



# FONDAMENTI DI AUTOMATICA

MATLAB

Introduzione



## Introduzione

- **MATLAB** è un linguaggio di alto livello e un ambiente interattivo per il calcolo numerico, la visualizzazione e la programmazione prodotto dalla Mathworks.
- Il linguaggio, gli strumenti e le funzioni matematiche integrate permettono di risolvere problemi in maniera intuitiva e più "semplice" rispetto ai tradizionali linguaggi di programmazione quali il C/C++ o il Java.
- MATLAB può essere usato in numerosi campi applicativi: processamento di segnali e telecomunicazioni, analisi di immagini e video, sistemi di controllo, modellazione, diagnosi, test, misura, finanza, biologia, ecc ...

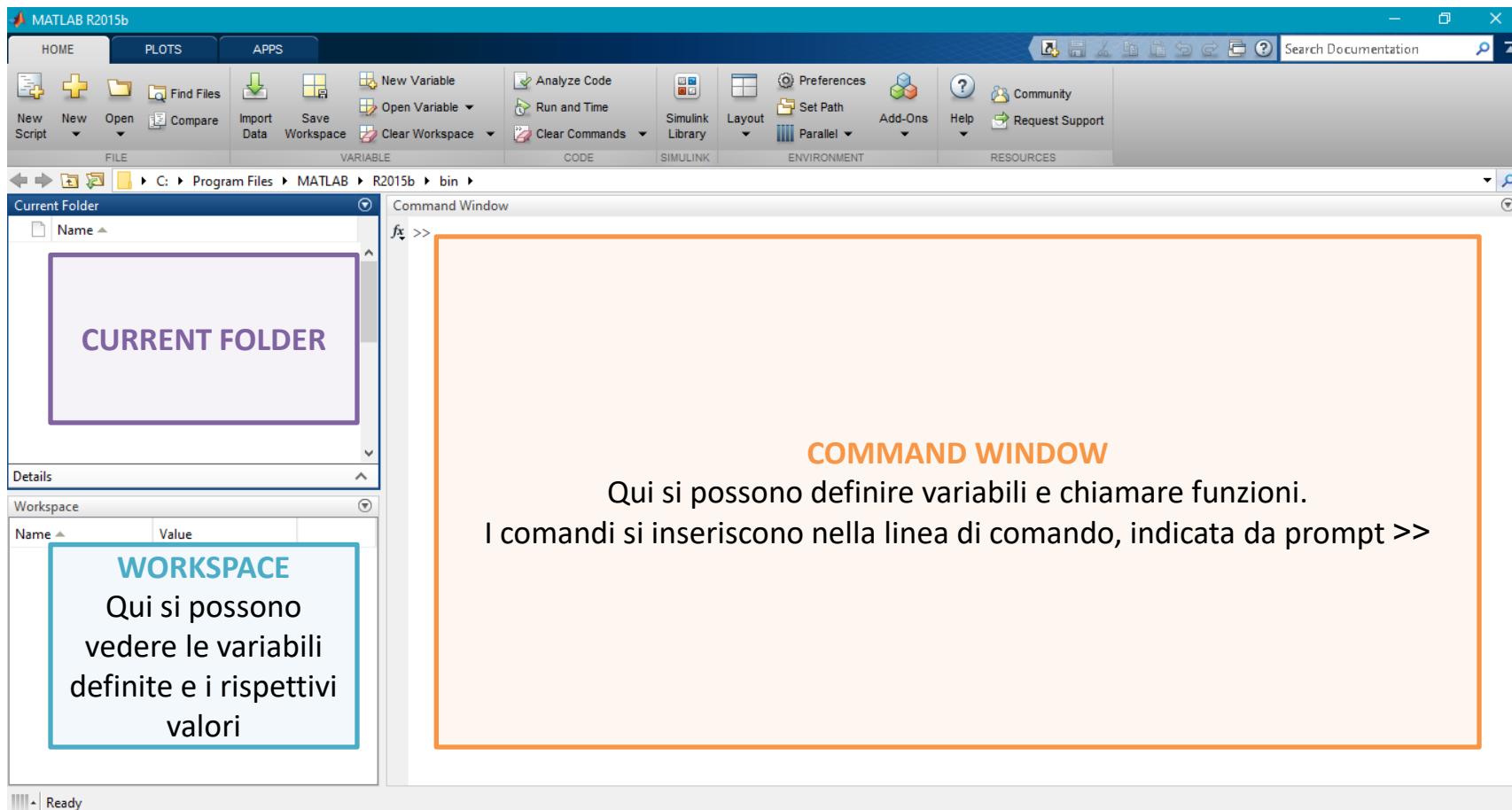


## Introduzione

- Caratteristiche chiave:
  - linguaggio di alto livello per il calcolo numerico, la visualizzazione e lo sviluppo di applicazioni;
  - ambiente interattivo per l'esplorazione iterativa, la visualizzazione e la risoluzione di problemi;
  - funzioni matematiche per l'algebra lineare, la statistica, l'analisi di Fourier, il filtraggio, l'ottimizzazione, l'integrazione numerica e la risoluzione di equazioni differenziali ordinarie;
  - grafica integrata per la visualizzazione di dati e strumenti di creazione di grafici;
  - strumenti di sviluppo per il miglioramento della qualità e manutenibilità del codice;
  - strumenti per lo sviluppo di applicazioni con interfacce personalizzabili;
  - funzioni per l'integrazione di algoritmi sviluppati in MATLAB con linguaggi di applicazione esterna, quali C, Java, .NET e Microsoft Excel.

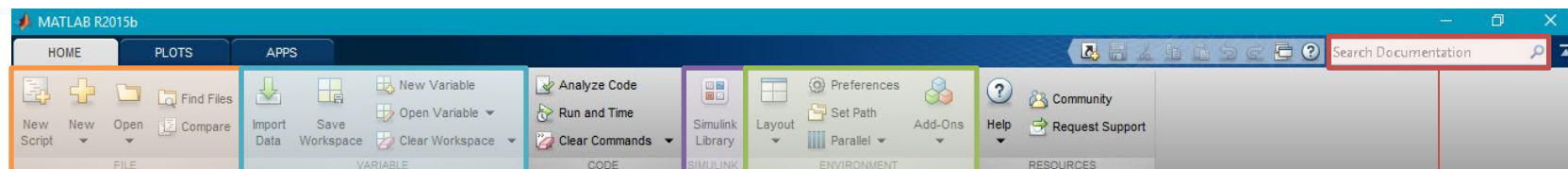


## Interfaccia grafica





## Barra dei comandi



### [FILE]

Aprire/creare  
file MATLAB

### [SIMULINK]

Consente di aprire  
la libreria  
SIMULINK

### [DOCUMENTATION]

Permette di cercare e  
accedere alla  
documentazione completa

### [VARIABLE]

È possibile creare/ aprire/ salvare/  
importare variabili

- Clear Workspace: consente di pulire le variabili e le funzioni definite
- Save Workspace: consente di salvare le variabili definite e i relativi valori

### [ENVIRONMENT]

Consente di gestire

- Preferenze
- Aggiungere cartelle all'interno del quale MATLAB può cercare file
- Cambiare l'aspetto dell'interfaccia grafica



## Comandi

- All'avvio di MATLAB, nella sezione di inserimento dei comandi (i.e. la *Command Window*), appare il prompt ">>" all'interno della quale eseguire i **comandi**. Vi sono due tipi di comandi:
  - **assegnamenti** | Il comando  
  
    >> variabile = espressione;  
  
    asigna il valore "espressione" a "variabile"
  - **valutazione di espressioni** | Il comando  
  
    >> espressione  
  
    genera una matrice che viene assegnata alla variabile indicata.
- Quando nell'istruzione non si specifica la variabile a cui assegnare il risultato, la valutazione dell'espressione viene assegnata alla **variabile di sistema** **ans** (abbreviazione di "answer").

### Esempio

```
>> 1+1  
ans =  
      2  
>> a=3*ans;  
>> a+1  
a =  
      7
```



## Comandi

- Notare che
  - Se un'espressione non termina con il punto e virgola il risultato della sua valutazione viene mostrato anche sullo schermo.
  - E' possibile richiamare comandi mediante le frecce direzionali, su linea vuota oppure dopo aver specificato l'iniziale del comando.
  - In MATLAB le variabili non devono essere dichiarate.
  - MATLAB è case-sensitive: la variabile a è diversa dalla variabile A.
  - MATLAB è l'abbreviazione di *MATrix LABoratory*: tutte le variabili di MATLAB sono array multidimensionali, a prescindere dal tipo di dato in essi contenuto.





# Vettori

- Un **vettore riga** (array monodimensionale) può essere definito inserendo gli elementi tra parentesi quadre, separati da una virgola oppure da uno spazio:

```
>> a = [1, 2, 3]          >> a = [1 2 3]
a =
    1  2  3                a =
    1  2  3
```

Un **vettore colonna** può esser definito inserendo un punto e virgola tra i valori:

```
>> a = [1; 2; 3]
a =
     1
     2
     3
```

- Il comando `x : step : y` crea un vettore riga di estremi  $x$  e  $y$ ; il parametro `step` è opzionale e indica l'intervallo tra ciascun elemento del vettore (valore di default pari a 1):

```
>> a = 0:2:6          >> a = 0:6
a =
    0  2  4  6        a =
    0  1  2  3  4  5  6
```





# Matrici

- Una matrice (array multidimensionale) può esser definite in diversi modi: i seguenti comandi generano la stessa matrice:

```
A = [  
    [1; 2; 3];  
    [4; 5; 6];  
];  
A = [[1 2 3]; [4 5 6]];  
A = [[1; 4], [2; 5], [3; 6]];  
A = [1, 2, 3; 4, 5, 6];
```

- Per definire matrici particolari è possibile usare funzioni già definite:

- |  |  |
|--|--|
| ■ Matrice identità di dimensione $N$ :                   | <code>&gt;&gt; A = eye(N);</code>              |
| ■ Matrice con $n$ righe e $m$ colonne con tutti 0:       | <code>&gt;&gt; A = zeros(n, m);</code>         |
| ■ Matrice con $n$ righe e $m$ colonne con tutti 1:       | <code>&gt;&gt; A = ones(n, m);</code>          |
| ■ Matrice di numeri casuali di dimensione $n \times m$ : | <code>&gt;&gt; A = rand(n, m);</code>          |
| ■ Matrice diagonale di dimensione $n \times n$ :         | <code>&gt;&gt; A = diag([a11 ... ann]);</code> |



## Array: definizione di una matrice

- MATLAB permette di operare su tutti gli elementi di una matrice con un singolo operatore:

```
>> B = A+1
```

```
>> C = sin(A)
```

B =

a =

2 3 4

5 6 7

0.8415 0.9093 0.1411

-0.7568 -0.9589 -0.2794

- Alcune delle funzioni già implementate in MATLAB che consentono di effettuare operazioni tra matrici sono:

- Calcolo della matrice trasposta: 

```
>> D = transpose(A); >> D = A.';
```
- Calcolo della matrice inversa: 

```
>> D = inv(A);
```
- Calcolo della pseudo-inversa: 

```
>> D = pinv(A);
```
- Operazioni con uno scalare  $k$ : 

```
>> D = A(+,-,*,/ )k;
```



# FONDAMENTI DI AUTOMATICA

## MATLAB

### Operazioni e Funzioni elementari



## Array: accesso ai singoli elementi (indexing)

- Data la matrice  $A$  di dimensione  $n \times m$  è possibile accedere/estrarre i singoli elementi mediante i seguenti comandi:
  - `>> A(i,j)` elemento  $(i,j)$  della matrice  $A$
  - `>> A(k)`  $k$ -esimo elemento, contato in ordine colonna, della matrice  $A$
  - `>> A(n,:)`  $n$ -esima riga della matrice  $A$
  - `>> A(:,m)`  $m$ -esima colonna della matrice  $A$
- Se si prova ad accedere ad un elemento al di fuori delle dimensioni  $n$  o  $m$  si ottiene un messaggio di errore:

```
>> A = [1,2,3;4,5,6];
```

```
>> size(A)
```

```
ans =
```

```
2     3
```

```
>> A(3,3)
```

**Index exceeds matrix dimensions.**



## Workspace

- *Workspace* contiene le variabili che l'utente crea o importa in MATLAB.
- Le variabili nel workspace sono visualizzabili con il comando `whos` oppure nel pannello workspace presente sul desktop
- Le variabili presenti nel workspace non rimangono all'uscita del MATLAB: pertanto è necessario salvare i dati mediante il comando `save` prima di uscire, il quale salva tutte le variabili contenute nel workspace in un file di estensione `.mat`; i dati così salvati sono poi recuperabili al successivo avvio di MATLAB mediante il comando `load`:  

```
>> save fileName.mat  
>> load fileName.mat
```

Name	Value
A	[1 2 3; 4 5 6]
B	[2 3 4; 5 6 7]

```
>> clear all  
>> A=[1 2 3; 4 5 6]  
  
A =  
  
     1     2     3  
     4     5     6  
  
>> whos  
Name      Size      Bytes  Class  Attributes  
A         2x3         48  double  
  
>> B=A+1  
  
B =  
  
     2     3     4  
     5     6     7  
  
>> whos  
Name      Size      Bytes  Class  Attributes  
A         2x3         48  double  
B         2x3         48  double  
  
>>
```

- Per cancellare le variabili dal workspace si usa il comando `clear`.



## Stringhe di caratteri

- Una **stringa** di caratteri è una **qualunque sequenza di caratteri racchiusa tra apici**: si può assegnare una stringa ad una variabile.

```
>> s = 'Hello, World'
```

```
s =
```

```
Hello, World
```

- Una variabile contenente una stringa è sempre un array in MATLAB, la cui classe (o tipo di dato) è `char` (abbreviazione di *character*).
- Per tradurre valori numerici in stringhe è possibile avvalersi delle funzioni integrate di MATLAB, quale ad esempio **num2str**.
- È possibile concatenare due stringhe usando il comando **strcat**.



## Chiamare le funzioni

- MATLAB fornisce un ampio numero di funzioni che eseguono elaborazioni.
- Le funzioni sono equivalenti a "subroutines" o "metodi" in altri linguaggi di programmazione.
- Per richiamare una funzione, ad esempio `max`, è necessario racchiudere i suoi argomenti tra parentesi tonde.
- Se la funzione ha più argomenti è necessario separarli con la virgola.
- Per memorizzare il valore di uscita di una funzione lo si assegna ad una variabile.
- Se la funzione restituisce uscite multiple, allora si assegnano tali uscite ad un vettore riga.

```
>> A = [1 3 5];  
max(A)
```

```
ans =  
    5
```

```
>> B = [10 6 4];  
max(A,B)
```

```
ans =  
    10     6     5
```

```
>> maxA = max(A)
```

```
maxA =  
    5
```

```
>> [maxA,location] = max(A)
```

```
maxA =  
    5
```

```
location =  
    3
```





## Operatori e funzioni matematiche elementari per scalari

- I principali **operatori aritmetici** presenti in MATLAB sono:

- +** e **-** somma e differenza;
- \*** e **/** prodotto e quoziente,
- ^** elevamento a potenza.

```
>> x = 2*((3+2-4)^2/13)
```

```
x =
```

```
0.1538
```

- Le **funzioni matematiche** elementari in MATLAB sono:

- abs** modulo (anche di un numero complesso);
- angle** fase di un numero complesso;
- conj** complesso coniugato;
- exp** elevamento a potenza in base  $e$ ;
- real** parte reale di un numero complesso;
- imag** parte immaginaria di un numero complesso;
- log** logaritmo naturale;
- log10** logaritmo in base 10;
- sqrt** radice quadrata.

```
>> z = 10 + 4i
```

```
z =
```

```
10.0000 + 4.0000i
```

```
>> Z = log10(real(z))+sqrt(imag(z))
```

```
Z =
```

```
3
```



## Funzioni trigonometriche elementari per scalari e costanti

- Le principali **funzioni trigonometriche** in MATLAB sono:

▪ <code>sin</code>	seno;
▪ <code>cos</code>	coseno;
▪ <code>tan</code>	tangente;
▪ <code>asin</code>	arcoseno;
▪ <code>acos</code>	arcocoseno;
▪ <code>atan</code>	arcotangente.

```
>> id=asin(sin(1))+acos(cos(1))-sin(1)^2-  
cos(1)^2  
  
id =  
    1
```

- Le principali **costanti** definite in MATLAB sono:

▪ <code>i</code> o <code>j</code>	unità complessa;
▪ <code>e</code>	costante di Eulero;
▪ <code>pi</code>	$\pi$ ;
▪ <code>Inf</code>	$\infty$ ;
▪ <code>NaN</code>	"not a number";
▪ <code>realmax</code>	massimo numero esprimibile;
▪ <code>realmin</code>	minimo numero esprimibile;
▪ <code>eps</code>	precisione macchina.

} dipendono dal  
calcolatore

```
>> z = 1/0  
  
z =  
    Inf  
  
>> z = 0/0  
  
z =  
    NaN  
  
>> eps  
  
ans =  
    2.2204e-16
```



## Operatori e funzioni matematiche elementari per matrici

- Le funzioni elementari per matrici sono le stesse viste per gli scalari, ad eccezione di:
  - `'` trasposizione complessa coniugata;
  - `\` divisione sinistra (i.e.  $X = A \backslash B$  è soluzione dell'equazione  $A * X = B$ ).
- L'operazione di somma o di sottrazione è definita tra matrici aventi le stesse dimensioni.
- Se uno dei due operandi è uno scalare, esso viene sommato o sottratto a tutti gli elementi della matrice.
- `.*`, `./` e `.^` effettuano le corrispondenti operazioni sui singoli elementi delle matrici coinvolte.
- Le funzioni matematiche elementari e trigonometriche, quando applicate alle matrici, si riferiscono ai singoli elementi della matrice.

- Le principali funzioni per matrici sono:

- `size` dimensioni;
- `det` determinante;
- `rank` rango;
- `eig` autovalori.

<pre>&gt;&gt; A = [1,3;4,2]</pre> <pre>A =</pre> <pre>     1     3</pre> <pre>     4     2</pre>	<pre>&gt;&gt; det(A)</pre> <pre>ans =</pre> <pre>    -10</pre>	<pre>&gt;&gt; size(A)</pre> <pre>ans =</pre> <pre>     2     2</pre>
--	--	--

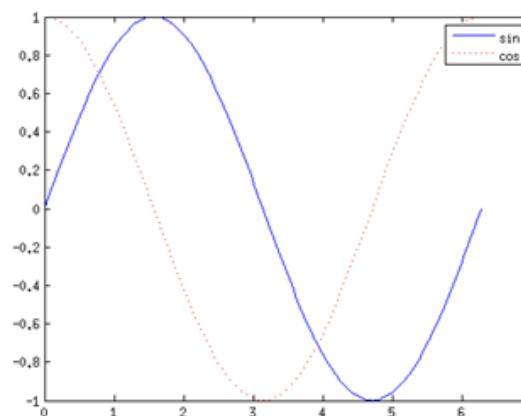
## Realizzare grafici bidimensionali

- Per creare grafici bidimensionali si usa la funzione `plot`.
- Il comando `plot` può essere integrato mediante argomenti aggiuntivi oppure ulteriori funzioni grafiche:

- `hold on`, rappresenta sullo stesso grafico più curve;
- `xlabel` e `ylabel`, etichette sugli assi  $x$  e  $y$ ;
- `grid on`, imposta una griglia;
- `'r:'`, argomento che specifica colore rosso e stile di linea;
- `legend`, crea una legenda;

- ecc ..

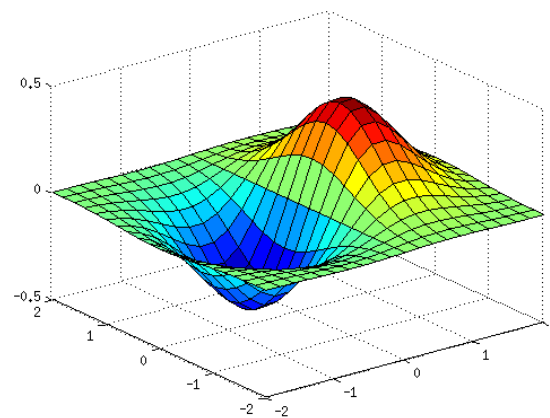
```
x = 0:pi/100:2*pi;  
y = sin(x);  
plot(x,y)  
hold on  
y2 = cos(x);  
plot(x,y2,'r:')  
legend('sin','cos')
```



## Realizzare grafici tridimensionali

- Un grafico tridimensionale raffigura tipicamente una superficie definita come funzione di due variabili, e.g.  $z = f(x,y)$ .
- Per disegnare  $z$  è prima necessario definire il set di punti  $(x,y)$  del dominio della funzione con **meshgrid** (analogamente a come si definisce in due dimensioni il dominio della funzione con un vettore riga); poi si definisce  $z$  in funzione di  $(x,y)$  e si usa la funzione **surf**.

```
>> [X,Y] = meshgrid(-2:2:2);  
Z = X .* exp(-X.^2 - Y.^2);  
surf(X,Y,Z)
```





## Programmazione e *script*

- L'esempio più semplice di programma MATLAB è detto *script*: **uno *script* è un file con estensione .m che contiene linee multiple e sequenziali di comandi e chiamate a funzione MATLAB.**
- Si può eseguire uno script digitando il suo nome nella linea di comando oppure posizionandosi su di esso con il cursore del mouse all'interno della directory corrente e cliccando esegui (o "F9" da tastiera).
- Ogni volta che si scrive codice in MATLAB è buona pratica aggiungere commenti che descrivano il codice: il simbolo che denota i commenti in MATLAB è il **%**.
- All'interno di uno script è possibile eseguire cicli di codice avvalendosi delle espressioni condizionali **for**, **while**, **if** e **switch**.





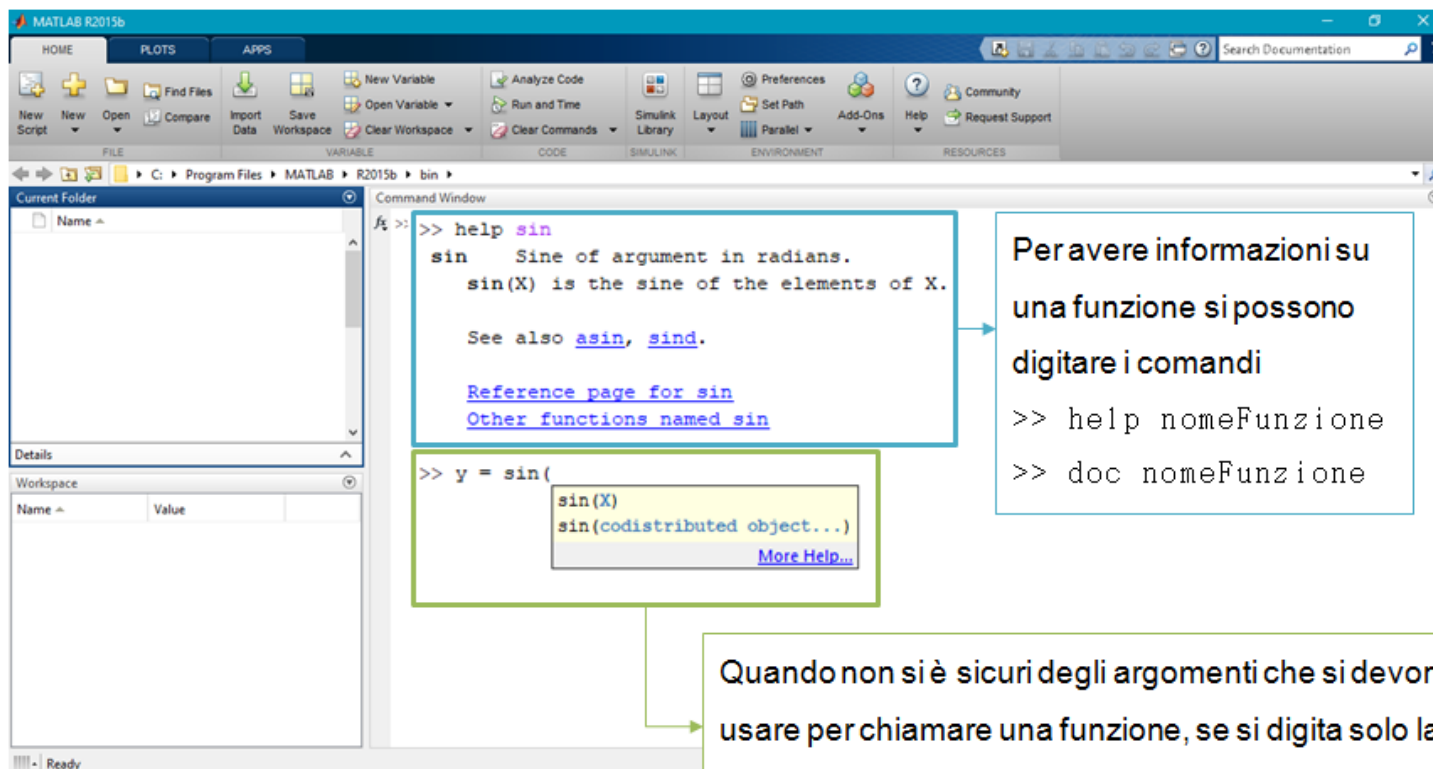
## Programmazione e *script*

- Affinché MATLAB possa eseguire uno script è necessario che esso si trovi nella directory corrente, oppure all'interno del percorso di ricerca che è modificabile dall'utente.
- Di default la directory di installazione di MATLAB è inserita nel percorso di ricerca: per utilizzare programmi installati in altre directory e aggiungerle al percorso è sufficiente selezionare la cartella di interesse, cliccare con il tasto destro del mouse e selezionare **Add to Path**.





## Help e documentazione





## Documentazione completa

The screenshot shows the MATLAB R2015b interface. The top toolbar includes a 'Search Documentation' button. Below the toolbar, the 'Current Folder' and 'Command Window' are visible. The 'Command Window' displays the search results for the function 'sin'. The results are categorized into 'FUNCTIONS', 'SYSTEM OBJECTS', and 'SEARCH SUGGESTIONS'. The 'FUNCTIONS' section lists several 'sin' functions from different toolboxes, including 'sin' from the Symbolic Math Toolbox and 'sin' from the Fixed-Point Designer. The 'SYSTEM OBJECTS' section lists objects like 'dsp.NCO System object' and 'hdlRAM System object'. The 'SEARCH SUGGESTIONS' section lists 'single', 'since', and 'sin'.

Se si vuole consultare la documentazione completa relativa ad una funzione si può digitare il nome della funzione nella barra di ricerca

Suggerimenti di ricerca



# Documentazione completa

The screenshot shows the MATLAB documentation website with a search bar at the top containing the text 'sin'. Below the search bar, there are three filters: 'Rate by Type', 'Rate by Product', and 'Rate by Category'. The 'Rate by Type' filter shows a list of categories with their respective counts: Functions (506), Blocks (88), System Objects (37), Categories (6), Examples and How To (476), Release Notes (8), and Concepts (112). The 'Rate by Product' filter shows a list of products with their respective counts: MATLAB (242), Simulink (32), Aerospace Blockset (40), Aerospace Toolbox (4), Aerospace Toolbox (4), Bioinformatics Toolbox (2), Communications System Toolbox (27), and Filter Implementation (34). The 'Rate by Category' filter shows a list of categories with their respective counts: MuPAD (457), Mathematics (107), Graphics (93), Mathematics (78), Coordinate Systems (38), Fixed-Point Design for MATLAB Code (35), and Filter Implementation (34). The main content area displays search results for 'sin', showing a list of functions and their descriptions. The results are numbered 1 through 10 of 1,731. The first result is 'sin - Sine of argument in radians', followed by 'sin - Symbolic sine function', 'sin - Sine of fixed-point values', 'sin - Sine function', 'pi - Ratio of circle's circumference to its diameter', 'Substitute Scalars with Matrices', 'sph2cart - Transform spherical coordinates to Cartesian', 'plotly - 2-D line plots with y-axes on both left and right side', 'inline - Construct inline object', and 'prg:untrace - Terminates observation of functions'. The bottom of the page contains a footer with the text '© 2007 - 2016 Università degli Studi eCampus - Via Isimbardi 10 - 22060 Novedrate (Co) - C.F. 9002752130 - Tel: 031.79421 - Fax: 031.7942501 - Mail: info@uniecampus.it'.



# FONDAMENTI DI AUTOMATICA

MATLAB

Approfondimento



## Approfondimento

- Consultare la documentazione relativa ai comandi e alle funzioni viste a lezione all'interno dell'help del MATLAB (tasto "F1" oppure prima riga della finestra **Help**)



# FONDAMENTI DI AUTOMATICA

## MATLAB

Domande aperte ed esercizi



## Domande aperte

- Che cos'è MATLAB e a cosa serve?
- Quali sono le due tipologie di comandi principali che è possibile impartire in MATLAB?
- A che cosa servono la Command Window, il Workspace e la Current Directory?
- Che cos'è un array in MATLAB?
- Come si crea un vettore in MATLAB?
- Come si crea una matrice in MATLAB?
- Come si accede agli elementi di un vettore/matrice in MATLAB?
- Come si crea un grafico in MATLAB?
- Che cos'è uno script in MATLAB?
- Come si accede alla documentazione in MATLAB?





## Esercizi

- Consultare la documentazione relativa ai comandi e alle funzioni viste a lezione all'interno dell'help del MATLAB (tasto "F1" oppure prima riga della finestra **Help**).
- Disegnare il grafico di  $\tan x$  nell'intervallo  $\left[-\frac{\pi}{2}; \frac{\pi}{2}\right]$
- Calcolare il massimo della funzione  $y = \tan x$
- Assegnare l'etichetta "x[rad]" all'asse delle  $x$
- Disegnare la parabola  $y = x^2$
- Creare la legenda
- Calcolare il minimo della funzione  $y = x^2$
- Includere tutto il codice all'interno di uno script eseguibile e commentato.