

Calcul numeric

Subspații. Matrice. Sisteme liniare pătratice.

Paul Irofti
Cristian Rusu
Andrei Pătrașcu

Departmentul de Informatică
Facultatea de Matematică și Informatică
Universitatea din București
Email: `prenume.nume@fmi.unibuc.ro`

Curs:

- ▶ Paul Irofti
- ▶ Cristian Rusu
- ▶ Andrei Pătrașcu

Laborator:

- ▶ Andra Băltoiu



Punctaj:

- ▶ Examen final scris - 50%
- ▶ Teme laborator - 40%
- ▶ Activitate (și prezență) seminar - 10%

Detalii examen:

- ▶ Dacă vor permite condițiile intenționăm să susținem examenul final fizic, altfel se va alege o variantă online
- ▶ **Promovare:** nota minim 5 la examen final, nota minim 5 la temele de laborator



- ▶ Van Loan, Charles F., and G. Golub. *Matrix computations (Johns Hopkins studies in mathematical sciences)*. (1996).
- ▶ Deisenroth, Marc Peter, A. Aldo Faisal, and Cheng Soon Ong. *Mathematics for machine learning*. Cambridge University Press, 2020.
- ▶ Friedman, Jerome H., *The elements of statistical learning: Data mining, inference, and prediction*. Springer, 2017.
- ▶ Călin, Ovidiu. *Deep learning architectures*. Springer International Publishing, 2020.
- ▶ Dumitrescu, Bogdan, and Paul Irofti. *Dictionary learning algorithms and applications*. Springer, 2018.
- ▶ Majoritatea vor fi disponibile pe platformele Teams/Moodle



- A. Sisteme liniare de ecuații. Problema celor mai mici pătrate
- B. Valori proprii
- C. Valori singulare
- D. Metode numerice pentru învățarea automată
- E. Metode numerice pentru învățarea dicționarelor



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



Vom lucra cu entități (vectori și matrice) construite cu numere reale

► scalar: 1.2

► vector: $\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$

► matrice: $\begin{bmatrix} 1 & 0 & 2 \\ 1 & 0 & 1 \\ 2 & 2 & 1 \end{bmatrix}$



Definiție. Un vector real x de dimensiune n este o colecție de n numere reale dispuse ordonat într-o coloană.

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

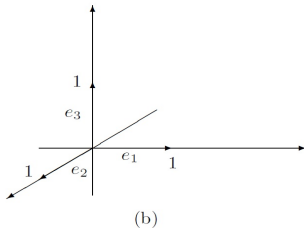
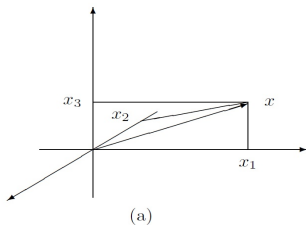


Fig. 1.1: (a) Un vector în \mathbb{R}^3 și coordonatele sale; (b) vectorii unitate în \mathbb{R}^3



► Suma: $z = x + y = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{bmatrix}$

► Înmulțire cu un scalar: $z = \alpha x = \begin{bmatrix} \alpha x_1 \\ \alpha x_2 \\ \vdots \\ \alpha x_n \end{bmatrix}$

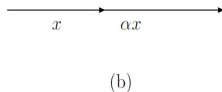
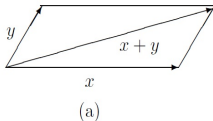


Fig. 1.2: (a) Suma a doi vectori în \mathbb{R}^2 ; (b) Produsul cu un scalar



Considerând vectorii $X = \{x_1, x_2, \dots, x_p\}$, atunci vectorul

$$Z = \alpha_1 x_1 + \dots + \alpha_p x_p = \sum_{i=1}^p \alpha_i x_i$$

se numeste combinație liniară a vectorilor din X cu coeficienții $\alpha_1, \dots, \alpha_p \in \mathbb{R}$.

Exemplu: $X = \left\{ \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}, \alpha = [1 \ 1/2]$

$$z = 1 \cdot \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} + 1/2 \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$



Considerând vectorii $X = \{x_1, x_2, \dots, x_p\}$, atunci vectorul

$$z = \alpha_1 x_1 + \dots + \alpha_p x_p = \sum_{i=1}^p \alpha_i x_i$$

se numește combinație liniară a vectorilor din X cu coeficienții $\alpha_1, \dots, \alpha_p \in \mathbb{R}$.

Exemplu: $X = \left\{ \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}, \alpha = [1 \ 1/2]$

$$z = 1 \cdot \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} + 1/2 \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Observați: dacă $\alpha = [1 \ -1/2]$ atunci $z = 0$



Vectorii $X = \{x_1, x_2, \dots, x_p\}$ se numesc **liniar dependenți** dacă $\exists \hat{\alpha} \neq 0 \in \mathbb{R}^p$ a.i.

$$z = \hat{\alpha}_1 x_1 + \dots + \hat{\alpha}_p x_p = \sum_{i=1}^p \hat{\alpha}_i x_i = 0$$

Altfel, se numesc **liniar independenți**.

Exemplu liniar dependenți: $X = \left\{ \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}, \hat{\alpha} = [1 \quad -1/2]$

$$z = 1 \cdot \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} - 1/2 \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Exemplu liniar independenți: $X = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}, \alpha = [\alpha_1 \quad \alpha_2]^T$

$$z = \alpha_1 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \alpha_2 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} \neq 0$$

Arătați că $\alpha(z)$ este unic pentru X liniar independenți! (exercițiu)



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



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

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

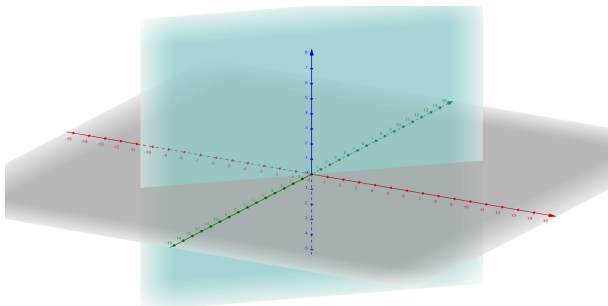


Subspațiu Liniar

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 $S = \{x \in \mathbb{R}\}$

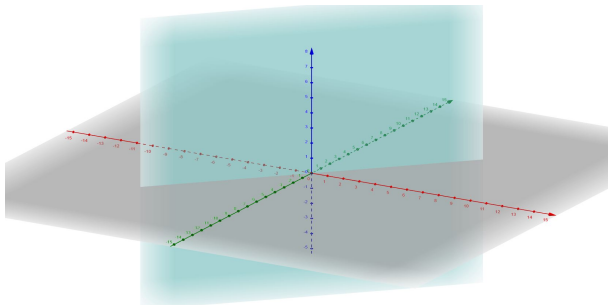


Subspațiu Liniar

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

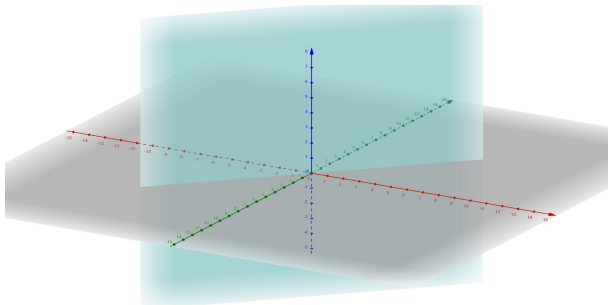


Subspațiu Liniar

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

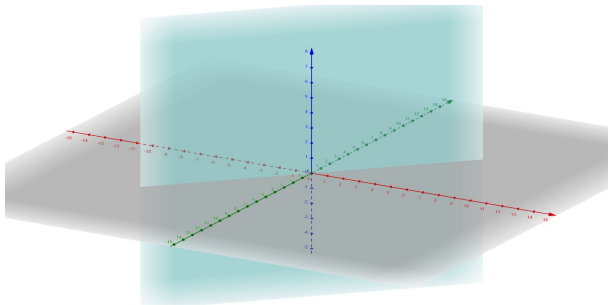


Subspațiu Liniar

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 $A \in \mathbb{R}^{m \times n}$



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 $X = \{x_1, x_2, \dots, x_p\}$ genereaza 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:

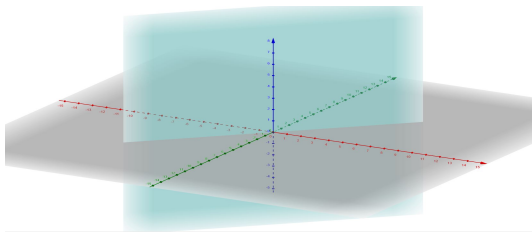
- ▶ $S = \left\{ x \in \mathbb{R}^n : x = \alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \alpha \in \mathbb{R} \right\}$ sau $S = \text{span} \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$
- ▶ $S = \{ x \in \mathbb{R}^n : x = \alpha_1 b_1 + \cdots + \alpha_p b_p, \alpha \in \mathbb{R}^p \}$ sau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



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

Exemple:

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

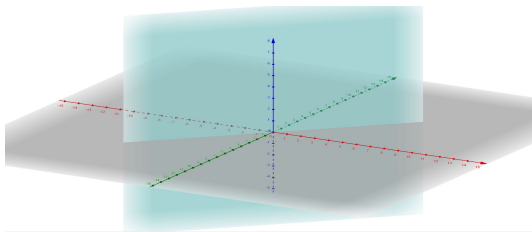


Dimensiunea subspațiului $S \subseteq \mathbb{R}^n$ = numărul de vectori din baza (nr. maxim de vectori linear independenți)

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**

Support Vector Machine:

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & \|w\|_2^2 \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \xi \geq 0 \end{aligned}$$

Regresie liniară:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \|x\|_2^2 \\ \text{s.t.} \quad & Ax = b \end{aligned}$$

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2$$



Produs scalar euclidian: $\langle x, y \rangle = x^T y = \sum_{i=1}^n x_i y_i$

- ▶ dacă x, y au norma unitate atunci $x^T y = \cos(\theta)$, unde θ este unghiul format de x și y

Norme $\| \cdot \|$: functii care satisfac urmatoarele conditii

- ▶ pozitivitate: $\|x\| > 0, \forall x \in \mathbb{R}^n, x \neq 0$



Produs scalar euclidian: $\langle x, y \rangle = x^T y = \sum_{i=1}^n x_i y_i$

- ▶ dacă x, y au norma unitate atunci $x^T y = \cos(\theta)$, unde θ este unghiul format de x și y
- ▶ în felul acesta, capătă sensul unei măsuri de similaritate (în opoziție cu distanța)

Norme $\| \cdot \|$: funcții care satisfac următoarele condiții

- ▶ pozitivitate: $\|x\| > 0, \forall x \in \mathbb{R}^n, x \neq 0$
- ▶ omogenitate: $\|\alpha x\| = |\alpha| \|x\|, \forall x \in \mathbb{R}^n, \alpha \in \mathbb{R}$



Produs scalar euclidian: $\langle x, y \rangle = x^T y = \sum_{i=1}^n x_i y_i$

- ▶ dacă x, y au norma unitate atunci $x^T y = \cos(\theta)$, unde θ este unghiul format de x și y
- ▶ în felul acesta, capătă sensul unei măsuri de similaritate (în opoziție cu distanța)

Norme $\| \cdot \|$: funcții care satisfac următoarele condiții

- ▶ pozitivitate: $\|x\| > 0, \forall x \in \mathbb{R}^n, x \neq 0$
- ▶ omogenitate: $\|\alpha x\| = |\alpha| \|x\|, \forall x \in \mathbb{R}^n, \alpha \in \mathbb{R}$
- ▶ inegalitatea triunghiului: $\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in \mathbb{R}^n$



Produs scalar euclidian: $\langle x, y \rangle = x^T y = \sum_{i=1}^n x_i y_i$

- ▶ dacă x, y au norma unitate atunci $x^T y = \cos(\theta)$, unde θ este unghiul format de x și y

Norme $\| \cdot \|$: functii care satisfac urmatoarele conditii

- ▶ pozitivitate: $\|x\| > 0, \forall x \in \mathbb{R}^n, x \neq 0$
- ▶ omogenitate: $\|\alpha x\| = |\alpha| \|x\|, \forall x \in \mathbb{R}^n, \alpha \in \mathbb{R}$
- ▶ inegalitatea triunghiului: $\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in \mathbb{R}^n$

$$\text{Exemple : } \|x\|_2 = \sqrt{x^T x}, \quad \|x\|_p := \left(\sum_i |x_i|^p \right)^{1/p}$$



Produs scalar euclidian: $\langle x, y \rangle = x^T y = \sum_{i=1}^n x_i y_i$

- ▶ dacă x, y au norma unitate atunci $x^T y = \cos(\theta)$, unde θ este unghiul format de x și y
- ▶ în felul acesta, capătă sensul unei măsuri de similaritate (în opoziție cu distanța)

Norme $\| \cdot \|$: funcții care satisfac următoarele condiții

- ▶ pozitivitate: $\|x\| > 0, \forall x \in \mathbb{R}^n, x \neq 0$
- ▶ omogenitate: $\|\alpha x\| = |\alpha| \|x\|, \forall x \in \mathbb{R}^n, \alpha \in \mathbb{R}$
- ▶ inegalitatea triunghiului: $\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in \mathbb{R}^n$

$$\text{Exemple: } \|x\|_2 = \sqrt{x^T x}, \quad \|x\|_p := \left(\sum_i |x_i|^p \right)^{1/p}$$

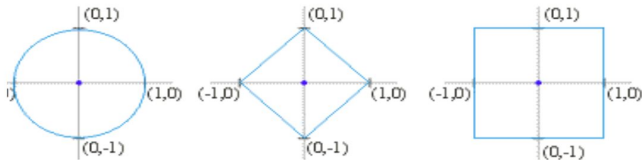


Produs scalar euclidian: $\langle x, y \rangle = x^T y = \sum_{i=1}^n x_i y_i$

- ▶ dacă x, y au norma unitate atunci $x^T y = \cos(\theta)$, unde θ este unghiul format de x și y

Norme $\| \cdot \|$: exemple

- ▶ norma 2: $\|x\|_2 = \sqrt{\sum_i x_i^2}$ (fig. stanga)

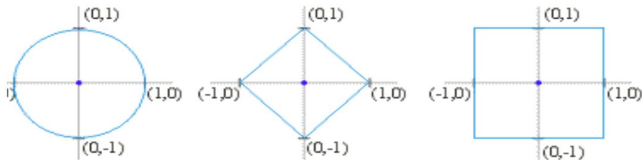


Produs scalar euclidian: $\langle x, y \rangle = x^T y = \sum_{i=1}^n x_i y_i$

- ▶ dacă x, y au norma unitate atunci $x^T y = \cos(\theta)$, unde θ este unghiul format de x și y
- ▶ în felul acesta, capătă sensul unei măsuri de similaritate (în opoziție cu distanța)

Norme $\| \cdot \|$: exemple

- ▶ norma 2: $\|x\|_2 = \sqrt{\sum_i x_i^2}$ (fig. stanga)
- ▶ norma 1: $\|x\|_1 = \sum_i |x_i|$ (fig. centru)

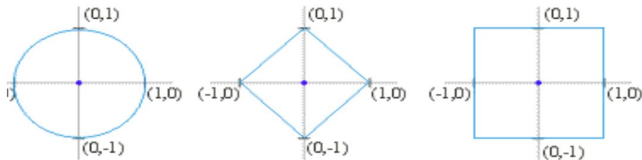


Produs scalar euclidian: $\langle x, y \rangle = x^T y = \sum_{i=1}^n x_i y_i$

- ▶ dacă x, y au norma unitate atunci $x^T y = \cos(\theta)$, unde θ este unghiul format de x și y
- ▶ în felul acesta, capătă sensul unei măsuri de similaritate (în opoziție cu distanța)

Norme $\| \cdot \|$: exemple

- ▶ norma 2: $\|x\|_2 = \sqrt{\sum_i x_i^2}$ (fig. stanga)
- ▶ norma 1: $\|x\|_1 = \sum_i |x_i|$ (fig. centru)
- ▶ norma ∞ : $\|x\|_\infty = \max_i |x_i|$ (fig. dreapta)



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



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 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 :

$$\text{Im}(A) = \{y \in \mathbb{R}^m : \exists x \in \mathbb{R}^n \text{ astfel incat } y = Ax\}$$

- ▶ $\text{rang}(A) = \dim(\text{Im}(A))$

Exemplu!



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

- ▶ Imaginea matricii A :

$$\text{Im}(A) = \{y \in \mathbb{R}^m : \exists x \in \mathbb{R}^n \text{ astfel incat } y = Ax\}$$

- ▶ $\text{rang}(A) = \dim(\text{Im}(A))$

- ▶ Nucleul matricii A : $\text{Ker}(A) = \{x \in \mathbb{R}^n : Ax = 0\}$

Exemplu!



O matrice $A \in \mathbb{R}^{m \times n}$ genereaza subspatiile:

- ▶ Imaginea matricii A :

$$\text{Im}(A) = \{y \in \mathbb{R}^m : \exists x \in \mathbb{R}^n \text{ astfel incat } y = Ax\}$$

- ▶ $\text{rang}(A) = \dim(\text{Im}(A))$

- ▶ Nucleul matricii A : $\text{Ker}(A) = \{x \in \mathbb{R}^n : Ax = 0\}$

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

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



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



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

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

- ▶ 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. $y_i = y_i + a_{ij}x_j$



$$C := AB = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1l} \\ a_{21} & a_{22} & \cdots & a_{2l} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{ml} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{l1} & b_{l2} & \cdots & b_{ln} \end{bmatrix}$$

$$\text{Observam : } c_{ij} = a^i b_j$$

- ▶ $A(BC) = (AB)C$
- ▶ $A(B + C) = AB + AC$
- ▶ $(AB)^T = B^T A^T$



Produs matrice-matrice: Forma 1

$$C := \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1l} \\ a_{21} & a_{22} & \cdots & a_{2l} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{ml} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{l1} & b_{l2} & \cdots & b_{ln} \end{bmatrix}$$
$$= [a_1 \cdots a_l] \begin{bmatrix} b^1 \\ \vdots \\ b^l \end{bmatrix}$$

$$\text{Forma 1 : } C = AB = \sum_{k=1}^l a_k b^k$$



$$C := \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1l} \\ a_{21} & a_{22} & \cdots & a_{2l} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{ml} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{l1} & b_{l2} & \cdots & b_{ln} \end{bmatrix}$$

$$\text{Forma 2 : } C = AB = \begin{bmatrix} \underbrace{Ab_1}_{c_1} & \underbrace{Ab_2}_{c_2} & \cdots & \underbrace{Ab_n}_{c_n} \end{bmatrix}$$



$$C := \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1l} \\ a_{21} & a_{22} & \cdots & a_{2l} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{ml} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{l1} & b_{l2} & \cdots & b_{ln} \end{bmatrix}$$

$$\text{Forma 3 : } C = AB = \begin{bmatrix} a^1 B \\ a^2 B \\ \cdots \\ a^m B \end{bmatrix} := \begin{bmatrix} c^1 \\ c^2 \\ \cdots \\ c^m \end{bmatrix}$$



$$C := \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1l} \\ a_{21} & a_{22} & \cdots & a_{2l} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{ml} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{l1} & b_{l2} & \cdots & b_{ln} \end{bmatrix}$$

Algorithm $MM(A, B)$

1. Pentru $i = 1 : m$

1. Pentru $j = 1 : n$

1. Pentru $k = 1 : l$

1. $c_{ij} = c_{ij} + a_{ik}b_{kj}$



- ▶ 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 :

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

- ▶ T tridiagonală:

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



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



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

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

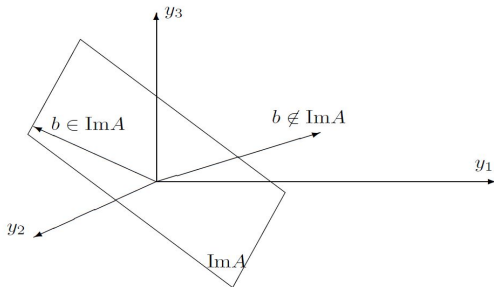


Sistemul

$$Ax = b$$

- ▶ Subdeterminat: $m < n$, posibil o infinitate de 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

$$\begin{cases} x_1 + x_2 = 2 \\ x_2 + x_3 = 1. \end{cases}$$

► O soluție particulară: $x = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$

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



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



Exista instanțe foarte simple de SL pătrate: sistem triunghiular!
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 pătratice: 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. $x_i = x_i / l_{ii}$

- 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: $O(n^2)$ (comparativ cazul general $O(n^3)$)



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. $x_i = x_i / l_{ii}$

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

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



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



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

$$M_k = I_n - m_k \mathbf{e}_k^T$$

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 μ_{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 μ_{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 μ_{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

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

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

.....

Pas k: $A_k = M_k \cdots M_2 M_1 A$ 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 $\mu_{ik} = \frac{a_{ik}^k}{a_{kk}^k}$
- ▶ Algoritmul produce $U = A_n = \underbrace{M_{n-1} M_{n-2} \cdots M_1}_M A$
- ▶ Matricea M este inferior triunghiulară (de ce?)
- ▶ Important: 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?)
- ▶ Important: 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?)



Eliminare gaussiană

Pseudocodul algoritmului EG:

Algorithm 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 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: $Ux = Mb$ 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: $Ux = Mb$ se rezolvă cu UTRIS
- ▶ Complexitate totală: $\sum_{k=1}^{n-1} (n - k + 2(n - k)^2) \approx \frac{2n^3}{3} = O(n^3)$



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: $Ux = Mb$ se rezolvă cu UTRIS
- ▶ Complexitate totală: $\sum_{k=1}^{n-1} (n - k + 2(n - k)^2) \approx \frac{2n^3}{3} = O(n^3)$
- ▶ 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: $Ux = Mb$ se rezolvă cu UTRIS
- ▶ Complexitate totală: $\sum_{k=1}^{n-1} (n - k + 2(n - k)^2) \approx \frac{2n^3}{3} = O(n^3)$
- ▶ 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 μ_{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 $i_k : |a_{i_k k}| = \max_{i=k:n} |a_{ik}|$



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.

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

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 $(M_k A)_{ik} = 0, i = k + 1 : n$
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 :

$$\sum_{k=1}^{n-1} (n - k + 1) \approx \frac{n^2}{2} = O(n^2)$$

