



Laboratorio di Matlab

Rudy Manganelli

Dipartimento di Ingegneria dell'Informazione,
Università di Siena

manganelli@dii.unisi.it

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

Febbraio-Marzo 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);
```

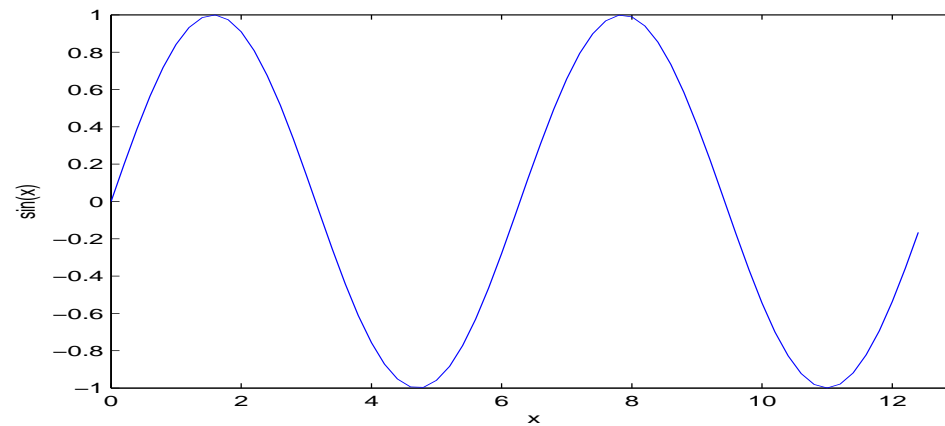
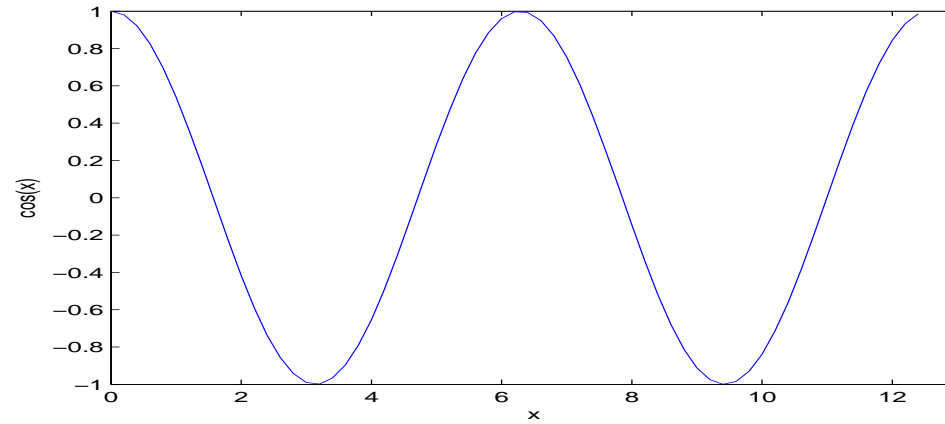
```
>> s = sin(x);
```

```
>> figure;
```

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

```
>> figure;
```

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

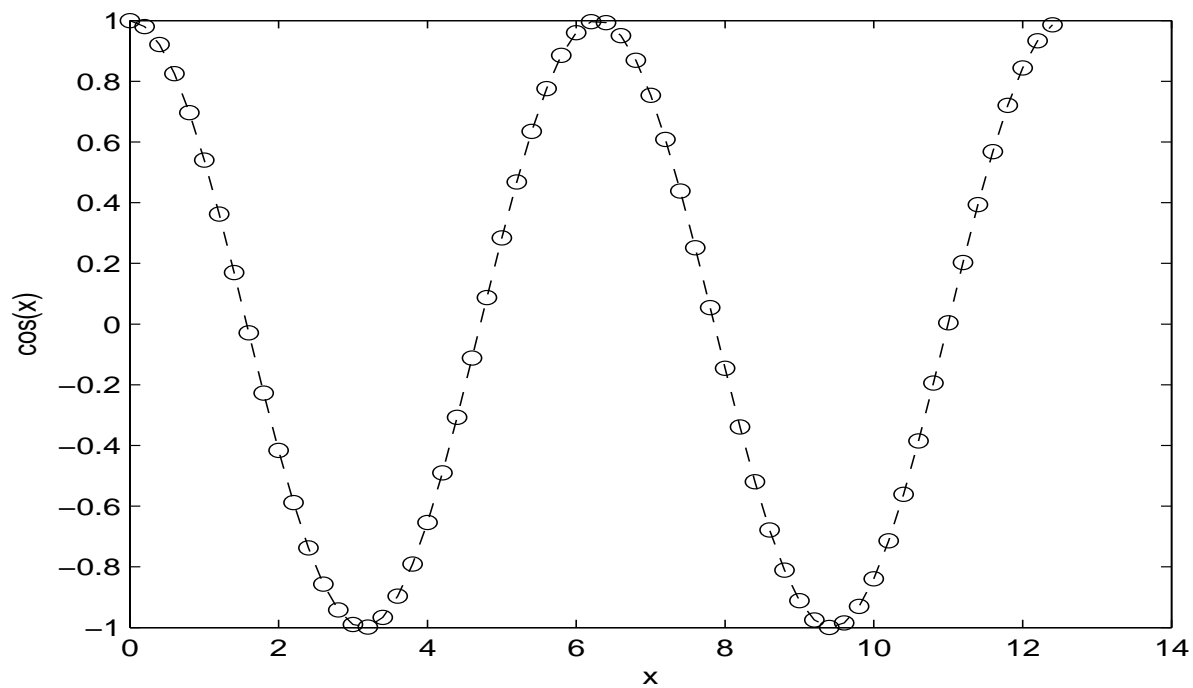


Funzioni correlate

- `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

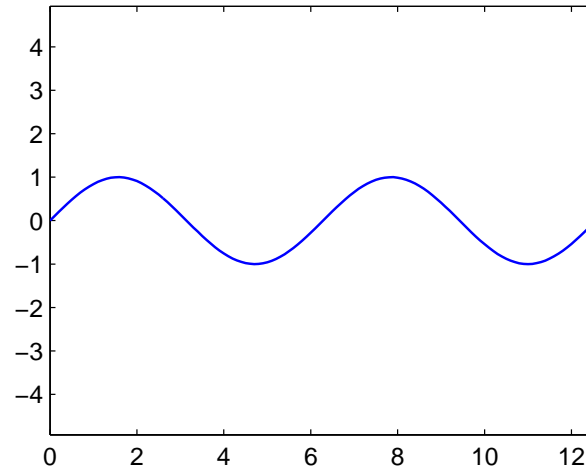
Per saperne di più ...

`help plot`

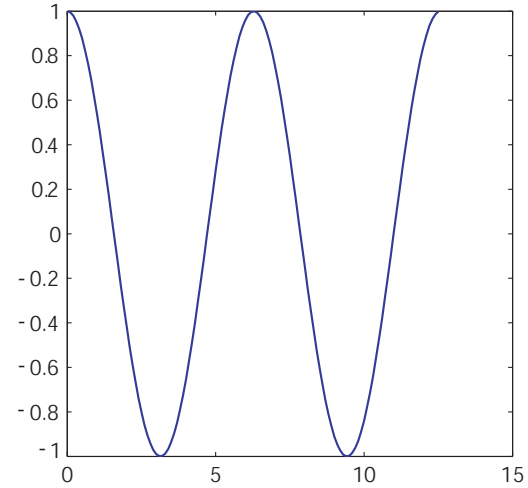
- `hold on` abilita la sovrapposizione di più grafici all'interno della stessa figura
- `hold off` disabilita la sovrapposizione di più grafici all'interno della stessa figura (una nuova istanza di plot elimina tutto quello che era stato tracciato precedentemente sulla figura di riferimento).
- `legend('S1', '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

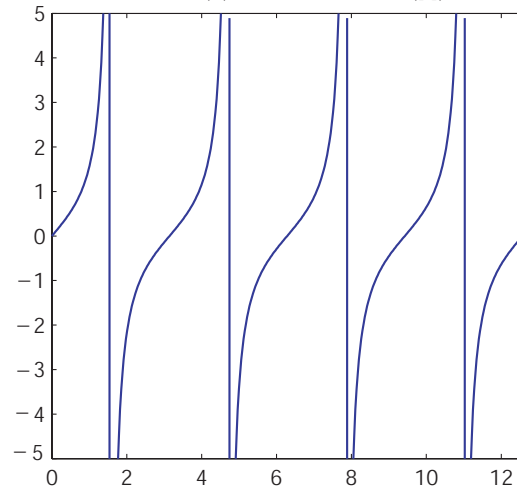
sin(x) con axis equal



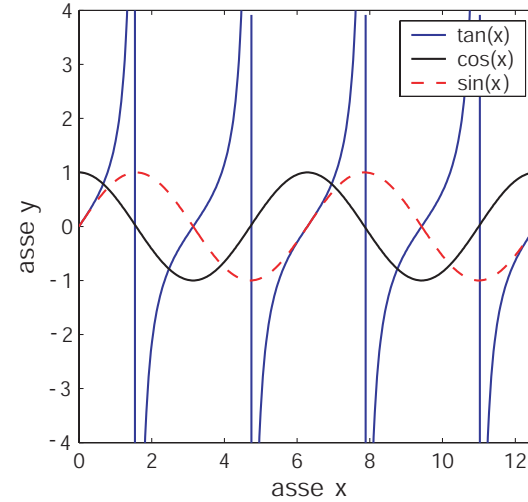
cos(x) con axis square

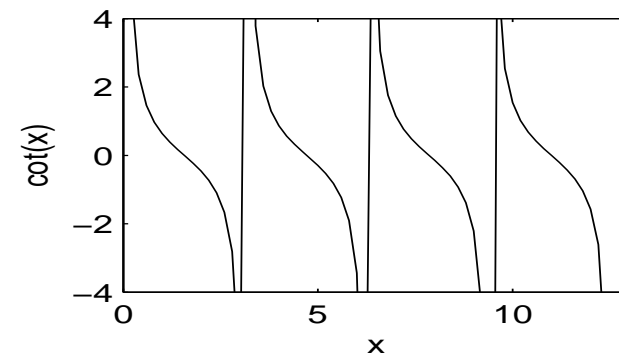
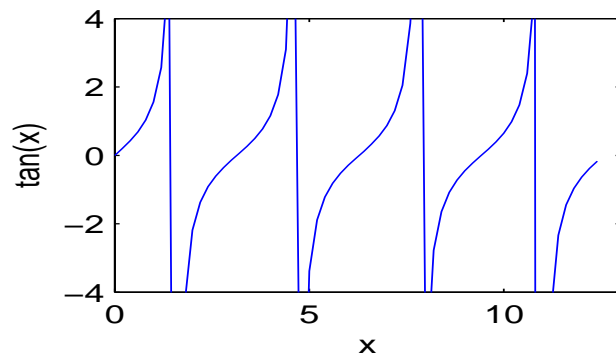
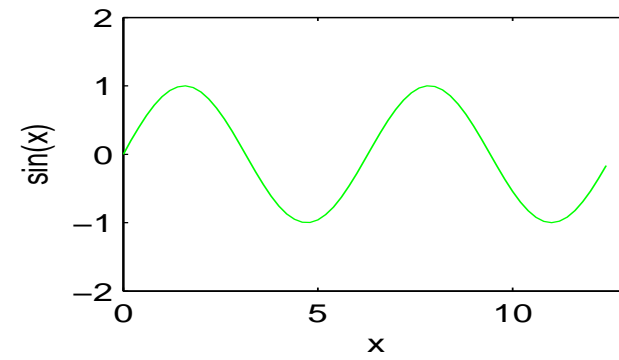
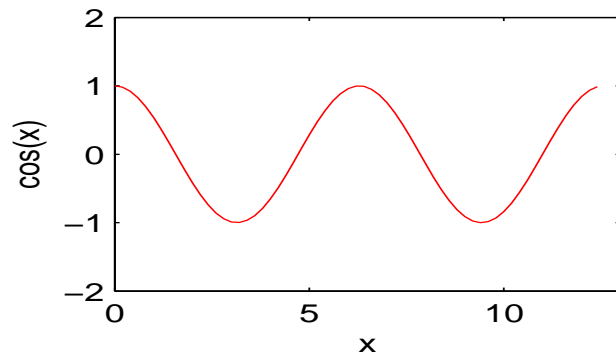


tan(x) con uso di axis([])



uso di legend e xlabel e ylabel



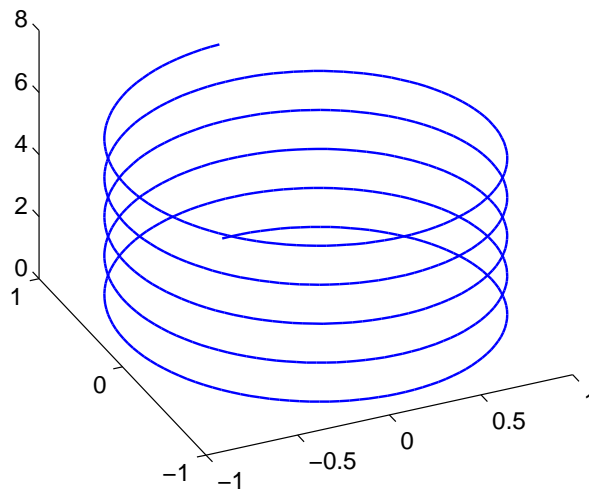


Traiettorie in \mathbb{R}^3

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}

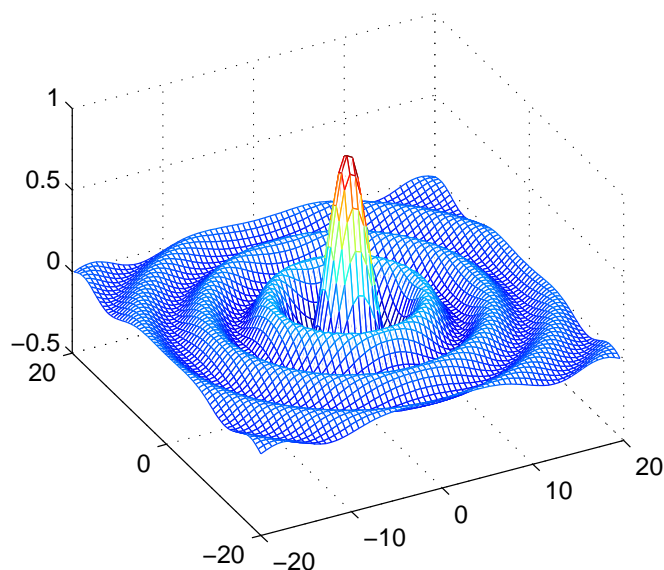
Esempio:



```
>> 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 :

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

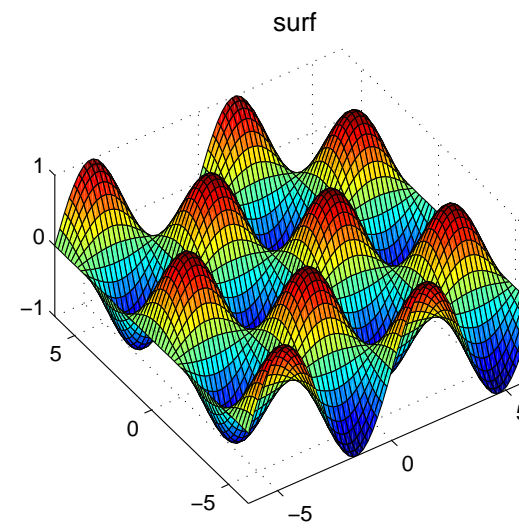
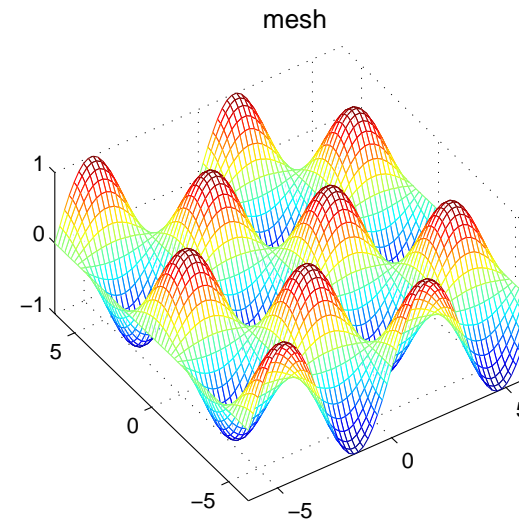
(senza nessun problema di dimensioni: prodotto cartesiano di x ed y)



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 $f(x, y)$.



Istruzioni di controllo di flusso

for

while

if

case

for

sintassi:

...

for *variabile = espressione*

istruzioni

end

...

Esempio 1:

```
n = 10;  
x = [];  
for i = 1:n  
    x = [x, i^2];  
end
```

Esempio 2:

```
passo = 0.1;  
t = 0.1:passo:100;  
f = log(t);  
plot(t, f, '-m');  
for i = 2:length(t)  
    dfdt(i-1) = (f(i)-f(i-1))/passo;  
end  
hold on  
plot(t(1:length(t)-1), dfdt, ':b');
```

while

sintassi:

```
...
```

```
while relazione
```

```
istruzioni
```

```
end
```

```
...
```

Esempio 3:

```
n = 0;  
a = 6;  
while n < exp(a)  
    n = n+1;  
end
```

Esempio 4:

```
n = 6;  
A = magic(n);  
B = zeros(n);  
C = [];  
k = 6;  
while k > 0  
    B(6-k+1,:) = A(k,:);  
    C = [C A(:,k)];  
    k = k-1;  
end
```

if

sintassi:

```
...  
if relazione 1  
istruzioni 1  
elseif relazione 2  
istruzioni 2  
else  
istruzioni 3  
end  
...
```

Esempio 5:

```
k = input('Inserisci un numero maggiore di cento :');
str = ischar(k);
if k > 100 & ~str
    disp('bravo!');
    delta = 100-k
elseif k > 50 & ~str
    disp('ci sei quasi!')
    delta = 100-k
elseif k > 0 & ~str
    disp('mica tanto vicino!');
    delta = 100-k
elseif str
    disp('ho detto NUMEROOOOOO');
else
    disp('allora sei duro!');
    delta = 100-k
end
```



case

sintassi:

...

switch *espressione di switch*

case *case valore 1*

istruzioni

case { *case valore 2, case valore 3, case valore 4,...* }

istruzioni

otherwise

istruzioni

end

...

Esempio 6:

```
k = input('Inserisci un numero da 1 a 3:');  
switch k  
case 1  
    disp('bravo!');  
case {2,3}  
    disp('esagerato!')  
otherwise  
    disp('Ho detto da 1 a 3 !!!');  
end
```

Esercizio 5

1. Generare una matrice random 100×100 utilizzando il comando `rand`
2. Estrarne la diagonale principale utilizzando un doppio ciclo `for` ed assegnarla ad un vettore `v1`
3. Estrarne la contro diagonale utilizzando `while` ed assegnarla ad un vettore `v2`
4. Plottare i vettori `v1` e `v2` sulla stessa figura visualizzando la legenda, il titolo e la descrizione degli assi.