Informatica per l'Ingegneria

Corsi M – N

A.A. 2023/2024

Angelo Cardellicchio

13 – Introduzione al MATLAB

Cosa è MATLAB

- MATLAB (MATrix LABoratory) fa riferimento a tre diverse cose:
  - Il *linguaggio MATLAB*, che utilizziamo per codificare i programmi.
  - L'*interprete* MATLAB, che viene invocato per eseguire i nostri programmi.
  - L'*ambiente di sviluppo integrato*, che permette di scrivere ed eseguire i programmi.
- È pensato (ed ottimizzato) per operare su matrici (ma include generiche funzionalità matematiche).
- MATLAB è uno strumento commerciale, su licenza NON gratuita.
  - La Student Edition è fornita dal Politecnico.

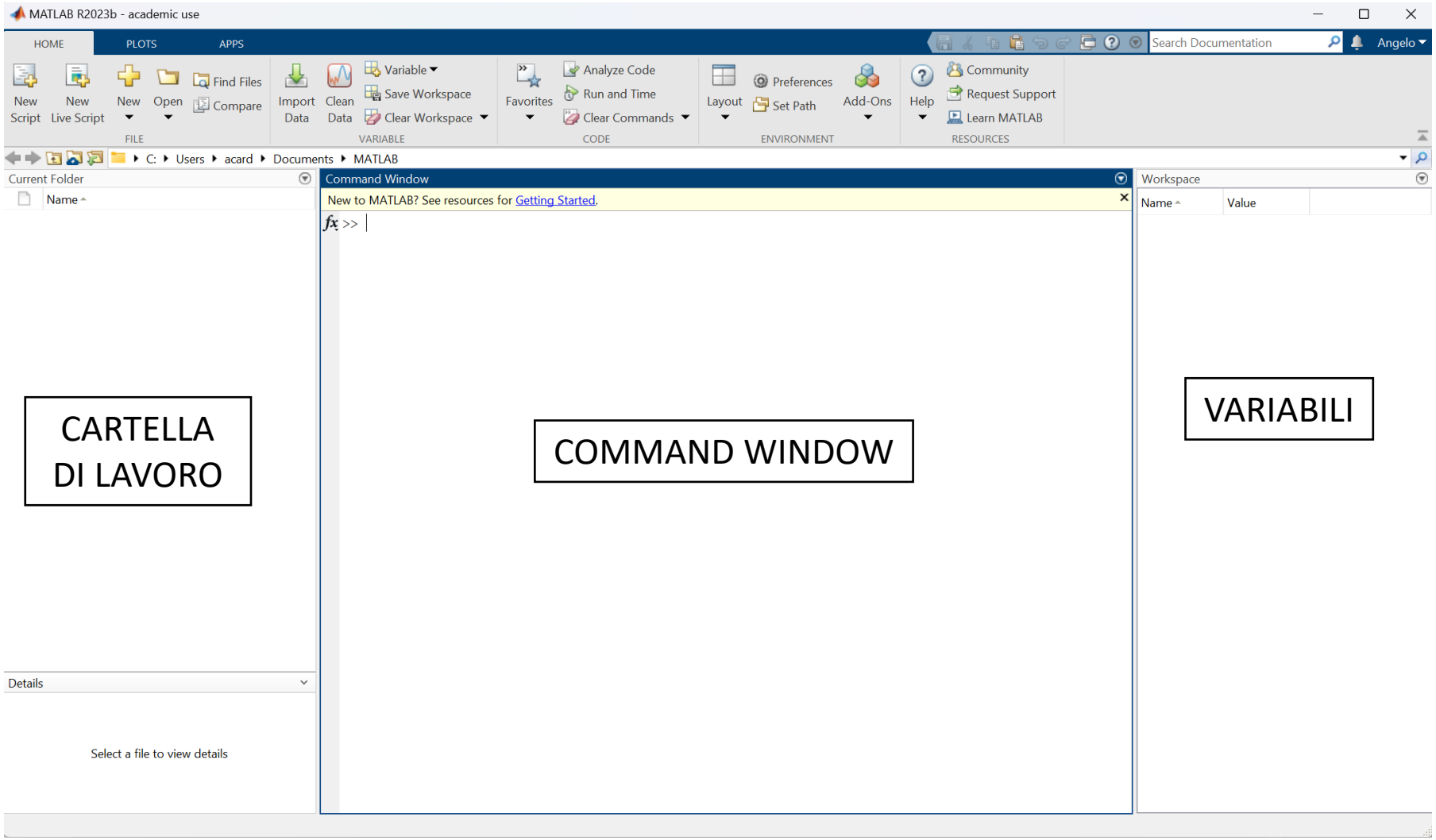
Caratteristiche del linguaggio MATLAB

- Linguaggio di alto livello
  - Simile a linguaggi di programmazione come Java o Python.
  - Possiede comandi sintetici per effettuare complesse elaborazioni numeriche.
- Linguaggio interpretato, comandi e istruzioni
  - NON sono tradotti in codice eseguibile dall'hardware.
  - Invia istruzioni ad un altro programma, l'**interprete**, che li analizza ed esegue azioni da essi descritte.

Caratteristiche del linguaggio MATLAB

- Linguaggio **dinamico** (non tipizzato).
- **NON occorre dichiarare le variabili:**
  - risultano definite al primo assegnamento;
  - vengono incluse in una struttura detta *tabella dei simboli*;
- **il tipo delle variabili è *dinamico*:**
  - a una variabile si possono assegnare, successivamente, valori di tipo diverso (scalari, stringhe, vettori, matrici…).

L’interfaccia MATLAB



Le istruzioni e la Command Window

- Le **istruzioni** possono essere **inviate** direttamente **all'interprete** se scritte nella Command Window (ovvero dopo il simbolo **>>**).
  - La Command Window è come una 'super calcolatrice'.
  - La Command Window ha un'interfaccia testuale che inizia con **>>**.

Esempio: le operazioni aritmetiche

- Nella Command Window è possibile eseguire qualsiasi operazione aritmetica.

```
>> 5 + 7

ans =

    12
```

```
>> 5 * 7

ans =

    35
```

```
>> 5 ^ 7

ans =

        78125
```

ELEVAZIONE A POTENZA

```
>> 5 / 7

ans =

    0.7143
```

```
>> 'a' + 2

ans =

    99
```

I caratteri alfanumerici si indicano con l'apice singolo: sono sempre legati agli interi mediante la tabella ASCII

Le istruzioni e il codice sorgente (1)

- Le istruzioni possono essere contenute in un **file sorgente**, in particolare:
  - uno *script*;
  - una *funzione*.
- Vengono eseguite in maniera sequenziale.
- L'esecuzione di un codice sorgente può essere visto come l'inserimento automatizzato delle varie istruzioni nella Command Window.

Le istruzioni e il codice sorgente (2)

- Le istruzioni *possono* terminare con un punto e virgola.
- Di default, il risultato di ogni istruzione viene visualizzato nella command window.
- Il **;** blocca la visualizzazione del risultato dell'istruzione. Usarlo comporta:
  - *maggiore velocità di esecuzione;*
  - *visualizzazione più compatta*.
- La best practice è quella di inserire sempre il **;** a meno che non si voglia ispezionare il valore di una variabile a scopo di debugging.

Istruzione di assegnamento

- Come nel C:

nomeVariabile = espressione

- A differenza del C:
  - non si deve (non è possibile) dichiarare la variabile **nomeVariabile** prima dell'assegnamento;
  - l'assegnamento comporta una dichiarazione implicita della variabile **nomeVariabile**;
  - è possibile eseguire assegnamento tra array;
  - non è richiesto i ; al termine dell'istruzione;
  - il risultato di un'operazione che non comporta un assegnamento viene assegnato alla variabile **ans**.

Assegnamento ed inizializzazione

- Quando si assegna un valore ad una variabile che non è stata inizializzata (e.g., **a**), la **variabile viene creata**.

```
>> a = 7

a =

     7
```

- Ovviamente, non è possibile assegnare ad una variabile il valore di un'altra variabile non esistente.

```
>> a = v

     Undefined function or variable 'v'.
```

Primi comandi (1)

- Definizione di una variabile scalare:

```
>> a = 3
```

- Sulla command window appare:

```
a =

     3
```

- Se scriviamo:

```
>> a = 3;
```

- L'echo viene eliminato.

Primi comandi (2)

- Definizione di un vettore riga:

```
>> v = [1 2 5 7];
```

- Definizione di una matrice:

```
>> A = [1 3 6; 4 0.4 12];
```

Gestione variabili

- Ogni variabile è memorizzata nel workspace
- Alla chiusura di MATLAB, si perde il lavoro!
- Per ispezionare il workspace:

```
>> who

Your variables are:

A  v

>> whos
  Name        Size              Bytes  Class     Attributes

  A           2x3                  48  double
  v           1x4                  32  double
```

```
>> clear v
>> clear all
>> save n_file
>> save n_f v
```

m-files

- MATLAB gestisce file di estensione *.m
- **Script:** sequenza di comandi, si avviano dal workspace

```
>> nome_file
```

- **Funzione:** riceve un input, restituisce un output

```
>> [a, b, c] = fn(C, D)
```

Istruzioni per m-files

- Istruzione con condizione

```
if [condizione]
        [istruzioni]
else
        [istruzioni]
end
```

- Cicli for e while

```
for n=1:100
        [istruzioni]
end

while [condizione]
        [istruzioni]
end
```

Istruzioni di controllo (1)

- **`for`:** ripetizione di un insieme di istruzioni per un numero predeterminato di iterazioni (deve terminare con **`end`**).
- **`while`:** ripetizione di un insieme di istruzioni finchè una determinata condizione rimane vera (deve terminare con **`end`**).
- **`if`:** istruzione condizionale (deve terminare con **`end`**) può utilizzare **`else`** ed **`elseif`**.
- **`else`:** identifica un blocco di istruzioni alternative.
- **`elseif`:** esegue un blocco di istruzioni se è soddisfatta una condizione alternativa.
- **`end`:** termina le istruzioni **`if`**, **`for`** e **`while`**.

Istruzioni di controllo (2)

- **`break`**: termina l'esecuzione di un ciclo **`for`** o **`while`**.
- **`switch`**: indirizza il controllo di un programma confrontando l'espressione di input con le espressioni associate alle clausole **`case`**.
- **`case`**: utilizzato con switch per controllare l'esecuzione di un programma.
- **`findstr('s1', 's2')`**: date le stringhe di caratteri **`s1`** ed **`s2`**, trova gli indici iniziali di qualsiasi ricorrenza della stringa più corta all'interno di quella più lunga.

Operatori relazionali

<	minore
<=	minore o uguale
>	maggiore
>=	maggiore o uguale
==	uguale
~=	diverso

- N.B. il simbolo ~ si digita tenendo premuto il tasto alt e digitando 126 sul tastierino numerico.

Domande?

42