lunedì 4 aprile 2016 08:42

funzione o comando FIND( )

Ha in ingresso un vettore o una matrice e dà in uscita le posizioni dove ci sono elementi diversi da 0.

(•) find([-1 0 1 0]) = [1 3]

(•) find([1 0 0 4; 2 3 0 0]) = [1 2 4 7]

Il find può essere combinato con altre operazioni.

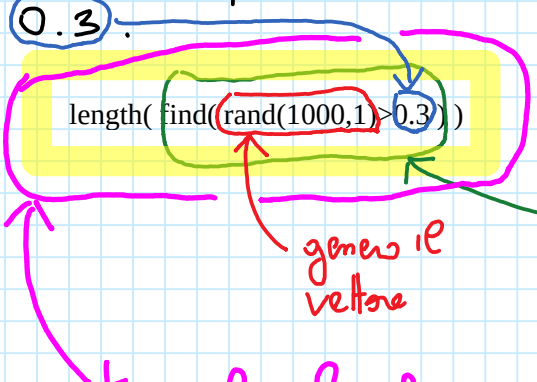
A = [1 0 0 4; 2 -3 0 0]

find(A > 0) = [1 2 7]

→ [1 0 0 1; 1 0 0 0]

La funzione V=rand(m,n) con m,n numeri naturali genera una variabile v (matrice m x n) con elementi distribuiti in modo uniforme nell'intervallo [0, 1].

ESERCIZIO Creare un vettore v di 1000 elementi con distribuzione uniforme e determinare quanti sono quelli che superano la soglia >= 0.3.



ossia trovo quanti elementi superano la soglia di 0.3.

lunedì 4 aprile 2016 08:42

PROGRAMMAZIONE

scelte : IF (help if)

```
if condizione logica
   istruzioni
end
```

```
if condizione logica
   istruzioni
else istruzioni
end
```

cicli
- FOR
- WHILE

Vedo FOR.

```
for k = vettore
   istruzioni
end
```

k assume, uno alla volta TUTTI gli elementi del vettore. Se vettore è vuoto il ciclo non è eseguito.

ESEMPIO

Trovare il massimo di un vettore v.

v = [1 -2 -1 (3) 0 1]

```
maxv ← v(1)
for k = 2:length(v)      = [2 3 4 5 6]
   if v(k) > maxv
      maxv ← v(k)
   end
end
```

file massimo.m

```
% definisco v
v = [1 -2 7 -2 5 1];

% trovo il massimo di v
maxv = v(1);
for k=2:length(v)
  if v(k)>maxv
    maxv = v(k);
  end
end
% metto a monitor maxv
maxv
```

va salvato in un file.
Eseguo i comandi scrivendo nella command Window il nome del file (senza il .m)

lunedì 4 aprile 2016 08:42

Il file di testo che contiene i comandi Matlab per trovare il massimo di un vettore è detto SCRIPT. Per eseguirlo, basta scrivere il nome del file sulla command window (senza scrivere l'estensione!)

FUNZIONI IN MATLAB

function [$o1, \ldots, om$] = nomeFunction($i1, \ldots, im$)

($o1, \ldots, om$: uscita; $i1, \ldots, im$: ingresso)

istruzioni.

(*) Per una sola uscita, si può omettere le parentesi quadre.

È buona prassi Matlab aggiungere dei commenti:

```
function maxv = maxvet( v )
%MAXVET calcola il massimo di un vettore
%LLLL MAXV = MAXVET(V) ......
% .....
:
% ....

% $ dalé e avanti %

istruzioni
```

questa riga viene usata da lookfor().

```
lookfor massimo
maxvet          MAXVET calcola massimo di un
  MAXV = MAXVET(V vettore
                V) trova il massimo di un ve
  ..............  ettore
>> % trovo il massimo di v
>> maxv = v(1);
error: 'v' undefined near line 2 c
```

lunedì 4 aprile 2016 08:42

```
function maxv = maxvet(v)
%MAXVET calcola massimo di un vettore
%   MAXV = MAXVET(V) trova il massimo di un vettore
%   .............
%   .............

%   $ 04/04/2016 $ Roberto Bertelle

% trovo il massimo di v
maxv = v(1);
for k=2:length(v)
  if v(k)>maxv
    maxv = v(k);
  end
end
```

```
while ( condizione logica )
    istruzioni
end
```

Il ciclo è eseguito finché la condizione logica è vera.
Per evitare i loop infiniti è bene aggiungere alla condizione logica di prima un'altra condizione sul NUMERO MASSIMO di volte che il ciclo può essere eseguito:

```
k = 1;
while ( condizione logica && k <= kmax )
    istruzioni
    k = k+1 ;
end
```

lunedì 4 aprile 2016 08:42

Dimostriamo che Newton con radice NON singola ha ordine di convergenza $p = 1$.

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

Supponiamo che la radice ξ abbia molteplicità $r > 1$; gli sviluppi di Taylor di $f(x_k)$ e $f'(x_k)$ sono

$$f(x_k) = \underbrace{f(\xi)}_{=0} + \underbrace{f'(\xi)}_{=0}(x_k - \xi) + \dots + \underbrace{\frac{f^{(r-1)}(\xi)}{(r-1)!}}_{=0}(x_k - \xi)^{r-1} + \underbrace{\frac{f^{(r)}(\xi)}{r!}}_{\neq 0}(x_k - \xi)^r + o\left((x_k - \xi)^r\right)$$

$$= \frac{f^{(r)}(\xi)}{r!}(x_k - \xi)^r + o\left((x_k - \xi)^r\right) \quad \text{per } x_k \to \xi$$

$$f'(x_k) = \frac{f^{(r)}(\xi)}{\cancel{r!}\,(r-1)!} \cdot \cancel{r} \cdot (x_k - \xi)^{r-1} + o\left((x_k - \xi)^{r-1}\right)$$

Riprendo da $\overset{(*)}{=}$:

$$\overset{(*)}{=} \xi - x_k + \frac{\frac{f^{(r)}(\xi)}{r!}(x_k - \xi)^r + o\left((x_k - \xi)^r\right)}{\frac{f^{(r)}(\xi)}{(r-1)!}(x_k - \xi)^{r-1} + o\left((x_k - \xi)^{r-1}\right)}$$

$$= \frac{\frac{f^{(r)}(\xi)}{(r-1)!}(x_k - \xi)^{r-1}(\xi - x_k) + \frac{f^{(r)}(\xi)}{r!}(x_k - \xi)^r + o\left((x_k - \xi)^r\right)}{\frac{f^{(r)}(\xi)}{(r-1)!}(x_k - \xi)^{r-1} + o\left((x_k - \xi)^{r-1}\right)}$$

$$= \frac{f^{(r)}(\xi)(x_k - \xi)^r \cdot \left(\frac{1}{r!} - \frac{1}{(r-1)!}\right) + o\left((x_k - \xi)^r\right)}{(x_k - \xi)^{r-1}\left[\frac{f^{(r)}(\xi)}{(r-1)!} + \frac{o\left((x_k - \xi)^{r-1}\right)}{(x_k - \xi)^{r-1}}\right]}$$

$$\frac{1}{r!} - \frac{1}{(r-1)!} = \frac{1 - r}{r!}$$

$$f^{(\tau)}(\xi)(x_n-\xi)^{\tau}\cdot\left(\frac{1}{\tau!}-\frac{1}{(\tau-1)!}\right)+o\left((x_n-\xi)^{\tau}\right)$$

$$=\frac{}{(x_n-\xi)^{\tau-1}\left[\frac{f^{(\tau)}(\xi)}{(\tau-1)!}+\frac{o\left((x_n-\xi)^{\tau-1}\right)}{(x_n-\xi)^{\tau-1}}\right]}$$

$$=\frac{(x_n-\xi)^{\tau}}{(x_n-\xi)^{\tau-1}}\cdot\frac{\frac{f^{(\tau)}(\xi)}{\tau!}(1-\tau)+\frac{o\left((x_n-\xi)^{\tau}\right)}{(x_n-\xi)^{\tau}}}{\frac{f^{(\tau)}(\xi)}{(\tau-1)!}+\frac{o\left((x_n-\xi)^{\tau-1}\right)}{(x_n-\xi)^{\tau-1}}}$$

$\to 0$ per $n\to+\infty$; $\to+\infty$

$$\to\frac{f^{(\tau)}(\xi)\cdot(\tau-1)!\cdot(1-\tau)}{\tau!\cdot f^{(\tau)}(\xi)}\cdot(x_n-\xi)$$

$$=\frac{1-\tau}{\tau}(x_n-\xi)=\frac{\tau-1}{\tau}(\xi-x_n)$$

$\xi-x_n=e_n$

Allora è

$$\lim_{n\to+\infty}\left|\frac{e_{n+1}}{e_n}\right|=\lim_{n\to+\infty}\left[\frac{\tau-1}{\tau}\frac{e_n}{e_n}\cdot\left|\frac{1+\dots}{1+\dots}\right|\right]=\frac{\tau-1}{\tau}$$

($\frac{1+\dots}{1+\dots}\to 1$)

$$=1-\frac{1}{\tau}$$

lunedì 4 aprile 2016 08:42

TEST DI ARRESTO

Un buon test di arresto per il metodo di Newton è

$$|x_{n+1} - x_n| < \text{toll}$$

con toll numero piccolo (esempio $\text{toll} = 10^{-6}$).

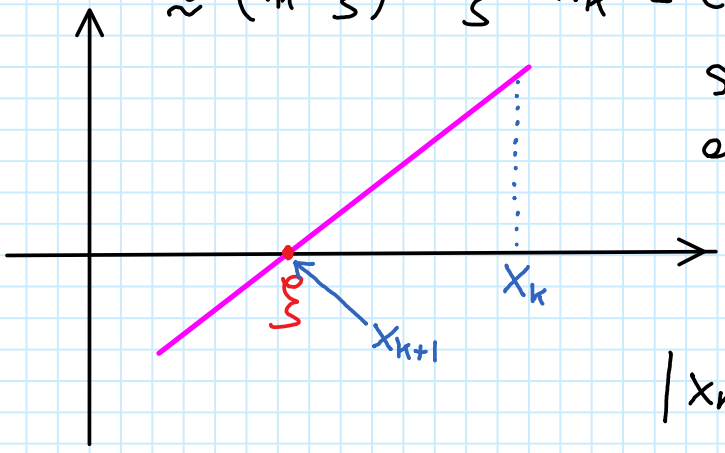
Vediamo di capire il legame tra questo test di arresto e l'errore $e_n = \xi - x_n$.

Ho due casi.

(1) $f'(\xi) \neq 0$. Ho $x_{n+1} - x_n \approx e_n$ perché

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

$$\approx -(x_n - \xi) = \xi - x_n = e_n$$

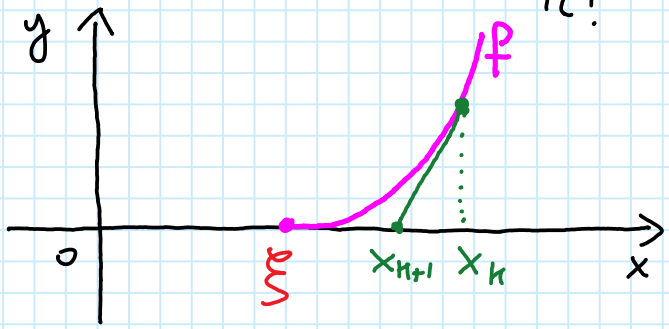


Se $x_n \approx \xi$ e $f'(\xi) \neq 0$ allora posso approssimare $f(x)$ con una retta $\Rightarrow$ $x_{n+1} \approx \xi \Rightarrow$

$$|x_{n+1} - x_n| \approx |\xi - x_n|$$

(2) $f'(\xi) = 0, \ldots, f^{(r-1)}(\xi) = 0, f^{(r)}(\xi) \neq 0$

Vicino a ξ ho $f(x) \cong \frac{f^{(r)}(\xi)}{r!}(x - \xi)^r$



In questo caso NON è più vero che $|x_{n+1} - x_n| \approx |e_n|$. Sorge il sospetto, dalla figura, che $|x_{n+1} - x_n| < |e_n|$

lunedì 4 aprile 2016 08:42

Questo significa che se $|x_{n+1} - x_n| < tol$, può essere $|e_n| > tol$. Vediamo il perché.

$$x_{n+1} - x_n = -\frac{f(x_n)}{f'(x_n)} = -\frac{\frac{f^{(r)}(\xi)}{r!}(x_n-\xi)^r + \dots}{\frac{f^{(r)}(\xi)}{(r-1)!}(x_n-\xi)^{r-1} + \dots}$$

(↑ sviluppi)

$$\approx -\frac{(r-1)!}{r!}(x_n - \xi) = \frac{1}{r} e_n$$

$$\Rightarrow \boxed{|e_{n+1}| \approx r \cdot |x_{n+1} - x_n|}$$

OSS. Conoscendo l'errore $|e_n|$ tramite le relazioni precedenti posso stimare l'errore $|e_{n+1}|$:

$$\frac{|e_{n+1}|}{|e_n|^p} \approx M \Rightarrow |e_{n+1}| \approx M \cdot |e_n|^p$$

OSS. Posso stimare M dalle iterate:

$$M \approx \frac{|e_{n+1}|}{|e_n|^p} \approx \frac{|x_{n+2} - x_{n+1}|}{|x_{n+1} - x_n|^p}$$

molto utile per i programmi di calcolatore!

lunedì 4 aprile 2016 08:42

METODI QUASI-NEWTON

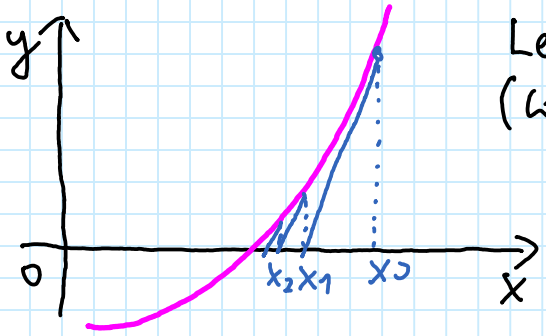
Questi metodi nascono dalla necessità di NON calcolare la $f'(x)$.

METODO TANGENTE FISSA

$$\begin{cases} x_0 \text{ dato (vicino a } \xi) \\ x_{k+1} = x_k - \dfrac{f(x_k)}{f'(x_0)} \quad , \quad k = 0, 1, 2, 3, \dots \end{cases}$$

($f'(x_0)$ costante)

A livello grafico ho



Le rette sono TUTTE parallele tra loro! (con coefficiente angolare $= f'(x_0)$)

Si dimostra, al solito modo, che si tratta di un metodo di ordine $p = 1$. Per provarlo, si scrive

$$e_{k+1} = \xi - x_{k+1} = \xi - x_k + \frac{f(x_k)}{f'(x_0)} \overset{\text{sviluppi in serie}}{=} \dots$$

Si trova anche che la costante asintotica dell'errore è

$$M = \left| 1 - \frac{f'(\xi)}{f'(x_0)} \right|$$

per cui il metodo converge se $M < 1$.

lunedì 4 aprile 2016 08:42

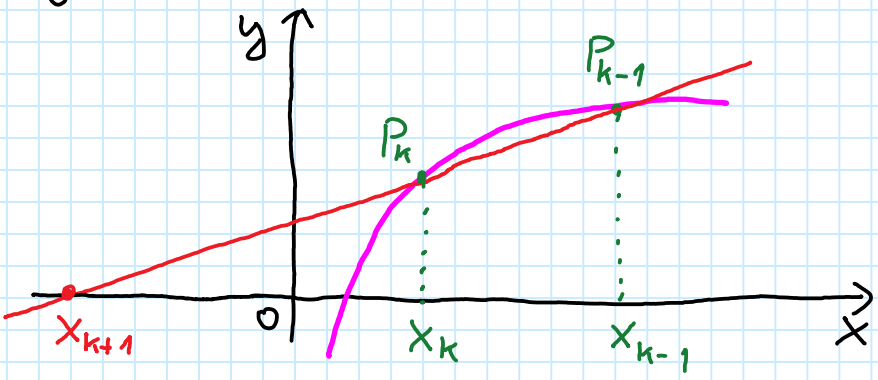
SECANTE VARIABILE

x_0, x_1 dati (vicini a ξ)

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

$$= x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}, \quad n = 1, 2, 3, \ldots$$

Si può dimostrare che c'è la seguente interpretazione geometrica:



Si dimostra, e non lo facciamo, che, nell'ipotesi $f'(\xi) \neq 0$, il metodo ha

(•) $p = \frac{1+\sqrt{5}}{2} \approx 1.6$

(•) $M = \left(\frac{1}{2} \frac{f''(\xi)}{f'(\xi)} \right)^{\frac{p}{p+1}} \approx 0.618$

lunedì 4 aprile 2016 08:42

ESERCIZIO Sia $f(x) = x - \sin(x)$.

(a) Dimostrare che $\xi = 0$ è radice di f. È l'unica?

(b) Partendo da $[a_0, b_0] = [-0.5, 1]$ calcolare le prime tre iterazioni x_0, x_1, x_2 con il metodo di bisezione. Dare una stima di quante iterazioni servono col metodo per avere un errore $|x_n - \xi| < 10^{-6}$.

(c) Determinare la molteplicità di ξ e stabilire l'ordine di convergenza di Newton. Trovare la costante asintotica dell'errore. Calcolare le prime tre iterate x_1, x_2, x_3 a partire da $x_0 = 1$.

(d) Stimare la costante asintotica dell'errore usando le iterate.

(e) Stabilire quante iterazioni sono necessarie a Newton per avere un errore $|e_n| < 10^{-9}$.

(f) Qual è il grafico qualitativo di $\log_{10}(|e_n|)$?

(g) Come va modificato Newton per ripristinare l'ordine di convergenza $p = 2$?

Calcolare le prime tre iterate x_1, x_2, x_3 e partire da $x_0 = 1$ e spiegare i risultati.

(h) Applicare il metodo delle secanti variabili per calcolare x_2 e x_3 a partire da $x_0 = 1$, $x_1 = 0.5$.