

Corso di Laurea in Ingegneria Aerospaziale/ Meccanica

Laboratorio di Analisi Numerica

A.A. 2015/2016 – II Ciclo

Esercitazione 1

Creare una cartella <cognome> in C: dove verranno salvati i file creati nella sessione di lavoro. Appena entrati in MATLAB posizionarsi in <cognome>. Risolvere in ambiente MATLAB i seguenti esercizi.

1. Dopo aver esplorato mediante l'*help* di Matlab, le potenzialità del comando **plot** e in particolare come utilizzare i parametri **color**, **style** e **marker**, scrivere uno script *ex1.m* che disegna in quattro sottofinestre i grafici delle funzioni

$$f_1(x) = 5x^2$$

$$f_2(x) = 5 \sin^2(x) + x \cos^2(x)$$

$$f_3(x) = xe^x$$

$$f_4(x) = \ln(x) + \sin(x)$$

Si utilizzi una discretizzazione dell'intervallo di definizione in 100 punti per opportuni intervalli. Si aggiungano titolo e label x ed y.

2. Scrivere uno script Matlab *ex2.m* che definisce la seguente matrice

$$P = [-6 \ -6 \ -7 \ 0 \ 7 \ 6 \ 6 \ -3 \ -3 \ 0 \ 0 \ -6; \ -7 \ 2 \ 1 \ 8 \ 1 \ 2 \ -7 \ -7 \ -2 \ -2 \ -7 \ -7]$$

di coordinate (x,y) di 12 punti sul piano (prima riga coord. x, seconda riga coord. y).

(a) Utilizzare il comando **plot()** per visualizzare il disegno rappresentato unendo i punti.

(b) Una moltiplicazione matrice per vettore con A matrice diagonale $A1 = [0.1 \ 0; 0 \ 1]$, oppure $A2 = [1 \ 0; 0 \ 0.5]$ ha l'effetto di scalare le coordinate. Mostrare in un'unica finestra il plot delle due trasformazioni applicate ai punti P avvalendosi dei comandi **subplot()**, **figure** per aprire una nuova finestra e **axis([-10,10, -10,10])**

(c) Una rotazione di un angolo α dei punti nel piano è definita dalla matrice

$$G(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}. \text{ Mostrare in un'unica finestra, l'effetto di due rotazioni di P}$$

per $\alpha = 15^\circ$, $\alpha = 45^\circ$, $\alpha = 90^\circ$, $\alpha = 215^\circ$, avvalendosi del comando **subplot()**.

(Attenzione, in MATLAB per angoli misurati in radianti, utilizzare **sin()** e **cos()**, altrimenti **sind()** e **cosd()**)

3. Creare una function $[s1,s2]=\text{somma}(n)$ che calcoli con i metodi 1) e 2) la somma dei primi n numeri naturali:

metodo 1) $s_1 = \sum_{k=1}^n k$

metodo 2) $s_2 = \frac{n(n+1)}{2}$

4. **Distanza dell'orizzonte.** Vi sarà capitato di trovarvi in cima a una collina o una montagna, e vi sarà sembrato di vedere fino all'infinito, ma quanto lontano si poteva vedere in realtà?

La distanza dell'orizzonte cresce a mano a mano si risale una montagna (o una collina) e quindi dipende da quanto la montagna sia alta rispetto al raggio della terra (o di un altro corpo planetario). La function *dist()* calcola la distanza dell'orizzonte mediante l'espressione

$$d = \sqrt{2rh + h^2}$$

dove d=distanza dell'orizzonte; r=raggio della terra; h= altezza della montagna.

Creare un programma MATLAB *distanza.m* che determini la distanza in km dell'orizzonte, sia sulla terra sia sulla luna, per montagne da 0 a 8000 metri (step 200mt). Si consideri:

- raggio della terra = 6378 km

- raggio della luna = 1737 km.

Riportare i risultati in una tabella [altezza_collina, distanza]

Salvare i dati della tabella nel file *dati.dat* in formato ascii. Verificare lanciandolo che lo script produca un corretto risultato.

5. Creare da command window una tabella delle temperature raggiunte da un liquido in un tubo ad ogni ora del giorno.

```
>>r(:,1) = (0:23)';  
>>r(:,2) = 100 + 2.*randn(24,1);  
>>save tab_temp.dat r -ascii
```

Un sensore che controlla la temperatura nel tubo memorizza i dati in tabella *tab_temp.dat* ogni giorno. Nello script *monitor_tubo.m*, leggere il file *tab_temp.dat* (mediante `load tab_temp.dat -ascii`) ed eseguire i seguenti controlli:

- La temperatura non dovrebbe eccedere i 103°F. Utilizzando la funzione `find`, determinare a quali ore del giorno viene superata tale temperatura.
- Determina quante volte la temperatura massima viene superata.
- La temperatura non dovrebbe scendere al di sotto dei 100°F. Utilizzando la funzione `find`, determinare a quali ore del giorno la temperatura è minore della minima.
- Determina quante si scende sotto alla temperatura minima.
- Determina la temperatura minima, massima e media raggiunta.

6. Provare nella command window a digitare *intmax* e *intmin* per ottenere il massimo e minimo intero rappresentabile in MATLAB. Scrivere poi due function che chiamerete *min_float.m*, *max_float.m* che calcolino il più piccolo e il più grande numero floating point della forma $xmin=2^p$ e $xmax=2^r$, rispettivamente. Fornire in uscita i valori $[xmin, p]$ e $[xmax, r]$. (Suggerimento: il risultato di un *overflow* è il “valore” Inf; il risultato di un *underflow* è il valore 0, impostare il formato di visualizzazione long di Matlab (`help format`)). Provare un esempio ($z=x*y$) che abbia come risultato overflow.