***Funzioni utili***

clear – Pulisce le variabili in memoria

clc – Utile per pulire la command window

sort – Per ordinare magari un array

abs – Valore assoluto

help – Per ricavare la funzionalità di un certo comando

size – Ricavarsi la dimensione di un array

median – Mediana

mean – Media (può avere due argomenti e il secondo argomento indica di quanti elementi fare la media)

***05/04/2022: Lezione 1: Introduzione a Matlab e primi programmi***

In poche parole, aperto Matlab si ha la *Command Window*, in cui si immettono i comandi, il proprio elenco di variabili (Working Space) e la *Current Folder*, cartella di lavoro attuale e contenuto, operando volendo con un Editor di testo

Matlab legge i file di tipo “.m”, soprattutto script (operazioni su variabili in memoria) e funzioni (calcolo esplicito dei dati di output partendo dagli input). In particolare si consiglia di lavorare nella stessa cartella dei file dove sono gli script, perché Matlab non vede quelli esterni.

La creazione delle *variabili* avviene di solito tramite *assegnazione* (=), usando principalmente il “double”, assieme a char (poco) e i vari tipi di interi (int32/64, uint32/64).

Le operazioni da noi usate sono quindi l’*assegnazione* (=), *uguale logico* (==), *addizione* (+), *sottrazione* (-), la *divisione* (/)

Altri ancora sono la (\*) moltiplicazione di due scalari o scalare vettore, che esegue l’operazione degli elementi se effettivamente compatibili, citando anche la moltiplicazione componente a componente (.\*) oppure anche la mancanza di output in una operazione (;)

Altro elemento a cui porre particolare attenzione è *l’operatore due punti* (:) che indica “*da a con passo 1*”, oppure anche *x0:dx : x1* che indica “*da x0 a x1 con passo dx*”.

Importante: Matlab parte a contare da 1!

La cosa importante e’ distinguere vettori (matrici ad una sola linea con le quadre) e matrici (spesso usate ed indicate con [], con elementi separati da spazi o virgole e righe separate da punto e virgola).

La creazione delle matrici segue questa logica:

Immagine che contiene testo

Descrizione generata automaticamente

Esempio di creazione di matrice:

A=[1 2; 4 1; 4 6]

A =

1 2

4 1

4 6

*Il ciclo for non è consigliato, il prof stesso dice che leva punti se lo utilizzi, perché risulta inefficiente su righe e colonne.*

Se le operazioni sono compatibili non genera errori facendo:

Immagine che contiene testo

Descrizione generata automaticamente

con le seguenti funzioni di utilità per vettori/matrici:

Immagine che contiene testo

Descrizione generata automaticamente

Matlab di default utilizza la rappresentazione in doppia precisione a 64 bit nell’intervallo [realmin, realmax] oppure anche tramite la precisione di macchina, quindi il più grande errore relativo compiuto nell’approssimazione di un numero reale.

Similmente numeri speciali sono *pi*, *+Inf, -Inf, NaN*.

Si può cambiare il formato di visualizzazione, che non interferisce con il formato di memorizzazioen dei numeri (es. *help format*), decidendo inoltre se memorizzare i numeri in precisione singola oppure precisione doppia (mettendo *single(variabile)* oppure *double(variabile)*).

Anche altri comandi come *fix, round, floor, ceil* servono ad arrotondare.

Per l’output su video si hanno tre modi:

* visualizzazione nome variabile tasto invio
* funzione *disp* (invocata come *disp(variabile)*
* funzione *fprintf* (caratteri speciali \n che va a capo , \t che indenta il testo che segue, %f che indica formato decimale fisso oppure %e che indica formato esponenziale)

Similmente per immettere un testo a video si usa il comando *input* per dare input a linea di comando e scrivere del testo. La command window è limitante, ragion per cui noi operiamo con l’editor di Matlab. É inoltre possibile commentare i programmi con %.

L’idea è quella di crearsi un file con estensione .m e cominciare a lavorarci sopra, ricordandosi di:

* salvare prima di eseguirlo (tramite tasto run dell’editor o scrivendo il nome dello script nella command window)
* dopo ogni istruzione consigliabile andare a capo, con output video minimo necessario
* dare nomi lunghi ma “esplicativi” allo scopo delle variabili
* mettere “clear all” solitamente pulisce il workspace per la nuova esecuzione, è consigliato (sembra l’uso sia deprecato, meglio mettere “clear (nomefile)”)

Andiamo quindi con il primo esercizio:

Immagine che contiene testo

Descrizione generata automaticamente

Una traccia di soluzione può essere:

clear all

crediti=[6 6 9 9 6];

voti=[22 25 27 22 19];

tot\_crediti=sum(crediti);

media\_pesata=sum(crediti.\*voti)/tot\_crediti;

fprintf('La media pesata è %2.3f\n', media\_pesata);

mediana=median(voti);

fprintf('La mediana è: %2.3f\n', mediana);

maximum=max(voti);

fprintf('Il massimo è: %2.3f\n', maximum);

minimum=min(voti);

fprintf('Il minimo è: %2.3f\n', minimum);

media=mean(voti);

fprintf('La media è: %2.3f\n', media);

Immagine che contiene testo

Descrizione generata automaticamente

Una possibile soluzione è:

clear analisivoti\_input

limite=input('Inserisci il limite dei vettori=');

n=1;

crediti=zeros(1, limite);

voti=zeros(1, limite);

while n<=limite

crediti(n)=input('Inserisci i crediti: ');

voti(n)=input('Inserisci il voto: ');

n=n+1;

end;

disp(crediti);

disp(voti);