Metodi Numerici con elementi di Programmazione

A.A. 2013-2014

Introduzione al MatLab
II parte

Docente: Vittoria Bruni

Email: vittoria.bruni@sbai.uniroma1.it

Ufficio: Via A. Scarpa,
Pal. B, I piano, Stanza n. 16
Tel. 06 49766648

Ricevimento: Giovedì 14.00-15.00

Il materiale didattico è disponibile sul sito http://ingaero.uniroma1.it/ nella pagina dedicata al corso Metodi Numerici con elementi di Programmazione

Per consultazione: Getting Started with MatLab – The mathworks
www.mathworks.com

Alcune operazioni

Gli operatori + - * / si applicano direttamente a vettori e matrici quando le dimensioni sono conformi

Somma di vettori o matrici

>> D = A + B

A e B devono avere la stessa dimensione!!!

D ha la stessa dimensione di A e B

Differenza di matrici o matrici

>> D = A - B

A e B devono avere la stessa dimensione!!!

D ha la stessa dimensione di A e B

Alcune operazioni

Esempio: Siano v = [0 -4 5 7] e w = [1;9;2;0].

>> v+w   non è corretto in quanto v è un vettore riga mentre w è un vettore colona

>> v+w'   è corretta

```
>> y=v+w'
y =
     1     5     7     7
```

Esempio: Siano A = [0 -4; 5 7] e B = [1 9; 2 0].

```
>> C = A - B
C =
    -1   -13
     3     7
```

Alcune operazioni

Prodotto righe-colonne di matrici

\>> C = A * B   numero di colonne di A deve essere uguale al numero di righe di B.

C ha dimensione = [num.righe di A, num.colonne di B]

Esempio: Siano A = [0 -4; 5 7] e B = [1 9; 2 0].

```
>> C = A * B
C =
    -8      0
    19     45
```

Esempio: Siano A = [0 -4; 5 7] e B = [1 9 ; 2 0: -3 1].

```
>> C=A*B
```

non è corretta!!!

Alcune operazioni

Esempio: Siano v = [0 -4 5 7] e w = [1;9;2;0].

>> v*w e w*v   **sono corrette**

```
>> v*w

ans =

   -26

>> w*v

ans =

     0    -4     5     7
     0   -36    45    63
     0    -8    10    14
     0     0     0     0
```

>> v*w' e w*v'   **non sono corrette!**

Alcune operazioni

Prodotto per uno scalare: k*v moltiplica ogni elemento del vettore (o matrice) v per lo scalare k

Esempio: Sia v = [4 2 -1]

```
>> 3*v

ans =

    12     6    -3
```

Somma o differenza con uno scalare: v + a somma ad ogni elemento di v (vettore o matrice) il numero a.

```
>> v-2

ans =

     2     0    -3
```

Potenza: A^n (A matrice quadrata, n intero >0) moltiplica A n volte per se stessa.

```
>> A = [4 1; 6 9];

>> A^3

   166   139

   834   861
```

Alcune operazioni

Operatori di divisione / (slash) \ (backslash)

Backlash: x=A\B indica la soluzione dell'equazione Ax=B.

Se B è un vettore, Ax=B è un ordinario sistema di equazioni lineari, ed x sarà la soluzione.

Se invece B è una matrice avente come colonne $b_1, b_2, \ldots b_n$, allora X=A\B sarà a sua volta una matrice, avente come colonne $x_1, x_2, \ldots x_n$, cioè le soluzioni dei sistemi lineari $Ax_1=b_1$, $Ax_2=b_2$, … $Ax_n=b_n$.

Slash: X=B/A indica la matrice tale che XA=B.

Operazioni puntuali

Prodotto elemento per elemento

>> C = A .* B   C(i,j) = A(i,j)*B(i,j)   per tutti gli indici i e j

A e B devono avere la stessa dimensione!!!

C ha la stessa dimensione di A e B

Divisione elemento per elemento

>> C = A./B   C(i,j) = A(i,j)/B(i,j)   per tutti gli indici i e j

A e B devono avere le stesse dimensioni

C ha la stessa dimensione di A e B

Potenza elemento per elemento

>>C = A .^B   C(i,j) = A(i,j)^B(i,j)   per tutti gli indici i e j

Ciascun elemento di A viene elevato a potenza, con esponente l'elemento corrispondente di B

A e B devono avere le stesse dimensioni

C ha la stessa dimensione di A e B

Operazioni puntuali

Esempio: Siano v = [0 -4 5 7] e w = [1;9;2;0].

\>> v.*w' è corretta   >> v.*w non è corretta!

```
>> v.*w'

ans =

     0    -36     10      0
```

Esempio: Siano A = [0 -4; 5 7] e B = [1 9; 2 0].

```
>> C = A * B

C =

    -8     0

    19    45

>> D = A.*B

D =

     0   -36

    10     0
```

Nota: $D \neq C$

Operazioni puntuali

Esempio: Siano $A = [0\ -8;\ 6\ 7]$ e $B = [1\ 4;\ 2\ 1]$.

```
>> D = A./B

D =

     0    -2

     3     7

>> D = A.^2

D =

     0    64

    36    49
```

Esempio: Siano $A = [0\ -8;\ 6\ 7]$ e $v = [1;\ 4]$.

```
>> D = A.*v
```

non è corretta!!!

Operazioni puntuali

Esempio: Siano $A = [0\ -8;\ 6\ 7]$ e $B = [1\ 4;\ 2\ 1]$.

```
>> D = A.^B

D =

     0    4096
    36       7
```

Esempio: Siano $A = [0\ -8;\ 6\ 7]$ e $v = [1;\ 4]$.

```
>> D = A.^v
```

non è corretta!!!

Operazioni puntuali

Le operazioni puntuali (elemento per elemento) non hanno una corrispondente operazione nell'algebra lineare

Sono definite solo per facilitare alcuni calcoli che vanno ripetuti su ogni singola componente di un vettore e/o una matrice

Permettono di scrivere sequenze di istruzioni in forma chiara e compatta, evitando, per esempio, l'uso di cicli

Operazioni puntuali

Esempio: Tabulare i valori della funzione e^x nell'intervallo [0,2]

```
>> x = linspace(0,2,5);
```

Genera un vettore x di 5 elementi equispaziati e contenuti nello intervallo [0,2]

```
>> f = exp(x);
```

Genera un vettore f di 5 elementi corrispondenti al valore della funzione e^x valutata in ogni elemento del vettore x

```
>> disp([x' f' ])
```

Visualizza gli elementi della matrice le cui colonne sono i vettori x ed f

```
      0    1.0000
 0.5000    1.6487
 1.0000    2.7183
 1.5000    4.4817
 2.0000    7.3891
```

Valori della funzione e^x nei punti 0, 1/2, 1, 3/2, 2

Operazioni puntuali

```
>> f = exp(x);
```

La funzione exp applicata al vettore x produce un vettore f della stessa dimensione di x tale che: la componente di indice i di f contiene il valore della funzione esponenziale nel punto x(i), cioè

$$f(i) = e^{x(i)}$$

f = [exp(x(1)), exp(x(2)), exp(x(3)), exp(x(4)),, exp(x(5))]

Operazioni puntuali

```
>> f = exp(x).*sin(x)+1;
```

la componente di indice i di f contiene il valore della funzione $f(x) = e^x \sin(x)+1$ nel punto $x(i)$, cioè

$$f(i) = e^{x(i)} \sin(x(i))+1$$

f = [exp(x(1))sin(x(1))+1, exp(x(2))sin(x(2))+1,....., exp(x(n))sin(x(i))+1]

Matrici

Aggiunta alla matrice A di una colonna: A=[A u] aggiunge ad A il vettore u come ultima colonna

(u deve essere vettore colonna, se u è un vettore riga allora A=[A u'])

Esempio: Sia A=[1 2 0; 3 4 1; 2 6 3]. Aumentarne la dimensione aggiungendo il vettore u = [-1; 4;0] come ultima colonna

```
>> A = [A u]

A =

     1     2     0    -1

     3     4     1     4

     2     6     3     0
```

Matrici

Aggiunta di una riga: A=[A; v] aggiunge ad A il vettore v come ultima riga

Esempio: Sia A=[1 2 0; 3 4 1; 2 6 3]. Aumentarne la dimensione aggiungendo il vettore u = [-1; 4;0] come ultima riga

```
>> A = [A; u']

A =

     1     2     0
     3     4     1
     2     6     3
    -1     4     0
```

Nota: u è definito come vettore colonna!

Separazione delle radici: tabulazione

Esercizio: Separare gli zeri della seguente funzione

$$f(\lambda) = e^{\lambda} + 0.435\,(e^{\lambda} - 1)\,/\,\lambda - 1.564$$

Si calcola il valore della funzione in corrispondenza di valori equidistanti della variabile in un certo intervallo e si osserva se il segno dei valori ottenuti:

```
>> lambda = linspace(0.10,0.20,6);

>>f=exp(lambda)+0.435*(exp(lambda)-1)./lambda- 1.564;

>> disp([lambda'   f'])
   0.10000000000000  -0.00133558829528
   0.12000000000000   0.02567293855461
   0.14000000000000   0.05319595959218
   0.16000000000000   0.08124355150079
   0.18000000000000   0.10982599066618
   0.20000000000000   0.13895375715853
```

L'intervallo [0.10, 0.12] contiene almeno uno zero della $f(\lambda)$

Separazione delle radici: tabulazione

Esercizio: Separare gli zeri della seguente funzione

$$f(\lambda) = e^{\lambda} + 0.435\,(e^{\lambda} - 1)\,/\,\lambda \; - 1.564$$

Si calcola il valore della funzione in corrispondenza di valori equidistanti della variabile in un certo intervallo e si osserva se il segno dei valori ottenuti:

```
>> lambda = (0.10:0.02:0.20);

>>f=exp(lambda)+0.435*(exp(lambda)-1)./lambda- 1.564;

>> disp([lambda'   f'])
   0.10000000000000  -0.00133558829528
   0.12000000000000   0.02567293855461
   0.14000000000000   0.05319595959218
   0.16000000000000   0.08124355150079
   0.18000000000000   0.10982599066618
   0.20000000000000   0.13895375715853
```

L'intervallo [0.10, 0.12] contiene almeno uno zero della $f(\lambda)$

Separazione delle radici: tabulazione

Esercizio: Separare gli zeri della seguente funzione

$$f(x) = log(x+1) + \sqrt{x+2} - 1 = 0$$

```
>>x = linspace(-0.9,0.5,10);
>>f = log(x+1)+sqrt(x+2)- 1;
>> disp([x' f'])
   -0.9000   -2.2538
   -0.7444   -1.2438
   -0.5889   -0.7010
   -0.4333   -0.3163
   -0.2778   -0.0131
   -0.1222    0.2400
    0.0333    0.4587
    0.1889    0.6525
    0.3444    0.8271
    0.5000    0.9866
```

L'intervallo [-0.2778,-0.1222] contiene almeno uno zero della f(x)

Separazione delle radici: tabulazione

Esercizio: Verificare che la funzione $f(x) = x^3 - 10x^2 + 5$ ha uno zero nell'intervallo [0.6,0.8]

```
>>x = 0.6:0.05:0.8;

>>f = x.^3-10*x.^2+5;

>> disp([x'   f'])
    0.6000    1.6160
    0.6500    1.0496
    0.7000    0.4430
    0.7500   -0.2031
    0.8000   -0.8880
```

Esercizi

- Costruire una matrice 5x5, ed assegnarla alla variabile A
- Calcolare il prodotto $A\ A^T$
- Calcolare il prodotto $A\ A^T$ elemento per elemento
- Sommare le righe di A e sia v il vettore risultante
- Sommare le colonne di A e sia w il vettore risultante.
- Eseguire i prodotti possibili tra v e w
- Elevare a potenza gli elementi di w usando gli elementi di v come esponenti e sia y il vettore risultante
- Sia B la matrice i cui vettori riga sono i vettori v, w e y opportunamente dimensionati
- Calcolare il prodotto degli elementi della quarta colonna di B (usare il comando lookfor)

Esercizi

- Costruire la tabella dei valori assunti dalla funzione $\log(x)$ nell'intervallo $[1/2, 3/2]$ usando un passo di discretizzazione pari a 0.025

- Valutare la funzione e^x in 100 valori equidistanti nell'intervallo $[0, 1]$ usando i primi cinque termini dello sviluppo in serie di Taylor attorno allo zero

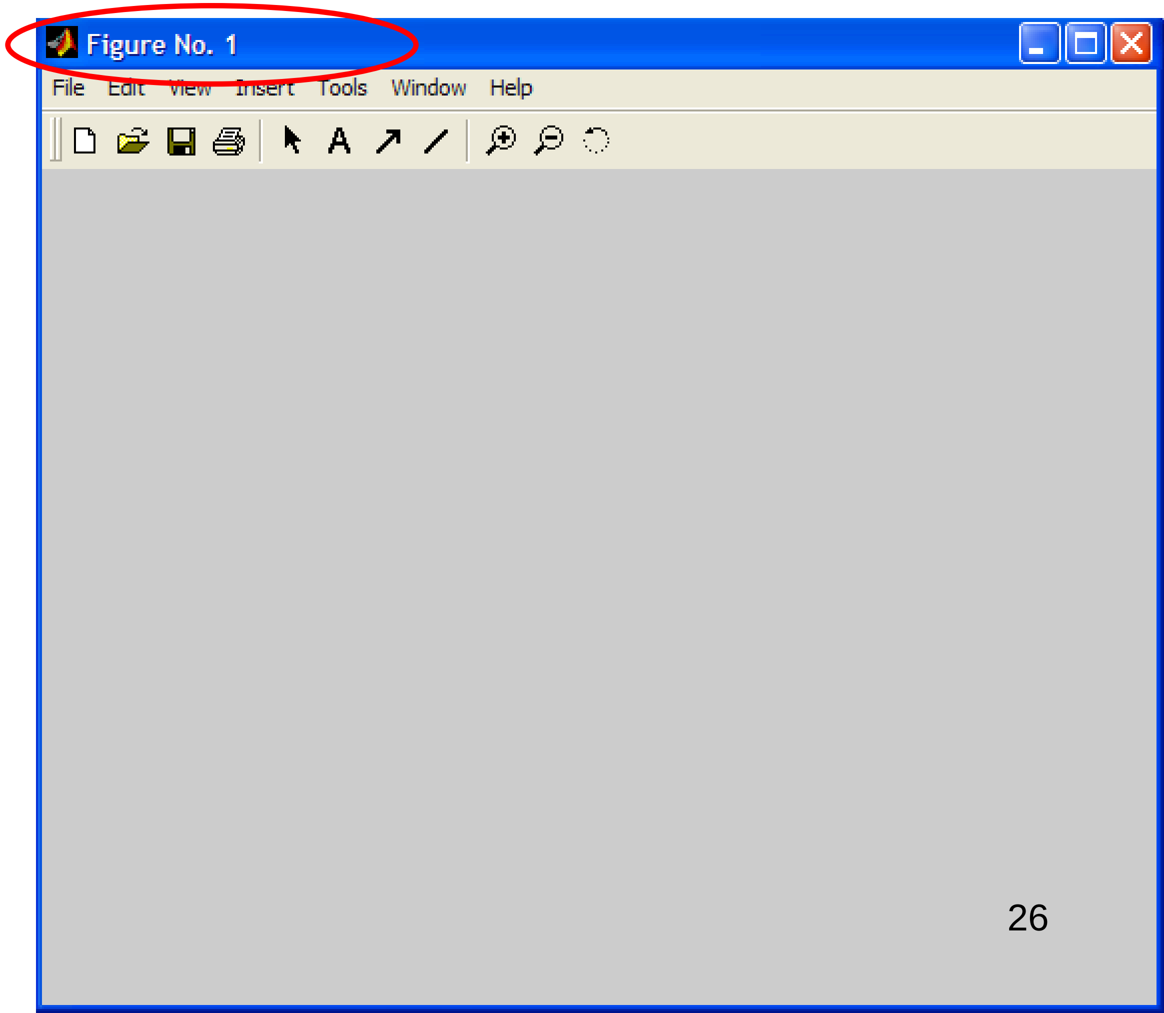
- Stabilire se le seguenti funzioni hanno almeno uno zero
  - $\log(x)-5+x$
  - $x^2-\log(x^2+2)$
  - $x^2-10x+23$
  - $x^2-\sin(\pi x)e^{-x}$

Grafici 1D

figure   apre la finestra grafica

figure(n)   n è il numero associato alla finestra grafica

```
>> figure(1)
```



Grafici

plot(vettore_x, vettore_y, 'opzioni')

vettore_x e vettore_y devono avere lo stesso numero di componenti

opzioni: colore, simbolo linea, marcatore di punto (si possono omettere)

Esempi colore: m magenta, r rosso, g verde, b blu, w bianco, k nero, y giallo

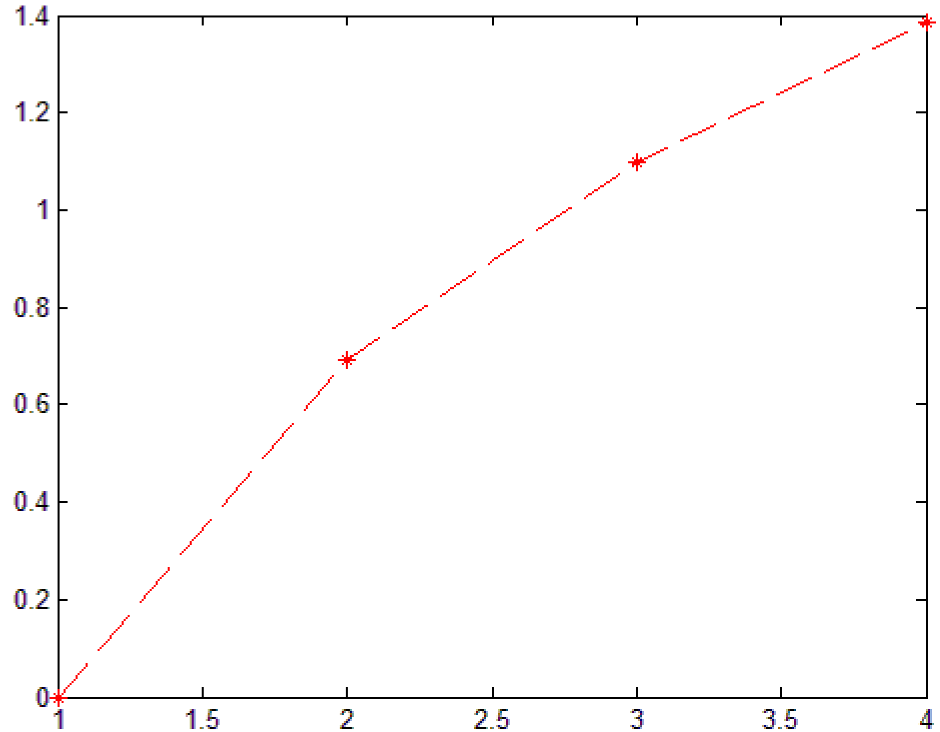
Esempi tipo di linea: - continua (default), -- tratteggiata, : punteggiata, -. Punto-linea

Esempi marcatore di punto + croce, o cerchietto, * asterisco, x ics,....

Grafici

Esempio: Siano x = [1 2 3 4] e y = log(x). Disegnare y in funzione di x usando una linea tratteggiata rossa e marcando i punti della curva con asterischi

```
>> x = 1:4;
>> y=log(x);
>> plot(x,y, 'r--*')
```



OSS: nella rappresentazione grafica, l'opzione linea unisce i punti con una linea retta!!!

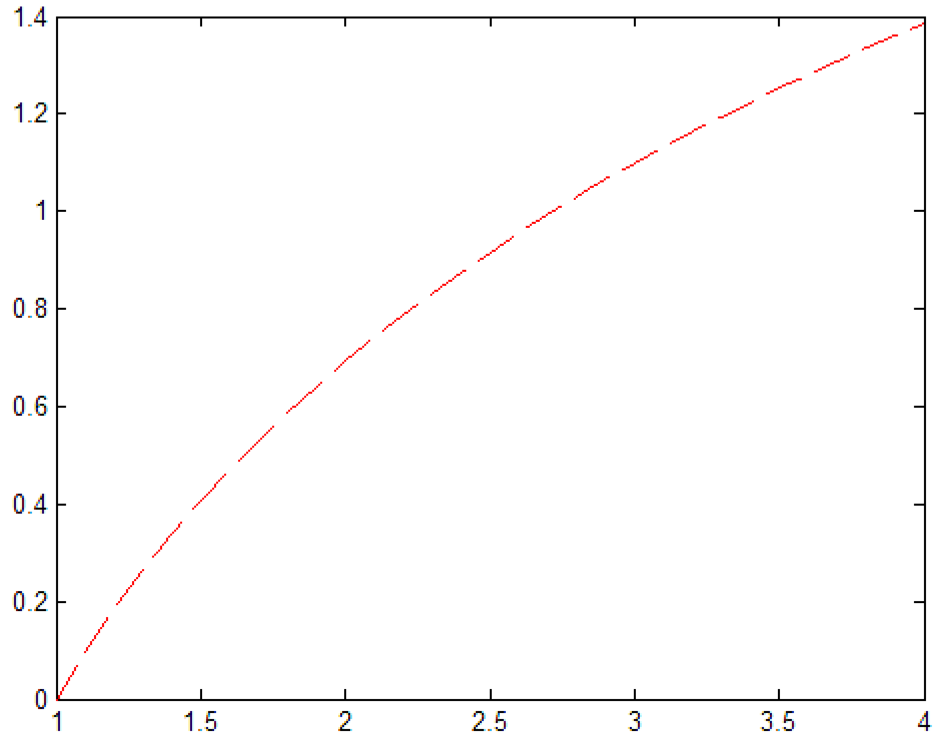
Grafici

OSS: nella rappresentazione grafica, l'opzione linea unisce i punti con una linea retta!!!

Se vogliamo produrre un grafico più preciso, dobbiamo aumentare il numero di punti nell'intervallo scelto

Esempio:

```
>> x = 1:.1:4;
>> y=log(x);
>> plot(x,y, 'r--')
```



Grafici: altre opzioni

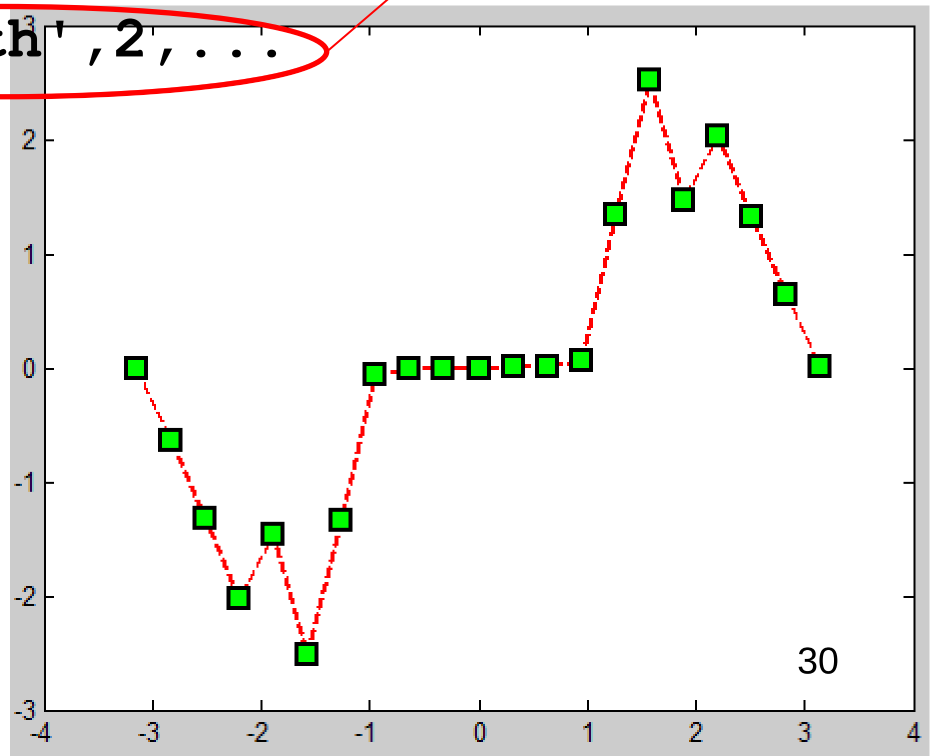
E' possibile migliorare la qualità visiva del grafico usando altre opzioni, come per esempio lo spessore di linea, il colore del marcatore di punto, etc.

Regola la dimensione dello spessore di linea

Esempio:

```
>>x = -pi:pi/10:pi;
>>y = tan(sin(x)) - sin(tan(x));
>>plot(x,y,'--rs','LineWidth',2,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','g',...
'MarkerSize',10)
```

Regolano rispettivamente:
- il colore del bordo
- il colore della parte interna
- la dimensione
del marcatore di punto



Grafici sovrapposti

Sia x il vettore delle ascisse e y_1 e y_2 due vettori aventi la stessa lunghezza di x.

Per visualizzare due grafici diversi sulla stessa finestra si usa il comando hold on

```
>>figure
>>plot(x,y1)
>>hold on
>>plot(x,y2)
>>hold off
```

Oppure la seguente istruzione

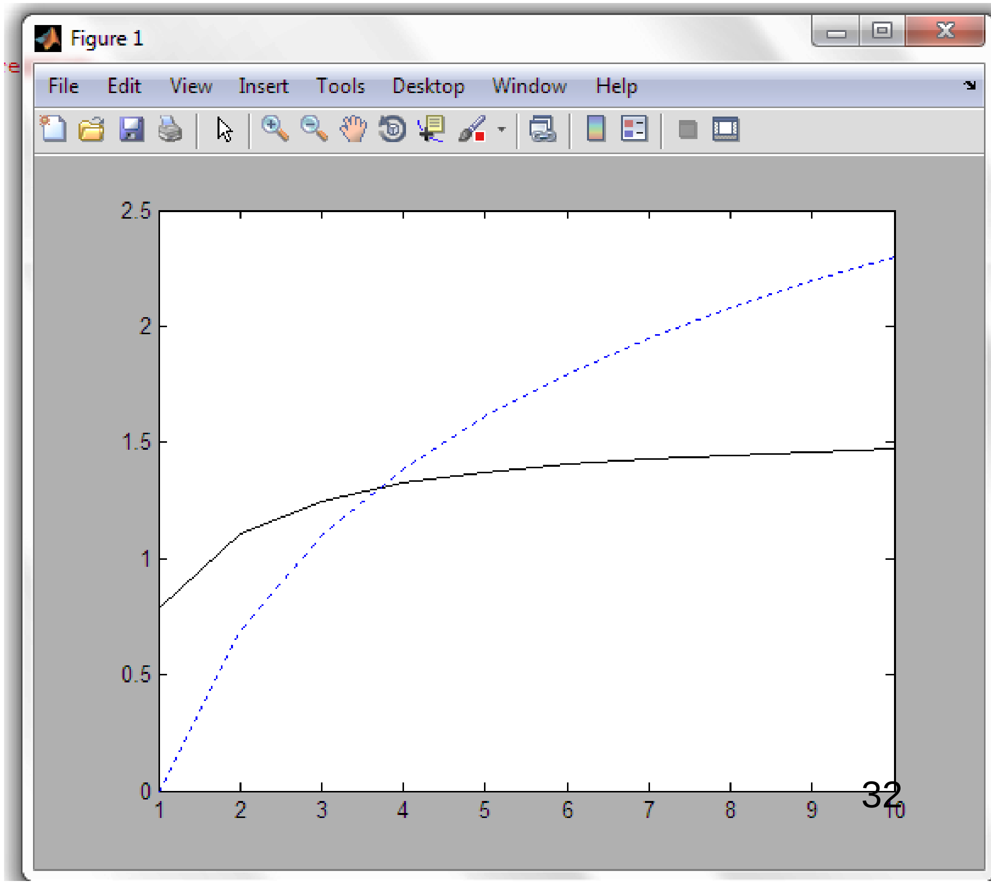
```
>>  plot(x, y1, ‘opzioni’,x, y2,’opzioni’)
```

(le opzioni si possono omettere)

Grafici sovrapposti

Esempio: Sia $x = 1{:}10$, y_1=atan(x) e y_2=log(x). Disegnare sulla stessa finestra y_1 e y_2 marcandoli opportunamente

```
>> x=1:10;
>> y1=atan(x);
>> y2=log(x);
>> figure,
>> plot(x,y1,'k-',x,y2,'b:')
```



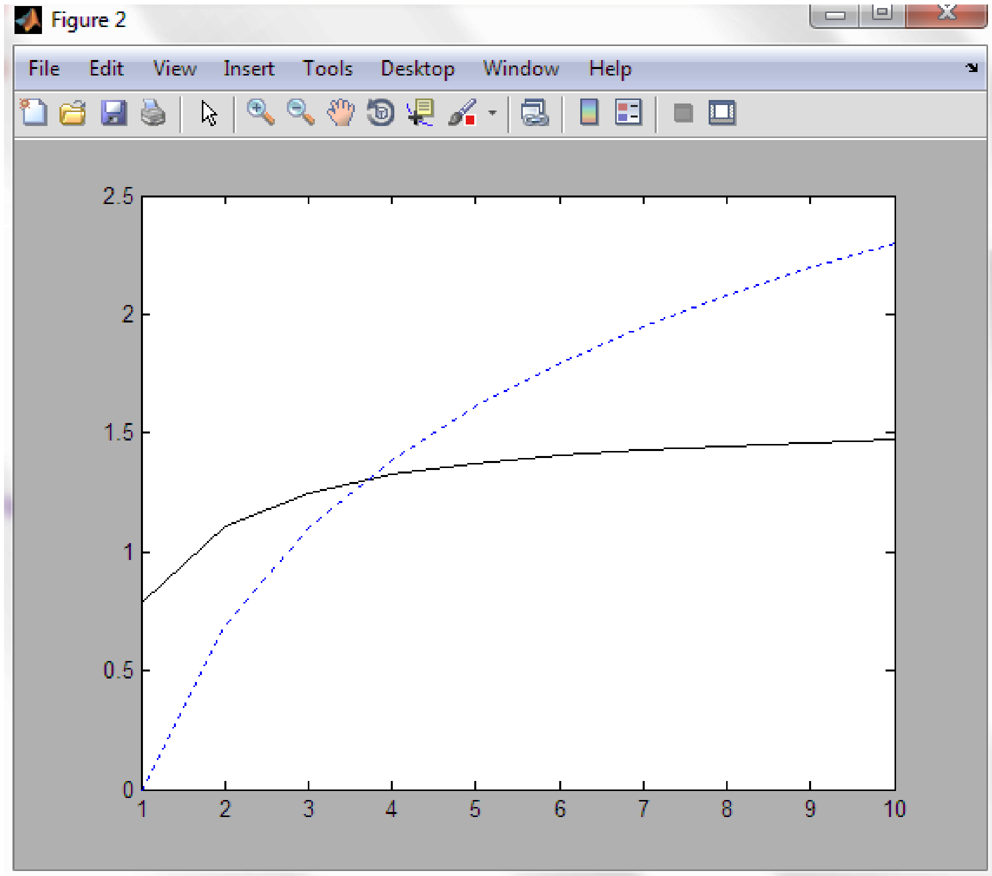
Grafici sovrapposti

Esempio: Sia $x = 1:10$, $y_1=atan(x)$ e $y_2=log(x)$. Disegnare sulla stessa finestra y_1 e y_2 marcandoli opportunamente

```
>> x=1:10;
>> y1=atan(x);
>> y2=log(x);
>> figure,
>> plot(x,y1,'k-',x,y2,'b:')
```

Oppure

```
>> figure,
>> plot(x,y1,'k-')
>> hold on
>> plot(x,y2,'b:')
```



Grafici sovrapposti

```
>> figure,
>> plot(x,y1,'k-')
>> plot(x,y2,'b:')
```

OSS: Se non si usa il comando hold on, il secondo grafico viene disegnato sulla stessa finestra grafica del primo, ma il primo grafico viene cancellato!!!

Grafici

Etichette

title('*stringa*') titolo del grafico

xlabel('*stringa*') etichetta per l' asse delle ascisse (asse x)

ylabel('*stringa*') etichetta dell'asse delle ordinate (asse y)

grid inserisce una griglia nel grafico

legend('*stringa1*', '*stringa2*',...)

legenda: identifica le curve disegnate nel grafico con stringhe diverse (stabilite dall'utente) nell'ordine in cui sono state disegnate

Grafici

Etichette

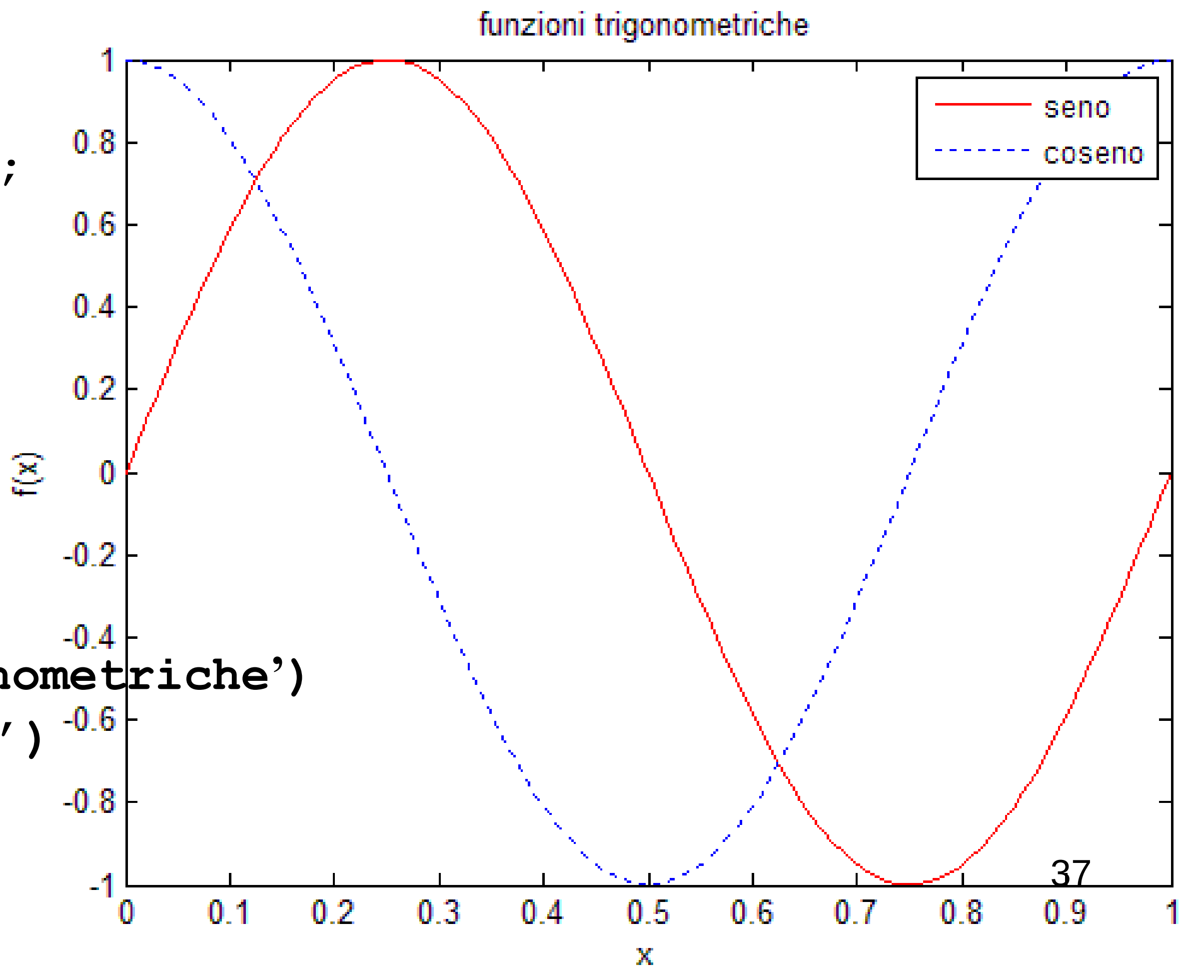
axis([xmin xmax ymin ymax]) regola la dimensione degli assi coordinati (determina il rettangolo nel quale si vogliono visualizzare i dati)

axis('equal') usa la stessa scala di misura sulle ascisse e le ordinate

Grafici

Esempio: Sia x un vettore composto da 100 elementi equidistanti e contenuti nell'intervallo [0,1]. Definire il vettore y = sin(2π x) e il vettore z = cos(2π x). Disegnarli sullo stesso grafico usando tratti diversi e colori diversi nel rettangolo [0 1]x[-1 1]. Etichettare gli assi rispettivamente con "x" e "f(x)" e dare il seguente titolo "funzioni trigonometriche". Includere una legenda.

```
>> x = linspace(0,1,100);
>> y = sin(2*pi*x);
>> z = cos(2*pi*x);
>> figure
>> plot(x,y,'r',x,z,'b:')
>> axis([0 1 -1 1])
>> xlabel('x')
>> ylabel('f(x)')
>> title('funzioni trigonometriche')
>> legend('seno','coseno')
```



Sottografici

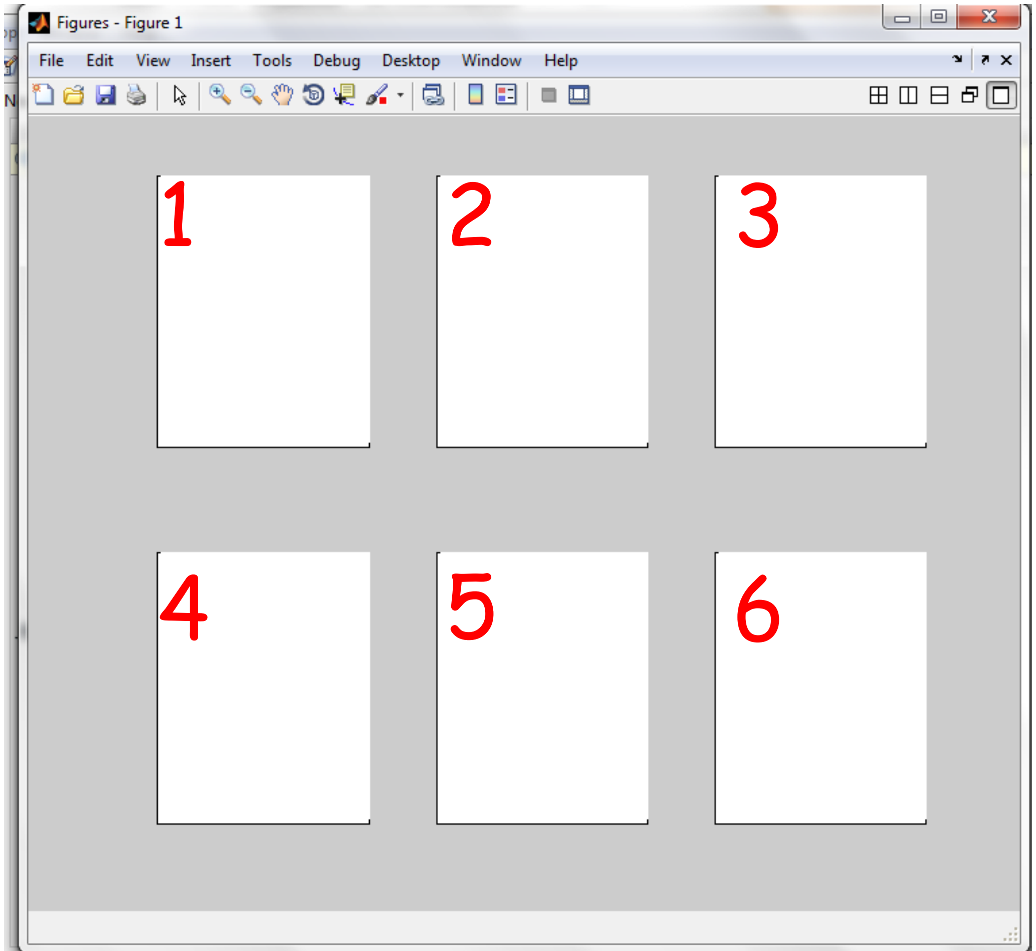
subplot(m,n,k) suddivide la finestra dei grafici in una matrice mxn e attiva la k-ma sottofinestra

Il comando subplot(m,n,k) è sempre seguito dal comando plot che disegna il grafico nella finestra attivata

Esempio

```
>>subplot(2,3,4)
```

divide la finestra grafica in 6 sottofinestre e attiva la quarta



Le sottofinestre sono ordinate da sinistra a destra, dall'alto verso il basso

Sottografici

Esempio

>>subplot(2,1,1)

divide la finestra grafica in 2 sottofinestre poste una sotto l'altra e attiva la prima

>>plot(x1,y1);

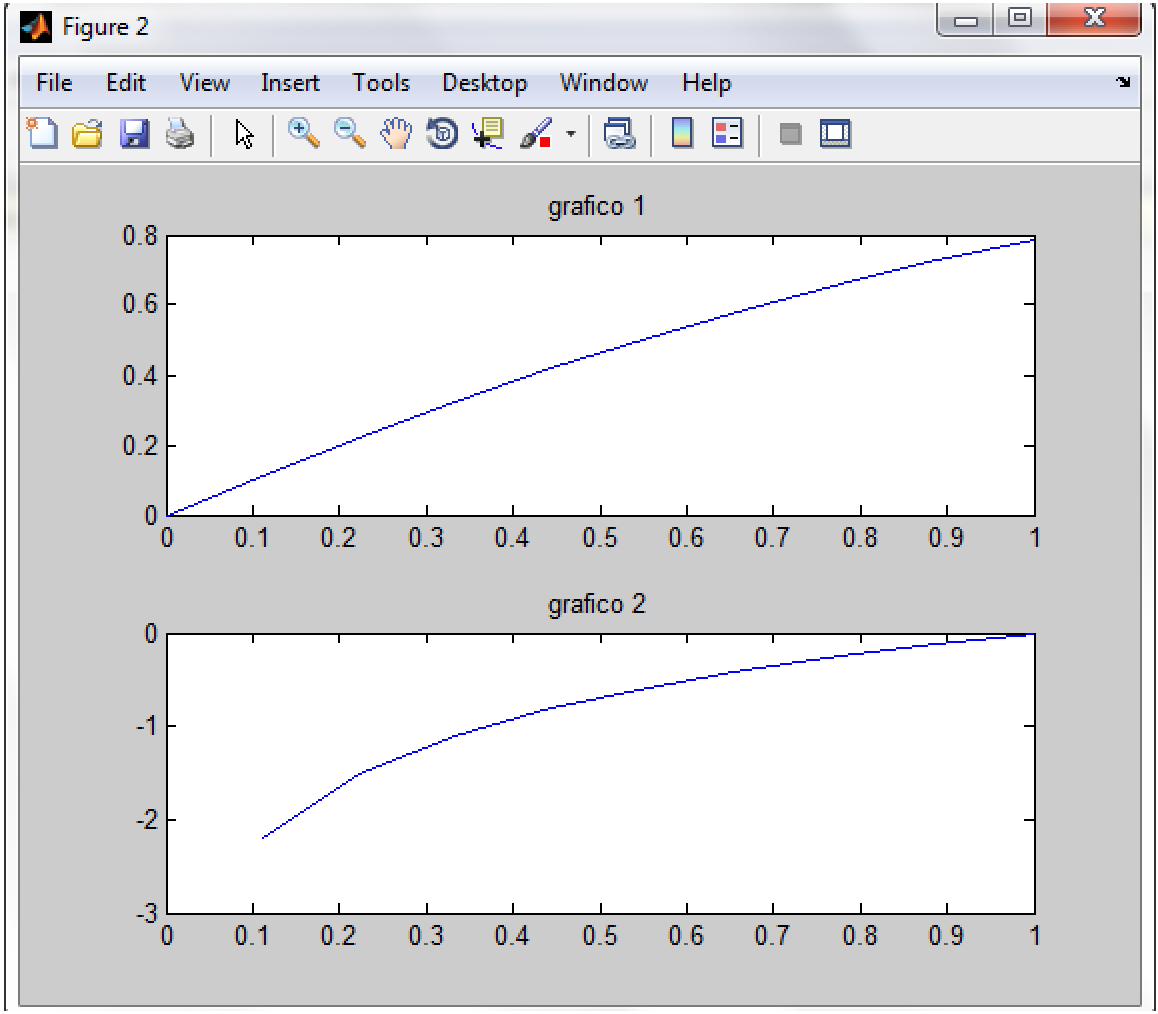
>>title('grafico1')

>>subplot(2,1,2);

attiva la seconda sottofinestra

>>plot(x2,y2);

>>title('grafico2')



Esercizi

1. Graficare tra 0 e 2*pi la funzione sin(x) e le sue approssimazioni con il polinomio di Taylor di 1° e 5° grado, cioè x e x-x^3/3!+x^5/5!. Le 3 curve devono essere rispettivamente a tratto continuo, tratteggiata e marcata con cerchietti.
2. Graficare le funzioni definite nell'esercizio precedente su una stessa finestra grafica ma usando opportunamente tre sottofinestre diverse.
3. Disegnare un cerchio
4. Disegnare in 6 sottofinestre distinte i nodi di Chebychev

$$x_i = -\cos\left(\frac{\pi i}{n}\right) \qquad i = 0,\ldots n$$

nell'intervallo [-1,1] con n=4,6,8,10,12,14

Esercizio

Considerare tre forme equivalenti di uno stesso polinomio:

$$\begin{aligned} p(x) &= (x-1)^6 \\ &= x^6 - 6x^5 + 15x^4 - 20x^3 + 15x^2 - 6x + 1 \\ &= 1 + x(-6 + x(15 + x(-20 + x(15 + x(-6 + x))))) \end{aligned}$$

Stabilire se, valutate al calcolatore in uno stesso punto, forniscono lo stesso risultato. Si supponga di lavorare in singola precisione.

Esercizio

```
>> x = 0.99:.0001:1.01;
>> x=single(x);

>> p1=(x-1).^6;
>> p2=x.^6-6*x.^5+15*x.^4-20*x.^3+15*x.^2-6*x+1;
>> p3=1+x.*(-6+x.*(15+x.*(-20+x.*(15+x.*(-6+x)))));

>> figure,
>> subplot(3,1,1), plot(x,p1)
>> title('(x-1)^6')

>> subplot(3,1,2), plot(x,p2)
>> title('x^6-6*x^5+15*x^4-20*x^3+15*x^2-6*x+1')

>> subplot(3,1,3), plot(x,p3)
>> title('1+x*(-6+x*(15+x*(-20+x*(15+x*(-6+x)))))')
```

Esercizio

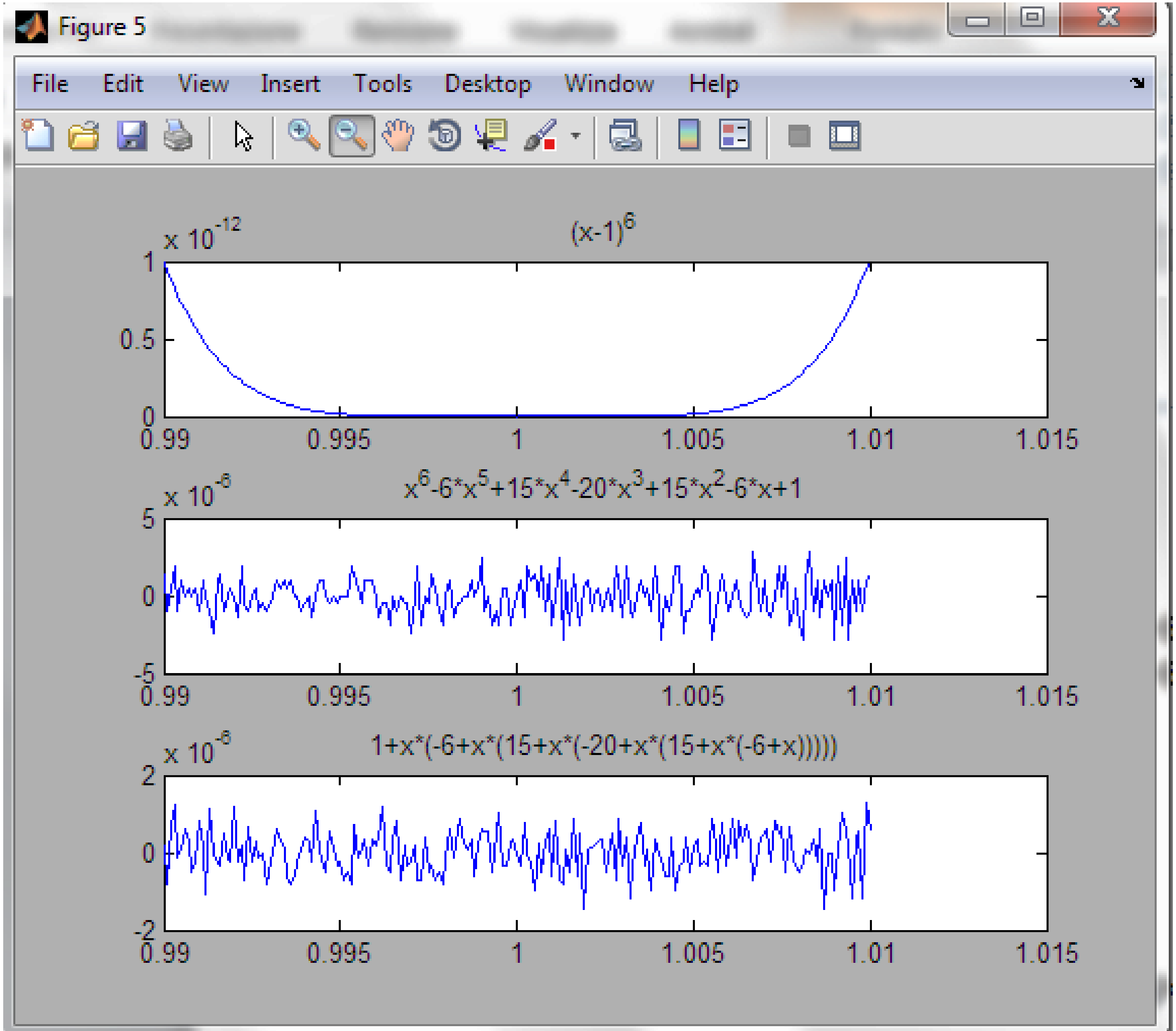


Figure 5
File Edit View Insert Tools Desktop Window Help
x 10^-12
(x-1)^6
1
0.5
0
0.99 0.995 1 1.005 1.01 1.015
x 10^-6
x^6-6*x^5+15*x^4-20*x^3+15*x^2-6*x+1
5
0
-5
0.99 0.995 1 1.005 1.01 1.015
x 10^-6
1+x*(-6+x*(15+x*(-20+x*(15+x*(-6+x)))))
2
0
-2
0.99 0.995 1 1.005 1.01 1.015

Funzioni e Grafici

Sebbene Matlab offra diverse funzioni matematiche predefinite, è possibile definirne altre nei modi seguenti:

- Stringa di caratteri (definita tra due apici ' ')

```
>> f='(x-1).^6'
 f =
          (x-1).^6
```

- Comando inline

```
>> f=inline('(x-1).^6')
 f =
          Inline function:
          f(x)  =  (x-1).^6
```

- Anonymous function mediante il simbolo @, che è un puntatore ad una funzione

```
>> f=@(x)[(x-1).^6]
 f =
          @(x)[(x-1).^6]
```

Funzioni e Grafici

Per valutare la funzione, definita secondo una delle tre modalità precedenti, in un punto x (oppure in un insieme di punti memorizzati in un vettore x), è necessario prima definire x e quindi usare i seguenti comandi

- Stringa di caratteri (definita tra due apici ' ')

```
>> f='(x-1).^6';
>> x=linspace(0.99,1.01,100);
>> y=eval(f);
```

y è un vettore della stessa dimensione di x

e contiene i valori della funzione f valutati in ogni elemento di x

Funzioni e Grafici

- Stringa di caratteri

```
>> f1='(x-1).^6';
>> f2='x.^6-6*x.^5+15*x.^4-20*x.^3+15*x.^2-6*x+1';
>> f3='1+x.*(-6+x.*(15+x.*(-20+x.*(15+x.*(-6+x)))))';

>> x = 0.99:.0001:1.01;
>> x=single(x);

>> p1 = eval(f1);
>> p2 = eval(f2);
>> p3 = eval(f3);

>> figure,
>> subplot(3,1,1), plot(x,p1)
>> title(f1)
>> subplot(3,1,2), plot(x,p2)
>> title(f2)
>> subplot(3,1,3), plot(x,p3)
>> title(f3)
```

Funzioni e Grafici

Per valutare la funzione, definita secondo una delle tre modalità precedenti, in un punto x (oppure in un insieme di punti memorizzati in un vettore x), è necessario prima definire x e quindi usare i seguenti comandi

- Comando inline

```
>> f=inline('(x-1).^6');
>> x=linspace(0.99,1.01,100);
>> y=f(x);
```

oppure

```
>>y=feval(f,x);
```

y è un vettore della stessa dimensione di x

e contiene i valori della funzione f valutati in ogni elemento di x

Funzioni e Grafici

- Comando inline

```
>> f1=inline('(x-1).^6');
>> f2=inline('x.^6-6*x.^5+15*x.^4-20*x.^3+15*x.^2-6*x+1');
>> f3=inline('1+x.*(-6+x.*(15+x.*(-20+x.*(15+x.*(-6+x)))))');

>> x = 0.99:.0001:1.01;
>> x=single(x);

>> p1 = f1(x);
>> p2 = f2(x);
>> p3 = f3(x);

>> figure,
>> subplot(3,1,1), plot(x,p1)
>> title('(x-1)^6')
>> subplot(3,1,2), plot(x,p2)
>> title('x^6-6*x^5+15*x^4-20*x^3+15*x^2-6*x+1')
>> subplot(3,1,3), plot(x,p3)
>> title('1+x*(-6+x*(15+x*(-20+x*(15+x*(-6+x)))))')
```

Funzioni e Grafici

- Comando inline

```
>> f1=inline(‘(x-1).^6’);
>> f2=inline(‘x.^6-6*x.^5+15*x.^4-20*x.^3+15*x.^2-6*x+1’);
>> f3=inline(‘1+x.*(-6+x.*(15+x.*(-20+x.*(15+x.*(-6+x)))))’);

>> x = 0.99:.0001:1.01;
>> x=single(x);

>> p1 = feval(f1,x);
>> p2 = feval(f2,x);
>> p3 = feval(f3,x);

>> figure,
>> subplot(3,1,1), plot(x,p1)
>> title('(x-1)^6')
>> subplot(3,1,2), plot(x,p2)
>> title('x^6-6*x^5+15*x^4-20*x^3+15*x^2-6*x+1')
>> subplot(3,1,3), plot(x,p3)
>> title('1+x*(-6+x*(15+x*(-20+x*(15+x*(-6+x)))))')
```

Funzioni e Grafici

Per valutare la funzione, definita secondo una delle tre modalità precedenti, in un punto x (oppure in un insieme di punti memorizzati in un vettore x), è necessario prima definire x e quindi usare i seguenti comandi

- Anonymous function mediante il simbolo @, che è un puntatore ad una funzione

```
>> f=@(x)[(x-1).^6];
>> x=linspace(0.99,1.01,100);
>> y=f(x);
```

oppure

```
>>y=feval(f,x);
```

y è un vettore della stessa dimensione di x

e contiene i valori della funzione f valutati in ogni elemento di x

Funzioni e Grafici

- Comando inline

```
>> f1=@(x)[(x-1).^6];
>> f2=@(x)[x.^6-6*x.^5+15*x.^4-20*x.^3+15*x.^2-6*x+1];
>> f3=@(x)[1+x.*(-6+x.*(15+x.*(-20+x.*(15+x.*(-6+x)))))];

>> x = 0.99:.0001:1.01;
>> x=single(x);

>> p1 = f1(x);
>> p2 = f2(x);
>> p3 = f3(x);

>> figure,
>> subplot(3,1,1), plot(x,p1)
>> title('(x-1)^6')
>> subplot(3,1,2), plot(x,p2)
>> title('x^6-6*x^5+15*x^4-20*x^3+15*x^2-6*x+1')
>> subplot(3,1,3), plot(x,p3)
>> title('1+x*(-6+x*(15+x*(-20+x*(15+x*(-6+x)))))')
```

Funzioni e Grafici

fplot(funzione,limiti_assi,'opzioni')

Permette di graficare la funzione (definita usando una delle modalità mostrate nelle slide precedenti) nell'intervallo definito da limiti_assi senza dover definire il vettore x

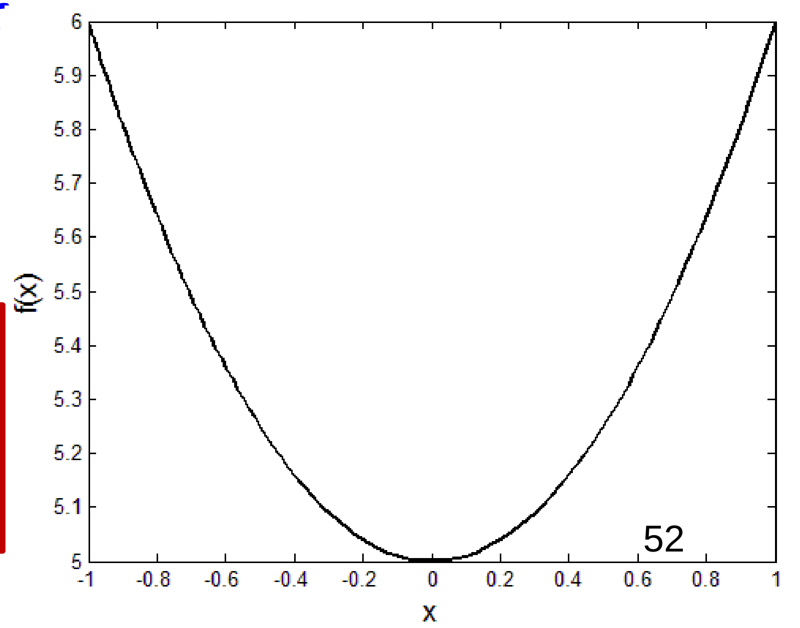
limiti_assi : [xmin xmax] è un vettore contenente i limiti inf e sup dell'intervallo in cui si vuole visualizzare funzione;

'opzioni': sono le opzioni del comando plot

Esempio:

```
>> f = @(x)[x^2+5];
>> fplot(f,[-1 1])
>> xlabel('x'), ylabel('f(x)')
```

Disegna il grafico della funzione f nell'intervallo [-1,1]



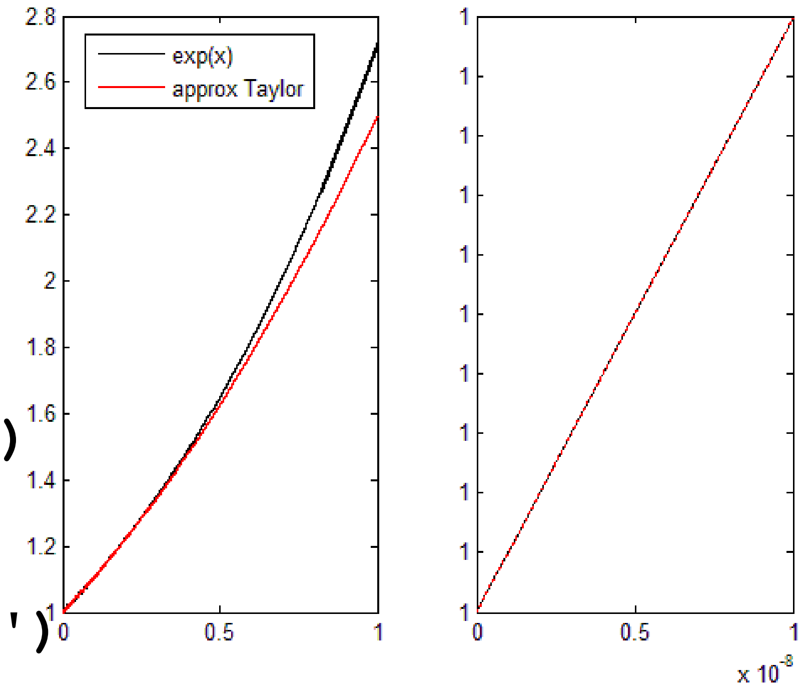
Esercizio

Consideriamo lo sviluppo in serie di Taylor di e^x attorno allo 0

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} e^{\xi} \qquad 0 \le \xi < x \le 1$$

Graficare e^x e la sua approssimazione negli intervalli [0,1] e [0 10^{-8}]

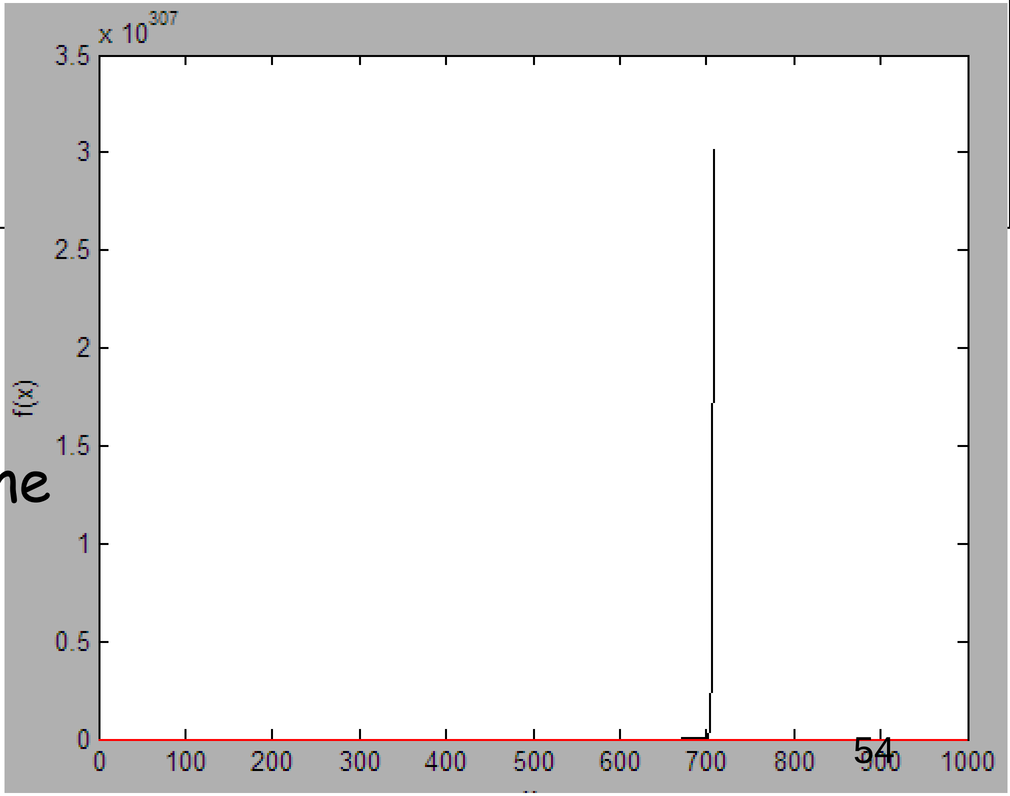
```
>> f = @(x)[exp(x)];
>> f2 = @(x)[1+x+x^2/2];
>> figure,
>> subplot(1,2,1),
>> fplot(f,[0 1],'k'), hold on,
>> fplot(f2,[0 1],'r'),
>> legend('exp(x)','approx Taylor')
>> subplot(1,2,2),
>> fplot(f,[0 10^-8],'k:'),
>> hold on, fplot(f2,[0 10^-8],'r:')
```



Separazione grafica delle radici

Esercizio: Separare graficamente gli zeri positivi della seguente funzione $f(\lambda) = e^{\lambda} + 0.435\,(e^{\lambda} - 1) / \lambda - 1.564$

```
>>f=@(lambda)[exp(lambda)+0.435*(exp(lambda)-1)/lambda-1.564];
>>assey=@(x)[0];
>>figure,
>>fplot(f,[0 1000],'k')
>>hold on,
>>fplot(assey,[0 1000],'r')
>>xlabel('x')
>>ylabel('f(x)')
```

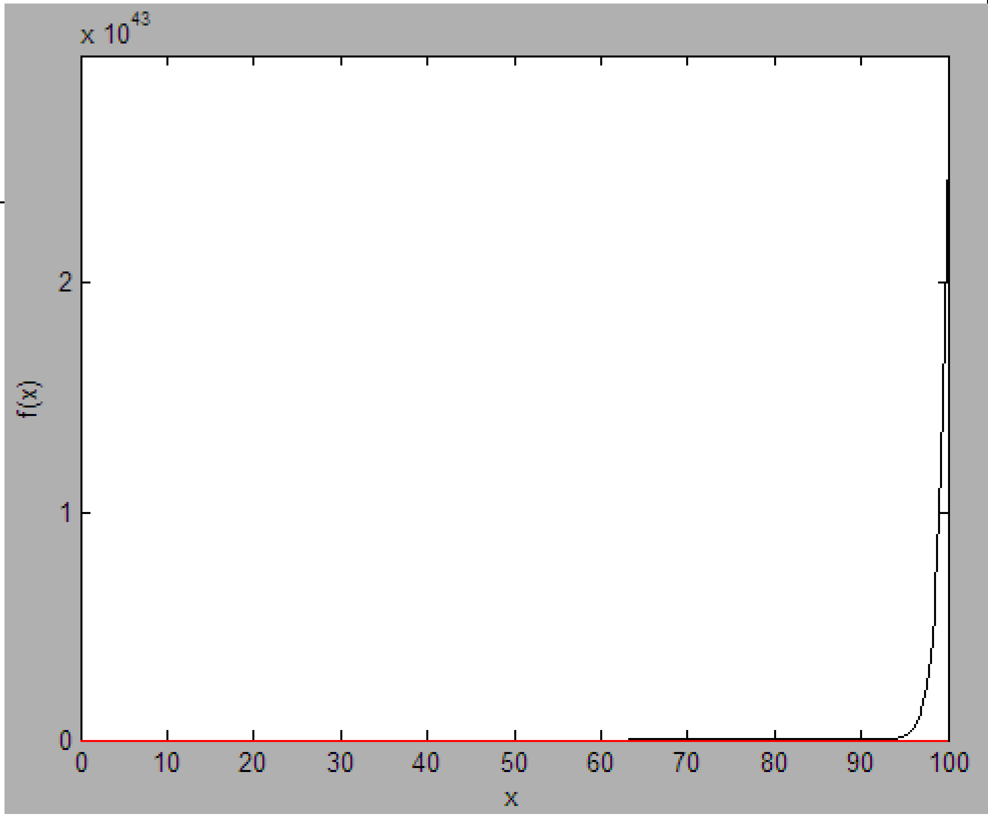


Dal grafico è evidente che per valori di x molto grandi la funzione è sicuramente diversa da zero
Si può, quindi, restringere l'intervallo di visualizzazione della funzione

Separazione grafica delle radici

Esercizio: Separare graficamente gli zeri positivi della seguente funzione $f(\lambda) = e^{\lambda} + 0.435\,(e^{\lambda} - 1) / \lambda - 1.564$

```
>>f=@(lambda)[exp(lambda)+0.435*(exp(lambda)-1)/lambda-1.564];
>>assey=@(x)[0];
>>figure,
>>fplot(f,[0 100],'k')
>>hold on,
>>fplot(assey,[0 100],'r')
>>xlabel('x')
>>ylabel('f(x)')
```

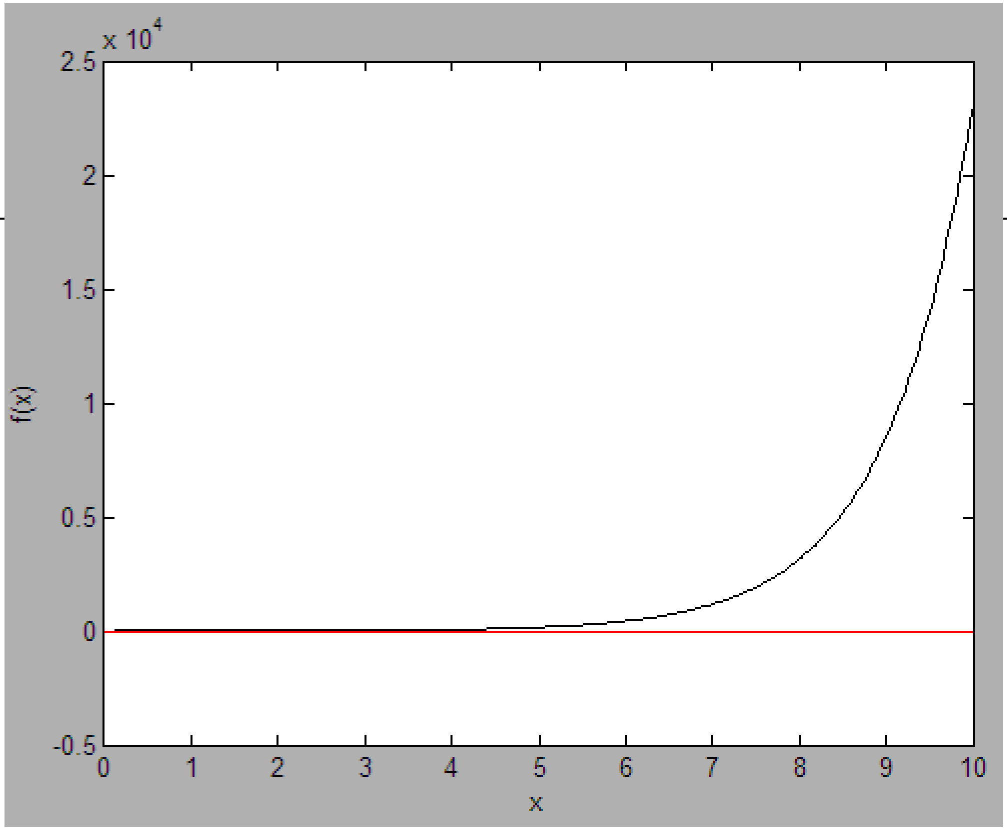


Si può restringere ancora l'intervallo di visualizzazione della funzione

Separazione grafica delle radici

Esercizio: Separare graficamente gli zeri positivi della seguente funzione $f(\lambda) = e^{\lambda} + 0.435\,(e^{\lambda} - 1) / \lambda - 1.564$

```
>>f=@(lambda)[exp(lambda)+0.435*(exp(lambda)-1)/lambda-1.564];
>>assey=@(x)[0];
>>figure,
>>fplot(f,[0 10],'k')
>>hold on,
>>fplot(assey,[0 10],'r')
>>xlabel('x')
>>ylabel('f(x)')
```



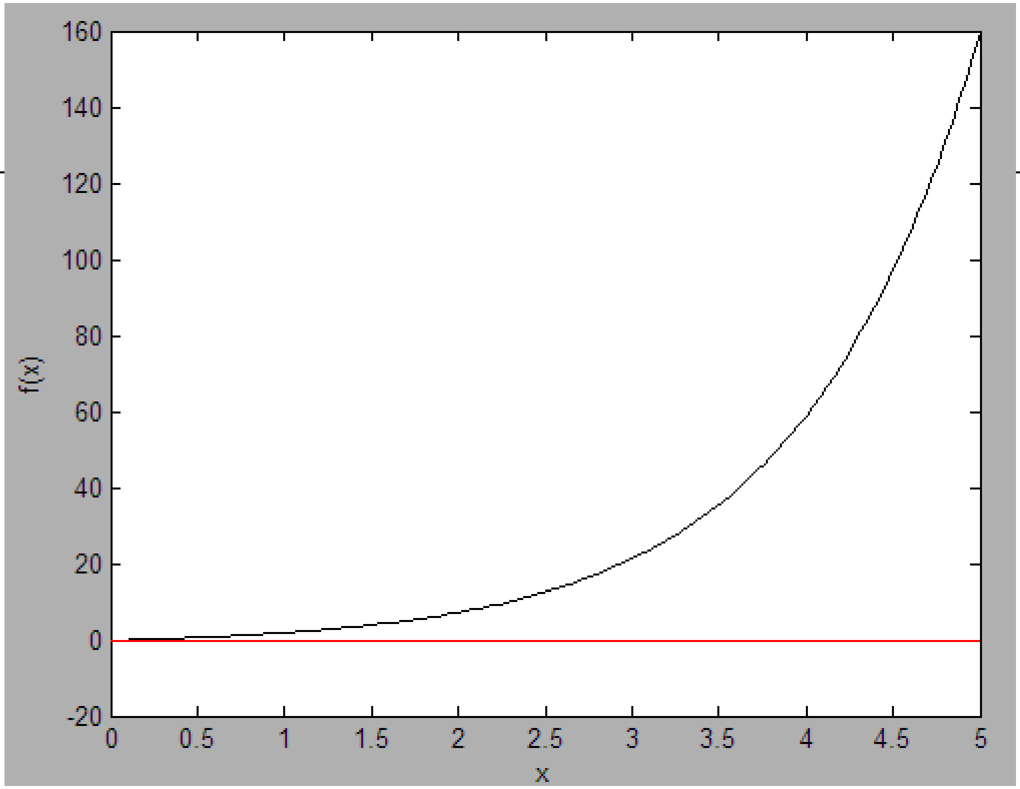
Si può restringere ancora l'intervallo di visualizzazione della funzione

Separazione grafica delle radici

Esercizio: Separare graficamente gli zeri positivi della seguente funzione $f(\lambda) = e^{\lambda} + 0.435\,(e^{\lambda} - 1) / \lambda - 1.564$

```
>>f=@(lambda)[exp(lambda)+0.435*(exp(lambda)-1)/lambda-1.564];
>>assey=@(x)[0];
>>figure,
>>fplot(f,[0 5],'k')
>>hold on,
>>fplot(assey,[0 5],'r')
>>xlabel('x')
>>ylabel('f(x)')
```

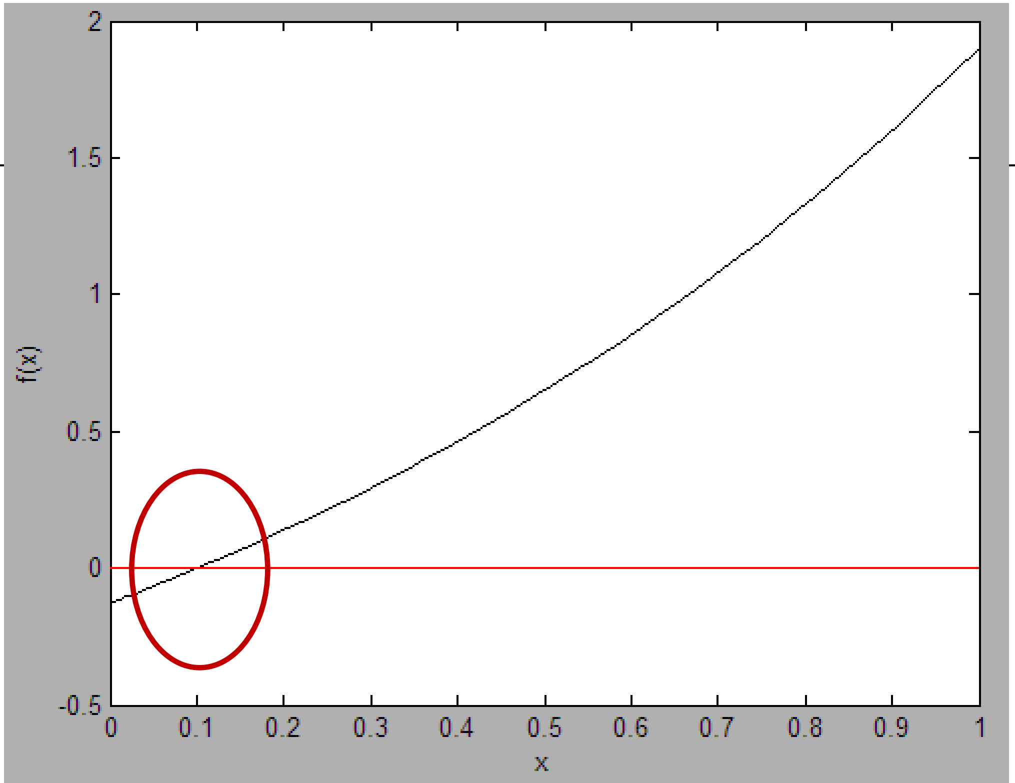
Si può restringere ancora l'intervallo di visualizzazione della funzione



Separazione grafica delle radici

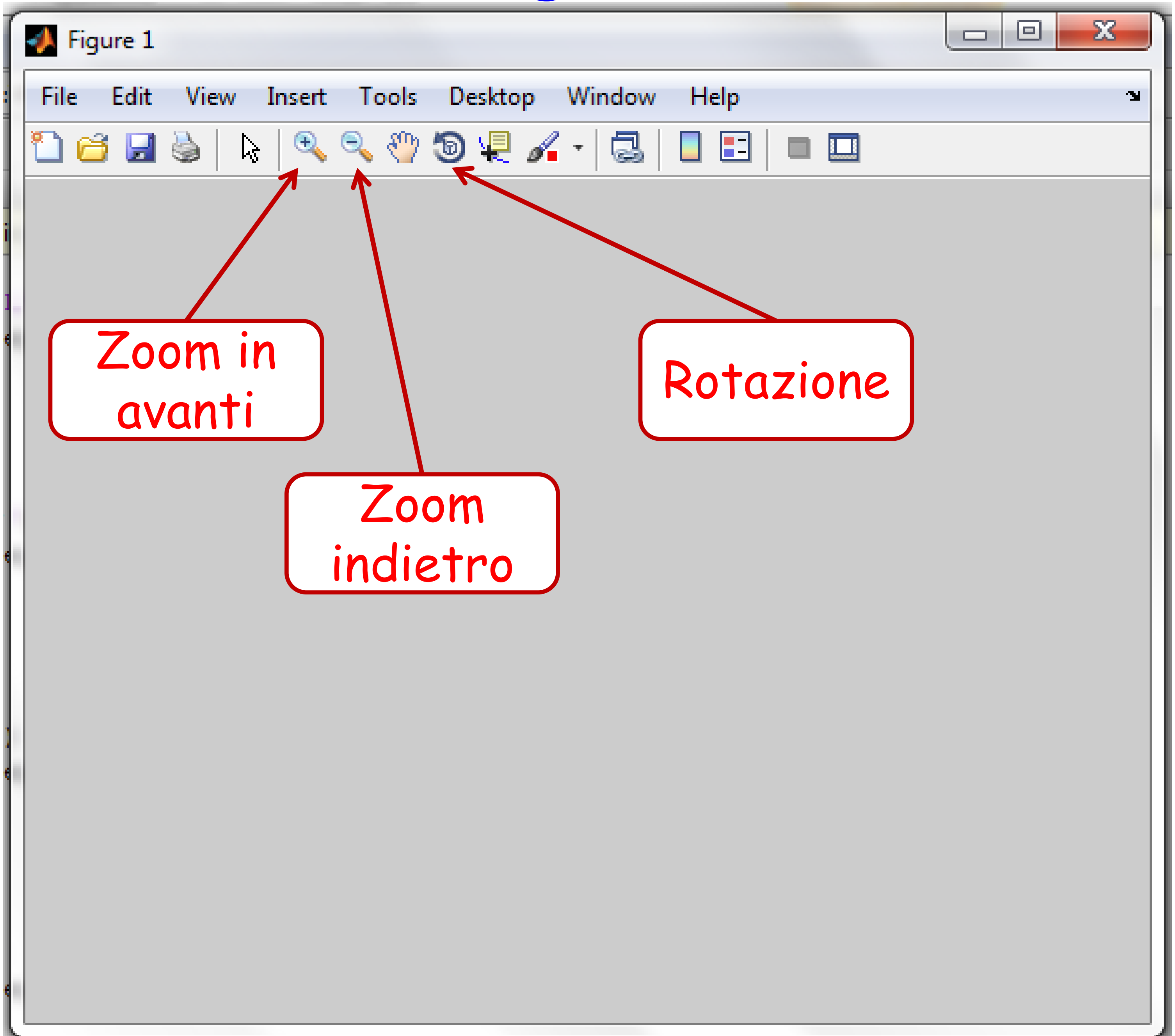
Esercizio: Separare graficamente gli zeri positivi della seguente funzione $f(\lambda) = e^{\lambda} + 0.435\,(e^{\lambda} - 1) / \lambda - 1.564$

```
>>f=@(lambda)[exp(lambda)+0.435*(exp(lambda)-1)/lambda-1.564];
>>assey=@(x)[0];
>>figure,
>>fplot(f,[0 1],'k')
>>hold on,
>>fplot(assey,[0 1],'r')
>>xlabel('x')
>>ylabel('f(x)')
```



Dal grafico è possibile determinare un buon intervallo di separazione dello zero della funzione, per esempio [0.05,0.15]

La finestra grafica di Matlab



Esercizi

1. Usare le diverse modalità per valutare e graficare le seguenti funzioni negli intervalli I specificati
   - $f(x) = 1/(1+x^2)$, $I=[-5,5]$
   - $f(x) = \sin(x)+\cos(x)$, $I=[-\pi, \pi]$

2. Separare gli zeri delle seguenti funzioni contenuti negli intervalli specificati
   - $f(x) = (x-1)^2-3\log(x)$ in $I=[0.5,2]$
   - $f(x) = \sin(x)+x-1$ in $I=[0,\pi]$
   - $f(x) = \log(x)+x^2-x$ in $I=[0.7,2.3]$

3. Separare le radici dell' equazione non lineare $\log(1/x)-x^2=0$ graficando su una stessa finestra le funzioni $f(x)=\log(1/x)$ e $g(x)=x^2$

Funzioni di input e output

Per visualizzare stringhe sullo schermo:

disp(‘*stringa di caratteri*’)

Esempio:

```
>> disp('Oggi e'' una bella giornata')
Oggi e' una bella giornata
```

Nota: Se la stringa contiene apostrofi e/o accenti è necessario il doppio apice " per evitare conflitti con l'apice singolo usato per racchiudere caratteri

```
>> disp(['Sabato ',' Domenica ',' Lunedi'''])
Sabato  Domenica  Lunedi'
```

Nota: Si visualizza un vettore le cui componenti sono stringhe

Funzioni di input e output

```
>> disp(['Sabato  ';'Domenica';'Lunedi'' '])
Sabato
Domenica
Lunedi`

>> disp(['Sabato';'Domenica';'Lunedi'''])
??? Error using ==> vertcat
CAT arguments dimensions are not consistent.
```

Nota:
Si visualizza un vettore colonna le cui componenti sono stringhe. In questo caso è necessario che le stringhe siano composte dallo stesso numero di caratteri, altrimenti si riceve un messaggio di errore. Per avere stringhe della stessa lunghezza sono stati usati gli spazi bianchi

Funzioni di input e output

E' possibile visualizzare caratteri e numeri contemporaneamente definendo un vettore composto da caratteri e numeri. Il numero, però, deve essere trasformato in stringa usando il comando num2str

num2str(x) trasforma il numero contenuto nella variabile x in un carattere

```
>> x=3;
>> disp(['la variabile x contiene il valore ' num2str(x)])
la variabile x contiene il valore 3
```

Funzioni di input e output

Per introdurre valori da tastiera:

nome_variabile = input(‘*stringa di caratteri tra apici*’)

Assegna alla variabile nome_variabile il valore (scalare, vettore o matrice) digitato dall’utente da tastiera subito dopo la ‘*stringa di caratteri tra apici*’ che viene visualizzata sulla finestra dei comandi

Esempio:

```
>>n=input(‘inserisci il numero di righe della matrice: ’)
inserisci il numero di righe: 3
n =
     3
>>m=input(‘inserisci il numero di colonne della matrice:’)
inserisci il numero di colonne: 7
m =
     7
>>A=zeros(n,m);(A è una matrice di zeri di dimensione nxm)
```

Output dei dati

Per visualizzare un insieme di dati (*variabili*) di output in un formato specifico, si può usare il comando

fprintf(*fid, formato, variabili*)

la stringa di formato contiene i codici che specificano il tipo di variabile che viene scritta nel file di output scelto (che può essere opzionale) il cui indicatore è fid. Se si omette fid, le variabili vengono visualizzate nel formato specificato sulla finestra dei comandi

Esempio

```
>> x = 3;
>> s = 'ciao';
>> fprintf('un intero %d e una stringa %s\n', x, s)
un intero 3 e una stringa ciao
```

Formattazione dell'output

Descrittori di formato: specificano tipo, allineamento, cifre significative, ampiezza di campo

%s -> formato stringa
%d -> formato numero intero
%f -> formato numero decimale
%e -> formato numero in notazione scientifica
%g -> formato numero in forma compatta

Esempi: %10s visualizza una stringa di 10 caratteri
%6.2f visualizza un numero con 6 cifre di cui 2 decimali
%-5d visualizza un intero con 5 cifre allineato a sinistra

Caratteri speciali
\n -> carattere di ritorno a capo
\t -> carattere di tabulazione

```
>> fprintf('un intero %d e una stringa %s\n', x, s)
un intero 3 e una stringa ciao
```

Formattazione dell'output

```
>> x=sqrt(2);

>> fprintf('il valore della variabile e'' %6.4f \n', x)
il valore della variabile e` 1.4142

>> fprintf('il valore della variabile e''  %6.4e \n', x)
il valore della variabile e` 1.4142e+000

>> fprintf('il valore della variabile e''  %6.4g \n', x)
il valore della variabile e` 1.414
```

Formattazione dell'output

Nota. fprintf può visualizzare vettori e matrici stampando gli elementi su una riga secondo l'ordinamento lessicografico (leggendoli per colonna) se non si indicano formati particolari.

```
>> A=[1 2 3; 4 5 6]
   A =
        1     2     3
        4     5     6
>> fprintf('%3d',A)
     1  4  2  5  3  6

>> fprintf('%3d%3d\n',A)
     1  4
     2  5
     3  6
```

Formattazione dell'output

Esercizio: Costruire e visualizzare la tabella dei valori delle funzioni sin(x) e cos(x) in corrispondenza di k punti equidistanti in $[0, \pi]$. k deve essere digitato da tastiera dall'utente

```
>>k=input('inserire il numero di punti in cui valutare le...
   funzioni    ');
inserire il numero di punti in cui valutare le funzioni   6
>>x=linspace(0,pi,k);
>>f1=cos(x);
>>f2=sin(x);
>> fprintf('n\t x(n)\t cos(x(n))\t sin(x(n))\n');,...
 fprintf('%d\t %3.2f\t %+6.5f\t %-6.5f\n',[(1:k);x;f1;f2])
n   x(n)        cos(x(n))     sin(x(n))
1   0.00        +1.00000      0.00000
2   0.63        +0.80902      0.58779
3   1.26        +0.30902      0.95106
4   1.88        -0.30902      0.95106
5   2.51        -0.80902      0.58779
6   3.14        -1.00000      0.00000
```

Formattazione dell'output

Infatti

```
>>disp([(1:k);x;f1;f2])
    1.0000    2.0000    3.0000    4.0000    5.0000    6.0000
         0    0.6283    1.2566    1.8850    2.5133    3.1416
    1.0000    0.8090    0.3090   -0.3090   -0.8090   -1.0000
         0    0.5878    0.9511    0.9511    0.5878    0.0000
```

```
n   x(n)        cos(x(n))     sin(x(n))
1   0.00        +1.00000      0.00000
2   0.63        +0.80902      0.58779
3   1.26        +0.30902      0.95106
4   1.88        -0.30902      0.95106
5   2.51        -0.80902      0.58779
6   3.14        -1.00000      0.00000
```

Esercizio

Quale comando si deve usare per ottenere la seguente stampa?

```
x=1.00  |  log(x)=0.00
x=1.25  |  log(x)=0.22
x=1.50  |  log(x)=0.41
x=1.75  |  log(x)=0.56
x=2.00  |  log(x)=0.69
```

Esercizio

Quale comando si deve usare per ottenere la seguente stampa?

x=1.00 | log(x)=0.00
x=1.25 | log(x)=0.22
x=1.50 | log(x)=0.41
x=1.75 | log(x)=0.56
x=2.00 | log(x)=0.69

```
>> x=1:.25:2; y=log(x);
>> fprintf('x=%-6.2f |  log(x)=%-6.2f\n',[x;y]);
```

Output dei dati

Per visualizzare un insieme di dati di output in un formato specifico, si può usare anche il comando

sprintf(*formato, variabili*)

la stringa di formato contiene i codici che specificano il tipo di variabile che deve essere visualizzata

L'output è indirizzato su una stringa di testo

Esempio

```
>>x = 3;
>>s = 'ciao';
>>stringa=sprintf('un intero %d e una stringa %s\n',x, s);
>> disp(stringa)
un intero 3 e una stringa ciao
```

Matrici

Eliminare righe o colonne:

$A(i,:) = []$ elimina la i-ma riga di A

$A(:, m:k) = []$ elimina tutte le colonne di indici da m a k

Esempio: Sia A = [1 2 0; 3 4 1; 2 6 3]. Eliminare la seconda riga.

```
>> A(2,:)=[]

   A =

        1        2        0

        2        6        3
```

Esempio: Sia A = [1 2 0; 3 4 1; 2 6 3]. Eliminare la prima e la seconda colonna.

```
>> A(:,[1 2])=[]

A =

      0

      1

      3
```

Matrici

Annullare righe o colonne:

$A(i,:) = 0$ sostituisce alla i-ma riga di A un vettore di elementi nulli

$A(:, m:k)=0$ sostituisce tutte le colonne di indici da m a k con vettori di elementi nulli

Esempio: Sia A = [1 2 ; 3 4]. Annullare la seconda riga.

```
>> A(2,:)=0

A =

     1     2

     0     0
```

Esempio: Sia A = [1 2 0; 3 4 1; 2 6 3]. Annullare la prima e la seconda colonna.

```
>> A(:,[1 2])=0

A =

     0     0     0

     0     0     1

     0     0     3
```

Modifica di Matrici

A(i,j)=x	Assegna il valore x all'elemento i,j della matrice *A*
A(i,:)=v	Assegna alla riga i della matrice *A* gli elementi del vettore riga v
A(:,j)=w	Assegna alla colonna j della matrice *A* gli elementi del vettore colonna w
A(i,m:p:n)=v	Assegna agli elementi di indice di colonna da m a n con passo p della i-esima riga gli elementi del vettore riga v
A(i,:)=[ ]	Elimina la riga i della matrice *A*
A(:,j)=[ ]	Elimina la colonna j della matrice *A*

Matrici

Il comando colon :

A(:) organizza gli elementi della matrice in un vettore colonna (posiziona le colonne di A una sotto l'altra)

Usando la stessa strategia A(k) indica il k-esimo elemento di A, ovvero il k-esimo elemento del vettore colonna in cui è memorizzata A

Esempio: Se A = [1 0 3; 4 2 1]

```
>> A(:)
ans =
     1
     4
     0
     2
     3
     1

>> A(5)
ans =
     3
```

Esercizi

- Costruire una matrice 5x5, ed assegnarla alla variabile *A*
- Azzerare la prima colonna di *A*
- Eliminare la seconda e l'ultima colonna di *A* ed indicare con *C* la matrice così ottenuta
- Estrarre da *A* la sottomatrice 2x2 costituita dalla seconda e terza riga e dalla seconda e terza colonna di *A* ed indicare con D la matrice ottenuta
- Estrarre da *C* una matrice di dimensione 2x2 e sottrarla alla matrice D