

# Corso di Laurea in Ingegneria Aerospaziale/ Meccanica

## Laboratorio di Analisi Numerica

### A.A. 2011/2012 – 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. L'equazione generale per determinare la distanza  $d$  attraversata da un oggetto in caduta libera nel tempo  $t$  trascurando l'attrito dell'aria è

$$d = \frac{1}{2}gt^2$$

L'accelerazione di gravità  $g$  su alcuni pianeti del nostro sistema solare e sulla luna, è data da:

|          |                       |
|----------|-----------------------|
| mercurio | $g=3.7\text{m/s}^2$   |
| venere   | $g=8.87\text{ m/s}^2$ |
| luna     | $g=1.6\text{ m/s}^2$  |
| marte    | $g=3.7\text{ m/s}^2$  |
| nettuno  | $g=11.0\text{ m/s}^2$ |

Nello script **pianeti.m** richiamare la function **[d]=caduta(g,t)** che calcola  $d$  per ogni pianeta nei primi 100 secondi di caduta e stampare una tabella  $[t,d]$ .

2. Scrivere uno script Matlab **test2.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  $\alpha$  dei punti nel piano è definita dalla matrice

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

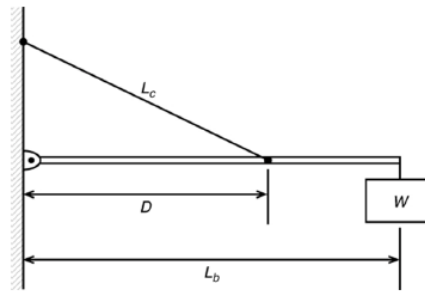
per  $\alpha=15^\circ$ ,  $\alpha=45^\circ$ ,  $\alpha=90^\circ$ ,  $\alpha=215^\circ$ , avvalendosi del comando **subplot()**.

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

3. Un cavo (lungo  $L_c$ ) regge una trave (lunga  $L_b$ ) a cui è attaccato un peso  $W$ .

La tensione del cavo vale:

$$T = \frac{L_b L_c W}{D \sqrt{L_b^2 - D^2}}$$



Supponendo che  $L_b=3\text{m}$ ,  $L_c=5\text{m}$ ,  $W=400\text{N}$ , calcolare la distanza  $D$  che minimizza la tensione del cavo.

Calcolare anche il valore della tensione minima.

**Suggerimento:** La distanza  $D$  assume valori nell'intervallo  $]0,3[$ . Creare dunque un vettore  $D$  discretizzando l'intervallo  $]0,3[$  con un certo passo. Utilizzare poi il comando *min* applicato ad un vettore  $T$  definito come sopra (attenzione ad utilizzare gli operatori elemento per elemento).

4. Creare una function **[ris]=converter(tipo,valore)** che permetta la conversione tra unità di misura diverse, distinte dalla variabile tipo:

tipo=1            converte da gradi Celsius a gradi Fahrenheit ( $T_f=9/5T_c+32$ )  
 tipo=2            converte da metri a miglia ( $\text{miles}=\text{mt}/1000*.6214$ )  
 tipo=3            converte da gradi centigradi a radianti ( $\text{radianti}=\text{gradi}*\pi/180$ )  
 tipo=4            converte da litri a galloni ( $1\text{ lt}=0.264\text{ gallon}$ )

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  $103^\circ\text{F}$ . 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  $100^\circ\text{F}$ . 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  $x_{\min}=2^{-p}$  e  $x_{\max}=2^r$ , rispettivamente. Fornire in uscita i valori  $[x_{\min}, p]$  e  $[x_{\max}, r]$ . (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*.