



---

# Laboratorio di Matlab

Alessandro Formaglio

Dipartimento di Ingegneria dell'Informazione,  
Università di Siena

[alex@dii.unisi.it](mailto:alex@dii.unisi.it)

<http://www.dii.unisi.it/~control/MatLab/LabMatlab.html>

Martedì 14 Luglio 2009



## Grafica

- Grafica 2D
- Traiettorie in  $\mathbb{R}^3$
- Superfici



## Grafica 2D

Si rappresentano funzioni di una variabile:  $y = f(x)$

### Esempio:

Funzioni del tipo:  $c(x) = \cos(x)$  e  $s(x) = \sin(x)$

1. Definizione del vettore 'x' :

```
>> x = 0:0.2:4*pi;
```

2. Generazione delle due funzioni calcolate su 'x' e plot:

```
>> c = cos(x);
```

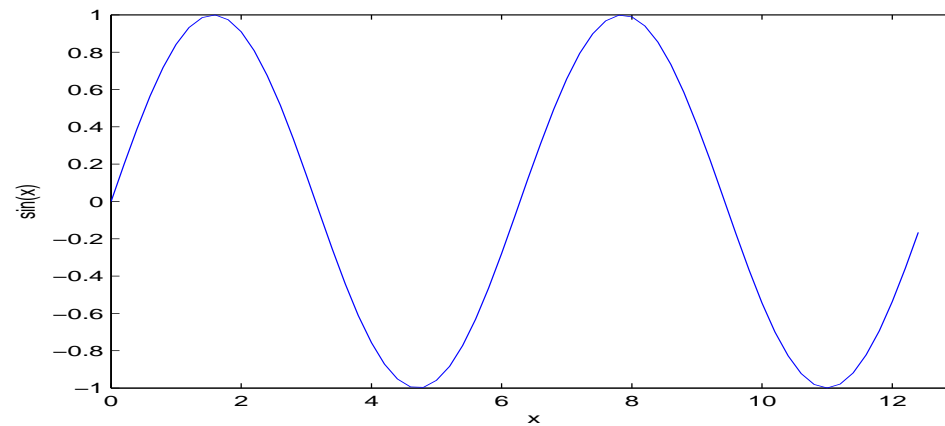
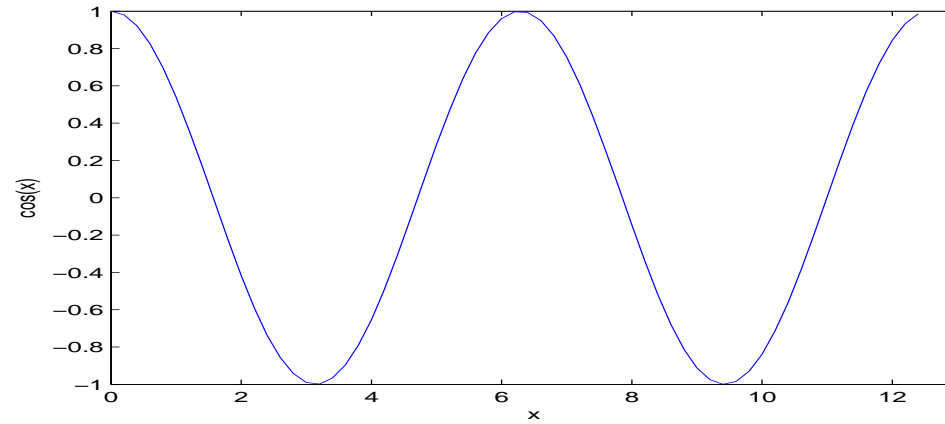
```
>> s = sin(x);
```

```
>> figure;
```

```
>> plot(x,c);
```

```
>> figure;
```

```
>> plot(x,s);
```

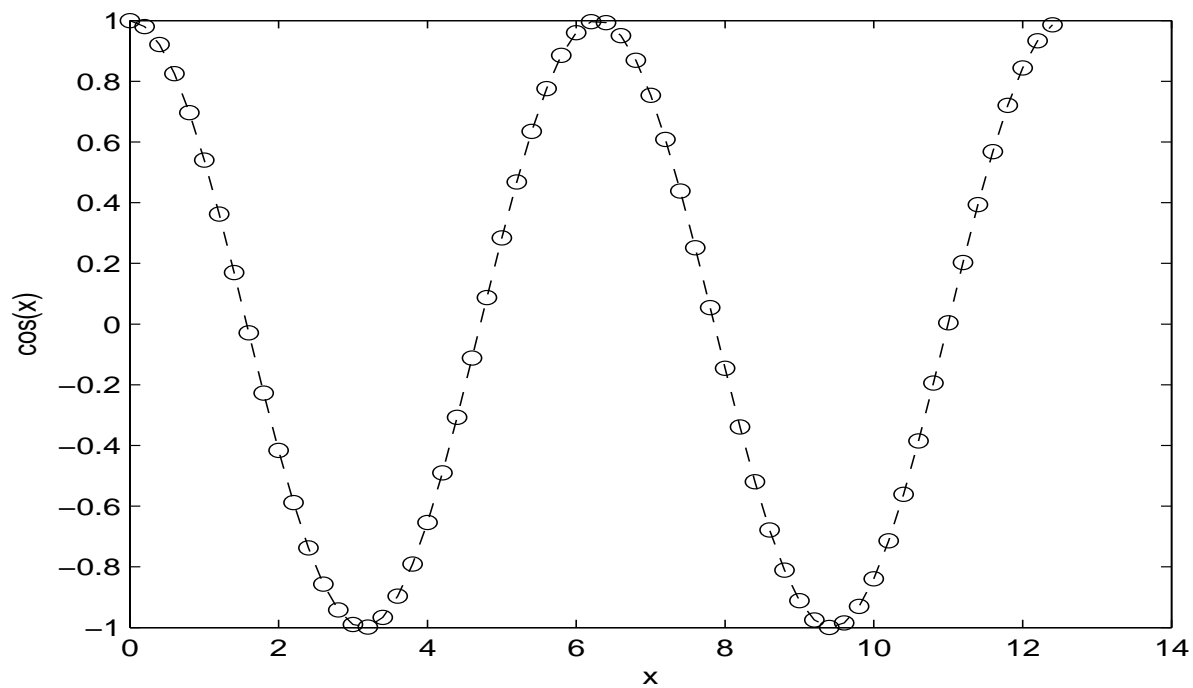


## Funzioni correlate

- `h = figure` genera una nuova istanza dell'oggetto figura ed assegna alla variabile `h` un *handle* a tale figura
- `close all` chiude tutte le figure aperte
- `plot(x, y)` plotta sul sistema  $O_{xy}$  tutte le coppie  $(x_i, y_i)$  congiungendo i punti successivi con una linea
- `plot(x, y, 'tratto|simbolo|colore')` definisce il tipo di tratto che congiunge punti successivi, il simbolo associato ad ogni punto, il colore

**Es.:**

```
>> plot(x, c, '--ok')
```



carattere	tratto	carattere	simbolo	carattere	colore
-	linea	+	crocetta	r	rosso
--	tratteggio	o	cerchietto	g	verde
:	puntini	*	asterisco	b	blue
-.	tratto punto	x	×	c	celeste
		d	rombo	m	magenta
		s	quadrato	y	giallo
				k	nero
				w	bianco

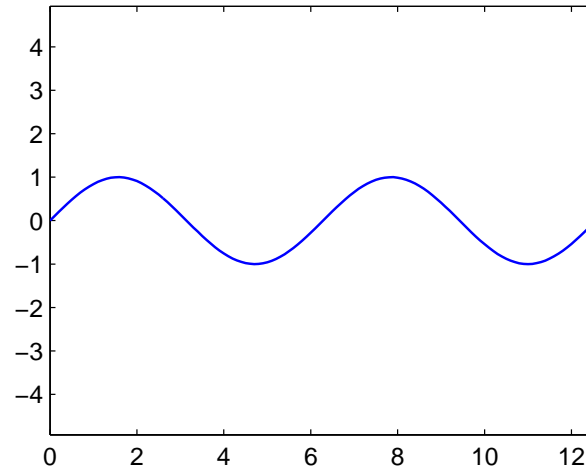
Per saperne di più ...

`help plot`

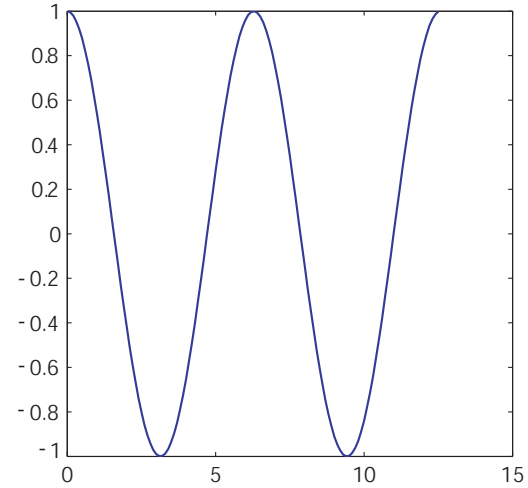
- `hold on` abilita la sovrapposizione di più grafici all'interno della stessa figura
- `hold off` disabilita la sovrapposizione di più grafici all'interno della stessa figura (una nuova istanza di plot elimina tutto quello che era stato tracciato precedentemente sulla figura di riferimento).
- `legend('S1', 'S2', ...)` aggiunge la legenda in base al numero di tracce: ad ogni traccia associa la stringa  $S_i$  corrispondente
- `title('Title')` associa la stringa `Title` al titolo della figura
- `xlabel('Sx')` associa la stringa  $S_x$  all'asse  $x$
- `ylabel('Sy')` associa la stringa  $S_y$  all'asse  $y$

- `axis([xmin xmax ymin ymax])` setta i limiti degli assi
- `axis equal` setta le stesse unità di misura e gli stessi aspect ratio sugli assi
- `axis square` rende la regione di rappresentazione quadrata
- `grid on` aggiunge una griglia alla figura
- `subplot(m, n, p)` crea  $m \times n$  finestre nella stessa figura e attiva la  $p$ -esima da sinistra in alto

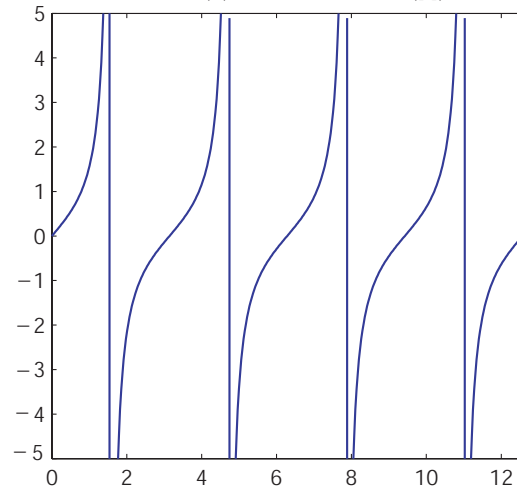
sin(x) con axis equal



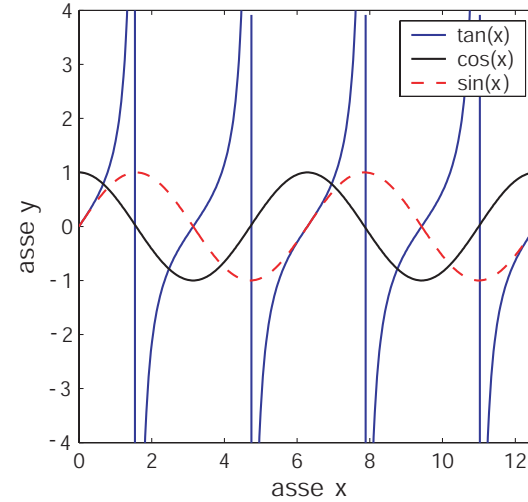
cos(x) con axis square

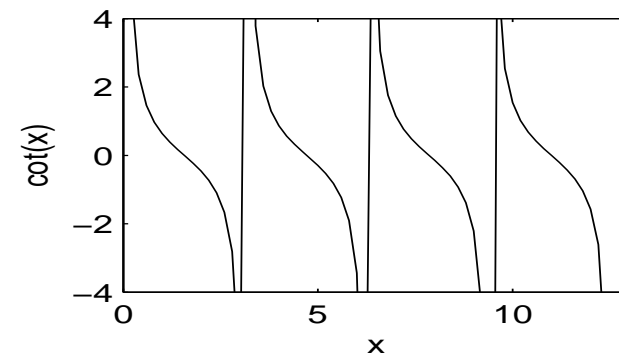
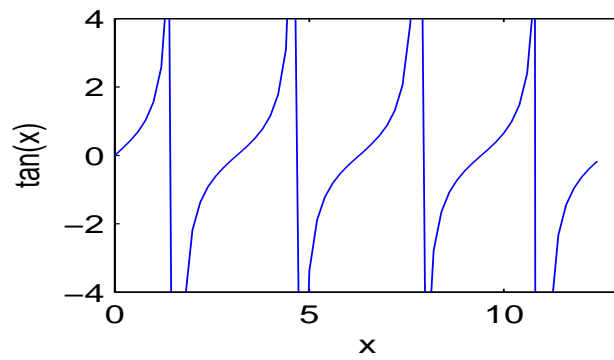
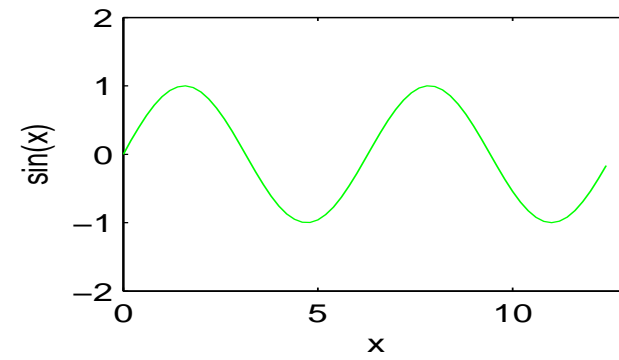
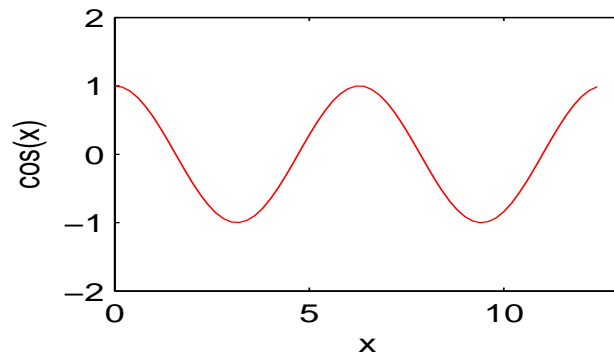


tan(x) con uso di axis([ ])



uso di legend e xlabel e ylabel



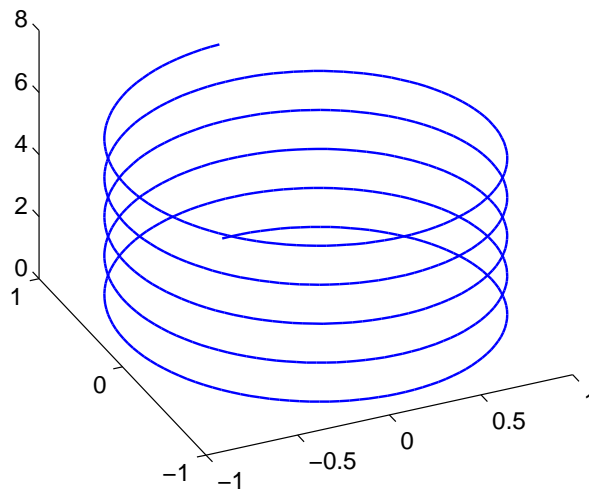


## Traiettorie in $\mathbb{R}^3$

Plot di funzioni  $f : \mathbb{R} \rightarrow \mathbb{R}^3$

`plot3(fx, fy, fz)` visualizza le terne  $(f_x, f_y, f_z)$  nel sistema  $O_{xyz}$

**Esempio:**



```
>> t = 0:0.01:2*pi;
```

```
>> fx = sin(5*t);
```

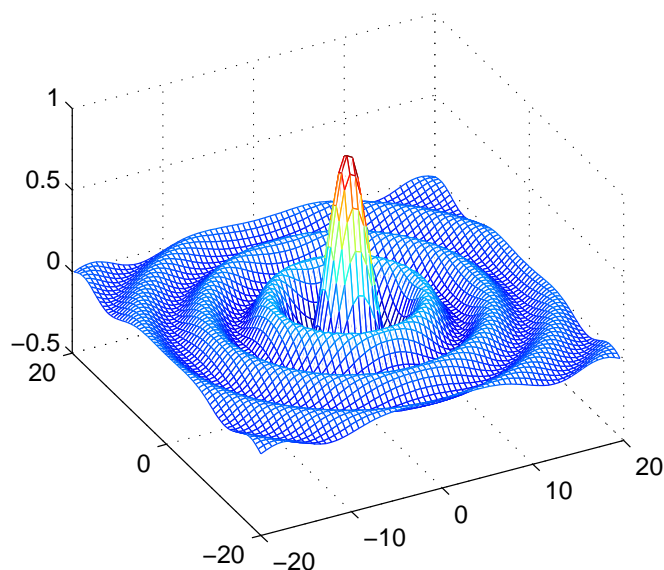
```
>> fy = cos(5*t);
```

```
>> fz = t;
```

```
>> plot3(fx, fy, fz);
```

**Attenzione:** Le dimensioni devono sempre essere concordi!

# Superfici



Plot di funzioni  $f(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$

- `surf(X, Y, Z)` genera superfici
- `mesh(X, Y, Z)` genera griglie superficiali

**Osservazione:**  $f(x, y)$  deve essere definita  $\forall (x, y) \in \mathcal{D}om(f)$

1. Si definiscano due vettori:

$$x = 0:0.2:2*\pi;$$

$$y = 0:0.2:4*\pi;$$

2. Una funzione generale  $z = f(x, y)$  assume valori su tutte le possibili coppie  $(x, y)$  da cui è evidente che  $\dim(z) = \dim(x) \times \dim(y)$

3.  $[X, Y] = \text{meshgrid}(x, y)$ ; genera le matrici:

$X$  contiene copie per righe di  $x$  tante volte quanto è la dimensione di  $y$

$Y$  contiene copie per colonne di  $y$  tante volte quanto è la dimensione di  $x$

4. A questo punto è possibile definire  $Z$ :

$$Z = f(X, Y);$$

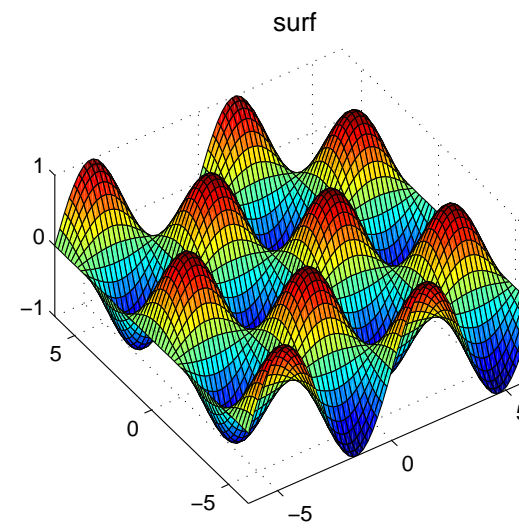
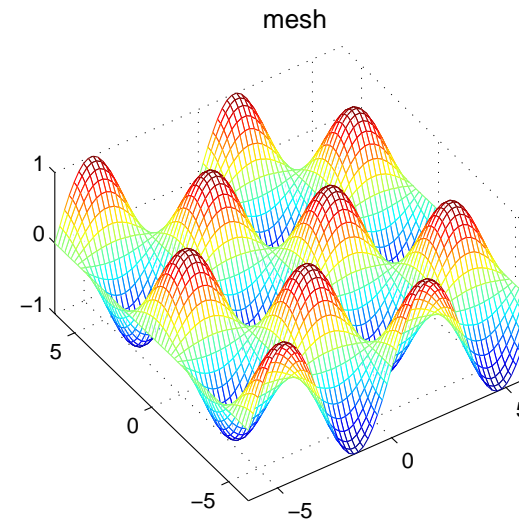
(senza nessun problema di dimensioni: prodotto cartesiano di  $x$  ed  $y$ )



**Esempio:**

```
>> x = -2*pi:pi/15:2*pi;  
>> y = -2*pi:pi/15:2*pi;  
>> [X Y] = meshgrid(x,y);  
>> Z = sin(X).*cos(Y);  
>> mesh(X,Y,Z);  
>> surf(X,Y,Z);
```

colormap definisce la  
colorazione del surf o del  
mesh



## Esercizio 3

1. Definire e plottare la seguente funzione  $f(x) = \begin{cases} x^2 & \text{per } x \leq 1, \\ 1/x & \text{per } x > 1. \end{cases}$
2. Calcolare la derivata di  $f(x)$  sia numericamente che in forma analitica e plottarne i valori
3. Utilizzando `subplot()` plottare nella stessa figura la funzione derivata e la funzione  $f(x)$ . Si utilizzino colori e tratti diversi per ogni funzione.

**Esercizio 4** Calcolare la funzione  $f(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$  definita come il quadrato della distanza della posizione  $x, y$ , dalla circonferenza centrata in un generico  $X_c$  di raggio  $r$ . Utilizzando `surf` e `mesh` tracciare l'andamento grafico di  $f(x, y)$ .



## Utilizzo File esterni

- Apertura di un file

```
fid = fopen(nomefile, permessi);
```

Permessi:

'r'	Aprire il file in sola lettura.
'w'	Aprire il file, o ne crea uno nuovo, in scrittura; se il file esiste ne cancella il contenuto.
'a'	Aprire il file, o ne crea uno nuovo, in scrittura; appende i dati inseriti alla fine del file.
'r+'	Aprire il file in lettura e scrittura.
'w+'	Aprire il file, o ne crea uno nuovo, in lettura e scrittura; se il file esiste ne cancella il contenuto.
'a+'	Aprire il file, o ne crea uno nuovo, in lettura e scrittura; appende i dati inseriti alla fine del file.

- Chiusura di un file

```
fclose(fid);
```

- Navigazione nel file

Per spostarsi all'interno di un file

```
fseek(fid, offset, origine);
```

`origine` può assumere i valori:

'bof' : Inizio del file

'cof' : Posizione corrente nel file

'eof' : Fine del file

Per conoscere la posizione attuale all'interno di un file

```
posizione = ftell(fid);
```

- Lettura di un file

Per leggere dati da un file

```
A = fread(fid, lunghezza, precisione);
```

Per leggere dati formattati da un file

```
A = fscanf(fid, formato);
```

Per leggere una riga intera

```
line = fgets(fid);
```

- Scrittura di un file

Per scrivere in un file

```
fwrite(fid, A, precisione);
```

Per scrivere dati formattati in un file

```
fprintf(fid, formato, A);
```

## Esempio:

Letture di dati dal file binario 'dati.dat':

```
fid = fopen('dati.dat', 'r');  
fseek(fid, 20, 'bof');  
A = fread(fid, 10, 'uint32');  
fclose(fid);
```

Scrittura di dati alla fine del file binario 'dati.dat':

```
fid = fopen('dati.dat', 'r+');  
fseek(fid, 0, 'eof');  
fwrite(fid, A, 'uint8');  
fclose(fid);
```

Scrittura di un file di testo con dati formattati:

```
x = 0:.1:1;  
y = [x; exp(x)];  
fid = fopen('exp.txt','w');  
fprintf(fid,'%6.2f %12.8f\n',y);  
fclose(fid);
```

Viene creato il file di testo 'exp.txt' contenente una tabella di valori della funzione esponenziale:

```
0.00      1.000000000  
0.10      1.10517092  
...  
1.00      2.71828183
```

Letture di un file di testo con dati formattati:

```
fid = fopen('exp.txt');  
a = fscanf(fid, '%f %f', [2 inf]);  
a = a';  
fclose(fid);
```

## Esercizio 8

Creare uno script che legge i dati delle temperature medie giornaliere dei sette giorni della settimana di dieci città salvati in 7 file di testo formattati nel seguente modo:

```
Bari          32
Bologna       28
...
Venezia       27
```

Salvare i dati in una matrice, calcolare la media settimanale x ogni città e salvare tutte le medie settimanali nel file di testo 'medie.txt' con la stessa formattazione dei file giornalieri.

Graficare l'andamento delle temperature settimanali di 4 città a scelta in una figura divisa in 4 grafici.