Métodos Numéricos para Engenharia

MÓDULO 1 - RAÍZES DE EQUAÇÕES – BISSEÇÃO ABS

PROFESSOR LUCIANO EMÍDIO DA FONSECA

Definições

Uma equação não linear pode ser representada na forma:

$$f(x) = 0$$

As soluções da equação acima, isto é, valores de x que tornam a equação nula, são denominadas de raízes da equação.

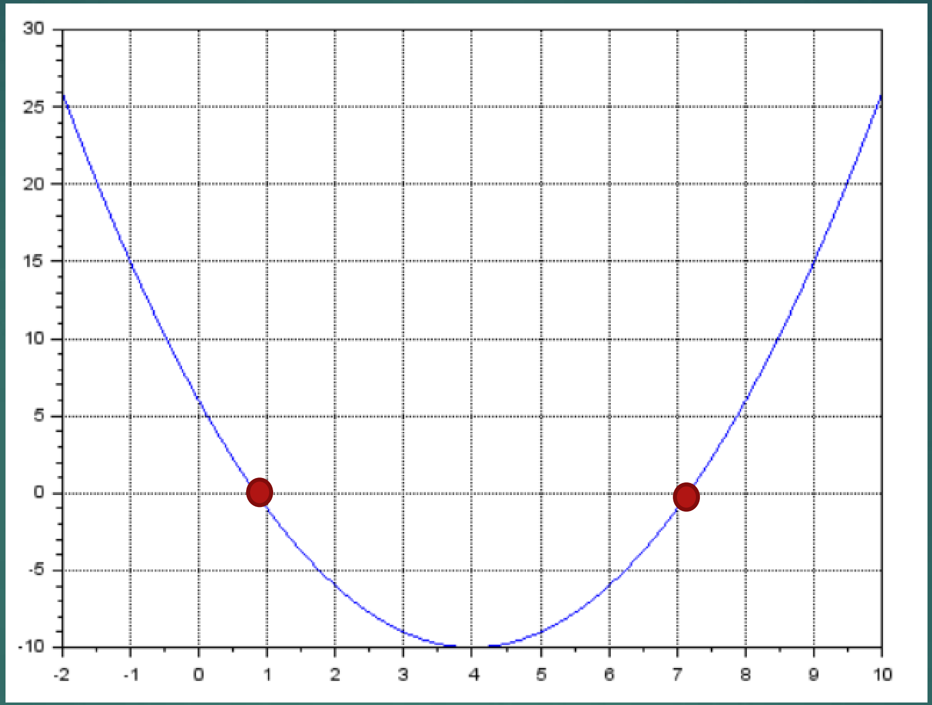
As raízes podem ser reais ou complexas, simples ou múltiplas.

Considere a seguinte equação não linear, para a qual queremos encontrar as raízes :

$$y = f(x) = x^2 - 8x + 6$$

Podemos plotar um gráfico

```
--> deff('y=f(x)','y=x^2-8*x+6');

--> x=[-2:10];

--> plot(x,f(x))

--> xgrid()
```



E fazer um tabela

```
--> x=[-2:10];

--> [x' f(x)']
 ans  =

  -2.    26.
  -1.    15.
   0.     6.
   1.    -1.
   2.    -6.
   3.    -9.
   4.   -10.
   5.    -9.
   6.    -6.
   7.    -1.
   8.     6.
   9.    15.
   10.   26.
```

Inversão
De sinal!!

E encontrar as raízes por Baskara

$$y = f(x) = x^2 - 8x + 6$$

```
--> a=1;

--> b=-8;

--> c=6;

--> delta=b^2-4*a*c
 delta  =

   40.

--> r1=(-b+sqrt(delta))/(2*a)
 r1  =

   7.1622777

--> r2=(-b-sqrt(delta))/(2*a)
 r2  =

   0.8377223
```

Considere agora esta equação não linear, para a qual queremos encontrar uma raiz real:

$$y = 4e^{\frac{x}{3}} - 20\mathrm{e}^{-\left(\frac{\mathrm{x}}{5}\right)}\sin(\mathrm{x})$$

Não há uma método direto para resolver a equação acima.

Uma primeira abordagem seria então montar uma tabela com pares ordenados (x, y=f(x)) espaçados de um intervalo constante. Se para dois valores consecutivos de x na tabela, digamos x_a e x_b, os valores correspondente de y (y_a e y_b) tiverem sinais opostos, podemos concluir que há uma raiz entre o valor x_a e x_b, pois a função y=f(x) tem que cortar o eixo neste intervalo para mudar de sinal.

```
--> deff ('y=f(x)' , 'y=4*exp(x/3)-20*exp(-x/5).*sin(x)')

--> x=[-10:10];

--> [x' f(x)']
 ans  =

  -10.   -80.253354
  -9.     50.062579
  -8.     98.284403
  -7.     53.672129
  -6.    -18.012501
  -5.    -51.377026
  -4.    -32.63151
  -3.     6.6142662
  -2.     29.183916
  -1.     23.421625
   0.     4.
   1.    -8.1963138
   2.    -4.3994697
   3.     9.3241613
   4.     21.975737
   5.     28.233331
   6.     31.239391
   7.     38.008816
   8.     53.572705
   9.     78.979693
   10.    113.599
```

A partir deste critério, vemos que há raízes reais nos intervalos:

[-10, -9)]

[-7, -6]

[-4,-3]

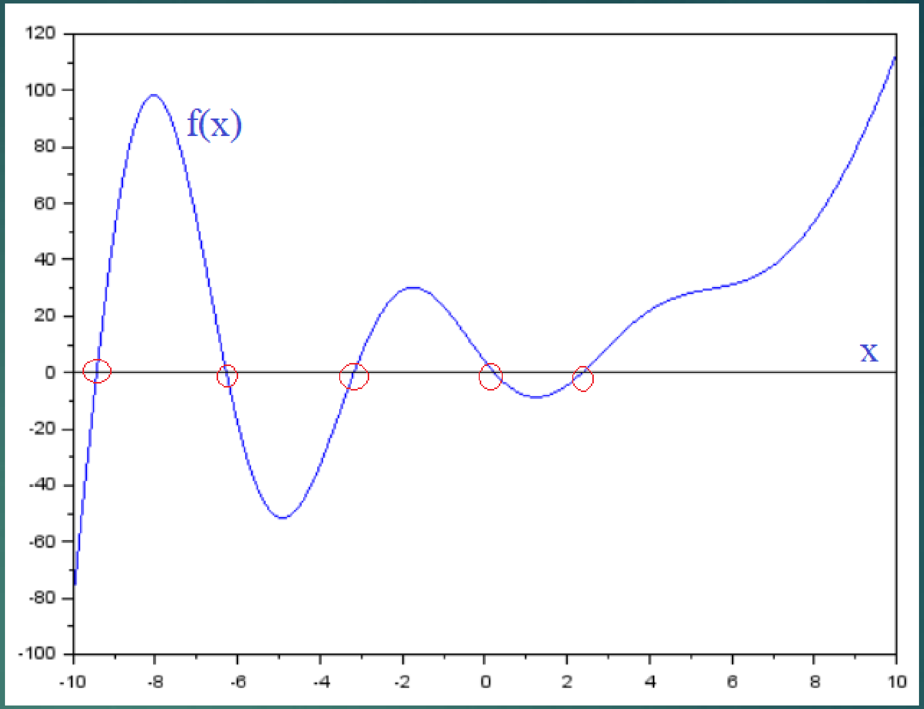
[0,1]

[2, 3]

Pois nestes intervalos, os valores de y=f(x) trocam de sinal.

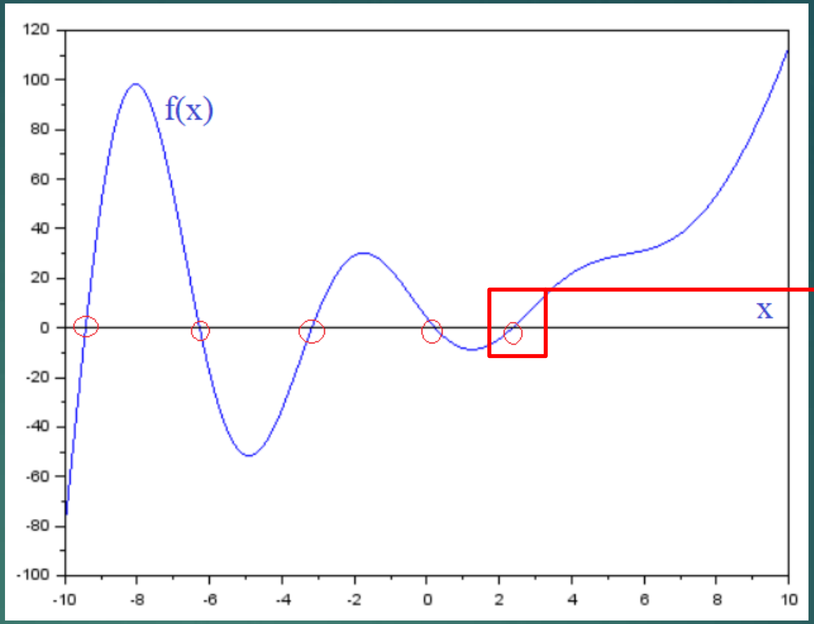
```
--> deff ('y=f(x)' , 'y=4*exp(x/3)-20*exp(-x/5).*sin(x)')

--> x=[-10:0.01:10];

--> plot(x,f(x))
```

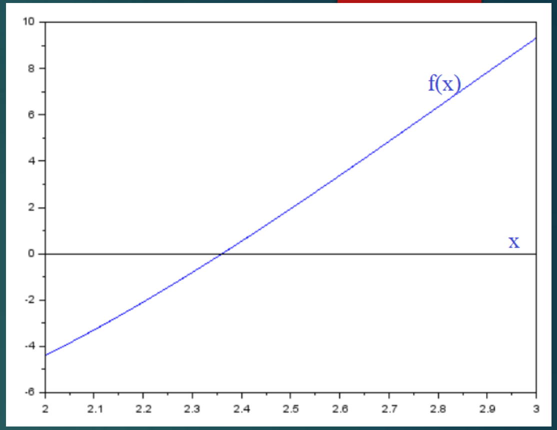
```
--> deff ('y=f(x)' , 'y=4*exp(x/3)-20*exp(-x/5).*sin(x)')

--> x=linspace(-10,10,1000);

--> plot(x,f(x))
```



- Para entendermos melhor podemos traçar um gráfico de y=f(x) no intervalo [-10,10].
- Vemos claramente que a curva cruza o eixo horizontal nos intervalos [-10,-9], [-7,-6], [-4,-3], [0,1] e [2,3], que são as mesmas conclusões que tivemos ao analisarmos a tabela.
- Podemos refinar a análise, por exemplo , no intervalo [2, 3]

```
--> deff ('y=f(x)' , 'y=4*exp(x/3)-20*exp(-x/5).*sin(x)')

--> x=[2:0.05:3];

--> y=f(x);

--> [x' y']
 ans  =

   2.     -4.3994697
   2.05   -3.8560919
   2.1    -3.2883686
   2.15   -2.6978195
   2.2    -2.0859863
   2.25   -1.4544274
   2.3    -0.804714
   2.35   -0.1384253
   2.4     0.5428558
   2.45    1.2375466
   2.5     1.9440695
   2.55    2.6608553
   2.6     3.3863478
   2.65    4.1190073
   2.7     4.8573149
   2.75    5.5997753
   2.8     6.3449212
   2.85    7.0913161
   2.9     7.8375576
   2.95    8.5822809
   3.      9.3241613

--> plot(x,y)
```



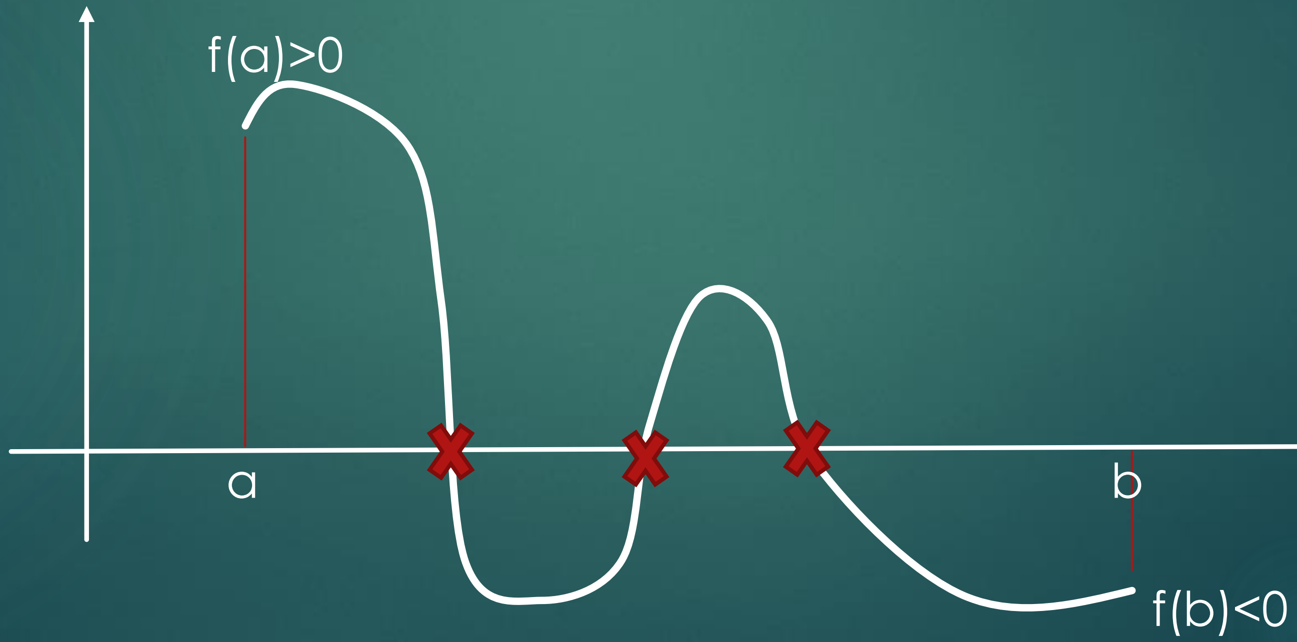


- Para entendermos melhor podemos a refinar tabela no intervalo [2 , 3], e plotar um gráfico com melhor resolução.
- Vemos agora que a curva cruza o eixo horizontal no [2.35, 2.40];
- Podemos prosseguir com esta a análise, subdividindo os intervalor até atingirmos uma resolução adequada.
- Esta é a lógica por traz dos métodos intervalares.

Métodos Intervalares

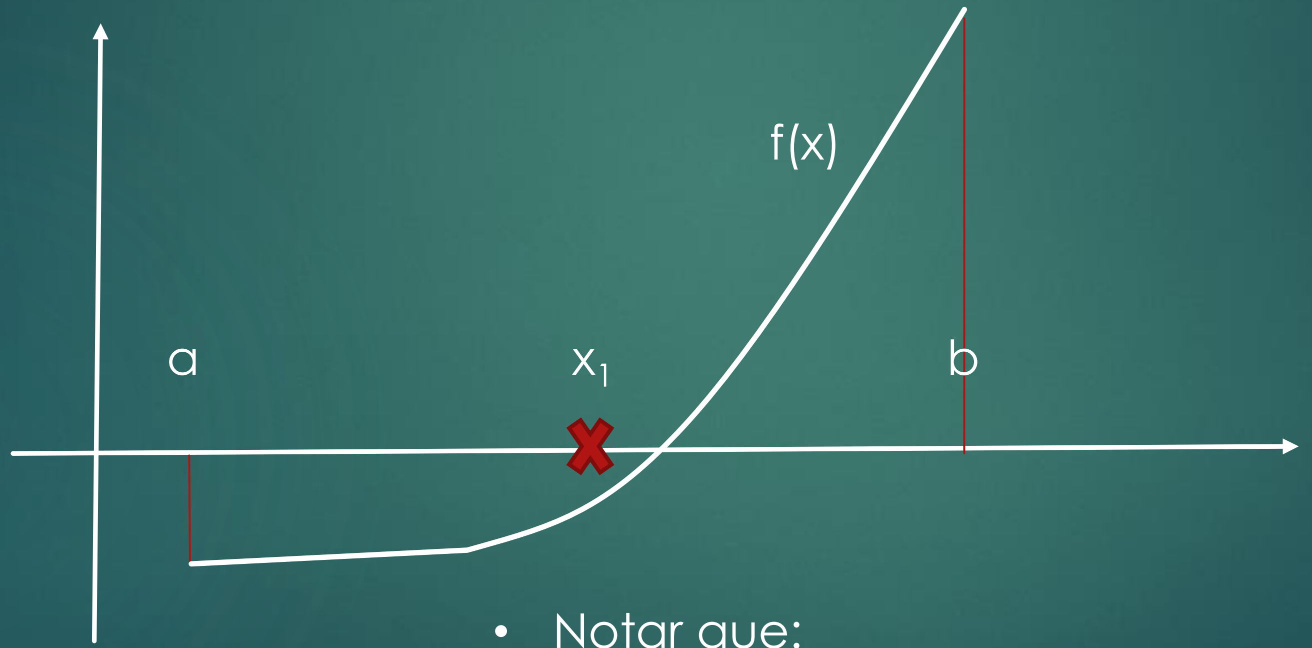
Os métodos intervalares são adequados para o cálculo de raízes reais e não-múltiplas (raízes simples).

- Se f(x) for contínua no intervalo [a,b] e se f(a)f(b)<0, então existe pelo menos uma raiz real no intervalo [a,b], isto é, existe pelo menos um ponto tal que f(x)=0.



Método da Bisseção

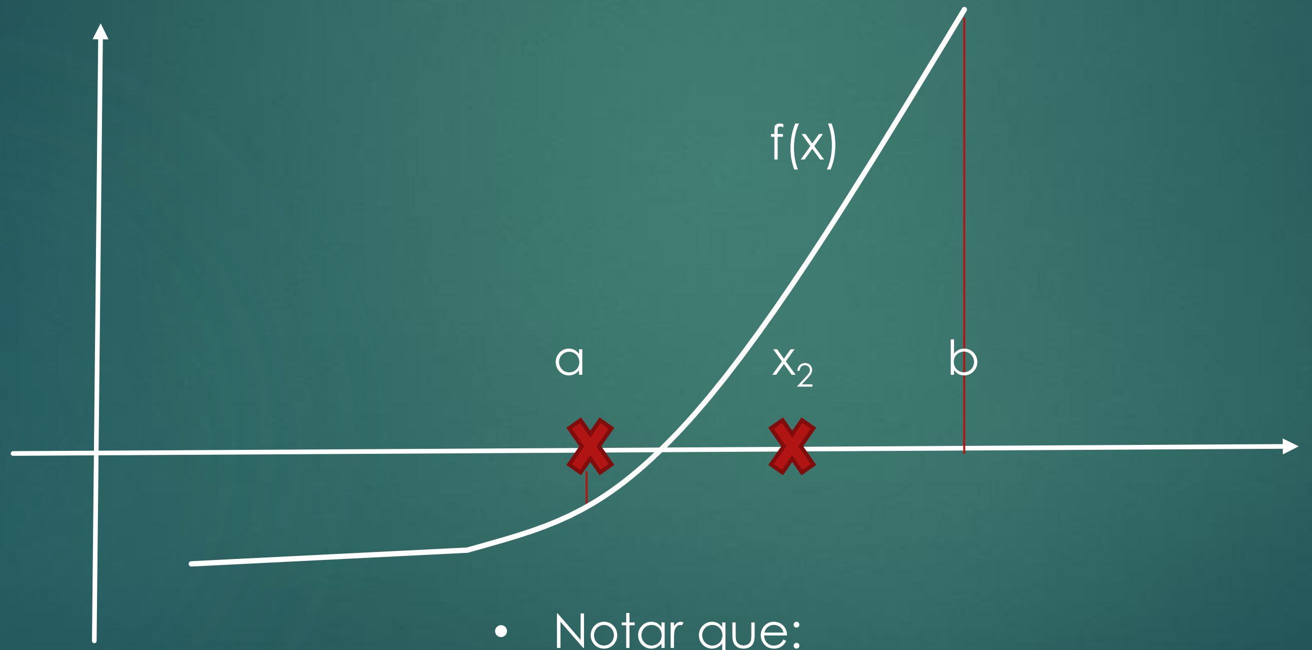
- O método Bisseção é o mais simples e o mais seguro
- Começa-se com um intervalo [a,b] que contenha uma raiz.
- A próxima estimativa para a raiz é simplesmente o meio do intervalo anterior.



- Notar que:
- x_1 está na metade do intervalo [a,b]

Método da Bisseção

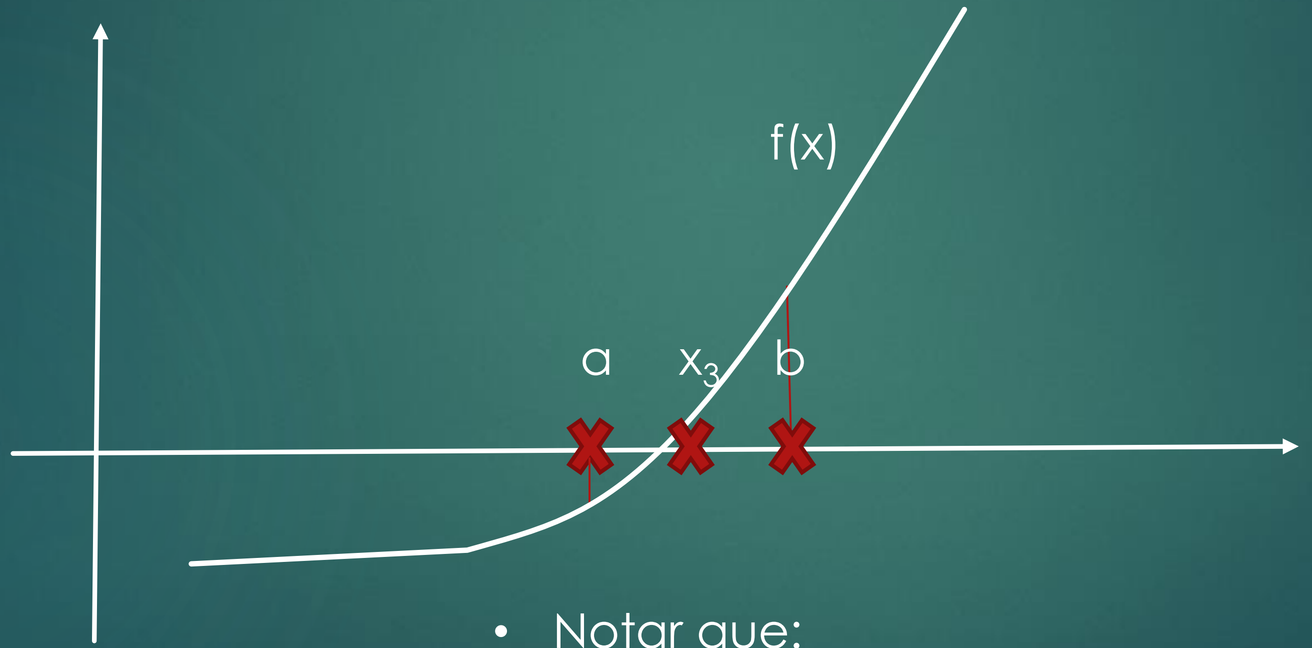
- O método Bisseção é o mais simples e o mais seguro
- Começa-se com um intervalo [a,b] que contenha uma raiz.
- A próxima estimativa para a raiz é simplesmente o meio do intervalo anterior.



- Notar que:
- x_1 é o novo "a"
- x_2 está na metade do intervalo [a,b]

Método da Bisseção

- O método Bisseção é o mais simples e o mais seguro
- Começa-se com um intervalo [a,b] que contenha uma raiz.
- A próxima estimativa para a raiz é simplesmente o meio do intervalo anterior.



- Notar que:
- x_2 é o novo “b”
- x_3 está na metade do intervalo [a,b]

Métodos Intervalares - Bisseção com Erro Absoluto

- Escolha um intervalo inicial [a,b] para se avaliar a função $f(x)$ tal que f(a)f(b)<0
- Defina uma tolerância (precisão) desejada

1 – Faça uma estimativa para a raiz da equação

$$x_o = \frac{a+b}{2}$$

2 – Se f(a)f(x_o) <0

Então $b = x_o$

Senão $a = x_o$

3 - Se $|a - b| <$ tolerância

Raiz =x_o

Pare

4 – Repita passo 1

```
function x1=Bissecao_abs(f,a,b,tol,prt) //erro absoluto
    if (f(a)*f(b)>0) then x1=[]; return end
    if (prt)  printf ('i \ta\t\t\tx1\t\t\tb\t\t\terro\n') end
    k=1
    while %t
        erro=abs(b-a)
        x1=(a+b)/2     // Bisseção
        if (prt)
            printf ("%d\t%.10f(%2d)\t%.10f(%2d)\t%.10f(%2d)\t%.1e\n",...
            k,a,sign(f(a)),x1,sign(f(x1)),b,sign(f(b)),erro)
        end
        if  ( (erro<tol) | (f(x1)==0) )  break  end
        if     f(x1)*f(a)<0  b=x1
        else   a=x1          end
        k=k+1
    end
endfunction
```

Algoritmo Scilab para o método da Bisseção com erro absoluto.

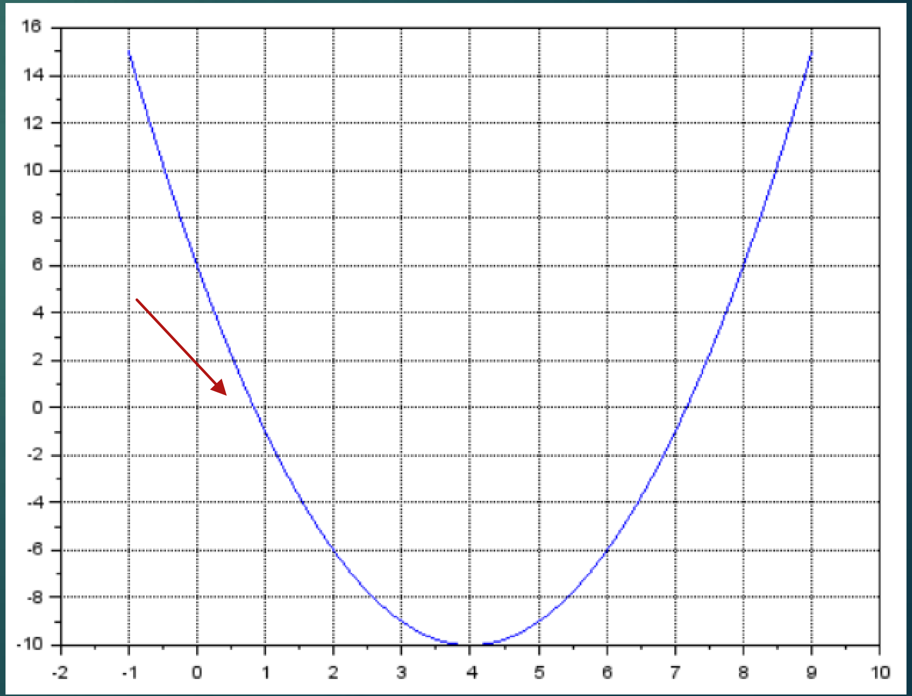
- f é a função para a qual se que calcular a raiz
- [a,b] é o intervalo inicial e tol a tolerância

$$f(x) = x^2 - 8x + 6 = 0$$

```
--> deff ('y=f(x)','y=x^2-8*x+6')

--> xr=Bissecao_abs(f,0,1,0.1,%F)
 xr  =

   0.84375

--> xr=Bissecao_abs(f,0,1,0.1,%T)
i        a                    x1                   b                    erro
1        0.0000000000( 1)     0.5000000000( 1)     1.0000000000(-1)     1.0e+00
2        0.5000000000( 1)     0.7500000000( 1)     1.0000000000(-1)     5.0e-01
3        0.7500000000( 1)     0.8750000000(-1)     1.0000000000(-1)     2.5e-01
4        0.7500000000( 1)     0.8125000000( 1)     0.8750000000(-1)     1.3e-01
5        0.8125000000( 1)     0.8437500000(-1)     0.8750000000(-1)     6.3e-02
 xr  =

   0.84375
```

```
--> deff ('y=f(x)' , 'y=x^2-8*x+6')

--> x=[-2:10];

--> [x' f(x)']
 ans  =

  -2.    26.
  -1.    15.
   0.     6.
   1.    -1.
   2.    -6.
   3.    -9.
   4.   -10.
   5.    -9.
   6.    -6.
   7.    -1.
   8.     6.
   9.    15.
   10.   26.
```

```
--> deff ('y=f(x)' , 'y=x^2-8*x+6')

--> x=linspace(-1,9,1000);

--> plot(x,f(x))

--> xgrid()
```



Exemplo Cálculo da Raiz por Bisseção

- $f(x) = x^2 - 8x + 6 = 0$
- $critério\ de\ parada:\ tolerância =\ \varepsilon =$ | b-a | < 0.1
- Intervalo Inicial [a,b]=[0, 1]
- Como f(a)f(b) < 0 então há uma raiz no intervalo

1) $x_1 = \frac{a+b}{2} = \frac{0+1}{2} = 0.5$

$\varepsilon =$ | b-a | $= |1.0 - 0.0| = 1.0$

$f(0.0) = 6$

$f(0.5) = 2.25$

$f(1.0) = -1$

Novo Intervalo [x_1,b]=[0.5,1]

2) $x_2 = \frac{a+b}{2} = \frac{0.5+1}{2} = 0.75$

$\varepsilon =$ | b-a | $= |1.0 - 0.5| = 0.5$

$f(0.5) = 2.25$

$f(0.75) = 0.5625$

$f(1.0) = -1$

Novo Intervalo [x_2,b]=[0.75,1]

3) $x_3 = \frac{a+b}{2} = \frac{0.75+1}{2} = 0.875$

$\varepsilon =$ | b-a | $= |1.0 - 0.75| = 0.25$

$f(0.75) = 0.5625$

$f(0.875) = -0.2344$

$f(1) = -1$

Novo Intervalo [a, x_3]=[0.75 , 0.875]

4) $x_4 = \frac{a+b}{2} = \frac{0.75+0.875}{2} = 0.8125$

$\varepsilon = |b-a| = |0.875 - 0.75| = 0.125$

$f(0.75) = 0.5625$

$f(0.8125) = 0.1602$

$f(0.875) = -0.2344$

Novo Intervalo [x_4,b]=[0.8125 , 0.875]

5) $x_5 = \frac{a+b}{2} = \frac{0.8125+0.875}{2} = 0.84375$

$\varepsilon = |b-a| = |0.875 - 0.8125| = 0.0625$

Parar, pois ε<0.1

Raiz = $0.84375A \cong\ 0.844$

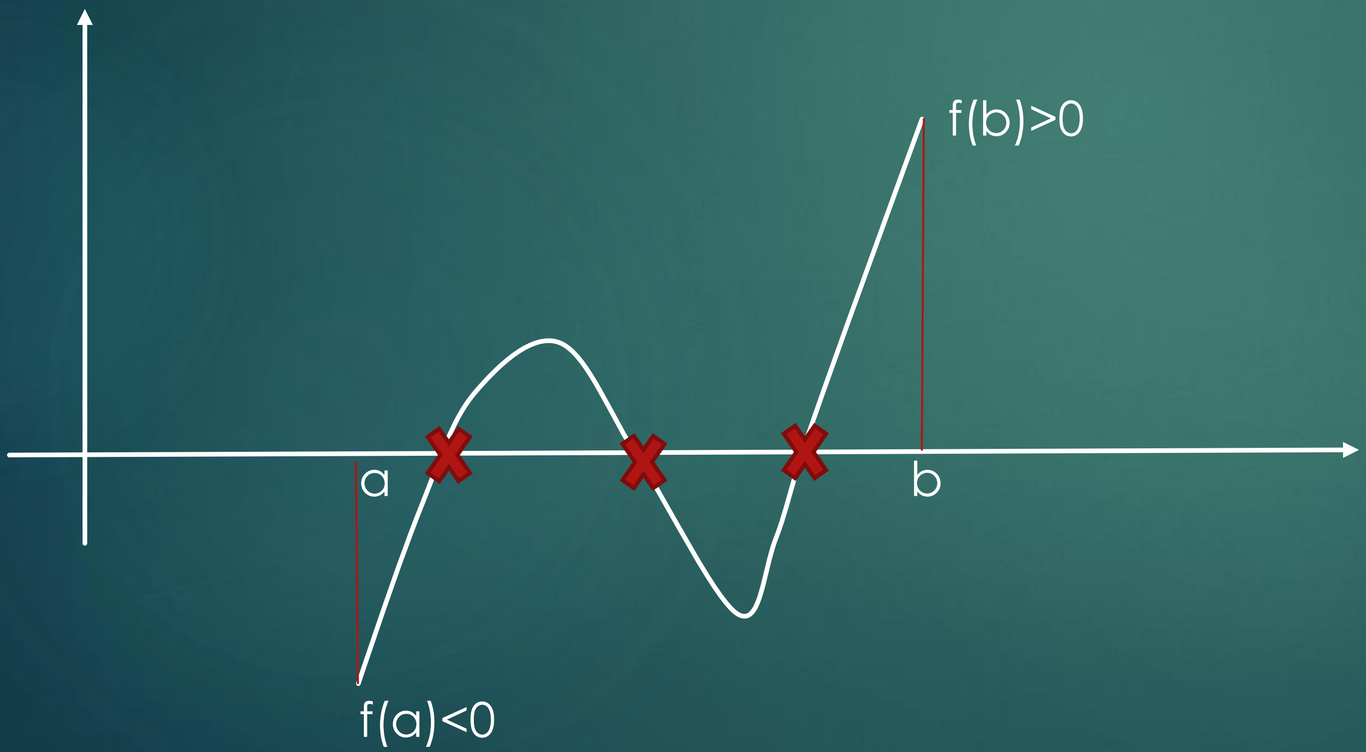
Situações nas quais os métodos Intervalares podem falhar, pela escolha equivocada do intervalo inicial

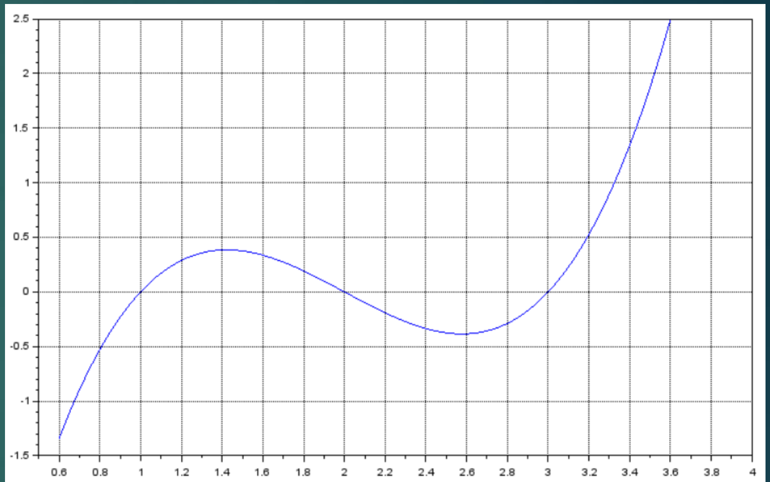
$$y = x^3 - 6x^2 + 11x - 6$$

```
--> deff ('y=f(x)' , 'y=x^3-6*x^2+11*x-6')

--> x=linspace(0.6,3.6,1000);

--> plot(x,f(x))

--> xgrid()
```

1) Número ímpar de raízes no intervalo!

Apesar de f(a)f(b)<0, o método vai convergir para apenas uma das raiz.






```
--> xr=Bissecao_abs(f,0.6,3.6,1e-8)
 xr  =

   3.0000000
```

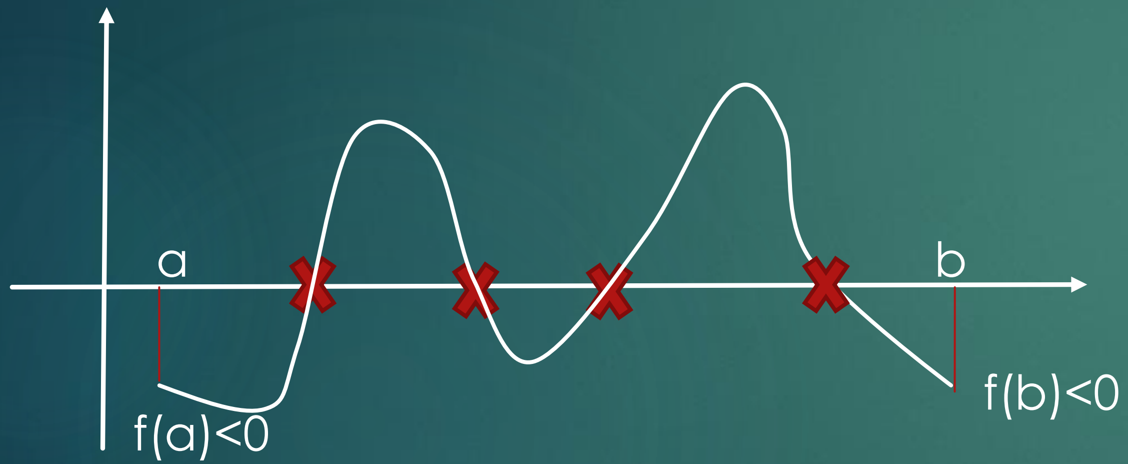
?

```
--> xr=Bissecao_abs(f,0.5,3.5,1e-8)
 xr  =

   2.
```

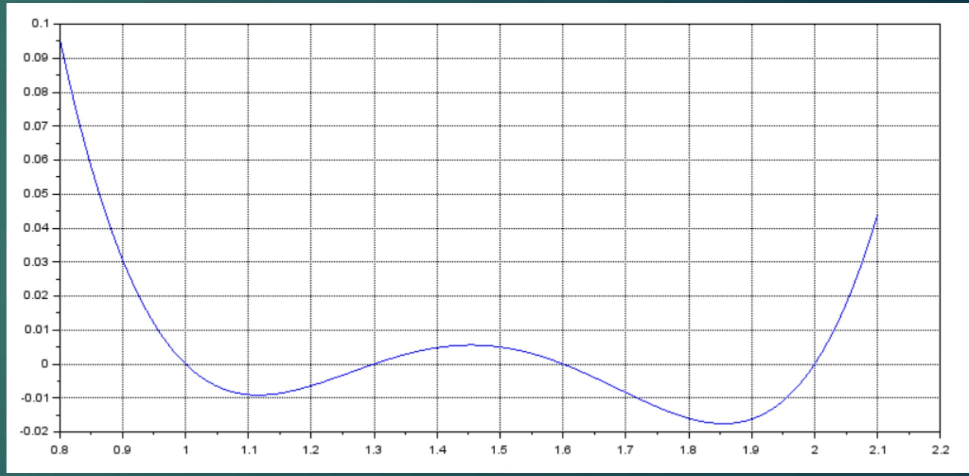
?

2) Número par de raízes no intervalo!

Como f(a)f(b)>0, o método não vai começar; e por conseguinte não vai encontrar nenhuma raiz.




```
--> deff ('y=f(x)' , 'y=x^4-5.9*x^3+12.78*x^2-12.04*x+4.16')

--> x=linspace(0.8,2.1,1000);

--> plot(x,f(x))

--> xgrid()
```

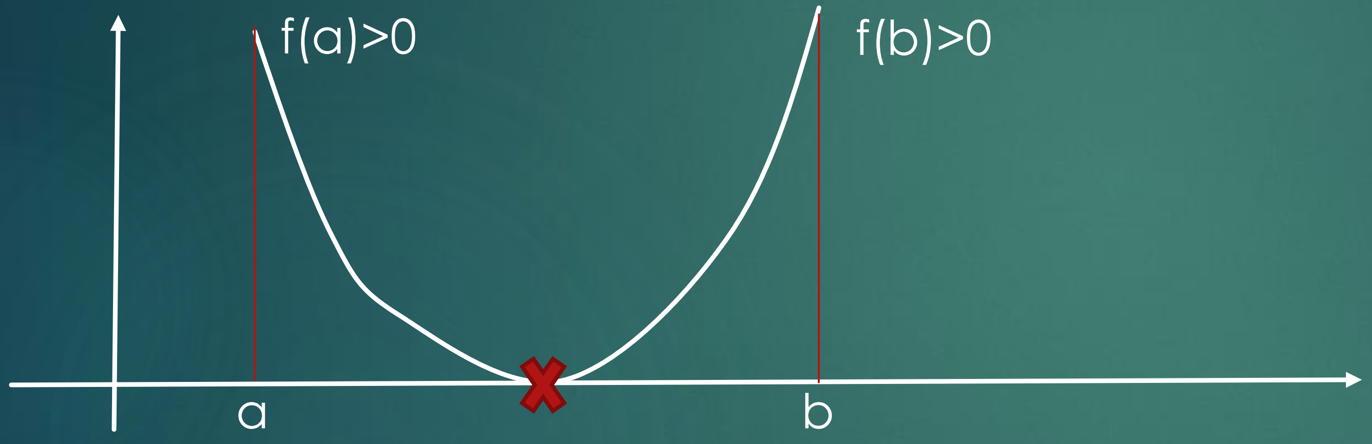



```
--> xr=Bissecao_abs(f,0.8,2.1,1e-8)
não há raiz no intervalo [0.800000,2.100000]]
Undefined variable 'x1' in function 'Bissecao_abs'.
```

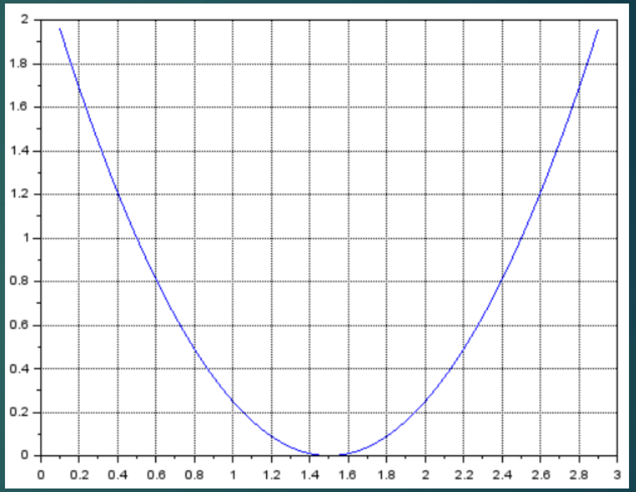
?

3) Raiz dupla

Como $f(a)f(b)>0$, o método não vai começar; e por conseguinte não vai encontrar nenhuma raiz.




```
--> deff ('y=f(x)' , 'y=x^2-3*x+2.25')

--> x=linspace(0.1,2.9,1000);

--> plot(x,f(x))

--> xgrid()
```




```
--> xr=Bissecao_abs(f,0.1,2.9,1e-8)
não há raiz no intervalo [0.100000,2.900000]]
Undefined variable 'x1' in function 'Bissecao_abs'.
```

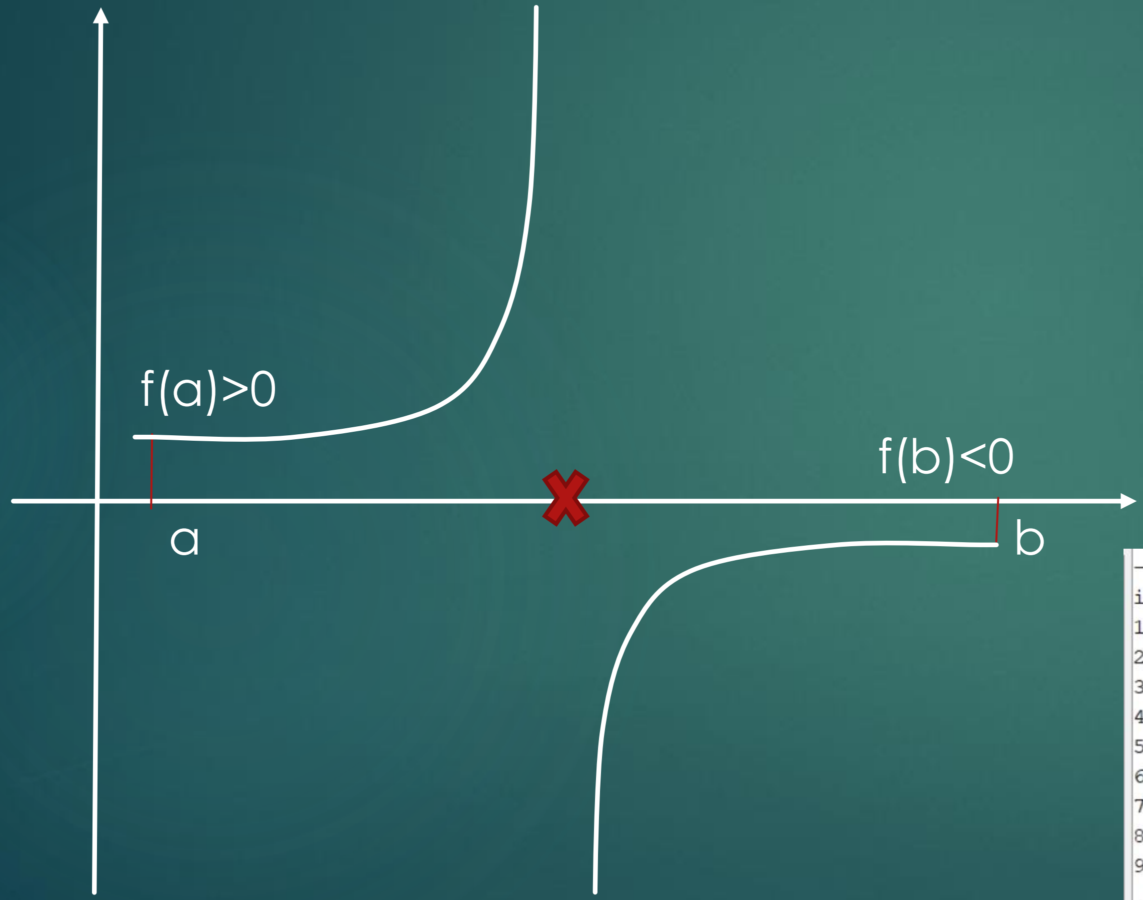
?

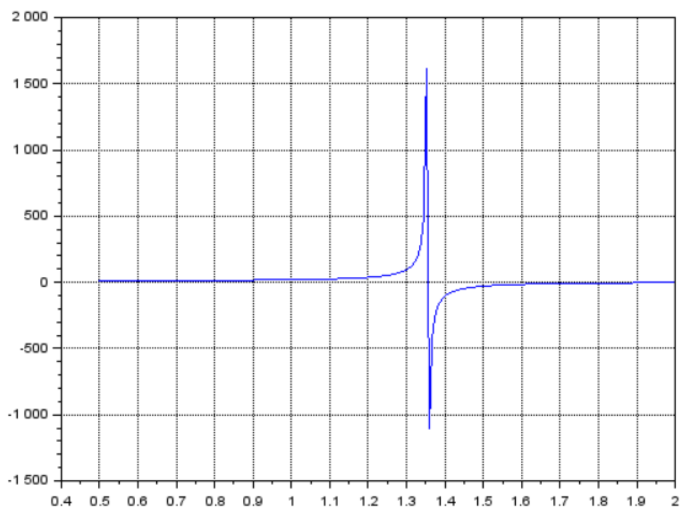
4) Singularidade

Como f(a)f(b)<0, o método vai convergir para a singularidade (a função diverge para $\pm\infty$!), e não para uma raiz.

$$f(x) = 4.9 + \frac{5.1x-22.3}{3.1x-4.2}$$ no intervalo [0.5 , 2.0].

```
--> deff ('y=f(x)' , 'y=4.9+(5.1*x-22.3)./(3.1*x-4.2)')

--> x=linspace(0.5,2.0,200);

--> plot(x,f(x))

--> xgrid()
```






```
--> xr=Bissecao_abs(f,0.5,2.0,1e-2,%t)
```

i	a	x1	b	erro
1	0.5000000000(1)	1.2500000000(1)	2.0000000000(-1)	1.5e+00
2	1.2500000000(1)	1.6250000000(-1)	2.0000000000(-1)	7.5e-01
3	1.2500000000(1)	1.4375000000(-1)	1.6250000000(-1)	3.8e-01
4	1.2500000000(1)	1.3437500000(1)	1.4375000000(-1)	1.9e-01
5	1.3437500000(1)	1.3906250000(-1)	1.4375000000(-1)	9.4e-02
6	1.3437500000(1)	1.3671875000(-1)	1.3906250000(-1)	4.7e-02
7	1.3437500000(1)	1.3554687500(-1)	1.3671875000(-1)	2.3e-02
8	1.3437500000(1)	1.3496093750(1)	1.3554687500(-1)	1.2e-02
9	1.3496093750(1)	1.3525390625(1)	1.3554687500(-1)	5.9e-03

```
 xr  =

  1.3525391
```

?