

**Calcul Numeric – Proba practică
Informatică, Anul III**

INSTRUCȚIUNI:

1. Comentați și explicați toate rezolvările trimise. Codurile necomentate/neexplicate nu se punctează.
2. Codurile vor fi salvate cu următoarea denumire [Nume_Prenume_Grupa.py](#) și vor fi trimise titularului de laborator până în data de **29 ianuarie 2021, ora 14:30**.

Factorizarea QR :

Fie $A = (a_{ij})_{i,j=\overline{1,n}} \in \mathcal{M}_n(\mathbb{R})$. Numim descompunere QR a matricei A , descompunerea de forma $A = QR$ unde $Q = (q_{ij})_{i,j=\overline{1,n}} \in \mathcal{M}_n(\mathbb{R})$ este o matrice ortogonală, i.e. $Q^T Q = Q Q^T = I$, iar $R = (r_{ij})_{i,j=\overline{1,n}} \in \mathcal{M}_n(\mathbb{R})$ este o matrice superior triunghiulară.

Dacă $A = (a_{ij})_{i,j=\overline{1,n}} \in \mathcal{M}_n(\mathbb{R})$ este o matrice inversabilă, atunci există și este unică descompunerea QR a matricei A cu $Q \in \mathcal{M}_n(\mathbb{R})$ o matrice ortogonală și $R \in \mathcal{M}_n(\mathbb{R})$ o matrice superior triunghiulară având componentele pe diagonala principală $r_{kk} > 0, k = \overline{1, n}$.

Sistemul $Ax = b$ devine $QRx = b$. Cum Q este ortogonală ($Q^T Q = I$), înmulțind relația $QRx = b$ cu Q^T obținem $Rx = Q^T b$. Cum R este superior triunghiulară sistemul $Rx = Q^T b$ se rezolvă conform metodei substituției descendente.

ALGORITM (Metoda Givens de descompunere QR)

Date de intrare: $A = (a_{ij})_{i,j=\overline{1,n}} \in \mathcal{M}_n(\mathbb{R})$, $b = (b_i)_{i=\overline{1,n}} \in \mathcal{M}_{n,1}(\mathbb{R})$;

Date de ieșire: $Q = (q_{ij})_{i,j=\overline{1,n}} \in \mathcal{M}_n(\mathbb{R})$, $R = (r_{ij})_{i,j=\overline{1,n}} \in \mathcal{M}_n(\mathbb{R})$, $x = (x_i)_{i=\overline{1,n}} \in \mathcal{M}_{n,1}(\mathbb{R})$;

PASUL 1: Inițializează $Q \leftarrow I_n$.

PASUL 2: Pentru $i = \overline{1, n}$ și $j = \overline{i+1, n}$ execută:

- Determină parametrii rotației Givens:

$$\sigma = \sqrt{a_{ii}^2 + a_{ji}^2}, \quad c = \frac{a_{ii}}{\sigma}, \quad s = \frac{a_{ji}}{\sigma}.$$

- Pentru $k = \overline{1, n}$ aplică matricea de rotație Givens matricei A , vectorului b și memorează rotația în matricea Q :

$$u \leftarrow ca_{ik} + sa_{jk}, \quad v \leftarrow -sa_{ik} + ca_{jk}, \quad a_{ik} \leftarrow u, \quad a_{jk} \leftarrow v;$$

$$u \leftarrow cq_{ik} + sq_{jk}, \quad v \leftarrow -sq_{ik} + cq_{jk}, \quad q_{ik} \leftarrow u, \quad q_{jk} \leftarrow v;$$

PASUL 3: Alege $R \leftarrow A$ și $Q \leftarrow Q^T$.

PASUL 4: Determină soluția x a sistemului $Rx = Q^T b$ conform metodei substituției descendente.

Ex. 1 Fie matricea $A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ și $b = (1, 2, 5)^T$.

- Să se implementeze în Python procedura **DescQRGivens**(A) care returnează matricele Q, R și soluția x a sistemului $Ax = b$;
- Să se rezolve numeric sistemul $Ax = b$ în baza procedurii **DescQRGivens**;
- Să se verifice soluția obținută.

Ex. 2

- (a) Creați funcția **newton_raphson** care determină numeric soluția ecuației:

$$f(x) = x^3 + 3x^2 - 25x + 21 = 0, \quad (1)$$

prin metoda Newton-Raphson și are ca **date de intrare**:

- funcția care determină ecuația (1), f ;
- derivata funcției care determină ecuația (1), df ;
- punctul de start al metodei Newton-Raphson, x_0 ;
- toleranța erorii specifice metodei Newton-Raphson, **eps**;

iar ca **date de ieșire**:

- soluția numerică obținută, x_{aprox} ;
- numărul de iterații necesare, N;

- (b) Alegeți subintervalele și punctele de start ale metodei respectând ipotezele teoremei de convergență ale metodei Newton-Raphson, astfel încât șirurile aproximărilor să rămână în subintervalele selectate și să converge la soluții. Justificați atât alegerea subintervalelor, cât și a valorilor inițiale.

Aflați toate soluțiile ecuației (1) apelând funcția **newton_raphson** cu eroarea de aproximare **eps** = 10^{-3} și construiți punctele obținute pe graficul funcției.