

UNIVERSITA' DEGLI STUDI DI PAVIA

FACOLTA' DI INGEGNERIA

**Corso di Identificazione dei Modelli e Analisi dei Dati**

Prof. G. De Nicolao e dott. G. R. Marseglia

---

**Introduzione a MATLAB**



# Accesso Account

---

**E' possibile loggarsi alle postazioni in C2-C3 tramite le seguenti credenziali:**

- 1. Username: cl123456 (123456 rappresenta il numero di matricola dello studente)**
- 2. Password: inf56 (56 rappresentano gli ultimi due numeri della matricola)**
- 3. Gli account dovrebbero già esistere dal primo anno (stesse credenziali usate per il corso di Fondamenti di Informatica)**
- 4. Per eventuali problemi di account recarsi in C4**

# Cos' è MATLAB

---

**MATLAB** è un programma interattivo che utilizza un linguaggio di alto livello e si è imposto in ambito ingegneristico mondiale come efficace strumento di calcolo e simulazione.

Il nome deriva da **MAT**rix **LAB**oratory: l'elemento base è la matrice.

Il software è corredato di una famiglia di soluzioni specifiche per varie applicazioni denominate Toolbox. Si tratta di insiemi di funzioni **MATLAB** aggiuntive che estendono l'ambiente di lavoro per risolvere particolari categorie di problemi quali ad esempio sistemi di controllo, matematica finanziaria, acquisizione e elaborazione di immagini...

# Cos' è MATLAB

---

E' possibile scaricare Matlab gratuitamente come presentato sulla pagina:

<http://news.unipv.it/?p=16675>

1. **Creare un Account MathWorks usando il proprio indirizzo e-mail universitario**
2. **Associare la Licenza usando il Codice di Attivazione di unipv**
3. **Fare il download per la release attuale.**
4. **Installare la versione**
5. **Attivare Matlab tramite l'utente appena creato**

# Cos' è MATLAB

---

Il modo più elementare per usare **MATLAB** è quello di utilizzarlo come una calcolatrice per fare operazioni quali somme, sottrazioni, moltiplicazioni, divisioni e potenze (+, -, \*, /, ^).

```
>> (4*10+8)/3
```

Il punto e virgola ; al termine di un'istruzione impedisce la visualizzazione del risultato dell'istruzione stessa.

**MATLAB** è un interprete di comandi che vengono eseguiti nell'ordine in cui sono impartiti.

# Variabili ed espressioni

---

Le istruzioni MATLAB sono nella forma

*variabile = espressione*

o semplicemente

*espressione*

Le espressioni sono costituite da operatori (+ \* - / ^) , funzioni, variabili e numeri

- funzioni: nomi simbolici con argomenti fra parentesi: `eye(2)`
- numeri: reali, ad es. 5, e complessi, ad es. `5 + 2*i` o indifferentemente `5 + 2*j`

Quando una variabile viene definita, essa viene memorizzata nel *workspace*.

# Definizione di vettori e matrici

---

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0; \\ 0 & 2 & 0; \\ 0 & 0 & 3 \end{bmatrix};$$

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1; \\ 2; \\ 3 \end{bmatrix};$$

$$A = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

# Funzioni utili per la definizione di matrici

---

**A = eye (3) ;** costruisce la matrice identità di dimensione 3,

**A = ones (3 , 2) ;** costruisce una matrice di uni di dimensione 3, 2

**A = zeros (4 , 5) ;** costruisce una matrice di zeri di dimensione 4, 5



# Esercizio 1: Matrici

---

Introdurre le matrici  $A = \begin{pmatrix} -1 & 3 & 2 \\ 5 & -8 & 6 \\ 1 & 0 & 1 \end{pmatrix}$   $B = \begin{pmatrix} 1 & 2 & 6 \\ 3 & 4 & 0 \\ -1 & -1 & 2 \end{pmatrix}$

e calcolare:  $A'$ ,  $A+B$ ,  $A-B$ ,  $4*A$ ,  $A*B$ ,  $A^2$ .

# Ancora sui vettori

---

Esistono vari modi per generare dei vettori

```
>> x=1:5
```

```
x =
```

```
1 2 3 4 5
```

```
>> x=1:0.5:2
```

```
x =
```

```
1.0000 1.5000 2.0000
```

# Ancora sulle matrici

---

- A(i, j)** seleziona l'elemento di posto (i,j),
- A(:, j)** seleziona tutta la colonna j-esima di A,
- A(i, :)** seleziona tutta la riga i-esima di A,
- A(:, end)** seleziona tutta l'ultima colonna di A,
- A(end, :)** seleziona tutta l'ultima riga di A,
- A=[]** crea una matrice A vuota o cancella il suo contenuto,
- v(i)** seleziona l' elemento i-esimo del vettore v.
- [m, n]=size(A)** numero di righe (n) e colonne (m) di A
- m=length(v)** numero di elementi del vettore v

# Operazioni elemento per elemento

---

Sono precedute da un punto : `.+` `.*` `.^` `./`

Se applicate a vettori (o matrici) agiscono su ogni elemento del vettore

```
>> x=1:5
```

```
x =
```

```
1 2 3 4 5
```

```
>> y=x.^2
```

```
y =
```

```
1 4 9 16 25
```

## Esercizio 2: Matrici

---

Si definisca la matrice:

$$A = \begin{pmatrix} -1 & 3 & 2 \\ 5 & -8 & 6 \\ 1 & 0 & 1 \end{pmatrix}$$

Sostituire alla seconda colonna della matrice A una colonna di uni e all'ultima riga una riga di zeri. Usare a tal fine la funzione `size`.

## Esercizio 3: Matrici

---

Si definisca il vettore:

$$A = \begin{pmatrix} 3 \\ -8 \\ 1 \end{pmatrix}$$

Si ottenga la matrice qui sotto utilizzando le istruzioni ones e  $\wedge 2$

$$B = \begin{pmatrix} 1 & 3 & 9 \\ 1 & -8 & 64 \\ 1 & 1 & 1 \end{pmatrix}$$

# Funzioni matematiche

---

Seno	<code>sin(z)</code>	<code>sind(z)</code>
Coseno	<code>cos(z)</code>	<code>cosd(z)</code>
Tangente	<code>tan(z)</code>	<code>tand(z)</code>
Arcotangente	<code>atan(y)</code>	<code>atand(y)</code>
Esponenziale	<code>exp(x)</code>	
Logaritmo naturale	<code>log(x)</code>	
Logaritmo in base 10	<code>log10(x)</code>	
Radice quadrata	<code>sqrt(x)</code>	
Distribuzione normale	<code>normpdf(x, mu, sigma)</code>	

Se gli argomenti sono vettori, le funzioni vengono applicate ad ogni singolo elemento (e restituiscono un vettore)

# Grafici

---

- l'istruzione `plot` consente di effettuare grafici bidimensionali.

`plot(x,y)`

con `x` e `y` vettori  $1 \times n$  produce un grafico con `x` in ascissa e `y` in ordinata;

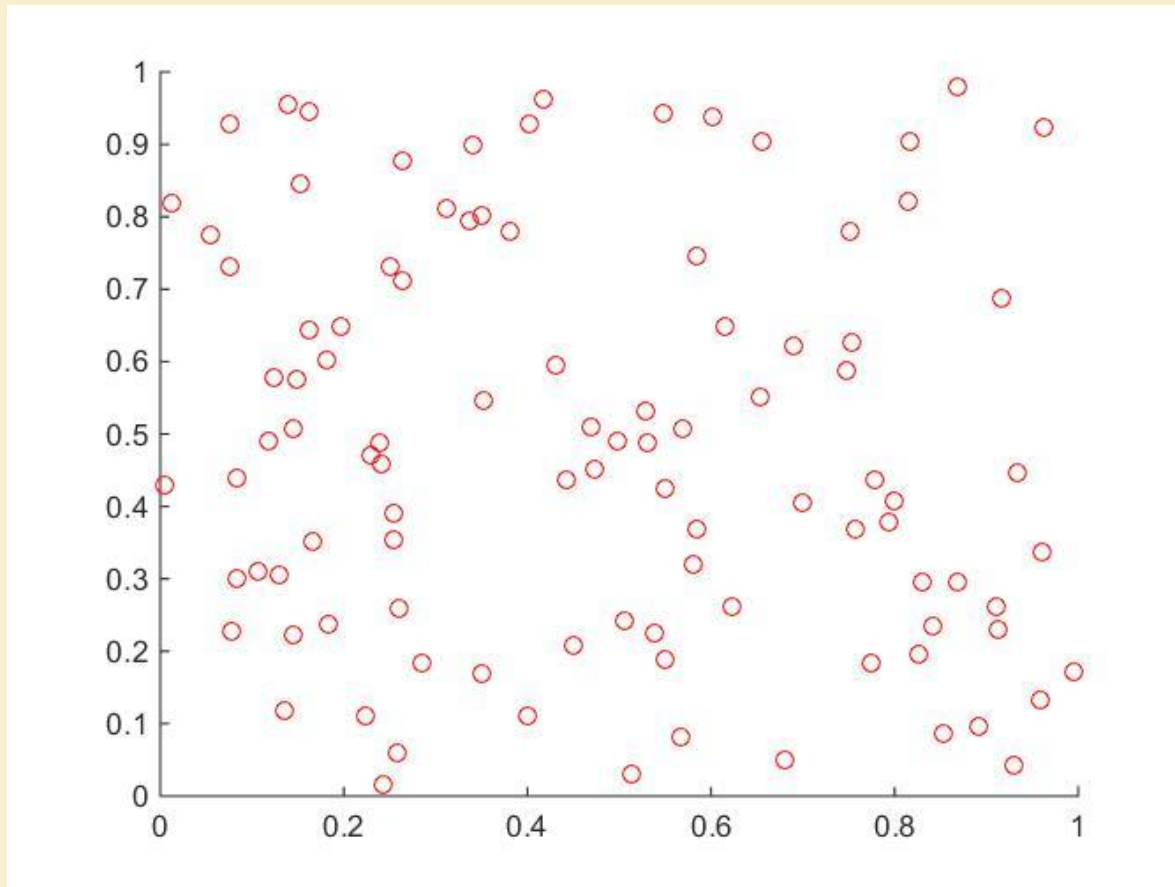
- l'istruzione `grid` visualizza una griglia sullo schermo;
- le istruzioni `title('titolo')`, `xlabel('label')` e `ylabel('label')` consentono di mettere il titolo e delle etichette sull'asse `x` e `y`, rispettivamente;
- l'istruzione `axis([xinit xend yinit yend])` consente di riscalarlo il grafico.



# Grafici

---

- l'istruzione **scatter(x,y)** consente di generare degli scatter plot con i punti nelle posizioni specificate dai vettori (x,y)



## Esercizio 4

---

Disegnare il grafico di  $y = \sin(x)$  e di  $y = \cos(x)$  fra 0 e 4 sulla stessa finestra in modo da poterli confrontare.

Per evitare la cancellazione di un grafico da parte di quello successivo, fra i diversi comandi `plot` va interposto il comando `hold on`.

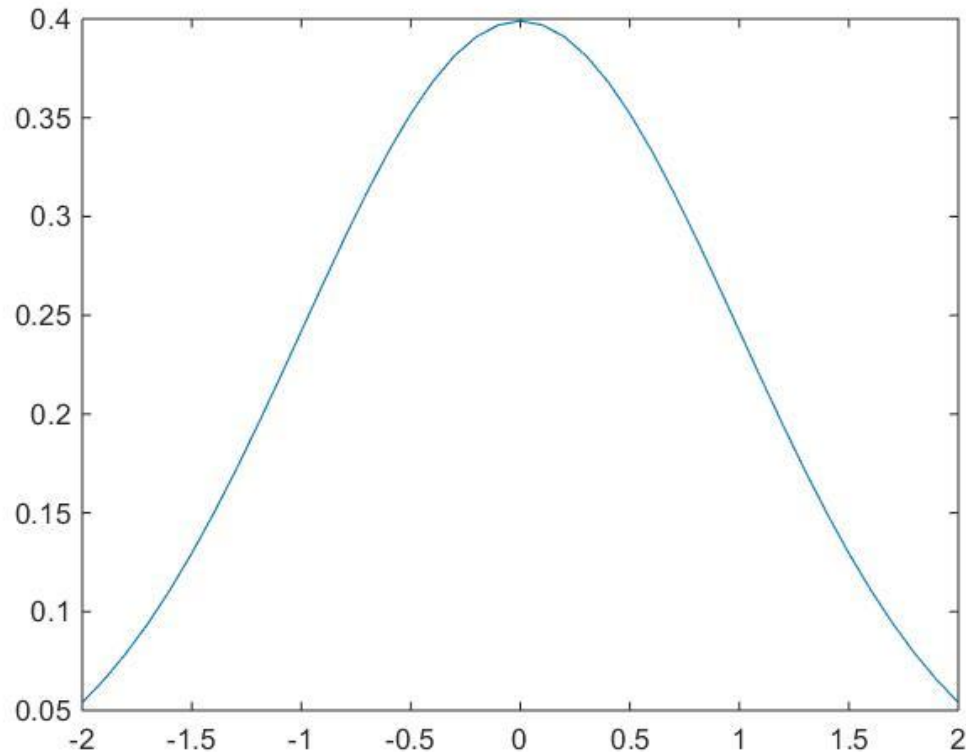
Se invece si desidera tracciare i grafici in differenti figure è necessario premettere all'istruzione `plot` il comando `figure(n)` che apre la figura numero n (o la seleziona se è già aperta).

Per cancellare la figura corrente: `clf`

## Esercizio 5

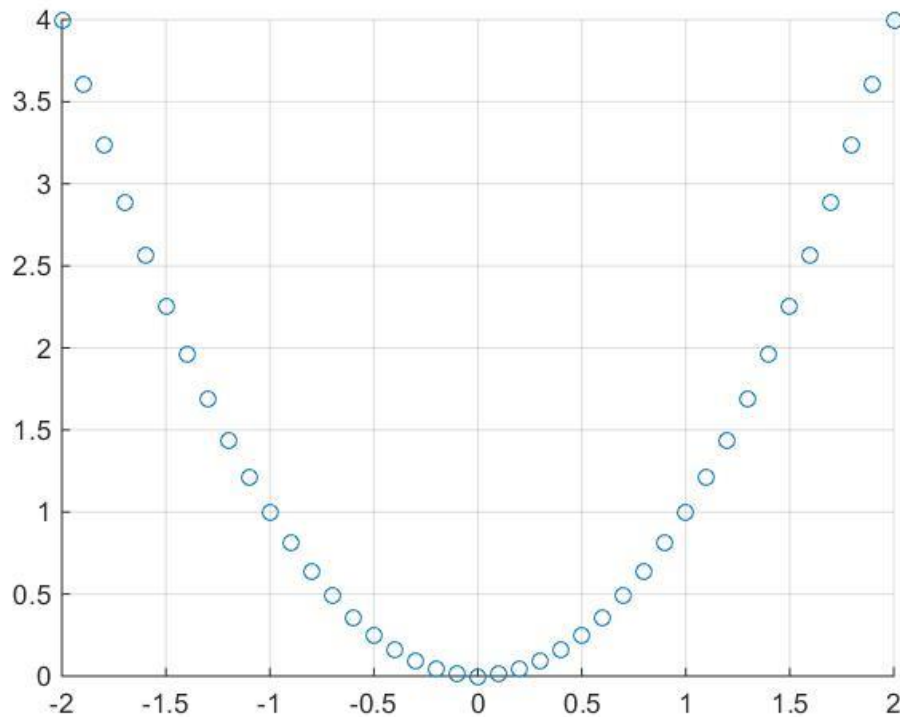
---

Disegnare il grafico della distribuzione normale con media zero e varianza unitaria tra -2 e 2. Utilizzare a tal fine la funzione `normpdf`



# Esercizio 6

Disegnare uno scatter plot che rappresenti  $y=x^2$



# Modifica del tipo e del colore della linea

È possibile far tracciare il grafico con linee di colori e tratti differenti da quelli di default usando la sintassi

```
plot(x,y,'colore_linea_simbolo')
```

'colore\_linea\_simbolo' è una particolare stringa formata da uno a quattro caratteri in combinazione fra quelli riportati in tabella.

Simbolo		Colore		Linea	
+	più	c	ciano	-	linea continua
o	circoletto	m	magenta	--	linea tratteggiata
*	asterisco	y	giallo	:	linea punteggiata
x	per	r	rosso	-.	linea tratto punto
s	quadrato	g	verde		
d	diamante	b	blu		
^	triangolo	w	bianco		
v	triangolo	k	nero		
>	triangolo				
<	triangolo				
'p'	pentagono				
'h'	esagono				

# M-files

---

I files che contengono istruzioni eseguibili da MATLAB sono detti m-files in quanto hanno come suffisso .m

- Se un m-file memorizza una sequenza di istruzioni è detto scriptfile.

Se contiene una nuova funzione viene detto function file.

- Un function file:

```
function y=linear(x,alfa,beta)
% questo file restituisce una
% trasformazione affine del vettore x;
y= alfa + beta *x;
```

```
>> x=[1 4 5 9];
```

```
>> linear(1,4,x)
```

```
ans =
```

```
5 8 9 13
```

# Cicli

---

MATLAB possiede le usuali istruzioni di controllo che consentono di effettuare programmazione:

**for** condizione ...istruzioni ... **End**

**while** condizione ...istruzioni ... **end**

**if** condizione ...istruzioni ... **else** ... istruzioni **end**

```
% script file sommatoria.m Calcola la somma
% a1+a2+ ... + a9 ove ai=i^2/(i+1)
s=0;
for i=1:9
    s=s+i^2/(i+1);
end
s
```


```
>> sommatoria
```

```
s =
37.9290
```

## Esercizio 7

---

Scrivere una funzione che calcoli il quadrato di un numero.  
Richiamare la funzione in Matlab per calcolare il quadrato di un intero vettore utilizzando un ciclo for.

The bottom right portion of the slide features several overlapping, wavy, light-colored lines that create a sense of motion and depth, serving as a decorative background element.



# Interpretazione dei comandi MatLab

---

Quando l'interprete di MATLAB trova un nome, ad esempio "prova", segue, nell'ordine, i seguenti passi:

1. cerca nel workspace la variabile di nome prova
2. cerca una funzione built-in di nome prova
3. cerca un m-file di nome prova nella directory da cui si è lanciato MATLAB
4. cerca un m-file di nome prova nel path indicato in una speciale variabile di sistema detta MATLABPATH. La variabile MATLABPATH può essere aggiornata (il modo dipende dal sistema operativo e dalla versione di MatLab)

# Comandi utili

---

**who** visualizza l'elenco delle variabili presenti nel workspace

**clear all** cancella tutte le variabili in memoria,

**clear nome\_variabile** cancella la variabile indicata,

**save nome\_file** salva le variabili in memoria e i dati in esse contenuti nel file specificato,

**load nome\_file** richiama in memoria le variabili e i dati contenuti nel file specificato,

**cd** permette di cambiare directory,

**dir** o **ls** esamina il contenuto della directory corrente

# I comandi più utili !

---

**help** richiama l' help in linea.

**help <nome istruzione>** richiama l' help per una specifica istruzione

**doc** finestra della documentazione (solo nelle versioni più recenti di MatLab)

... intere generazioni di ricercatori hanno imparato MatLab analizzando m-files e imparando nuove istruzioni con il comando help !