

Grafici in matlab

- Matlab ha un sacco di funzioni per le rappresentazioni grafiche!
- Oggi ne vediamo alcune.
- Per ulteriori informazioni:
`help graph2d`
`help graph3d`

1

Un esempio di grafico:

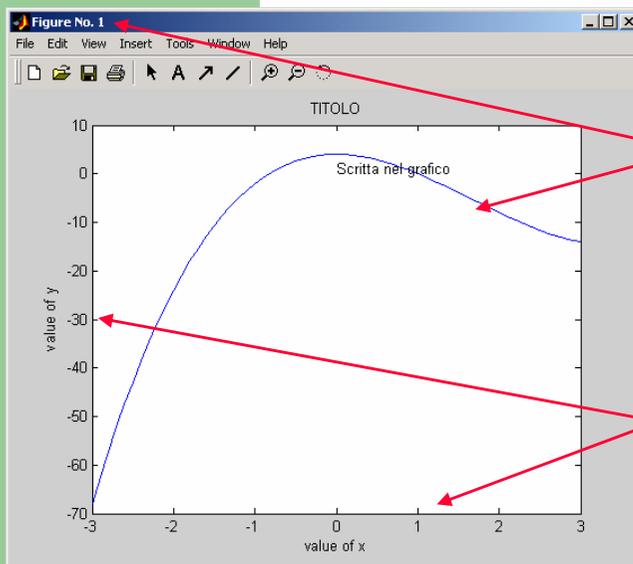
Notazione con l'operatore ":" e operazioni elemento per elemento

```
xv = -3:0.1:3;
yv = xv.^3 - 5*xv.^2 + 4;
plot(xv,yv)
xlabel('value of x')
ylabel('value of y')
title('TITOLO')
text(0,1, 'Scritta nel grafico')
```

← Uso standard del comando plot

← Per annotare

2

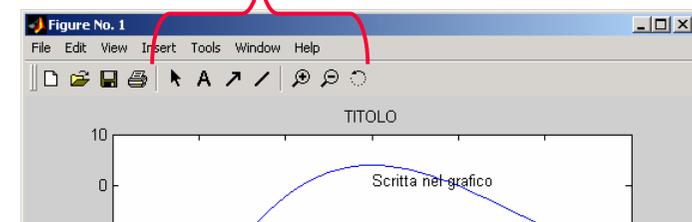


Il default è "figure1",
 linea blu e sottile

Gli assi sono scalati in automatico

3

- Matlab mette a disposizione dei comandi interattivi per modificare il grafico

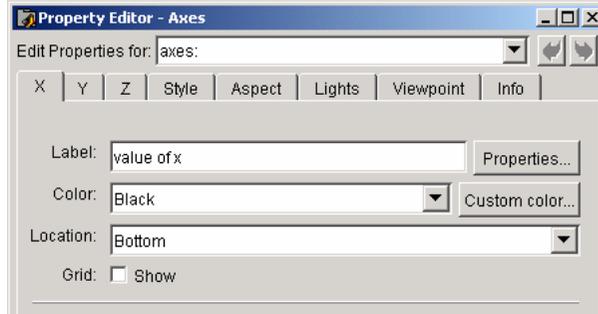


4

Click sulla freccia per modificare

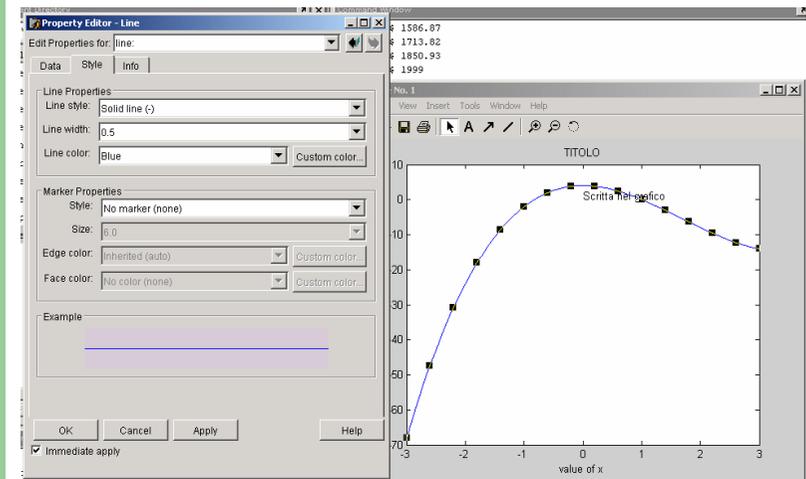


Click destro -> properties su una parte della figura per modificarla



5

O anche le linee:



6

- In pratica, il **property editor** è simile agli strumenti di excel però...
 - È limitato ad una singola figura
 - Ripeterlo per ogni nuovo grafico è noioso
- La funzione plot mette a disposizione degli argomenti addizionali per modificare da riga di comando:


```
plot(x,y, 'linespec' , 'Propname' , PropValue )
```

 - Specifiche di linea: colore, tipo di linea, punti dei dati
 - Proprietà e valore: spessore, dimensione ecc.

7

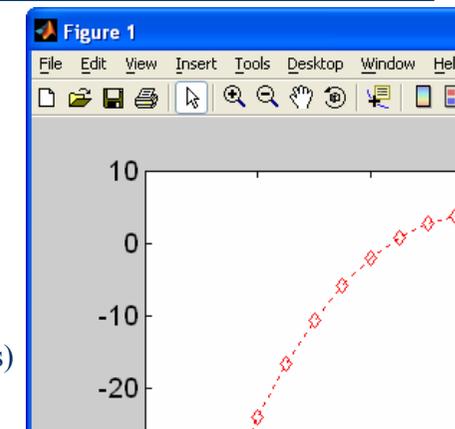
Specifiche di Linea

`plot(x,y, 'r : d ')`

rosso

punteggiata

Rombi
(diamonds)



8

Forma Generale: `plot(x,y, 'OBJ' 'OBJ' 'OBJ')`

Colore:

k black
r red
b blue
g green
y yellow
c cyan
w white
m magenta

Simbolo:

. point
o circle
x x-mark
s square
d diamond
Etc (+ , * ,
^ , > , <
, v , p , h) .

Tipo Linea:

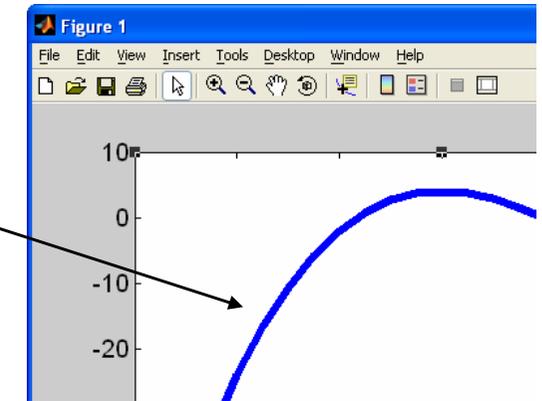
- solid
: dotted
-- dashed
-. dash-dot

L'ordine non è importante!

Proprietà e valori

`plot(x,y, 'linewidth',5)`

Lo spessore della linea è 5 punti



Forma Generale: `plot(x,y, 'OBJ',value)`

Proprietà:

linewidth
markersize
markeredgecolor
markerfacecolor

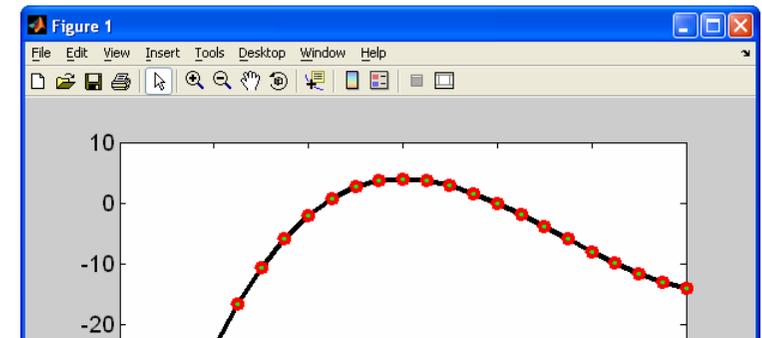
Valore: dipende dalla proprietà

Dimensioni in punti
Colori

Possono esserci coppie multiple !

• Esempio:

`plot(x,y, '- k o' , 'LineWidth' , 3 , 'MarkerSize', 6,...
'MarkerEdgeColor','red','MarkerFaceColor','green')`

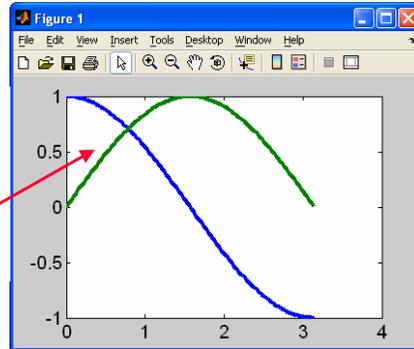


Più grafici negli stessi assi

- Possiamo disegnare diversi grafici sugli stessi assi:

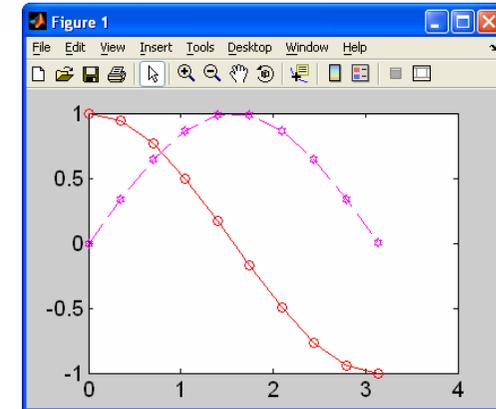
```
x = linspace(0,pi);
y1 = cos(x);
y2 = sin(x);
plot(x,y1,x,y2)
```

I colori cambiano...



13

```
x = linspace(0,pi,10);
y1 = cos(x);
y2 = sin(x);
plot(x,y1,'ro-',x,y2,'mh--')
```

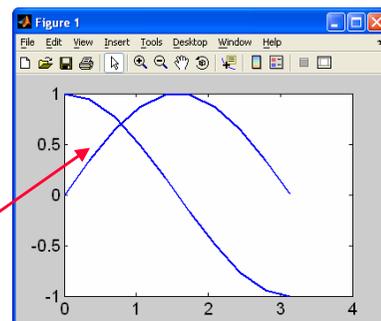


14

- Possiamo usare **hold** per “congelare” il grafico

```
x = linspace(0,pi,10);
y1 = cos(x);
y2 = sin(x);
plot(x,y1,'linewidth',2)
hold on
plot(x,y2,'linewidth',2)
hold off
```

Notare i colori adesso!



15

Attenzione...

- Se passiamo delle matrici:
 - `plot(x_matrice, y_matrice)`
 - Grafico colonna per colonna
 - Cambia i colori ogni volta
- Per un argomento singolo, `plot(x)`:
 - Reale/immaginario se x è complesso
 - x /indice se x è reale

16

Altri due comandi utili

- **figure**
 - Da solo apre una nuova finestra
 - **figure(n)** ci porta alla finestra n-esima
- **ginput(1)**
 - Crea un mirino sulla figura
 - restituisce la posizione (x,y) al click del mouse
 - **ginput(n)** restituisce n coppie di coordinate

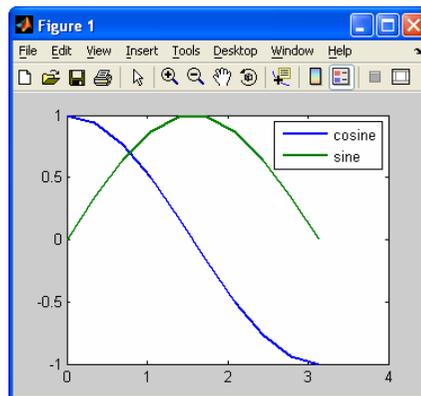
17

Aggiungere del testo

- Conosciamo già:
 - **xlabel('string')**
 - **ylabel('string')**
 - **title('string')**
 - **text(x, y, 'string')**
- In più:
 - **gtext('string')** – controllato dal puntatore
 - **legend('string1', ... 'stringn', loc)**

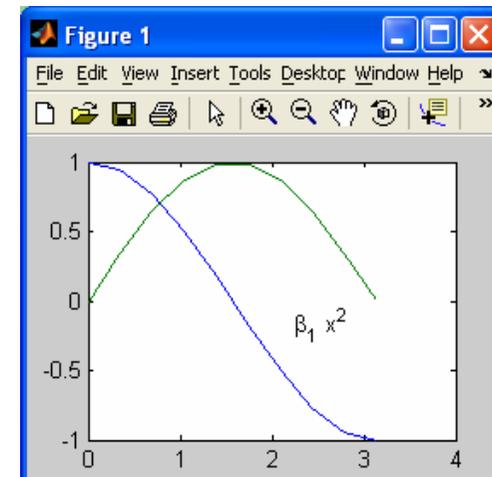
18

```
x = linspace(0,pi,10);
y1 = cos(x);
y2 = sin(x);
plot(x,y1,x,y2,'linewidth',2)
legend('cosine','sine')
```



19

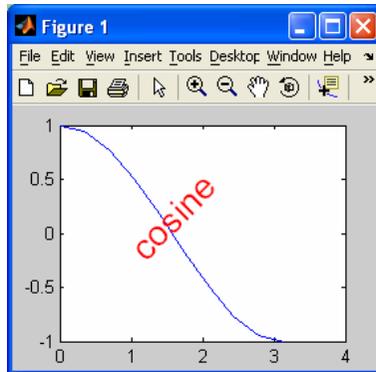
- Ci sono lettere greche, apici e pedici
- Es. **gtext('\beta_1 x^2')**



20

- E possiamo modificare l'aspetto del testo
- Es.

```
gtext('cosine', 'fontsize', 20, 'rotation', 45,
      'color', 'red', )
```



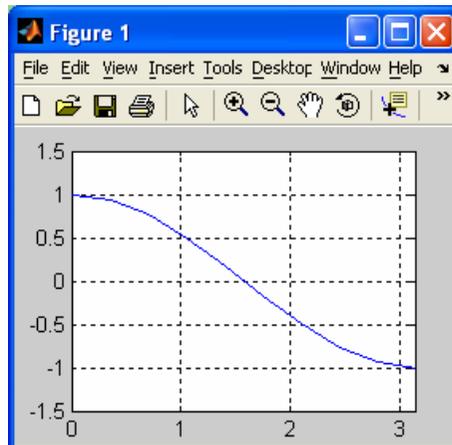
21

Gli assi

- Possiamo aggiungere una griglia
grid
- O settare i limiti degli assi:
axis([xmin xmax ymin ymax])

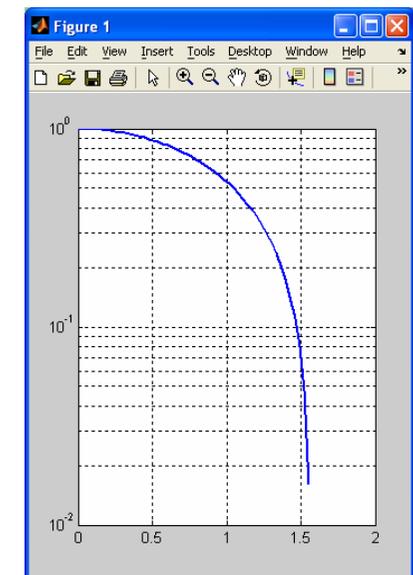
22

- Es.
- ```
plot(x,y)
grid on
axis([0 pi -1.5 1.5])
```



23

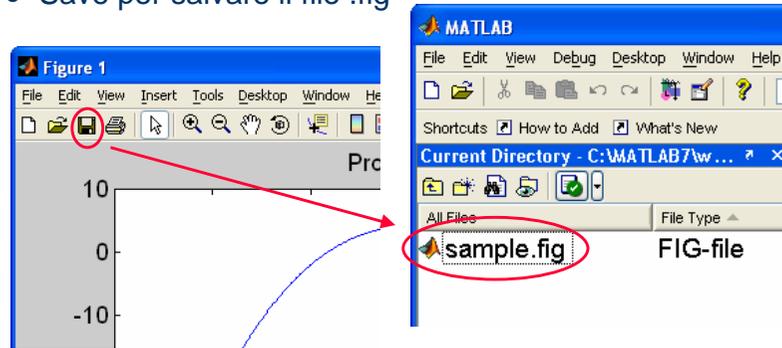
- Possiamo fare dei grafici logaritmici:  
**semilogx(x,y)**  
**semilogy(x,y)**  
**loglog(x,y)** →
- I dati negativi vengono ignorati



24

## Figure Files

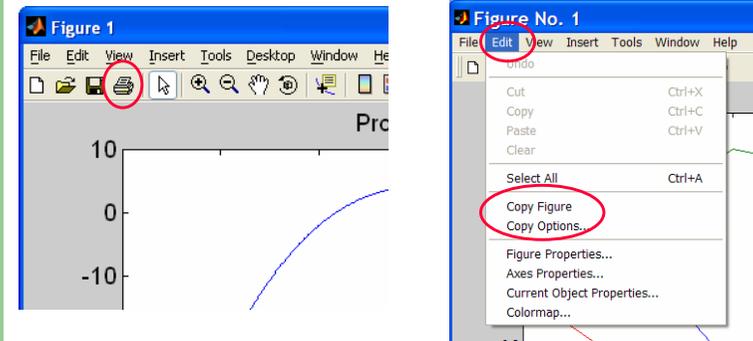
- Save per salvare il file .fig



25

## Per esportare le figure

- Stampare:
- Copia (per incollare)

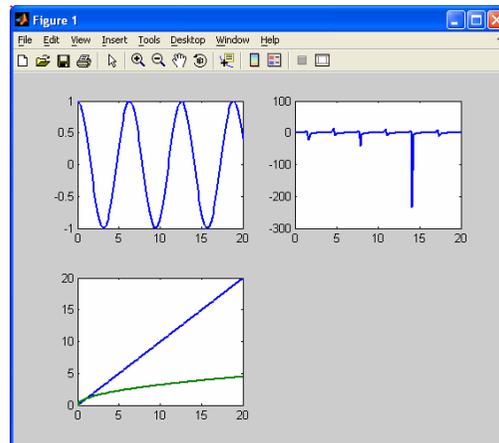


26

## Subplot: una figura, più assi

subplot(2,2,1)  
plot(x1,y1)  
subplot(2,2,2)  
etc.

I parametri sono:  
numero di righe,  
numero di colonne,  
settore scelto



27

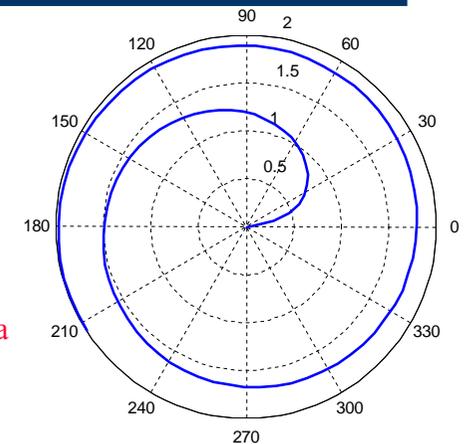
## Altri tipi di grafici 2-D - Polari

$x = 1:100;$   
 $r = \log_{10}(x);$   
 $t = x/10;$

polar( t, r )

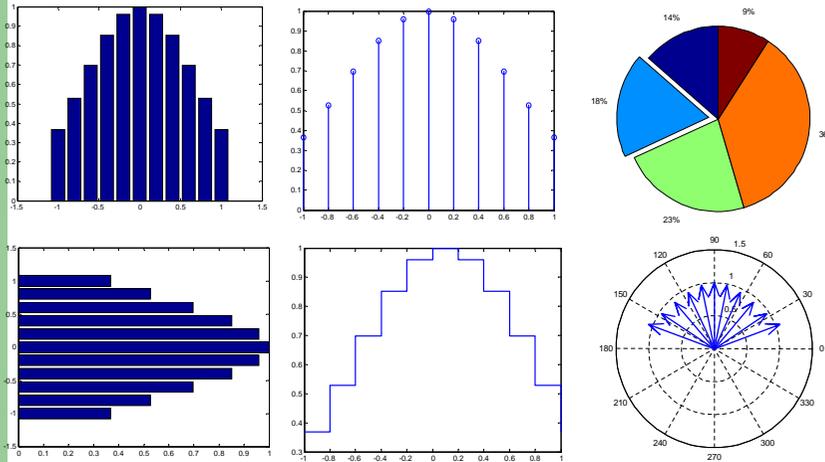
Angolo in radianti

Ampiezza



28

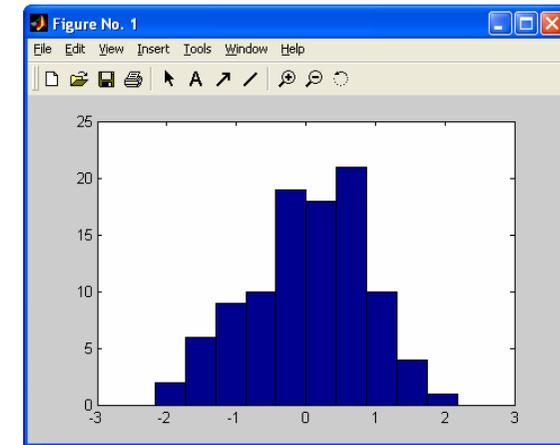
- Vertical and horizontal **bar** plots, **stem** and **stair** plots, **pie** and **compass** plots:



29

- Istogrammi:

```
x = randn(1,100);
hist(x)
```

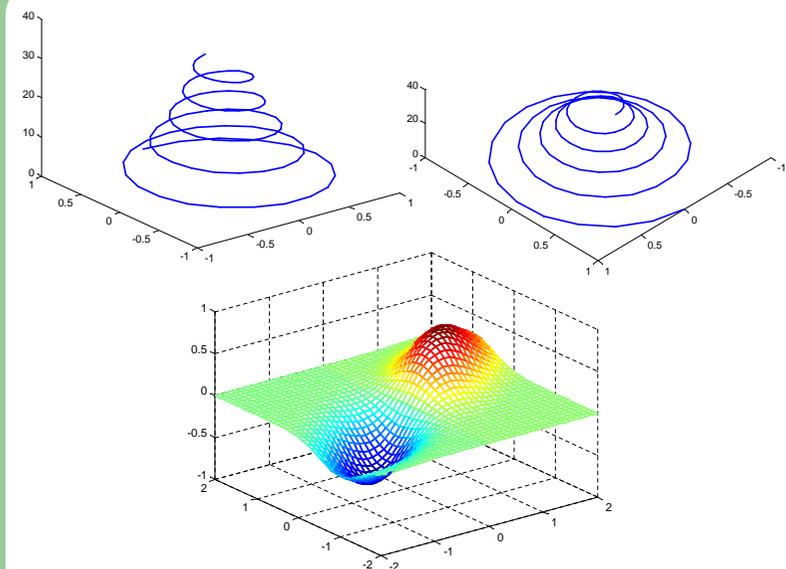


30

## Inoltre...

- Specifiche di linea, proprietà, assi, griglie... funzionano con gran parte dei tipi di grafico
- Molti di questi strumenti hanno funzioni extra... usate l' **help!!!**
- Possiamo anche fare dei grafici 3-D!

31



32

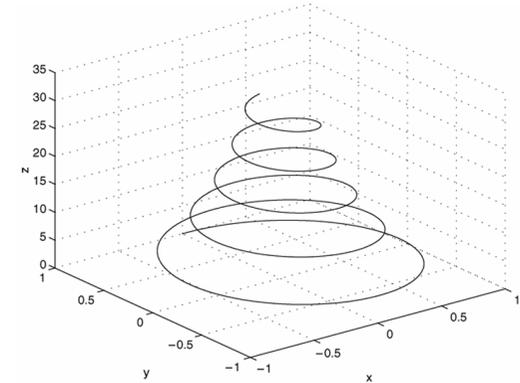
## Grafici 3-D

Per tracciare una linea ( $x,y,z=f(t)$ ) in un diagramma tridimensionale usiamo la `plot3`:

```
>>t = [0:pi/50:10*pi];
>>plot3(exp(-0.05*t).*sin(t),...
exp(-0.05*t).*cos(t),t),...
xlabel('x'),ylabel('y'),zlabel('z'),grid
```

33

La curva  $x = e^{-0.05t} \sin t$ ,  $y = e^{-0.05t} \cos t$ ,  $z = t$ , disegnata con `plot3`



34

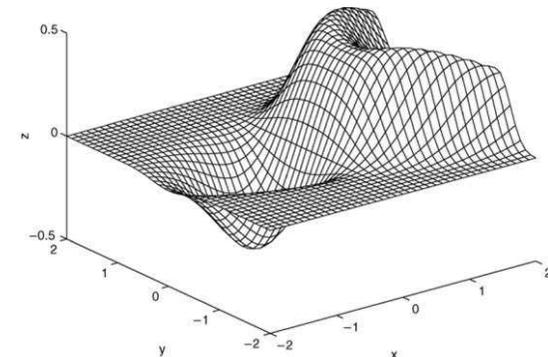
## Superfici

Funzione  $z = xe^{-(x-y^2)^2+y^2}$ ,  
per  $-2 \leq x \leq 2$  and  $-2 \leq y \leq 2$ , con intervallo 0.1.

```
>>[X,Y] = meshgrid(-2:0.1:2);
>>Z = X.*exp(-((X-Y.^2).^2+Y.^2));
>>mesh(X,Y,Z),xlabel('x'),ylabel('y'),...
zlabel('z')
```

35

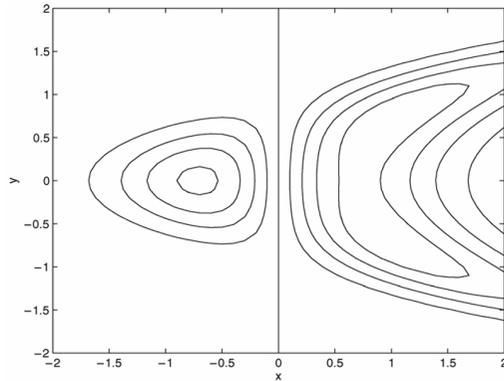
Funzione  $z = xe^{-(x-y^2)^2+y^2}$



36

## Disegno dei contorni

```
>>[X,Y] = meshgrid(-2:0.1:2);
>>Z = X.*exp(-((X- Y.^2).^2+Y.^2));
>>contour(X,Y,Z),xlabel('x'),ylabel('y')
```



37

## Funzione

```
contour(x,y,z)
```

```
mesh(x,y,z)
```

```
meshc(x,y,z)
```

```
meshz(x,y,z)
```

```
surf(x,y,z)
```

```
surfc(x,y,z)
```

```
[X,Y] = meshgrid(x,y) Crea le matrici x e y dai vettori x e y per definire la griglia
```

```
[X,Y] = meshgrid(x) Come [X,Y]= meshgrid(x,x).
```

```
waterfall(x,y,z) Come mesh ma disegna linee solo in una direzione.
```

## Descrizione

Contorni.

Superficie 3D (griglia).

Come mesh ma con i contorni sotto.

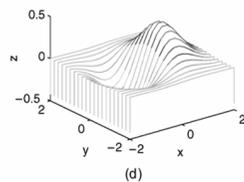
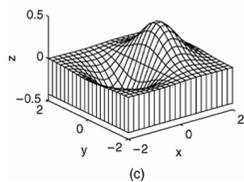
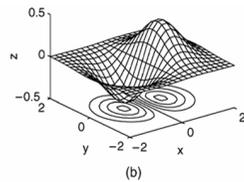
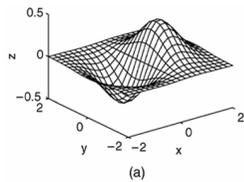
Come mesh ma disegna linee verticali ai bordi

Superficie sfumata 3D.

Come surf ma con i contorni.

38

4 varianti per  $z = xe^{-(x^2+y^2)}$ : meshc, meshz, and waterfall.  
a) mesh, b) meshc, c) meshz, d) waterfall



39

## ● Ancora...

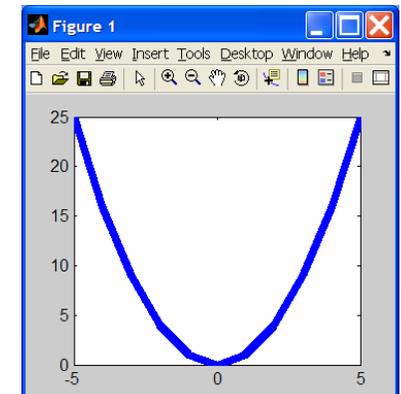
– Manipolare i grafici (set) ...

```
>> x = -5:5;
>> y = x.^2;
>> h = plot(x,y)

h =

 157.0101

>> set(h, 'linewidth', 5)
>> |
```



40

## - GUIs – interfacce di alto livello

