# Calcul Numeric – Laboratorul#4
## Calculatoare şi Tehnologia Informaţiei, Anul I

**Algorithm 1:** Metoda substituţiei ascendente

**Input:** $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^n$
**Result:** $\mathbf{x} \in \mathbb{R}^n$

**Pasul 1:** $x_1 = \dfrac{1}{a_{1,1}} \cdot b_1;$
**Pasul 2:** **for** $i \leftarrow 2$ **to** $n$ **do**
$$x_i \longleftarrow \frac{1}{a_{i,i}} \cdot \left[ b_i - \sum_{j=1}^{n-1} a_{i,j} \cdot x_j \right]$$
**end**
**Pasul 3:** OUTPUT($x$)
STOP.

**Algorithm 2:** Factorizarea $LU$ folosind Metoda de eliminare Gauss cu pivotare parţială

**Input:** $\mathbf{A} \in \mathbb{R}^{n\times n}$
**Result:** ($\mathbf{L} \in \mathbb{R}^{n\times n}$, $\mathbf{U} \in \mathbb{R}^{n\times n}$, $\mathbf{w} \in \mathbb{R}^n$) / mesaj de eroare

**Pasul 1:** (Iniţializări)
$L \longleftarrow I_n$
$w \longleftarrow \overline{1,n}$
**Pasul 2:** **for** $i \leftarrow 1$ **to** $n-1$ **do**
**Pasul 3:** Caută cel mai mic întreg $p$ , unde $i \leq p \leq n$ cu proprietatea că $|a_{p,i}| = \max_{j=\overline{i,n}} |a_{j,i}| \neq 0$.
Dacă nu există niciun întreg $p$ cu proprietatea de mai sus:
PRINT('Matricea $A$ nu admite factorizarea $LU$!')
STOP.
**Pasul 4:** **if** $p \neq i$ **then**
$A_p \longleftrightarrow A_i$ (Schimbă linia $p$ cu linia $i$)
$w_p \longleftrightarrow w_i$ (Schimbă ordinea indexului)
**Pasul 5:** **if** $i \neq 1$ **then**
$L_{p,\overline{1:i-1}} \longleftrightarrow L_{i,\overline{1:i-1}}$ (Schimbă sublinii in $L$)
**end**
**end**
**Pasul 6:** **for** $j \leftarrow i+1$ **to** $n$ **do**
**Pasul 7:** $l_{j,i} \longleftarrow \dfrac{1}{a_{i,i}} \cdot a_{j,i}$ (Elementele coloanei $j$ din matricea $L$)
**Pasul 8:** $A_j \longleftarrow A_j - l_{j,i} \cdot A_i$ (Zero sub pivot $\rightarrow MEG\_PP$)
**end**
**end**
**Pasul 9:** **if** $a_{n,n} = 0$ **then**
PRINT('Matricea $A$ nu admite factorizarea $LU$!')
STOP.
**end**
**Pasul 10:** $U \longleftarrow A$
**Pasul 11:** OUTPUT($L, U, w$)
STOP.

**Algorithm 3:** Metoda Givens de descompunere QR

**Input:** $\mathbf{A} \in \mathbb{R}^{n \times n}$
**Result:** $(\mathbf{Q} \in \mathbb{R}^{n \times n}, \mathbf{R} \in \mathbb{R}^{n \times n})$

**Pasul 1:** Initializează $Q \longleftarrow I_n$
**Pasul 2:** **for** $i \leftarrow 1$ **to** $n$ **do**
**Pasul 3:**     **for** $j \leftarrow i+1$ **to** $n$ **do**
        (Determină parametrii de rotaţie)
        $\sigma = \sqrt{a_{i,i}^2 + a_{j,i}^2}$
        $c = \frac{a_{i,i}}{\sigma}$
        $s = \frac{a_{j,i}}{\sigma}$
**Pasul 4:**         **for** $k \leftarrow 1$ **to** $n$ **do**
            (Aplică matricea de rotaţie Givens)
            $u \longleftarrow c \cdot a_{i,k} + s \cdot a_{j,k}$
            $v \longleftarrow -s \cdot a_{i,k} + c \cdot a_{j,k}$
            $a_{i,k} \longleftarrow u$
            $a_{j,k} \longleftarrow v$
            (Memorează rotaţia in matricea $Q$)
            $u \longleftarrow c \cdot q_{i,k} + s \cdot q_{j,k}$
            $v \longleftarrow -s \cdot q_{i,k} + c \cdot q_{j,k}$
            $q_{i,k} \longleftarrow u$
            $q_{j,k} \longleftarrow v$
        **end**
    **end**
**end**
**Pasul 5:** $R \longleftarrow A$
$Q \longleftarrow Q^T$
**Pasul 6:** OUTPUT$(Q, R)$
STOP.

## – Exerciţii –

**Ex.** 1

Implementează in **python** metoda *substituţiei ascendente* cu numele **subs_asc**. Pentru implementare, urmăreşte algoritmul de mai sus.

(a) În implementarea metodei **subs_asc**, verifică dacă:

(i) Matricea $A$ este pătratică;

(ii) Matricea $A$ este inferior triunghiulară;

(iii) Matricea $A$ şi vectorul $\underline{b}$ sunt compatibili;

(iv) Matricea $A$ este inversabilă.

(b) Pentru implementare, verifică rezolvarea sistemului $A \cdot \underline{\boldsymbol{x}} = \underline{b}$, unde:

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 3 & 4 & 0 \\ 1 & 3 & 1 \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} 2 \\ 11 \\ 10 \end{bmatrix}, \quad \underline{\boldsymbol{x}} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \tag{1}$$

**Ex.** 2

Implementează in **python** factorizarea $LU$ cu numele **fact_lu**. Pentru implementare, urmăreşte algoritmul de mai sus.

(a) În implementarea metodei **fact_lu**, verifică dacă:

(i) Matricea $A$ este pătratică;

(ii) Matricea $A$ este inversabilă.

(b) Pentru implementare, verifică rezolvarea sistemului $A \cdot \underline{\boldsymbol{x}} = \underline{b}$, unde:

$$A = \begin{bmatrix} 2 & 3 & 0 \\ 3 & 4 & 2 \\ 1 & 3 & 1 \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} 8 \\ 17 \\ 10 \end{bmatrix}, \quad \underline{\boldsymbol{x}} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \tag{2}$$

*Help:* Ţine cont că rezolvarea sistemului $A \cdot \underline{\boldsymbol{x}} = \underline{b}$ atunci când $A = LU$ folosind algoritmul de mai sus se reduce la a rezolva doua sisteme

$$L \cdot \underline{\boldsymbol{y}} = \underline{b}' \tag{3}$$

$$U \cdot \underline{\boldsymbol{x}} = \underline{y} \tag{4}$$

unde

$$\underline{b}'_j = \underline{b}_{w_j}, \quad j \in \overline{1, n} \tag{5}$$

**Ex.** 3

Implementează in **python** factorizarea $QR$ cu numele **fact_qr**. Pentru implementare, urmăreşte algoritmul de mai sus.

(a) În implementarea metodei **fact_qr**, verifică dacă:

(i) Matricea $A$ este pătratică;

(ii) Matricea $A$ este inversabilă.

(b) Pentru implementare, verifică rezolvarea sistemului $A \cdot \underline{\boldsymbol{x}} = \underline{b}$, unde:

$$A = \begin{bmatrix} 2 & 3 & 0 \\ 3 & 4 & 2 \\ 1 & 3 & 1 \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} 8 \\ 17 \\ 10 \end{bmatrix}, \quad \underline{\boldsymbol{x}} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \tag{6}$$

*Help:* Ţine cont că rezolvarea sistemului $A \cdot \underline{\boldsymbol{x}} = \underline{b}$ atunci când $A = QR$ folosind algoritmul de mai sus se reduce la a rezolva sistemul $R \cdot \underline{\boldsymbol{x}} = Q^T \cdot \underline{b}$.