UD6 - MATLAB

Gestione della grafica

Grafici

- MatLab può produrre grafici 2D e 3D

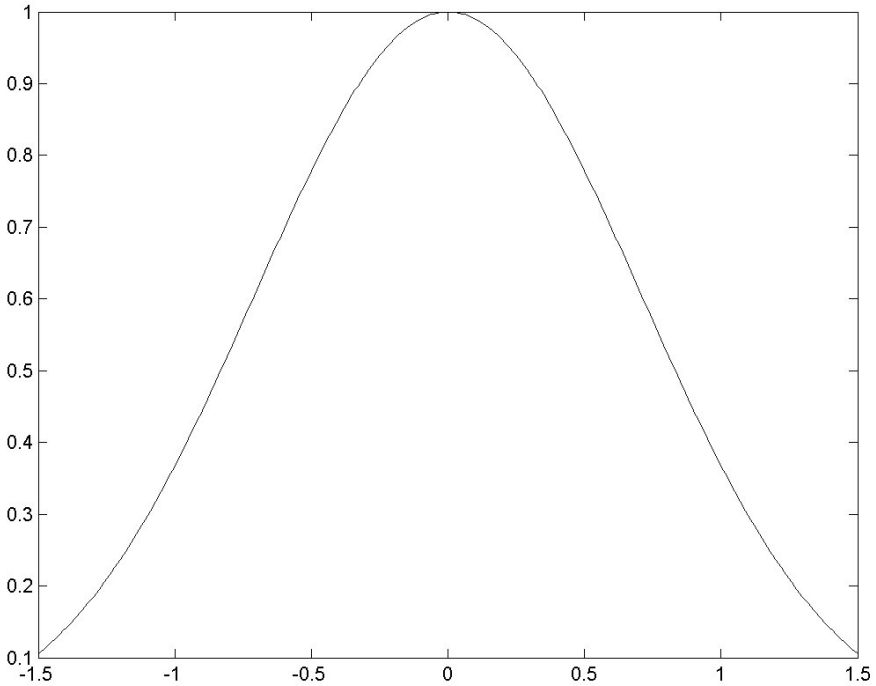
Disegnare un Grafico

- Il comando `plot` produce grafici in 2 dimensioni;
- `plot(x,y)` apre una finestra e disegna il punto `(x,y)`;
- Se `x` e `y` sono vettori, `plot(x,y)` traccia il grafico dei punti (x_i, y_i);
- Es: `x = -4:.01:4; y = sin(x); plot(x,y)` traccia il grafico della funzione `sin` nell'intervallo `[-4, 4]` con passo `0.01`;
- Il comando `shg` apre una finestra grafica;

Esempio di Grafico 2D

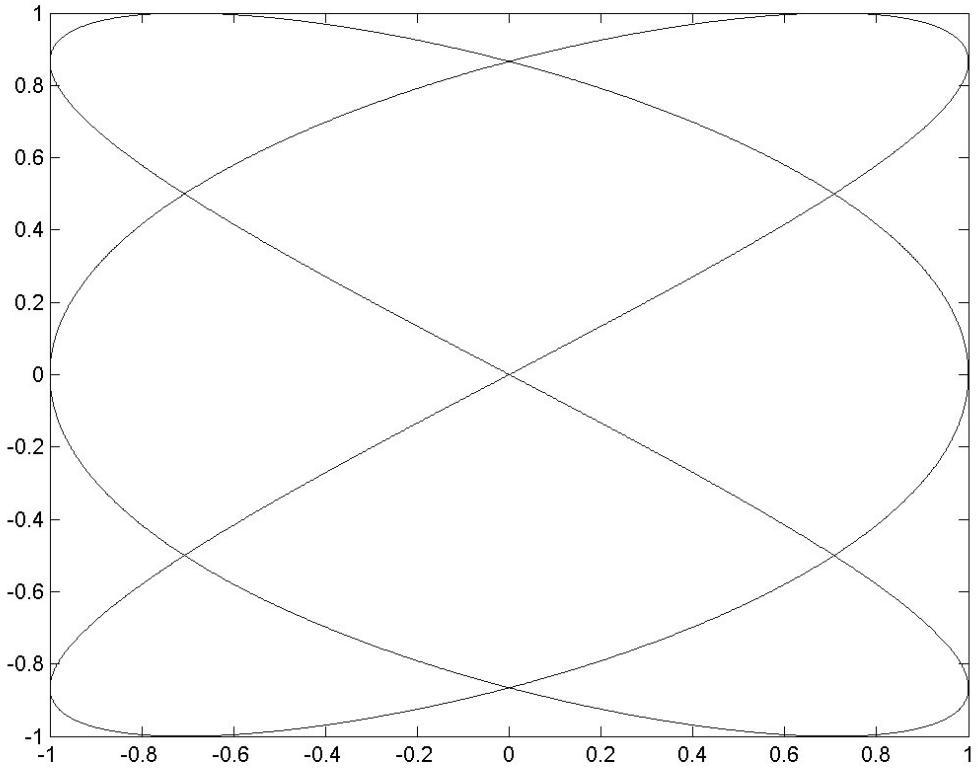
- Es: Grafico della funzione $y = e^{-x^2}$

```
x = -1.5:.01:1.5; y = exp(-x.^2); plot(x,y)
```



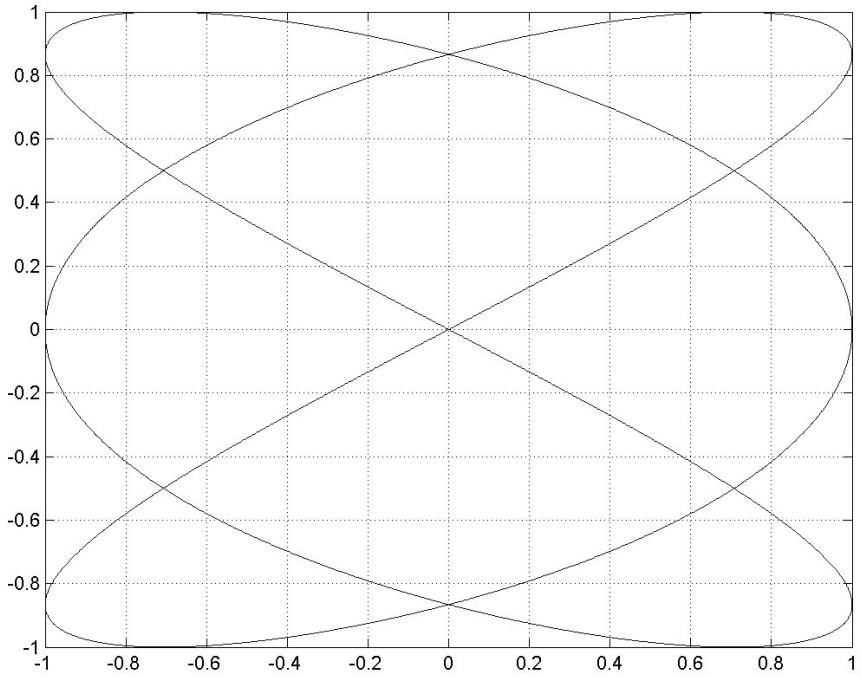
Esempio di Grafico 2D

```
t=0:.001:2*pi; x=cos(3*t); y=sin(2*t); plot(x,y)
```



Griglia su grafico 2D

- Il comando `grid` disegna una griglia sul grafico corrente



Etichette su grafici

- Ad un grafico possono essere assegnati

`title('testo da inserire')`	titolo del grafico
`xlabel('…')`	etichetta asse X
`ylabel('…')`	etichetta asse Y
`gtext('…')`	testo da posizionare interattivamente col mouse
`text(xi,yi, 'testo')`	testo da posizionare al punto di coordinate (x_i,y_i)

- Tali operazioni possono essere effettuate anche mediante l'interfaccia grafica;

Fattore di scala per le assi

- Per default l'unità di misura per le assi è calcolata in maniera automatica (scala automatica);
- Il comando `axis([Xmin,Xmax,Ymin,Ymax])` assegna in maniera esplicita un intervallo (e quindi una scala) per la rappresentazione dei valori sull'asse X e Y;
- `axis('square')` assicura che il grafico sia quadrato;

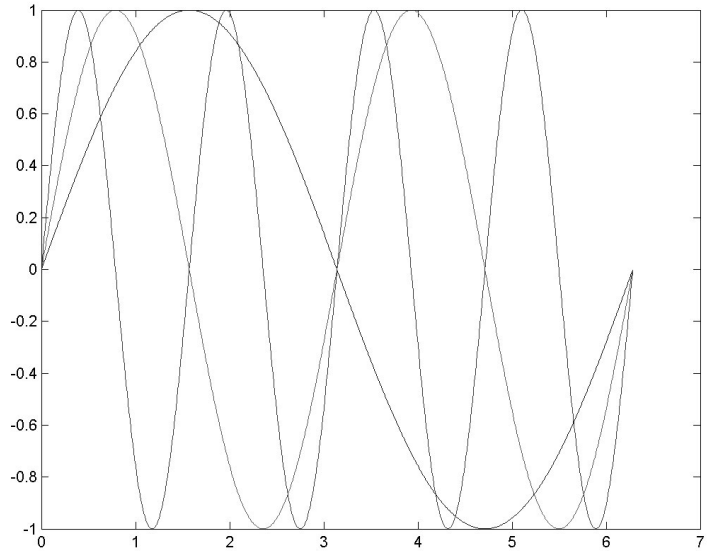
Tracciamento multiplo di grafici (1)

- Es:

```
x=0:.01:2*pi; y1=sin(x); y2=sin(2*x); y3=sin(4*x);

plot(x,y1,x,y2,x,y3)
```

oppure:

```
x=0:.01:2*pi; Y=[sin(x)', sin(2*x)', sin(4*x)'];

plot(x,Y)
```



Tracciamento multiplo di grafici (2)

- Un altro modo è fornito dal comando `hold`:
  - `hold on` mantiene un grafico precedente e un nuovo grafico viene sovrimposto;
  - `hold off` rilascia un grafico precedente e un nuovo grafico viene sostituito al precedente;
  - `hold` modifica l'impostazione hold precedente

Tipi di simboli per il tracciamento di grafici

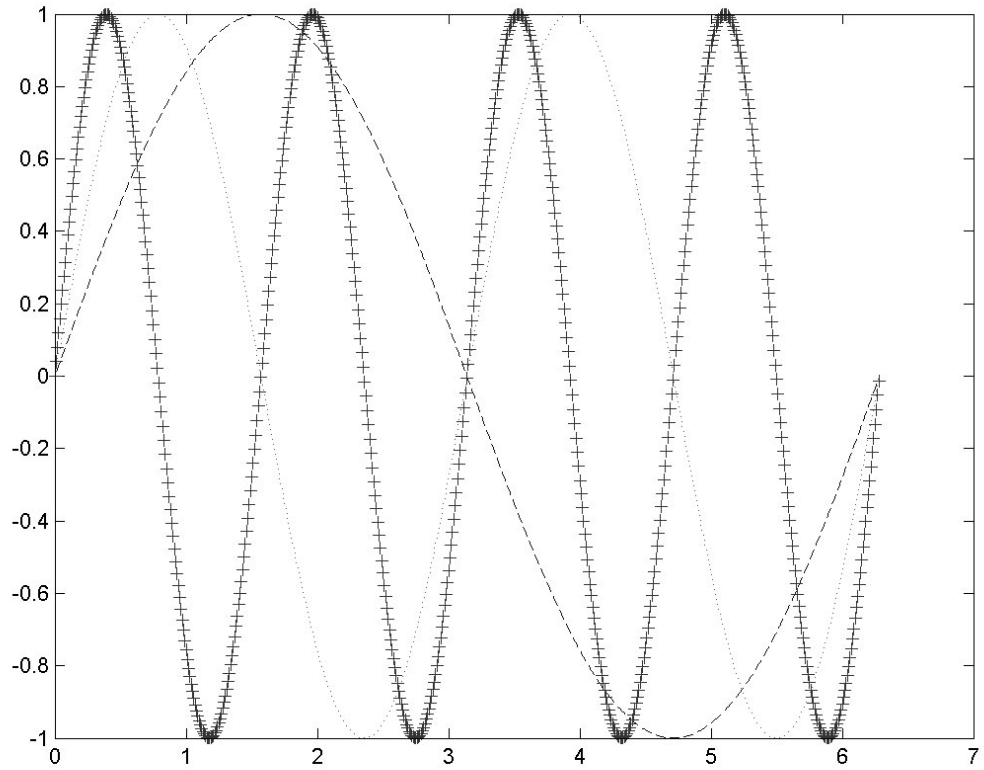
- E’ possibile modificare il tipo di simbolo da utilizzare per il tracciamento di un grafico utilizzando il comando `plot(x,y,s)`, dove s è una stringa che può assumere uno dei simboli per ciascuna delle seguenti 3 colonne:

b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star		
y	yellow	s	square		
k	black	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

Esempio

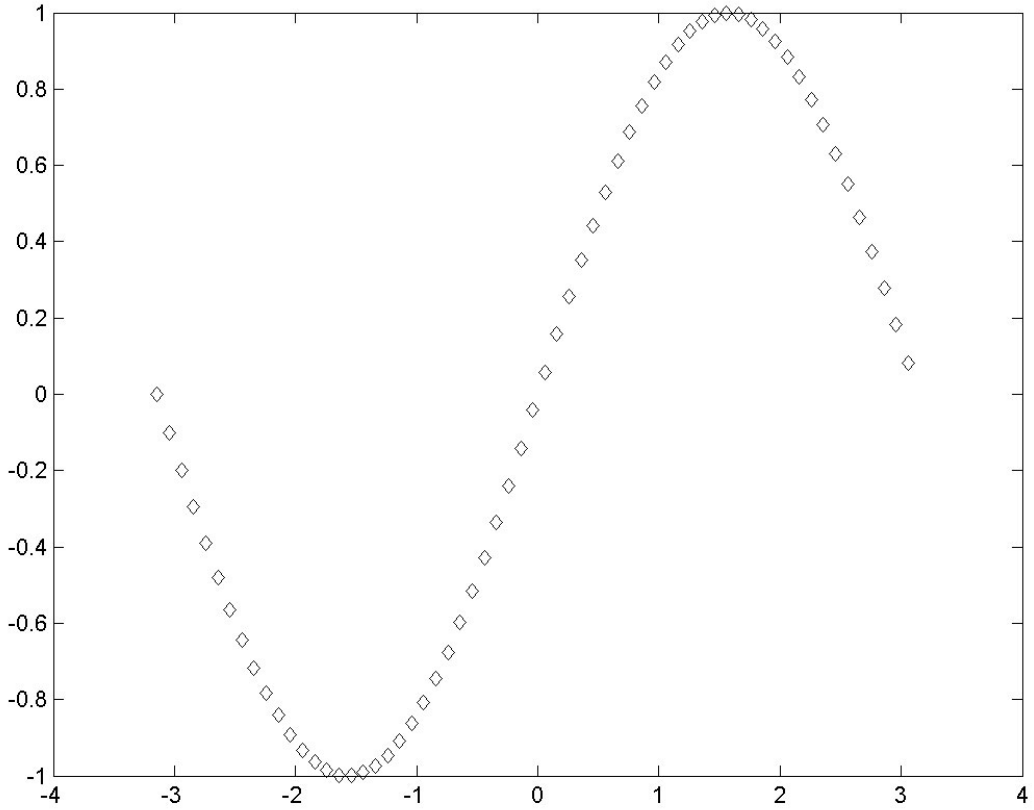
- Es:

```
x=0:.01:2*pi;
y1=sin(x);
y2=sin(4*x);
plot(x,y1,'--',
     x,y2,'+')
```



Esempio

```
x=-pi:.1:pi; y = sin(x); plot(x,y,'bd')
```

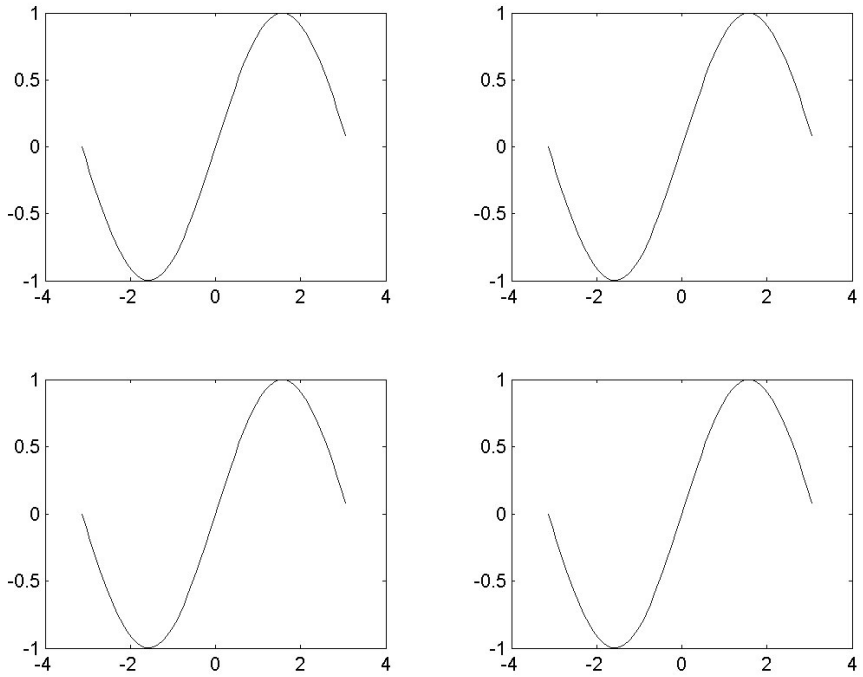


Partizionamento dello schermo

- Il comando `subplot` può essere usato per partizionare lo schermo:

  `subplot(m,n,p)`

  partiziona lo schermo in una matrice `m x n` e posiziona il successivo plot nella cella p-esima (numerate dalla cella in alto a sinistra fino a quella in basso a destra)



Comando `fplot`

- `fplot` consente di determinare automaticamente il numero dei punti da utilizzare (valori della variabile indipendente) in modo che il diagramma presenti tutte le caratteristiche della funzione;

  `fplot('stringa', [xmin xmax])`

  `stringa =` funzione da rappresentare

  `[xmin xmax] =` valori min e max della variabile indipendente

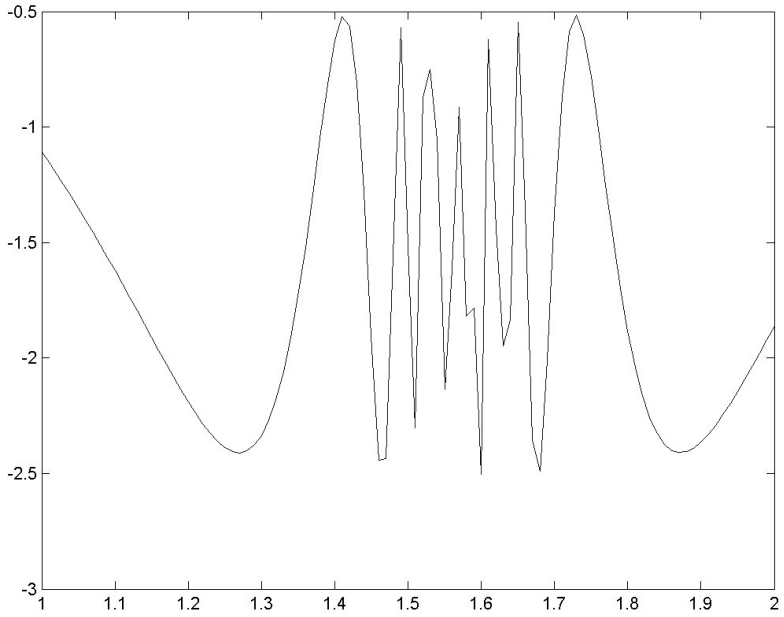
  Altra forma:

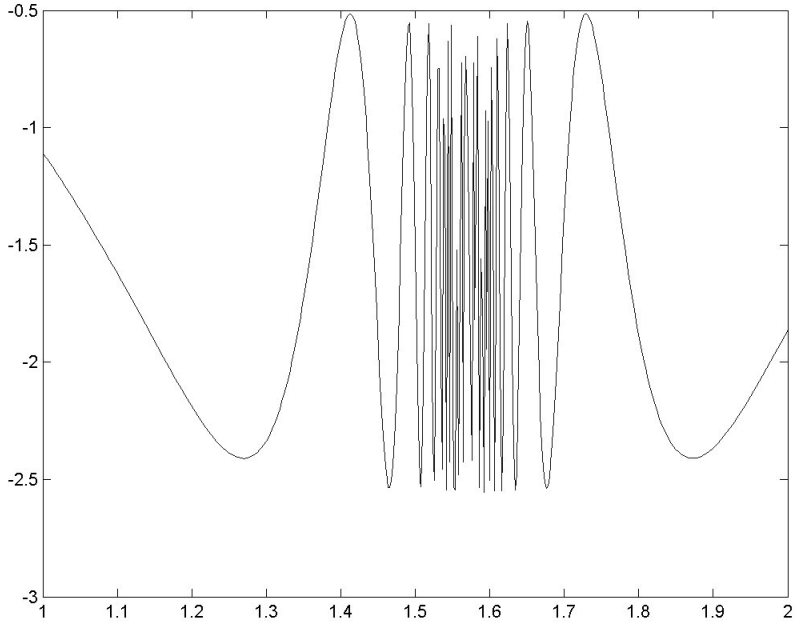
  `[x,y] = fplot('stringa', [xmin xmax ymin ymax])`

  Non genera alcun grafico ma restituisce i valori di x e y calcolati negli intervalli [xmin, xmax] [ymin, ymax] con un passo scelto automaticamente.

Confronto comandi `plot/fplot`

```
>> x = [1:0.01:2];
>> y = cos(tan(x)) - tan(sin(x));
>> plot(x,y);
```






```
>> f = 'cos(tan(x)) - tan(sin(x))'
>> fplot(f,[1 2])
```

Grafico di un polinomio

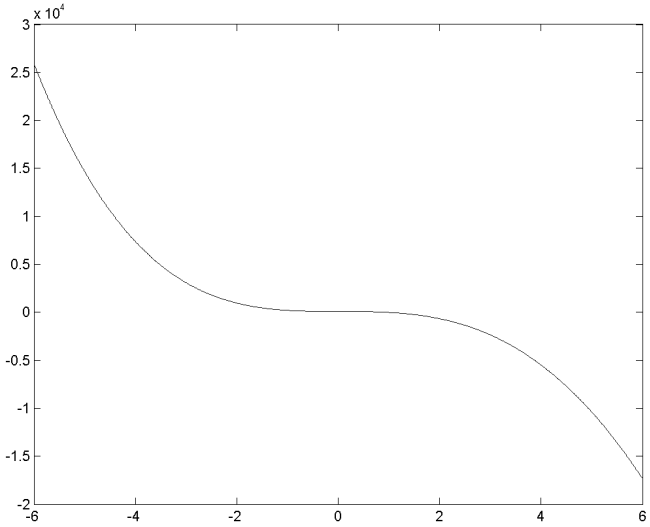
- La funzione `y = polyval(p,x)`, dove:

  `p` = vettore di dimensione N+1 dei coefficienti del polinomio di grado N;

  `x` = vettore dei valori della variabile indipendente,

  calcola i valori del polinomio e li assegna al vettore y;
- `plot(x, polyval(p,x))` disegna il grafico del polinomio;
- Es:

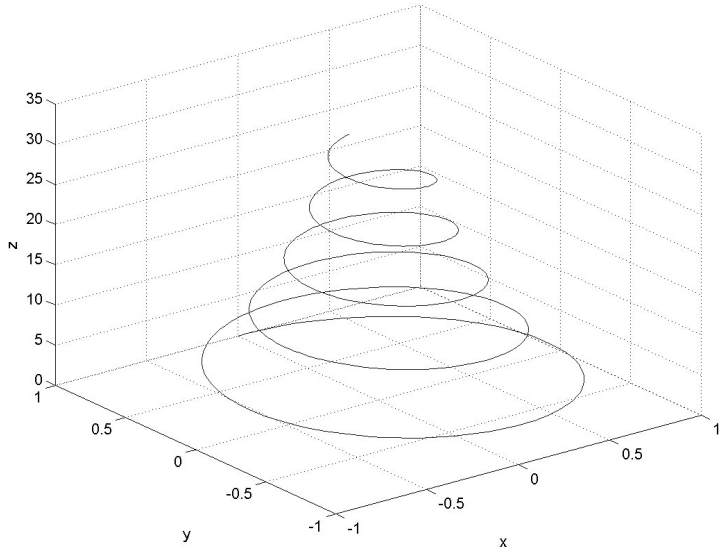
```
x = [-6;.01;6]
p = [3.2,-100,2,-7,90]
plot(x,polyval(p,x))
```



Grafici 3D a linea

- Le linee nello spazio tridimensionale possono essere rappresentate con il comando `plot3(x,y,z);`
- Es:

$$x = e^{-0.05t}\sin(t)$$
$$y = e^{-0.05t}\cos(t)$$
$$z = t$$




```
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
```

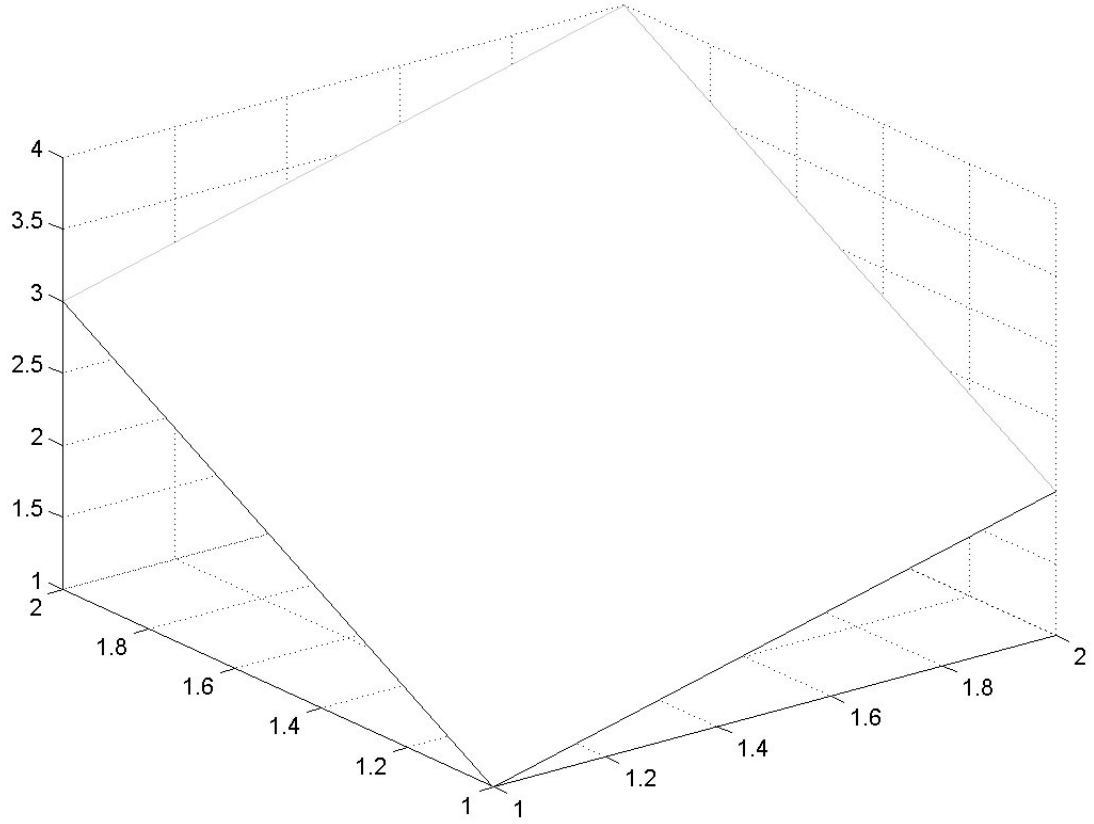
Grafici 3D a superficie

- I grafici tridimensionali a superficie si ottengono con la funzione `mesh`;
- `mesh(z)` crea una prospettiva tridimensionale dei dati della matrice z;
- La superficie mesh è definita dagli elementi della matrice z come valori z(x,y) dei punti di coordinate (x,y);

Esempio

- Es:

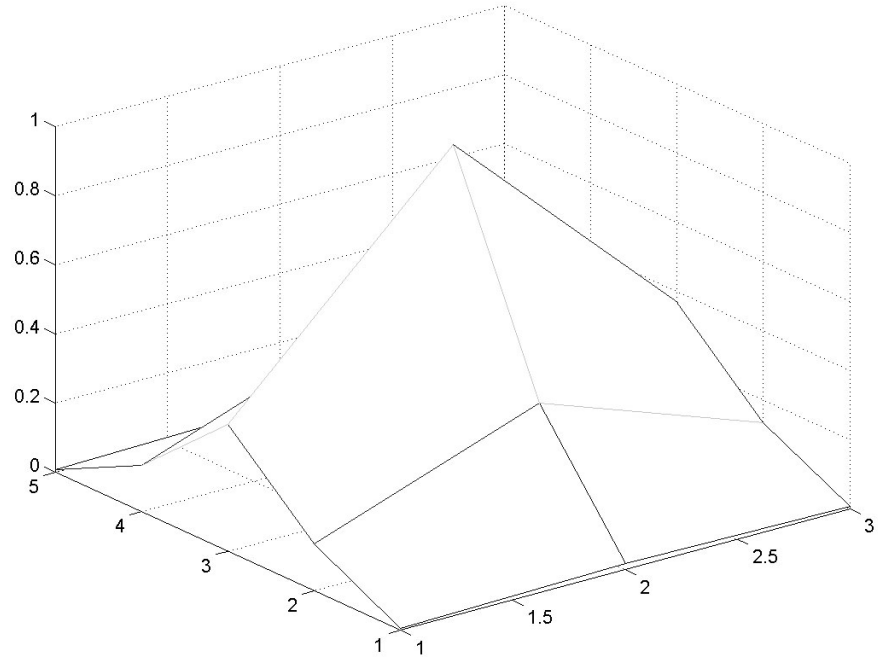
```
z = [1 2;
     3 4]
```



Grafici di funzioni a 2 variabili

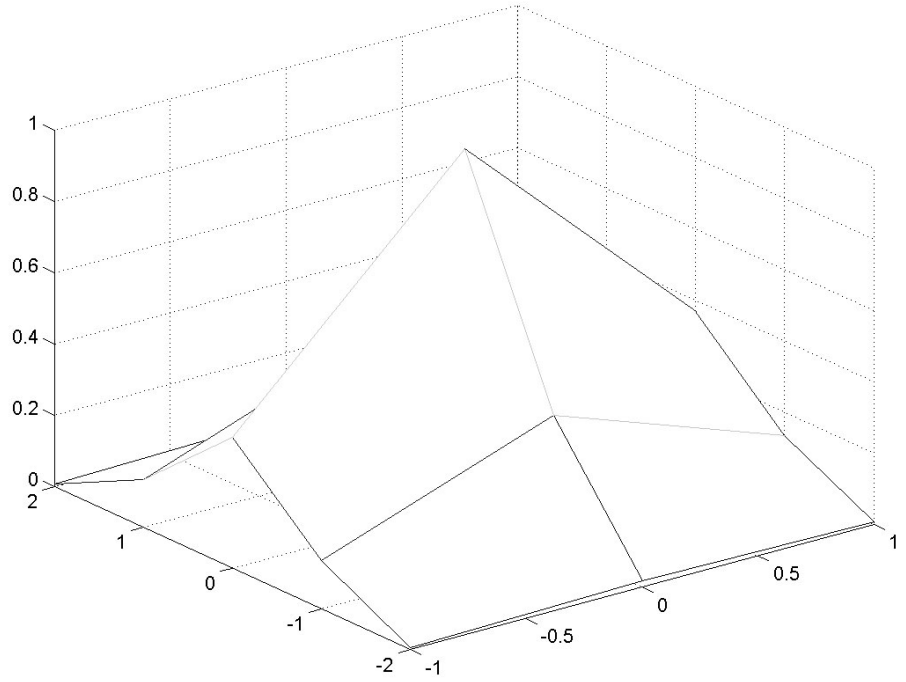
- Per visualizzare il grafico di una funzione `z=f(x,y)` su un rettangolo, è necessario creare due vettori `xx` e `yy` che contengono i valori degli intervalli per x e y sui quali calcolare la funzione;
- Mediante la funzione `[X,Y] = meshgrid(xx,yy)` è possibile creare:
  - una matrice X di dim(yy) righe, le cui righe sono rappresentate dai valori di xx;
  - una matrice Y di dim(xx) colonne, le cui colonne sono rappresentare dai valori di yy;
- In questo modo per calcolare il valore della funzione $z = f(x, y)$ è sufficiente calcolare $z = f(X(i, j), Y(i, j)), i \in [1, m],\ j \in [1, n]$ ottenendo il grafico per i valori normalizzati delle variabili x e y negli intervalli `x = [1:m] y = [1:n]`

```
[x,y] = meshgrid(-1:1:1,-2:1:2);
x = -1     0     1
    -1     0     1
    -1     0     1
    -1     0     1
    -1     0     1

y = -2    -2    -2
    -1    -1    -1
     0     0     0
     1     1     1
     2     2     2

z = exp(-x.^2 - y.^2)
z = 0.0067    0.0183    0.0067
    0.1353    0.3679    0.1353
    0.3679    1.0000    0.3679
    0.1353    0.3679    0.1353
    0.0067    0.0183    0.0067
mesh(z)
```



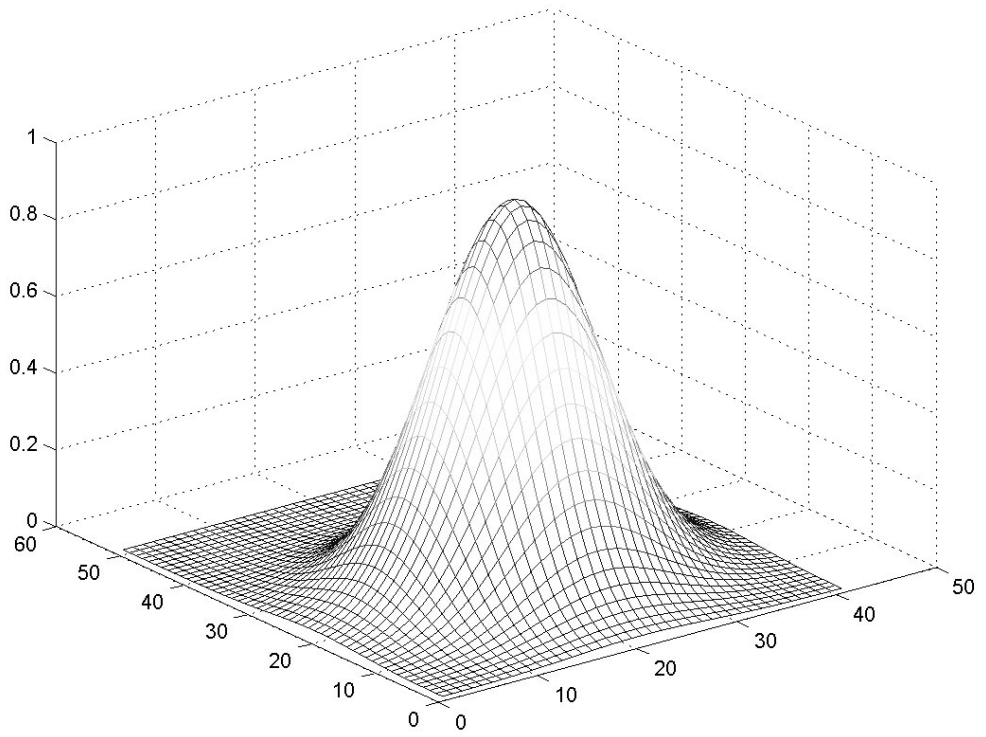
Grafici funzioni a 2 variabili: visualizzazione dei domini per le variabili

```
[x,y] = meshgrid(-1:1:1,-2:1:2);
z = exp(-x.^2 - y.^2);
mesh(x,y,z)
```



Esempio

```
[x,y] = meshgrid(-2:.1:2,-3:.1:3);
z = exp(-x.^2 - y.^2);
mesh(z);
```



Esempio

```
[x,y] = meshgrid(-2:.1:2,-3:.1:3);
z = exp(-x.^2 - y.^2);
mesh(x,y,z);
```