

# Método de Newton

Prof. Americo Cunha

Universidade do Estado do Rio de Janeiro – UERJ

[americo.cunha@uerj.br](mailto:americo.cunha@uerj.br)

[www.americocunha.org](http://www.americocunha.org)



Encontre uma solução para a equação cúbica:

$$x^3 - 2x - 4 = 0$$



Encontre uma solução para a equação cúbica:

$$x^3 - 2x - 4 = 0$$

Candidatos à solução:  $\pm 2$  ,  $\pm 4$



Encontre uma solução para a equação cúbica:

$$x^3 - 2x - 4 = 0$$

Candidatos à solução:  $\pm 2$  ,  $\pm 4$

Olhômetro:  $x = 2$  é solução!



Encontre uma solução para a equação cúbica:

$$x^3 - 2x - 4 = 0$$

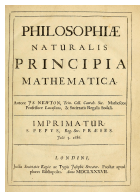
Candidatos à solução:  $\pm 2$  ,  $\pm 4$

Olhômetro:  $x = 2$  é solução!

E se a equação fosse

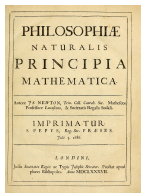
$$x^3 - 2x - 4,1 = 0 \quad ?$$





$$x^3 - 2x - 4,1 = 0$$

**Sir Isaac Newton (1687)**

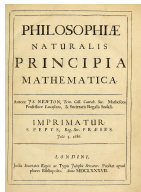
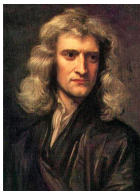


$$x^3 - 2x - 4,1 = 0$$

2 está “perto” da raiz  
(continuidade)

**Sir Isaac Newton (1687)**





Sir Isaac Newton (1687)

$$x^3 - 2x - 4,1 = 0$$

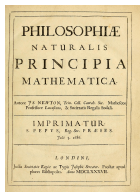
2 está “perto” da raiz  
(continuidade)

$$x = 2 + \Delta x$$

“raiz = chute + correção”







**Sir Isaac Newton (1687)**

$$\Rightarrow (2 + \Delta x)^3 - 2(2 + \Delta x) - 4,1 = 0$$

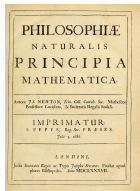
$$x^3 - 2x - 4,1 = 0$$

2 está “perto” da raiz  
(continuidade)

$$x = 2 + \Delta x$$

“raiz = chute + correção”





$$x^3 - 2x - 4,1 = 0$$

2 está “perto” da raiz  
(continuidade)

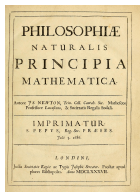
$$x = 2 + \Delta x$$

“raiz = chute + correção”

**Sir Isaac Newton (1687)**

$$\Rightarrow (2 + \Delta x)^3 - 2(2 + \Delta x) - 4,1 = 0$$

$$\Rightarrow 8 + 6\Delta x^2 + 12\Delta x + \Delta x^3 - 4 - 2\Delta x - 4,1 = 0$$



**Sir Isaac Newton (1687)**

$$x^3 - 2x - 4,1 = 0$$

2 está “perto” da raiz  
(continuidade)

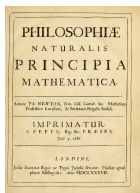
$$x = 2 + \Delta x$$

“raiz = chute + correção”

$$\Rightarrow (2 + \Delta x)^3 - 2(2 + \Delta x) - 4,1 = 0$$

$$\Rightarrow 8 + 6\cancel{\Delta x^2} + 12\Delta x + \cancel{\Delta x^3} - 4 - 2\Delta x - 4,1 = 0$$





**Sir Isaac Newton (1687)**

$$x^3 - 2x - 4,1 = 0$$

2 está “perto” da raiz  
(continuidade)

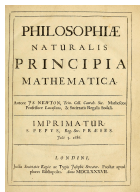
$$x = 2 + \Delta x$$

“raiz = chute + correção”

$$\Rightarrow (2 + \Delta x)^3 - 2(2 + \Delta x) - 4,1 = 0$$

$$\Rightarrow 8 + 6\cancel{\Delta x^2} + 12\Delta x + \cancel{\Delta x^3} - 4 - 2\Delta x - 4,1 = 0$$

$$\Rightarrow 8 + 12\Delta x - 4 - 2\Delta x - 4,1 \approx 0$$



**Sir Isaac Newton (1687)**

$$x^3 - 2x - 4,1 = 0$$

2 está “perto” da raiz  
(continuidade)

$$x = 2 + \Delta x$$

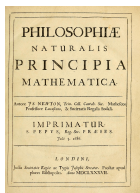
“raiz = chute + correção”

$$\Rightarrow (2 + \Delta x)^3 - 2(2 + \Delta x) - 4,1 = 0$$

$$\Rightarrow 8 + 6\cancel{\Delta x^2} + 12\Delta x + \cancel{\Delta x^3} - 4 - 2\Delta x - 4,1 = 0$$

$$\Rightarrow 8 + 12\Delta x - 4 - 2\Delta x - 4,1 \approx 0$$

$$\Rightarrow 10\Delta x - 0,1 \approx 0$$



**Sir Isaac Newton (1687)**

$$x^3 - 2x - 4,1 = 0$$

2 está “perto” da raiz  
(continuidade)

$$x = 2 + \Delta x$$

“raiz = chute + correção”

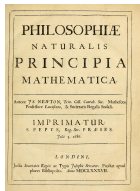
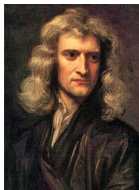
$$\Rightarrow (2 + \Delta x)^3 - 2(2 + \Delta x) - 4,1 = 0$$

$$\Rightarrow 8 + 6\cancel{\Delta x^2} + 12\Delta x + \cancel{\Delta x^3} - 4 - 2\Delta x - 4,1 = 0$$

$$\Rightarrow 8 + 12\Delta x - 4 - 2\Delta x - 4,1 \approx 0$$

$$\Rightarrow 10\Delta x - 0,1 \approx 0$$

$$\Rightarrow \Delta x \approx 0,01$$



**Sir Isaac Newton (1687)**

$$x^3 - 2x - 4,1 = 0$$

2 está “perto” da raiz  
(continuidade)

$$x = 2 + \Delta x$$

“raiz = chute + correção”

$$\Rightarrow (2 + \Delta x)^3 - 2(2 + \Delta x) - 4,1 = 0$$

$$\Rightarrow 8 + 6\cancel{\Delta x^2} + 12\Delta x + \cancel{\Delta x^3} - 4 - 2\Delta x - 4,1 = 0$$

$$\Rightarrow 8 + 12\Delta x - 4 - 2\Delta x - 4,1 \approx 0$$

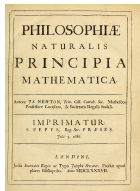
$$\Rightarrow 10\Delta x - 0,1 \approx 0$$

$$\Rightarrow \Delta x \approx 0,01$$

$$x \approx 2 + 0,01 = 2,01$$

(aproximação)





**Sir Isaac Newton (1687)**

$$x^3 - 2x - 4,1 = 0$$

2 está “perto” da raiz  
(continuidade)

$$x = 2 + \Delta x$$

“raiz = chute + correção”

$$\Rightarrow (2 + \Delta x)^3 - 2(2 + \Delta x) - 4,1 = 0$$

$$\Rightarrow 8 + 6\cancel{\Delta x^2} + 12\Delta x + \cancel{\Delta x^3} - 4 - 2\Delta x - 4,1 = 0$$

$$\Rightarrow 8 + 12\Delta x - 4 - 2\Delta x - 4,1 \approx 0$$

$$\Rightarrow 10\Delta x - 0,1 \approx 0$$

$$\Rightarrow \Delta x \approx 0,01$$

$$x \approx 2 + 0,01 = 2,01$$

(aproximação)

$$x = 2,00994 \dots$$

(valor exato)





# Método de Newton

É um *método numérico* para calcular uma *solução aproximada* para a equação  $f(x) = 0$  partindo de um chute inicial  $x_0$ .

Hipóteses:

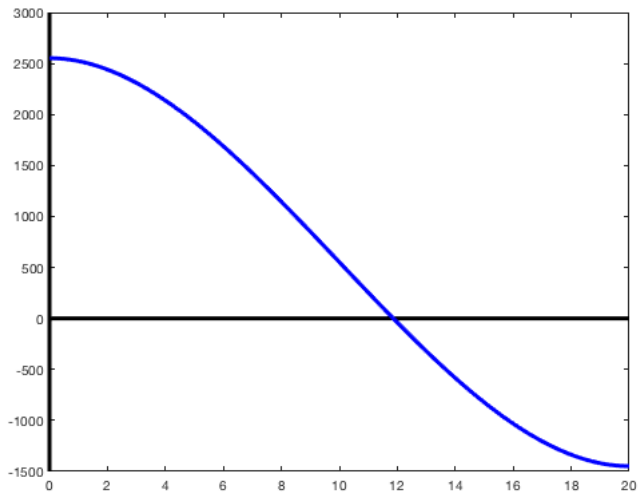
$$f \in C^2[a, b] \text{ e existe } x^* \in [a, b] \text{ tal que } f(x^*) = 0 \text{ e } f'(x^*) \neq 0$$

Ideia do método:

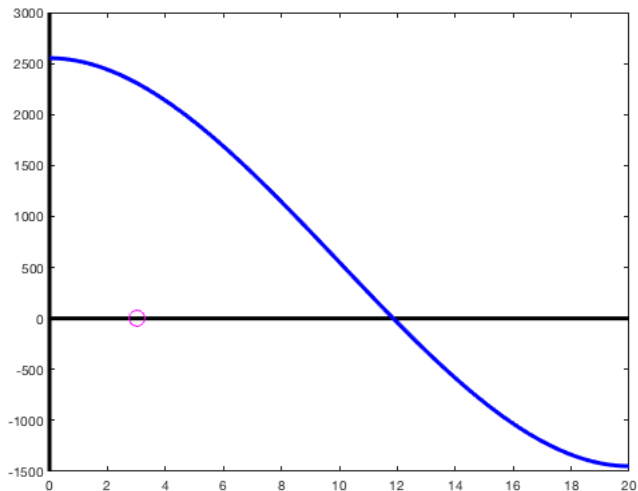
- A partir de um dado *chute inicial*  $x_0$ , calcule a *reta tangente* ao gráfico de  $f$  no ponto  $(x_0, f(x_0))$
- A *interseção* entre essa reta tangente e o eixo  $x$  fornece uma *nova aproximação* para  $x^*$
- Repita esse procedimento até obter uma *aproximação* com a *tolerância desejada*



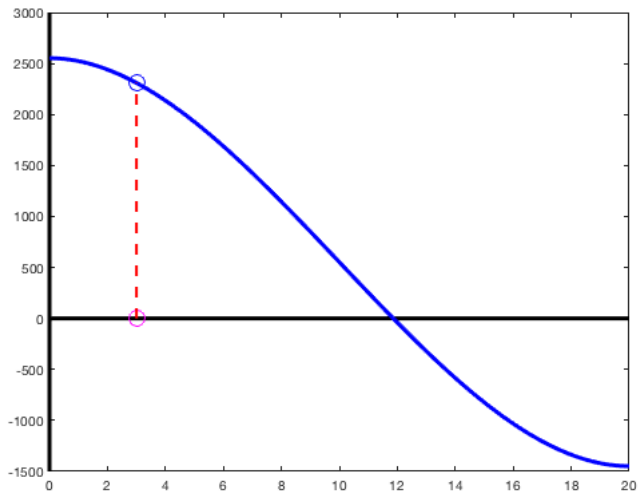
# Interpretação geométrica do método de Newton



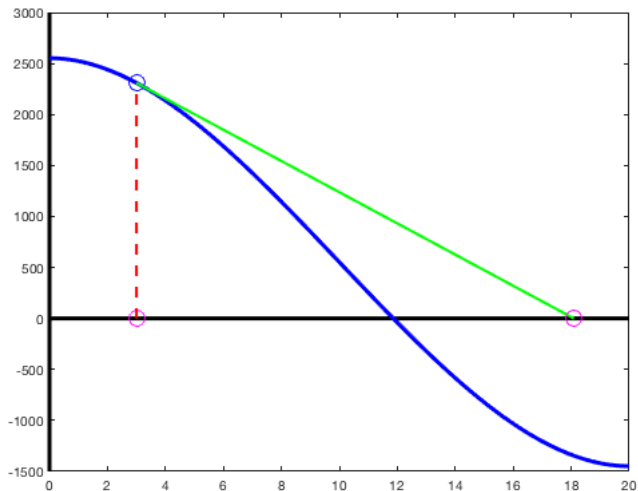
# Interpretação geométrica do método de Newton



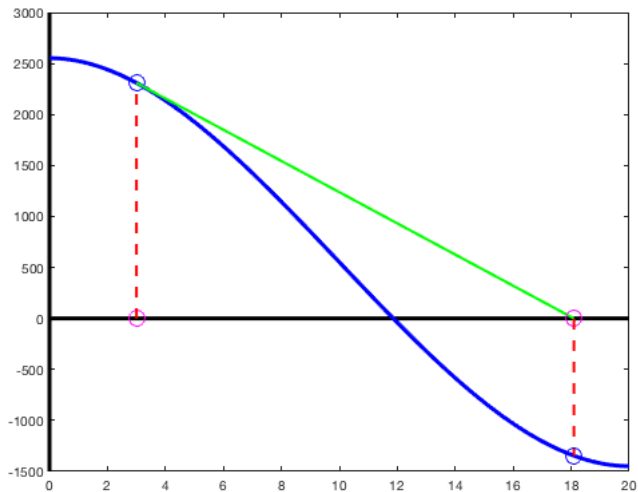
# Interpretação geométrica do método de Newton



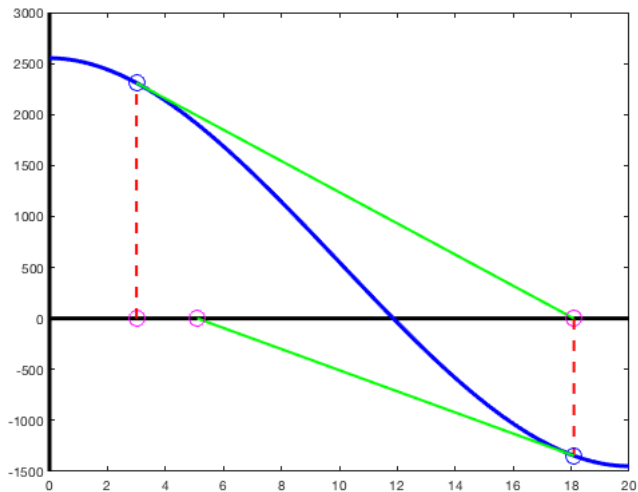
# Interpretação geométrica do método de Newton



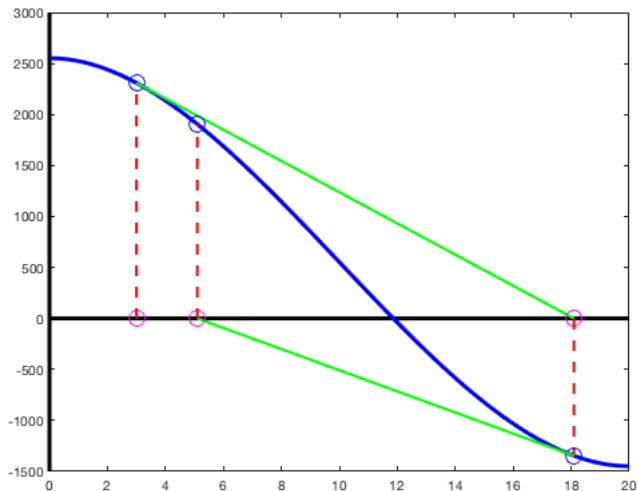
# Interpretação geométrica do método de Newton



# Interpretação geométrica do método de Newton

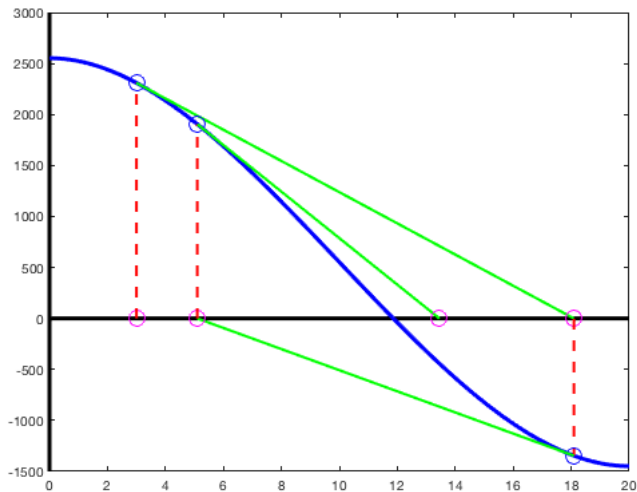


# Interpretação geométrica do método de Newton

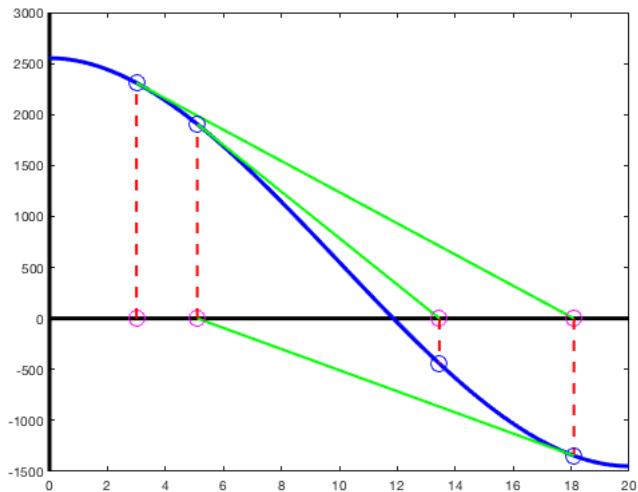




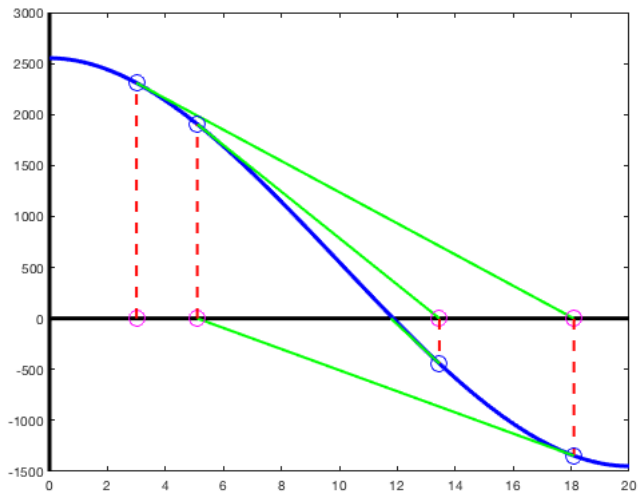
# Interpretação geométrica do método de Newton



# Interpretação geométrica do método de Newton



# Interpretação geométrica do método de Newton



# Desenvolvimento analítico do método de Newton



# Desenvolvimento analítico do método de Newton

Reta tangente ao gráfico de  $f$  em  $(x_n, f(x_n))$ :

$$y = f(x_n) + f'(x_n)(x - x_n)$$



# Desenvolvimento analítico do método de Newton

Reta tangente ao gráfico de  $f$  em  $(x_n, f(x_n))$ :

$$y = f(x_n) + f'(x_n)(x - x_n)$$

Na interseção entre a reta e o eixo  $x$  temos  $y = 0$  e  $x = x_{n+1}$ :

$$0 = f(x_n) + f'(x_n)(x_{n+1} - x_n)$$



# Desenvolvimento analítico do método de Newton

Reta tangente ao gráfico de  $f$  em  $(x_n, f(x_n))$ :

$$y = f(x_n) + f'(x_n)(x - x_n)$$

Na interseção entre a reta e o eixo  $x$  temos  $y = 0$  e  $x = x_{n+1}$ :

$$0 = f(x_n) + f'(x_n)(x_{n+1} - x_n)$$

Isolando  $x_{n+1}$ , obtemos:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad f'(x_n) \neq 0$$



# Desenvolvimento analítico do método de Newton

Reta tangente ao gráfico de  $f$  em  $(x_n, f(x_n))$ :

$$y = f(x_n) + f'(x_n)(x - x_n)$$

Na interseção entre a reta e o eixo  $x$  temos  $y = 0$  e  $x = x_{n+1}$ :

$$0 = f(x_n) + f'(x_n)(x_{n+1} - x_n)$$

Isolando  $x_{n+1}$ , obtemos:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad f'(x_n) \neq 0$$

Esquema iterativo do *método de Newton*:

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





# Algoritmo do método de Newton

Input:  $f$ ,  $f'$ ,  $x_0$ ,  $\text{tol}$  and  $\text{maxiter}$

```
1: iter = 0
2: Error =  $\infty$ 
3: while termination criterion is not met (Error, tol, maxiter) do
4:   iter = iter + 1
5:   Compute the correction  $\Delta x = -f(x_0)/f'(x_0)$ 
6:   Update the approximation  $x_{\text{new}} = x_0 + \Delta x$ 
7:   Estimate the Error based on  $x_{\text{new}}$  and  $x_0$ 
8:   Update the initial guess  $x_0 = x_{\text{new}}$ 
9: end while
10: if Error > tol then
11:    $x_{\text{new}} = \text{NaN}$ 
12: end if
13: return
```

Output:  $x_{\text{new}}$ , iter



# Implementação em GNU Octave

```
function [xnew,iter] = newton(f,df,x0,tol,maxiter)
    iter = 0;
    Error = inf;
    while Error > tol && iter < maxiter
        iter = iter + 1;
        dx = - f(x0)/df(x0);
        xnew = x0 + dx;
        Error = abs(xnew-x0);
        x0 = xnew;
        fprintf([' iter = %3d    ',...
                ' root = %.16f ',...
                'Error = %.16f \n'],iter,xnew,Error);
    end
    if Error > tol
        xnew = NaN;
    end
end
```



# Experimento computacional 1



```
>> x0 = 0.5; tol = 1.0e-9; maxiter = 100;  
>> f = @(x) x*exp(x)-1;  
>> df = @(x) (1+x)*exp(x);  
>> root1 = newton(f,df,x0,tol,maxiter);
```



```
>> g = @(x) exp(-x);  
>> root2 = fixedpoint(g,x0,tol,maxiter);
```



```
>> a = 0.0; b = 1.0;  
>> root3 = bisection(f,a,b,tol);
```

# Experimento computacional 1



```
>> x0 = 0.5; tol = 1.0e-9; maxiter = 100;  
>> f = @(x) x*exp(x)-1;  
>> df = @(x) (1+x)*exp(x);  
>> root1 = newton(f,df,x0,tol,maxiter);
```



```
>> g = @(x) exp(-x);  
>> root2 = fixedpoint(g,x0,tol,maxiter);
```



```
>> a = 0.0; b = 1.0;  
>> root3 = bisection(f,a,b,tol);
```

**Em geral Newton é muito mais rápido  
que os outros métodos!**



## Experimento computacional 2



```
>> tol = 1.0e-9; maxiter = 100;
>> f = @(x) x^2 - 2;
>> df = @(x) 2*x;
>> x0 = 2.0;
>> root1 = newton(f,df,x0,tol,maxiter);
>> x0 = -2.0;
>> root2 = newton(f,df,x0,tol,maxiter);
```



## Experimento computacional 2



```
>> tol = 1.0e-9; maxiter = 100;  
>> f = @(x) x^2 - 2;  
>> df = @(x) 2*x;  
>> x0 = 2.0;  
>> root1 = newton(f,df,x0,tol,maxiter);  
>> x0 = -2.0;  
>> root2 = newton(f,df,x0,tol,maxiter);
```

**Chute iniciais diferentes podem convergir  
para raízes diferentes!**



**Mas nem tudo são flores ...**



# Patologia 1: chute inicial “longe” da raiz



```
>> x0 = 2.0; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) x*exp(-x ^ 2);  
>> df = @(x) exp(-x ^ 2) - 2*(x^ 2)*exp(-x ^ 2);  
>> root = newton(f,df,x0,tol,maxiter);
```

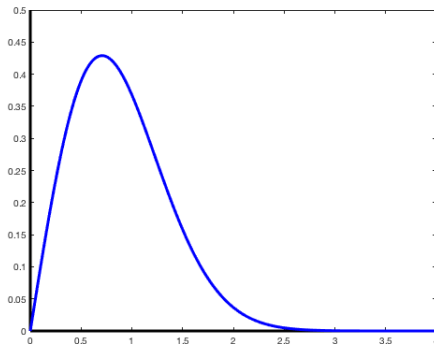




# Patologia 1: chute inicial “longe” da raiz



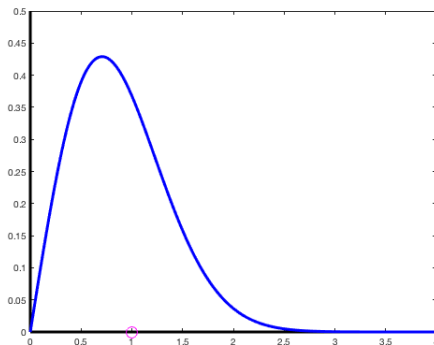
```
>> x0 = 2.0; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) x*exp(-x ^ 2);  
>> df = @(x) exp(-x ^ 2) - 2*(x^ 2)*exp(-x ^ 2);  
>> root = newton(f,df,x0,tol,maxiter);
```



# Patologia 1: chute inicial “longe” da raiz



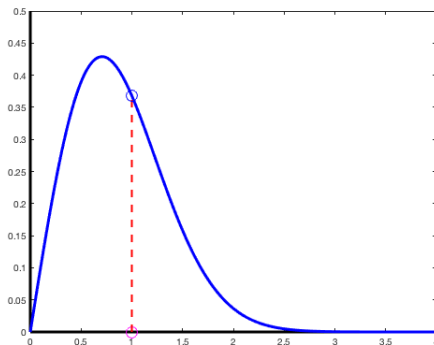
```
>> x0 = 2.0; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) x*exp(-x ^ 2);  
>> df = @(x) exp(-x ^ 2) - 2*(x^ 2)*exp(-x ^ 2);  
>> root = newton(f,df,x0,tol,maxiter);
```



# Patologia 1: chute inicial “longe” da raiz



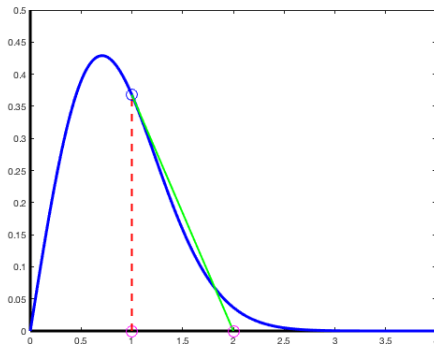
```
>> x0 = 2.0; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) x*exp(-x ^ 2);  
>> df = @(x) exp(-x ^ 2) - 2*(x^ 2)*exp(-x ^ 2);  
>> root = newton(f,df,x0,tol,maxiter);
```



# Patologia 1: chute inicial “longe” da raiz



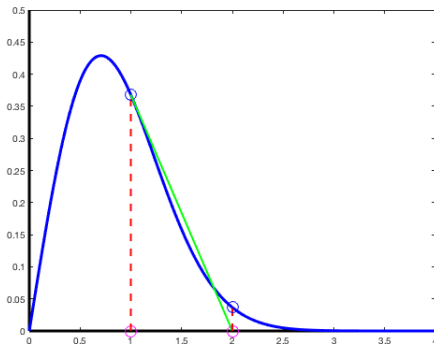
```
>> x0 = 2.0; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) x*exp(-x ^ 2);  
>> df = @(x) exp(-x ^ 2) - 2*(x^ 2)*exp(-x ^ 2);  
>> root = newton(f,df,x0,tol,maxiter);
```



# Patologia 1: chute inicial “longe” da raiz



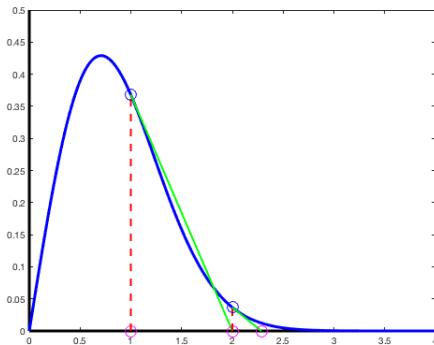
```
>> x0 = 2.0; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) x*exp(-x ^ 2);  
>> df = @(x) exp(-x ^ 2) - 2*(x^ 2)*exp(-x ^ 2);  
>> root = newton(f,df,x0,tol,maxiter);
```



# Patologia 1: chute inicial “longe” da raiz



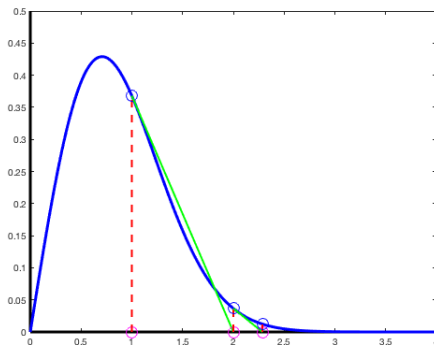
```
>> x0 = 2.0; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) x*exp(-x ^ 2);  
>> df = @(x) exp(-x ^ 2) - 2*(x^ 2)*exp(-x ^ 2);  
>> root = newton(f,df,x0,tol,maxiter);
```



# Patologia 1: chute inicial “longe” da raiz



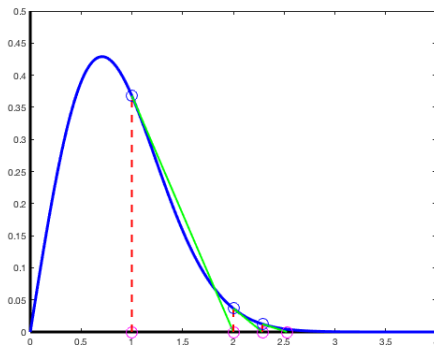
```
>> x0 = 2.0; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) x*exp(-x ^ 2);  
>> df = @(x) exp(-x ^ 2) - 2*(x^ 2)*exp(-x ^ 2);  
>> root = newton(f,df,x0,tol,maxiter);
```



# Patologia 1: chute inicial “longe” da raiz



```
>> x0 = 2.0; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) x*exp(-x ^ 2);  
>> df = @(x) exp(-x ^ 2) - 2*(x^ 2)*exp(-x ^ 2);  
>> root = newton(f,df,x0,tol,maxiter);
```

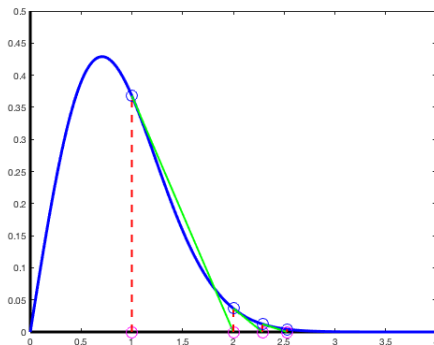




# Patologia 1: chute inicial “longe” da raiz



```
>> x0 = 2.0; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) x*exp(-x ^ 2);  
>> df = @(x) exp(-x ^ 2) - 2*(x^ 2)*exp(-x ^ 2);  
>> root = newton(f,df,x0,tol,maxiter);
```



## Patologia 2: encontrar um extremo local



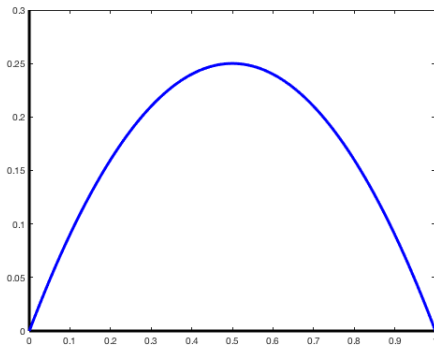
```
>> x0 = 0.5; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) x*(1-x);  
>> df = @(x) 1 - 2*x;  
>> root = newton(f,df,x0,tol,maxiter);
```



## Patologia 2: encontrar um extremo local



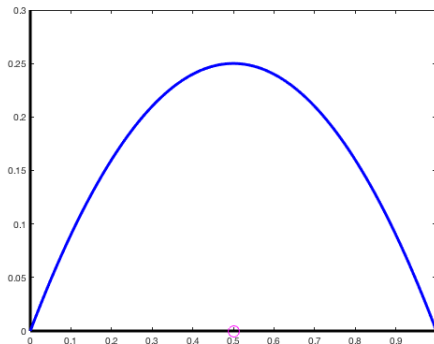
```
>> x0 = 0.5; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) x*(1-x);  
>> df = @(x) 1 - 2*x;  
>> root = newton(f,df,x0,tol,maxiter);
```



## Patologia 2: encontrar um extremo local



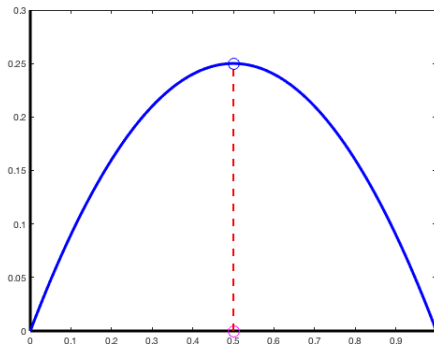
```
>> x0 = 0.5; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) x*(1-x);  
>> df = @(x) 1 - 2*x;  
>> root = newton(f,df,x0,tol,maxiter);
```



## Patologia 2: encontrar um extremo local



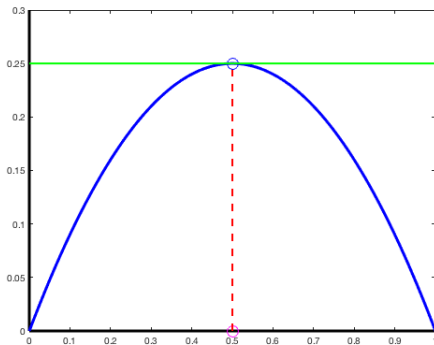
```
>> x0 = 0.5; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) x*(1-x);  
>> df = @(x) 1 - 2*x;  
>> root = newton(f,df,x0,tol,maxiter);
```



## Patologia 2: encontrar um extremo local



```
>> x0 = 0.5; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) x*(1-x);  
>> df = @(x) 1 - 2*x;  
>> root = newton(f,df,x0,tol,maxiter);
```



## Patologia 3: ficar “preso” numa “região oscilatória”



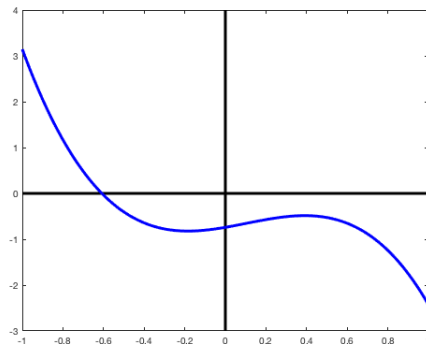
```
>> x0 = 5/9; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) - 0.74 + 0.765*x + 1.1*x^2 - 3.55*x^3;  
>> df = @(x) 0.765 + 2.2*x - 3*3.55*x^2  
>> root = newton(f,df,x0,tol,maxiter);
```



## Patologia 3: ficar “preso” numa “região oscilatória”



```
>> x0 = 5/9; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) - 0.74 + 0.765*x + 1.1*x^2 - 3.55*x^3;  
>> df = @(x) 0.765 + 2.2*x - 3*3.55*x^2  
>> root = newton(f,df,x0,tol,maxiter);
```

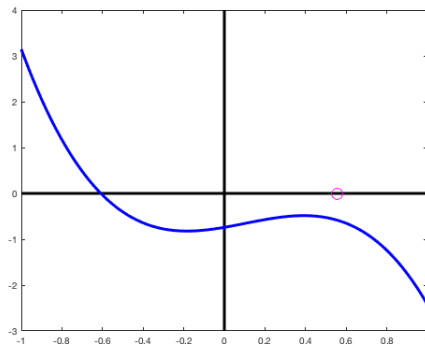




## Patologia 3: ficar “preso” numa “região oscilatória”



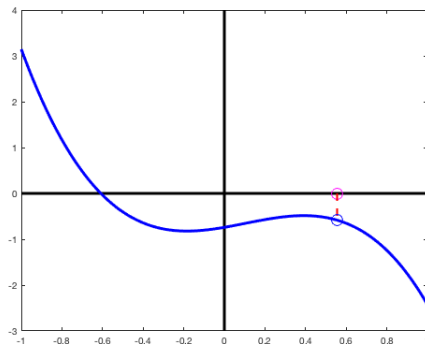
```
>> x0 = 5/9; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) - 0.74 + 0.765*x + 1.1*x^2 - 3.55*x^3;  
>> df = @(x) 0.765 + 2.2*x - 3*3.55*x^2  
>> root = newton(f,df,x0,tol,maxiter);
```



## Patologia 3: ficar “preso” numa “região oscilatória”



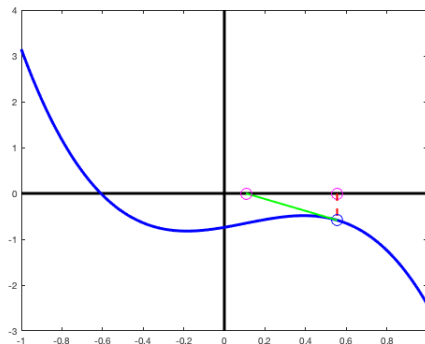
```
>> x0 = 5/9; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) - 0.74 + 0.765*x + 1.1*x^2 - 3.55*x^3;  
>> df = @(x) 0.765 + 2.2*x - 3*3.55*x^2  
>> root = newton(f,df,x0,tol,maxiter);
```



## Patologia 3: ficar “preso” numa “região oscilatória”



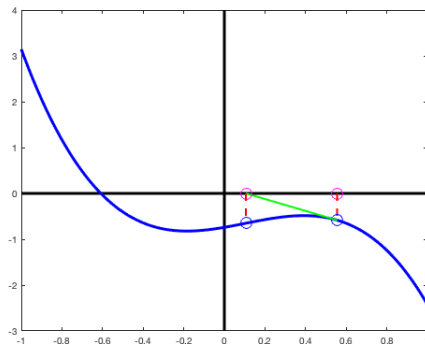
```
>> x0 = 5/9; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) - 0.74 + 0.765*x + 1.1*x^2 - 3.55*x^3;  
>> df = @(x) 0.765 + 2.2*x - 3*3.55*x^2  
>> root = newton(f,df,x0,tol,maxiter);
```



## Patologia 3: ficar “preso” numa “região oscilatória”



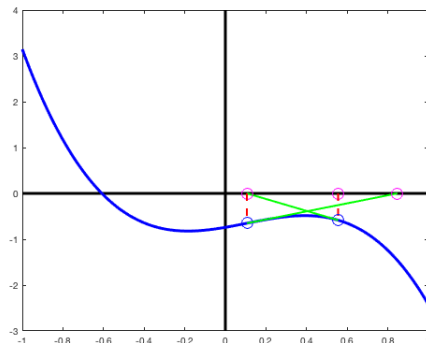
```
>> x0 = 5/9; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) - 0.74 + 0.765*x + 1.1*x^2 - 3.55*x^3;  
>> df = @(x) 0.765 + 2.2*x - 3*3.55*x^2  
>> root = newton(f,df,x0,tol,maxiter);
```



## Patologia 3: ficar “preso” numa “região oscilatória”



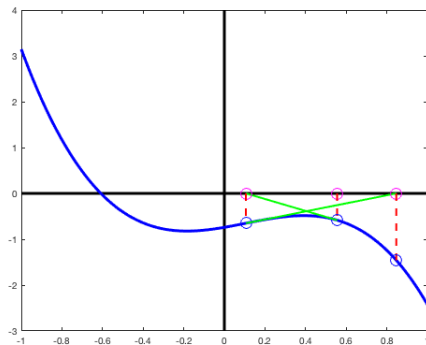
```
>> x0 = 5/9; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) - 0.74 + 0.765*x + 1.1*x^2 - 3.55*x^3;  
>> df = @(x) 0.765 + 2.2*x - 3*3.55*x^2  
>> root = newton(f,df,x0,tol,maxiter);
```



## Patologia 3: ficar “preso” numa “região oscilatória”



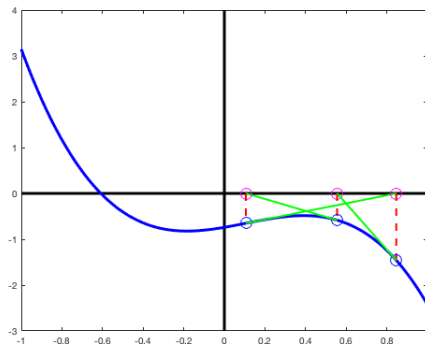
```
>> x0 = 5/9; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) - 0.74 + 0.765*x + 1.1*x^2 - 3.55*x^3;  
>> df = @(x) 0.765 + 2.2*x - 3*3.55*x^2  
>> root = newton(f,df,x0,tol,maxiter);
```



## Patologia 3: ficar “preso” numa “região oscilatória”



```
>> x0 = 5/9; tol = 1.0e-6; maxiter = 20;  
>> f = @(x) - 0.74 + 0.765*x + 1.1*x^2 - 3.55*x^3;  
>> df = @(x) 0.765 + 2.2*x - 3*3.55*x^2  
>> root = newton(f,df,x0,tol,maxiter);
```



# Fundamentação teórica

## Teorema (convergência do método de Newton)

Seja  $f \in C^2[a, b]$ , onde  $[a, b]$  é um intervalo que contém uma raiz de  $f$ , i.e., existe  $x^* \in [a, b]$  tal que  $f(x^*) = 0$ .

Se  $f'(x^*) \neq 0$ , então:

1. existe  $\delta > 0$  tal que para todo  $x_0 \in (x^* - \delta, x^* + \delta) \subset [a, b]$

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

é uma sequência que converge para  $x^*$ , i.e.,  $x_n \rightarrow x^*$

2. a convergência é quadrática, i.e., para algum  $M > 0$  tem-se

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^2} = M$$





# Demonstração do teorema

## Parte 1 (convergência):



# Demonstração do teorema

## Parte 1 (convergência):

O método de Newton é uma iteração de ponto fixo

$$x_{n+1} = \underbrace{x_n - f(x_n)/f'(x_n)}_{g(x_n)}$$



# Demonstração do teorema

## Parte 1 (convergência):

O método de Newton é uma iteração de ponto fixo

$$x_{n+1} = \underbrace{x_n - f(x_n)/f'(x_n)}_{g(x_n)}$$

Alguns fatos:



# Demonstração do teorema

## Parte 1 (convergência):

O método de Newton é uma iteração de ponto fixo

$$x_{n+1} = \underbrace{x_n - f(x_n)/f'(x_n)}_{g(x_n)}$$

Alguns fatos:

- $g'(x) = f(x) f''(x)/f'(x)^2$



# Demonstração do teorema

## Parte 1 (convergência):

O método de Newton é uma iteração de ponto fixo

$$x_{n+1} = \underbrace{x_n - f(x_n)/f'(x_n)}_{g(x_n)}$$

Alguns fatos:

- $g'(x) = f(x) f''(x) / f'(x)^2$
- $f \in C^2[a, b] \implies f, f', f'' \in C[a, b] \implies g' \in C[a, b]$



# Demonstração do teorema

## Parte 1 (convergência):

O método de Newton é uma iteração de ponto fixo

$$x_{n+1} = \underbrace{x_n - f(x_n)/f'(x_n)}_{g(x_n)}$$

Alguns fatos:

- $g'(x) = f(x) f''(x)/f'(x)^2$
- $f \in C^2[a, b] \implies f, f', f'' \in C[a, b] \implies g' \in C[a, b]$
- $f(x^*) = 0 \implies g'(x^*) = 0$



# Demonstração do teorema

## Parte 1 (convergência):

O método de Newton é uma iteração de ponto fixo

$$x_{n+1} = \underbrace{x_n - f(x_n)/f'(x_n)}_{g(x_n)}$$

Alguns fatos:

- $g'(x) = f(x) f''(x)/f'(x)^2$
- $f \in C^2[a, b] \implies f, f', f'' \in C[a, b] \implies g' \in C[a, b]$
- $f(x^*) = 0 \implies g'(x^*) = 0$
- Existe  $(x^* - \delta, x^* + \delta) \subset [a, b]$  tal que  $|g'(x)| < 1$



# Demonstração do teorema

## Parte 1 (convergência):

O método de Newton é uma iteração de ponto fixo

$$x_{n+1} = \underbrace{x_n - f(x_n)/f'(x_n)}_{g(x_n)}$$

Alguns fatos:

- $g'(x) = f(x) f''(x)/f'(x)^2$
- $f \in C^2[a, b] \implies f, f', f'' \in C[a, b] \implies g' \in C[a, b]$
- $f(x^*) = 0 \implies g'(x^*) = 0$
- Existe  $(x^* - \delta, x^* + \delta) \subset [a, b]$  tal que  $|g'(x)| < 1$

Teorema de ponto fixo  $\implies x_{n+1} = g(x_n)$  converge em  $(x^* - \delta, x^* + \delta)$





# Demonstração do teorema

## Parte 1 (convergência):

O método de Newton é uma iteração de ponto fixo

$$x_{n+1} = \underbrace{x_n - f(x_n)/f'(x_n)}_{g(x_n)}$$

Alguns fatos:

- $g'(x) = f(x) f''(x)/f'(x)^2$
- $f \in C^2[a, b] \implies f, f', f'' \in C[a, b] \implies g' \in C[a, b]$
- $f(x^*) = 0 \implies g'(x^*) = 0$
- Existe  $(x^* - \delta, x^* + \delta) \subset [a, b]$  tal que  $|g'(x)| < 1$

Teorema de ponto fixo  $\implies x_{n+1} = g(x_n)$  converge em  $(x^* - \delta, x^* + \delta)$

**O método de Newton converge para qualquer  $x_0$   
“suficientemente próximo” de  $x^*$ .**



# Demonstração do teorema

## Parte 2 (velocidade de convergência):



# Demonstração do teorema

## Parte 2 (velocidade de convergência):

$$f(x^*) = 0 \iff f(x_n + x^* - x_n) = 0$$



# Demonstração do teorema

## Parte 2 (velocidade de convergência):

$$f(x^*) = 0 \iff f(x_n + \underbrace{x^* - x_n}_{\Delta x_n}) = 0$$



# Demonstração do teorema

## Parte 2 (velocidade de convergência):

$$f(x^*) = 0 \iff f(x_n + \underbrace{x^* - x_n}_{\Delta x_n}) = 0$$

### Teorema de Taylor:

$$f(x_n) + f'(x_n)(x^* - x_n) + \frac{1}{2} f''(\xi)(x^* - x_n)^2 = 0, \quad \xi \in (x^*, x_n)$$



# Demonstração do teorema

## Parte 2 (velocidade de convergência):

$$f(x^*) = 0 \iff f(x_n + \underbrace{x^* - x_n}_{\Delta x_n}) = 0$$

### Teorema de Taylor:

$$f(x_n) + f'(x_n)(x^* - x_n) + \frac{1}{2} f''(\xi)(x^* - x_n)^2 = 0, \quad \xi \in (x^*, x_n)$$

$$\implies \frac{f(x_n)}{f'(x_n)} + x^* - x_n = -\frac{1}{2} \frac{f''(\xi)}{f'(x_n)} (x^* - x_n)^2$$



# Demonstração do teorema

## Parte 2 (velocidade de convergência):

$$f(x^*) = 0 \iff f(x_n + \underbrace{x^* - x_n}_{\Delta x_n}) = 0$$

### Teorema de Taylor:

$$f(x_n) + f'(x_n)(x^* - x_n) + \frac{1}{2} f''(\xi)(x^* - x_n)^2 = 0, \quad \xi \in (x^*, x_n)$$

$$\implies \frac{f(x_n)}{f'(x_n)} + x^* - x_n = -\frac{1}{2} \frac{f''(\xi)}{f'(x_n)} (x^* - x_n)^2$$

$$\implies x^* - \underbrace{\left( x_n - \frac{f(x_n)}{f'(x_n)} \right)}_{x_{n+1}} = -\frac{1}{2} \frac{f''(\xi)}{f'(x_n)} (x^* - x_n)^2$$



# Demonstração do teorema

## Parte 2 (velocidade de convergência):

$$f(x^*) = 0 \iff f(x_n + \underbrace{x^* - x_n}_{\Delta x_n}) = 0$$

### Teorema de Taylor:

$$f(x_n) + f'(x_n)(x^* - x_n) + \frac{1}{2} f''(\xi)(x^* - x_n)^2 = 0, \quad \xi \in (x^*, x_n)$$

$$\implies \frac{f(x_n)}{f'(x_n)} + x^* - x_n = -\frac{1}{2} \frac{f''(\xi)}{f'(x_n)} (x^* - x_n)^2$$

$$\implies x^* - \underbrace{\left(x_n - \frac{f(x_n)}{f'(x_n)}\right)}_{x_{n+1}} = -\frac{1}{2} \frac{f''(\xi)}{f'(x_n)} (x^* - x_n)^2$$

$$\implies |x^* - x_{n+1}| = \frac{1}{2} \left| \frac{f''(\xi)}{f'(x_n)} \right| |x^* - x_n|^2$$





# Demonstração do teorema

$$|x^* - x_{n+1}| = \frac{1}{2} \left| \frac{f''(\xi)}{f'(x_n)} \right| |x^* - x_n|^2$$



# Demonstração do teorema

$$|x^* - x_{n+1}| = \frac{1}{2} \left| \frac{f''(\xi)}{f'(x_n)} \right| |x^* - x_n|^2$$

erro da iteração  $n+1 \sim$  quadrado do erro da iteração  $n$



# Demonstração do teorema

$$|x^* - x_{n+1}| = \frac{1}{2} \left| \frac{f''(\xi)}{f'(x_n)} \right| |x^* - x_n|^2$$

erro da iteração  $n+1 \sim$  quadrado do erro da iteração  $n$

Definindo  $M = \frac{1}{2} \left| \frac{f''(\xi)}{f'(x^*)} \right| > 0$ , temos que

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^2} = M$$



# Demonstração do teorema

$$|x^* - x_{n+1}| = \frac{1}{2} \left| \frac{f''(\xi)}{f'(x_n)} \right| |x^* - x_n|^2$$

erro da iteração  $n+1 \sim$  quadrado do erro da iteração  $n$

Definindo  $M = \frac{1}{2} \left| \frac{f''(\xi)}{f'(x^*)} \right| > 0$ , temos que

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^2} = M$$

**O método de Newton converge quadraticamente!**



# Características do método de Newton

- ☺ Simples e fácil de implementar
- ☺ Fácil de generalizar para várias variáveis
- ☺ Convergência rápida
- ☹ Requer que  $f$  seja diferenciável
- ☹ O cálculo de  $f'$  pode ser computacionalmente caro
- ☹ Convergência não garantida e dependente do chute inicial



## Como citar esse material?

A. Cunha, *Método de Newton*, Universidade do Estado do Rio de Janeiro – UERJ, 2020.

Essas notas de aula podem ser compartilhadas nos termos da licença Creative Commons BY-NC-ND 3.0, com propósitos exclusivamente educacionais.

