

Corso di Segnali e Sistemi
Ingegneria Elettronica
Università degli Studi di Padova
(Materiale dei Proff. N. Benvenuto e C. Dalla Man,
aggiornato da M. Cagnazzo)

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
 - Niente dichiarazioni di tipi, allocazione di memoria, funzioni virtuali, polimorfismo etc...
 - Tuttavia, se si vuole, si possono usare classi e compilatori (ma noi non lo faremo in questi lab)
- Esportabilità
 - Le funzioni MatLab sono file di testo

Programmare in MatLab

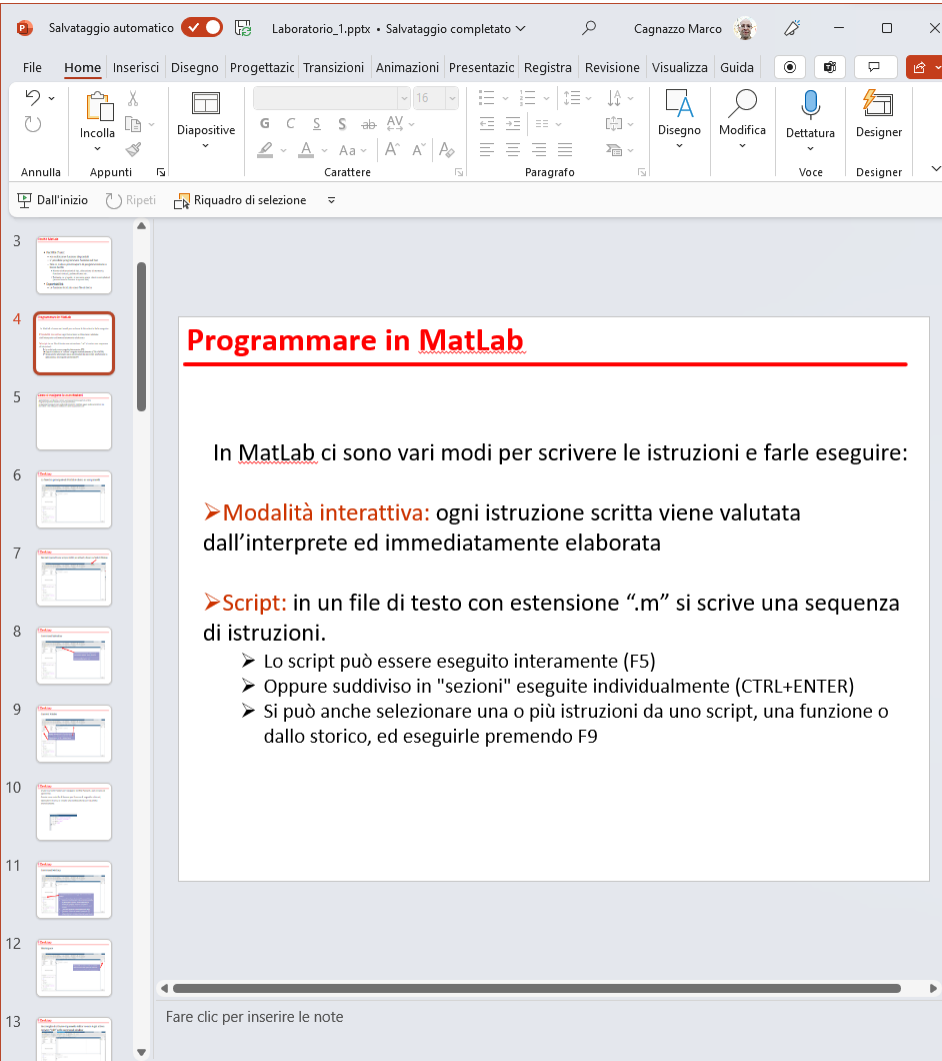
In MatLab ci sono vari modi per scrivere le istruzioni e farle eseguire:

- **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.
 - Lo script può essere eseguito interamente (F5)
 - Oppure suddiviso in "sezioni" eseguite individualmente (CTRL+ENTER)
 - Si può anche selezionare una o più istruzioni da uno script, una funzione o dallo storico, ed eseguirle premendo F9

Come si svolgono le esercitazioni

Aprirete Matlab in una finestra, e tenete questa presentazione aperta in un'altra. Eseguite le operazioni indicate in questa presentazione.

Le operazioni da eseguire sono quelle negli screenshot di Matlab, oppure quelle evidenziate in **blu**. Non esitate a fare delle prove cambiando i valori dei parametri usati.

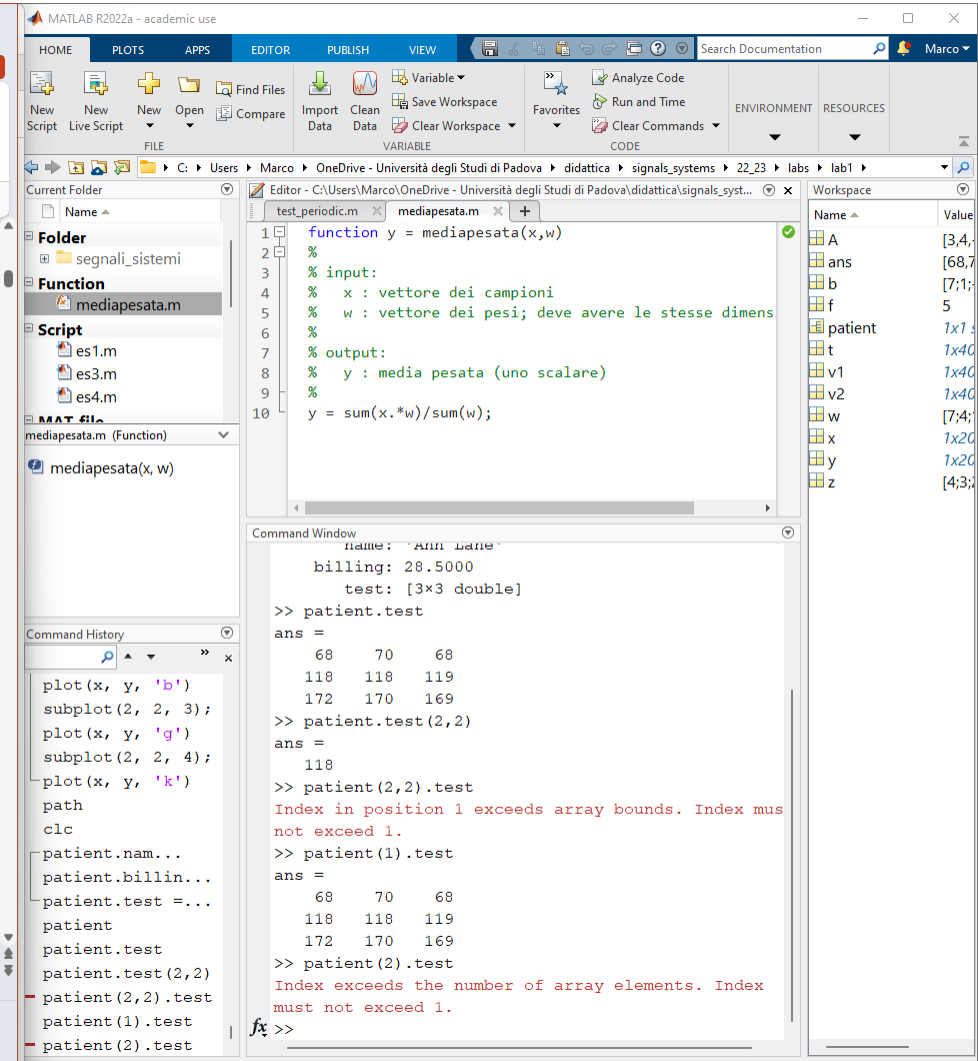


Programmare in MatLab

In MatLab ci sono vari 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.
 - Lo script può essere eseguito interamente (F5)
 - Oppure suddiviso in "sezioni" eseguite individualmente (CTRL+ENTER)
 - Si può anche selezionare una o più istruzioni da uno script, una funzione o dallo storico, ed eseguirle premendo F9

Fare clic per inserire le note



MATLAB R2022a - academic use

Editor - C:\Users\Marco\OneDrive - Università degli Studi di Padova\didattica\signals_syst... \ labs \ lab1

```
function y = mediapesata(x,w)
%
% input:
% x : vettore dei campioni
% w : vettore dei pesi; deve avere le stesse dimens
%
% output:
% y : media pesata (uno scalare)
%
y = sum(x.*w)/sum(w);
```

Command Window

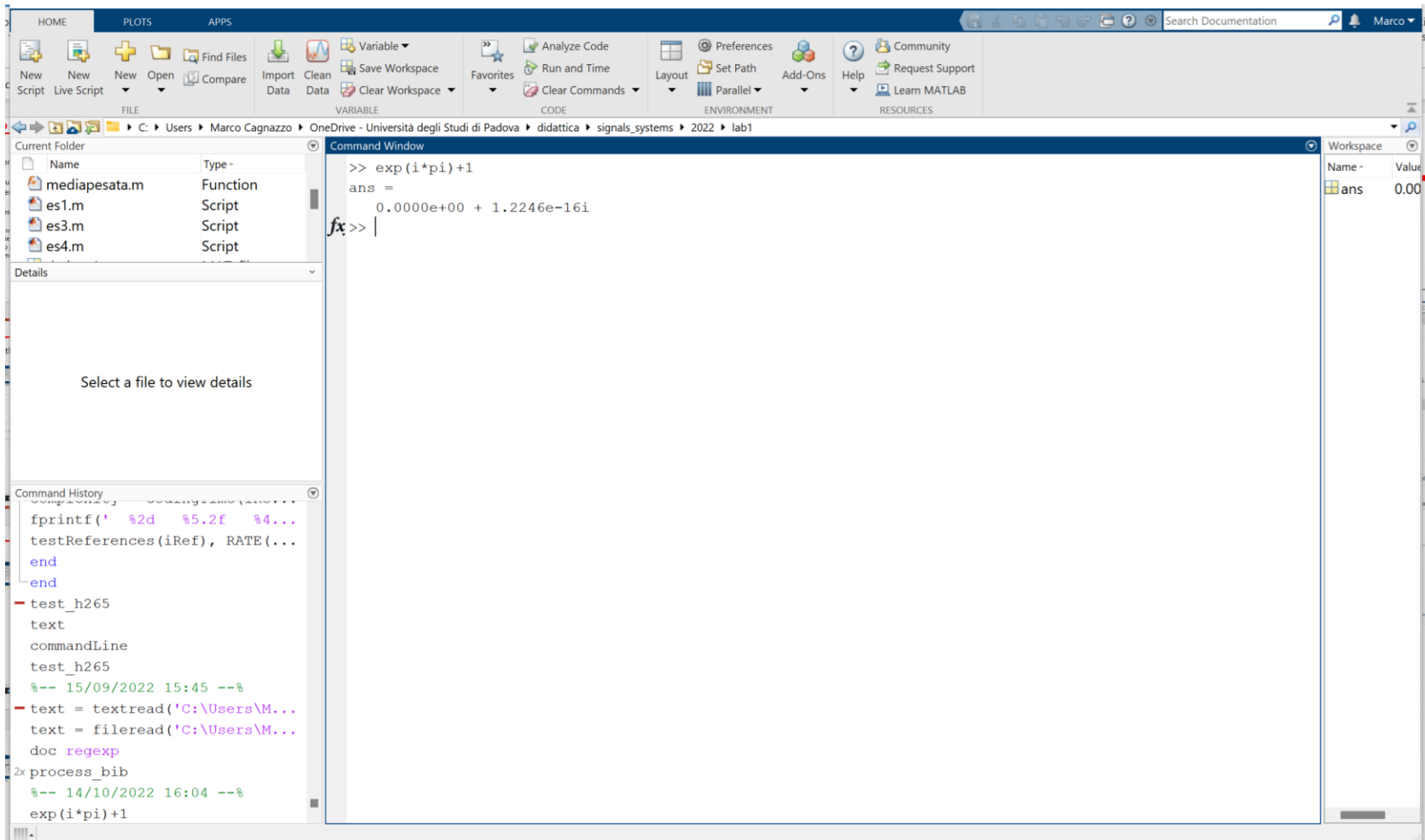
```
name: 'Ann Lane'
billing: 28.5000
test: [3x3 double]
>> patient.test
ans =
    68    70    68
   118   118   119
   172   170   169
>> patient.test(2,2)
ans =
    118
>> patient(2,2).test
Index in position 1 exceeds array bounds. Index must not exceed 1.
>> patient(1).test
ans =
    68    70    68
   118   118   119
   172   170   169
>> patient(2).test
Index exceeds the number of array elements. Index must not exceed 1.
fx >>
```

Command History

```
plot(x, y, 'b')
subplot(2, 2, 3);
plot(x, y, 'g')
subplot(2, 2, 4);
plot(x, y, 'k')
path
clc
patient.nam...
patient.billin...
patient.test =...
patient
patient.test
patient.test(2,2)
patient(2,2).test
patient(1).test
patient(2).test
```

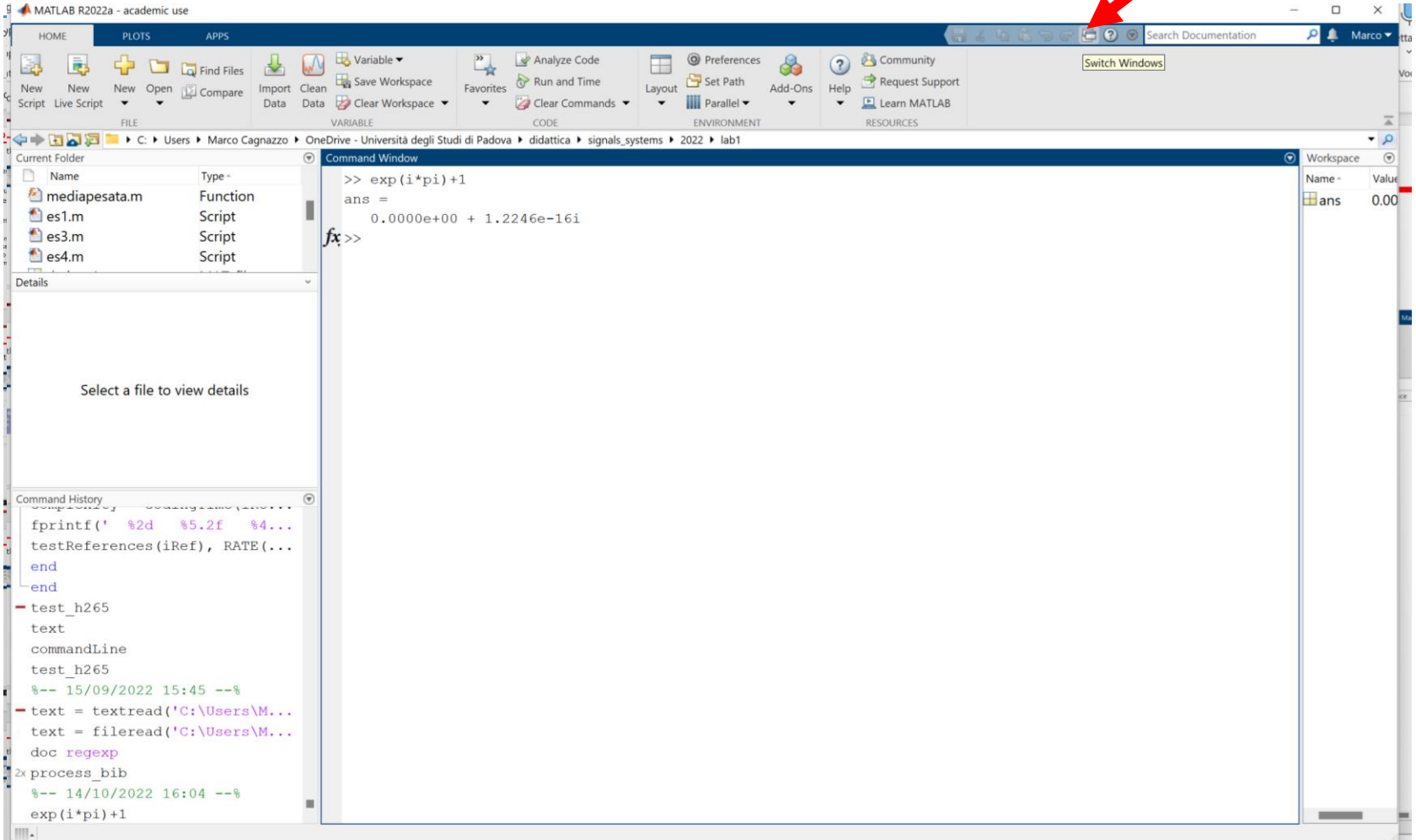
II Desktop

La finestra principale di Matlab è divisa in vari pannelli



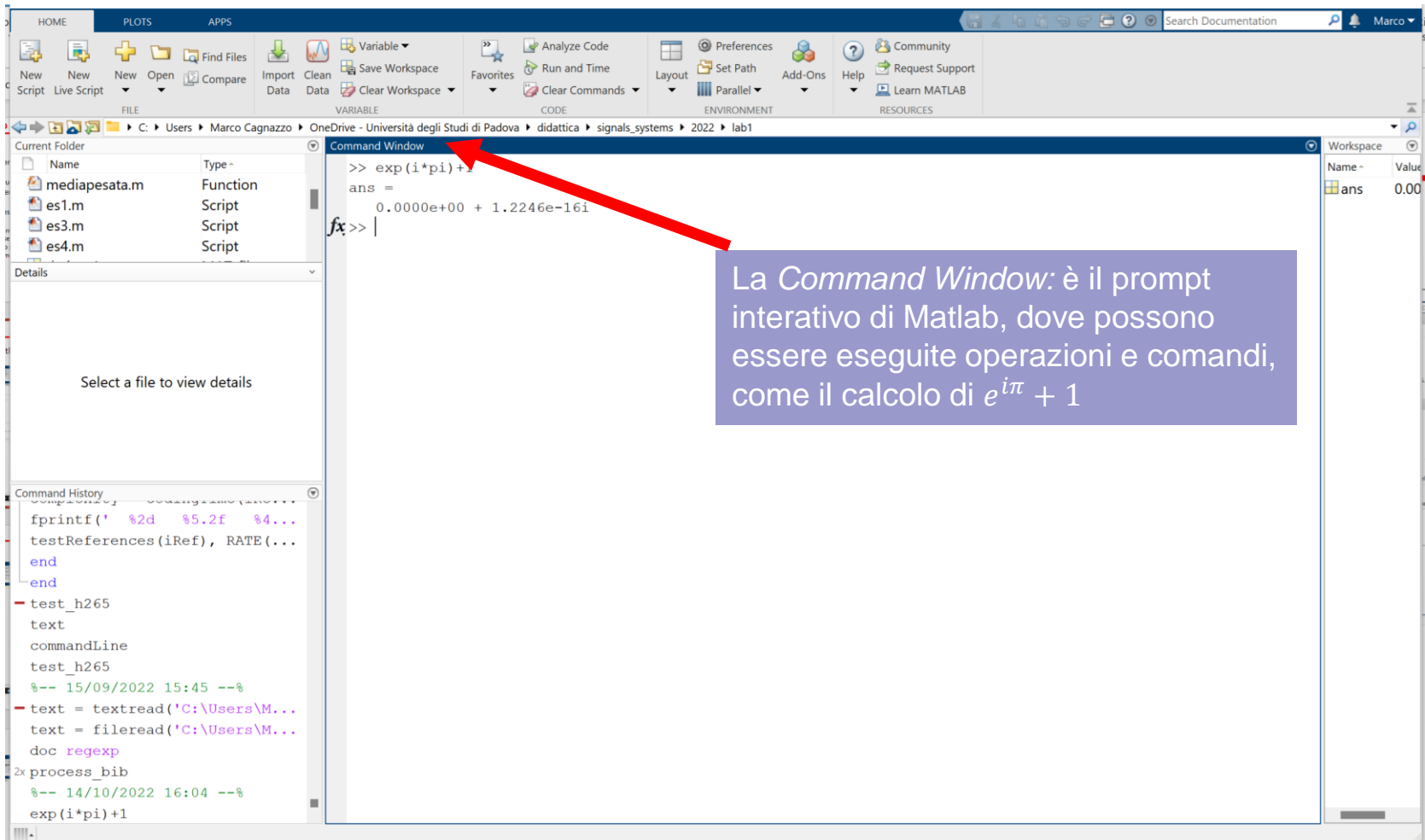
II Desktop

Non tutti i pannelli sono sempre visibili: per attivarli, cliccare su Switch Window



II Desktop

Command Window



The screenshot displays the MATLAB software interface. The Command Window is the central focus, showing the following text:

```
>> exp(i*pi)+1  
ans =  
0.0000e+00 + 1.2246e-16i  
fx>> |
```

A red arrow points from the text box to the Command Window. The Command History window at the bottom left shows the following commands:

```
fprintf(' %2d %5.2f %4...  
testReferences(iRef), RATE(...  
end  
end  
- test_h265  
text  
commandLine  
test_h265  
%-- 15/09/2022 15:45 --%  
- text = textread('C:\Users\M...  
text = fileread('C:\Users\M...  
doc regexp  
2x process_bib  
%-- 14/10/2022 16:04 --%  
exp(i*pi)+1
```

The Workspace window on the right shows the variable 'ans' with a value of 0.00.

La *Command Window*: è il prompt interattivo di Matlab, dove possono essere eseguite operazioni e comandi, come il calcolo di $e^{i\pi} + 1$

Il Desktop

Current Folder

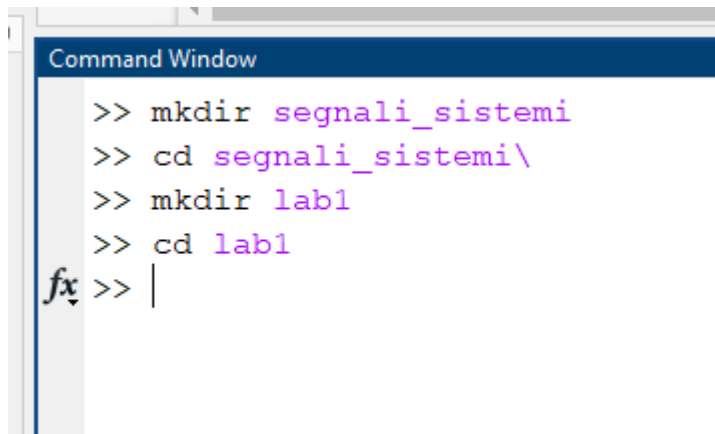
The screenshot displays the MATLAB software interface. At the top, the 'Current Folder' browser shows the current directory path: `C:\Users\Marco Cagnazzo\OneDrive - Università degli Studi di Padova\didattica\signals_systems\2022\lab1`. Below this, a list of files is shown, including `mediaplata.m`, `es1.m`, `es3.m`, and `es4.m`. The 'Details' section below the list is currently empty, with the text 'Select a file to view details' visible. To the right, the 'Command Window' shows the execution of the command `>> exp(i*pi)+1`, resulting in `ans = 0.0000e+00 + 1.2246e-16i`. The 'Workspace' window on the far right shows the variable `ans` with a value of `0.00`. At the bottom, the 'Command History' window shows a list of previously executed commands, including `fprintf`, `testReferences`, `test_h265`, `textread`, `fileread`, `doc regex`, `process_bib`, and `exp(i*pi)+1`.

Current Folder mostra il contenuto della cartella corrente, il cui path è in alto. Selezionando un file, i dettagli sono visibili qui

Il Desktop

Usate Current Folder per navigare nel File System, per creare ed aprire file.

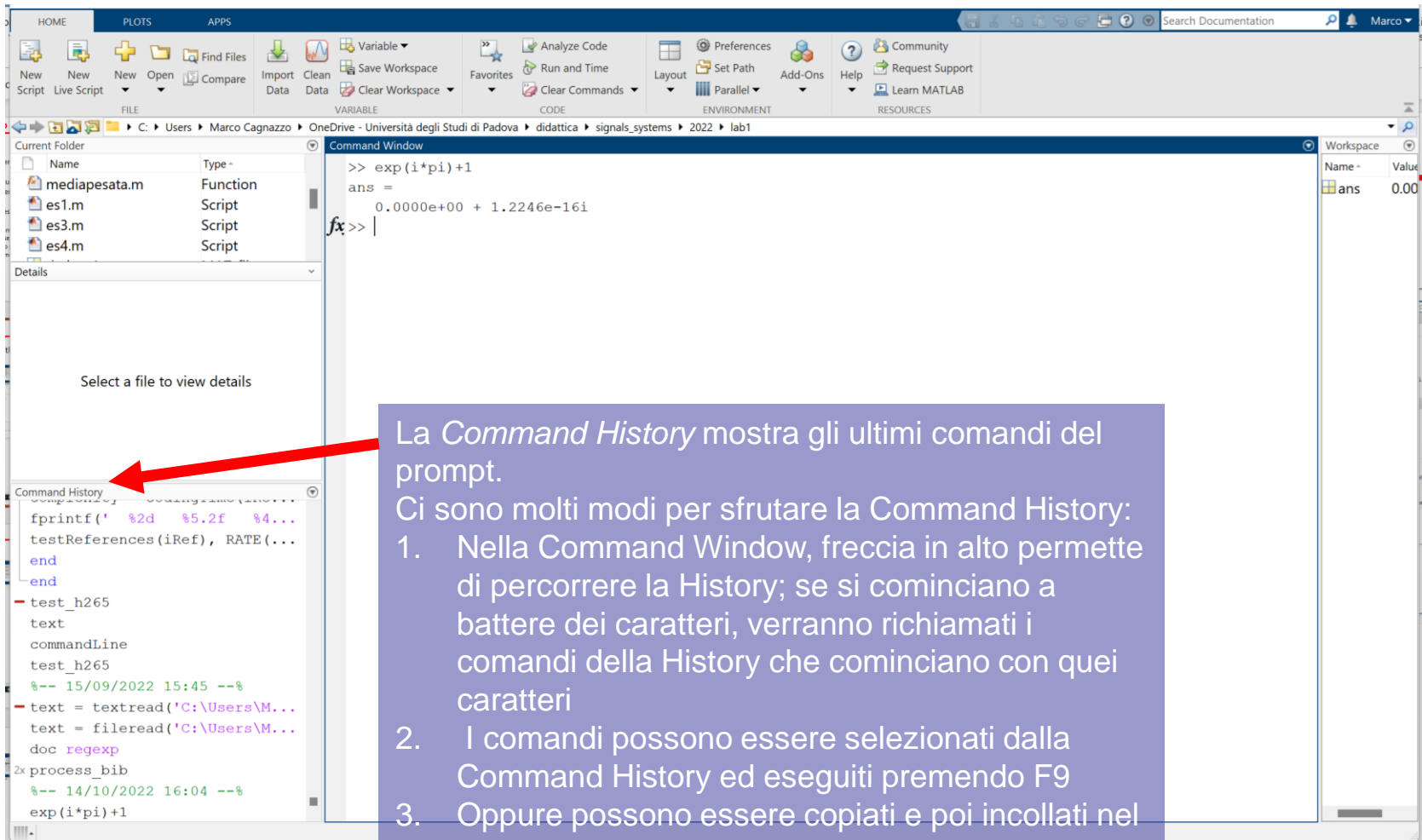
Create una cartella di lavoro per il corso di segnali e sistemi, spostatevi in essa e create una sottocartella per la prima esercitazione.



```
Command Window
>> mkdir segnali_sistemi
>> cd segnali_sistemi\
>> mkdir lab1
>> cd lab1
fx >> |
```

II Desktop

Command History



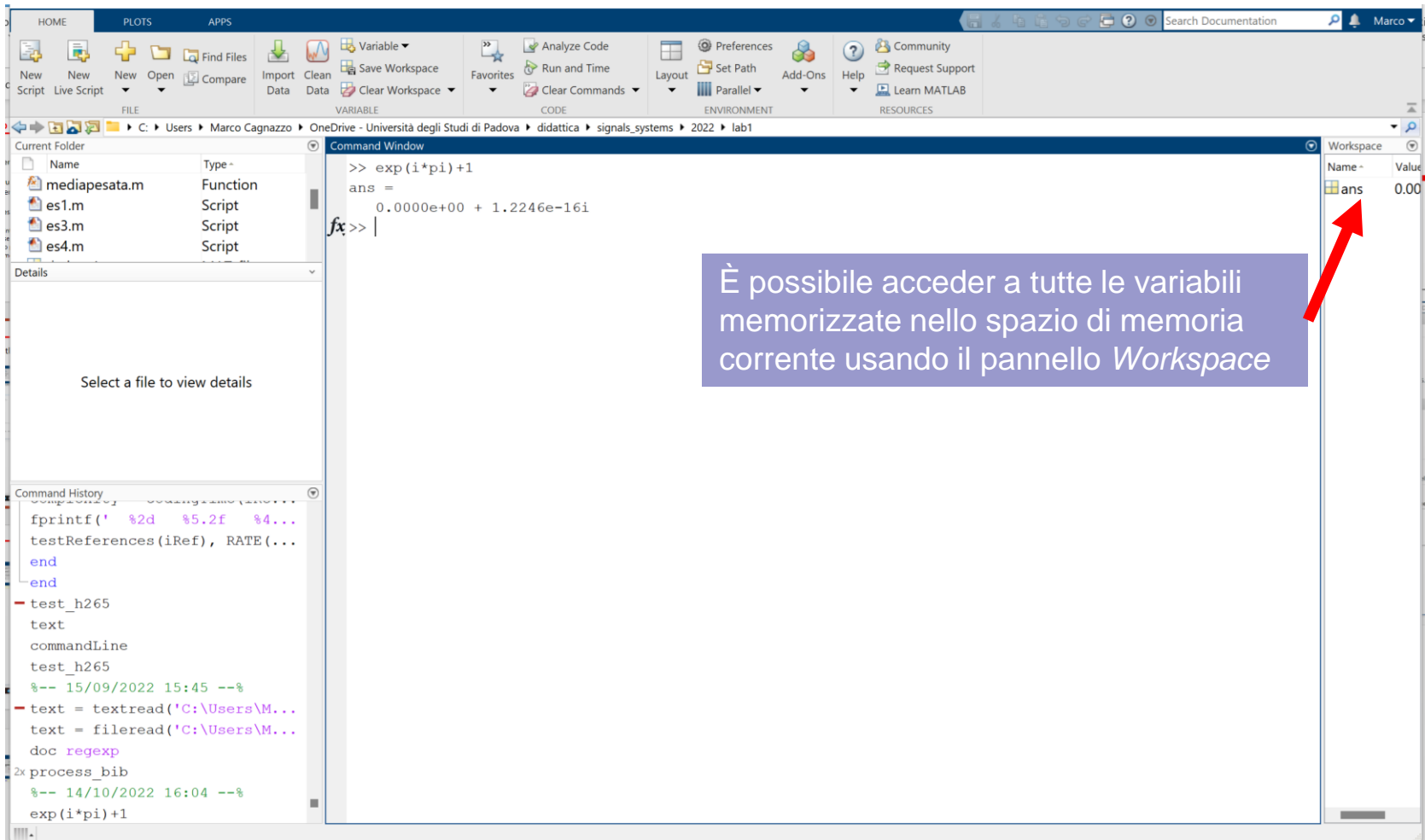
La *Command History* mostra gli ultimi comandi del prompt.

Ci sono molti modi per sfruttare la *Command History*:

1. Nella *Command Window*, freccia in alto permette di percorrere la *History*; se si cominciano a battere dei caratteri, verranno richiamati i comandi della *History* che cominciano con quei caratteri
2. I comandi possono essere selezionati dalla *Command History* ed eseguiti premendo F9
3. Oppure possono essere copiati e poi incollati nel prompt oppure in un file di comandi (detto script)

II Desktop

Workspace



The screenshot displays the MATLAB R2022a desktop environment. The Command Window shows the execution of the command `exp(i*pi)+1`, resulting in the value `ans = 0.0000e+00 + 1.2246e-16i`. The Workspace panel on the right shows the variable `ans` with its value `0.0000e+00 + 1.2246e-16i`. A red arrow points from the text box to the `ans` variable in the Workspace panel.

È possibile accedere a tutte le variabili memorizzate nello spazio di memoria corrente usando il pannello *Workspace*

II Desktop

Si consiglia di attivare il pannello Editor se non è già attivo:
scrivere “edit” nella command window

The screenshot displays the MATLAB desktop interface. The top ribbon includes tabs for HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. The EDITOR tab is active, showing a toolbar with icons for file operations (New, Open, Save, Compare, Print), navigation (Go To, Find, Bookmark), code editing (Refactor, Analyze), and execution (Run, Step, Stop). The main workspace is divided into several panes:

- Current Folder:** Shows a list of files in the current directory: mediapesata.m (Function), es1.m (Script), es3.m (Script), and es4.m (Script).
- Editor - untitled:** A text editor window with a single line of code: `1 |`.
- Workspace:** A table showing the current workspace variables:

Name	Value
ans	0.00
- Command History:** A list of previously executed commands, including `testReferences(iRef), RATE(...)`, `test_h265`, and `exp(i*pi)+1`.
- Command Window:** The active window showing the current command session:

```
>> exp(i*pi)+1
ans =
    0.0000e+00 + 1.2246e-16i
>> edit
fx>>
```

The status bar at the bottom indicates the zoom level is 100%, the encoding is UTF-8, the line ending is CRLF, the current file is script, and the cursor is at line 1, column 1.

II Desktop

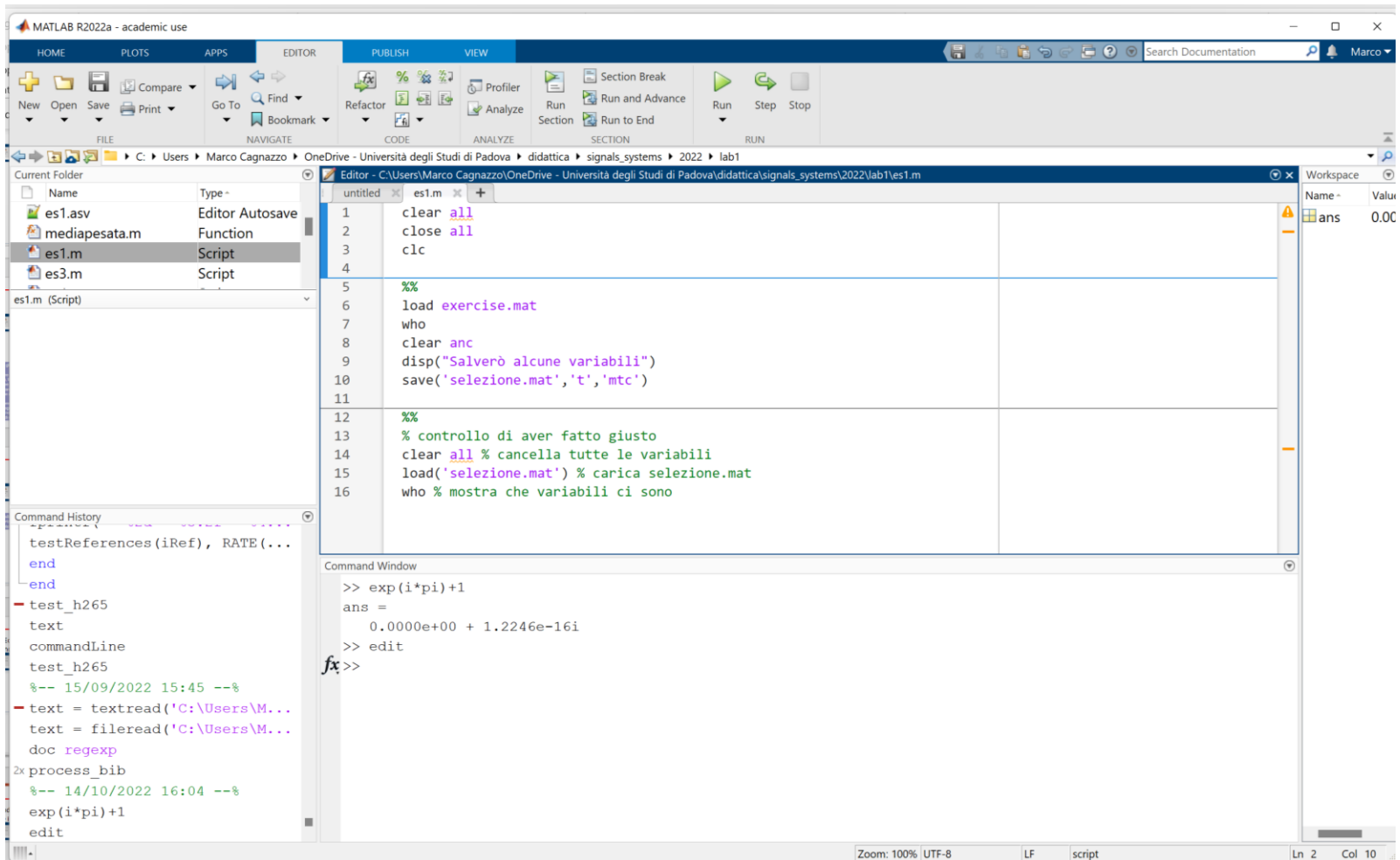
Dal pannello editor si possono modificare e lanciare file di comandi (script) e funzioni, anche in modalità “section”

The screenshot displays the MATLAB desktop environment. The top ribbon includes tabs for HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. The EDITOR tab is active, showing a toolbar with icons for New, Open, Save, Compare, Go To, Find, and Bookmark. The main editor window is titled "Editor - untitled" and contains a single line of code: `1 |`. To the left of the editor is a file browser showing the current folder `C:\Users\Marco Cagnazzo\OneDrive - Università degli Studi di Padova\didattica\signals_systems\2022\lab1`. The file list includes `mediapesata.m` (Function), `es1.m` (Script), `es3.m` (Script), and `es4.m` (Script). Below the file browser is a "Details" section with the text "Select a file to view details". At the bottom left is the Command History window, showing a list of executed commands and their timestamps. At the bottom right is the Command Window, which displays the output of the `edit` command: `>> exp(i*pi)+1`, `ans =`, `0.0000e+00 + 1.2246e-16i`, `>> edit`, and `fx>>`. The status bar at the bottom indicates "Zoom: 100% UTF-8 CRLF script Ln 1 Col 1".

II Desktop

Doppio clic su un file per aprirlo

Per eseguirlo, clic su Run, oppure F5, oppure scrivere il nome nella CW



The screenshot displays the MATLAB R2022a desktop environment. The main window is the Editor, showing a script named 'es1.m' with the following code:

```
1 clear all
2 close all
3 clc
4
5 %%
6 load exercise.mat
7 who
8 clear anc
9 disp("Salverò alcune variabili")
10 save('selezione.mat','t','mtc')
11
12 %%
13 % controllo di aver fatto giusto
14 clear all % cancella tutte le variabili
15 load('selezione.mat') % carica selezione.mat
16 who % mostra che variabili ci sono
```

The Command Window (CW) at the bottom shows the execution of the script, resulting in the following output:

```
>> exp(i*pi)+1
ans =
    0.0000e+00 + 1.2246e-16i
>> edit
fx>>
```

The Command History window on the left shows the sequence of commands entered in the Command Window, including the execution of the script and the subsequent 'edit' command.

Assegnazione di una variabile

PROVATE ANCHE VOI I COMANDI DEGLI ESEMPI NELLE SLIDE SEGUENTI!!!!

>> **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: 0x0)

Notare che il tipo viene assegnato automaticamente al momento della creazione della variabile

```
>> 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
i New to MATLAB? Watch this Video
>> 6*4
ans =
    24
>>
```

```
Command Window
i New to MATLAB? Watch this Video
>> 2^4
ans =
    16
>>
```

```
Command Window
i New to MATLAB? Watch this Video
>> sqrt(81)
ans =
     9
>> |
```

Caratteri speciali

Il simbolo % introduce un commento: tutto quello che segue sulla stessa linea è ignorato

Il simbolo ; sopprime l'output e separa le righe delle matrici

Il simbolo . è usato per distinguere le operazioni elemento per elemento da quelle su matrici

Il simbolo , è usato per separare le colonne di vettori e matrici

Il simbolo ' è usato per il trasposto coniugato

Le parentesi quadre sono usate per creare vettori e matrici, le parentesi tonde per accedere agli elementi delle matrici e per chiamare le funzioni

Il primo elemento di un vettore x è accessibile con x(1)

```
>> x = [1, 2, 3]
x =
     1     2     3
>> y = [3, 2, 1]'
y =
     3
     2
     1
>> z=y+[1; 1; 1]
z =
     4
     3
     2
>> w=2*z-x';
>> w(1)
ans =
     7
```

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

The screenshot shows the MATLAB interface. On the left, the 'Workspace' window displays a table with two columns: 'Name' and 'Value'. A row is highlighted with a yellow background, showing the variable 'v' with the value '[7,3,9]'. A black oval encircles this row. On the right, the 'Command Window' shows the command `>> v=[7 3 9]` and the resulting output: `v =` followed by the numbers 7, 3, and 9 on separate lines. A black arrow originates from the oval in the workspace window and points downwards towards the text below.

Name	Value
v	[7,3,9]

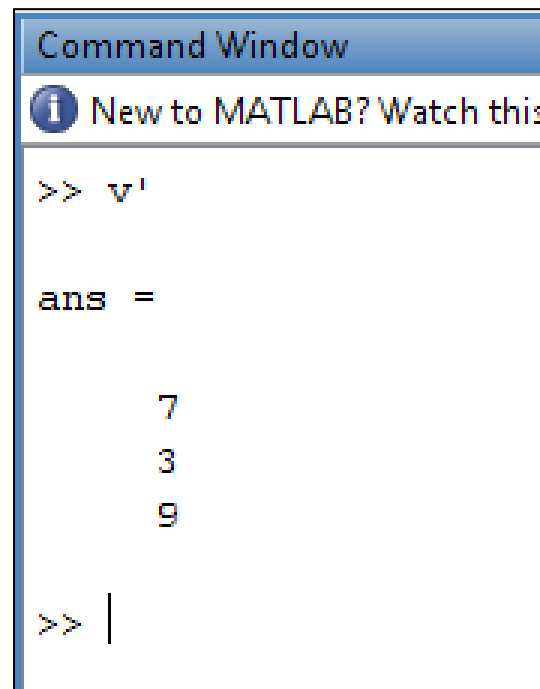
```
>> v=[7 3 9]
v =
    7     3     9
>>
```

La variabile è salvata nel workspace

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

Vettori e matrici

' per il trasposto e coniugato, .' per il = trasposto)

A screenshot of the MATLAB Command Window. The title bar reads "Command Window". Below the title bar is an information icon and the text "New to MATLAB? Watch this". The main area shows the command prompt ">> v'" followed by the output "ans = 7 3 9". At the bottom, the prompt ">> |" is visible.

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

>> v'

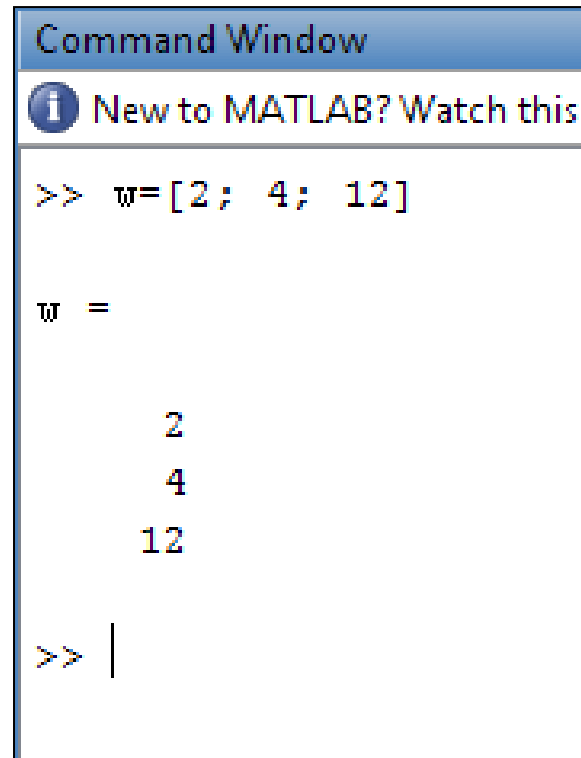
ans =

     7
     3
     9

>> |
```

Vettori e matrici

Creare un vettore colonna



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

w =

     2
     4
    12

>> |
```

Vettori e matrici

Creare matrici: virgola o spazio per separare le colonne,
punto e virgola o Enter per separare le righe
`whos` permette di visualizzare le variabili presenti in memoria

```
Command Window
>> a = [2,3 ; 4, 5];
>> b = [ 2, 5, 1; 4, 6, 7];
>> whos a b
    Name      Size      Bytes  Class  Attributes
    a         2x2         32  double
    b         2x3         48  double

>> a*b
ans =
    16    28    23
    28    50    39
fx >>
```

Vettori e matrici

Creare una matrice di zeri o di uno:

zeros(N) matrice di zeri NxN.

zeros(M,N) matrice di zeri MxN

ones(M,N) matrice di tutti uno 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

L'operatore `*` calcola il prodotto matriciale e dà errore se il numero di colonne della prima variabile è diverso dal numero di righe della seconda

L'operatore `.*` calcola il prodotto elemento per elemento e dà errore se le due variabili non hanno la stessa dimensione.

Esempi

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

```
Command Window
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 =

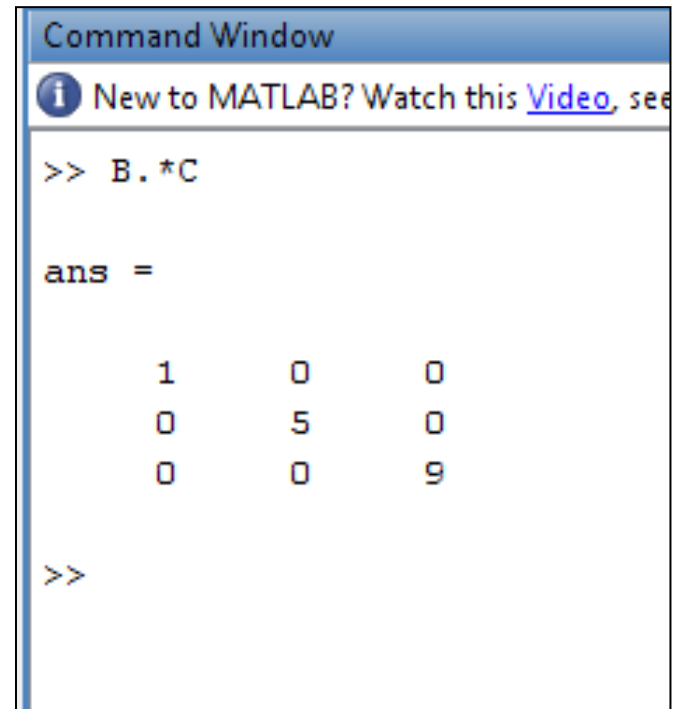
     1     2     3
     4     5     6
     7     8     9

>>
```

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
```

Vettori e matrici

Intervalli numerici: si usa il carattere speciale *due punti*

```
Command Window
>> t = -5:5
t =
    -5    -4    -3    -2    -1     0     1     2     3     4     5
>> t=-5:2:5
t =
    -5    -3    -1     1     3     5
>> t = -10:0.1:10;
>> size(t)
ans =
     1    201
fx>> |
```

Vettori e matrici

Funzioni su vettori: la maggior parte delle funzioni di Matlab è definita su vettori:

Command Window

```
>> t = -5:5;
>> x = sin(t)
x =
Columns 1 through 10
    0.9589    0.7568   -0.1411   -0.9093   -0.8415         0    0.8415    0.9093    0.1411   -0.7568
Column 11
   -0.9589
>> y = exp(t)
y =
Columns 1 through 10
    0.0067    0.0183    0.0498    0.1353    0.3679    1.0000    2.7183    7.3891   20.0855   54.5982
Column 11
  148.4132
fx>>
```

Notare l'uso dei due punti e del punto e virgola:

I due punti servono a creare vettori, con le due possibili sintassi:

```
vector = start:stop
```

```
vector = start:step:end
```

Se assente, lo step è uguale a 1

Invece il punto e virgola serve a sopprimere l'echo dell'istruzione (valore calcolato) e per separare le righe nella dichiarazione di vettori e matrici

Vettori e matrici

Qualche funzione utile:

Command Window

```
>> x = rand % Valori casuali tra 0 e 1
x =
    0.9058
>> X = rand(2,3) % Matrice 2x3 di valori casuali
X =
    0.1270    0.6324    0.2785
    0.9134    0.0975    0.5469
>> mCol = max(X) % massimo colonna per colonna
mCol =
    0.9134    0.6324    0.5469
>> XV = X(:) % "vettorizzazione" di X
XV =
    0.1270
    0.9134
    0.6324
    0.0975
    0.2785
    0.5469
>> max(XV)
ans =
    0.9134
fx >> |
```

Command Window

```
>> X = rand(2,3) % Matrice 2x3 di valori casuali
X =
    0.9575    0.1576    0.9572
    0.9649    0.9706    0.4854
>> mean(X) % media per colonne
ans =
    0.9612    0.5641    0.7213
>> mean(X,2) % media per righe
ans =
    0.6908
    0.8070
>> mean(X(:)) % media globale
ans =
    0.7489
>> sum(X) % somma per colonne
ans =
    1.9224    1.1282    1.4425
>> X > 0.5 % operatore logico
ans =
    2x3 logical array
    1    0    1
    1    1    0
```


Vettori e matrici

Come usare Matlab per risolvere un sistema di equazioni lineari

$$\begin{cases} 3x_1 + 4x_2 - 5x_3 = 7 \\ x_1 - x_3 = 1 \\ x_1 + x_2 + x_3 = -1 \end{cases}$$

```
Command Window
>> A = [3, 4, -5; 1, 0, -1; 1, 1, 1]; %coefficienti
>> b = [7; 1; -1]; % termine noto
>> x=A\b % Notare il backslash

x =

    -0.2000
     0.4000
    -1.2000

fx >> |
```

Notare il simbolo % che introduce un commento in linea

Provate a calcolare $A*x$

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
 - » **clearvars** – cancella tutte le variabili
 - » **close all** – chiude tutte le figure
 - » **load** - carica le variabili o i file di dati
 - » **save** – salva le variabili in un file .mat

Load & Save

➤ 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"

➤ COMANDO **load**

sintassi

load filename

carica tutte le variabili da un file

load filename X

loads only the variable X

load filename X Y Z

loads the variables X Y & Z

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

>> **doc *command_name***
Apri il browser della
documentazione

Nella Barra degli strumenti

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

**Sul web trovate quasi
sicuramente la risposta 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

fprintf e sprintf

fprintf e sprintf sono molto più potenti di display

Esempi:

Command Window

```
>> number = 10; name = 'John';  
>> fprintf('%s has %d apples\nWriting Pi with 4 decimal digits gives %1.4f\n', name, number, pi)  
John has 10 apples  
Writing Pi with 4 decimal digits gives 3.1416  
>> string = sprintf('%s has %d apples\nWriting Pi with 4 decimal digits gives %1.4f\n', name, number, pi);  
fx>> |
```

Il simbolo % introduce uno specificatore di formato

Il primo %s significa: stampa la prima variabile (name) come stringa

Il secondo %d: stampa la seconda (number) come numero intero

La terza %1.4f: stampa la terza variabile (pi) come numero decimale con 4 cifre dopo la virgola

Differenza tra fprintf e sprintf: il primo stampa a schermo (o su file)

il secondo crea una stringa

Grafici e Figure

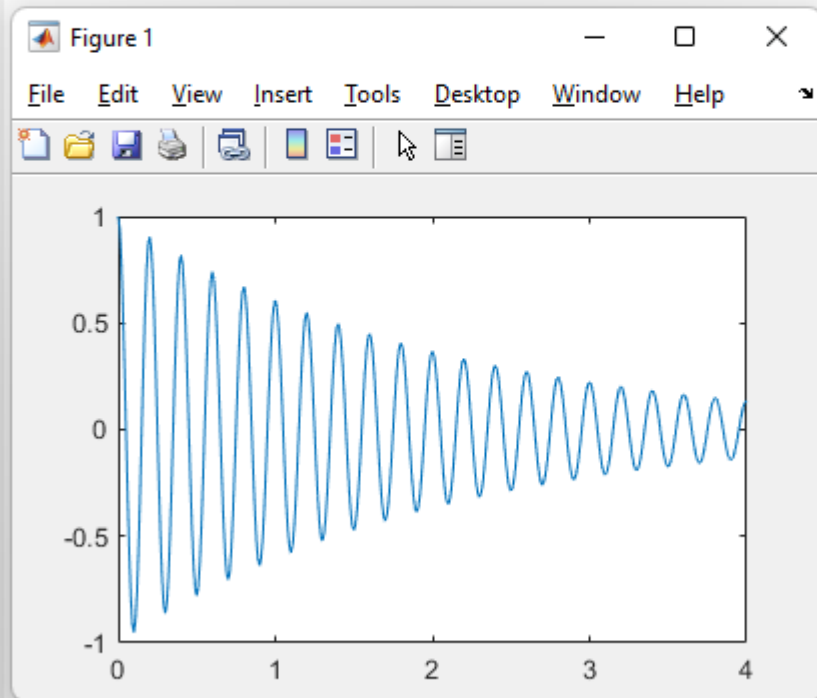
PLOT(X,Y,' ') = plotta il vettore Y versus il vettore X usando il tipo di linea indicata tra apici ' '

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

Command Window

```
>> t = 0:1e-2:4;  
>> x = (cos(2*pi*f*t)).*exp(-t/2); %notare il .*  
>> plot(t,x);  
fx >>
```



plot crea una nuova finestra (detta «figure» in matlab) e disegna il grafico di x in funzione di t

plot per default disegna nell'ultima figure attiva e, se non ce ne sono, ne crea una

Per forzare la creazione di una nuova figura, usare il comando figure

Per forzare plot ad usare una figura specifica, per esempio la figura 2, usare figure(2)

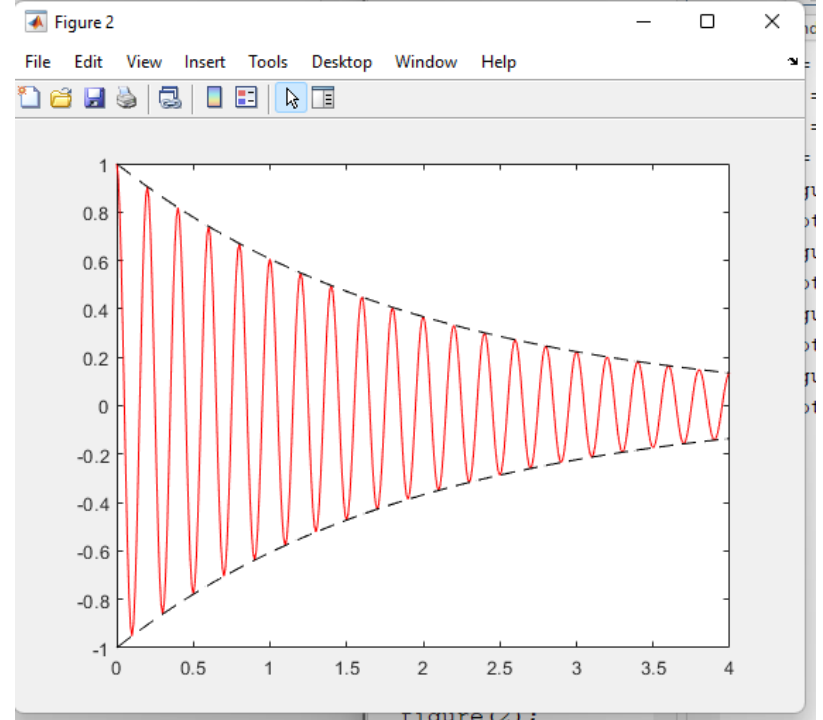
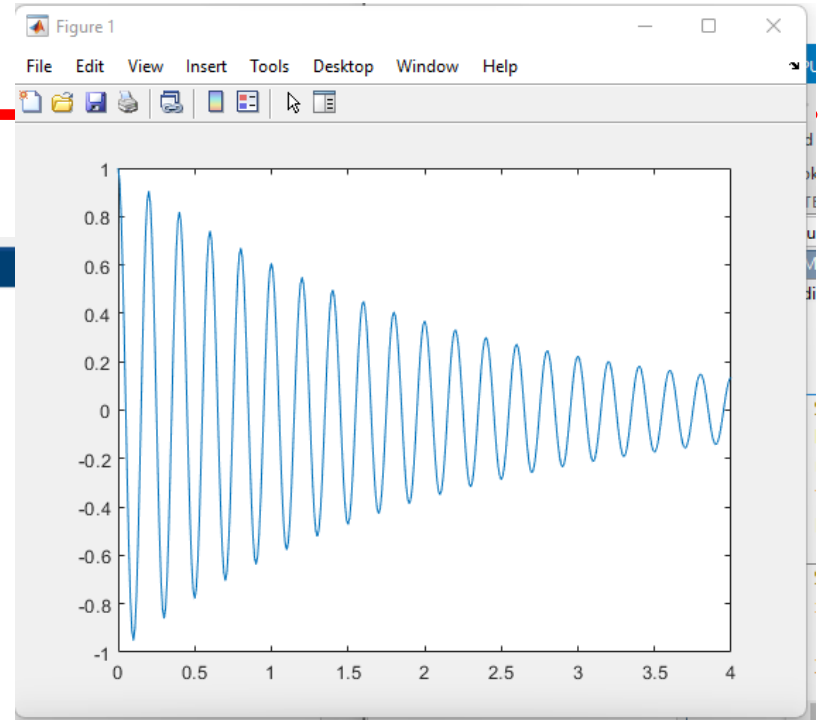
Grafici e Figure

Command Window

```
>> t = 0:1e-2:4;  
>> v1 = (cos(2*pi*f*t));  
>> v2 = exp(-t/2);  
>> x = v1.*v2;  
>> figure(1);  
>> plot(t,x);  
>> figure(2);  
>> plot(t,x,'r',t,v2,'--k',t,-v2,'--k');  
fx >> |
```

Nella figura 2, si disegnano tre curve, ognuna rappresentata da una terna: ascissa, ordinata, formato

formato è una stringa, dove si indica colore e/o formato della linea



Grafici e Figure

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

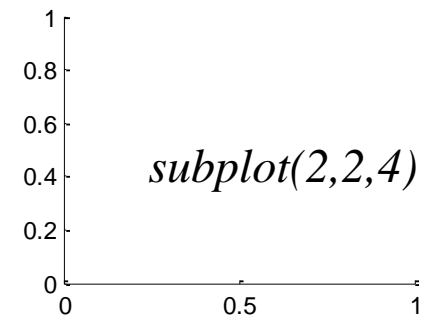
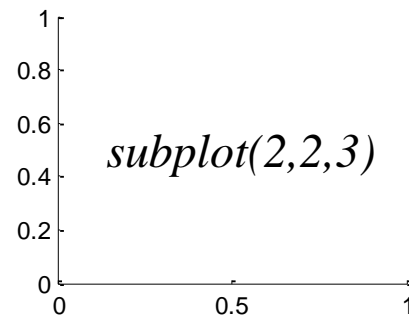
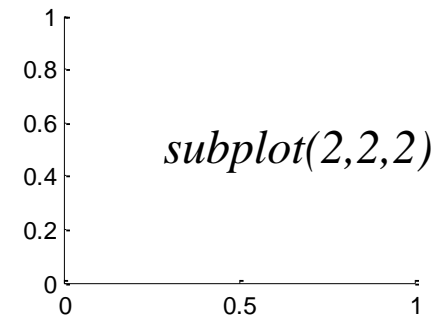
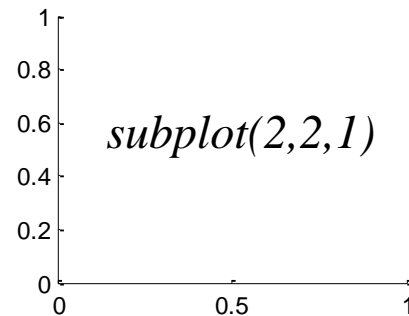
Esempio

subplot(2,2,1)

subplot(2,2,2)

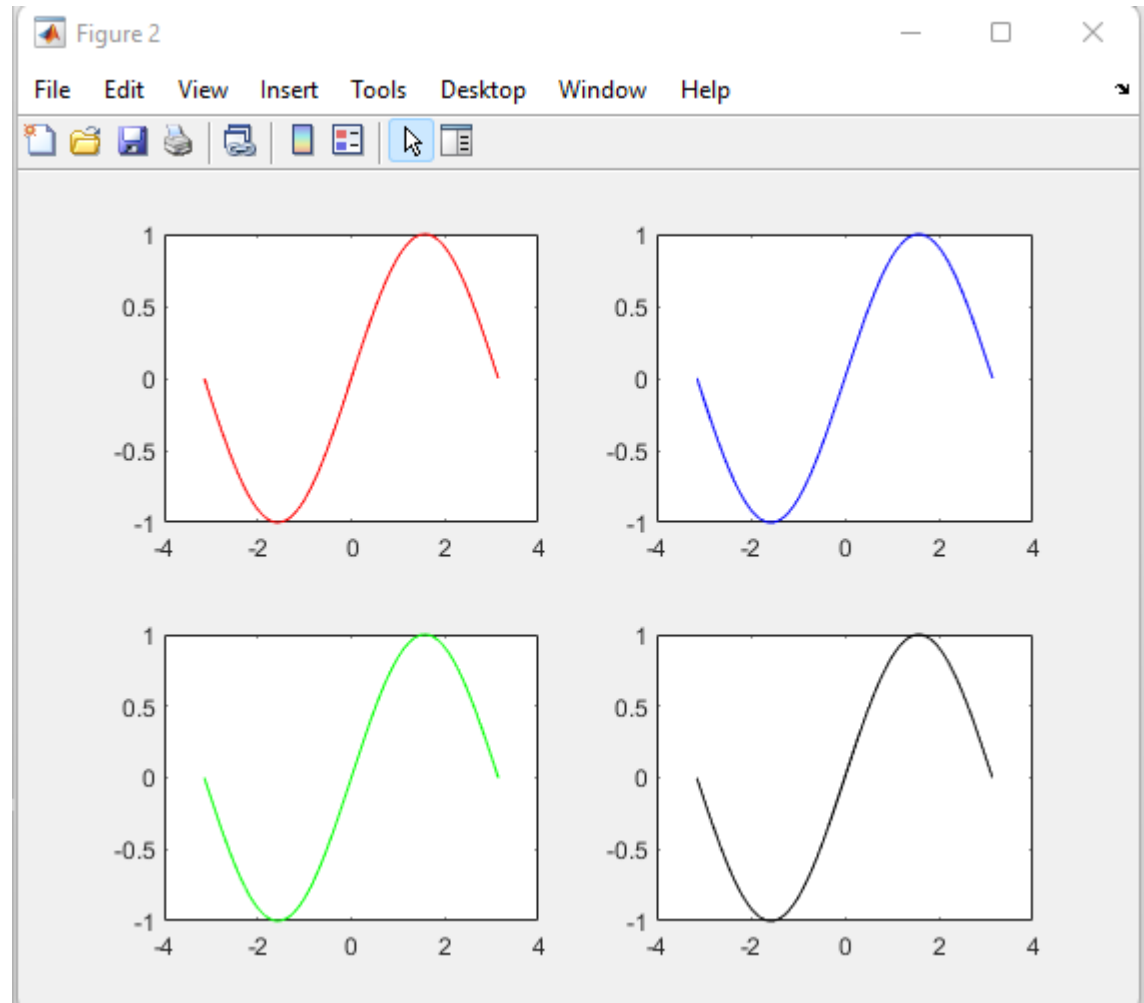
subplot(2,2,3)

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')
```

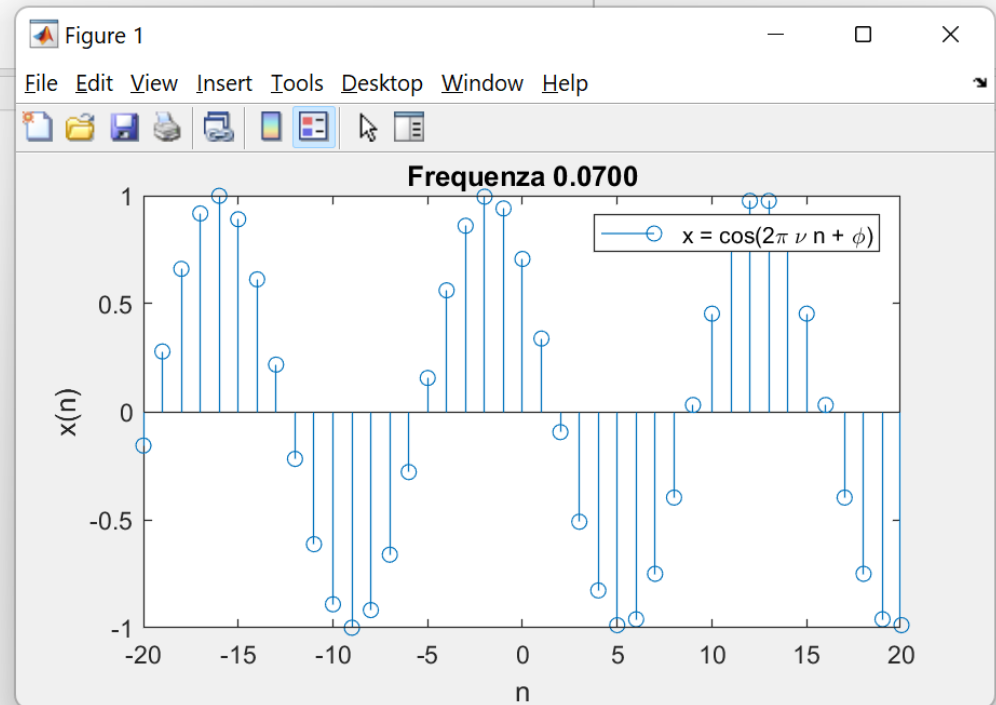


Grafici e Figure

stem crea grafici adeguati ai segnali discreti

Command Window

```
>> n = -20:20;  
>> nu = 0.07;  
>> phi = pi/4;  
>> x = cos(2*pi*nu*n + phi);  
>> figure  
>> stem(n,x);  
>> xlabel('n'); ylabel('x(n)');  
>> legend('x = cos(2\pi \nu n + \phi)');  
>> title(sprintf('Frequenza %3.4f',nu))  
fx>>
```



Cosa fanno i comandi xlabel, legend, title?

Qual è il periodo di questo segnale?

Ripetere i comandi, creando ogni volta una nuova figura, con

nu = 1.07, nu = 0.93 e nu = -0.07

Che cosa osservate?

Costrutti logici

if e switch-case

```
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
```

Controllo di flusso

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 le variabili **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 file di testo, script e funzioni
- 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
- I commenti possono cominciare anche sulla stessa linea di un comando
- 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. **Le ultime versioni di Matlab includono i LiveScripts che sono molto simili ai notebook di Python**

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

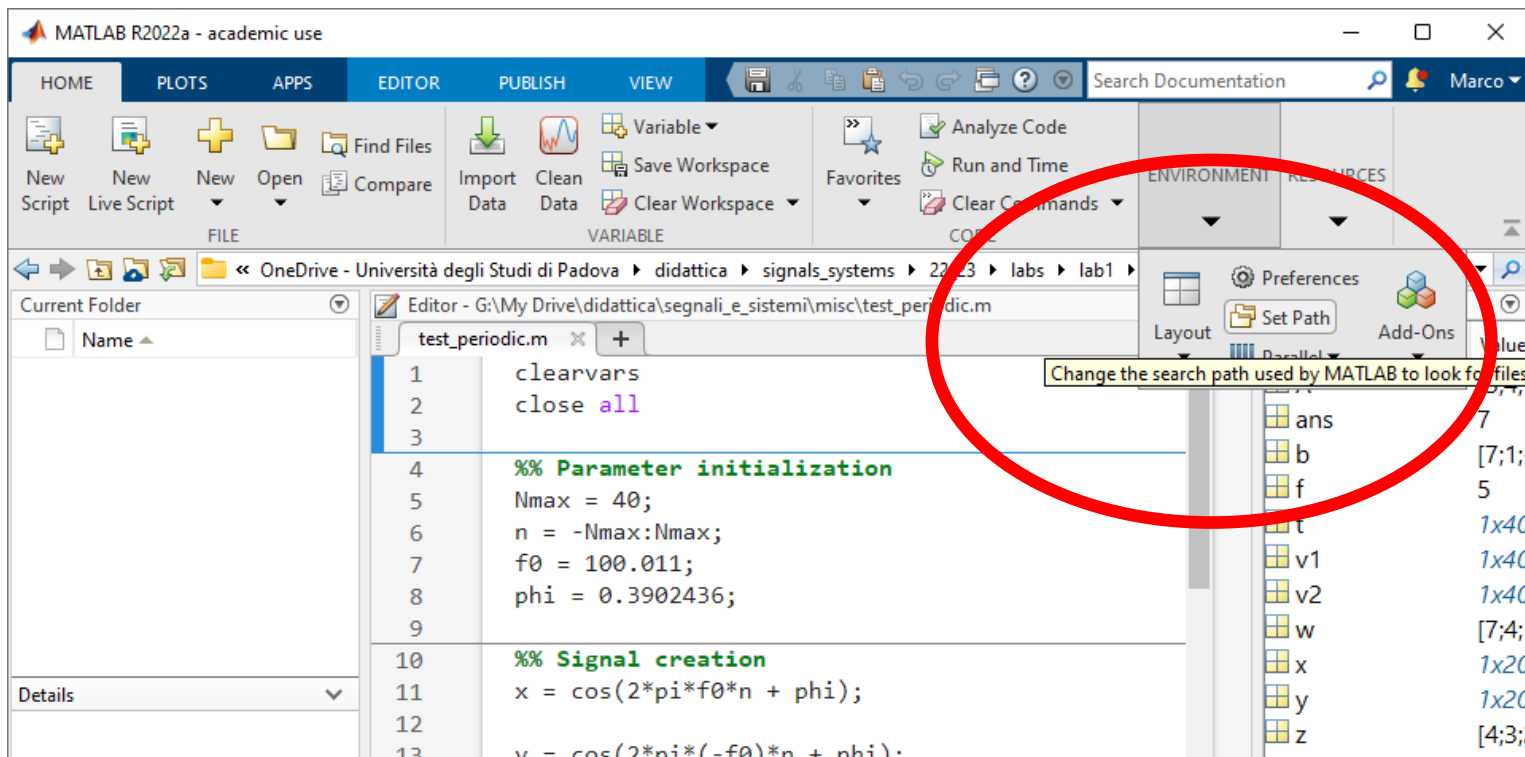
Una funzione può essere chiamata in un'altra funzione o in uno script se:

il file .m della funzione da chiamare è nella cartella corrente; oppure

il file .m della funzione da chiamare è nel path

File .m

Per gestire il path, usare il comando **Set Path** del menu Environment



Esempio di Function

Keyword: function

Function Name (same as file name .m)

Output Argument(s)

Input Argument(s)

Online Help

MATLAB
Code

```
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.
[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

Esempio di Script con *Sections*

univ di padova > didattica > signals_systems > 2022 > lab1

```
Editor - C:\Users\Marco Cagnazzo\OneDrive - Università degli Studi di Padova\didattica\signals_systems\2022\lab1\tmp.m
untitled x es1.m x tmp.m x untitled3.mlx x +
1      %% Grafici di sinuoidi discrete: studio del valore della frequenza numerica
2
3      %% Caso 1
4      nu = 0.07;
5      n = -20:20;
6      x = cos(2*pi*nu*n + phi);
7      % Creazione grafico
8      stem(n,x);
9      xlabel('n'); ylabel('x(n)');
10     title(sprintf('Frequenza %3.4f',nu));
11
12     %% Caso 2
13     nu = 0.93;
14     figure;
15     x = cos(2*pi*nu*n + phi);
16     stem(n,x);
17     title(sprintf('Frequenza %3.4f',nu))
```

Le sezioni sono delimitate da righe che cominciano con “%% ” (due percento e spazio)

Nell’Editor, premendo CTRL+ENTER si esegue unicamente la Sezione corrente

Con CTRL+SHIFT+ENTER si esegue la sezione corrente e si passa alla seguente

Nei due casi precedenti il file NON viene salvato: utile per delle prove!

Infine, con F5 o Run, il file viene salvato ed eseguito interamente

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=1:15`, più compatto)

- Creare i vettori:

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

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

come vettori colonna.

- Plottare **y1** rispetto a **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 o fprintf)

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**