

# CALCUL NUMERIC

## Curs 1

Paul Irofti  
Cristian Rusu  
Andrei Pătrașcu

Departament Informatică  
Universitatea din București

- **Subspații liniare**
- Matrice. Operații elementare. Proprietăți
- Sisteme de ecuații liniare pătratice
- Algoritmi de rezolvare a SL pătratice
- Ortogonalitate. Sisteme generale



O mulțime  $S$  de vectori din  $\mathbb{R}^n$  este numită **subspațiu liniar** al spațiului  $\mathbb{R}^n$  dacă:

- $x + y \in S, \forall x, y \in S$
- $\alpha x \in S, \forall x \in S, \alpha \in \mathbb{R}$

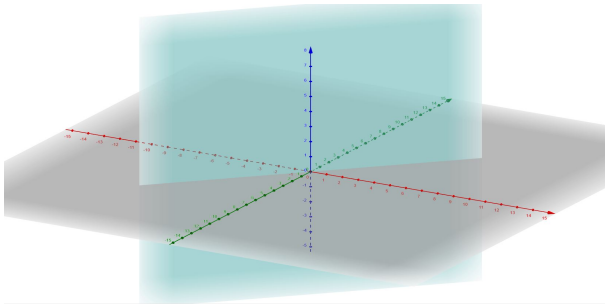


## Subspațiu Liniar

O mulțime  $S$  de vectori din  $\mathbb{R}^n$  este numită **subspațiu liniar** al spațiului  $\mathbb{R}^n$  dacă: (i)  $x + y \in S, \forall x, y \in S$ ; (ii)  $\alpha x \in S, \forall x \in S, \alpha \in \mathbb{R}$

**Exemple:**

- in  $\mathbb{R}$  : axa reală  $S = \{x \in \mathbb{R}\}$



Geogebra.org

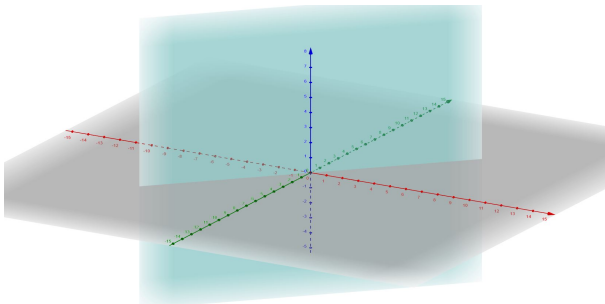


## Subspațiu Liniar

O mulțime  $S$  de vectori din  $\mathbb{R}^n$  este numită **subspațiu liniar** al spațiului  $\mathbb{R}^n$  dacă: (i)  $x + y \in S, \forall x, y \in S$ ; (ii)  $\alpha x \in S, \forall x \in S, \alpha \in \mathbb{R}$

**Exemple:**

- in  $\mathbb{R}$  : axa reală  $S = \{x \in \mathbb{R}\}$
- in  $\mathbb{R}^2$  : dreapta  $S = \{x \in \mathbb{R}^2 : x_1 + 2x_2 = 0\}$



Geogebra.org

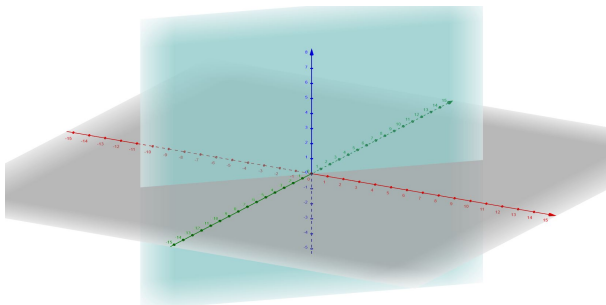


## Subspațiu Liniar

O mulțime  $S$  de vectori din  $\mathbb{R}^n$  este numită **subspațiu liniar** al spațiului  $\mathbb{R}^n$  dacă: (i)  $x + y \in S, \forall x, y \in S$ ; (ii)  $\alpha x \in S, \forall x \in S, \alpha \in \mathbb{R}$

### Example:

- in  $\mathbb{R}$  : axa reală  $S = \{x \in \mathbb{R}\}$
- in  $\mathbb{R}^2$  : dreapta  $S = \{x \in \mathbb{R}^2 : x_1 + 2x_2 = 0\}$
- in  $\mathbb{R}^3$  : plan  $S = \{x \in \mathbb{R}^3 : x_1 + 2x_2 + x_3 = 0, x_1 - x_2 = 0\}$



Geogebra.org

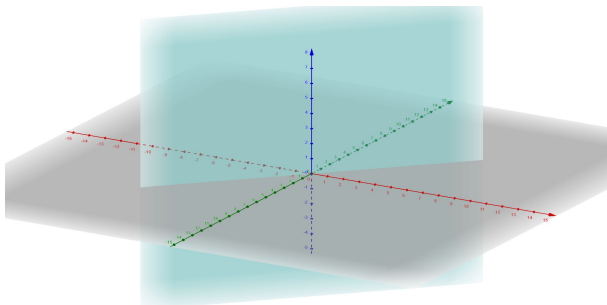


# Subspațiu Liniar

O mulțime  $S$  de vectori din  $\mathbb{R}^n$  este numită **subspațiu liniar** al spațiului  $\mathbb{R}^n$  dacă: (i)  $x + y \in S, \forall x, y \in S$ ; (ii)  $\alpha x \in S, \forall x \in S, \alpha \in \mathbb{R}$

## Example:

- in  $\mathbb{R}$  : axa reală  $S = \{x \in \mathbb{R}\}$
- in  $\mathbb{R}^2$  : dreapta  $S = \{x \in \mathbb{R}^2 : x_1 + 2x_2 = 0\}$
- in  $\mathbb{R}^3$  : plan  $S = \{x \in \mathbb{R}^3 : x_1 + 2x_2 + x_3 = 0, x_1 - x_2 = 0\}$
- in  $\mathbb{R}^n$  : subspațiu  $S = \{x \in \mathbb{R}^n : Ax = 0\}$ , unde  $A \in \mathbb{R}^{m \times n}$



Geogebra.org



O mulțime  $S$  de vectori din  $\mathbb{R}^n$  este numită **subspațiu liniar** al spațiului  $\mathbb{R}^n$  dacă:

- $x + y \in S, \forall x, y \in S$
- $\alpha x \in S, \forall x \in S, \alpha \in \mathbb{R}$

### Exemple:

- in  $\mathbb{R}$  : axa reală  $S = \{x \in \mathbb{R}\}$
- in  $\mathbb{R}^2$  : dreapta  $S = \{x \in \mathbb{R}^2 : x_1 + 2x_2 = 0\}$
- in  $\mathbb{R}^3$  : plan  $S = \{x \in \mathbb{R}^3 : x_1 + 2x_2 + x_3 = 0, x_1 - x_2 = 0\}$
- in  $\mathbb{R}^n$  : subspațiu  $S = \{x \in \mathbb{R}^n : Ax = 0\}$ , unde  $A \in \mathbb{R}^{m \times n}$

Combinatiile liniare ale vectorilor  $X = \{x_1, x_2, \dots, x_p\}$  generează un subspațiu liniar!





O mulțime  $B$  de vectori din  $\mathbb{R}^n$  este numită **bază** al spațiului  $S \subseteq \mathbb{R}^n$  dacă:

- elementele din  $B$  sunt liniar independente
- $B$  generează  $S$

**Exemple:**

- $S = \left\{ x \in \mathbb{R}^n : x = \alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \alpha \in \mathbb{R} \right\}$  sau  $S = \text{span} \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$



O mulțime  $B$  de vectori din  $\mathbb{R}^n$  este numită **bază** al spațiului  $S \subseteq \mathbb{R}^n$  dacă:

- elementele din  $B$  sunt liniar independente
- $B$  generează  $S$

**Exemple:**

- $S = \left\{ x \in \mathbb{R}^n : x = \alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \alpha \in \mathbb{R} \right\}$  sau  $S = \text{span} \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$
- $S = \{ x \in \mathbb{R}^n : x = \alpha_1 b_1 + \cdots + \alpha_p b_p, \alpha \in \mathbb{R}^p \}$  sau  $S = \text{span} \{ b_1, \dots, b_p \}$



O mulțime  $B$  de vectori din  $\mathbb{R}^n$  este numită **bază** al spațiului  $S \subseteq \mathbb{R}^n$  dacă:

- elementele din  $B$  sunt liniar independente
- $B$  generează  $S$

**Exemple:**

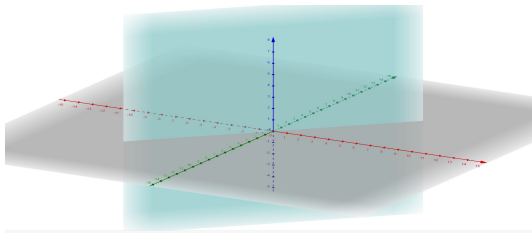
- $S = \left\{ x \in \mathbb{R}^n : x = \alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \alpha \in \mathbb{R} \right\}$  sau  $S = \text{span} \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$
- $S = \{ x \in \mathbb{R}^n : x = \alpha_1 b_1 + \cdots + \alpha_p b_p, \alpha \in \mathbb{R}^p \}$  sau  $S = \text{span} \{ b_1, \dots, b_p \}$
- $\{ e_1, \dots, e_n \}$  baza (canonică) pentru spațiul  $\mathbb{R}^n$



**Dimensiunea** subspațiului  $S \subseteq \mathbb{R}^n$  = numărul de vectori din bază (nr. maxim de vectori liniar independenți)

**Exemple:**

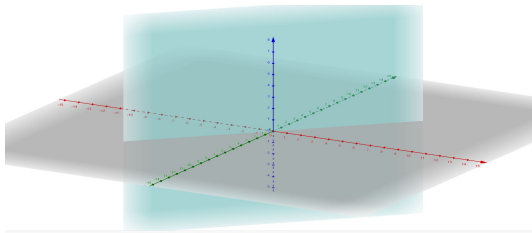
- $S = \left\{ x \in \mathbb{R}^n : x = \alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \alpha \in \mathbb{R} \right\} \Rightarrow \text{Dimensiune } \text{Dim}(S) = 1$



**Dimensiunea** subspațiului  $S \subseteq \mathbb{R}^n$  = numărul de vectori din bază (nr. maxim de vectori liniar independenți)

**Exemple:**

- $S = \left\{ x \in \mathbb{R}^n : x = \alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \alpha \in \mathbb{R} \right\} \Rightarrow \text{Dimensiune } \text{Dim}(S) = 1$
- $S = \{ x \in \mathbb{R}^n : x = \alpha_1 b_1 + \cdots \alpha_p b_p, \alpha \in \mathbb{R}^p \} \Rightarrow \text{Dim}(S) = p$



**În orice (sub)spațiu este necesară o metrică de măsură a distanțelor**

Multe probleme de calcul numeric și învățare automată se formulează în termeni de distanțe:

- SVM, CMMP: Care este cel mai "apropiat" punct ce aparține unei mulțimi?



## **În orice (sub)spațiu este necesară o metrică de măsură a distanțelor**

Multe probleme de calcul numeric și învățare automată se formulează în termeni de distanțe:

- SVM, CMMP: Care este cel mai "apropiat" punct ce aparține unei mulțimi?
- Regresie liniară: Care este funcția liniară care explică cel mai bine un set de date? (minimizează "distanța" de la model la fiecare punct dat)



- Subspații liniare.
- **Matrice. Operații elementare. Proprietăți**
- Sisteme de ecuații liniare pătratice
- Algoritmi de rezolvare a SL pătratice
- Ortogonalitate. Sisteme generale





O matrice  $A \in \mathbb{R}^{m \times n}$  reprezintă un tablou bidimensional de numere reale de forma:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

- dacă  $m = n$  atunci  $A$  este matrice patrata



O matrice  $A \in \mathbb{R}^{m \times n}$  reprezinta un tablou bidimensional de numere reale de forma:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

- dacă  $m = n$  atunci  $A$  este matrice patrata
- $A$  patrata  $\Rightarrow$  diagonala principala este mulțimea pozitiilor pentru care  $i = j$



O matrice  $A \in \mathbb{R}^{m \times n}$  reprezinta un tablou bidimensional de numere reale de forma:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

- dacă  $m = n$  atunci  $A$  este matrice patrata
- $A$  patrata  $\Rightarrow$  diagonala principala este mulțimea pozitiilor pentru care  $i = j$
- $C = A + B \Leftrightarrow c_{ij} = a_{ij} + b_{ij}$



O matrice  $A \in \mathbb{R}^{m \times n}$  reprezinta un tablou bidimensional de numere reale de forma:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

- dacă  $m = n$  atunci  $A$  este matrice patrata
- $A$  patrata  $\Rightarrow$  diagonala principala este mulțimea pozitiilor pentru care  $i = j$
- $C = A + B \Leftrightarrow c_{ij} = a_{ij} + b_{ij}$
- $C = \alpha A \Leftrightarrow c_{ij} = \alpha a_{ij}$



O matrice  $A \in \mathbb{R}^{m \times n}$  reprezinta un tablou bidimensional de numere reale de forma:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

- dacă  $m = n$  atunci  $A$  este matrice patrata
- $A$  patrata  $\Rightarrow$  diagonala principala este mulțimea pozitiilor pentru care  $i = j$
- $C = A + B \Leftrightarrow c_{ij} = a_{ij} + b_{ij}$
- $C = \alpha A \Leftrightarrow c_{ij} = \alpha a_{ij}$
- Transpusa  $B := A^T \Leftrightarrow b_{ij} = a_{ji}$



O matrice  $A \in \mathbb{R}^{m \times n}$  generează subspațiile:

- Imaginea matricii  $A$ :  $Im(A) = \{y \in \mathbb{R}^m : \exists x \in \mathbb{R}^n \text{ astfel incat } y = Ax\}$ 
  - $\text{rang}(A) = \dim(Im(A))$
  - $Im\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) := \left\{x \in \mathbb{R}^2 : x = \alpha \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \alpha \in \mathbb{R}\right\} \rightarrow \text{dimensiune?}$



O matrice  $A \in \mathbb{R}^{m \times n}$  generează subspațiile:

- Imaginea matricii  $A$ :  $Im(A) = \{y \in \mathbb{R}^m : \exists x \in \mathbb{R}^n \text{ astfel incat } y = Ax\}$ 
  - $\text{rang}(A) = \dim(Im(A))$
  - $Im\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) := \left\{x \in \mathbb{R}^2 : x = \alpha \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \alpha \in \mathbb{R}\right\} \rightarrow \text{dimensiune?}$
- Nucleul matricii  $A$ :  $Ker(A) = \{x \in \mathbb{R}^n : Ax = 0\}$ 
  - exemplu:  $Ker\left(\begin{bmatrix} 1 & 2 \end{bmatrix}\right) := \{x \in \mathbb{R}^2 : x_1 + 2x_2 = 0\}$



O matrice  $A \in \mathbb{R}^{m \times n}$  generează subspațiile:

- Imaginea matricii  $A$ :  $Im(A) = \{y \in \mathbb{R}^m : \exists x \in \mathbb{R}^n \text{ astfel incat } y = Ax\}$ 
  - $\text{rang}(A) = \dim(Im(A))$
  - $Im\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) := \left\{x \in \mathbb{R}^2 : x = \alpha \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \alpha \in \mathbb{R}\right\} \rightarrow \text{dimensiune?}$
- Nucleul matricii  $A$ :  $Ker(A) = \{x \in \mathbb{R}^n : Ax = 0\}$ 
  - exemplu:  $Ker\left(\begin{bmatrix} 1 & 2 \end{bmatrix}\right) := \{x \in \mathbb{R}^2 : x_1 + 2x_2 = 0\}$

$$\begin{bmatrix} 0 \\ 5 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix} + \begin{bmatrix} -2 \\ 1 \end{bmatrix}$$





O matrice  $A \in \mathbb{R}^{m \times n}$  generează subspațiile:

- Imaginea matricii  $A$ :  $Im(A) = \{y \in \mathbb{R}^m : \exists x \in \mathbb{R}^n \text{ astfel incat } y = Ax\}$ 
  - $\text{rang}(A) = \dim(Im(A))$
  - $Im\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) := \left\{x \in \mathbb{R}^2 : x = \alpha \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \alpha \in \mathbb{R}\right\} \rightarrow \text{dimensiune?}$
- Nucleul matricii  $A$ :  $Ker(A) = \{x \in \mathbb{R}^n : Ax = 0\}$ 
  - exemplu:  $Ker\left(\begin{bmatrix} 1 & 2 \end{bmatrix}\right) := \{x \in \mathbb{R}^2 : x_1 + 2x_2 = 0\}$

**Teorema.**  $Im(A) \perp Ker(A^T)$  si orice  $x \in \mathbb{R}^m$  se decompune

$$x = u + v, u \in Im(A), v \in Ker(A^T)$$



O matrice  $A \in \mathbb{R}^{m \times n}$  reprezinta un tablou bidimensional de numere reale de forma:

$$A := \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = [a_1 \quad a_2 \quad \cdots \quad a_n]$$

- dacă  $A \in \mathbb{R}^{m \times n}$  și  $x \in \mathbb{R}^n$



O matrice  $A \in \mathbb{R}^{m \times n}$  reprezinta un tablou bidimensional de numere reale de forma:

$$A := \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = [a_1 \quad a_2 \quad \cdots \quad a_n]$$

- dacă  $A \in \mathbb{R}^{m \times n}$  si  $x \in \mathbb{R}^n$
- Produs M-V:  $y = Ax := \sum_{j=1}^n a_j x_j$



O matrice  $A \in \mathbb{R}^{m \times n}$  reprezinta un tablou bidimensional de numere reale de forma:

$$y = Ax := \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix}$$

**Algorithm**  $GAXPY(A, x, y)$

1. **Pentru**  $i = 1 : m$ 
  1. **Pentru**  $j = 1 : n$ 
    1.  $y_i = y_i + a_{ij}x_j$



- Triunghiulare
- Hessenberg
- Diagonale - Bidiagonale - Tridiagonale
- $U$  superior triunghiulara  $u_{ij} = 0$  pentru  $j < i$ :

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

- $L$  inferior triunghiulara  $l_{ij} = 0$  pentru  $j > i$ :

$$\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}$$



- $H$  superior Hessenberg  $h_{ij} = 0$  pentru  $j < i - 1$ :

$$\begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n-1} & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n-1} & h_{2n} \\ 0 & h_{32} & \cdots & h_{3n-1} & h_{3n} \\ \vdots & \vdots & & \vdots & \\ 0 & 0 & \cdots & h_{nn-1} & h_{nn} \end{bmatrix}$$

- $H$  inferior Hessenberg  $h_{ij} = 0$  pentru  $i < j - 1$ :

$$\begin{bmatrix} h_{11} & h_{12} & 0 & \cdots & 0 \\ h_{21} & h_{22} & h_{23} & \cdots & 0 \\ \vdots & \vdots & & \vdots & \\ h_{n-11} & h_{n-12} & h_{n-13} & \cdots & h_{n-1n} \\ h_{n1} & h_{n2} & h_{n3} & \cdots & h_{nn} \end{bmatrix}$$



- $D$  matrice diagonală  $d_{ij} = 0$  pentru  $j \neq i$
- $B$  matrice bidiagonală :

$$\begin{bmatrix} b_{11} & b_{12} & 0 & \cdots & 0 \\ 0 & b_{22} & b_{23} & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & b_{n-1,n-1} & b_{n-1,n} \\ 0 & 0 & \cdots & 0 & b_{nn} \end{bmatrix} \quad \begin{bmatrix} b_{11} & 0 & 0 & \cdots & 0 \\ b_{21} & b_{22} & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & b_{n-1,n-1} & 0 \\ 0 & 0 & \cdots & b_{nn-1} & b_{nn} \end{bmatrix}$$

- $T$  tridiagonală:

$$\begin{bmatrix} t_{11} & t_{12} & 0 & \cdots & 0 \\ t_{21} & t_{22} & t_{23} & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & t_{n-1,n-1} & t_{n-1,n} \\ 0 & 0 & \cdots & t_{nn-1} & t_{nn} \end{bmatrix}$$



- Subspații liniare. Produs scalar. Norme
- Matrice. Operații elementare. Proprietăți
- **Sisteme de ecuații liniare pătratice**
- Algoritmi de rezolvare a SL pătratice
- Ortogonalitate. Sisteme generale





Un sistem de  $m$  ecuații cu  $n$  necunoscute are forma:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n & = & b_2 \\ \dots & & \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n & = & b_m \end{cases},$$

Forma matriceala

$$Ax = b,$$

unde  $A$  este matricea coeficienților,  $x$  vector necunoscutelor,  $b$  termen liber

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

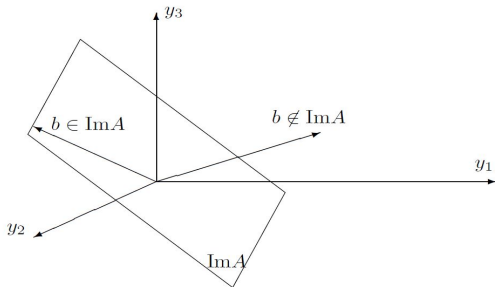


## Sistemul

$$Ax = b$$

- Subdeterminat:  $m < n$ , posibil o infinitate de soluții
- Determinat:  $m = n$ , adesea soluție unică
- Supradeterminat:  $m > n$ , adesea nu are soluție

**Teoremă.** Sistemul  $Ax = b$  are soluție dacă și numai dacă  $b \in \text{Im}(A)$ .



**Exemplu:** Fie  $A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ ,  $b = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ , atunci  $Ax = b$  este

$$\begin{cases} x_1 + x_2 = 0 \\ x_2 + x_3 = 0. \end{cases}$$

- O soluție particulară:  $x = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$
- Mulțimea tuturor soluțiilor:  $X = \left\{ x = \begin{bmatrix} 1 + \alpha \\ 1 - \alpha \\ \alpha \end{bmatrix}, \alpha \in \mathbb{R} \right\}$



Notiuni legate de solvabilitatea unui sistem liniar:

- $A$  singulară = neinvertibilă  $\rightarrow$  exemplu:  $\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$
- $A$  nesingulară = invertibilă  $\rightarrow AA^{-1} = A^{-1}A = I_n$



Notiuni legate de solvabilitatea unui sistem liniar:

- $A$  singulară = neinvertibilă  $\rightarrow$  exemplu:  $\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$
- $A$  nesingulară = invertibilă  $\rightarrow AA^{-1} = A^{-1}A = I_n$
- În alte forme:
  - $\det(A) = 0$
  - coloane liniar dependente
  - Nucleu  $\text{Ker}(A)$  nenul



Notiuni legate de solvabilitatea unui sistem liniar:

- $A$  singulară = neinvertibilă  $\rightarrow$  exemplu:  $\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$
- $A$  nesingulară = invertibilă  $\rightarrow AA^{-1} = A^{-1}A = I_n$
- În alte forme:
  - $\det(A) = 0$
  - coloane liniar dependente
  - Nucleu  $\text{Ker}(A)$  nenul
- Rang = număr maxim de coloane liniar independente



Notiuni legate de solvabilitatea unui sistem liniar:

- $A$  singulară = neinvertibilă  $\rightarrow$  exemplu:  $\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$
- $A$  nesingulară = invertibilă  $\rightarrow AA^{-1} = A^{-1}A = I_n$
- În alte forme:
  - $\det(A) = 0$
  - coloane liniar dependente
  - Nucleu  $\text{Ker}(A)$  nenul
- Rang = număr maxim de coloane liniar independente
- Matricile singulare au rang  $< n$ , i.e. fie o infinitate de soluții, fie fără soluție



**Teoremă.** Dacă  $A \in \mathbb{R}^{n \times n}$  atunci sistemul are soluție unică dacă și numai dacă:

$$x = A^{-1}b.$$

- Nu este o formulă adecvată calculului numeric (vrem sa evităm  $A^{-1}$ )





**Teoremă.** Dacă  $A \in \mathbb{R}^{n \times n}$  atunci sistemul are soluție unică dacă și numai dacă:

$$x = A^{-1}b.$$

- Nu este o formulă adecvată calculului numeric (vrem sa evităm  $A^{-1}$ )
- In general, existența unei soluții se determină greu



**Teoremă.** Dacă  $A \in \mathbb{R}^{n \times n}$  atunci sistemul are soluție unică dacă și numai dacă:

$$x = A^{-1}b.$$

- Nu este o formulă adecvată calculului numeric (vrem sa evităm  $A^{-1}$ )
- In general, existența unei soluții se determină greu
- In particular, existența unei soluții se detectează imediat in cazul matricilor triunghiulare



Matricea  $A$  diagonală:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$



Matricea  $A$  diagonală:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 = 1$$

$$2x_2 = 1$$

$$2x_3 = 1$$



Matricea  $A$  diagonală:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 = 1$$

$$2x_2 = 1$$

$$2x_3 = 1$$

Rezolvare:

$$x_1 = 1$$

$$x_2 = 1/2$$

$$x_3 = 1/2$$



Matricea  $A$  diagonală:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 = 1$$

$$2x_2 = 1$$

$$2x_3 = 1$$

În general, pentru  $A \in \mathbb{R}^{n \times n}$  diagonală observăm că:

- $A_{ii} \neq 0$  este necesar pentru soluție unică



Matricea  $A$  diagonală:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 = 1$$

$$2x_2 = 1$$

$$2x_3 = 1$$

În general, pentru  $A \in \mathbb{R}^{n \times n}$  diagonală observăm că:

- $A_{ii} \neq 0$  este necesar pentru soluție unică
- Complexitate algoritm:



Matricea  $A$  diagonală:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 = 1$$

$$2x_2 = 1$$

$$2x_3 = 1$$

În general, pentru  $A \in \mathbb{R}^{n \times n}$  diagonală observăm că:

- $A_{ii} \neq 0$  este necesar pentru soluție unică
- Complexitate algoritm:
- $\mathcal{O}(n)$





Matricea  $A$  (superior) bidiagonală:

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$



Matricea  $A$  (superior) bidiagonală:

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$\begin{aligned} x_1 + x_2 &= 1 \\ 2x_2 + 3x_3 &= 1 \\ 2x_3 &= 1 \end{aligned}$$



Matricea  $A$  (superior) bidiagonală:

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 + x_2 = 1$$

$$2x_2 + 3x_3 = 1$$

$$2x_3 = 1$$

Pas 1:  $x_3 = \frac{1}{2}$



Matricea  $A$  (superior) bidiagonală:

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 + x_2 = 1$$

$$2x_2 + 3x_3 = 1$$

$$2x_3 = 1$$

Pas 1:  $x_3 = \frac{1}{2}$

Pas 2:  $2x_2 + 3x_3 = 1$



Matricea  $A$  (superior) bidiagonală:

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 + x_2 = 1$$

$$2x_2 + 3x_3 = 1$$

$$2x_3 = 1$$

Pas 1:  $x_3 = \frac{1}{2}$

Pas 2:  $2x_2 + 3x_3 = 1 \rightarrow x_2 = -\frac{1}{4}$



Matricea  $A$  (superior) bidiagonală:

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 + x_2 = 1$$

$$2x_2 + 3x_3 = 1$$

$$2x_3 = 1$$

Pas 1:  $x_3 = \frac{1}{2}$

Pas 2:  $2x_2 + 3x_3 = 1 \rightarrow x_2 = -\frac{1}{4}$

Pas 3:  $x_1 + x_2 = 1$



Matricea  $A$  (superior) bidiagonală:

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 + x_2 = 1$$

$$2x_2 + 3x_3 = 1$$

$$2x_3 = 1$$

Pas 1:  $x_3 = \frac{1}{2}$

Pas 2:  $2x_2 + 3x_3 = 1 \rightarrow x_2 = -\frac{1}{4}$

Pas 3:  $x_1 + x_2 = 1 \rightarrow x_1 = \frac{5}{4}$



Matricea  $A$  (superior) bidiagonală:

$$\begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 & 0 \\ 0 & a_{22} & a_{23} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & 0 & \cdots & 0 & a_{n-1,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$





Matricea  $A$  (superior) bidiagonală:

$$\begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 & 0 \\ 0 & a_{22} & a_{23} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1n-1} & a_{n-1n} \\ 0 & 0 & 0 & \cdots & 0 & a_{n-1n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

Echivalent:

$$a_{11}x_1 + a_{12}x_2 = b_1$$

$$a_{22}x_2 + a_{23}x_3 = b_2$$

$$a_{33}x_3 + a_{34}x_4 = b_3$$

.....

$$a_{n-1n}x_n = b_n$$



Matricea  $A$  (superior) bidiagonală:

$$\begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 & 0 \\ 0 & a_{22} & a_{23} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & 0 & \cdots & 0 & a_{n-1,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

**Algorithm**  $BDTris(A, b)$ :



Matricea  $A$  (superior) bidiagonală:

$$\begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 & 0 \\ 0 & a_{22} & a_{23} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & 0 & \cdots & 0 & a_{n-1,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

**Algorithm**  $BDTris(A, b)$ :

1.  $x := b, x_n = x_n / a_{nn}$



Matricea  $A$  (superior) bidiagonală:

$$\begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 & 0 \\ 0 & a_{22} & a_{23} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & 0 & \cdots & 0 & a_{n-1,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

**Algorithm**  $BDTris(A, b)$ :

1.  $x := b, x_n = x_n / a_{nn}$
2. **Pentru**  $i = n - 1 : -1 : 1$



Matricea  $A$  (superior) bidiagonală:

$$\begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 & 0 \\ 0 & a_{22} & a_{23} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & 0 & \cdots & 0 & a_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

**Algorithm**  $BDTris(A, b)$ :

1.  $x := b, x_n = x_n / a_{nn}$
2. **Pentru**  $i = n - 1 : -1 : 1$ 
  1.  $x_i = x_i - a_{i,i+1} x_{i+1}$



Matricea  $A$  (superior) bidiagonală:

$$\begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 & 0 \\ 0 & a_{22} & a_{23} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & 0 & \cdots & 0 & a_{n-1,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

**Algorithm**  $BDTris(A, b)$ :

1.  $x := b, x_n = x_n / a_{nn}$
2. **Pentru**  $i = n - 1 : -1 : 1$ 
  1.  $x_i = x_i - a_{i,i+1} x_{i+1}$
  2.  $x_i = x_i / a_{ii}$



Matricea  $A$  (superior) triunghiulară:

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$



Matricea  $A$  (superior) triunghiulară:

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 + x_2 + x_3 = 1$$

$$2x_2 + 3x_3 = 1$$

$$2x_3 = 1$$





Matricea  $A$  (superior) triunghiulară:

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 + x_2 + x_3 = 1$$

$$2x_2 + 3x_3 = 1$$

$$2x_3 = 1$$

Pas 1:  $x_3 = \frac{1}{2}$



Matricea  $A$  (superior) triunghiulară:

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 + x_2 + x_3 = 1$$

$$2x_2 + 3x_3 = 1$$

$$2x_3 = 1$$

Pas 1:  $x_3 = \frac{1}{2}$

Pas 2:  $2x_2 + 3x_3 = 1$



Matricea  $A$  (superior) triunghiulară:

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 + x_2 + x_3 = 1$$

$$2x_2 + 3x_3 = 1$$

$$2x_3 = 1$$

Pas 1:  $x_3 = \frac{1}{2}$

Pas 2:  $2x_2 + 3x_3 = 1 \rightarrow x_2 = -\frac{1}{4}$



Matricea  $A$  (superior) triunghiulară:

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 + x_2 + x_3 = 1$$

$$2x_2 + 3x_3 = 1$$

$$2x_3 = 1$$

Pas 1:  $x_3 = \frac{1}{2}$

Pas 2:  $2x_2 + 3x_3 = 1 \rightarrow x_2 = -\frac{1}{4}$

Pas 3:  $x_1 + x_2 + x_3 = 1$



Matricea  $A$  (superior) triunghiulară:

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Echivalent:

$$x_1 + x_2 + x_3 = 1$$

$$2x_2 + 3x_3 = 1$$

$$2x_3 = 1$$

Pas 1:  $x_3 = \frac{1}{2}$

Pas 2:  $2x_2 + 3x_3 = 1 \rightarrow x_2 = -\frac{1}{4}$

Pas 3:  $x_1 + x_2 + x_3 = 1 \rightarrow x_1 = \frac{3}{4}$



Exista instanțe foarte simple de SL pătratice: sistem triunghiulare!  
Distingem:

- $A = U$  superior triunghiulara  $u_{ij} = 0$  pentru  $j < i$ :

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

- $A = L$  inferior triunghiulara  $l_{ij} = 0$  pentru  $j > i$ :

$$\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$



Exista instanțe foarte simple de SL pătratice: sistem triunghiulare!

Distingem:  $A = L$  inferior triunghiulara  $l_{ij} = 0$  pentru  $j > i$ :

$$\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

- Observăm că:  $x_1 = \frac{b_1}{l_{11}}$



Exista instanțe foarte simple de SL pătratică: sistem triunghiular!

Distingem:  $A = L$  inferior triunghiulara  $l_{ij} = 0$  pentru  $j > i$ :

$$\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

- Observăm că:  $x_1 = \frac{b_1}{l_{11}}$
- Dacă se cunosc  $x_1, \dots, x_{i-1}$  atunci :

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij} x_j}{l_{ii}}$$





## Algorithm *LTRIS*( $L, b$ )

1.  $x := b$
2. **Pentru**  $i = 1 : n$ 
  1. **Pentru**  $j = 1 : i - 1$ 
    1.  $x_i = x_i - l_{ij}x_j$
    2.  $x_j = x_j / l_{ij}$

- La fiecare pas al buclei necesită  $2(i - 1)$  flopi



## Algorithm *LTRIS*( $L, b$ )

1.  $x := b$
2. **Pentru**  $i = 1 : n$ 
  1. **Pentru**  $j = 1 : i - 1$ 
    1.  $x_i = x_i - l_{ij}x_j$
    2.  $x_j = x_j / l_{ij}$

- La fiecare pas al buclei necesită  $2(i - 1)$  flopi
- Complexitate:  $\mathcal{O}(n^2)$  (comparativ cazul general  $\mathcal{O}(n^3)$ )



## Algorithm *LTRIS*( $L, b$ )

1.  $x := b$
2. **Pentru**  $i = 1 : n$ 
  1. **Pentru**  $j = 1 : i - 1$ 
    1.  $x_i = x_i - l_{ij}x_j$
    2.  $x_j = x_j / l_{ij}$

- La fiecare pas al buclei necesită  $2(i - 1)$  flopi
- Complexitate:  $\mathcal{O}(n^2)$  (comparativ cazul general  $\mathcal{O}(n^3)$ )
- Exercițiu: Scrieti pseudocodul alg. UTRIS



## Algorithm *LTRIS*( $L, b$ )

1.  $x := b$
2. **Pentru**  $i = 1 : n$ 
  1. **Pentru**  $j = 1 : i - 1$ 
    1.  $x_i = x_i - l_{ij}x_j$
    2.  $x_j = x_j / l_{ij}$

- La fiecare pas al buclei necesită  $2(i - 1)$  flopi
- Complexitate:  $\mathcal{O}(n^2)$  (comparativ cazul general  $\mathcal{O}(n^3)$ )
- Exercițiu: Scrieti pseudocodul alg. UTRIS
- **Idee: Putem reduce un sistem general la unul simplu (e.g. triunghiular)?**



- Subspații liniare. Produs scalar. Norme
- Matrice. Operații elementare. Proprietăți
- Sisteme de ecuații liniare pătratice
- **Algoritmi de rezolvare a SL pătratice**
- Ortogonalitate. Sisteme generale



**Scop:** echivalarea sistemului generic

$$Ax = b$$

printr-un număr finit de transformări, cu un sistem triunghiular

$$Ux = \tilde{b}$$



**Scop:** echivalarea sistemului generic

$$Ax = b$$

printr-un număr finit de transformări, cu un sistem triunghiular

$$Ux = \tilde{b}$$

Pornind de la sistemul inițial:  $\begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ 10 \end{bmatrix}$  triangularizăm  
succesiv coloanele lui  $A$  pentru a obține forma:

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \tilde{b}_3 \end{bmatrix}$$



**Scop:** echivalarea sistemului generic

$$Ax = b$$

printr-un număr finit de transformări, cu un sistem triunghiular

$$Ux = \tilde{b}$$

Pornind de la sistemul inițial:  $\begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ 10 \end{bmatrix}$  triangularizăm

succesiv coloanele lui  $A$  pentru a obține forma:

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \tilde{b}_2 \end{bmatrix}$$

**Problema se reduce la găirea unei transformări elementare  $M$  care produce 0-uri subdiagonale în coloana  $a_i$ , i.e.**

$$\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} \rightarrow Ma_1 := \begin{bmatrix} \tilde{a}_{11} \\ 0 \\ 0 \end{bmatrix}$$





- $a_1 = \begin{bmatrix} 2 \\ 4 \\ -2 \end{bmatrix}$



- $a_1 = \begin{bmatrix} 2 \\ 4 \\ -2 \end{bmatrix}$
- $M_{11} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 0 \\ 4 \\ -2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$



$$\bullet a_1 = \begin{bmatrix} 2 \\ 4 \\ -2 \end{bmatrix}$$

$$\bullet M_{11} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 0 \\ 4 \\ -2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$M_{11}a_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ -2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$



- $a_1 = \begin{bmatrix} 2 \\ 4 \\ -2 \end{bmatrix}$



- $a_1 = \begin{bmatrix} 2 \\ 4 \\ -2 \end{bmatrix}$
- $M_{21} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{4} \begin{bmatrix} 0 \\ 0 \\ -2 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$



$$\bullet a_1 = \begin{bmatrix} 2 \\ 4 \\ -2 \end{bmatrix}$$

$$\bullet M_{21} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{4} \begin{bmatrix} 0 \\ 0 \\ -2 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

$$M_{21} a_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ -2 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 0 \end{bmatrix}$$



**Definiție.** Transformare inferior triunghiulară elementară (ITE) de ordin  $n$  și indice  $k$  are forma:

$$M_k = I_n - m_k e_k^T$$

unde

$$m_k = [0 \quad 0 \quad \cdots \quad 0 \quad \mu_{k+1,k} \quad \cdots \quad \mu_{n,k}]$$

are primele  $k$  elemente nenule.

$$M_k = \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 \\ & & \ddots & & & \\ 0 & 0 & \cdots & 1 & \cdots & 0 \\ 0 & 0 & \cdots & -\mu_{k+1,k} & \cdots & 0 \\ & & \cdots & \cdots & \ddots & 0 \\ 0 & 0 & \cdots & -\mu_{nk} & \cdots & 1 \end{bmatrix}.$$



Proprietăți ale matricilor de transformare ITE:

- $M_k$  este inversabilă și  $M_k^{-1} = I_n + m_k e_k^T$





Proprietăți ale matricilor de transformare ITE:

- $M_k$  este inversabilă și  $M_k^{-1} = I_n + m_k e_k^T$
- $(M_k x)_i := \begin{cases} x_i & \text{pentru } i = 1 : k \\ x_i - \mu_{ik} x_k & \text{pentru } i = k + 1 : n \end{cases}$



Proprietăți ale matricilor de transformare ITE:

- $M_k$  este inversabilă și  $M_k^{-1} = I_n + m_k e_k^T$
- $(M_k x)_i := \begin{cases} x_i & \text{pentru } i = 1 : k \\ x_i - \mu_{ik} x_k & \text{pentru } i = k + 1 : n \end{cases}$
- Pe scurt,  $M_k$  schimbă elementele din  $x$  de la indicele  $k$  mai departe



## Proprietăți ale matricilor de transformare ITE:

- $M_k$  este inversabilă și  $M_k^{-1} = I_n + m_k e_k^T$
- $(M_k x)_i := \begin{cases} x_i & \text{pentru } i = 1 : k \\ x_i - \mu_{ik} x_k & \text{pentru } i = k + 1 : n \end{cases}$
- Pe scurt,  $M_k$  schimbă elementele din  $x$  de la indicele  $k$  mai departe
- Dacă alegem valori potrivite pentru multiplicatorii  $\mu_{ik}$ , putem obține  $(M_k x)_i := \begin{cases} x_i & \text{pentru } i = 1 : k \\ 0 & \text{pentru } i = k + 1 : n \end{cases}$



### Proprietăți ale matricilor de transformare ITE:

- $M_k$  este inversabilă și  $M_k^{-1} = I_n + m_k e_k^T$
- $(M_k x)_i := \begin{cases} x_i & \text{pentru } i = 1 : k \\ x_i - \mu_{ik} x_k & \text{pentru } i = k + 1 : n \end{cases}$
- Pe scurt,  $M_k$  schimbă elementele din  $x$  de la indicele  $k$  mai departe
- Dacă alegem valori potrivite pentru multiplicatorii  $\mu_{ik}$ , putem obține

$$(M_k x)_i := \begin{cases} x_i & \text{pentru } i = 1 : k \\ 0 & \text{pentru } i = k + 1 : n \end{cases}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \quad M_2 x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ 0 \end{bmatrix} \quad \text{alegând } \mu_{i1} = \frac{x_i}{x_2}$$



### Proprietăți ale matricilor de transformare ITE:

- $M_k$  este inversabilă și  $M_k^{-1} = I_n + m_k e_k^T$
- $(M_k x)_i := \begin{cases} x_i & \text{pentru } i = 1 : k \\ x_i - \mu_{ik} x_k & \text{pentru } i = k + 1 : n \end{cases}$
- Pe scurt,  $M_k$  schimbă elementele din  $x$  de la indicele  $k$  mai departe
- Dacă alegem valori potrivite pentru multiplicatorii  $\mu_{ik}$ , putem obține  $(M_k x)_i := \begin{cases} x_i & \text{pentru } i = 1 : k \\ 0 & \text{pentru } i = k + 1 : n \end{cases}$
- Dacă  $x_k = 0$ , atunci  $M_k x = x$

$$x = \begin{bmatrix} x_1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad M_2 x = \begin{bmatrix} x_1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{alegând } \mu_{i1} = \frac{x_i}{x_2}$$



Algoritmul de EG propune triangularizarea progresiva a matricii  $A$

**Initializare:**  $A_1 = A, b_{(1)} = b$

**Pas 1:**  $A_2 = M_1 A$  are elementele sub-diagonale de pe coloana 1 egale cu 0

$$A_2 = \begin{bmatrix} a_{11}^{(2)} & a_{12}^{(2)} & \dots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} \\ 0 & a_{32}^{(2)} & \dots & a_{3n}^{(2)} \\ \dots & \dots & \dots & \dots \\ 0 & a_{n2}^{(2)} & \dots & a_{nn}^{(2)} \end{bmatrix}.$$

**Pas 2:**  $A_3 = M_2 M_1 A$  are elementele sub-diagonale de pe coloanele 1 si 2 egale cu 0

.....

**Pas k:**  $A_k = M_k \dots M_2 M_1 A$  are elementele sub-diagonale de pe coloanele 1 : k egale cu 0; nu sunt afectate primele  $k - 1$  coloane



## Algoritm $EG(A)$

1. **Pentru**  $k = 1 : n - 1$

1. **Se calculează matricea**  $M_k$  (adică multiplicatorii  $\mu_{ik}, i = k + 1 : n$ ), astfel încât  $(M_k A)_i = 0$ , pentru  $i = k + 1 : n$

2.  $A = M_k A$

- Multiplicatorii necesari  $\mu_{ik} = \frac{a_{ik}^k}{a_{kk}^k}$



## Algoritm $EG(A)$

1. **Pentru**  $k = 1 : n - 1$

1. **Se calculează matricea**  $M_k$  (adică multiplicatorii  $\mu_{ik}, i = k + 1 : n$ ), astfel încât  $(M_k A)_i = 0$ , pentru  $i = k + 1 : n$

2.  $A = M_k A$

- Multiplicatorii necesari  $\mu_{ik} = \frac{a_{ik}^k}{a_{kk}^k}$
- Algoritmul produce  $U = A_n = \underbrace{M_{n-1} M_{n-2} \cdots M_1}_M A$





## Algoritm $EG(A)$

1. **Pentru**  $k = 1 : n - 1$

1. **Se calculează matricea**  $M_k$  (adică multiplicatorii  $\mu_{ik}, i = k + 1 : n$ ), astfel încât  $(M_k A)_i = 0$ , pentru  $i = k + 1 : n$

2.  $A = M_k A$

- Multiplicatorii necesari  $\mu_{ik} = \frac{a_{ik}^k}{a_{kk}^k}$
- Algoritmul produce  $U = A_n = \underbrace{M_{n-1} M_{n-2} \cdots M_1}_M A$
- Matricea  $M$  este inferior triunghiulară (de ce?)



## Algoritm $EG(A)$

1. **Pentru**  $k = 1 : n - 1$

1. **Se calculează matricea**  $M_k$  (adică multiplicatorii  $\mu_{ik}, i = k + 1 : n$ ), astfel încât  $(M_k A)_i = 0$ , pentru  $i = k + 1 : n$

2.  $A = M_k A$

- Multiplicatorii necesari  $\mu_{ik} = \frac{a_{ik}^k}{a_{kk}^k}$
- Algoritmul produce  $U = A_n = \underbrace{M_{n-1} M_{n-2} \cdots M_1}_M A$
- Matricea  $M$  este inferior triunghiulară (de ce?)
- **Teoremă.** Dacă toate matricile lider principale  $A^{[k]}$  din  $A$  sunt nesingulare, atunci algoritmul produce un  $U$  nesingular (sistemul are soluție unică!)



## Algoritm $EG(A)$

1. **Pentru**  $k = 1 : n - 1$ 
    1. **Se calculează matricea**  $M_k$  (adică multiplicatorii  $\mu_{ik}, i = k + 1 : n$ ), astfel încât  $(M_k A)_i = 0$ , pentru  $i = k + 1 : n$
    2.  $A = M_k A$
- Multiplicatorii necesari  $\mu_{ik} = \frac{a_{ik}^k}{a_{kk}^k}$
  - Algoritmul produce  $U = A_n = \underbrace{M_{n-1} M_{n-2} \cdots M_1}_M A$
  - Matricea  $M$  este inferior triunghiulară (de ce?)
  - **Teoremă.** Dacă toate matricile lider principale  $A^{[k]}$  din  $A$  sunt nesingulare, atunci algoritmul produce un  $U$  nesingular (sistemul are soluție unică!)
  - Altfel EG produce un  $U$  singular (care sunt consecințele?)



- $A = \begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix}$



$$\bullet A = \begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix}$$

$$\bullet M_1 A = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix} = \begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 1 & 5 \end{bmatrix}$$



$$\bullet A = \begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix}$$

$$\bullet M_1 A = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix} = \begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 1 & 5 \end{bmatrix}$$

$$\bullet M_2 M_1 A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 1 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 0 & 4 \end{bmatrix}$$



## Triangularizare - exemplu

$$\bullet A = \begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix}$$

$$\bullet M_1 A = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix} = \begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 1 & 5 \end{bmatrix}$$

$$\bullet M_2 M_1 A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 1 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 0 & 4 \end{bmatrix}$$

$$\bullet \text{În final rezolvăm, } \begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = M_2 M_1 b = \begin{bmatrix} 2 \\ 4 \\ 8 \end{bmatrix}$$



- Fie  $L_1, L_2 \in \mathbb{R}^{n \times n}$  doua matrici inferior triunghiulare, atunci:

$$R = L_1 L_2,$$

satisface  $R_{ij} = (L_1)^i(L_2)_j = \begin{bmatrix} (L_1)_{i1} & (L_1)_{i2} & \cdots & (L_1)_{ii} & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ (L_2)_{ij} \\ (L_2)_{i+1j} \\ \vdots \\ (L_2)_{nj} \end{bmatrix}$

- Dacă  $i < j$ , atunci nu au loc suprapuneri de elemente!**
- Consecință:**  $R$  este inferior triunghiulară





- Acumulare multiplicatori:  $M = M_2 M_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix}$
- In general:  $M = M_{n-1} \cdots M_2 M_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \mu_{21} & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \mu_{n1} & \mu_{n2} & \cdots & 1 \end{bmatrix}$
- $\mu_{ij}$  ocupă triunghiul strict inferior, A are progresiv aceste poziții libere. Idee?



Pseudocodul algoritmului EG:

**Algoritm  $G(A)$**

1. **Pentru**  $k = 1 : n - 1$ 
  1. **Pentru**  $i = k + 1 : n$ 
    1.  $a_{ik} \leftarrow \mu_{ik} = \frac{a_{ik}}{a_{kk}}$
  2. **Pentru**  $j = k + 1 : n$ 
    1. **Pentru**  $i = k + 1 : n$ 
      1.  $a_{ij} \leftarrow a_{ij} - \mu_{ik} a_{kj}$

Multiplicatorii  $\mu_{ik}$  se pot memora în triunghiul inferior al matricii  $A$

$$\begin{bmatrix} u_{11} & u_{12} & \dots & u_{1k} & u_{1,k+1} & \dots & u_{1n} \\ \mu_{21} & u_{22} & \dots & u_{2k} & u_{2,k+1} & \dots & u_{2n} \\ & & \dots & & & \dots & \\ \mu_{k1} & \mu_{k2} & \dots & u_{kk} & u_{k,k+1} & \dots & u_{kn} \\ \mu_{k+1,1} & \mu_{k+1,2} & \dots & \mu_{k+1,k} & a_{k+1,k+1}^{(k+1)} & \dots & a_{k+1,n}^{(k+1)} \\ & & \dots & & & \dots & \\ \mu_{n1} & \mu_{n2} & \dots & \mu_{nk} & a_{n,k+1}^{(k+1)} & \dots & a_{nn}^{(k+1)} \end{bmatrix}$$

După pasul  $k$

$$\begin{bmatrix} u_{11} & u_{12} & \dots & u_{1k} & \dots & u_{1n} \\ \mu_{21} & u_{22} & \dots & u_{2k} & \dots & u_{2n} \\ & & \dots & & \dots & \\ \mu_{k1} & \mu_{k2} & \dots & u_{kk} & \dots & u_{kn} \\ & & \dots & & \dots & \\ & & \dots & & \dots & \\ \mu_{n1} & \mu_{n2} & \dots & \mu_{nk} & \dots & u_{nn} \end{bmatrix}$$

În final



Pseudocodul algoritmului EG:

**Algoritm**  $G(A)$

1. **Pentru**  $k = 1 : n - 1$

1. **Pentru**  $i = k + 1 : n$

1.  $a_{ik} \leftarrow \mu_{ik} = \frac{a_{ik}}{a_{kk}}$

2. **Pentru**  $j = k + 1 : n$

1. **Pentru**  $i = k + 1 : n$

1.  $a_{ij} \leftarrow a_{ij} - \mu_{ik} a_{kj}$

- În final scopul este atins:  $MA = U$ , deci sub condițiile teoremei  $Ax = b$  are aceleași soluții cu  $Ux = Mb$ ; se rezolvă cu UTRIS



Pseudocodul algoritmului EG:

**Algoritm**  $G(A)$

1. **Pentru**  $k = 1 : n - 1$ 
  1. **Pentru**  $i = k + 1 : n$ 
    1.  $a_{ik} \leftarrow \mu_{ik} = \frac{a_{ik}}{a_{kk}}$
  2. **Pentru**  $j = k + 1 : n$ 
    1. **Pentru**  $i = k + 1 : n$ 
      1.  $a_{ij} \leftarrow a_{ij} - \mu_{ik} a_{kj}$

- În final scopul este atins:  $MA = U$ , deci sub condițiile teoremei  $Ax = b$  are aceleași soluții cu  $Ux = Mb$ ; se rezolvă cu UTRIS
- $MA = U \Leftrightarrow A = M^{-1}U = LU$



Pseudocodul algoritmului EG:

## Algoritm $G(A)$

1. **Pentru**  $k = 1 : n - 1$ 
  1. **Pentru**  $i = k + 1 : n$ 
    1.  $a_{ik} \leftarrow \mu_{ik} = \frac{a_{ik}}{a_{kk}}$
  2. **Pentru**  $j = k + 1 : n$ 
    1. **Pentru**  $i = k + 1 : n$ 
      1.  $a_{ij} \leftarrow a_{ij} - \mu_{ik} a_{kj}$

- În final scopul este atins:  $MA = U$ , deci sub condițiile teoremei  $Ax = b$  are aceleași soluții cu  $Ux = Mb$ ; se rezolvă cu UTRIS
- $MA = U \Leftrightarrow A = M^{-1}U = LU$
- Complexitate totală:  $\sum_{k=1}^{n-1} (n - k + 2(n - k)^2) \approx \frac{2n^3}{3} = \mathcal{O}(n^3)$



Pseudocodul algoritmului EG:

**Algoritm  $G(A)$**

1. **Pentru**  $k = 1 : n - 1$ 
  1. **Pentru**  $i = k + 1 : n$ 
    1.  $a_{ik} \leftarrow \mu_{ik} = \frac{a_{ik}}{a_{kk}}$
  2. **Pentru**  $j = k + 1 : n$ 
    1. **Pentru**  $i = k + 1 : n$ 
      1.  $a_{ij} \leftarrow a_{ij} - \mu_{ik} a_{kj}$

- În final scopul este atins:  $MA = U$ , deci sub condițiile teoremei  $Ax = b$  are aceleași soluții cu  $Ux = Mb$ ; se rezolvă cu UTRIS
- $MA = U \Leftrightarrow A = M^{-1}U = LU$
- Complexitate totală:  $\sum_{k=1}^{n-1} (n - k + 2(n - k)^2) \approx \frac{2n^3}{3} = \mathcal{O}(n^3)$
- **Probleme: Ce se întâmplă dacă o submatrice lider principală este singulară?**



Pseudocodul algoritmului EG:

**Algoritm**  $G(A)$

1. **Pentru**  $k = 1 : n - 1$ 
  1. **Pentru**  $i = k + 1 : n$ 
    1.  $a_{ik} \leftarrow \mu_{ik} = \frac{a_{ik}}{a_{kk}}$
  2. **Pentru**  $j = k + 1 : n$ 
    1. **Pentru**  $i = k + 1 : n$ 
      1.  $a_{ij} \leftarrow a_{ij} - \mu_{ik} a_{kj}$

- În final scopul este atins:  $MA = U$ , deci sub condițiile teoremei  $Ax = b$  are aceleași soluții cu  $Ux = Mb$ ; se rezolvă cu UTRIS
- $MA = U \Leftrightarrow A = M^{-1}U = LU$
- Complexitate totală:  $\sum_{k=1}^{n-1} (n - k + 2(n - k)^2) \approx \frac{2n^3}{3} = \mathcal{O}(n^3)$
- **Probleme: Ce se întâmplă dacă o submatrice lider principală este singulară?**
- Răspuns: La pasul  $k$  pivotul  $a_{kk}^{(k)}$  este nul; cum se pot calcula, în situația aceasta, multiplicatorii  $\mu_{ik}$  ?



Adaptați algoritmul  $G$  pentru cazul în care  $A := H$  are forma superior Hessenberg. Ce formă va avea matricea multiplicatorilor  $M$ ?

$$\begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n-1} & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n-1} & h_{2n} \\ 0 & h_{32} & \cdots & h_{3n-1} & h_{3n} \\ \vdots & \vdots & & \vdots & \\ 0 & 0 & \cdots & h_{nn-1} & h_{nn} \end{bmatrix}$$

- Coloana  $h_i$  are un singur element subdiagonal





Adaptați algoritmul  $G$  pentru cazul în care  $A := H$  are forma superior Hessenberg. Ce formă va avea matricea multiplicatorilor  $M$ ?

$$\begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n-1} & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n-1} & h_{2n} \\ 0 & h_{32} & \cdots & h_{3n-1} & h_{3n} \\ \vdots & \vdots & & \vdots & \\ 0 & 0 & \cdots & h_{nn-1} & h_{nn} \end{bmatrix}$$

- Coloana  $h_i$  are un singur element subdiagonal
- Este necesar un singur multiplicator  $\mu_{k+1k} = \frac{h_{k+1k}}{h_{kk}}$  la fiecare pas



$$H = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 5 & 3 & 2 & 2 \\ 0 & 3 & 4 & 3 \\ 0 & 0 & 4 & 5 \end{bmatrix}$$



$$H = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 5 & 3 & 2 & 2 \\ 0 & 3 & 4 & 3 \\ 0 & 0 & 4 & 5 \end{bmatrix}$$

$$\text{Pas 1 : } M_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -5/2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$H = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 5 & 3 & 2 & 2 \\ 0 & 3 & 4 & 3 \\ 0 & 0 & 4 & 5 \end{bmatrix}$$

$$\text{Pas 1 : } M_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -5/2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_1 H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -5/2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 1 \\ 5 & 3 & 2 & 2 \\ 0 & 3 & 4 & 3 \\ 0 & 0 & 4 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 0 & 1/2 & -1/2 & -1/2 \\ 0 & 3 & 4 & 3 \\ 0 & 0 & 4 & 5 \end{bmatrix}$$



$$H = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 5 & 3 & 2 & 2 \\ 0 & 3 & 4 & 3 \\ 0 & 0 & 4 & 5 \end{bmatrix}$$

$$M_1 A = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 0 & 1/2 & -1/2 & -1/2 \\ 0 & 3 & 4 & 3 \\ 0 & 0 & 4 & 5 \end{bmatrix}$$



$$H = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 5 & 3 & 2 & 2 \\ 0 & 3 & 4 & 3 \\ 0 & 0 & 4 & 5 \end{bmatrix}$$

$$M_1 A = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 0 & 1/2 & -1/2 & -1/2 \\ 0 & 3 & 4 & 3 \\ 0 & 0 & 4 & 5 \end{bmatrix}$$

$$\text{Pas 2 : } M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -6 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow M_2 M_1 A = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 0 & 1/2 & -1/2 & -1/2 \\ 0 & 0 & 7 & 6 \\ 0 & 0 & 4 & 5 \end{bmatrix}$$



$$H = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 5 & 3 & 2 & 2 \\ 0 & 3 & 4 & 3 \\ 0 & 0 & 4 & 5 \end{bmatrix}$$

$$M_2 M_1 A = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 0 & 1/2 & -1/2 & -1/2 \\ 0 & 0 & 7 & 6 \\ 0 & 0 & 4 & 5 \end{bmatrix}$$



$$H = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 5 & 3 & 2 & 2 \\ 0 & 3 & 4 & 3 \\ 0 & 0 & 4 & 5 \end{bmatrix}$$

$$M_2 M_1 A = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 0 & 1/2 & -1/2 & -1/2 \\ 0 & 0 & 7 & 6 \\ 0 & 0 & 4 & 5 \end{bmatrix}$$

$$\text{Pas 3 : } M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -4/7 & 1 \end{bmatrix} \Rightarrow M_3 M_2 M_1 A = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 0 & 1/2 & -1/2 & -1/2 \\ 0 & 0 & 7 & 6 \\ 0 & 0 & 0 & 11/7 \end{bmatrix}$$





$$H = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 5 & 3 & 2 & 2 \\ 0 & 3 & 4 & 3 \\ 0 & 0 & 4 & 5 \end{bmatrix}$$

Observăm că:

- $M$  este inferior bidiagonală:

$$M := M_1 M_2 M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -5/2 & 1 & 0 & 0 \\ 0 & -6 & 1 & 0 \\ 0 & 0 & -4/7 & 1 \end{bmatrix}$$

- Doar 3 pași sunt necesari
- Cum generalizăm algoritmul  $G$ ?



Algoritmului EG pentru  $A = H$  superior Hessenberg:

**Algoritm**  $G(A)$  % Varianta generală



Algoritmului EG pentru  $A = H$  superior Hessenberg:

**Algoritm**  $G(A)$  % Varianta Hessenberg

**Algoritm**  $G(A)$  % Varianta generală

1. **Pentru**  $k = 1 : n - 1$



Algoritmului EG pentru  $A = H$  superior Hessenberg:

**Algoritm**  $G(A)$  % Varianta Hessenberg

1. **Pentru**  $k = 1 : n - 1$

**Algoritm**  $G(A)$  % Varianta generală

1. **Pentru**  $k = 1 : n - 1$

1. **Pentru**  $i = k + 1 : n$

1.  $a_{ik} \leftarrow \mu_{ik} = \frac{a_{ik}}{a_{kk}}$



Algoritmului EG pentru  $A = H$  superior Hessenberg:

**Algoritm**  $G(A)$  % Varianta Hessenberg

1. **Pentru**  $k = 1 : n - 1$

1. **Pentru**  $i = k + 1 : n$

1.  $a_{k+1k} \leftarrow \mu_{k+1k} = \frac{a_{k+1k}}{a_{kk}}$

**Algoritm**  $G(A)$  % Varianta generală

1. **Pentru**  $k = 1 : n - 1$

1. **Pentru**  $i = k + 1 : n$

1.  $a_{ik} \leftarrow \mu_{ik} = \frac{a_{ik}}{a_{kk}}$

2. **Pentru**  $j = k + 1 : n$



Algoritmului EG pentru  $A = H$  superior Hessenberg:

**Algoritm** G(A) % Varianta Hessenberg

1. **Pentru**  $k = 1 : n - 1$

1. ~~**Pentru**  $i = k + 1 : n$~~

1.  $a_{k+1k} \leftarrow \mu_{k+1k} = \frac{a_{k+1k}}{a_{kk}}$

**Algoritm** G(A) % Varianta generală

1. **Pentru**  $k = 1 : n - 1$

1. **Pentru**  $i = k + 1 : n$

1.  $a_{ik} \leftarrow \mu_{ik} = \frac{a_{ik}}{a_{kk}}$

2. **Pentru**  $j = k + 1 : n$

1. **Pentru**  $i = k + 1 : n$



Algoritmului EG pentru  $A = H$  superior Hessenberg:

**Algoritm**  $G(A)$  % Varianta Hessenberg

1. **Pentru**  $k = 1 : n - 1$

1. **Pentru**  $i = k + 1 : n$

1.  $a_{k+1k} \leftarrow \mu_{k+1k} = \frac{a_{k+1k}}{a_{kk}}$

2. **Pentru**  $j = k + 1 : n$

**Algoritm**  $G(A)$  % Varianta generală

1. **Pentru**  $k = 1 : n - 1$

1. **Pentru**  $i = k + 1 : n$

1.  $a_{ik} \leftarrow \mu_{ik} = \frac{a_{ik}}{a_{kk}}$

2. **Pentru**  $j = k + 1 : n$

1. **Pentru**  $i = k + 1 : n$

1.  $a_{ij} \leftarrow a_{ij} - \mu_{ik} a_{kj}$



Algoritmului EG pentru  $A = H$  superior Hessenberg:

**Algoritm**  $G(A)$  % Varianta Hessenberg

1. **Pentru**  $k = 1 : n - 1$

1. ~~**Pentru**  $i = k + 1 : n$~~

1.  $a_{k+1k} \leftarrow \mu_{k+1k} = \frac{a_{k+1k}}{a_{kk}}$

2. **Pentru**  $j = k + 1 : n$

1. ~~**Pentru**  $i = k + 1 : n$~~

1.  $a_{k+1j} \leftarrow a_{k+1j} - \mu_{k+1k} a_{kj}$

**Algoritm**  $G(A)$  % Varianta generală

1. **Pentru**  $k = 1 : n - 1$

1. **Pentru**  $i = k + 1 : n$

1.  $a_{ik} \leftarrow \mu_{ik} = \frac{a_{ik}}{a_{kk}}$

2. **Pentru**  $j = k + 1 : n$

1. **Pentru**  $i = k + 1 : n$

1.  $a_{ij} \leftarrow a_{ij} - \mu_{ik} a_{kj}$





Algoritmului EG pentru  $A = H$  superior Hessenberg:

**Algorithm**  $G(A)$  % Varianta Hessenberg

1. **Pentru**  $k = 1 : n - 1$

1. ~~**Pentru**  $i = k + 1 : n$~~

1.  $a_{k+1k} \leftarrow \mu_{k+1k} = \frac{a_{k+1k}}{a_{kk}}$

2. **Pentru**  $j = k + 1 : n$

1. ~~**Pentru**  $i = k + 1 : n$~~

1.  $a_{k+1j} \leftarrow a_{k+1j} - \mu_{k+1k} a_{kj}$

Important:

- Complexitate redusă de la  $\mathcal{O}(n^3)$  la  $\mathcal{O}(n^2)$
- Forma matricii multiplicatorilor  $M$  este inferior bidiagonală
- Forma economică rezultată  $\tilde{A}$  este superior Hessenberg



Modificăm algoritmul  $G$  prin interschimbarea de linii (și/sau coloane) pentru a aduce în poziția pivotului un element nenul.

$$A_k = \begin{bmatrix} u_{11} & \dots & u_{1k} & \dots & u_{1n} \\ 0 & \ddots & & & \\ & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & 0 & & \dots & \\ & & a_{i_k k}^{(k)} & \dots & a_{i_k n}^{(k)} \\ & 0 & & \dots & \\ & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{bmatrix} \quad P_{ki_k} A_k = \begin{bmatrix} u_{11} & \dots & u_{1k} & \dots & u_{1n} \\ 0 & \ddots & & & \\ & & a_{i_k k}^{(k)} & \dots & a_{i_k n}^{(k)} \\ & 0 & & \dots & \\ & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & 0 & & \dots & \\ & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{bmatrix}$$

## Modificare Pas k:

1. Se determină cel mai mic  $i_k : |a_{i_k k}| = \max_{i=k:n} |a_{ik}|$



Modificăm algoritmul  $G$  prin interschimbarea de linii (și/sau coloane) pentru a aduce în poziția pivotului un element nenul.

$$A_k = \begin{bmatrix} u_{11} & \dots & u_{1k} & \dots & u_{1n} \\ 0 & \ddots & & & \\ & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & 0 & & \dots & \\ & & a_{i_k k}^{(k)} & \dots & a_{i_k n}^{(k)} \\ & 0 & & \dots & \\ & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{bmatrix} \quad P_{ki_k} A_k = \begin{bmatrix} u_{11} & \dots & u_{1k} & \dots & u_{1n} \\ 0 & \ddots & & & \\ & & a_{i_k k}^{(k)} & \dots & a_{i_k n}^{(k)} \\ & 0 & & \dots & \\ & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & 0 & & \dots & \\ & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{bmatrix}$$

## Modificare Pas k:

1. Se determină cel mai mic  $i_k : |a_{i_k k}| = \max_{i=k:n} |a_{ik}|$
2. Se interschimbă liniile  $i_k$  și  $k$ :  $A \leftarrow P_{i_k k} A$



Modificăm algoritmul  $G$  prin interschimbarea de linii (și/sau coloane) pentru a aduce în poziția pivotului un element nenul.

$$A_k = \begin{bmatrix} u_{11} & \dots & u_{1k} & \dots & u_{1n} \\ 0 & \ddots & & & \\ & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & 0 & & \dots & \\ & & a_{i_k k}^{(k)} & \dots & a_{i_k n}^{(k)} \\ & 0 & & \dots & \\ & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{bmatrix} \quad P_{ki_k} A_k = \begin{bmatrix} u_{11} & \dots & u_{1k} & \dots & u_{1n} \\ 0 & \ddots & & & \\ & & a_{i_k k}^{(k)} & \dots & a_{i_k n}^{(k)} \\ & 0 & & \dots & \\ & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & 0 & & \dots & \\ & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{bmatrix}$$

## Modificare Pas k:

1. Se determină cel mai mic  $i_k : |a_{i_k k}| = \max_{i=k:n} |a_{ik}|$
2. Se interschimbă liniile  $i_k$  și  $k$ :  $A \leftarrow P_{i_k k} A$
3. Se calculează  $M_k$  pentru  $(M_k A)_{ik} = 0, i = k + 1 : n$



Modificăm algoritmul  $G$  prin interschimbarea de linii (și/sau coloane) pentru a aduce în poziția pivotului un element nenul.

$$A_k = \begin{bmatrix} u_{11} & \dots & u_{1k} & \dots & u_{1n} \\ 0 & \ddots & & & \\ & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & 0 & & \dots & \\ & & a_{ik}^{(k)} & \dots & a_{in}^{(k)} \\ & 0 & & \dots & \\ & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{bmatrix} \quad P_{ki_k} A_k = \begin{bmatrix} u_{11} & \dots & u_{1k} & \dots & u_{1n} \\ 0 & \ddots & & & \\ & & a_{i_k k}^{(k)} & \dots & a_{i_k n}^{(k)} \\ & 0 & & \dots & \\ & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & 0 & & \dots & \\ & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{bmatrix}$$

## Modificare Pas k:

1. Se determină cel mai mic  $i_k : |a_{i_k k}| = \max_{i=k:n} |a_{ik}|$
2. Se interschimbă liniile  $i_k$  și  $k$ :  $A \leftarrow P_{i_k k} A$
3. Se calculează  $M_k$  pentru  $(M_k A)_{ik} = 0, i = k + 1 : n$
4. Se aplică transformarea  $A \leftarrow M_k A$



Modificăm algoritmul  $G$  prin interschimbarea de linii (și/sau coloane) pentru a aduce în poziția pivotului un element nenul.

$$A_k = \begin{bmatrix} u_{11} & \dots & u_{1k} & \dots & u_{1n} \\ 0 & \ddots & & & \\ & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & 0 & & \dots & \\ & & a_{ik}^{(k)} & \dots & a_{in}^{(k)} \\ & 0 & & \dots & \\ & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{bmatrix} \quad P_{ki_k} A_k = \begin{bmatrix} u_{11} & \dots & u_{1k} & \dots & u_{1n} \\ 0 & \ddots & & & \\ & & a_{i_k k}^{(k)} & \dots & a_{i_k n}^{(k)} \\ & 0 & & \dots & \\ & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & 0 & & \dots & \\ & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{bmatrix}$$

## Modificare Pas $k$ :

1. Se determină cel mai mic  $i_k : |a_{i_k k}| = \max_{i=k:n} |a_{ik}|$
2. Se interschimbă liniile  $i_k$  și  $k$ :  $A \leftarrow P_{i_k k} A$
3. Se calculează  $M_k$  pentru  $(M_k A)_{ik} = 0, i = k + 1 : n$
4. Se aplică transformarea  $A \leftarrow M_k A$

Pe scurt:  $A_{k+1} = M_k P_k A_k$

În final:  $U := A_n = M_{n-1} P_{n-1} M_{n-2} P_{n-2} \dots M_1 P_1 A_k$



## Algoritm $GPP(A)$

1. **Pentru**  $k = 1 : n - 1$

1. Se determină cel mai mic  $i_k : |a_{i_k k}| = \max_{i=k:n} |a_{ik}|$

2.  $p(k) = i_k$

3. **Pentru**  $j = k : n$

1.  $a_{kj} \leftrightarrow a_{i_k j}$

4. **Pentru**  $i = k + 1 : n$

1.  $a_{ik} \leftarrow \mu_{ik} = \frac{a_{ik}}{a_{kk}}$

5. **Pentru**  $j = k + 1 : n$

1. **Pentru**  $i = k + 1 : n$

1.  $a_{ij} \leftarrow a_{ij} - \mu_{ik} a_{kj}$

- În final:  $Ux = Mb$  se rezolvă cu UTRIS



## Algoritm $GPP(A)$

1. **Pentru**  $k = 1 : n - 1$

1. Se determină cel mai mic  $i_k : |a_{i_k k}| = \max_{i=k:n} |a_{ik}|$

2.  $p(k) = i_k$

3. **Pentru**  $j = k : n$

1.  $a_{kj} \leftrightarrow a_{i_k j}$

4. **Pentru**  $i = k + 1 : n$

1.  $a_{ik} \leftarrow \mu_{ik} = \frac{a_{ik}}{a_{kk}}$

5. **Pentru**  $j = k + 1 : n$

1. **Pentru**  $i = k + 1 : n$

1.  $a_{ij} \leftarrow a_{ij} - \mu_{ik} a_{kj}$

- În final:  $Ux = Mb$  se rezolvă cu UTRIS
- Complexitate suplimentară față de  $G$ :  $\sum_{k=1}^{n-1} (n - k + 1) \approx \frac{n^2}{2} = \mathcal{O}(n^2)$





- Introducere. Vectori. Operații elementare
- Subspații liniare. Produs scalar. Norme
- Matrice. Operații elementare. Proprietăți
- Sisteme de ecuații liniare pătratice
- Algoritmi de rezolvare a SL pătratice
- **Ortogonalitate. Sisteme generale**



- Vectorii  $u, v \in \mathbb{R}^{n \times n}$  se numesc ortogonali dacă  $u^T v = 0$ ;



- Vectorii  $u, v \in \mathbb{R}^{n \times n}$  se numesc ortogonali dacă  $u^T v = 0$ ;
- Vectorii  $u, v \in \mathbb{R}^{n \times n}$  se numesc ortonormali dacă  $u^T v = 0$  și  $\|u\|_2 = \|v\|_2 = 1$



- Vectorii  $u, v \in \mathbb{R}^{n \times n}$  se numesc ortogonali dacă  $u^T v = 0$ ;
- Vectorii  $u, v \in \mathbb{R}^{n \times n}$  se numesc ortonormali dacă  $u^T v = 0$  și  $\|u\|_2 = \|v\|_2 = 1$
- O matrice pătrată  $Q \in \mathbb{R}^{n \times n}$  se numește ortogonală dacă  $Q^T Q = I_n$ ;  
echivalent,  $q_i^T q_j = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$  (coloanele sunt vectori ortonormali)



- Vectorii  $u, v \in \mathbb{R}^{n \times n}$  se numesc ortogonali dacă  $u^T v = 0$ ;
- Vectorii  $u, v \in \mathbb{R}^{n \times n}$  se numesc ortonormali dacă  $u^T v = 0$  și  $\|u\|_2 = \|v\|_2 = 1$
- O matrice pătrată  $Q \in \mathbb{R}^{n \times n}$  se numește ortogonală dacă  $Q^T Q = I_n$ ;  
echivalent,  $q_i^T q_j = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$  (coloanele sunt vectori ortonormali)
- Evident,  $Q^{-1} = Q^T$



- Vectorii  $u, v \in \mathbb{R}^{n \times n}$  se numesc ortogonali dacă  $u^T v = 0$ ;
- Vectorii  $u, v \in \mathbb{R}^{n \times n}$  se numesc ortonormali dacă  $u^T v = 0$  și  $\|u\|_2 = \|v\|_2 = 1$
- O matrice pătrată  $Q \in \mathbb{R}^{n \times n}$  se numește ortogonală dacă  $Q^T Q = I_n$ ;  
echivalent,  $q_i^T q_j = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$  (coloanele sunt vectori ortonormali)
- Evident,  $Q^{-1} = Q^T$
- O matrice  $Q \in \mathbb{R}^{m \times n}$  are coloanele ortonormale dacă  $Q^T Q = I_n$  ( $m > n$ ,  $Q Q^T \neq I_m$ ); de asemenea,  $Q$  are liniile ortonormale dacă  $Q Q^T = I_m$  ( $m < n$ ,  $Q^T Q \neq I_n$ )



- Vectorii  $u, v \in \mathbb{R}^{n \times n}$  se numesc ortogonali dacă  $u^T v = 0$ ;
- Vectorii  $u, v \in \mathbb{R}^{n \times n}$  se numesc ortonormali dacă  $u^T v = 0$  și  $\|u\|_2 = \|v\|_2 = 1$
- O matrice pătrată  $Q \in \mathbb{R}^{n \times n}$  se numește ortogonală dacă  $Q^T Q = I_n$ ;  
echivalent,  $q_i^T q_j = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$  (coloanele sunt vectori ortonormali)
- Evident,  $Q^{-1} = Q^T$
- O matrice  $Q \in \mathbb{R}^{m \times n}$  are coloanele ortonormale dacă  $Q^T Q = I_n$  ( $m > n$ ,  $Q Q^T \neq I_m$ ); de asemenea,  $Q$  are liniile ortonormale dacă  $Q Q^T = I_m$  ( $m < n$ ,  $Q^T Q \neq I_n$ )
- Evident,  $Q^{-1} = Q^T$



- Vectorii  $u, v \in \mathbb{R}^{n \times n}$  se numesc ortogonali dacă  $u^T v = 0$ ;
- Vectorii  $u, v \in \mathbb{R}^{n \times n}$  se numesc ortonormali dacă  $u^T v = 0$  și  $\|u\|_2 = \|v\|_2 = 1$
- O matrice pătrată  $Q \in \mathbb{R}^{n \times n}$  se numește ortogonală dacă  $Q^T Q = I_n$ ;  
echivalent,  $q_i^T q_j = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$  (coloanele sunt vectori ortonormali)
- Evident,  $Q^{-1} = Q^T$
- O matrice  $Q \in \mathbb{R}^{m \times n}$  are coloanele ortonormale dacă  $Q^T Q = I_n$  ( $m > n$ ,  $Q Q^T \neq I_m$ ); de asemenea,  $Q$  are liniile ortonormale dacă  $Q Q^T = I_m$  ( $m < n$ ,  $Q^T Q \neq I_n$ )
- Evident,  $Q^{-1} = Q^T$
- Exemple:  $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}$





- O matrice  $Q \in \mathbb{R}^{n \times n}$  ortogonală reprezintă o transformare liniară ce conservă norma euclidiană (Frobenius), i.e.

$$\|Qx\|_2 = \|x\|_2$$



- O matrice  $Q \in \mathbb{R}^{n \times n}$  ortogonală reprezintă o transformare liniară ce conservă norma euclidiană (Frobenius), i.e.

$$\|Qx\|_2 = \|x\|_2$$

- Verificare:  $\|Qx\|_2 = \sqrt{x^T \underbrace{Q^T Q}_{I_n} x} = \sqrt{x^T x} = \|x\|_2$

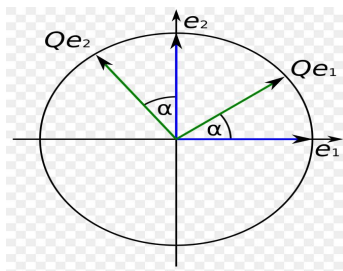


- O matrice  $Q \in \mathbb{R}^{n \times n}$  ortogonală reprezintă o transformare liniară ce conservă norma euclidiană (Frobenius), i.e.

$$\|Qx\|_2 = \|x\|_2$$

- Verificare:  $\|Qx\|_2 = \sqrt{x^T \underbrace{Q^T Q}_{I_n} x} = \sqrt{x^T x} = \|x\|_2$

- Transformarea  $Q$  aplicată lui  $x$  aduce schimbări asupra direcției lui  $x$  (fără a schimba "lungimea" sa)



de.wikiversity.org

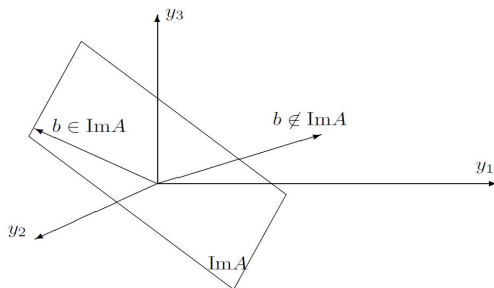


## Sistemul

$$Ax = b$$

- Subdeterminat:  $m < n$ , posibil o infinitate de soluții
- Determinat:  $m = n$ , adesea soluție unică
- Supradeterminat:  $m > n$ , adesea nu are soluție

**Teoremă.** Sistemul  $Ax = b$  are soluție dacă și numai dacă  $b \in \text{Im}(A)$ .



## Sistemul

$$Ax = b$$

Redefinim noțiunea de soluție a sistemului:

- $m \geq n : \|b - Ax^*\| = \min_{x \in \mathbb{R}^n} \|b - Ax\|$  (problema CMMP)



## Sistemul

$$Ax = b$$

Redefinim noțiunea de soluție a sistemului:

- $m \geq n : \|b - Ax^*\| = \min_{x \in \mathbb{R}^n} \|b - Ax\|$  (problema CMMP)
- $m \leq n : \|x^*\| = \min_{s.t. Ax=b} \|x\|$  (soluție normală CMMP)



**Definiție.** Fie  $u \in \mathbb{R}^m$  normat, i.e.  $\|u\| = 1$ . O matrice  $U \in \mathbb{R}^{m \times m}$  de forma

$$U = I_m - 2uu^T$$

se numește reflector elementar Householder de ordin  $m$

- $U^T U = (I_m - 2uu^T)^T (I_m - 2uu^T) = I_m - 4uu^T + 4u(u^T u)u^T = I_m$



**Definiție.** Fie  $u \in \mathbb{R}^m$  normat, i.e.  $\|u\| = 1$ . O matrice  $U \in \mathbb{R}^{m \times m}$  de forma

$$U = I_m - 2uu^T$$

se numește reflector elementar Householder de ordin  $m$

- $U^T U = (I_m - 2uu^T)^T (I_m - 2uu^T) = I_m - 4uu^T + 4u(u^T u)u^T = I_m$
- $U^T = U$





**Definiție.** Fie  $u \in \mathbb{R}^m$  normat, i.e.  $\|u\| = 1$ . O matrice  $U \in \mathbb{R}^{m \times m}$  de forma

$$U = I_m - 2uu^T$$

se numește reflector elementar Householder de ordin  $m$

- $U^T U = (I_m - 2uu^T)^T (I_m - 2uu^T) = I_m - 4uu^T + 4u(u^T u)u^T = I_m$
- $U^T = U$
- $Ux = (I_m - (1/\beta)uu^T)x = x - \left(\frac{u^T x}{\beta}\right)u$



**Definiție.** Fie  $u \in \mathbb{R}^m$  normat, i.e.  $\|u\| = 1$ . O matrice  $U \in \mathbb{R}^{m \times m}$  de forma

$$U = I_m - 2uu^T$$

se numește reflector elementar Householder de ordin  $m$

- $U^T U = (I_m - 2uu^T)^T (I_m - 2uu^T) = I_m - 4uu^T + 4u(u^T u)u^T = I_m$
- $U^T = U$
- $Ux = (I_m - (1/\beta)uu^T)x = x - \left(\frac{u^T x}{\beta}\right)u$
- pentru  $u = [0 \ \cdots \ 0 \ u_k \ \cdots \ u_m]^T$ ,  $U_k$  are forma:

$$U_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & \tilde{U} \end{bmatrix}$$



**Definiție.** Fie  $u \in \mathbb{R}^m$  normat, i.e.  $\|u\| = 1$ . O matrice  $U \in \mathbb{R}^{m \times m}$  de forma

$$U = I_m - 2uu^T$$

se numește reflector elementar Householder de ordin  $m$

- $U^T U = (I_m - 2uu^T)^T (I_m - 2uu^T) = I_m - 4uu^T + 4u(u^T u)u^T = I_m$
- $U^T = U$
- $Ux = (I_m - (1/\beta)uu^T)x = x - \left(\frac{u^T x}{\beta}\right)u$
- pentru  $u = [0 \ \cdots \ 0 \ u_k \ \cdots \ u_m]^T$ ,  $U_k$  are forma:

$$U_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & \tilde{U} \end{bmatrix}$$

- dacă  $\|u\| \neq 1$ , putem exprima  $U$ :

$$U = I_m - \frac{uu^T}{\beta}, \quad \beta = \frac{\|u\|^2}{2}$$



$$Ux = \left( I_m - (1/\beta)uu^T \right) x = x - \underbrace{\left( \frac{u^T x}{\beta} \right)}_{\nu} u$$

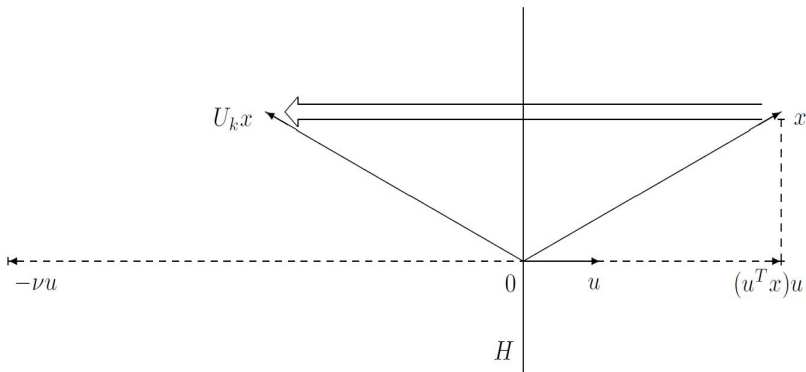


Fig. 3.1: Efectul aplicării unui reflector  $U$  asupra unui vector  $x$ , în  $\mathbb{R}^2$



Fie  $\sigma = \sqrt{\sum_{i=k}^m x_i^2}$  și reflectorul  $U_i = I_m - \frac{uu^T}{\beta}$ , unde:  $u = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ x_k + \sigma \\ x_{k+1} \\ \vdots \\ x_m \end{bmatrix}$ ,  $\beta = \sigma u_k$

atunci

$$X = \begin{bmatrix} x_1 \\ \dots \\ x_{k-1} \\ x_k \\ x_{k+1} \\ \dots \\ x_m \end{bmatrix} \rightarrow U_i X = \begin{bmatrix} x_1 \\ \dots \\ x_{k-1} \\ -\sigma \\ 0 \\ \dots \\ 0 \end{bmatrix}$$



Pasul 1: Calculăm reflectorul  $U_1$  astfel încât:

$$U_1 A = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}$$

---

## Algorithm 1: TrOrto(A)

---

```
1 for  $i = 1 : n$  do
2   |   Se calculează reflectorul Householder  $U_i$ 
3   |   Se realizează  $A = U_i A$ 
4 end
```

---



Pasul 2: Calculăm reflectorul  $U_2$  astfel încât:

$$U_2 U_1 A = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{bmatrix}$$

---

## Algorithm 1: TrOrto(A)

---

```
1 for  $i = 1 : n$  do
2   |   Se calculează reflectorul Householder  $U_i$ 
3   |   Se realizează  $A = U_i A$ 
4 end
```

---



Pasul 3: Calculăm reflectorul  $U_3$  astfel încât:

$$U_3 U_2 U_1 A = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

---

## Algorithm 1: TrOrto(A)

---

```
1 for  $i = 1 : n$  do  
2   |   Se calculează reflectorul Householder  $U_i$   
3   |   Se realizează  $A = U_i A$   
4 end
```

---





Pasul 4: Calculăm reflectorul  $U_4$  astfel încât:

$$U_4 U_3 U_2 U_1 A = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

---

## Algorithm 1: TrOrto(A)

---

```
1 for  $i = 1 : n$  do  
2   |   Se calculează reflectorul Householder  $U_i$   
3   |   Se realizează  $A = U_i A$   
4 end
```

---



Pasul 5: Calculăm reflectorul  $U_5$  astfel încât:

$$U_5 U_4 U_3 U_2 U_1 A = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

---

## Algorithm 1: TrOrto(A)

---

```
1 for  $i = 1 : n$  do  
2   | Se calculează reflectorul Householder  $U_i$   
3   | Se realizează  $A = U_i A$   
4 end
```

---



## Exercițiu

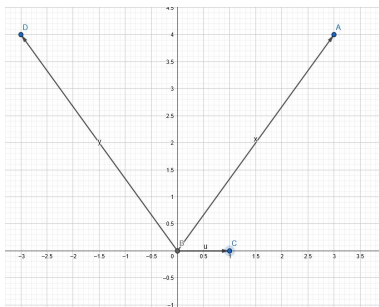
Fie vectorul  $x = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$  și vectorul Householder  $u = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ . Să se construiască vectorul transformat  $y = Ux$ . Indicație:  $U := I_n - 2uu^T$



## Exercițiu

Fie vectorul  $x = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$  și vectorul Householder  $u = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ . Să se construiască vectorul transformat  $y = Ux$ . Indicație:  $U := I_n - 2uu^T$

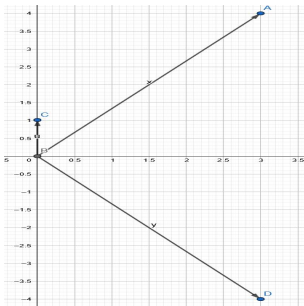
$$y = Ux = x - 2u^T x u = \begin{bmatrix} 3 \\ 4 \end{bmatrix} - 6 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -3 \\ 4 \end{bmatrix}$$



## Exercițiu

Fie vectorul  $x = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$  și vectorul Householder  $u = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ . Să se construiască vectorul transformat  $y = Ux$ .

$$y = Ux = x - 2u^T x u = \begin{bmatrix} 3 \\ 4 \end{bmatrix} - 8 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ -4 \end{bmatrix}$$



### Exercițiu

Fie vectorul  $x \in \mathbb{R}^n$ . Să se construiască reflectorul  $U$  care anulează elementele  $x_i, i = 2 : n$ .



$$Ux = \left( I_m - \frac{uu^T}{\beta} \right) x = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \beta = \frac{\|x\|_2^2}{2}$$



$$Ux = \left( I_m - \frac{uu^T}{\beta} \right) x = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \beta = \frac{\|x\|_2^2}{2}$$

Transformările ortogonale conservă norma (imaginați-vă o rotire a vectorului  $x$ ). De aceea deducem că  $\|x\| = \|Ux\| = \|\alpha e_1\| = |\alpha|$ , care implică

$$\alpha = \pm \|x\| \Rightarrow Ux = \pm \|x\| e_1 = x - (u^T x)u \quad | \quad \cdot x^T$$





$$Ux = \left( I_m - \frac{uu^T}{\beta} \right) x = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \beta = \frac{\|x\|_2^2}{2}$$

Transformările ortogonale conservă norma (imaginați-vă o rotire a vectorului  $x$ ). De aceea deducem că  $\|x\| = \|Ux\| = \|\alpha e_1\| = |\alpha|$ , care implică

$$\alpha = \pm \|x\| \Rightarrow Ux = \pm \|x\| e_1 = x - (u^T x)u \quad | \quad \cdot x^T$$

Înmulțim ambii membrii cu  $x$  (produs scalar)  $\Rightarrow (u^T x)^2 = \|x\|^2 \mp \|x\| x_1$



$$Ux = \left( I_m - \frac{uu^T}{\beta} \right) x = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \beta = \frac{\|x\|_2^2}{2}$$

Transformările ortogonale conservă norma (imaginați-vă o rotire a vectorului  $x$ ). De aceea deducem că  $\|x\| = \|Ux\| = \|\alpha e_1\| = |\alpha|$ , care implică

$$\alpha = \pm \|x\| \Rightarrow Ux = \pm \|x\| e_1 = x - (u^T x)u \quad | \quad \cdot x^T$$

Înmulțim ambii membrii cu  $x$  (produs scalar)  $\Rightarrow (u^T x)^2 = \|x\|^2 \mp \|x\| x_1$

$$\pm \|x\| e_1 = x - \sqrt{\|x\|^2 \mp \|x\| x_1} u \Rightarrow u = \frac{1}{\sqrt{\|x\|^2 \mp \|x\| x_1}} (x \mp \|x\| e_1)$$



$$Ux = \left( I_m - \frac{uu^T}{\beta} \right) x = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \beta = \frac{\|x\|_2^2}{2}$$

Transformările ortogonale conservă norma (imaginați-vă o rotire a vectorului  $x$ ). De aceea deducem că  $\|x\| = \|Ux\| = \|\alpha e_1\| = |\alpha|$ , care implică

$$\alpha = \pm \|x\| \Rightarrow Ux = \pm \|x\| e_1 = x - (u^T x)u \quad | \cdot x^T$$

Inmultim ambii membrii cu  $x$  (produs scalar)  $\Rightarrow (u^T x)^2 = \|x\|^2 \mp \|x\| x_1$

$$\pm \|x\| e_1 = x - \sqrt{\|x\|^2 \mp \|x\| x_1} u \Rightarrow u = \frac{1}{\sqrt{\|x\|^2 \mp \|x\| x_1}} (x \mp \|x\| e_1)$$

$$\Rightarrow Ux = x - (x \mp \|x\| e_1) = \pm \|x\| e_1$$



Fie  $\sigma = \sqrt{\sum_{i=k}^m x_i^2}$  și reflectorul  $U_i = I_m - \frac{uu^T}{\beta}$ , unde  $\beta = \sigma u_k$  atunci alegând

$$u = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ x_k + \sigma \\ x_{k+1} \\ \vdots \\ x_m \end{bmatrix} \rightarrow U_i x = \begin{bmatrix} x_1 \\ \vdots \\ x_{k-1} \\ -\sigma \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

---

### Algorithm 1: TrOrto(A)

---

```
1 for  $i = 1 : n$  do
2   | Se calculează reflectorul Householder  $U_i$ 
3   | Se realizează  $A = U_i A$ 
4 end
```



Fie  $A \in \mathbb{R}^{m \times n}$  cu  $m \geq n$  și  $\text{rang}(A) = n$  atunci

$$\text{CMMP} : \min_x \|Ax - b\|_2$$

Din alg. 1  $\Rightarrow U_n U_{n-1} \cdots U_1 A = R$ , adică

$$A = (U_n U_{n-1} \cdots U_1)^T R = Q \left\{ \begin{array}{c} R' \\ 0 \end{array} \right\} \begin{array}{c} n \\ m - n \end{array}$$

$R' \in \mathbb{R}^{n \times n}$  superior triunghiulară,  $Q$  ortonormală

Calcul soluție CMMP: dacă  $Qb = \left\{ \begin{array}{c} d' \\ d'' \end{array} \right\} \begin{array}{c} n \\ m - n \end{array}$  atunci

$$\|Ax - b\|_2^2 = \|Q^T Ax - Q^T b\|_2^2 = \|R'x - d'\|_2^2 + \|d''\|_2^2$$



Fie  $A \in \mathbb{R}^{m \times n}$  cu  $m \geq n$  și  $\text{rang}(A) = n$  atunci

$$CMMP : \min_x \|Ax - b\|_2$$

Din alg. 1  $\Rightarrow U_n U_{n-1} \cdots U_1 A = R$ , adică

$$A = (U_n U_{n-1} \cdots U_1)^T R = Q \left\{ \begin{array}{c} R' \\ 0 \end{array} \right\} \quad \left. \begin{array}{l} n \\ m-n \end{array} \right\}$$

$R' \in \mathbb{R}^{n \times n}$  superior triunghiulară,  $Q$  ortonormală

Deoarece  $\text{rang}(A) = \text{rang}(R') = n$  avem:

$$R' x_{CMMP}^* = d'$$

$$r(x_{GMP}^*) = \|d''\|_2^2$$



### Factorizare QR cu reflectori Householder:

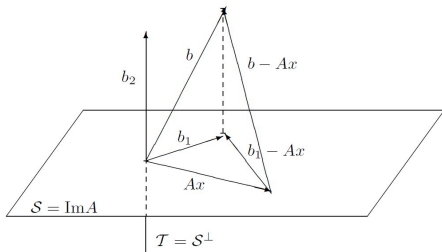
$$Ax \stackrel{CMMP}{\approx} b \Rightarrow Q^T A = R \Rightarrow A = QR = Q \begin{bmatrix} R' \\ 0 \end{bmatrix}$$

$R' \in \mathbb{R}^{n \times n}$  superior triunghiulară,  $Q$  ortonormală

Calcul pseudosoluție:

$$b = [b_1 \quad b_2]^T, Q = [Q' \quad Q''], b_1 \in \mathbb{R}^n, b_2 \in \mathbb{R}^{m-n}, Q' \in \mathbb{R}^{n \times n}, Q'' \in \mathbb{R}^{m-n \times n}$$

$$\min_x \|b - QRx\| = \min_x \|b - \begin{bmatrix} Q' & Q'' \end{bmatrix} \begin{bmatrix} R' \\ 0 \end{bmatrix} x\| = \min_x \left\| \begin{bmatrix} b_1 - Q'R'x \\ b_2 \end{bmatrix} \right\|$$



Factorizare QR cu reflectori Householder:

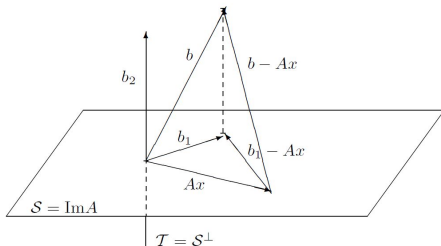
$$Ax \stackrel{CMMP}{\approx} b \Rightarrow Q^T A = R \Rightarrow A = QR = Q \begin{bmatrix} R' \\ 0 \end{bmatrix}$$

$R' \in \mathbb{R}^{n \times n}$  superior triunghiulară,  $Q$  ortonormală

Calcul pseudosoluție:

$$b = \begin{bmatrix} b_1 & b_2 \end{bmatrix}^T, Q = \begin{bmatrix} Q' & Q'' \end{bmatrix}, b_1 \in \mathbb{R}^n, b_2 \in \mathbb{R}^{m-n}, Q' \in \mathbb{R}^{n \times n}, Q'' \in \mathbb{R}^{m-n \times n}$$

$$R' x^* = d', \text{ unde } Qb = \begin{bmatrix} d' \\ d'' \end{bmatrix}$$





Factorizare QR cu reflectori Householder:

$$Ax \stackrel{CMMP}{=} b \Rightarrow Q^T A = R \Rightarrow A = QR = Q \begin{bmatrix} R' \\ 0 \end{bmatrix}$$

$R' \in \mathbb{R}^{n \times n}$  superior triunghiulară,  $Q$  ortonormală

Calcul pseudosoluție:

$$Q' R' x^* = b'$$

$$R' x^* = d', \text{ unde } Qb = \begin{bmatrix} d' \\ d'' \end{bmatrix}$$

Soluția CMMP satisface:

$$x^* = \arg \min_x \|Ax - b\|$$



Factorizare QR cu reflectori Householder:

$$Ax \stackrel{CMMP}{=} b \Rightarrow Q^T A = R \Rightarrow A = QR = Q \begin{bmatrix} R' \\ 0 \end{bmatrix}$$

Algoritm:

- 1. Factorizare  $QR$  :  $Q^T A = \begin{bmatrix} R \\ 0 \end{bmatrix}$



Factorizare QR cu reflectori Householder:

$$Ax \stackrel{CMMP}{=} b \Rightarrow Q^T A = R \Rightarrow A = QR = Q \begin{bmatrix} R' \\ 0 \end{bmatrix}$$

Algoritm:

- 1. Factorizare  $QR$  :  $Q^T A = \begin{bmatrix} R \\ 0 \end{bmatrix}$
- 2. Calcul :  $Q^T b = d = \begin{bmatrix} d' \\ d'' \end{bmatrix}$



Factorizare QR cu reflectori Householder:

$$Ax \stackrel{CMMP}{=} b \Rightarrow Q^T A = R \Rightarrow A = QR = Q \begin{bmatrix} R' \\ 0 \end{bmatrix}$$

Algoritm:

- 1. Factorizare  $QR$  :  $Q^T A = \begin{bmatrix} R \\ 0 \end{bmatrix}$
- 2. Calcul :  $Q^T b = d = \begin{bmatrix} d' \\ d'' \end{bmatrix}$
- 3. Rezolvă:  $Rx = d'$



- Carte BD,CP,BJ
- <http://web.stanford.edu/class/cs205l/lectures.html>

