CALCUL NUMERIC
Curs 2

Paul Irofti
Cristian Rusu
Andrei Pătraşcu

Departament Informatică
Universitatea din Bucureşti

- **Subspaţii liniare**
- Matrice. Operaţii elementare. Proprietăţi
- Sisteme de ecuaţii liniare pătratice
- Algoritmi de rezolvare a SL pătratice
- Ortogonalitate. Sisteme generale

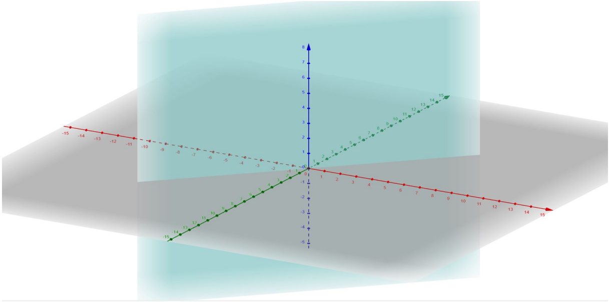
O mulţime S de vectori din $\mathbb{R}^n$ este numită **subspaţiu liniar** al spaţiului $\mathbb{R}^n$ dacă:

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

O mulţime S de vectori din $\mathbb{R}^n$ este numită **subspaţiu liniar** al spaţiului $\mathbb{R}^n$ 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 $S = \{x \in \mathbb{R}\}$



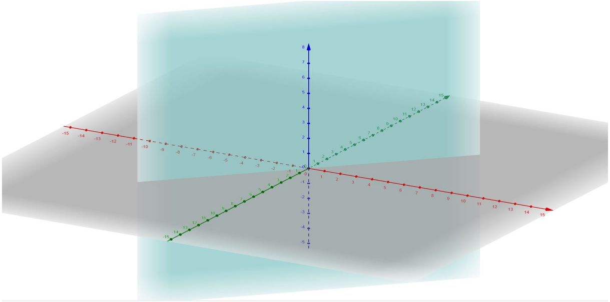
Geogebra.org

Subspaţiu Liniar

O mulţime S de vectori din $\mathbb{R}^n$ este numită **subspaţiu liniar** al spaţiului $\mathbb{R}^n$ 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 $S = \{x \in \mathbb{R}^2 : x_1 + 2x_2 = 0\}$



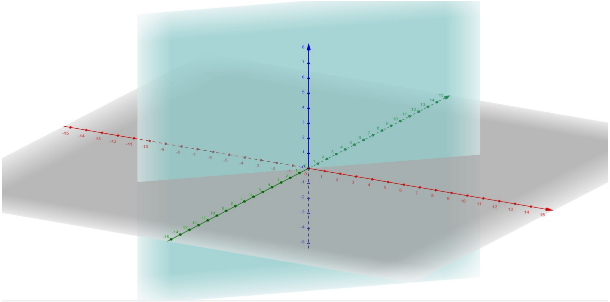
Geogebra.org

Subspaţiu Liniar

O mulţime S de vectori din $\mathbb{R}^n$ este numită **subspaţiu liniar** al spaţiului $\mathbb{R}^n$ 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 $S = \{x \in \mathbb{R}^2 : x_1 + 2x_2 = 0\}$
- in $\mathbb{R}^3$: plan $S = \{x \in \mathbb{R}^3 : x_1 - x_2 = 0\}$

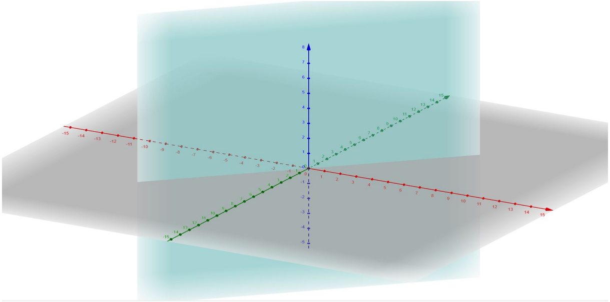


Geogebra.org

O mulţime S de vectori din $\mathbb{R}^n$ este numită **subspaţiu liniar** al spaţiului $\mathbb{R}^n$ 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 $S = \{x \in \mathbb{R}^2 : x_1 + 2x_2 = 0\}$
- in $\mathbb{R}^3$: plan $S = \{x \in \mathbb{R}^3 : x_1 - x_2 = 0\}$
- in $\mathbb{R}^n$: subspaţiu $S = \{x \in \mathbb{R}^n : Bx = 0\}$, unde $B \in \mathbb{R}^{m \times n}$



Geogebra.org

O mulţime S de vectori din $\mathbb{R}^n$ este numită **subspaţiu liniar** al spaţiului $\mathbb{R}^n$ 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 $S = \{x \in \mathbb{R}^3 : x_1 - x_2 = 0\}$
- in $\mathbb{R}^n$: subspaţiu $S = \{x \in \mathbb{R}^n : Ax = 0\}$, unde $A \in \mathbb{R}^{m \times n}$

Combinatiile liniare ale vectorilor $B = \{b_1, b_2, \cdots, b_p\}$ generează un subspaţiu liniar!

O mulţime B de vectori din $\mathbb{R}^n$ este numită **bază** al spaţiului $S \subseteq \mathbb{R}^n$ 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 $S = span \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$

O mulţime B de vectori din $\mathbb{R}^n$ este numită **bază** al spaţiului $S \subseteq \mathbb{R}^n$ 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 $S = span \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$
- $S = \{x \in \mathbb{R}^n : x = \alpha_1 b_1 + \cdots \alpha_p b_p, \alpha \in \mathbb{R}^p\}$ sau $S = span\{b_1, \cdots, b_p\}$

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

- elementele din B sunt liniar independente
- B generează S

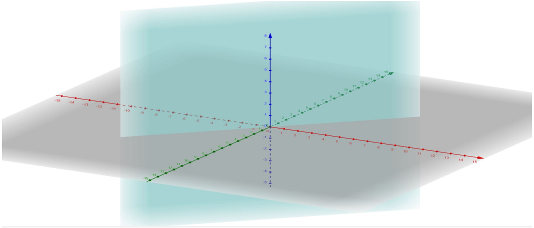
Exemple:

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

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

Exemple:

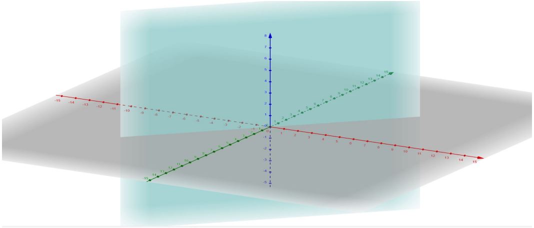
- $S = \left\{ x \in \mathbb{R}^n : x = \alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \alpha \in \mathbb{R} \right\}$ => Dimensiune $Dim(S) = 1$



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

Exemple:

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



În orice (sub)spaţiu este necesară o metrică de măsură a distanţelor

Multe probleme de calcul numeric şi învăţare automată se formulează în termeni de distanţe:

- SVM, CMMP: Care este cel mai ”apropiat” punct ce aparţine unei mulţimi?

În orice (sub)spaţiu este necesară o metrică de măsură a distanţelor

Multe probleme de calcul numeric şi învăţare automată se formulează în termeni de distanţe:

- SVM, CMMP: Care este cel mai ”apropiat” punct ce aparţine unei mulţimi?
- Regresie liniară: Care este funcţia liniară care explică cel mai bine un set de date? (minimizează ”distanţa” de la model la fiecare punct dat)

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

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 mulţimea 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 mulţimea 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 mulţimea 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 mulţimea 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 subspaţiile:

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

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

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

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

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

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

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

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

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:

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

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

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

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

- 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 :

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

- T tridiagonală:

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

Cuprins

- Subspaţii liniare. Produs scalar. Norme
- Matrice. Operaţii elementare. Proprietăţi
- **Sisteme de ecuaţii liniare pătratice**
- Algoritmi de rezolvare a SL pătratice
- Ortogonalitate. Sisteme generale

Un sistem de *m* ecuaţii cu *n* necunoscute are forma:

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

Forma matriceala

$$Ax = b,$$

unde A este matricea coeficienţilor, x vector necunoscutelor, b termen liber

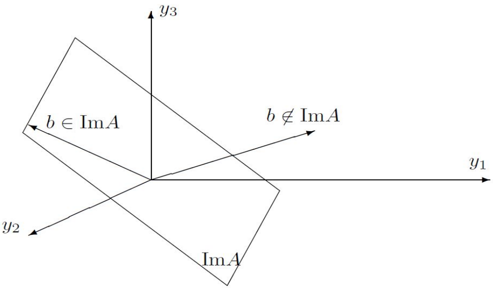
$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Sistemul

$$Ax = b$$

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

Teoremă. Sistemul $Ax = b$ are soluţie dacă şi numai dacă $b \in Im(A)$.



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 soluţie particulară: $x = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$
- Mulţimea tuturor soluţiilor: $X = \left\{ x = \begin{bmatrix} 1+\alpha \\ 1-\alpha \\ \alpha \end{bmatrix}, \alpha \in \mathbb{R} \right\}$

Notiuni legate de solvabilitatea unui sistem liniar pătratic:

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

Notiuni legate de solvabilitatea unui sistem liniar pătratic:

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

Notiuni legate de solvabilitatea unui sistem liniar pătratic:

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

Notiuni legate de solvabilitatea unui sistem liniar pătratic:

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

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

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

- Nu este o formulă adecvată calculului numeric (vrem sa evităm A^{-1})
- In general, existenţa unei soluţii este dificil de determinat

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:

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

Caz diagonal

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:

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

Rezolvare:

$$\begin{aligned} \color{blue}{x_1} &= 1 \\ \color{blue}{x_2} &= 1/2 \\ \color{blue}{x_3} &= 1/2 \end{aligned}$$

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:

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

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

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

Caz diagonal

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:

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

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

- $A_{ii} \neq 0$ este necesar pentru soluţie 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:

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

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

- $A_{ii} \neq 0$ este necesar pentru soluţie 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:

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

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:

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

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:

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

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:

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

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:

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

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:

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

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:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &= b_1 \\ a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{33}x_3 + a_{34}x_4 &= b_3 \\ \cdots&\cdots \\ a_{n-1n}x_n &= b_n \end{aligned}$$

Caz bidiagonal general

Matricea A (superior) bidiagonală:

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

Algorithm $BDTris(A, b)$:

Matricea A (superior) bidiagonală:

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

Algorithm $BDTris(A, b)$:

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

Matricea A (superior) bidiagonală:

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

Algorithm $BDTris(A, b)$:

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

Matricea A (superior) bidiagonală:

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

Algorithm $BDTris(A, b)$:

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

Matricea A (superior) bidiagonală:

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

Algorithm $BDTris(A, b)$:

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

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

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:

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

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:

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

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:

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

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:

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

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:

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

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

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

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. $x_i = x_i / l_{ii}$

- 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. $x_i = x_i / l_{ii}$

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

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. $x_i = x_i / l_{ii}$

- La fiecare pas al buclei necesită $2(i-1)$ flopi
- Complexitate: $\mathcal{O}\left(n^2\right)$ (comparativ cazul general $\mathcal{O}\left(n^3\right)$)
- Exercitiu: 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. $x_i = x_i / l_{ii}$

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

Cuprins

- Subspaţii liniare. Produs scalar. Norme
- Matrice. Operaţii elementare. Proprietăţi
- Sisteme de ecuaţii 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 iniţial: $\begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ 10 \end{bmatrix}$ triangularizăm succesiv coloanele lui A pentru a obţine forma:

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

Scop: echivalarea sistemului generic

$$Ax = b$$

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

$$Ux = \tilde{b}$$

Pornind de la sistemul iniţial: $\begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ 10 \end{bmatrix}$ triangularizăm succesiv coloanele lui A pentru a obţine 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 gă sirea unei transformări elementare M **care produce** $0-$**uri subdiagonale în coloana** a_i, i.e.

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

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

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

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

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

Triangularizare - exemplu

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

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

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

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

Definiţie. Transformare inferior triunghiulară elementară (ITE) de ordin n şi indice k are forma:

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

unde

$$m_k = \begin{bmatrix} 0 & 0 & \cdots & 0 & \mu_{k+1,k} & \cdots & \mu_{n,k} \end{bmatrix}$$

are primele k elemente nenule.

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

Proprietăţi ale matricilor de transformare ITE:

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

Proprietăţi ale matricilor de transformare ITE:

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

Proprietăţi ale matricilor de transformare ITE:

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

Proprietăţi ale matricilor de transformare ITE:

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

Proprietăţi ale matricilor de transformare ITE:

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

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \qquad M_2 x = \begin{bmatrix} x_1 \\ x_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ alegând } \mu_{i2} = \frac{x_i}{x_2}$$

Proprietăţi ale matricilor de transformare ITE:

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

$$x = \begin{bmatrix} x_1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad M_2 x = \begin{bmatrix} x_1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ alegând } \mu_{i2} = \frac{x_i}{x_2}$$

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

$$\dots\dots\dots$$

Pas k: $A_k = M_k \cdots M_2 M_1 A$ 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 $\mu_{ik} = \frac{a^k_{ik}}{a^k_{kk}}$
- 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.** Daca toate matricile lider principale $A^{[k]}$ din A sunt nesingulare, atunci algoritmul produce un U nesingular (sistemul are soluţie 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 $\mu_{ik} = \frac{a^k_{ik}}{a^k_{kk}}$
- 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.** Daca toate matricile lider principale $A^{[k]}$ din A sunt nesingulare, atunci algoritmul produce un U nesingular (sistemul are soluţie unică!)
- Altfel EG produce un U singular (care sunt consecinţele?)

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

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

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

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

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

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

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

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

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

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

- **Dacă $i < j$, atunci nu au loc suprapuneri de elemente!**
- **Consecinţă:** 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: $M = M_{n-1} \cdots M_2 M_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \mu_{21} & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \mu_{n1} & \mu_{n2} & \cdots & 1 \end{bmatrix}$
- μ_{ij} ocupă triunghiul strict inferior, A are progresiv aceste poziţii 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 μ_{ik} se pot memora in triunghiul inferior al matricii A

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

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 condiţiile teoremei $Ax = b$ are aceleaşi soluţii 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 condiţiile teoremei $Ax = b$ are aceleaşi soluţii 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 condiţiile teoremei $Ax = b$ are aceleaşi soluţii cu $Ux = Mb$; se rezolvă cu UTRIS
- $MA = U \Leftrightarrow A = M^{-1}U = LU$
- Complexitate totală: $\sum_{k=1}^{n-1}(n-k+2(n-k)^2) \approx \frac{2n^3}{3} = \mathcal{O}\left(n^3\right)$

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 condiţiile teoremei $Ax = b$ are aceleaşi soluţii cu $Ux = Mb$; se rezolvă cu UTRIS
- $MA = U \Leftrightarrow A = M^{-1}U = LU$
- Complexitate totală: $\sum_{k=1}^{n-1}(n-k+2(n-k)^2) \approx \frac{2n^3}{3} = \mathcal{O}\left(n^3\right)$
- 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 condiţiile teoremei $Ax = b$ are aceleaşi soluţii cu $Ux = Mb$; se rezolvă cu UTRIS
- $MA = U \Leftrightarrow A = M^{-1}U = LU$
- Complexitate totală: $\sum_{k=1}^{n-1}(n-k+2(n-k)^2) \approx \frac{2n^3}{3} = \mathcal{O}\left(n^3\right)$
- 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 situaţia aceasta, multiplicatorii μ_{ik} ?

Adaptaţi 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

Adaptaţi 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 paşi 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**~~ $\cancel{i = k + 1 : n}$
    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:

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

Important:

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

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

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

Modificare Pas k:

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

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

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

Modificare Pas k:

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

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

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

Modificare Pas k:

1. Se determină cel mai mic i_k : $|a_{i_k k}| = \max\limits_{i=k:n} |a_{ik}|$
2. Se interschimbă liniile i_k şi 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 (şi/sau coloane) pentru a aduce în poziţia pivotului un element nenul.

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

Modificare Pas k:

1. Se determină cel mai mic i_k : $|a_{i_k k}| = \max\limits_{i=k:n} |a_{ik}|$
2. Se interschimbă liniile i_k şi 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 (şi/sau coloane) pentru a aduce în poziţia pivotului un element nenul.

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

Modificare Pas k:

1. Se determină cel mai mic $i_k : |a_{i_k k}| = \max\limits_{i=k:n} |a_{ik}|$
2. Se interschimbă liniile i_k şi 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: $U := A_n = M_{n-1} P_{n-1} M_{n-2} P_{n-2} \cdots M_1 P_1 A_k$

Algoritm $GPP(A)$

1. **Pentru** $k = 1 : n - 1$
    1. Se determină cel mai mic $i_k : |a_{i_k k}| = \max\limits_{i=k:n} |a_{ik}|$
    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 : $|a_{i_k k}| = \max\limits_{i=k:n} |a_{ik}|$
    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 faţă de G: $\sum_{k=1}^{n-1}(n-k+1) \approx \frac{n^2}{2} = \mathcal{O}(n^2)$

- 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 $u^T v = 0$ şi $\|u\|_2 = \|v\|_2 = 1$

- 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 $u^T v = 0$ şi $\|u\|_2 = \|v\|_2 = 1$
- O matrice pătrată $Q \in \mathbb{R}^{n\times n}$ se numeşte 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 $u^T v = 0$ şi $\|u\|_2 = \|v\|_2 = 1$
- O matrice pătrată $Q \in \mathbb{R}^{n\times n}$ se numeşte 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 $u^T v = 0$ şi $\|u\|_2 = \|v\|_2 = 1$
- O matrice pătrată $Q \in \mathbb{R}^{n\times n}$ se numeşte 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 $Q^T Q = I_n (m > n, QQ^T \neq I_m)$; de asemenea, Q are liniile ortonormale dacă $QQ^T = I_m (m < n, Q^T Q \neq I_n)$

- 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 $u^T v = 0$ şi $\|u\|_2 = \|v\|_2 = 1$
- O matrice pătrată $Q \in \mathbb{R}^{n\times n}$ se numeşte 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 $Q^T Q = I_n (m > n, QQ^T \neq I_m)$; de asemenea, Q are liniile ortonormale dacă $QQ^T = I_m (m < n, Q^T Q \neq I_n)$
- 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 $u^T v = 0$ şi $\|u\|_2 = \|v\|_2 = 1$
- O matrice pătrată $Q \in \mathbb{R}^{n\times n}$ se numeşte 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 $Q^T Q = I_n (m > n, QQ^T \neq I_m)$; de asemenea, Q are liniile ortonormale dacă $QQ^T = I_m (m < n, Q^T Q \neq I_n)$
- 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.

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

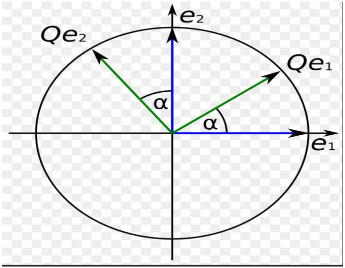
- Verificare: $\|Qx\|_2 = \sqrt{x^T \underbrace{Q^T Q}_{I_n} x} = \sqrt{x^T x} = \|x\|_2$

- 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 direcţiei lui x (fără a schimba "lungimea" sa)



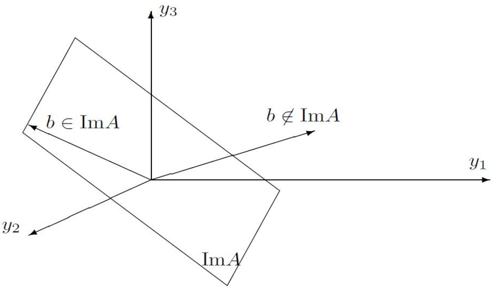
de.wikiversity.org

Sistemul

$$Ax = b$$

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

Teoremă. Sistemul $Ax = b$ are soluţie dacă şi numai dacă $b \in Im(A)$.



Sistemul

$$Ax = b$$

Redefinim noţiunea de soluţie a sistemului:

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

O aplicaţie vedeţi cursul următor!

Sistemul

$$Ax = b$$

Redefinim noţiunea de soluţie a sistemului:

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

O aplicaţie vedeţi cursul următor!

Sistemul

$$Ax = b$$

Redefinim noţiunea de soluţie a sistemului:

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

O aplicaţie de regresie liniară vedeţi cursul următor!

Sistemul

$$Ax = b$$

Redefinim noţiunea de soluţie a sistemului:

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

O aplicaţie de regresie liniară vedeţi cursul următor!

Definiţie. 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 numeşte reflector elementar Householder de ordin m

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

Definiţie. 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 numeşte reflector elementar Householder de ordin m

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

Definiţie. 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 numeşte reflector elementar Householder de ordin m

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

Definiţie. 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 numeşte reflector elementar Householder de ordin m

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

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

Definiţie. 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 numeşte reflector elementar Householder de ordin m

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

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

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

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

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

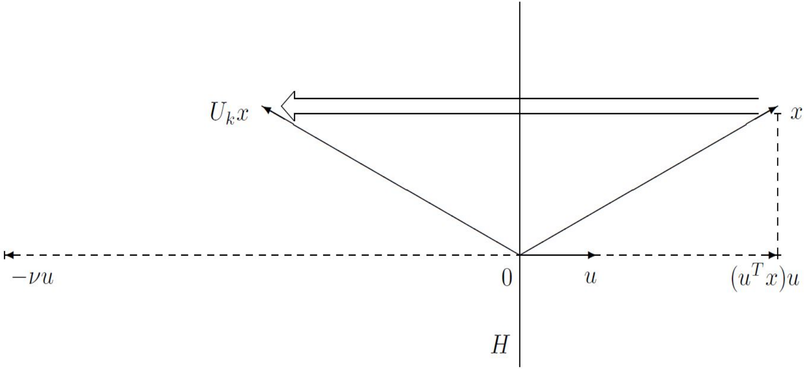


Fig. 3.1: Efectul aplicării unui reflector U asupra unui vector x, în $\mathbb{R}^2$

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

atunci

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

Factorizare QR cu reflectori Householder:

$$Ax \overset{CMMP}{=} b \Rightarrow \underbrace{U_1 A}_{A_1} x \Rightarrow U_2 U_1 A x \Rightarrow$$

$$\cdots \Rightarrow$$

$$\underbrace{U_m \cdots U_1 A}_{R} x = \underbrace{U_m \cdots U_1}_{Q^T} b$$

$$A_2 = U_1 A_1 = [U_1 a_1 \; U_1 a_2 \; \ldots \; U_1 a_n] = \begin{bmatrix} r_{11} & r_{12} & \ldots & r_{1n} \\ 0 & a_{22}^{(2)} & \ldots & a_{2n}^{(2)} \\ 0 & a_{32}^{(2)} & \ldots & a_{3n}^{(2)} \\ & & \ldots & \\ 0 & a_{m2}^{(2)} & \ldots & a_{mn}^{(2)} \end{bmatrix}$$

Factorizare QR cu reflectori Householder:

$$Ax \overset{CMMP}{=} b \Rightarrow \underbrace{U_1 A}_{A_1} x \Rightarrow U_2 U_1 A x \Rightarrow$$

$$\cdots \Rightarrow$$

$$\underbrace{U_m \cdots U_1 A}_{R} x = \underbrace{U_m \cdots U_1}_{Q^T} b$$

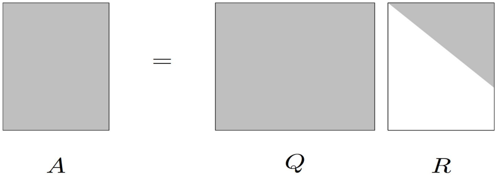
$$A_k = [a_1^{(k)} \ \dots \ a_k^{(k)} \ \dots \ a_n^{(k)}] = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1,k-1} & r_{1k} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2,k-1} & r_{2k} & \dots & r_{2n} \\ & & \ddots & \vdots & \vdots & & \vdots \\ & & & r_{k-1,k-1} & r_{k-1,k} & \dots & r_{k-1,n} \\ & & & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & 0 & & & a_{k+1,k}^{(k)} & \dots & a_{k+1,n}^{(k)} \\ & & & & \vdots & & \vdots \\ & & & & a_{mk}^{(k)} & \dots & a_{mn}^{(k)} \end{bmatrix}.$$

Factorizare QR cu reflectori Householder:

$$Ax \overset{CMMP}{=} b \Rightarrow \underbrace{U_1 A}_{A_1} x \Rightarrow U_2 U_1 A x \Rightarrow$$

$$\cdots \Rightarrow$$

$$\underbrace{U_m \cdots U_1 A}_{R} x = \underbrace{U_m \cdots U_1}_{Q^T} b$$



https://www.ics.uci.edu/ xhx/courses/CS206/NLA-QR.pdf

Factorizare QR cu reflectori Householder:

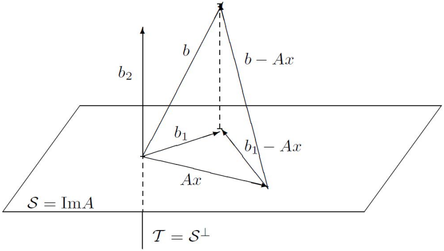
$$Ax \overset{CMMP}{\approx} b \Rightarrow Q^T A = R \Rightarrow A = QR = Q \begin{bmatrix} R' \\ 0 \end{bmatrix}$$

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

Calcul pseudosoluţie:

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

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



Factorizare QR cu reflectori Householder:

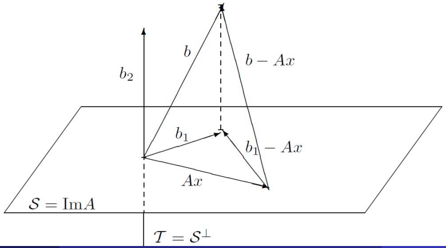
$$Ax \overset{CMMP}{\approx} b \Rightarrow Q^T A = R \Rightarrow A = QR = Q \begin{bmatrix} R' \\ 0 \end{bmatrix}$$

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

Calcul pseudosoluţie:

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

$$R'x^* = d', \text{unde } Q^T b = \begin{bmatrix} d' \\ d'' \end{bmatrix}$$



Factorizare QR cu reflectori Householder:

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

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

Calcul pseudosoluţie:

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

$$R' x^* = d', \text{unde } Q^T b = \begin{bmatrix} d' \\ d'' \end{bmatrix}$$

Soluţia CMMP satisface:

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

Factorizare QR cu reflectori Householder:

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

Algoritm:

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

Factorizare QR cu reflectori Householder:

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

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:

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

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: $Rx = d'$

Referinţe

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