

---

**PON 2007 – 2013**  
**Liceo Scientifico *Leonardo da Vinci***  
**Vallo della Lucania**  
**Nuovi percorsi matematici:**  
**Osservare, descrivere, costruire.**

---

Matlab - 5: Grafici di funzione

Arturo Stabile  
Vallo della Lucania  
13 Ottobre 2008

---

# Diagrammi xy – 1

- Un diagramma xy rappresenta una funzione del tipo  $y=f(x)$
- I valori di  $x$  sono rappresentati sull'asse orizzontale (ascisse) e i valori di  $y$  sono rappresentati sull'asse verticale (ordinate)
- Matlab ha molte funzioni e comandi che consentono di produrre vari tipi di diagrammi con caratteristiche speciali

---

# Diagrammi xy – 2

- Per visualizzare una qualsiasi funzione  $y=f(x)$  in Matlab, è “quasi” sempre necessario creare i vettori  $x$  ed  $y$  nel dominio di interesse
  - $x$  ed  $y$  devono avere la stessa dimensione
- È importante selezionare la *risoluzione*\* lungo l'asse  $x$  sufficientemente elevata

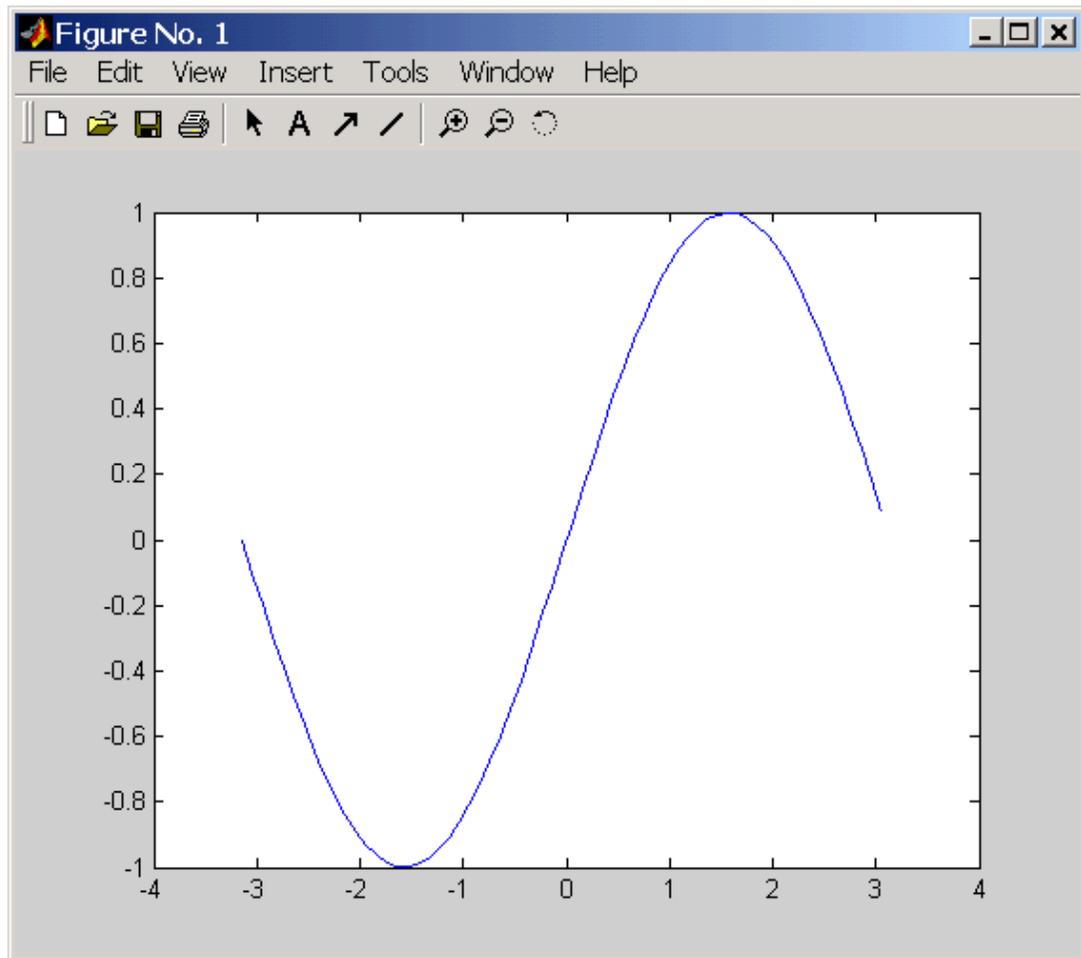
---

# I grafici in Matlab

- I grafici vengono visualizzati in finestre (*figure window*)
- Il comando di base è **plot**
  - `plot(y)`
    - Visualizza gli elementi del vettore `y` rispetto agli indici del vettore stesso
  - `plot(x,y)`
    - Visualizza gli elementi del vettore `y` rispetto agli elementi del vettore `x`

# Esempio

```
>> x=[-pi:0.1:pi];  
>> y=sin(x);  
>> plot(x,y)
```



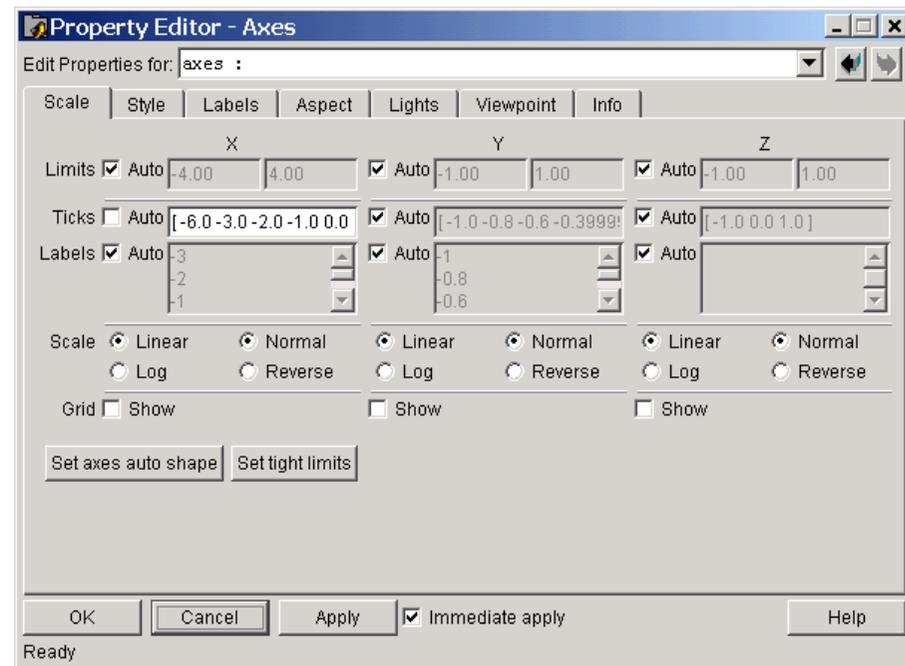
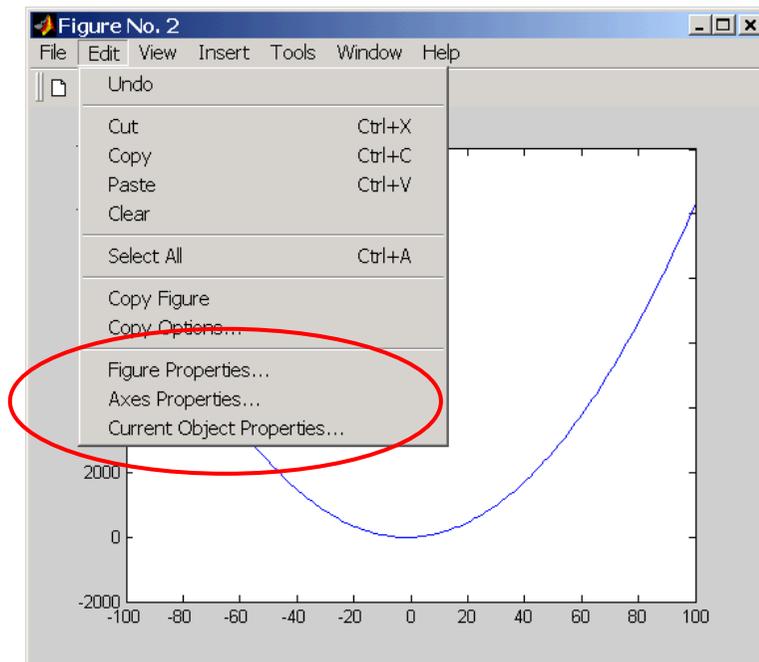
---

# Come creare un grafico in Matlab

1. Generare i dati
2. Scegliere la finestra dove mostrare il grafico
3. Tracciare il grafico
4. Fissare gli assi, la griglia, ...
5. Inserire le etichette degli assi, titolo, legenda
6. Stampare o esportare il grafico

# Modifica del grafico – 1

- È possibile modificare in modo interattivo l'aspetto dei grafici mediante il *Plot Editing Mode*, oppure mediante riga di comando



# Modifica del grafico – 2

- Attraverso il Plot Editing Mode possiamo modificare proprietà della figura, quali
  - Zoom
  - Aggiungere frecce e testo alle figure
  - Modificare le proprietà delle linee, degli assi



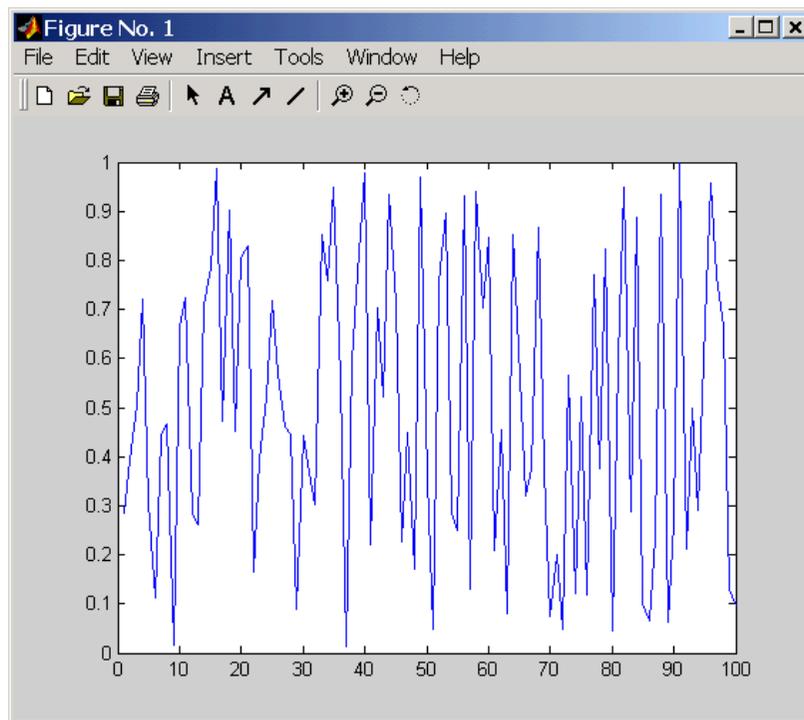
---

# Scelta della finestra

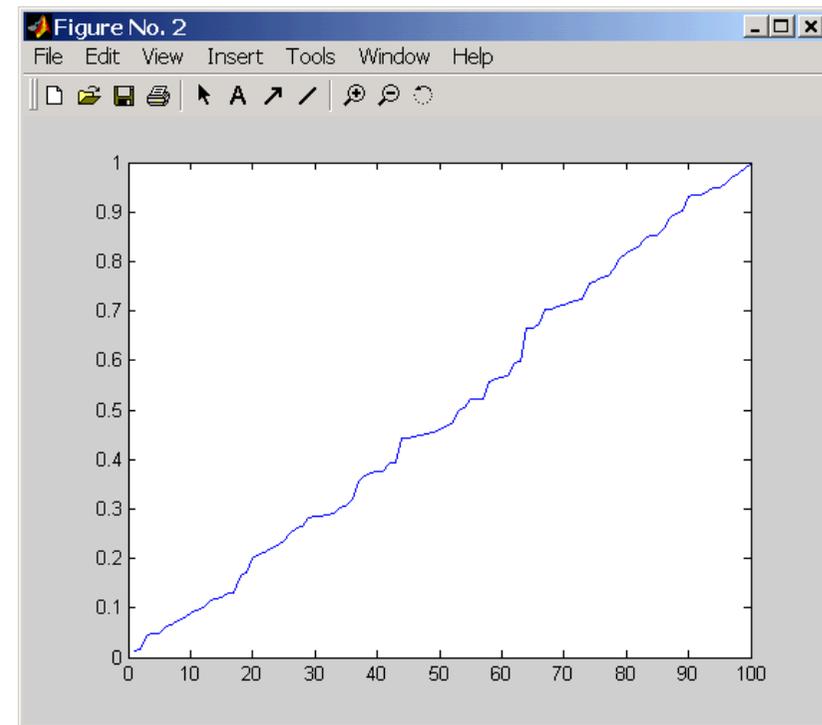
- Per scegliere la finestra in cui visualizzare il grafico si usa il comando `figure(NumeroFinestra)`
  - Se non lo si usa, la finestra utilizzata è sempre la numero uno
  - Se si eseguono successivamente, senza cambiare finestra, due `plot`, il secondo cancella il primo grafico
- Per cancellare il contenuto della finestra corrente si usa il comando `clf`
- Per cancellare un grafico si usa il comando `cla`

# Esempio

```
>> y=rand(1,100);  
>> plot(y)
```

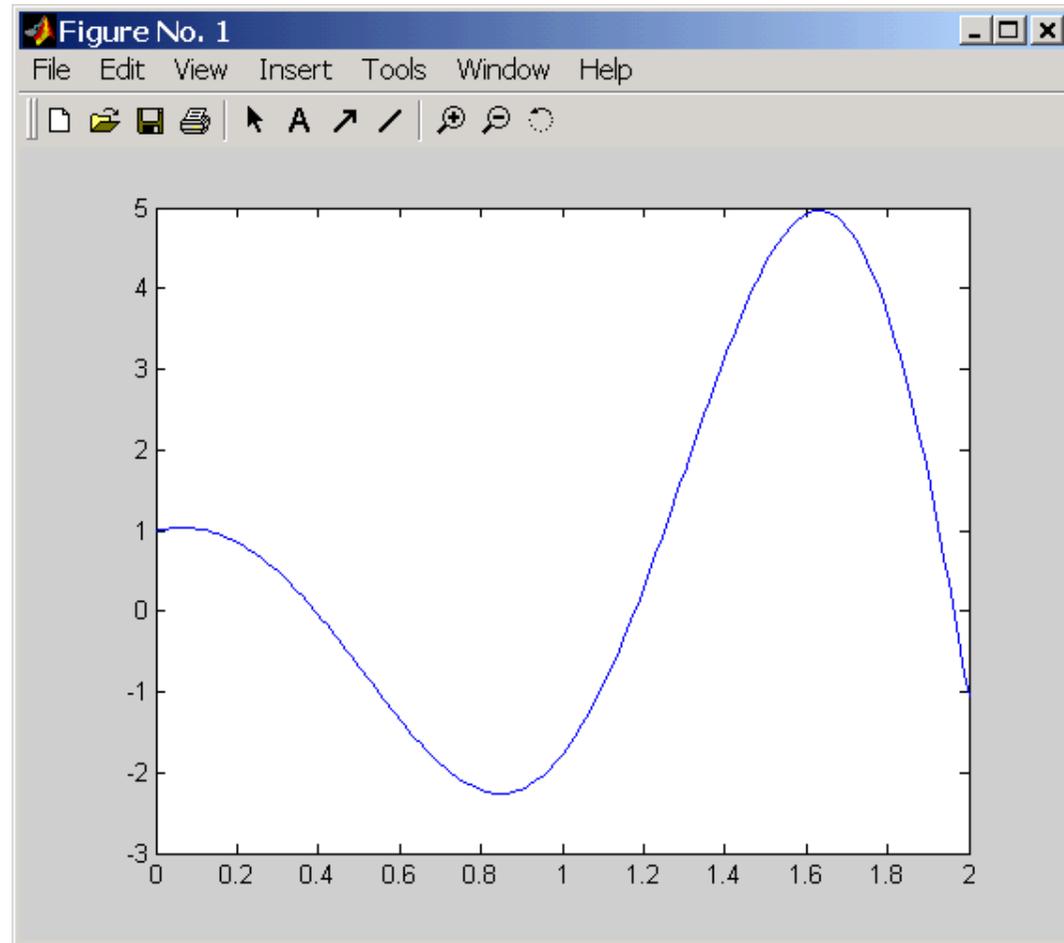


```
>> ys=sort(y);  
>> figure(2)  
>> plot(ys)
```



# Grafico di $\cos(4x)e^x$ su $[0,2]$

```
>> x=0:0.01:2;  
>> y=cos(4*x).*exp(x);  
>> plot(x,y)
```



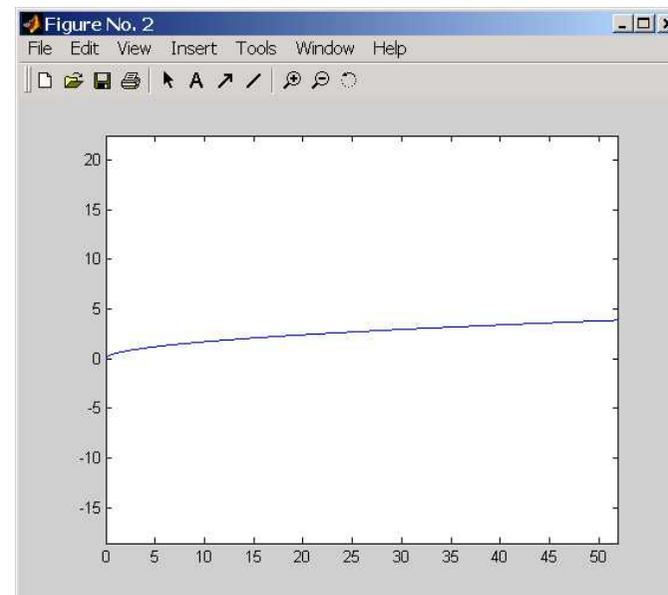
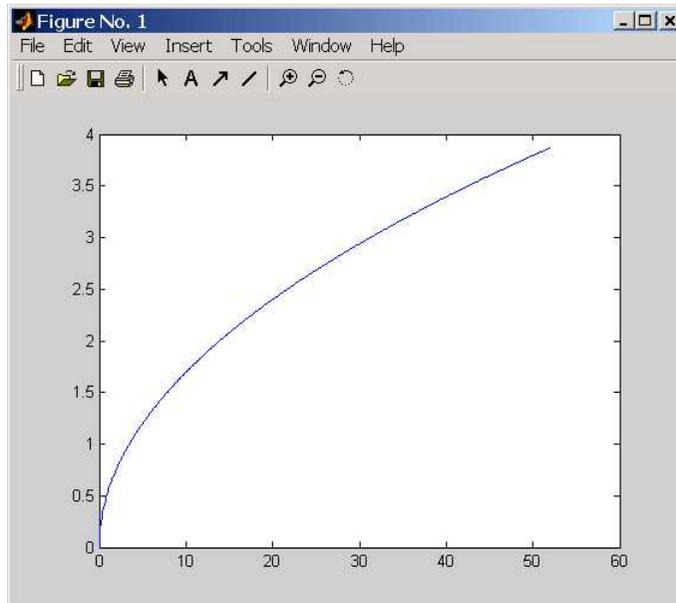
---

# Controllare gli assi xy – 1

- È possibile, tramite il comando **axis**, controllare la modalità con cui vengono rappresentati gli assi xy
- `axis([XMIN XMAX YMIN YMAX])`
  - Setta la scala degli assi xy per il grafico corrente
- `axis square`
  - Seleziona i fattori di scala in modo che il diagramma sia quadrato
- `axis auto`
  - Calcola i limiti degli assi in modo automatico

# Controllare gli assi xy – 2

- axis equal
  - Seleziona i fattori di scala e la spaziatura dei segni di graduazione in modo che siano uguali nei due assi x ed y



---

# Esercizio – 1

- Disegnare il grafico della funzione
$$f(x) = 4\sqrt{2.3x}$$
 per  $x \in [0, 60]$ 
  - Disegnare lo stesso grafico in finestre differenti
  - Controllare gli assi in maniera differente per ogni finestra

---

## Esercizio – 2

- Disegnare il grafico della funzione
$$\sin^2(x)\cos(x)+(\sin(e^x))^2+1$$
nell'intervallo  $[0,2\pi]$ 
  - Dividere l'intervallo in 10 punti equidistanti
  - Dividere l'intervallo in 100 punti equidistanti
  - Dividere l'intervallo in 1000 punti equidistanti

---

## Esercizio – 3

- Disegnare il grafico delle seguenti funzioni
  - $y = xe^x$  con  $x \in [-3,4]$
  - $y = x \sin(2x+\pi/3)$  con  $x \in [-2\pi, 2\pi]$
  - $y = \sin^2(x)$  con  $x \in [-2\pi, 2\pi]$
  - $y = x \sin(x)$  con  $x \in [-2\pi, 2\pi]$
  - $y = x^2 \sin(x)$  con  $x \in [-2\pi, 2\pi]$
  - $y = \sin(x)/x$  con  $x \in [-2\pi, 2\pi]$

## Esercizio – 4

- Disegnare il grafico delle funzioni:
  - $h(t) = v_0 \cdot t \cdot \text{sen}(A) - 0.5 \cdot g \cdot t^2$
  - $v(t) = v_0^2 - 2 \cdot v_0 \cdot g \cdot t \cdot \text{sen}(A) + g^2 \cdot t^2$ 
    - $A=30^\circ$ ,  $v_0=40\text{m/s}$ ,  $g=9.81\text{m/s}^2$
  - Considerare l'intervallo  $[0, 2(v_0/g)\text{sen}(A)]$ 
    - Trasformare A in radianti
      - $\text{deg} = \text{rad} \cdot 180/\pi$

---

# La griglia

- È possibile aggiungere una griglia al grafico in una finestra, si usa il comando **grid**
  - `grid on`
    - Aggiunge la griglia
  - `grid off`
    - Toglie la griglia
  - `grid minor`
    - Aggiunge/toglie linee dalla griglia

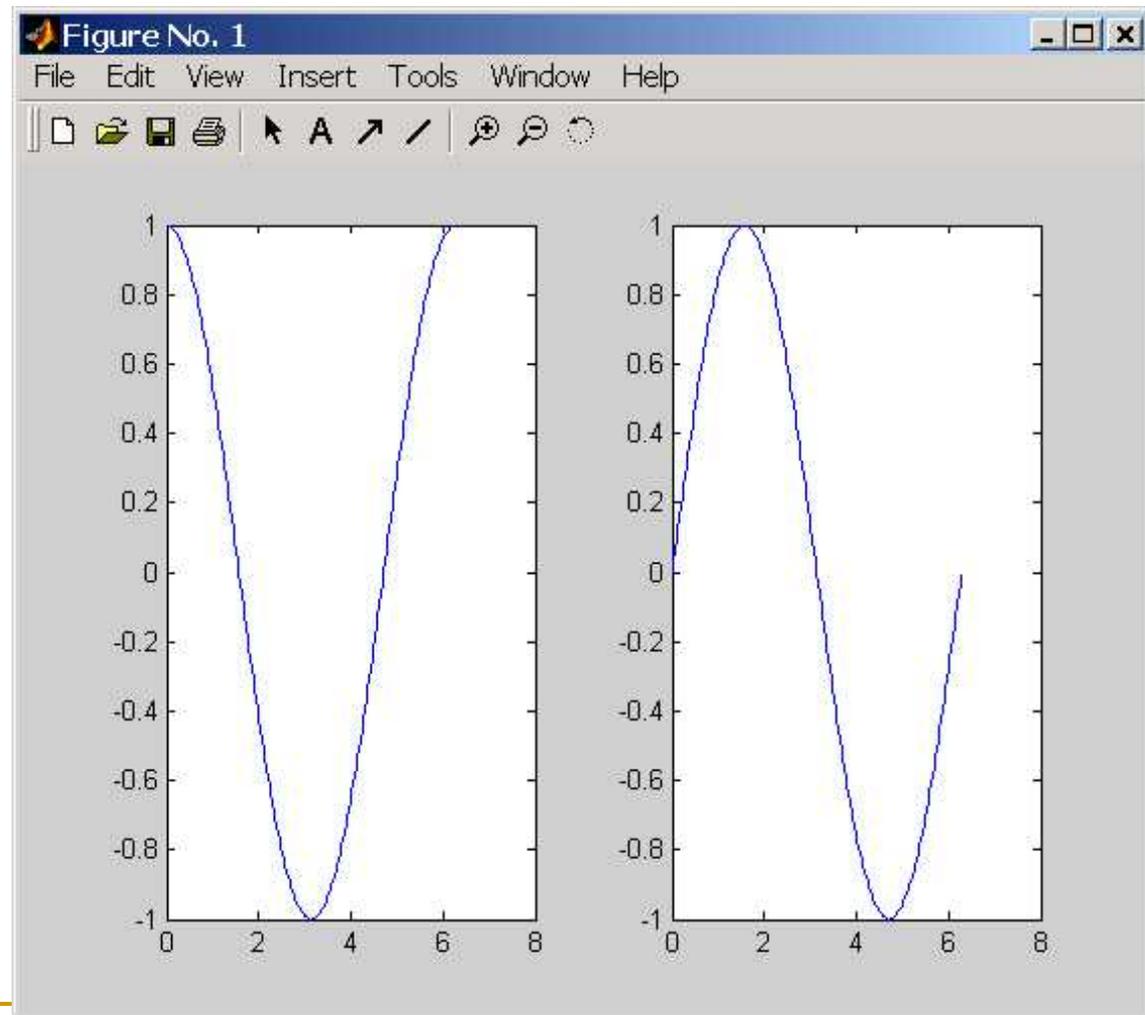
---

# Il comando `subplot`

- Serve a visualizzare contemporaneamente diversi grafici
- Il comando `subplot` suddivide la finestra dei grafici in più parti e seleziona la sottofinestra in cui far apparire il grafico
- Tramite il comando `subplot(n,m,p)` la finestra viene divisa in  $n \times m$  celle e la cella  $p$ -esima viene selezionata
  - Le celle sono “numerate” per righe

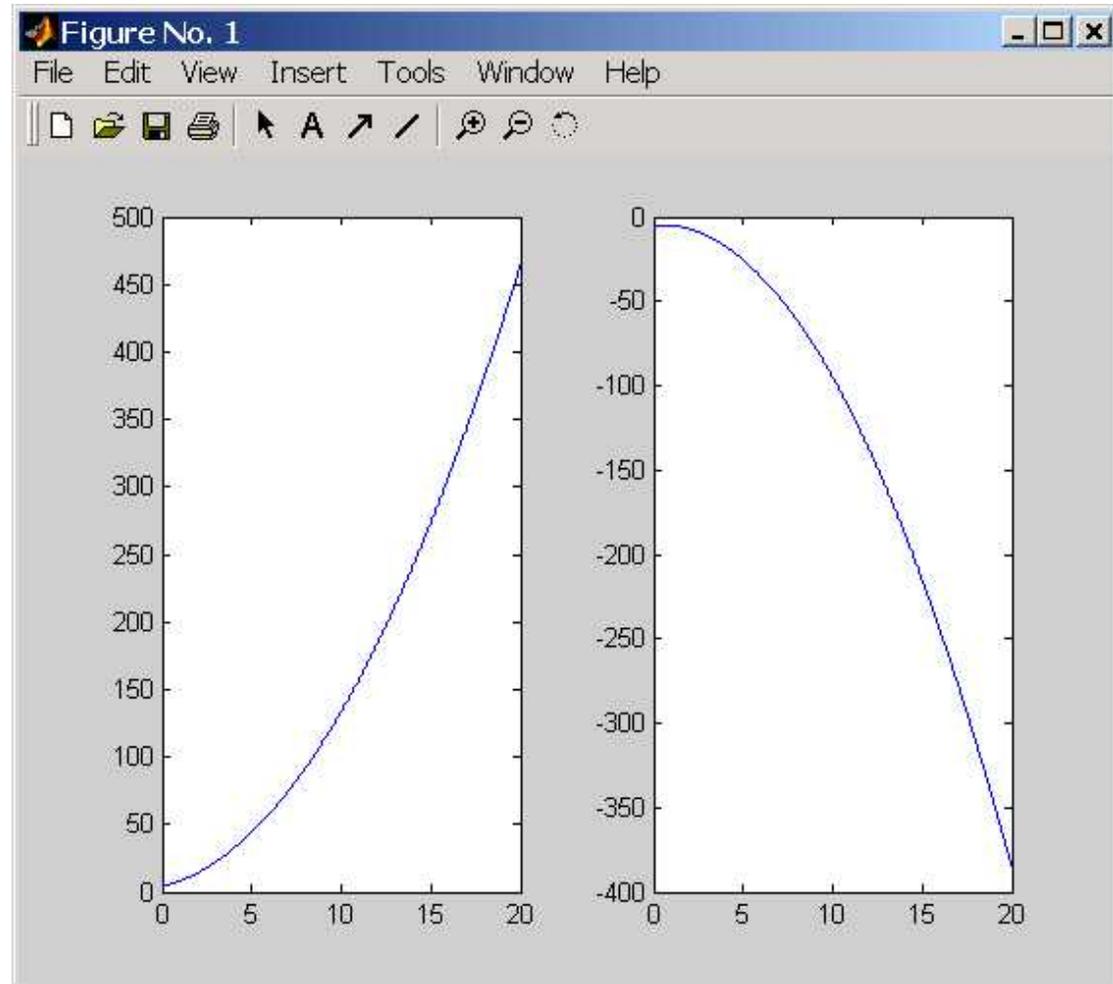
# Esempio – 1

```
>> t=[0:pi/100:2*pi];  
>> x=cos(t);  
>> y=sin(t);  
>> subplot(1,2,1);  
>> plot(t,x)  
>> subplot(1,2,2);  
>> plot(t,y)
```



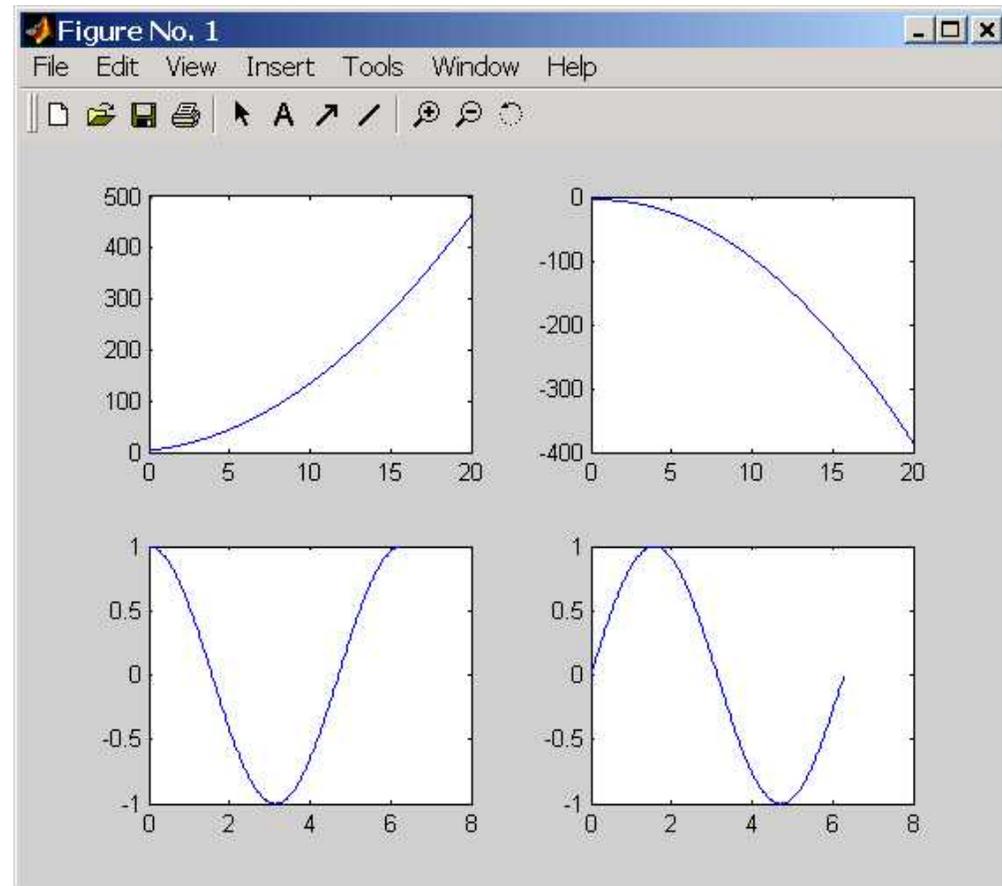
# Esempio – 2

```
>> x=[0:0.01:20];  
>> y1=x.^2+3*x+4;  
>> y2=-x.^2+x-5;  
>> subplot(1,2,1);  
>> plot(x,y1)  
>> subplot(1,2,2);  
>> plot(x,y2)
```



## Esempio – 3

```
>> t=[0:pi/100:2*pi];  
>> x=[0:0.01:20];  
>> y1=x.^2+3*x+4;  
>> y2=-x.^2+x-5;  
>> y3=cos(t);  
>> y4=sin(t);  
>> subplot(2,2,1);  
>> plot(x,y1)  
>> subplot(2,2,2);  
>> plot(x,y2)  
>> subplot(2,2,3);  
>> plot(t,y3)  
>> subplot(2,2,4);  
>> plot(t,y4)
```



---

# Esercizio

- Usare il comando **subplot** per creare nella stessa finestra grafica i diagrammi delle funzioni

$$z = e^{-0,5t} \cos(20t-6) \text{ per } 0 \leq t \leq 8$$

$$u = 6 \log_{10}(v^2+20) \text{ per } -8 \leq v \leq 8$$

---

# Aggiungere etichette

- Per aggiungere etichette agli assi cartesiani si usa i comandi **xlabel** ed **ylabel**, per gli assi x ed y, rispettivamente

`xlabel('etichettaX')` `ylabel('etichettaY')`

- Per aggiungere un titolo al grafico si usa il comando `title`

`title('titolo del grafico')`

- Le stringhe da mostrare come etichette o titolo vengono racchiuse tra apici ‘ ’

# Caratteri tipografici particolari

- Nelle etichette e nel titolo è possibile aggiungere
  - Lettere greche minuscole
  - Alcune lettere greche maiuscole
  - Alcuni caratteri speciali
- Si usa la stessa sintassi utilizzata per scrivere documenti con TeX/LaTeX
  - $x^{12}$  è reso come `x^{12}`
  - $\tau x_{23}$  è reso come `\tau x_{23}`

# Esempi

$\Gamma$	<code>\Gamma</code>	$\Lambda$	<code>\Lambda</code>	$\Sigma$	<code>\Sigma</code>	$\Psi$	<code>\Psi</code>
$\Delta$	<code>\Delta</code>	$\Xi$	<code>\Xi</code>	$\Upsilon$	<code>\Upsilon</code>	$\Omega$	<code>\Omega</code>
$\Theta$	<code>\Theta</code>	$\Pi$	<code>\Pi</code>	$\Phi$	<code>\Phi</code>		

$\alpha$	<code>\alpha</code>	$\theta$	<code>\theta</code>	$o$	<code>o</code>	$\tau$	<code>\tau</code>
$\beta$	<code>\beta</code>	$\vartheta$	<code>\vartheta</code>	$\pi$	<code>\pi</code>	$v$	<code>\upsilon</code>
$\gamma$	<code>\gamma</code>	$\iota$	<code>\iota</code>	$\varpi$	<code>\varpi</code>	$\phi$	<code>\phi</code>
$\delta$	<code>\delta</code>	$\kappa$	<code>\kappa</code>	$\rho$	<code>\rho</code>	$\varphi$	<code>\varphi</code>
$\epsilon$	<code>\epsilon</code>	$\lambda$	<code>\lambda</code>	$\varrho$	<code>\varrho</code>	$\chi$	<code>\chi</code>
$\varepsilon$	<code>\varepsilon</code>	$\mu$	<code>\mu</code>	$\sigma$	<code>\sigma</code>	$\psi$	<code>\psi</code>
$\zeta$	<code>\zeta</code>	$\nu$	<code>\nu</code>	$\varsigma$	<code>\varsigma</code>	$\omega$	<code>\omega</code>
$\eta$	<code>\eta</code>	$\xi$	<code>\xi</code>				

$\pm$	<code>\pm</code>	$\mp$	<code>\mp</code>	$\times$	<code>\times</code>	$\circ$	<code>\circ</code>
$\cap$	<code>\cap</code>	$\cup$	<code>\cup</code>	$\nabla$	<code>\nabla</code>	$\parallel$	<code>\parallel</code>
$\ell$	<code>\ell</code>	$\Re$	<code>\Re</code>	$\Im$	<code>\Im</code>	$\partial$	<code>\partial</code>
$\infty$	<code>\infty</code>	$\exists$	<code>\exists</code>	$\forall$	<code>\forall</code>		
$\leq$	<code>\leq</code>	$\subset$	<code>\subset</code>	$\subseteq$	<code>\subseteq</code>	$\geq$	<code>\geq</code>
$\ll$	<code>\ll</code>	$\supset$	<code>\supset</code>	$\supseteq$	<code>\supseteq</code>	$\gg$	<code>\gg</code>
$\in$	<code>\in</code>	$\equiv$	<code>\equiv</code>	$\sim$	<code>\sim</code>	$\approx$	<code>\approx</code>
$\neq$	<code>\neq</code>	$\propto$	<code>\propto</code>	$\div$	<code>\div</code>	$*$	<code>*</code>
$\rightarrow$	<code>\rightarrow</code>	$\mapsto$	<code>\mapsto</code>	$\Rightarrow$	<code>\Rightarrow</code>	$\Longrightarrow$	<code>\Longrightarrow</code>
$\sum$	<code>\sum</code>	$\int$	<code>\int</code>	$\prod$	<code>\prod</code>	$\oint$	<code>\oint</code>

---

# Aggiungere testo al grafico

- È possibile aggiungere del testo al grafico nella posizione che vogliamo
- Vengono usati i comandi **text** e **gtext**
  - Con **text** indichiamo la posizione dove il testo deve apparire
  - Con **gtext** usiamo il mouse per indicare la posizione dove far apparire il testo

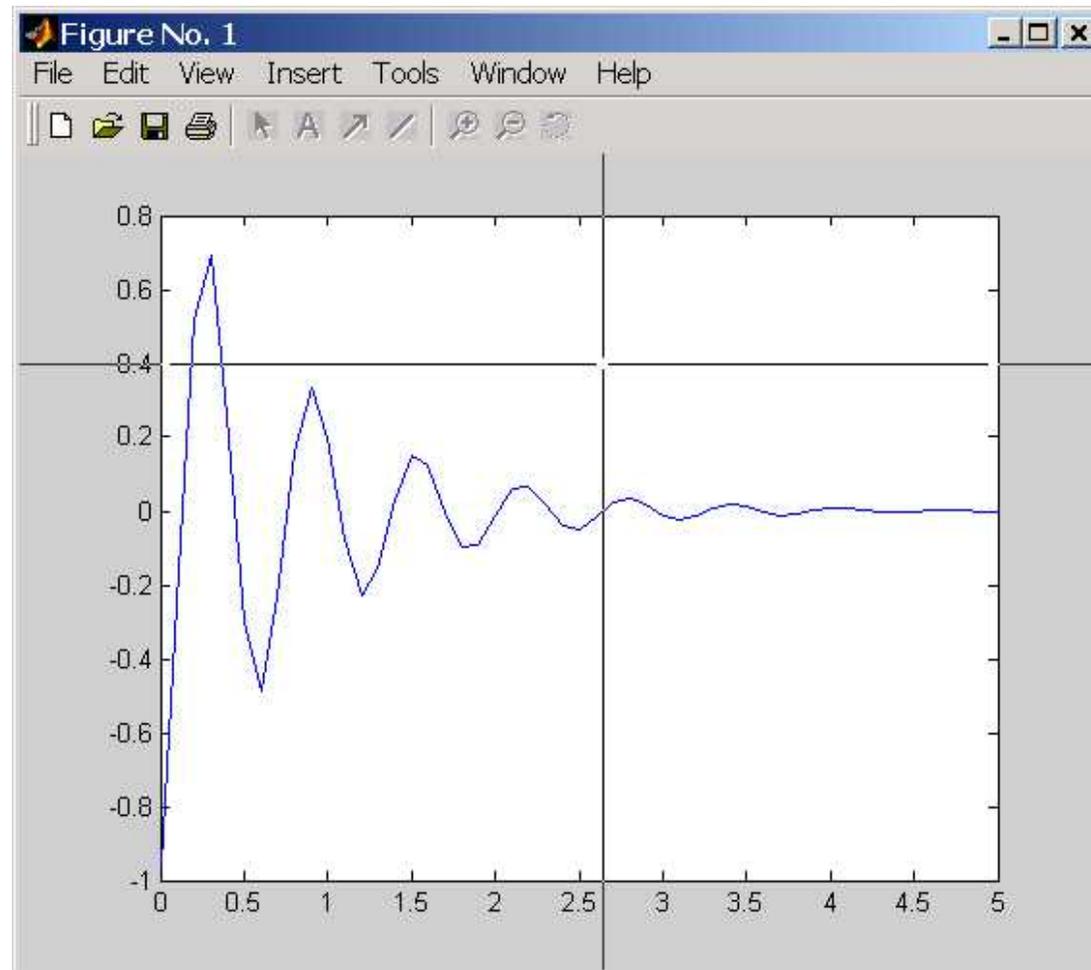
---

# Sintassi

- `text(x,y,'testo')`
  - Aggiunge la stringa `testo` al grafico posizionandola alle coordinate `x,y`
- `gtext('testo')`
  - Aggiunge la stringa `testo` al grafico posizionandola dove si clicca con il mouse
    - Sul grafico compaiono due assi indicanti il punto dove la stringa può essere posizionata

# Esempio

```
>> gtext('qui')
```

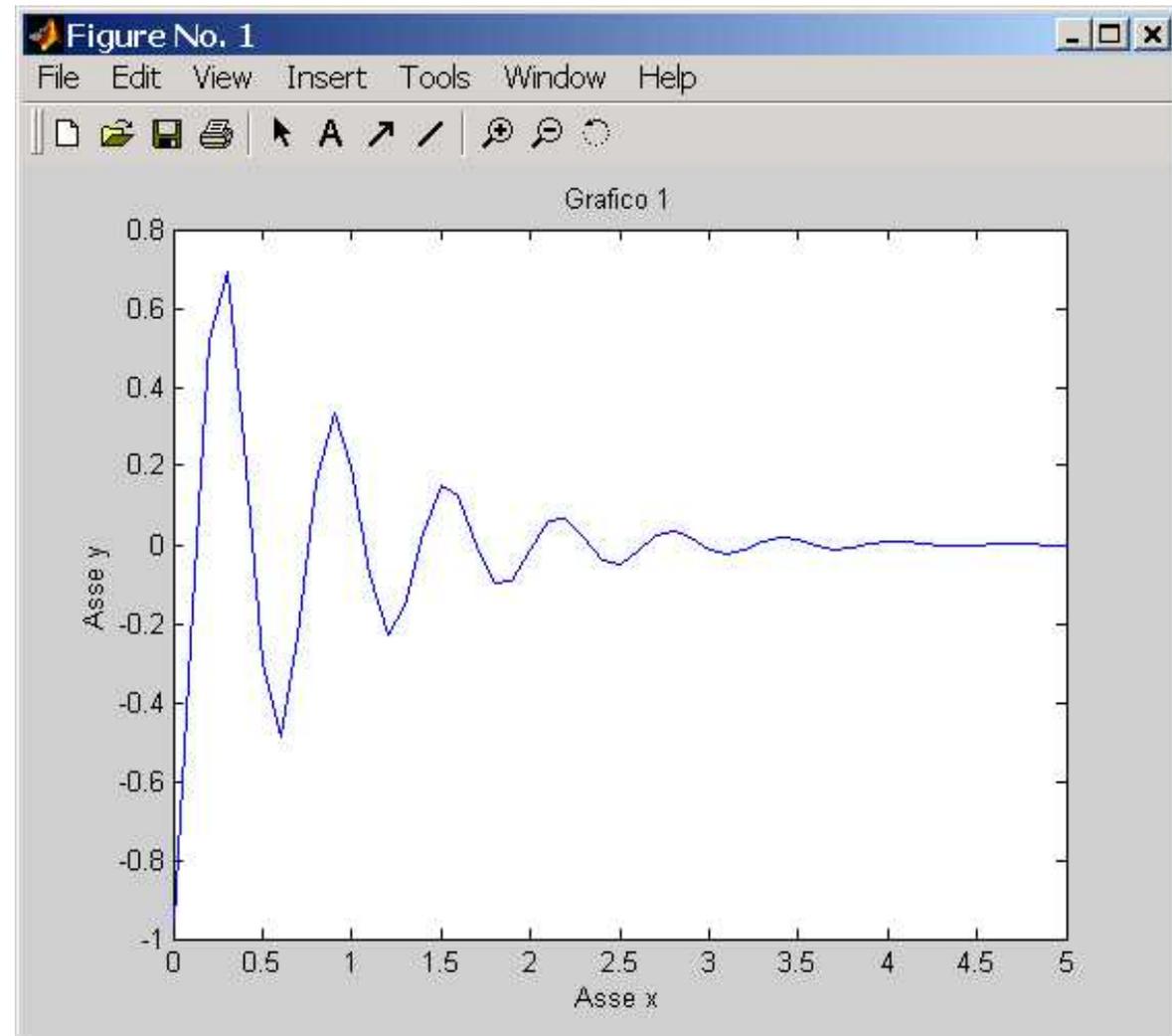


---

# Esercizio

- Disegnare la funzione  $e^{-1,2x}\sin(10x+5)$  nell'intervallo  $[0,5]$ 
  - Aggiungere il titolo “Grafico 1” al grafico
  - Aggiungere l'etichetta “asse x”
  - Aggiungere l'etichetta “asse y”

# Risultato



---

# Esercizio

- Disegnare in un'unica finestra i grafici delle quattro funzioni  $\sin$ ,  $\cos$ ,  $\tan$ ,  $\text{atan}$  nell'intervallo  $[0, 2\pi]$ 
  - Aggiungere titolo ed etichette ad ogni grafico

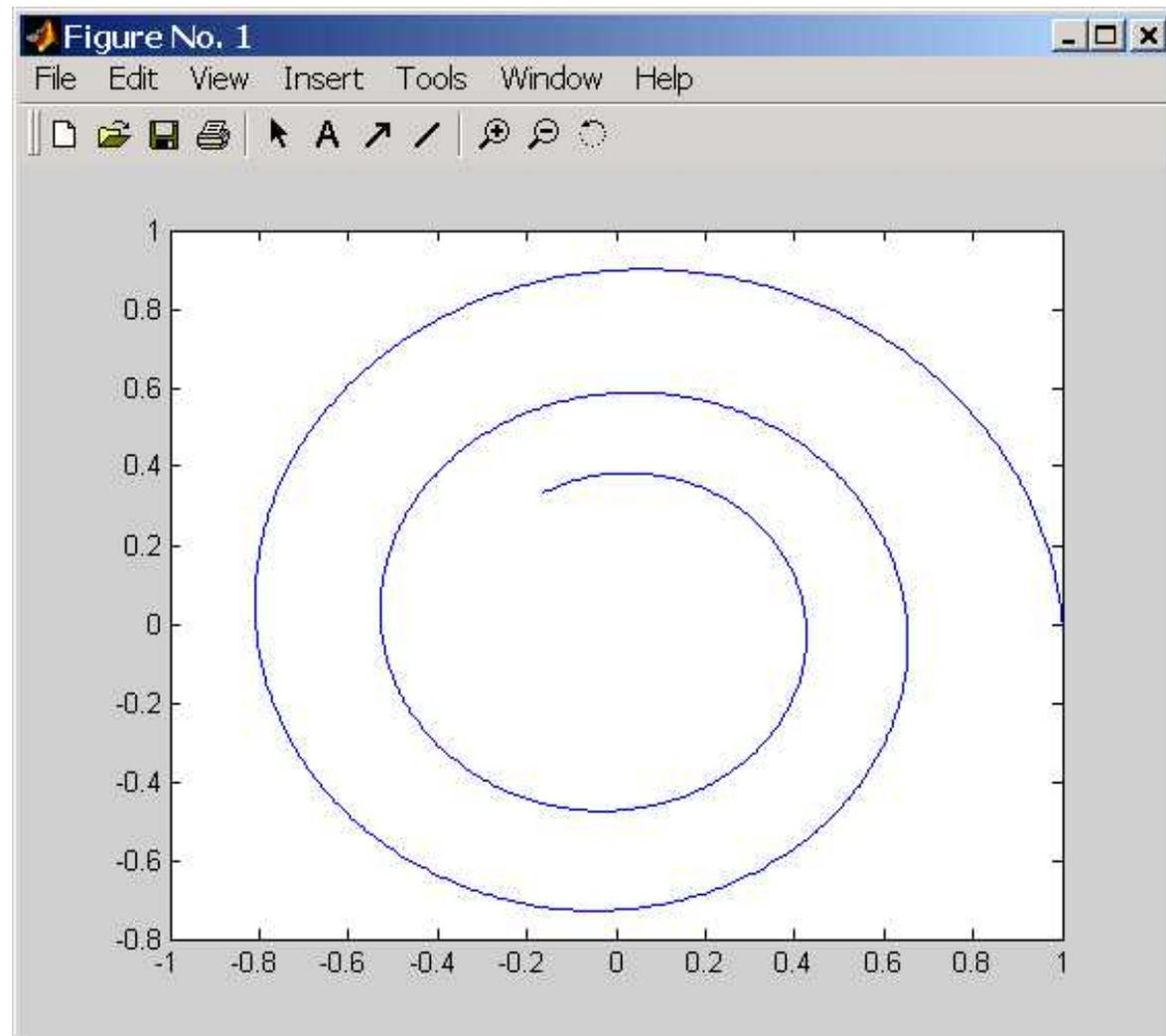
---

# Diagrammi di numeri complessi

- Se  $y$  è un vettore di numeri complessi allora `plot(y)` è equivalente a `plot(real(y),imag(y))`
- Questo è l'unico caso in cui la funzione `plot` gestisce le parti immaginarie, in tutte le altre varianti della funzione `plot` le parti immaginarie vengono ignorate

# Esempio

```
>> z=.1+.9i;  
>> n=0:.01:10;  
>> plot(z.^n)
```



---

# Esercizio

- Creare un diagramma per rappresentare la parte immaginaria di  $(0,2+0,8i)^n$  in funzione della parte reale per  $0 \leq n \leq 20$ .
  - Scegliere un numero sufficiente di punti per ottenere una curva regolare
  - Assegnare un titolo ai due assi e al diagramma
  - Aggiungere una griglia al grafico
  - Utilizzare il comando **axis** per cambiare la spaziatura dei segni di graduazione degli assi

---

# Il comando `fplot` – 1

- Esamina la funzione da rappresentare nel diagramma e stabilisce automaticamente il numero di punti da utilizzare
- La sintassi è `fplot(fun, limit)`
  - `fun` è una stringa che rappresenta una funzione
  - `limit` rappresenta l'intervallo da usare, può essere indicato secondo due modalità
    - `[xmin xmax]`
    - `[xmin xmax ymin ymax]`

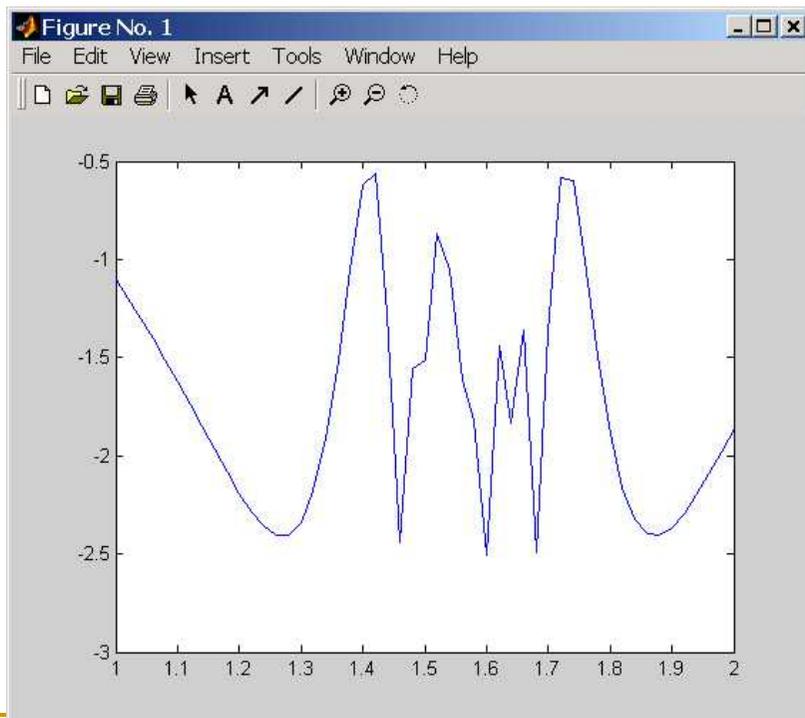
---

## Il comando `fplot` – 2

- Un'altra forma del comando `fplot` è  
 $[x,y] = \text{fplot}(\text{fun}, \text{limit})$
- Con questa forma il comando restituisce i valori delle ascisse e delle ordinate nei vettori `x` ed `y` senza generare alcun grafico

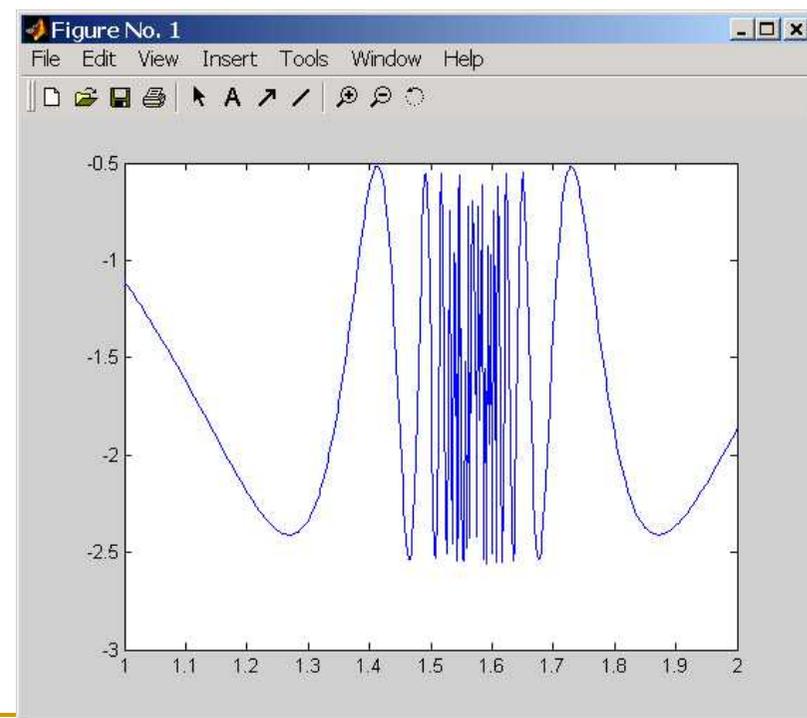
# Esempio

```
>> x=1:.02:2;  
>> y=cos(tan(x))-tan(sin(x));  
>> plot(x,y)
```



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

apice



---

# Esercizio

- Creare il grafico della funzione

$$y = .4\sqrt{2.3x} \text{ per } 0 \leq x \leq 35 \text{ e } 0 \leq y \leq 3,5$$

- Utilizzare il comando **fplot** per analizzare la funzione

$$\tan(\cos x) - \sin(\tan x) \text{ per } 0 \leq x \leq 2\pi$$

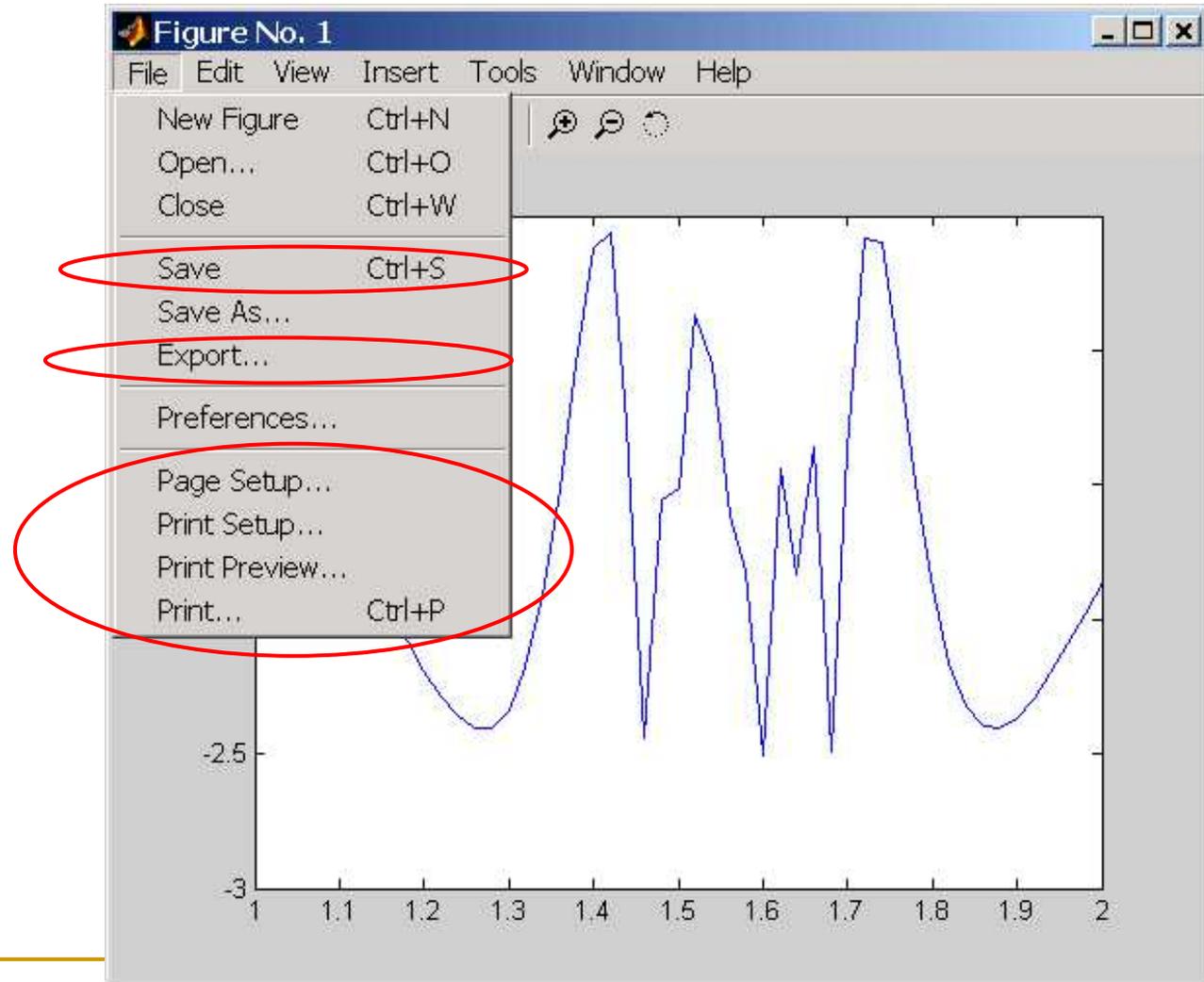
- Quanti valori sono richiesti per ottenere lo stesso diagramma utilizzando la funzione plot?

---

# Esportare un grafico

- È possibile stampare un grafico
  - Menu file -> Page Setup, Print Preview, ...
- È possibile salvarla come figura Matlab
  - Menu file -> Save, Save As
- Esportarla in formato grafico
  - Menu file -> Export
    - Formati bmp, jpg, tiff, emf, ...

# Graficamente



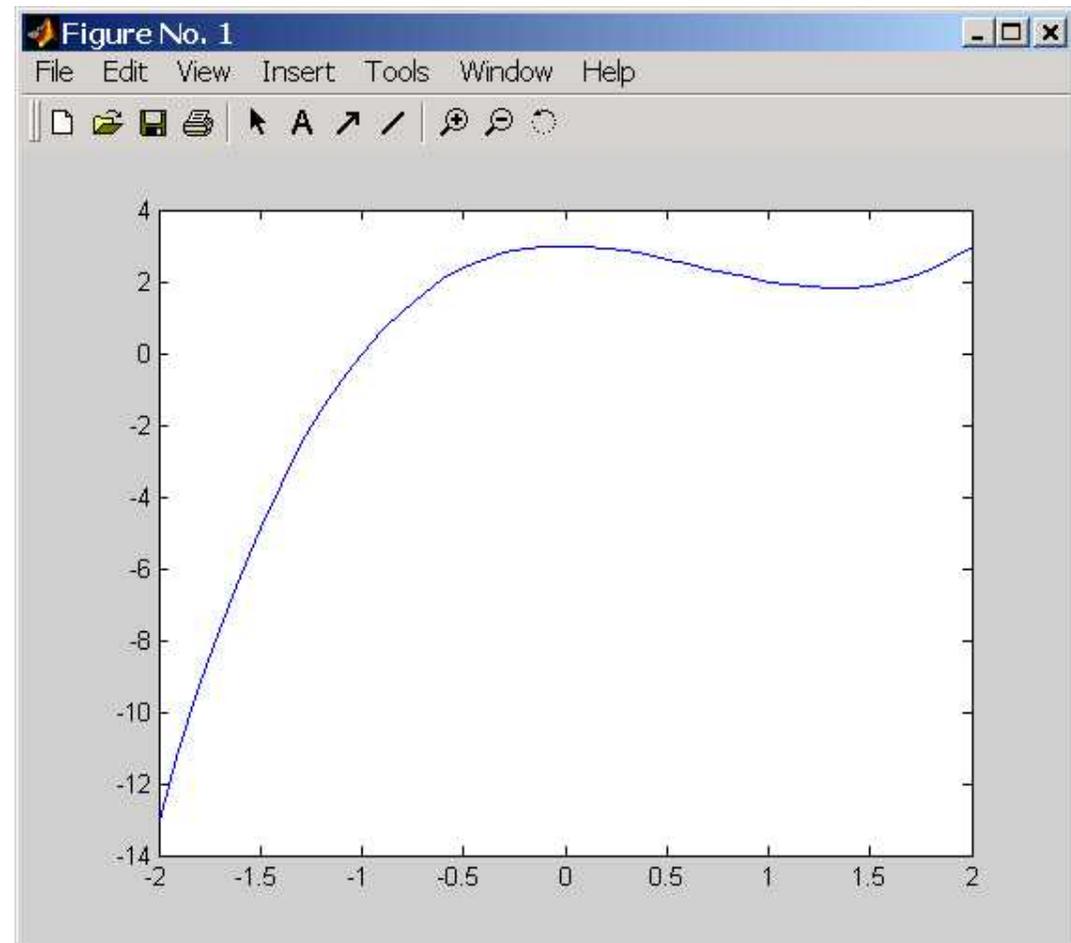
---

# Grafici di polinomi

- Possiamo disegnare un polinomio  $p$ 
  - Considerandolo come una funzione della variabile  $x$ 
    - $p=x.^3+2$ ; `plot(x,p)`
  - Rappresentandolo come un vettore e poi utilizzando opportunamente la funzione `polyval`
    - `plot(x,polyval(p,x))`

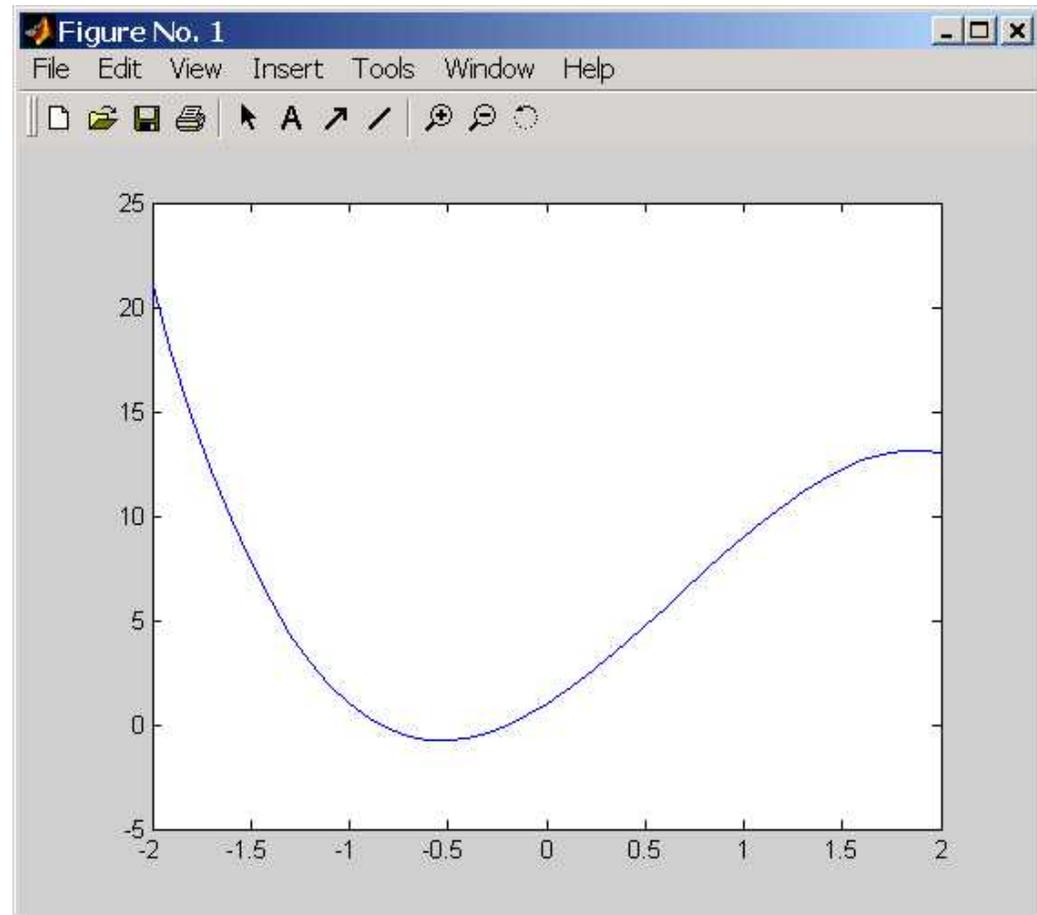
# Esempio – 1

```
>> x=-2:0.1:2;  
>> p=x.^3-2*x.^2+3;  
>> plot(x,p)
```



# Esempio – 2

```
>> x=-2:0.1:2;  
>> p=[-2 4 6 1];  
>> plot(x,polyval(p,x))
```



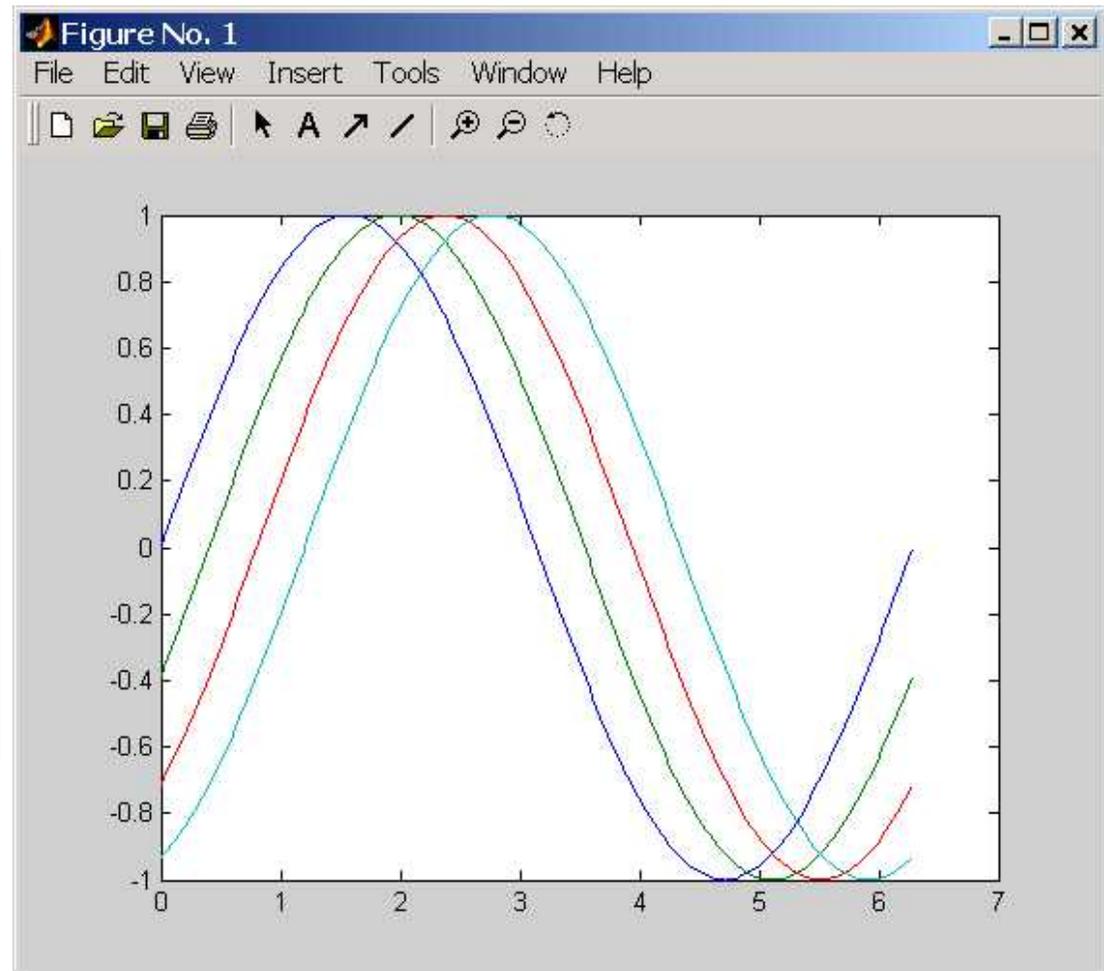
---

# Diagrammi sovrapposti

- Per poter inserire in una stessa finestra più diagrammi sovrapposti è sufficiente indicare nel comando plot diverse coppie x-y (una per ogni grafico)
- Matlab traccia ogni curva con un colore differente
  - L'utente può intervenire nella scelta del colore

# Esempio

```
>> x=0:pi/100:2*pi;  
>> y1=sin(x);  
>> y2=sin(x - .40);  
>> y3=sin(x - .80);  
>> y4=sin(x - 1.2);  
>> plot(x,y1,x,y2,x,y3,x,y4)
```



---

# Grafici sovrapposti

- Invece di usare la sintassi di **plot** vista nelle slide precedenti, possiamo utilizzare il comando **hold on**
- Attraverso **hold on** è possibile disegnare sulla stessa finestra attraverso l'esecuzione di **plot** indipendenti
  - Si torna alla situazione di default (la finestra del grafico viene sovrascritta) tramite il comando **hold off**

# Esempio

```
>> x=0:pi/100:2*pi;  
>> y1=sin(x);  
>> plot(x,y1);  
>> hold on  
>> y2=sin(x - .40);  
>> plot(x,y2);
```

```
>> y3=sin(x - .80);  
>> plot(x,y2);  
>> y4=sin(x - 1.2);  
>> plot(x,y4)
```

---

# Il comando `legend`

- Aggiunge una legenda al grafico
- Sintassi:
  - `legend(stringa1, stringa2, ...)`
    - Inseriamo tante stringhe quanti sono grafici nella finestra
  - `legend(stringa1, stringa2, ..., Pos)`
    - Pos indica la posizione (numero tra -1 e 4)
    - Maggiori dettagli con l'help in linea

# Esercizio

- Disegnare nella stessa finestra i grafici delle funzioni

$$x^3-3x+e^{x/2} \quad \text{per } x \in [3,30]$$

$$\sin(\tan(x)) \quad \text{per } x \in [3,9\pi]$$

$$\log_{10}(x+3)*\sin(2x) \quad \text{per } x \in [1,4\pi+2.3]$$

$$\sin(x)+\cos(x)+e^{3/x} \quad \text{per } x \in [10,26]$$

Aggiungere una legenda al grafico

---

# Tipi di linee e simboli

- È possibile indicare colore, stile della linea e marcatori specificandoli nel comando **plot**
- La sintassi è  
`plot(x,y,'colore_stile_marcatore')`
  - **colore\_stile\_marcatore** è una stringa contenente da uno a quattro caratteri

---

# Colori

- Le stringhe che rappresentano i colori sono:

□ c	ciano	g	verde
□ m	magenta	b	blu
□ y	giallo	w	bianco
□ r	rosso	k	nero

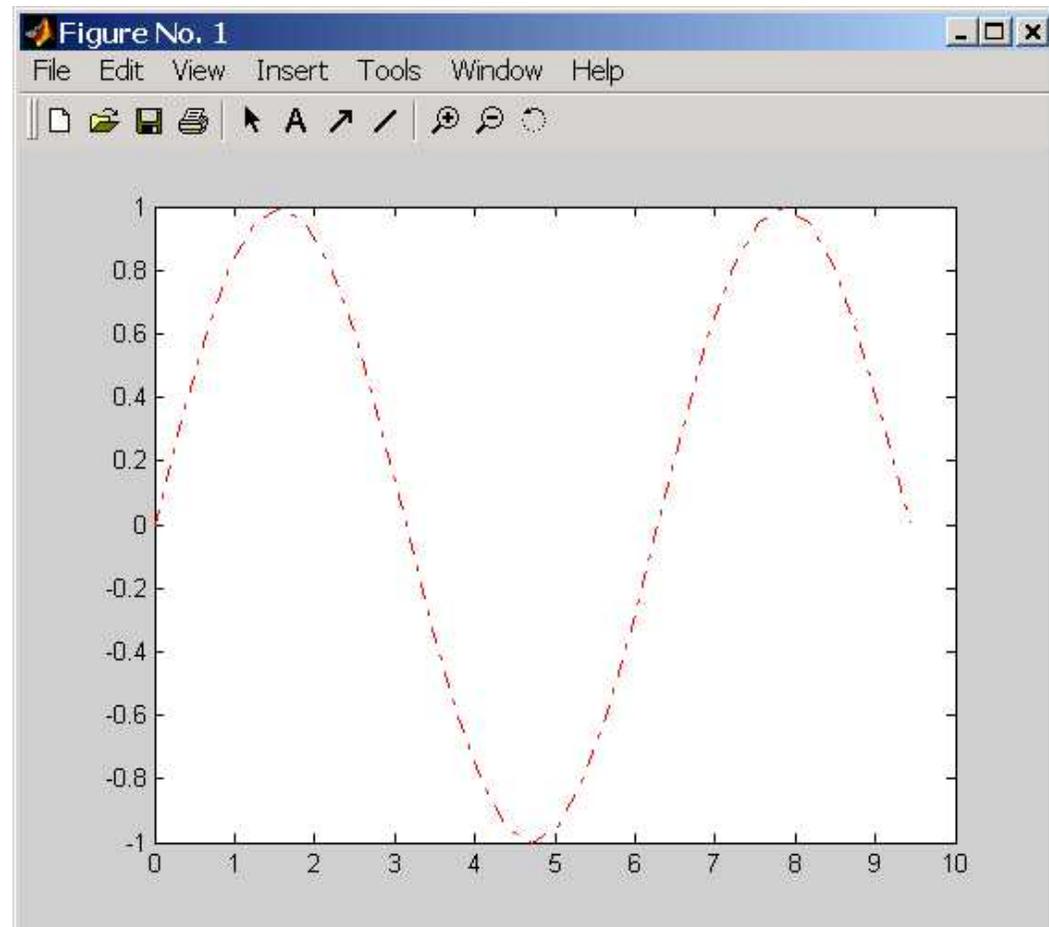
---

# Stili di linee

- Gli stili che possono essere usati per disegnare le linee che uniscono punti successivi sono:
  - - stile solido
  - -- stile tratteggiato
  - : stile punteggiato
  - -. stile tratto-punto
- Omettere il carattere per non avere linee di collegamento tra punti successivi

# Esempio

```
>> x=[0:pi/10:3*pi];  
>> y=sin(x);  
>> plot(x,y,'r-.')
```



# Stile per marcatori

- Gli stili che possono essere usati per disegnare i marcatori sono:

- + o \* x

- s quadrato d diamante

- ^ triangolo superiore

- v triangolo inferiore

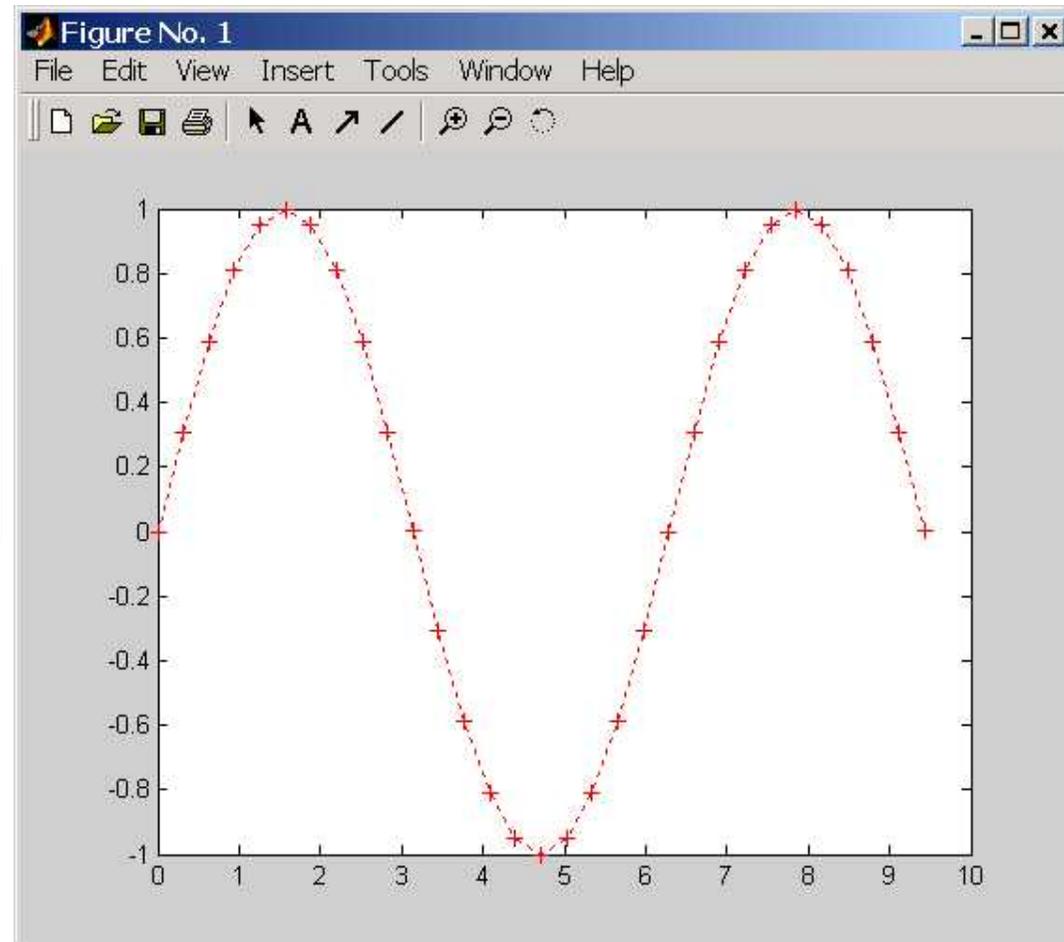
- < triangolo sinistro

- > triangolo destro

- h esagono p pentagono

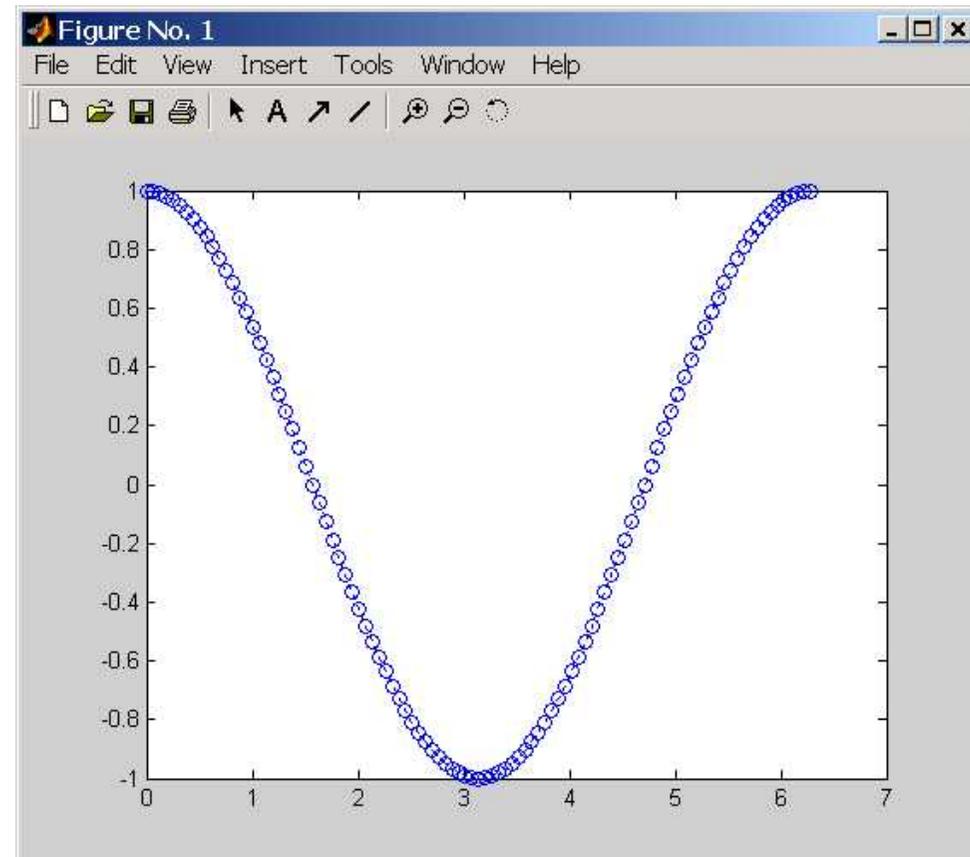
# Esempio – 1

```
>> x=[0:pi/10:3*pi];  
>> y=sin(x);  
>> plot(x,y,'r:+')
```



# Esempio – 2

```
>> x=[0:pi/100:2*pi];  
>> y=cos(x);  
>> plot(x,y, 'b:o')
```



---

## Esercizio – 1

- Disegnare sullo stesso grafico le funzioni  $2\sin(x)$  e  $\cos^2(x)$  nello stesso intervallo  $[0, 2\pi]$
- La prima curva deve essere di colore giallo
- La seconda curva deve essere di colore rosso
- Usare il cerchio per entrambi i marcatori

---

## Esercizio – 2

- Disegnare il grafico della funzione immaginaria  $e^{it}$ 
  - Il marcatore deve essere il cerchio
  - La linea di congiunzione dei marcatori deve avere lo stile solido
  - La linea deve essere di colore verde
  - Gli assi devono essere uguali

---

## Esercizi – 3

- Tracciare nella stessa finestra i grafici sovrapposti delle funzioni
  - $y=x^k$  con  $k=1,3,5,7,9$  ed  $x\in[-4,4]$
- Tracciare nella stessa finestra i grafici sovrapposti delle funzioni
  - $y=a^x$  con  $a$  che varia da 0.1 a 1.1 con passo 0.4 ed  $x\in[0,4]$

## Esercizi – 4

- Tracciare nella stessa finestra i grafici delle funzioni
  - $y = \log_a x$  con  $a$  che varia da 0.1 a 1.9 con passo 0.5 ed  $x \in [0.1, 6]$ 
    - Ricordare che  $\log_a x = \log_b x / \log_b a$
- Tracciare nella stessa finestra i grafici delle funzioni trigonometriche inverse
  - $y = a \sin x$  con  $x \in [-1, 1]$ ,  $y = a \cos x$  con  $x \in [-1, 1]$ ,  
 $y = a \tan x$  con  $x \in [-5, 5]$ ,

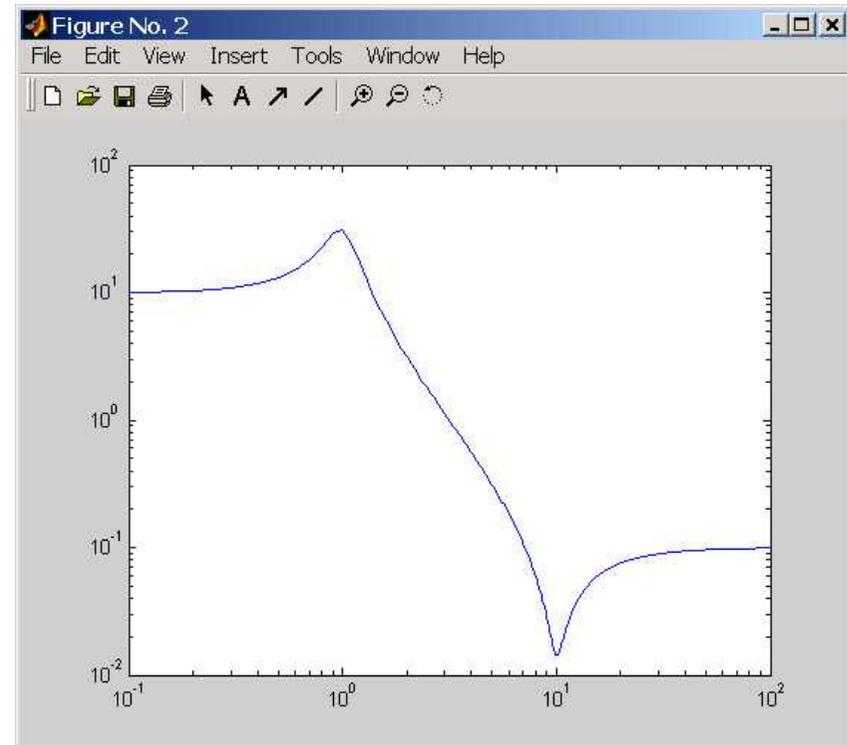
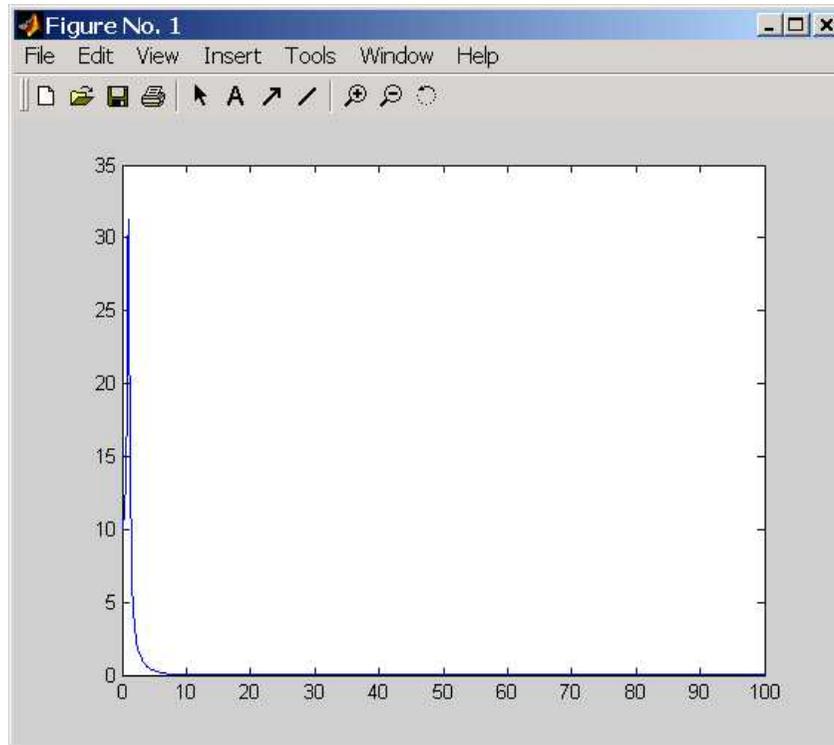
---

# Altre tipi di scale

- Oltre alla scala decimale è possibile disegnare dei grafici usando altre scale (ad esempio quella logaritmica)
  - Si possono rappresentare dati che si estendono per un insieme molto vasto di valori
  - Si mettono in evidenza andamenti nella variazione dei dati
- Invece del comando plot si usa
  - loglog – semilogx – semilogy

# Esempio

$$\sqrt{\frac{100(1-0.01x^2)+0.02x^2}{(1-x^2)^2+0.1x^2}}$$



```
x=0.1:0.1:100;
```

```
y=sqrt((100*(1-0.01*x.^2).^2+0.02*x.^2)./((1-x.^2).^2+0.1*x.^2));
```

---

# Note su scala logaritmica

- Non è possibile rappresentare numeri negativi
- Non è possibile rappresentare il numero zero
- Le etichette dei segni di graduazione corrispondono ai valore realmente rappresentati, non i loro logaritmi
- I numeri che si differenziano per un fattore fissato (ad esempio 10) sono separati dalla stessa distanza
  - La distanza tra 0,3 e 3 è la stessa tra 30 e 300

---

# Modifica spaziatura ed etichette

- È possibile modificare la spaziatura e le etichette dei segni di graduazione come segue

```
set(gca, 'XTick', [xmin:dx:xmax], 'YTick', [ymin:dy:ymax])
```

```
set(gca, 'XTicklabels', [l1; l2; ... ;lk])
```

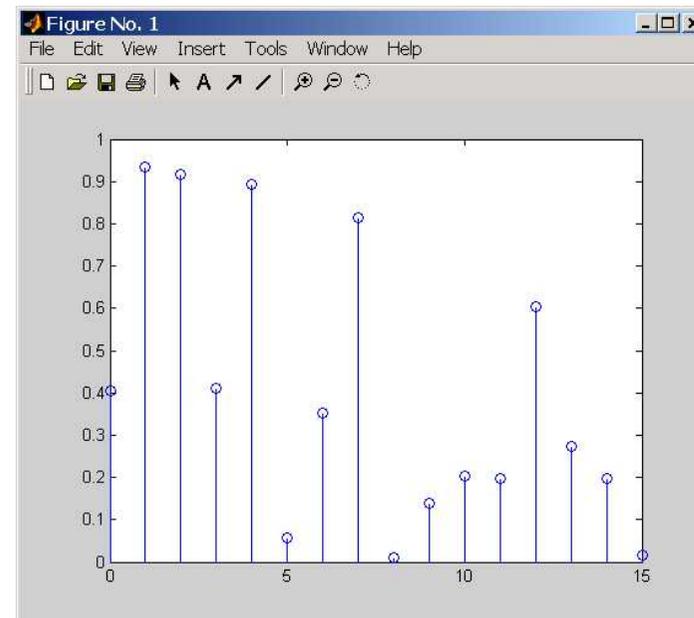
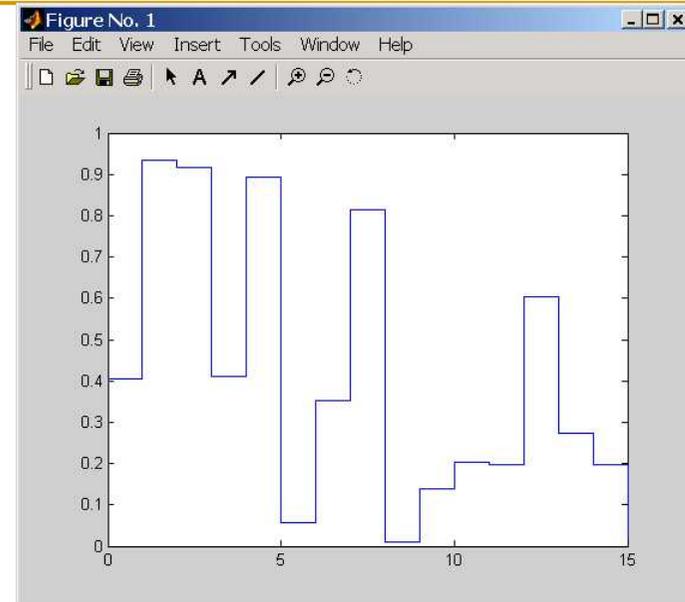
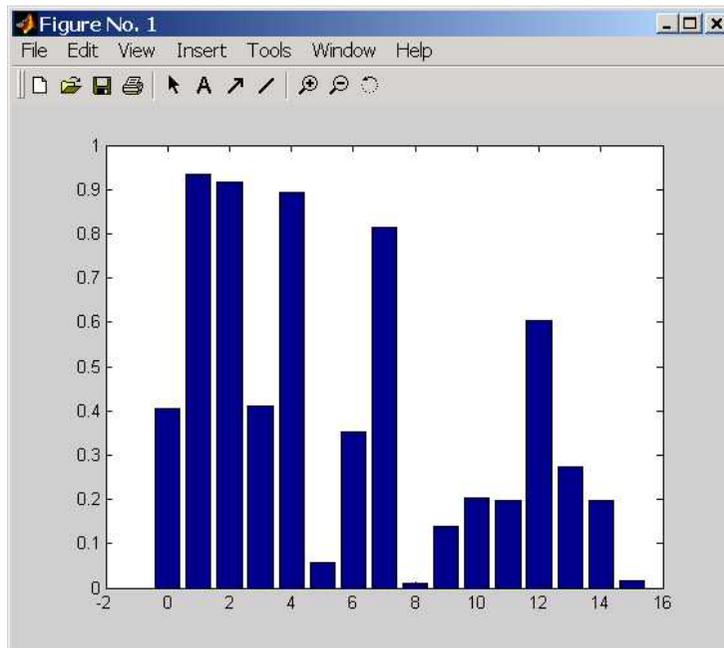
- $l_i$  indica un'etichetta (stringa racchiusa tra apici) da inserire al posto del segno di graduazione

---

# Grafici particolari

- Matlab dispone di altri tipi di diagrammi per rappresentare i dati
  - diagrammi a barre `bar(x,y)`
  - diagrammi a gradini `stairs(x,y)`
  - diagrammi a steli `stem(x,y)`

# Esempi

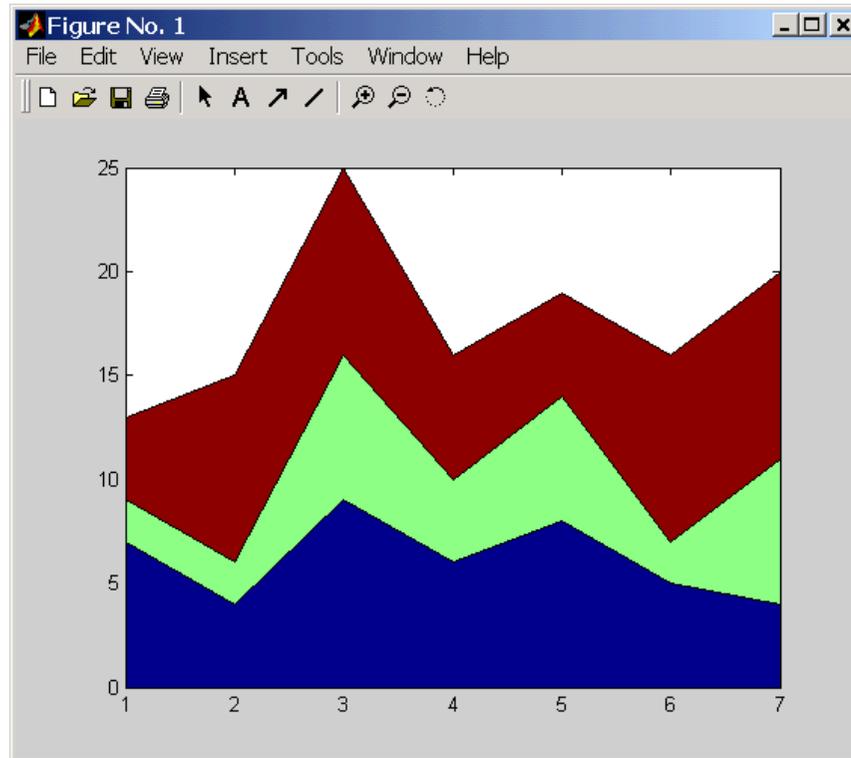


---

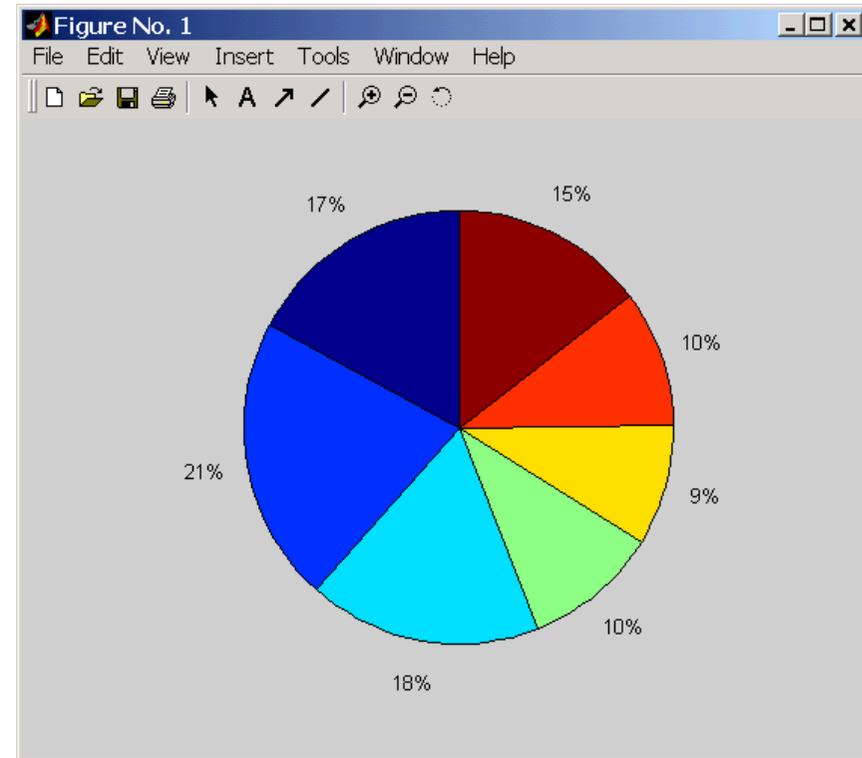
# Altri tipi di grafici

- Si possono usare i comandi **area** e **pie** per creare altri tipi di grafici
  - **area**(y) produce il grafico delle **n** curve rappresentate dalle colonne di y (y è una matrice  $m \times n$ ).
    - L'area sottostante le curve è colorata.
    - L'altezza dell'area del grafico i-esimo è la somma degli elementi nelle righe i, i+1, ..., m
  - **pie**(x) produce un diagramma a torte del vettore x

# Esempi



```
>> y=fix(rand(7,3)*10+1);  
>> area(y)
```



```
>> x=rand(1,7)*10+1;  
>> pie(x)
```

---

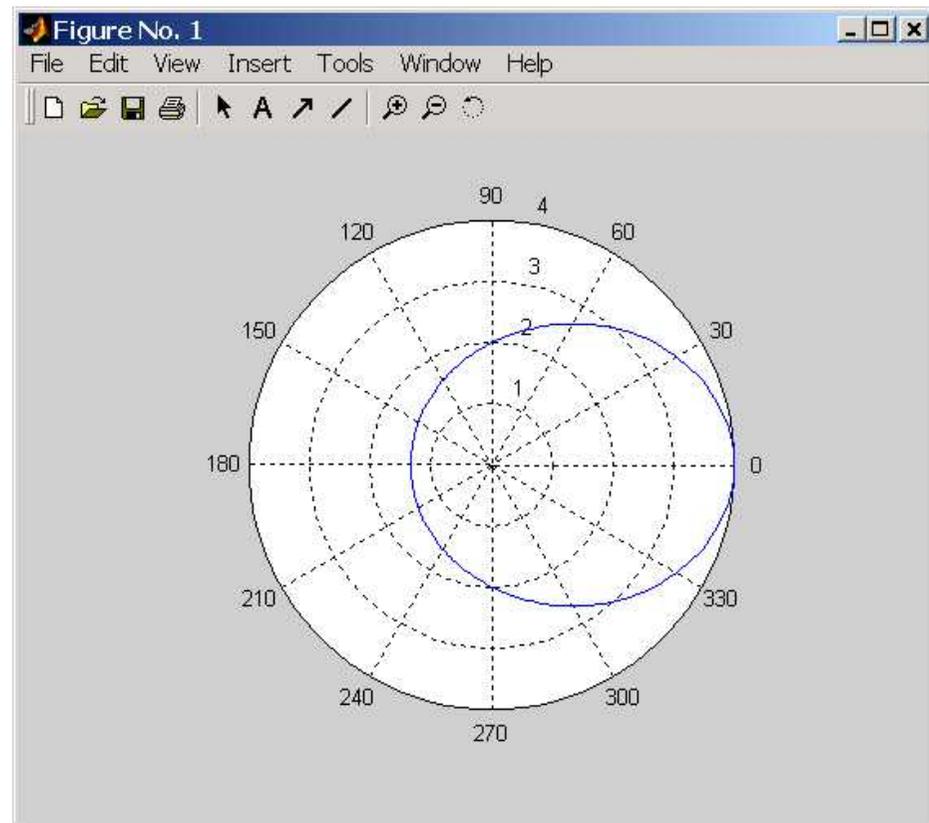
# Diagrammi polari

- I diagrammi polari sono diagrammi bidimensionali che usano coordinate polari
- Il comando è `polar(teta,r)`
  - `teta` è la coordinata polare
  - `r` è la coordinata radiale

# Diagramma polare di un'orbita

- $r = p/(1-\varepsilon\cos \theta)$ 
  - $\varepsilon$  è l'eccentricità
  - $p$  è la dimensione

```
>> teta=0:pi/100:2*pi;  
>> r=2./(1-0.5*cos(teta));  
>> polar(teta,r)
```



---

# Esercizi

- Creare il diagramma delle funzioni descritte dalle coordinate polari:
  - $r=a\theta$  con  $\theta \in [0,4\pi]$  ed  $a=2$
  - $r=\tan(\cos(\theta))+\sin(\tan(\theta))$  con  $\theta \in [0,2\pi]$
  - $r=\cos(\theta)+3\theta +\sin(\tan(\theta))$  con  $\theta \in [0,5\pi]$