

ESERCIZIO 1

Scrivere la funzione Matlab **find_val** che riceva in input un vettore **v** e una variabile stringa **tipo** che puo' assumere solo i valori 'min' e 'max' e restituisca in output il valore minimo o massimo **M** (in base al valore della variabile tipo) contenuto nel vettore v. Se eventualmente richiesto, si restituisca in output anche l'indice **ind** del vettore in cui si ha il valore minimo o massimo.

(NON usare le funzioni predefinite di Matlab per il calcolo del minimo/massimo)

Applicare la funzione find_val ad un vettore w di 20 elementi generato mediante la funzione rand.

ESERCIZIO 2

Scrivere la funzione Matlab **find_roots** che riceva in input i coefficienti **a,b,c** di un'equazione di secondo grado e restituisca in output le due radici (reali o complesse) **x1** e **x2** . La funzione deve disegnare la parabola di coefficienti a,b,c.

Applicare la funzione find_roots ad un'equazione di secondo grado qualsiasi.

ESERCIZIO 3

Generare lo script Matlab **script_matrix** in cui si crei una matrice quadrata di ordine N con elementi generati a caso da una distribuzione uniforme nell'intervallo [0,1]; in particolare, calcolare:

- un vettore che contiene il valore medio degli elementi di ciascuna colonna;
- un vettore che contiene il valore medio degli elementi di ciascuna riga;
- la somma degli elementi al quadrato della matrice;
- il massimo degli elementi della matrice;
- il rango della matrice;
- il determinante della matrice.

Stampare a video i valori ottenuti.

ESERCIZIO 4

Scrivere la funzione Matlab **fibonacci** che riceva in input un intero positivo **N** e restituisca in output un vettore **v** contenente i primi N elementi della successione di Fibonacci.

La successione di Fibonacci è definita in maniera ricorsiva come:

$$F(i) = F(i-2)+F(i-1), F(1) = 0; F(2) = 1;$$

La funzione deve inoltre disegnare la successione in un grafico bidimensionale usando la funzione **stem()**.

Dalla command window richiamare la funzione per diversi valori di N.

ESERCIZIO 5

Scrivere la funzione Matlab **has_zero** che riceva in input una funzione **f**, e gli estremi di un intervallo **[a,b]** e restituisca una variabile di tipo logico **zero** che assuma il valore 1 se la funzione ha almeno uno zero in [a,b], il valore 0 altrimenti.

Applicare la funzione has_zero alle seguenti funzioni e intervalli

$$y = x^2 \quad [-1,1]$$

$$y = x^3 \quad [-1,1]$$

ESERCIZIO 6

Scrivere una funzione **plot_points** che riceva in input una variabile di tipo struct **coo** contenente le coordinate di due punti del piano cartesiano e restituisca in output la distanza euclidea **d** tra i due punti. La funzione deve disegnare sul piano i due punti.

Si applichi la funzione a coppie random di punti nel piano cartesiano.

ESERCIZIO 7

Data la serie

$$\sum_{j=0}^n x^j \xrightarrow{n \rightarrow \infty} \frac{1}{1-x} \quad \text{se } |x| < 1$$

si scriva una funzione che prenda in input un valore reale **x** e una precisione **TOL** e restituisca in output l'intero **n** tale che la somma n-esima approssimi il valore limite con una precisione pari a TOL.

ESERCIZIO 8

Si scriva una funzione che prenda in input un vettore **v** e un valore di soglia **T** (T numero intero) e restituisca in output due vettori **u** e **l** che contengano i valori sopra e sotto soglia.

Applicare la funzione a diversi vettori

ESERCIZIO 9

Si scriva la funzione **is_prime** che prenda in input un numero intero **n** e restituisca in output una variabile logica che assuma il valore 1 se n è primo, 0 altrimenti.

Si applichi la funzione a diversi numeri interi.