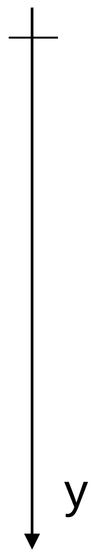
Métodos de Determinação de Zeros de funções a uma dimensão e mais dimensões

Método de Newton

Problema

Lança-se um corpo em queda. Um atirador terá de o alvejar.
Qual o ângulo com que deverá disparar?

$$m\frac{d^2 y_c}{dt^2} = -\upsilon\frac{dy_c}{dt} + mg \longrightarrow \left\{ \begin{aligned} & m\frac{dv_c}{dt} = -\upsilon\, v_c + mg \\ & \frac{dy_c}{dt} = v_c \end{aligned} \right.$$



A primeira equação tem uma solução exacta simples: $v_c = \dfrac{mg}{\upsilon} - u$

$$-m\frac{du}{dt} = -\upsilon\left(\frac{mg}{\upsilon} - u\right) + mg \Leftrightarrow m\frac{du}{dt} = -\upsilon\, u$$

Donde:

$$u(t) = A\exp(-\frac{\upsilon}{m}t) \qquad v_c(t) = \frac{mg}{\upsilon} - A\exp(-\frac{\upsilon}{m}t)$$

Se: $v_c(t) = v_0$ então: $v_c(0) = v_0 = \frac{mg}{\upsilon} - A \Leftrightarrow A = \frac{mg}{\upsilon} - v_0$

Assim: $$v_c(t) = \frac{mg}{\upsilon}\left(1 - \exp(-\frac{\upsilon}{m}t)\right) + v_0 \exp(-\frac{\upsilon}{m}t)$$

E de: $\frac{dy_c}{dt} = v_c$ vem por integração:

$$y_c(t) = \frac{mg}{\upsilon}\left(t + \frac{m}{\upsilon}\exp(-\frac{\upsilon}{m}t)\right) - \frac{mv_0}{\upsilon}\exp(-\frac{\upsilon}{m}t) + C$$

Finalmente, se em t=0, y=0, temos:

$$C = \frac{mv_0}{\upsilon} - \frac{m^2 g}{\upsilon^2}$$

donde:

$$y_c(t) = \frac{mg}{\upsilon}t + \frac{m^2 g}{\upsilon^2}\exp(-\frac{\upsilon}{m}t) - \frac{mv_0}{\upsilon}\exp(-\frac{\upsilon}{m}t) + \frac{mv_0}{\upsilon} - \frac{m^2 g}{\upsilon^2}$$

$$y_c(t) = \frac{mg}{\upsilon}t - \frac{m^2 g}{\upsilon^2}\left(1 - \exp(-\frac{\upsilon}{m}t)\right) + \frac{mv_0}{\upsilon}\left(1 - \exp(-\frac{\upsilon}{m}t)\right)$$

$$\boxed{y_c(t) = \frac{mg}{\upsilon}t + \left(\frac{mv_0}{\upsilon} - \frac{m^2 g}{\upsilon^2}\right)\left(1 - \exp(-\frac{\upsilon}{m}t)\right)}$$

Questão para os alunos mais interessados:

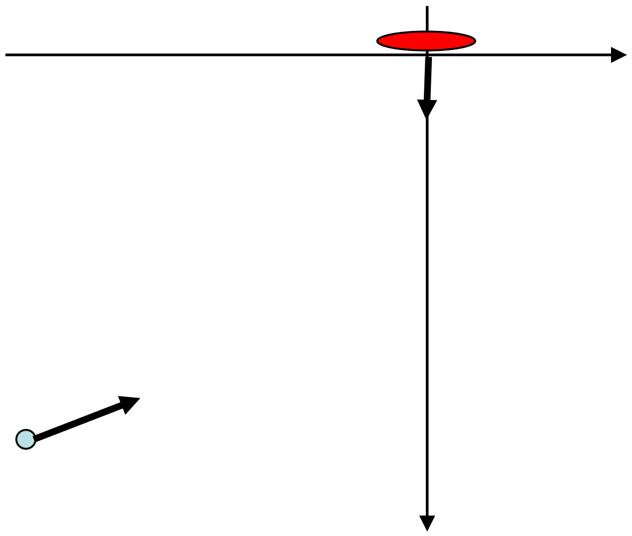
Como recuperar a solução sem viscosidade, com esta solução????

Hint: Usar a decomposição em Taylor, admitindo
que a viscosidade é um parâmtero pequeno

(Experimentem!!!!)

Equação do projétil

Para simplificar vamos assumir que, ao contrário do corpo em queda, o projéctil comporta-se muito bem de um ponto de vista aerodinâmico. Por isso, vamos desprezar o efeito do atrito. Assim:



$$x_p(t) = x_0 + v_{0x} t$$

$$y_p(t) = y_0 + v_{0y} t + \frac{1}{2} g t^2$$

$$x_c(t) = 0$$

$$y_c(t) = \frac{mg}{\upsilon} t + \left(\frac{m v_0}{\upsilon} - \frac{m^2 g}{\upsilon^2} \right) \left(1 - \exp(-\frac{\upsilon}{m} t) \right)$$

Situação do problema:

Temos uma espingarda, que dispara tiros com uma velocidade estabelecida.
O que podemos “experimentalmente” alterar é o ângulo de disparo.
Tudo o resto são parâmetros do problema que não poderemos controlar.

$$v_{0x} = v_b \cos\theta$$
$$v_{0y} = -v_b \sin\theta$$

onde v_b é a velocidade da bala (constante)

Podemos resolver este problema de duas formas:

$$0 = x_0 + v_b \cos\theta\, t$$

$$y_0 - v_b \sin\theta\ t + \frac{1}{2} g t^2 = \frac{mg}{\upsilon} t + \left(\frac{m v_0}{\upsilon} - \frac{m^2 g}{\upsilon^2} \right) \left(1 - \exp(-\frac{\upsilon}{m} t) \right)$$

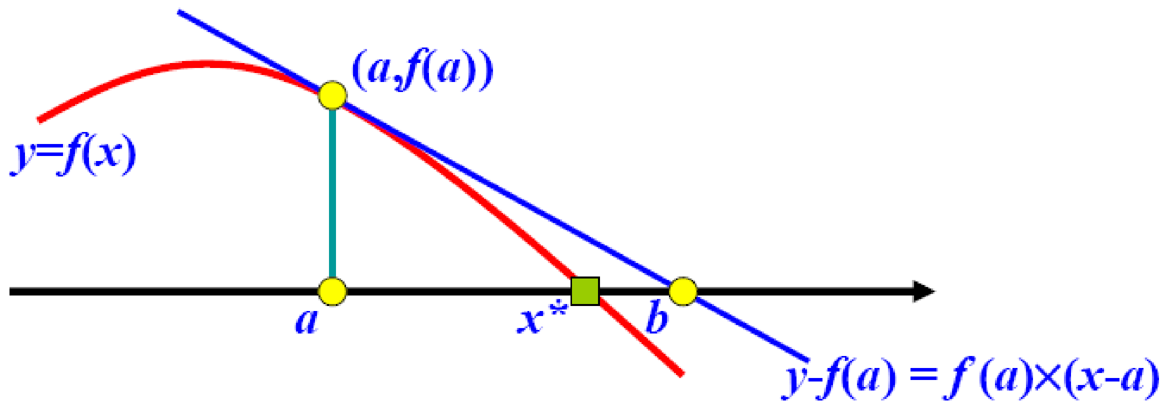
Poderiamos obter da primeira equação uma expressão para t, e introduzi-la na segunda. Teriamos uma equação não linear cuja única incógnita é θ.

Alternativamente poderiamos pensar que temos um sistema com duas equações e duas incógnitas. Poderiamos então pensar em resolver o sistema de equações:

$$F\left(\vec{v}\right) = 0$$

Método de Newton

Intuitivamente tem-se:



Queremos encontrar a raiz x* da equação f(x)=0.

Poderíamos pensar em determinar a tangente à função f, no ponto x=a (onde a é uma solução aproximada de x*), que passa por (x,f(x))=(a,f(a)), e tem o declive de f nesse ponto. Encontrando onde esta recta cruza o eixo das abcissas, temos uma nova estimativa de x*.

Formalização desta ideia

$$f(x) = f(x_0) + \left.\frac{df}{dx}\right|_{x=x_0} (x - x_0) + \ldots$$

$$0 \qquad x_k \qquad x_{k+1}$$

grande assumpção!

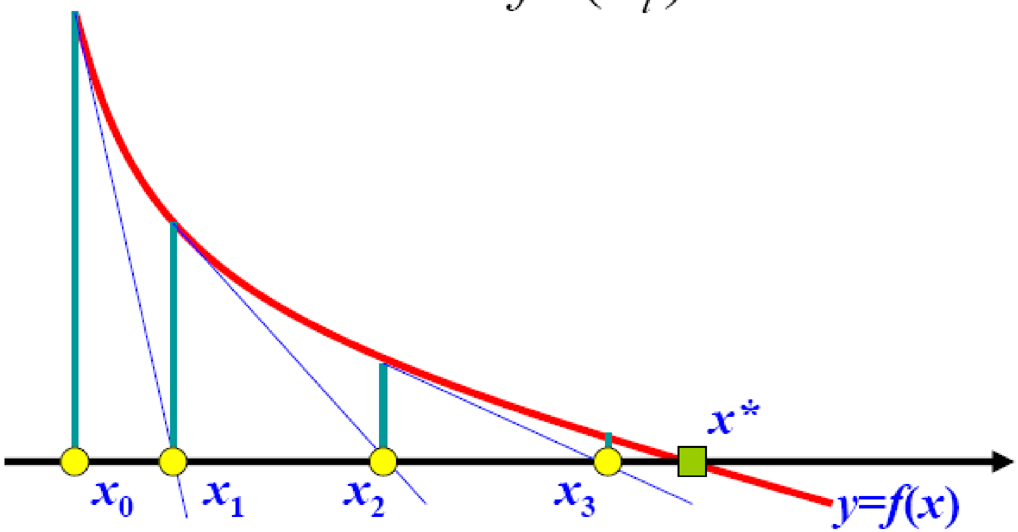
$$0 = f(x_k) + \left.\frac{df}{dx}\right|_{x=x_k} (x_{k+1} - x_k)$$

$$\boxed{x_{k+1} = x_k - \frac{f(x_k)}{\left.\frac{df}{dx}\right|_{x=x_k}}}$$

Método de Newton-Raphson

ideia:

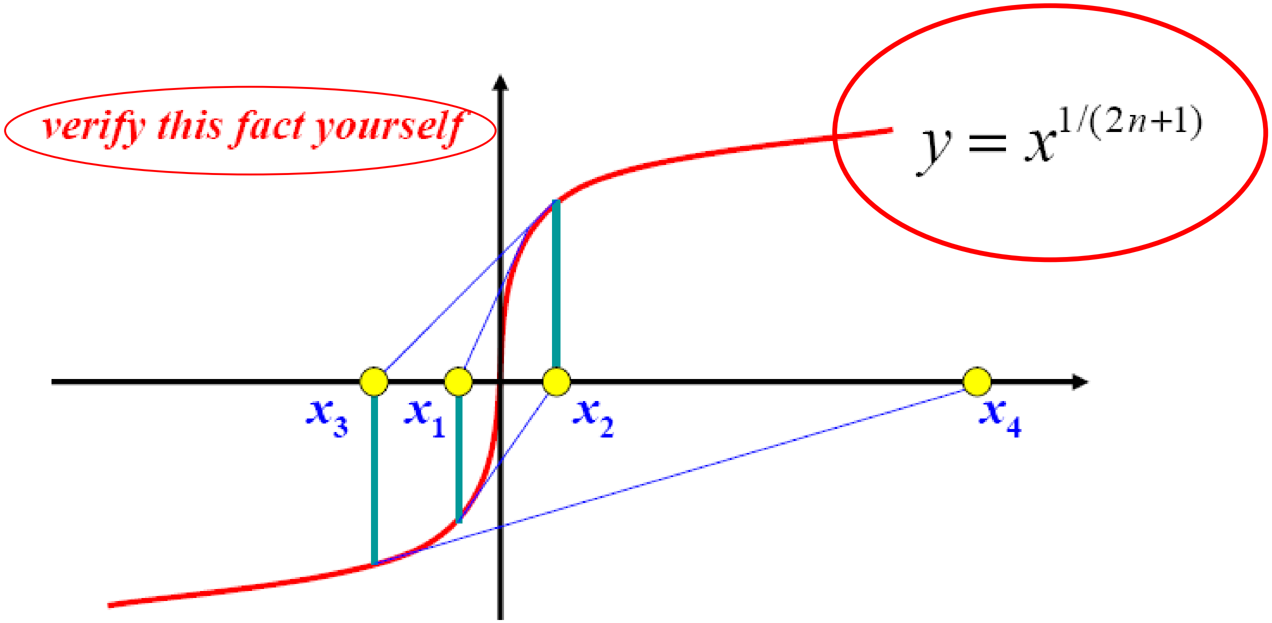
$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$



Temos de conhecer a derivada!

Na realidade podemos também estimá-la aproximadamente (veremos mais à frente com fazê-lo)

Problemas



$$x_{n+1} = x_n - (2n+1)\frac{x_n^{1/(2n+1)}}{x_n^{1/(2n+1)-1}} = x_n - (2n+1)x_n = 2\ n\ x_n$$

Em cima, a função não é real para valores negativos de x, e n<1. Ora, o método aparentemente funcionaria na mesma, e atiraria para valores de x para os quais a função não está definida.

O caso precedente mostrou que o método pode não convergir, independentemente de estarmos perto ou longe da solução (nota: não é também por acaso que o exemplo precedente utilizou uma função não analítica...)

Há outra situações em que, mesmo com funções analíticas, o método pode não convergir, ficando a oscilar entre dois valores:

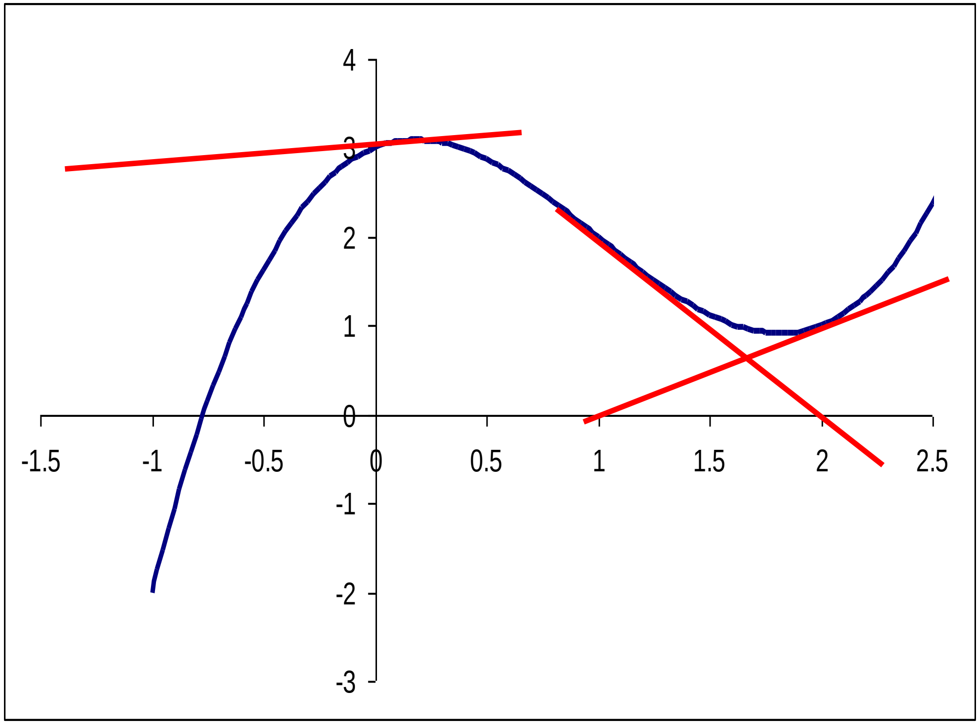
$$f(x) = x^3 - 3x^2 + x + 3 \qquad \frac{df(x)}{dx} = 3x^2 - 6x + 1$$

$$x_{n+1} = x_n - \frac{x_n^3 - 3x_n^2 + x_n + 3}{3x_n^2 - 6x_n + 1}$$

Se começarmos em x_0=1, temos: x_1=2, depois x_2=1, etc.

No entanto, se começarmos noutro ponto (por exemplo, x_0=-1), conseguimos ter uma convergência muito rápida.

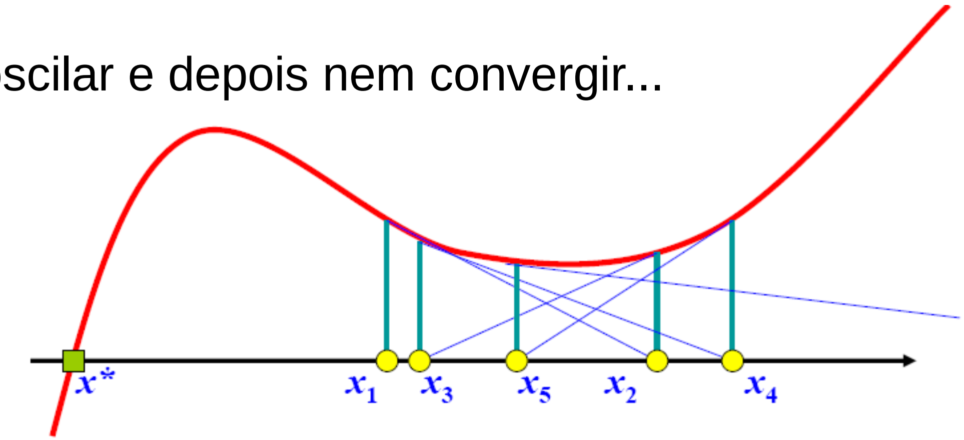
1	x	f(x)
2	0.2	3.088
3	38.8	53936.552
4	26.21129648	15976.11621
5	17.82412832	4730.443026
6	12.24020415	1399.631593
7	8.52791512	413.5474012
8	6.06644825	121.9172462
9	4.441030033	35.86210189
10	3.371224102	10.59024243
11	2.658945167	3.247691115
12	2.139837617	1.201235758
13	1.506838471	1.116522378
14	2.415064413	2.003406475
15	1.915115437	0.936119978
16	0.087858134	3.06537916
17	-6.09223774	-340.553921
18	-3.80509842	-99.3345803
19	-2.32837566	-28.2152757
20	-1.42503166	-7.41101087
21	-0.95125262	-1.52666816
22	-0.78922309	-0.14942817
23	-0.76957172	-0.00206528
24	-0.76929241	-4.1414E-07
25	-0.76929235	-1.7319E-14
26	-0.76929235	0



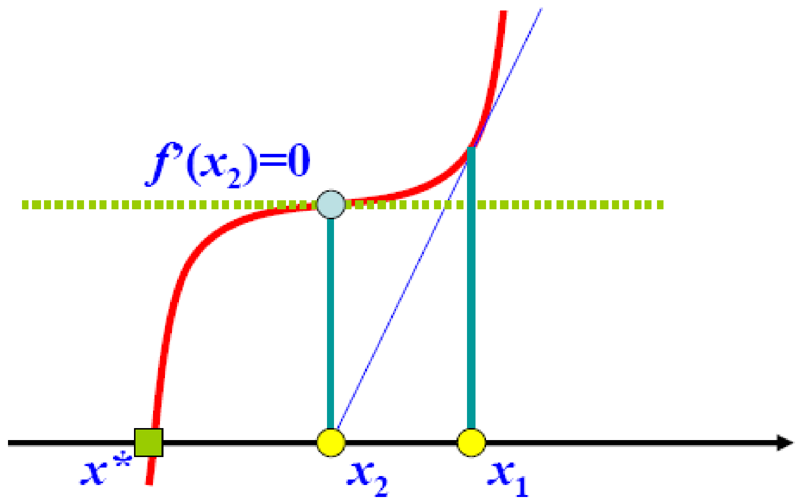
Assim, as soluções que vamos obter dependem da escolha da condição inicial do método!

Mais problemas

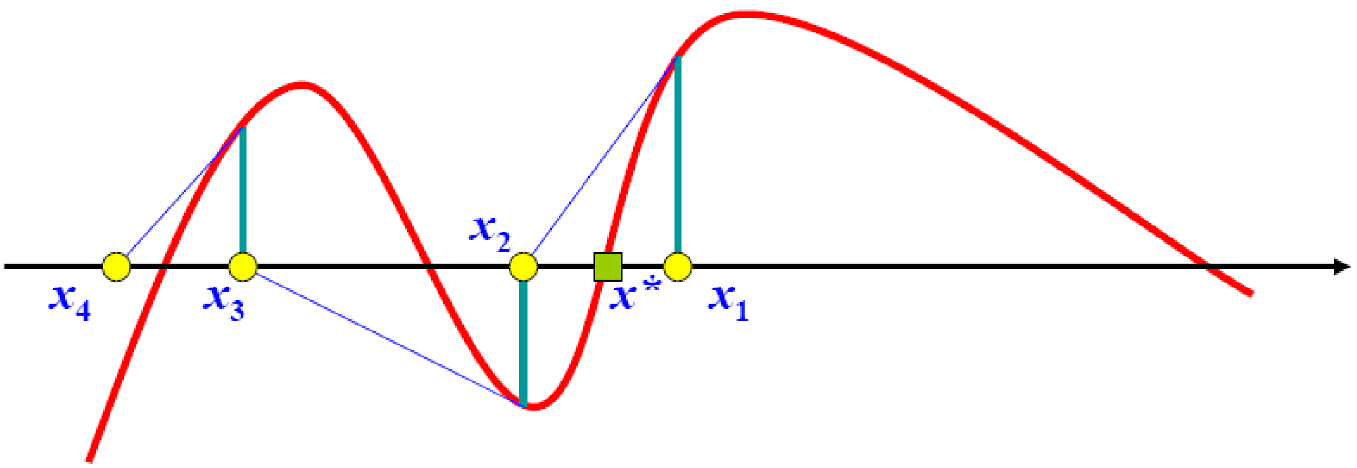
O método pode oscilar e depois nem convergir...



Os pontos de inflexão podem exigir uma boa escolha da iteração inicial, pois o método poderia em princípio convergir, mas os pontos de inflexão não permitirem a convergência, ou torná-la muito lenta!



E o método não garante uma convergência para a raiz pretendida mesmo que a escolha da iteração inicial tenha sido boa!



exemplo de aplicação

Como calcular uma raiz quadrada com um computador?

Método de Newton: $x = \sqrt{a}$?

$$f(x) = x - \sqrt{a} \qquad\qquad f(x) = x^2 - a$$

$$x_{n+1} = x_n - \frac{x_n - \sqrt{a}}{1} \qquad\qquad x_{n+1} = x_n - \frac{{x_n}^2 - a}{2x_n} = \frac{x_n}{2} + \frac{a}{2x_n}$$

Método de Newton Generalizado

$$\begin{cases} f(x,y,z,\ldots)=0 \\ g(x,y,z,\ldots)=0 \\ h(x,y,z,\ldots)=0 \\ \ldots \end{cases}$$

$$\vec{F}(\vec{r}) = \vec{F}(\vec{r}_0) + J_{\vec{F}}(\vec{r}_0)\cdot(\vec{r}-\vec{r}_0) + \ldots$$

Matriz Jacobiana

$$J_{\vec{F}}(\vec{r}_0) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \ldots \\ \frac{\partial F_2}{\partial x_1} & \ldots & \ldots \\ \ldots & \ldots & \ldots \end{bmatrix}$$

Generalização do método de Newton

$$0 = \vec{F}(\vec{r}_0) + J_{\vec{F}}(\vec{r}_0)\cdot(\vec{r} - \vec{r}_0) + \ldots$$

$$\vec{r}_{k+1} = \vec{r}_k - \left[J_{\vec{F}}(\vec{r}_k)\right]^{-1} \vec{F}(\vec{r}_k)$$

Matriz inversa da
matriz Jacobiana

Exemplo:

$$f(x,y) = \exp(-xy) + x$$

$$g(x,y) = x^3 - y$$

$$J = \begin{bmatrix} -y\exp(-xy)+1 & -x\exp(-xy) \\ 3x^2 & -1 \end{bmatrix}$$

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \begin{bmatrix} -y_k\exp(-x_k\, y_k)+1 & -x_k\exp(-x_k\, y_k) \\ 3x_k^2 & -1 \end{bmatrix}^{-1} \begin{bmatrix} \exp(-x_k\, y_k)+x_k \\ x_k^3 - y_k \end{bmatrix}$$

Como fazer:

Editor - C:\MATLAB701\work\aulas SM\Untitled.m

```
x=0.1;    % iterações iniciais
y=0.1;    % iterações iniciais
n=1000;
r=[x;y];

while n>0.00001
    J=[-y.*exp(-x.*y)+1,-x.*exp(-x.*y);3*x.^2, -1];
    f=[exp(-x.*y)+x;x.^3-y];
    d=J^(-1)*f;   % vector diferença entre iterações sucessivas
    r= r- d;
    x=r(1); y=r(2);
    n= norm(d);
end
x
y
f
```

Command Window

```
x =

   -0.7404


y =

   -0.4059


f =

  1.0e-009 *

    0.8330
   -0.1343
```

Nota suplementar:

Para evitar o cálculo da matriz inversa pode-se usar o seguinte procedimento:

Dado que: $$0 = \vec{F}\left(\vec{r}_0\right) + J_{\vec{F}}\left(\vec{r}_0\right) \cdot \left(\vec{r} - \vec{r}_0\right)$$

Resolve-se:

$$J_{\vec{F}}\left(\vec{r}_k\right) \cdot \Delta\vec{r} = -\vec{F}\left(\vec{r}_k\right)$$

Uma vez determinado o vector $\Delta\vec{r}$ calcula-se a nova estimativa da soução:

$$\vec{r}_{k+1} = \vec{r}_k + \Delta\vec{r}$$

Assim, em vez de inverter uma matriz, resolve-se um sistema de equações, o que pode envolver menos operações.

$$f(x,y) = -xy + x$$
$$g(x,y) = x^3 - y$$

$$J = \begin{bmatrix} -y+1 & -x \\ 3x^2 & -1 \end{bmatrix}$$

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \begin{bmatrix} -y_k+1 & -x_k \\ 3x_k^2 & -1 \end{bmatrix}^{-1} \begin{bmatrix} -x_k y_k + x_k \\ x_k^3 - y_k \end{bmatrix}$$

Ordem de Convergência

A ordem de convergência do método é uma medida da velocidade de convergência do método. O método diz-se de ordem p, se:

$$\left|x_{k+1} - x^*\right| < C\left|x_k - x^*\right|^p$$

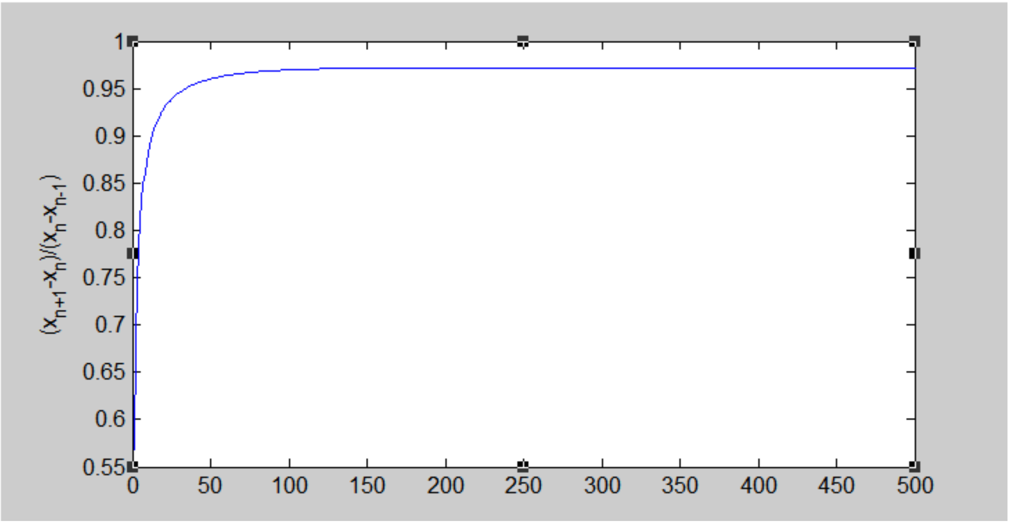
O método da bisecção é um exemplo de um método de ordem 1 (linear):

$$e_n = (1/2)^n\, e_0$$

O método do ponto fixo é em geral um método de ordem linear. No entanto o método de Newton é um caso particular de um método do ponto fixo que tem convergência quadrática (em geral)

$$f(x)=x^3-10z^2+10z+15$$

```
s=zeros(1,1000);
ct=1;
z=-10;
d=10;
lambda= 0.001;
while abs(d)>0.00000001
    d= -lambda*(z.^3-10*z.^2+10*z+15);
    s(ct)=d; ct= ct+1;
    z= z + d;
    d= abs(d./z);
 end
 s1= s(1:end-1); s2=s(2:end);
 s= s2./s1;
 s= s(s~=0);
 plot(s)
 ylabel('(x_{n+1}-x_{n})/(x_{n}-x_{n-1})');
```



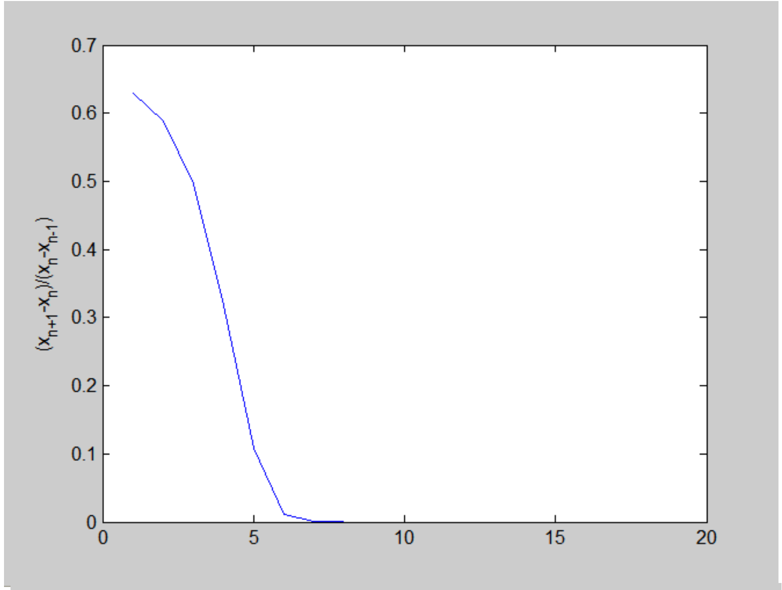
São necessárias mais do que 500 iteracções.

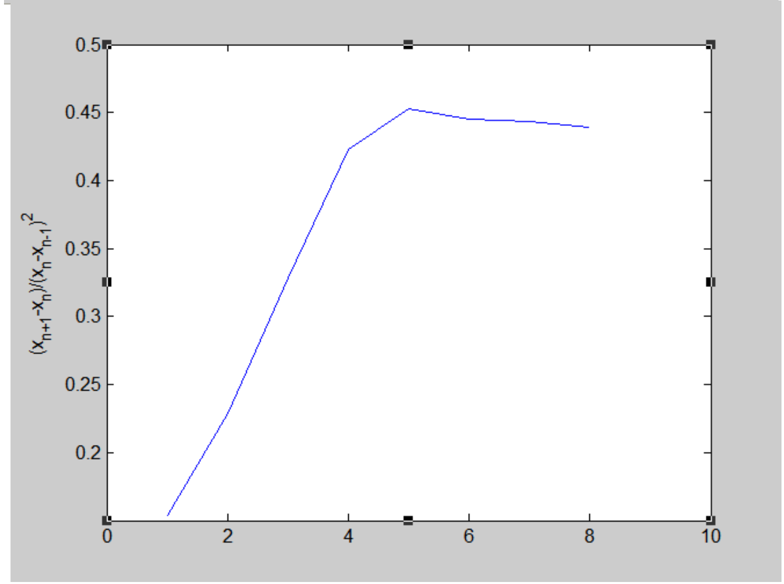
Fazendo o mesmo para o método de Newton

Começando no mesmo ponto, precisamos só de 10 iterações!

```
figure(2);
s=zeros(1,1000);
ct=1;
z=-10;
d=10;
lambda= 0.001;
while d>0.00000001
    d= -(z.^3-10*z.^2+10*z+15)./(3*z.^2-20*z+10)
    s(ct)=d; ct= ct+1;
    z= z + d;
    d= abs(d./z);
 end
 s1= s(1:end-1); s2=s(2:end);
 %s= s2./s1;   % esta possibilidade é para
              % testar convergencia linear
 s= s2./s1.^2;

 s= s(s~=0);
 plot(s)
 ylabel('(x_{n+1}-x_{n})/(x_{n}-x_{n-1})^2');
```





Podemos compreender porque é que o método de Newton converge quadráticamente usando uma decomposição em Taylor à volta do zero da função, x^*. Temos que:

$$f(x_i) = f(x^* + \epsilon_i) = f(x^*) + \epsilon_i \frac{df(x^*)}{dx} + \frac{(\epsilon_i)^2}{2}\frac{d^2 f(x^*)}{dx^2} + \cdots \cong \epsilon_i \frac{df(x^*)}{dx} + \frac{(\epsilon_i)^2}{2}\frac{d^2 f(x^*)}{dx^2}$$

$$f'(x_i) = f'(x^* + \epsilon_i) \cong \frac{df(x^*)}{dx} + \epsilon_i \frac{d^2 f(x^*)}{dx^2} + \cdots$$

onde nestas expressões ϵ_i é o erro cometido na iteração i (i.e., $\epsilon_i = x_i - x^*$)

Aplicando na fórmula do método de Newton ($x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$) vem:

$$x_{i+1} - x^* = x_i - x^* - \frac{f(x_i)}{f'(x_i)} \Leftrightarrow \epsilon_{i+1} = \epsilon_i - \frac{f(x_i)}{f'(x_i)}$$

$$\frac{f(x_i)}{f'(x_i)} = \frac{\epsilon_i \frac{df(x^*)}{dx} + \frac{(\epsilon_i)^2}{2}\frac{d^2 f(x^*)}{dx^2}}{\frac{df(x^*)}{dx} + \epsilon_i \frac{d^2 f(x^*)}{dx^2}} = \epsilon_i \frac{\frac{df(x^*)}{dx} + \epsilon_i \frac{d^2 f(x^*)}{dx^2}}{\frac{df(x^*)}{dx} + \epsilon_i \frac{d^2 f(x^*)}{dx^2}} - \epsilon_i \frac{\frac{\epsilon_i}{2}\frac{d^2 f(x^*)}{dx^2}}{\frac{df(x^*)}{dx} + \epsilon_i \frac{d^2 f(x^*)}{dx^2}} =$$

$$\frac{f(x_i)}{f'(x_i)} = \epsilon_i - \epsilon_i \frac{\frac{\epsilon_i}{2}\frac{d^2 f(x^*)}{dx^2}}{\frac{df(x^*)}{dx} + \epsilon_i \frac{d^2 f(x^*)}{dx^2}} \cong \epsilon_i - \frac{(\epsilon_i)^2}{2}\frac{\frac{d^2 f(x^*)}{dx^2}}{\frac{df(x^*)}{dx}}$$

$$\epsilon_{i+1} = \epsilon_i - \epsilon_i + \frac{(\epsilon_i)^2}{2}\frac{\frac{d^2 f(x^*)}{dx^2}}{\frac{df(x^*)}{dx}}$$

$$\boxed{\epsilon_{i+1} = -\frac{(\epsilon_i)^2}{2}\frac{\frac{d^2 f(x^*)}{dx^2}}{\frac{df(x^*)}{dx}}}$$

Ou seja, se as derivadas não divergirem, então o erro varia quadráticamente ao longo das iterações.