Métodos de Determinação de Zeros de funções a uma dimensão

Método da Bissecção

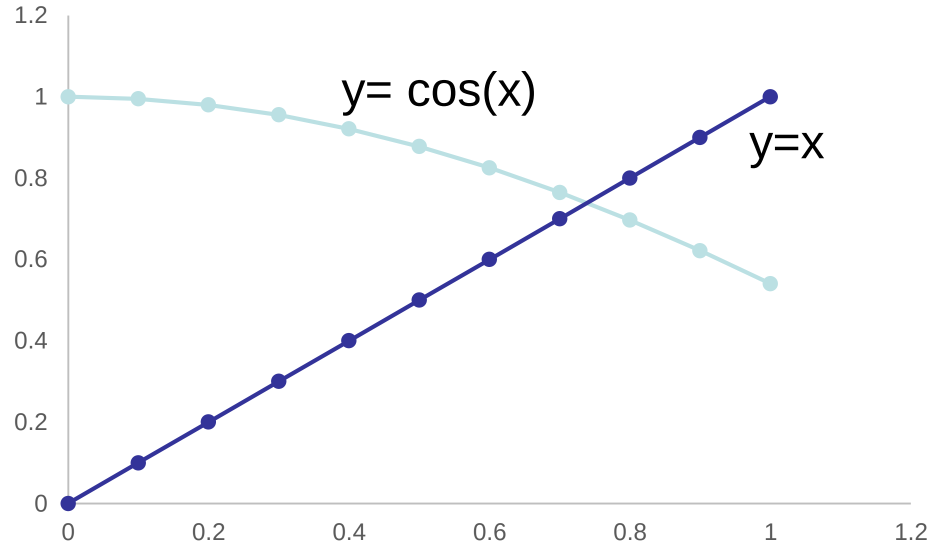
Método do Ponto Fixo

Zero de uma função

Determinar x tal que f(x)=0

Exemplo de aplicação:

cos(x)=x



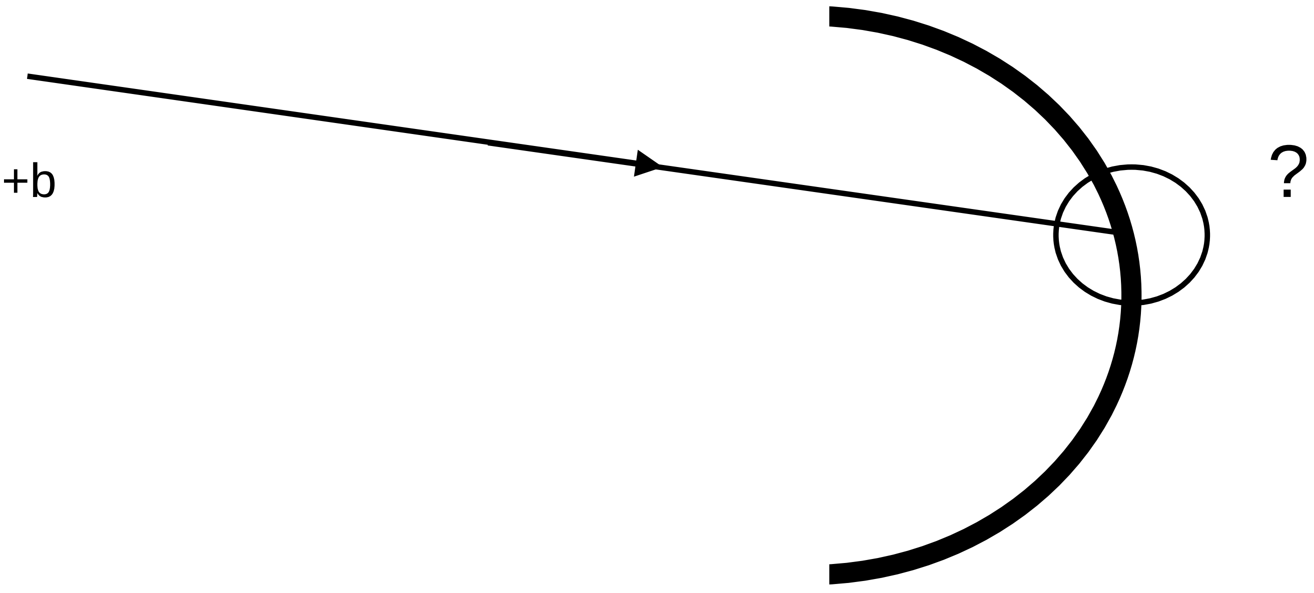
Como determinar o zero z, tal que |x*-z|<0.000001 ?

Se não souber nada sobre a sua localização, posso ter que começar, por exemplo, em x=0. Será que ir testando todos os valores separados de 0.000001 até o coseno se tornar inferior a y=x é uma boa estratégia? Precisariamos quase de 1 milhão de iterações até encontrar o zero! Queremos um método mais rápido…

Outro exemplo de aplicação

Como determinar em que ponto um raio luminoso bate num espelho esférico?

$y = m\,x + b$

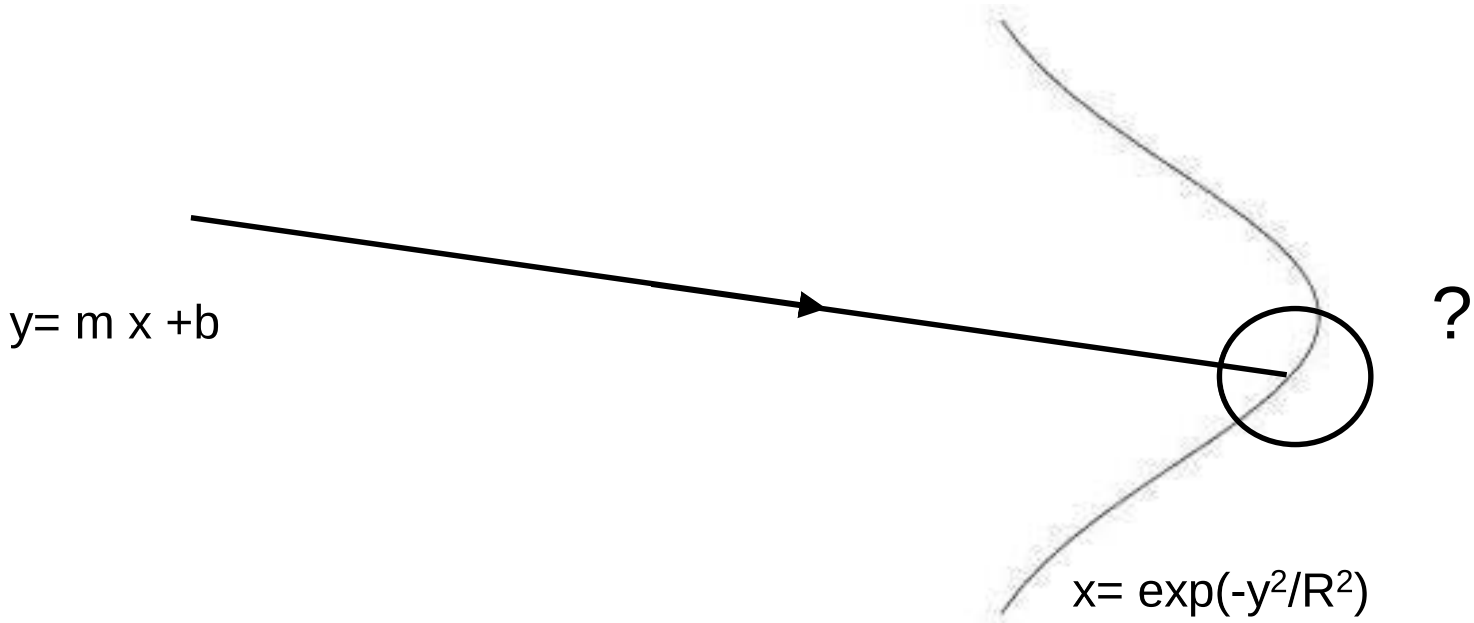


$x^2 + (m\,x + b)^2 = R^2$

$x^2 + y^2 = R^2$

Neste caso o problema tem solução exacta

Como determinar em que ponto um raio luminoso bate num espelho com forma gaussiana?

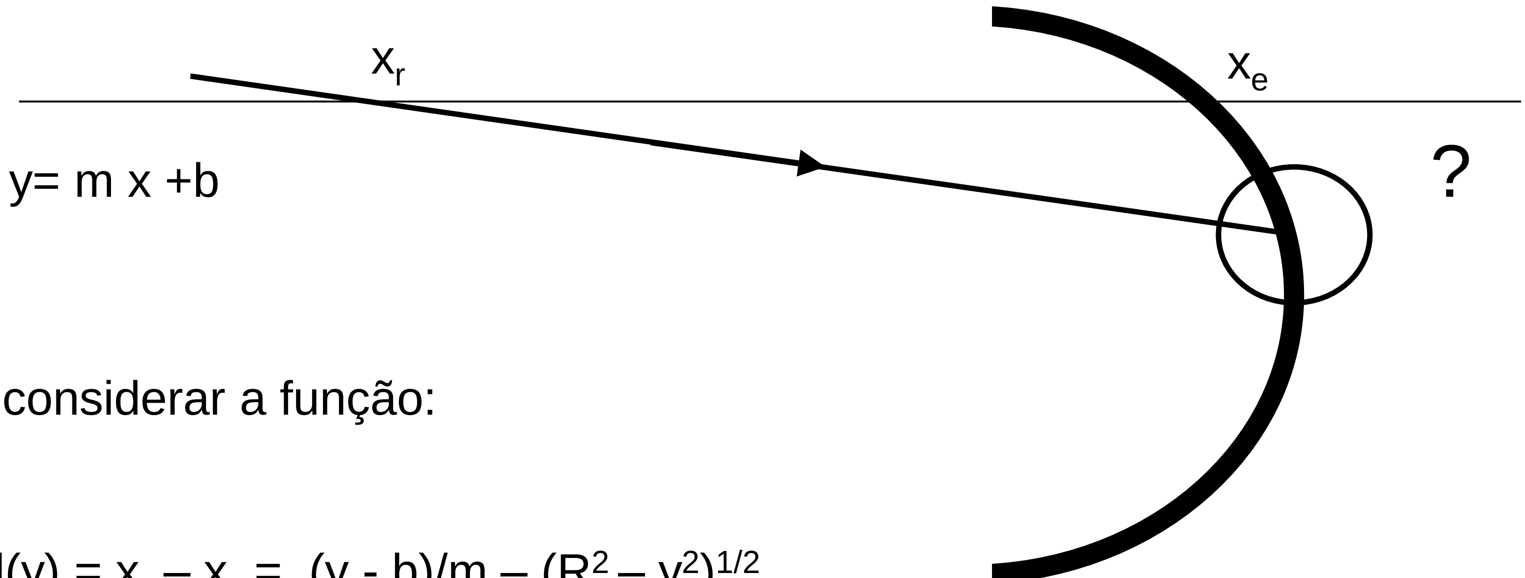


$$y = m \exp(-y^2/R^2) + b$$

Neste caso o problema não tem solução analítica.

Como determinar numéricamente?

Como determinar em que ponto um raio luminoso bate num espelho esférico?

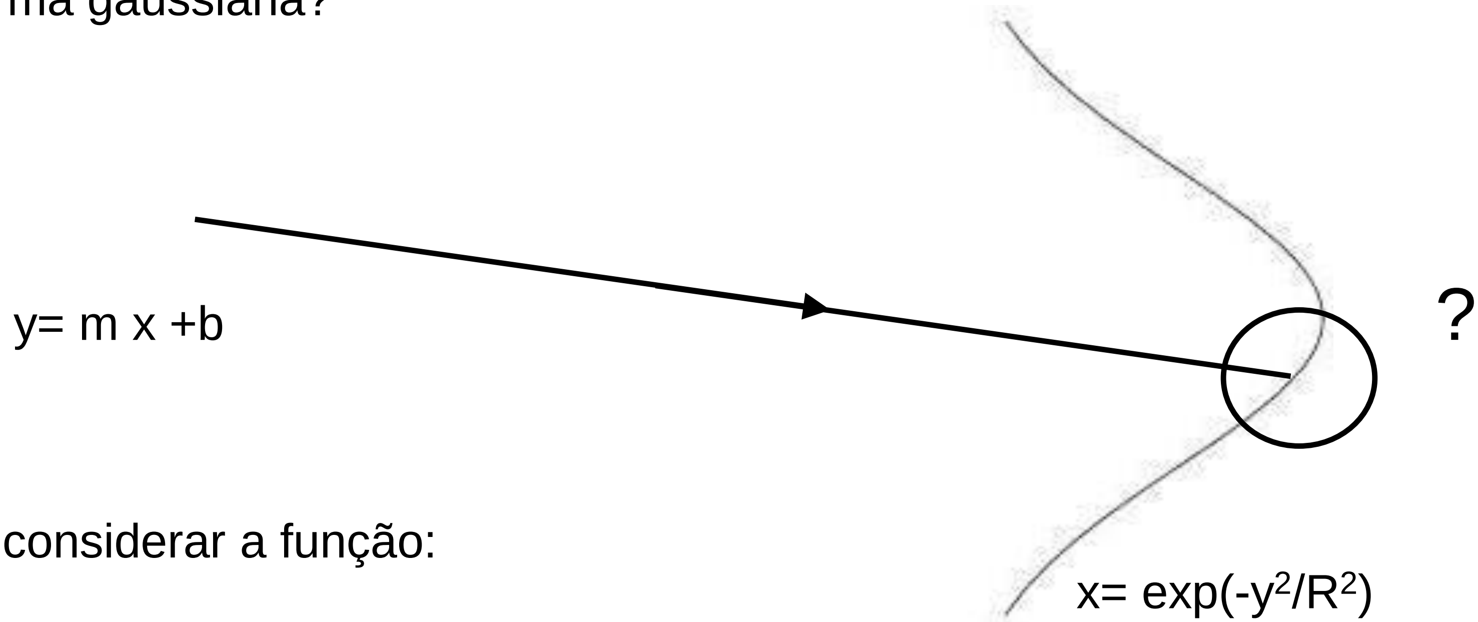


Podemos considerar a função:

$$d(y) = x_r - x_a = (y - b)/m - (R^2 - y^2)^{1/2}$$

$$x^2 + y^2 = R^2$$

Como determinar em que ponto um raio luminoso bate num espelho com forma gaussiana?



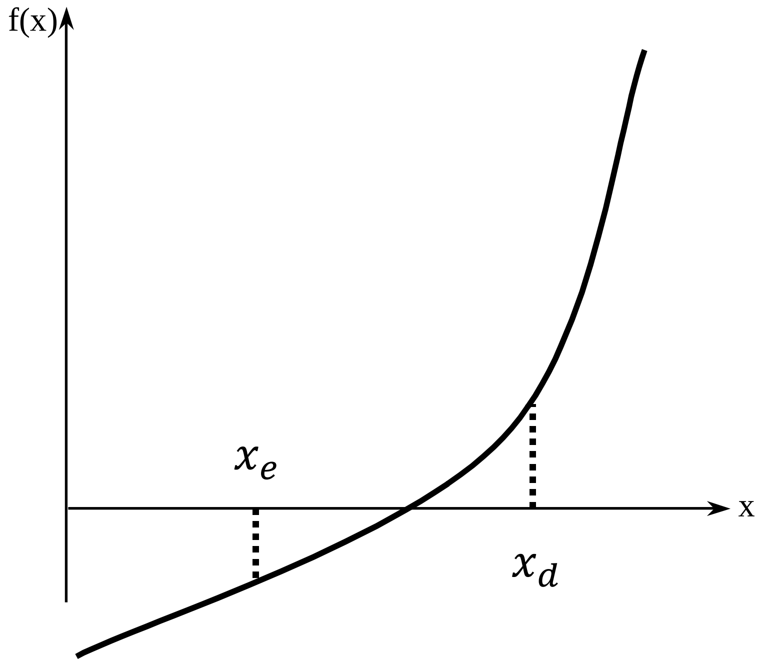
Podemos considerar a função:

$$d(y) = x_r - x_a = (y - b)/m - \exp(-y^2/R^2)$$

Como determinar o ponto de intersecção?

Método da Bisecção: ideia

Se se estimar que há um zero entre x_e e x_d, então podemos calcular o valor da função no centro do intervalo. Depois constroi-se um intervalo menor dentro do qual esteja o zero.

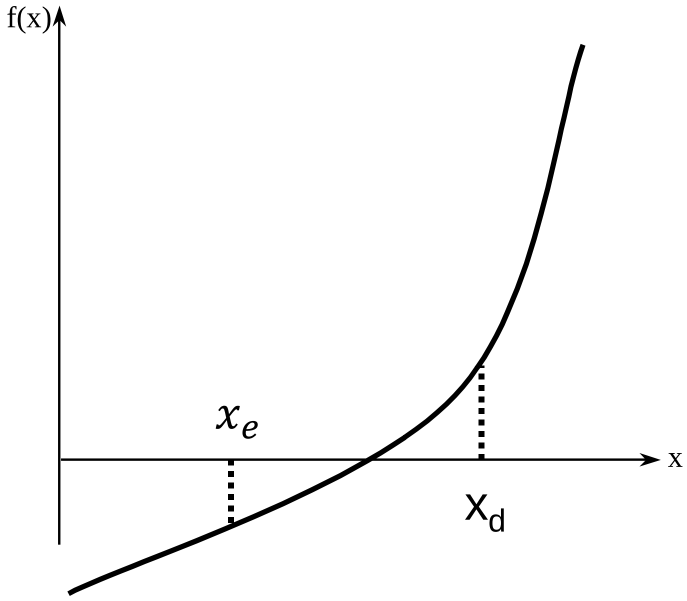


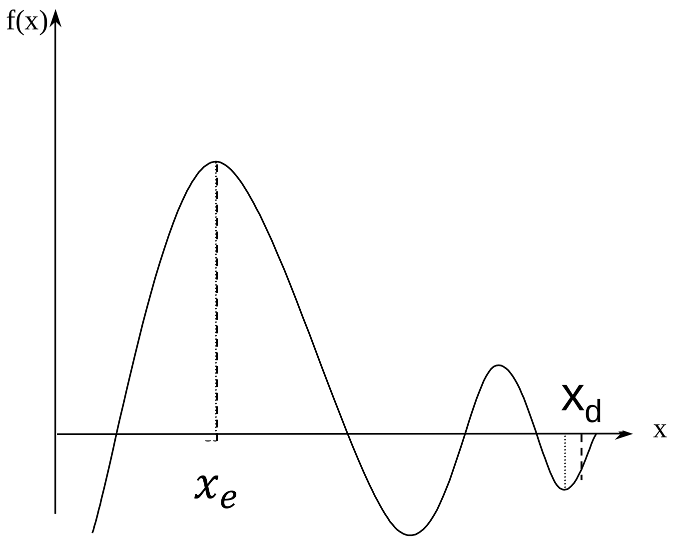
Como fazê-lo?

Considerações importantes

A equação f(x)=0, onde f(x) é uma função contínua e real, tem pelo menos uma solução entre x_e and x_d se:

$$f(x_e) * f(x_d) < 0.$$

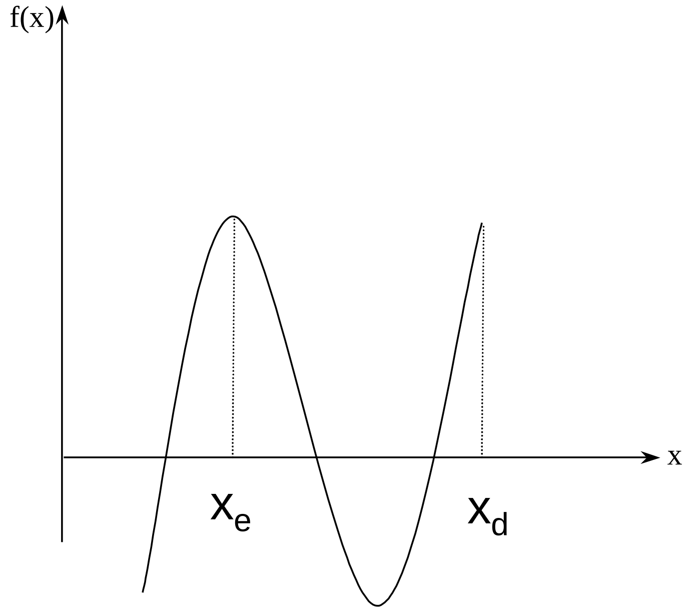




Considerações importantes

A equação f(x)=0, onde f(x) é uma função contínua e real, pode ter mais que uma solução entre x_e and x_d se:

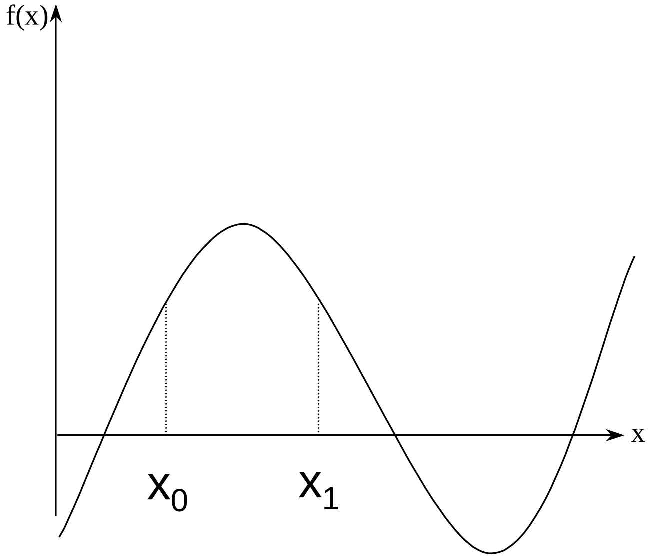
$$f(x_e) * f(x_d) > 0.$$

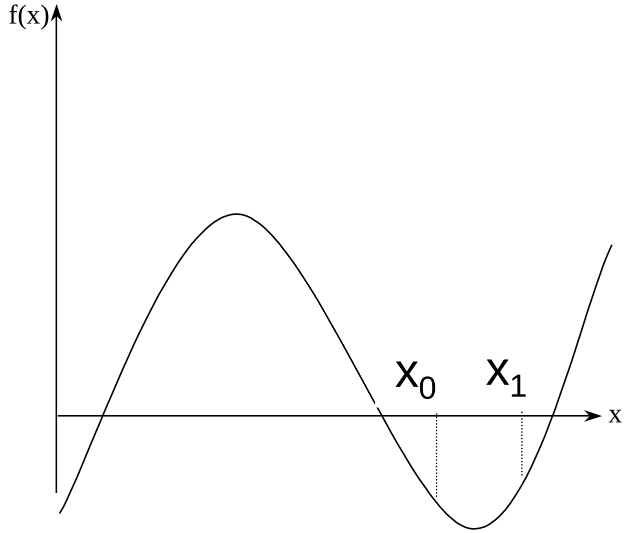


Considerações importantes

A equação f(x)=0, onde f(x) é uma função contínua e real, pode não ter solução entre x_e and x_d se:

$$f(x_e) * f(x_d) > 0.$$





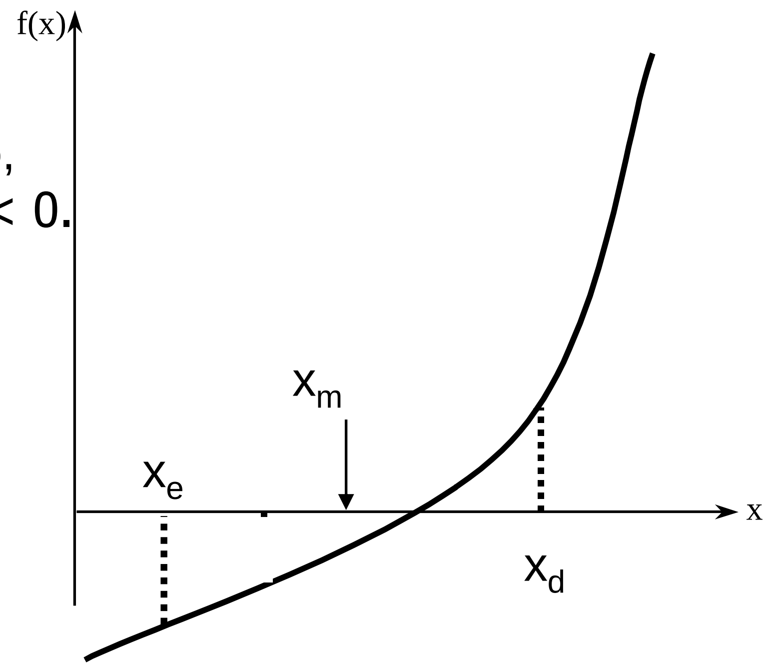
Método da bisecção

1) Escolher dois pontos para os quais $f(x_e) * f(x_d) < 0.$
Isto é, a função muda de sinal entre x_e e x_d

2) Determinar o ponto médio do intervalo: $x_m = (x_e + x_d)/2$

3) Considerar um novo intervalo entre x_m e x_e ou x_d, de forma a que o zero esteja dentro do novo intervalo, i.e., $f(x_m) * f(x_e) < 0$ ou $f(x_d) * f(x_m) < 0.$

4) Parar se a estimativa do zero satisfizer um critério de paragem.



Critério de Paragem

Podia parar o algoritmo se $f(x_n)$ fosse inferior a um determinado número?

Mas seria: “Parar quando $f(x_n)<0.00001$” um bom critério?

E se na iteração inicial a função fosse 0.0009???

Não é um bom critério!

(pode depender das unidades de f)

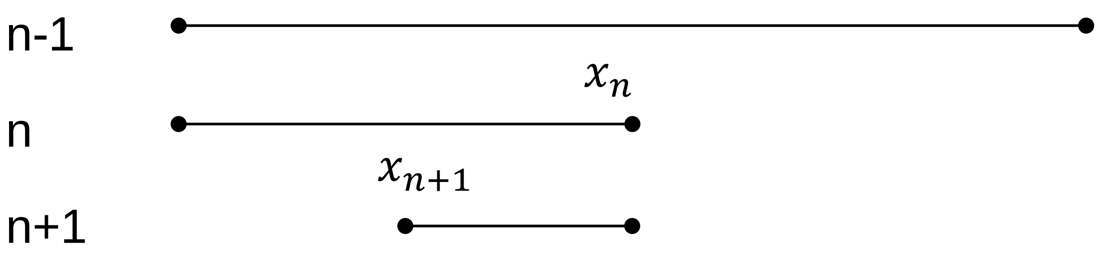
Alternativamente poderemos pensar num critério baseado na aproximação à solução.

Estimativa do erro:

Se o intervalo na iteração n, for constituída pelos pontos $x_{e,n}$ e $x_{d,n}$, respetivamente à esquerda e à direita, o erro pode ser majorado:

$$erro_n \leq x_{d,n} - x_{e,n}$$

Na iteração seguinte, ou o $x_{e,n}$ ou o $x_{d,n}$ serão os novos extremos do intervalo, e o ponto médio, o outro.



Se considerarmos que o ponto médio do intervalo é estimativa do zero na iteração seguinte, então podemos escrever que:

$$|x_{n+1} - x_n| \geq erro_{n+1}$$

Critérios de Paragem

relacionado com a estimativa indireta do Erro Relativo

O critério de paragem pode utilizar o "erro relativo" na estimativa da raiz da equação:

$$|\delta| = \left| \frac{x_{n+1} - x_n}{x_{n+1}} \right| < \delta_p$$

Exemplo: $\delta_p = 0.01\%$

Em cima escrevi erro relativo com aspas, porque $\left| x_{n+1} - x_n \right|$ não é o erro

(pois desconhecemos a solução), mas pode-se relacionar com uma sua majoração. Alternativamente podemos usar um critério relacionado com o erro absoluto:

$$|x_{n+1} - x_n| < \epsilon$$

Última consideração: velocidade de convergência

O intervalo é divido ao meio a cada iteração. Por isso as majorações do erro cometido a cada iteração numa estimativa, decrescem também por 1/2.

$$erro_n = (1/2)\ erro_{n-1}$$

Assim, o erro na estimativa na iteração n, $erro_n$ evolui com:

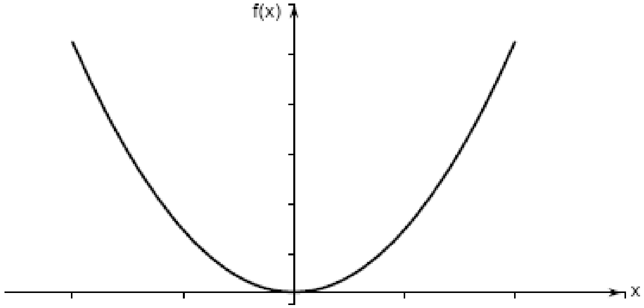
$$erro_n = (1/2)^n\ erro_0$$

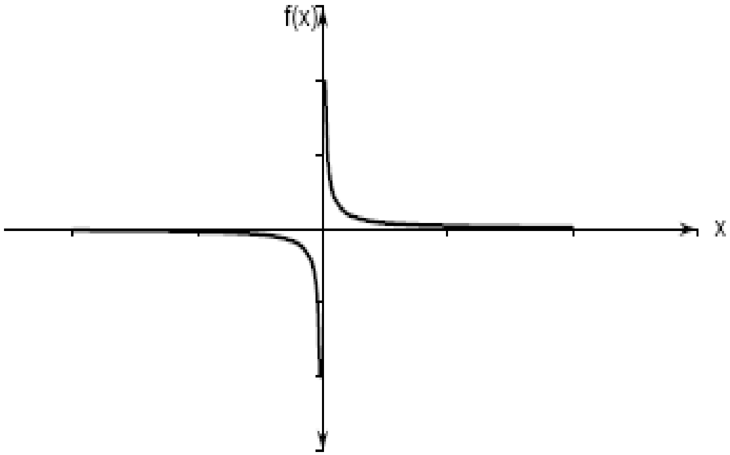
onde e_0 é o erro no ínicio (igual ao comprimento do intervalo).

O método tem convergência linear pois o expoente p=1 em:

$$erro_n = C\ (\ erro_{n-1}\ )^p \qquad \text{com C constante}$$

Problemas do Método



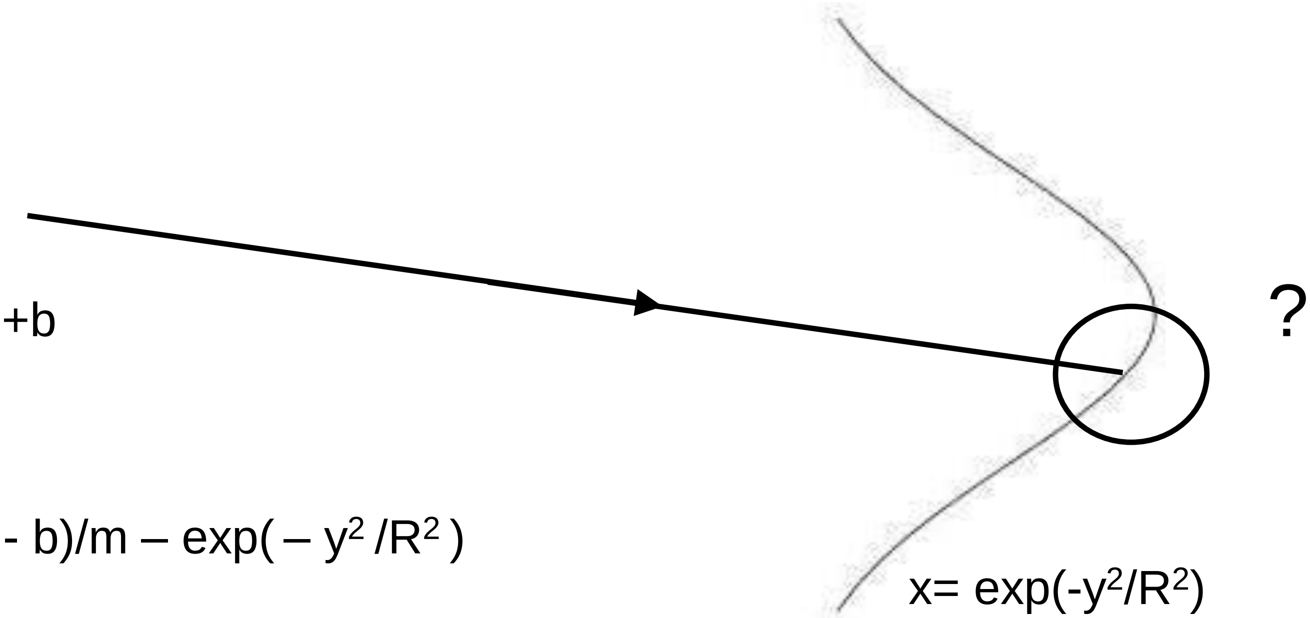


Como resolver: com a derivada...

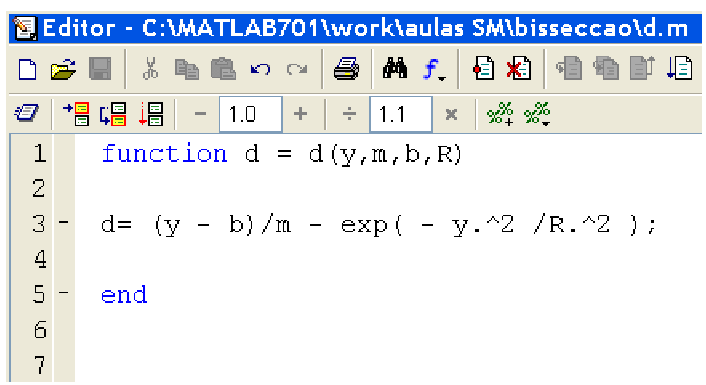
Como resolver: com cuidado, pois não há zero. O programa não deve ser aplicado no caso de existirem divergências no intervalo inicial

Exemplo anterior: como fazer?

$y = m\,x + b$



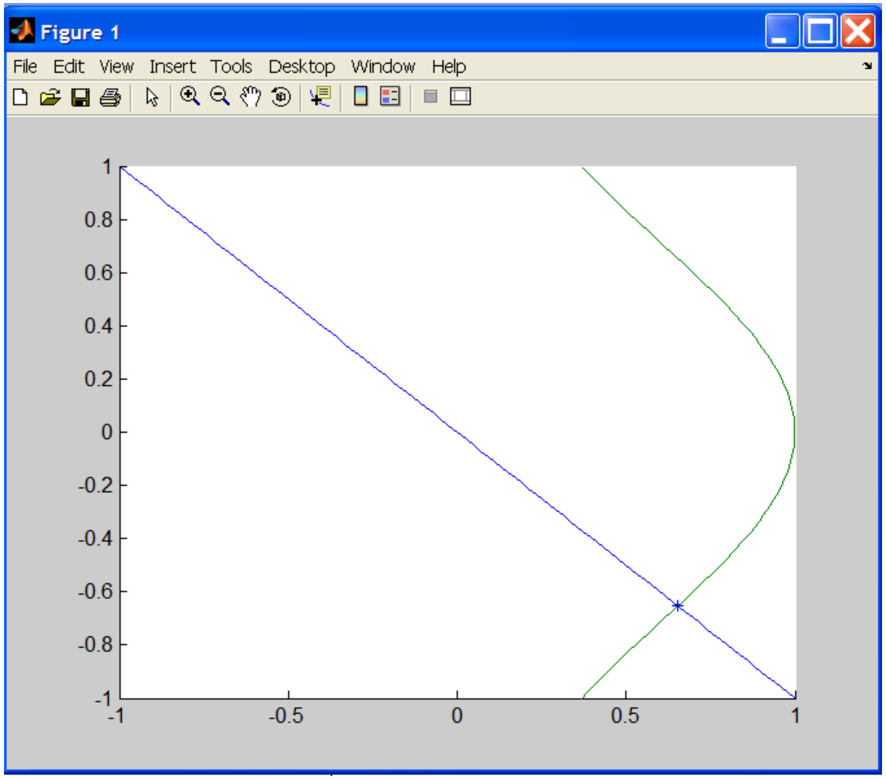
$$d(y) = x_r - x_a = (y - b)/m - \exp(-y^2/R^2)$$



```
y0=-1;
y1=1;
m= -1;
R=1;
b=0;
deltac=0.0001;
ya=y0;
yb=y1;

d1= d(ya,m,b,R);
d2= d(yb,m,b,R);
if(d1*d2>0)
    display('Escolha outros pontos iniciais');
    return;
end

delta=1;
```

```
while delta>deltac

    y= (ya+yb)/2;


    if(d(ya,m,b,R)*d(y,m,b,R)<0)
        delta= abs(yb-y)/(abs(y));
        yb=y;
    else
        delta= abs(ya-y)/(abs(y));
        ya=y;
    end

end
ys=y;
xs= (ys-b)/m;


sprintf('(x,y)= (%g,%g)',xs,ys)
```

```
y= linspace(-1,1,100);
xr= (y-b)./m;
xg= exp( - y.^2 /R.^2 );
hold on
plot(xr,y,xg,y);
plot(xs,ys,'+');
```



No exemplo anterior os pontos iniciais eram parâmetros a definir. Podemos reduzir a dependência do algoritmo numa boa escolha destes parâmetros.
Um algoritmo para a implementação do método da bissecção deve iniciar-se procurando dois pontos a partir dos quais o algoritmo encontrará o zero. Deverá ser um algoritmo de procura grosseira:

```
xi=-100;  % imaginemos que queremos começar em x=-100!
Dx=1;
xe=procura_intervalo(xi,Dx);
%display(xi)
xd=xe+Dx;
while xd-xe>0.000001
    if (f((xd+xe)/2)*f(xe)>0)
        xe=(xd+xe)/2;
    else
        xd=(xd+xe)/2;
    end
end
sprintf('zero= %g',(xe+xd)/2)
```

```
function x=procura_intervalo(x,incremento)
x0=x;
fx0=f(x);
x=x+incremento;
while(f(x)*f(x0)>0)
    x=x+incremento;
end
x=x-incremento;
```

```
function y=f(x)
y=cos(x)-x;
```

Note que deve colocar em funções pedaços do código que sabe que funcionam bem e que são independentes.

Método do Ponto Fixo

$f(x)=0$ é equivalente a: $x = x - \lambda f(x)$

A ideia está em criar a equação de recorrência:

$$x_{n+1} = x_n - \lambda_n f(x_n)$$

Se esta equação tender para o ponto fixo (isto é, não divergir), então ao fim muitas iterações teremos determinado um valor de x muito perto deste ponto fixo. Este valor de x, verifica:

$x^* = x^* - \lambda f(x^*)$ ou seja resolve: $f(x^*)=0$

Convergência

Como vimos na aula anterior, para ter convergência para o ponto fixo, este deve ser estável. Para tal podemos utilizar a técnica de linearização à volta do ponto fixo:

$$x_{n+1} = x_n - \lambda_n f(x_n)$$

$$x_{n+1} = \underbrace{x^* - \lambda_n f\left(x^*\right)}_{x^*} + \left(1 - \lambda_n \left.\frac{df}{dx}\right|_{x=x^*}\right)(x_n - x^*)$$

o sinal de λ é determinante

$$x_{n+1} - x^* = \underbrace{\left(1 - \lambda_n \left.\frac{df}{dx}\right|_{x=x^*}\right)}_{K}(x_n - x^*)$$

$$K \longrightarrow \left|1 - \lambda_n \left.\frac{df}{dx}\right|_{x=x^*}\right| < 1 \Leftrightarrow \boxed{0 < \lambda_n \left.\frac{df}{dx}\right|_{x=x^*} < 2}$$

Escolha do λ

Deve ter o sinal correcto. Porém isso pode não garantir a convergência. De facto, a análise anterior é válida só quando estivermos suficientemente perto do ponto fixo e implicaria conhecer a derivada de f.

Determinar a dimensão do intervalo de convergência nem sempre é possível e pode ser complicado, pelo que o método aplica-se de forma empírica.

Aplicação do método do ponto fixo no exemplo anterior

O que muda?

```
lambda=0.1;
d1= d(ya,m,b,R);
y = ya - lambda .*d1;
d2= d(y,m,b,R);

if abs(d1)<abs(d2)     % neste caso estará a divergir
    lambda= -lambda;
end

delta=1;
while delta>deltac
    d1= d(y,m,b,R);
    y = y - lambda .*d1;
    delta= abs(lambda .*d1)/(abs(y));
end
```

Exemplo em excel

1			lambda=	0.3
2				
3	0.1			
4	0.368501			
5	0.537811			
6	0.634118			
7	0.685561			
8	0.712112			
9	0.725573			
10	0.732337			
11	0.73572			
12	0.737408			
13	0.73825			
14	0.738669			
15	0.738878			
16	0.738982			
17	0.739034			
18	0.73906			
19	0.739072			
20	0.739079			
21	0.739082			
22	0.739084			
23	0.739084