

**Corso di Laurea Magistrale in
Ingegneria Biomedica
e Ingegneria elettronica e telecomunicazioni per l'energia
Laboratorio di Analisi Numerica
A.A. 2018/2019 – I 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. Creare la seguente matrice ed eseguire i punti richiesti in *ex1.m*:

$$A = \begin{bmatrix} 2 & 6 & -4 & 12 \\ -5 & -9 & 10 & 2 \\ -6 & 12 & 8 & 16 \\ 15 & -3 & 12 & 2 \end{bmatrix}$$

- Creare un vettore v formato dagli elementi della seconda riga di A ;
- Creare una matrice B 4×3 formata dagli elementi dalla seconda alla quarta colonna di A ;
- Creare A trasposta;
- Valutare se la matrice è simmetrica e/o ortogonale
- Calcola la somma di tutti gli elementi di A .

2. CANCELLAZIONE NUMERICA. Nello script *ex2.m* dati i due numeri x, y :

$$x = 5$$

$$y = 5 - a \Rightarrow (x - y) = a$$

calcolare l'errore relativo sulla loro differenza $\varepsilon_{x-y} = \frac{fl(x-y) - (x-y)}{(x-y)}$ e stamparlo in una

tabella insieme all'errore relativo percentuale, al diminuire di a nel range $[1e-1:1e-18]$.

NB: con $fl(x-y)$ si considera il numero finito con cui il computer approssima il risultato di $x-y$.

3. 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 *ex3.m* che disegna in quattro sottofinestre i grafici delle funzioni

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

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

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

$$f_4(x) = \begin{cases} e^{-x^2} & -\pi \leq x \leq 0 \\ \cos(x) & 0 \leq x \leq \pi \end{cases}$$

Si utilizzi una discretizzazione dell'intervallo di definizione in 100 punti per opportuni intervalli.

Si aggiungano titolo e label x ed y .

4. Disegnare nello script **ex4.m** il percorso definito dalla curva di equazione parametrica:

$$F(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} 6 \\ 0 \end{pmatrix} t + \begin{pmatrix} -9 \\ -3 \end{pmatrix} t^2 + \begin{pmatrix} 4 \\ 4 \end{pmatrix} t^3$$

per $t \in [0,1]$. Disegnare una particella (simbolo 'o') che si muove lungo la curva per $t \in [0,1]$ (usare comando `drawnow`). Provare anche il comando **comet(x,y)**.
Calcolare e graficare la velocità della particella ($F'(t)$) al variare di t .

5. Sia definita nella function **funz.m** la funzione:

$$Z = \sin(\sqrt{X^2+Y^2});$$

Nello script **plot_superficie.m** visualizzare i seguenti grafici:

- In figura 1 visualizzare la superficie descritta da **funz.m** in un dominio X,Y in $[-5,5]$ con step 0.5. (Utilizzare `meshgrid()` e `mesh`)
- In figura 2 visualizzare la superficie descritta da **funz.m** in un dominio X,Y in $[-5,5]$. (Utilizzare `meshgrid()` e `surf(Z)` o `surf(X,Y,Z)`)
- Modificare il grafico di figura 2 con `shading interpolato` e `altra colormap` (Utilizzare `shading` e `colormap`) escludere gli assi.
- Generare in figura 3 un `contour plot` di Z
- Generare in figura 4 una combinazione della superficie Z e suo `contour plot`

6. Una lastra metallica quadrata è riscaldata a 90°C in corrispondenza dell'angolo ($x=L, y=L$), dove L definisce la lunghezza del lato del quadrato. La distribuzione della temperatura all'interno del quadrato $[0, L] \times [0, L]$ è fornita dalla seguente funzione bivariata:

$$T(x, y) = 90 \cdot e^{-0.1*(x-L)^2} \cdot e^{-0.05*(y-L)^2}$$

In **ex6.m** creare il `contour plot` e il `plot` di superficie per la temperatura.
Valutare la temperatura nell'angolo ($x=0, y=0$).

7. Scrivere ed eseguire un programma che implementi la seguente ricorrenza:

$$\begin{aligned} x_0 &= 1/2 \\ x_{k+1} &= x_k / 2 \quad \text{se } x_k + 1 > 1 \\ &\text{altrimenti } STOP \end{aligned}$$

Giustificare l'output prodotto dal programma.

8. PROPAGAZIONE DELL'ERRORE DI APPROSSIMAZIONE. Dall'analisi è noto il limite fondamentale $e = \exp(1) = \lim_{n \rightarrow \infty} (1 + \frac{1}{n})^n$. Questo implica che all'aumentare di n il valore di

$f(n) = (1 + \frac{1}{n})^n$ aumenta indefinitamente di accuratezza nell'approssimazione di e .

In **approxexp.m** calcolare per diversi valori di n ($n=0:2:16$) il valore di $f(n)$ e l'errore commesso nell'approssimazione di e . Commentare in % i risultati ottenuti.

9. Nella command window digitare **intmax** e **intmin** per ottenere il massimo e minimo intero rappresentabile in MATLAB. Scrivere poi due function **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.

10. Si scriva uno script *ex10.m* che, utilizzando un ciclo **for**, sommi il valore 0.1 per 100 volte (usare **format long**). Motivare il risultato inserendo un commento (%) nel programma.

11. 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 ($miles=mt/1000*.6214$)
tipo=3	converte da gradi centigradi a radianti ($radianti=gradi*pi/180$)
tipo=4	converte da litri a galloni (1 lt= 0.264 gallon)

12. Consideriamo il problema di realizzare un generico menu a pulsanti che consenta di effettuare diverse scelte. Scrivere uno script *menu1.m* che, a seconda della scelta utente, disegnerà un triangolo, un quadrato, un pentagono o un cerchio. (utilizzare il comando **menu**)