

Laborator 2

Citiți din fișierul pdf *Lab_1.pdf* materialele referitoare la reprezentări grafice, fișiere script și fișiere funcție și instrucțiuni de control în Matlab (de la pagina 13 până la pagina 20).

După ce sunteți familiarizați cu noțiunile descrise și implementare lor în Matlab rezolvați exercițiile de mai jos.

Exercițiul 1. Scrieți codul Matlab (realizând fie o funcție, fie un script, fie scriind codul în linia de comandă) care trasează graficul funcției următoare:

$$f: [-10, 10] \rightarrow \mathbb{R} \quad f(x) = \begin{cases} 2x + 8, & \text{dacă } x \leq 2 \\ 3x^2, & \text{dacă } x > 2 \end{cases}$$

Colorați diferit cele două "ramuri" ale funcției modificând și alte proprietăți ale graficelor (markeri, grosime, etc.).

Exercițiul 2. Scrieți o funcție în Matlab cu numele *genereazaMatrice.m* care primește ca argument un număr natural nenul n și întoarce o matrice A , cu n linii și $n+1$ coloane ale cărei elemente sunt:

$$A(i,j) = \begin{cases} 2, & \text{dacă } i = j; \\ -1, & \text{dacă } |i - j| = 1; \text{ (în Matlab funcția modul este } abs) \\ 0, & \text{în rest} \end{cases}$$

Exercițiul 3

a. Scrieți o funcție în Matlab cu numele *adunaElemente.m* folosind instrucțiuni de control (for, while, break, etc.) care primește ca argument o matrice și returnează suma tuturor elementelor matricei, parcurse linie cu linie, până la întâlnirea celui mai mic element al matricei (acesta nu este luat în calculul sumei).

Exemplu: pentru matricea A de forma

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 0 & 1 & 2 \end{bmatrix} \quad \text{suma returnată va fi } 1+2+\dots+9 = 45.$$

- b. Generați cu ajutorul funcției *rand* o matrice A de dimensiuni $n \times n$ pentru numere aleatoare în (0,1). Calculați timpul de calcul necesar rulării funcției scrise folosind funcțiile *tic* și *toc* pentru valori ale lui n egale cu 10, 100, 1000, 10000.
- c. Puteți scrie o implementare mai bună din punct de vedere al timpului de rulare pentru funcția *adunaElemente.m*? (folosiți proprietățile de calcul vectorial implementate eficient în Matlab).
- d. Plotați timpul de rulare al funcției pentru cele două implementări (cu instrucțiuni de control vs programare vectorială) pentru diferite valori ale lui n .

Exercițiul 4. Implementați înmulțirea a două matrice A și B de dimensiuni $n \times n$ folosind instrucțiunile de control și comparați din punct de vedere al timpului de rulare cu implementarea eficientă din Matlab ($C = A * B$).

Exercițiul 5. (Exercițiul de la seminar) Sunteți concurent într-un joc spectacol în care un premiu este ascuns în spatele uneia dintre cele trei cortine. Veți câștiga premiul dacă alegeți cortina corectă. După ce ați ales cortina, dar înainte ca ea să fie ridicată, prezentatorul ridică una dintre celelalte cortine, arătând o scenă goală și vă întreabă dacă doriți să vă schimbați opțiunea curentă pentru cortina rămasă. Scrieți două funcții în Matlab, *strategie_1* și *strategie_2* în care să implementați următoarele 2 strategii: păstrarea opțiunii curente și schimbarea opțiunii curente. Care strategie este mai bună?