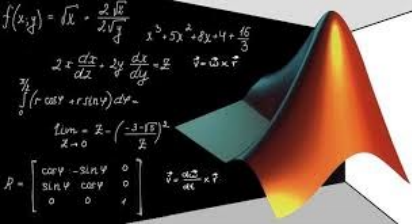


Matlab

Utilizzo del software più idoneo per accompagnare gli studenti del corso di Ingegneria Elettrica per la e-mobility nel raggiungere i propri ambiziosi traguardi



Istallazione

MATLAB®
& SIMULINK®



unipa

ITA ENG

INFORMAZIONI ▾ DIDATTICA ▾ DOCENTI MOBILITÀ E BORSE DI STUDIO ▾ QUALITÀ ▾ DIDATTICA ON-LINE SOFTWARE ≡

2224 - INGEGNERIA ELETTRICA PER LA E-MOBILITY

HOME > Dipartimenti > Ingegneria > 2224 - INGEGNERIA ELETTRICA PER LA E-MOBILITY



Sede: PALERMO
Anni attivi: I,II,III
Classe: L-9
COORDINATORE: FABIO VIOLA

Didattica erogata
Calendario didattico
Orario delle lezioni
Calendario Esami

camera semi anecoica

Ascolta

In basso un video sulla nostra camera semi-anechoica, un ambiente schermato dove i nostri studenti testano i disturbi radiati e condotti di dispositivi innovativi realizzati negli altri nostri laboratori

camera semi-anechoica

<https://servizisia.unipa.it/cms/matlab#access>

dj dipartimento
di ingegneria
unipa

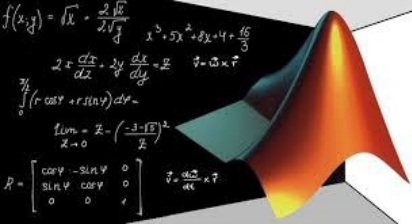
Prof. Fabio Viola



Laboratory of Electrical Application

Download del programma di installazione

1. Creare un account MathWorks utilizzando l'indirizzo e-mail della tua università: mathworks.com/accesslogin/createProfile.do.
2. Accedere al License Center: <http://mathworks.com/licensecenter>.
3. Fare clic su Aggiungi licenza nell'angolo in alto a destra.
4. Scegliere Collega mediante codice di attivazione e fare clic su Continua.
5. Quando richiesto, inserire il Codice di attivazione. ([Richiedere il codice di attivazione inserendo le proprie credenziali di Ateneo](#))
6. Accedere a: mathworks.com/downloads/web_downloads/select_release.
7. Fare clic sul pulsante scarica per ottenere la versione più recente.
8. Scaricare il programma di installazione cliccando sull'apposito pulsante.



Istallazione

Installazione e attivazione

1. Salvare il programma di installazione scaricato in un file browser. Dovrebbe trovarsi nella posizione predefinita di download, a meno che non ne sia stata specificata una diversa. Il nome del programma di installazione è il seguente:

1. Windows: matlab__win64.exe
2. Mac OS X: matlab__maci64.zip Dove rappresenta il numero della release.

2. Avviare il programma di installazione:

1. Windows: fare doppio clic sul file del programma di installazione che è stato scaricato nella fase precedente. Viene eseguito il programma di autoestrazione di Windows e il programma di installazione si avvia.
2. Mac OS X: fare doppio clic sul file del programma di installazione che è stato scaricato nella fase precedente. Questa operazione estrae i file e crea un'altra cartella denominata matlab__maci64, dove rappresenta il numero della release. All'interno della cartella, fare doppio clic su InstallForMacOSX per avviare il programma di installazione.

3. Nel programma di installazione MathWorks, selezionare Accedi con un account Mathworks e seguire le istruzioni online.

4. Quando richiesto, selezionare la licenza che si desidera utilizzare.

5. Selezionare i prodotti che si desidera scaricare e installare.

6. Dopo aver scaricato e installato i prodotti desiderati, mantenere selezionata la casella Attiva MATLAB e fare clic su Avanti.

7. Quando viene richiesto di fornire un nome utente, verificare che il nome utente visualizzato sia corretto. Continuare con la procedura fino al completamento dell'attivazione



ITA ENG

INFORMAZIONI ▾ DIDATTICA ▾ DOCENTI MOBILITÀ E BORSE DI STUDIO ▾ QUALITÀ ▾ DIDATTICA ON-LINE SOFTWARE ≡

2224 - INGEGNERIA ELETTRICA PER LA E-MOBILITY

HOME > Dipartimenti > Ingegneria > 2224 - INGEGNERIA ELETTRICA PER LA E-MOBILITY



Sede: PALERMO
Anni attivi: I,II,III
Classe: L-9
COORDINATORE: FABIO VIOLA

Didattica erogata
Calendario didattico
Orario delle lezioni
Calendario Esami

camera semi anecoica

Ascolta

In basso un video sulla nostra camera semi-anechoica, un ambiente schermato dove i nostri studenti testano i disturbi radiati e condotti di dispositivi innovativi realizzati negli altri nostri laboratori

camera semi-anechoica





interfaccia

The screenshot shows the MATLAB software interface. At the top, there is a ribbon menu with tabs for HOME, PLOTS, and APPS. Below the ribbon are several toolbars for file operations, variable management, and code execution. The main workspace is divided into three panes: a file browser on the left showing a folder structure, a large central Command Window with a yellow background, and a Workspace pane on the right. The Command Window contains the text: "COMMAND WINDOW", "area dove è possibile inserire i comandi che vogliamo far eseguire da Matlab", and "Si deve rispettare la sequenza temporale delle istruzioni". The Workspace pane shows a table with columns for Name and Value.



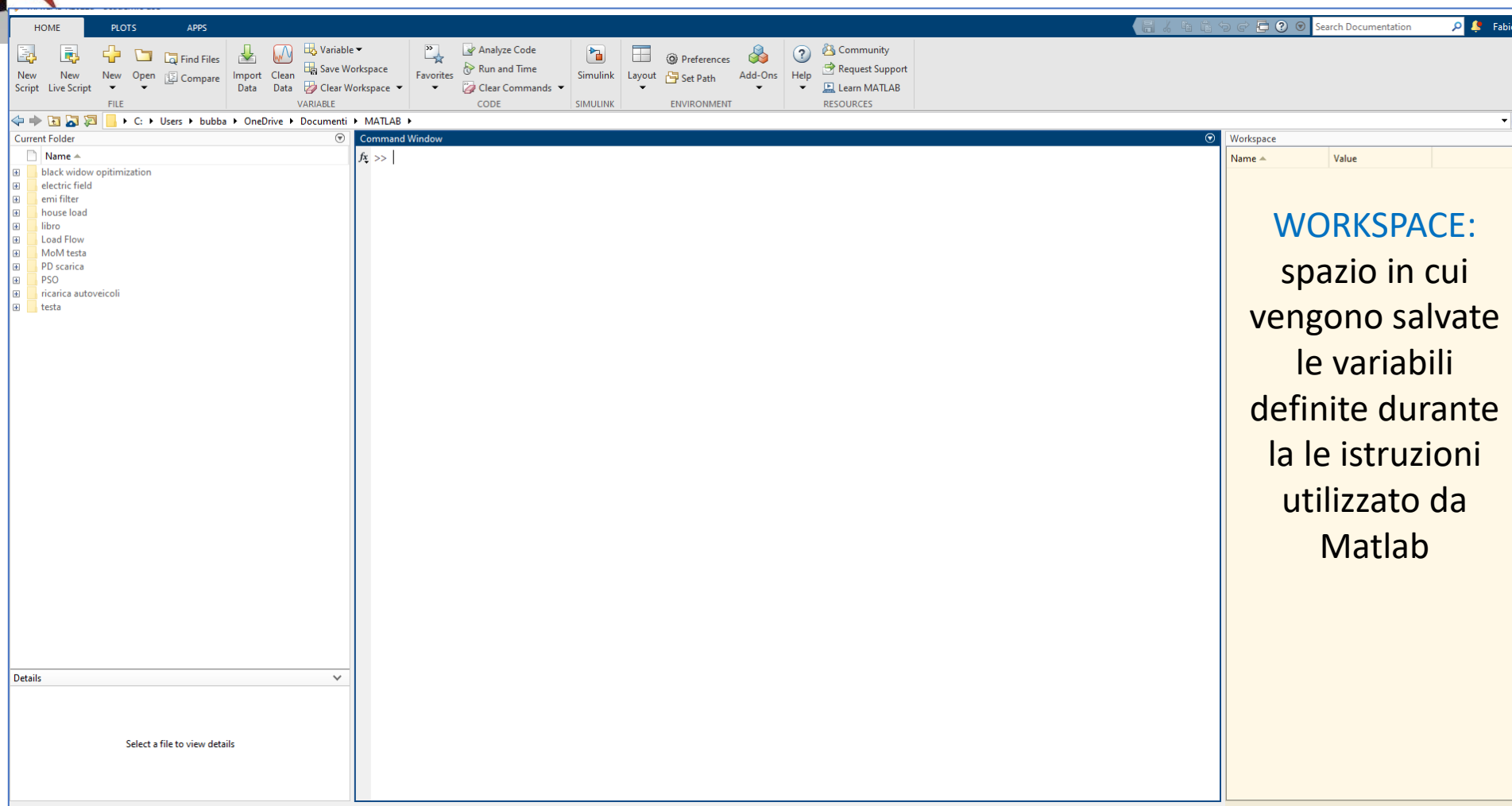
interfaccia

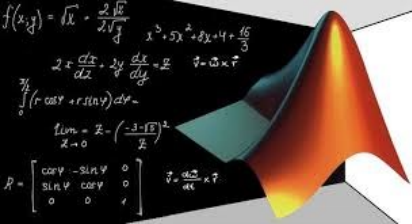
A screenshot of the MATLAB software interface. The top ribbon contains tabs for HOME, PLOTS, and APPS. Below the ribbon is a toolbar with icons for file operations (New Script, New Live Script, New, Open, Find Files, Compare, Import Data, Clean Data, Save Workspace, Clear Workspace), code execution (Analyze Code, Run and Time, Clear Commands), and environment management (Simulink, Layout, Set Path, Add-Ons, Help, Preferences, Community, Request Support, Learn MATLAB). The main window is divided into three panes: 'Current Folder' on the left, 'Command Window' in the center, and 'Workspace' on the right. The 'Current Folder' pane shows the path 'C:\Users\bubba\OneDrive\Documents\MATLAB' and a list of subfolders: black widow optimization, electric field, emi filter, house load, libro, Load Flow, MoM testa, PD scarica, PSO, ricarica autoveicoli, and testa. The 'Command Window' shows the prompt 'fx >> |'. The 'Workspace' pane is empty with columns for Name and Value. A large text overlay is present in the 'Current Folder' pane.

CURRENT FOLDER
percorso utilizzato
da Matlab ed
insieme delle
cartelle di lavoro



interfaccia





Primi passi

Command window
Barra di comando

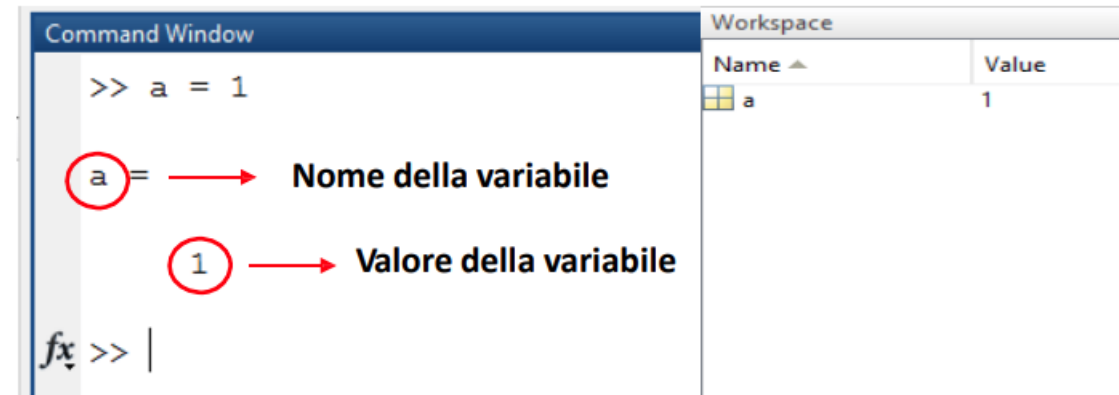
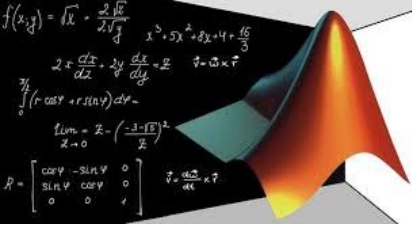
Script

Function

La programmazione può avvenire in modo differente:

- 1) Tramite barra di comando, eseguendo dei passaggi che abbiamo noi in mente
- 2) Su script (copione) eseguendo istruzioni da noi inserite in una sequenza
- 3) Tramite funzioni che possono essere richiamate all'occorrenza

Barra di comando



Sintassi base

Per definire una nuova variabile, in Matlab = e affiancarla al suo valore numerico.

```
>> a = 1
```

Matlab è case-sensitive: la variabile "Temperatura" è diversa dalla variabile "TemPeraTura"

Operazioni tra scalari

a+b somma tra scalari

```
>> a = 1
```

a =

1

```
>> b = 3
```

b =

3

```
>> c = a + b
```

c =

4

a-b differenza tra scalari

```
>> d = a - b
```

d =

-2

a*b prodotto tra scalari

```
>> e = a * b
```

e =

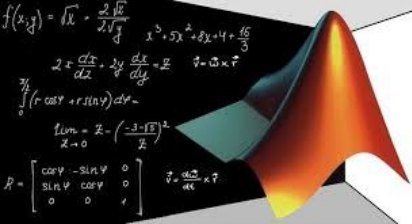
3

a/b rapporto tra scalari

```
>> f = a/b
```

f =

0.3333



Barra di comando

Sintassi base

Un vettore riga viene definito tra parentesi quadre separando i singoli elementi da uno spazio oppure da una virgola. Un vettore colonna viene definito tra parentesi quadre separando i singoli elementi da un punto e virgola. Per trasformare un vettore riga in un vettore colonna si usa l'apice `.

```
>> v1 = [ 1 2 3]
```

```
v1 =
```

```
     1     2     3
```

```
>> v2 = [1;2;3]
```

```
v2 =
```

```
     1  
     2  
     3
```

```
>> v3 = v2'
```

```
v3 =
```

```
     1     2     3
```

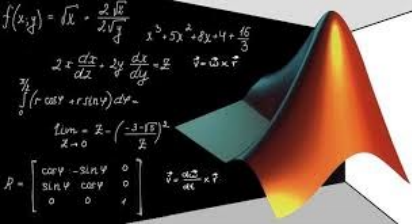
Per accedere all'elemento v(i) del vettore:

```
>> v1 = [ 1 2 3]
```

```
>> v1(2)
```

```
ans =
```

```
     2
```



Barra di comando

Sintassi base

In un array (vettore) si possono facilmente individuare massimo e minimo:

```
>> v1 = [ 1 2 3]
```

```
>> max(v1)
```

```
ans =
```

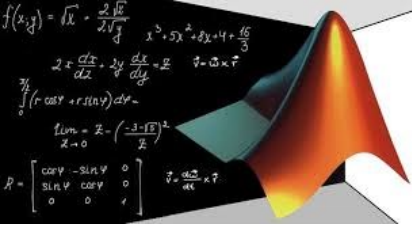
```
3
```

```
>> min(v1)
```

```
ans =
```

```
1
```

Barra di comando



Si possono estendere le operazioni base ai vettori:

```
>> v1
```

```
v1 =
```

```
1     2     3
```

```
>> v2
```

```
v2 =
```

```
4     5     6
```



```
>> v1+v2
```

```
ans =
```

```
5     7     9
```

Gli elementi che occupano lo stesso posto nel vettore sono sommati tra di loro

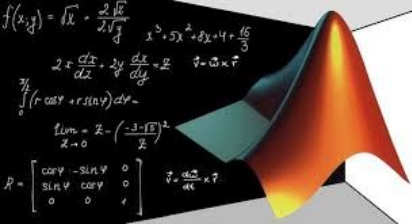
```
>> v1.*v2
```

```
ans =
```

```
4    10    18
```

Gli elementi che occupano lo stesso posto nel vettore sono moltiplicati tra di loro, ma serve mettere il . prima di *

L'istruzione v1*v2 darebbe come risultato 32 facendo la somma di 4 +10 +18



Barra di comando

Se i vettori sono composti da una sequenza ordinata di valori è possibile inserirli con la funzione linspace:

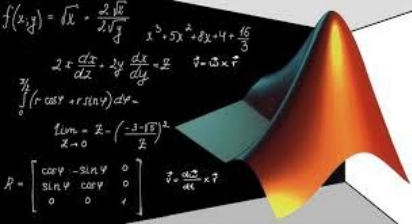
```
>> linspace(0,5,10)
```

```
ans =
```

```
0    0.5556    1.1111    1.6667    2.2222    2.7778    3.3333    3.8889    4.4444    5.0000
```

```
>>
```

I primi due input del comando linspace corrispondono agli estremi del vettore desiderato, mentre il terzo input corrisponde al numero totale di elementi che si desiderano nel vettore.



Barra di comando

I seguenti esempi sono utilizzati per fornire una conoscenza delle comuni routine di rappresentazione grafica di MATLAB ed esercitarsi nell'utilizzo della barra di comando. Dallo studio di questo breve tutorial lo studente sarà in grado di rappresentare grafici bidimensionali, includendo differenti tracce, etichette, titoli e leggende.

Sintassi base

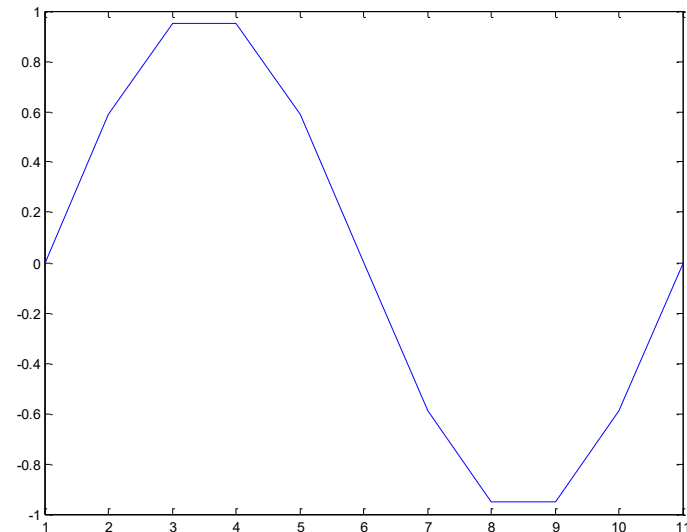
Il comando utilizzato dal Matlab per creare dei grafici è il plot. Questo comando richiede l'inserimento di alcune informazioni, sempre più dettagliate man mano che si raffina la rappresentazione.


Ingresso singolo

L'utilizzo più semplice del comando plot prevede l'utilizzo di un vettore numerico come argomento.

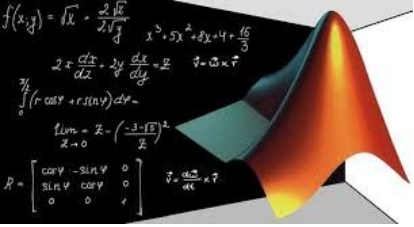
Per esempio:

```
>> x=10:20;  
>> y=sin(2*pi*x/10);  
>> plot(y)
```



 **YouTube** <https://youtu.be/7pYxGtn-aPo>

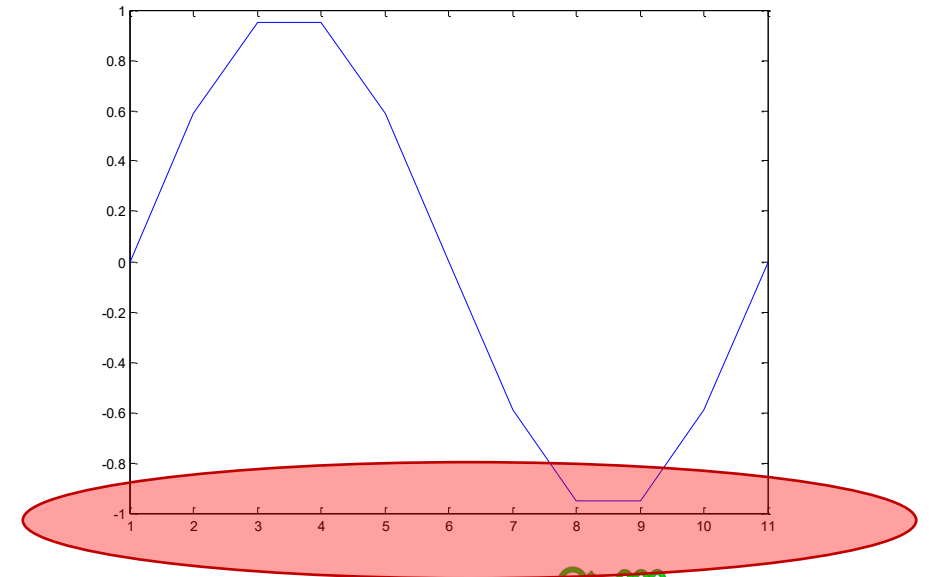
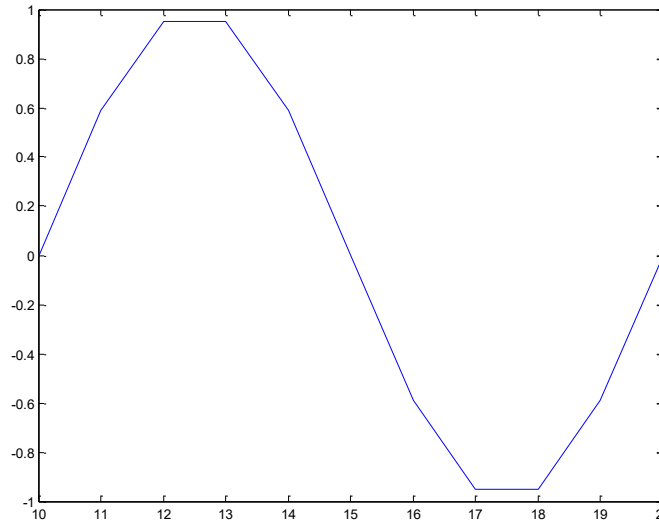
Barra di comando



Due ingressi (due vettori)

Per aggiungere al precedente grafico i valori della funzione argomento x si procede come nell'esempio riportato. Ovviamente x e y devono avere le stesse dimensioni. (11 campioni)

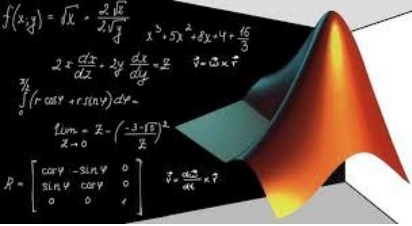
```
>> x=10:20;  
>> y=sin(2*pi*x/10);  
>> plot(x, y)
```



osservazioni

Il campionamento x=10:20 inserisce elementi unitari tra il valore 10 e 20, realizzando una spaziatura grezza

Barra di comando



Puliamo il workspace

Liberiamo il workspace dai precedenti calcoli

```
>> clear
```

Puliamo la command window

Liberiamo la command windows dalle istruzioni scritte precedentemente

```
>> clc
```

Due ingressi (una matrice, un vettore)

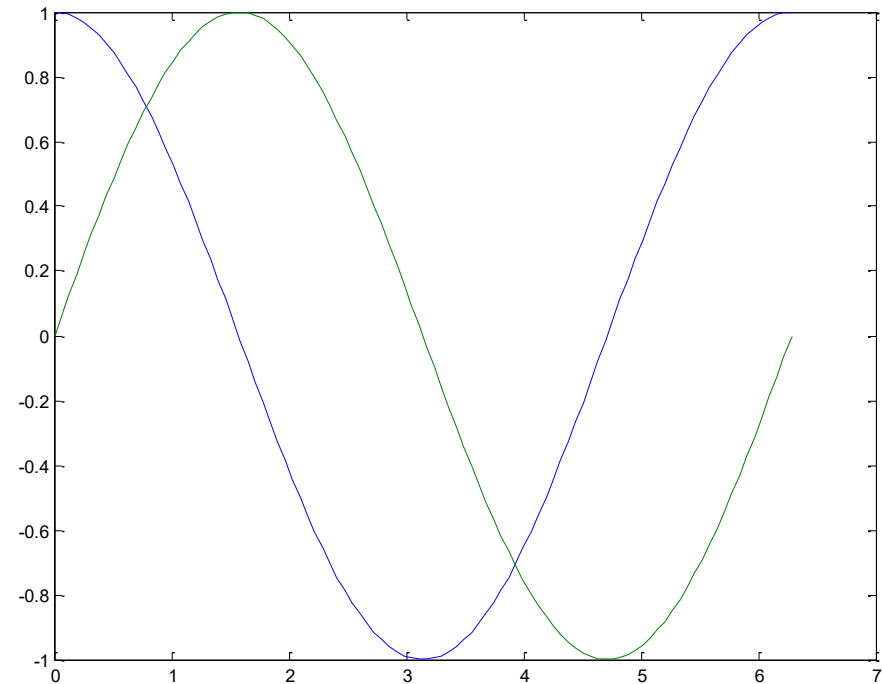
Un altro esempio della sintassi a due ingressi è quando y è dato da una matrice. In questo caso, ogni riga in y è tracciata in funzione dell'argomento x, da cui dipendono le righe s. Nel seguente esempio, sia la x che y hanno lo stesso numero di colonne (100):

```
>> x = linspace(0, 2*pi, 100);
>> y(1, :) = cos(x);
>> y(2, :) = sin(x);
>> plot(x, y);
```

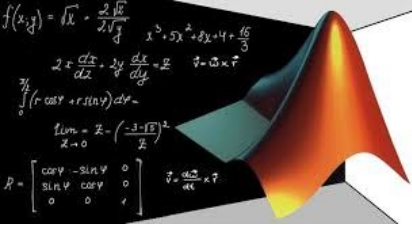
 **YouTube** <https://youtu.be/6J-gvQExH68>

osservazioni

Quando sia la x che la y sono matrici, i risultati non sono necessariamente intuitivi. La funzione linspace crea una mappatura più fine con 100 campioni



Barra di comando

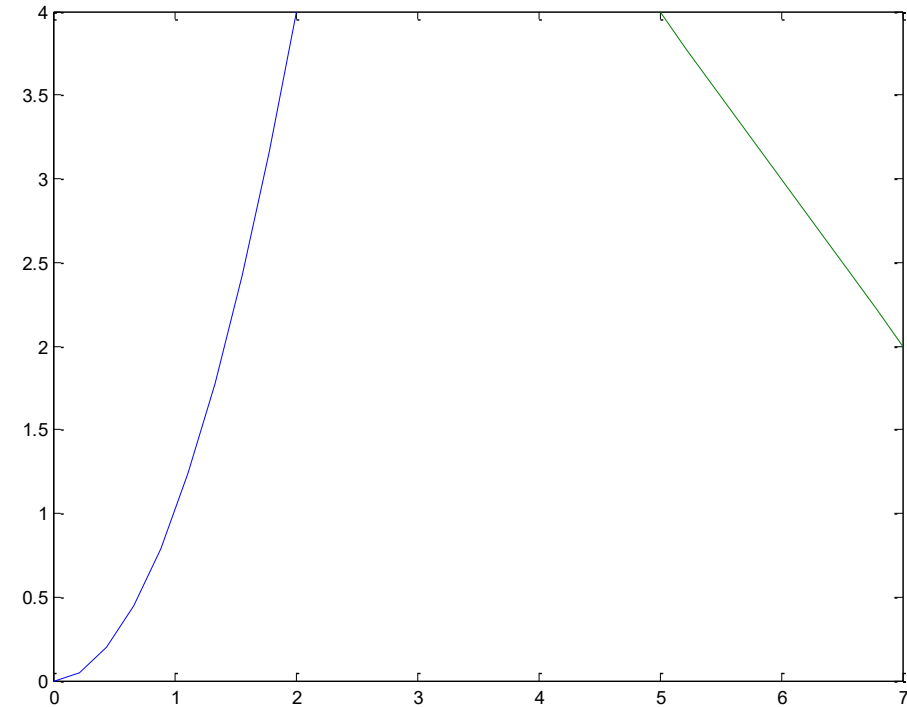


Ingressi multipli

Si possono poi tracciare sullo stesso grafico andamenti che non hanno ascisse in comune .

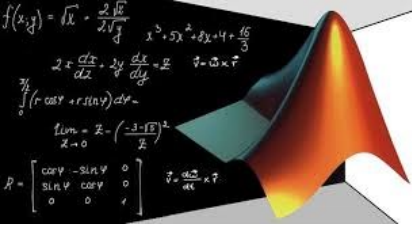
Consideriamo l'esempio:

```
>> x1 = linspace(0,2,10);
>> x2 = linspace(5,7,10);
>> y1 = x1.^2;
>> y2 = 9-x2;
>> plot(x1, y1, x2, y2);
```



 **YouTube** <https://youtu.be/cBPiorMZ-hs>

Barra di comando



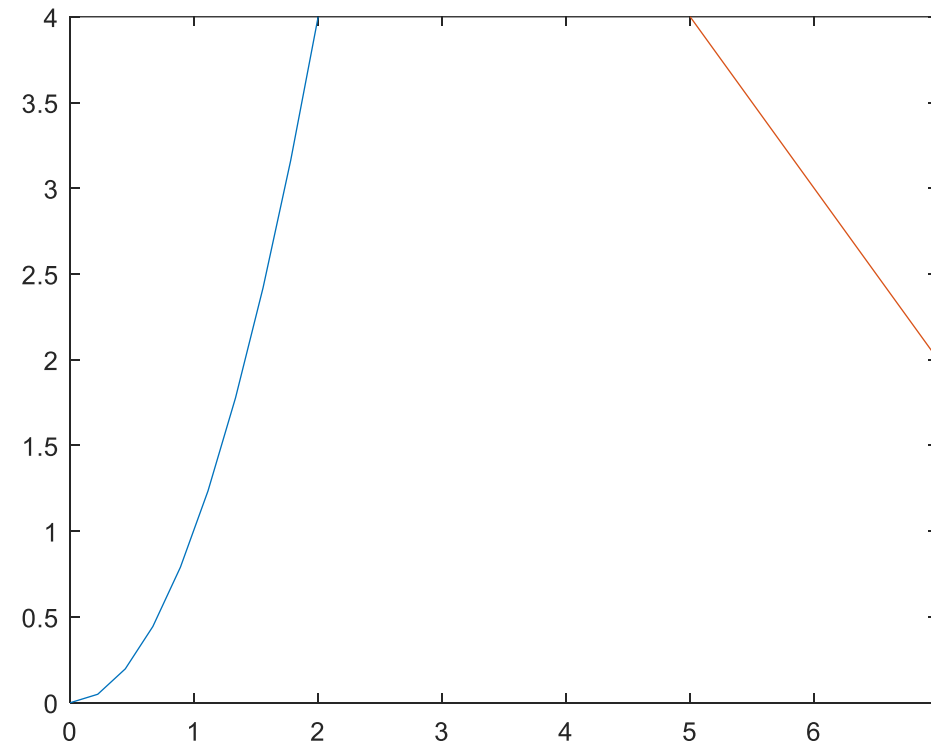
Comando hold

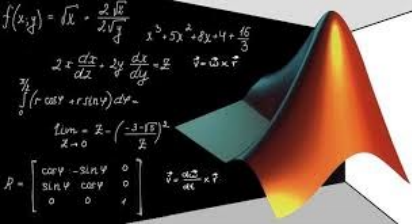
Per concludere, i diagrammi multipli possono essere tracciati usando il comando hold on ed il comando hold off.

Il comando hold on ordina a MATLAB di mantenere tutte le tracce sul diagramma in uso e di aggiungere le eventuali tracce supplementari.

Il comando hold off impone a MATLAB di cessare di rappresentare gli andamenti nel foglio corrente.

```
>> x1 = linspace(0,2,10);
>> x2 = linspace(5,7,10);
>> y1 = x1.^2;
>> y2 = 9-x2;
>> plot(x1, y1);
>> hold on;
>> plot(x2, y2);
>> hold off;
```





Barra di comando

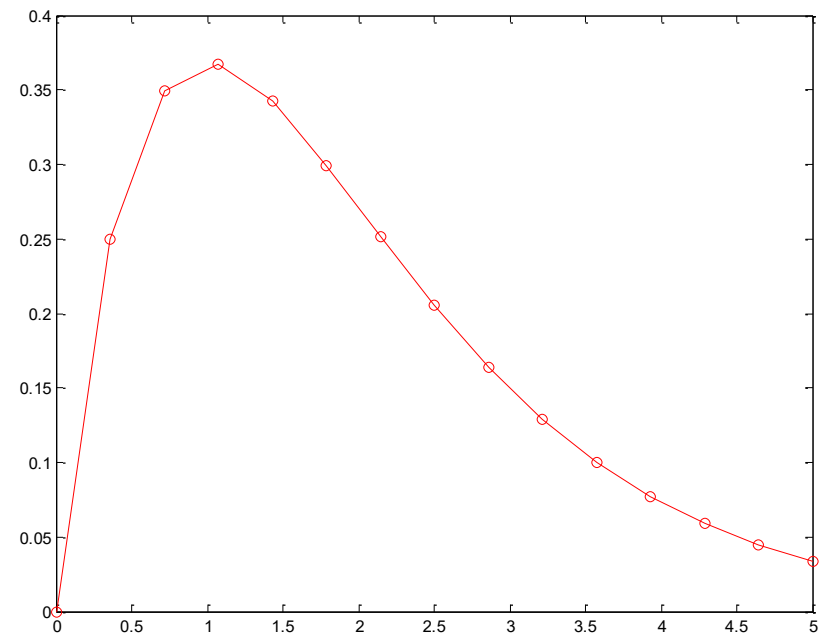
Opzioni della rappresentazione

Ci sono tre opzioni principali per ogni traccia: colore, simbolo e stile della linea.

Queste opzioni sono fornite in una stringa (cioè circondato dalle singoli apici) dopo gli argomenti della curva da tracciare.

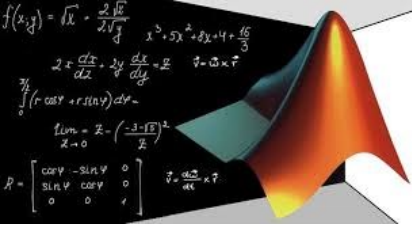
Per esempio, la curva sotto è di una linea rossa con un cerchietto o ad ogni punto e ad una linea continua che collega i cerchietti:

```
>> x = linspace(0,5,15);  
>> y = x.*exp(-x);  
>> plot(x, y, 'ro-');
```



<https://youtu.be/Q6aymGnN7ZE>

Barra di comando



Colori

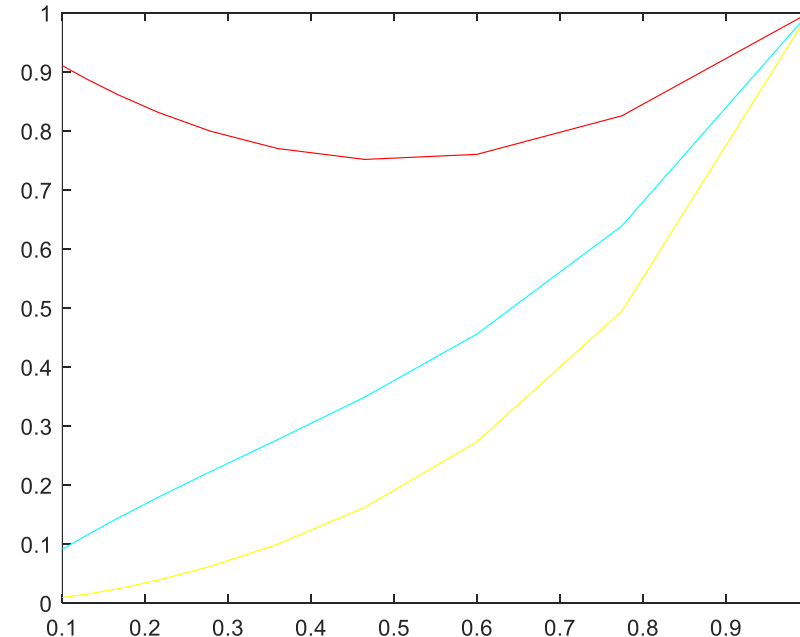
Ogni curva può avere un singolo codice di colore. I colori disponibili sono 8 e sono (b)lue, (g)reen, (r)ed, (c)yan, (m)agenta, (y)ellow e (w)hite. Se si usa la forma multipla di ingresso non è necessario specificare i colori, le tracce differenti saranno dal colore azzurro al colore giallo; il bianco non è usato poiché gli sfondi sono generalmente bianchi.

codice 1:

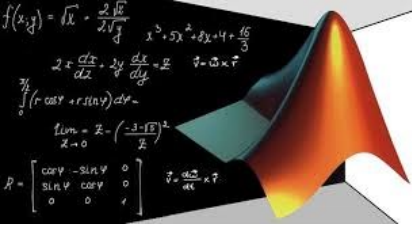
```
>> x = logspace(-1,0,10);
>> y1 = x.^2-x+1;
>> y2 = x.^3-x.^2+x;
>> y3 = x.^4-x.^3+x.^2;
>> plot(x, y1, 'r', x, y2, 'c', x, y3, 'y');
```

codice 2:

```
>> x = logspace(-1,0,10);
>> y1 = x.^2-x+1;
>> y2 = x.^3-x.^2+x;
>> y3 = x.^4-x.^3+x.^2;
>> plot(x, y1, 'r');
>> hold on;
>> plot(x, y2, 'c');
>> plot(x, y3, 'y');
>> hold off;
```



Barra di comando



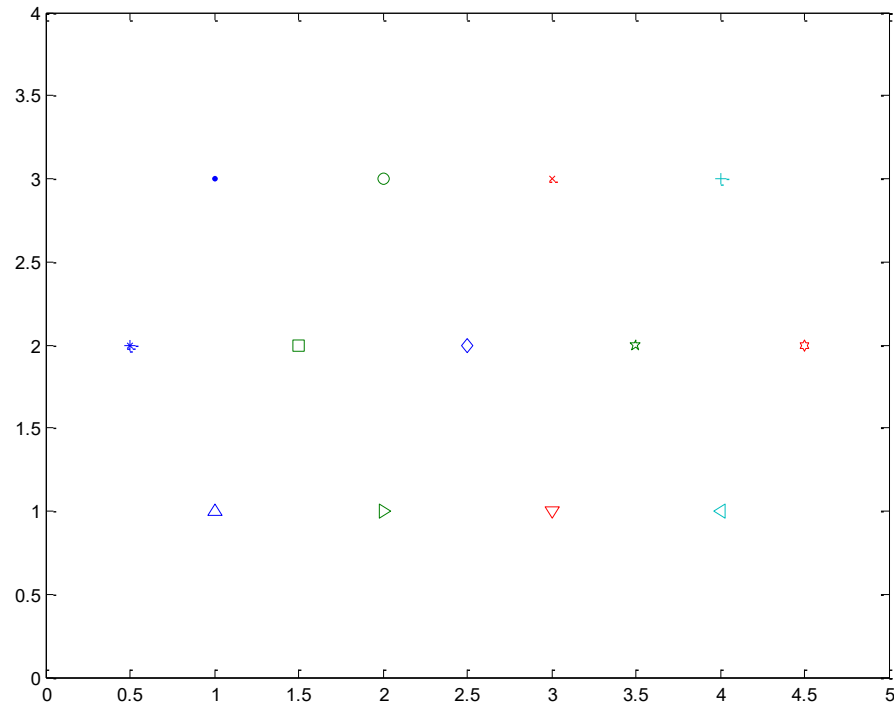
Simboli nelle tracce

La versione corrente di MATLAB utilizzata ha 13 simboli differenti che possono essere disposti ad ogni punto di computazione.

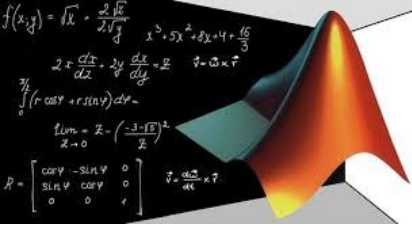
Includono il punto (.), il cerchio (o), la croce (x), il più (+), la stella (*), il (s) quadrato, il diamante (d), il pentagramma (p), l'exagramma (h) ed i triangoli che indicano in quattro sensi (^, >, v e <).

Il grafico sotto mostra questi simboli:

```
>> plot(1,3,'.',2,3,'o',3,3,'x',4,3,'+');
>> hold on
>> plot(0.5,2,'*',1.5,2,'s')
>> plot(2.5,2,'d',3.5,2,'p', 4.5, 2, 'h');
>> plot(1,1,'^',2,1,'>',3,1,'v',4,1,'<');
>> axis([0 5 0 4])
```



Barra di comando



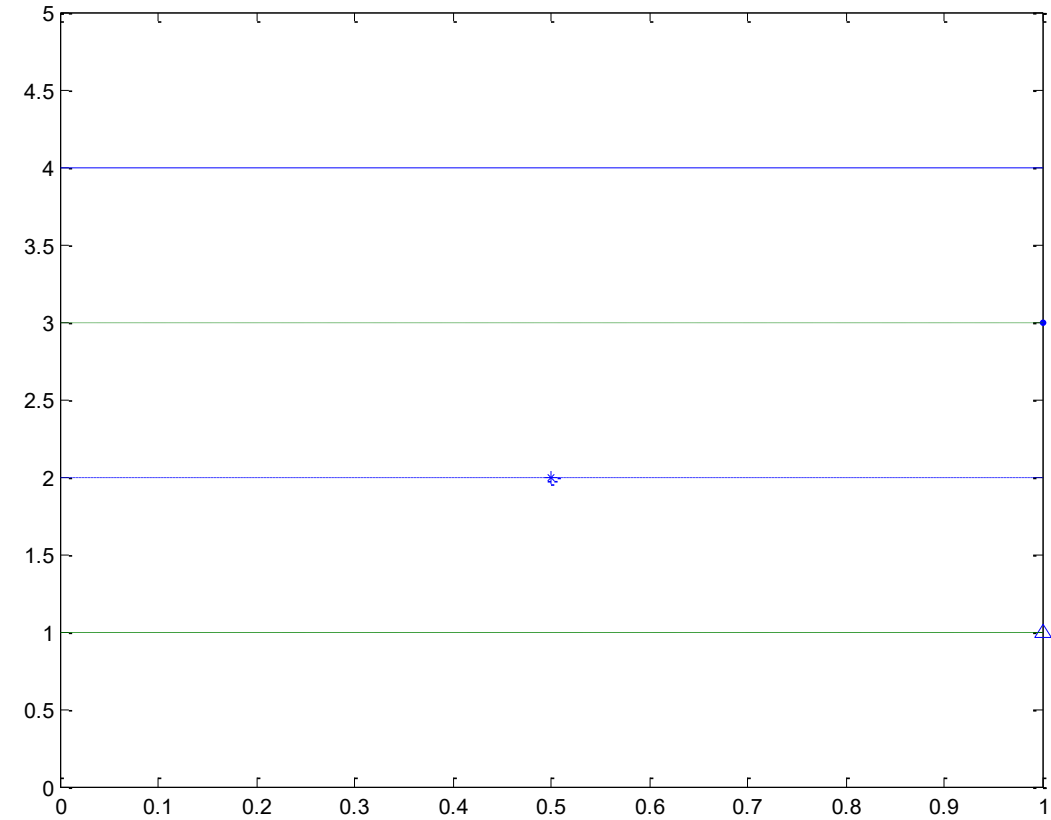
Stili di tratteggio

MATLAB inoltre ha 4 stili differenti di rappresentazione della linea.

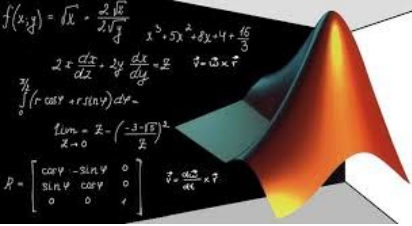
Sono tratto continuo (-), punteggiato (:), il tratto-punto (-.), ed il tratteggiato(--).

Il grafico seguente mostra ciascuno di questi stili:

```
>> x = [0 1];
>> y = [1 1];
>> plot(x, 4*y, '-', x, 3*y, ':');
>> hold on;
>> plot(x, 2*y, '-.', x, 1*y, '--');
>> hold off;
>> axis([0 1 0 5]);
```



Barra di comando

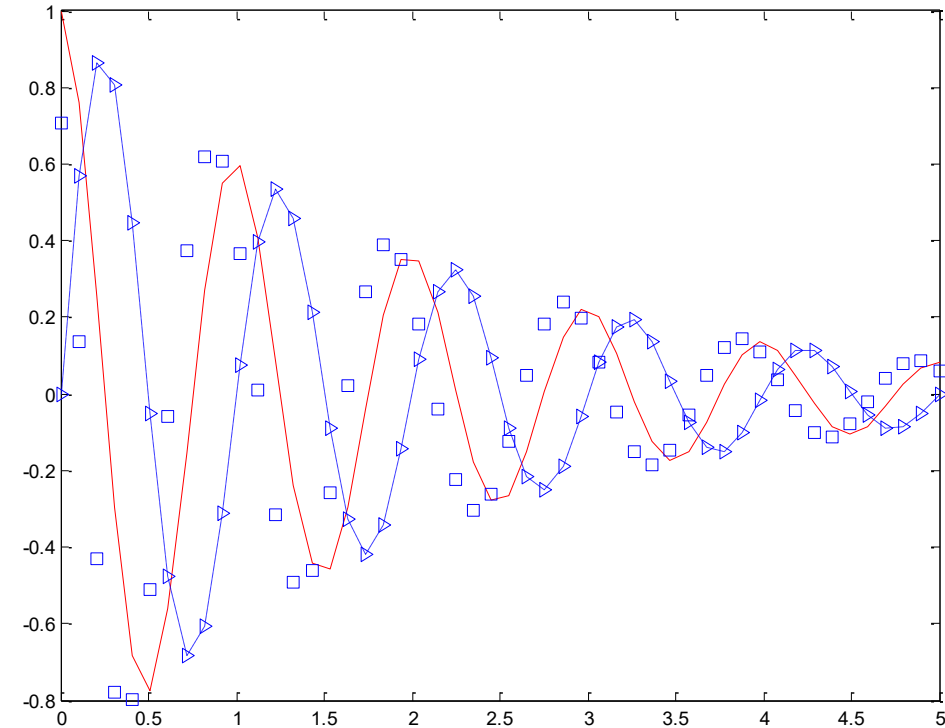


Combinazioni

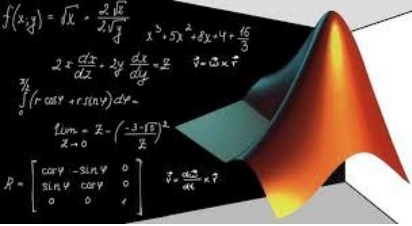
Con MATLAB, si possono usare combinazione di colori, simboli e di stili di tratteggio.

```
>> x = linspace(0, 5, 50);
>> y1 = cos(2*pi*x).*exp(-x/2);
>> y2 = sin(2*pi*x).*exp(-x/2);
>> y3 = cos(2*pi*x+pi/4).*exp(-x/2);
>> plot(x,y1,'r-',x,y2,'--',x,y3,'bs');
```

Per la prima curva, l'assenza di una specificazione del simbolo richiede che nessun simbolo sia tracciato. Per la seconda linea, la mancanza di specifica di colore fa sì che ci si riferisca al primo colore nella rotazione (blu). Per la terza linea, la mancanza di specifica della linea significa che nessuna linea è tracciata.



Barra di comando



Etichettatura

Descrizione assi e titolo

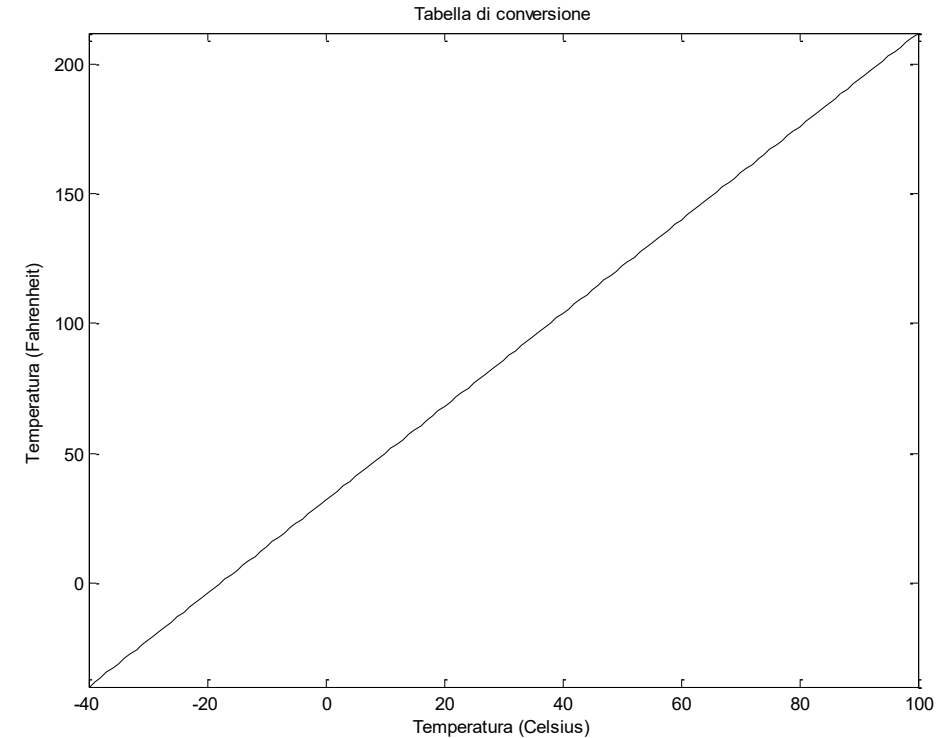
Il comando xlabel etichetta l'asse delle ascisse mentre ylabel etichetta l'asse delle ordinate.


Il comando title aggiunge il titolo del diagramma.

Il comando e grid on e grid off dispongono o rimuovono un reticolo dalla figura.

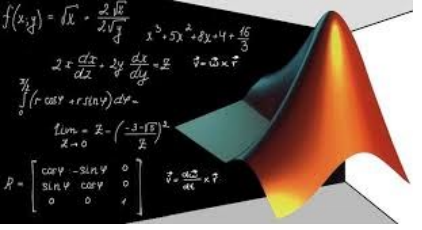
Per concludere il comando axis permette di specificare i limiti degli assi.

```
>> TC = -40:1:100;
>> TF = TC*1,8+32;
>> plot(TC, TF, 'k-');
>> xlabel('Temperatura (Celsius)');
>> ylabel('Temperatura (Fahrenheit)');
>> title('Tabella di conversione');
>> axis([-40 100 min(TF) max(TF)]);
```



 **YouTube** <https://youtu.be/gTCyJtUmKs8>

Barra di comando

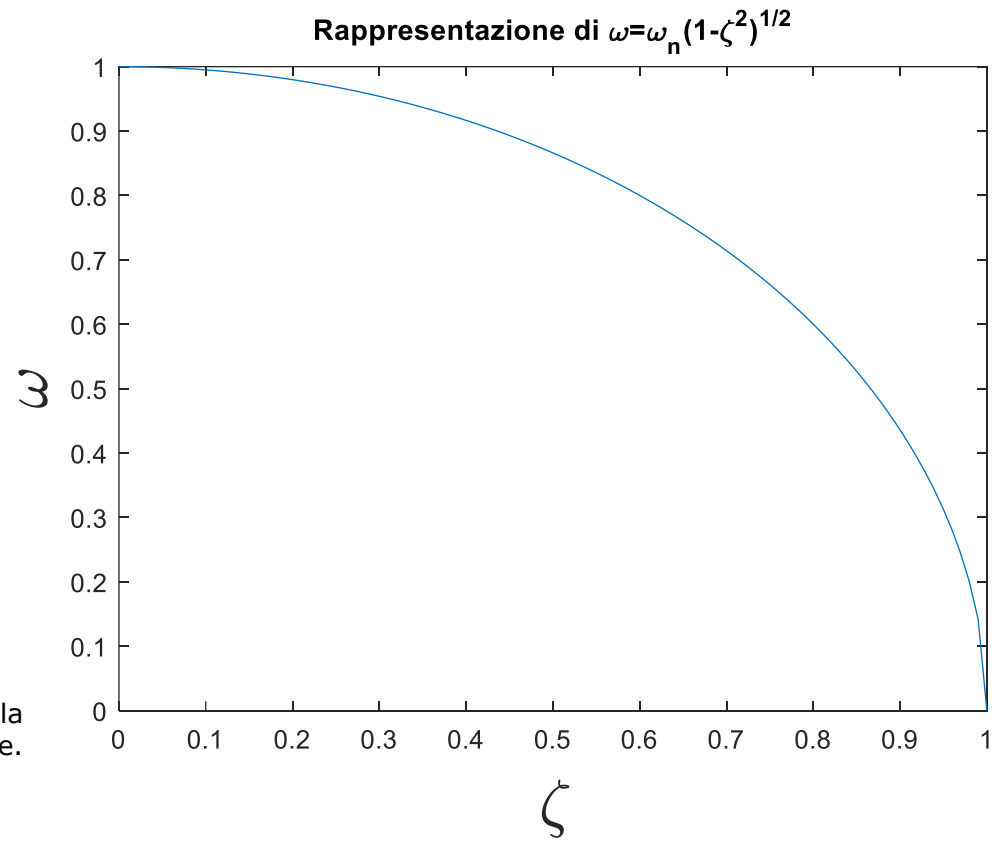


Etichettatura

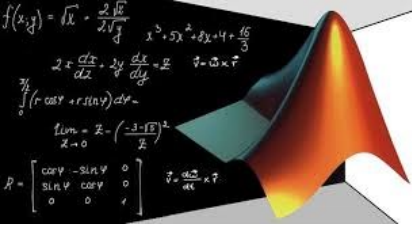
Ci sono diverse opzioni per xlabel, ylabel e per il titolo. La maggior parte di queste vanno oltre l'intento di questa breve guida, ma per completare la descrizione dell'etichettatura dei diagrammi si deve introdurre la formattazione LaTeX ed il dimensionamento dei caratteri. Usando la tipologia di comandi LaTeX si possono inserire lettere greche, pedici ed apici. Usando il comando FontSize, si può potete cambiare il formato del testo.

```
>> omega_n=1;
>> zeta = linspace(0, 1, 100);
>> omega = omega_n*sqrt(1-zeta.^2);
>> plot(zeta, omega);
>> xlabel('\zeta', 'FontSize', 24);
>> ylabel('\omega', 'FontSize', 24);
>> title('Rappresentazione di \omega=\omega_n(1-\zeta^2)^{1/2}');
```

In generale, le lettere greche possono essere ottenute scrivendo una slash seguito dalla lettera greca. Apici e pedici sono generati disponendo ^ o _ dopo il carattere da utilizzare. Per inserire una parola è necessario che il testo sia accluso dentro { }.



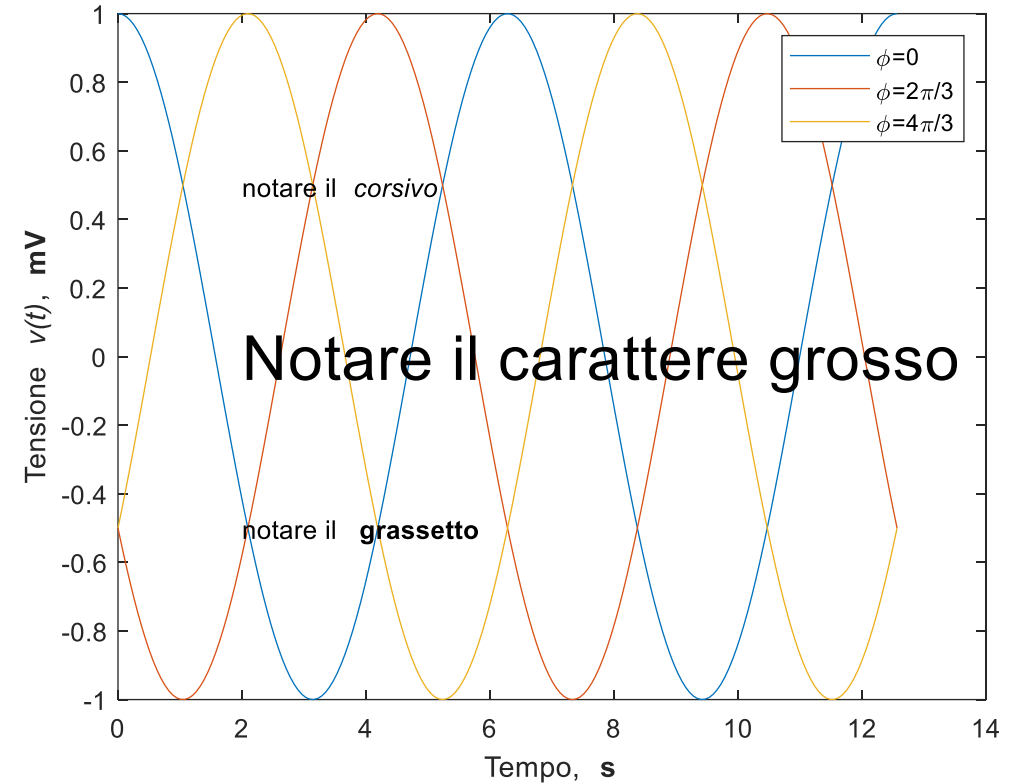
Barra di comando

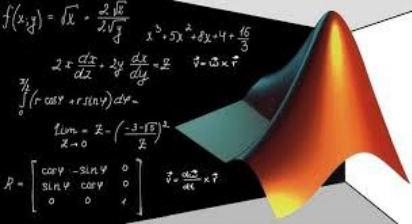


Legenda e testo

E' possibile inserire una legenda nel grafico. Il comando legend consente di specificare il nome per ogni curva seguendo l'ordine di ingresso dei dati nella funzione plot. Il comando text permette che specificiate una posizione per una stringa di testo ed il suo contenuto. Il comando gtext consente di specificare il testo solamente e la posizione sarà specificata mediante utilizzo del mouse.

```
>> t = linspace(0, 4*pi, 200);
>> v1 = cos(t);
>> v2 = cos(t+2*pi/3);
>> v3 = cos(t+4*pi/3);
>> plot(t, v1, t, v2, t, v3);
>> legend('\phi=0', '\phi=2\pi/3', '\phi=4\pi/3');
>> xlabel('Tempo, {\bf s}');
>> ylabel('Tensione {\it v(t)}, {\bf mV}');
>> text(2, 0.5, 'notare il {\it corsivo}');
>> text(2, 0, 'Notare il carattere grosso', 'FontSize', 24);
>> text(2, -0.5, 'notare il {\bf grassetto}');
```





Barra di comando

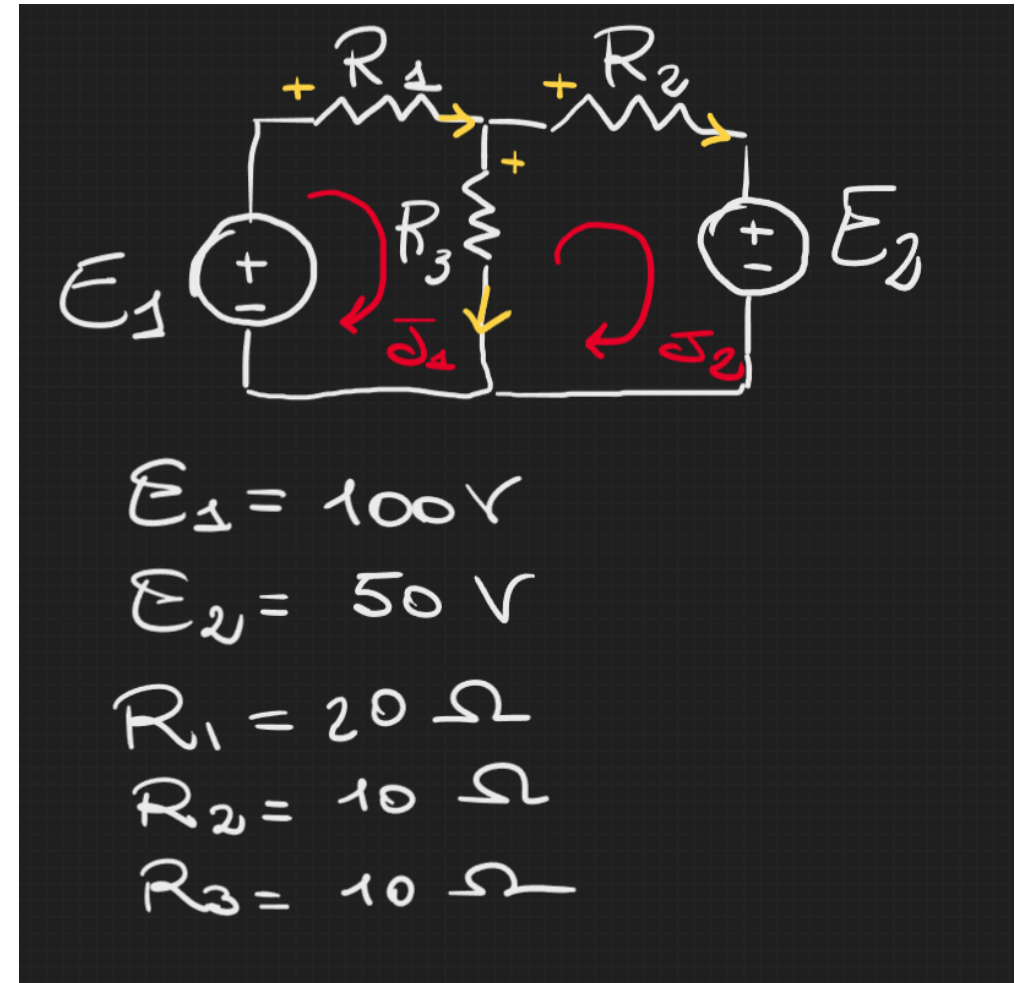
Utilizzo delle matrici

E' possibile risolvere problemi dell'ingegneria elettrica sfruttando le operazioni tra vettori e matrici.

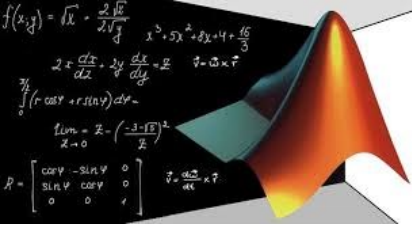
L'equazione matriciale è data da $[R][J] = [E]$, con $[J]$ vettore delle incognite



<https://youtu.be/Q6aymGnN7ZE>



Barra di comando



Nello specifico con il metodo delle correnti d'anello ed i valori riportati si scrive:

```
>> R = [30 -10; -10 20];
>> E = [100; -50];
J = inv(R)*E
```

Ottenendo J = [3; -1]

```
Command Window
>> R = [30 -10;-10 20]

R =

    30    -10
   -10     20

>> E = [100; -50]

E =

    100
    -50

>> J =inv(R)*E

J =

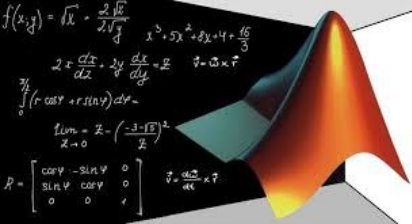
    3.0000
   -1.0000

fx >> |
```

$E_1 (R_1 + R_3) J_1 - R_3 J_2 = E_1$
 $-R_3 J_1 + (R_2 + R_3) J_2 = -E_2$

$E_1 = 100 \text{ V}$
 $E_2 = 50 \text{ V}$
 $R_1 = 20 \ \Omega$
 $R_2 = 10 \ \Omega$
 $R_3 = 10 \ \Omega$

$$\begin{bmatrix} R_1 + R_3 & -R_3 \\ -R_3 & R_2 + R_3 \end{bmatrix} \begin{pmatrix} J_1 \\ J_2 \end{pmatrix} = \begin{pmatrix} E_1 \\ -E_2 \end{pmatrix}$$



Primi passi

Command window
Barra di comando

Script

Function

Con questa presentazione termina l'utilizzo della barra di comando. Nella prossima sarà affrontato l'utilizzo degli script.