Calcul numeric

Subspații. Matrice. Sisteme liniare pătratice.

Paul Irofti
Cristian Rusu
Andrei Pătraşcu

Departmentul de Informatică
Facultatea de Matematică și Informatică
Universitatea din București

Cuprins - Curs 1

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

Vom lucra cu entităţi (vectori şi matrice) construite cu numere reale

- scalar: 1.2
- vector: $\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$
- matrice: $\begin{bmatrix} 1 & 0 & 2 \\ 1 & 0 & 1 \\ 2 & 2 & 1 \end{bmatrix}$

Definiţie. Un vector real *x* de dimensiune *n* este o colecţie de *n* numere reale dispuse ordonat într-o coloană.

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

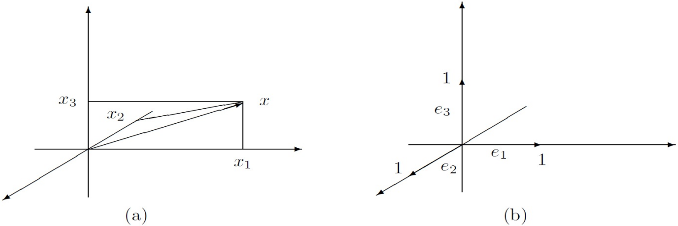

Fig. 1.1: (a) Un vector în $\mathbb{R}^3$ şi coordonatele sale; (b) vectorii unitate în $\mathbb{R}^3$

- Suma: $z = x + y = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{bmatrix}$
- Înmulţire cu un scalar: $z = \; x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

(a) (b)

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

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

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

se numeste combinaţie liniară a vectorilor din X cu coeficienţii $\alpha_1, \cdots, \alpha_p \in \mathbb{R}$.

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

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

Liniar dependenta

Considerând vectorii $X = f x_1; x_2; \quad ; x_p g$, atunci vectorul

$$z = \ {}_1 x_1 + \qquad {}_p x_p = \sum_{i=1}^{p} \ {}_i x_i$$

se numeste combinaţie liniară a vectorilor din X cu coeficienţii ${}_1; \quad ; {}_p\ 2\ \mathrm{R}$.

Exemplu: $X = f \begin{matrix} 1=2 \\ 1=2 \end{matrix} ; \begin{matrix} 1 \\ 1 \end{matrix} g; \quad = [1\ 1=2]$

$$z = 1 \quad \begin{matrix} 1=2 \\ 1=2 \end{matrix} + 1=2 \quad \begin{matrix} 1 \\ 1 \end{matrix} = \begin{matrix} 1 \\ 1 \end{matrix}$$

Observati: dacă $= [1 \quad 1=2]$ atunci $z = 0$

Vectorii $X = \{x_1; x_2; \ldots; x_p\}$ se numesc **liniar dependenţi** dacă $\exists \lambda \neq 0 \in \mathbb{R}^p$ a.i.

$$z = \lambda_1 x_1 + \cdots + \lambda_p x_p = \sum_{i=1}^{p} \lambda_i x_i = 0$$

Altfel, se numesc **liniar independenţi**.

Exemplu liniar dependenţi: $X = \left\{ \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}; \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}; \lambda = [1 \quad 1/2]$

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

Exemplu liniar independenţi: $X = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}; \quad = [\ _1 \quad _2]^T$

$$z = \ _1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \ _2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} _1 \\ _2 \end{bmatrix} \neq 0$$

Arătaţi că $\ (z)$ este unic pentru X liniar independenţi! (exerciţiu)

Cuprins

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

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

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

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

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) $\lambda x \in S, \forall x \in S, \lambda \in \mathbb{R}$

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) $\lambda x \in S, \forall x \in S, \lambda \in \mathbb{R}$

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

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) $\lambda x \in S, \forall x \in S, \lambda \in \mathbb{R}$

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 $S = \{x \in \mathbb{R}^2 : x_1 + 2x_2 + x_3 = 0, x_1 - x_2 = 0\}$
- in $\mathbb{R}^n$: subspaţiu $S = \{x \in \mathbb{R}^n : Ax = 0\}$, unde $A \in \mathbb{R}^{m \times n}$

Geogebra.org

Subspaţiu Liniar

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

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

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 $S = \{x \in \mathbb{R}^2 : x_1 + 2x_2 + x_3 = 0, x_1 - x_2 = 0\}$
- 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 $X = \{x_1, x_2, \cdots, x_p\}$ genereaza 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 genereaza 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 genereaza 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 $S = span\{b_1, \ldots, b_p\}$

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 genereaza 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 $S = span\,\{b_1, \dots, b_p\}$
- $\{e_1, \dots, e_n\}$ baza (canonica) pentru spaţiul $\mathbb{R}^n$

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

Exemple:

- $S = \; x \in \mathbb{R}^n : x = \; \begin{matrix} 1 \\ 1 \end{matrix} \; ; \; \in \mathbb{R}$ => Dimensiune $Dim(S) = 1$

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

Exemple:

- $S = \quad x \in \mathbb{R}^n : x = \begin{matrix} 1 \\ 1 \end{matrix} ; \quad \in \mathbb{R}$ => Dimensiune $Dim(S) = 1$
- $S = \{ x \in \mathbb{R}^n : x = \;_1 b_1 + \qquad _p b_p ; \quad \in \mathbb{R}^p \} => Dim(S) = p$

Norme

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**

Support Vector Machine:

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

Regresie liniară:

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

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

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 si y

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

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

Norme. Produs Scalar

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 si y
- in felul acesta, capata sensul unei măsuri de similaritate (în opoziţie cu distanţa)

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

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

Norme. Produs Scalar

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

- daca x, y au norma unitate atunci $x^T y = \cos(\theta)$; unde θ este unghiul format de x si y
- in felul acesta, capata sensul unei masuri de similaritate (în opoziţie cu distanţa)

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: $\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in \mathbb{R}^n$

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

- daca x, y au norma unitate atunci $x^T y = \cos(\theta)$, unde θ este unghiul format de x si 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: $\langle x, y\rangle = x^T y = \sum_{i=1}^{n} x_i y_i$

- daca x, y au norma unitate atunci $x^T y = \cos(\theta)$, unde θ este unghiul format de x si y
- in felul acesta, capata sensul unei masuri de similaritate (in opozitie cu distanta)

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

- daca x, y au norma unitate atunci $x^T y = \cos(\ )$, unde este unghiul format de x si y

Norme $\| \ \|$: exemple

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

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

- daca x, y au norma unitate atunci $x^T y = \cos(\theta)$, unde θ este unghiul format de x si y
- in felul acesta, capata sensul unei masuri de similaritate (in opozitie cu distanta)

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)

w3.cs.jmu.edu

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

- daca x, y au norma unitate atunci $x^T y = \cos(\theta)$; unde θ este unghiul format de x si y
- in felul acesta, capata sensul unei masuri de similaritate (in opozitie cu distanta)

Norme $\| \cdot \|$: exemple

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

Cuprins

- Introducere. Vectori. Operaţii elementare
- Subspaţii liniare. Produs scalar. Norme
- Matrice. Operaţii elementare. Propriet aţi
- Sisteme de ecuaţii liniare patratice
- Algoritmi de rezolvare a SL patratice
- Sisteme speciale

Matrice

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

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

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

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

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

- daca $m = n$ 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:

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

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

Matrice

O matrice $A\ 2\ R^{m\ \ n}$ reprezinta un tablou bidimensional de numere reale de forma:

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

- daca $m = n$ atunci A este matrice patrata
- A patrata) diagonala principala este mulţimea pozitiilor pentru care $i = j$
- $C = A + B$, $c_{ij} = a_{ij} + b_{ij}$
- $C = \ A$, $c_{ij} = \ a_{ij}$

Matrice

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

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

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

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

- Imaginea matricii A:
  $\mathrm{Im}(A) = \{y \in \mathbb{R}^m : \exists x \in \mathbb{R}^n \text{ astfel incat } y = Ax\}$
  - rang(A) = dim(Im(A))

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

- Imaginea matricii A:
  $\mathrm{Im}(A) = \{y \in \mathbb{R}^m : \exists x \in \mathbb{R}^n \text{ astfel incat } y = Ax\}$
  - rang(A) = dim(Im(A))
- Nucleul matricii A: $\mathrm{Ker}(A) = \{x \in \mathbb{R}^n : Ax = 0\}$

Matrice. Subspatii asociate

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

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

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

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

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

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

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

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

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

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

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

Algorithm GAXPY(A, x, y)

1: Pentru $i = 1 : m$

    1: Pentru $j = 1 : n$

        1: $y_i = y_i + a_{ij} x_j$

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

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

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

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

$$= [\, \color{blue}{a_1} \; \cdots \; a_l \,] \begin{bmatrix} \color{red}{b^1} \\ \vdots \\ b^l \end{bmatrix}$$

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

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

$$\text{Forma 2} : C = AB = \Big[\underbrace{A\color{blue}{b_1}}_{c_1} \quad \underbrace{Ab_2}_{c_2} \quad \cdots \quad \underbrace{Ab_n}_{c_n}\Big]$$

Produs matrice-matrice: Forma 3

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

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

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

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 $j < i$:

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

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

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

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

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

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

$$\begin{bmatrix} h_{11} & h_{12} & 0 & & 0 \\ h_{21} & h_{22} & h_{23} & & 0 \\ \vdots & \vdots & & \vdots & \\ h_{n-1,1} & h_{n-1,2} & h_{n-1,3} & & h_{n-1,n} \\ h_{n1} & h_{n2} & h_{n3} & & h_{nn} \end{bmatrix}$$

Matrici structurate

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

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

- T tridiagonala:

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

Cuprins

- Introducere. Vectori. Operaţii elementare
- Subspaţii liniare. Produs scalar. Norme
- Matrice. Operaţii elementare. Proprietaţi
- Sisteme de ecuaţii liniare p atratice
- Algoritmi de rezolvare a SL patratice
- Sisteme speciale

Un sistem de m ecuaţii cu n necunoscute are forma:

Forma matriceala

$$Ax = b;$$

unde A este matricea coe cienţilor, x vector necunoscutelor, b termen liber

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

Sistemul

$$Ax = b$$

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

Teorem a. Sistemul $Ax = b$ are soluţie daca şi numai daca b 2 $\mathrm{Im}(A)$.

Sisteme de ecuaţii liniare

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

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

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

Teorema. Daca $A \in \mathbb{R}^{n \times n}$ atunci sistemul are soluţie unica daca şi numai daca:

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

- Nu este o formula adecvata calculului numeric (vrem sa evitam A^{-1})

Teorema. Daca $A \in \mathbb{R}^{n \times n}$ atunci sistemul are soluţie unica daca şi numai daca:

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

- Nu este o formula adecvata calculului numeric (vrem sa evitam A^{-1})
- In general, existenţa unei soluţii se determina greu

Teoremă. Daca $A \in \mathbb{R}^{n \times n}$ atunci sistemul are soluţie unica daca şi numai daca:

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

- Nu este o formula adecvata calculului numeric (vrem sa evitam A^{-1})
- In general, existenţa unei soluţii se determina greu
- In particular, existenţa unei soluţii se detecteaza imediat in cazul matricilor triunghiulare

Exista instanţe foarte simple de SL patratice: 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 patratice: sistem triunghiulare!
Distingem: $A = L$ inferior triunghiulara $l_{ij} = 0$ pentru $j > i$:

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

- Observam ca: $x_1 = \frac{b_1}{l_{11}}$

Exista instanţe foarte simple de SL patratice: sistem triunghiulare!
Distingem: $A = L$ inferior triunghiulara $l_{ij} = 0$ pentru $j > i$:

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

- Observam ca: $x_1 = \frac{b_1}{l_{11}}$
- Daca se cunosc $x_1; \cdots; x_{i-1}$ atunci :

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

Algoritm LTRIS

Algorithm LTRIS$(L; b)$

1: $x := b$

2: Pentru $i = 1 : n$

  1: Pentru $j = 1 : i \quad 1$

    1. $x_i = x_i \quad l_{ij} x_j$

  2: $x_i = x_i = l_{ii}$

- La ecare pas al buclei necesit a $2(i \quad 1)$ opi

Algoritm LTRIS

Algorithm LTRIS$(L; b)$
1: $x := b$
2: Pentru $i = 1 : n$
    1: Pentru $j = 1 : i \quad 1$
        1. $x_i = x_i \quad l_{ij} x_j$
    2: $x_i = x_i = l_{ii}$

- La ecare pas al buclei necesit a $2(i \quad 1)$ opi
- Complexitate: $O(n^2)$ (comparativ cazul general $O(n^3)$)

Algoritm LTRIS

Algorithm LTRIS$(L; b)$

1: $x := b$

2: Pentru $i = 1 : n$

   1: Pentru $j = 1 : i \quad 1$

      1. $x_i = x_i \quad l_{ij} x_j$

   2: $x_i = x_i = l_{ii}$

- La ecare pas al buclei necesit a $2(i \quad 1)$ opi
- Complexitate: $O(n^2)$ (comparativ cazul general $O(n^3)$)
- Exercitiu: Scrieti pseudocodul alg. UTRIS

Algorithm LTRIS$(L; b)$

1: $x := b$

2: Pentru $i = 1 : n$

  1: Pentru $j = 1 : i \quad 1$

    1. $x_i = x_i \quad l_{ij} x_j$

  2: $x_i = x_i = l_{ii}$

- La ecare pas al buclei necesit a $2(i \quad 1)$ opi
- Complexitate: $O(n^2)$ (comparativ cazul general $O(n^3)$)
- Exercitiu: Scrieti pseudocodul alg. UTRIS
- Idee: Putem reduce un sistem general la unul triunghiular?

Cuprins

- Introducere. Vectori. Operaţii elementare
- Subspaţii liniare. Produs scalar. Norme
- Matrice. Operaţii elementare. Proprietaţi
- Sisteme de ecuaţii liniare patratice
- Algoritmi de rezolvare a SL p   atratice
- Sisteme speciale

De niţie. Transformare inferior triunghiulara elementara (ITE) de ordin n şi indice k are forma:

$$M_k = I_n \quad m_k e_k^T$$

unde

$$m_k = \quad 0 \quad 0 \quad \quad 0 \quad {}_{k+1;k} \quad \quad {}_{n;k}$$

are primele k elemente nenule.

Proprietaţi ale matricilor de transformare ITE:

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

Transformari elementare

Proprietaţi ale matricilor de transformare ITE:

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

Proprietaţi ale matricilor de transformare ITE:

- M_k este inversabila 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

Proprietaţi ale matricilor de transformare ITE:

- M_k este inversabila 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
- Daca alegem valori potrivite pentru multiplicatorii μ_{ik}, putem obţine $(M_k x)_i := \begin{cases} x_i & \text{pentru } i = 1 : k \\ 0 & \text{pentru } i = k+1 : n \end{cases}$

Proprietaţi ale matricilor de transformare ITE:

- M_k este inversabila 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
- Daca alegem valori potrivite pentru multiplicatorii μ_{ik}, putem obţine $(M_k x)_i := \begin{cases} x_i & \text{pentru } i = 1 : k \\ 0 & \text{pentru } i = k+1 : n \end{cases}$

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

Proprietaţi ale matricilor de transformare ITE:

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

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

Eliminare gaussiana

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

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

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

Eliminare gaussiana

Algoritm EG(A)

1: Pentru $k = 1 : n \quad 1$

1: Se calculeaz a matricea M_k (adica multiplicatorii $_{ik}$; $i = k + 1 : n$), astfel ˆncât $(M_k A)_i = 0$; pentru $i = k + 1 : n$

2: $A = M_k A$

- Multiplicatorii necesari $_{ik} = \frac{a^k_{ik}}{a^k_{kk}}$

Algoritm $EG(A)$

1: **Pentru** $k = 1 : n \quad 1$

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

 2: $A = M_k A$

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

Eliminare gaussiană

Algoritm $EG(A)$

1: **Pentru** $k = 1 : n \quad 1$

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

   2: $A = M_k A$

- Multiplicatorii necesari $_{ik} = \frac{a_{ik}^k}{a_{kk}^k}$
- Algoritmul produce $U = A_n = \underbrace{M_n \; {}_1 M_n \; {}_2 \quad M_1}_{M} A$
- Matricea M este inferior triunghiulară (de ce?)

Algoritm $EG(A)$

1: **Pentru** $k = 1 : n \quad 1$

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

  2: $A = M_k A$

- Multiplicatorii necesari $_{ik} = \frac{a^k_{ik}}{a^k_{kk}}$
- Algoritmul produce $U = A_n = \underbrace{M_{n \quad 1} M_{n \quad 2} \quad M_1}_{M} A$
- Matricea M este inferior triunghiulară (de ce?)
- Important: 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** $k = 1 : n \quad 1$

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

  2: $A = M_k A$

- Multiplicatorii necesari $_{ik} = \frac{a^k_{ik}}{a^k_{kk}}$
- Algoritmul produce $U = A_n = \underbrace{M_{n\quad 1} M_{n\quad 2} \quad M_1}_{M} A$
- Matricea M este inferior triunghiulară (de ce?)
- Important: 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?)

Pseudocodul algoritmului EG:

Algoritm $G(A)$

1: **Pentru** $k = 1 : n \quad 1$

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

    1: $a_{ik} \qquad {}_{ik} = \frac{a_{ik}}{a_{kk}}$

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

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

      1: $a_{ij} \qquad a_{ij} \qquad {}_{ik} a_{kj}$

Multiplicatorii ${}_{ik}$ se pot memora in triunghiul inferior al matricii A

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

După pasul k — În final

Eliminare gaussiană

Pseudocodul algoritmului EG:

Algoritm $G(A)$

1: **Pentru** $k = 1 : n \quad 1$

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

    1: $a_{ik} \qquad {}_{ik} = \frac{a_{ik}}{a_{kk}}$

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

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

      1: $a_{ij} \quad a_{ij} \quad {}_{ik} a_{kj}$

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

Eliminare gaussiană

Pseudocodul algoritmului EG:

Algoritm $G(A)$

1: **Pentru** $k = 1 : n \quad 1$

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

    1: $a_{ik} \qquad {}_{ik} = \frac{a_{ik}}{a_{kk}}$

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

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

      1: $a_{ij} \quad a_{ij} \quad {}_{ik} a_{kj}$

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

Pseudocodul algoritmului EG:

Algoritm $G(A)$

1: **Pentru** $k = 1 : n \quad 1$

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

    1: $a_{ik} \quad _{ik} = \frac{a_{ik}}{a_{kk}}$

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

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

      1: $a_{ij} \quad a_{ij} \quad _{ik} a_{kj}$

- În final: $Ux = Mb$ se rezolvă cu UTRIS
- Complexitate totală: $\sum_{k=1}^{n \quad 1} (n \quad k + 2(n \quad k)^2) \quad \frac{2n^3}{3} = O(n^3)$
- Probleme: Ce se întâmplă dacă o submatrice lider principală este singulară?

Eliminare gaussiană

Pseudocodul algoritmului EG:

Algoritm $G(A)$

1: **Pentru** $k = 1 : n \quad 1$

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

    1: $a_{ik} \qquad {}_{ik} = \frac{a_{ik}}{a_{kk}}$

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

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

      1: $a_{ij} \quad a_{ij} \qquad {}_{ik} a_{kj}$

- În final: $Ux = Mb$ se rezolvă cu UTRIS
- Complexitate totală: $\sum_{k=1}^{n \quad 1} (n \quad k + 2(n \quad k)^2) \quad \frac{2n^3}{3} = O(n^3)$
- 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 situaţia aceasta, multiplicatorii ${}_{ik}$?

Pivotare parţială

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

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 (şi/sau coloane) pentru a aduce în poziţia pivotului un element nenul.

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 şi k: $A \quad P_{i_k k}A$

Pivotare parţială

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

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 şi k: $A \quad P_{i_k k}A$

3: Se calculează M_k pentru $(M_k A)_{ik} = 0; i = k + 1 : n$

Pivotare parţială

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

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 $A \leftarrow M_k A$

Pivotare parţială

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

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 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** $k = 1 : n \quad 1$

  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: $a_{ik} \quad {}_{ik} = \frac{a_{ik}}{a_{kk}}$

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

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

      1: $a_{ij} \quad a_{ij} \quad {}_{ik} a_{kj}$

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

Algoritm $GPP(A)$

1: **Pentru** $k = 1 : n \quad 1$

  1: Se determină cel mai mic $i_k : ja_{i_k k}j = \max_{i=k:n} ja_{ik}j$

  2: $p(k) = i_k$

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

    1: $a_{kj} \; \$ \; a_{i_k j}$

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

    1: $a_{ik} \qquad {}_{ik} = \frac{a_{ik}}{a_{kk}}$

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

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

      1: $a_{ij} \quad a_{ij} \quad {}_{ik} a_{kj}$

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