



 POLITECNICO DI MILANO

Dipartimento di  
Elettronica e Informazione

# MATLAB: Script, Input/Output

Matteo Ferroni  
matteo.ferroni@polimi.it

20/11/2018



POLITECNICO  
DI MILANO





# Caratteristiche del linguaggio di Matlab/ Octave (1)



- Linguaggio di **alto livello**
  - simile a linguaggi di programmazione C, Java, Pascal
  - possiede comandi sintetici per effettuare complesse elaborazioni numeriche
- Linguaggio **interpretato**, comandi e istruzioni
  - NON tradotti in codice eseguibile dall'hardware
  - MA manipolati da un programma (l'interprete) che li analizza ed esegue azioni da essi descritte
- Linguaggio **dinamico**
  - NON occorre dichiarare le variabili
    - risultano definite dal punto in cui vengono introdotte
      - e 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...)



Le 3 cose fondamentali:

- 1) In Matlab/Octave tutto è un **array**
- 2) Ogni cosa è un **array** in Matlab/Octave
- 3) Ripetete con me: “tutto è un **array**”

- Script
- Input/Output



- Uno **script** è un file di testo contenente una sequenza di comandi MATLAB
  - non deve contenere caratteri di formattazione (solo testo puro)
  - viene salvato con estensione .m
- I comandi all'interno di uno script sono eseguiti **sequenzialmente**, come se fossero scritti nella finestra dei comandi
  - Per eseguire il file si digita il suo nome (senza .m)
  - I risultati appaiono nella finestra dei comandi (se non usiamo il ; )



- Uno script può
  - essere ri-eseguito
  - essere facilmente modificato
  - essere spedito a qualcuno
- Uno script NON
  - accetta variabili di input
  - genera variabili di output
- Uno script opera sulle **variabili del workspace**, che può essere arricchito introducendone di nuove durante l'esecuzione dello script stesso

- Può essere creato utilizzando un qualsiasi editor di testo
  - Ricordarsi di salvare il file come “solo testo” e di dare l'**estensione .m**
  - Il file di script deve essere presente nella **directory corrente** o il cammino (path) che identifica la directory in cui si trova lo script deve essere salvato tra i path di Matlab



- Il **nome** del file deve cominciare con una lettera e può contenere cifre e il carattere underscore, fino a 31 caratteri
- **Non** dare lo *stesso nome al file di script e a una variabile*
- **Non** chiamare uno script con lo *stesso nome di un comando o funzione MATLAB*
- Per verificare se esiste già qualcosa che ha un certo nome si può utilizzare la funzione *exist*





# Editor per script in GUI-Octave



POLITECNICO  
DI MILANO

DIPARTIMENTO DI ELETTRONICA E INFORMAZIONE

Virtual Appliance [Running]

**Figure 1**

cubica

ordinata

ascissa

View -12,1620, -1136,05

File list

- PrimoScript.m

Small Clipboard

xlabel("");

PrimoScript.m

```
1 x= -10:0.1:10;  
2 y=x.^3;  
3 plot(x,y);  
4 xlabel('ascissa');  
5 ylabel('ordinata');  
6 title('cubica');
```

Line: 6 Col: 17

```
>>> y=x.^3;  
>>> plot(x,y);  
>>> xlabel('ascisse');  
>>> ylabel('ordinate');  
>>> title('cubica');  
>>> PrimoScript  
>>> PrimoScript;  
>>>
```

Command line>>

QtOctave [Empty] Editor gplt

00:03 Left



- Sezione dei **commenti**:
  - Il nome del programma e le parole chiave, nella *prima riga*
  - La data di creazione e i nomi degli autori nella *seconda riga*
  - La definizione dei nomi delle *variabili* per ogni variabile di input e di output
  - Il nome di ogni *funzione* creata dall'utente che viene usata nel programma
  - Il comando help visualizza tutta la sezione dei commenti all'inizio dello script
- Sezione di **Input**: inserimento dei dati in input e/o uso di funzioni di input
- Sezione di **calcolo**
- Sezione di **output**: uso si funzioni per visualizzare i risultati del programma



- Gli script *non accettano argomenti* d'entrata e d'uscita
- Usano
  - variabili *già presenti* nel workspace
  - variabili *acquisite da tastiera o file*
  - *nuove variabili* introdotte nello script
- Le **variabili interne** allo script diventano variabili del workspace
  - **Permangono dopo l'esecuzione** dello script



- **Calcoli** matematici
- **Assegnamenti**
- Strutture di **controllo**
  - Condizioni
  - Cicli
- Comandi per la costruzione di **grafici**
- Chiamate a **funzioni**

- ~~Script~~
- Input/Output



- Funzione **input**
  - `valore = input('inserisci un valore ');`
- Matlab *stampa a video* la stringa 'inserisci un valore' ed aspetta di ricevere un dato
  - Valore **scalare**
  - **Array** racchiuso tra [ e ]
  - **Stringa** racchiusa tra ' e '
- Il dato inserito dall'utente viene memorizzato nella **variabile valore**

- I **risultati** di un'operazione sono mostrati immediatamente se non si inserisce il ;
- Altri due modi
  - **disp**
    - accetta come parametro un array. Se questo array è di tipo char, lo stampa
    - viene usato in congiunzione con num2str
    - Esempio:
      - *str = ['il valore di pi e` ' num2str(pi)];*
      - *disp(str);*
    - Stampa: “il valore di pi e` 3.1416”

- ...altro modo
  - **fprintf**
    - `fprintf('Il valore di pi e` %f \n', pi);`
    - stringhe di formato: %d (**interi**), %e (**formato esponenziale**), %f (**virgola mobile**), %g (il più corto tra il formato esponenziale e quello in virgola mobile)
- *disp* vs. *fprintf*
  - **disp** è in grado di stampare anche **valori complessi**
    - `x=2*(1-2*i)^3;`
    - `str=['disp: x = ' num2str(x)];`
    - `disp(str); disp: x = -22+4i`
  - **fprintf** ne stampa solo la **parte reale**
    - `fprintf('fprintf: x = %8.4f\n', x);`
      - `fprintf: x = -22.0000`





- Tipi di **file** gestiti
  - ascii = file di testo
  - .mat = file proprietari di Matlab
- Comandi più semplici da usare
  - **save**
  - **load**



- **file .mat**
  - **save filename**: salva su filename.mat tutte le variabili contenute nello spazio di lavoro
  - **save filename array1 array2**: salva su filename.mat le variabili array1 e array2
- I file .mat hanno un formato **compatto**
- **Contengono**
  - Nomi, tipi e valori di ogni variabile
  - La dimensione degli array
  - ... in generale, **tutto ciò che serve** per ripristinare lo stato dello spazio di lavoro
  - Possono essere portati da un computer all'altro, anche con sistemi operativi diversi



- **Limitazione** dei file .mat
  - E` un formato proprietario di MATLAB.
  - Non è utilizzabile per leggere/scrivere dati con un altro programma
    - Es, un editor di testi, excel, ...
- **Soluzione**
  - Uso dei **file di testo** (ascii), esempio
    - $x = [1.23 \ 3.14 \ 6.28; -5.1 \ 7.00 \ 0];$
    - **save -ascii filename.dat x;**
      - Produce il file filename.dat organizzato come segue

1.23000000e+000	3.14000000e+000	6.28000000e+000
-5.10000000e+000	7.00000000e+000	0.00000000e+000
- Nota: si può usare *qualsiasi estensione* per questi file, è buona norma distinguerli dai file .mat

- **load** carica dati da file nello spazio di lavoro di MATLAB
  - **load filename**: carica nello spazio di lavoro tutte le variabili nel file
  - *load filename x y*: carica nello spazio di lavoro solo le variabili x ed y
  - Se filename non ha estensione o ha estensione .mat, viene trattato come un **file .mat**
  - File ascii
    - **load filename.dat**: crea una variabile di nome filename che conterrà i dati in filename.dat
    - Il file deve contenere dati *separati da virgole o spazi*



- $A = \mathbf{xlsread}('filename')$ 
  - importa il file di Microsoft Excel *filename.xls* nella matrice  $A$
- Alcuni fogli di calcolo salvano i dati nel formato *.wk1*  
 $M = \mathbf{wk1read}('filename')$   
per importare questi dati nella matrice  $M$