

# ELEMENTI DI GRAFICA IN MATLAB/OCTAVE

MANOLO VENTURIN

UNIVERSITÀ DEGLI STUDI DI PADOVA  
DIP. DI MATEMATICA PURA ED APPLICATA

A. A. 2007/2008

# INDICE

- Creazione di un vettore  
(es. punti equispaziati)
- Definizione della funzione da visualizzare  
(formulazione puntuale)
- Visualizzazione
- Aggiungere informazioni ai grafici  
(titolo, etichette agli assi, ...)
- Altri tipi di grafici

# Creazione di un vettore (es. punti equispaziati)

```
>> % 6 punti equispaziati tra 0 e 10
```

```
>> x = linspace(0,10,6);
```

```
>> % Vediamo il contenuto di x
```

```
>> x
```

```
x =
```

```
    0     2     4     6     8    10
```

# Esercizio in classe

## TESTO

- Generare un vettore di tra -2.5 e 2.5 di 11 elementi equispaziati
- Dire se lo 0 è un elemento di tale vettore
- Fornire l'ampiezza del passo

# Esercizio in classe

## SOLUZIONE

```
>> % 11 punti equispaziati tra -2.5 e 2.5  
>> x = linspace(-2.5,2.5,11);
```

```
>> % Vediamo il contenuto di x  
>> x
```

```
x =  
-2.5000 -2.0000 -1.5000 -1.0000  
-0.5000 0 0.5000 1.0000  
1.5000 2.0000 2.5000
```

# Esercizio in classe

## SOLUZIONE

```
>> % lo zero e' presente nel vettore x
```

```
>> % la lunghezza dell'intervallo e'
```

```
>> -2 - (-2.5)
```

```
ans =  
    0.5000
```

```
>> % soluzione alternativa
```

```
>> abs(-2.5-(-2))
```

```
ans =  
    0.5000
```

# Definizione funzione

## ESEMPIO

```
>> % Creazione vettore x
```

```
>> x = linspace(0,10,6);
```

```
>> % Es: def. funzione  $y = 2*x*\exp(-x)$ 
```

```
>> y = 2*x .*exp(-x);
```

```
>> % Attenzione alle sostituzioni sui vettori
```

```
>> % * che diventa .*
```

```
>> % / che diventa ./
```

```
>> % ^ che diventa .^
```

# Definizione funzione

## CONTENUTO VARIABILI

```
>> % Contenuto vettore x
```

```
>> x
```

```
x =
```

```
    0     2     4     6     8    10
```

```
>> % Contentuo vettore y
```

```
>> y
```

```
y =
```

```
    0    0.5413    0.1465    0.0297  
0.0054    0.0009
```

# Definizione Funzione

## VERIFICA

```
>> % Verifica in x uguale a 0
```

```
>> 2*0.*exp(-0)
```

```
ans =  
    0
```

```
>> % Verifica in x uguale a 2
```

```
>> 2*2.*exp(-2)
```

```
ans =  
    0.5413
```

# Visualizzazione ESEMPIO

```
>> % Creazione vettore x
```

```
>> x = linspace(0,10,6);
```

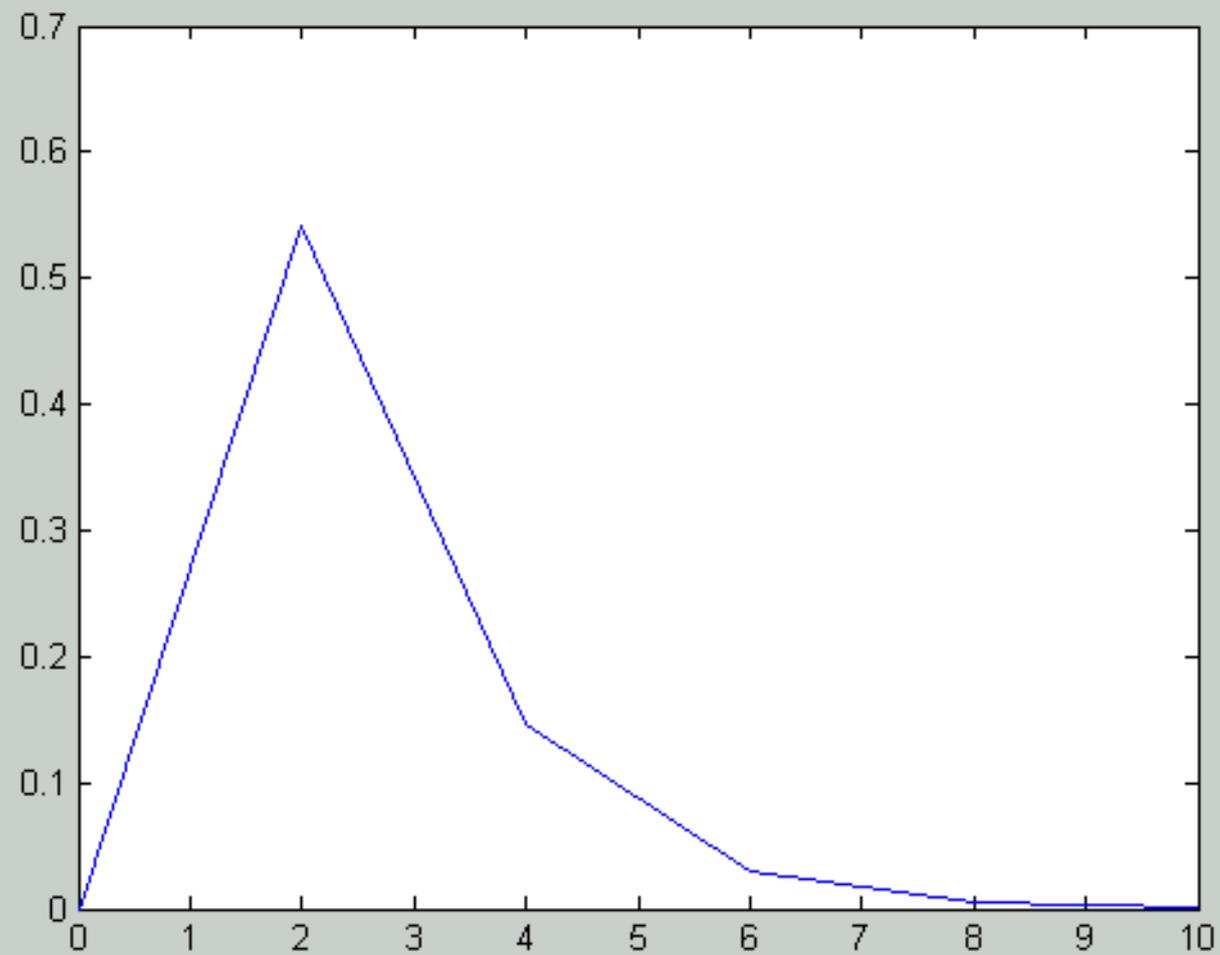
```
>> % Def. funzione  $y = 2*x*\exp(-x)$ 
```

```
>> y = 2*x .*exp(-x);
```

```
>> % Visualizzazione funzione
```

```
>> plot(x,y);
```

# Visualizzazione RISULTATO



# Esercizio in classe

## TESTO

- Generare un vettore  $x$  tra  $-5.0$  e  $5.0$  di 101 elementi equispaziati
- Disegnare le seguenti 2 funzioni:  
(chiamare rispettivamente i vettori delle soluzioni  $y_1$  e  $y_2$ )

- $$y_1 = \frac{1}{x^2 + 1} \quad y_2 = \sin(x * \exp(-|x|))$$

Oss: `abs` è la routine che implementa il valore assoluto

# Esercizio in classe

## SOLUZIONE

```
>> % Creazione vettore x
```

```
>> x = linspace(-5,5,101);
```

```
>> % Def. funzione y1
```

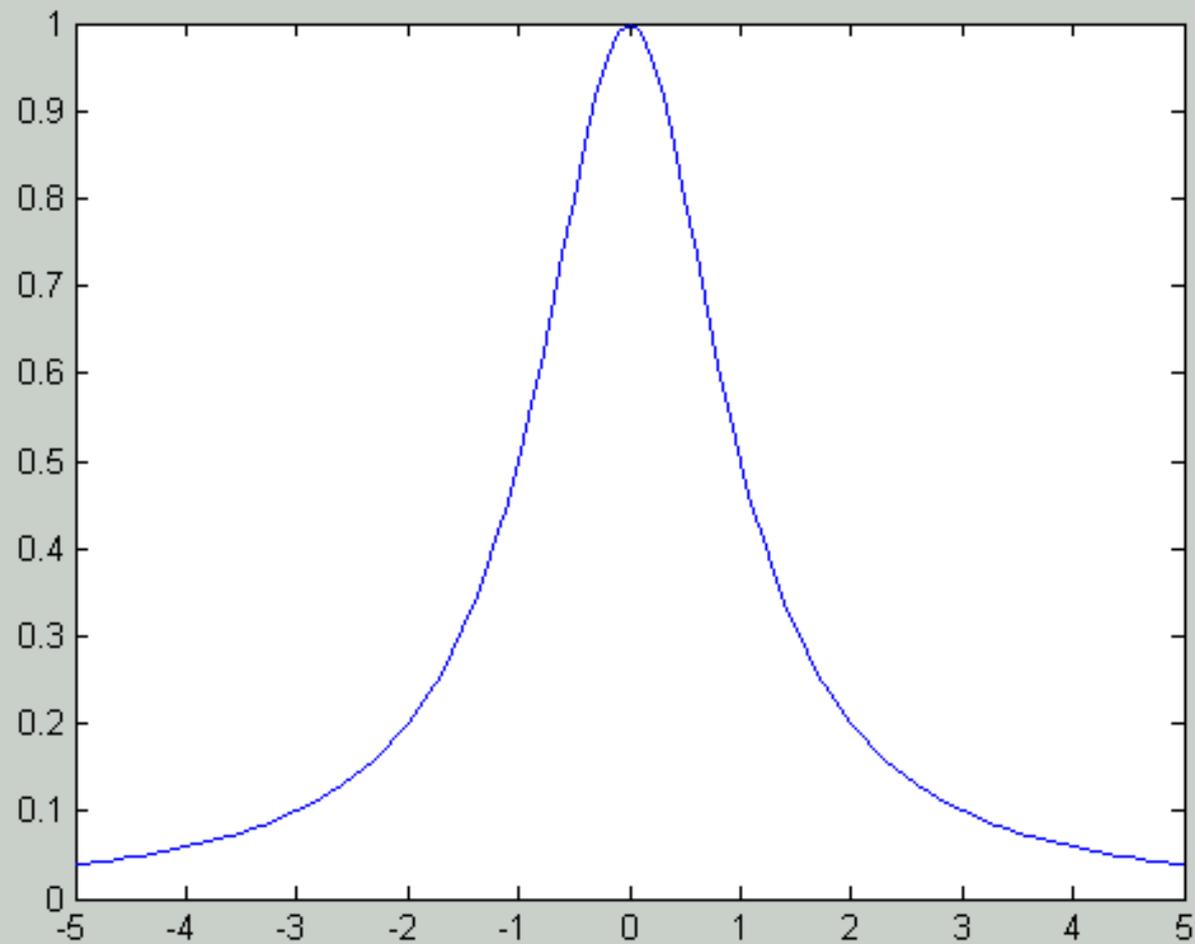
```
>> y1 = 1./(x.^2+1);
```

```
>> % Visualizzazione funzione
```

```
>> plot(x,y1);
```

# Esercizio in classe

## SOLUZIONE



# Esercizio in classe

## SOLUZIONE

```
>> % Creazione vettore x
```

```
>> x = linspace(-5,5,101);
```

```
>> % Def. funzione y2
```

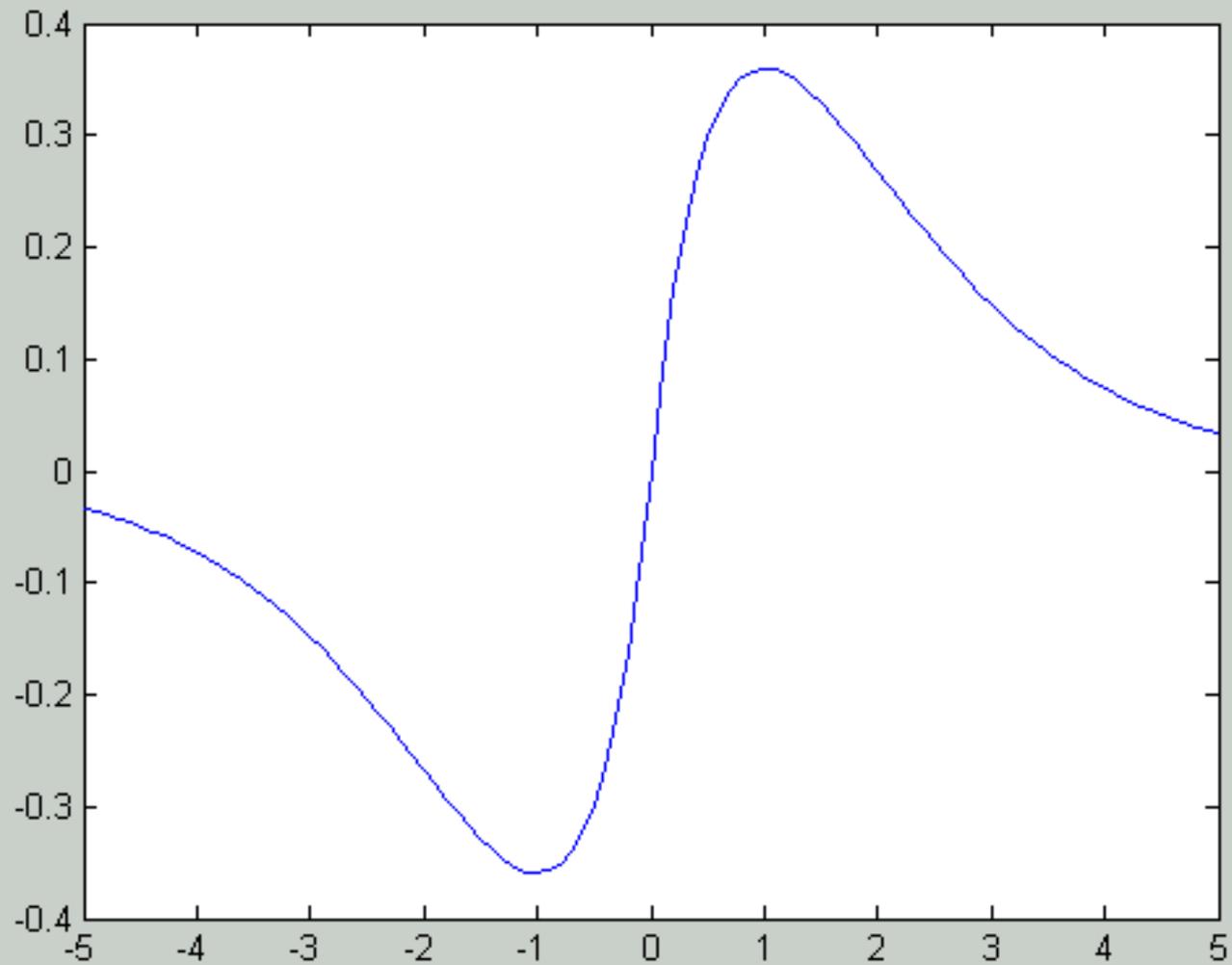
```
>> y2 = sin(x.*exp(-abs(x)));
```

```
>> % Visualizzazione funzione
```

```
>> plot(x,y2);
```

# Esercizio in classe

## SOLUZIONE



# Visualizzazione avanzata

- Titolo
- Etichette assi x e y
- Cambiare colore e forma ai grafici
- Più funzioni in una finestra
- Legenda
- Griglia
- Più finestre in contemporanea

# Visualizzazione avanzata

## TITOLO-ETICHETTE ASSI

```
>> % Ridisegnamo y2
```

```
>> plot(x,y2);
```

```
>> % Titolo
```

```
>> title('titolo della figura');
```

```
>> % Asse x
```

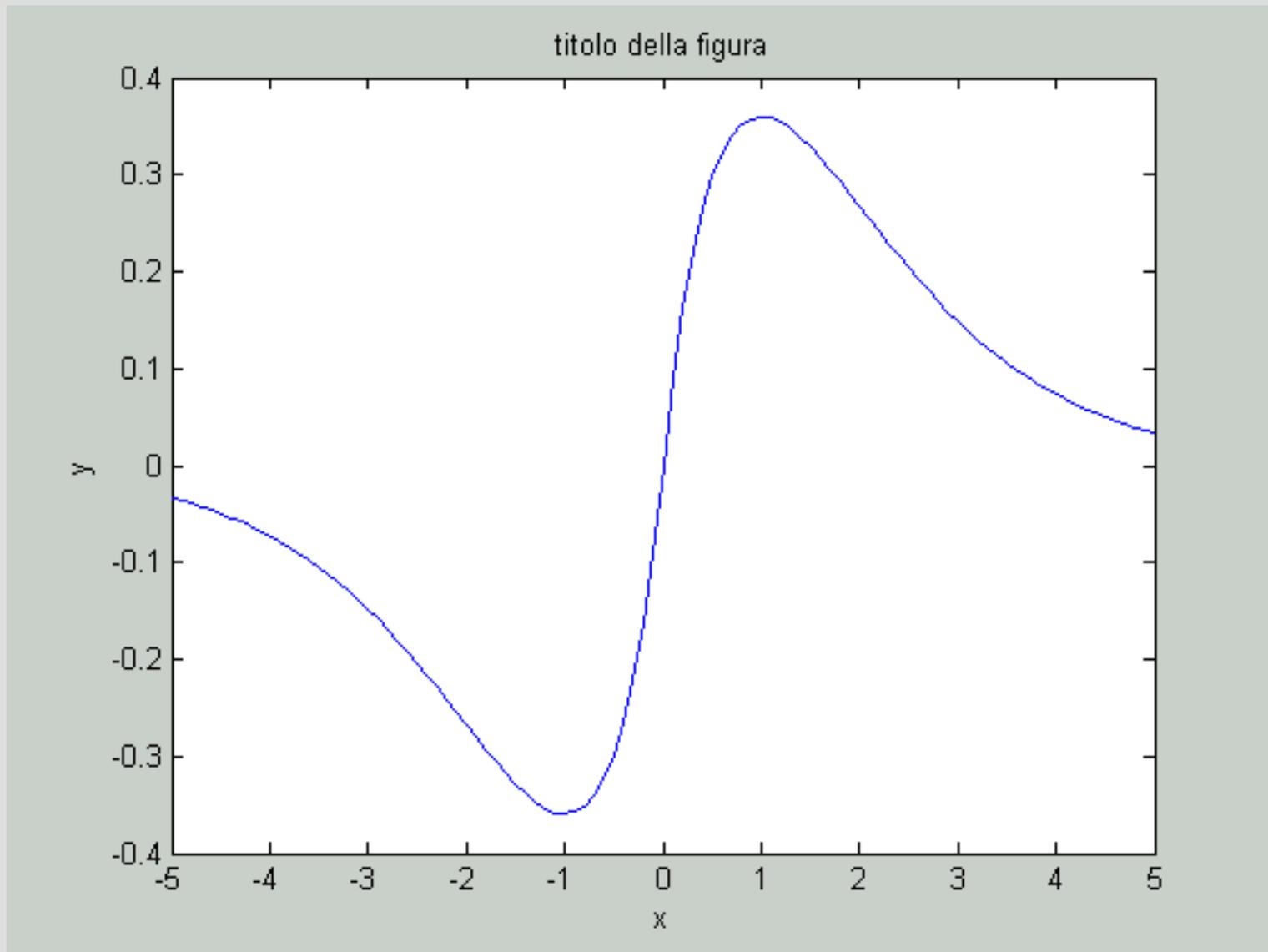
```
>> xlabel('x');
```

```
>> % Asse y
```

```
>> ylabel('y');
```

# Visualizzazione avanzata

## TITOLO-ETICHETTE ASSI



# Visualizzazione avanzata

## COLORE

```
>> % y2 in colore rosso
```

```
>> plot(x,y2,'r');
```

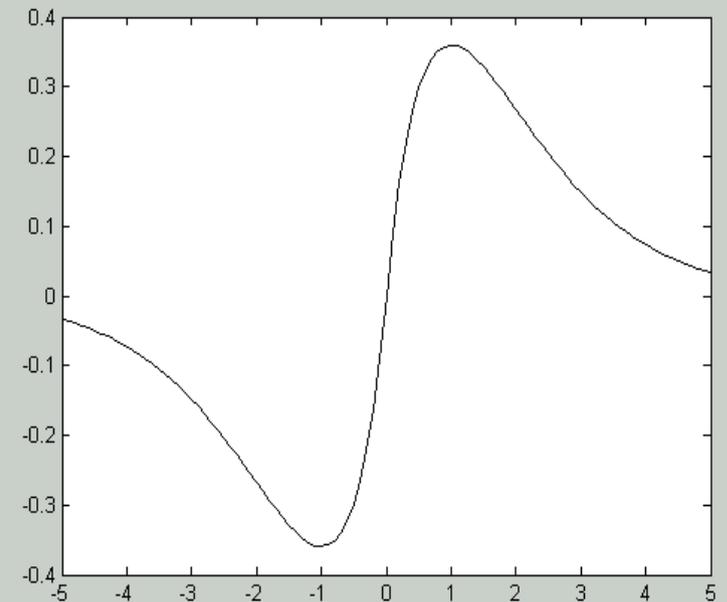
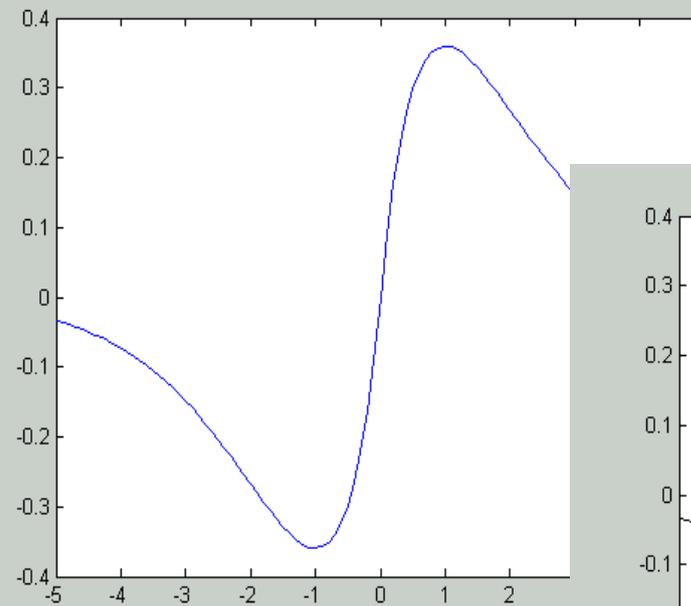
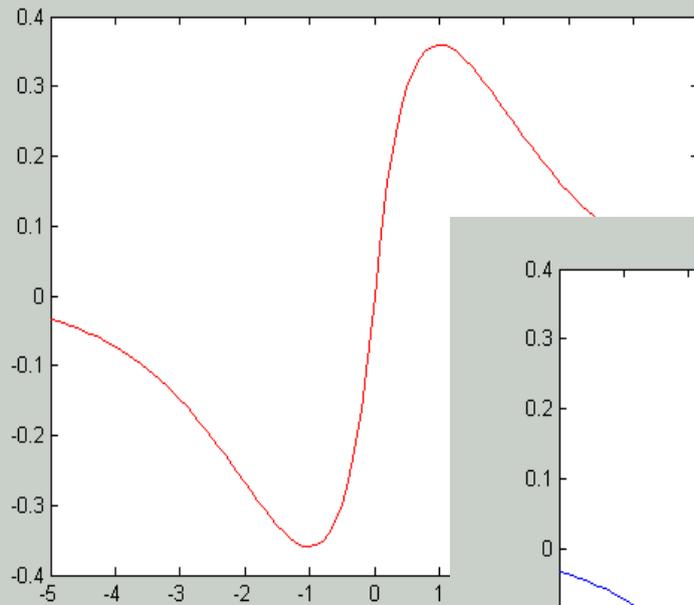
```
>> % y2 in colore blue
```

```
>> plot(x,y2,'b');
```

```
>> % y2 in colore nero
```

```
>> plot(x,y2,'k');
```

# Visualizzazione avanzata COLORE



# Visualizzazione avanzata

## FORMA

>> % y2: linea continua rossa in cui i  
nodi sono evidenziati con dei cerchi

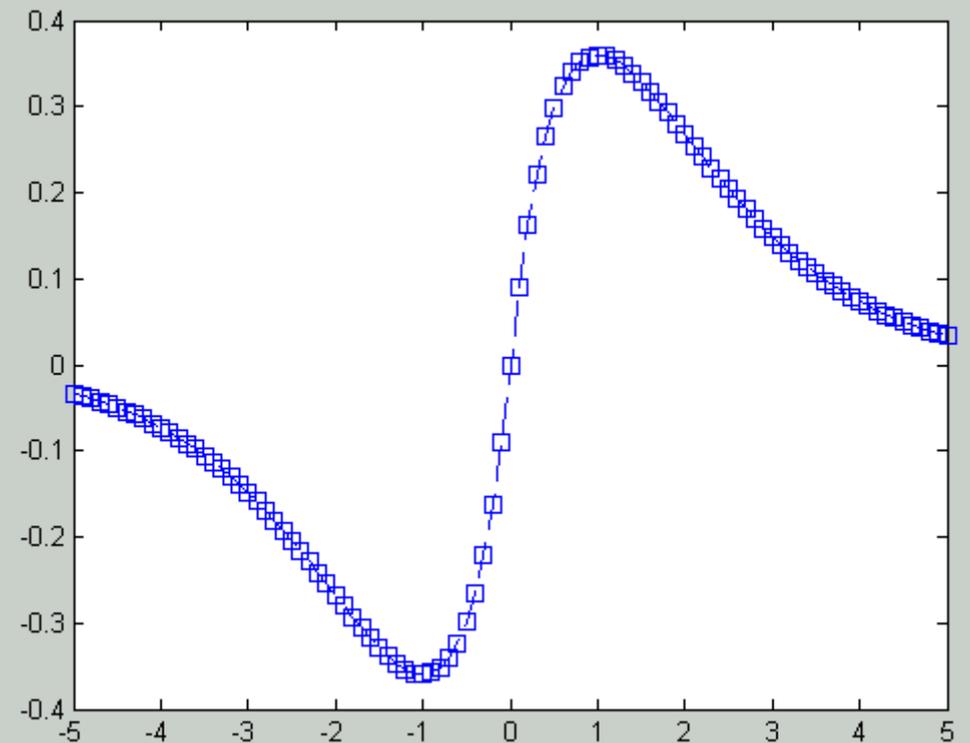
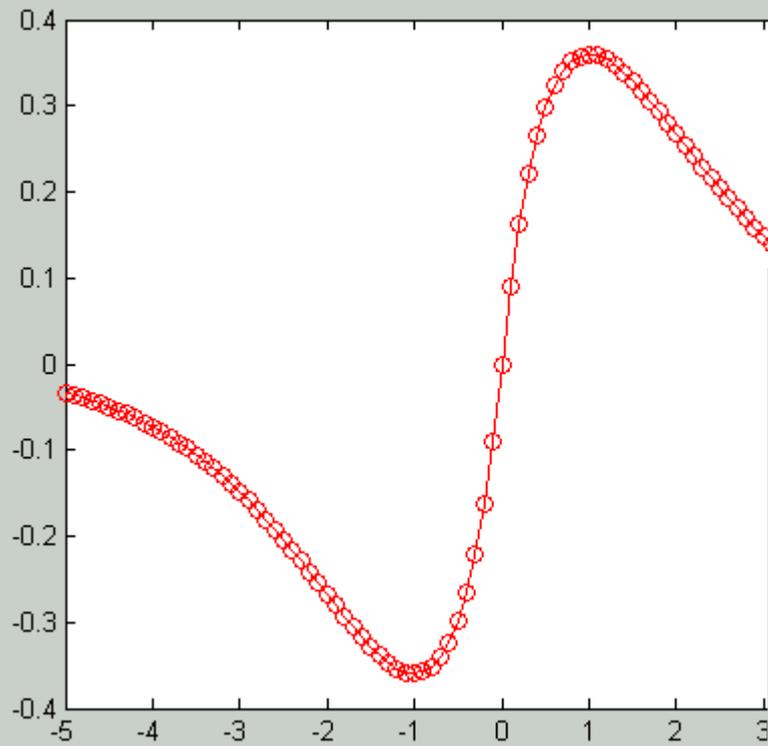
```
>> plot(x,y2,'ro-');
```

>> % y2: linea tratteggiata blu in cui i  
nodi sono evidenziati con dei quadrati

```
>> plot(x,y2,'bs--');
```

>> % maggiori info con help plot

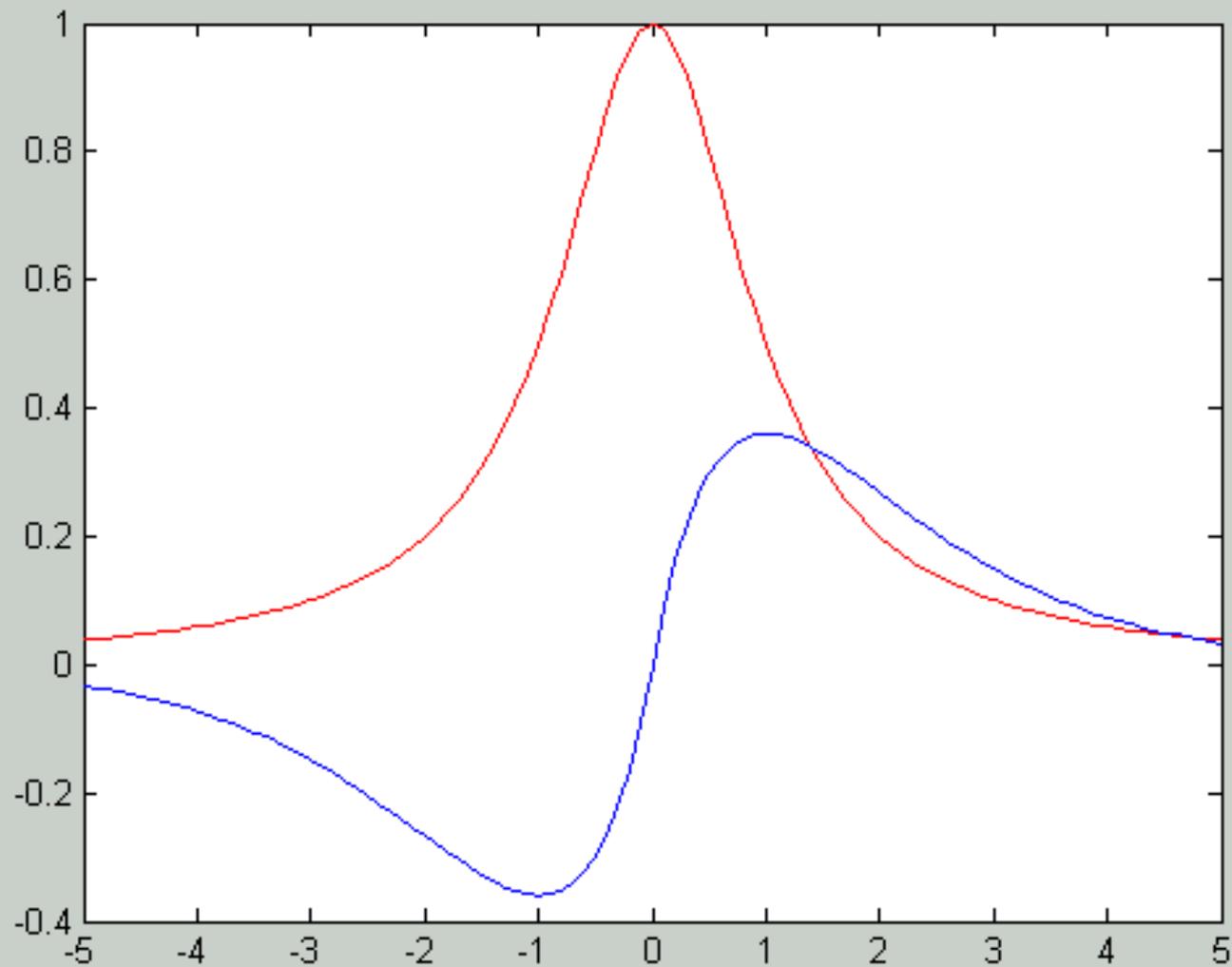
# Visualizzazione avanzata FORMA



# Visualizzazione avanzata PIÙ FUNZIONI IN FINESTRA

```
>> % visualizziamo y1 e y2 assieme  
>> % Disegna y1  
>> plot(x,y1,'r');  
  
>> % Trattieni ambiente figura  
>> hold on;  
>> % Disegna y2  
>> plot(x,y2,'b');  
>> % Rilascia ambiente figura  
>> hold off;
```

# Visualizzazione avanzata PIÙ FUNZIONI IN FINESTRA



# Visualizzazione avanzata

## LEGENDA e Griglia

```
>> % titolo, asse x, asse y
```

```
>> title('Bella Figura');
```

```
>> xlabel('Asse x');
```

```
>> ylabel('Asse y');
```

```
>> % Aggiungo legenda
```

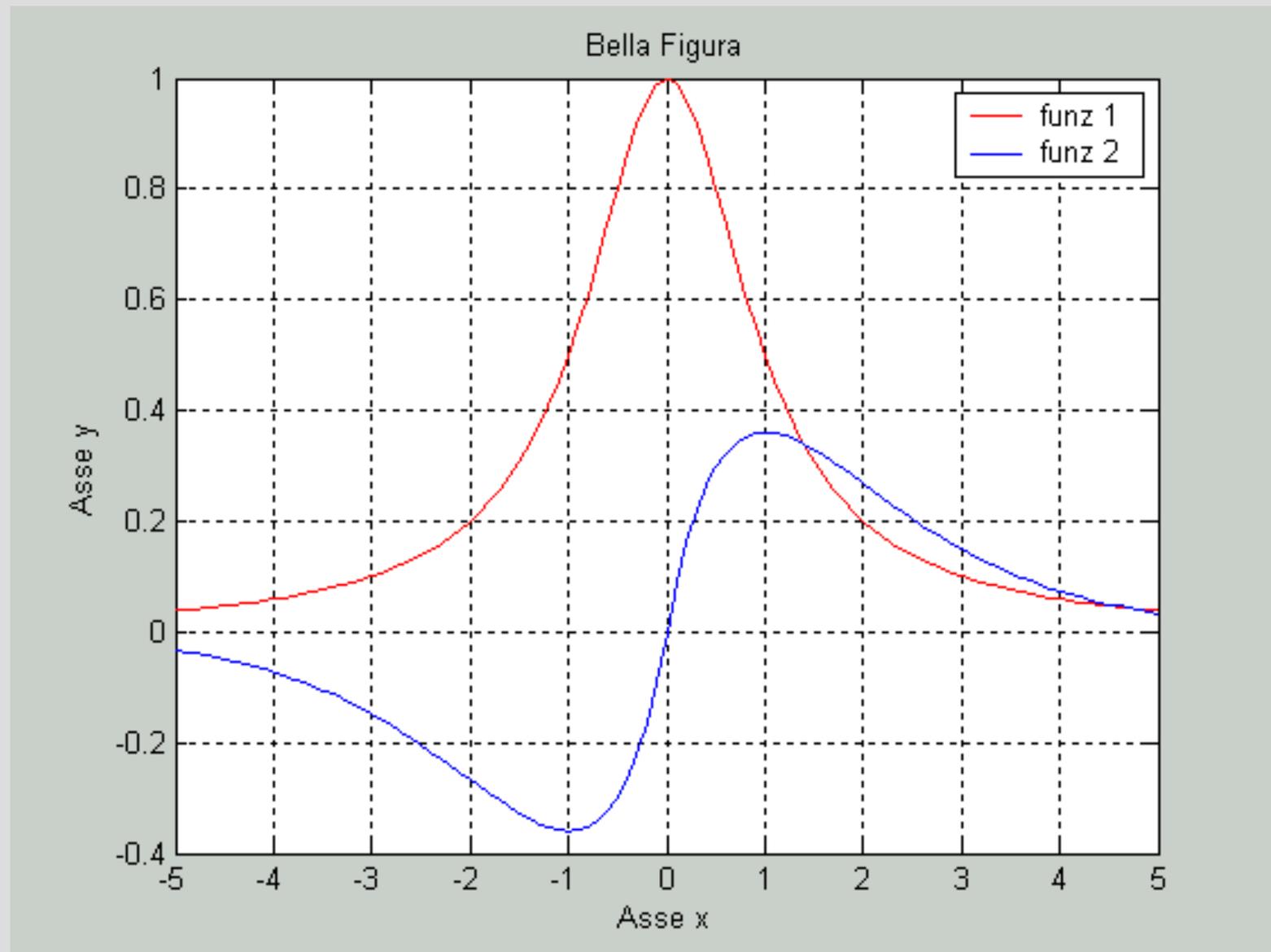
```
>> legend('funz 1','funz 2');
```

```
>> % Aggiungo griglia
```

```
>> grid on;
```

# Visualizzazione avanzata

## LEGENDA



# Visualizzazione avanzata

## PIÙ FINESTRE

```
>> % Si utilizza il comando figure con il
    numero della figura
>> % Attivo finestra 1
>> figure(1);
>> % Disegna y1
>> plot(x,y1,'r');
>> % Attivo finestra 2
>> figure(2);
>> % Disegna y2
>> plot(x,y2,'r');
```

# Visualizzazione avanzata PIÙ FINESTRE

Figure No. 1

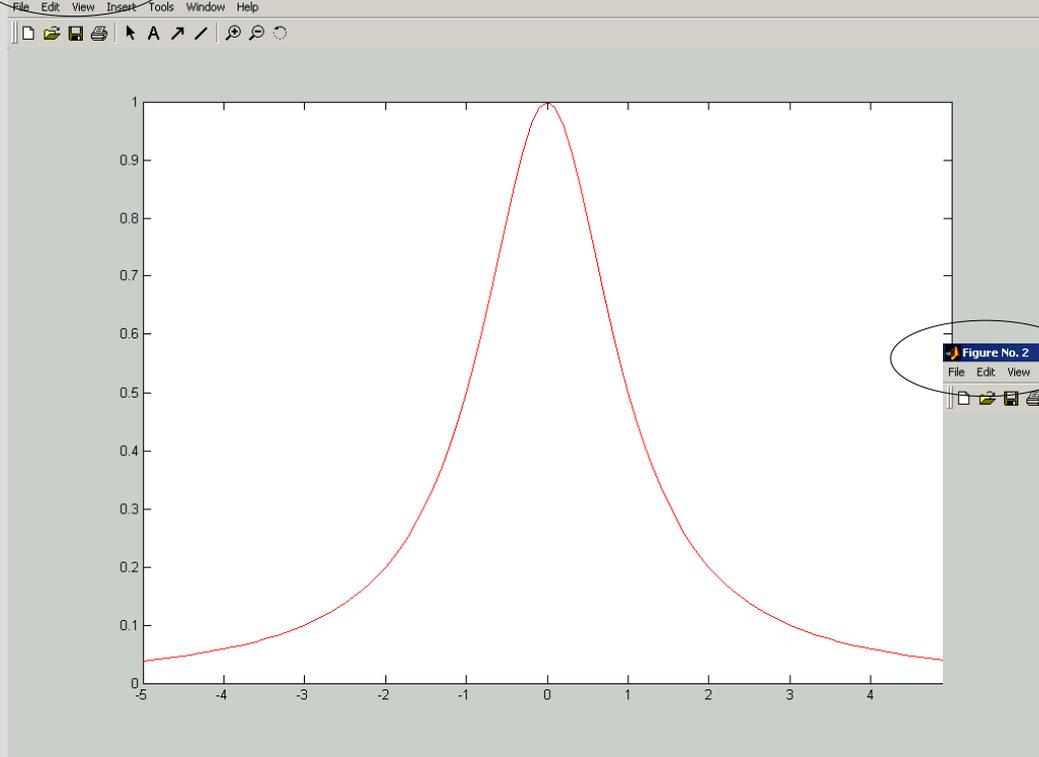
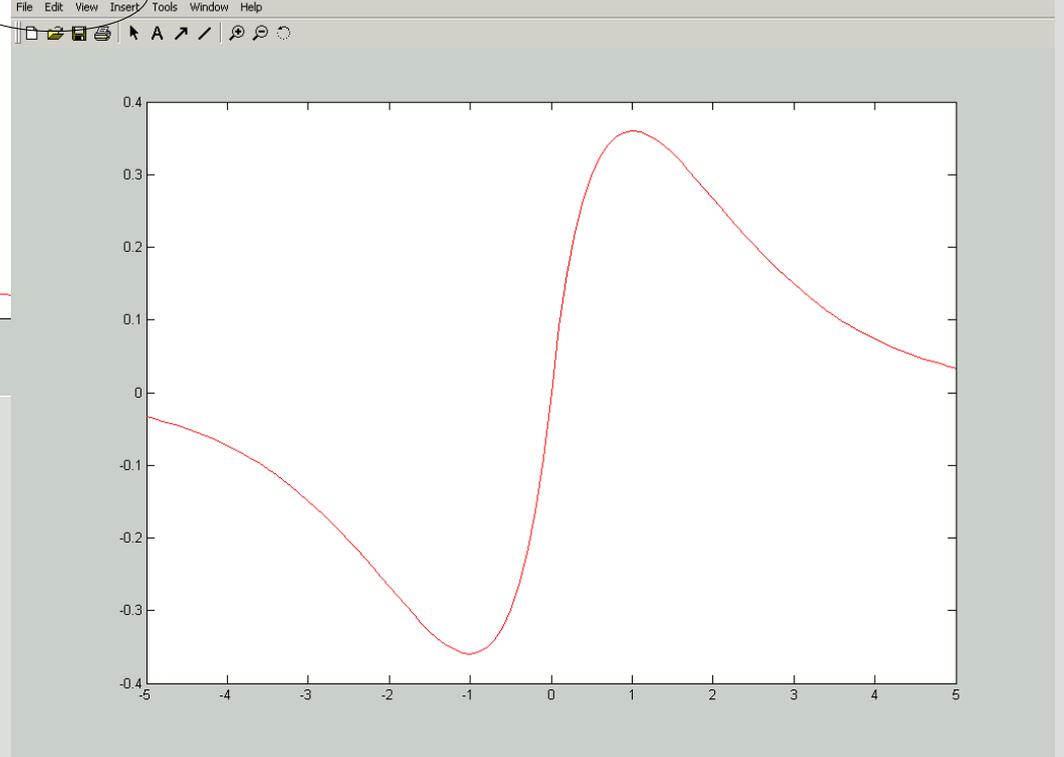


Figure No. 2



# Altri grafici

## SEMILOGY

```
>> % Grafico utile nello studio degli  
errori
```

```
>> % semilogy: log nelle y
```

```
>> x = linspace(0,12,13);
```

```
>> y = 10.^(-x);
```

```
>> % visualizzo differenza
```

```
>> figure(1); semilogy(x,y); grid on
```

```
>> figure(2); plot(x,y); grid on
```

# Altri grafici SEMILOGY

