Zeros de Funções

J. A. Salvador

Introdução

Muitos problemas que aparecem na natrureza que consistem em resolver uma equação do tipo $f(x)=0$, isto é, encontrar um ponto x_0 tal $f(x_0)=0$.
As equações de primeiro e segundo graus são fáceis de obter solução.

```
> r[1] := solve( a*x + b = 0, x);
```

$$r_1 := -\frac{b}{a}$$

```
> r[1,2] := solve( a*x^2 + b*x + c =0, x);
```

$$r_{1,2} := \frac{-b+\sqrt{b^2-4\,a\,c}}{2\,a},\ \frac{-b-\sqrt{b^2-4\,a\,c}}{2\,a}$$

```
>
```

Ex. Até uma função $f(x)=x^3-2\,x$ do terceiro grau que possui 3 zeros: $x_0=0$, $x_1=\sqrt{2}$ e $x_2=-\sqrt{2}$ não é difícil de encontrá-los.

```
> r[1,2,3] := solve( x^3 - 2*x = 0, x);
```

$$r_{1,2,3} := 0, \sqrt{2}, -\sqrt{2}$$

ou mesmo uma equação geral do terceiro grau possui uma fórmula que nos dá as suas raízes.

```
> r[1,2,3] := solve( a*x^3 + b*x^2 + c*x + d =0, x);
```

$$r_{1,2,3} :=$$

$$\frac{1}{6}\frac{(36\,b\,c\,a-108\,d\,a^2-8\,b^3+12\sqrt{3}\sqrt{4\,a\,c^3-c^2\,b^2-18\,b\,c\,a\,d+27\,d^2\,a^2+4\,d\,b^3}\,a)^{(1/3)}}{a}$$

$$-\frac{2}{3}\frac{3\,a\,c-b^2}{a\,(36\,b\,c\,a-108\,d\,a^2-8\,b^3+12\sqrt{3}\sqrt{4\,a\,c^3-c^2\,b^2-18\,b\,c\,a\,d+27\,d^2\,a^2+4\,d\,b^3}\,a)^{(1/3)}}$$

$$-\frac{1}{3}\frac{b}{a},\ -\frac{1}{12}$$

$$\frac{\left(36\, b\, c\, a - 108\, d\, a^{2} - 8\, b^{3} + 12\, \sqrt{3}\, \sqrt{4\, a\, c^{3} - c^{2}\, b^{2} - 18\, b\, c\, a\, d + 27\, d^{2}\, a^{2} + 4\, d\, b^{3}\, a}\right)^{(1/3)}}{a} +$$

$$\frac{1}{3}$$

$$\frac{3\, a\, c - b^{2}}{a \left(36\, b\, c\, a - 108\, d\, a^{2} - 8\, b^{3} + 12\, \sqrt{3}\, \sqrt{4\, a\, c^{3} - c^{2}\, b^{2} - 18\, b\, c\, a\, d + 27\, d^{2}\, a^{2} + 4\, d\, b^{3}\, a}\right)^{(1/3)}}$$

$$-\frac{1}{3}\frac{b}{a} + \frac{1}{2} I \sqrt{3} \Bigg($$

$$\frac{1}{6}\frac{\left(36\, b\, c\, a - 108\, d\, a^{2} - 8\, b^{3} + 12\, \sqrt{3}\, \sqrt{4\, a\, c^{3} - c^{2}\, b^{2} - 18\, b\, c\, a\, d + 27\, d^{2}\, a^{2} + 4\, d\, b^{3}\, a}\right)^{(1/3)}}{a}$$

$$+\frac{2}{3}$$

$$\frac{3\, a\, c - b^{2}}{a \left(36\, b\, c\, a - 108\, d\, a^{2} - 8\, b^{3} + 12\, \sqrt{3}\, \sqrt{4\, a\, c^{3} - c^{2}\, b^{2} - 18\, b\, c\, a\, d + 27\, d^{2}\, a^{2} + 4\, d\, b^{3}\, a}\right)^{(1/3)}}\Bigg)$$

$$, -\frac{1}{12}$$

$$\frac{\left(36\, b\, c\, a - 108\, d\, a^{2} - 8\, b^{3} + 12\, \sqrt{3}\, \sqrt{4\, a\, c^{3} - c^{2}\, b^{2} - 18\, b\, c\, a\, d + 27\, d^{2}\, a^{2} + 4\, d\, b^{3}\, a}\right)^{(1/3)}}{a} +$$

$$\frac{1}{3}$$

$$\frac{3\, a\, c - b^{2}}{a \left(36\, b\, c\, a - 108\, d\, a^{2} - 8\, b^{3} + 12\, \sqrt{3}\, \sqrt{4\, a\, c^{3} - c^{2}\, b^{2} - 18\, b\, c\, a\, d + 27\, d^{2}\, a^{2} + 4\, d\, b^{3}\, a}\right)^{(1/3)}}$$

$$-\frac{1}{3}\frac{b}{a} - \frac{1}{2} I \sqrt{3} \Bigg($$

$$\frac{1}{6}\frac{\left(36\, b\, c\, a - 108\, d\, a^{2} - 8\, b^{3} + 12\, \sqrt{3}\, \sqrt{4\, a\, c^{3} - c^{2}\, b^{2} - 18\, b\, c\, a\, d + 27\, d^{2}\, a^{2} + 4\, d\, b^{3}\, a}\right)^{(1/3)}}{a}$$

$$+\frac{2}{3}$$

$$\frac{3\, a\, c - b^{2}}{a \left(36\, b\, c\, a - 108\, d\, a^{2} - 8\, b^{3} + 12\, \sqrt{3}\, \sqrt{4\, a\, c^{3} - c^{2}\, b^{2} - 18\, b\, c\, a\, d + 27\, d^{2}\, a^{2} + 4\, d\, b^{3}\, a}\right)^{(1/3)}}\Bigg)$$

Como encontrar os zeros de uma equação ou as raízes de uma equação qualquer? Ou de uma função menos conhecida e com uma expressão mais complicada?

Neste caso, uma idéia seria localizarmos os intervalos que contém cada uma das raízes e isolá-los. Em seguida refiná-los cada vez mais.

```
>
```

Ex.

```
> f := x -> x^3 - 9*x +3: 'f(x)'= f(x);
```

$$f(x) = x^3 - 9x + 3$$

Verificamos o sinal de f em alguns intervalos.

```
> for i from -4 to 4 by 1 do print( [i, f(i)]) od;
```

$$[-4, -25]$$
$$[-3, 3]$$
$$[-2, 13]$$
$$[-1, 11]$$
$$[0, 3]$$
$$[1, -5]$$
$$[2, -7]$$
$$[3, 3]$$
$$[4, 31]$$

```
>
```

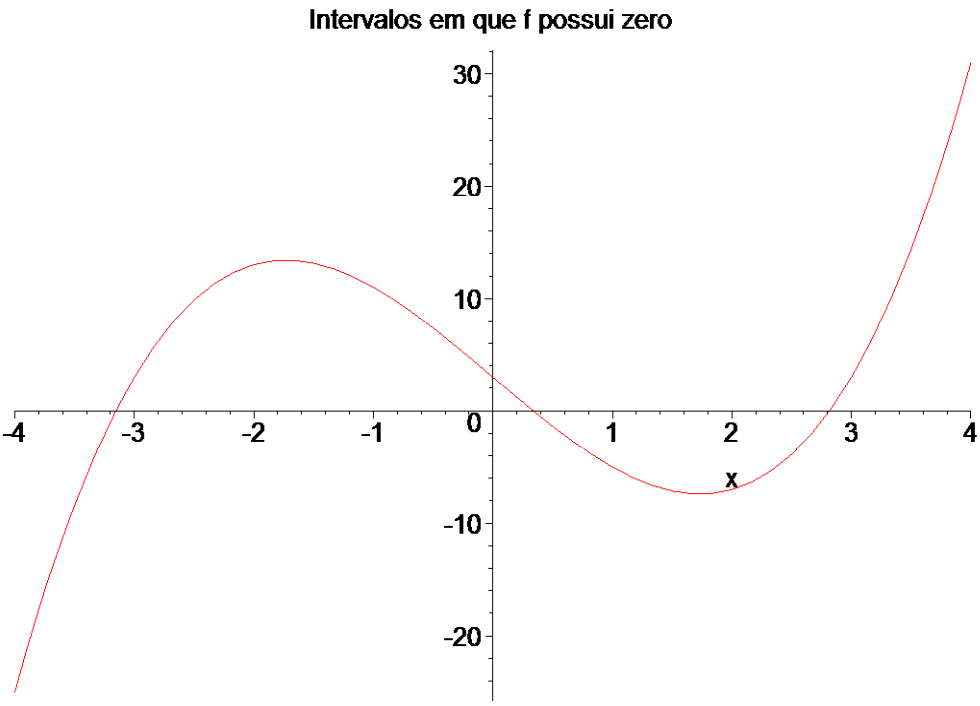
Como f é uma função contínua para todo valor real x, observamos que ela contém pelo menos um zero em cada intervalo em que ela muda de sinal, que são os intervalos [-4, -3], [0, 1] e [2, 3].

Para descobrir um intervalo que contém as raízes de uma função contínua usamos o Teorema do Valor Intermediário do Cálculo Diferencial e Integral, traduzido na seguinte forma:

TVI: Se $f(x)$ é uma função contínua num intervalo $[a, b]$, e se $f(a)\,f(b) < 0$ então existe pelo menos um ponto $x = \xi$ entre a e b que é zero de $f(x)$, isto é, tal que $f(\xi) = 0$.
[Ver dem. Guidorizzi 2001]

Vejamos o gráfico de $f(x) = x^3 - 9x + 3$.

```
> plot(f(x), x=-4..4, title = `Intervalos em que f possui zero`);
```




```
>
```

Observe que

```
> ' f(-4) * f(-3)' = f(-4) * f(-3); 'f(0) * f(1)' = f(0) * f(1);
  'f(2) * f(3)' = f(2) * f(3);
```

$$f(-4)\,f(-3) = -75$$

$$f(0)\,f(1) = -15$$

$$f(2)\,f(3) = -21$$

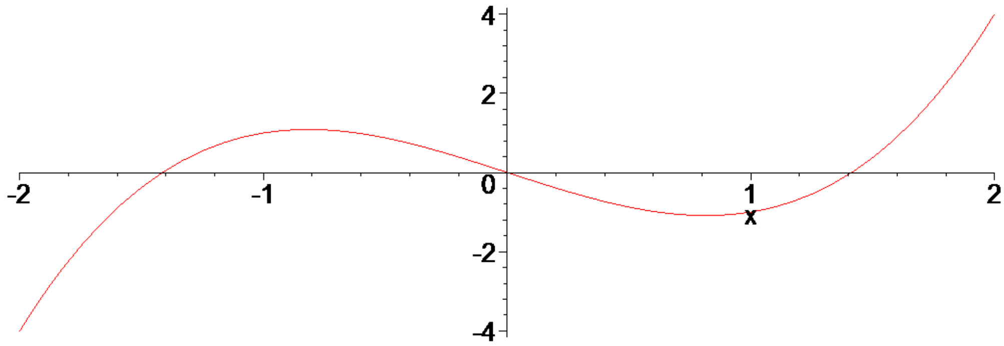
É fácil de verificar que nos intervalo em que f muda de sinal ela possui um zero.

Seja agora $f(x) = x^3 - 2\,x$:

```
> f := x -> x^3 - 2*x;
```

$$f := x \to x^3 - 2\,x$$

```
> plot( f(x) , x=-2..2);
```



Observe que

```
>  'f(1)'= f(1);  'f(2)'= f(2);
```

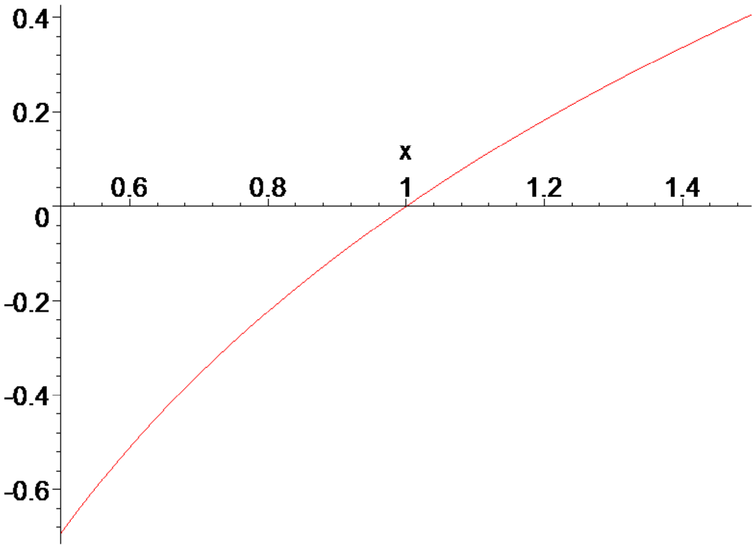
$$f(1) = -1$$

$$f(2) = 4$$

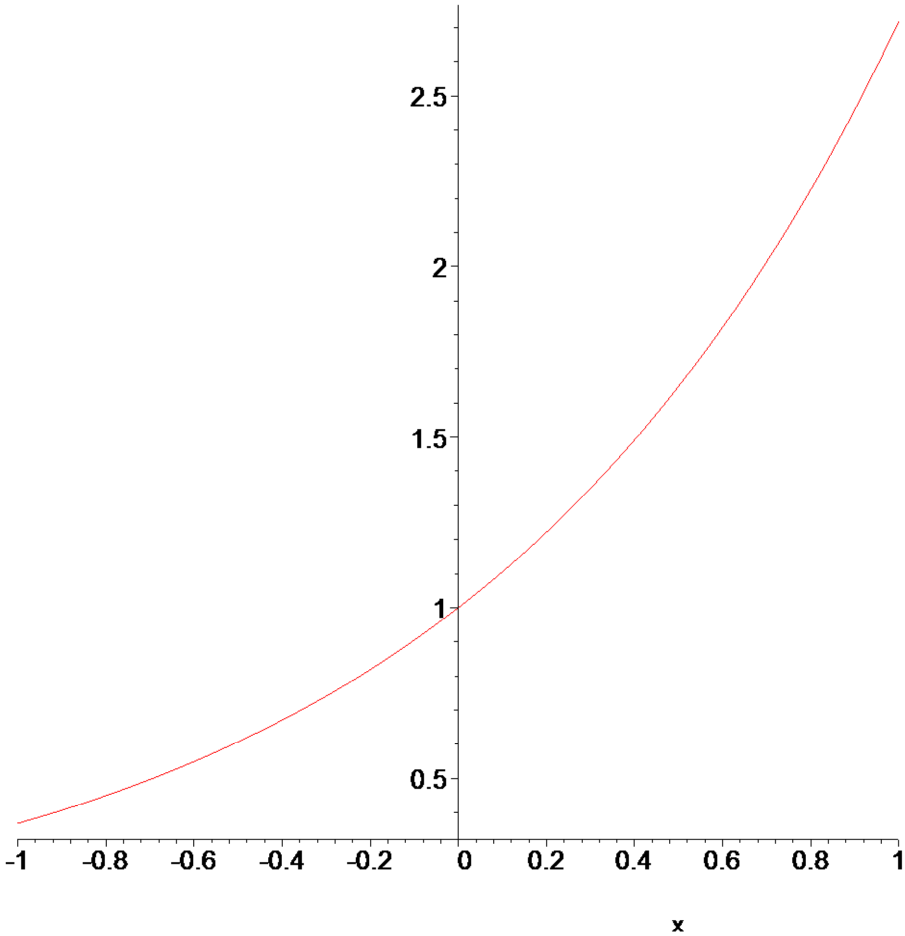
Observe que f muda de sinal no intervalo [1, 2], ou seja, $f(1)\,f(2) < 0$, logo existe um zero de f entre $a = 1$ e $b = 2$.

Zero de f(x)
Se f : [a, b] -> R é uma função dada, um ponto x ε [a, b], é um ZERO (ou RAIZ) de f se f(x) =0.

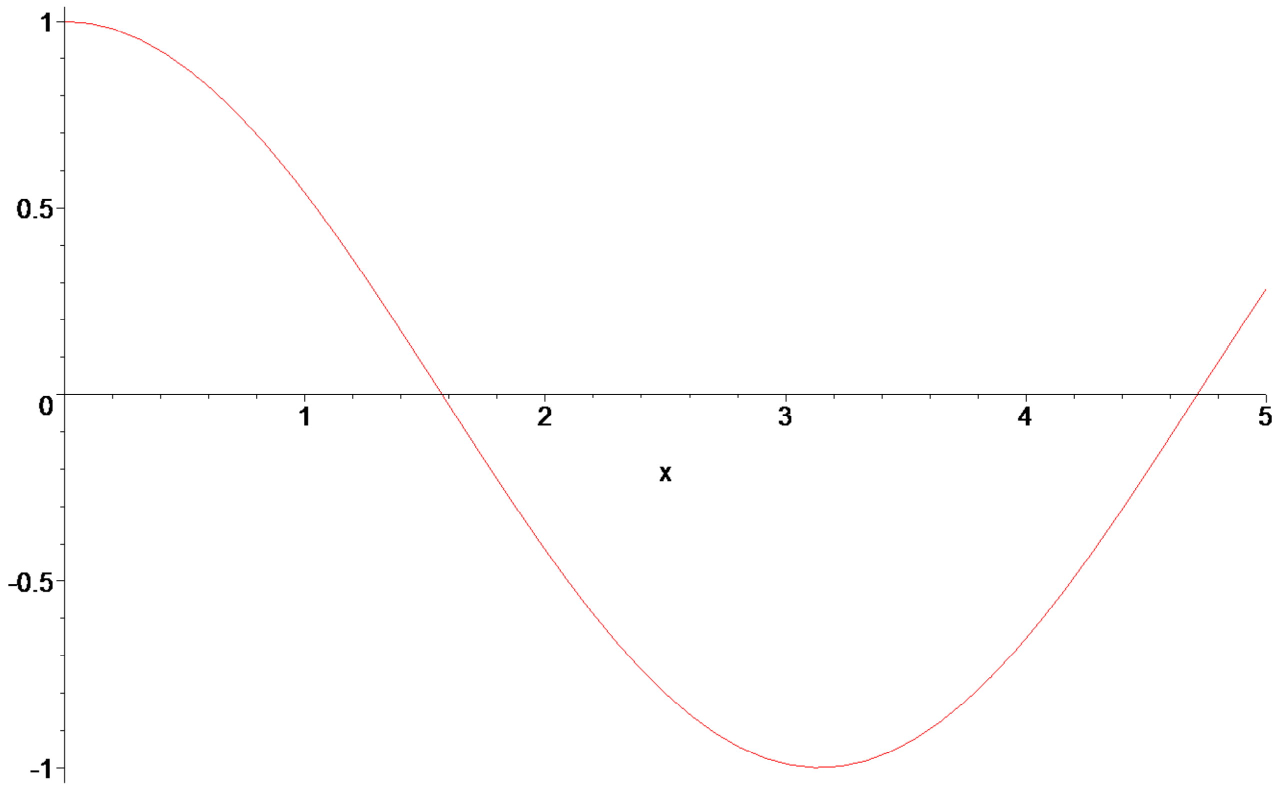
```
> plot(ln(x), x=0.5..1.5);
```




```
> plot( exp(x), x=-1..1);
```




```
> plot( cos(x), x=0..5);
```




```
> [0, cos(0)]; [1, evalf(cos(1))]; [2, evalf(cos(2))]; [3,
  evalf(cos(3))]; [4, evalf(cos(4))]; [5, evalf(cos(5))];
```

[0, 1]

[1, 0.5403023059]

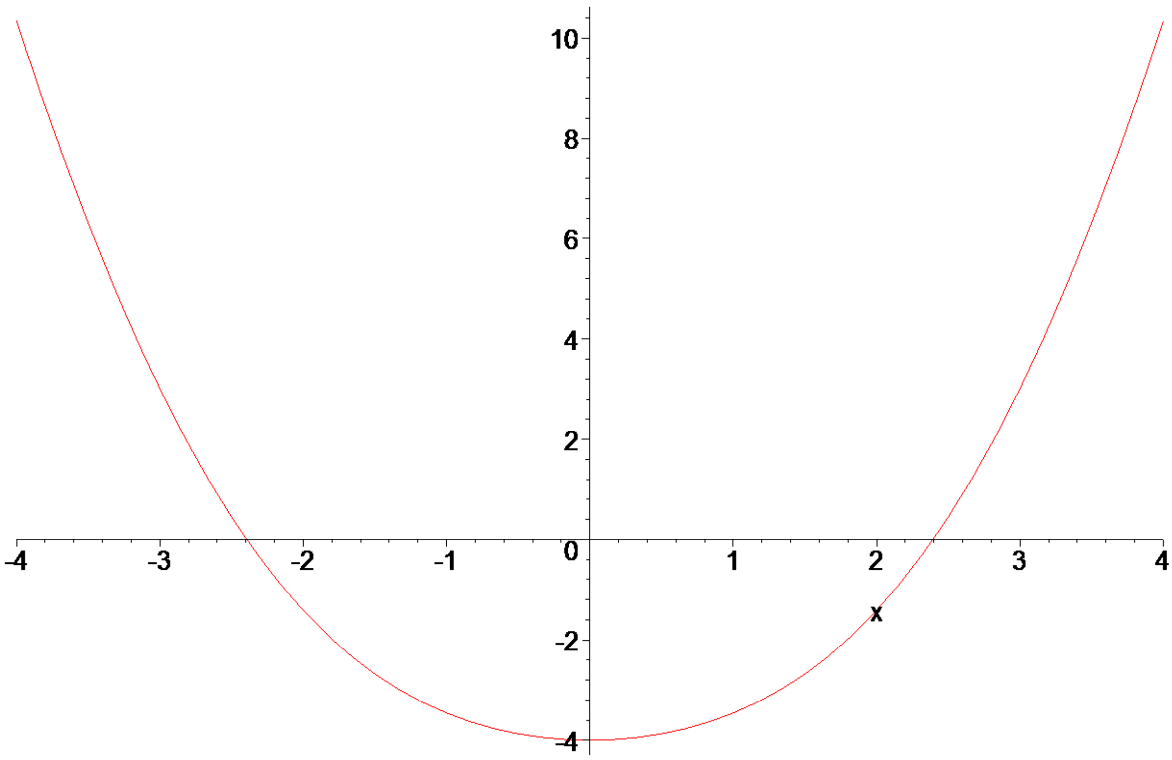
[2, -0.4161468365]

[3, -0.9899924966]

[4, -0.6536436209]

[5, 0.2836621855]

```
> plot( cos(x) + x^2 - 5, x=-4..4);
```




```
> solve(cos(x) + x^2 + 5 = 0, x);
```

$$\mathrm{RootOf}(\cos(\_Z) + \_Z^2 + 5, label = \_L6)$$

```
>
```

Além disso, se nas condições anteriores a derivada de $f(x)$ existir no intervalo (a, b) e preservar o sinal em (a, b), então ele é único aí.

Caso em que $0 < f(a)\, f(b)$ podemos ter várias situações.

Uma análise gráfica de $f(x)$ que pode ser feito com alguns softwares disponíveis e é aconselhável para se obter boa aproximação inicial para a raíz.

No exemplo acima, usando o Maple, basta clicar no ponto onde f intercepta o eixo ox no intervalo $[1, 2]$ e obtermos $x_0 = 1.41$, por exemplo.

O modo de refinamento do intervalo para obter a raíz é que nos dá os vários métodos iterativos.

Um método iterativo consiste em uma seqüência de instruções que são executadas passo a passo, algumas delas repetidas em ciclos.

A execução de cada ciclo é chamada de iteração.

Cada iteração utiliza resultados das iterações anteriores e efetua determinados testes para ver se atinge um resultado próximo o suficiente do resultado esperado. Os métodos iterativos são métodos que nos dão aproximação para a solução exata.

Como testar para verificar se a solução está próxima da solução exata desejada se não a conhecemos?

O que entendemos por solução aproximada?

Uma interpretação para raíz aproximada é dizer que x_k é uma raiz aproximada de ξ com precisão ε se

```
> abs( x[k] - xi) < epsilon; # teste 1
```

$$|x_k - \xi| < \varepsilon$$

ou

```
> f(x[k]) < epsilon;  # teste 2
```

$$f(x_k) < \varepsilon$$

```
>
```

O problema é como efetuar o teste1 se não conhecemos a solução exata ξ ?

Se podermos reduzir o intervalo que contém a raíz até conseguir um intervalo $[a_k, b_k]$ de modo que

```
> b[k] - a[k] < epsilon;
```

$$b_k - a_k < \varepsilon$$

```
>
```

então para todo x em $[a_k, b_k]$ temos que $|x - \xi| < \varepsilon$. Portanto, qualquer x em $[a_k, b_k]$ pode ser tomado como x_k.

Nem sempre é possível ter os dois testes satisfeitos simultameamente e os métodos numéricos são desenvolvidos de modo a satisfazer pelo menos um dos critérios.

Dependendo da ordem de grandeza dos números é aconselhável também usar o Teste do Erro Relativo,

```
>   abs( f(x[k])/ f(x)) < epsilon;
```

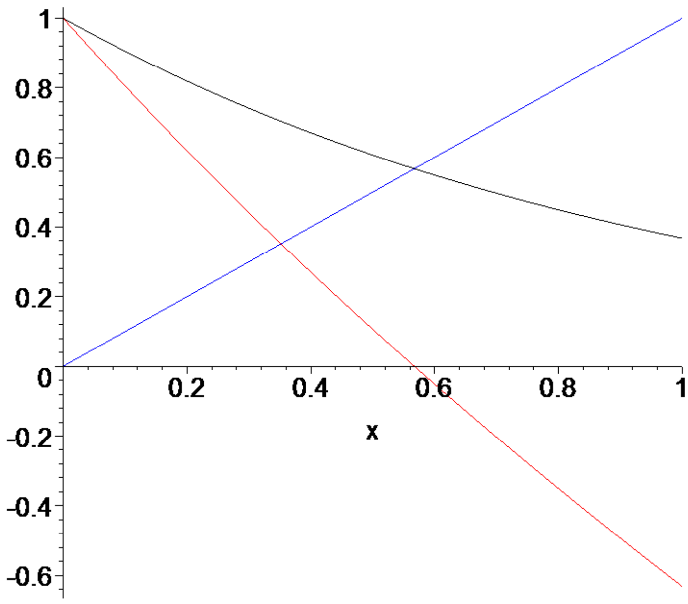
$$\left|\frac{f(x_k)}{f(x)}\right| < \varepsilon$$

para algum x escolhido numa vizinhança de ξ.

Nos programas computacionais pode ser usado um critério de parada para cada método, estipulando um número máximo de iterações para evitar looping devido a erros que podem ocorrer no programa ou devido ao fato do método usado não ser adequado para o problema em questão.

Ex. Ache os zeros de $f(x) = \mathbf{e}^{(-x)} - x$.

```
> plot( [exp(-x) - x, x, exp(-x)], x=0..1, color = [red, blue,
  black]);
```



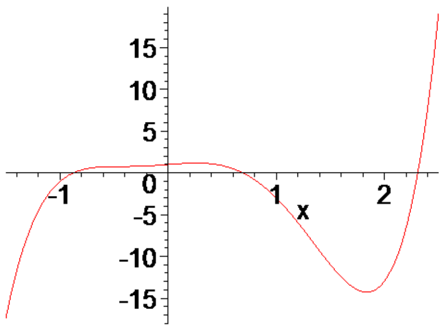

```
> restart;
```

Metodo da bisseção

```
> f :=  x -> 2*x^5-3*x^4-4*x^3+x+1;
```

$$f := x \rightarrow 2\,x^5 - 3\,x^4 - 4\,x^3 + x + 1$$

```
> plot(f(x), x=-1.5..2.5);
```



Observamos que *f* possui uma raíz entre -1 e -0.5;

```
> ' f(-1)' = f(-1); ' f(-0.5)' =f(-0.5);
```

$$f(-1) = -1$$

$$f(-0.5) = 0.75000$$

Dando uma aproximação inicial, como o ponto médio do intervalo onde se encontra a raíz,

```
> x0 := (-1+ (-0.5))/2;
```

$$x0 := -0.7500000000$$

```
> f(x0);
```

$$0.5136718750$$

Testamos

```
> x1 := (-1+x0)/2;
```

$$x1 := -0.8750000000$$

```
> 'f(x1)' = f(x1);
```

$$f(x1) = 0.0203247070$$

```
> x2 := (-1+x1)/2;
```

$$x2 := -0.9375000000$$

```
> x3 := (x1+x2)/2;
```

$$x3 := -0.9062500000$$

```
> 'f(x3)' = f(x3);
```

$$f(x3) = -0.175184071$$

```
> x4 := (x1+x3)/2;
```

$$x4 := -0.8906250000$$

```
> 'f(x4)' = f(x4);
```

$$f(x4) = -0.073102938$$

```
> x3-x1;
```

$$-0.0312500000$$

```
>
```

Problema:

Vamos calcular sem o uso de computadores a $\sqrt{45}$. Este problema equivale a encontrar a raíz ou o zero da função $f(x) = x^2 - 45$.

A raíz da equação $f(x) = x^2 - 45$ pelo método da bisseção pode ser feita do seguinte modo:

```
> restart;
> x := 45:
> n := 1:
> if((x/n+n)/2)^2 - x < 0.001 then Raiz := ((x/n+n)/2) else
  n:=((x/n+n)/2.) fi;
```

$$n := 23.00000000$$

```
>
```

do mesmo modo, podemos encontrar a $\sqrt{98}$.

```
>  n := 2: x := 98:
   while ( ((x/n + n)/2 )^2 - x) > 0.001 do n := ((x/n+n)/2.) od:
  'n'= n;
```

$$n = 9.927704354$$

```
>
```

De um modo geral, podemos construir o seguinte procedimento para o cálculo da raíz quadrada, cuja sintaxe é:

raiz(Número)

```
> restart;
> raiz := proc(x)
>         local n;    n:=1:
>         while ( ((x/n + n)/2 )^2 - x) > 0.001 do n:=((x/n+n)/2.)
  od: n;  end;
```

raiz := **proc**(*x*)

local *n*;

  n := 1;

  while 0.001 < (1 / 2*x / n + 1 / 2*n)^2 − x **do**

$n := 0.5000000000 * x / n + 0.5000000000 * n$

end do;

n

end proc

```
> 'sqrt(45)'= raiz(45);
```

$$\sqrt{45} = 6.709101664$$

```
> 'sqrt(2)'= raiz(2);
```

$$\sqrt{2} = 1.416666666$$

```
> 'sqrt(25)' = raiz(25);
```

$$\sqrt{25} = 5.015247602$$

```
> 'sqrt(121)'= raiz(121);
```

$$\sqrt{121} = 11.00018829$$

```
>
> restart:
> f := x -> x^3-x^2+x-1;
```

$$f := x \to x^3 - x^2 + x - 1$$

```
> a := 0.5; b := 2; # usamos representação decimal
```

$$a := 0.5$$

$$b := 2$$

```
> for i from 1 to 5
  do c := (a+b)/2:
  if f(a)*f(c)> 0 then a := c else b := c fi
  od;
```

$$c := 1.250000000$$

$$c := 0.8750000000$$

$$c := 1.062500000$$

$$c := 0.9687500000$$

$$c := 1.015625000$$

```
>
```

4 - Programa B:

Baseado no sucesso do Programa A faremos um *procedimento Maple*. O novo programa também testará se a função f possui sinais contrários nos extremos do intervalo [*a*,*b*]. O teste é feito através dos comandos if --- then --- ERROR --- fi . Além disso, o programa para quando o erro na aproximação for menor que erro. A sintaxe do procedimento é:

`bissec( função , a , b , precisão ).`

```
> bissec := proc(f, a, b, erro)
```

```
    local aa, bb, cc, k: # definindo variáveis locais
    aa := evalf(a):  bb := evalf(b):
    if f(aa)*f(bb) > 0 then ERROR(`intervalo inválido`) fi:
    for k from 1 do cc := (aa+bb)/2:
    if f(aa)*f(cc) > 0 then aa := cc
     elif f(aa)*f(cc) < 0 then bb := cc
      else break fi:
    if evalf(abs(aa-bb)) < erro then break fi:
     od:  # agora k já é k+1.
  print(`A raiz aproximada após`,k-1,`bissecções :`):
  print(cc):
  end:
```

```
> bissec(cos, 1, 3, .0001);
```

A raiz aproximada após, 14, *bissecções :*

1.570739746

```
> bissec(cos, 0, 1, .0001);
Error, (in bissec) intervalo inválido
```

```
> g := x -> x^3-exp(x);
```

$$g := x \to x^3 - \mathbf{e}^x$$

```
> bissec(g , 1, 4, .001);
```

A raiz aproximada após, 11, *bissecções :*

1.857666016

```
>
```

Método da Posição Falsa

No método da posição falsa para uma função $\mathrm{f}(x)$ contínua num intervalo [a, b] e tal que $\mathrm{f}(a)\,\mathrm{f}(b) < 0$ e supondo que f tenha uma única raiz em (a, b), podemos conseguir uma raiz aproximada x_0 usando as informações de $\mathrm{f}(x)$ nos extremos de cada intervalo disponíveis a cada iteração.

Ex. No caso da função

```
> f := x -> x^3 - 9*x + 1: 'f(x)' = f(x);
```

$$\mathrm{f}(x) = x^3 - 9\,x + 1$$

sabemos que f tem uma raíz no intervalo [0, 1]. De fato;

```
> 'f(0)'= f(0);  'f(1)'= f(1);
```

$$\mathrm{f}(0) = 1$$

$$\mathrm{f}(1) = -7$$

Observamos que

```
> 'abs(f(0))' = abs(f(0)); 'abs(f(1))' = abs(f(1));
```

$$|\mathrm{f}(0)| = 1$$

$$|f(1)| = 7$$

O valor absoluto de f no ponto $x = a = 0$ é menor do que o valor absoluto de f no ponto $x = b = 1$. Logo, é mais provável que o zero de f esteja mais próximo de $a = 0$ do que $b = 1$. Pelo menos isto ocorreria se f fosse linear!

Então, em vez de tomarmos x_0 como a média aritmética de a e b como no método da Bissecção, escolhemos a média ponderada entre a e b com pesos $|f(b)|$ e $|f(a)|$, isto é,

$$x_0 = \frac{a|f(b)| + b|f(a)|}{|f(b)| + |f(a)|} = \frac{a\,f(b) - b\,f(a)}{f(b) - f(a)}$$

No exemplo;

```
> a0 := 0: b0 := 1.:
> x[0] := (a0*abs(f(b0))  + b0* abs(f(a0)))/( abs(f(b0))  +
  abs(f(a0)));
```

$$x_0 := 0.1250000000$$

```
> 'f(x[0])' = f(x[0]);
```

$$f(x_0) = -0.123046875$$

Escolhemos agora o intervalo [a1, b1] = [0, 0.125]

```
> a1 := 0: b1 := 0.125:
> x[1] := (a1*abs(f(b1))  + b1* abs(f(a1)))/( abs(f(b1))  +
  abs(f(a1)));
```

$$x_1 := 0.1113043478$$

```
> 'f(x[1])' = f(x[1]);
```

$$f(x_1) = -0.000360219$$

Escolhemos agora o intervalo $[a_2, b_2]$ = [0.1113043478, 0.125]

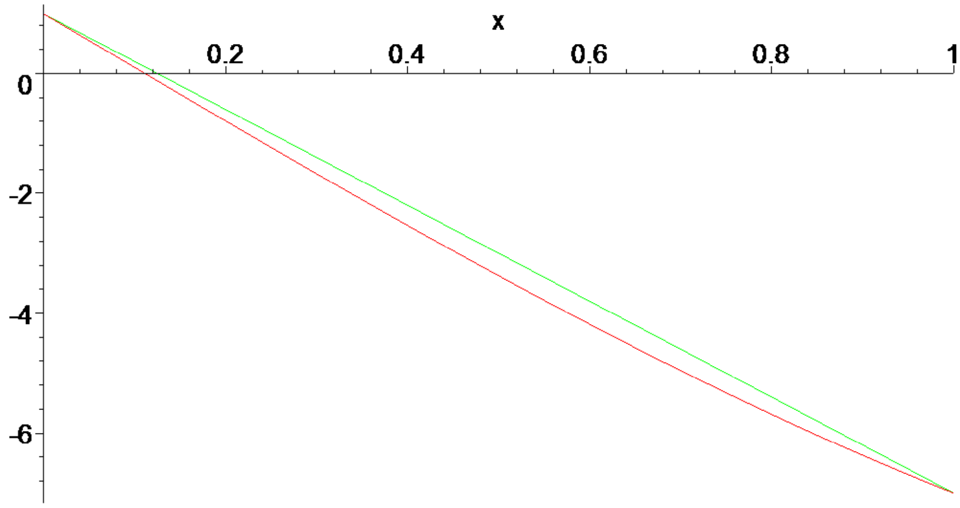
```
> a2 := 0.1113043478: b2 := 0.125:
> x[2] := (a2*abs(f(b2))  + b2* abs(f(a2)))/( abs(f(b2)) +
  abs(f(a2)));
```

$$x_2 := 0.1113443247$$

```
> 'f(x[2])' = f(x[2]);
```

$$f(x_2) = -0.000718524$$

```
> plot( {f(x), [[a0, f(a0)], [b0, f(b0)]]}, x=0..1);
```

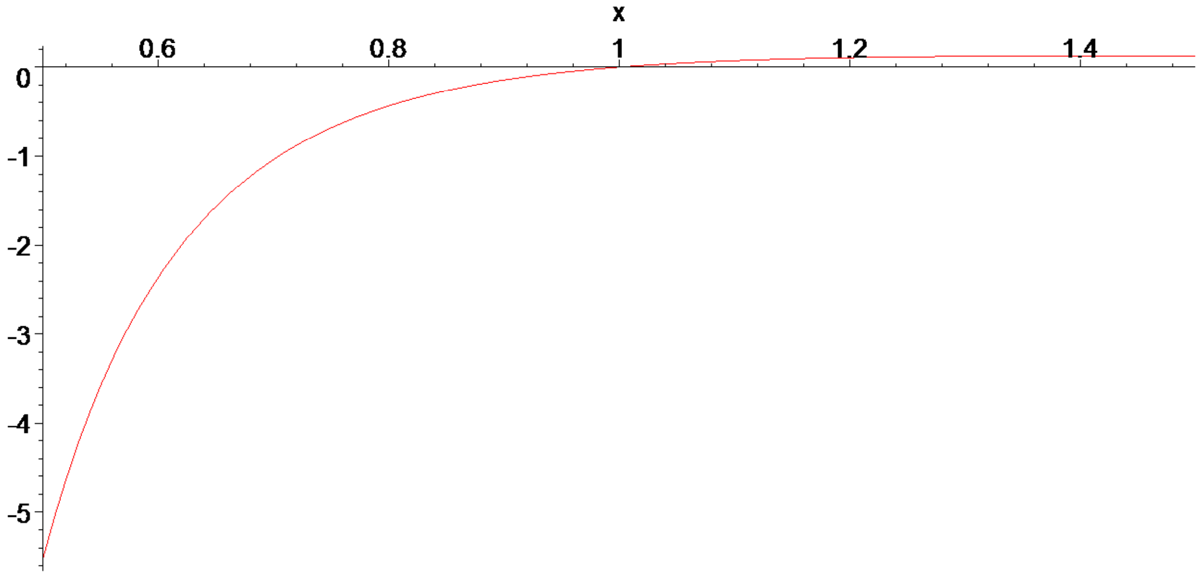


>

Observamos que quando *f* é derivável duas vezes em [a, b] e a derivada segunda de *f*, $\frac{\partial^2}{\partial x\,\partial x}\mathrm{f}(x)$ também não muda de sinal neste intervalo, é bastante intuitivo verificar a convergência graficamente.
Observamos também que o método da posição falsa pode chegar a raiz aproximada no qual $|\mathrm{f}(x_k)| < \varepsilon$ sem que o intervalo $[a_k, b_k]$ seja suficientemente pequeno.

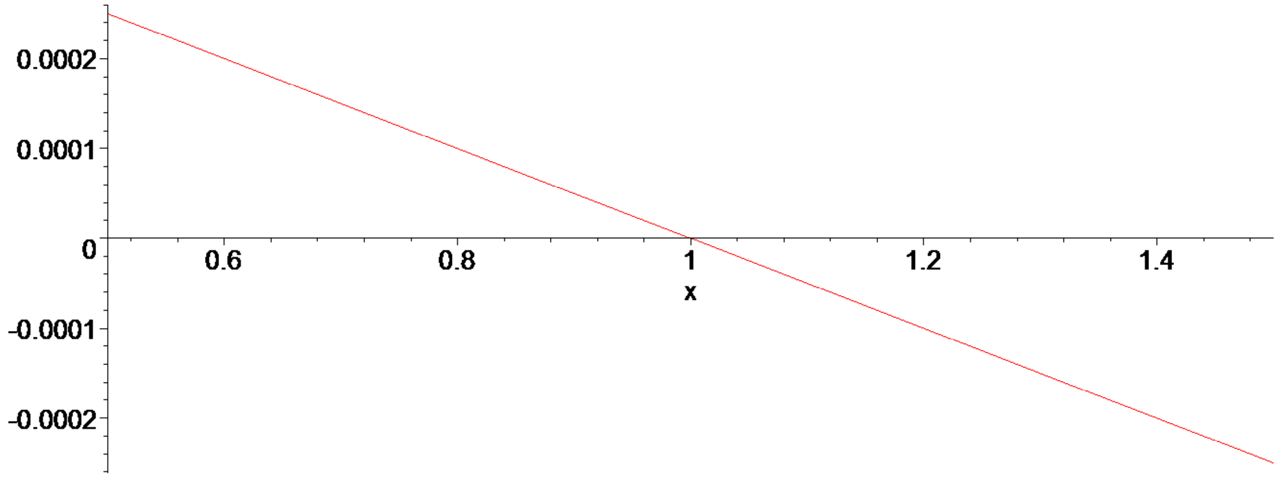
Obs.

```
> plot(x^(-3)*ln(x), x=0.5..1.5);
```



Se usarmos erro absoluto precisamos muitas iterações. Ex. $Ea < 10^{(-4)}$

```
> plot( (x-1)/(x-2000), x=0.5..1.5);
```


0.0002
0.0001
0
-0.0001
-0.0002
0.6
0.8
1
1.2
1.4
x

>

Método do Ponto Fixo

O método do ponto fixo é um método iterativo de aproximações sucessivas.
O Método das Aproximações Sucessivas é destinado para se determinar recursivamente pontos fixos de funções.
Dizemos que um ponto u é *ponto fixo* de função real f: R -> R quando $f(u) = u$.
É como se o ponto u permanecesse fixo em relação a função ou aplicação f . Aqui só consideramos funções f : R -> R. Veremos mais adiante que existe uma estreita relação entre ponto fixos e raízes de funções.

Para obter o ponto fixo de uma função, geralmetne começamos "chutando" um valor inicial x_0 para o ponto fixo procurado. Em seguida fazemos as iterações:

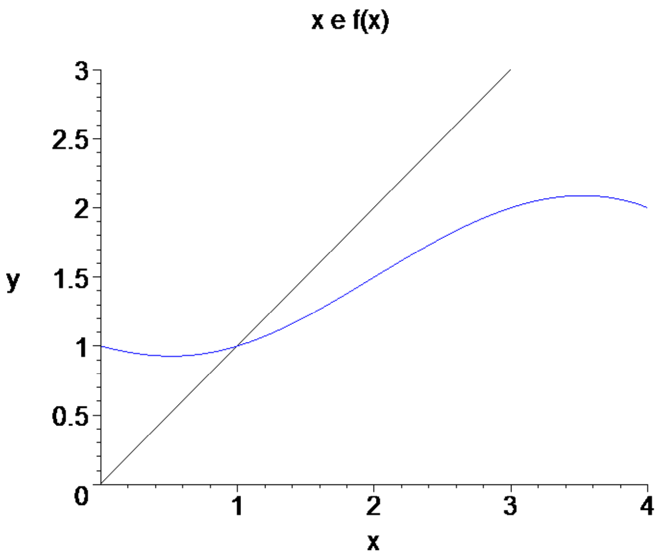
$$\mathbf{x_1 = f(x_0),}$$
$$\mathbf{x_2 = f(x_1),}$$
$$\mathbf{x_3 = f(x_2),}$$
etc ...

Se as inclinações das tangentes ao gráfico da função f, próximo ao ponto fixo, não forem muito acentuadas (verticalmente), então essa seqüência x_n deverá convergir para o ponto fixo procurado. Veja a figura abaixo para se ter uma idéia geométrica do método.

Código do programa que gera a figura abaixo

```
> restart;
> poly := interp( [-1,0,1,2,3,4,5], [1, 1, 1 , 1.5, 2. , 2 ,1 ]
  , z ):
> fp := unapply(poly, z):
> p[0] := [3.,0]:   p[1] := [3., fp(3.)]:
> for k from 2 to 9 do
>   if type(k, even) then p[k] := [ p[k-1][2] , p[k-1][2] ]
>   else p[k] := [ p[k-1][2] , fp(p[k-1][2]) ]
>   fi
> od:
> FP := plot( fp(x), x=0..4, y=0..3, color=blue, thickness=1):
> BI := plot( x, x=0..4, y=0..3, color=black):
> for j from 0 to 9 do
>    lista := [ seq( p[i] , i=0..j ) ]:
>    AP[j] := plot( lista , color=red):
>    G1[j] := plots[display]({ FP, BI, AP[j] })
> od:
> figura1 := plots[display]( seq(G1[i], i=0..9),
  insequence=true, scaling= constrained, title = `x e f(x)`):
```

```
> figura1; # Clique na figura
```



Toda função *contínua* f : R -> R possui um ponto fixo.
Isso é verdade, pois se $f(x)$ é contínua em R, a função $g(x) = x - f(x)$ possui um zero $[a, b]$.
O teorema abaixo estabelece as condições necessárias para que f tenha um único ponto fixo e fornece um *algoritmo* para calculá-lo.

Teorema: Seja f uma função derivável tal $|\mathbf{Df}(x)| < \mathbf{1}$ (o valor absoluto da derivada de *f* é menor do que 1) para todo x em $[a, b]$. Então f possui exatamente um ponto fixo ξ. Além disso, dado qualquer x_0 em $[a, b]$, a seqüência x_n definida por $x_n = f(x_{n-1})$ com $n = 1, 2, 3, ...$ converge para o ponto fixo.

Algoritmo Básico: Procurando um ponto fixo de $f(x) = \sqrt{1 + x}$. Tomaremos como um chute inicial $x_0 = 2$ e nestecaso faremos 7 iterações.

```
> restart:
> x[0] := 2.:
> for j from 1 to 7 do
> x[j] := sqrt( 1 + x[j-1] ):
> od;
```

$$x_1 := 1.732050808$$

$$x_2 := 1.652891650$$

$$x_3 := 1.628769981$$

$$x_4 := 1.621348199$$

$$x_5 := 1.619057812$$

$$x_6 := 1.618350337$$

$$x_7 := 1.618131743$$

De fato, resolvendo numericamente a equação $\sqrt{x + 1} - x = 0$ obtemos o ponto fixo

```
> xi = fsolve( sqrt(x+1) = x, x);
```

$$\xi = 1.618033989$$

```
>
```

Modelo de Procedimento: Mostraremos aqui um simples programa para o método das aproximações sucessivas com critério de parada. A sintaxe do procedimento é a seguinte:

AproxSucess(função, chute inicial, número de iterações, precisão).

```
> # procedimento AproxSucess
> AproxSucess := proc( f , a , imax , Erro )
> local xx , j:
> xx := evalf(a):
> # Aqui começa o loop
>   for j from 1 to imax do
>     if abs(f(xx)-xx) < Erro then break
>     else xx := f(xx)
>     fi
>   od:
> # Nesta etapa j já é j+1.
>     if j-1 = imax then print(`Não convergiu após`, j-1
  ,`iteradas`)
>     else print(`Solução Aproximada após`, j-1,`iteradas é `, xx
  )
>     fi;
> end:
```

Lembrando que o procedimento tem como variáveis de entrada

AproxSucess(**função, chute inicial, número de iterações, precisão**).

```
> h1 := x -> sqrt(x+2);
```

$$h1 := x \rightarrow \sqrt{x+2}$$

```
> AproxSucess(h1 , 18 , 15, .0001);
```

Solução Aproximada após, 8, *iteradas é* , 2.000127204

De fato;

```
> xi1 := fsolve( sqrt(x+2) = x, x);
```

$$\xi 1 := 2.000000000$$

E para

```
> h2 := x -> 2/x;
```

$$h2 := x \rightarrow \frac{2}{x}$$

```
> AproxSucess(h2, 2 , 150 , .0001);
```

Não convergiu após, 150, *iteradas*

```
> xi2 := fsolve( h2(x) = x, x);
```

$$\xi 2 := -1.414213562$$

```
>
```

Seja f(x) uma função contínua em [a, b] que contém um zero ξ de *f*. O método do ponto fixo consiste em transformar a equação f(x) = 0 em uma equação equivalente $x = \phi(x)$, e a partir de uma aproximação inicial x_0, gerar a sequência $\{x_k\}$ de aproximações para ξ pela relação

$$x_{k+1} = \phi(x_k)$$

pois a função $\phi(x)$ é tal que $f(\xi) = 0$ se e somente se $\phi(\xi) = \xi$.

Transformemos então o problema de encontrar um zero de $f(x)$ num problema de encontrar um ponto fixo de $\phi(x)$.

Uma função $\phi(x)$ que satisfaz a iteração acima é chamada de função de iteração.

Ex. Para a função

```
> f := x -> x^2 + x - 6: 'f(x) ' = f(x);
```

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

temos várias funções de iteração:

```
> phi[1] := x -> -x^2 + 6;
```

$$\phi_1 := x \to -x^2 + 6$$

```
> phi[2] := x -> sqrt( 6- x);
```

$$\phi_2 := x \to \sqrt{6 - x}$$

```
> phi[3] := x -> 6/x - 1;
```

$$\phi_3 := x \to \frac{6}{x} - 1$$

```
> phi[4] := x ->  6/(x+1);
```

$$\phi_4 := x \to \frac{6}{x+1}$$

```
>
```

A forma geral das funções de iteração $\phi(x)$ é

$$\phi(x) = x + A(x)\, f(x)$$

com a condição que em ξ, ponto fixo de $\phi(x)$, se tenha $A(\xi) \neq 0$.

Vamos mostrar que $f(\xi) = 0$ se e somente se $\xi = \phi(\xi)$.

1) Consideramos ξ tal que $f(\xi) = 0$.

A função de iteração no ponto ξ satisfaz, $\phi(\xi) = \xi + A(\xi)\, f(\xi)$, o que implica que $\phi(\xi) = \xi$ (pois $f(\xi) = 0$)

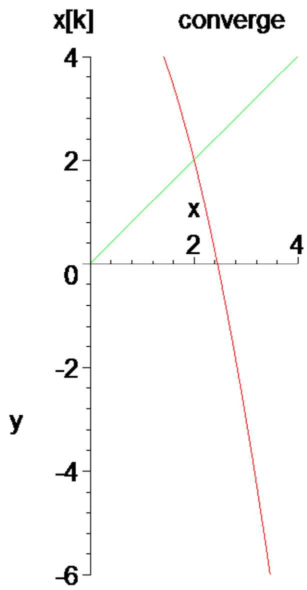
2) Consideramos agora $\phi(\xi) = \xi$.

Assim, $\xi = \xi + A(\xi)\, f(\xi)$ o que implica que $A(\xi)\, f(\xi) = 0$. como $A(\xi) \neq 0$ temos que $f(\xi) = 0$..

Verificamos então que existem muitas funções de iteração $\phi(x)$ para uma equação $f(x) = 0$.

A interpretação geométrica deste método é que a raíz da equação $f(x) = 0$ (ou da equação $x = \phi(x)$) é a interseção dos gráficos de $y = x$ e $y = \phi(x)$.

```
> plot( [-x^2 + 6, x], x=0..4, y=-6..4, scaling = constrained,
  tickmarks=[2,4], title = `x[k]          converge `);
```



Tomando $x_0 = 1.0$, temos

```
> phi :=  x -> - x^2 + 6;
```

$$\phi := x \to -x^2 + 6$$

```
> x1 := phi(1.);
```

$$x1 := 5.$$

```
> 'f(x1)' = f(x1);  'abs(f(x1))' = abs(f(x1));
```

$$f(x1) = 24.$$

$$|f(x1)| = 24.$$

```
> x2 := phi(x1);
```

$$x2 := -19.$$

```
> 'f(x2)' = f(x2);  'abs(f(x1))' = abs(f(x1));
```

$$f(x2) = 336.$$

$$|f(x1)| = 24.$$

```
>
```

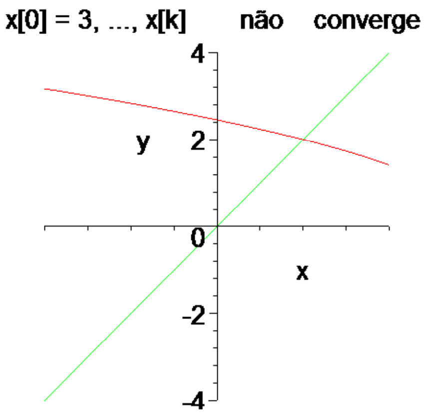
Verificamos que a sequência x_k está divergindo!

Agora vamos considerar a função iterativa

```
> phi2 := x -> sqrt(6-x):  'phi2(x)' = phi2(x);
```

$$\phi 2(x) = \sqrt{6 - x}$$

```
> plot( [sqrt(6-x), x], x=-4..4, y=-4..4, scaling = constrained,
  tickmarks=[2,4], title = `x[0] = 3, ..., x[k]       não
  converge `);
```



Tomando $x_0 = 1.0$, temos

```
> x1 := phi2(1.);
```

$$x1 := 2.236067977$$

```
> 'f(x1)' = f(x1);  'abs(f(x1))' = abs(f(x1));
```

$$f(x1) = 1.236067975$$

$$|f(x1)| = 1.236067975$$

```
> x2 := phi2(x1);
```

$$x2 := 1.940085571$$

```
> 'f(x2)' = f(x2);  'abs(f(x2))' = abs(f(x2));
```

$$f(x2) = -0.295982406$$

$$|f(x2)| = 0.295982406$$

```
> x3 := phi2(x2);
> 'f(x3)' = f(x3);  'abs(f(x3))' = abs(f(x3));
```

$$x3 := 2.014922934$$

$$f(x3) = 0.074837364$$

$$|f(x3)| = 0.074837364$$

```
>
```

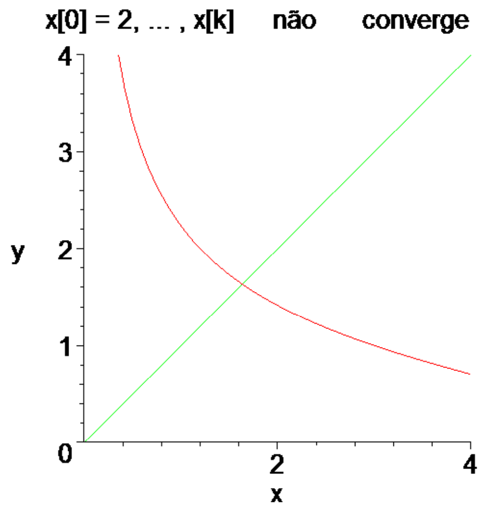
Verificamos que a sequência x_k obtida da função iterativa $\phi 2(x)$ está convergindo para o zero de $f(x)$ que é $\xi = 2$.

Ex. Verifique se as outras funções iterativas levam a uma sequência iterativa

```
> phi3 := x -> sqrt(6/x -1): 'phi3(x)' = phi3(x);
```

$$\phi 3(x) = \sqrt{\frac{6}{x} - 1}$$

```
> plot( [sqrt(6/x -1), x], x=0..4, y=0..4, scaling = constrained,
  tickmarks=[2,4], title = `x[0] = 2, ... , x[k]      não
  converge `);
```



>

Consequentemente, para certas funções iterativas escolhidas o Método do Ponto Fixo pode não convergir! Quais são as condições suficientes que devemos impor às funções iterativas para que ocorra a convergência?

Seja ξ uma raíz da equação $f(x) = 0$, isolada num intervalo I centrado em ξ e seja $\phi(x)$ é uma função de iteração para a equação $f(x) = 0$. Se

a) $\phi(x)$ e $\frac{d}{dx}\phi(x)$ são contínuas em I,

b) $\left|\frac{d}{dx}\phi(x)\right| \leq M < 1$, para todo $x \, \varepsilon \, I$ e

c) x_0 pertence a I,

então a sequência $\{x_k\}$ gerada pelo processo iterativo $x_{k+1} = \phi(x_k)$ converge para ξ.

Ex. Observe a primeira função iterativa

```
> 'phi(x)' = phi(x);
```

$$\phi(x) = -x^2 + 6$$

```
> Diff(phi(x), x) = diff(phi(x), x);
```

$$\frac{d}{dx}(-x^2 + 6) = -2x$$

Ambas as funções $\phi(x)$ e $\frac{d}{dx}\phi(x)$ são contínuas em R.

```
> 'abs(diff(phi(x), x)) < 1' ; abs(diff(phi(x), x)) < 1;
```

$$\left|\frac{d}{dx}\phi(x)\right| < 1$$

$$2|x| < 1$$

```
> solve( abs(diff(phi(x), x)) < 1, x);
```

$$\mathrm{RealRange}\left(\mathrm{Open}\left(\frac{-1}{2}\right), \mathrm{Open}\left(\frac{1}{2}\right)\right)$$

>

Observamos que não existe um intervalo I centrado em $\xi = 2$ tal que $\left| \frac{d}{dx} \phi(x) \right| < 1$ para todo x em I.

Portanto, a função iterativa $\phi(x) = 6 - x^2$ não satisfaz a condição 2) com relação a $\xi = 2$.

```
> restart;
```

Método de Newton - Raphson

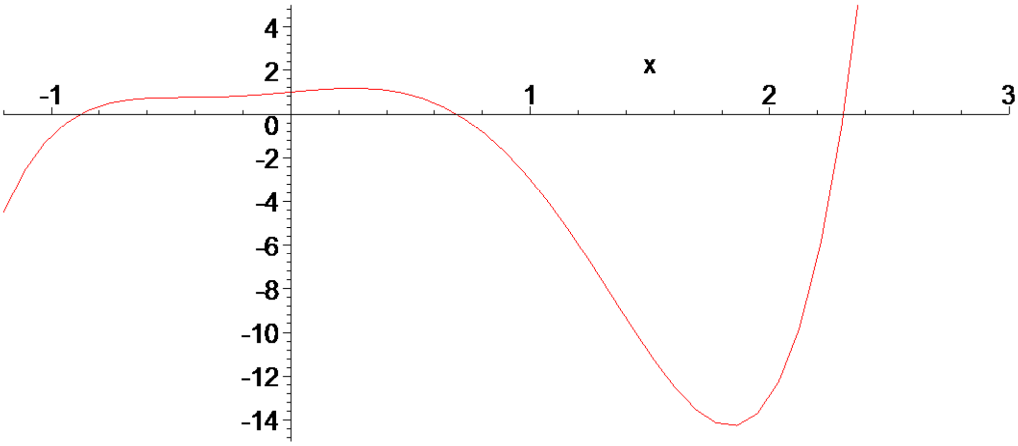
Seja

```
> f := x -> 2*x^5-3*x^4-4*x^3+x+1: 'f(x)'= f(x);
```

$$f(x) = 2\,x^5 - 3\,x^4 - 4\,x^3 + x + 1$$

ccujo gráfico

```
> plot(f(x), x=-1.2..3, -15..5);
```



nos mostra que f tem um zero entre $a = 0$ e $b = 1$. De fato;

```
> 'f(0)' = f(0); 'f(1)' = f(1);
```

$$f(0) = 1$$

$$f(1) = -3$$

A derivada de f

```
> 'df' = D(f)(x);
```

$$df = 10\,x^4 - 12\,x^3 - 12\,x^2 + 1$$

Escolhendo como uma aproximação inicial

```
> x0 := -1.;
```

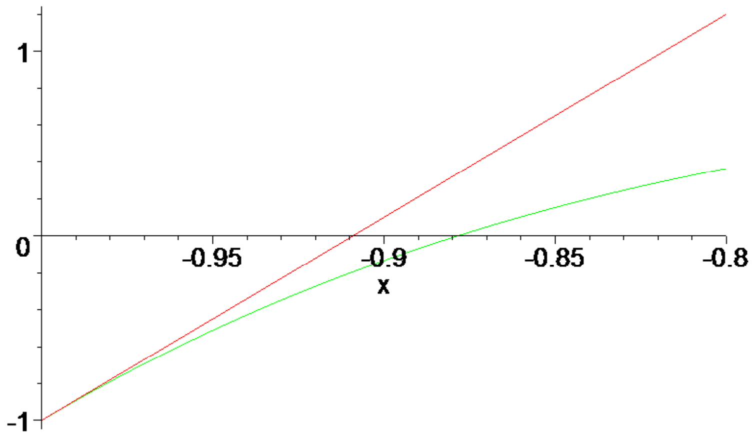
$$x0 := -1.$$

```
> x1 := '(x0 - f(x0)/D(f)(x0))'; x1 := evalf(x0 - f(x0)/D(f)(x0));
```

$$x1 := x0 - \frac{f(x0)}{D(f)(x0)}$$

$$x1 := -0.9090909091$$

```
> with(student): # geometricamente
> showtangent(f(x), x=-1, x=-1..-0.8, tickmarks=[4,3]);
```




```
> x2 := 'x1 - f(x1)/D(f)(x1)' ;  x2 := evalf(x1 - f(x1)/D(f)(x1));
```

$$x2 := x1 - \frac{f(x1)}{D(f)(x1)}$$

$$x2 := -0.8809876947$$

E assim sucessivamente, até que

```
> 'abs(f(xn))' < 10^(-m);
```

$$|f(xn)| < 10^{(-m)}$$

onde m é a ordem da precisão desejada.

```
>
```

O Método do Ponto Fixo, nos diz que uma das condições de convergência é que

```
> abs( diff(phi(x), x)) <= M;
```

$$\left|\frac{d}{dx}\phi(x)\right| \le M$$

em que

```
> M <= 1;
```

$$M \le 1$$

para todo x ε I, onde I é o intervalo centrado na raíz. A convergência será mais rápida quanto menor for $\left|\frac{d}{dx}\phi(x)\right|$.

```
> restart;
```

O que o Método de Newton faz é garantir a aceleração da convergência do Método do Ponto Fixo, escolhendo a função iterativa $\phi(x)$ tal que $\frac{d}{dx}\phi(x) = 0$.

A forma geral das funções de iteração $\phi(x)$ é

$$\phi(x) = x + A(x)\, f(x)$$

com a condição que em ξ, ponto fixo de $\phi(x)$, se tenha $A(\xi) \ne 0$. Neste caso queremos obter a função $A(x)$ tal que $\frac{d}{dx}\phi(x) = 0$.

Assim, derivando a expressão de $\phi(x)$ temos

$$D(\phi)(x) = 1 + D(A)(x)\, f(x) + A(x)\, D(f)(x) = 0$$

No ponto $x = \xi$, temos

$$D(\phi)(\xi) = 1 + D(A)(\xi)\, f(\xi) + A(\xi)\, D(f)(\xi)$$

E como $F(\xi) = 0$, vem que

$$D(\phi)(\xi) = 1 + A(\xi)\, D(f)(\xi)$$

e que

$$D(\phi)(\xi) = 0\ ,$$

temos

$$A(\xi)\, D(f)(\xi) = -1\ ,$$

o que implica que

$$A(\xi) = -\frac{1}{D(f)(\xi)}.$$

```
>
```

Assim, podemos tomar

$$A(x) = -\frac{1}{D(f)(x)}.$$

E a função iterativa,

$$\phi(x) = x - \frac{f(x)}{D(f)(x)}$$

será tal que $D(\phi)(\xi) = 0$.

De fato;

```
> phi := x -> x - f(x)/D(f)(x): 'phi(x)' = phi(x);
```

$$\phi(x) = x - \frac{f(x)}{D(f)(x)}$$

```
> diff(phi(x),x);
```

$$1 - \frac{\frac{d}{dx} f(x)}{D(f)(x)} + \frac{f(x)\, (D^{(2)})(f)(x)}{D(f)(x)^2}$$

```
> simplify(%);
```

$$\frac{f(x)\, (D^{(2)})(f)(x)}{D(f)(x)^2}$$

```
>
```

E como $f(\xi) = 0$, $D(\phi)(\xi) = 0$.

Desde que $D(f)(\xi) \neq 0$, uma vez que temos escolhido x_0, a sequência x_k determinada por

$$x_{k+1} = x_k - \frac{f(x_k)}{D(f)(x_k)}\ .$$

Geometricamente, o método de Newton é obtido da seguinte forma:

Pelo ponto $[x_k, f(x_k)]$ traçamos a reta tangente à curva $y = f(x)$ neste ponto

```
> L[k](x) = f(x[k]) + D(f)(x[k])*(x - x[k]);
```

$$L_k(x) = f(x_k) + D(f)(x_k)\,(x - x_k)$$

que é um modelo linear que aproxima a função $f(x)$ numa vizinhança de x_k. Encontrando o zero deste modelo obtemos

$$L_k(x) = 0 \text{ se e somente se } x = x_k - \frac{f(x_k)}{D(f)(x_k)}.$$

E assim, fazemos $x = x_{k+1}$.

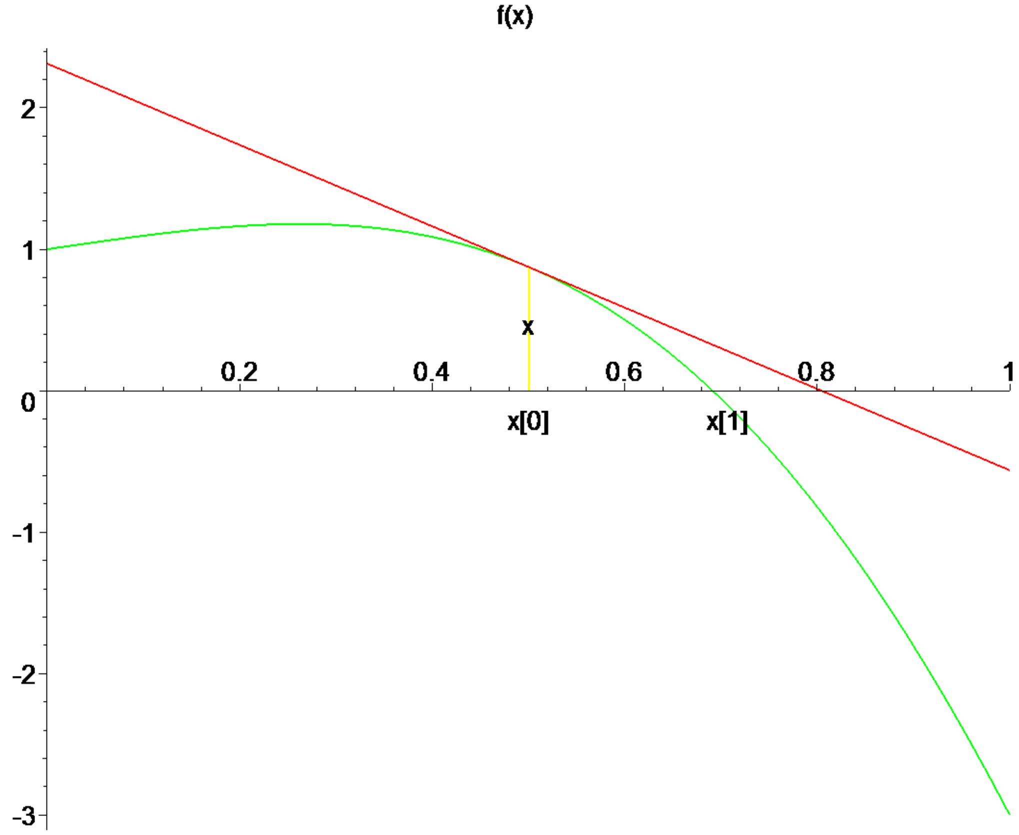
Vejamos a outra raíz de f no intervalo [0, 1].

```
> restart; with(student): with(plots):
  f := x -> 2*x^5-3*x^4-4*x^3+x+1: 'f(x)'= f(x);
```

```
Warning, the name changecoords has been redefined
```

$$f(x) = 2x^5 - 3x^4 - 4x^3 + x + 1$$

```
> g1 := showtangent( f(x), x=0.5, x=0..1, title= `f(x)`):
> g2 := textplot( [0.5, -0.2, `x[0]`],align={BELLOW,BELLOW}):
> g3 := textplot( [.7069175386, -0.2,
  `x[1]`],align={BELLOW,BELLOW}):
> display( g1 ,g2, g3, thickness=2);
```




```
> x0 := 0.5;
> x1 := '(x0 - f(x0)/D(f)(x0))'; x1 := evalf(x0 - f(x0)/D(f)(x0));
```

$$x0 := 0.5$$

$$x1 := x0 - \frac{f(x0)}{D(f)(x0)}$$

$$x1 := 0.8043478261$$

```
> x2 := '(x1 - f(x1)/D(f)(x1))'; x2 := evalf(x1 - f(x1)/D(f)(x1));
```

$$x2 := x1 - \frac{f(x1)}{D(f)(x1)}$$

$$x2 := 0.7069175386$$

```
> x3 := '(x2 - f(x2)/D(f)(x2))'; x3 := evalf(x2 - f(x2)/D(f)(x2));
```

$$x3 := x2 - \frac{f(x2)}{D(f)(x2)}$$

$$x3 := 0.6917398665$$

```
> x4 := '(x3 - f(x3)/D(f)(x3))';  x4 := evalf(x3 -
  f(x3)/D(f)(x3));
```

$$x4 := x3 - \frac{f(x3)}{D(f)(x3)}$$

$$x4 := 0.6913678786$$

```
> x5 := '(x4 - f(x4)/D(f)(x4))'; x5 := evalf(x4 - f(x4)/D(f)(x4));
```

$$x5 := x4 - \frac{f(x4)}{D(f)(x4)}$$

$$x5 := 0.6913676565$$

```
> 'f(x5)' = f(x5);
```

$$f(x5) = 0.5\ 10^{-9}$$

```
>
```

Ex. Analise a convergência do Método de Newton.
Ex. Faça um algorítmo para o Método de Newton.

Observação
O Método de Newton é também conhecido como Método das tangentes, devido sua interpretação gráfica natural.

Convergência do Método de Newton
A convergência do Método de Newton pode ser tratada usando o Teorema de convergência do Método das Aproximações Sucessivas, o qual pode ser visto nas Referência Bibliográfica [1].

Definição 3.2 Convergência quadrática
Dizemos que um método iterativo possui convergência quadrática se , em que k é chamada constante assintótica de proporcionalidade, e são os erros cometidos nas iterações correspondentes.

Teorema 3.1
O Método de Newton possui convergência quadrática.
Prova: Referência Bibliográfica [1].
Para observar as boas propriedades de convergência do Método de Newton, exibimos o seguinte exemplo:

Exemplo 3.4

O cálculo do número , usando o Método de Newton, consiste na resolução da equação e, usando uma tolerância fixa com temos:

A partir de uma solução inicial, geramos a sequência de soluções aproximadas:

solução inicial dada

0.1429 >

0.0103 >

0.0000 <

Como o critério de parada está satisfeito, podemos parar, e observar que esta sequência converge para

Desta forma, podemos observar que na medida em que os valores de aproximam da raiz , a convergência torna-se muito rápida, isto devido a propriedade da convergência quadrática do Método de Newton.

Algoritmo 3.2

1. Defina as funções f(x), e > 0 uma tolerância fixa.
2. Escolha uma solução inicial.
   Faça Pare=Falso e i=0
3. Enquanto Pare=Falso faça:
   3.1.
   3.2. Se , então Pare = Verdade
   Senão i = i+1

Observação

O leitor pode incluir nesse algoritmo, uma modificação no critério de parada, considerando o valor da função no ponto , isto é, .

Exemplo 3.5

Usando o Método de Newton, resolvemos a equação cos(x) - x = 0 com = 0.001.

A partir do processo iterativo , geramos a sequência:

solução inicial dada

0.0533 >

0.0000 <

Como o critério de parada está satisfeito, tomamos como solução aproximada para f(x) a solução .

Observação

Podemos ainda, modificar o Método de Newton, da seguinte forma:

O valor calculado da derivada na 1.a iteração, , em que , é fixado e substituído no processo iterativo de Newton durante as iterações:

Assim, temos:

o qual, é conhecido como Método Modificado de Newton.

```
>
```

O procedimento NR para o Método de Newton Raphson possui a seguinte sintaxe:

NR(função, chute inicial , número de iterações , precisão).

```
> restart:
> NR := proc( f, chute, imax , Erro)
> local x , n:
> x := chute:
> # Aqui começa o loop. "D" é o operador diferencial.
>   for n from 1 to imax do
>     if abs( f(x)/D(f)(x) ) < Erro then break
>     else x := x - ( f(x)/D(f)(x) ):
>     fi
>   od:
> # Agora n já é n+1.
>     if n-1 = imax then print(`Não convergiu após`, n-1
,`iteradas`)
>     else print(`Solução Aproximada com`, n-1,`iteradas é :`, x)
>     fi;
> end:
```

Exemplo

```
> h1 := x -> x-2*sin(x);
```

$$h1 := x \to x - 2\sin(x)$$

```
> NR(h1, 0.5 , 10, .0001);
```

Solução Aproximada com, 3, *iteradas é :*, -0.3950 10^{-9}

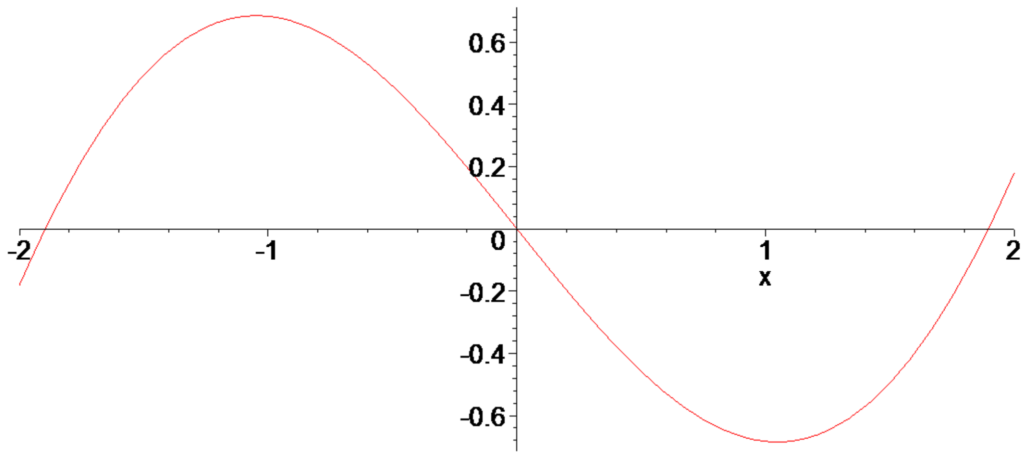
```
> NR(h1, 1.2 , 20, .0001);
```

Solução Aproximada com, 4, *iteradas é :*, 1.895505322

```
> NR(h1, 1.1 , 50, .0001);
```

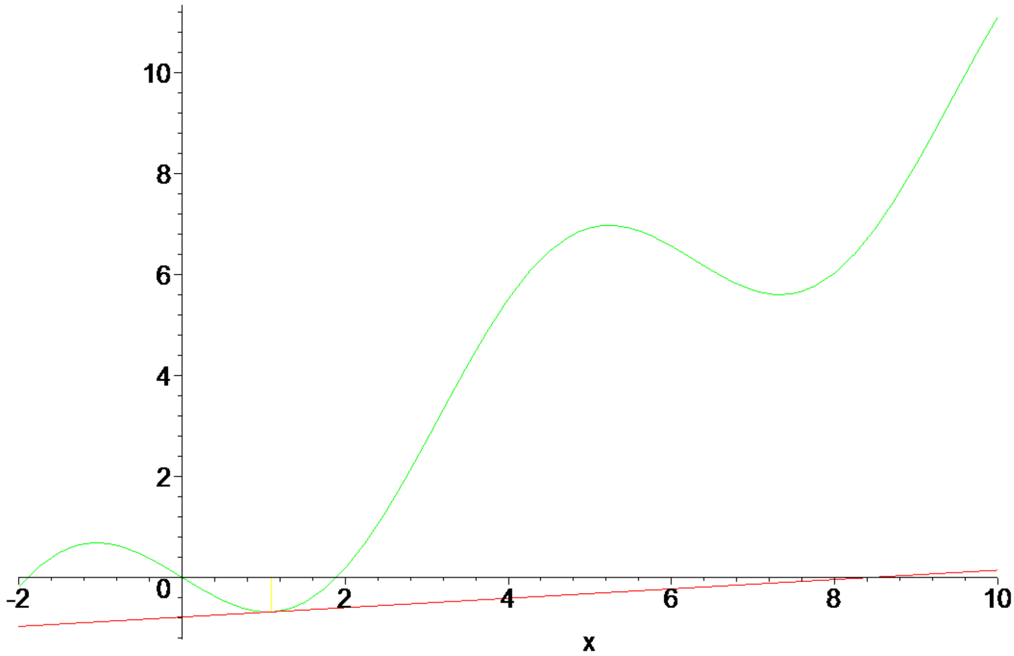
Não convergiu após, 50, *iteradas*

```
> plot( h1(x), x=-2..2);
```



Observe que dando um chute inicial $x_0 = 1.1$ o métod de Newton não converge.

```
> student[ showtangent](h1(x), x = 1.1, x=-2..10);
```




```
>
```

Método de Newton para polinômios

O Método de Newton pode ser utilizado no cálculo de raízes de polinômios. Entretanto, devido ao excessivo cálculo de potências x^n existentes nos polinômios e nas suas derivadas, o desempenho do método pode ser prejudicado. Com o uso de multiplicações encaixadas mostraremos uma maneira iterativa para se calcular os valores de polinômios. Em seguida adaptamos o Método de Newton-Raphson para esse ponto de vista.

Tomemos como exemplo o polinômio $p_3(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0$. Notemos que este polinômio pode ser escrito sob a forma $p_3(x) = ((a_3 x + a_2) x + a_1) x + a_0$. Então cálculo do valor de $p(x)$ poderá ser efetuado recursivamente:

$$b_3 = a_3$$
$$b_2 = a_2 + b_3 x$$
$$b_1 = a_1 + b_2 x$$
$$b_0 = a_0 + b_1 x,$$

de forma que $p_3(x) = b_0$. De uma forma geral, para

$$p_n(x) = a_n x^n + a_{n-1} x^{(n-1)} + \ldots + a_1 x + a_0,$$

o cálculo de $p(x)$ é dado pela seguinte seqüência b_n:

$$b_n = a_n$$
$$b_{n-j} = a_{n-j} + b_{n-j+1} x \quad \text{com} \quad j = 1, 2, \ldots, n,$$

de forma que $p_n(x) = b_0$.

Analogamente, uma vez calculados os b_j, o valor da derivada $\frac{d}{dx} p(x)$ é dado pela seqüência c_n:

$$c_n = b_n$$
$$c_{n-j} = b_{n-j} + c_{n-j+1}\, x \quad \text{com} \quad j = 1, 2, ..., n-1,$$

de forma que $\frac{d}{dx} p_n(x) = c_1$.

Finalmente, considerando essa nova maneira de se calcular $p(x)$ e $\frac{d}{dx}p(x)$ reescrevemos o algoritmo de Newton.

O Procedimento NewtonPol : Este simples procedimento não contém comandos para testar a compatibilidade dos dados fornecidos e nem possui critérios de parada. A sintaxe básica é a seguinte:

NewtonPol(polinômio , variável , chute inicial , número de iterações).

```
> restart:
> NewtonPol := proc(p, t , chute, imax)
> local x, a, s, b, c, k, j, nmax:
> x := evalf(chute):
> # grau de p
> nmax := degree(p):
> # extraindo os coeficientes
>    for s from 0 to nmax do
>    a[s] := coeff(p,t,s)
>    od:
> # começando o loop.
>    for k from 1 to imax do
>       b := a[nmax]:  c := a[nmax]:
>       for j from 1 to nmax-1 do
>         b := a[nmax-j] + b*x:
>         c := b + c*x:
>       od:
>       # Neste momento já temos b[1]=b e c[1]=c.
>    b := a[0] + b*x:  # este é o b[0].
>    x := x - b/c: # Fórmula de Newton-Raphson
>    od:
> end:
```

Vejamos alguns exemplos

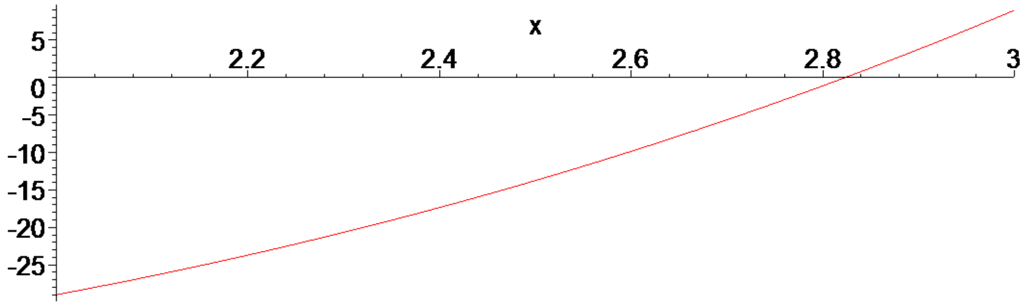
```
> p1 := 2*x^3-45;
```

$$p1 := 2\,x^3 - 45$$

```
> NewtonPol( p1, x , 6 , 10);
```

$$2.823108087$$

```
> plot( p1, x=2..3);
```



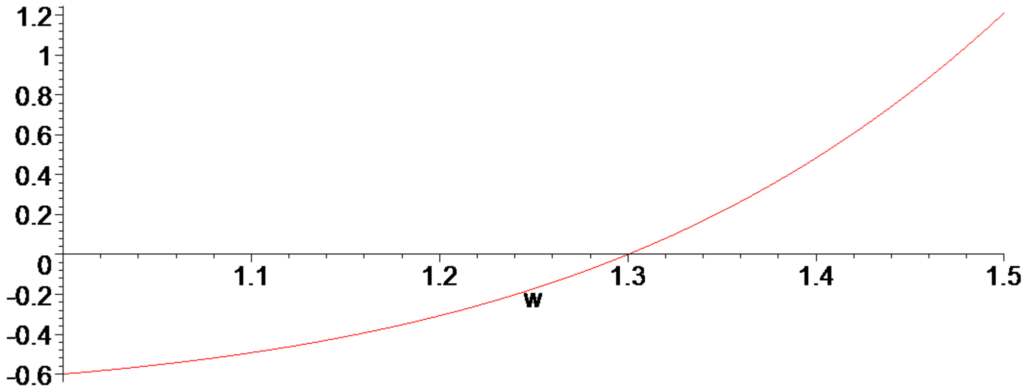

```
> p2 := w^5 - 1.3*w^4 + w -1.3;
```

$$p2 := w^5 - 1.3\, w^4 + w - 1.3$$

```
> NewtonPol( p2 , w , 2 , 10 );
```

$$1.300000000$$

```
> plot( p2, w=1..1.5);
```




```
>
```

Método da Secante

Uma das desvantagens do Método de Newton é a necessidade de se obter a derivada $\mathrm{D}(f)(x)$ da função $\mathrm{f}(x)$ e calcular o seu valor numérico a cada iteração.
Para contornar este problema, fazemos uma aproximação da derivada usando razão de diferenças.

```
>  D(f)(x[k]) = (f(x[k]) - f(x[k-1]) ) / (x[k] - x[k-1]);
```

$$\mathrm{D}(f)(x_k) = \frac{\mathrm{f}(x_k) - \mathrm{f}(x_{k-1})}{x_k - x_{k-1}}$$

onde x_{k-1} e x_k são duas aproximações para o zero de f.
Assim, a função de iteração fica definida como

```
> x[k+1] := x[k] - f(x[k]) / (( f(x[k]) - f(x[k-1]) )/ (x[k] -
  x[k-1]) );
```

$$x_{k+1} := x_k - \frac{\mathrm{f}(x_k)\,(x_k - x_{k-1})}{\mathrm{f}(x_k) - \mathrm{f}(x_{k-1})}$$

ou então,

```
> phi(x[k]) = simplify(  x[k] - f(x[k]) / (( f(x[k]) - f(x[k-1])
  )/ (x[k] - x[k-1]) ));
```

$$\phi(x_k) = -\frac{x_k\, f(x_{k-1}) - f(x_k)\, x_{k-1}}{f(x_k) - f(x_{k-1})}$$

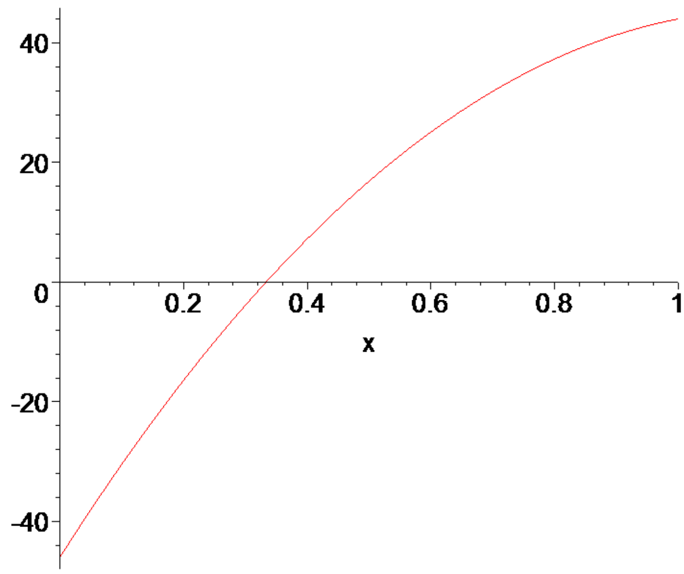
>

Ex. Encontre o zero da função polinomial $p(x) = 3\,x^2 - 76\,x^2 + 163\,x - 46$.

```
> p := x -> 3*x^3 - 76*x^2 + 163*x - 46;
```

$$p := x \rightarrow 3\,x^3 - 76\,x^2 + 163\,x - 46$$

```
> plot( p(x), x=0..1);
```




```
> 'p(0)' = p(0); 'p(1)' = p(1); 'p(2)' = p(2); 'p(20)' = p(20);
  'p(25)' = p(25);
```

$$p(0) = -46$$

$$p(1) = 44$$

$$p(2) = 0$$

$$p(20) = -3186$$

$$p(25) = 3404$$

>

A função polinomial tem um zero entre $a = 0$ e $b = 1$.

Pelo método da secante, podemos escolher a secante inicial passando por $(a, f(a))$ e $(b, f(b))$

```
> x[0] := 0: x[1] := 1:
> x[2] :=  evalf( x[1] - p(x[1]) / (( p(x[1]) - p(x[0]) )/ (x[1] -
  x[0]) ) );
```

$$x_2 := 0.5111111111$$

```
> 'p(x[2])' = p(x[2]);
```

$$p(x_2) = 17.85784362$$

```
> x[3] :=  evalf( x[2] - p(x[2])/(( p(x[2]) - p(x[1]))/(x[2] -
  x[1]) ) );
```

$$x_3 := 0.1771485566$$

```
> 'p(x[3])' = p(x[3]);
```

$$p(x_3) = -19.49311009$$

```
> x[4] :=  evalf( x[3] - p(x[3])/((p(x[3]) - p(x[2]))/(x[3] -
  x[2]) ) );
```

$$x_4 := 0.3514404607$$

```
> 'p(x[4])' = p(x[4]);
```

$$p(x_4) = 2.02822454$$

```
> x[5] :=  evalf( x[4] - p(x[4]) / (( p(x[4]) - p(x[3]))/(x[4] -
  x[3])));
```

$$x_5 := 0.3350147546$$

```
> 'p(x[5])' = p(x[5]);
```

$$p(x_5) = 0.19035471$$

```
> x[6] :=  evalf( x[5] - p(x[5]) / (( p(x[5]) - p(x[4]))/(x[5] -
  x[4])));
```

$$x_6 := 0.3333134858$$

```
> 'p(x[6])' = p(x[6]);
```

$$p(x_6) = -0.00224941$$

```
>
```

Na 6a. iteração obtemos um erro da ordem de $10^{(-2)}$.

Pelo método de Newton, teríamos

```
> x0 := 0;
> x1 := '(x0 - p(x0)/D(p)(x0))'; x1 := evalf(x0 - p(x0)/D(p)(x0));
```

$$x0 := 0$$

$$x1 := x0 - \frac{p(x0)}{D(p)(x0)}$$

$$x1 := 0.2822085890$$

```
> 'p(x1)' = p(x1);
```

$$p(x1) = -5.98534155$$

```
> x2 := '(x1 - p(x1)/D(p)(x1))'; x2 := evalf(x1 - p(x1)/D(p)(x1));
```

$$x2 := x1 - \frac{p(x1)}{D(p)(x1)}$$

$$x2 := 0.3317474779$$

```
> 'p(x2)' = p(x2);
```

$$p(x2) = -0.17991388$$

```
> x3 := '(x2 - p(x2)/D(p)(x2))'; x3 := evalf(x2 - p(x2)/D(p)(x2));
```

$$x3 := x2 - \frac{p(x2)}{D(p)(x2)}$$

$$x3 := 0.3333317165$$

```
> 'p(x3)' = p(x3);
```

$$p(x3) = -0.00018324$$

```
>
```

Ex. O polinômio

```
> p := x -> x^3 - 3*x + 1: `p(x)`= p(x);
```

$$p(x) = x^3 - 3\,x + 1$$

possui um zero entre

```
> 'p(-3)'= p(-3);  'p(-1.5)'= p(-1.5);
```

$$\mathrm{p}(-3) = -17$$

$$\mathrm{p}(-1.5) = 2.125$$

Usando O método de Newton, temos

```
> x0 := -2;
> x1 := '(x0 - p(x0)/D(p)(x0))'; x1 := evalf(x0 - p(x0)/D(p)(x0));
```

$$x0 := -2$$

$$x1 := x0 - \frac{\mathrm{p}(x0)}{\mathrm{D}(p)(x0)}$$

$$x1 := -1.888888889$$

```
> 'p(x1)' = p(x1);
```

$$\mathrm{p}(x1) = -0.072702333$$

```
> x2 := '(x1 - p(x1)/D(p)(x1))'; x2 := evalf(x1 - p(x1)/D(p)(x1));
```

$$x2 := x1 - \frac{\mathrm{p}(x1)}{\mathrm{D}(p)(x1)}$$

$$x2 := -1.879451567$$

```
> 'p(x2)' = p(x2);
```

$$\mathrm{p}(x2) = -0.000503850$$

```
> x3 := '(x2 - p(x2)/D(p)(x2))'; x3 := evalf(x2 - p(x2)/D(p)(x2));
```

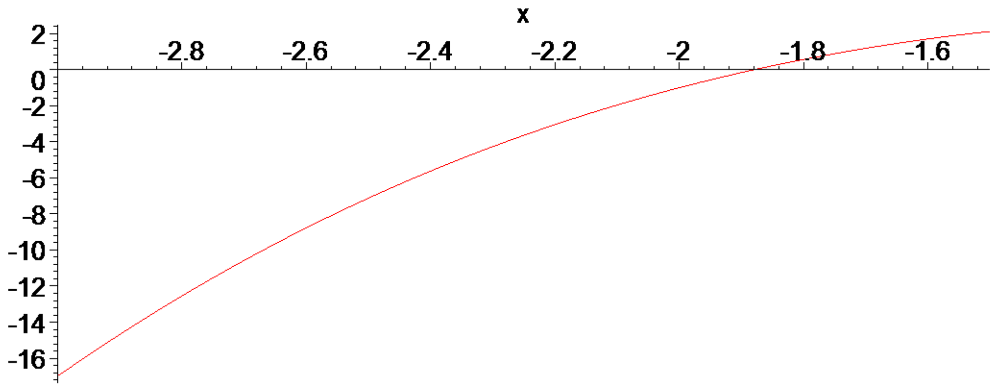
$$x3 := x2 - \frac{\mathrm{p}(x2)}{\mathrm{D}(p)(x2)}$$

$$x3 := -1.879385245$$

```
> 'p(x3)' = p(x3);
```

$$\mathrm{p}(x3) = -0.26\ 10^{-7}$$

```
> plot(p(x), x=-3..-1.5);
```




```
>
```

Problemas

O critério comumente usado para parar uma sequênciade aproximações x^k de uma raíz r da equação $f(x)=0$ é escolhermos um número $0<\varepsilon$ e tomarmos como aproximação X, se $\left|x^k - X\right| \le \varepsilon$.
Sabendo porém, que isto não implica que $\left|x^k - r\right| \le \varepsilon$.
Determine que condição $D(\phi)(x)$, onde $\phi(x)$ é a função de iteração, deve satisfazer a fim de que $\left|x^k - x^{(k-1)}\right| \le \varepsilon$ implique $\left|x^k - r\right| \le \varepsilon$.

Considerando que $\phi_1(x)$ e $\phi_2(x)$ são duas funções de iteração que convergem para a uma solução r da equação $f(x)=0$, como determinar a que converge mais rápido?

O método de Newton de Newton aplicado a $f(x)=\frac{1}{x}-q=0$ produz a iteração $x_n = x_{n-1}\,(2-q\,x_{n-1})$ que nos dá o inverso do número q sem efetuar divisões. Faça um programa que use esta iteração para o número $q=\mathbf{e}=2.718218$ começando com $x_0=.3$ e com $x_o=1$.
Explique cada passo e o que ocorre em cada um dos casos.

Ache o valor de x para o qual $y=\mathbf{e}^{(-x)}\log(x)$ tem um ponto de inflexão

Seja r uma raíz dupla da equação $f(x)=0$, isto é, podemos escrever $f(x)=(x-r)^2\,g(x)$, sendo $g(r)\neq 0$.
a) Qual a ordem de convergência do método de Newton neste caso? Explique a sua resposta.
b) Se usarmos a função de iteração $\phi(x)=x-\frac{2\,f(x)}{D(f)(x)}$, que é uma modificação do Método de Newton, determine qual é a ordem de convergência desta iteração e justifique.

O Método de Newton para algumas funções é inconveniente porque precisamos da derivada de $f(x)$.
Um método que converge mas tem ordem de convergência 2 é dado pelo seguinte procedimento chamado Método da Falsa Posição.
Seja I um intervalo que contém uma única raíz r de $f(x)=0$. Sejam x_0 e x_1 dois pontos pertencentes a I tais que $f(x_0)\,f(x_1)<0$. O ponto x_2 é a interseção da reta que contém $(x_0, f(x_0))$ e $(x_1, f(x_1))$ com o eixo dos x que está mais perto da raíz r do que pelo menos um dos pontos x_0 e x_1.
Obtém - se uma nova aproximação x_3,
usando x_0 e x_2 se $f(x_2)\,f(x_0)<0$ ou
usando x_2 e x_1 se $f(x_2)\,f(x_1)<0$. Formalize um algorítmo para este método e dê um exemplo em que podemos usá-lo.

JAS Zero_de_func10