Risoluzione di equazioni e sistemi non lineari

PROBLEMA

Data f:[a,b]$\subseteq\Re\rightarrow\Re$, determinare x*in [a,b] tale che:

$$f(x^*) = 0$$

IL PRESENTE MATERIALE È RISERVATO AL PERSONALE DELL'UNIVERSITÀ DI BOLOGNA E NON PUÒ ESSERE UTILIZZATO AI TERMINI DI LEGGE DA ALTRE PERSONE O PER FINI NON ISTITUZIONALI

Esempi di equazioni non lineari

$$e^x + 1 = 0$$

$$e^{-x} - x = 0$$

$$x^2 - 4\sin(x) = 0$$

$$x^3 + 6x^2 + 11x - 6 = 0$$

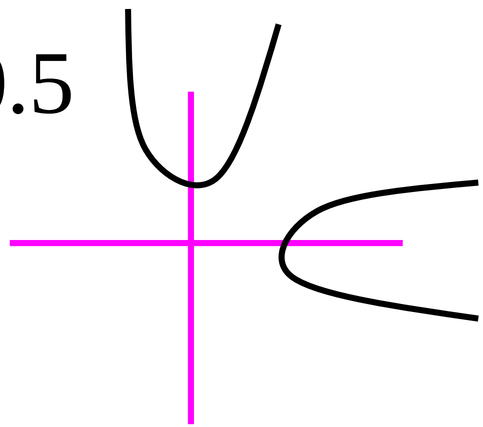
$$\sin(x) = 0$$

- No soluzione
- Una soluzione
- Due soluzioni
- Tre soluzioni
- Infinite soluzioni

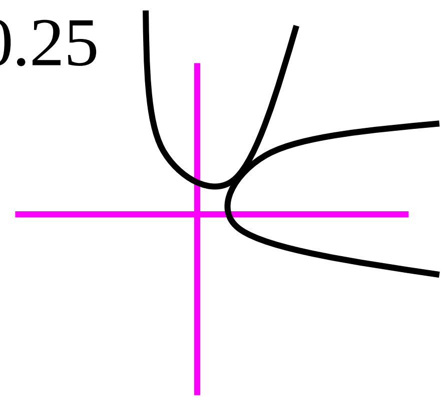
Esempi di
Sistemi di equazioni non lineari

$$f(x)=\begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix}=\begin{bmatrix} x_1^{\ 2}-x_2+\gamma \\ -x_1+x_2^{\ 2}+\gamma \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

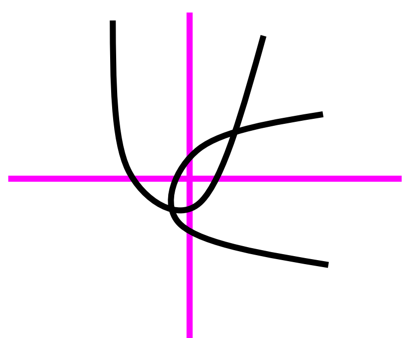
$\gamma = 0.5$



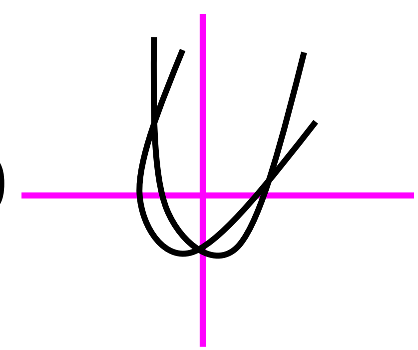
$\gamma = 0.25$



$\gamma = -0.5$



$\gamma = -1.0$



Considerazioni generali

- Le radici di un'equazione non lineare

$$f(x)=0$$

non possono in generale venire espresse in "forma chiusa" e anche quando cio' e' possibile la corrispondente espressione puo' risultare molto complessa

- Si ricorre a metodi numerici iterativi approssimanti

Metodi Numerici Iterativi per la soluzione di equazioni non lineari:

- **Metodo di Bisezione**
- **Metodo delle secanti**
- **Metodo di Newton**
- **Metodo di regula falsi**

$$f(x^*) = 0$$

Si costruisce una successione di valori x_k tali che

$$\lim_{k\to\infty} x_k = x^*$$

Condizionamento del problema

- Problema matematico: determinare x* tale che $f(x^*)=0$
- Problema perturbato (lavorando in aritmetica finita):

Data una perturbazione sui dati: funzione **εg** tale che **f_e=f+εg**

determinare x_e=x*+h tale che

$$f_e(x^*+h)=0$$

h è la perturbazione sui risultati.

Sviluppando con Taylor:

$$f(x^*+h)+\varepsilon g(x^*+h)=0$$

$$\left[\underbrace{f(x^*)}_{=0}+hf'(x^*)+\cancel{\frac{1}{2}h^2f''(\xi)}\right]+\varepsilon\left[g(x^*)+\cancel{hg'(x^*)}+\cancel{\frac{1}{2}h^2g''(\eta)}\right]=0$$

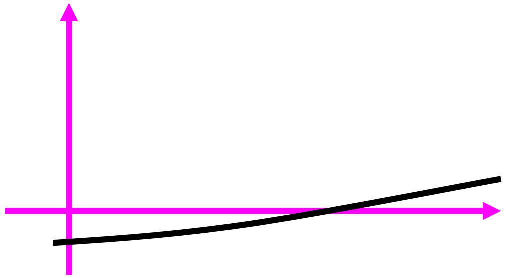
$$\boxed{h\approx-\varepsilon\frac{g(x^*)}{f'(x^*)}}$$

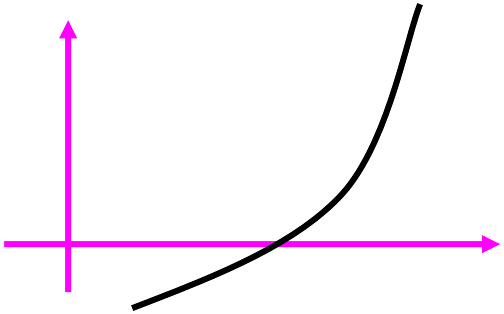
Condizionamento del problema

$$h \approx -\varepsilon g(x^*) \boxed{\frac{1}{f^{'}(x^*)}}$$

Indice di condizionamento del problema

Se f'(x*) molto piccolo, vicino allo zero, allora il problema è mal condizionato, viceversa, il problema risulta ben condizionato e $f_e(x)=0$ ha una radice che non differisce troppo da x*.

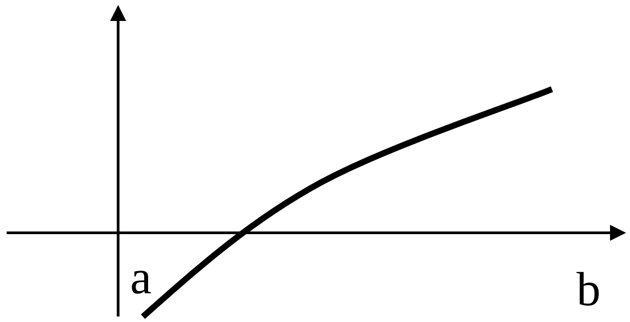


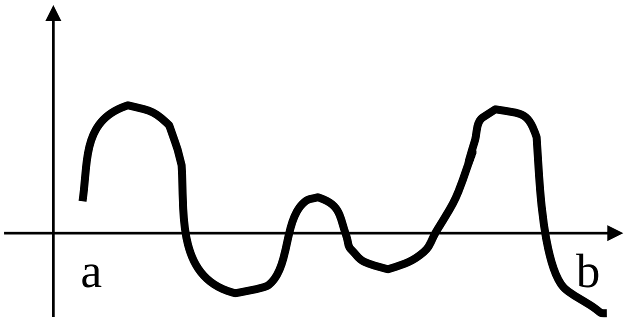


Teorema degli zeri per le funzioni continue

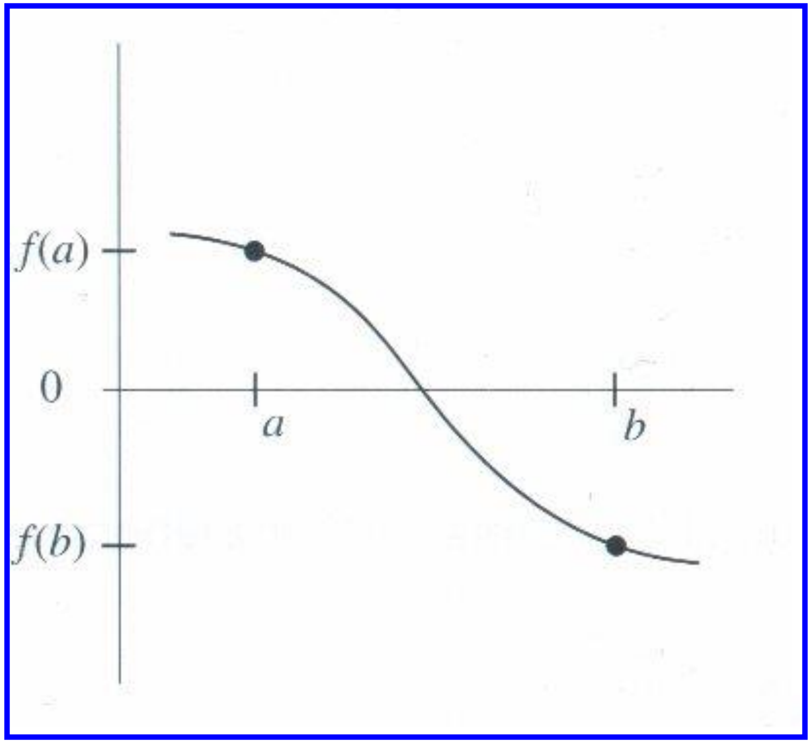
Data una funzione **continua** $f:[a,b]\rightarrow\Re$ e tale che $f(a)*f(b)<0$,

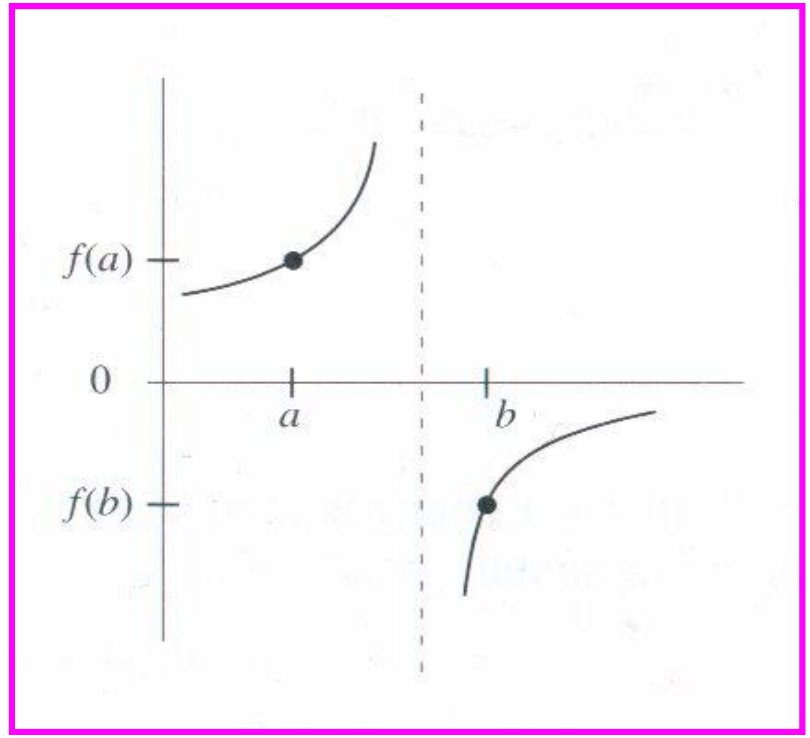
allora esiste **almeno** una soluzione $x^*\in[a,b]$ tale che $f(x^*)=0$





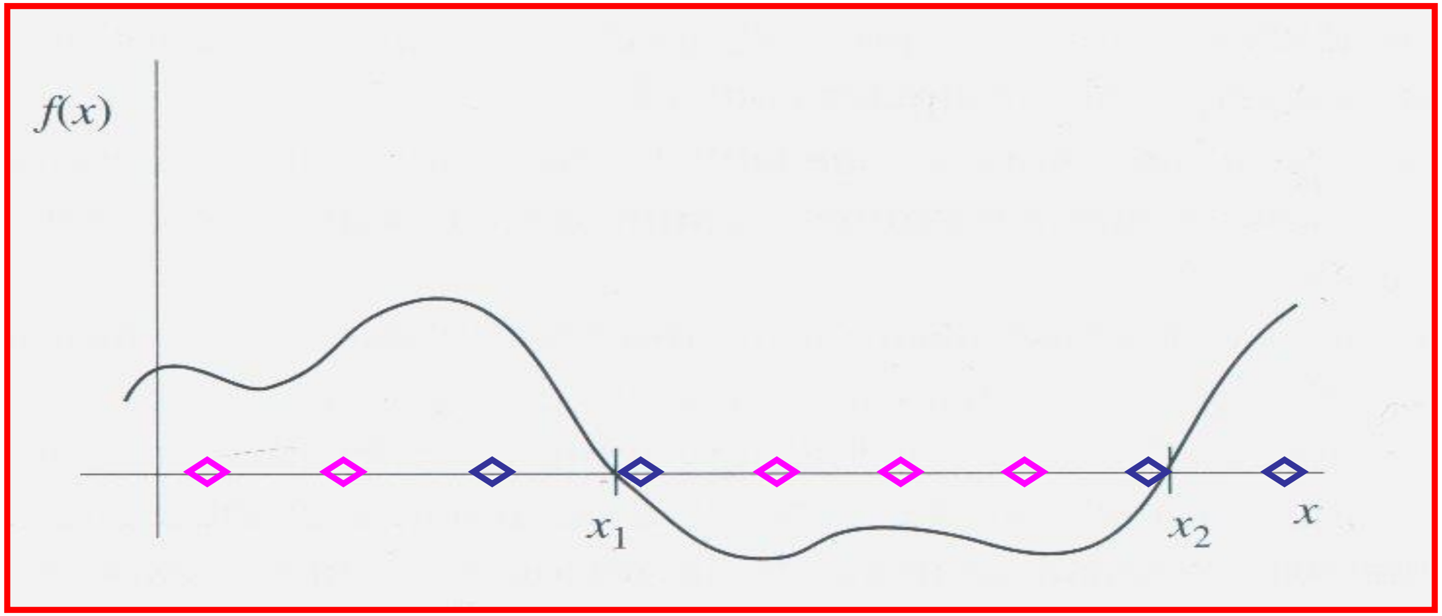
La funzione ha delle singolarità?





Alcune procedure per il calcolo delle radici convergono sia a singolarità che a radici. Questo deve essere prevenuto.

Localizzare le radici



- Localizzare le radici: determinare il numero delle soluzioni e *separare* ogni soluzione, cioè individuare, per ogni soluzione, un intervallo che non ne contenga altre. (discretizzando l'intervallo iniziale)
- Applicare per ogni intervallo determinato un metodo iterativo fino alla convergenza ad una soluzione (radice).

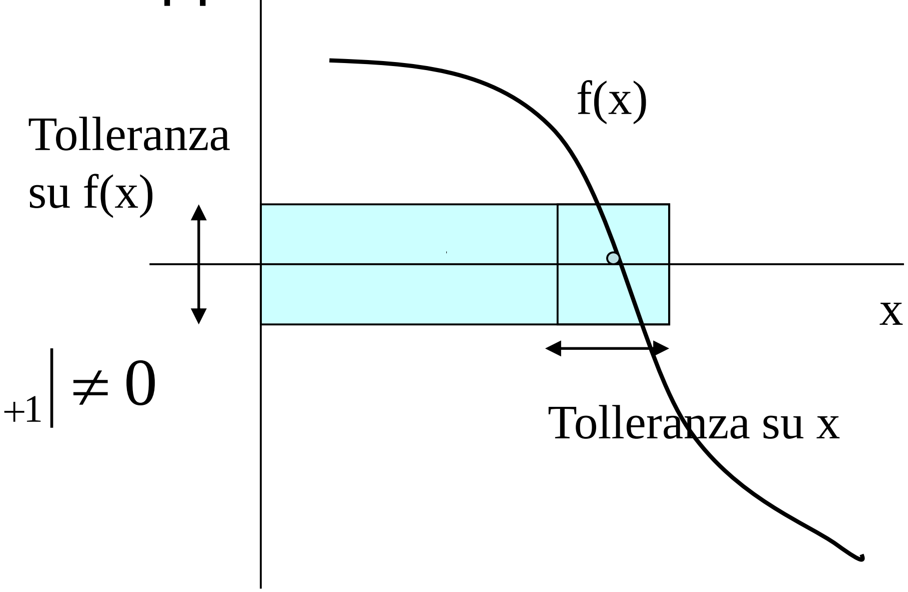
Criteri di arresto

Date due tolleranze $\varepsilon_1 > 0, \varepsilon_2 > 0$

1. Variazione tra due successive approssimazioni della radice:

$$\left| x_{k+1} - x_k \right| < \varepsilon_1$$

$$\frac{\left| x_{k+1} - x_k \right|}{\min(\left| x_k \right|, \left| x_{k+1} \right|)} < \varepsilon_1 \qquad \left| x_k \right|, \left| x_{k+1} \right| \neq 0$$



2. Valore della funzione:

$$\left| f(x_k) \right| < \varepsilon_2$$

3. Numero massimo iterazioni: *nmax*

Metodo di bisezione in [a,b]

- **$a_0 = a$, $b_0 = b$**. Si individua un intervallo **$[a_0 , b_0]$** contenente la radice x* e tale che **$f(a_0)\, f(b_0)<0$**
- Si calcola il punto medio di tale intervallo **$x_m = (a_0 + b_0)/2$**

- Se **$f(a_0)\, f(x_m)<0$** si prosegue con l'intervallo **$[a_0 , x_m]$**
- Se **$f(b_0)\, f(x_m)<0$** si prosegue con l'intervallo **$[x_m , b_0]$**
- Se **$f(x_m)=0$** **x_m** è la radice cercata

Il procedimento definisce una successione di intervalli $[a_i, b_i]$ contenenti x*, ciascuno di lunghezza metà del precedente.

Quindi la successione dei x_{mi} converge ad x*.

Metodo di bisezione in [a,b]

```
Do while |b - a| >= tolerance value

     mid = (a+b)/2;
     fmid = fname(mid); % evaluate function f at mid

     IF fa of opposite sign of fmid
          % radice in [a,mid]
          set b  = mid;     fb = fmid;
       ELSE
          % radice in [mid,b]
          set a  = mid;     fa = fmid;
     END if
  END loop
radice=(b+a)/2
```

Metodo di bisezione: velocità di convergenza

- Si genera una successione di intervalli $[a_k, b_k]$
- Dopo k passi otteniamo un intervallo $[a_k, b_k]$ contenente la radice cercata di ampiezza

$$b_k - a_k = \frac{b_{k-1} - a_{k-1}}{2} = \ldots = \frac{b_0 - a_0}{2^k}$$

$$x_m = \frac{1}{2}(a_k + b_k) \qquad \textbf{Stima della radice}$$

$$x^* = x_m \pm e_{k+1} \qquad \textbf{Errore assoluto } e_{k+1}$$

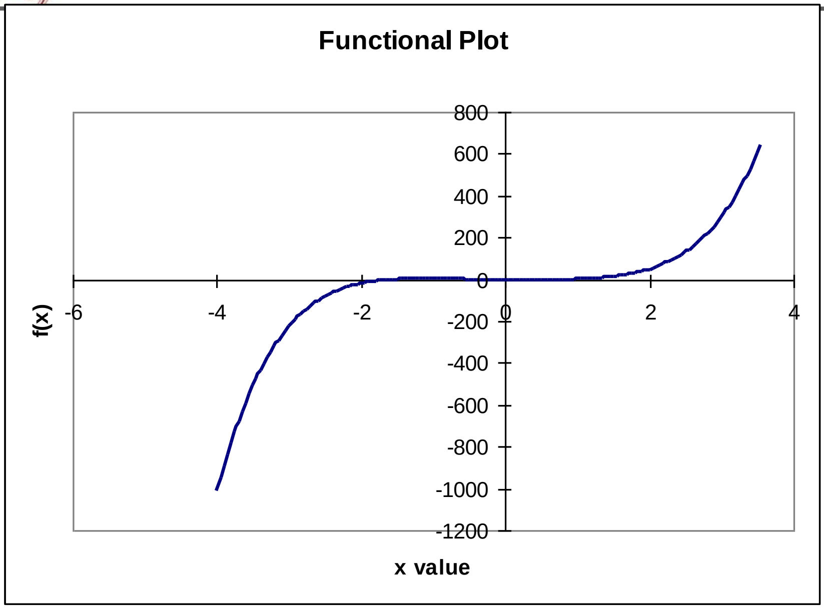
$$|e_{k+1}| \leq \frac{b_k - a_k}{2} = \frac{b_0 - a_0}{2^{k+1}}$$

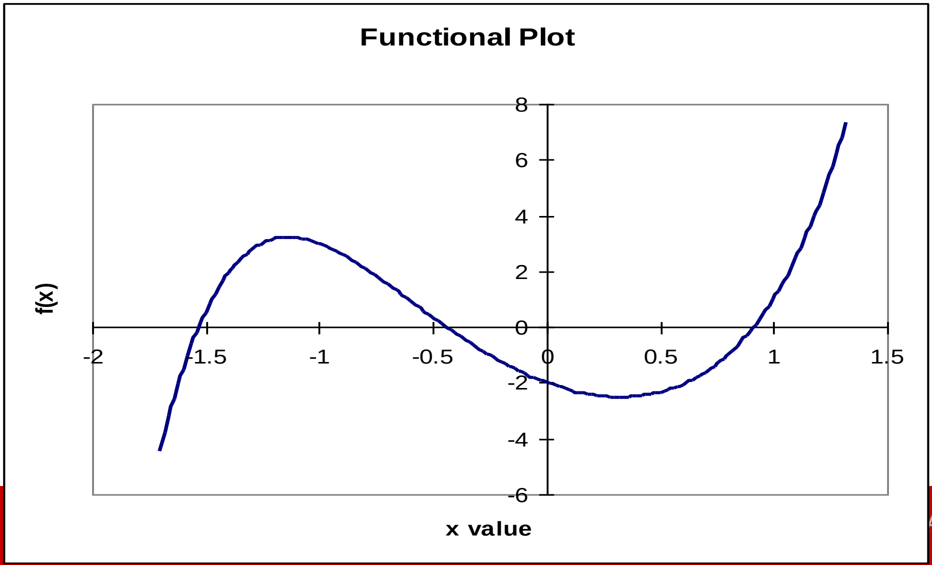
$$da \quad cui \quad lim_{k \to \infty} |e_k| = 0$$

- Sempre convergente ma convergenza lenta.

  Ad ogni passo “si guadagna” una cifra binaria ma $10^{-1} \cong 2^{-3.3}$

Esempio: Metodo di bisezione





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

Ci sono 3 radici

(a) $-2 < x < -1$

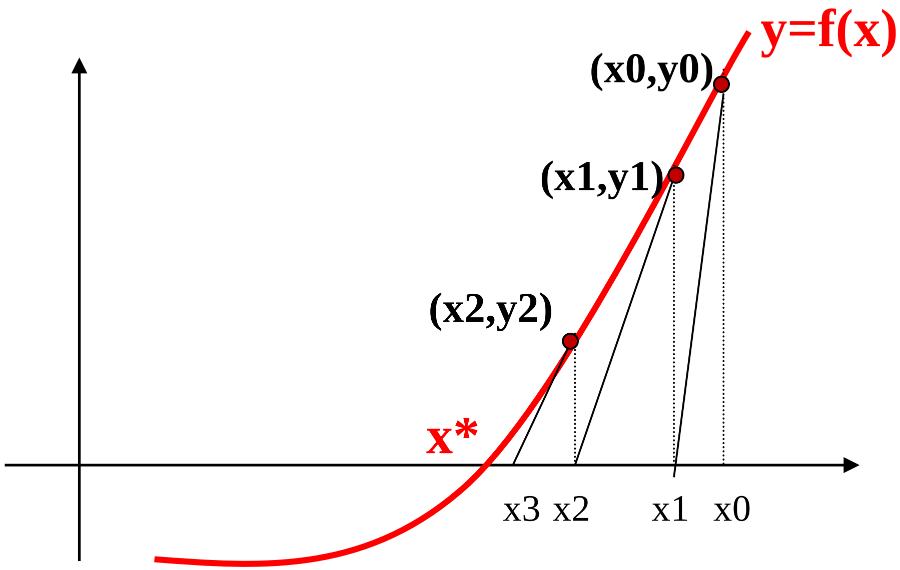
(b) $-1 < x < 0$

(c) $0.5 < x < 1.5$

Nel caso di un numero dispari di radici il metodo di bisezione determina un'approssimazione di una sola di esse.

Idea

Partendo da un'approssimazione iniziale x_0, generiamo i valori successivi nel modo seguente



Ad ogni iterazione sostituiamo la funzione f non lineare con una funzione lineare più semplice: una retta con pendenza k_k passante da $(x_k,f(x_k))$:

$$y - f(x_k) = k_k(x - x_k) \quad k = 0,1,2,..$$

come nuova approssimazione x_{k+1} della radice x* si calcola l'intersezione esatta di tale retta con y=0

$$f(x_k) + k_k(x - x_k) = 0 \quad k = 0,1,2,..$$

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

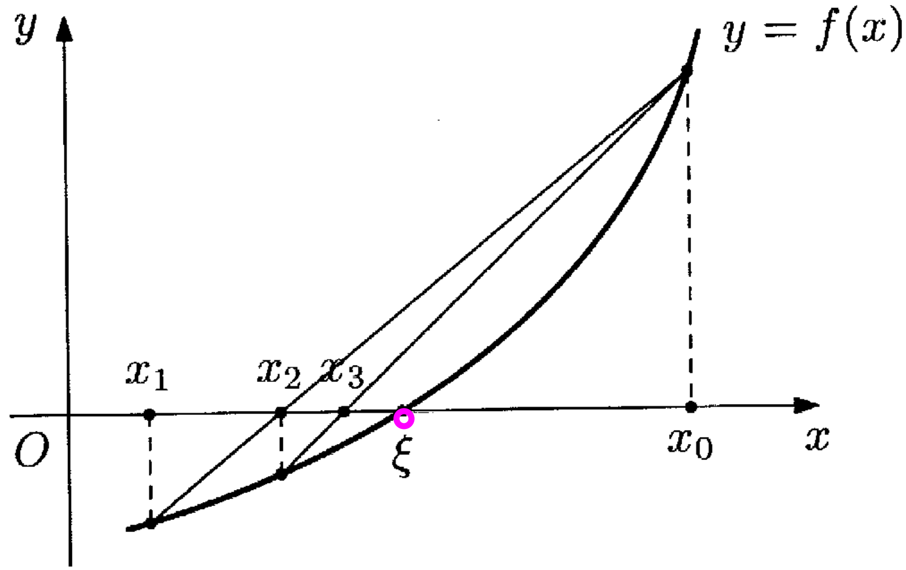
I coefficienti angolari k possono essere scelti in vari modi.

Metodo di Regula Falsi

Come approssimazione della funzione si considera la retta per i punti $(x_k,f(x_k)),(x_n,f(x_n))$ con $n<k$ massimo indice tale che $f(x_k)*f(x_n)<0$

$$\frac{x - x_k}{x_n - x_k} = \frac{y - f(x_k)}{f(x_n) - f(x_k)}$$

$$k_k = \frac{f(x_k) - f(x_n)}{x_k - x_n}$$



che interseca l'asse x nel punto di ascissa

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

Metodo di Regula Falsi

- Si individua un intervallo $[x_1 , x_2]$ tale che $f(x_1)*f(x_2)<0$
- Si costruisce la retta passante per $\mathbf{x_1}$ e $\mathbf{x_2}$:

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

- Si ricava $\mathbf{x_3}$ come intersezione della retta con l'asse x:

$$x_3 = x_2 - \frac{f(x_2)}{f(x_2) - f(x_1)}(x_2 - x_1)$$

- Si valuta il segno di $\mathbf{f(x_3)}$, il punto $\mathbf{x_3}$ sostituisce $\mathbf{x_1}$ o $\mathbf{x_2}$ in base alla concordanza di segno di $f(x_1)$ e $f(x_2)$ con $f(x_3)$.
- In questo modo la radice è sempre racchiusa nell'intervallo $\mathbf{[x_1, x_2]}$

Metodo di Regula Falsi

```
Do while |x2 - x1| >= tolerance value 1
             or |f(x3)|>= tolerance value 2

  Set x3 = x2 - f(x2)*(x2 - x1)/(f(x2)-f(x1))

  IF f(x3) of opposite sign of f(x1);
     Set x2 = x3;
  ELSE
     Set x1 = x3;
  ENDIF
END loop
```

Metodo delle Secanti

Simile al Regula Falsi ma ogni volta si procede con gli ultimi due punti trovati in successione senza tener conto del valore positivo o negativo della funzione.

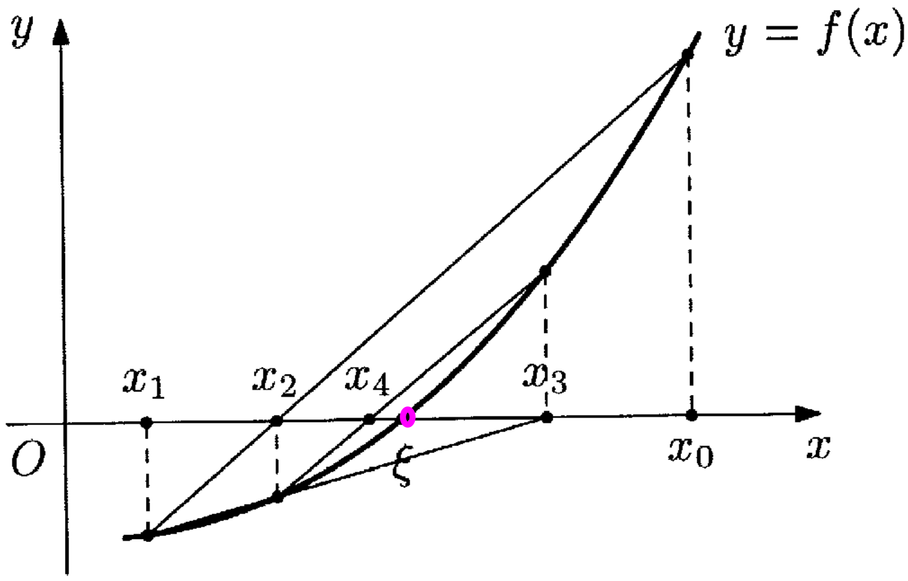
Assegnati i due valori iniziali x_0, x_1 , al passo k l'approssimazione della funzione f nell'intervallo $[x_{k-1}, x_k]$ è la retta per i punti

$(x_{k-1}, f(x_{k-1})), (x_k, f(x_k))$

con coefficiente angolare

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

che interseca l'asse x nel punto di ascissa



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

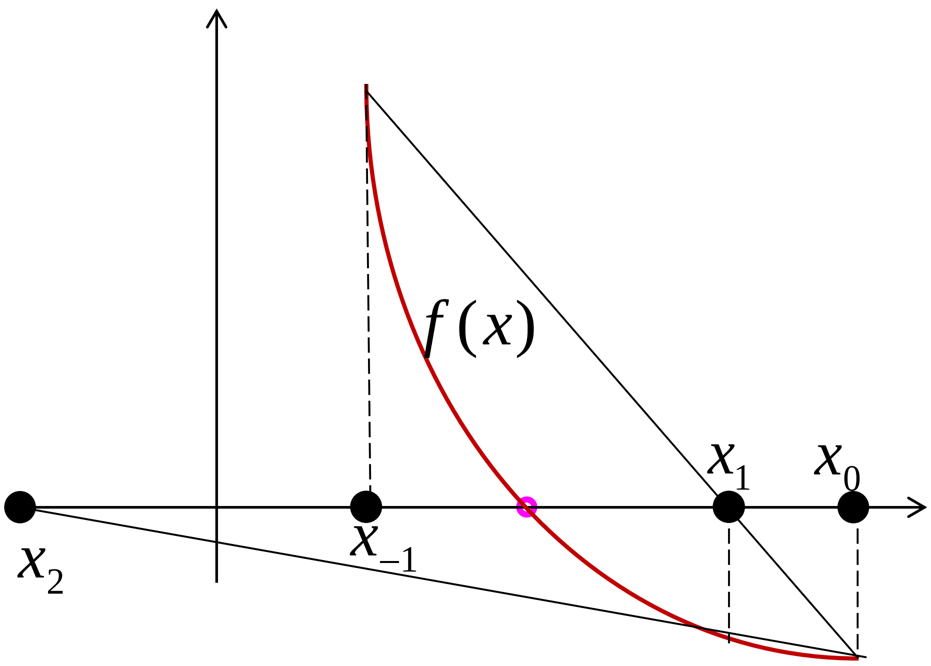
Algoritmo delle Secanti

```
function [x,cont]= secanti(fun,x0,x1,ep)
% INPUT
% fun     puntatore alla funzione non lineare
% x0,x1   vettori contenenti le approssimazioni iniziali
% ep      parametro di tolleranza per l'errore
% OUTPUT:
% x    vettore soluzione dell' equazione non lineare
% cont numero di iterazioni per ottenere l'approssimazione desiderata
x=x1;cont=0;x1=x0;
y1=fun(x1);
while (abs(x-x1) > ep) & (cont<100)
   x0=x1;  y0=y1;   x1=x;
   y1=fun(x1);
   if (abs(y1)>ep)
      x=x1-y1*(x1-x0)./(y1-y0);
      cont=cont+1;
   end
end
if cont==100
   disp('Il procedimento non converge con la precisione desiderata');
end
```

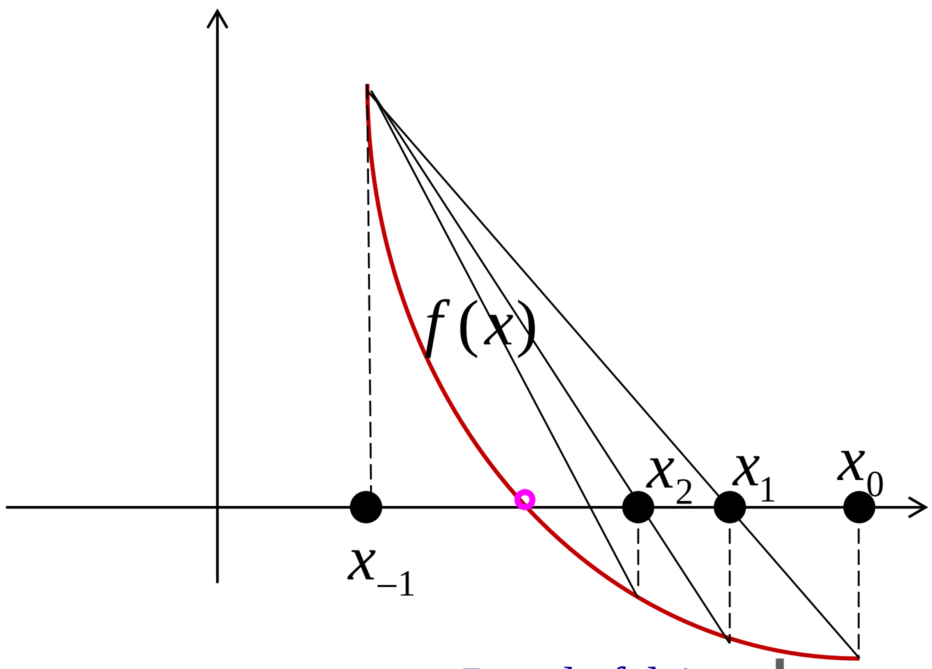
Confronto tra i metodi Regula Falsi e Secanti

Il metodo delle secanti può essere più veloce ma non converge sempre.

Non c'è più la certezza di avere sempre il punto cercato all'interno dell'intervallo.



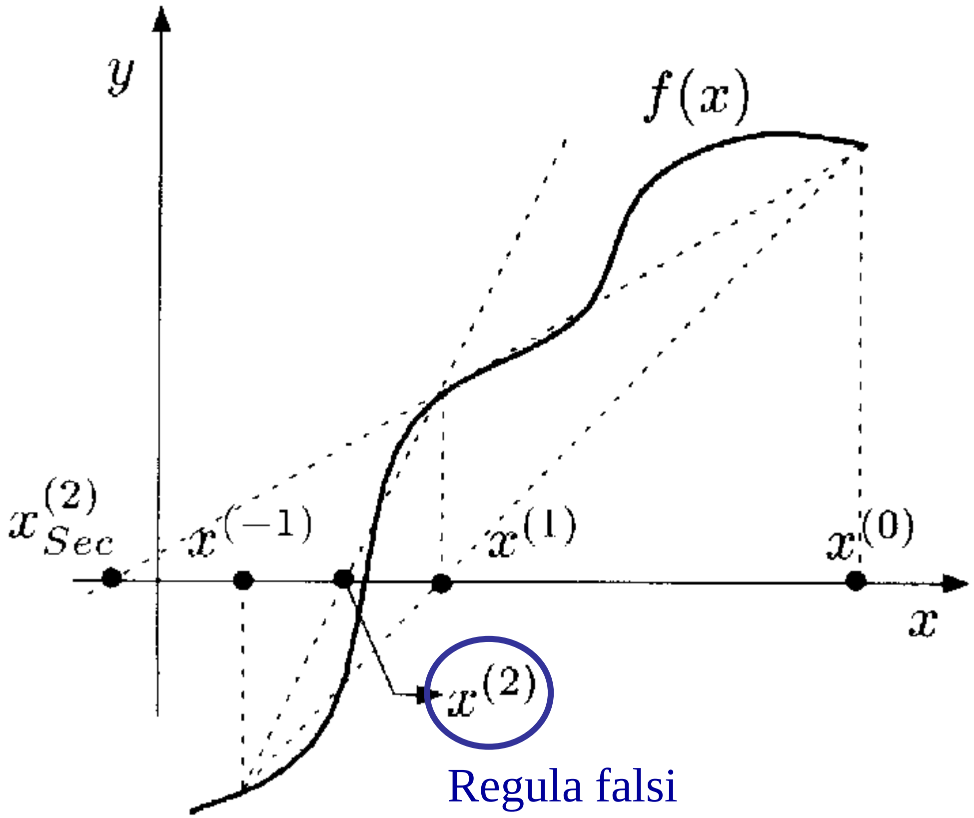
secanti



Regula falsi

Confronto tra i metodi Regula Falsi e Secanti

Secanti



Regula falsi

Metodo di Newton

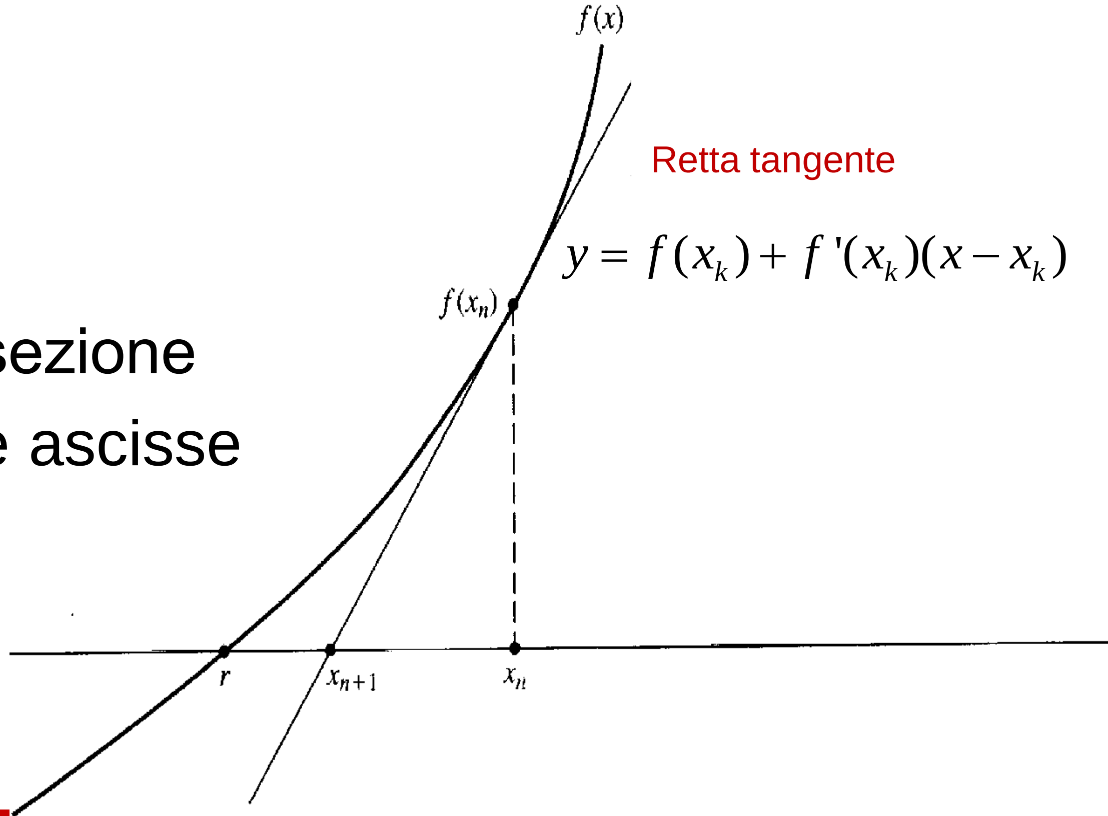
Isaac Newton (1642-1727)

Partendo da una stima iniziale x_0 della soluzione si genera una successione $\{x_k\}$ approssimando ad ogni passo k la curva f(x) mediante la retta tangente ad f nel punto $(x_k, f(x_k))$

$$k_k = f'(x_k)$$

e calcolando x_{k+1} come l'intersezione della tangente con l'asse delle ascisse

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



Metodo di Newton o delle tangenti

L'idea viene dagli sviluppi in serie di Taylor centrata in x_k, dove si conosce la funzione e la sua derivata prima:

$$f(x_{k+1}) = f(x_k) + (x_{k+1} - x_k) f'(x_k) + ...$$

Poichè lo scopo è quello di avere f(x)=0, poniamo

$f(x_{k+1}) = 0$ e tralasciamo i termini di ordine superiore:

$$0 \approx f(x_k) + (x_{k+1} - x_k) f'(x_k) + ...$$

$$x_{k+1} \cong x_k - \frac{f(x_k)}{f'(x_k)}$$

Metodo di Newton o delle tangenti

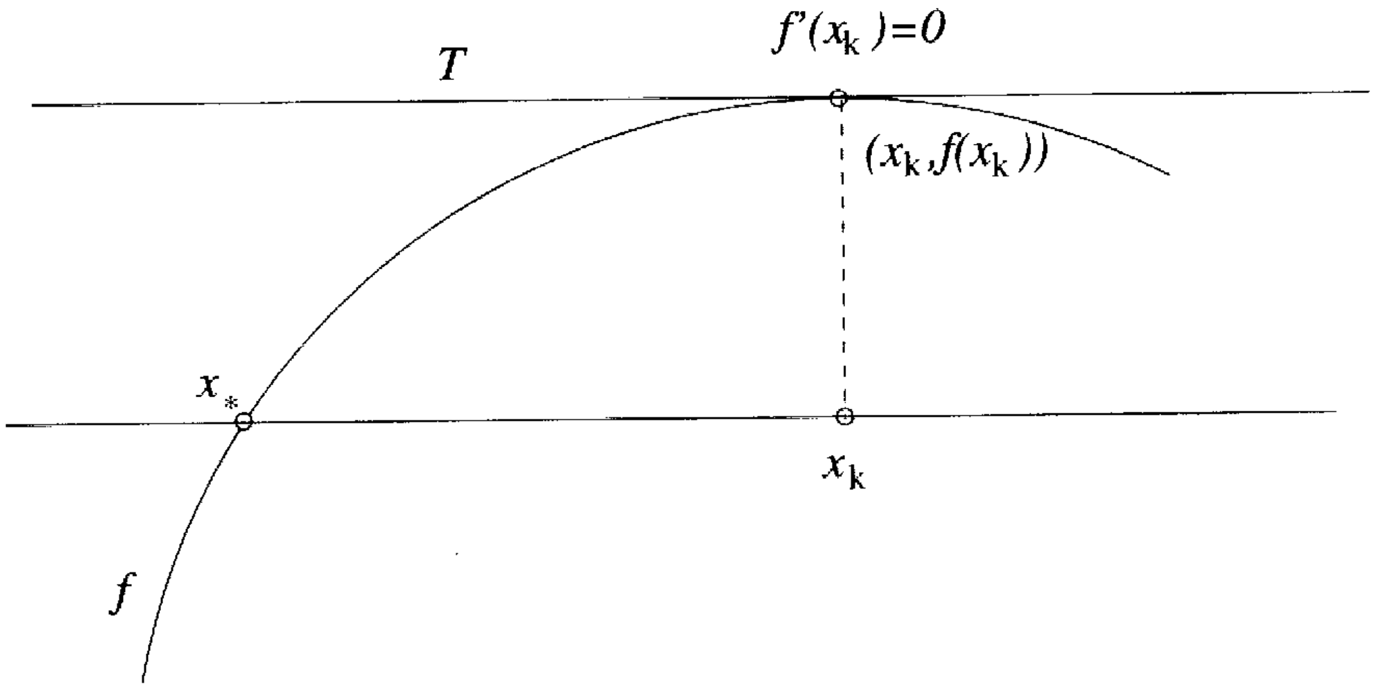
```
function [x,cont]= newton(fun,jac,x0,ep)
% INPUT fun, jac  puntatori rispettivamente alla funzione non lineare
%       e alla matrice Jacobiana della funzione
% x0   vettore contenente l'approssimazione iniziale della soluzione
% ep   parametro di tolleranza per l'errore
% OUTPUT:
% x    vettore soluzione del sistema (o equazione) non lineare
% cont numero di iterazioni per ottenere l'approssimazione desiderata

y=jac(x0)\fun(x0);
x=x0-y;
cont=1;
while (norm(x-x0,'inf') > ep) & (cont<100)
   x0=x;
   y=jac(x0)\fun(x0);
   x=x0-y;
   cont=cont+1;
end
if cont==100
   disp('Il procedimento non converge con la precisione desiderata ');
end
```

Metodo di Newton o delle tangenti

Il metodo genera la successione

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, se\ f'(x_k) \neq 0, k = 0,1,2,..$$

Problemi con il metodo di Newton



Ordine (velocità) di convergenza

Sia $\{x_k\}$ una successione convergente a x* e sia $x_k \neq x^*$ per ogni k.

Se esiste un numero reale $\boldsymbol{p \geq 1}$ tale che:

$$\boxed{\lim_{k \to \infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^p} = \gamma}$$

si dice che la successione ha

ordine di convergenza p e **fattore di convergenza γ**

Un metodo iterativo è convergente di ordine p se tale è la successione da esso generata

$se \;\; p = 1 \quad occorre\ che \quad 0 < \gamma \leq 1$ per avere convergenza lineare

Tanto più piccolo è γ

tanto migliore è la convergenza

Significato del concetto di ordine di convergenza

$$\boxed{e_k = x_k - x^* \qquad |e_k| \le \frac{1}{2}10^{-n}}$$

x_k ha n decimali corretti

$$|e_{k+1}| \cong \gamma\left(\frac{1}{2}10^{-n}\right)^p = \frac{\gamma}{2^p}10^{-pn}$$

X_{k+1} ha pn decimali corretti

Il numero di decimali corretti tende ad essere moltiplicato per p ad ogni passo solo per $k \to \infty$

Per valori finiti di k (e soprattutto nei primi passi) l'aumento di cifre corrette dipende anche dalla costante

$$\gamma_k : \qquad |e_{k+1}| = \gamma_k |e_k|^p \qquad \lim_{k\to\infty} \gamma_k = \gamma$$

Ordine del metodo di Newton: p=2

Nell'ipotesi x* radice semplice: $f(x^*) = 0 \qquad f'(x^*) \neq 0$

$$Posto: \quad e_k = x_k - x^*$$

$$f(x^*) = 0 = f(x_k) + (x^* - x_k) f'(x_k) + \frac{(x^* - x_k)^2}{2} f''(\xi)$$

$$dividendo \quad per \quad f'(x_k)$$

$$\frac{f(x_k)}{f'(x_k)} + x^* - x_k + \frac{\frac{(x^* - x_k)^2}{2} f''(\xi)}{f'(x_k)} = x^* - x_{k+1} + \frac{\frac{(x^* - x_k)^2}{2} f''(\xi)}{f'(x_k)} = 0$$

$$e_{k+1} = \frac{e_k^{\,2}}{2} \frac{f''(\xi)}{f'(x_k)} \quad \Rightarrow \quad \boxed{\frac{e_{k+1}}{e_k^{\,2}} \xrightarrow[x_k \to x^*]{} \frac{1}{2} \frac{f''(x^*)}{f'(x^*)}}$$

2° ordine

Ordine di convergenza

p=1	$0 < \gamma < 1$	**convergenza lineare**
p=1	$\gamma = 1$	**convergenza sublineare**
1<p<2		**convergenza superlineare**
p=2		**convergenza quadratica**
p=3		**convergenza cubica**

Ordine dei metodi

Sia x* radice semplice $f(x^*) = 0$ $f'(x^*) \neq 0$

Metodo di Newton o delle tangenti **convergenza quadratica p=2**

Metodo delle secanti **convergenza superlineare**

$$p = \frac{1+\sqrt{5}}{2} \sim 1.618$$ sezione aurea

Metodo regula falsi **convergenza superlineare**

Metodo di bisezione **convergenza lineare p=1**

Esempio: problema di Archimede

$$f(x)=0 \qquad f(x) = x^3 - 3x^2 + 1 \quad \text{in } [0,1]$$

metodo bisezione

k	x_k	$f(x_k)$
1	0.50000	3.7500×10^{-1}
2	0.75000	-2.6563×10^{-1}
3	0.62500	7.2266×10^{-2}
4	0.68750	-9.3018×10^{-2}
5	0.65625	-9.3689×10^{-3}
6	0.64063	3.1712×10^{-2}
7	0.64844	1.1236×10^{-2}
8	0.65234	9.4932×10^{-4}
9	0.65430	-4.2058×10^{-3}
10	0.65332	-1.6273×10^{-3}

Secanti

k	x_k	$f(x_k)$
1	0.50000	3.7500×10^{-1}
2	0.63636	4.2825×10^{-2}
3	0.65130	3.7093×10^{-3}
4	0.65259	3.1166×10^{-4}
5	0.65269	2.6116×10^{-5}

Arresto con

Tol=10^{-3} sull' ampiezza dell'intervallo.

Newton (x_0 = 1)

k	x_k	$f(x_k)$
1	0.66667	3.7037×10^{-2}
2	0.65278	1.9558×10^{-4}
3	0.65270	5.7248×10^{-9}

Se però scegliamo x0 = 0.1 si ottiene x1 = 1.8035 che è fuori dall'intervallo [0,1]. Infatti con questa scelta di x0 la successione ottenuta ha come limite il punto -0.53209, che è un'altra soluzione della stessa equazione, ma inaccettabile per il nostro problema che richiede soluzioni in [0,1].

Scelta dell’iterato iniziale

- Metodi a convergenza locale
  - La convergenza è assicurata per x_0 appartenente ad un intorno della soluzione.

    (Secanti, Newton)
- Metodi a convergenza globale
  - La convergenza è assicurata pe qualsiasi scelta del punto iniziale appartenente all’intervallo che racchiude la radice, cioè x_0 in [a,b]

    (Bisezione, Regula Falsi)

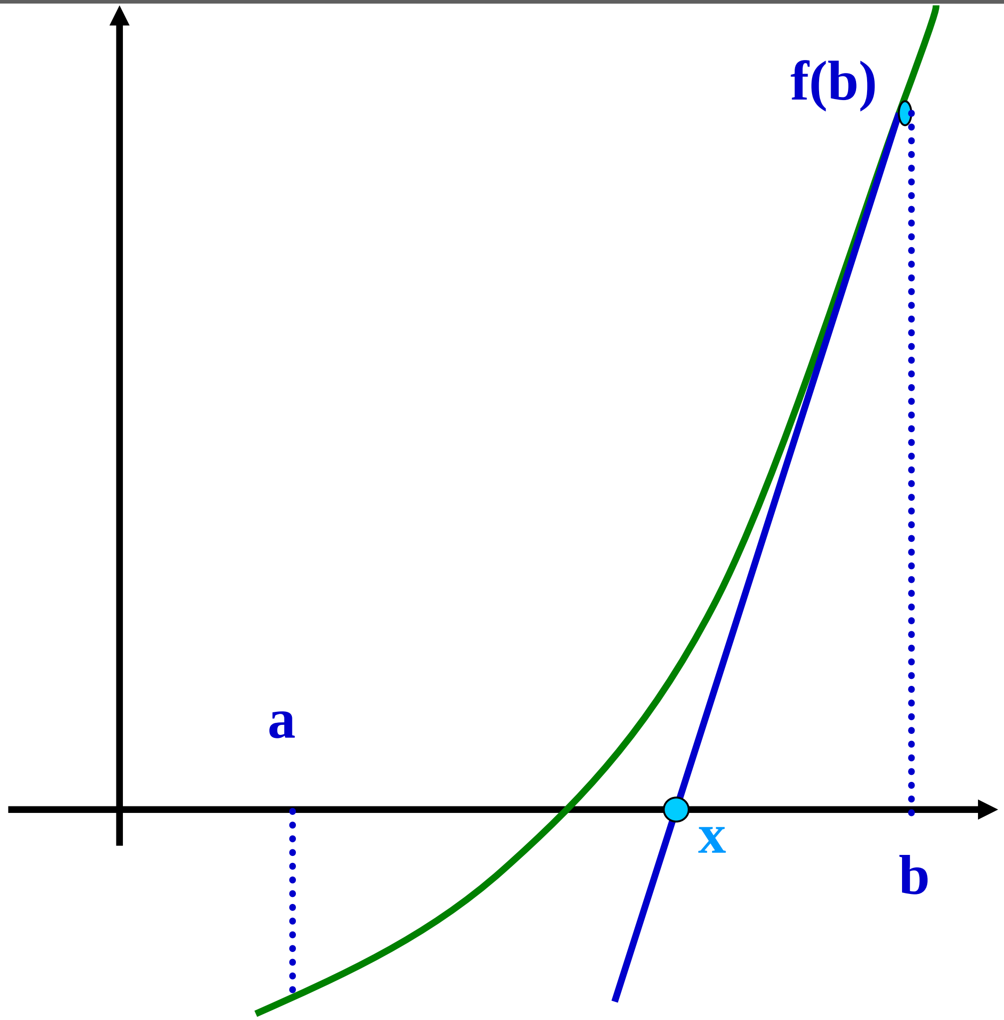
Teorema di Convergenza globale

Sia x* uno zero semplice di f:[a, b] → R. Si supponga inoltre che $f(x) \in C^2[a,b]$ Se

1. $f(a)f(b) < 0$
2. $f'(x) \neq 0 \qquad \forall x \in [a,b]$
3. $f''(x) > 0 \qquad oppure \qquad f''(x) < 0 \qquad \forall x \in [a,b]$
4. $\left|\frac{f(a)}{f'(a)}\right| < b-a \qquad \left|\frac{f(b)}{f'(b)}\right| < b-a$

allora il metodo di Newton **converge** all'unica soluzione $x^* \in [a,b]$, **per ogni scelta di x_0 in [a,b].**

Ipotesi 4 del Teorema di Convergenza



$$\begin{cases} y - f(b) = f'(b)(x-b) \\ y = 0 \end{cases}$$

x interno ad [a,b]

$$\left|\frac{f(b)}{f'(b)}\right| = |x-b| < b-a$$

La tangente negli estremi interseca l'asse x all'interno dell'intervallo [a,b]

Sistemi di equazioni non lineari

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

Data $F : R^n \to R^m$ calcolare $x^* \in R^n$ tale che

$$F(x^*) = 0,$$

$$x = (x_1, x_2, .., x_n)^T, \quad F = (f_1, f_2, .., f_m)^T$$

Preliminari

- $f : R^n \to R$ è continuamente differenziabile se

  $\dfrac{\partial f}{\partial x_i}$ esiste ed è continua per i=1,2,..,n

- Il gradiente di f in x è dato da

$$\nabla f(x) = \begin{bmatrix} \dfrac{\partial f(x)}{\partial x_1} \\ \dfrac{\partial f(x)}{\partial x_2} \\ .... \\ \dfrac{\partial f(x)}{\partial x_n} \end{bmatrix}$$

Preliminari

- $F : R^n \rightarrow R^m$ funzione a valori vettoriali

$$F : \begin{bmatrix} x_1 \\ x_2 \\ \cdots\cdot \\ x_n \end{bmatrix} \rightarrow \begin{bmatrix} f_1(x) \\ f_2(x) \\ \cdots\cdot \\ f_m(x) \end{bmatrix}$$

- Derivata di F in x o Jacobiano è la matrice

$$J(x_k) = \begin{bmatrix} \left.\frac{\partial f_1}{\partial x_1}\right|_{x=x_k} & \left.\frac{\partial f_1}{\partial x_2}\right|_{x=x_k} & \cdots\cdot & \left.\frac{\partial f_1}{\partial x_n}\right|_{x=x_k} \\ \left.\frac{\partial f_2}{\partial x_1}\right|_{x=x_k} & \left.\frac{\partial f_2}{\partial x_2}\right|_{x=x_k} & \cdots\cdot & \left.\frac{\partial f_2}{\partial x_n}\right|_{x=x_k} \\ \cdots\cdot & \cdots\cdot & \cdots\cdot & \cdots\cdot \\ \left.\frac{\partial f_m}{\partial x_1}\right|_{x=x_k} & \left.\frac{\partial f_m}{\partial x_2}\right|_{x=x_k} & \cdots\cdot & \left.\frac{\partial f_m}{\partial x_n}\right|_{x=x_k} \end{bmatrix}$$

- Gradiente di F in x è la matrice: $\nabla F(x) = J(x)^T$

Metodo di Newton-Raphson

Algoritmo

Dato $x_0 \in R^n$ e F, per ogni iterazione k :

0. Valutare $J(x_k)$
1. Risolvere il sistema lineare

$$J(x_k)s_k = -F(x_k)$$

2. Porre

$$x_{k+1} = x_k + s_k$$

Il metodo ha convergenza *locale* quadratica

(se x_0 è sufficientemente vicino alla soluzione)

Varianti del Metodo di Newton-Raphson

La valutazione dello Jacobiano richiede di conoscere o poter valutare n^2 derivate parziali.

<u>Alcune varianti al metodo per migliorarne l'efficienza:</u>

1. Approssimazione con rapporti incrementali:

$$\frac{\partial f_j}{\partial x_i}\bigg|_{x=x_k} \approx (J^k)_{ij} = \frac{f_j(x_k + e_i s_{ij}) - f_j(x_k)}{s_{ij}}$$

e_i **i – esimo vettore della base canonica** R^n

s_{ij} **incrementi scelti ad ogni passo** k

Il metodo che si ottiene è l'analogo n-dimensionale di quello delle secanti

Varianti del Metodo di Newton-Raphson

2. Metodo della corda:

si utilizza il medesimo Jacobiano o una sua approssimazione $J(x_0)$ o $A(x_0)$ per tutte le iterazioni k. Si potrebbe quindi fattorizzare $J(x_0)=LU$ e utilizzare i medesimi L ed U per ogni iterazione

3. Metodo di Shamanskii

Si valuta lo Jacobiano ogni m iterazioni, e quindi lo si utilizza per le m iterazioni successive:

$$J^{k+l} = J^k \quad l = 1,..,m$$

Giunti ad x_{k+m+1} si rivaluta lo Jacobiano,..

Minimizzazione di una funzione

Problema di ottimizzazione non vincolato

Data $f : R^n \rightarrow R,\ f \in C^2$ due volte continuamente differenziabile, trovare $x \in R^n$ tale che

$$\min_{x \in R^n} f$$

$$x = (x_1, x_2, .., x_n)^T,$$

Se f è $C^1(R^n)$ i punti di stazionarietà locale x* (massimi, minimi, sella) sono soluzione del seguente sistema (non lineare):

$$\nabla f(x^*) = 0$$

Minimizzazione di una funzione

Applichiamo il metodo di Newton-Raphson al sistema non lineare:

$$\nabla f(x) = 0 \quad \Rightarrow \quad \begin{cases} \dfrac{\partial f(x)}{\partial x_1} = 0 \\ \quad \dots \\ \dfrac{\partial f(x)}{\partial x_n} = 0 \end{cases}$$

Per verificare poi se tale punto è un massimo, un minimo oppure un punto di sella, occorrerà in generale esaminare la matrice hessiana $H(x) = \nabla^2 f(x)$

$$(H(x))_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}, \quad i, j = 1,..,n$$

Metodo di Newton-Raphson per MINIMIZZAZIONE

Algoritmo

Dato $x_0 \in R^n$ e f, per ogni iterazione k :

1. Valutare $\nabla^2 f(x_k)$
1. Risolvere il sistema lineare

$$\nabla^2 f(x_k)s_k = -\nabla f(x_k)$$

2. Porre

$$x_{k+1} = x_k + s_k$$

s_k definisce una direzione di discesa da x_k a x_{k+1}