

## Metode Iterative pentru SEL

Valentin-Ioan VINTILĂ

Facultatea de Automatică și Calculatoare - CTI  
Universitatea POLITEHNICA București

21 martie 2023 (Lab. 4)



## Cuprins

- 1 Clarificări
- 2 Introducere
- 3 Metoda lui Jacobi
- 4 Metoda Gauss-Seidel
- 5 Metoda suprarelaxării
- 6 Bibliografie



### Clarif. 1 - Givens (din temă) (1)

Vrem factorizarea Givens a lui  $A = \begin{bmatrix} 0 & -1 & 1 \\ 4 & 2 & 0 \\ 3 & 4 & 0 \end{bmatrix}$

**Pasul 1.** Scăpăm de  $A_{21}$ , adică de  $\begin{bmatrix} 0 & -1 & 1 \\ 4 & 2 & 0 \\ 3 & 4 & 0 \end{bmatrix}$

Vrem să scăpăm de  $A_{21}$ , deci formăm  $G_{12} = \begin{bmatrix} \cos & \sin & 0 \\ -\sin & \cos & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Avem  $\mathbf{x} = \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$ , așadar  $\begin{cases} \cos = x_1 / \|\mathbf{x}\| = 0 \\ \sin = x_2 / \|\mathbf{x}\| = 1 \end{cases}$



### Clarif. 1 - Givens (din temă) (2)

Factorizarea Givens pentru  $A = \begin{bmatrix} 0 & -1 & 1 \\ 4 & 2 & 0 \\ 3 & 4 & 0 \end{bmatrix}$

**Pasul 1.** Aplicăm  $G_{12} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  pe  $A$ :  $G_{12}A = \begin{bmatrix} 4 & 2 & 0 \\ 0 & 1 & -1 \\ 3 & 4 & 0 \end{bmatrix}$



### Clarif. 1 - Givens (din temă) (3)

Givens pentru  $A = \begin{bmatrix} 0 & -1 & 1 \\ 4 & 2 & 0 \\ 3 & 4 & 0 \end{bmatrix}$ , știind  $G_{12}A = \begin{bmatrix} 4 & 2 & 0 \\ 0 & 1 & -1 \\ 3 & 4 & 0 \end{bmatrix}$

**Pasul 2.** Scăpăm de  $(G_{12}A)_{31}$ , adică de  $\begin{bmatrix} 4 & 2 & 0 \\ 0 & 1 & -1 \\ 3 & 4 & 0 \end{bmatrix}$

Formăm  $G_{13} = \begin{bmatrix} \cos & 0 & \sin \\ 0 & 1 & 0 \\ -\sin & 0 & \cos \end{bmatrix}$ , unde  $\mathbf{x} = \begin{bmatrix} (G_{12}A)_{11} \\ (G_{12}A)_{31} \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$ , așadar

$\begin{cases} \cos = x_1 / \|\mathbf{x}\| = 0.8 \\ \sin = x_2 / \|\mathbf{x}\| = 0.6 \end{cases}$ . Aplicăm  $G_{13} = \begin{bmatrix} 0.8 & 0 & 0.6 \\ 0 & 1 & 0 \\ -0.6 & 0 & 0.8 \end{bmatrix}$  pe  $G_{12}A$ .



### Clarif. 1 - Givens (din temă) (4)

Givens pentru  $A = \begin{bmatrix} 0 & -1 & 1 \\ 4 & 2 & 0 \\ 3 & 4 & 0 \end{bmatrix}$ , știind  $G_{13}G_{12}A = \begin{bmatrix} 5 & 4 & 0 \\ 0 & 1 & -1 \\ 0 & 2 & 0 \end{bmatrix}$

**Pasul 3.** Scăpăm de  $(G_{13}G_{12}A)_{32}$ , adică de  $\begin{bmatrix} 5 & 4 & 0 \\ 0 & 1 & -1 \\ 0 & 2 & 0 \end{bmatrix}$

Formăm  $G_{23} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos & \sin \\ 0 & -\sin & \cos \end{bmatrix}$ , unde  $\mathbf{x} = \begin{bmatrix} (G_{13}G_{12}A)_{22} \\ (G_{13}G_{12}A)_{32} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ , așadar

$\begin{cases} \cos = x_1 / \|\mathbf{x}\| = 1/\sqrt{5} \\ \sin = x_2 / \|\mathbf{x}\| = 2/\sqrt{5} \end{cases}$ . Aplicăm  $G_{23} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/\sqrt{5} & 2/\sqrt{5} \\ 0 & -2/\sqrt{5} & 1/\sqrt{5} \end{bmatrix}$



### Clarif. 1 - Givens (din temă) (5)

Am obținut deci:

$$R = G_{23}G_{13}G_{12}A = \begin{bmatrix} 5 & 4 & 0 \\ 0 & \sqrt{5} & -1/\sqrt{5} \\ 0 & 0 & 2/\sqrt{5} \end{bmatrix}$$

Totodată, matricea  $Q$  se calculează ușor:

$$Q = G_{12}^T G_{13}^T G_{23}^T = \dots$$



### Clarif. 2 - Condiționarea matricelor (1)

Să considerăm sistemul de ecuații liniare  $A\mathbf{x} = \mathbf{b}$ :

$$\begin{bmatrix} 1 & 1 \\ 1 & 1.001 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \Rightarrow \begin{cases} x + y = 2 \\ x + 1.001y = 2 \end{cases}$$

Soluția este evidentă:  $x = 2$  și  $y = 0$ .

Alterăm foarte puțin vectorul  $\mathbf{b}$ :

$$\begin{bmatrix} 1 & 1 \\ 1 & 1.001 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ 2.001 \end{bmatrix} \Rightarrow \begin{cases} x + y = 2 \\ x + 1.001y = 2.001 \end{cases}$$

Efectul este **exagerat**:  $x = 1$  și  $y = 1!$

Spunem despre astfel de sisteme că sunt **rău condiționate**.



## Clarif. 2 - Condiționarea matricelor (2)

### Definiție (norma matriceală)

Fie  $\|\cdot\|_p$  o normă vectorială,  $p \in [1, +\infty]$ . Atunci, **norma matriceală** a matricei  $A \in \mathbb{R}^{n \times n}$  ( $n \in \mathbb{N}^*$ ) se definește drept:

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$$

**NU** folosim definiția, ci o consecință a acesteia.



## Clarif. 2 - Condiționarea matricelor (3)

Pentru norma euclidiană ( $p = 2$ , deci  $\|\cdot\|_2$ , notată  $\|\cdot\|$ ), vom folosi următoarea consecință a definiției anterioare:

$$\|A\| = \sqrt{\lambda_{\max}(A^T A)}$$



## Clarif. 2 - Condiționarea matricelor (4)

### Definiție (numărul de condiționare)

Fie  $A \in \mathbb{R}^{n \times n}$  o matrice nesingulară,  $n \in \mathbb{N}^*$ . Definim **numărul de condiționare** al matricei  $A$  drept:

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|$$

Ca o regulă *după ureche*, pierdem până la aproximativ  $\log_{10} \kappa(A)$  cifre de acuratețe **pe lângă** ceea ce pierde metoda în sine.



## Clarif. 2 - Condiționarea matricelor (5)

Pe exemplul anterior,  $A = \begin{bmatrix} 1 & 1 \\ 1 & 1.001 \end{bmatrix}$ , deci  $A^{-1} = \begin{bmatrix} 1001 & -1000 \\ -1000 & 1000 \end{bmatrix}$

Știm  $A^T A = \begin{bmatrix} 2 & 2.001 \\ 2.001 & 2.002 \end{bmatrix}$ , deci  $\begin{cases} \lambda_1 \approx 0 \\ \lambda_2 = 4.002 \end{cases}$

Știm și  $(A^{-1})^T A^{-1} = 10^6 \cdot \begin{bmatrix} 2.002 & -2.001 \\ -2.001 & 2 \end{bmatrix}$ , deci  $\begin{cases} \lambda'_1 \approx 0.25 \\ \lambda'_2 = 4.002 \cdot 10^6 \end{cases}$

Obținem  $\|A\| = \sqrt{\lambda_2} = 2.0005$ , respectiv  $\|A^{-1}\| = \sqrt{\lambda'_2} = 2000.5$ .

Astfel,  $\kappa(A) = \|A\| \cdot \|A^{-1}\| = 4002$  - o valoare foarte mare!



## Introducere (1)

Până acum am rezolvat sistemele prin metode directe (LU, G ș.a.).

### Ce nu ne convine?

- 1 Aceste metode au o complexitate de  $O(n^3)$ ;
- 2 Acuratețea este greu de controlat.



## Introducere (2)

Introducem acum **metodele iterative**:

- 1 **Aproximează** soluția din ce în ce mai bine;
- 2 Complexitatea depinde de numărul de pași:  $O(\alpha n^2)$ .



## Motorul metodelor (1)

Pornim de la un sistem de ecuații liniare,  $Ax = b$ , unde  $A \in \mathbb{R}^{n \times n}$ ,  $x, b \in \mathbb{R}^n$  și  $n \in \mathbb{N}^*$ .

Este evident că  $A$  poate fi scris ca o diferență de matrice. Cunoaștem așadar următoarele lucruri:

$$\begin{cases} Ax = b \\ A = N - P \end{cases}$$



## Motorul metodelor (2)

Înlocuim  $A$  în prima ecuație:

$$\begin{cases} Ax = b \\ A = N - P \end{cases} \Rightarrow (N - P)x = b$$

Vrem să rămânem doar cu  $x$ . Presupunem că  $\exists N^{-1}$ , deci:

$$\begin{aligned} (N - P)x = b &\Rightarrow Nx - Px = b \\ &\Rightarrow Nx = Px + b \\ &\Rightarrow x = N^{-1}Px + N^{-1}b \end{aligned}$$



Ce am obținut în scrierea  $\mathbf{x} = N^{-1}P\mathbf{x} + N^{-1}\mathbf{b}$ ?

**O modalitate de a-l scrie pe  $\mathbf{x}$  față de el însuși!**

Aplicând deci formula, presupunând că știm o aproximare de început, vom găsi o **aproximare mai bună**.

Putem apoi să folosim aproximarea obținută ca să găsim o aproximare și mai bună - de aici "metode **iterative**".



Amintim formula  $\mathbf{x} = N^{-1}P\mathbf{x} + N^{-1}\mathbf{b}$ . Notăm:

- 1 Vom nota aproximarea de la început cu  $\mathbf{x}^{(0)}$ , apoi  $\mathbf{x}^{(1)}$  va fi aproximarea la primul pas,  $\mathbf{x}^{(2)}$  la al doilea etc.;
- 2 Notăm  $G = N^{-1}P$ , denumită **matrice de iteratie**;
- 3 Mai notăm și  $\mathbf{c} = N^{-1}\mathbf{b}$ , numit **vector de iteratie**.

În aceste condiții, formula finală devine:

$$\mathbf{x}^{(p+1)} = G\mathbf{x}^{(p)} + \mathbf{c}$$

Numim această formulă **forma matriceală**.



## Descompunerea lui $A$

**Teoremă (unicitatea descompunerii  $A = D - L - U$ )**

O matrice  $A \in \mathbb{R}^{n \times n}$ ,  $n \in \mathbb{N}^*$ , poate fi scrisă în mod unic sub forma:

$$A = D - L - U$$

unde  $D$ ,  $L$ ,  $U$  au fost alese astfel încât să reprezinte:

- $D$  - matrice diagonală;
- $L$  - matrice inferior triunghiulară;
- $U$  - matrice superior triunghiulară.

...dar de ce nu  $A = D + L + U$ ?



## Condiția de convergență (1)

Metodele iterative **NU** converg întotdeauna!

Pentru ca o metodă iterativă să convergă, este **imperios necesar** ca raza spectrală a matricei de iteratie să fie subunitară. Matematic,  $\rho(G) < 1$ .



## Condiția de convergență (2)

**Definiție (matrice diagonal dominantă)**

Fie  $A \in \mathbb{R}^{n \times n}$  o matrice pătratică oarecare,  $n \in \mathbb{N}^*$ . Această se consideră **diagonal dominantă** dacă și numai dacă:

$$|A_{ii}| \geq \sum_{j \neq i} |A_{ij}|, \quad \forall i \in \overline{1, n}$$

Matricele diagonal dominante **converg întotdeauna** (adică matricele lor de iteratie au rază spectrală subunitară).



## Carl Gustav Jacob Jacobi



Carl Gustav Jacob Jacobi (1804-1825)



## Metoda lui Jacobi (1)

Avem un sistem oarecare  $A\mathbf{x} = \mathbf{b}$ , cu  $A$  scris drept  $A = N - P$ , respectiv drept  $A = D - L - U$ .

**Metoda lui Jacobi** alege:

$$\begin{cases} N = D \\ P = L + U \end{cases}$$

Vom scrie acum și  $G_{\text{Jacobi}}$  și  $\mathbf{c}_{\text{Jacobi}}$  cu aceste valori:

$$\begin{cases} G_{\text{Jacobi}} = N^{-1}P = D^{-1}(L + U) \\ \mathbf{c}_{\text{Jacobi}} = N^{-1}\mathbf{b} = D^{-1}\mathbf{b} \end{cases}$$



## Metoda lui Jacobi (2)

Se poate desfășura forma matriceală  $\mathbf{x}^{(p+1)} = G_{\text{Jacobi}}\mathbf{x}^{(p)} + \mathbf{c}_{\text{Jacobi}}$  pentru a se obține **forma analitică**.

*Fără demonstrație*, aceasta este:

$$x_i^{(p+1)} = \frac{b_i - \sum_{j=1, j \neq i}^n a_{ij}x_j^{(p)}}{a_{ii}}, \quad \forall i \in \overline{1, n}$$



## Metoda lui Jacobi - implementare (1)

Știm formula pentru orice element al oricărei iterații:

$$x_i^{(p+1)} = \frac{b_i - \sum_{j=1, j \neq i}^n a_{ij} x_j^{(p)}}{a_{ii}}, \quad \forall i = \overline{1, n}$$

Cum implementăm?



## Metoda lui Jacobi - implementare (2)

Începem prin a defini două valori:

- **Toleranța** (not.  $\varepsilon$ ) - diferența **minimă** dintre soluțiile a două iterații consecutive necesară pentru a continua cu algoritmul.

Cu alte cuvinte, oprim algoritmul când  $\|x^{(p+1)} - x^{(p)}\| < \varepsilon$ .

- **Numărul maxim de iterații** - este posibil să nu convergă (suficient de repede) algoritmul, așa dar îl oprim forțat mai devreme.



## Metoda lui Jacobi - implementare (3)

Implementarea este încărcată pe **Teams** - să vedem...



## Metoda lui Jacobi - exercitiu rapid

Fie sistemul  $\begin{bmatrix} 1 & 1 & 1 \\ -2 & 6 & 1 \\ -1 & 1 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 9 \\ -6 \end{bmatrix}$ . Dacă  $x^{(0)} = \mathbf{0}$ , rulați codul pentru câteva iterații și observați la ce rezultat se ajunge după câteva iterații, de pildă 6.

**Ne-am apropiat de soluția corectă?**

Soluția era  $x = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$ , iar  $x^{(6)} = \begin{bmatrix} 1.0034 \\ 2.0855 \\ -0.9603 \end{bmatrix}$



## Metoda lui Jacobi - concluzii

O folosim în practică? **NU** - converge greu, după cum am observat!



## Gauss și von Seidel



Carl Friedrich Gauss (1777-1855)



Philipp Ludwig von Seidel (1821-1896)



## Metoda Gauss-Seidel (1)

Avem un sistem oarecare  $Ax = b$ , cu  $A$  scris drept  $A = N - P$ , respectiv drept  $A = D - L - U$ .

**Metoda Gauss-Seidel** alege:

$$\begin{cases} N = D - L \\ P = U \end{cases}$$

Vom scrie acum și  $G_{\text{Gauss-Seidel}}$  și  $c_{\text{Gauss-Seidel}}$  cu aceste valori:

$$\begin{cases} G_{\text{Gauss-Seidel}} = N^{-1}P = (D - L)^{-1}U \\ c_{\text{Gauss-Seidel}} = N^{-1}b = (D - L)^{-1}b \end{cases}$$



## Metoda Gauss-Seidel (2)

Prin desfășurarea formei matriceale

$$x^{(p+1)} = G_{\text{Gauss-Seidel}} x^{(p)} + c_{\text{Gauss-Seidel}}$$
 se obține **forma analitică**.

Fără demonstrație, aceasta este:

$$x_i^{(p+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(p+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(p)}}{a_{ii}}, \quad \forall i = \overline{1, n}$$

Deci, din punct de vedere al interpretării, folosim **mereu** cele mai recente valori pe care le avem.



Rapid, *adaptati* algoritmul Jacobi pentru a calcula Gauss-Seidel:

$$x_i^{(p+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(p+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(p)}}{a_{ii}}, \forall i = \overline{1, n}$$

Testati apoi același sistem:  $\begin{bmatrix} 1 & 1 & 1 \\ -2 & 6 & 1 \\ -1 & 1 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 9 \\ -6 \end{bmatrix}$ , cu  $x^{(0)} = \mathbf{0}$ .

Considerati mai puține iteratii (de exemplu 4).

Ce observati? Soluția era  $x = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$ , iar GS a dat  $x^{(4)} = \begin{bmatrix} 0.9368 \\ 1.9768 \\ -1.0057 \end{bmatrix}$



O folosim în practică? **Oarecum** - nu tocmai în această formă... Vedem imediat metoda suprarelaxării!



Young și Frankel



David M. Young Jr. (1923-2008)



Stanley Phillips Frankel (1919-1978)



Metoda SOR (1)

Inovația față de Gauss-Seidel este introducerea pe cale **pur artificială** a unei constante  $\omega \in (0, 1)$ .

Rămânem la scrierea  $A = D - L - U$ , dar redefinim  $\omega A = N - P$ . Pentru metoda SOR, alegem:

$$\begin{cases} N = D - \omega L \\ P = (1 - \omega)D + \omega U \end{cases}$$

Vom scrie acum și  $G_{SOR}$  și  $C_{SOR}$  cu aceste valori:

$$\begin{cases} G_{SOR} = N^{-1}P = (D - \omega L)^{-1}[(1 - \omega)D + \omega U] \\ C_{SOR} = N^{-1}b = (D - \omega L)^{-1}(\omega b) \end{cases}$$



Metoda SOR (2)

**Forma analitică**, cea care ne interesează de fapt, este asemănătoare cu Gauss-Seidel:

$$x_i^{(p+1)} = (1 - \omega)x_i^{(p)} + \frac{\omega}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(p+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(p)} \right), \forall i = \overline{1, n}$$

Să digerăm această ecuație:

- Termenul cu **roșu** este doar o scalare adusă metodei Gauss-Seidel, având o pondere  $\omega \in (0, 1)$ ;
- Termenul cu **portocaliu** este nou. Amintim că  $\omega \in (0, 1)$ , deci  $(1 - \omega) \in (0, 1)$ . Deci, **un fel de medie ponderată!**



Metoda SOR - implementare

Rapid, *adaptati* algoritmul Gauss-Seidel pentru a calcula SOR:

$$x_i^{(p+1)} = (1 - \omega)x_i^{(p)} + \frac{\omega}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(p+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(p)} \right), \forall i = \overline{1, n}$$

Testati apoi același sistem:  $\begin{bmatrix} 1 & 1 & 1 \\ -2 & 6 & 1 \\ -1 & 1 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 9 \\ -6 \end{bmatrix}$ , cu  $x^{(0)} = \mathbf{0}$  și  $\omega = 0.85$ . Considerati mai puține iteratii (de exemplu 2).

Ce observati? Soluția era  $x = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$ , iar SOR a dat  $x^{(2)} = \begin{bmatrix} 1.0870 \\ 1.9507 \\ -0.9438 \end{bmatrix}$



Metoda SOR - alegerea lui  $\omega$

Nu există o formulă concretă general valabilă pentru a calcula cea mai bună valoare a lui  $\omega$ .

În anumite situații, David Young a demonstrat că:

$$\omega_{opt} = 1 + \left( \frac{\mu_{max}}{1 + \sqrt{1 - \mu_{max}^2}} \right)^2$$

unde prin  $\mu_{max}$  se înțelege valoarea proprie maximă a matricei  $G_{Jacobi}$ .



Bibliografie

Pentru aceste prezentări, am utilizat:

1. Cărțile *Matrix Decomposition and Applications*, respectiv *Numerical Matrix Decomposition and its Modern Applications: A Rigorous First Course* ale lui **Jun Lu**.
2. Articolul *Successive Over-Relaxation (SOR)* din *Advanced Linear Algebra: Foundations to Frontiers* scrisă de **Robert van de Geijn** și **Margaret Myers**.



**Mulțumesc frumos pentru atenție!**

Vă rog frumos să **completați formularul de feedback!**

