

Nota: Creare sul desktop una cartella in cui mettere tutti i file.
Come nome della cartella usare il proprio cognome.
Al termine della prova cliccare sul tasto start, scegliere il menu "Computer" e aprire il disco (W:Consegna).
Trascinare l'icona della cartella contenente i file dentro la finestra che si è aperta nel momento in cui si è fatto doppio click sul disco W:

Nel primo file creato, scrivere, come commento, anche Nome, Cognome e indirizzo e-mail
È possibile (anzi consigliabile) aggiungere, ove occorra, righe di commento agli m-file

Si richiede di rispondere ad almeno 5 dei seguenti 6 quesiti contrassegnati con le lettere l, m, n, o, p, q

\mathcal{L} È dato il file esterno "datiq.txt" contenente una matrice 30×30
Sia \mathbf{q} questa matrice.
Esaminare tutte le sottomatrici 10×10 di \mathbf{q} aventi righe e colonne consecutive e dire quale di esse ha il numero di condizionamento più basso.

Salvare i comandi relativi in un m-file di tipo script col nome "lira.m".

\mathcal{M} Sempre nella matrice \mathbf{q} , esaminare tutte le sottomatrici 3×3 di \mathbf{q} aventi righe e colonne consecutive ed elencare quelle di esse che sono diagonalmente dominanti per colonne.

Suggerimento:

- Si può usare la funzione **diag** e la funzione **sum** che (leggere le istruzioni) somma le colonne di una matrice.

Salvare i comandi relativi in un m-file di tipo script col nome "mela.m".

\mathcal{N} Costruire un m-file **funzione** col nome **nova(n, t)**. Le variabili di input saranno due numeri \mathbf{n} e \mathbf{t} , dove \mathbf{n} è un numero intero ≥ 3 e \mathbf{t} è un numero qualunque.

La funzione calcolerà innanzitutto il vettore $v = [t \quad t + 0.1 \quad t + 0.2 \quad \dots]$.

Il vettore v avrà lunghezza \mathbf{n} se \mathbf{n} è pari.

Il vettore v avrà lunghezza $\mathbf{n}+1$ se \mathbf{n} è dispari.

La matrice **nova(n, t)** sarà costruita come segue:

$\left(\begin{array}{cccc cc} & v & & & & \\ n & n & n & n & n & n \\ 1 & 0 & \dots & 0 & n & n \\ 0 & 2 & \dots & 0 & n & n \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & N & n & t^2 \end{array} \right)$	<p>La prima riga è \mathbf{v}.</p> <p>Gli elementi della seconda riga sono tutti \mathbf{n}.</p> <p>Gli elementi delle ultime due colonne sono tutti \mathbf{n}, a parte la prima riga e a parte l'ultimo elemento che è t^2.</p> <p>N è ovviamente $\mathbf{n}-2$ se \mathbf{n} è pari e $\mathbf{n}-1$ se \mathbf{n} è dispari.</p>
--	--

Consigli per un file di buona qualità:

1. Nel listato evitare, dove possibile, i comandi del tipo **for...end**.
2. Controllare bene le variabili di input e output.
3. C'è qualche difficoltà a costruire \mathbf{v} perché non sempre termina correttamente: esaminare la possibilità di costruirlo sovrabbondante per poi troncarlo alla lunghezza giusta.
4. Porre (dopo aver verificato che il file funzioni) dei simboli ; alla fine di ogni istruzione.
5. Sarebbe bene che la funzione fosse *a prova di errore*:
Se n non è intero o è inferiore a 3, la funzione potrebbe sostituire 3 a n e dare un avvertimento (comando **warning**)

O Costruire un file funzione di nome **orsa(d)** che, dato **d** numero intero ≥ 3 , calcoli per ogni $t \in [-1, 2]$ (passo 0.1) la norma del vettore x soluzione del sistema lineare $Ax = b$, dove A è la matrice **nova(d, t)** e b è la colonna di numeri 1.
Come output la funzione darà la matrice $[p, m]$, dove p è il punto di minimo della funzione nell'intervallo $[-1, 2]$ e m è il valore del minimo (a meno di 0.1).
Inoltre la funzione disegnerà il grafico monometrico della funzione.

P Per ogni $d = 3, \dots, 30$ calcolare il minimo della funzione definita sopra nel file **orsa** e disegnare il grafico di tale funzione.

Salvare i comandi relativi in un m-file di tipo script col nome "piva.m".

Q Costruire una funzione **s=quiz(t, n)**
Usando l'algoritmo di DeCasteljau, la funzione calcolerà n punti della porzione di cubica avente come poligono di controllo $(0, 0)$ $(1, t)$ $(3, 2)$ $(4, 1)$.
La funzione avrà come variabile di output la tabulazione della curva e cioè una matrice $n \times 2$, e disegnerà la porzione di cubica e il poligono di controllo.

Consiglio:

Considerare la possibilità di usare la funzione **linspace**.