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 $error(5,4) == 5.0 * 10^{-4}$. 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.

```
error = lambda y,x: y*(10**(-x))

F = lambda x: (10*mt.exp(-0.5*x)*mt.cos(2*x) -4) #F(x)

F_deriv = lambda x: mt.exp(-0.5*x)*(-20*mt.sin(2*x)-5*mt.cos(2*x)) #F'(x)

F_dderiv = lambda x: mt.exp(-0.5*x)*(20*mt.sin(2*x)-37.5*mt.cos(2*x)) #F''(x)
```

Figura 1-Notação geral

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

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

$$F(x) = 10e^{-0.5x} \cdot \cos(2x) - 4$$

$$F'(x) = e^{-0.5x} \cdot (-5cos(2x) - 20sen(2x)) \qquad F''(x) = e^{-0.5x} \cdot (20sen(2x) - 37.5cos(2x))$$

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*error²;
                x0<- x1; i<- i+1;
        escreve(i)
        escreve(x0);
        escreve(error)
```

Figura 2-Pseudocódigo

```
def newtonMethod(x0,a,b,x_der_min,x_dder_max,eps):
    min_fd=F_deriv(x_der_min) #F' min value
    max_fdd=F_dderiv(x_dder_max) #F'' max value
    m=(1/2)*(min_fd/max_fdd) #multiplier
    err=abs(b-a) #x0 error
    x1=0
    i=0; #iterations
    while(err>eps):
        x1=x0-(F(x0)/F_deriv(x0))
        err=m*(err**2)
        x0=x1
        i+=1
    print("Iterações: "+str(i)+"\n")
    print("Raíz: "+str(dm.Decimal(x0))+"\n")
    print("Erro: "+str(dm.Decimal(err))+"\n")
    return
```

Figura 3- Implementação em Python 3.7

2.2. Separação de Raiz e Determinação de Intervalo

$$F(x) = 0 \equiv 10e^{-0.5x} \cdot \cos 2x = 4 \equiv e^{-0.5x} = \frac{2}{5 \cdot cos2x} \equiv x = -2ln(\frac{2}{5 \cdot cos2x})$$

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

As Figuras *4*, *5* e *6* apresentam os gráficos de $F(x)$, $F'(x)$ e $F''(x)$ respetivamente.

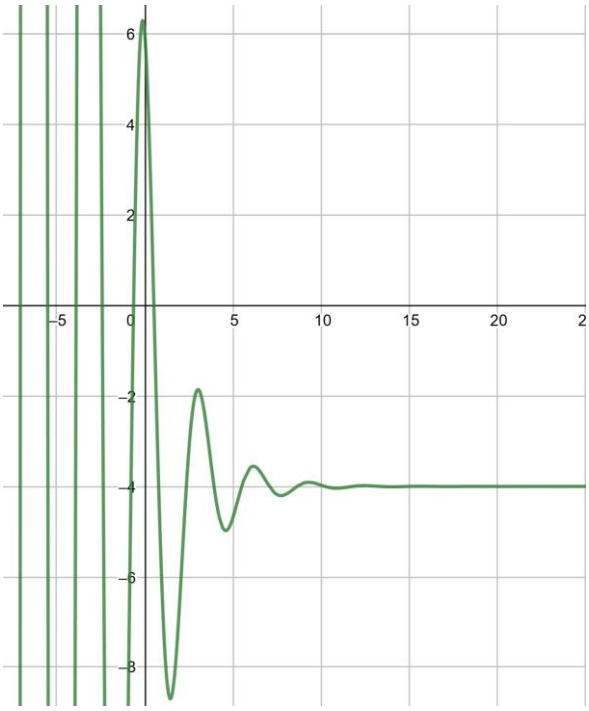

Figura 4-Gráfico de F(x)

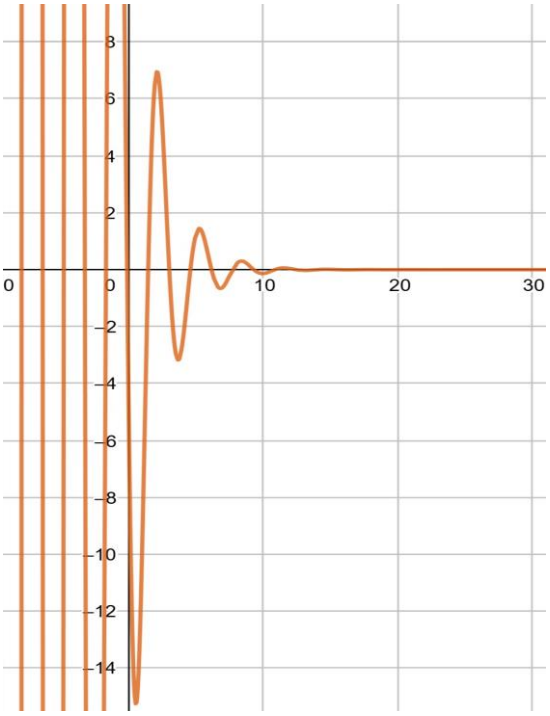

Figura 5-Gráfico de $F'(x)$

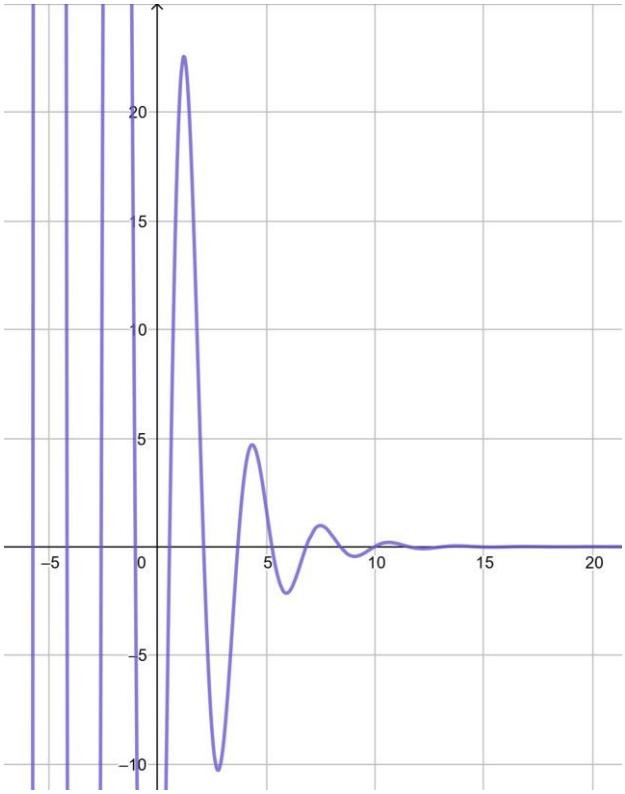

Figura 6-Gráfico de $F''(x)$

A *Figura 7* apresenta o gráfico de $-2ln(\frac{2}{5 \cdot cos2x})$ e a *Figura 8* o gráfico de F(x)=0 ($x = -2ln(\frac{2}{5 \cdot cos2x})$), com um intervalo menor.

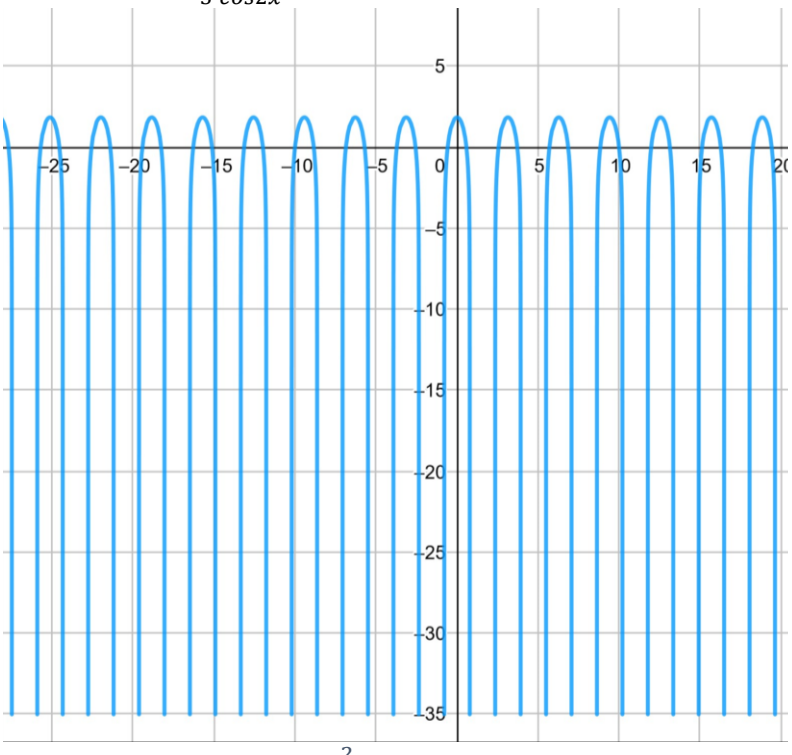

Figura 7-Gráfico de $-2ln(\frac{2}{5 \cdot cos2x})$

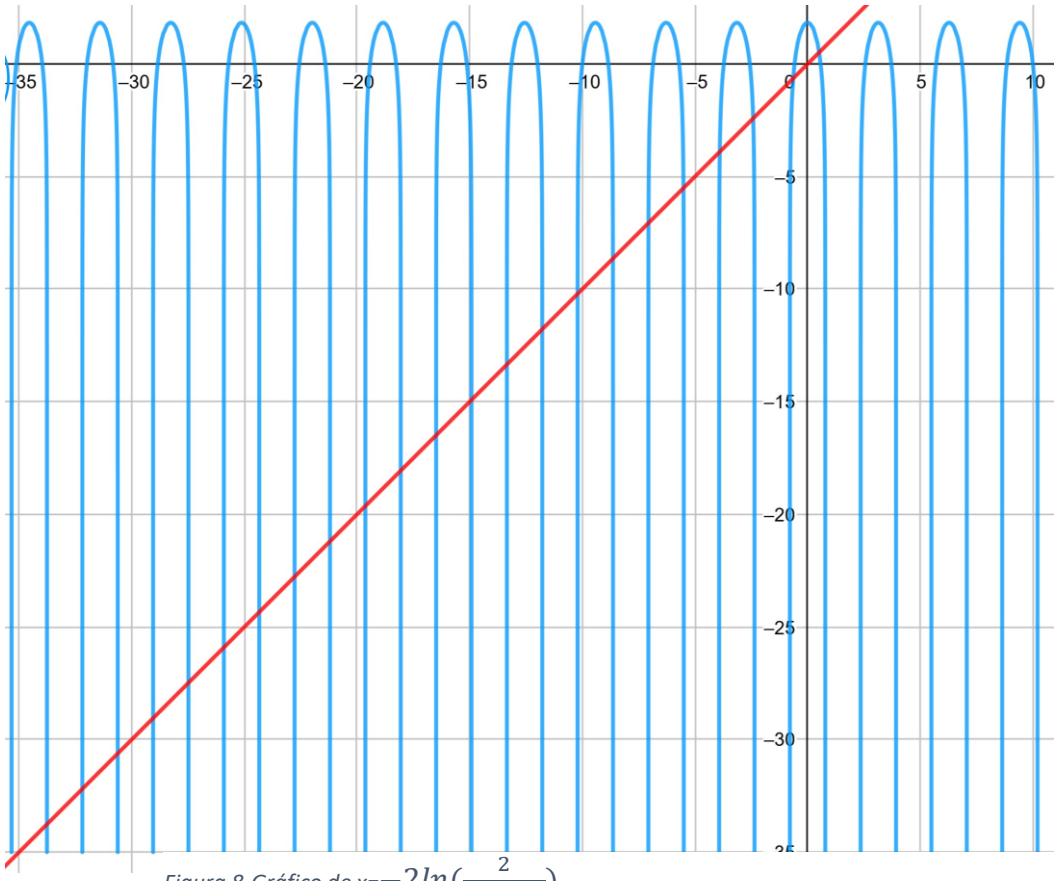

Figura 8-Gráfico de x=$-2ln(\frac{2}{5\cdot cos2x})$

Por observação gráfica, verificamos que existem, pelo menos 23 raízes de F(x)=0 (pontos de intersecção de x com $-2ln(\frac{2}{5\cdot cos2x})$, 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 10^{-1} que contenha uma raiz de F(x)=0 é o intervalo I=[0.5,0.6].

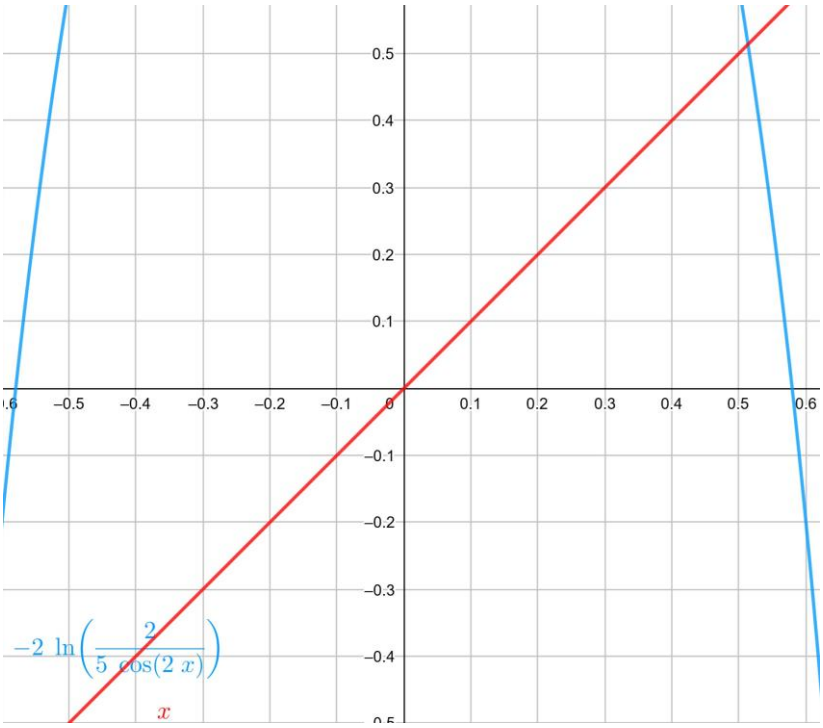

Figura 9- Gráfico de x=$-2ln(\frac{2}{5\cdot cos2x})$

2.3. 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. F, F' e F'' existem e são contínuas em [a,b]
2. $F(a) \cdot F(b) < 0$;
3. $F'(x) \neq 0 \; \forall x \; \in [a, b]$;
4. $F''(x) \geq 0 \; ou \; F''(x) \leq 0 \; \forall x \; \in [a, b]$;
5. 2 opções (ii se i falhar):
   i. $F(x_0) \cdot F'(x_0) > 0, x_0 \in [a, b]$
   ii. $\left|\frac{F(c)}{F'(c)}\right| < b - a$, c é extremo de [a,b]

Por observação das figuras 4-6, F, F' e F'' 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.

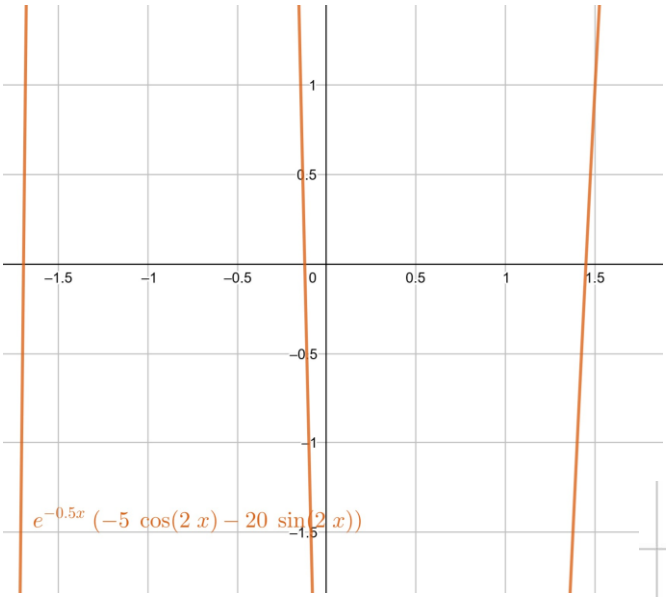

Figura 10-Gráfico de $F'(x)$

A partir da *Figura 10*, verificamos que $F'(x) \neq 0 \; \forall x \; \in [0.5,0.6]$ 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, $F''(0.5) < 0$ e $F''(0.6) > 0$, basta ir diminuindo b até que $F''(b) < 0$.

A *Figura 12* apresenta esse cálculo.

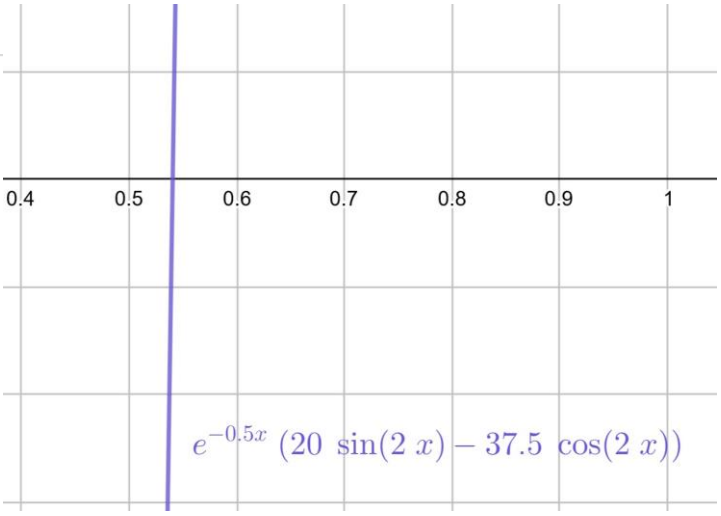

Figura 11-Gráfico de $F''(x)$

x	$F''(x)$
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, $F''(x) < 0 \; \forall x \in [0.5,0.54]$, F(x) não mantem o mesmo sinal no intervalo, o que viola a condição 5.i. sendo necessário testar a condição 5.ii. (ou 5^*).

$\left|\frac{F(c)}{F\prime(c)}\right| < b - a \equiv \left|\frac{F(0.54)}{F'(0.54)}\right| < 0.54 - 0.50 \equiv \left|\frac{-0.40..}{-15.26..}\right| < 0.27 \cong 0.026.. < 0.27$, logo 5^* (**5.ii.**) é satisfeita, sendo $x_0 = 0.54$.

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], pelo que a sucessão gerada por recorrência é convergente.

2.4. 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, $F''(x)$ é monótona crescente e negativa em todo o intervalo (*Figura 13)* sendo assim, $max_{x\in[0.5,0.54]}|F''(x)| = |F''(0.5)| = 2.7741..$, ou seja *x_dder_max* na *Figura 3* será igual a 0.5.

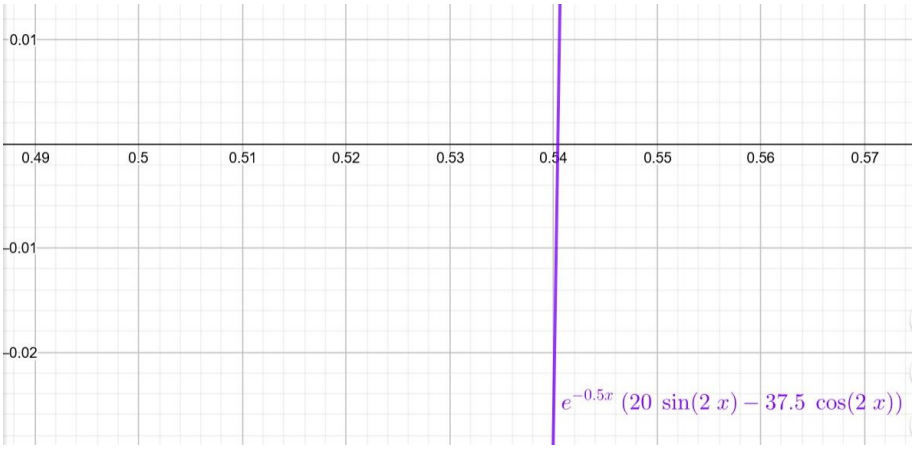

Figura 13-Gráfico de $F''(x)$ para I

Através da *Figura 14*, é visível que, para este intervalo, $F'(0.54)$ será o mínimo, no entanto F' é negativa neste mesmo intervalo, como tal, $min_{x\in[0.5,0.54]}|F'(x)| = |F'(0.5)| = 15.2107..$, ou seja, *x_der_min* na *Figura 3* será igual a 0.5.

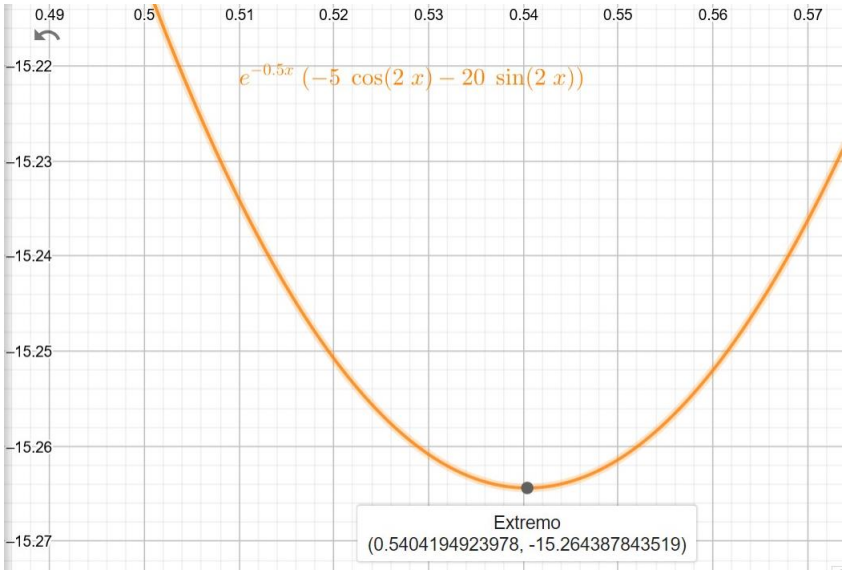

Figura 14-Gráfico de $F'(x)$ em [0.5,0.54]

Sendo assim, $M = \frac{1}{2} \cdot \frac{max_{x\in[0.5,0.54]}|F''(x)|}{min_{x\in[0.5,0.54]}|F'(x)|} = \frac{1}{2} \cdot \frac{|F''(0.5|}{|F'(0.5)|}$ e, sendo erro absoluto inferior a $5x10^{-12}$, 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 $error(5{,}12) = 5x10^{-12}$ e 0.54, 0.5, 0.54, 0.5 e 0.5 são respetivamente, x_0, *a*, *b*, *x_der_min* e *x_dder_max*. A *Figura 15* apresenta a tabela com os resultados.

ε	n	F(x)=0	$\lvert\Delta x_n\rvert$
$5x10^{-12}$	4	0.51365209475085	$2.79x10^{-16}$

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

2.5. Iterações para Majoração inferior a $5x10^{-14}$

Para o intervalo[0.5,0.54] e um erro $\varepsilon = 5x10^{-14}$, o número de iterações n é tal que $|\Delta x_n| \leq M^{2n-1}|\Delta x_0|^{2^n} \leq \varepsilon$ => $n \geq \frac{\ln \alpha}{\ln 2}$ onde $\alpha = \frac{ln\varepsilon + \ln M}{lnM + ln|\Delta x_0|} \approx 3.9393..$ onde $M = \frac{1}{2} \cdot \frac{|F''(0.5|}{|F'(0.5)|}$, correspondendo ao valor obtido pela aplicação do método na *Figura15*.

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

3.1. Algoritmo

Para este problema, consideramos F(x)=0 $\equiv x = f(x)$ onde $f(x) = \frac{1}{2} arcos(\frac{2e^{0.5x}}{5})$, uma vez que, caso $f(x) = -2ln(\frac{2}{5 \cdot cos2x})$, o programa terminava devido a um erro de cálculo no ln, assim sendo, a *Figura 16* é a versão atualizada da *Figura 1*.

```
error = lambda y,x: y*(10**(-x))

F = lambda x: (10*mt.exp(-0.5*x)*mt.cos(2*x) -4) #F(x)

F_deriv = lambda x: mt.exp(-0.5*x)*(-20*mt.sin(2*x)-5*mt.cos(2*x)) #F'(x)

F_dderiv = lambda x: mt.exp(-0.5*x)*(20*mt.sin(2*x)-37.5*mt.cos(2*x)) #F''(x)

f = lambda x: 0.5*(mt.acos((2*mt.exp(0.5*x))/5))#f(x)=1/2*arcos(2e^0.5x/5)
```

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 17-Pseudocódigo

```
def itMethod(x0,eps):
    x1=f(x0)
    err=abs(x1-x0)
    i=1
    while err > eps:
        x0=x1
        x1=f(x0)
        err=abs(x1-x0)
        i+=1
    print("Iterações: "+str(i)+"\n")
    print("Raíz: "+str(dm.Decimal(x1))+"\n")
    print("Erro: "+str(dm.Decimal(err))+"\n")
    return
```

Figura 18-Código

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 x_0 de **2.5.** (0.54) e como erro/eps $5x10^{-12}$. A *Figura 19* apresenta a tabela com os resultados do método.

ε	n	F(x)=0	$\lvert\Delta x_n\rvert$
$5x10^{-12}$	13	0.5136520947528	$4.31x10^{-12}$

Figura 19- Tabela de resultados aplicando método iterativo simples para cálculo de raiz de F(x)=0 para $x_0 = 0.54$

4. Conclusões

É necessário referir que, as condições de aplicabilidade do método iterativo simples não são necessárias para a sucessão convergir (pode convergir ou não, falhando o teorema), no entanto, se passar no teorema do método iterativo, fica garantida a convergência da sucessão. Para este problema não foi testada nenhuma condição de aplicabilidade, no entanto, dado os resultados, podemos concluir que a sucessão converge para uma raiz (sendo esta próxima da obtida pelo método de Newton), assim sendo, podemos assumir que a função satisfaz as condições de aplicabilidade do método iterativo simples.

O método de Newton pode ser visto como uma aplicação iterativa simples, rapidamente convergente, para poder ser aplicado, este exige muito a uma função (daí a dificuldade em encontrar um intervalo que valide todas as condições de aplicabilidade), no entanto, quando um dado intervalo e uma função passam nesses testes, fica automaticamente garantida a convergência da sucessão (nesse intervalo), sendo assim, um forte indício de que a função passa nas condições de aplicabilidade do método iterativo simples.

Pela análise e comparação dos resultados das figuras *15* e *19*, podemos imediatamente observar a diferença no número de iterações para o método encontrar a raiz, para o método de Newton, são necessárias 4 iterações, no entanto, para o método iterativo simples, são 13 iterações, provando assim que o método de Newton é, dos dois, o método que converge mais rapidamente para a raiz.

Um resultado interessante, é o facto de, no método de Newton, com apenas 4 iterações somos capazes de obter para a mesma raiz um erro muito menor ($2.79x10^{-16}$) que o obtido pelo método iterativo simples em 13 iterações ($4.31x10^{-12}$), daí a existência de alguma diferença entre as raízes, no entanto, é importante referir que o critério de paragem do método iterativo aplicado na *Figura 18* não é o critério mais preciso, tendo sido escolhido devido às restrições impostas no enunciado)

Podemos então concluir que o método de Newton, é o método (dos dois que aplicamos neste trabalho) que produz resultados mais precisos e no menor número de iterações possível, razão essa que pode ser em parte explicada pelas restrições que o método impõem à função em estudo.