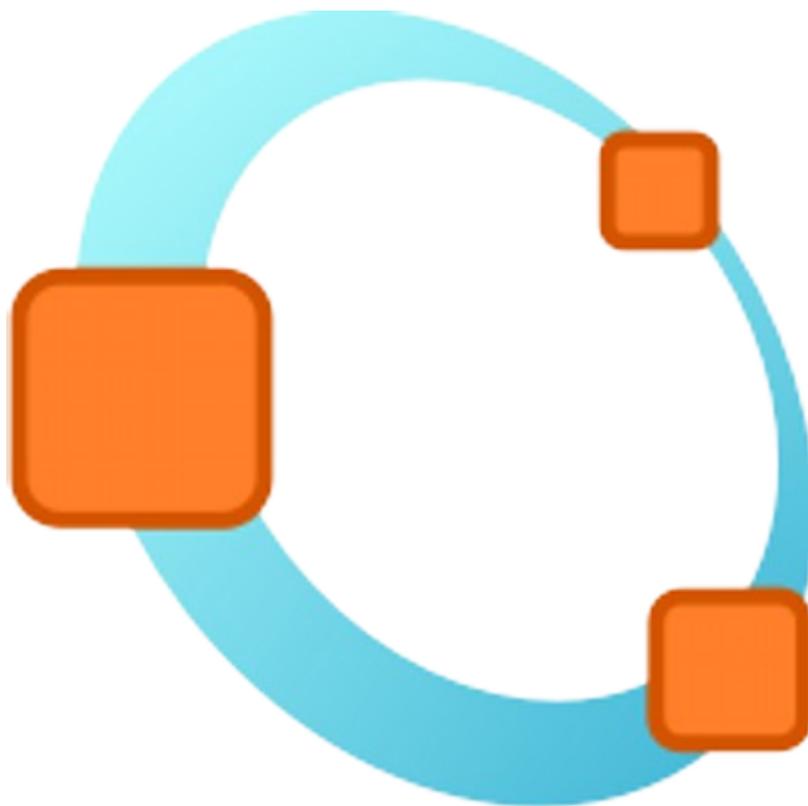


# Computação Numérica

e-Fólio A

22 Novembro 2015



**Marco Paulo Silva Martins**  
**1100619**

## Índice

Nota introdutória:.....	3
Pergunta 1:.....	4
Pergunta 2:.....	5
Pergunta 3:.....	6
Pergunta 4:.....	7

**Nota introdutória:**

Informação sobre o sistema onde resolvi o eFólio:

```
octave:10> OCTAVE_VERSION ()  
ans = 3.8.1
```

```
octave:11> uname  
ans =
```

scalar structure containing the fields:

```
sysname = Linux  
nodename = Asus  
release = 3.13.0-24-generic  
version = #47-Ubuntu SMP Fri May 2 23:30:00 UTC 2014  
machine = x86_64
```

Linux Mint 17 Qiana

**Pergunta 1:**

Para a função  $\cos x = e^{ax} - 1$  tem-se  $h(x) = \cos x$  e  $g(x) = e^{ax} - 1$  sabe-se que os valores da abcissa dos pontos de intersecção das funções  $h(x)$  e  $g(x)$  são as raízes da função.

Como a função  $\cos x$  varia entre 1 e -1 é óbvio que não vão existir intersecções quando  $e^{ax} > 1$ . Se iterar os valores de  $x$  de  $\pi$  em  $\pi$  cada iteração vai incluir os troços positivos, por exemplo de  $0 \rightarrow \pi$  tem incluído o troço positivo  $1 \rightarrow 0$  e de  $\pi \rightarrow 2\pi$  tem incluído o troço positivo  $0 \rightarrow 1$ .

Assim vou iterar os valores de  $x$  de  $\pi$  em  $\pi$  dentro de um ciclo com a condição  $e^{ax} - 1 \leq 1$  e incrementar 1 no numero de raízes da função.

O ficheiro é o nraizes.m

exemplo de uma execução:

```
octave:3> nraizes(0.03);
A função tem 7 raízes!
```

A função:

```
function n = nraizes(a)
    n = 0;
    x = pi;
    while (exp(a.*x) - 1 <= 1)
        n += 1;
        x += pi;
    end
    if n == 0
        fprintf('A função não tem raízes!\n',n)
    else
        fprintf('A função tem %d raízes!\n',n)
    end
end
```

**Pergunta 2:**

A função para determinar a matriz com os intervalos que contem as raízes da função está centrada num ciclo que corre até a matriz estar preenchida.

A matriz X é iniciada com o numero de linhas igual ao numero de raízes recebido da função nraizes(a) e 2 colunas e preenchida com zeros. O intervalo inicial é  $[0, 1] \times \text{Pi}/2$ .

Enquanto a matriz não tem as linhas todas preenchidas se o coseno de um dos valores do intervalo for 1, como a distancia entre os valores do intervalo é  $\text{Pi}/2$  certamente que o coseno do outro valor também é maior ou igual a zero, logo grava esse intervalo na matriz e incrementa o intervalo e a linha.

Por exemplo o intervalo  $[0, 1] \times \text{Pi}/2$  como o  $\cos(0 \times \text{Pi}/2) == 1$  o valor do  $\cos(1 \times \text{Pi}/2)$  certamente é zero portanto sempre que um dos cosenos é 1 estamos perante um troço positivo ou  $0 \rightarrow 1$  ou  $1 \rightarrow 0$  ou seja um intervalo com uma raiz da função logo adiciona-se este intervalo na matriz e incrementa-se o numero da linha.

O ficheiro é intervalos.m

exemplo de uma execução:

```
octave:20> x = intervalos(0.03)
```

```
A função tem 7 raízes!
```

```
x =
```

```

0  1
3  4
4  5
7  8
8  9
11 12
12 13
```

A função:

```
function X = intervalos(a)
    k = pi./2;
    n = nraizes(a);
    X = zeros(n, 2);
    l = 1;
    Xmin = 0;
    Xmax = 1;

    while (l <= n)
        if( ((cos(Xmin .* k)) == 1) || ((cos(Xmax .* k)) == 1))
            X(l,1) = Xmin;
            X(l,2) = Xmax;
            l += 1;
        end
        Xmin += 1;
        Xmax += 1;
    end
end
```

**Pergunta 3:**

Para calcular as raízes da função escrevi a função na forma  $f(x)=h(x) - g(x)$ , recorri à função intervalos(a) para obter a matriz com todos os intervalos que contem zeros e multipliquei por  $\pi/2$  de forma a ficar cada linha no formato **[Xmin, Xmax]  $\pi/2$**

O ciclo for corre todas as linhas da matriz e chama o método da secante em cada iteração enviando por argumento a função, o Xmin, o Xmax, o erro e o máximo de iterações.

Os resultados vem no formato [raiz, erro, iteração] e são guardados numa matriz e por fim retiro a primeira coluna da matriz de resultados que tem todos os zeros da função.

Quanto ao método da secante fiz recorrendo ao algoritmo do manual da UC. A forma de o chamar é baseado nos exercícios das AF's

**[r,e,i] = metodo\_secante( funcao, Xmin, Xmax, erro, kmax );**

Os ficheiros é **raizes.m** , **metodo\_secante.m**

exemplo de uma execução:

```
octave:44> r = raizes(0.03)
A função tem 7 raízes!
[ Raiz = 1.523975e+00 | erro = 1.638627e-08 | iteracoes = 3 ]
[ Raiz = 4.870373e+00 | erro = 1.527412e-10 | iteracoes = 5 ]
[ Raiz = 7.595276e+00 | erro = 3.762878e-08 | iteracoes = 4 ]
[ Raiz = 1.141633e+01 | erro = 4.104647e-10 | iteracoes = 6 ]
[ Raiz = 1.360869e+01 | erro = 1.394686e-08 | iteracoes = 5 ]
[ Raiz = 1.808301e+01 | erro = 5.741678e-08 | iteracoes = 7 ]
[ Raiz = 1.950121e+01 | erro = 4.545522e-08 | iteracoes = 6 ]
r =

    1.5240
    4.8704
    7.5953
   11.4163
   13.6087
   18.0830
   19.5012
```

A função:

```
function r = raizes(a)
    funcao = @(x) cos(x) - (exp(a.*x) -1);
    X = intervalos(a) * pi/2;
    erro = 1*10^(-7);
    kmax = 50;
    matriz = [];
    for i = 1:size(X) (1,1)
        [r,e,i] = metodo_secante(funcao,X(i,1),X(i,2),erro,kmax);
        matriz = [matriz; [r,e,i]];
    end

    r = matriz(:,1);
end
```

**Pergunta 4:**

Para desenhar o gráfico de demonstração fiz o gráfico da função  $\cos(x) - (e^{(a \cdot x)} - 1)$  e o gráfico com os pontos correspondentes às raízes no mesmo plot

```
plot(x,y,'cos(x) - (e^(a.*x) -1)'; r,0, 'marker', '*', 'color', 'r');
```

No primeiro os valores de x vão de 0 a 50 de 0.1 em 0.1

e os valores de y são  $\cos(x) - (\exp(a \cdot x) - 1)$

No segundo os valores de x são as raízes

e os valores de y é 0 com o marker definido como '\*' e a cor definida como 'r' (red).

Os valores das raízes são obtidos recorrendo à função raizes()

O ficheiro é **demonstracao.m**

exemplo de uma execução:

```
octave:56> demonstracao
```

```
Introduza o valor de a 0.015
```

```
A função tem 14 raízes!
```

```
[ Raiz = 1.547310e+00 | erro = 5.555348e-10 | iteracoes = 3 ]
[ Raiz = 4.786953e+00 | erro = 4.808241e-08 | iteracoes = 4 ]
[ Raiz = 7.730720e+00 | erro = 5.733928e-11 | iteracoes = 4 ]
[ Raiz = 1.117918e+01 | erro = 4.590966e-10 | iteracoes = 5 ]
[ Raiz = 1.390320e+01 | erro = 1.790667e-08 | iteracoes = 4 ]
[ Raiz = 1.758548e+01 | erro = 6.795213e-08 | iteracoes = 5 ]
[ Raiz = 2.006160e+01 | erro = 6.377653e-11 | iteracoes = 5 ]
[ Raiz = 2.401037e+01 | erro = 5.868647e-10 | iteracoes = 6 ]
[ Raiz = 2.620122e+01 | erro = 9.447297e-09 | iteracoes = 5 ]
[ Raiz = 3.046293e+01 | erro = 4.115916e-12 | iteracoes = 7 ]
[ Raiz = 3.231327e+01 | erro = 5.717878e-11 | iteracoes = 6 ]
[ Raiz = 3.696286e+01 | erro = 2.916412e-08 | iteracoes = 7 ]
[ Raiz = 3.837788e+01 | erro = 5.519898e-08 | iteracoes = 6 ]
[ Raiz = 4.358688e+01 | erro = 1.239248e-08 | iteracoes = 9 ]
```

**Marco Paulo Silva Martins**  
1100619