

Corso di Segnali e Sistemi
Ingegneria Biomedica-Ingegneria Elettronica
Università degli Studi di Padova
(Proff. N. Benvenuto e C. Dalla Man)

Laboratorio 01

- Introduzione a MATLAB

Cos'è MatLab

MatLab = Matrix Laboratory

Utilizza come oggetti fondamentali da gestire le **matrici**:

Una **matrice NxM** indica una matrice con N righe ed M colonne, Cioè un oggetto che contiene NxM elementi disposti ordinatamente su N righe ed M colonne, ed in cui ogni elemento è identificato dalle sue coordinate.

Matlab consente:

- L'accesso ad un ambiente di calcolo
- L'utilizzo di funzioni specializzate
- La programmazione

Perchè MatLab

- Facilità d'uso:
 - Ha moltissime funzioni disponibili
 - E' possibile programmare funzioni ad hoc
 - Non ci si deve preoccupare di programmazione a basso livello
- Esportabilità
 - Le funzioni MatLab sono file di testo

Programmare in MatLab

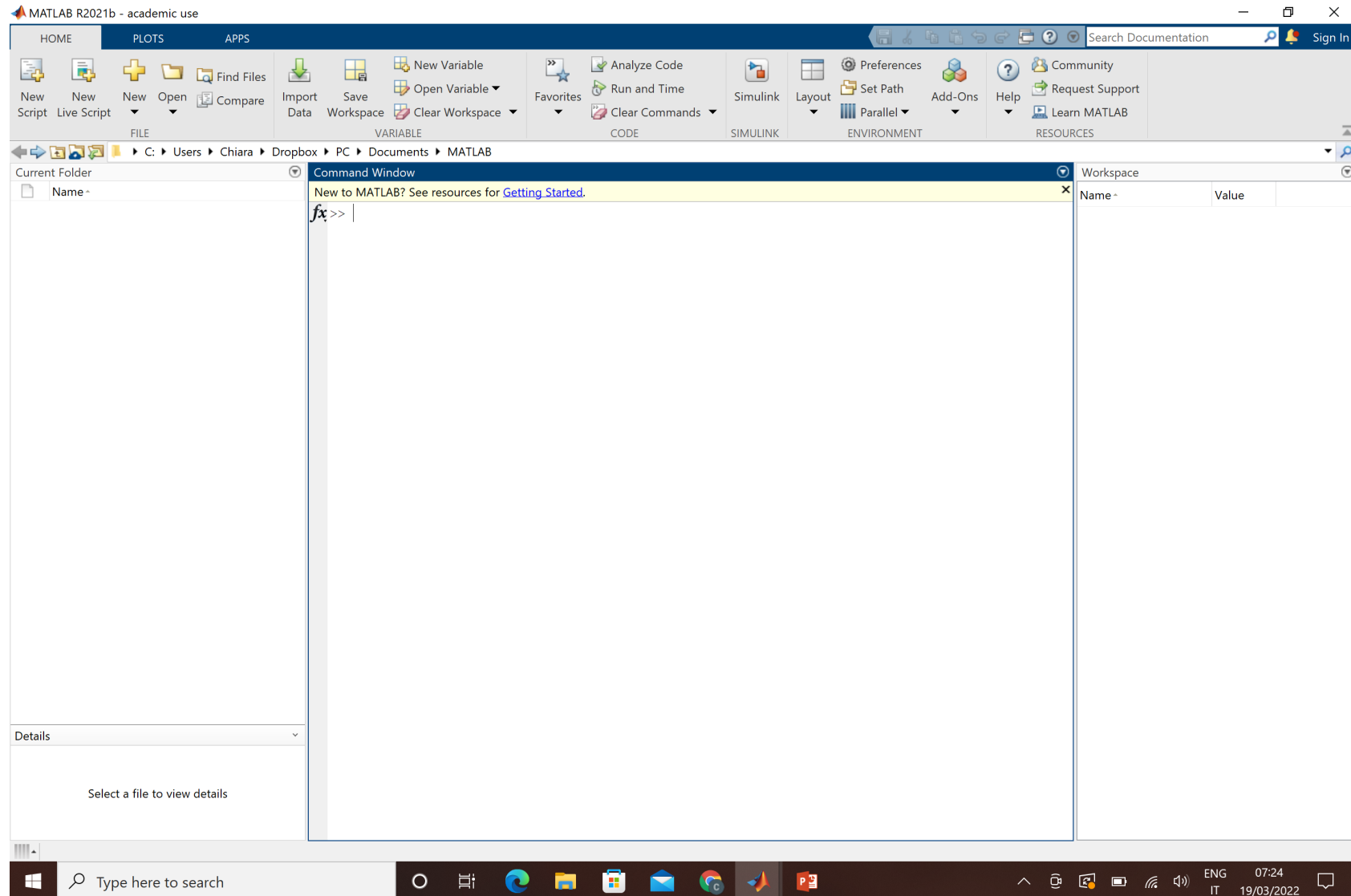
In MatLab ci sono due modi per scrivere le istruzioni e farle eseguire:

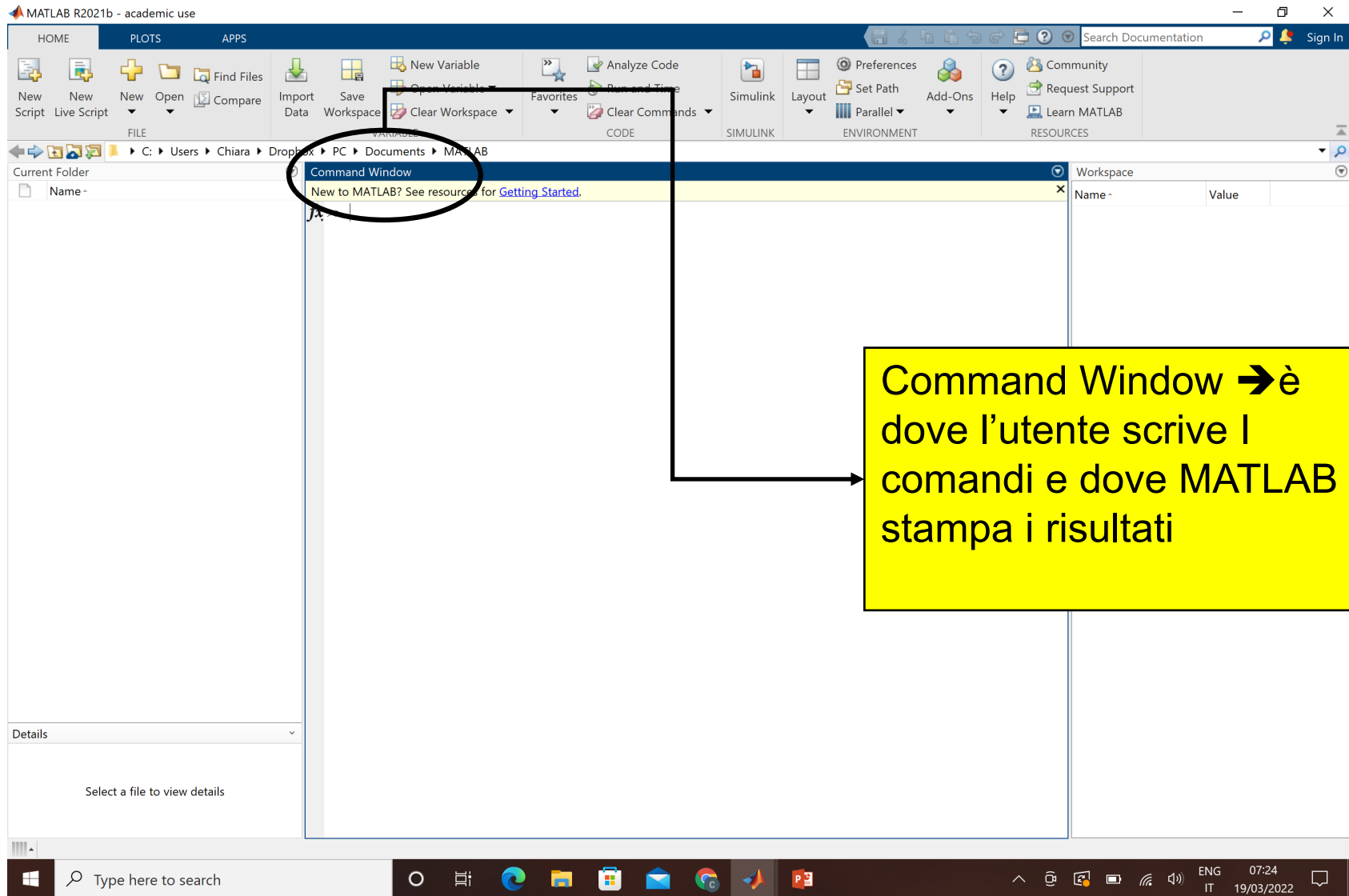
- **Modalità interattiva:** ogni istruzione scritta viene valutata dall'interprete ed immediatamente elaborata
- **Script:** in un file di testo con estensione “.m” si scrive una sequenza di istruzioni. Si chiede che venga eseguita tutta la sequenza

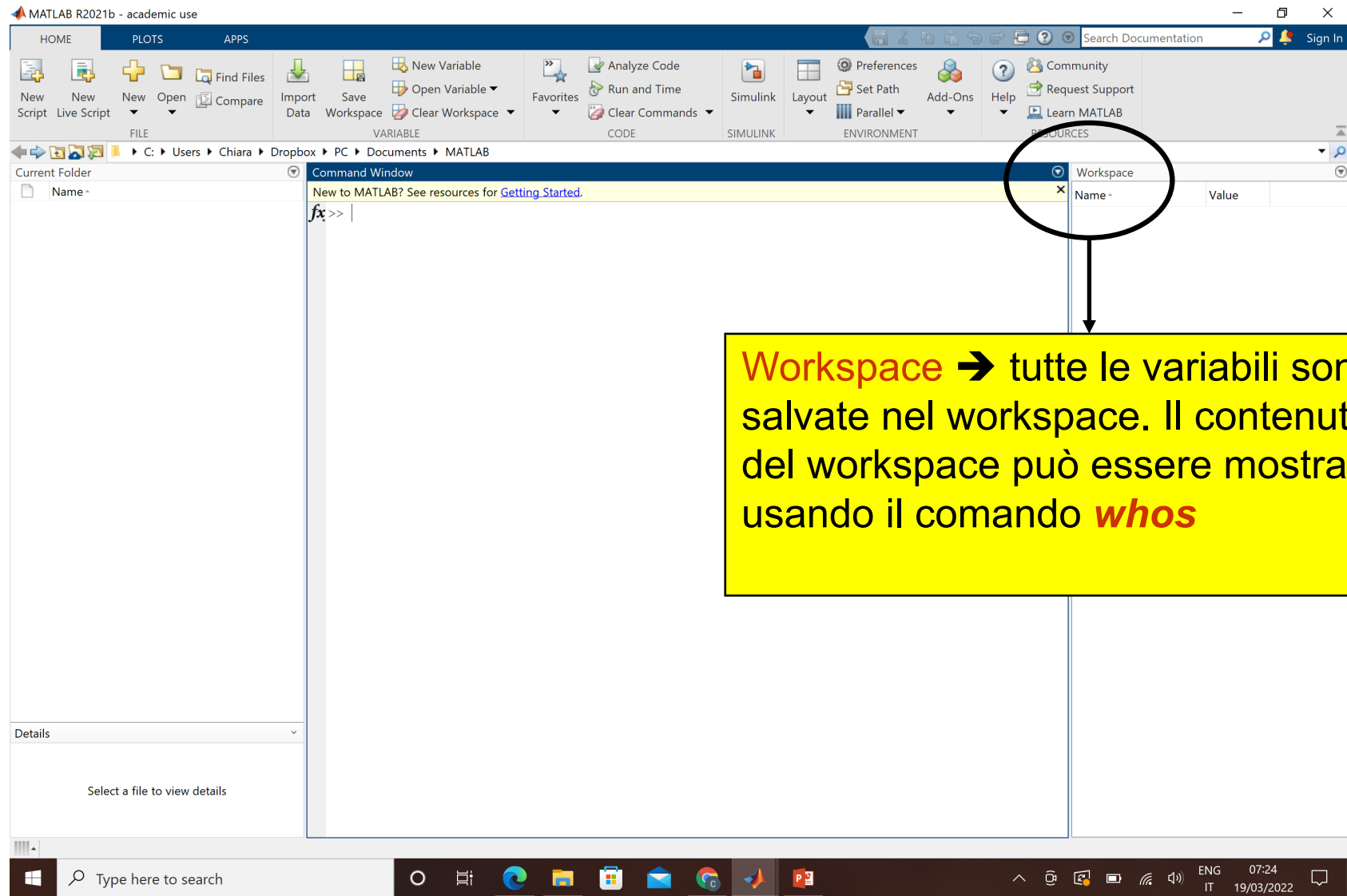
II Desktop

- 4 finestre:
 - » *Command Window*
 - » *Workspace*
 - » *Current directory*
 - » *Command history*

II Desktop

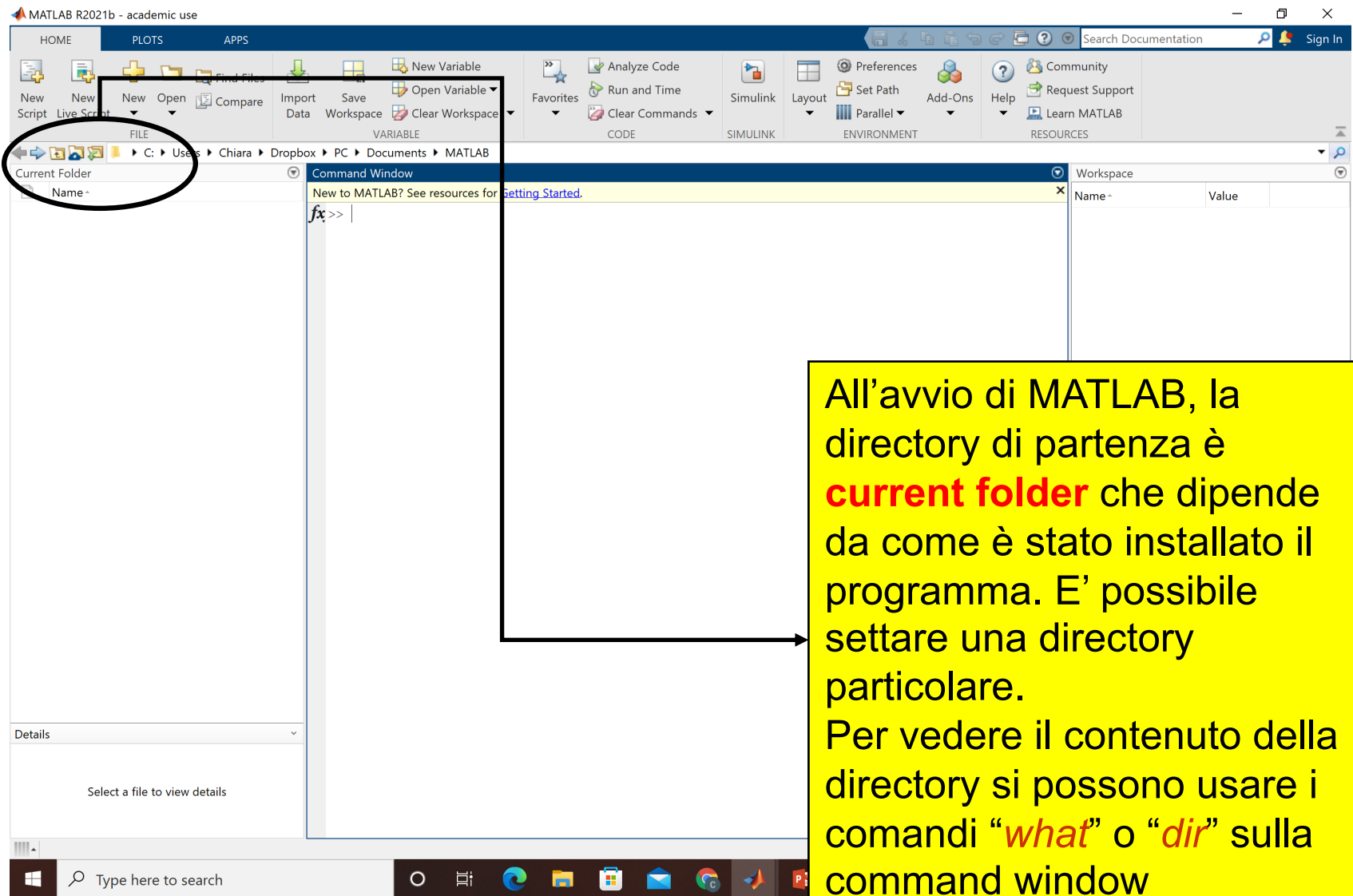


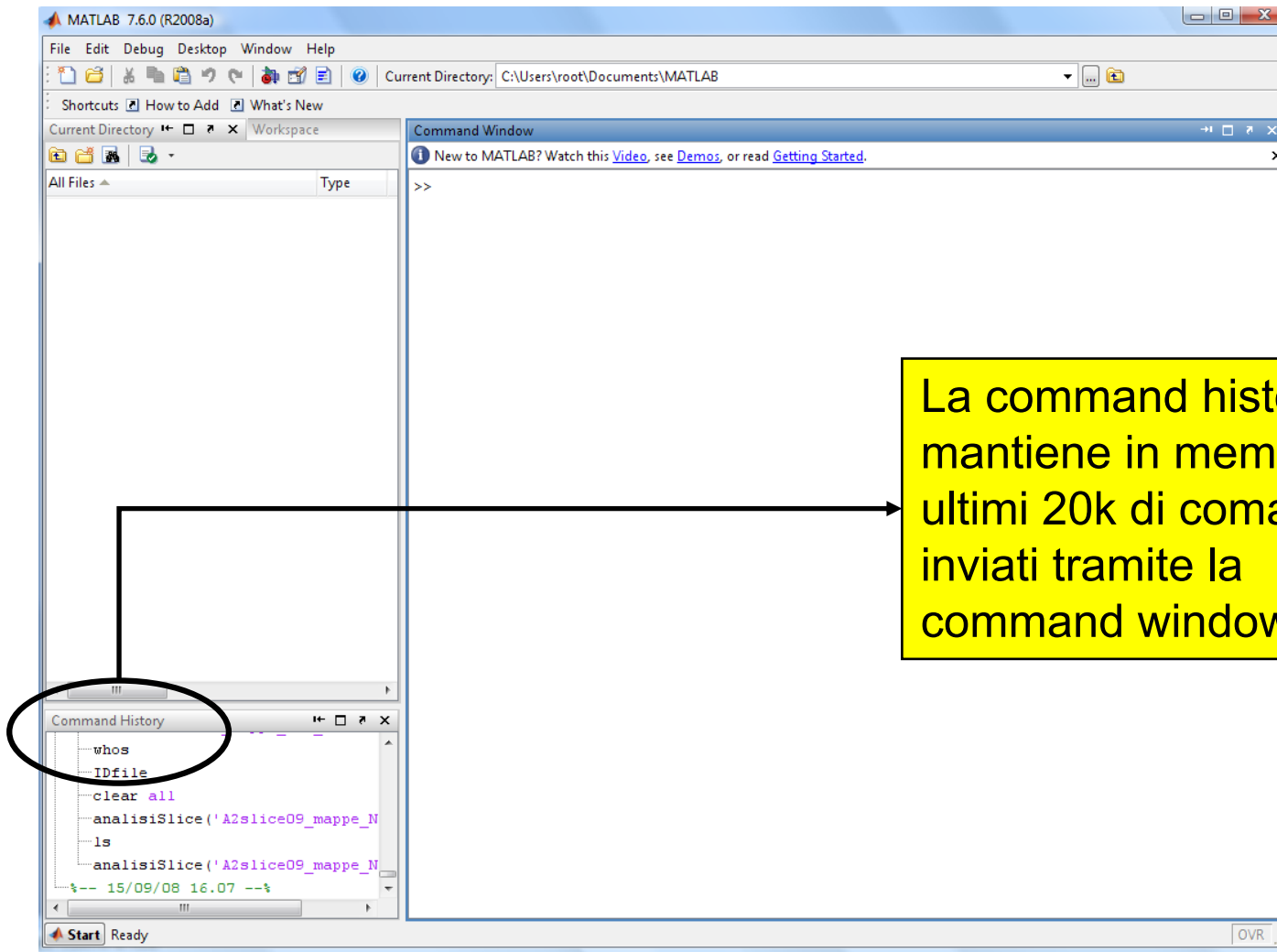




L'uso interattivo della command window è spesso poco conveniente e noioso:

- è difficile trovare gli errori,
- per ripetere le operazioni bisogna riscrivere tutto ogni volta





La command history mantiene in memoria gli ultimi 20k di comandi inviati tramite la command window

Assegnazione di una variabile

>> **a=10** – crea la variabile a e le assegna il valore 10

>> **b=[]** – crea la variabile b e le assegna il vettore vuoto

>> **size(a)** – restituisce le dimensioni della variabile (in questo caso:

```
>> a=10
a =
    10
>> size(a)
ans =
     1     1
>> b=[]
b =
     []
>> size(b)
ans =
     0     0
```

Operazioni numeriche

- + somma
- - sottrazione
- * moltiplicazione
- / divisione
- ^ potenza
- Funzioni: function-name (value)
esempio: $\text{sqrt}(4) \rightarrow 2$

```
Command Window
New to MATLAB? Watch this Video

>> 6*4

ans =

    24

>>
```

```
Command Window
New to MATLAB? Watch this Video

>> 2^4

ans =

    16

>>
```

```
Command Window
New to MATLAB? Watch this Video

>> sqrt(81)

ans =

     9

>> |
```

Formati

- **format short** (default) - 5 digits
Example: 0.0300
- **format short e** - float, 5 digits
Example: 3.0000e-002
- **format short g** - best of fixed or floating point format, 5 digits
Example: 0.03

- **format long** - 15 digits
Example: 0.030000000000000
- **format long e** - float, 15 digits
Example: 3.000000000000000e-002
- **format long g** - best of fixed or floating point format, 15 digits
Example: 0.03

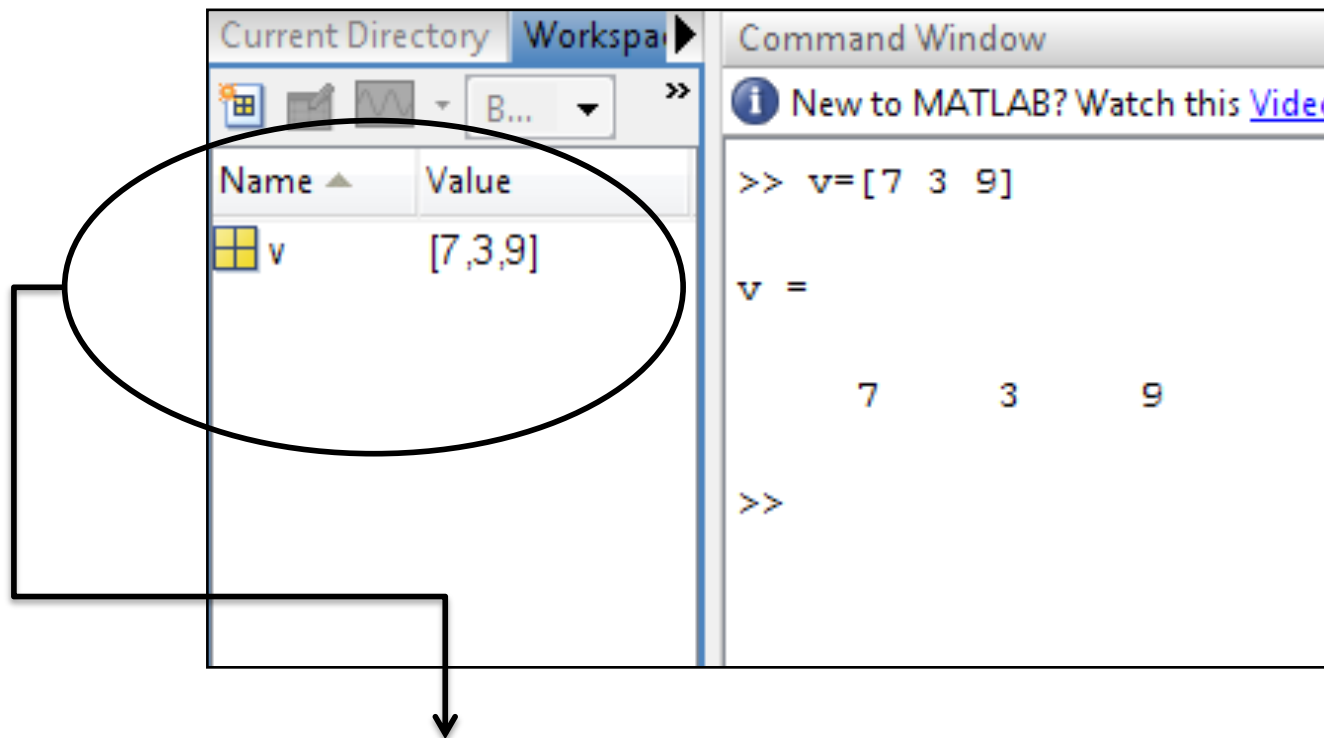
Variabili e valori speciali

- **ans** = ultimo valore calcolato
- **Inf** = infinito
- **NaN** = Not a Number
- **eps** = precisione della macchina = $2.2204e-016$
- **i** e **j** = unità immaginaria = $0 + 1.0000i$
- **pi** = π = 3.1416
- **realmax** = il numero più grande rappresentabile = $1.7977e+308$
- **realmin** = il numero più piccolo rappresentabile = $2.2251e-308$

Vettori e matrici

Creare un vettore riga

Nella Command Window

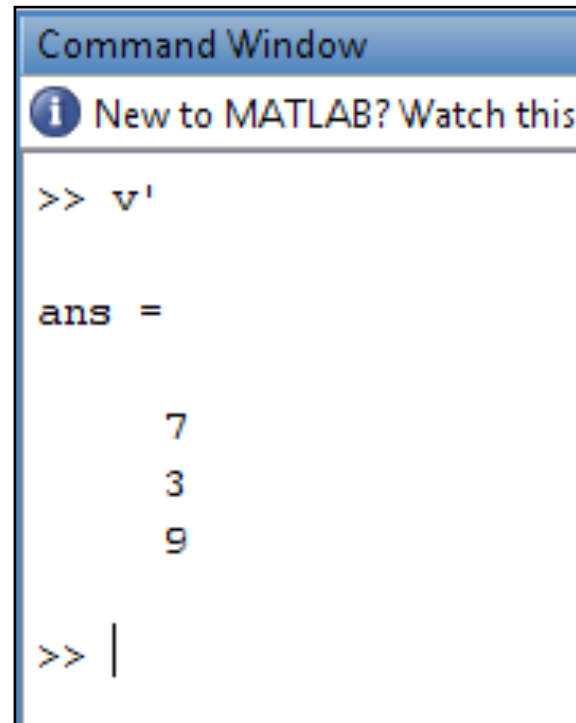


La variabile è salvata provvisoriamente nel workspace

Attenzione: il primo elemento del vettore `v` è `v(1)`

Vettori e matrici

' per trasporre ('= trasposto e coniugato, .' = trasposto)



```
Command Window
i New to MATLAB? Watch this

>> v'

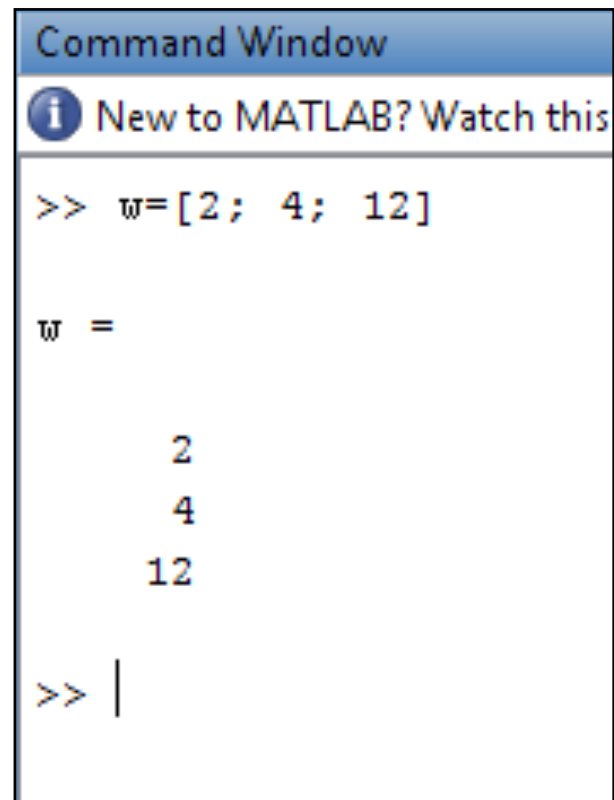
ans =

     7
     3
     9

>> |
```


Vettori e matrici

Creare un vettore colonna



```
Command Window
i New to MATLAB? Watch this
>> w=[2; 4; 12]

w =

     2
     4
    12

>> |
```

Vettori e matrici

Creare una matrice

```
Command Window
i New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> A=[ 1 0 3; 4 0 -4; 2 -8 -7; 1 2 8]

A =

     1     0     3
     4     0    -4
     2    -8    -7
     1     2     8

>> |
```

Vettori e matrici

Creare una matrice di zeri:

zeros(N) matrice di zeri NxN.

zeros(M,N) matrice di zeri MxN

Esempi:

matrix1=zeros(3); → $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

matrix2=zeros(2,3); → $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Vettori e matrici

Operazioni con vettori e matrici

Symbol	Operation
<code>.*</code>	element-by-element multiplication
<code>./</code>	element-by-element “right” division
<code>.\</code>	element-by-element “left” division
<code>.^</code>	element-by-element exponentiation

Vettori e matrici

esempi

$$1) \quad v \times w = [7 \quad 3 \quad 9] \times \begin{bmatrix} 2 \\ 4 \\ 12 \end{bmatrix} = 134 \quad \longrightarrow$$

```
Command Window
i New to MATLAB? Watch t
>> v*w
ans =
    134
>>
```

Vettori e matrici

2) MATLAB usa il semplice asterisco “*” per indicare la moltiplicazione tra matrici

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$



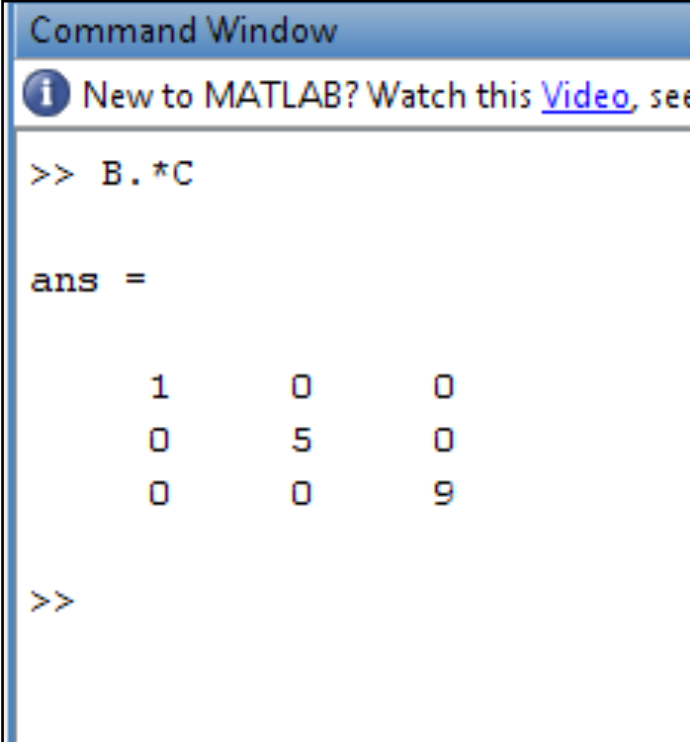
$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
Command Window
i New to MATLAB? Watch this Video, se
>> B*C
ans =
     1     2     3
     4     5     6
     7     8     9
>>
```

Vettori e matrici

3) MATLAB usa “`.*`” per eseguire la moltiplicazione elemento per elemento

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 1 \times 1 & 2 \times 0 & 3 \times 0 \\ 4 \times 0 & 5 \times 1 & 6 \times 0 \\ 7 \times 0 & 8 \times 0 & 9 \times 1 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



```
Command Window
i New to MATLAB? Watch this Video, see
>> B.*C

ans =

     1     0     0
     0     5     0
     0     0     9

>>
```

Vettori e matrici

4)

```
>> A = [1 2 3 4; 5 6 7 8];
>> B = [8 7 6 5; 4 3 2 1];
>> A.*B
ans =
     8    14    18    20
    20    18    14     8

>> A*B
??? Error using ==> *
Inner matrix dimensions must agree.
```

5)

```
>> A*B'
ans =
     60    20
    164    60

>> A.^2
ans =
     1     4     9    16
    25    36    49    64
```


Operazioni sul Workspace

- Nella Command Window
 - » **who** – mostra tutte le variabili
 - » **whos** – mostra il nome delle variabili, il tipo e la taglia
 - » **clear *var1 var2*...** - cancella le variabili riportate
 - » **clear** – cancella tutte le variabili
 - » **close** – chiude le figure
 - » **load** - carica le variabili o i file di dati
 - » **save** – salva le variabili in un file .mat

Load & Save

➤ COMANDO **load**

sintassi

load filename

loads all the variables

load filename X

loads only the variable X

load filename X Y Z

loads the variables X Y & Z

➤ COMANDO **save**

sintassi

save

saves all the variables in a file called matlab.mat

save filename

save all the variables under a file "filename.mat"

save filename X Y Z

save only the variables X Y and Z under a file "filename.mat"

L'Help

Nella Command Window

>> **help *command_name*** – list help in command window

Es: help load

>> **lookfor *keyword*** – search for a word in all commands on disk

Es: lookfor interpolation

Nella Barra degli strumenti

>> **?** – Button on the right = **Help**

Sul web trovate soluzione a tutte le vostre domande



N.B. MATLAB è case sensitive

La funzione Display

>> **disp('text')** – mostra il testo tra apici

Es: *disp('have a nice day')*

disp(['I am ', num2str(18), ' year old'])

N.B. **num2str** – converte il numero in una stringa

Grafici e Figure

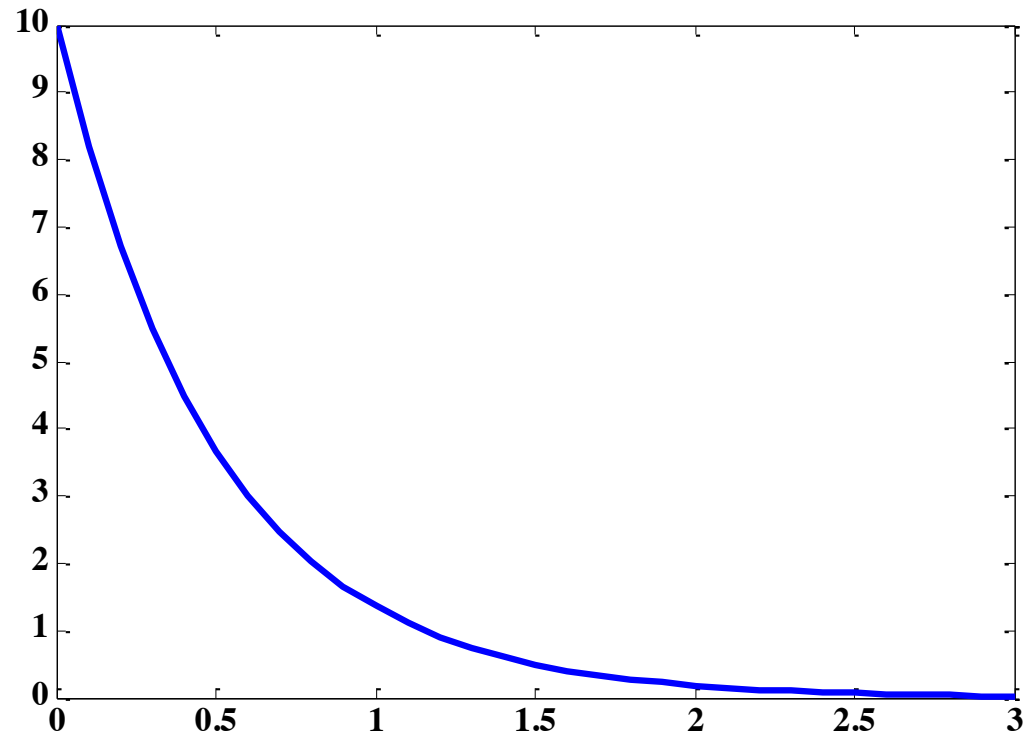
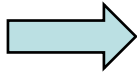
PLOT(X,Y,' ') = plotta il vettore Y versus il vettore X usando il tipo di linea indicata tra apici ' '

Color		Symbol		Line	
b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-. 	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
w	white	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

Grafici e Figure

Esempio

```
>> t=[0:0.1:3];  
>> y=10*exp(-2*t);  
>> plot(t,y)  
>>
```



Grafici e Figure

subplot (m, n, k) divide la figura in $m \times n$ sottofinestre e mette il plot seguente nella finestra numero k

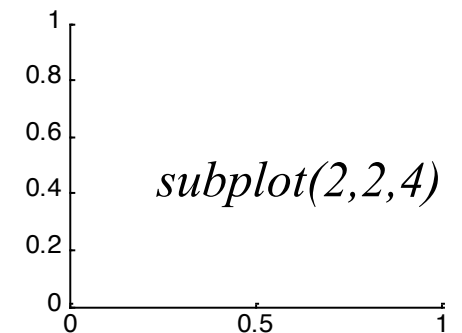
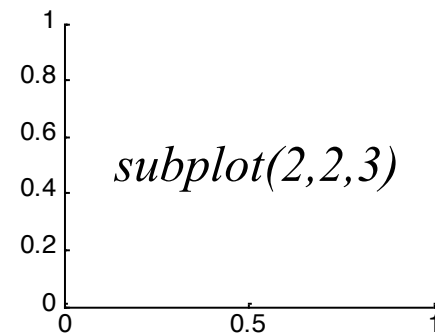
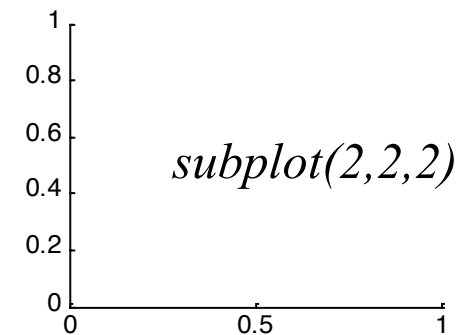
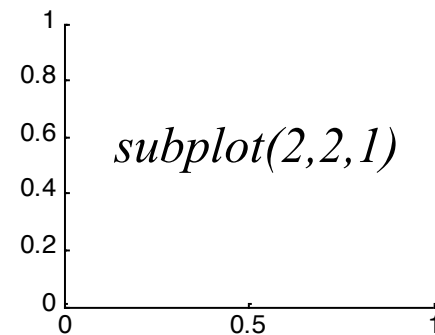
Esempio

subplot(2,2,1)

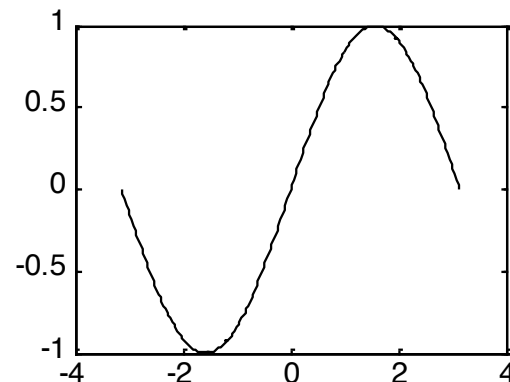
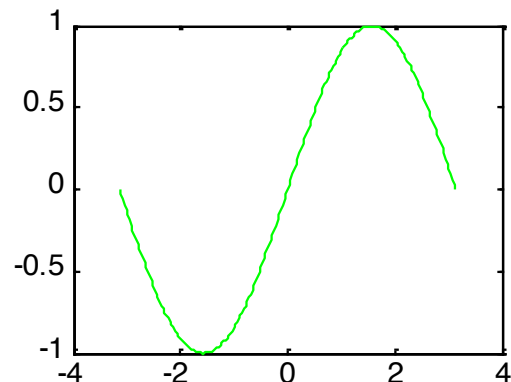
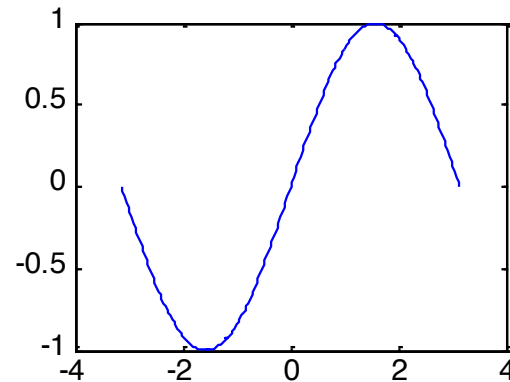
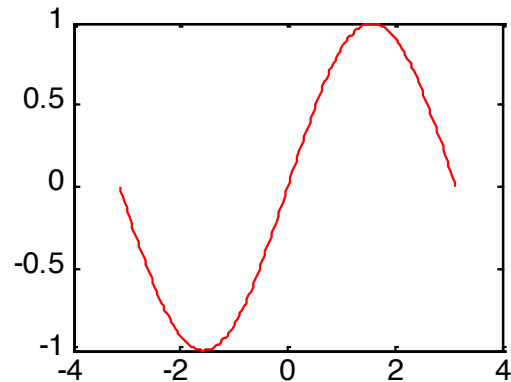
subplot(2,2,2)

subplot(2,2,3)

subplot(2,2,4)



In ogni figura si può mettere un plot diverso



```
x = -pi:pi/100:pi;  
y = sin(x);  
subplot(2, 2, 1);  
plot(x, y, 'r')  
subplot(2, 2, 2);  
plot(x, y, 'b')  
subplot(2, 2, 3);  
plot(x, y, 'g')  
subplot(2, 2, 4);  
plot(x, y, 'k')
```


Grafici e Figure

Altri tipi di grafici

Bar & Area Graphs

bar	Displays columns of m-by-n matrix as m groups of n vertical bars
barh	Displays columns of m-by-n matrix as m groups of n horizontal bars
bar3	Displays columns of m-by-n matrix as m groups of n vertical 3-D bars
bar3h	Displays columns of m-by-n matrix as m groups of n horizontal 3-D bars
area	Displays vector data as stacked area plots

Pie Charts

pie	
pie3	

Histograms

hist	Displays vector data as stacked area plots
rose	Displays data in a polar coordinate system

Costrutti logici

if e switch

```
if I == J
    A(I,J) = 2;
elseif abs(I-J) == 1
    A(I,J) = -1;
else
    A(I,J) = 0;
end
```

```
switch algorithm
case 'ode23'
    str = '2nd/3rd order';
case {'ode15s', 'ode23s'}
    str = 'stiff system';
otherwise
    str = 'other algorithm';
end
```

Costrutti logici

Costrutti iterativi: for e while

```
N = 10;  
for I = 1:N  
    for J = 1:N  
        A(I,J) = 1/(I+J-1);  
    end  
end
```

```
I = 1; N = 10;  
while I <= N  
    J = 1;  
    while J <= N  
        A(I,J) = 1/(I+J-1);  
        J = J+1;  
    end  
    I = I+1;  
end
```

Operatori

- **Operatori relazionali:**

= = (equal to)

< (less than)

> (greater than)

~ = (not equal)

< = (less than or equal to)

> = (greater than or equal to)

- **Operatori logici:**

& (and)

| (or)

~ (not)

Esercizi

Esercizio 1

- caricare il file matlab exercise.mat
- listare tutte le variabili caricate
- cancellare la variabile “anc”
- fare il display della stringa : **“Salverò alcune variabili”**
- salvare la variabile **t** e **mtc** in un file Matlab .mat e nominarlo: **selezione**
- cancellare tutte le variabili

Esercizio 2

- creare la matrice $m_1 = \begin{bmatrix} 6 & 8 & 10 & 3 \\ 28 & 0 & 6 & 1 \\ 1 & 3 & 29 & 31 \end{bmatrix}$ ed il vettore $v_1 = \begin{bmatrix} 3 \\ 18 \\ 1 \end{bmatrix}$
- eseguire la seguente operazione: $m_2 = m_1^T \cdot v_1$ (T = trasposta)
- eseguire la seguente operazione: $m_3 = (m_1)^2$; **(dà errore!!! Leggere il messaggio di errore!!)**
- cancellare tutte le variabili

Tipi di File

- Programmi: .m m-files ascii (text)
- Dati: .mat MATLAB internal format
- Figure: .fig MATLAB internal format
- Altri formati :
 - » Text (e.g. tables)
 - » Pictures (e.g. .jpg)
 - » Worksheets (e.g. Excel)
 - » Sound (e.g. .wav)

File .m

- Gli m-file sono file di testo che contengono una sequenza di comandi MATLAB
- Le linee di commento devono iniziare con il carattere "%"
 % Questa è una linea di commento
- Ci sono due tipi di .m file:

Scripts, che non accettano in ingresso degli argomenti, né restituiscono variabili in uscita. Lavorano sui dati del workspace.

Functions, che possono accettare in ingresso e restituire in uscita delle variabili. Le variabili interne sono locali (non sono visibili dal workspace).

Esempio di Function

Keyword: function

Function Name (same as file name .m)

Output Argument(s)

Input Argument(s)

Online Help

```
function y = mean(x)
% MEAN Average or mean value.
% For vectors, MEAN(x) returns the mean value.
% For matrices, MEAN(x) is a row vector
% containing the mean value of each column.
```

MATLAB
Code

```
[m,n] = size(x);
if m == 1
    m = n;
end
y = sum(x)/m;
```

»output_value = mean(input_value) ← Command Line Syntax

Esempio di Script

Salvato in un
.m file
chiamato
try_plots.m

```
%use plotyy and hist

t = 0:900; A = 1000; a = 0.005;
b = 0.005;
z1 = A*exp(-a*t);
z2 = sin(b*t);
figure(1)
[haxes,hline1,hline2] =
plotyy(t,z1,t,z2,'semilogy','plo
t');
figure(2)
y(:,1)=randn(1000,1);
y(:,2)=0.5*randn(1000,1);
y(:,3)=0.02+0.5*randn(1000,1);
hist(y)
```

Quando si invoca lo script (cioè digitando `try_plots` nella Command Window), MATLAB semplicemente esegue le istruzioni riportate nel file. Ogni variabile creata dalle istruzioni rimane nel workspace

Esercizi

Esercizio 3

Creare lo script es3.m e al suo interno:

- Creare il vettore riga

```
t=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15];
```

(oppure `t=[1:1:15]`, più compatto)

- Creare i vettori:

```
y1=0.05*t;
```

```
y2=exp(-0.5*t);
```

come vettori colonna.

- Plottare **y1 vs t** con una linea continua rossa
- creare un subplot 2x1 che mostra il plot di **y1** (con linea continua verde) nella finestra superiore ed il plot di **y2** (in magenta - circle (symbol) – linea punteggiata) nella finestra inferiore.

Esercizi

Esercizio 4

- Creare la function **mediapesata.m** che dati in ingresso il vettore riga **x** ed il vettore riga dei pesi **w** (della stessa lunghezza, n) restituisca la media pesata di **x** secondo **w**:

$$y = \frac{\sum_{i=1}^n \mathbf{x}(i) \mathbf{w}(i)}{\sum_{i=1}^n \mathbf{w}(i)}$$

- Creare uno script e nominarlo es4.m
- Cancellare tutte le variabili (clear all)
- Chiudere tutte le finestre aperte (close all)
- Cancella la command window (clc)
- Caricare i dati in dati_es4.mat

Il file contiene la matrice X (5x50) ed il vettore w (1x50)

- Per ogni riga i della matrice X, trovare la media pesata (secondo w) degli elementi della riga e salvarla in posizione i-esima di un vettore y (5x1)
- Mostrare il risultato (disp)

Strutture

VETTORI DI STRUTTURE

Le strutture sono vettori di MATLAB i cui campi possono contenere tipi diversi di dati.

Per esempio, un campo può contenere del testo, un altro uno scalare, un terzo una matrice etc.

Esempio:

```
patient.name = 'Ann Lane';  
patient.billing = 28.50;  
patient.test = [68 70 68; 118 118 119; 172 170 169];
```

“patient” è il nome della struttura. Per accedere, per esempio al campo “name”, si deve scrivere **patient.name**