

UNIVERSITÀ DEGLI STUDI DI SALERNO

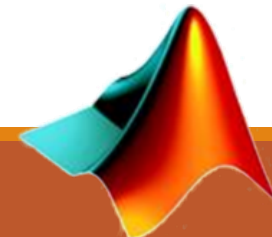
# Fondamenti di Informatica

---

Grafici in MATLAB

Prof. Arcangelo Castiglione

A.A. 2016/17



MATLAB®

# Grafici in MATLAB – 1/13

---

- `plot(x, y)`

- MATLAB genera un grafico basandosi sull'array `x` per l'**asse X** e sull'array `y` per l'**asse Y**
  - **NOTA**: Gli array `x` ed `y` devono essere della stessa lunghezza

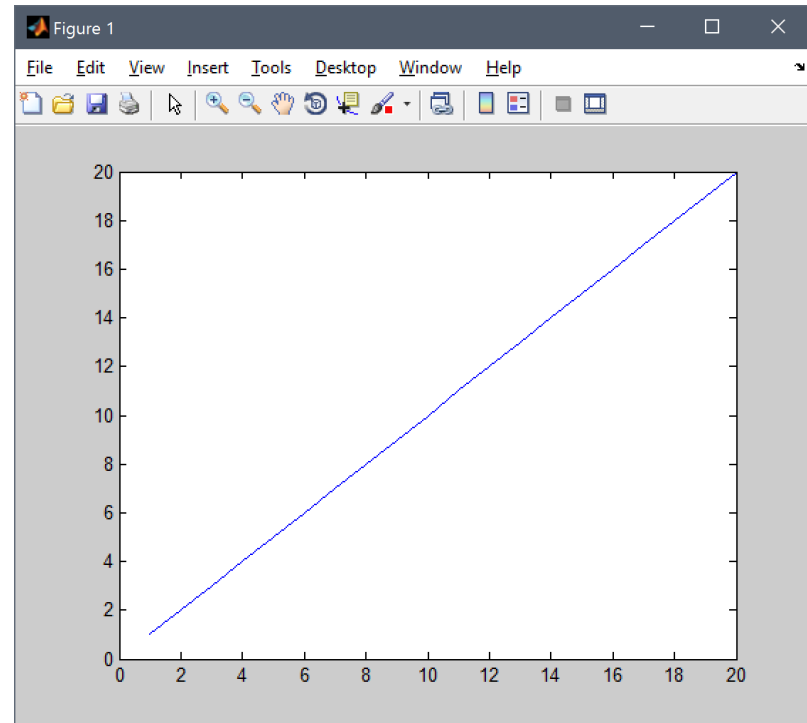
- `plot(x)`

- MATLAB genera un grafico lineare basandosi sull'array `x` sia per l'**asse X** che per l'**asse Y**
  - Ciascun punto ha lo stesso valore sia per la coordinata sull'asse X che per quella sull'asse Y

# Grafici in MATLAB – 2/13

- *Esempio 1 (array  $x$  su asse  $X$  ed asse  $Y$ )*

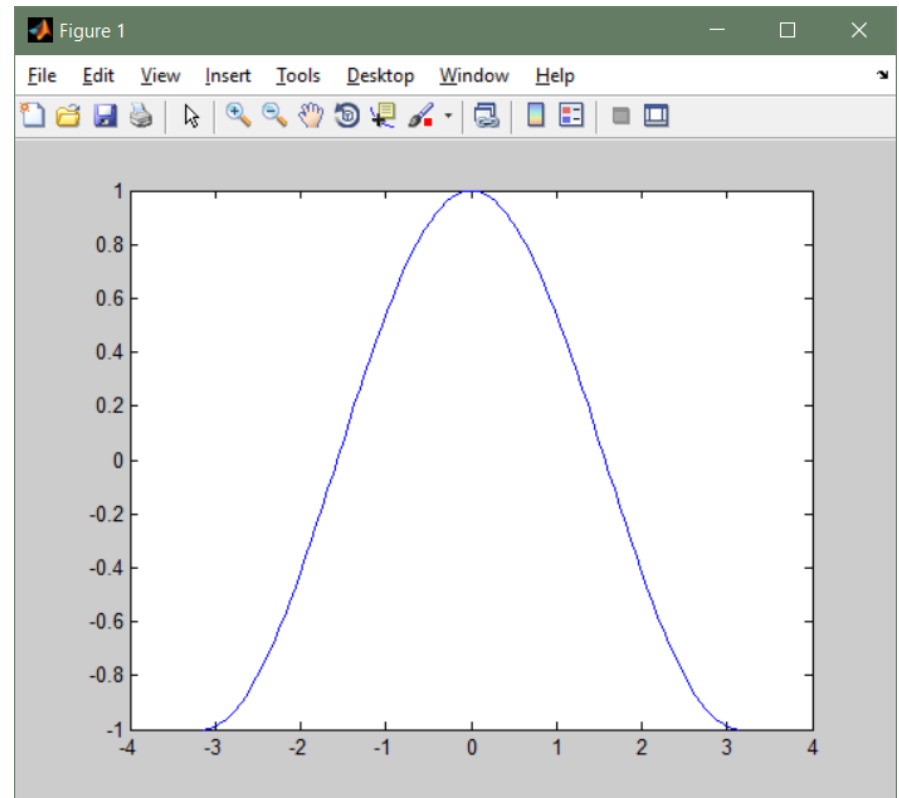
```
x = [1:1:20]  
plot(x)
```



# Grafici in MATLAB – 3/13

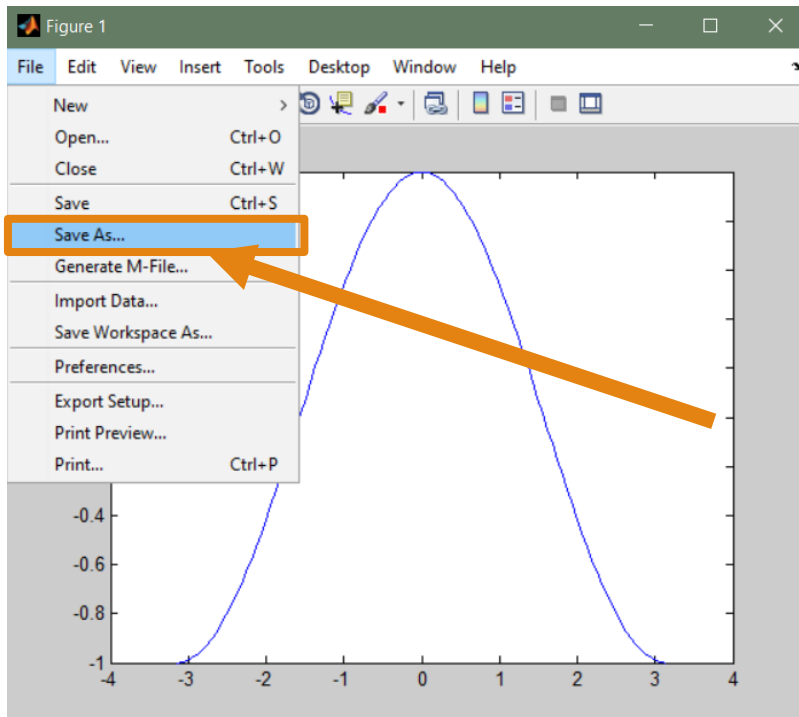
- *Esempio 3 (array  $x$  su asse  $X$  ed array  $y$  su asse  $Y$ )*

```
x = linspace(-pi, pi);  
y = cos(x);  
plot(x, y)
```



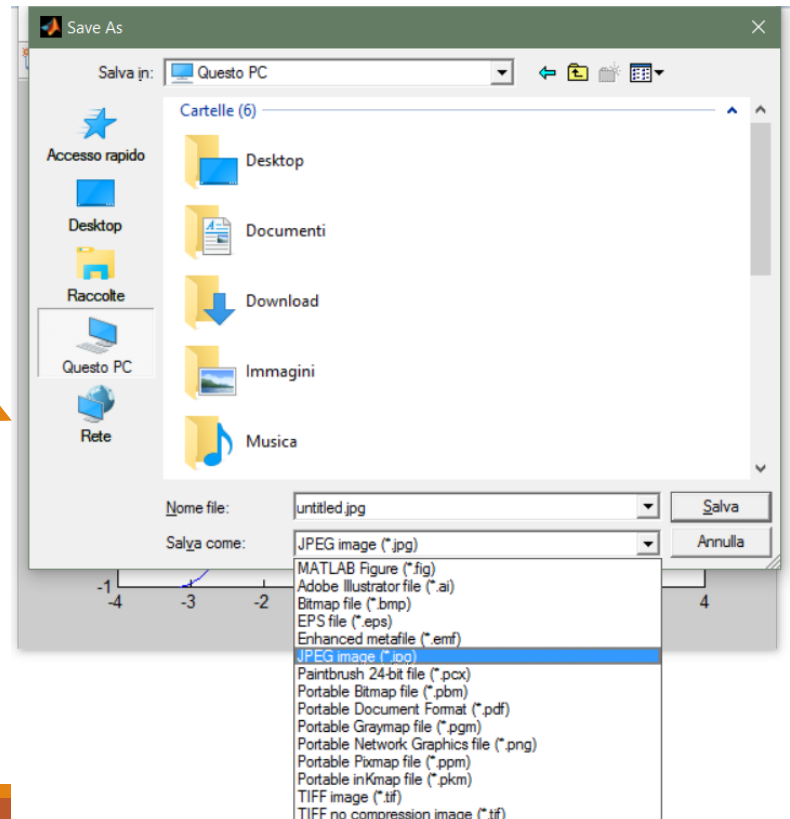
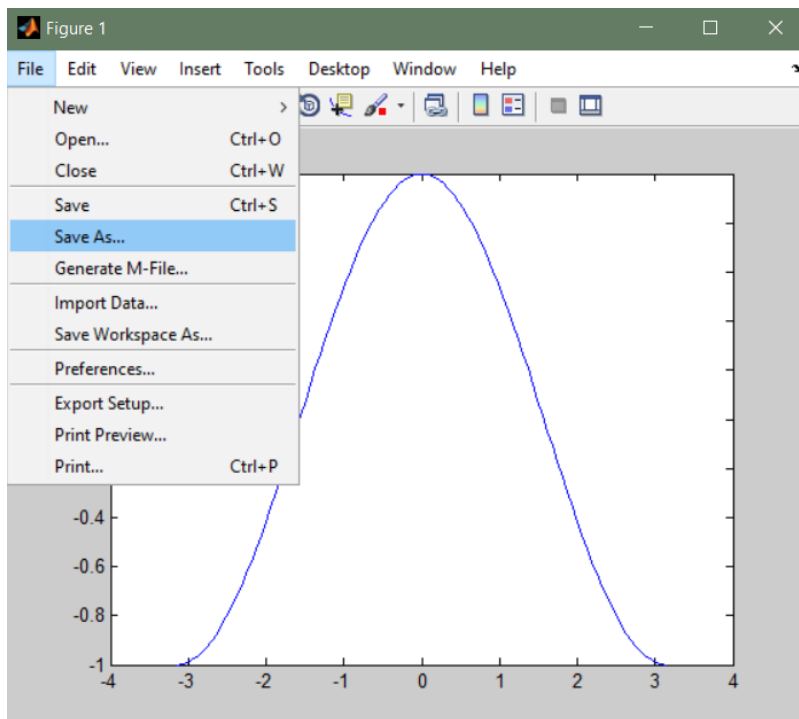
# Grafici in MATLAB – 4/13

- Possiamo salvare un grafico per poterlo usare altrove (esempio *Microsoft Word, OpenOffice, TeX, etc*)
- Andando sul menu «**File**» e «**Save As...**»



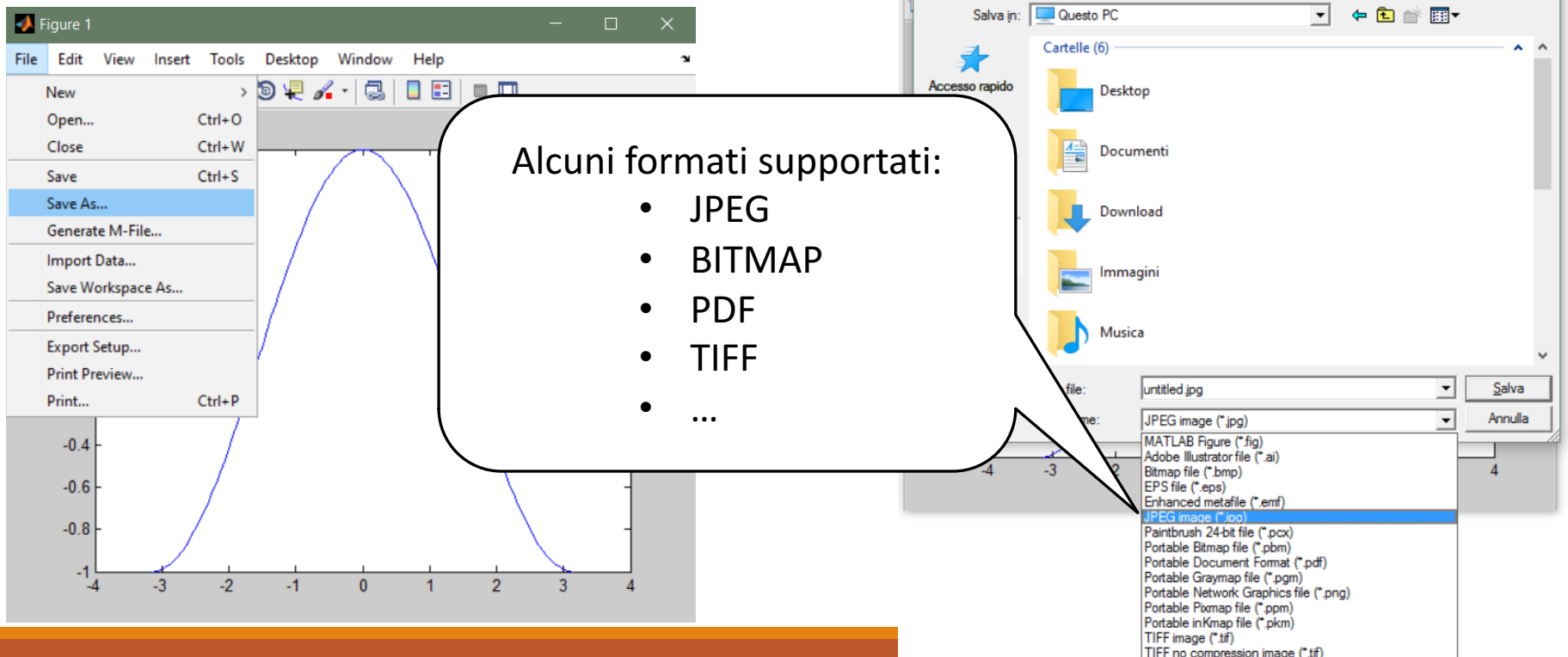
# Grafici in MATLAB – 4/13

- Possiamo salvare un grafico per poterlo usare altrove (esempio *Microsoft Word, OpenOffice, TeX, etc*)
- Andando sul menu «**File**» e «**Save As...**»



# Grafici in MATLAB – 4/13

- Possiamo salvare un grafico per poterlo usare altrove (esempio *Microsoft Word, OpenOffice, TeX, etc*)
- Andando sul menu «**File**» e «**Save As...**»



The image shows a MATLAB Figure window titled 'Figure 1' with a plot of a sine wave. The 'File' menu is open, and 'Save As...' is selected. A 'Save As' dialog box is open, showing the file name 'untitled.jpg' and a list of supported file formats. A callout box points to the list of formats.

Alcuni formati supportati:

- JPEG
- BITMAP
- PDF
- TIFF
- ...

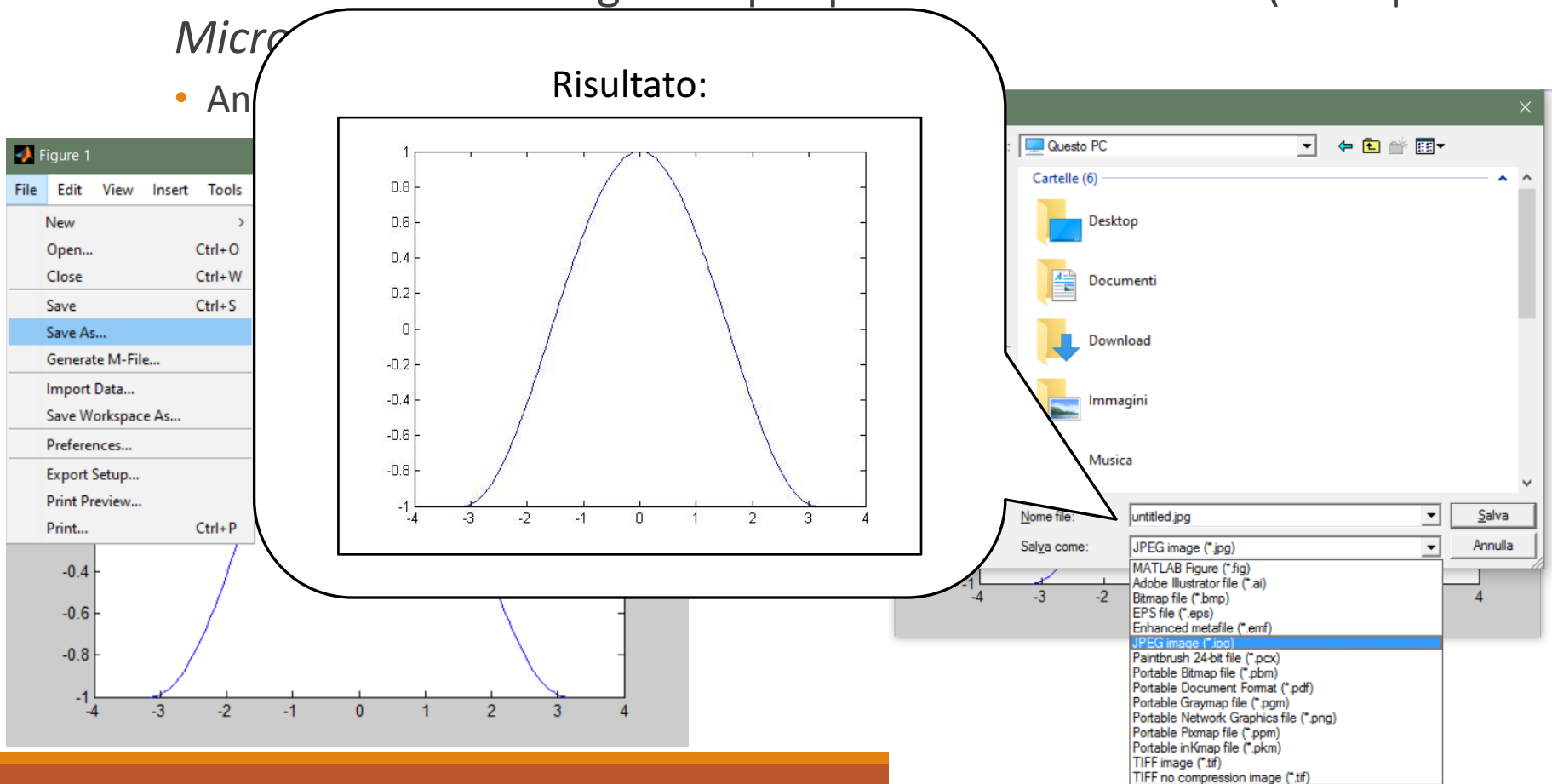
The 'Save As' dialog box shows the following file formats in the list:

- JPEG image (\*.jpg)
- Paintbrush 24-bit file (\*.pcx)
- Portable Bitmap file (\*.pbm)
- Portable Document Format (\*.pdf)
- Portable Graymap file (\*.pgm)
- Portable Network Graphics file (\*.png)
- Portable Pixmap file (\*.ppm)
- Portable inKmap file (\*.pkm)
- TIFF image (\*.tif)
- TIFF no compression image (\*.tif)

# Grafici in MATLAB – 4/13

- Possiamo salvare un grafico per poterlo usare altrove (esempio *Micro*)
- An

Risultato:



The image shows a MATLAB interface with a plot of a Gaussian function. The plot has an x-axis from -4 to 4 and a y-axis from -1 to 1. A callout bubble labeled 'Risultato:' points to the plot. To the left, the 'File' menu is open, with 'Save As...' selected. To the right, a 'Save As' dialog box is open, showing the file name 'untitled.jpg' and a list of file formats, with 'JPEG image (\*.jpg)' selected. The dialog box also shows the file name field, the 'Salva' button, and the 'Annulla' button.



# Grafici in MATLAB – 5/13

## Colori e Stile di Indicatori e Linee

---

- Mediante la funzione **plot** è anche possibile personalizzare il grafico, per quanto riguarda
  - **Colore**
    - **Linea e Indicatori** (detti anche *marcatori* o *simboli*)
  - **Stile**
    - **Linea** (tratteggiata, etc)
    - **Indicatori** (cerchio, puntino, etc)
- **plot(x, y, personalizzazione)**

# Grafici in MATLAB – 5/13

## Colori e Stile di Indicatori e Linee

---

- `plot(x, y, personalizzazione)`
- **Esempio di personalizzazione**
  - `plot(x, y, 'r^:')`
  - La stringa di personalizzazione `r^:` è composta da tre parti
    - `r`
      - Indica il colore di linea e indicatori
      - **Colore rosso (red)** in questo esempio
    - `^`
      - Indica lo stile degli indicatori
      - **Triangolo** in questo esempio
    - `:`
      - Indica lo stile della linea
      - **Punteggiata** in questo esempio

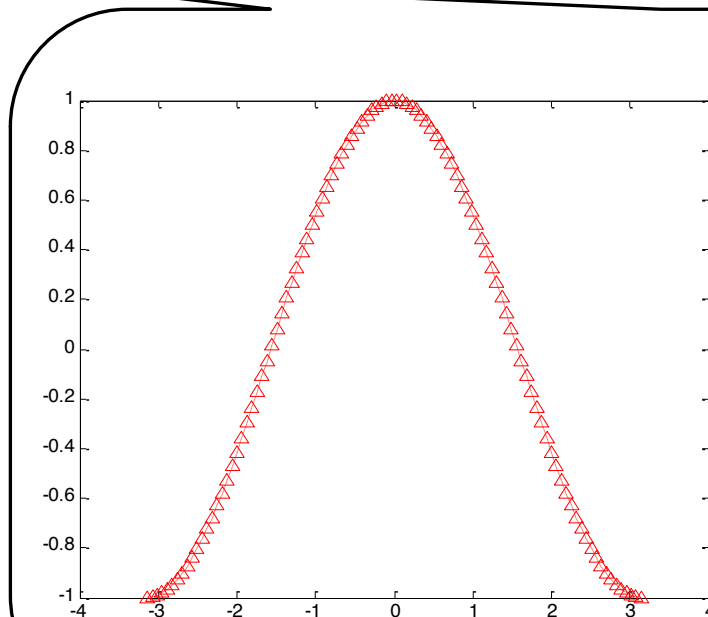
# Grafici in MATLAB – 5/13

## Colori e Stile di Indicatori e Linee

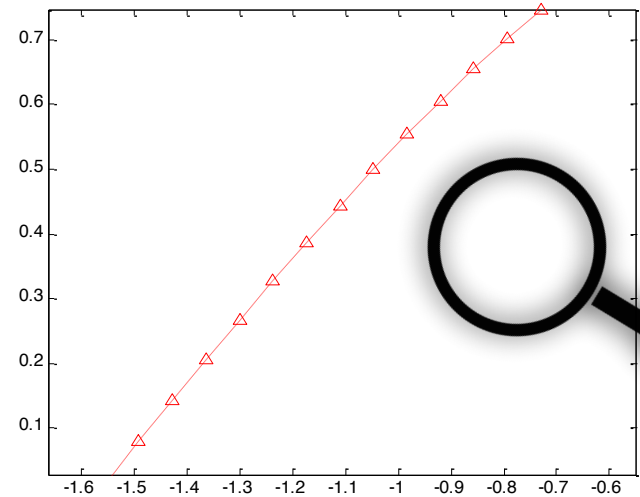
- `plot(x, y, personalizzazione)`

- *Esempio di personalizzazione*

- `plot(x, y, 'r^:')`



Risultato plot



Zoom Risultato plot

# Grafici in MATLAB – 6/13

## Colori e Stile di Indicatori e Linee

• `plot(x, y, '123')`

<b>1</b>	<b>Colori</b>	<b>2</b>	<b>Indicatori</b>	<b>3</b>	<b>Stile Linea</b>
<b>b</b>	blue	.	punto	-	continua
<b>g</b>	green (verde)	<b>o</b>	cerchio	:	punteggiata
<b>r</b>	red (rosso)	<b>x</b>	croce a x	-.	punto+tratteggio
<b>c</b>	cyan (ciano)	<b>+</b>	più	--	tratteggiata
<b>m</b>	magenta	<b>s</b>	square (quadrato)		nessuna linea
<b>y</b>	yellow (giallo)	<b>^</b>	triangolo		
<b>k</b>	black (nero)	<b>*</b>	stella		
<b>w</b>	white (bianco)	<b>h</b>	hexagon (esagono)		
	default	<b>p</b>	Pentagramma		
			nessun indicatore		

# Grafici in MATLAB – 6/13

## Colori e Stile di Indicatori e Linee

• `plot(x, y, '123')`

1	Colori	2	Indicatori	3	Stile Linea
<b>b</b>	blue	.	punto	-	continua
<b>g</b>	green (verde)	o	cerchio	---	punteggiata
<b>r</b>	red (rosso)	x	croce	-.	punto+tratteggio
<b>c</b>	cyan (ciano)	+	più	---	tratteggiata
<b>m</b>	magenta	s	square (quadrato)		nessuna linea
<b>y</b>	yellow (giallo)	^	triangolo		
<b>k</b>	black (nero)	*	stella		
<b>w</b>	white (bianco)	h	hexagon (esagono)		
	default	p	Pentagramma		
			nessun indicatore		

Non tutti i parametri sono obbligatori

# Grafici in MATLAB – 6/13

## Colori e Stile di Indicatori e Linee

• `plot(x, y, '123')`

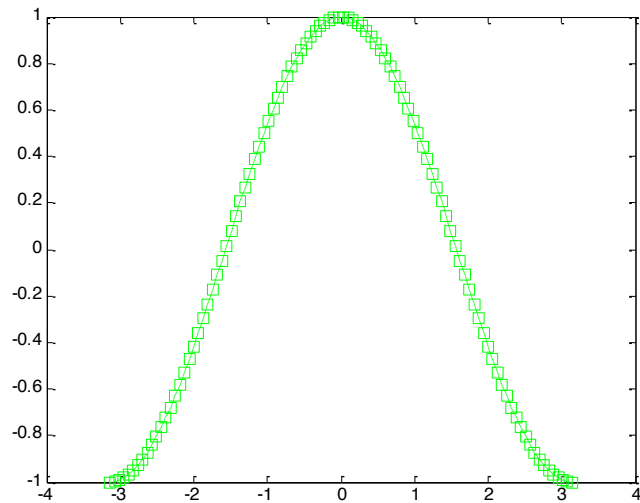
<b>1</b>	<b>Colori</b>	<b>2</b>	<b>Indicatori</b>	<b>3</b>	<b>Stile Linea</b>
<b>b</b>	blue	.	punto	-	continua
<b>g</b>	green (verde)	<b>o</b>	cerchio	:	punteggiata
<b>r</b>	red (rosso)	<b>x</b>	croce a x	-.	punto+tratteggio
<b>c</b>	cyan (ciano)	<b>+</b>	più	--	tratteggiata
<b>m</b>	magenta	<b>s</b>	square (quadrato)		nessuna linea
<b>y</b>	yellow (giallo)	<b>^</b>	triangolo		
<b>k</b>	black (nero)	<b>*</b>	stella		
<b>w</b>	white (bianco)	<b>h</b>	hexagon (esagono)		
	default	<b>p</b>	Pentagramma		
			nessun indicatore		

# Grafici in MATLAB – 7/13

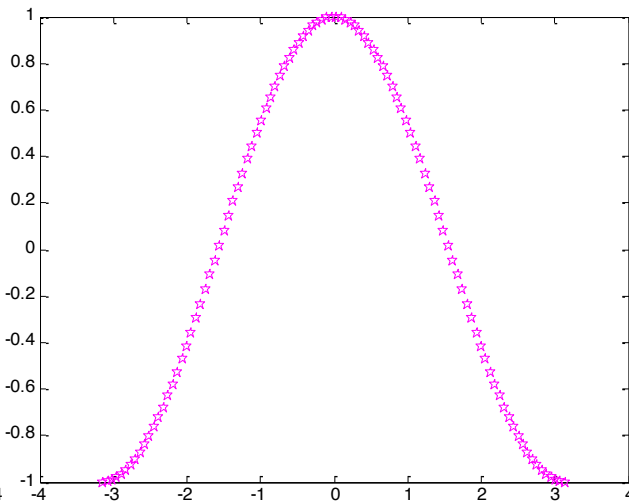
## Colori e Stile di Indicatori e Linee

---

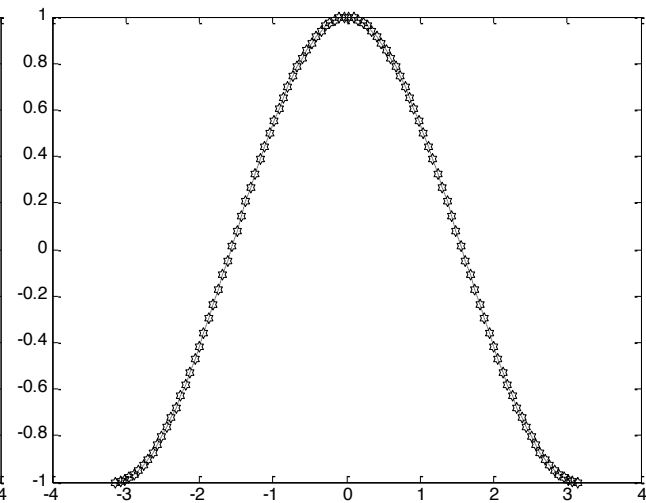
- **Esempi** (sono stati usati  $x$  ed  $y$  definiti nell'**Esempio 3**)
  1. `plot(x, y, 'gs--')`
  2. `plot(x, y, 'mp')`
  3. `plot(x, y, 'kh:')`



1



2



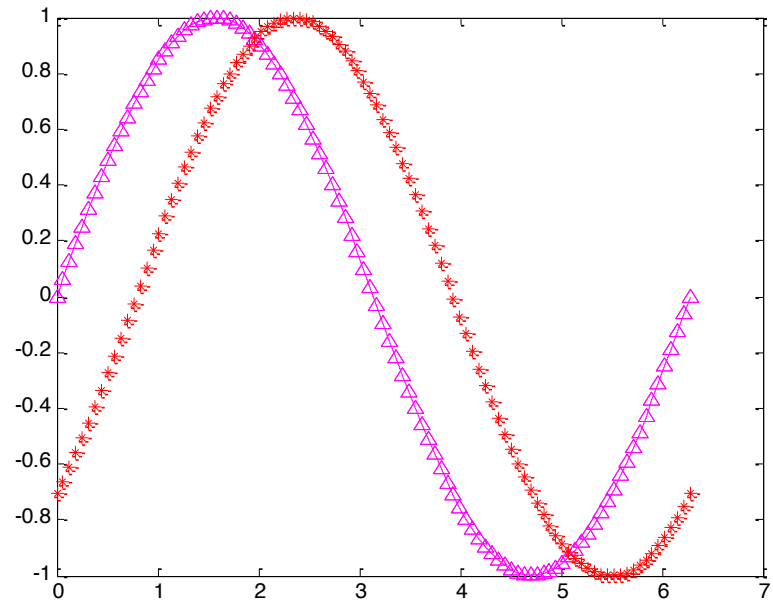
3

# Grafici in MATLAB – 8/13

## Grafico con Informazioni Multiple

- Con il comando `hold on` è possibile creare un **grafico che rappresenti più informazioni**, mediante **più linee e/o indicatori**

```
x = linspace(0, 2*pi, 100);  
y1 = sin(x);  
y2 = sin(x - pi / 4);  
hold on;  
plot(x, y1, 'm^--');  
plot(x, y2, 'r*');  
hold off;
```





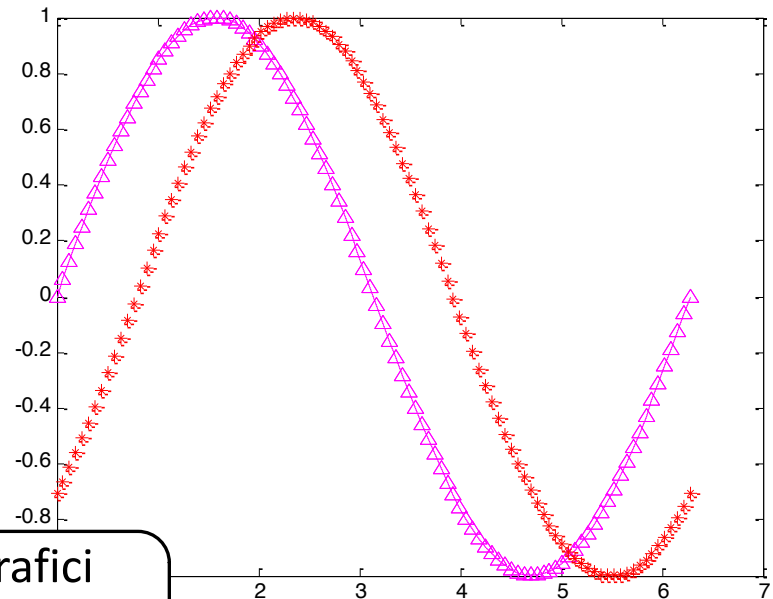
# Grafici in MATLAB – 8/13

## Grafico con Informazioni Multiple

- Con il comando **hold on** è possibile creare un **grafico che rappresenti più informazioni**, mediante **più linee e/o indicatori**

```
x = linspace(0, 2*pi, 100);  
y1 = sin(x);  
y2 = sin(x - pi / 4);  
hold on;  
plot(x, y1, 'm^--');  
plot(x, y2, 'r*');  
hold off;
```

Da ora in poi tutti i grafici saranno sovrapposti al grafico di partenza



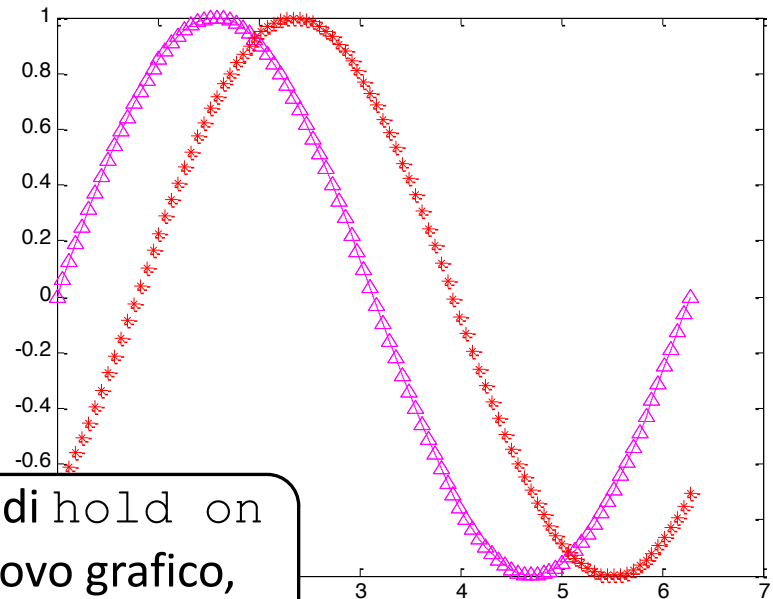
# Grafici in MATLAB – 8/13

## Grafico con Informazioni Multiple

- Con il comando **hold on** è possibile creare un **grafico che rappresenti più informazioni**, mediante **più linee e/o indicatori**

```
x = linspace(0, 2*pi, 100);  
y1 = sin(x);  
y2 = sin(x - pi / 4);  
hold on;  
plot(x, y1, 'm^--');  
plot(x, y2, 'r*');  
hold off;
```

Annulla l'effetto di **hold on** e genera un nuovo grafico, cancellando quelli vecchi



# Grafici in MATLAB – 8/13

## Alcune Note

---

- Ogni volta che si utilizza il comando **plot**, esso sovrascrive l'ultimo grafico che è stato creato
  - A meno che non venga utilizzato il comando **hold on**
- È possibile creare/gestire più istanze di grafici contemporaneamente, senza sovrascriverne nessuna, mediante il comando **figure**
  - `help figure` per maggiori informazioni
- Per chiudere tutte le istanze, si può usare il comando **close all**

# Grafici in MATLAB – 9/13

## Personalizzazione Titoli ed Etichette Assi

- MATLAB permette di aggiungere varie etichette al grafico

- **Esempio**

- `title('titolo')`

- Aggiunge/Modifica l'etichetta relativa al titolo del grafico

- `xlabel('asse x')`

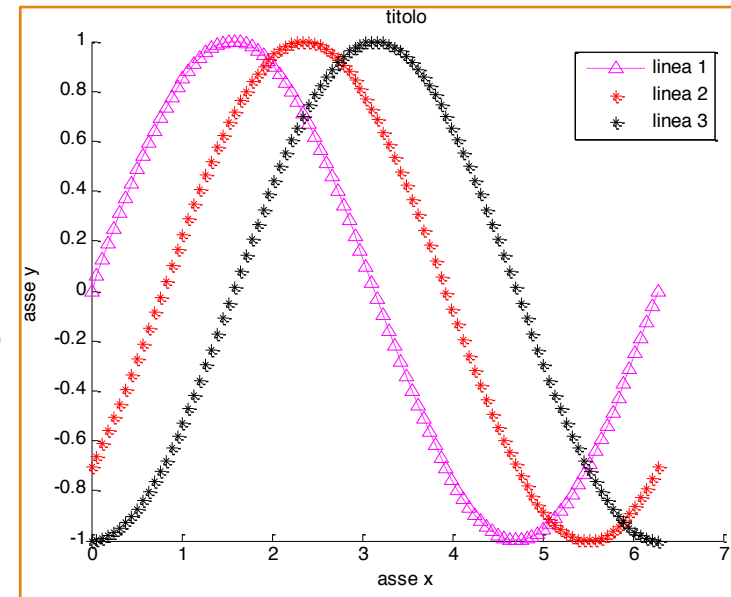
- Aggiunge/Modifica l'etichetta relativa all'asse X

- `ylabel('asse y')`

- Aggiunge/Modifica l'etichetta relativa all'asse Y

- `legend('linea 1', 'linea 2', 'linea 3')`

- Aggiunge la leggenda al grafico



# Grafici in MATLAB – 10/13

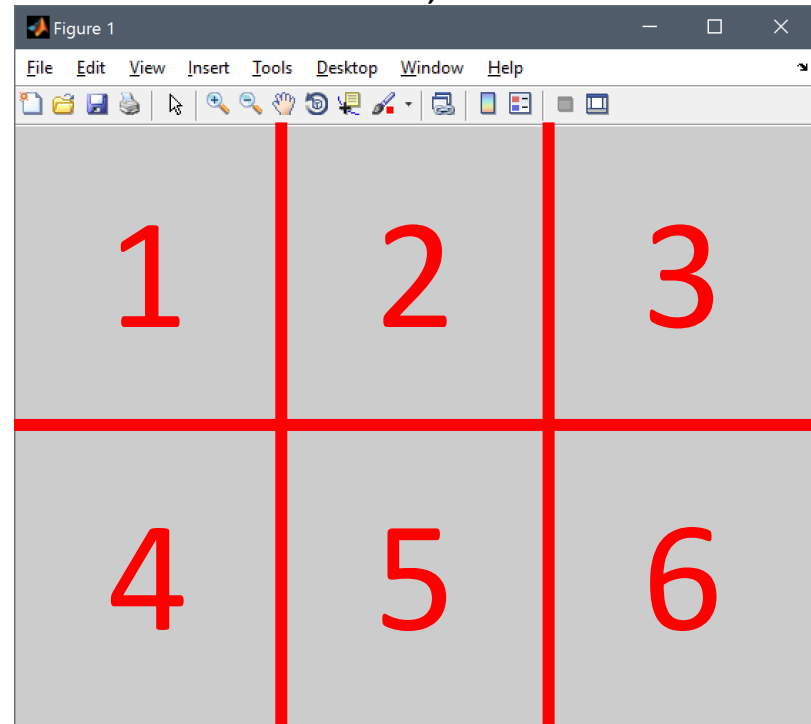
- MATLAB permette di creare più sotto-grafici all'interno di una singola figura, mediante il comando **subplot**

- **subplot**(*m*, *n*, *p*)

- Crea una figura partizionata come una **griglia**, inizialmente vuota, composta da *m* righe ed *n* colonne
- *p* indica una posizione all'interno della griglia (nell'esempio, *p* varia da 1 a 6)
- Una volta specificata con **subplot** la posizione all'interno della griglia, sarà possibile utilizzare la funzione **plot** (con gli opportuni parametri) per creare un sotto-grafico in tale posizione

## *Esempio*

*m* = 2, *n* = 3



# Grafici in MATLAB – 11/13

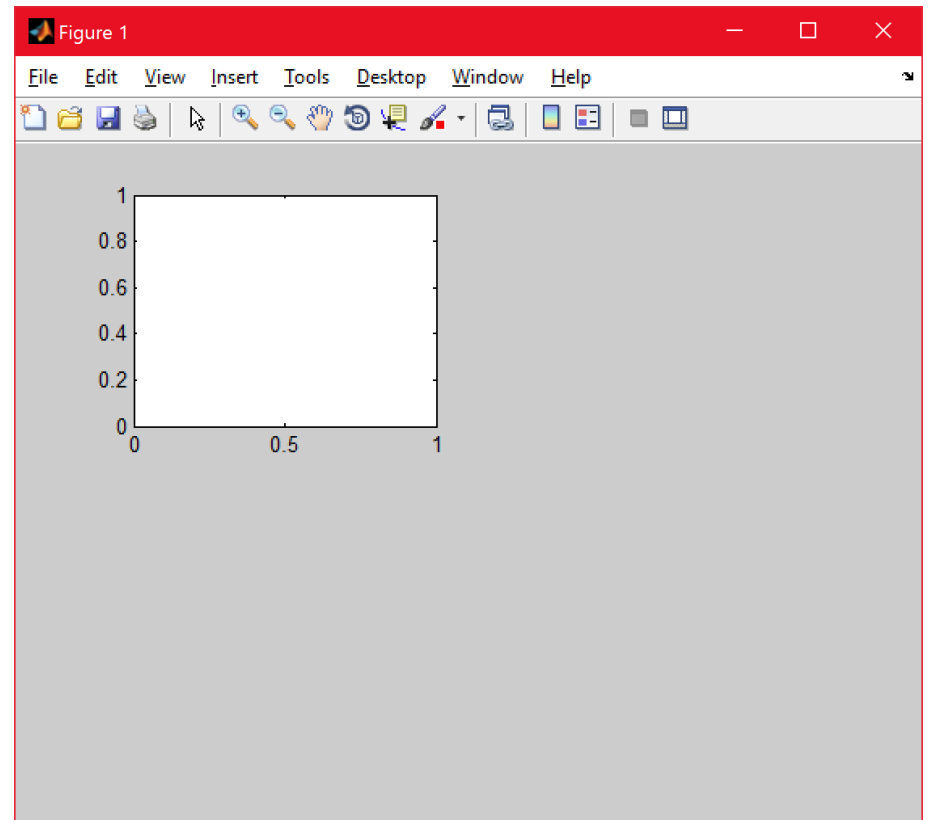
---

```
x = linspace(0,2*pi,100);  
y1 = sin(x);  
y2 = sin(x-pi/4);  
y3 = sin(x-pi/2);  
y4 = sin(x-2*pi/3);
```

# Grafici in MATLAB – 11/13

---

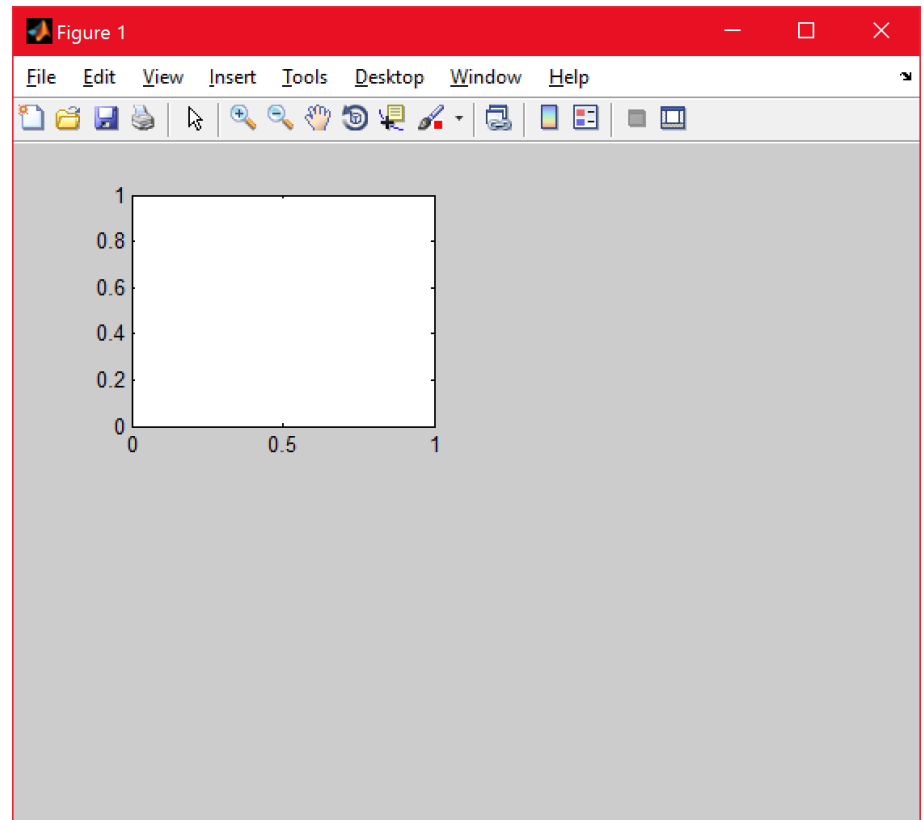
```
x = linspace(0,2*pi,100);  
y1 = sin(x);  
y2 = sin(x-pi/4);  
y3 = sin(x-pi/2);  
y4 = sin(x-2*pi/3);  
hold on;  
subplot(2,2,1);
```



# Grafici in MATLAB – 11/13

```
x = linspace(0,2*pi,100);  
y1 = sin(x);  
y2 = sin(x-pi/4);  
y3 = sin(x-pi/2);  
y4 = sin(x-2*pi/3);  
hold on;  
subplot(2,2,1);
```

Crea una griglia di **2** righe e **2** colonne,  
ed imposta la posizione corrente a **1**

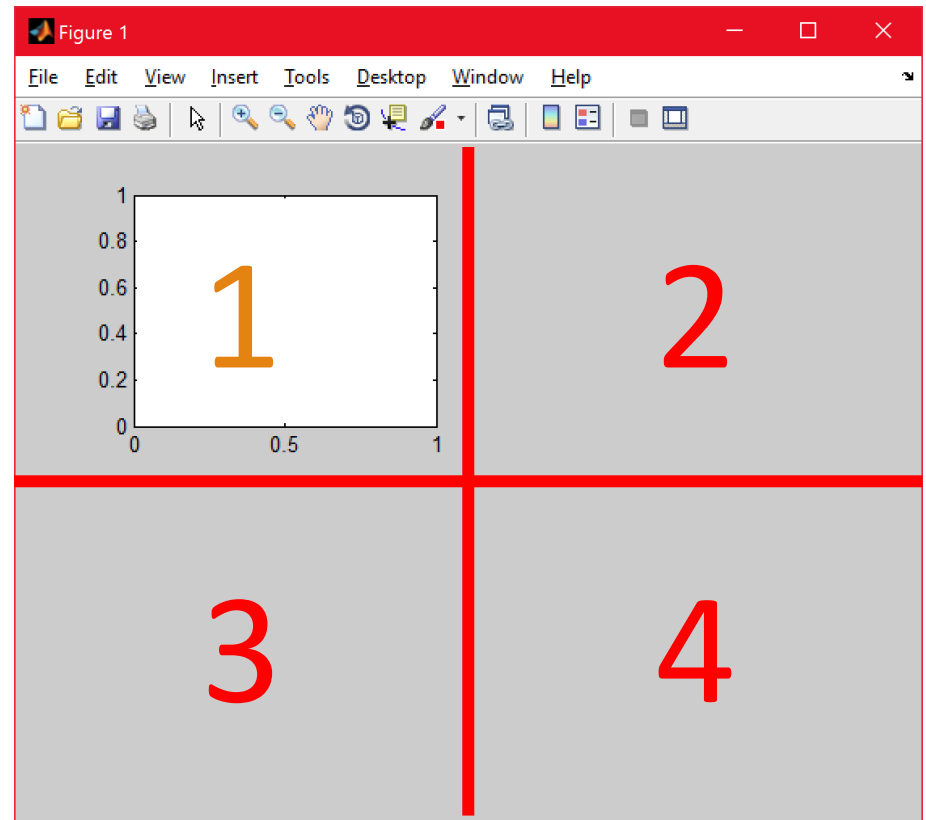




# Grafici in MATLAB – 11/13

```
x = linspace(0,2*pi,100);  
y1 = sin(x);  
y2 = sin(x-pi/4);  
y3 = sin(x-pi/2);  
y4 = sin(x-2*pi/3);  
hold on;  
subplot(2,2,1);
```

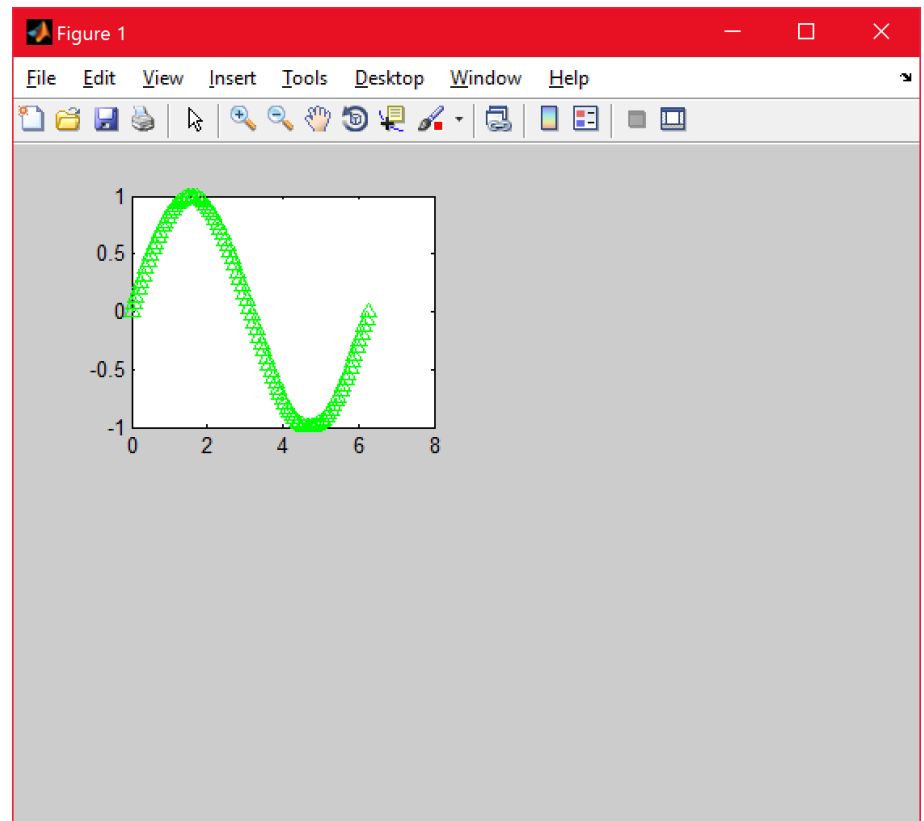
Crea una griglia di **2** righe e **2** colonne,  
ed imposta la posizione corrente a **1**



# Grafici in MATLAB – 11/13

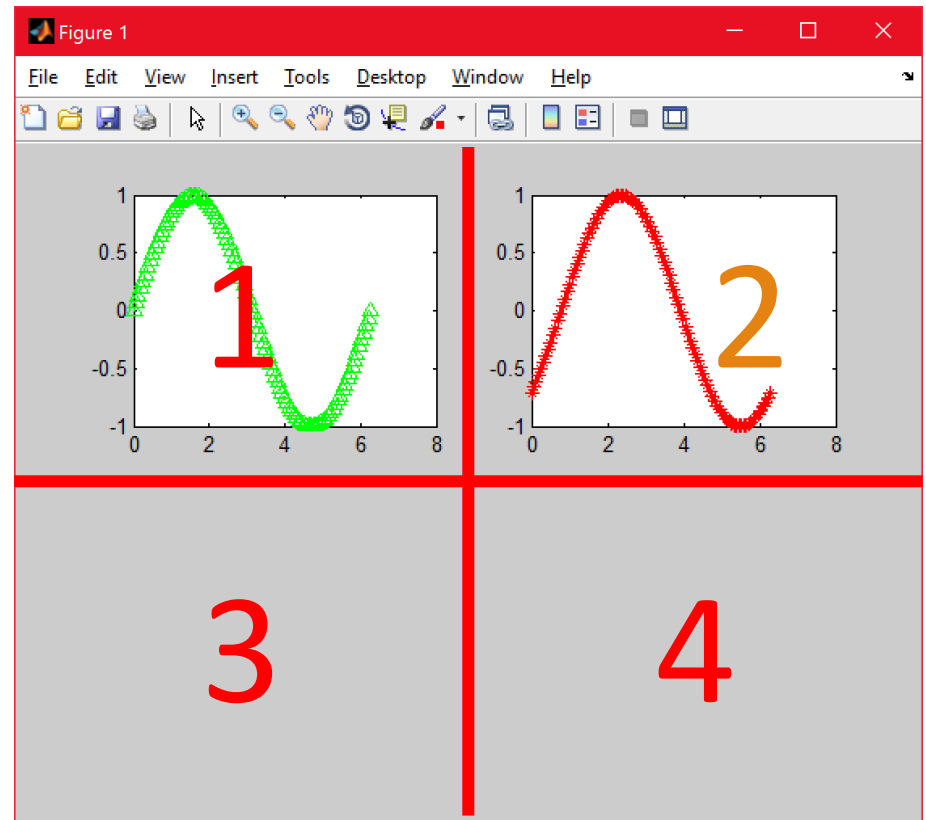
```
x = linspace(0,2*pi,100);  
y1 = sin(x);  
y2 = sin(x-pi/4);  
y3 = sin(x-pi/2);  
y4 = sin(x-2*pi/3);  
hold on;  
subplot(2,2,1);  
plot(x,y1,'g^:');
```

**Crea il primo grafico nella posizione 1**



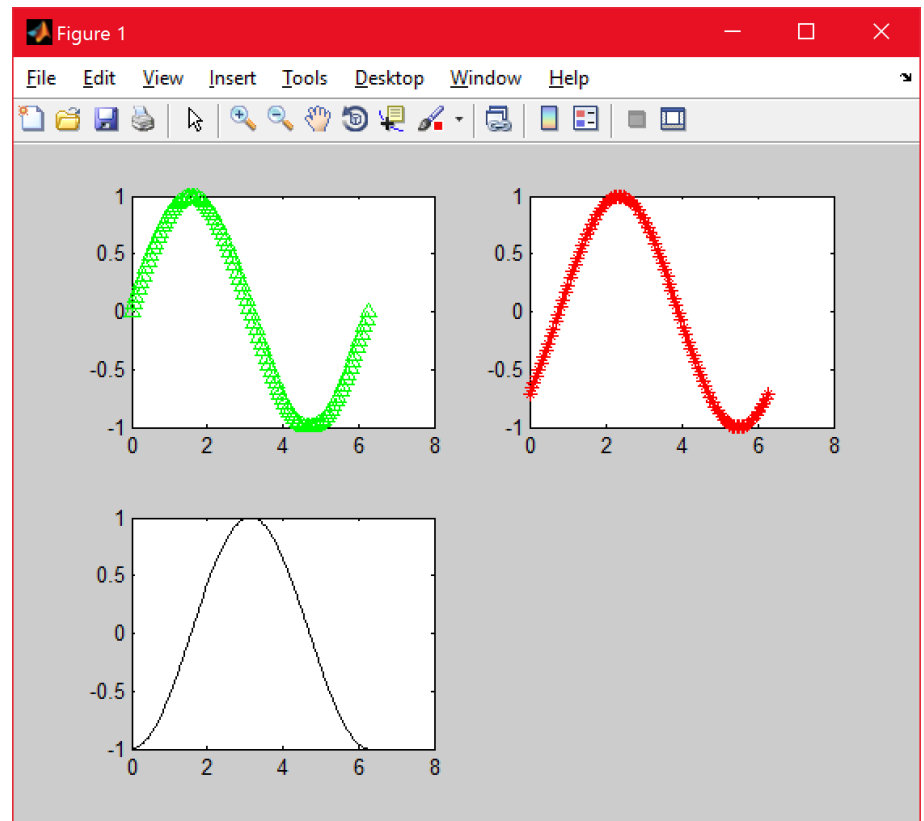
# Grafici in MATLAB – 11/13

```
x = linspace(0,2*pi,100);  
y1 = sin(x);  
y2 = sin(x-pi/4);  
y3 = sin(x-pi/2);  
y4 = sin(x-2*pi/3);  
hold on;  
subplot(2,2,1);  
plot(x,y1,'g^:');  
subplot(2,2,2);  
plot(x,y2,'r*--');
```



# Grafici in MATLAB – 11/13

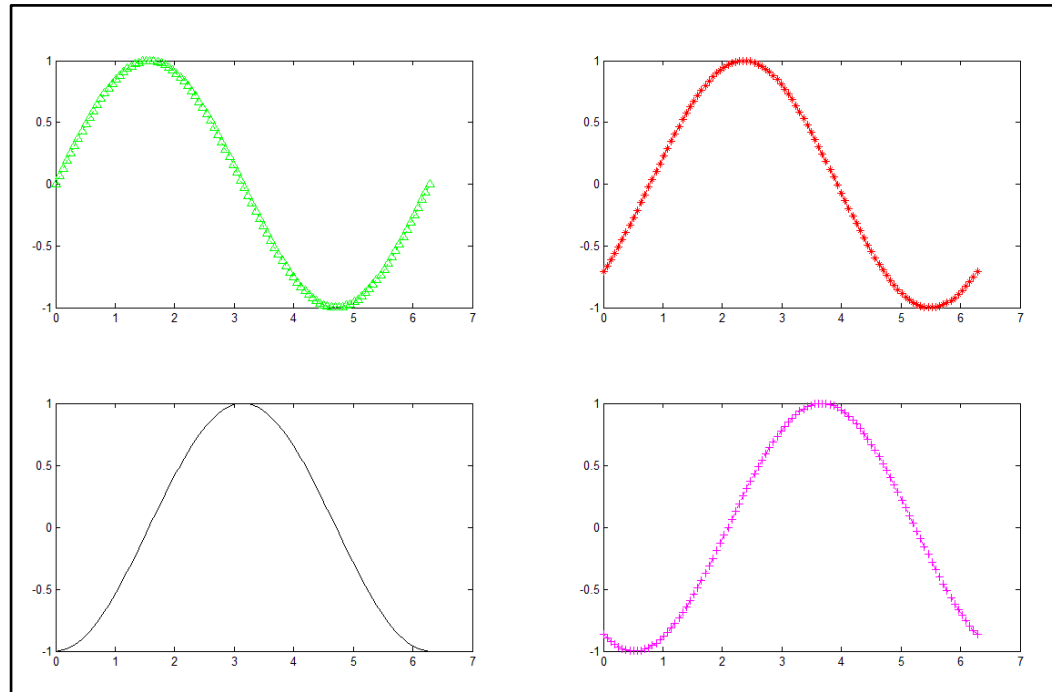
```
x = linspace(0,2*pi,100);  
y1 = sin(x);  
y2 = sin(x-pi/4);  
y3 = sin(x-pi/2);  
y4 = sin(x-2*pi/3);  
hold on;  
subplot(2,2,1);  
plot(x,y1,'g^:');  
subplot(2,2,2);  
plot(x,y2,'r*--');  
subplot(2,2,3);  
plot(x,y3,'k');
```



# Grafici in MATLAB – 11/13

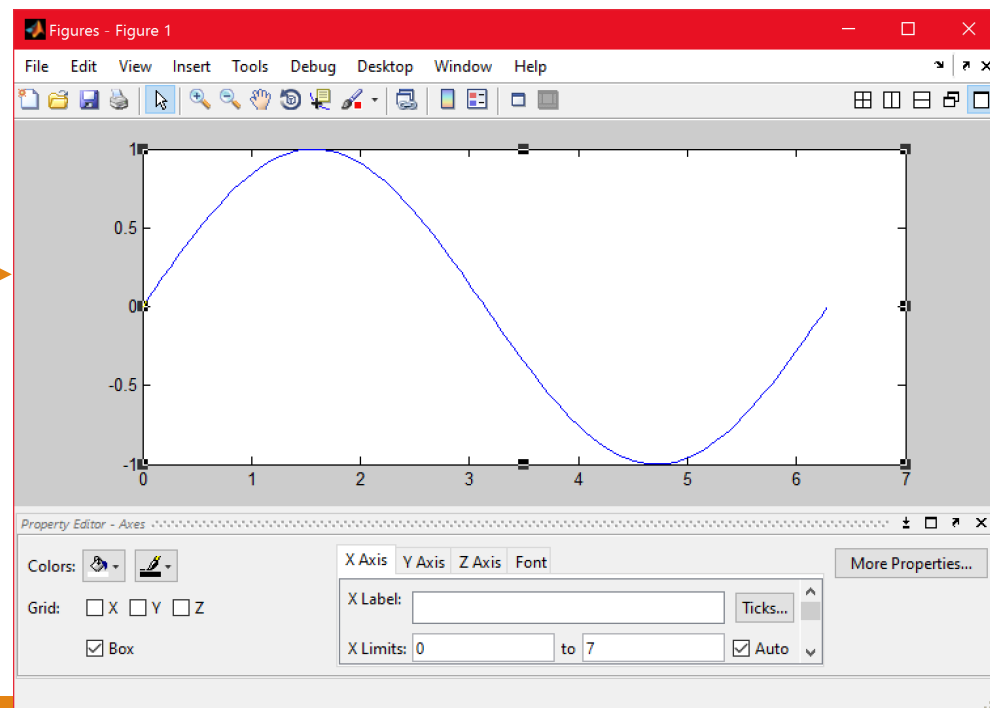
```
x = linspace(0,2*pi,100);  
y1 = sin(x);  
y2 = sin(x-pi/4);  
y3 = sin(x-pi/2);  
y4 = sin(x-2*pi/3);  
hold on;  
subplot(2,2,1);  
plot(x,y1,'g^:');  
subplot(2,2,2);  
plot(x,y2,'r*--');  
subplot(2,2,3);  
plot(x,y3,'k');  
subplot(2,2,4);  
plot(x,y4,'m+-');  
hold off;
```

**Output Finale:**



# Grafici in MATLAB – 12/13

- MATLAB fornisce anche un editor visuale per modificare le varie proprietà del grafico
  - Titolo
  - Etichette assi
  - Griglie
  - Font
  - Tipo di grafico
  - ...

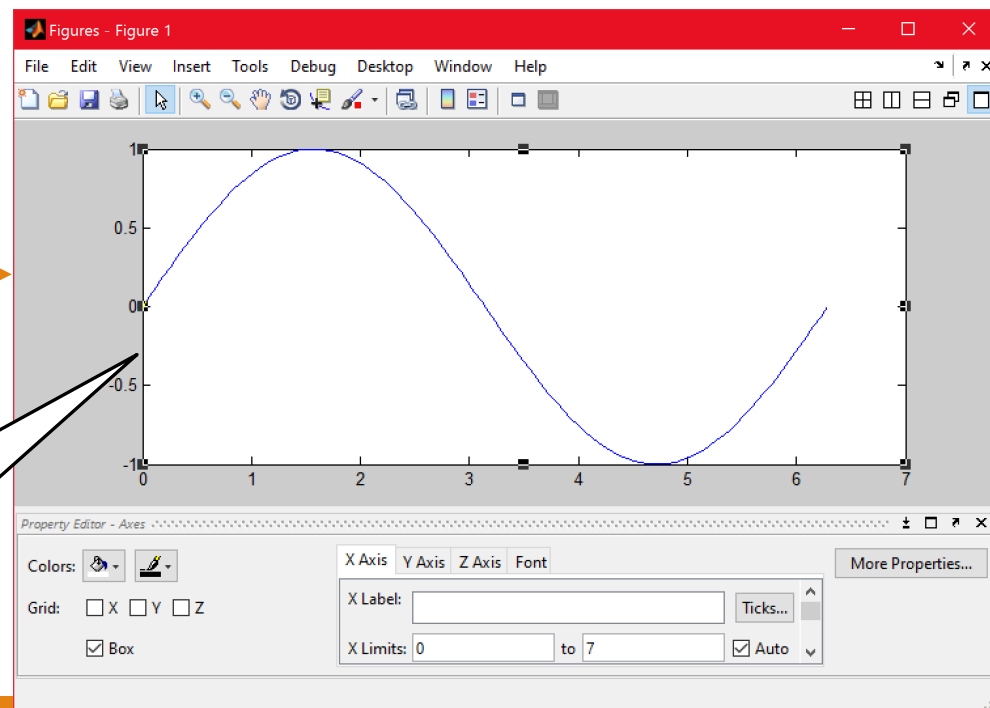


# Grafici in MATLAB – 12/13

- MATLAB fornisce anche un editor visuale per modificare le varie proprietà del grafico
  - Titolo
  - Etichette assi
  - Griglie
  - Font
  - Tipo di grafico
  - ...



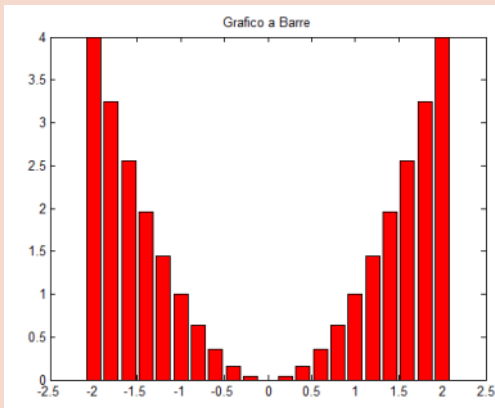
Cliccando due volte sul grafico si possono modificare le proprietà ad esso relative



# Grafici in MATLAB – 13/13

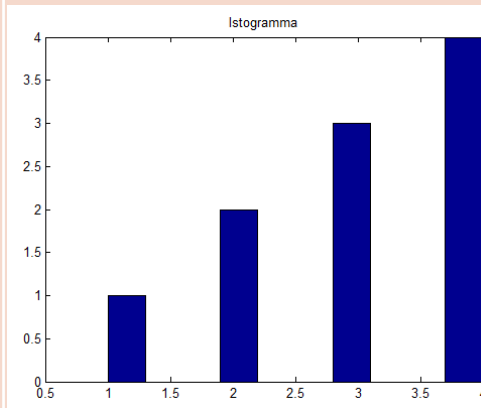
## Altre Tipologie di Grafici | Esempi

### Grafico a Barre



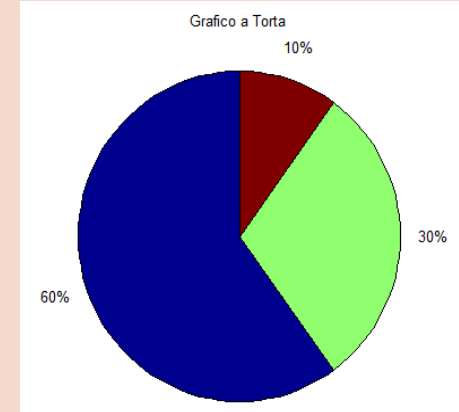
```
x = -2:0.2:2;  
y = x .* x;  
bar(x, y, 'r');  
title('Grafico a Barre');
```

### Istogramma



```
y = [ 1 2 2 3 3 3 4 4 4 4 ];  
hist(y);  
title('Istogramma');
```

### Grafico a Torta



```
x = [60 30 10];  
pie(x)  
title('Grafico a Torta');
```