliotecas como *numpy*, forçar o programa a trabalhar apenas em precisão simples, no entanto, todos os cálculos seriam realizados em precisão dupla e só depois convertidos/arredondados para precisão simples, o que acabaria por transmitir erros de arredondamento para os resultados.

Uma nota importante para utilização dos próximos métodos em Python é a necessidade de configurar a divisão inteira como sendo uma divisão de pontos flutuantes, caso contrário, operações como 1/4=0 e não 0.25, para isso é necessário importar um parâmetro de uma biblioteca: *from \_\_future\_\_ import division*

É necessário também referir alguma notação que iremos utilizar para este trabalho. Primeiramente a variável *error* que iremos utilizar, irá representar o erro limite para o programa . Como bibliotecas principais importadas, temos *numpy (as np)*, *math (as mt)*, *decimal (as dm)*. Consideramos *F*,*F\_deriv* e *F\_dderiv* como sendo F(x),F’(x) e F’’(x), respetivamente. A *Figura 1* apresenta estas notações.

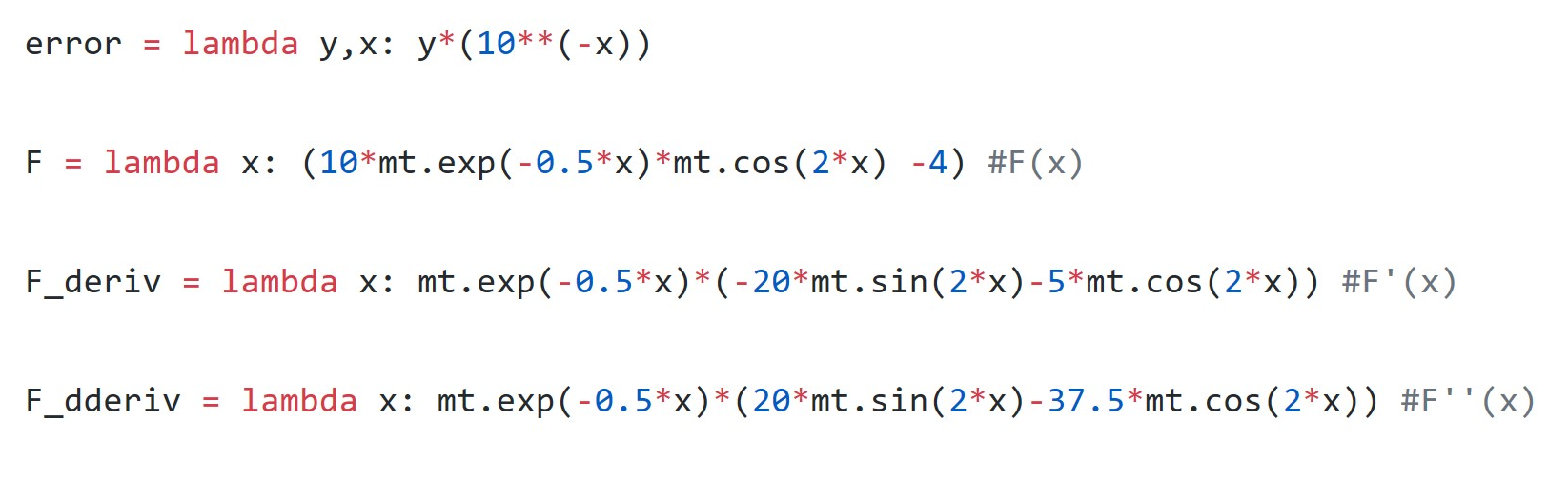


Figura -Notação geral

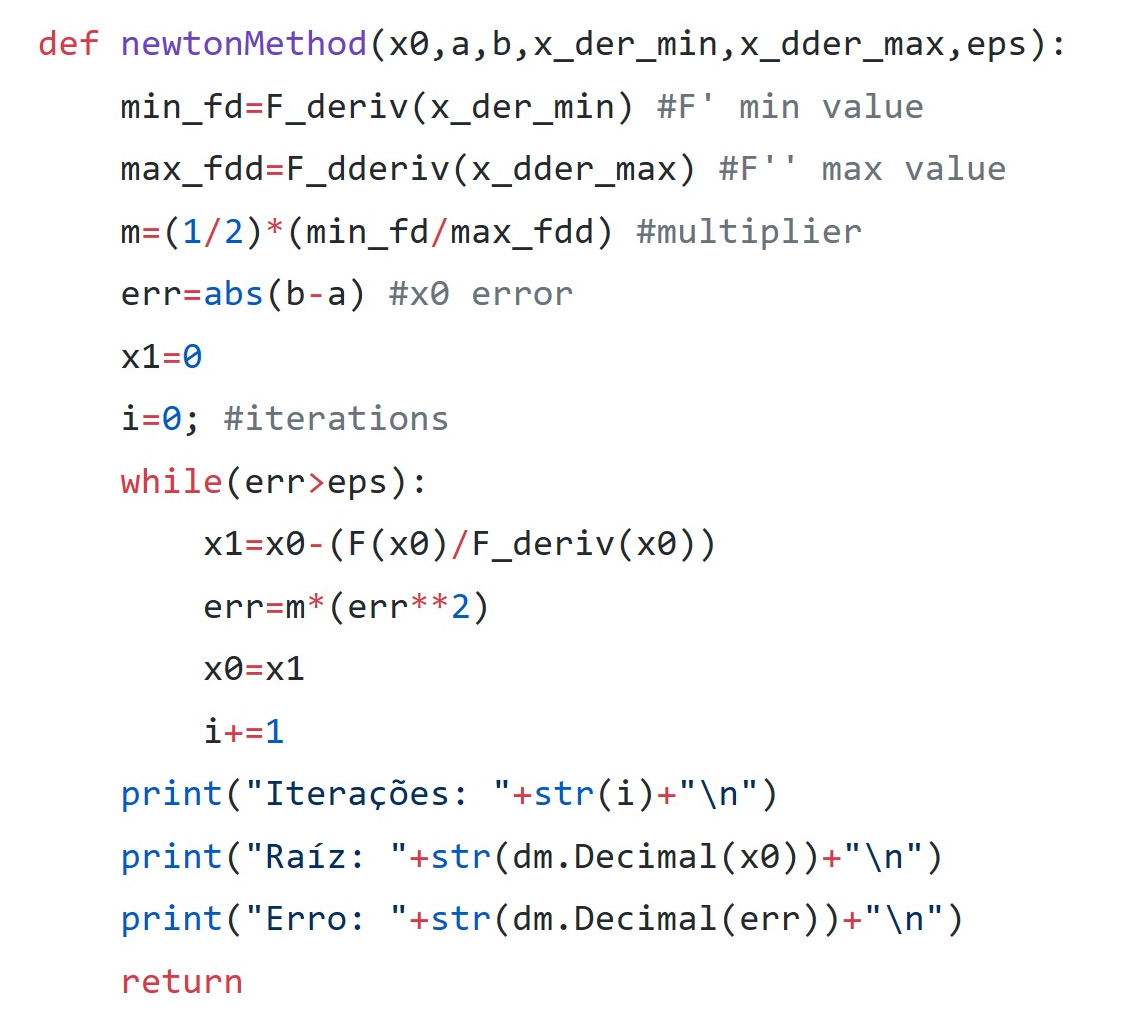
Todos estes programas podem ser encontrados no Github no repositório da equipa, <https://github.com/thejoblessducks/Trabalho2An.git>

1. **Método de Newton para determinar raiz de F(x)=0 (Exercício 1)**

**2.1. Programa de cálculo de raiz**

O programa desenvolvido nesta parte, tem como objetivo calcular um valor aproximado da raiz de F(x)=0 para um intervalo I=[a,b]. Para o nosso programa, a função *newtonMethod* recebe 7 parâmetros, x0 (o ponto de arranque para o programa), a, b (os extremos do intervalo I), x\_der\_min (o ponto em I onde |F’(x)| é mínima), x\_dder\_max (o ponto em I onde |F’’(x)| é máxima) e eps (o valor limitador do erro absoluto)

A *Figura 2* representa o pseudocódigo a implementar, e a *Figura 3* a sua respetiva implementação em Python (em inglês).



**newtonMethod(x0,a,b,x\_min,x\_max,eps):**

min\_der<- F’(x\_min);

max\_dder<- F’’(x\_max);

multiplier<- ½(min\_der/max\_dder);

error<- |b-a|;

x1<- 0; i<-0;

Enquanto error > eps fazer

X1<- x0-(F(x0)/F’(x0));

error<- m\*;

x0<- x1; i<- i+1;

escreve(i)

escreve(x0);

escreve(error)

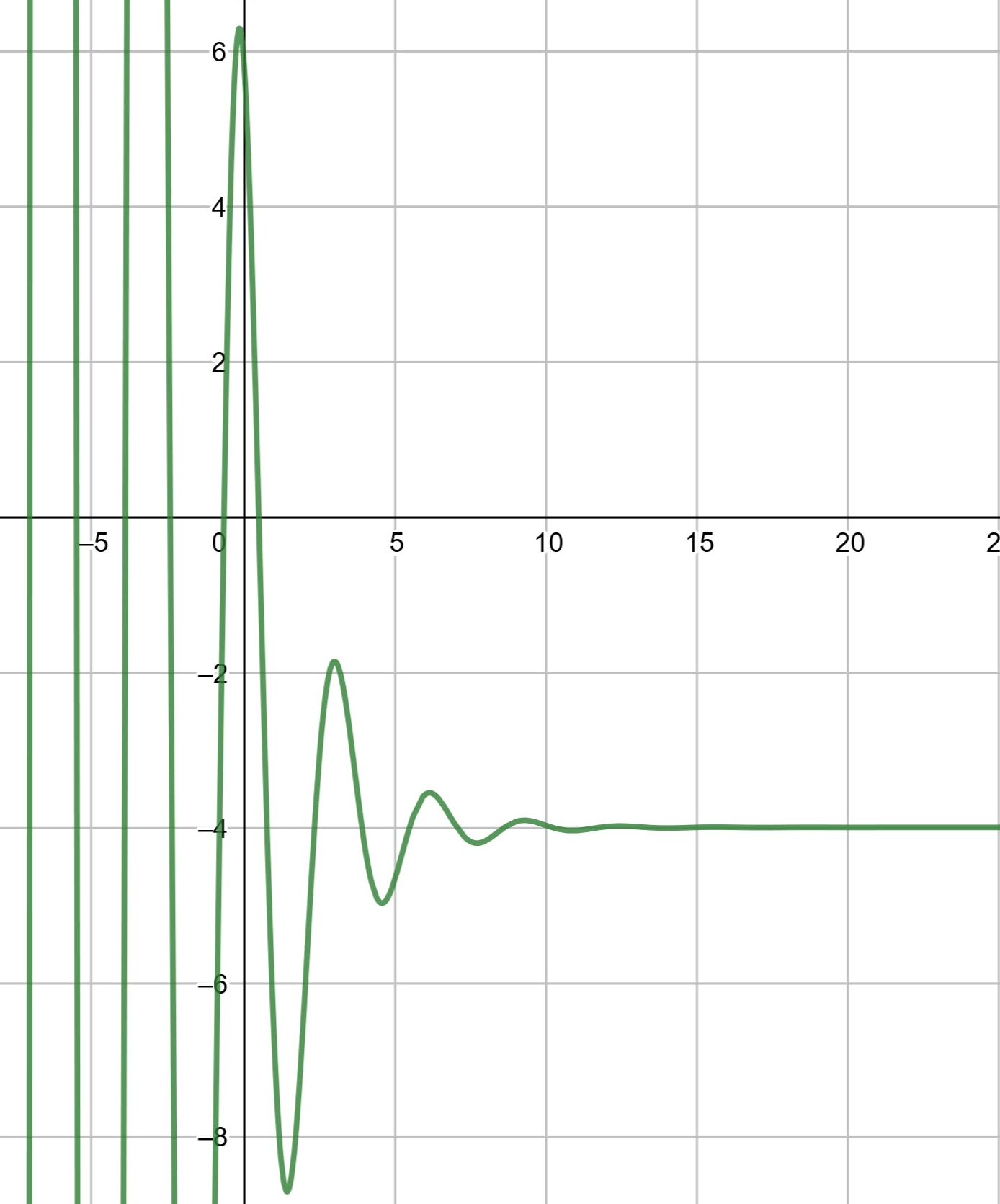
Figura 2-Pseudocódigo

Figura 3- Implementação em Python 3.7

* 1. **Separação de Raiz e Determinação de Intervalo**

Toda a próxima interpretação gráfica foi feita utilizando a Calculadora Gráfica GeoGebra [<https://www.geogebra.org/graphing?lang=pt-PT>](https://www.geogebra.org/graphing?lang=pt-PT).

As Figuras *4*, *5* e *6* apresentam os gráficos de , e respetivamente.



Figura

Figura 4-Gráfico de F(x)

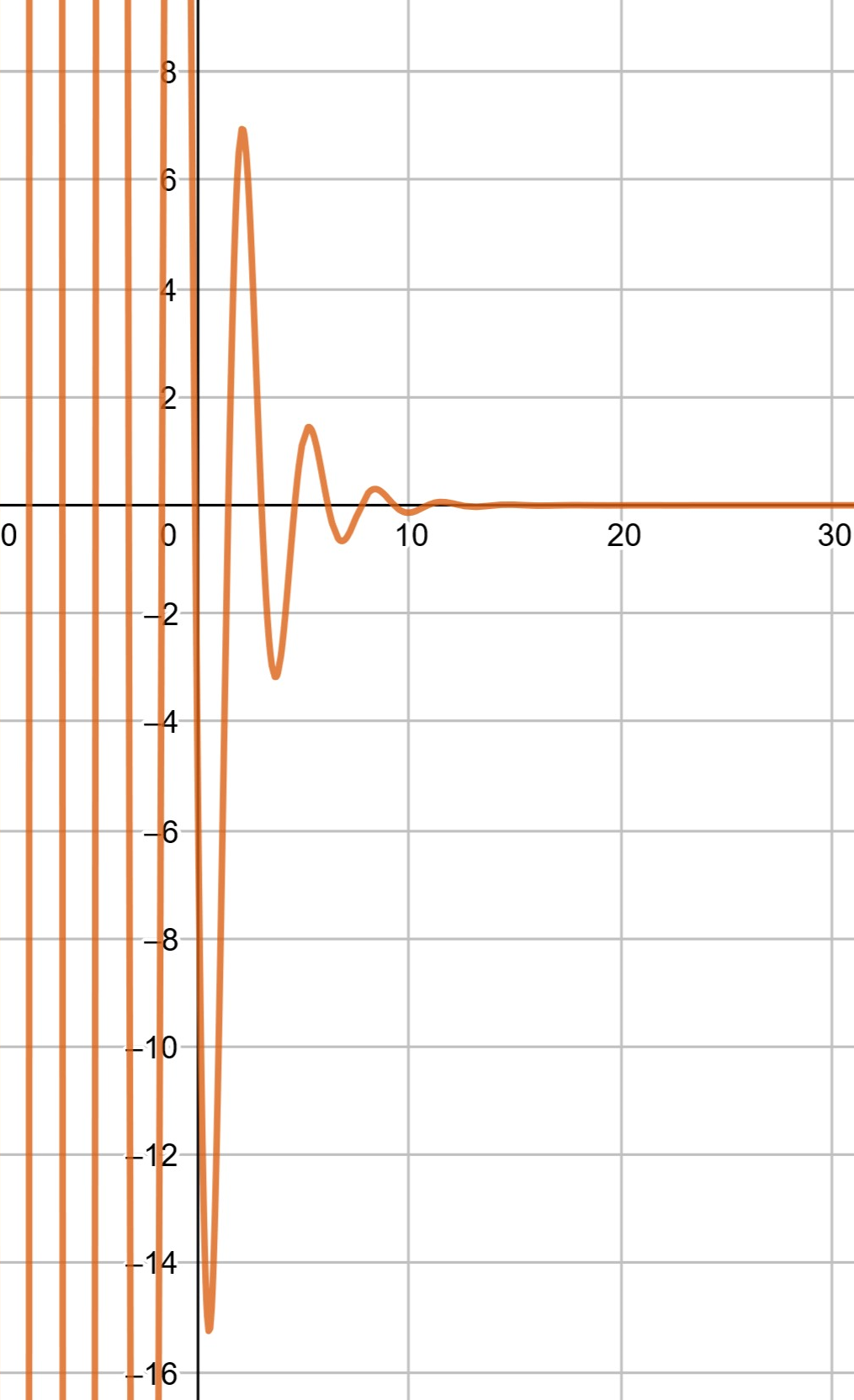


Figura 5-Gráfico de

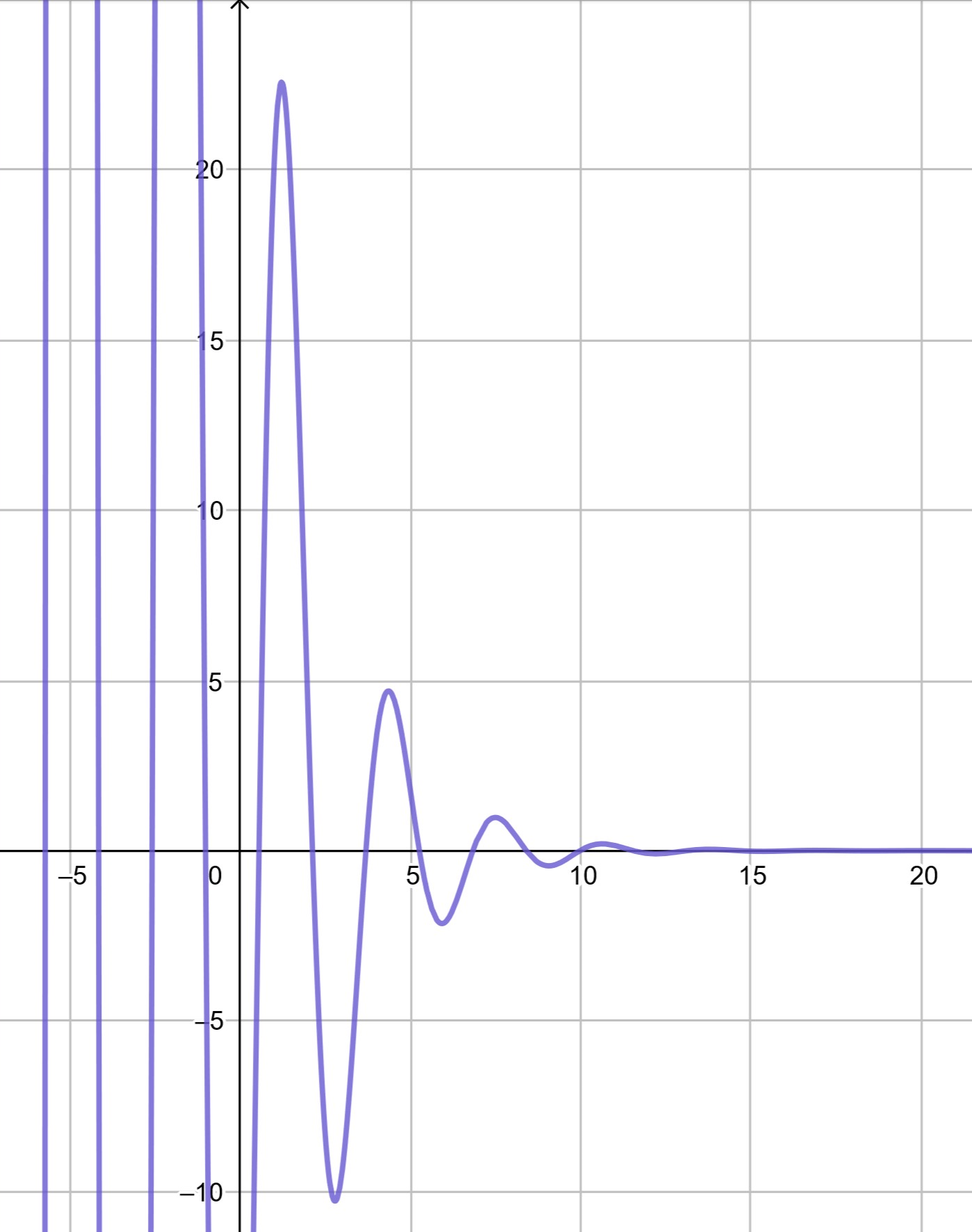


Figura 6-Gráfico de

A *Figura 7* apresenta o gráfico de

e a *Figura 8* o gráfico de F(x)=0

(), com um intervalo menor.

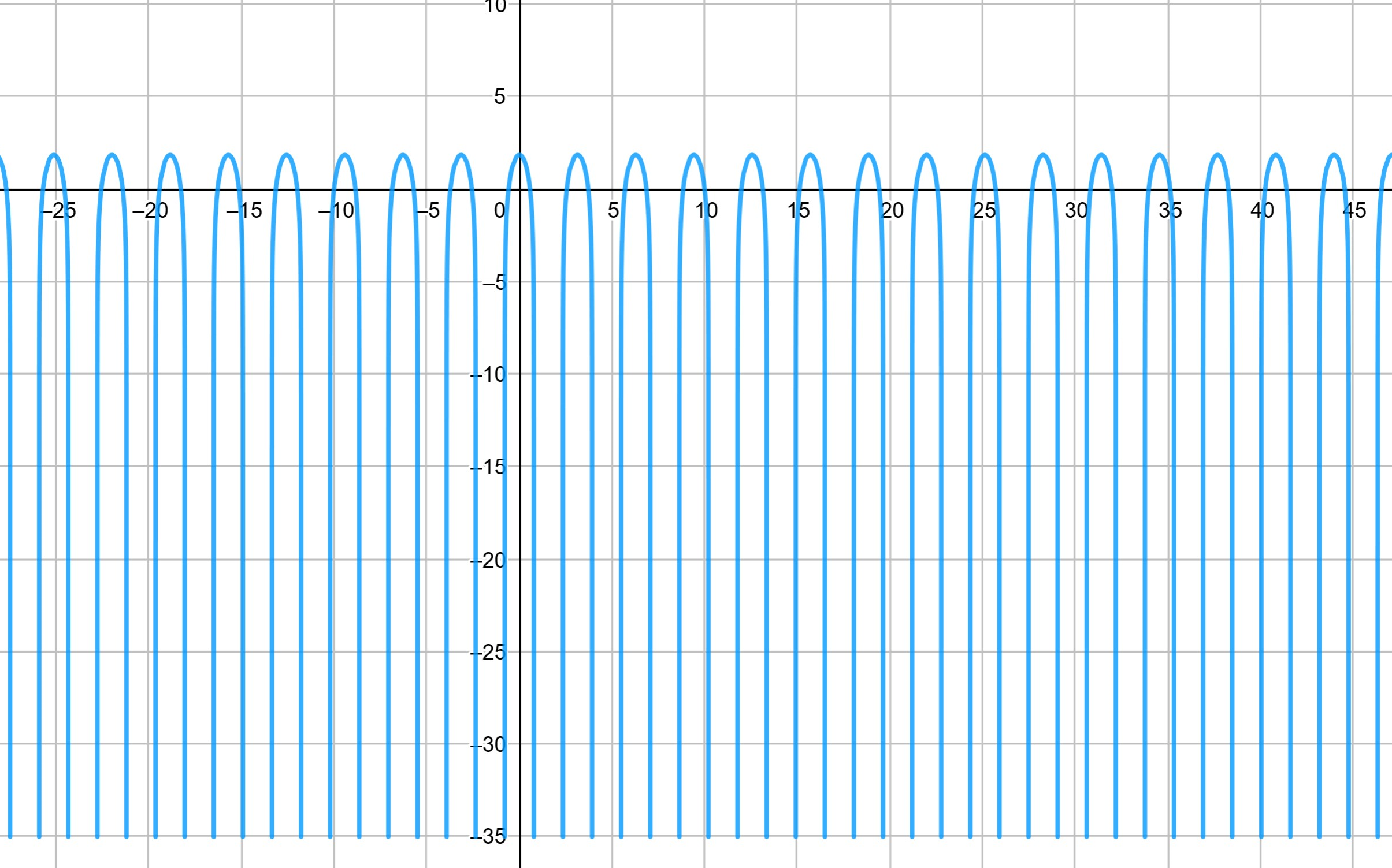
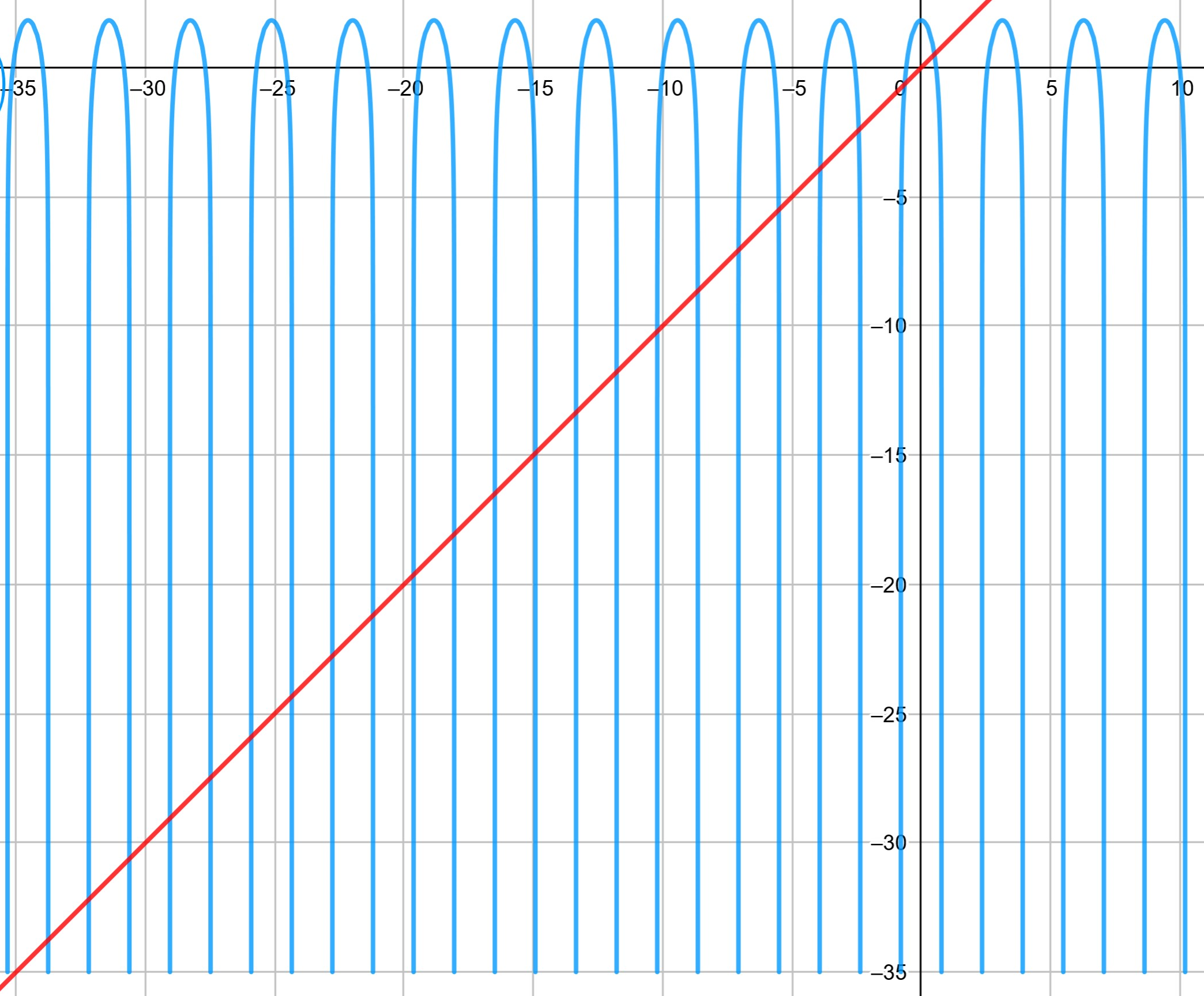


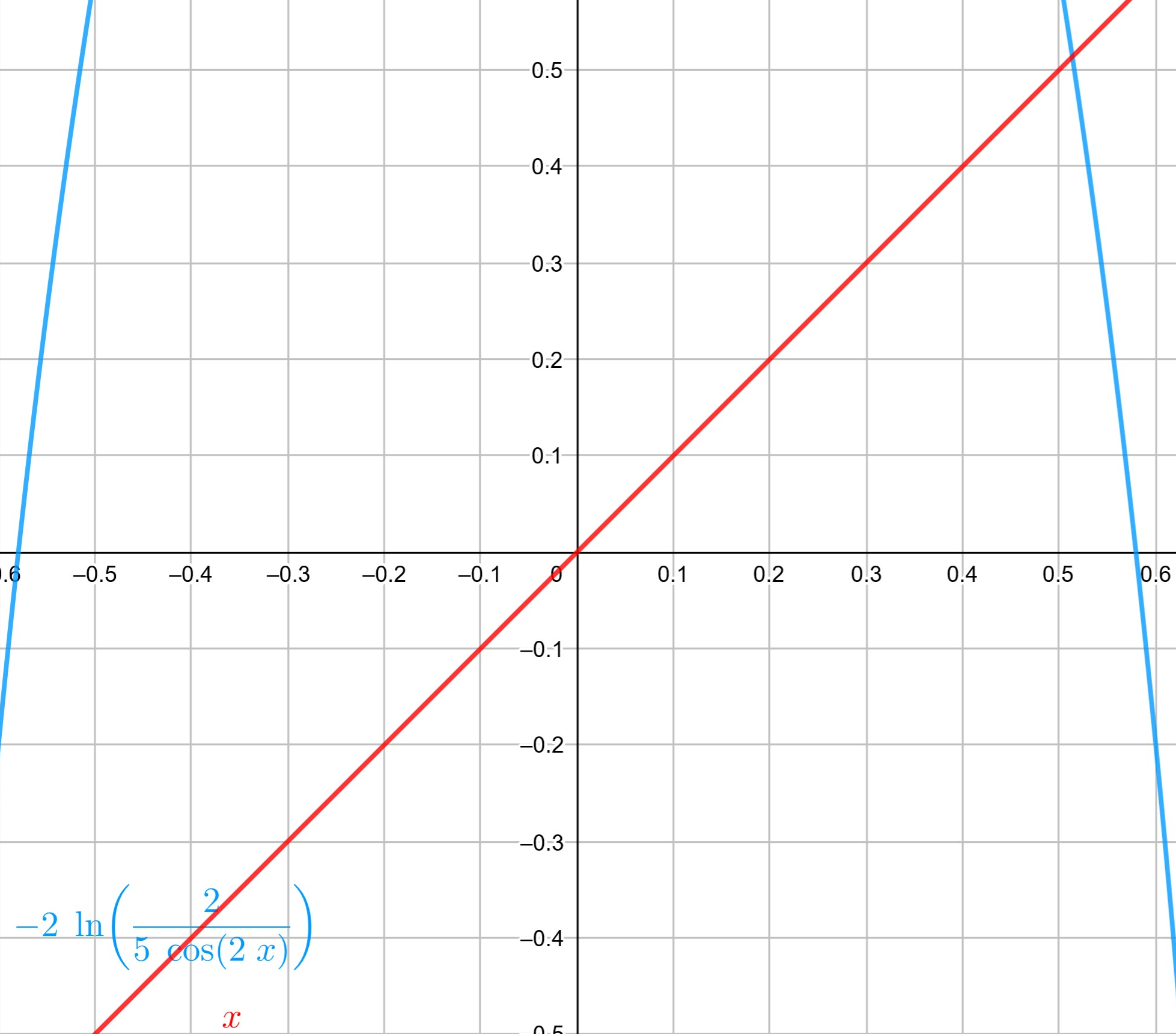
Figura 7-Gráfico de



Figura

Figura 8-Gráfico de x=

Por observação gráfica, verificamos que existem, pelo menos 23 raízes de F(x)=0 (pontos de intersecção de x com , uma vez que, por observação das figuras 4-6, iremos focar-nos mais em intervalos de -1 a 10, uma vez que, as funções têm um comportamento mais fácil de estudar, como tal, a *Figura 9* apresenta o gráfico da *Figura 8* em um intervalo menor.



Por observação gráfica, podemos

concluir que, um intervalo I de

amplitude que contenha uma

raiz de F(x)=0 é o intervalo

I=[0.5,0.6].

Figura 9- Gráfico de x=

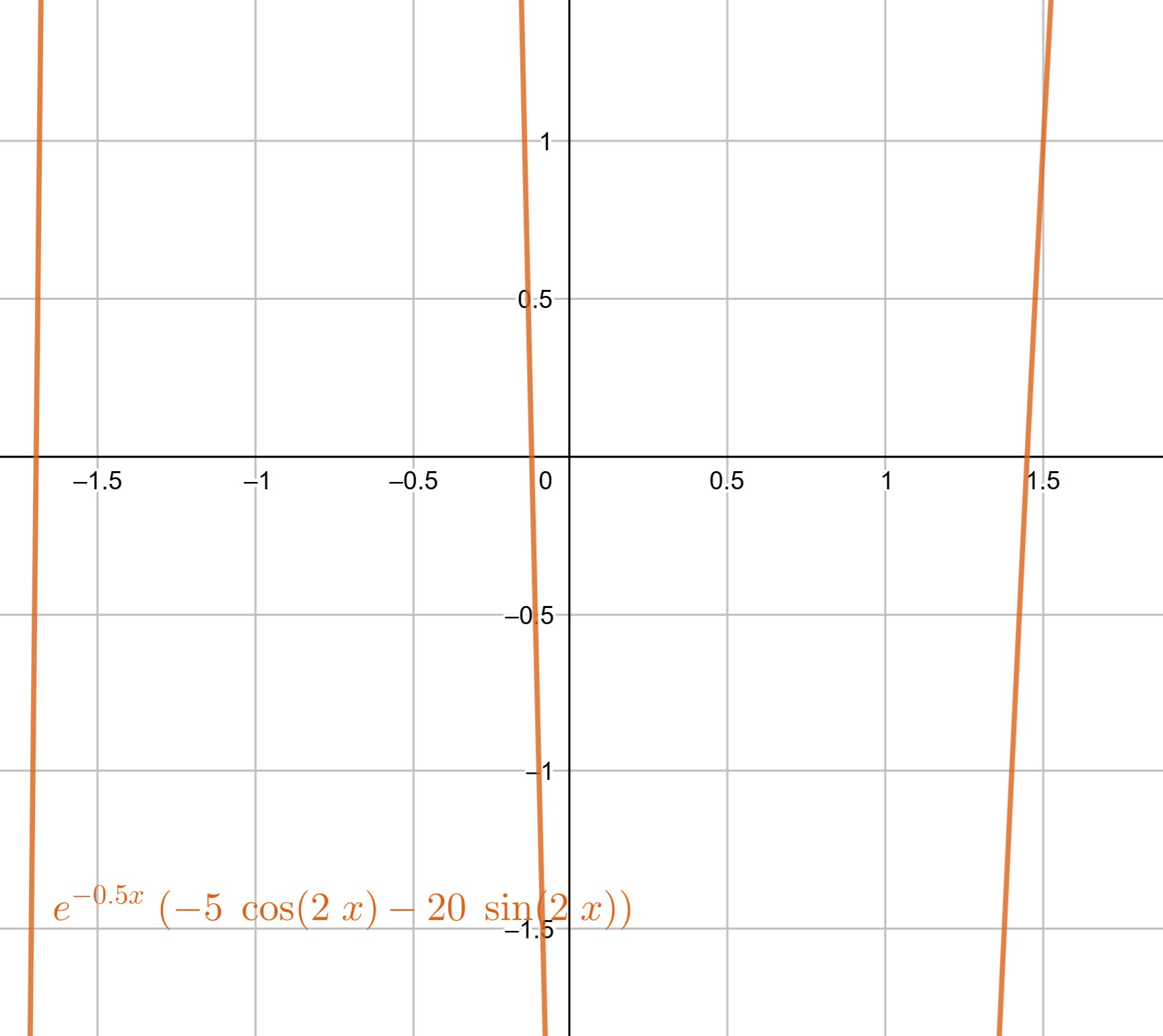
* 1. **Verificação das Condições de Aplicabilidade para I=[0.5,0.6]**

As condições de aplicabilidade do método de Newton para um intervalo I=[a,b] são:

1. , e existem e são contínuas em [a,b]
2. ;
3. ;
4. ;
5. 2 opções (ii se i falhar):
   1. ]
   2. , c é extremo de [a,b]

Por observação das figuras 4-6, , e existem e são contínuas em [0.5,0.6] (são compostas por funções contínuas), logo a condição **1** é satisfeita.

F(0.5)=0.2078785891.. >0 e F(0.6)=-1.325587731.. <0 logo a condição **2** também é satisfeita. A



A partir da *Figura 10*, verificamos que

pelo que é

satisfeita a condição **3**

Para o intervalo considerado, a condição **5**

não é satisfeita, como mostra a *Figura 11*,

pelo que, é necessário corrigir o intervalo,

uma vez que, e ,

basta ir diminuindo b até que .

A *Figura 12* apresenta esse cálculo.



Figura 10-Gráfico de

Figura 11-Gráfico de

|  |  |
| --- | --- |
|  |  |
| 0.5 | -2.6727794736.. **<0** |
| 0.6 | 3.742884736.. **>0** |
| 0.59 | 3.13257.. **>0** |
| 0.58 | 2.51478.. **>0** |
| 0.56 | 1.2575614.. **>0** |
| 0.55 | 0.6185.. **>0** |
| 0.54 | -0.027220.. **<0** |

Figura 12-Cálculo de novo extrema superior para I=[0.5,0.6]

Através da *Figura 12*, escolhemos como novo intervalo **I=[0.5,0.54]**, sendo assim, a condição **4** é satisfeita.

Apesar de, , F(x) não mantem o mesmo sinal no intervalo, o que viola a condição 5.i. sendo necessário testar a condição 5.ii. (ou ).

, logo (**5.ii.**) é satisfeita, sendo .

Sendo assim, verificamos que todas as condições de aplicabilidade do método de Newton são satisfeitas para a função F(x) e para o intervalo I=[0.5,0.54].

* 1. **Aplicação do Método**

Para a aplicação do método de newton para F(x) é necessário primeiro calcular o valor do multiplicador para o intervalo [0.5,0.54]. No intervalo considerado, é monótona crescente e negativa em todo o intervalo (*Figura 13)* sendo assim, , ou seja *x\_dder\_max* na *Figura 3* será igual a 0.5.

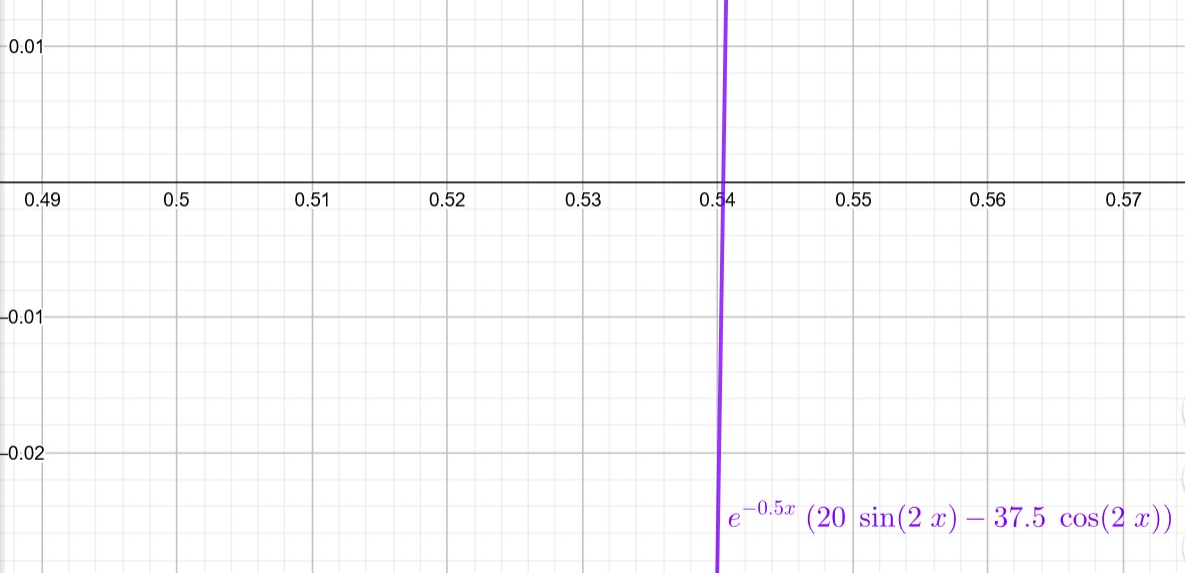


Figura 13-Gráfico de para I

Através da *Figura 14*, é visível que, para este intervalo, será o mínimo, no entanto é negativa neste mesmo intervalo, como tal, , ou seja, *x\_der\_min* na *Figura 3* será igual a 0.5.

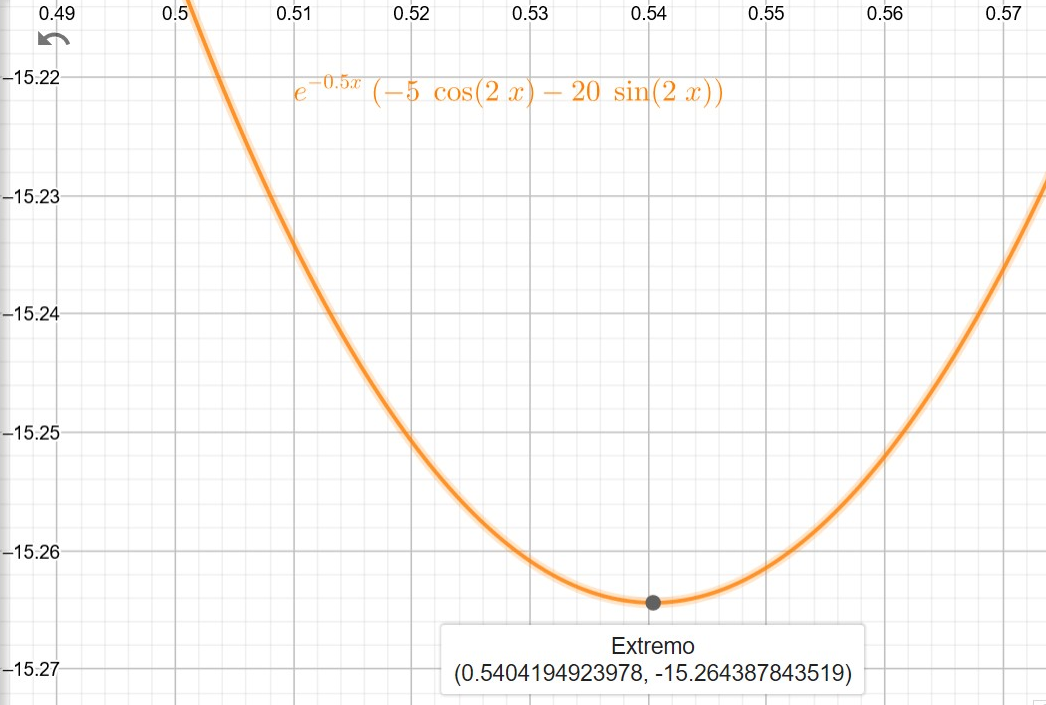


Figura 14-Gráfico de em [0.5,0.54]

Sendo assim, e, sendo erro absoluto inferior a , o algoritmo que implementa o método de Newton para F(x) é chamado da seguinte forma: newtonMethod(0.54,0.5,0.54,0.5,0.5,error(5,12)) onde e 0.54, 0.5, 0.54, 0.5 e 0.5 são respetivamente, , *a*, *b*, *x\_der\_min* e *x\_dder\_max*. A *Figura 15* apresenta a tabela com os resultados.

|  |  |  |  |
| --- | --- | --- | --- |
|  | n | F(x)=0 |  |
|  | 4 | 0.51365209475085 |  |

Figura 15-Tabela de resultados para cálculo de raiz de F(x)=0 para intervalo [0.5,0.54]

* 1. **Iterações para Majoração inferior a**

Para o intervalo[0.5,0.54] e um erro , o número de iterações n é tal que => onde onde .

1. **Método Iterativo Simples para F(x)=0 (Exercício 3)**

**3.1.1 Algoritmo**

Para este problema, consideramos F(x)=0 onde ), uma vez que, caso , o programa terminava devido a um erro de cálculo no ln, assim sendo, a *Figura 16* é a versão atualizada da *Figura 1*.

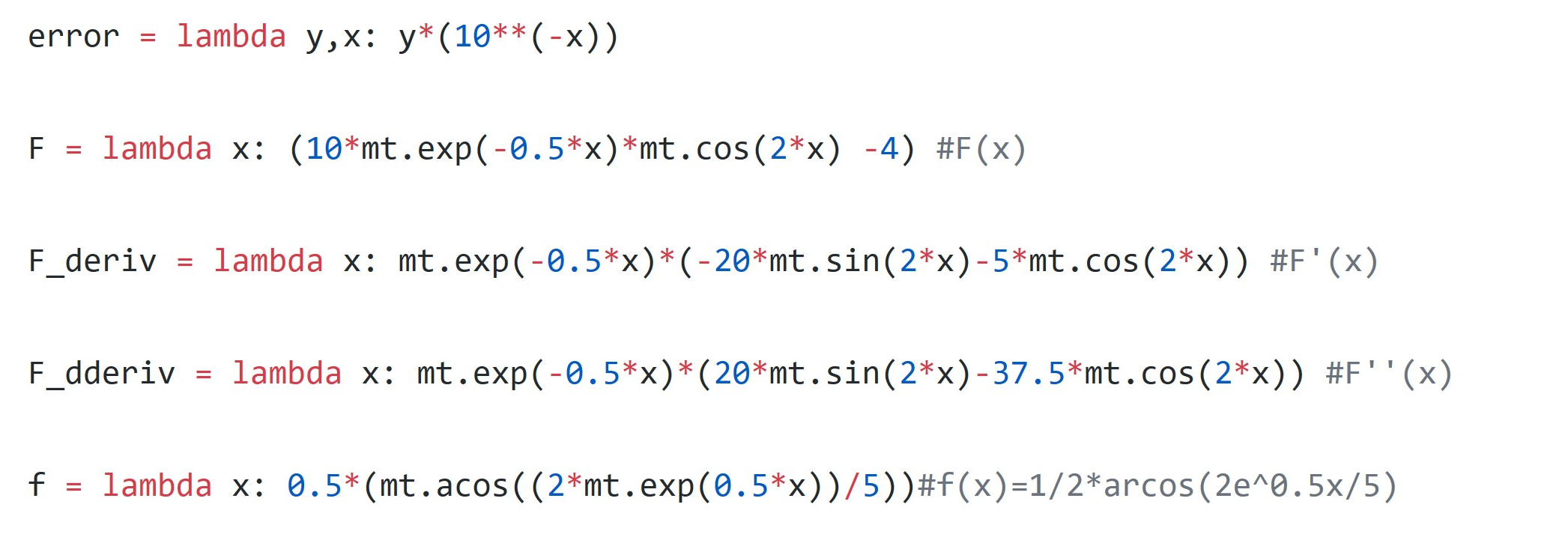
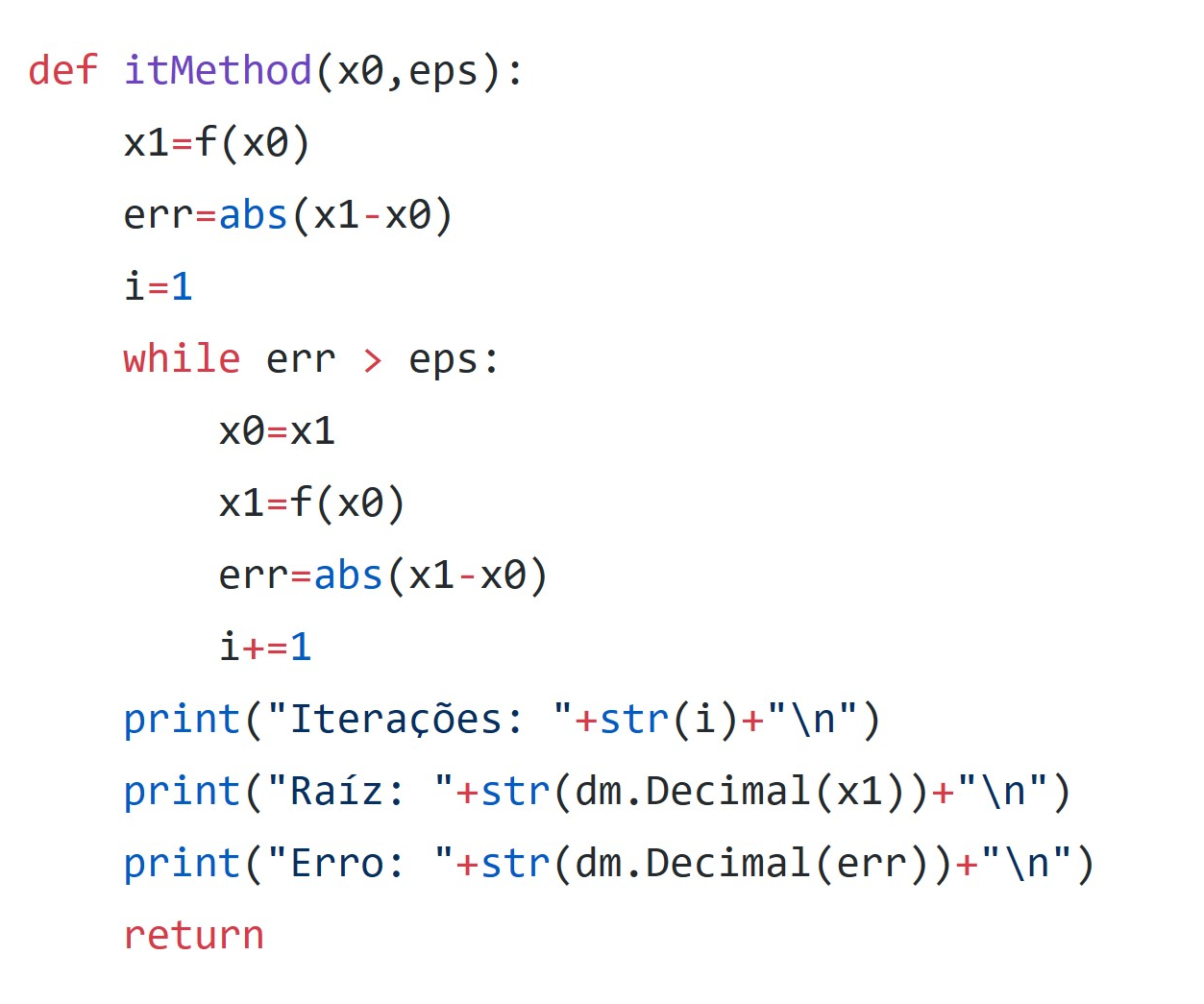


Figura 16-Versão atualizada de Figura 1

A *Figura 17* e *18* representam o pseudocódigo e o código para este problema respetivamente.



**iterativeMethod(x0,eps,f):**

x1<- f(x0);

error<- |x1-x0|;

i<-1;

Enquanto error > eps fazer

X0<- x1;

X1<- f(x0);

error<- |x1-x0|;

i<- i+1;

escreve(i)

escreve(x1);

escreve(error)

Figura 18-Código

Figura 17-Pseudocódigo

**3.2. Aplicação a F(x)**

Para a aplicação deste método em F(x), nou houve uma verificação inicial de aplicabilidade do método iterativo, tendo sido fornecido ao método como parâmetros o de **2.5.**(0.54) e como erro/eps . A *Figura 19* apresenta a tabela com os resultados do método.

|  |  |  |  |
| --- | --- | --- | --- |
|  | n | F(x)=0 |  |
|  | 13 | 0.5136520947528 |  |

Figura 19- Tabela de resultados aplicando método iterativo simples para cálculo de raiz de F(x)=0 para