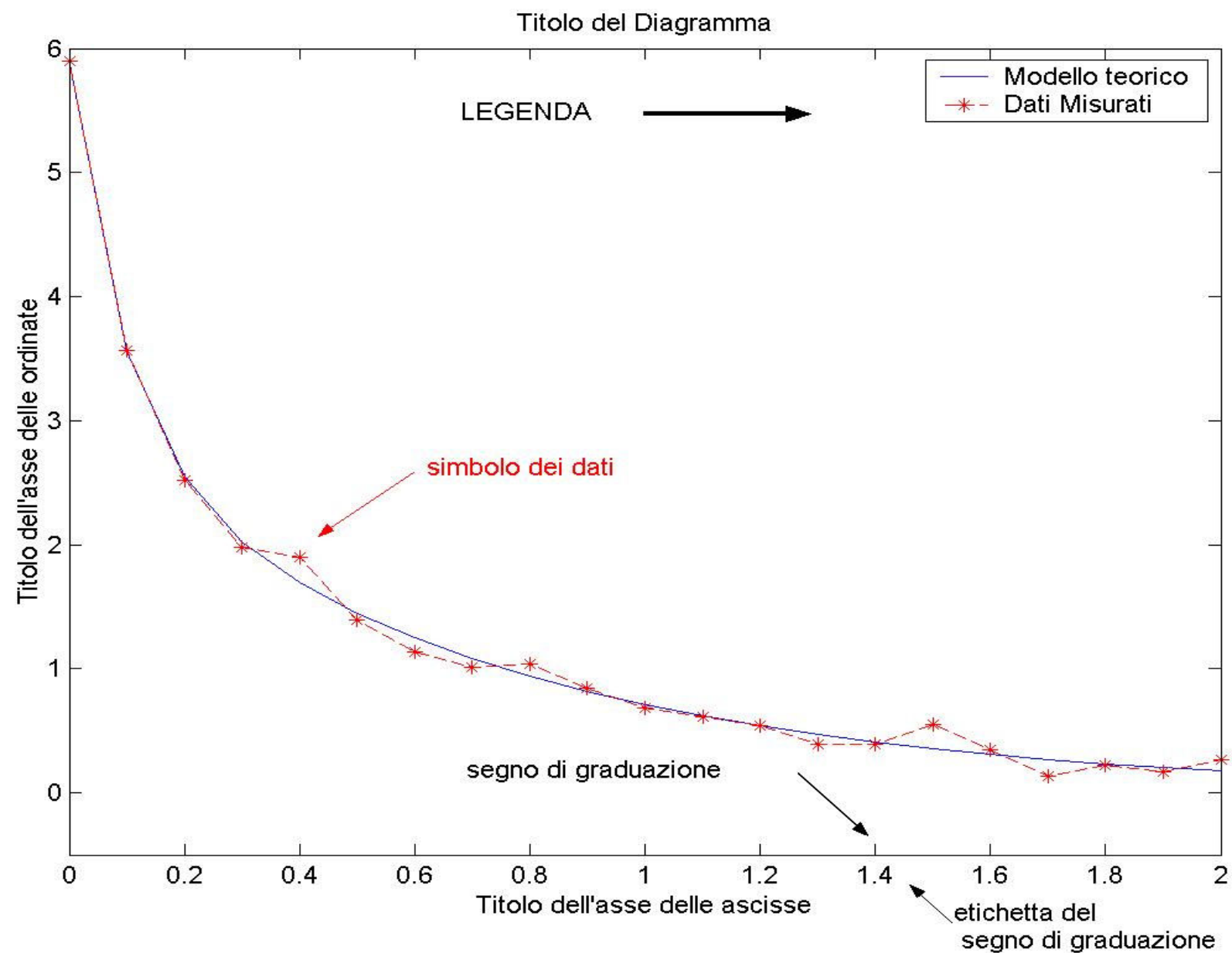


La Grafica in MATLAB



- La scala nei due grafici fa riferimento all'intervallo e alla spaziatura dei numeri rappresentati.
- Entrambi gli assi del diagramma precedente sono lineari, in quanto la spaziatura dei numeri è costante, un altro tipo di scala è quella logaritmica.
- I segni di graduazione (tick mark) sono posti sull'asse per facilitare la valutazione dei numeri rappresentati nel diagramma.
- Le etichette dei segni di graduazione (tick mark label) sono i numeri che corrispondono alle posizioni dei segni di graduazione;

- La funzione `plot(x,y)` di MatLab imposta automaticamente la spaziatura dei segni di graduazione per i due assi cartesiani e pone le etichette appropriate: questa caratteristica si chiama *scala automatica* (autoscaling).
- Il comando `axis` serve a modificare le impostazioni dei valori limite che MATLAB ha scelto per gli assi. La sintassi di base del comando è:

`axis([xmin xmax ymin ymax])`

Esistono diverse varianti di questo comando:

- `axis square` :seleziona i limiti degli assi in modo che il diagramma sia quadrato;
- `axis equal`: seleziona i fattori di scala e la spaziatura dei segni di graduazione in modo che siano uguali nei due assi;
- `axis auto`: attiva le impostazioni di MATLAB che calcolano i limiti ideali degli assi in modo automatico.
- Il comando `grid` visualizza le linee di una griglia in corrispondenza delle etichette dei segni di graduazione di un diagramma.
- Digitando `grid on` si aggiunge la griglia, `grid off` la si esclude.

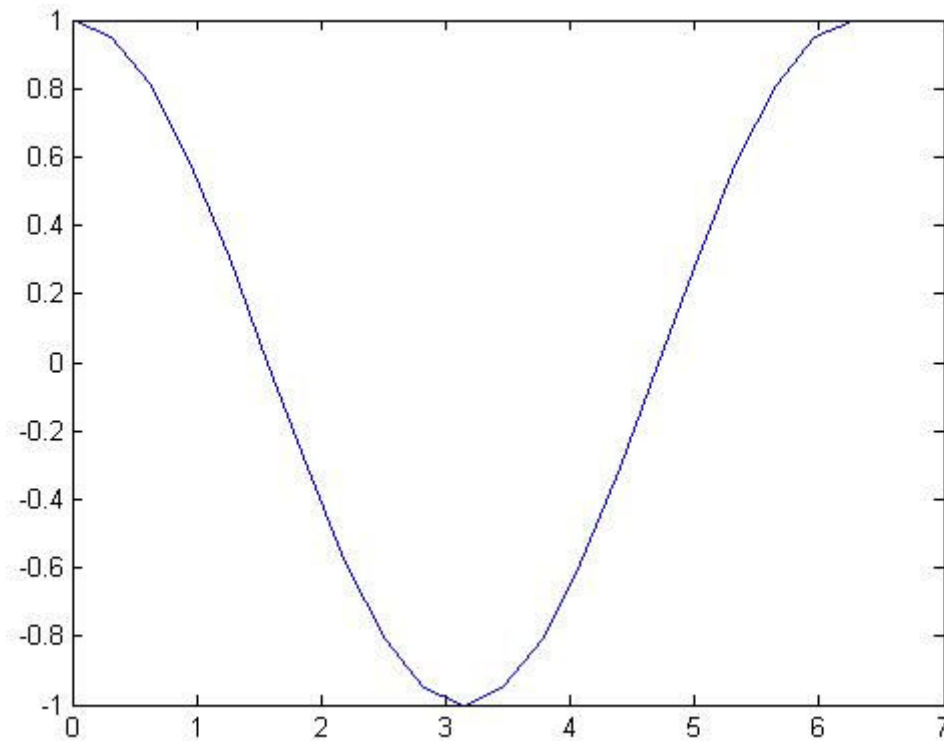
- Ogni asse può avere un'etichetta o titolo dell'asse, questo titolo assegna il nome all'asse e descrive la quantità rappresentata lungo l'asse.
- Solitamente un diagramma ha anche un titolo che ne descrive il contenuto; questo titolo è posto nella parte superiore del diagramma;
- Un diagramma può essere ottenuto da dati sperimentali o da un'equazione.
- Quando i dati vengono riportati nel diagramma, ogni dato viene rappresentato con un simbolo o marcatore;

- MATLAB possiede numerose capacità grafiche che consentono di rappresentare dati memorizzati in vettori e matrici.
- Il grafico bidimensionale di una funzione $y = f(x)$ di variabile x , può essere realizzato con il comando `plot` la cui sintassi nella forma più semplice è `plot(x,y)` essendo x ed y due vettori di ugual lunghezza.
- La rappresentazione di grafici in scala logaritmica sull'asse x , sull'asse y o su entrambi può essere realizzata rispettivamente con i comandi `semilogx(x,y)`, `semilogy(x,y)`, `loglog(x,y)`.

- Esempio Supponiamo di volere visualizzare in forma grafica una successione di valori della funzione $y = \cos(x)$ sull'intervallo $[0; 2\pi]$.
- Per prima cosa dovremo creare due vettori x e y contenenti rispettivamente la successione di valori nell'intervallo ed i corrispondenti valori della funzione.

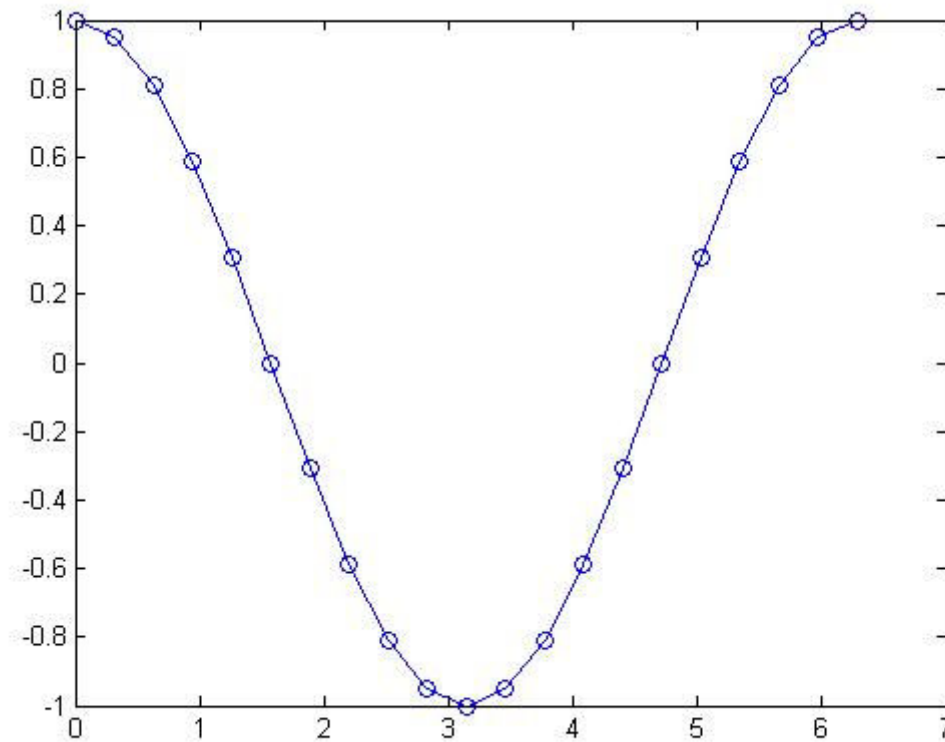
```
>> n = 21;  
>> x = linspace(0,2*pi,n);  
>> y = cos(x);
```
- dove in questo caso si sono scelti 21 punti equispaziati.
- Per realizzare il grafico basta aggiungere il comando

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



- In questo modo si ottiene una rappresentazione grafica ottenuta raccordando con segmenti di retta nel piano xy i vertici $(x(i); y(i))$ in modo ordinato al variare di i da 1 a n .

- La scelta della scala di visualizzazione è automatica.
- Se si vogliono evidenziare i vertici della poligonale così costruita si può utilizzare:
`>> plot(x,y,'-o')`



- A volte i simboli dei dati sono collegati da una linea per agevolare l'interpretazione dei dati, specialmente se i dati del diagramma sono pochi;
- Quando vengono rappresentate più curve o serie di dati nello stesso diagramma, deve essere possibile distinguerle.
- Un metodo per farlo consiste nell'utilizzare una legenda che mette in relazione il simbolo della serie di dati o il tipo di curva con le quantità rappresentate nel diagramma;

Opzioni per il comando plot

colore	significato	simbolo	significato	linea	significato
w	bianco	.	punto	-	continua
y	giallo	o	circoletto	:	punteggiata
r	rosso	x	per	-.	tratto-punto
g	verde	+	più	--	tratteggiata
b	blu	*	asterisco		
k	nero	s	quadrato		

- La sintassi di base del comando plot è quindi
`plot(Vettore1, Vettore2, Opzioni)`
- dove Vettore1 e Vettore2 sono i vettori di dati di ugual lunghezza;
- Opzioni è una stringa opzionale che definisce il tipo di colore, di simbolo e di linea usato nel grafico.
- E' possibile inoltre commentare il grafico aggiungendo le istruzioni
 - >> `title('Grafico della funzione cos(x)')`
 - >> `xlabel('x')`
 - >> `ylabel('y')`
- che consentono di inserire informazioni sul titolo del grafico e sugli assi x e y.

Comandi per commentare un grafico

Comando	Descrizione
title	Inserisce un titolo nel grafico
xlabel	Inserisce un nome per l'asse x
ylabel	Inserisce un nome per l'asse y
grid	Inserisce una griglia sugli assi x ed y
legend	Inserisce una legenda per identificare rappresentazioni diverse
text	Inserisce una stringa di testo in una specifica posizione
gtext	Inserisce una stringa di testo in una posizione individuata tramite mouse

Diagrammi multipli e sovrapposti

- MatLab è in grado di creare grafici che contengono più diagrammi distinti o diagrammi sovrapposti, si tratta di diagrammi particolarmente utili per confrontare gli stessi dati rappresentati in tipi di assi differenti.
- Il comando `subplot(m,n,p)` suddivide la finestra grafica di MATLAB in una serie di pannelli rettangolari disposti su m righe e n colonne. La variabile p indica a MATLAB di porre l'output del comando `plot`, che segue il comando `subplot`, nel p -esimo pannello.

Esempio (Grafici in sottofinestre)

- Per disegnare in diverse sottofinestre i grafici delle funzioni $f(x)=\exp(-x^2)\cos(k\pi x)$ sull'intervallo $[-2; 2]$ al variare di $k=1,2,3,4$:

```
>> x=linspace(-2,2);  
>> y=exp(-x.^2).*cos(pi*x);  
>> subplot(2,2,1);  
>> plot(x,y); title('k=1');  
>> y=exp(-x.^2).*cos(2*pi*x);  
>> subplot(2,2,2);  
>> plot(x,y); title('k=2');  
>> y=exp(-x.^2).*cos(3*pi*x);  
>> subplot(2,2,3);  
>> plot(x,y); title('k=3');  
>> y=exp(-x.^2).*cos(4*pi*x);  
>> subplot(2,2,4);  
>> plot(x,y); title('k=4');
```

- Esempio Grafici sovrapposti delle funzioni seno e coseno sull'intervallo $[0; 2\pi]$ nella stessa finestra grafica
- Dopo aver costruito i vettori di valori per le due funzioni tramite i comandi:

```
>> x = linspace(0,2*pi);
```

```
>> y1 = cos(x);
```

```
>> y2 = sin(x);
```

- Per disegnarne i grafici sovrapposti si può scrivere

```
>> plot(x,y1,'-')
```

```
>> hold on
```

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

```
>> hold off
```


- Il comando `hold on` fa sovrapporre tutti i grafici successivi nella finestra grafica.
- Il comando `hold off` ritorna all'impostazione originale.
- Lo stesso risultato, con l'aggiunta della legenda, può essere ottenuto tramite la singola istruzione
`>> plot(x,y1,'-',x,y2,'--'),legend('Coseno', 'Seno')`

