

Raízes e otimização

Renato Assunção

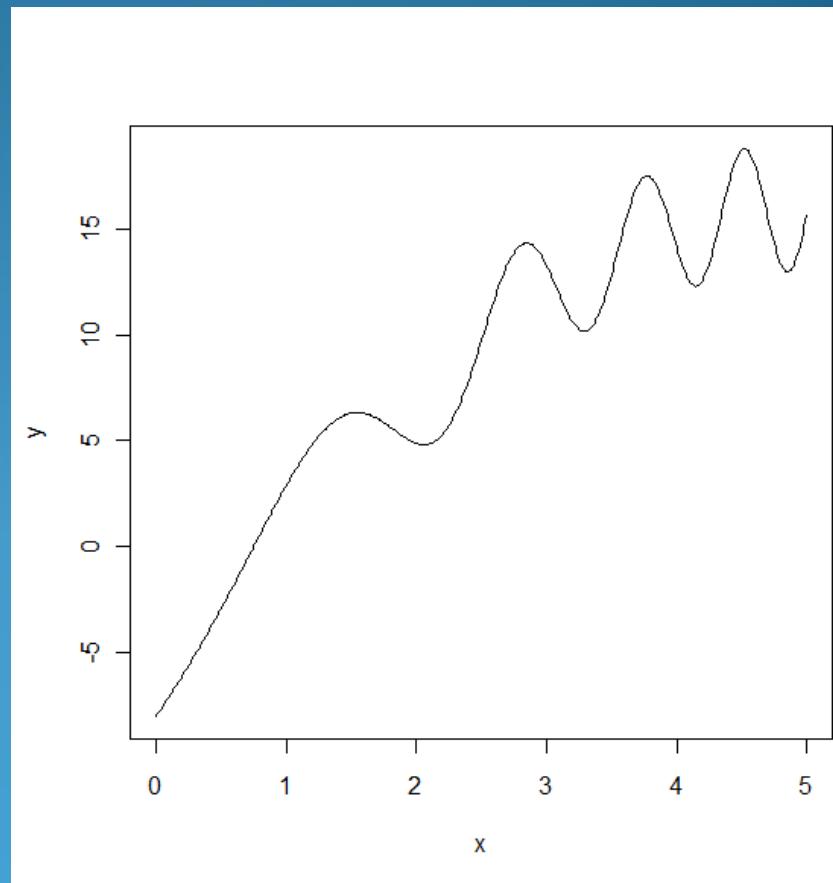
DCC, UFMG

Raízes de equações

- Um tipo de problema bastante comum é o de achar raízes de equações da forma $f(x) = 0$, onde $f(x)$ pode ser um polinômio ou uma função transcendental
- O valor de x que satisfaz $f(x) = 0$ é chamado de raiz da equação.
- Raramente podemos obter as raízes de tais funções de modo exato.
- Vários procedimentos fornecem métodos para calcular uma sequência de aproximações, que convergem para uma solução tão precisa quanto necessária, resguardadas algumas condições

De funções relativamente simples...

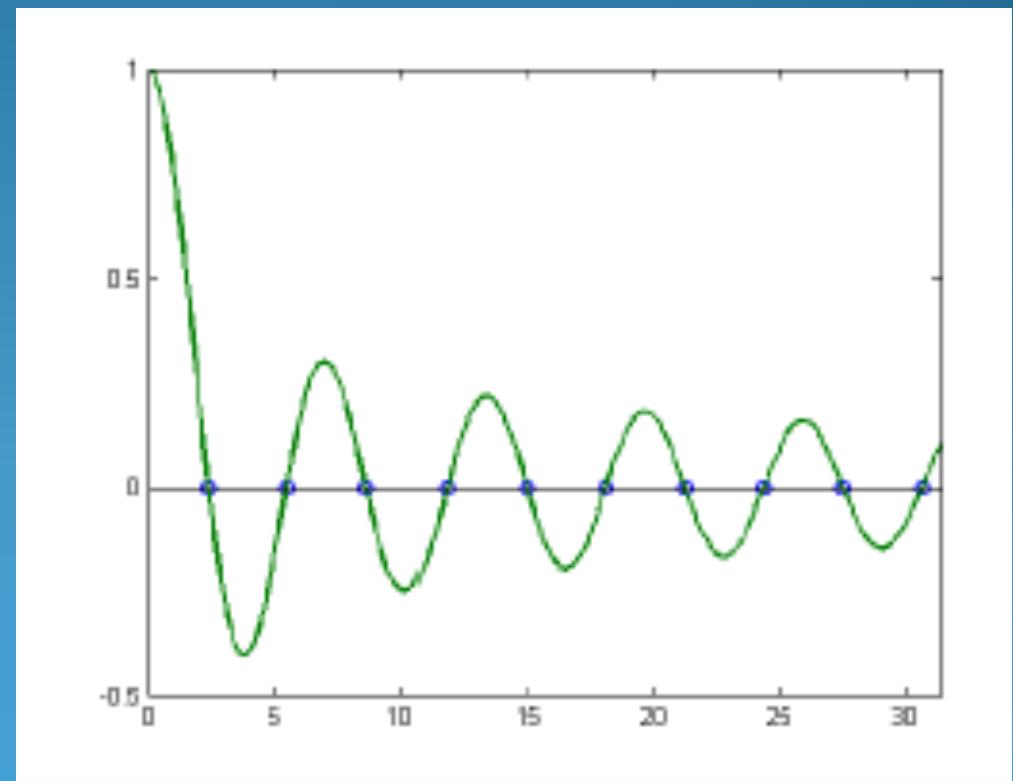
- Polinômios e
- combinações finitas de funções
- transcendentais
- Fácil de plotar.
- $f(x)=3^*\sin(x^2) + e^{-x} - (x-3)^2 + 4^*x$



...a funções mais complexas

- Achar a primeira raiz positiva da função de Bessel de primeira ordem (solução de certas equações diferenciais ordinárias)

$$J_0(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{2^{2k}(k!)^2} x^{2k}.$$



Outro problema: maximização

- Maximizar uma função: otimização de recursos.
- No fundo, problema pode ser reduzido a encontrar a raiz de uma função.
- Achar $\text{Max}_x f(x)$
- É equivalente a achar a raiz da função derivada
- $f'(x) = 0$
- Assim, maximizar reduz-se a achar raízes de equações não-lineares.

Sistemas de equações não-lineares

Most of this chapter will be concerned with the solution of a **single nonlinear equation**. However, **systems of nonlinear equations** are also important (and difficult) to solve.

Example

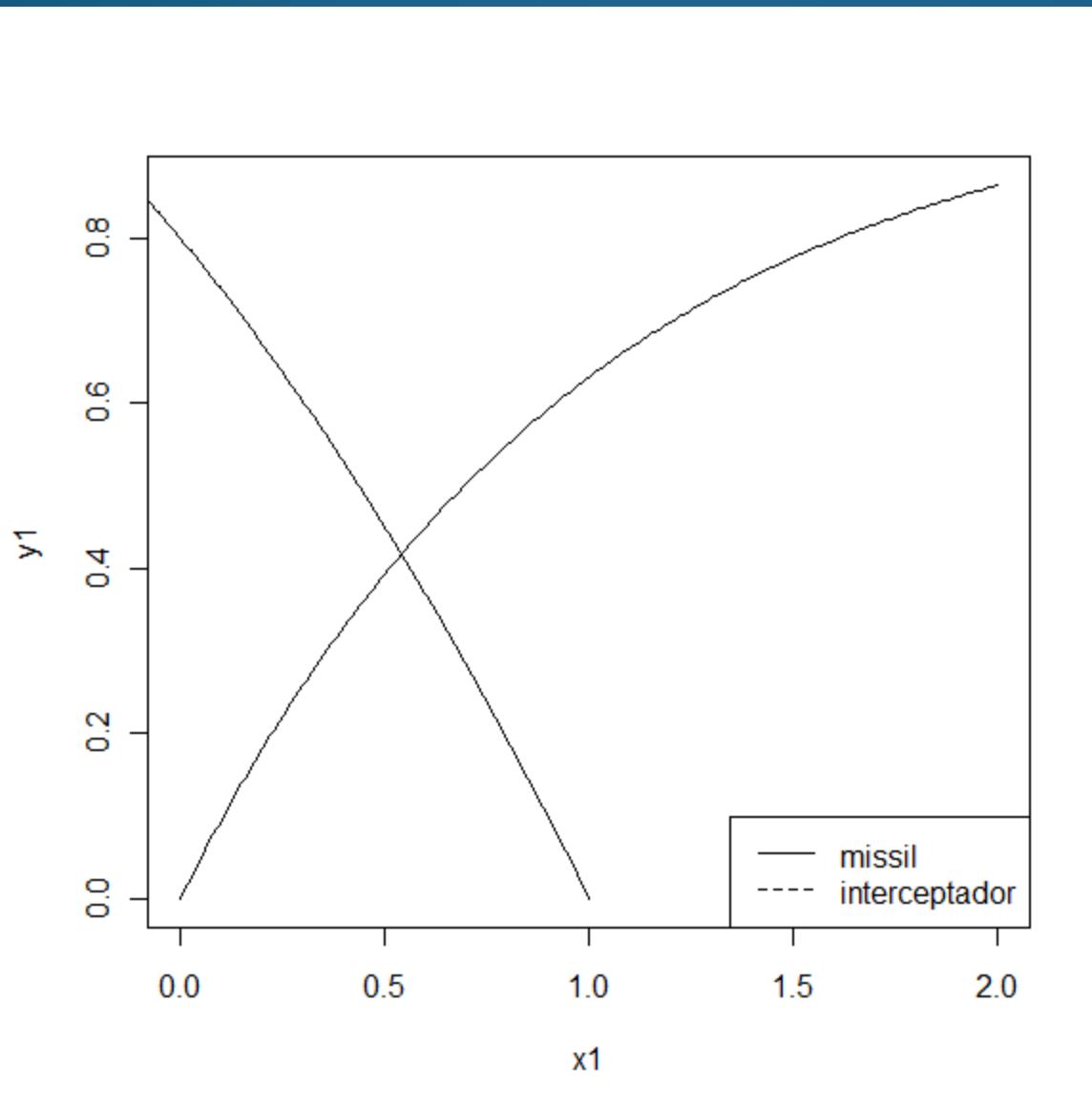
Consider a missile M following the parametrized path

$$x_M(t) = t, \quad y_M(t) = 1 - e^{-t},$$

and a missile interceptor I whose launch angle α we want to determine so that it will intersect the missile's path. Let the parametrized path for the interceptor be given as

$$x_I(t) = 1 - t \cos \alpha, \quad y_I(t) = t \sin \alpha - \frac{t^2}{10}.$$

$$\alpha = \pi/4$$



Sistema de equações não-lineares

$$x_M(t) = t, \quad y_M(t) = 1 - e^{-t}, \quad \text{Equação do míssil}$$

$$x_I(t) = 1 - t \cos \alpha, \quad y_I(t) = t \sin \alpha - \frac{t^2}{10}. \quad \text{Equação do interceptador}$$

Thus, we want to solve the nonlinear system

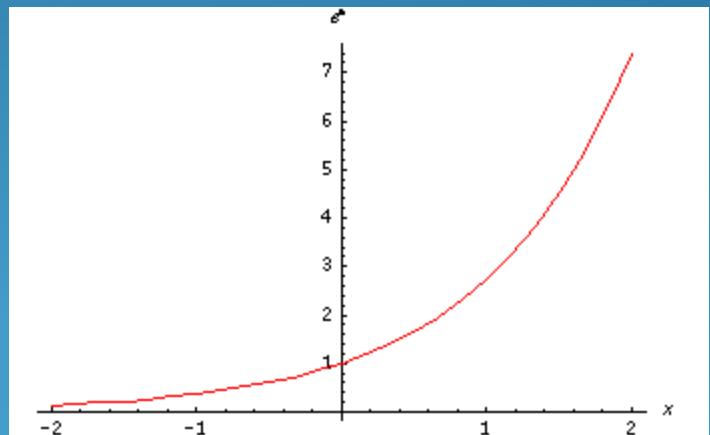
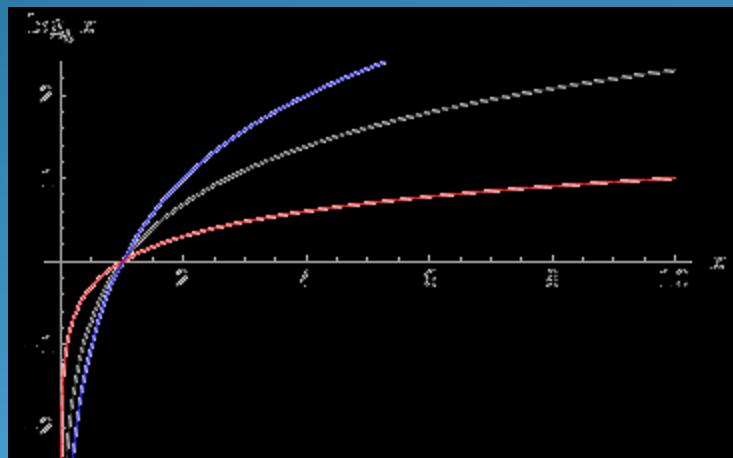
$$\begin{cases} t &= 1 - t \cos \alpha \\ 1 - e^{-t} &= t \sin \alpha - \frac{t^2}{10} \end{cases} \quad \text{or} \quad \begin{cases} f(t, \alpha) &= t - 1 + t \cos \alpha = 0 \\ g(t, \alpha) &= 1 - e^{-t} - t \sin \alpha + \frac{t^2}{10} = 0. \end{cases}$$

Um teorema que dispensa prova

- Antes de examinarmos vários métodos para determinar raízes isoladas de $f(x) = 0$, vamos ver o teorema abaixo e alguns exemplos
- **Teorema:** Suponha que uma função contínua $f(x)$ assume valores de sinais opostos nos pontos extremos do intervalo $[a,b]$.
- Isto é, suponha que $f(a) * f(b) < 0$
- Então existe pelo menos um ponto $x' \in [a,b]$, tal que $f(x') = 0$
- Isto é, existe uma raiz entre a e b .

Exemplos

- Vamos examinar o comportamento das funções $f(x) = \ln(cx^p)$ e $f(x) = e^x$



Método da Bissecção

- Considere o intervalo $[a,b]$ para o qual $f(a) * f(b) < 0$.
- No método da bissecção nós calculamos o valor da função $f(x)$ no ponto médio $x_1 = (a + b)/2$
- Caso $f(x) = 0$, x_1 é a raiz procurada e o processo para.
- Se $f(a) * f(x_1) < 0$, a raiz procurada está entre a e x_1 , e repete-se o processo para o intervalo $[a, x_1]$.
- Caso contrário, $f(x_1) * f(b) < 0$, e a raiz procurada está entre x_1 e b . Logo, repete-se o processo para o intervalo $[x_1, b]$

Método da Bissecção

○ Algoritmo:

- Para $k=1, 2, \dots$, faça

$$x_k = \frac{a+b}{2}$$

$$\text{Se } f(a) \times f(x_k) \begin{cases} < 0, & b = x_k \\ > 0, & a = x_k \end{cases}$$

Para usar um método numérico para encontrar uma raiz devemos ter uma idéia inicial da localização da mesma, o que geralmente é feito através da observação do gráfico da função ou do teorema mencionado anteriormente

Para obtermos uma raiz com uma determinada precisão ϵ devemos, durante o processo iterativo, efetuar o seguinte teste

$$\frac{|x_{k+1} - x_k|}{|x_{k+1}|} < \epsilon,$$

onde ϵ é uma precisão pré-fixada.

Se isto for verdade, temos $\bar{x} = x_{k+1}$ e' a raiz procurada

Precisão e parada: cuidados

- Geralmente usamos $\epsilon = 10^{-m}$, onde m é o número de casa decimais corretas que desejamos
- Alguns autores usam $f(x_{k+1}) < \epsilon$ como critério de parada, mas isso não garante a proximidade com a raiz procurada.
- Alguns autores usam o erro absoluto como critério de parada, isto é, $|x_{k+1} - x_k| < \epsilon$. Isso pode determinar uma condição de parada muito exigente

Precisão e parada: cuidados

- Finalmente, quando calculamos o erro relativo de um processo iterativo, devemos usar a fórmula $|x_{k+1} - x_k| < \epsilon \times \max\{1, |x_{k+1}|\}$ para que o processo iterativo não estacione em valores de $|x_{k+1} - x_k|$ muito próximos de 0
- Além disso, devemos estabelecer um número máximo de iterações para evitar loops infinitos

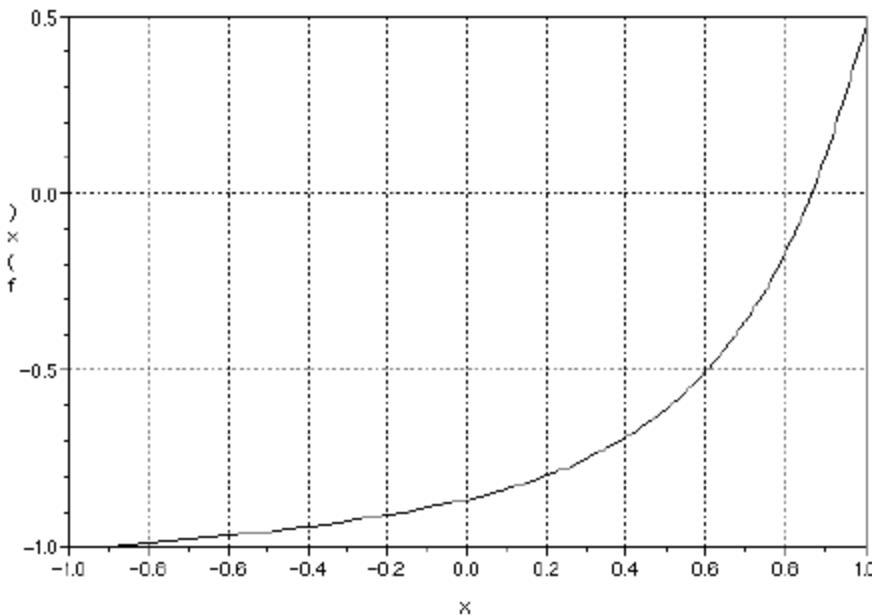
Bisection Algorithm

```
while abs (b-a) > eps*abs (b)
    x = (a + b)/2;
    if sign(f(x)) == sign(f(b))
        b = x;      % set [a,x] as new [a,b]
    else
        a = x;      % set [x,b] as new [a,b]
    end
end
```

Falta ainda o limite maximo do numero de iterações.
Se atingido, enviar uma mensagem de warning: limite atingido

Exemplo

- Usando o método da bissecção, calcular a raiz positiva de $f(x) = (x+1)^2 e^{x^2-2} - 1 = 0$



k	a	b	x_k	$f(x_k)$
1	0	1	0.5	-0.609009
2	0.5	1	0.75	-0.272592
3	0.75	1	0.875	0.023105
4	0.75	0.875	0.8125	-0.139662
5	0.8125	0.875	0.84375	-0.062448
6	0.84375	0.875	0.859375	-0.020775

Vantagens e desvantagens

- Sempre converge
- O intervalo é diminuído pela metade a cada iteração

- Convergência é lenta
- Se uma das estimativas é próxima da solução, a convergência é mais lenta ainda
- Em alguns casos, não conseguimos determinar estimativas inferior e superior para o intervalo

Método de Newton

By Taylor's theorem (assuming $f''(\xi)$ exists) we have

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2}f''(\xi).$$

So, for values of x_0 reasonably close to x we can approximate

$$f(x) \approx f(x_0) + (x - x_0)f'(x_0).$$

Since we are trying to find a root of f , i.e., we are hoping that $f(x) = 0$, we have

$$0 \approx f(x_0) + (x - x_0)f'(x_0) \iff x - x_0 \approx -\frac{f(x_0)}{f'(x_0)}.$$

This motivates the Newton iteration formula

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

where an initial guess x_0 is required to start the iteration.



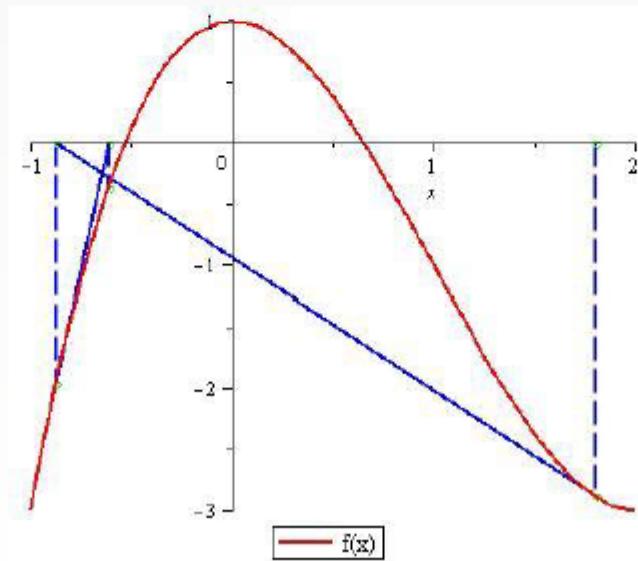
Graphical Interpretation

Consider the tangent line to the graph of f at x_n :

$$y - f(x_n) = f'(x_n)(x - x_n) \implies y = f(x_n) + (x - x_n)f'(x_n).$$

To see how this relates to Newton's method, set $y = 0$ and solve for x :

$$0 = f(x_n) + (x - x_n)f'(x_n) \iff x = x_n - \frac{f(x_n)}{f'(x_n)}.$$



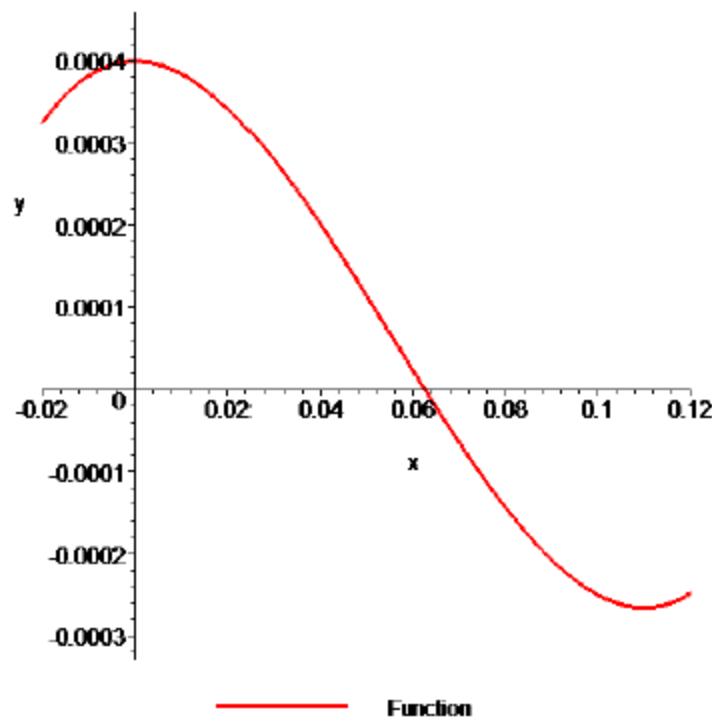
Newton Iteration

```
while abs(x - xprev) > eps*abs(x)
    xprev = x;
    x = x - f(x) / fprime(x);
end
```

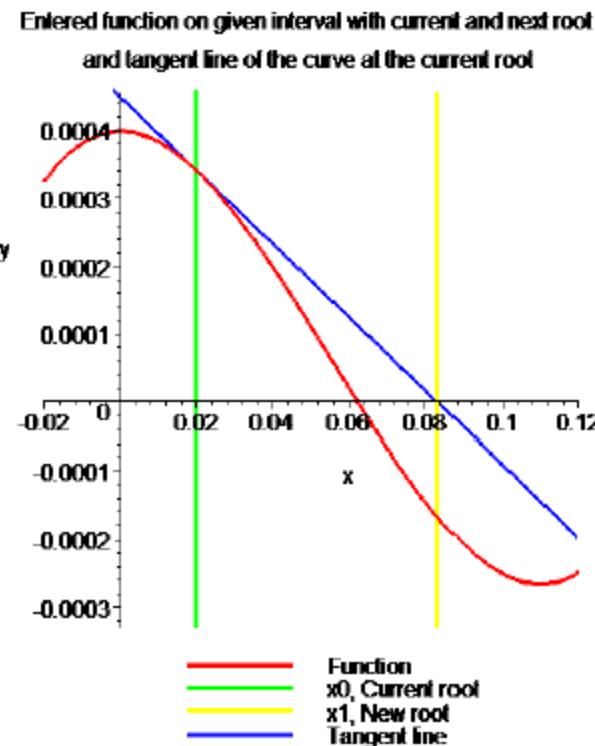
Convergence of Newton's method depends quite a bit on the choice of the initial guess x_0 . If successful, the algorithm above converges very quickly to within machine accuracy.

$$f(x) = x^3 - 0.165x^2 + 3.993 \times 10^{-4}$$

Entered function on given interval



Iteração 1



$$x_0 = 0.02$$

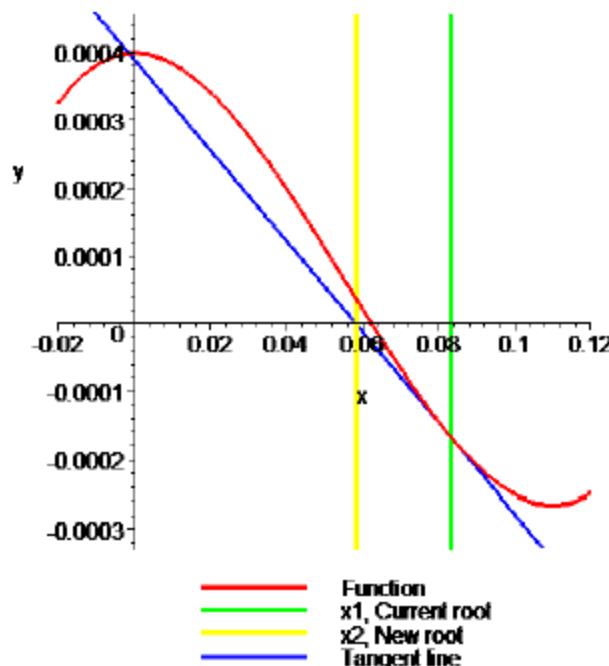
$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

$$x_1 = 0.02 - \frac{3.413 \times 10^{-4}}{-5.4 \times 10^{-3}}$$
$$= 0.08320$$

$$|\epsilon_a| = 75.96\%$$

Iteração 2

Entered function on given interval with current and next root
and tangent line of the curve at the current root



$$x_1 = 0.08320$$

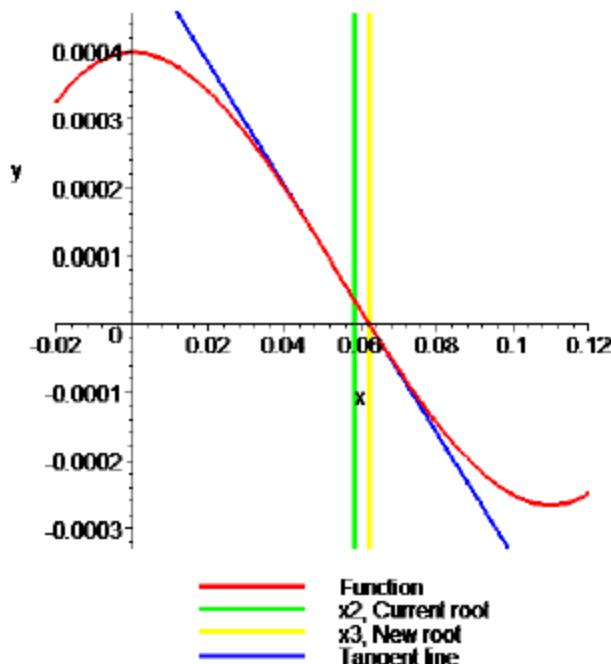
$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

$$x_2 = 0.08320 - \frac{-1.670 \times 10^{-4}}{-6.689 \times 10^{-3}}$$
$$= 0.05824$$

$$|\epsilon_a| = 42.86\%$$

Iteração 3

Entered function on given interval with current and next root
and tangent line of the curve at the current root



$$x_2 = 0.05824$$

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$$

$$= 0.05284 - \frac{3.717 \times 10^{-5}}{-9.043 \times 10^{-3}}$$

$$= 0.06235$$

$$|e_a| = 6.592\%$$

Problem

How quickly does Newton's method converge? How fast does the error decrease from one iteration to the next?

Solution

Let's assume $f''(x)$ exists and $f'(x) \neq 0$ for all x of interest.

- Denote the root of f by x_* ,
- and the error in iteration n by $e_n = x_n - x_*$.

Then

$$\begin{aligned} e_{n+1} &= x_{n+1} - x_* \\ &= x_n - \frac{f(x_n)}{f'(x_n)} - x_* \\ &= e_n - \frac{f(x_n)}{f'(x_n)} \\ &= \frac{e_n f'(x_n) - f(x_n)}{f'(x_n)} \end{aligned} \tag{1}$$

Solution (cont.)

On the other hand, a Taylor expansion gives

$$0 = f(x_*) = f(x_n - e_n) = f(x_n) - e_n f'(x_n) + \frac{e_n^2}{2} f''(\xi)$$

Rearrange:

$$e_n f'(x_n) - f(x_n) = \frac{e_n^2}{2} f''(\xi) \quad (2)$$

(2) in (1):

$$e_{n+1} = \frac{\frac{e_n^2}{2} f''(\xi)}{f'(x_n)}.$$

If x_n is close enough to x_* (so that also ξ is close to x_*) we have

$$e_{n+1} \approx \frac{f''(x_*)}{2f'(x_*)} e_n^2 \implies e_{n+1} = \mathcal{O}(e_n^2).$$

This is known as **quadratic convergence**, and implies that the **number of correct digits approximately doubles** in each iteration.

Vantagens

- Converge rápido, se convergir!!
- Só precisa de uma estimativa inicial

Desvantagens

Pode não convergir.

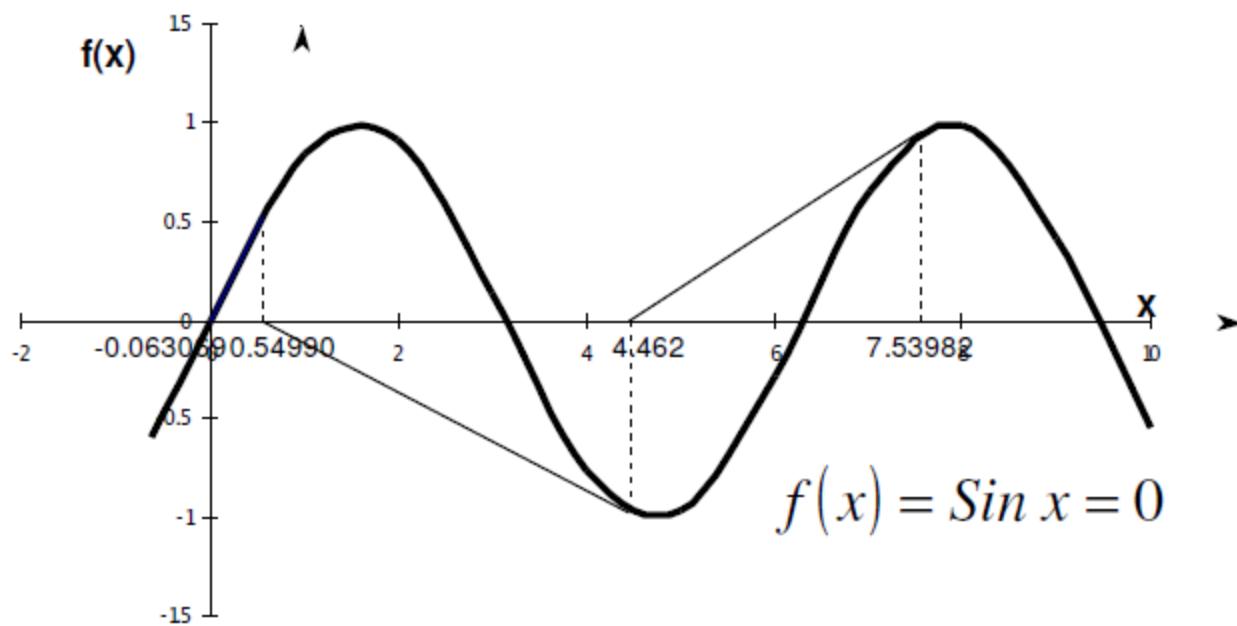
Precisa fornecer a derivada da função
(pode ser um problema se a função for multidimensional)

Condição de convergência

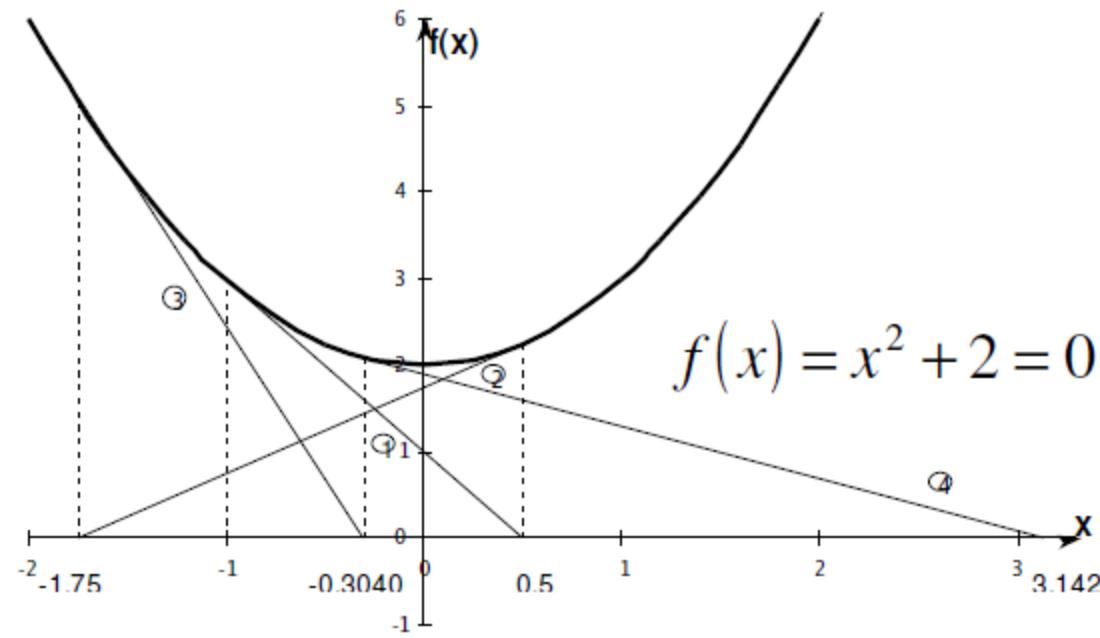
- Uma situação em que é garantido que o método de Newton converge para a raiz da função $f(x)$.
- Seja $f(x)$ uma função definida no intervalo $[a,b]$ tal que exista uma raiz no intervalo e tal que:
 - $f'(x) \neq 0$ para todo x
 - $f''(x)$ não troca de sinal no intervalo
- Então o método de Newton converge para a raiz qualquer que seja o valor inicial.



Situação de não-convergência



Alternância de raízes (root jumping)



Oscilações próximo de máximos ou mínimos locais

Outras desvantagens: divisão por zero ou QUASE zero

