Argomenti

(1): Introduzione a Matlab
(2): *Scripts* – Funzioni – Operatori
(3): *Input-Output* dei dati
(4): Grafici e visualizzazione dei dati
(5): Analisi dei dati
(6): Analisi nonlineare ed approssimazioni di funzioni e di dati
(7): Differenziazione ed integrazione numerica – calcolo simbolico
(8): Sistemi lineari
(9): Soluzioni *ODEs*
(10): Cenni a soluzioni *PDEs*

Grafici e Visualizzazioni dei dati

- Plots
  - Plot semplici - Subplots
  - Mesh plots - Surface Plots - Contour Plots
- Animazioni

Grafici 2-D (>>help graph2d)

X, Y vettori di dimensione *n*;

`plot(X,Y)` apre una finestra grafica dove traccia una spezzata che unisce tutte le coppie

(X1,Y1), (X2,Y2),…., (Xn,Yn)

riscala automaticamente gli assi

L'istruzione `plot` usa la finestra grafica aperta o ne apre una e la rende corrente

`figure` apre una nuova finestra senza cancellare le precedenti

(ricordare: `clf, close, close all, figure(n)`)

`hold on` è l'istruzione per tracciare più figure nella stessa finestra

(`hold off` di default)

Gli argomenti della funzione `plot` possono anche essere matrici

Grafici 2-D: plot (>>help plot)

Sintassi del comando plot:

`plot(vettore1, vettore2, opzioni)`

`opzioni` contiene al più 3 elementi, per i quali l'ordine non importa

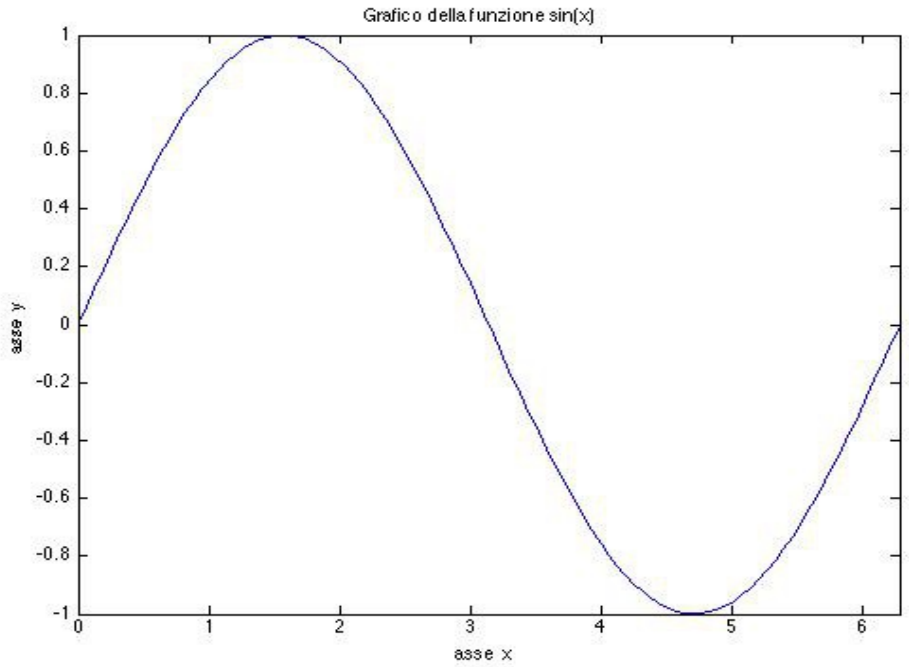
'y:*' ':y*' etc sono equivalenti

Colore	Linea	Simbolo
y giallo	- Linea continua	. Punto
m magenta	: linea punteggiata	o circoletto
c ciano	-. Linea punto	x per
r rosso	-- linea tratteggiata	+ più
g verde		* asterisco
b blu		s quadratino
w bianco		d diamante
k nero		v triangolo

Grafici 2-D: `plot`

Esempio:

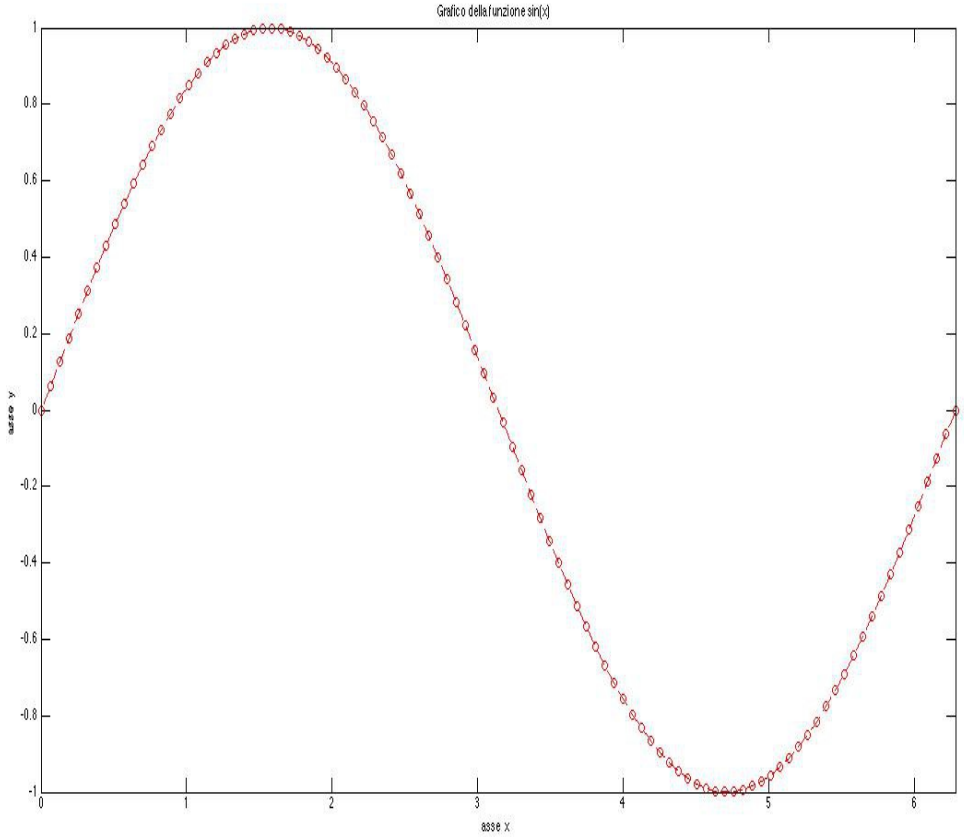
```
x=linspace(0,2*pi);
y=sin(x);
plot(x,y);
title('Grafico della
funzione sin(x)')
xlabel('asse  x')
ylabel('asse  y')
axis([0 2*pi -1 1])
```



Grafici 2-D: `plot`

Esempio:

```
>>
x=linspace(0,2*pi);
y=sin(x);
plot(x,y,'ro--');
title('Grafico della
funzione sin(x)')
xlabel('asse  x')
ylabel('asse  y')
axis([0 2*pi -1 1])
```



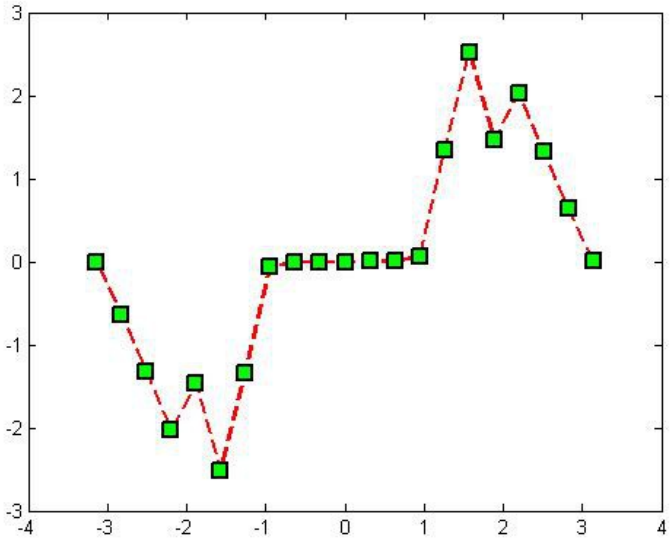
Grafici 2-D: `plot`

Altri argomenti della funzione `plot`:

```
LineWidth (0.5 valore di default)
FontSize (10      “                )
FontAngle ( normal   “             )
MarkerSize (6           “          )
   MarkerEdgeColor (auto   “   )
   MarkerFaceColor (none   “    )
```

Grafici 2-D: `plot`

Esempio:

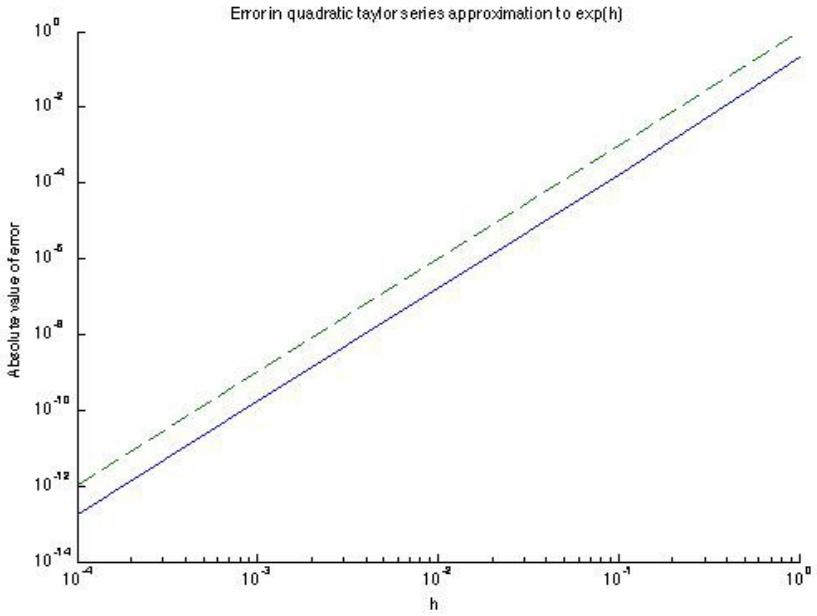



```
x = -pi:pi/10:pi;
y = tan(sin(x)) - sin(tan(x));
plot(x,y,'--rs','LineWidth',2,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',10)
```

Grafici 2-D: loglog

Esempio:

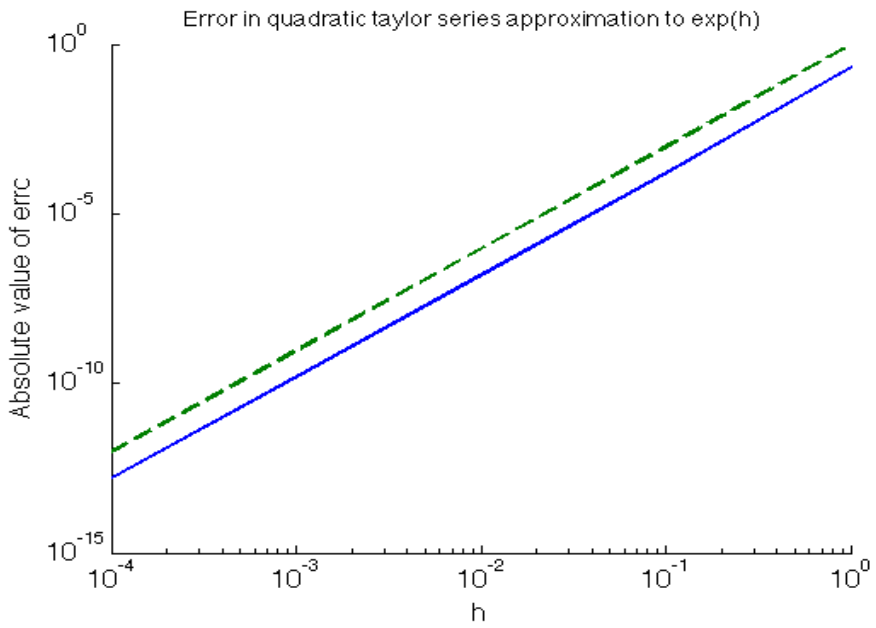
```
%un esempio con loglog per evidenziare una legge di potenza
 h=10.^[0:-1:-4];
taylorerr=abs((1+h+h.^2/2)-exp(h));
loglog(h,taylorerr,'-', h,h.^3,'--')
xlabel('h')
ylabel('Absolute value of error')
title('Error in quadratic taylor series approximation to
exp(h)')
box off
```



Grafici 2-D: loglog

Esempio:

```
%un esempio con loglog per evidenziare una legge di potenza
h=10.^[0:-1:-4];
taylorerr=abs((1+h+h.^2/2)-exp(h));
axes('FontSize',16)
loglog(h,taylorerr,'-', h,h.^3,'--','LineWidth',2)

xlabel('h','FontSize',16)
ylabel('Absolute value of error','FontSize',16)
title('Error in quadratic taylor series approximation to exp(h)','FontSize',14)
box off
```



Grafici 2-D: `plot`

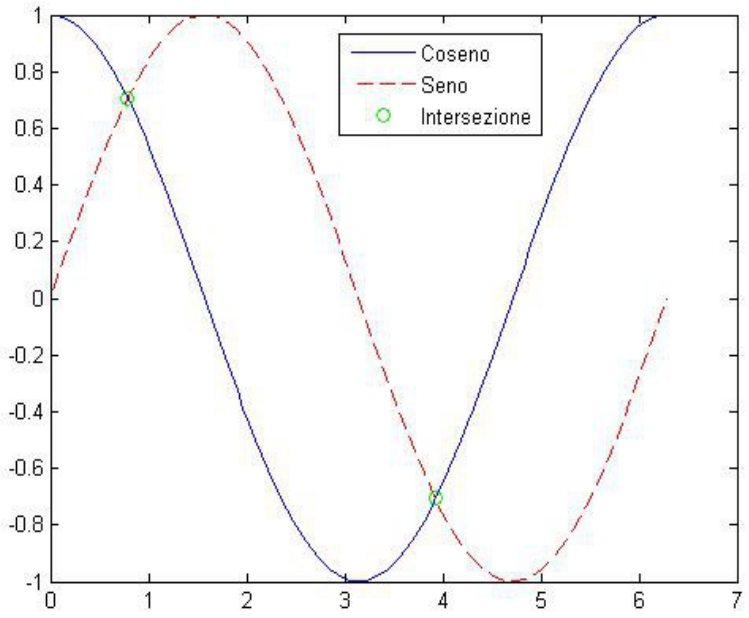
Funzioni correlate ai grafici assi e annotazioni:

```
axis([xmin xmax ymin ymax]) (>>help axis):
xlim([xmin xmax])
ylim([ymin ymax])
axis('equal'),
axis square
axis tight
axis auto (default)
grid (on - off)
title('stringa')
xlabel('stringa')
ylabel('stringa')
zlabel('stringa')
gtext('stringa')
text(x,y,'stringa') (>>doc text_props)
```

Grafici 2-D: `plot`

Esempio:

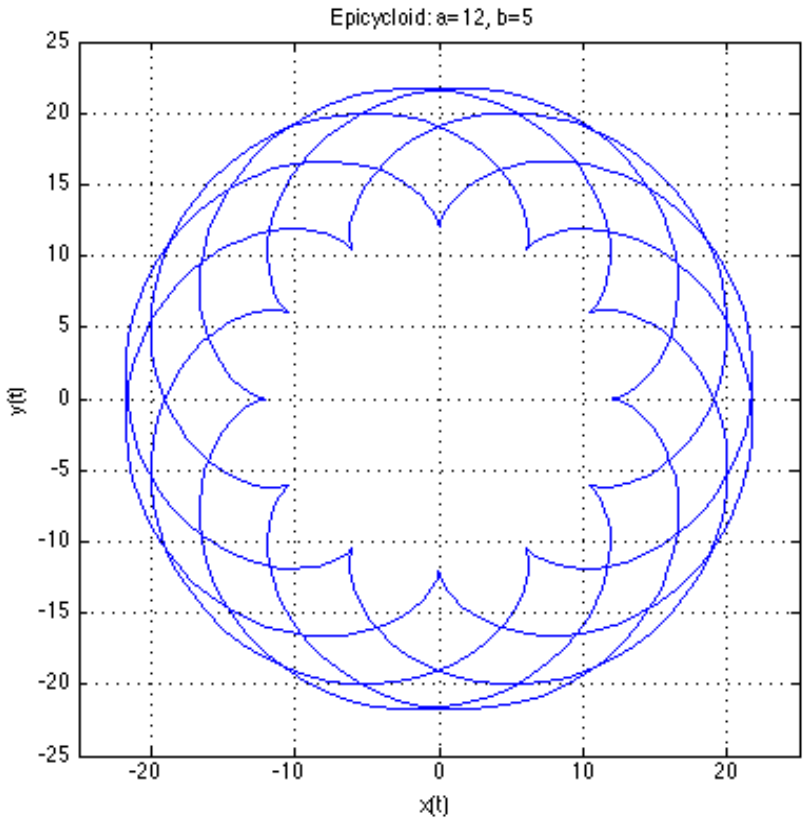
```
%Grafici sovrapposti:
x = linspace(0,2*pi);
y1 = cos(x);
y2 = sin(x);
plot(x,y1,'-')
hold on
plot(x,y2,'--r')
plot([1 5]*pi/4, [1 -1]/sqrt(2),'og')
legend('Coseno', 'Seno', 'Intersezione')
hold off
```



Grafici 2-D: `plot`

Esempio:

```
% Epicycloid

a=12; b=5;
t=0:0.05:10*pi;
x=(a+b)*cos(t)-b*cos((a/b+1)*t);
y=(a+b)*sin(t)-b*sin((a/b+1)*t);
plot(x,y)
axis equal
axis([-25 25 -25 25])
grid on
title('Epicycloid: a=12, b=5')
xlabel('x(t)'), ylabel('y(t)')
```



Grafici 2-D: `plot`

Sintassi del comando subplot:

```
subplot(righe, colonne, sottofinestre)
```

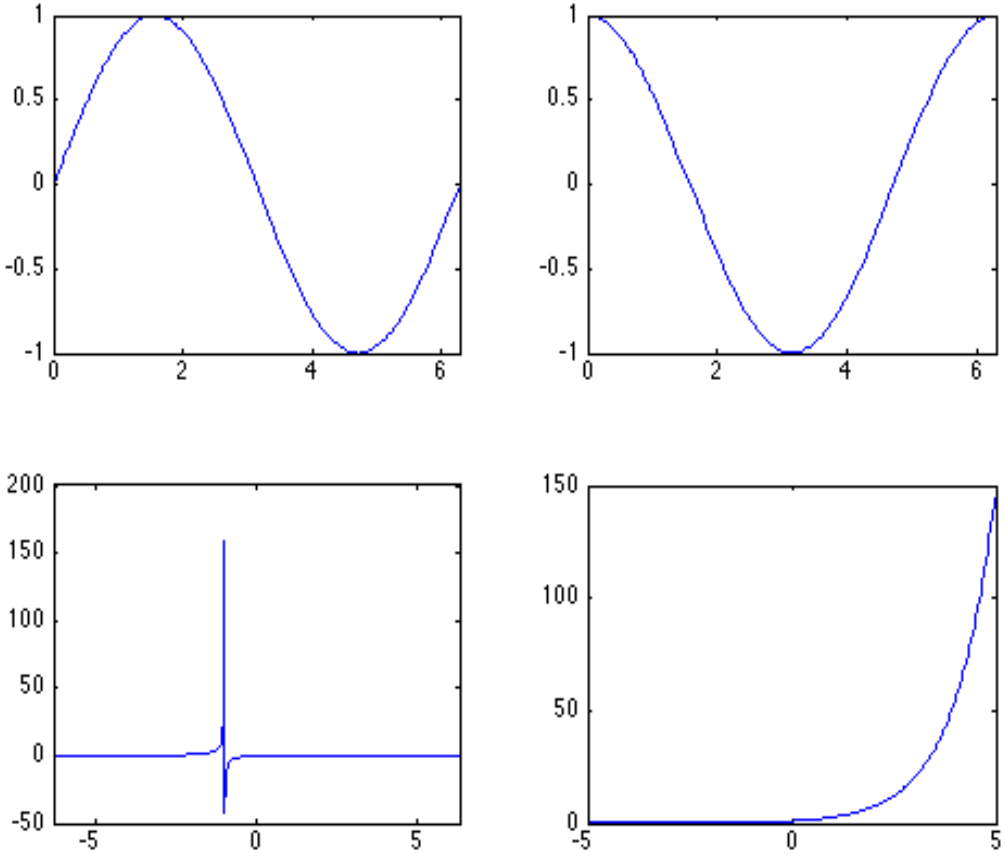
Esempio: `subplot(2,3,1)`

1	2	3
4	5	6

Grafici 2-D: subplot

Esempio:

```
subplot(2,2,1)
fplot('sin(x)',[0 2*pi])
subplot(2,2,2)
fplot('cos(x)',[0 2*pi])
subplot(2,2,3)
fplot('sin(x)/(1+x)',[-2*pi 2*pi])
subplot(2,2,4)
fplot('exp(x)',[-5 5])
```



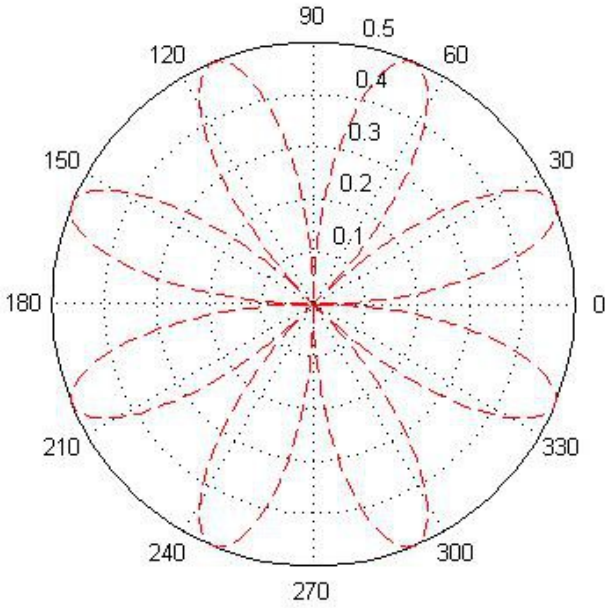
Grafici 2-D: lista di funzioni

```
loglog
semilogx
semilogy
plotyy
polar
fplot
ezplot
ezpolar
fill
```

```
area
bar
barh
hist
pie
errorbar
comet
quiver
scatter
stairs
```

Grafici 2-D: `polar(teta,r)`

Esempio:

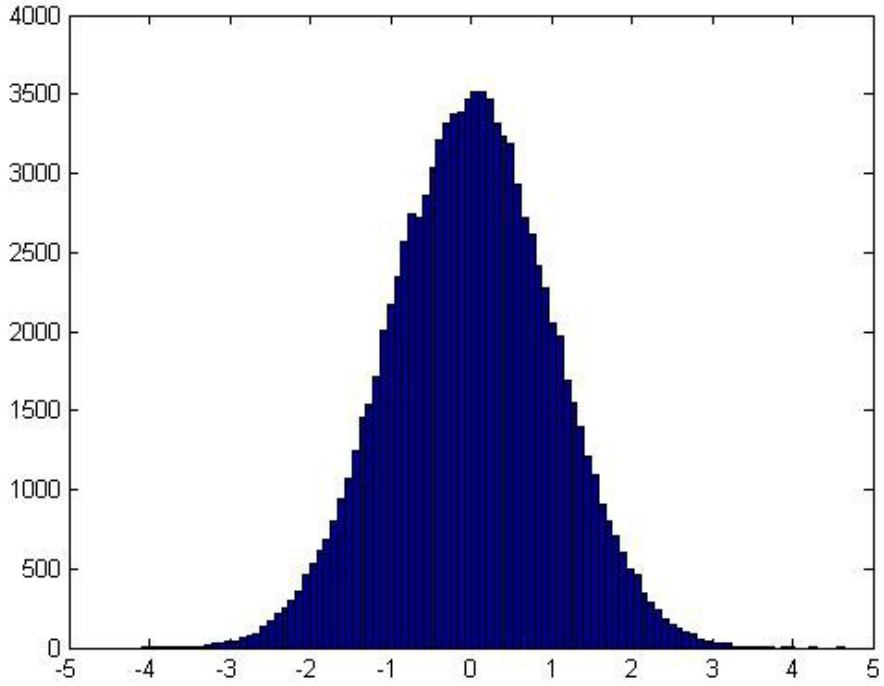



```
t = 0:.01:2*pi;
polar(t,sin(2*t).*cos(2*t),'--r')
```

Grafici 2-D: `hist(x,nbin)`

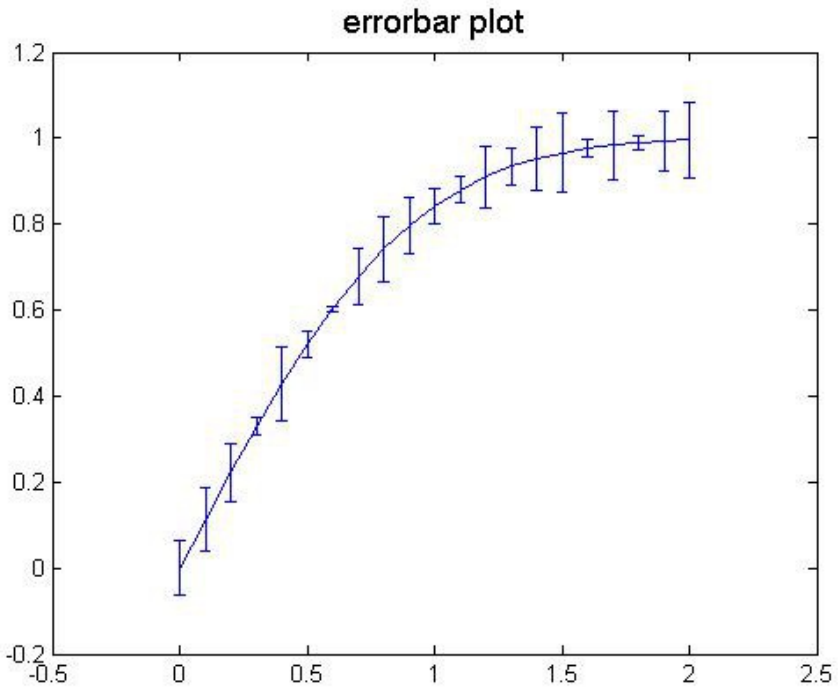
Esempio:

```
x=randn(1,100000);
>> hist(x,100)
```



Grafici 2-D: `errorbar(x,y,e)`

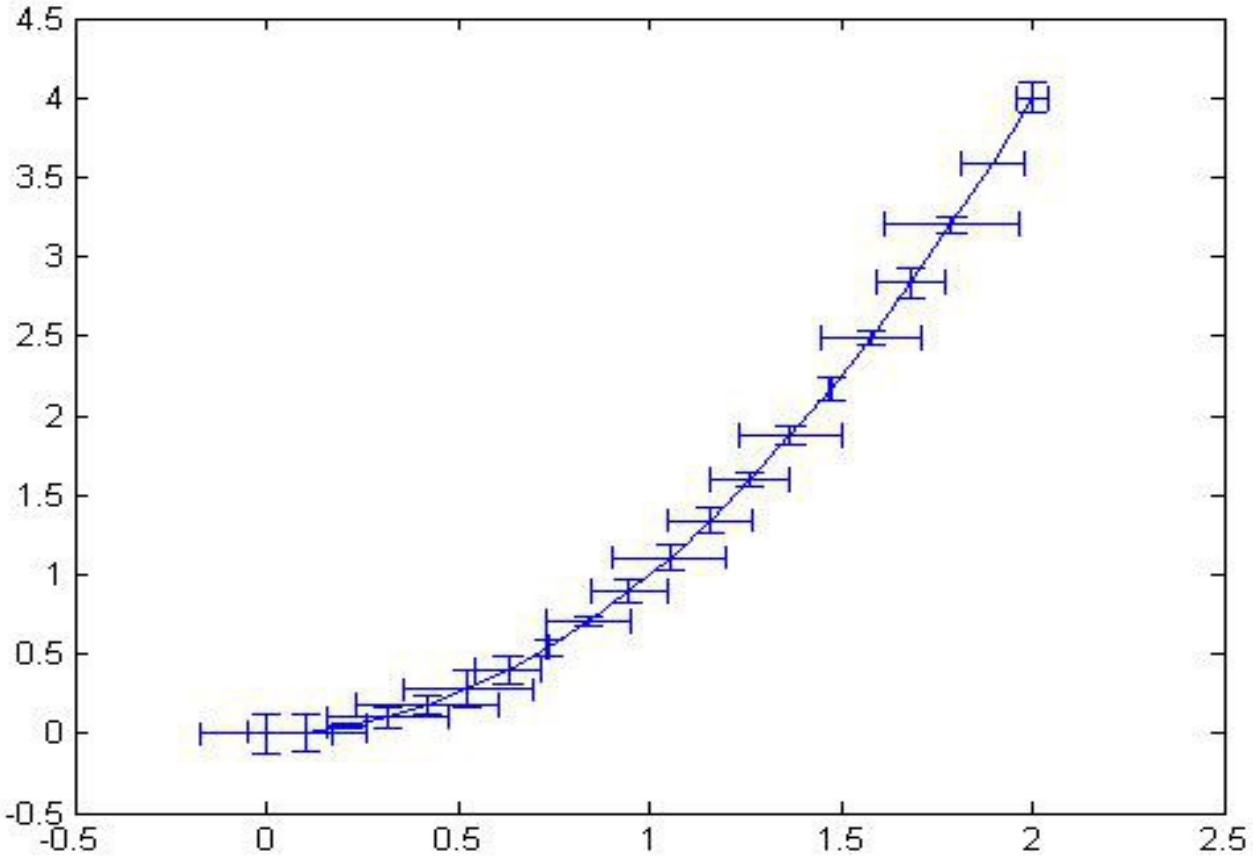
Esempio:




```
x=linspace(0,2,21);
y=erf(x);
e=rand(size(x))/10;
errorbar(x,y,e)
title('errorbar plot')
```

Grafici 2-D:
function ploterr(x,y,xerr,yerr)

Esempio:

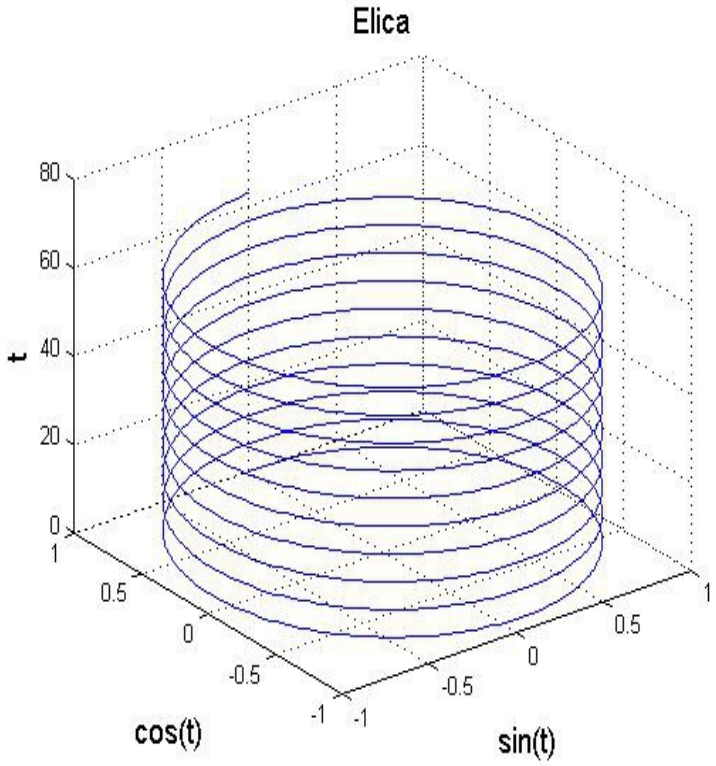


Grafici 3D: linea

```
plot3(x1,y1,z1,opzioni1,x2,y2,z2,opzioni2,...)
```

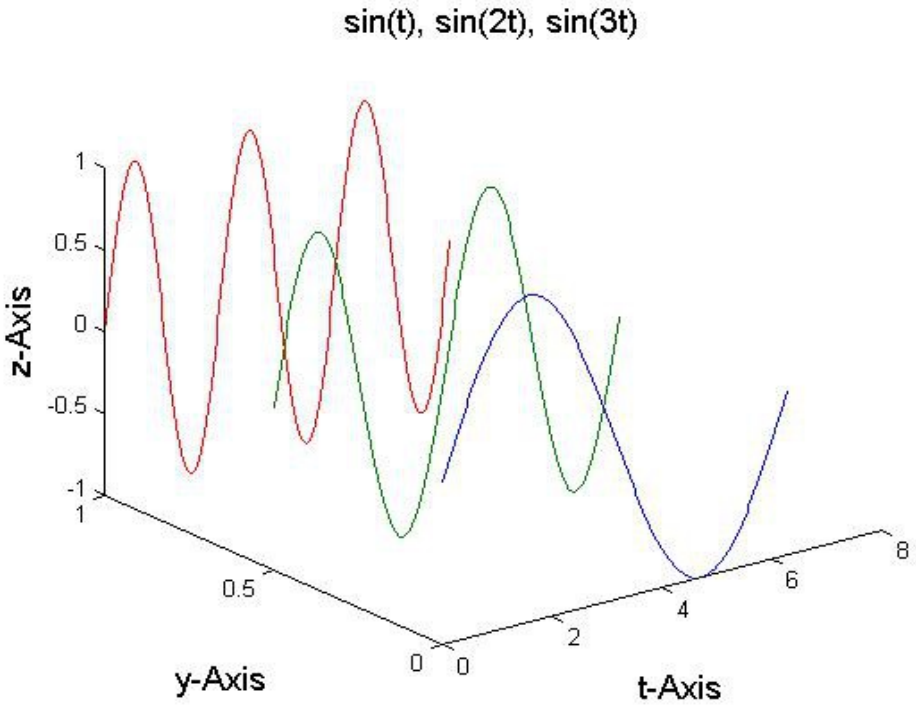
Esempio:

```
t=linspace(0,20*pi,2000);
plot3(sin(t),cos(t),t)
xlabel(('sin(t)')),
ylabel(('cos(t)')),zlabel(('t'))
grid on
title('Elica')
```



Grafici 3D: linee

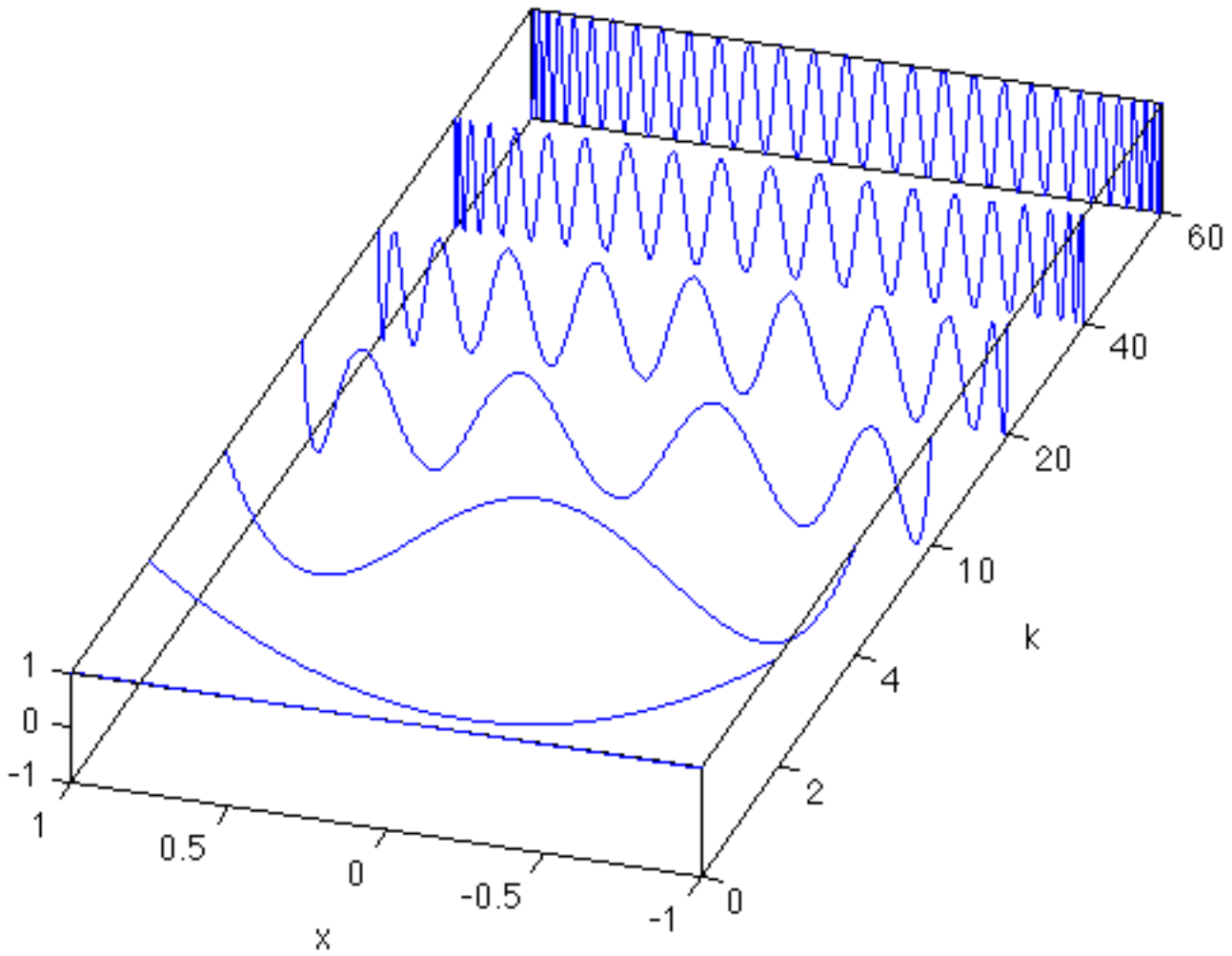
Esempio:




```
t=linspace(0,2*pi,200);
z1=sin(t);
z2=sin(2*t);
z3=sin(3*t);
y1=zeros(size(t));
y3=ones(size(t));
y2=y3/2;
grid on
plot3(t,y1,z1,t,y2,z2,t,y3,z3)
xlabel(('t-Axis'),'FontSize',14),ylabel(('y-Axis'),...
    'FontSize',14),zlabel(('z-Axis'),'FontSize',14)
```

Grafici 3D: linee (`cheb3plot.m`)

Esempio:



Grafici 3D: superfici

Rappresentazione di funzione scalare di 2 variabili:

```
z=f(x,y)
```

Dati i vettori x e y,
z e' una matrice data da:

```
z(i,:) = f(x,y(i))  e  z(:,j) = f(x(j),:)
```

Grafici 3D: superfici (`meshgrid`)

z(i,j) =f(x(j),y(i))

Esempio: Z=X+Y

```
>> x=-2:2
x =
   -2   -1    0    1    2
>> y=-4:4
y =
   -4   -3   -2   -1    0    1    2    3    4
```

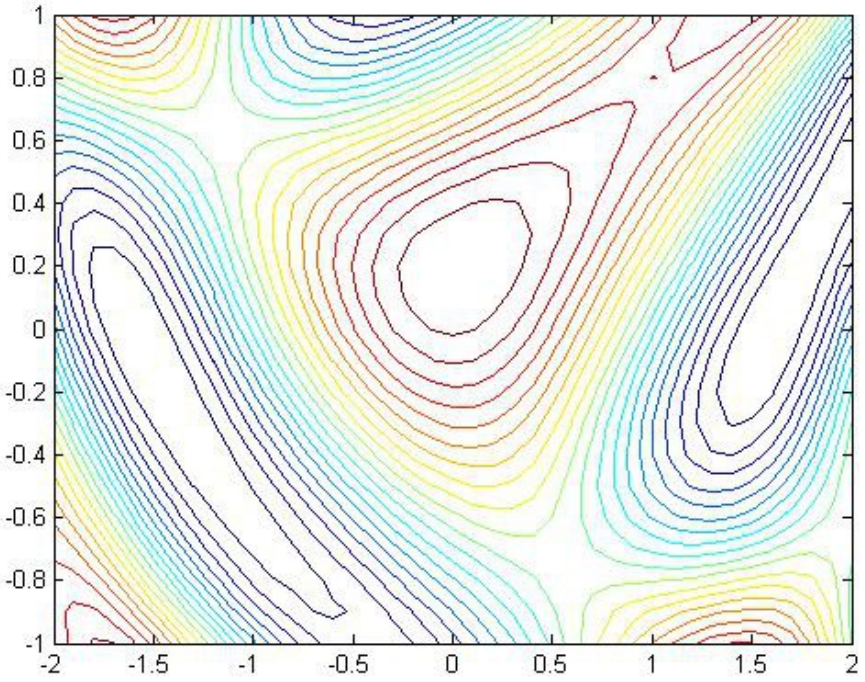
```
>> [X,Y]=meshgrid(x,y)
X =
   -2   -1    0    1    2
   -2   -1    0    1    2
   -2   -1    0    1    2
   -2   -1    0    1    2
   -2   -1    0    1    2
   -2   -1    0    1    2
   -2   -1    0    1    2
   -2   -1    0    1    2
   -2   -1    0    1    2
Y =
   -4   -4   -4   -4   -4
   -3   -3   -3   -3   -3
   -2   -2   -2   -2   -2
   -1   -1   -1   -1   -1
    0    0    0    0    0
    1    1    1    1    1
    2    2    2    2    2
    3    3    3    3    3
    4    4    4    4    4
```

```
>> Z=X+Y
Z =
   -6   -5   -4   -3   -2
   -5   -4   -3   -2   -1
   -4   -3   -2   -1    0
   -3   -2   -1    0    1
   -2   -1    0    1    2
   -1    0    1    2    3
    0    1    2    3    4
    1    2    3    4    5
    2    3    4    5    6
```

Grafici 3D: superfici (contour)

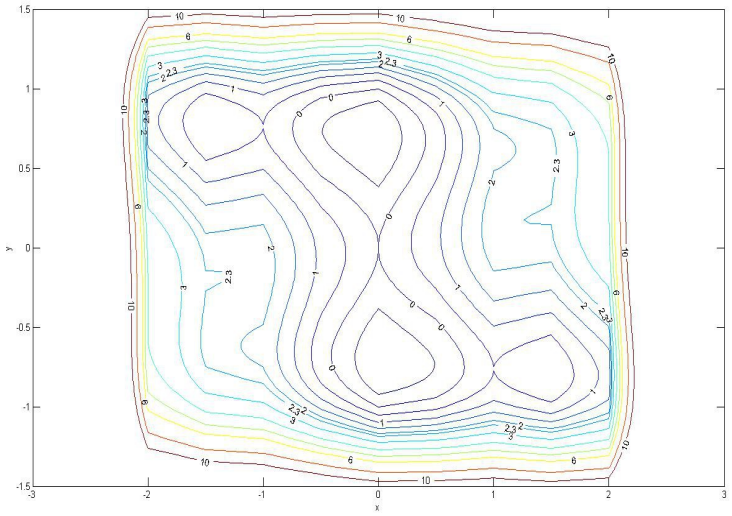
Esempio:

```
x=-2:0.1:2;
y=-1:0.1:1;

[X,Y]=meshgrid(x,y);

Z=sin(3*Y-X.^2+1)+cos(2*Y.^2-2*X);

contour(x,y,Z,20)
```

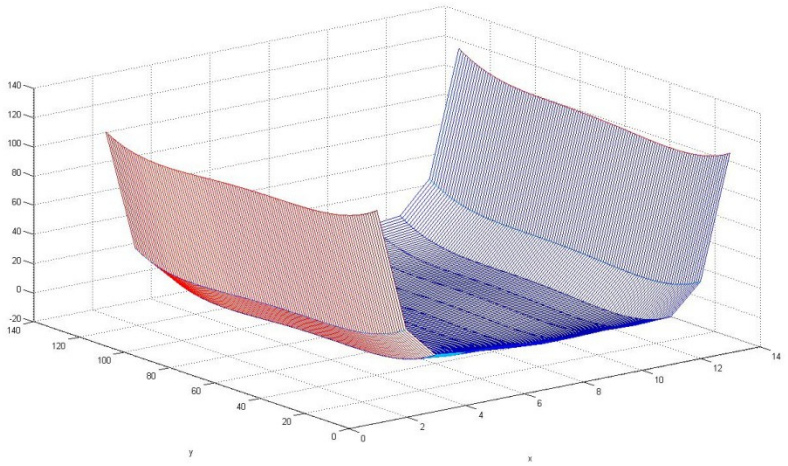


Grafici 3D: superfici (`contour - mesh`)

Esempi:



```
[X,Y]=meshgrid(-3:0.5:3,-1.5:0.025:1.5);
Z=4*X.^2-2.1*X.^4+X.^6/3+X.*Y-4*Y.^2+4*Y.^4;
cvals=[-2:.5:2 2.3 3:5 6:2:10];
[C,h]=contour(X,Y,Z,cvals);
clabel(C,h,cvals([1:2:9 10 11 14 16]))
xlabel('x'), ylabel('y')
```

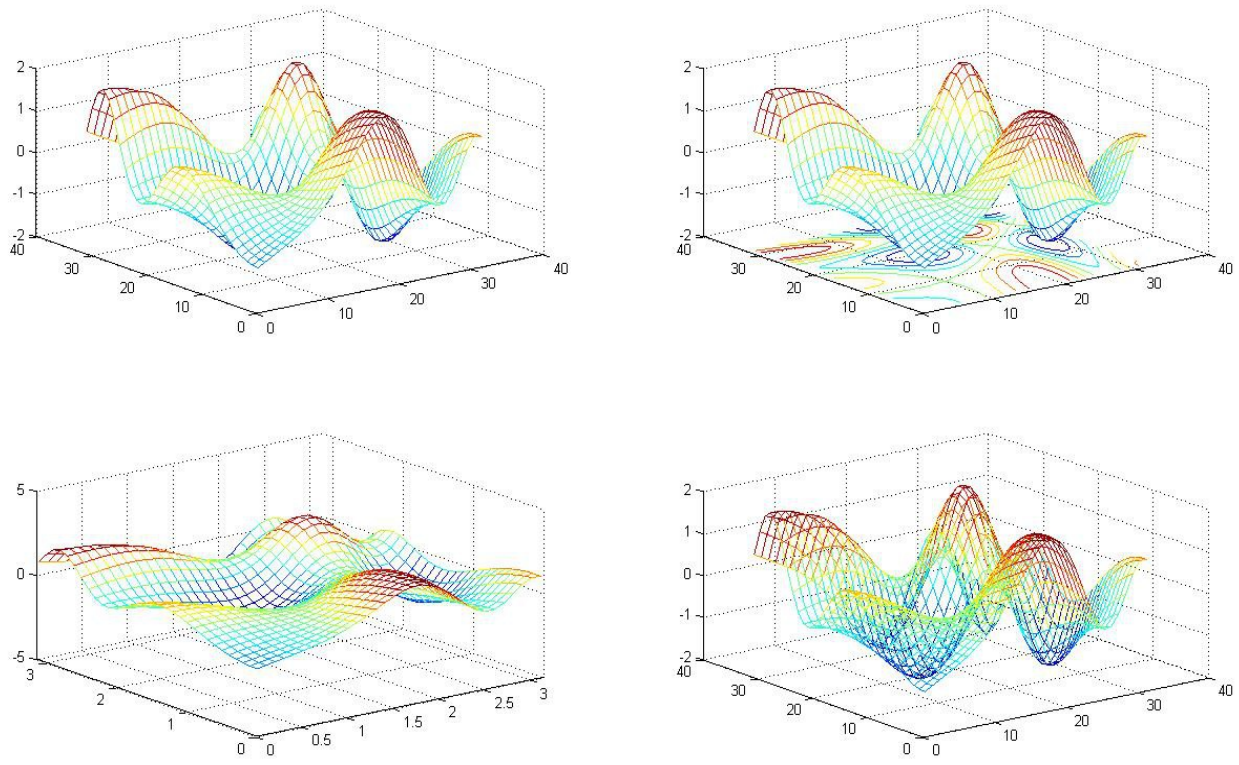


```
[X,Y]=meshgrid(-3:0.5:3,-1.5:0.025:1.5);
Z=4*X.^2-2.1*X.^4+X.^6/3+X.*Y-4*Y.^2+4*Y.^4;
mesh(Z)
xlabel('x'), ylabel('y')
```

Grafici 3D: superfici (mesh-meshc)

Esempi:

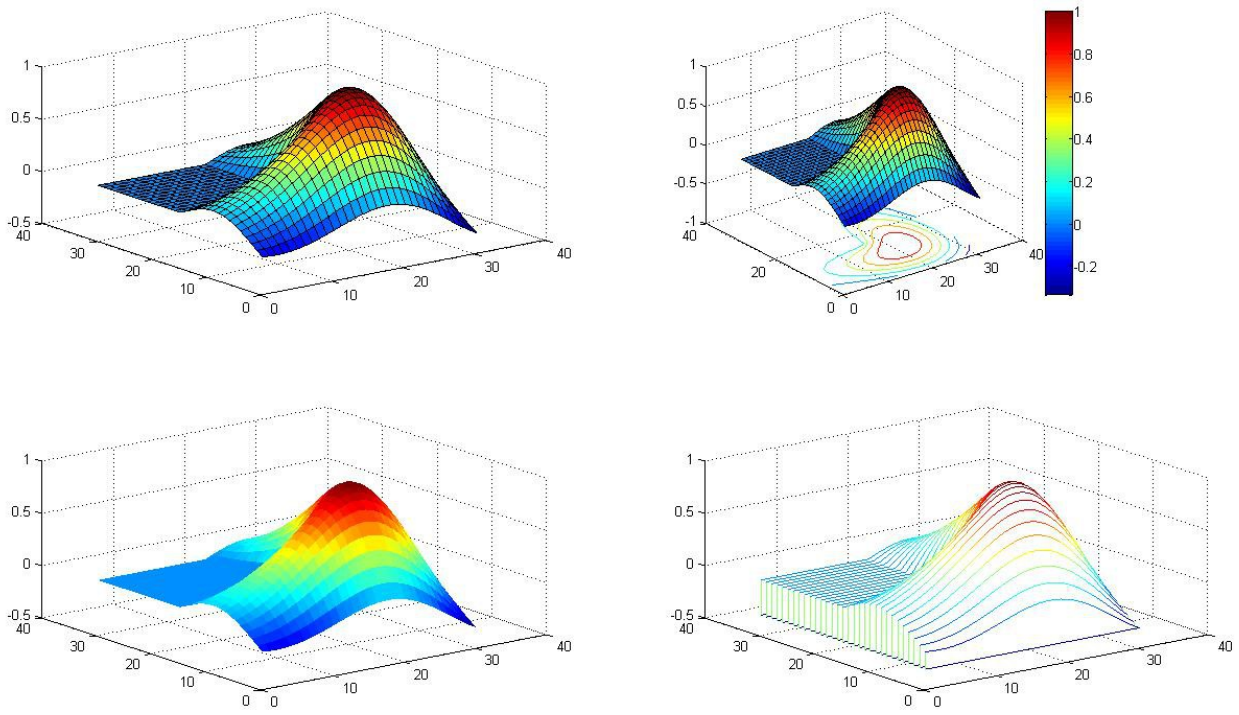
```
x=0:.1:pi; y=0:.1:pi;
[X,Y]=meshgrid(x,y);
Z=sin(Y.^2+X) -
cos(Y-X.^2);

subplot(2,2,1)
mesh(Z)

subplot(2,2,2)
meshc(Z)

subplot(2,2,3)
mesh(x,y,Z)
axis([0 pi 0 pi -5 5])

subplot(2,2,4)
mesh(Z)
hidden off
```



Grafici 3D: superfici (`surf-surfc-waterfall`)

Esempi:

```
Z=membrane;

subplot(2,2,1)
surf(Z)

subplot(2,2,2)
surfc(Z)
colorbar

subplot(2,2,3)
surf(Z)
shading flat

subplot(2,2,4)
waterfall(Z)
```

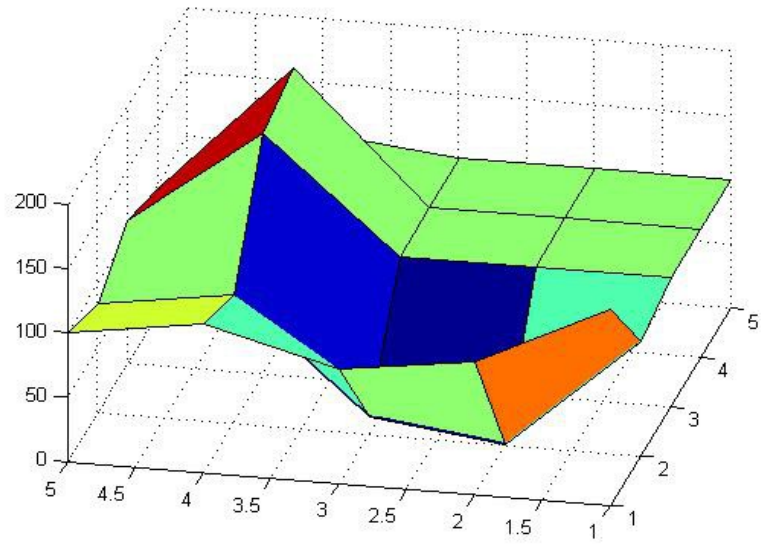


Grafici 3D: superfici (Caso terne di valori sperimentali)

Esempio:

```
A=[
1   1   152
2   1   89
3   1   100
4   1   100
5   1   100
1   2   103
2   2   0
3   2   100
4   2   100
5   2   100
1   3   89
2   3   13
3   3   100
4   3   100
5   3   100
1   4   115
2   4   100
3   4   187
4   4   200
5   4   111
1   5   100
2   5   85
3   5   111
4   5   97
5   5   48];
```

```
X=transp(reshape(A(:,1),5,5))
Y=transp(reshape(A(:,2),5,5))
Z=transp(reshape(A(:,3),5,5))
mesh(X,Y,Z)
```



Grafici 3-D: lista funzioni

```
plot3
contour
contour3
contourf
mesh
meshc
surf
surfc
waterfall
meshz
comet3
```

```
bar3
bar3h
pie
fill3
scatter3
stem3
```

Handle graphics

Le funzioni grafiche in Matlab sono parte di un sistema grafico *object-oriented* noto come *handle graphics.*

Un grafico è un oggetto composto da altri oggetti più elementari, in relazione gerarchica:

Root

Figure

Axes UI Objects

Image Light Line Patch Rectangle Surface Text

Handle graphics

Ogni oggetto ha un indentificatore unico (`handle`), che e' un *floating-point number*:

`root` `0`

`figure` `1,2,3,.....`

Una volta creati gli oggetti grafici, questi possono essere modificati, facendo riferimento ai loro `handles`, per mezzo delle funzioni `get` e `set`

Handle graphics: `get - set`

Words with most meanings in the Oxford English Dictionary:

1. set

.....

.....

6. get

(Russel Ash. The top ten of everything), citato in Higham-Higham, Matlab guide

Handle graphics

```
>> plot(sin(0:pi/2:2*pi))

>> h=findobj

h =

     0
  1.0000
160.0016
161.0021
```

```
>> get(h,'Type')

ans =

    'root'
    'figure'
    'axes'
    'line'
```

Handle graphics

Per avere la lista delle proprieta' associate all'handle, usiamo `set`:

```
>> set(h(3))
```

ActivePositionProperty: [position | {outerposition}]
ALim
ALimMode: [{auto} | manual]
AmbientLightColor
Box: [on | {off}]
CameraPosition
CameraPositionMode: [{auto} | manual]
CameraTarget
CameraTargetMode: [{auto} | manual]
CameraUpVector
CameraUpVectorMode: [{auto} | manual]
CameraViewAngle
CameraViewAngleMode: [{auto} | manual]
CLim
CLimMode: [{auto} | manual]
……..
…..

```
>> set(h(4))
```

Color: {}
EraseMode: {4x1 cell}
LineStyle: {5x1 cell}
LineWidth: {}
Marker: {14x1 cell}
MarkerSize: {}
MarkerEdgeColor: {2x1 cell}
MarkerFaceColor: {2x1 cell}
XData: {}
YData: {}
ZData: {}
ButtonDownFcn: {}
Children: {}
Clipping: {2x1 cell}
CreateFcn: {}
DeleteFcn: {}
BusyAction: {2x1 cell}
HandleVisibility: {3x1 cell}
……

Handle graphics

Esempi:

```
>> x=get(h(4),'XData'), y=get(h(4),'YData')
x =
   1    2    3    4    5
y =
      0    1.0000    0.0000   -1.0000   -2.0000

>> set(h(4),'Marker')
[ + | o | * | . | x | square | diamond | v | ^ | > | < | pentagram | hexagram | {none} ]

>> get(h(4),'Marker')
ans =
None

>> set(h(4),'Marker','s','MarkerSize',18)
>> set(h(3),'XScale')
[ {linear} | log ]
>> set(h(3),'Xscale','log')
```

Animazioni

Esistono due tipi di animazioni in Matlab:

1) Salvare una sequenza di figure, eseguendola con `movie`;

2) Animare un `plot`, agendo sulle proprieta'
   Xdata
   Ydata
   Zdata
   dell'oggetto.

Animazioni: movie

Esempio:

```
Z = peaks;
surf(Z);
axis tight
set(gca,'nextplot','replacechildren');

% Record the movie
for j = 1:20
   surf(sin(2*pi*j/20)*Z,Z)
   F(j) = getframe;
end


movie(F,10) % Play the movie ten times
```

Animazioni: comet

Esempio:

```
x=linspace(-2,2,50000);
y=exp(x).*sin(1./x);
comet(x,y)
```

Animazioni: comet (Epicycloid.m)

Esempio:

```
% Epicycloid

a=12; b=5;
t=0:0.05:10*pi;
x=(a+b)*cos(t)-b*cos((a/b+1)*t);
y=(a+b)*sin(t)-b*sin((a/b+1)*t);
figure
axis equal
axis([-25 25 -25 25])
grid on
title('Epicycloid: a=12, b=5','FontSize', 16)
xlabel('x(t)', 'FontSize', 16), ylabel('y(t)','FontSize',
16)
comet(x,y)
```

Animazioni: comet3

Esempio:

```
t = -pi:pi/50000:pi;
comet3(sin(5*t),cos(3*t),tan(t))
```

Animazioni: drawnow

Esempio:

```
x=linspace(-pi,pi,200000);
y=cos(tan(x))-tan(sin(x));
p=plot(x(1),y(1),'.','EraseMode','none','MarkerSize', 2);
axis([min(x) max(x) min(y) max(y)])
hold on
for i=1:length(x)
set(p,'XData',x(i),'YData',y(i))
drawnow
end
hold off
```

Notare: '*EraseMode*', '*none*'

Animazioni: drawnow

Esempio:

```
x=linspace(-pi,pi,200000);
y=cos(tan(x))-tan(sin(x));
p=plot(x(1),y(1),'.','EraseMode', 'background','MarkerSize', 5);
axis([min(x) max(x) min(y) max(y)])
hold on
for i=1:length(x)
set(p,'XData',x(i),'YData',y(i))
drawnow
end
hold off
```

Notare: '*EraseMode*', '*background':* I punti vengono cancellati dopo essere stati plottati

Animazioni: drawnow

Esempio:

```
% Programma animate.m - Anima il moto del proiettile
% Usa le funzioni xcoord, ycoord e vertvel.

th=45*(pi/180);
g=9.8;
v0=30;

tmax=2*v0*sin(th)/g;         %tempo totale di volo
xmax=xcoord(tmax,v0,th);     %gittata
ymax= ycoord(tmax/2,v0,th,g);  % altezza massima
vmax=vertvel(0,v0,th,g);
w=linspace(0,tmax,500);

subplot(2,1,1)
plot(xcoord(w,v0,th),ycoord(w,v0,th,g)), %hold
h1= plot(xcoord(w,v0,th),ycoord(w,v0,th,g), 'o','EraseMode','xor');
axis([0 xmax 0 1.1*ymax]), xlabel('x'),ylabel('y');
subplot(2,1,2)
plot(xcoord(w,v0,th),vertvel(w,v0,th,g)), %hold
h2= plot(xcoord(w,v0,th),vertvel(w,v0,th,g), 's','EraseMode','xor');
axis([0 xmax -1.1*vmax 1.1*vmax]), xlabel('x'),ylabel('Vertical Velocity');

for t=[0:0.01:tmax]
    set(h1,'Xdata',xcoord(t,v0,th),'Ydata',ycoord(t,v0,th,g))
    set(h2,'Xdata',xcoord(t,v0,th),'Ydata',vertvel(t,v0,th,g))
    drawnow
    pause(0.001)
end
hold
```

Animazioni: drawnow

Esempio:

```
t=[0:0.05:100];
b=1;
p = plot(t,t.*exp(-t/b),'EraseMode','xor'),axis([0 100 0 10]), xlabel('t');
for b=2:20
    set(p,'Xdata',t,'Ydata',t.*exp(-t/b)),axis([0 100 0 10]),xlabel('t');
drawnow
pause(0.1)
end
```

Ultimo esempio sulla versatilità di matlab nella grafica

```
%GARDEN

% Cols: Carrots|Broccoli|Green Beans|Cucumbers|Chard.  Rows are months.
Y = [0.4 0.3 0.0 0.0 0.9
     0.6 0.4 0.0 0.0 1.0
     0.7 0.8 0.3 0.2 1.2
     0.6 0.5 0.9 0.4 1.1
     0.4 0.4 0.7 0.6 0.9];

t = [13 15 22 24 18]; % Temperature.

b = bar(Y,'stacked');
ylabel('Yield (kg)'), ylim([0 4])

h1 = gca; % Handle of first axis.
set(h1,'XTickLabel','May|June|July|August|September')

% Create a second axis at same location as first and plot to it.
h2 = axes('Position',get(h1,'Position'));
p = plot(t,'Marker','square','MarkerSize',12,'LineStyle','-',...
         'LineWidth',2,'MarkerFaceColor',[.6 .6 .6]);
ylabel('Degrees (Celsius)')
title('Fran''s vegetable garden','FontSize',14)

% Align second x-axis with first and remove tick marks and tick labels.
set(h2,'Xlim',get(h1,'XLim'),'XTick',[],'XTickLabel',[])
% Locate second y-axis on right, make background transparent.
set(h2,'YAxisLocation','right','Color','none')

% Make second y-axis tick marks line up with those of first.
ylimits = get(h2,'YLim');
yinc = (ylimits(2)-ylimits(1))/4;
set(h2,'Ytick',[ylimits(1):yinc:ylimits(2)])

% Give legend the Axes handles and place top left.
legend([b,p],'Carrots','Broccoli','Green Beans','Cucumbers',...
       'Swiss Chard','Temperature','Location','NW')
```