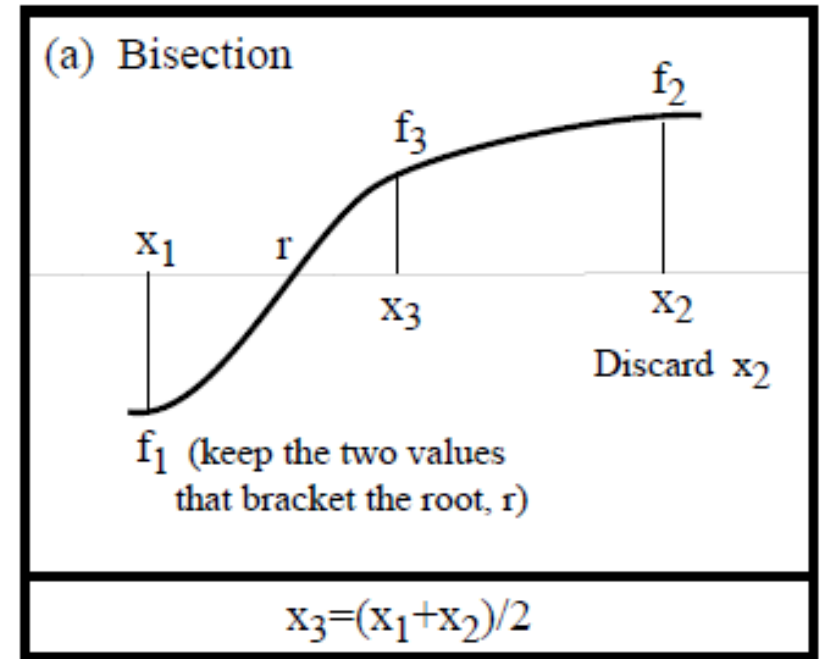


Equações não lineares

Resolver a equação: $f(x) = 0, x \in \mathbb{R}^1$ é uma raiz da equação.

Método da bisecção:

1. Dado $[a_i, b_i]$: $f(a_i).f(b_i) < 0$
2. Calcular $p_i = (a_i + b_i)/2$
3. Enquanto $||p_{i+1} - p_i|| > \varepsilon$ faz:
 Se $f(a_i).f(p_i) < 0$ então $a_{i+1} = a_i$ e $b_{i+1} = p_i$
 senão $a_{i+1} = p_i$ e $b_{i+1} = b_i$
 Calcular $p_{i+1} = (a_{i+1} + b_{i+1})/2$



Método da bissecção: $f(x) = x^3 - 2x - 5$

k	x_a	\bar{x}	x_b	$f(x_a)$	$f(\bar{x})$	$f(x_b)$	δ
1	1.5000000	2.0000000	2.5000000	-4.6250000	-1.0000000	5.6250000	0.5000000
2	2.0000000	2.2500000	2.5000000	-1.0000000	1.8906250	5.6250000	0.2500000
3	2.0000000	2.1250000	2.2500000	-1.0000000	0.3457031	1.8906250	0.1250000
4	2.0000000	2.0625000	2.1250000	-1.0000000	-0.3513184	0.3457031	0.0625000
5	2.0625000	2.0937500	2.1250000	-0.3513184	-0.0089417	0.3457031	0.0312500
6	2.0937500	2.1093750	2.1250000	-0.0089417	0.1668358	0.3457031	0.0156250
7	2.0937500	2.1015625	2.1093750	-0.0089417	0.0785623	0.1668358	0.0078125
8	2.0937500	2.0976563	2.1015625	-0.0089417	0.0347143	0.0785623	0.0039063
9	2.0937500	2.0957031	2.0976563	-0.0089417	0.0128623	0.0347143	0.0019531
10	2.0937500	2.0947266	2.0957031	-0.0089417	0.0019543	0.0128623	0.0009766
11	2.0937500	2.0942383	2.0947266	-0.0089417	-0.0034951	0.0019543	0.0004883
12	2.0942383	2.0944824	2.0947266	-0.0034951	-0.0007708	0.0019543	0.0002441
13	2.0944824	2.0946045	2.0947266	-0.0007708	0.0005917	0.0019543	0.0001221
14	2.0944824	2.0945435	2.0946045	-0.0007708	-0.0000896	0.0005917	0.0000610
15	2.0945435	2.0945740	2.0946045	-0.0000896	0.0002511	0.0005917	0.0000305
16	2.0945435	2.0945587	2.0945740	-0.0000896	0.0000807	0.0002511	0.0000153
17	2.0945435	2.0945511	2.0945587	-0.0000896	-0.0000044	0.0000807	0.0000076
18	2.0945511	2.0945549	2.0945587	-0.0000044	0.0000382	0.0000807	0.0000038
19	2.0945511	2.0945530	2.0945549	-0.0000044	0.0000169	0.0000382	0.0000019
20	2.0945511	2.0945520	2.0945530	-0.0000044	0.0000062	0.0000169	0.0000010
21	2.0945511	2.0945516	2.0945520	-0.0000044	0.0000009	0.0000062	0.0000005
22	2.0945511	2.0945513	2.0945516	-0.0000044	-0.0000017	0.0000009	0.0000002
23	2.0945513	2.0945514	2.0945516	-0.0000017	-0.0000004	0.0000009	0.0000001
24	2.0945514	2.0945515	2.0945516	-0.0000004	0.0000002	0.0000009	0.0000001
25	2.0945514	2.0945515	2.0945515	-0.0000004	-0.0000001	0.0000002	0.0000000
26	2.0945515	2.0945515	2.0945515	-0.0000001	0.0000001	0.0000002	0.0000000

Método da Falsa posição (ou Corda-falsa “Regula-Falsi”):

Vamos aproximar $f(x)$ por uma recta entre os pontos a, b e limitar o intervalo da raiz da equação pelo ponto da recta que passa por zero.

1. Dado $[a_i, b_i]$: $f(a_i) \cdot f(b_i) < 0$

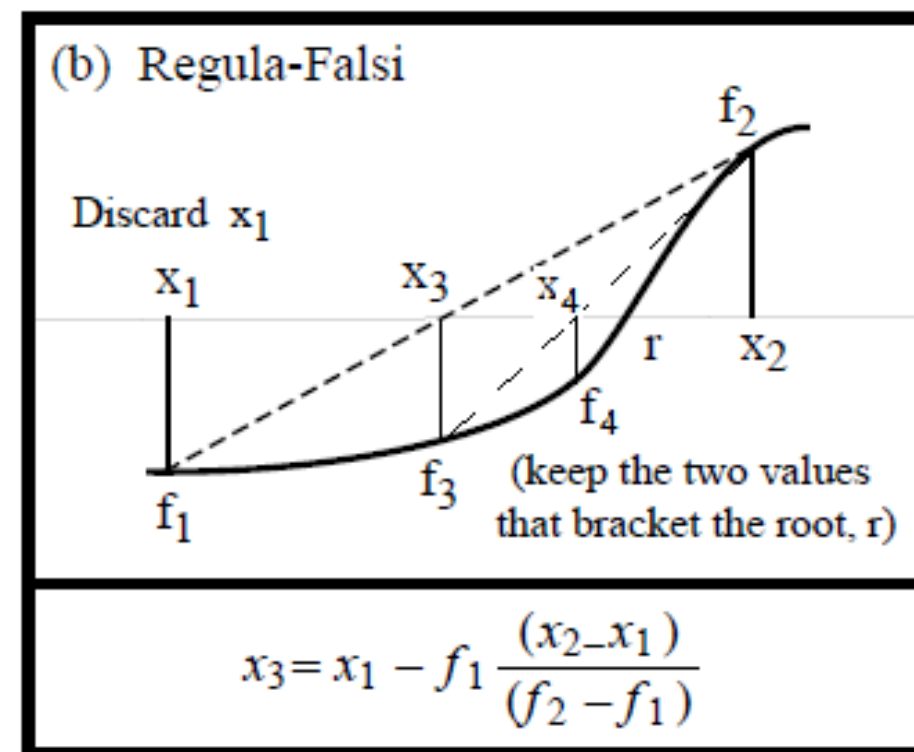
2. Calcular $p_i = (a_i + b_i) / 2$

3. Enquanto $|f(p_i)| > \epsilon$ faz:

Se $f(a_i) \cdot f(p_i) < 0$ então $a_{i+1} = a_i$ e $b_{i+1} = p_i$

senão $a_{i+1} = p_i$ e $b_{i+1} = b_i$

Calcular $p_{i+1} = a_{i+1} - f(a_{i+1})(b_{i+1} - a_{i+1}) / (f(b_{i+1}) - f(a_{i+1}))$



Se a concavidade de $f(x)$ se mantiver em $[a, b]$ um dos extremos fica fixo pelo que para parar o método não se deve usar intervalo de localização da raiz, pois a sua amplitude pode manter-se sempre acima de um certo valor.

k	x_a	\bar{x}	x_b	$f(x_a)$	$f(\bar{x})$	$f(x_b)$	δ
1	1.5000000	1.9512195	2.5000000	-4.6250000	-1.4736437	5.6250000	0.4512195
2	1.9512195	2.0651437	2.5000000	-1.4736437	-0.3228248	5.6250000	0.1139241
3	2.0651437	2.0887460	2.5000000	-0.3228248	-0.0645863	5.6250000	0.0236023
4	2.0887460	2.0934144	2.5000000	-0.0645863	-0.0126835	5.6250000	0.0046684
5	2.0934144	2.0943291	2.5000000	-0.0126835	-0.0024817	5.6250000	0.0009147
6	2.0943291	2.0945080	2.5000000	-0.0024817	-0.0004852	5.6250000	0.0001789
7	2.0945080	2.0945430	2.5000000	-0.0004852	-0.0000949	5.6250000	0.0000350
8	2.0945430	2.0945498	2.5000000	-0.0000949	-0.0000185	5.6250000	0.0000068
9	2.0945498	2.0945512	2.5000000	-0.0000185	-0.0000036	5.6250000	0.0000013
10	2.0945512	2.0945514	2.5000000	-0.0000036	-0.0000007	5.6250000	0.0000003
11	2.0945514	2.0945515	2.5000000	-0.0000007	-0.0000001	5.6250000	0.0000001
12	2.0945515	2.0945515	2.5000000	-0.0000001	-0.0000000	5.6250000	0.0000000

Método do ponto fixo (ou da substituição sucessiva)

Vamos reorganizar o problema $f(x) = 0$ e transformá-lo em: $x = g(x)$

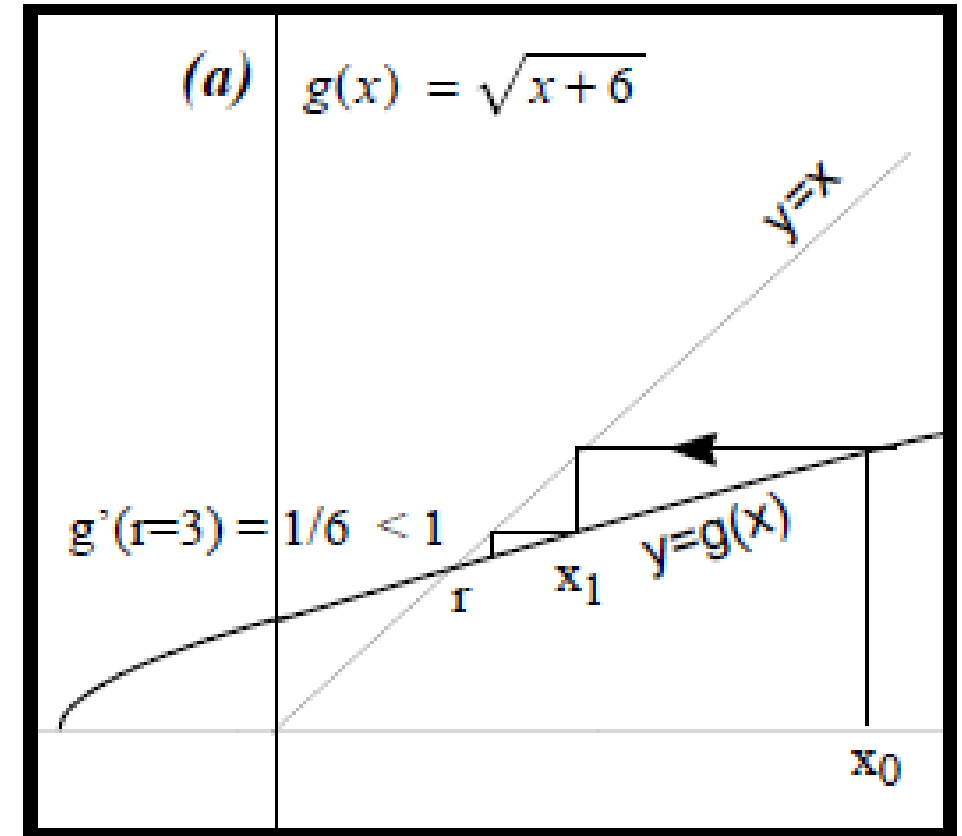
1. Dado x_0
2. Enquanto $||x_{i+1} - x_i|| > \varepsilon$ faz:

$$x_{i+1} = g(x_i)$$

Este método converge para a raiz se $|g'(r)| < 1$!

Exemplo:

Calcular raízes de $f(x) = x^2 - x - 6 = 0$. Experimentar com $g(x) = (x+6)^{0.5}$ e $g(x) = x^2 - 6$.



Método de Newton-Raphson

Expandindo $f(x)$ em série de Taylor

$$\tilde{f}(x_i + \delta) = f(x_i) + f'(x_i)\delta + f''(x_i)\frac{\delta^2}{2!} + \mathcal{O}(\delta^3)$$

e desprezando os termos de ordem superior a 2 temos:

$$\tilde{f}(x_i + \delta) = 0 = f(x_i) + f'(x_i)(x_{i+1} - x_i) + \mathcal{O}(\delta^2)$$

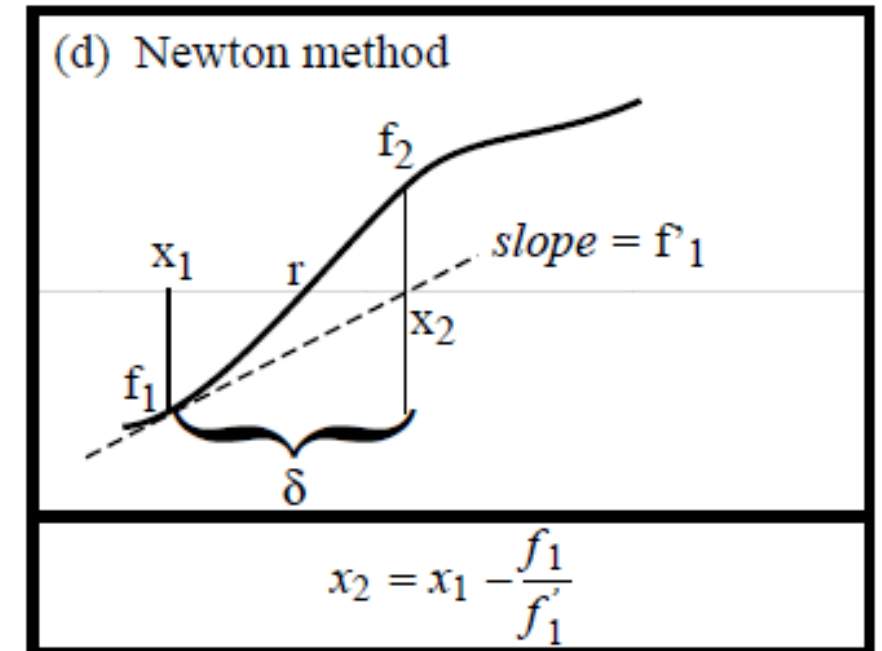
1. Dado x_0

2. Enquanto $||x_{i+1} - x_i|| > \varepsilon$ faz:

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

Exemplo:

Comparar o método de Newton com a função do Matlab `fzero(f,x0,tol)` para determinar raízes.



Método de Newton: $f(x) = x^3 - 2x - 5$

k	\bar{x}	$f(\bar{x})$	δ
0	1.5000000	-4.6250000	0.0000038
1	2.4736842	5.1893862	-0.9736842
2	2.1564330	0.7149858	0.3172512
3	2.0966046	0.0229423	0.0598284
4	2.0945539	0.0000264	0.0020508
5	2.0945515	0.0000000	0.0000024

k	\bar{x}	$f(\bar{x})$	δ
0	2.5000000	5.6250000	0.0000000
1	2.1641791	0.8079451	0.3358209
2	2.0971354	0.0288817	0.0670437
3	2.0945552	0.0000419	0.0025801
4	2.0945515	0.0000000	0.0000038

Método de Newton: $f(x) = x^3 - 2x - 5$

k	\bar{x}	$f(\bar{x})$	δ
0	0.8160000	-6.0886615	0.0000024
1	-2502.7454737	-15676529183.9705276	2503.5614737
2	-1668.4971598	-4644897166.5104160	-834.2483139
3	-1112.3317056	-1376265581.2256701	-556.1654541
4	-741.5548686	-407782230.9435600	-370.7768370
5	-494.3705087	-120824256.1601983	-247.1843599
6	-329.5812313	-35799707.6590384	-164.7892774
7	-219.7221541	-10607273.2570007	-109.8590773
8	-146.4834243	-3142864.5242462	-73.2387298
9	-97.6585727	-931198.7126192	-48.8248516
10	-65.1100917	-275897.5569029	-32.5484810
11	-43.4131617	-81739.0728663	-21.6969300
12	-28.9514644	-24213.8464756	-14.4616974
13	-19.3143498	-7171.4757192	-9.6371145

k	\bar{x}	$f(\bar{x})$	δ
13	-19.3143498	-7171.4757192	-9.6371145
14	-12.8948098	-2123.3093018	-6.4195401
15	-8.6210817	-628.5029153	-4.2737281
16	-5.7767800	-186.2244480	-2.8443017
17	-3.8787300	-55.5962734	-1.8980500
18	-2.5897993	-17.1903410	-1.2889307
19	-1.6411667	-6.1380315	-0.9486325
20	-0.6316694	-3.9887012	-1.0094973
21	-5.5990345	-169.3271102	4.9673651
22	-3.7594734	-50.6160980	-1.8395611
23	-2.5066282	-15.7363530	-1.2528452
24	-1.5726954	-5.7444683	-0.9339328
25	-0.5128525	-4.1091843	-1.0598430
26	-3.9062167	-56.7906882	3.3933643

Método da Secante:

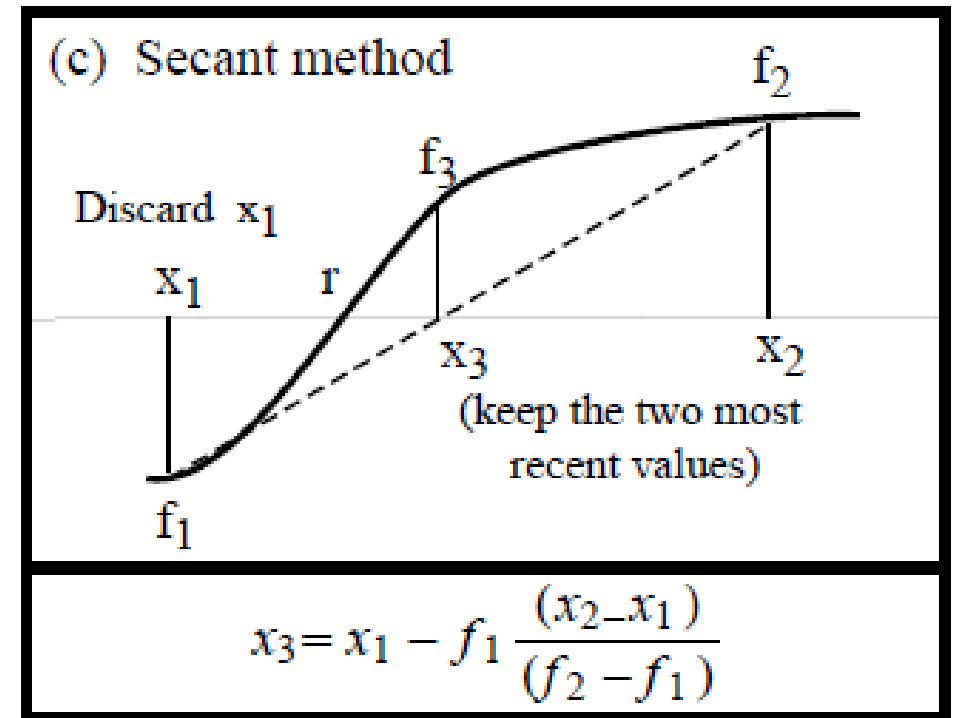
Vamos aproximar $f(x)$ por uma recta entre os pontos a, b e aproximar a raiz da equação pelo ponto dessa recta que passa por zero.

1. Dado $[x_0, x_1]$: $f(x_0).f(x_1) < 0$

2. Enquanto $|f(x_{i+1})| > \varepsilon$ faz:

$$x_{i+1} = x_i - f(x_i) \cdot (x_i - x_{i-1}) / (f(x_i) - f(x_{i-1}))$$

Este método é mais rápido que da Corda Falsa, no entanto não mantém o intervalo de localização da raiz. É uma variante do método de Newton, que usa uma aproximação numérica da derivada.



Razão da convergência

Iterações $x_0, x_1, x_2, x_3 \dots \rightarrow \bar{x}$.

Se existem um parâmetro p e uma constante $C \neq 0$ tal que

$$\lim_{k \rightarrow \infty} \frac{|\bar{x} - x_{k+1}|}{|\bar{x} - x_k|^p} = C$$

$\Rightarrow p =$ **razão de convergência** do método e

$\Rightarrow C =$ **constante de convergência assintótica**.

bisseccção: $p = 1$

convergência linear

secante: $p = \frac{1}{2}(1 + \sqrt{5}) \simeq 1.618 \dots$

convergência super-linear

Newton: $p = 2$

convergência quadrática

Técnica da deflação para determinar várias raízes.

Se a raiz r é conhecida para $f(x)=0$ outras raízes podem ser determinadas resolvendo o problema:

$$g(x) = f(x)/(x-r) = 0$$

Uma vez que r agora não é raiz de $g(x)$. O processo pode ser generalizado se várias raízes forem conhecidas.

No caso de raízes com multiplicidade $m > 1$ pode-se usar o método de Newton da seguinte forma:

1. Dado x_0
2. Enquanto $||x_{i+1} - x_i|| > \varepsilon$ faz:

$$x_{i+1} = x_i - m \cdot f(x_i) / f'(x_i)$$

Exemplo:

Determinar os zeros de $f(x) = (x-2)^2 \exp(-0.8x)$.

Sistemas de equações não lineares.

$$\begin{cases} f_1(x_1, x_2) = 0 \\ f_2(x_1, x_2) = 0 \end{cases}$$

Expandindo f_1 e f_2 em série de Taylor:

$$f_1(x_1^{(0)} + \delta_1, x_2^{(0)} + \delta_2) = f_1(x_1^{(0)}, x_2^{(0)}) + \underbrace{\frac{\partial f_1}{\partial x_1} \Big|_{[x_1^{(0)}, x_2^{(0)}]} \delta_1}_{\text{variation due to } x_1} + \underbrace{\frac{\partial f_1}{\partial x_2} \Big|_{[x_1^{(0)}, x_2^{(0)}]} \delta_2}_{\text{variation due to } x_2} + \mathcal{O}(\delta^2)$$

$$f_2(x_1^{(0)} + \delta_1, x_2^{(0)} + \delta_2) = f_2(x_1^{(0)}, x_2^{(0)}) + \underbrace{\frac{\partial f_2}{\partial x_1} \Big|_{[x_1^{(0)}, x_2^{(0)}]} \delta_1}_{\text{variation due to } x_1} + \underbrace{\frac{\partial f_2}{\partial x_2} \Big|_{[x_1^{(0)}, x_2^{(0)}]} \delta_2}_{\text{variation due to } x_2} + \mathcal{O}(\delta^2)$$

Temos:

$$\begin{aligned} 0 &= f_1(x_1^{(0)}, x_2^{(0)}) + \frac{\partial f_1}{\partial x_1} \Big|_{[x_1^{(0)}, x_2^{(0)}]} \delta_1 + \frac{\partial f_1}{\partial x_2} \Big|_{[x_1^{(0)}, x_2^{(0)}]} \delta_2 \\ 0 &= f_2(x_1^{(0)}, x_2^{(0)}) + \frac{\partial f_2}{\partial x_1} \Big|_{[x_1^{(0)}, x_2^{(0)}]} \delta_1 + \frac{\partial f_2}{\partial x_2} \Big|_{[x_1^{(0)}, x_2^{(0)}]} \delta_2 \end{aligned} \quad \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix}$$

Sistemas de equações não lineares.

Método de Newton:

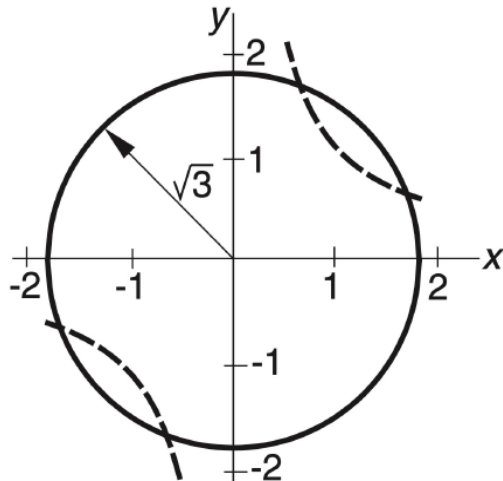
1) Resolver o sistema de equações e obter δ :

$$\delta^{(k)} = -[J^{(k)}]^{-1} f^{(k)}$$

onde J se designa por **matriz Jacobiana**.

2) Obter nova estimativa para a solução:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \delta^{(k)} \quad k = 0, 1, \dots$$



$$f_1(x, y) = x^2 + y^2 - 3 = 0$$

$$f_2(x, y) = xy - 1 = 0$$

```
from scipy.optimize import fsolve
import math

def equations(p):
    x, y = p
    return (x**2+y**2-3, x*y - 1)

x, y = fsolve(equations, (1, 1))

print(x,y)
print(equations((x, y)))
```

Sistemas de equações não lineares.

Método de Broyden:

1) Na resolução do sistema de equações para obter os acréscimos:
vamos usar uma aproximação para a matriz Jacobiana:

$$\mathbf{J}_n(\mathbf{x}_n - \mathbf{x}_{n-1}) \simeq -\mathbf{f}(\mathbf{x}_{n-1})$$

com \mathbf{J}_n sendo actualizada de acordo:

$$\mathbf{J}_n = \mathbf{J}_{n-1} + \frac{\Delta \mathbf{f}_n - \mathbf{J}_{n-1} \Delta \mathbf{x}_n}{\|\Delta \mathbf{x}_n\|^2} \Delta \mathbf{x}_n^T$$

2) Obter nova estimativa para a solução:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta \mathbf{x}^{(k)} \quad k = 0, 1, \dots$$