Esercitazioni di MatLab

Supporto alla didattica - Lezione 1

Serena Crisci

Università di Ferrara - Dip. di Matematica e Informatica
e-mail: serena.crisci@unife.it

A.A. 2017/18

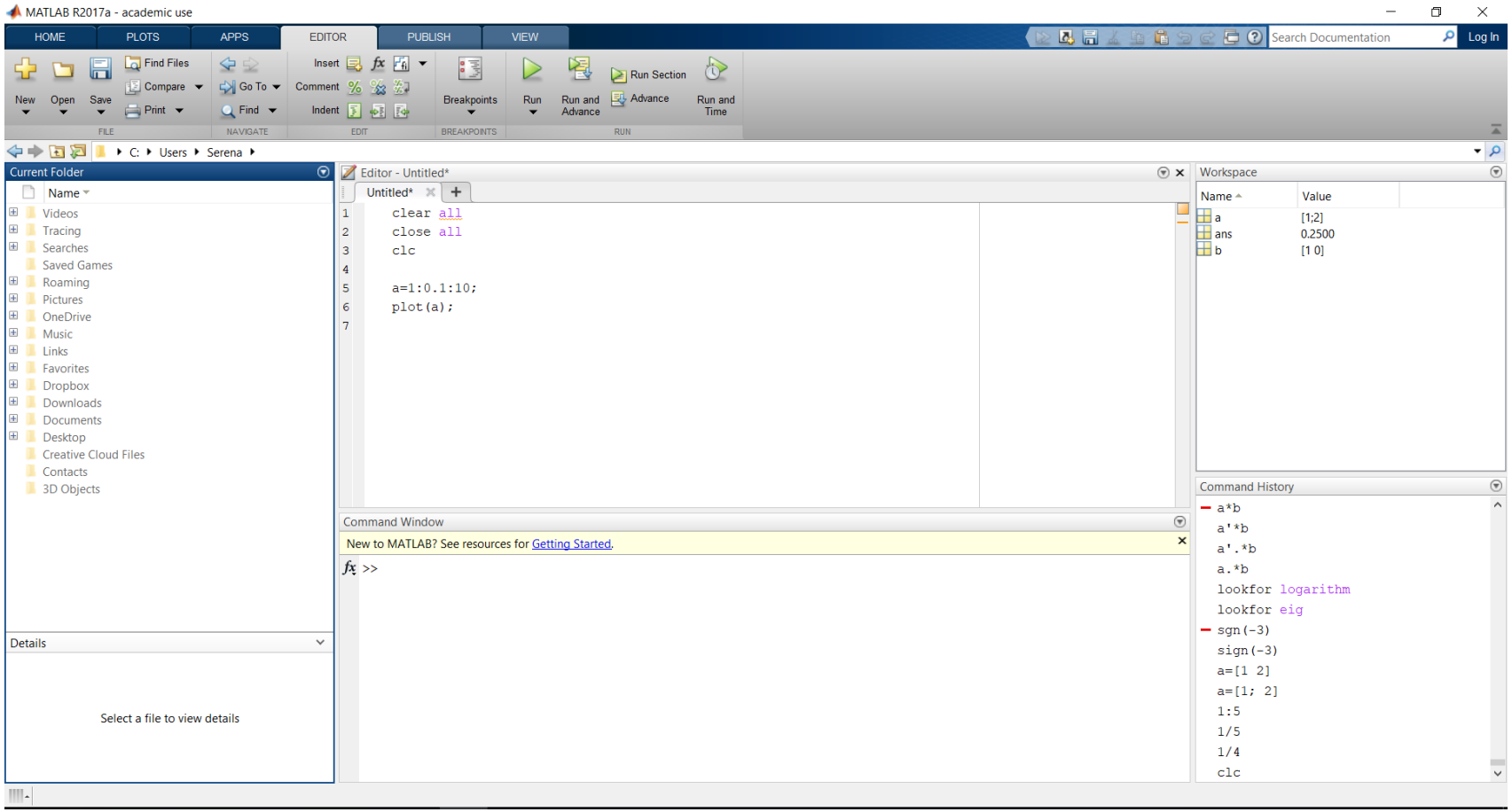
Sommario

Cos'è MatLab

Matlab (Matrix Laboratory) è un ambiente integrato per il calcolo scientifico, basato sul calcolo matriciale; consente di eseguire un'istruzione o comando per volta, tali comandi permettono di definire variabili, valutare formule, disegnare grafici 2D e 3D a schermo...e molto altro!

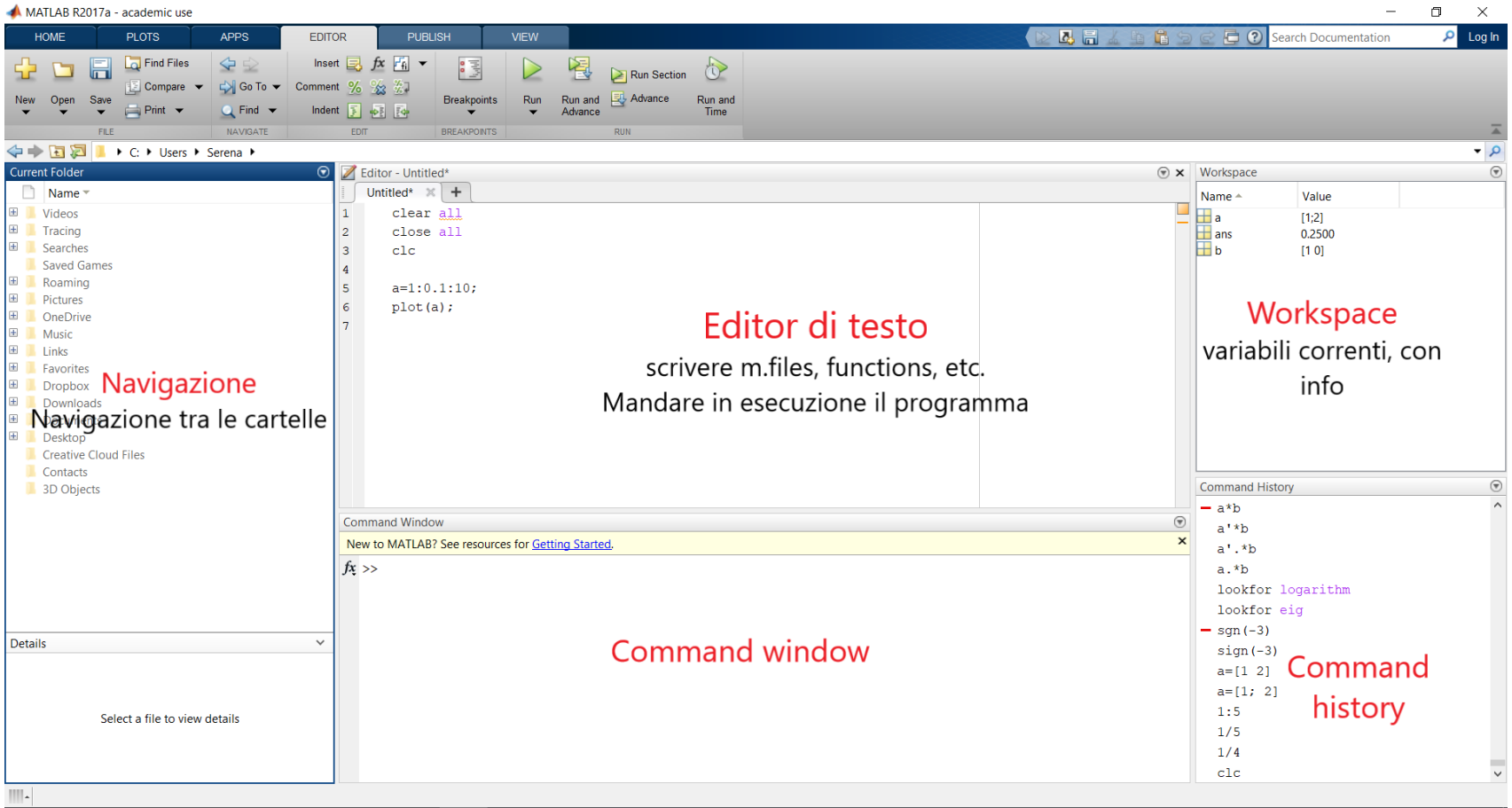
Matlab è un ambiente per studiare soluzioni numeriche di problemi di tipo fisico, economico, statistico, ingegneristico, in molto campi delle scienze applicate. Per aiutare gli utenti nella soluzione di questi problemi, sono stati creati vari Toolbox, ovverossia librerie con funzioni già implementate per affrontare le tematiche particolari (ad esempio, lo Statistic Toolbox è stato creato per effettuare analisi statistiche di dati.) Per maggiori informazioni, visitare ▸ www.mathworks.com

Ambiente MatLab



Ogni finestra può essere estratta (unlocked), integrata (locked), massimizzata, minimizzata, chiusa, etc.

Ambiente MatLab



Ogni finestra può essere estratta (unlocked), integrata (locked), massimizzata, minimizzata, chiusa, etc.

Navigazione

Nel menù di navigazione è possibile controllare in quale directory si sta lavorando.

Questo è importante in quanto se si vanno a caricare dati o funzioni non presenti nel path corrente MatLab restituisce un messaggio di errore.

In questa finestra è possibile selezionare la cartella di lavoro, aprire m-files, caricare dati salvati mediante il doppio clic del mouse.

Alcuni comandi da utilizzare nella finestra di comando per la navigazione sono i seguenti:
`cd name_dir` entra nella cartella `name_dir`;
`cd ..` sale di un livello;
`load name_file.mat` carica dati salvati nel file `name_file.mat`;
`edit name_script.m` apre nella finestra dell'editor il file `name_script.m`.

Workspace & Command history

Nel **Workspace** viene visualizzato l'elenco delle variabili.

Viene visualizzato il nome della variabile, il valore (e in aggiunta la dimensione, il tipo, massimo, il valore minimo, etc.)

Cliccando due volte su di una variabile nel workspace, si apre una finestra simile ad un foglio elettronico contenente i valori della variabile. In questo ambiente 'e possibile modificare la variabile presa in esame.

Nella **Command History** sono presenti tutti i comandi digitati. Cliccando due volte su di un comando presente nell'elenco tale comando viene eseguito.

Command window

La **Command Line** o **Shell** è l'ambiente con cui si interagisce effettivamente con MatLab. Sulla linea di comando si definiscono le variabili, si richiamano funzioni, si eseguono script...

```
>> 3+2=
ans =
     5
>>
>> sin (3/2 * pi ) + log(25) - tan(2)
ans =
    4.4039
>>
```

Si può utilizzare Matlab come una calcolatrice...

... anche di alto livello!

Per eseguire un comando si digita tale comando sulla tastiera e poi si preme `Invio`. Per eseguire più comandi sulla stessa linea si separano con una virgola; i risultati vengono visualizzati in ordine.

`ans` è la variabile temporanea in cui MatLab memorizza il risultato dell'ultima istruzione digitata. Viene sovrascritta ogni volta!

Command window: alcuni comandi utili

```
>> pwd
```

Directory di lavoro

```
>> help name_fun
```

Apre una breve descrizione della funzione `name_fun`

```
>> doc
```

Apre il browser con il manuale di MatLab

Editor

I programmi Matlab possono raggiungere facilmente le centinaia di righe di codice, quindi è necessario poter disporre di un file sorgente in cui memorizzare le istruzioni da eseguire. Nell'**Editor** è possibile salvare un file di testo con estensione `.m` in cui scrivere le istruzioni del nostro algoritmo. L'editor di MatLab è un comune editor di testo con solo alcune funzioni aggiuntive, comode per poter lavorare con questo linguaggio di programmazione.

Nell'editor son presenti strumenti di debug e di esecuzione del programma, utili quando vi sono errori difficili da trovare.

Una volta scritte le istruzioni, queste possono essere richiamate digitando il nome del file nella shell: in questo modo le istruzioni presenti nel file verranno eseguite in sequenza, una alla volta.

Per poter utilizzare un m-file, dobbiamo trovarci nella stessa directory in cui è stato salvato!

Variabili numeriche

Digitando in successione due comandi »3+2 e »7+3 il risultato viene memorizzato nella variabile ans, che memorizza solo il risultato dell'ultima operazione.
Se volessimo memorizzare questi risultati dobbiamo dichiarare una variabile:

```
>> a= 3+2
a=
  5
>> a= 3+2;

>> a= 3+2;
>> a= 10+3;
```

Il risultato viene memorizzato nella variabile a e viene anche visualizzato.

Il risultato viene memorizzato nella variabile a ma non viene visualizzato.

In questo modo il valore assegnato ad a viene sovrascritto:

Per memorizzare un valore in una variabile:

```
nome_variabile = valore_variabile
```

Variabili numeriche

MatLab è case sensitive: la variabile `M` è diversa dalla variabile `m`. Inoltre, il nome di una variabile non può iniziare con una cifra.

Variabile NULL

Utilizzando la seguente istruzione:

```
>> x = [];
```

si crea una variabile vuota (NULL) che può contenere dati di qualsiasi tipo (può essere riempita con un'assegnazione successiva).

Vettori

MatLab è ottimizzato per il calcolo matriciale, tutti i dati vengono visti come matrici.

I vettori sono particolari matrici.

Riepilogo di alcuni comandi di base:

```
d = linspace(1,50,100)
```

Crea il vettore d di 100 elementi, il primo elemento 1 e l'ultimo è 50. Il passo è dato da $d_i - d_{i-1}$ (automaticamente)

```
dot(a,b)
```

Calcola il prodotto scalare tra i vettori a e b

```
sum(v)
```

Calcola la somma degli elementi del suo argomento

Vettori

Riepilogo di alcuni comandi di base:

`prod(v)`	Calcola il prodotto degli elementi del suo argomento.)
`numel(v)`	Calcola il numero degli elementi del suo argomento.
`length(v)`	Restituisce la lunghezza del vettore
`size(v)`	Restituisce le dimensioni del suo argomento
`find(v<10)`	La funzione `find(logic_expr)` restituisce gli indici per cui `logic_expr` è verificata.

Operatori di confronto

Gli operatori di confronto in Matlab sono

- == confronto
- < maggiore
- > minore
- >= maggiore o uguale
- <= minore o uguale
- ~= diverso (per digitare ~ in ambiente Windows: ALT+0126. In ambiente Linux: ALTGR+ì. In ambiente MacOS ALT+5).

Esercizi sui vettori

1) Generare i vettori $v = (1, 2, \ldots, 6)^T$ e $w = (1, 2, \ldots, 6)^T$ (vettori colonna) [in tre modi diversi!]. Calcolare i seguenti punti:

- calcolare $a = v + w$ e $b = 4v$ e determinarne la lunghezza
- calcolare il prodotto di v per w elemento per elemento e salvare il risultato nella variabile c
- dividere ogni elemento di w per 2 e salvare il risultato nella variabile d
- dividere ogni elemento di v per il corrispondente elemento di w e salvare il risultato nella variabile e

Esercizi sui vettori

- calcolare il prodotto scalare di v e di w, salvare il risultato in f
- calcolare $g = 2v - 6w$
- memorizzare negli elementi di posto pari del vettore $g1$ gli elementi di posto pari di d e negli elementi di posto dispari di $g1$ gli elementi di posto dispari di c
- creare il vettore h con 5 copie del vettore v
- osservare il comportamento del comando $h([6 : 6 : end])$
- sostituire 0 negli elementi con indice multiplo di 5 in h e 1 negli elementi con indice multiplo di 6
- dato $u = (1, 2, \ldots, 6)$ (vettore riga) calcolare $u * w$ e $w * u$: osservare i risultati e capire cosa succede.

Esercizi sulle matrici

Creare una matrice A di dimensioni 3×3, che abbia tutti gli elementi uguali a zero tranne quelli sulla diagonale.

- Siano a e b i vettori di elementi $[1, 2, 7]$ e $[3, 4, 1]$, rispettivamente;
- costruire la matrice B che abbia b come prima riga e a come seconda;
- sia C la trasposta di B;
- creare un vettore colonna w di 3 elementi di numeri casuali compresi tra 0 e 3;
- aggiungere a C una terza colonna corrispondente al vettore w;
- elevare al cubo gli elementi della matrice C;
- moltiplicare le matrici A e C, ottenendo la matrice D;
- estrarre la sottomatrice E 2×2, composta dagli elementi $(1, 1), (1, 2), (2, 1), (2, 2)$ di D.

Script

Supponiamo di dover ripetere le stesse istruzioni con input diversi.

Si dovrebbe ripartire dall'inizio e riscrivere tutte le istruzioni?

M-file

Per fortuna no! Basta salvare la lista delle istruzioni in quello che viene chiamato M-file. Una volta scritta questa lista e salvato il file, possiamo richiamare questa serie di comandi semplicemente scrivendo il nome del M-file nella shell di comando.

Questo tipo di M-file prende il nome di script.

Script: directory, workspace e variabili

Alcune accortezze:

- Per eseguire uno script, è necessario essere nella directory dove tale script è stato salvato
- uno script può accedere a tutte le variabile del workspace corrente, modificandole. Inoltre, tutte le varibili create nello script rimarranno in memoria.
- uno script può chiamare un altro script

Script: directory, workspace e variabili

Alcune accortezze:

- per rendere effettive le modifiche fatte ad uno script, è necessario salvarle. Quando le modifiche fatte non sono state salvate, viene eseguita l'ultima versione dello script. Si nota che non si è effettuato il salvataggio delle modifiche perchè nella finestra dell'editor accanto al nome del file appare un asterisco.
- per avviare uno script, si può digitare il nome del file nella shell oppure cliccare sul bottone a forma di play sulla barra di modifica dell'editor: in questo modo, si salvano le modifiche e si eseguono le istruzioni contenute nello script.
- Le prime tre righe del primo script che viene chiamato dovrebbe contenere le seguenti istruzioni:
  `clear all`: pulisce il workspace
  `close all`: chiude tutte le finestre aperte
  `clc`: pulisce la shell di comando