

Sisteme de ecuații algebrice liniare - metode directe

Prof.dr.ing. Gabriela Ciuprina

Universitatea "Politehnica" București, Facultatea de Inginerie Electrică,
Departamentul de Electrotehnică

Suport didactic pentru disciplina *Algoritmi Numerici*, 2017-2018

Notes

Cuprins

- 1 Formularea problemei
 - Enunț
 - Buna formulare matematică
 - Condiționarea problemei
- 2 Metoda Gauss
 - Idee
 - Algoritm
 - Pivotare
 - Concluzii
 - Cazul sistemelor multiple
- 3 Metoda factorizării LU
 - Varianta Doolittle
 - Varianta Cholesky
- 4 Matrice rare
 - Formate de memorare
 - Adaptarea metodelor directe - exemplu

Notes

Formulara problemei

Sistem de n ecuații algebrice liniare cu n necunoscute:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n. \end{cases} \quad (1)$$

Notes

Formulara problemei

Se dă matricea coeficienților

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \in \mathbb{R}^{n \times n} \quad (2)$$

și vectorul termenilor liberi

$$\mathbf{b} = [b_1 \quad b_2 \quad \dots \quad b_n]^T \in \mathbb{R}^n, \quad (3)$$

se cere să se rezolve sistemul

$$\mathbf{Ax} = \mathbf{b}, \quad (4)$$

unde \mathbf{x} este soluția

$$\mathbf{x} = [x_1 \quad x_2 \quad \dots \quad x_n]^T \in \mathbb{R}^n, \quad (5)$$

Notes

Buna formulare matematică

Problema este bine formulată din punct de vedere matematic (soluția există și este unică)

⇔

matricea **A** este nesingulară (are determinantul nenul).

Se scrie formal:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

trebuie citită ca:

"x este soluția sistemului algebric liniar Ax = b"

și **NU** "se calculează inversa matricei **A** care se înmulțește cu vectorul **b**".

Notes

Condiționarea problemei

Condiționarea

se referă la comportarea **problemei matematice** la perturbații ale datelor.

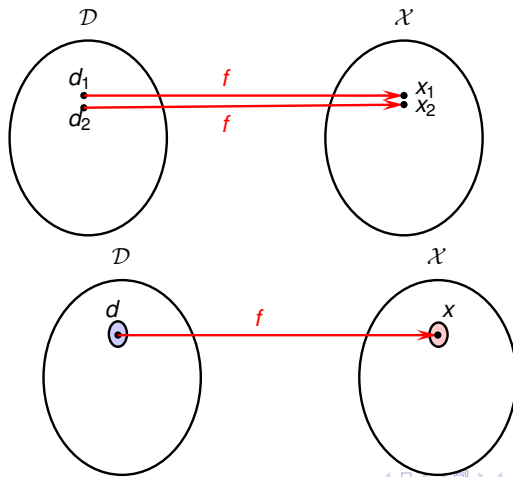
Problemă matematică f formulată explicit:

Fie $f: \mathcal{D} \rightarrow \mathcal{X}$ și $\mathbf{d} \in \mathcal{D}$.

Să se găsească $\mathbf{x} \in \mathcal{X}$ astfel încât $f(\mathbf{d}) = \mathbf{x}$. (6)

Notes

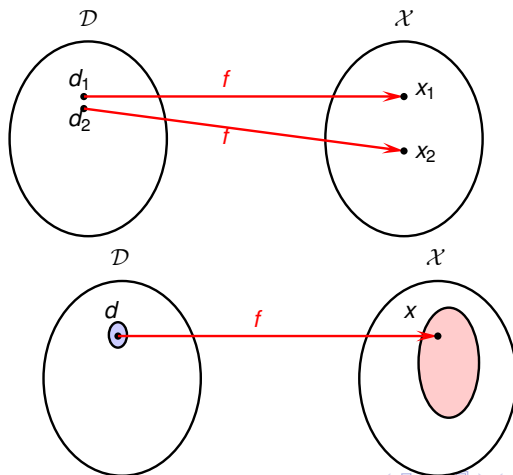
Reprezentări intuitive - problemă bine condiționată



7/91

Notes

Reprezentări intuitive - problemă prost condiționată



8/91

Notes

Numărul de condiționare

$$\|\mathbf{b}\| = \|\mathbf{Ax}\| \leq \|\mathbf{A}\|\|\mathbf{x}\| \Rightarrow \|\mathbf{x}\| \geq \frac{\|\mathbf{b}\|}{\|\mathbf{A}\|}. \quad (12)$$

Un majorant pentru eroarea asupra soluției

$$\varepsilon_x = \frac{\|\mathbf{e}_x\|}{\|\mathbf{x}\|} \leq \frac{\|\mathbf{A}^{-1}\|\|\mathbf{e}_b\|}{\frac{\|\mathbf{b}\|}{\|\mathbf{A}\|}} = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|\varepsilon_b. \quad (13)$$

$$\kappa(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^{-1}\| \quad (14)$$

număr de condiționare la inversare al matricei \mathbf{A} .

$$\varepsilon_x \leq \kappa(\mathbf{A})\varepsilon_b, \quad (15)$$

Notes

Numărul de condiționare - altă demonstrație

Pornind de la definiția generală:

$$\kappa = \lim_{\delta \rightarrow 0} \sup_{\|\delta \mathbf{d}\| < \delta} \frac{\|\delta f\| / \|f(\mathbf{d})\|}{\|\delta \mathbf{d}\| / \|\mathbf{d}\|}, \quad (16)$$

sau, scris mai simplu în ipoteza unor variații infinitesimale

$$\kappa = \sup_{\|\delta \mathbf{d}\|} \frac{\|\delta f\| / \|f(\mathbf{d})\|}{\|\delta \mathbf{d}\| / \|\mathbf{d}\|}. \quad (17)$$

Dacă f este derivabilă, atunci

$$k = \frac{\|\mathbf{J}(\mathbf{d})\|}{\|f(\mathbf{d})\| / \|\mathbf{d}\|}. \quad (18)$$

Notes

Numărul de condiționare - altă demonstrație

Presupunem doar datele \mathbf{b} perturbate, iar soluția problemei este scrisă formal ca $\mathbf{x} = \mathbf{f}(\mathbf{b}) = \mathbf{A}^{-1}\mathbf{b}$, având matricea Jacobian $\mathbf{J}(\mathbf{b}) = \mathbf{A}^{-1}$.

$$\begin{aligned} \kappa &= \frac{\|\mathbf{J}(\mathbf{b})\|}{\|\mathbf{f}(\mathbf{b})\|/\|\mathbf{b}\|} = \frac{\|\mathbf{A}^{-1}\|}{\|\mathbf{A}^{-1}\mathbf{b}\|/\|\mathbf{b}\|} = \frac{\|\mathbf{A}^{-1}\|\|\mathbf{b}\|}{\|\mathbf{A}^{-1}\mathbf{b}\|} = \\ &= \frac{\|\mathbf{A}^{-1}\|\|\mathbf{A}\mathbf{A}^{-1}\mathbf{b}\|}{\|\mathbf{A}^{-1}\mathbf{b}\|} \leq \\ &\leq \frac{\|\mathbf{A}^{-1}\|\|\mathbf{A}\|\|\mathbf{A}^{-1}\mathbf{b}\|}{\|\mathbf{A}^{-1}\mathbf{b}\|} = \|\mathbf{A}^{-1}\|\|\mathbf{A}\| = \kappa(\mathbf{A}), \end{aligned} \quad (19)$$

Notes

Marginea inferioară pentru eroarea asupra soluției

$$\|\mathbf{e}_b\| = \|\mathbf{A}\mathbf{e}_x\| \leq \|\mathbf{A}\|\|\mathbf{e}_x\| \Rightarrow \|\mathbf{e}_x\| \geq \frac{\|\mathbf{e}_b\|}{\|\mathbf{A}\|}. \quad (20)$$

$$\|\mathbf{x}\| = \|\mathbf{A}^{-1}\mathbf{b}\| \leq \|\mathbf{A}^{-1}\|\|\mathbf{b}\|. \quad (21)$$

\Rightarrow

$$\varepsilon_x = \frac{\|\mathbf{e}_x\|}{\|\mathbf{x}\|} \geq \frac{\|\mathbf{e}_b\|}{\|\mathbf{A}\|\|\mathbf{A}^{-1}\|\|\mathbf{b}\|} = \frac{\varepsilon_b}{\kappa(\mathbf{A})}. \quad (22)$$

$$\frac{\varepsilon_b}{\kappa(\mathbf{A})} \leq \varepsilon_x \leq \kappa(\mathbf{A})\varepsilon_b. \quad (23)$$

Notes

Numărul de condiționare - proprietăți

- Numărul de condiționare este întotdeauna supraunitar $\kappa(\mathbf{A}) \geq 1$:

$$1 = \|\mathbf{I}\| = \|\mathbf{A}\mathbf{A}^{-1}\| \leq \|\mathbf{A}\| \|\mathbf{A}^{-1}\| = \kappa(\mathbf{A}). \quad (24)$$

Cazul cel mai favorabil: $n_A = 1$ și $\varepsilon_x = \varepsilon_b$. (matrice ortogonală)

- Numărul de condiționare este o proprietate a matricei și nu are legătură nici cu metoda de rezolvare propriu-zisă, nici cu erorile de rotunjire care apar în mediul de calcul.

În practică:

Dacă $\kappa(\mathbf{A}) > 1/\text{eps}$ problema se consideră slab condiționată.

Notes

Perturbații în matricea coeficienților

$$(\mathbf{A} + \mathbf{e}_A)(\mathbf{x} + \mathbf{e}_x) = \mathbf{b}. \quad (25)$$

$$\mathbf{A}\mathbf{e}_x = -\mathbf{e}_A(\mathbf{x} + \mathbf{e}_x), \quad (26)$$

$$\|\mathbf{e}_x\| = \|\mathbf{A}^{-1}\mathbf{e}_A(\mathbf{x} + \mathbf{e}_x)\| \leq \|\mathbf{A}^{-1}\| \|\mathbf{e}_A\| \|\mathbf{x} + \mathbf{e}_x\|. \quad (27)$$

$$\varepsilon_x \approx \frac{\|\mathbf{e}_x\|}{\|\mathbf{x} + \mathbf{e}_x\|} \leq \|\mathbf{A}^{-1}\| \|\mathbf{e}_A\| = \|\mathbf{A}^{-1}\| \|\mathbf{A}\| \frac{\|\mathbf{e}_A\|}{\|\mathbf{A}\|} = \kappa(\mathbf{A})\varepsilon_A. \quad (28)$$

Deoarece $\|\mathbf{x} + \mathbf{e}_x\| \leq \|\mathbf{x}\| + \|\mathbf{e}_x\|$, rezultă că

$$\|\mathbf{x}\| \geq \|\mathbf{x} + \mathbf{e}_x\| - \|\mathbf{e}_x\|.$$

Notes

Perturbații în matricea coeficienților

Dacă presupunem că

$$\|\mathbf{x} + \mathbf{e}_x\| - \|\mathbf{e}_x\| > 0, \quad (29)$$

⇒

$$\begin{aligned} \varepsilon_x &= \frac{\|\mathbf{e}_x\|}{\|\mathbf{x}\|} \leq \frac{\|\mathbf{e}_x\|}{\|\mathbf{x} + \mathbf{e}_x\| - \|\mathbf{e}_x\|} = \frac{\|\mathbf{e}_x\|/\|\mathbf{x} + \mathbf{e}_x\|}{1 - \|\mathbf{e}_x\|/\|\mathbf{x} + \mathbf{e}_x\|} \leq \\ &\leq \frac{\kappa(\mathbf{A})\varepsilon_A}{1 - \kappa(\mathbf{A})\varepsilon_A}, \end{aligned} \quad (30)$$

relație valabilă în ipoteza $\kappa(\mathbf{A})\varepsilon_A < 1$.

Clasificarea metodelor

- 1 **Metode directe** - găsesc soluția teoretică a problemei într-un **număr finit de pași**. (Gauss, factorizare LU)
- 2 **Metode iterative** - generează un **șir de aproximații** ale soluției care se dorește a fi convergent către soluția exactă.
 - **staționare**: Jacobi, Gauss-Seidel, SOR, SSOR
 - **nestaționare (semiiterative)**: gradienti conjugați (GC), reziduu minim (MINRES), reziduu minim generalizat (GMRES), gradienti biconjugați (BIGC), etc.

Notes

Notes

Ideea metodei Gauss

$$\mathbf{Ax} = \mathbf{b} \quad \begin{array}{c} \Leftrightarrow \\ \text{eliminare} \end{array} \quad \mathbf{Ux} = \mathbf{b}' \quad \begin{array}{c} \Rightarrow \\ \text{subst.regresivă} \end{array} \quad \mathbf{x} = \mathbf{U}^{-1}\mathbf{b}'. \quad (31)$$

Notes

Un exemplu simplu

$$\begin{cases} x_1 + 2x_2 - x_3 = -1, \\ -2x_1 + 3x_2 + x_3 = 0, \\ 4x_1 - x_2 - 3x_3 = -2. \end{cases} \quad (32)$$

$$\begin{cases} x_1 + 2x_2 - x_3 = -1, \\ 7x_2 - x_3 = -2, \\ -9x_2 + x_3 = 2. \end{cases} \quad (33)$$

$$\begin{cases} x_1 + 2x_2 - x_3 = -1, \\ 7x_2 - x_3 = -2, \\ -2/7x_3 = -4/7. \end{cases} \quad (34)$$

$$\begin{aligned} x_3 &= (-4/7)/(-2/7) = 2, \\ x_2 &= (-2 + x_3)/7 = 0, \\ x_1 &= -1 - 2x_2 + x_3 = 1. \end{aligned} \quad (35)$$

Notes

Algoritmul metodei Gauss - etapa de eliminare

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \rightarrow \begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{bmatrix} \rightarrow \begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \end{bmatrix} \rightarrow \dots \rightarrow \begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix}$$

$A_0 \qquad A_1 \qquad A_2 \qquad \dots \qquad A_{n-1}$

Eliminare în metoda Gauss: pentru un sistem de dimensiune n există $n - 1$ sub-etape de eliminare. La final matricea este superior triunghiulară. Matricea inițială este notată A_0 iar matricea superior triunghiulară obținută este notată A_{n-1} . În realitate, transformările sunt memorate "în loc", în același tablou bidimensional.

Notes

Algoritmul metodei Gauss - etapa de eliminare

Modificarea ecuației a doua în prima sub-etapă de eliminare poate fi descrisă astfel:

; anularea elementului a_{21}

$p = -a_{21}/a_{11}$; element de multiplicare

pentru $j = 1, n$; parcurge coloanele

$$a_{2j} = a_{2j} + pa_{1j}$$

•

$$b_2 = b_2 + pb_1$$

2 \rightarrow i inserată într-un ciclu cu contor

Notes

Algoritmul metodei Gauss - etapa de eliminare

Prima sub-etapă de eliminare:

; prima sub-etapă de eliminare

pentru $i = 2, n$; parcurge liniile
 $p = -a_{i1}/a_{11}$; element de multiplicare
pentru $j = 2, n$; parcurge coloanele
 $a_{ij} = a_{ij} + pa_{1j}$

•
 $b_i = b_i + pb_1$

•

OBS: În ciclul în j contorul începe cu valoarea 2.

$1 \rightarrow k, 2 \rightarrow k + 1$ inserate într-un ciclu cu contor.

Notes

Algoritmul metodei Gauss - etapa de eliminare

Secvența de cod corespunzătoare etapei de eliminare

; etapa de eliminare din metoda Gauss

pentru $k = 1, n - 1$
pentru $i = k + 1, n$; parcurge liniile
 $p = -a_{ik}/a_{kk}$; element de multiplicare
pentru $j = k + 1, n$; parcurge coloanele
 $a_{ij} = a_{ij} + pa_{kj}$

•
 $b_i = b_i + pb_k$

•

•

Notes

Algoritmul metodei Gauss - substituție regresivă

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \dots \\ a_{nn}x_n = b_n. \end{cases} \quad (36)$$

$$x_n = b_n/a_{nn}, \quad (37)$$

$$a_{ij}x_i + a_{i,i+1}x_{i+1} + \dots + a_{in}x_n = b_i, \quad (38)$$

⇒

$$x_i = \frac{b_i - \sum_{j=i+1}^n a_{ij}x_j}{a_{ij}}. \quad (39)$$

Notes

Algoritmul metodei Gauss - substituție regresivă

$$x_n = b_n/a_{nn}, \quad (40)$$

$$x_i = \frac{b_i - \sum_{j=i+1}^n a_{ij}x_j}{a_{ij}}. \quad (41)$$

; etapa de retrosubstituție

$$x_n = b_n/a_{nn}$$

pentru $i = n - 1, 1, -1$

$$s = 0$$

pentru $j = i + 1, n$

$$s = s + a_{ij}x_j$$

•

$$x_i = (b_i - s)/a_{ii}$$

•

Notes

Algoritmul metodei Gauss

```
procedură Gauss( $n, a, b, x$ )  
; rezolvă sistemul algebric liniar  $ax = b$  prin metoda Gauss  
întreg  $n$  ; dimensiunea sistemului  
tablou real  $a[n][n]$  ; matricea coeficienților - indici de la 1  
tablou real  $b[n]$  ; vectorul termenilor liberi  
tablou real  $x[n]$  ; vectorul soluție  
întreg  $i, j, k$   
real  $p, s$   
; etapa de eliminare  
pentru  $k = 1, n - 1$  ;  
; aici se poate introduce pivotarea  
pentru  $i = k + 1, n$  ; parcurge liniile  
     $p = -a_{ik} / a_{kk}$  ; element de multiplicare  
    pentru  $j = k + 1, n$  ; parcurge coloanele  
         $a_{ij} = a_{ij} + pa_{kj}$   
        •  
         $b_i = b_i + pb_k$   
    •  
•
```

Algoritmul metodei Gauss

```
; etapa de retroversubstituție  
 $x_n = b_n / a_{nn}$   
pentru  $i = n - 1, 1, -1$   
     $s = 0$   
    pentru  $j = i + 1, n$   
         $s = s + a_{ij}x_j$   
    •  
     $x_i = (b_i - s) / a_{ii}$   
•  
retur
```

Algoritmul poate fi îmbunătățit prin folosirea la fiecare etapă de eliminare a unei strategii de pivotare.

Notes

Notes

Evaluarea algoritmului

Din punct de vedere al timpului de calcul:

$$\begin{aligned} T_e &= \sum_{k=1}^{n-1} [2(n-k) + 3](n-k) \approx \sum_{k=1}^{n-1} 2(n-k)^2 = \\ &= 2 \frac{(n-1)n(2n-1)}{6} \approx \frac{2n^3}{3}. \end{aligned} \quad (42)$$

$$T_s = \sum_{i=1}^{n-1} [2(n-i) + 2] \approx \sum_{i=1}^{n-1} [2(n-i)] = 2 \frac{n(n-1)}{2} \approx n^2. \quad (43)$$

$$T_{\text{Gauss}} = O(2n^3/3 + n^2) = O(2n^3/3) - \text{costisitor}$$

Din punct de vedere al necesarului de memorie:

$$M = n^2 + 2n + 2 \Rightarrow M = O(n^2)$$

Notes

Evaluarea algoritmului

Din punct de vedere al erorilor: erorile inerente, erori de rotunjire.

- Cu cât sistemul este de dimensiune mai mare, cu atât erorile acumulate datorită rotunjirii cresc.
- O diminuare a erorilor de rotunjire se poate obține dacă se includ în algoritm strategii de pivotare.

Din punct de vedere al stabilității: algoritmul Gauss poate să nu fie stabil chiar dacă problema matematică este bine formulată și bine condiționată (numărul de condiționare al matricei **A** este mic). Acest lucru se întâmplă atunci când numărul de condiționare al matricei **U** este mare. Remediul îl constituie în acest caz pivotarea.

Notes

Strategii de pivotare

Pivoți

Elementele diagonale a_{kk} obținute în urma etapei de eliminare.

Determinantul sistemului = produsul pivoților.

⇒

Problema este bine formulată matematic \Leftrightarrow toți pivoții sunt nenuli.

Elementele de multiplicare: $p = -a_{ik}/a_{kk}$. a_{kk} = pivot,

Pivotare

Operație de permutare care urmărește obținerea valorilor nenule pentru pivoți.

Trebuie făcută înainte de calculul multiplicatorului.

31/91

Notes

Strategii de pivotare

Strategii de pivotare:

- Pivotarea pe linii (parțială)
- Pivotarea pe coloane
- Pivotarea totală (completă sau maximală)
- Pivotarea diagonală

$$\begin{bmatrix} X & X & X & X & X & X \\ 0 & X & X & X & X & X \\ 0 & 0 & a_{kk} & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \end{bmatrix} \quad \begin{bmatrix} X & X & X & X & X & X \\ 0 & X & X & X & X & X \\ 0 & 0 & a_{kk} & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \end{bmatrix}$$

$$\begin{bmatrix} X & X & X & X & X & X \\ 0 & X & X & X & X & X \\ 0 & 0 & a_{kk} & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \end{bmatrix} \quad \begin{bmatrix} X & X & X & X & X & X \\ 0 & X & X & X & X & X \\ 0 & 0 & a_{kk} & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \end{bmatrix}$$

32/91

Notes

Algoritmul pivotării pe linii

$p = 0$
pentru $i = k, n$; parcurge coloana k
dacă $|a_{ik}| > p$ atunci
 $l = i$; memorează poziția potențialului pivot
 $p = |a_{ik}|$

•
dacă $p = 0$ atunci
 scrie "problema este prost formulată matematic"

altfel
pentru $j = k, n$; permută linia l cu linia k
 $p = a_{kj}$
 $a_{kj} = a_{lj}$
 $a_{lj} = p$

•
 $p = b_k$
 $b_k = b_l$; permută termenii liberi
 $b_l = p$

Pivotarea parțială (pe linii)

Observații:

- 1 Într-o implementare eficientă nu se face efectiv rocada liniilor, ci este memorată permutarea necesară într-un vector.
- 2 O variantă a acestei pivotări se numește *pivotare scalată*. Se selectează ca linie pivot, linia pentru care posibilul element pivot este cel mai mare în raport cu valorile elementelor corespunzătoare acestei linii. O astfel de strategie este utilă atunci când elementele dintr-o linie sunt foarte diferite ca ordine de mărime, fiind mai eficientă decât pivotarea parțială clasică. Pentru detalii, consultați [Cheney].

Notes

Notes

Avantajele pivotării

Pivotarea

- 1 necesară dacă pe parcursul algoritmului se întâlnește un pivot nul.
- 2 efect benefic asupra stabilității și acurateții.

Notes

Avantajele pivotării

Exemplu

$$\begin{cases} 10^{-20}x + y = 1, \\ x + y = 0, \end{cases} \quad (44)$$

soluția corectă $(x, y) \approx (-1, 1)$.

Gauss și presupunem că $\text{eps} = 10^{-16}$:

$$\begin{cases} 10^{-20}x + y = 1, \\ (1 - 10^{20})y = -10^{20}. \end{cases} \quad (45)$$

$$\begin{cases} 10^{-20}x + y = 1, \\ -10^{20}y = -10^{20}. \end{cases} \quad (46)$$

Rezultatul final: $(x, y) = (0, 1)$ extrem de eronat.

Explicație: $\kappa(\mathbf{A}) \approx 2.6$, dar $\kappa(\mathbf{U}) = 10^{40}$!

Notes

Metoda Gauss - Concluzii

- Este o metodă directă - găsește soluția teoretică a problemei într-un **număr finit de pași**.
- Calculele sunt afectate de **erori de rotunjire** \Rightarrow nu se obține soluția exactă, ci o aproximare a ei.
- Se transformă **sistemului de ecuații** într-unul **echivalent** din punct de vedere al soluției (Δ sup.), mult mai ușor de rezolvat (subs. regr.).
- Pivotarea: esențială pentru a asigura pivoți nenuli; utilă pentru a crește stabilitatea algoritmului și acuratețea soluției.

Notes

Metoda Gauss - Concluzii

- Pivotarea parțială are un efort de implementare nesemnificativ.
- Pivotarea totală este rareori aplicată deoarece duce la o creștere semnificativă a timpului de calcul, nerealizând decât o îmbunătățire nesemnificativă a acurateții soluției.
- Dezavantajul metodei Gauss: în anumite situații, **efortul de generare a problemei echivalente (eliminarea) este mare sau, necesarul de memorie poate deveni extrem de mare**.

Notes

Formularea problemei

Fie m sisteme de ecuații algebrice liniare

$$\mathbf{Ax}^{(1)} = \mathbf{b}^{(1)}, \quad \mathbf{Ax}^{(2)} = \mathbf{b}^{(2)}, \quad \dots, \quad \mathbf{Ax}^{(m)} = \mathbf{b}^{(m)}, \quad (47)$$

Se dau: $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{b}^{(k)} \in \mathbb{R}^{n \times 1}$, $k = 1, m$

Se cer: $\mathbf{x}^{(k)} \in \mathbb{R}^{n \times 1}$,

Notăm

$$\mathbf{B} = [\mathbf{b}^{(1)} \quad \mathbf{b}^{(2)} \quad \dots \quad \mathbf{b}^{(m)}] \in \mathbb{R}^{n \times m} \quad (48)$$

$$\mathbf{X} = [\mathbf{x}^{(1)} \quad \mathbf{x}^{(2)} \quad \dots \quad \mathbf{x}^{(m)}] \in \mathbb{R}^{n \times m} \quad (49)$$

Se cere să se rezolve sistemul

$$\mathbf{AX} = \mathbf{B}. \quad (50)$$

Varianta I

Varianta I - aplicarea **succesivă a algoritmului Gauss**

Efort de calcul: $m(2n^3/3 + n^2) \approx 2mn^3/3$.

Etapa de eliminare este repetată inutil, de m ori.

Cea mai proasta idee.

Notes

Notes

Varianta II

Varianta II - rezolvarea **simultană** prin adaptarea algoritmului Gauss

```
procedură Gauss_multiplu(n, m, a, B, X)
; rezolvă simultan sistemele algebrice liniare  $aX = B$  prin metoda Gauss
  întreg n ; dimensiunea sistemului
  întreg m ; numărul de sisteme
  tablou real a[n][n] ; matricea coeficienților - indici de la 1
  tablou real B[n][m] ; matricea termenilor liberi
  tablou real X[n][m] ; matricea soluție
  întreg i, j, k
  real p, s
; etapa de eliminare
pentru k = 1, n - 1 ; parcurge sub-etape ale eliminării
  ; aici se poate introduce pivotarea
  pentru i = k + 1, n ; parcurge liniile
    p = -aik / akk ; element de multiplicare
    pentru j = k + 1, n ; parcurge coloanele
      aij = aij + p akj
    •
    pentru j = 1, m ; parcurge coloanele termenilor liberi
      bij = bij + p bkj
  •
•
```

Notes

Varianta II

```
; etapa de retrosubstituție
pentru k = 1, m
  xnk = bnk / ann
  pentru i = n - 1, 1, -1
    s = 0
    pentru j = i + 1, n
      s = s + aij xjk
    •
    xik = (bik - s) / aii
  •
•
retur
```

Notes

Varianta II

Efort de calcul

$$T_e = \sum_{k=1}^{n-1} [2(n-k) + 2m + 1](n-k) \approx \sum_{k=1}^{n-1} [2(n-k)^2 + 2m(n-k)] =$$

$$= 2 \frac{(n-1)n(2n-1)}{6} + 2m \frac{n(n-1)}{2} \approx \frac{2n^3}{3} + mn^2. \quad (51)$$

$$T_s = m \sum_{i=1}^{n-1} [2(n-i) + 2] \approx m \sum_{i=1}^{n-1} [2(n-i)] = 2m \frac{n(n-1)}{2} \approx mn^2. \quad (52)$$

$T = O(2n^3/3 + 2mn^2)$, mai mic decât în cazul variantei I.

Notes

Varianta III

Varianta III - rezolvarea succesivă a sistemelor folosind calculul inversei

- Se calculează \mathbf{A}^{-1}
- Se calculează $\mathbf{x}^{(k)} = \mathbf{A}^{-1} \mathbf{b}^{(k)}$ imediat ce este cunoscut termenul liber.

Notes

Varianta III

```
functie invA(n, a)
; calculează inversa matricei a
intreg n ; dimensiunea matricei
tablou real a[n][n] ; matricea, indici de la 1
; alte declarații
....
pentru i = 1, n
    pentru j = 1, n
        •
        Bij = 0
    •
        Bij = 1
    Gauss_multiplu(n, n, a, B, X)
    intoarce X ; X este inversa matricei
```

Complexitatea calcului inversei: $2n^3/3 + 2mn^2 = 8n^3/3$
COSTISITOR!

Varianta III

```
functie produs_Mv(n, M, v)
; calculează produsul dintre o matrice pătrată M și un vector coloană v
intreg n ; dimensiunea problemei
tablou real M[n][n] ; matricea, indici de la 1
tablou real v[n] ; vectorul
tablou real p[n] ; rezultatul p = Mv
; alte declarații
....
pentru i = 1, n
    pi = 0
    pentru j = 1, n
        •
        pi = pi + Mijvj
    •
    intoarce p
```

Complexitatea înmulțirii dintre o matrice și un vector: $2n^2$
Efortul total de calcul : $O(8n^3/3 + 2mn^2)$.

Există o variantă mai eficientă bazată pe **factorizarea** matricei coeficienților.

Notes

Notes

Ideea metodei

$$\mathbf{Ax} = \mathbf{b}, \tag{53}$$

$$\mathbf{A} = \mathbf{LU}, \quad \text{factorizare} \tag{54}$$

$$\mathbf{A} = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix} \quad \mathbf{L} = \begin{bmatrix} * & 0 & 0 & 0 & 0 & 0 \\ * & * & 0 & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 \\ * & * & * & * & 0 & 0 \\ * & * & * & * & * & 0 \\ * & * & * & * & * & * \end{bmatrix} \quad \mathbf{U} = \begin{bmatrix} * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 & * \end{bmatrix}$$

$$\mathbf{LUx} = \mathbf{b}. \tag{55}$$

Notes

Ideea metodei

$$\mathbf{Ax} = \mathbf{b}, \tag{56}$$

$$\mathbf{A} = \mathbf{LU}, \quad \text{factorizare} \tag{57}$$

$$\mathbf{LUx} = \mathbf{b}. \tag{58}$$

Notăm

$$\mathbf{y} = \mathbf{Ux}, \tag{59}$$

(89) ⇔

$$\begin{aligned} \mathbf{Ly} &= \mathbf{b}, & \text{substituție} & \text{ progresivă} \\ \mathbf{Ux} &= \mathbf{y}. & \text{substituție} & \text{ progresivă} \end{aligned} \tag{60}$$

Notes

Un exemplu simplu - pornind de la Gauss

$$\begin{cases} x_1 + 2x_2 - x_3 = -1, \\ -2x_1 + 3x_2 + x_3 = 0, \\ 4x_1 - x_2 - 3x_3 = -2. \end{cases} \quad (61)$$

$$\begin{cases} x_1 + 2x_2 - x_3 = -1, \\ 7x_2 - x_3 = -2, \\ -9x_2 + x_3 = 2. \end{cases} \quad (62)$$

$$\begin{cases} x_1 + 2x_2 - x_3 = -1, \\ 7x_2 - x_3 = -2, \\ -2/7x_3 = -4/7. \end{cases} \quad (63)$$

$$\begin{aligned} x_3 &= (-4/7)/(-2/7) = 2, \\ x_2 &= (-2 + x_3)/7 = 0, \\ x_1 &= -1 - 2x_2 + x_3 = 1. \end{aligned} \quad (64)$$

Notes

Un exemplu simplu - pornind de la Gauss

Factorizare

$$\mathbf{U} = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 7 & -1 \\ 0 & 0 & -2/7 \end{bmatrix}. \quad (65)$$

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ -2/1 & 1 & 0 \\ 4/1 & -9/7 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 4 & -9/7 & 1 \end{bmatrix}. \quad (66)$$

Verificare: $\mathbf{LU} = \mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & -1 \\ -2 & 3 & 1 \\ 4 & -1 & -3 \end{bmatrix}. \quad (67)$$

Notes

Un exemplu simplu - pornind de la Gauss

Substituție progresivă

$$Ly = b$$

$$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 4 & -9/7 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ -2 \end{bmatrix}, \quad (68)$$

$$\begin{cases} y_1 & = -1 \\ -2y_1 + y_2 & = 0 \\ 4y_1 - 9/7y_2 + y_3 & = -2 \end{cases} \quad (69)$$

$$\begin{aligned} y_1 &= -1 \\ y_2 &= 2y_1 = -2 \\ y_3 &= -2 - 4y_1 + 9/7y_2 = -4/7. \end{aligned} \quad (70)$$

Notes

Un exemplu simplu - pornind de la Gauss

Substituție regresivă

$$Ux = y$$

$$\begin{bmatrix} 1 & 2 & -1 \\ 0 & 7 & -1 \\ 0 & 0 & -2/7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \\ -4/7 \end{bmatrix}, \quad (71)$$

$$\begin{cases} x_1 + 2x_2 - x_3 & = -1 \\ 7x_2 - x_3 & = -2 \\ -2/7x_3 & = -4/7. \end{cases} \quad (72)$$

$$\begin{aligned} x_3 &= (-4/7)/(-2/7) = 2, \\ x_2 &= (-2 + x_3)/7 = 0, \\ x_1 &= -1 - 2x_2 + x_3 = 1. \end{aligned} \quad (73)$$

Notes

Variante de factorizare

Factorizare nu este unică. Variante standard:

- Doolittle: $l_{ij} = 1$ - se aplică la orice matrice nesingulară
- Crout: $u_{ij} = 1$ - se aplică la orice matrice nesingulară
- Cholesky: $\mathbf{L} = \mathbf{U}^T$ - se aplică doar matricelor simetrice și pozitiv definite

$$\begin{bmatrix} 3 & 2 \\ 6 & 1 \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}. \quad (74)$$

$$\begin{cases} l_{11}u_{11} = 3 \\ l_{12}u_{12} = 2 \\ l_{21}u_{11} = 6 \\ l_{21}u_{12} + l_{22}u_{22} = 1 \end{cases} \quad (75)$$

Sistemul devine determinat doar dacă fixăm oricare două valori. 53/91

Notes

Variante de factorizare

Exemplu:

$$\begin{bmatrix} 3 & 2 \\ 6 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ 0 & -3 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 6 & -3 \end{bmatrix} \begin{bmatrix} 1 & 2/3 \\ 0 & 1 \end{bmatrix}. \quad (76)$$

$$\begin{bmatrix} 9 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 2/3 & \sqrt{5}/3 \end{bmatrix} \begin{bmatrix} 3 & 2/3 \\ 0 & \sqrt{5}/3 \end{bmatrix}. \quad (77)$$

Notes

Algoritmul variantei Doolittle

$$\mathbf{A}_0 = \mathbf{A}, \quad (78)$$

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{E}_1 \mathbf{A}_0, \\ \mathbf{A}_2 &= \mathbf{E}_2 \mathbf{A}_1 = \mathbf{E}_2 \mathbf{E}_1 \mathbf{A}_0, \\ &\dots \end{aligned} \quad (79)$$

$$\begin{aligned} \mathbf{A}_{n-1} &= \mathbf{E}_{n-1} \mathbf{A}_{n-2} = \mathbf{E}_{n-1} \mathbf{E}_{n-2} \dots \mathbf{E}_2 \mathbf{E}_1 \mathbf{A}_0. \\ \mathbf{U} &= \mathbf{A}_{n-1}. \end{aligned} \quad (80)$$

$$\mathbf{E} = \mathbf{E}_{n-1} \mathbf{E}_{n-2} \dots \mathbf{E}_2 \mathbf{E}_1, \quad (81)$$

$$\mathbf{U} = \mathbf{E} \mathbf{A}. \quad (82)$$

Dar \mathbf{E} este nesingulară și:

$$\mathbf{L} = \mathbf{E}^{-1}. \quad (83)$$

55/91

Notes

Algoritmul variantei Doolittle

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & a'_{32} & a'_{33} \end{bmatrix} = \mathbf{E}_1 \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}. \quad (84)$$

$$\mathbf{E}_1 = \begin{bmatrix} 1 & 0 & 0 \\ -a_{21}/a_{11} & 1 & 0 \\ -a_{31}/a_{11} & 0 & 1 \end{bmatrix}. \quad (85)$$

$$\mathbf{E}_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ a_{21}/a_{11} & 1 & 0 \\ a_{31}/a_{11} & 0 & 1 \end{bmatrix}, \quad \mathbf{E}_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & a'_{32}/a'_{22} & 1 \end{bmatrix}. \quad (86)$$

$$\mathbf{E}^{-1} = \mathbf{E}_1^{-1} \mathbf{E}_2^{-1} \dots \mathbf{E}_{n-2}^{-1} \mathbf{E}_{n-1}^{-1}. \quad (87)$$

$$\mathbf{E}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ a_{21}/a_{11} & 1 & 0 \\ a_{31}/a_{11} & a'_{32}/a'_{22} & 1 \end{bmatrix} = \mathbf{L}. \quad (88)$$

56/91

Notes

Algoritmul variantei Doolittle

```
; etapa de eliminare din metoda Gauss cu memorarea opuselor elementelor  
; de multiplicare în triunghiul inferior al matricei  
pentru k = 1, n - 1 ; parcurge sub-etape ale eliminării  
    pentru i = k + 1, n ; parcurge liniile  
        p = -aik/akk ; element de multiplicare  
        pentru j = k + 1, n ; parcurge coloanele  
            aij = aij + p akj  
            •  
            aik = -p  
        •  
    •  
procedură factorizare_LU(n, a)  
; factorizează "in loc" matricea a  
; varianta Doolittle  
; declarații  
...  
pentru k = 1, n - 1 ; parcurge sub-etape ale eliminării  
    pentru i = k + 1, n ; parcurge liniile  
        aik = aik/akk ; element de multiplicare  
        pentru j = k + 1, n ; parcurge coloanele  
            aij = aij - aikakj ; Factorizare "pe loc": "A = L + U - I"  
        •  
    •  
retur
```

Notes

Calculul soluției după factorizare

$$\mathbf{LUx} = \mathbf{b}. \quad (89)$$

Notăm

$$\mathbf{y} = \mathbf{Ux}, \quad (90)$$

(89) \Leftrightarrow

$$\mathbf{Ly} = \mathbf{b}, \quad (91)$$

$$\mathbf{Ux} = \mathbf{y}. \quad (92)$$

" $\mathbf{y} = \mathbf{L}^{-1}\mathbf{b}$ " se rezolvă prin substituție progresivă:

$$\begin{cases} l_{11}y_1 = b_1, \\ l_{21}y_1 + l_{22}y_2 = b_2, \\ \dots \\ l_{n1}y_1 + l_{n2}y_2 + \dots + l_{nn}y_n = b_n, \end{cases} \Rightarrow \begin{cases} y_1 = b_1/l_{11}, \\ y_2 = (b_2 - l_{21}y_1)/l_{22}, \\ \dots \\ y_n = (b_n - \sum_{k=1}^{n-1} l_{nk}y_k)/l_{nn}. \end{cases}$$

Notes

Calculul soluției după factorizare

$$y_1 = b_1/l_{11}, \quad (94)$$

$$y_i = \left(b_i - \sum_{j=1}^{i-1} l_{ij}y_j \right) / l_{ii}, \quad i = 2, \dots, n. \quad (95)$$

" $\mathbf{x} = \mathbf{U}^{-1}\mathbf{y}$ " se rezolvă prin **substituție regresivă**:

$$x_n = y_n/u_{nn}, \quad (96)$$

$$x_i = \left(y_i - \sum_{j=i+1}^n u_{ij}x_j \right) / u_{ii}, \quad i = n-1, \dots, 1. \quad (97)$$

Notes

Calculul soluției după factorizare

```
procedură rezolvă_LU(n, a, b, x)
; rezolvă sistemul de ecuații  $ax = b$  prin factorizare LU
; matricea este presupusă a fi deja factorizată în loc
; varianta Doolittle
; declarații
...
; substituție progresivă
 $y_1 = b_1$  ; formula (94), unde  $l_{11} = 1$ 
pentru  $i = 2, n$ 
     $s = 0$ 
    pentru  $j = 1, i-1$ 
         $s = s + a_{ij}y_j$ ; formula (95), unde  $\mathbf{L}$  este memorat în a
    •
     $y_i = b_i - s$ ; deoarece  $l_{ii} = 1$ 
•
; substituție regresivă
 $x_n = y_n/a_{nn}$  ; formula (96), unde  $\mathbf{U}$  este memorat în a
pentru  $i = n-1, 1, -1$ 
     $s = 0$ 
    pentru  $j = i+1, n$ 
         $s = s + a_{ij}x_j$ 
    •
     $x_i = (y_i - s)/a_{ii}$ 
•
retur
```

Notes

Evaluarea algoritmului

Complexitate:

- Factorizarea propriu-zisă a: $T_f = O(2n^3/3)$
- Rezolvările: $T_s = O(2n^2)$.
- Necesarul de memorie: $M = O(n^2)$

Erori:

- Nu există erori de trunchiere;
- Erorile de rotunjire pot fi micșorate dacă se aplică strategii de pivotare.

Notes

Pivotare

Matrice de permutare:

- matrice care are exact un element egal cu 1 pe fiecare linie și pe fiecare coloană, și 0 în rest;
- inversa unei matrice de permutare este o matrice de permutare;
- produsul a două matrice de permutare este de asemenea o matrice de permutare;

Pivotarea pe linie

poate fi descrisă prin înmulțirea la stânga cu o matrice de permutare notată **P**.

Pivotarea pe coloane

poate fi descrisă prin înmulțirea la dreapta cu o matrice de permutare ce va fi notată cu **Q**.

Notes

Pivotare

$$\mathbf{A}_0 = \mathbf{P}_1 \mathbf{A}. \quad (98)$$

Presupunând că la fiecare etapă de eliminare se efectuează o permutare parțială, relațiile (79) se scriu

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{E}_1 \mathbf{A}_0 = \mathbf{E}_1 \mathbf{P}_1 \mathbf{A}, \\ \mathbf{A}_2 &= \mathbf{E}_2 \mathbf{P}_2 \mathbf{A}_1 = \mathbf{E}_2 \mathbf{P}_2 \mathbf{E}_1 \mathbf{P}_1 \mathbf{A}, \\ &\dots \\ \mathbf{A}_{n-1} &= \mathbf{E}_{n-1} \mathbf{P}_{n-1} \dots \mathbf{E}_2 \mathbf{P}_2 \mathbf{E}_1 \mathbf{P}_1 \mathbf{A}. \end{aligned} \quad (99)$$

Pivotare

$$\mathbf{U} = \mathbf{E}_{n-1} \mathbf{P}_{n-1} \dots \mathbf{E}_2 \mathbf{P}_2 \mathbf{E}_1 \mathbf{P}_1 \mathbf{A}. \quad (100)$$

$$\begin{aligned} \mathbf{U} &= \mathbf{E}_3 \mathbf{P}_3 \mathbf{E}_2 \mathbf{P}_2 \mathbf{E}_1 \mathbf{P}_1 \mathbf{A} = \\ &= \underbrace{\mathbf{E}_3}_{\mathbf{E}'_3} \underbrace{\mathbf{P}_3 \mathbf{E}_2 \mathbf{P}_3^{-1}}_{\mathbf{E}'_2} \underbrace{\mathbf{P}_3 \mathbf{P}_2 \mathbf{E}_1 \mathbf{P}_2^{-1} \mathbf{P}_3^{-1}}_{\mathbf{E}'_1} \mathbf{P}_3 \mathbf{P}_2 \mathbf{P}_1 \mathbf{A} = \\ &= \underbrace{\mathbf{E}'_3 \mathbf{E}'_2 \mathbf{E}'_1}_{\mathbf{L}^{-1}} \underbrace{\mathbf{P}_3 \mathbf{P}_2 \mathbf{P}_1}_{\mathbf{P}} \mathbf{A} \end{aligned} \quad (101)$$

Notes

Notes

Pivotare

Factorizarea cu pivotare pe linii:

$$PA = LU, \quad (102)$$

Factorizarea LU cu pivotare totală (rareori aplicată)

$$PAQ = LU, \quad (103)$$

Notes

Cazul sistemelor multiple

Rezolvate cu factorizare: $T = O(2n^3/3 + 2mn^2)$, mai mic decât cel necesar calculului inversei.

Efort de calcul pentru rezolvarea sistemelor multiple.

Nr. sisteme	Metoda	Complexitate T
1	Gauss	$2n^3/3 + n^2$
	LU	$2n^3/3 + 2n^2$
m - simultan	Gauss	$2n^3/3 + 2mn^2$
m - succesiv	folosind inversa	$8n^3/3 + 2mn^2$
	LU	$2n^3/3 + 2mn^2$

Notes

Varianta Cholesky

Dacă \mathbf{A} este simetrică, atunci este de dorit ca

$$\mathbf{U} = \mathbf{L}^T. \quad (104)$$

Aceasta se poate realiza doar dacă \mathbf{A} este pozitiv definită. Pp.

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T. \quad (105)$$

fie \mathbf{x} nenul; atunci $\mathbf{y} = \mathbf{L}^T\mathbf{x}$ va fi nenul

$$\mathbf{x}^T\mathbf{A}\mathbf{x} = \mathbf{x}^T\mathbf{L}\mathbf{L}^T\mathbf{x} = \mathbf{y}^T\mathbf{y} = \sum_{i=1}^n y_i^2 > 0.$$

Notes

Varianta Cholesky

Teoremă:

Dacă \mathbf{A} este o matrice simetrică și pozitiv definită, atunci factorizarea ei Cholesky există în mod unic, adică există în mod unic o matrice triunghiular inferioară \mathbf{L} cu elementele diagonale pozitive, astfel încât

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T$$

Notes

Modul de generare al matricei L

$$\begin{bmatrix} \alpha & \mathbf{a}^T \\ \mathbf{a} & \mathcal{A} \end{bmatrix} = \begin{bmatrix} \lambda & \mathbf{0} \\ \mathbf{I} & \mathcal{L} \end{bmatrix} \begin{bmatrix} \lambda & \mathbf{I}^T \\ \mathbf{0} & \mathcal{L}^T \end{bmatrix}. \quad (106)$$

$$\lambda = \sqrt{\alpha} \quad (107)$$

$$\mathbf{I} = \mathbf{a}/\lambda \quad (108)$$

$$\mathcal{L}\mathcal{L}^T = \mathcal{A} - \mathbf{I}\mathbf{I}^T. \quad (109)$$

Complementul Schur al lui α :

$$\mathbf{S} = \mathcal{A} - \mathbf{I}\mathbf{I}^T = \mathcal{A} - \mathbf{a}\mathbf{a}^T/\alpha. \quad (110)$$

Se poate demonstra că \mathbf{S} este simetrică și pozitiv definită și, în consecință $\mathcal{L}\mathcal{L}^T$ este factorizarea ei Cholesky.

Notes

Modul de generare al matricei L

$$\mathbf{A} = \mathbf{A}_0 = \begin{bmatrix} \lambda & \mathbf{0} \\ \mathbf{I} & \mathcal{L} \end{bmatrix} \begin{bmatrix} \lambda & \mathbf{I}^T \\ \mathbf{0} & \mathcal{L}^T \end{bmatrix} = \begin{bmatrix} \lambda & \mathbf{0} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \lambda & \mathbf{I}^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \mathbf{L}_1 \mathbf{A}_1 \mathbf{L}_1^T. \quad (111)$$

Similar,

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{L}_2 \mathbf{A}_2 \mathbf{L}_2^T, \\ &\vdots \\ \mathbf{A}_{n-1} &= \mathbf{L}_n \mathbf{A}_n \mathbf{L}_n^T, \end{aligned} \quad (112)$$

unde $\mathbf{A}_n = \mathbf{I}$.

$$\mathbf{A} = \underbrace{\mathbf{L}_1 \mathbf{L}_2 \cdots \mathbf{L}_n}_{\mathbf{L}} \underbrace{\mathbf{L}_n^T \mathbf{L}_{n-1}^T \cdots \mathbf{L}_1^T}_{\mathbf{L}^T} \quad (113)$$

Notes

Algoritm

$$l_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2}, \quad k = 1, \dots, n \quad (114)$$

$$l_{ik} = (a_{ik} - \sum_{j=1}^{k-1} l_{ij}l_{kj}) / l_{kk}, \quad i = k + 1, \dots, n. \quad (115)$$

Notes

Algoritm

```
procedură factorizare_LU_Cholesky(n, a, l)  
; factorizează matricea a, presupusă simetrică și pozitiv definită  
; întoarce matricea triunghiular inferioară l  
; varianta Cholesky  
; declarații  
...  
pentru k = 1, n ; parcurge sub-etape ale eliminării  
    pentru i = k, n ; calculează coloana, sub diagonală  
        s = aik  
        pentru j = 1, k - 1  
            s = s - lijlkj  
        •  
        dacă i = k  
            lkk = √s  
        altfel  
            lik = s / lkk  
        •  
    •  
retur
```

Notes

Algoritm

Efortul de calcul

$$\begin{aligned} T_e &\approx \sum_{k=1}^n [2k(n-k)] = -2 \sum_{k=1}^n [(n-k-n)(n-k)] = \\ &= -2 \left[\sum_{k=1}^n (n-k)^2 - n \sum_{k=1}^n (n-k) \right] = \\ &= -2 \left[\frac{(n-1)n(2n-1)}{6} - n \frac{n(n-1)}{2} \right] \approx -2 \left(\frac{2n^3}{6} - \frac{n^3}{2} \right) = \frac{n^3}{3}. \end{aligned}$$

Algoritmul Cholesky este întotdeauna stabil și nu are nevoie de pivotare. Aceasta se datorează proprietăților speciale ale matricei **A**, care fiind pozitiv definită este și diagonal dominantă.

Ce sunt matricele rare

Matrice rară = matrice care conține un număr foarte mare de elemente nenule.

O matrice care nu este rară se numește matrice **densă** sau **plină**.

Densitatea unei matrice = raportul dintre numărul de elemente nenule și numărul total de elemente al matricei.

Dacă, pentru o anumită matrice care are și elemente nule, se poate elabora un algoritm care exploatează această structură și care, este mai eficient decât algoritmul conceput pentru matricea plină, atunci aceasta este o matrice rară.

Notes

Notes

Formate de memorare a matricelor rare

Matrix Market: se memorează doar valorile nenule și "coordonatele" lor în matrice. <http://math.nist.gov/MatrixMarket>

Exemplu: tablou bidimensional de dimensiune $m \times n$:

$$M_{plin} = 8mn \text{ B}$$

$$M_{rar,coord} = 8 * n_{nz} + 4 * 2n_{nz} = 16n_{nz} \text{ B.}$$

$$\mathbf{M} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 5 & 1 \\ 2 & 3 & 0 & 7 \end{bmatrix} \Rightarrow \begin{cases} val & = [4 & 5 & 1 & 2 & 3 & 7] \\ r_idx & = [1 & 2 & 2 & 3 & 3 & 3] \\ c_idx & = [1 & 3 & 4 & 1 & 2 & 4] \end{cases}$$

Notes

Formate de memorare a matricelor rare

Formatul **Yale** sau **CRS - Compressed Row Storage**:

$$M_{rar,CRS} = 8n_{nz} + 4(m + 1) + 4n_{nz} = 12n_{nz} + 4(m + 1) \text{ B.}$$

$$\mathbf{M} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 5 & 1 \\ 2 & 3 & 0 & 7 \end{bmatrix} \Rightarrow \begin{cases} val & = [4 & 5 & 1 & 2 & 3 & 7] \\ r_ptr & = [1 & 2 & 4 & 7] \\ c_idx & = [1 & 3 & 4 & 1 & 2 & 3] \end{cases}$$

Similar, **CCS (Compressed Column Storage)**- cunoscut și sub numele **Harwell - Boeing**.

$$M_{rar,CCS} = 8n_{nz} + 4(n + 1) + 4n_{nz} = 12n_{nz} + 4(n + 1) \text{ B.}$$

$$\mathbf{M} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 5 & 1 \\ 2 & 3 & 0 & 7 \end{bmatrix} \Rightarrow \begin{cases} val & = [4 & 2 & 3 & 5 & 1 & 7] \\ c_ptr & = [1 & 3 & 4 & 5 & 7] \\ r_idx & = [1 & 3 & 3 & 2 & 2 & 3] \end{cases}$$

Notes

Formate de memorare a matricelor rare

Matricelor **bandă** (de exemplu matrice tridiagonală):

$$M = \begin{bmatrix} q_1 & r_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ p_2 & q_2 & r_2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & p_3 & q_3 & r_3 & \cdots & 0 & 0 & 0 \\ \cdots & & & & & & & \\ \cdots & & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & p_{n-1} & q_{n-1} & r_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & p_n & q_n \end{bmatrix}$$

Memorare cu ajutorul a trei vectori (**CDS - Compressed Diagonal Storage**):

$$M_{\text{rar}} = \begin{bmatrix} 0 & p_2 & p_3 & \cdots & p_{n-1} & p_n \\ q_1 & q_2 & q_3 & \cdots & q_{n-1} & q_n \\ r_1 & r_2 & r_3 & \cdots & r_{n-1} & 0 \end{bmatrix}$$

Notes

Metode directe pentru matrice rare

Gauss pentru matrice tridiagonală, matricea la subetapa k de eliminare:

$$\begin{bmatrix} * & * & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & * & * & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & * & * & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & q_k & r_k & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & p_{k+1} & q_{k+1} & r_{k+1} & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & p_{k+2} & q_{k+2} & r_{k+2} & \cdots & 0 & 0 \\ \cdots & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & p_n & q_n \end{bmatrix}$$

Un singur element de multiplicare $m = -p_{k+1}/q_k$.

Singura modificare suferind-o ecuația $k + 1$:

$q_{k+1} = q_{k+1} + m * r_k$, și termenul liber corespunzător.

Notes

Metode directe pentru matrice rare

Gauss pentru matrice tridiagonală, matricea după eliminare.

$$\begin{bmatrix}
 q_1 & r_1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
 0 & q_2 & r_2 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 0 & 0 & 0 & q_k & r_k & 0 & 0 & \dots & 0 & 0 \\
 0 & 0 & 0 & 0 & q_{k+1} & r_{k+1} & 0 & \dots & 0 & 0 \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & q_{n-1} & r_{n-1} \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & q_n
 \end{bmatrix}$$

Retrosubstituție

$$x_n = b_n/q_n, \tag{116}$$

$$q_i x_i + r_i x_{i+1} = b_i \Rightarrow x_i = (b_i - r_i x_{i+1})/q_i, \quad i = n - 1, \dots, 1.$$

(117)

Notes

Metode directe pentru matrice rare

```

procedură Gauss_tridiag(n, p, q, r, b, x)
; rezolvă sistemul algebric liniar ax = b prin metoda Gauss
; matricea a este tridiagonală, memorată în p, q, r
întreg n ; dimensiunea sistemului
tablou_real p[n], q[n], r[n] ; "matricea" coeficienților - indici de la 1
tablou_real b[n] ; vectorul termenilor liberi
tablou_real x[n] ; vectorul soluție
întreg i, k
; etapa de eliminare din metoda Gauss
pentru k = 1, n - 1
    m = -p_{k+1}/q_k ; parcurge sub-etape ale eliminării
    q_{k+1} = q_{k+1} + mr_k ; element de multiplicare
    b_{k+1} = b_{k+1} + mb_k ; modifică element în linia k + 1
    ; modifică termenul liber al ecuației k + 1
; etapa de retrosubstituție
x_n = b_n/q_n
pentru i = n - 1, 1, -1
    x_i = (b_i - r_i x_{i+1})/q_i
retur

```

$$T = O(8n), M = O(5n).$$

Notes

Metode directe pentru matrice rare

- Pentru matrice rare fără o structură particulară, algoritmi trebuie adaptați memorării de tip CRS sau CCS.
- La eliminare matricea se poate umple, a.î. pivotarea urmărește nu numai stabilitatea numerică, ci și minimizarea umplerilor, adică a elementelor nenule nou apărute.
- La matrice rare inversarea este practic imposibilă datorită fenomen de **umplere**.

Notes

Metode directe pentru matrice rare

- Factorizarea unei matrice rare poate salva raritatea dacă matricea are o anumită structură.

$$A_1 = \begin{bmatrix} * & 0 & \dots & 0 & 0 & * \\ 0 & * & \dots & 0 & 0 & * \\ \dots & & & & & \\ 0 & 0 & \dots & * & 0 & * \\ 0 & 0 & \dots & 0 & * & * \\ * & * & \dots & * & * & * \end{bmatrix} \quad A_2 = \begin{bmatrix} * & * & * & \dots & * & * \\ * & * & 0 & \dots & 0 & 0 \\ * & 0 & * & \dots & 0 & 0 \\ \dots & & & & & \\ * & 0 & \dots & 0 & * & 0 \\ * & 0 & \dots & 0 & 0 & * \end{bmatrix}$$

Matricea A_1 are factorii LU rari, în timp ce matricea A_2 are factorii LU plini.

Structura matricei joacă deci un rol important în conceperea algoritmului de rezolvare.

Notes

Referințe

- [Ciuprina13a] Gabriela Ciuprina - Algoritmi numerici pentru calcule științifice în ingineria electrică , Editura MatrixROM, 2013, pag 51-66.

disponibilă la http://www.lmn.pub.ro/~gabriela/books/AlgNr_MatrixRom2013.pdf

- [Cheney00] Ward Cheney and David Kincaid, *Numerical Mathematics and Computing*, Brooks/Cole publishing Company,2000. (Capitolul *Systems of Linear Equations*)

O pot împrumuta, la cerere.

Referințe

- [Davis06] Timothy Davis, *Direct methods for sparse linear systems*, SIAM 2006.
- [Davis16] Timothy A. Davis, Sivasankaran Rajamanickam, and Wissam M. Sid-Lakhdar, *A survey of direct methods for sparse linear systems*, 2016, disponibilă la



http://faculty.cse.tamu.edu/davis/publications_files/survey_tech_report.pdf

Notes

Notes








Referințe

Pachete existente: ([Davis16])

BCSLIB-EXT Ashcraft (1995), Ashcraft et al. (1998), Pierce and Lewis (1997), analytics.com **BSMP** Bank and Smith (1987), www.netlib.org/linalg/bsmp.f **CHOLMOD** Chen et al. (2008), suitesparse.com **CSparse** Davis (2006), suitesparse.com **DSPACK** Heath and Raghavan (1995) (1997), Raghavan (2002), www.cse.psu.edu/?raghavan. Also CAPSS.Elemental Poulson, libelemental.org **ESSL** www.ibm.com **GPLU** Gilbert and Peierls (1988), www.mathworks.com **IMSL** www.roguewave.com **KLU** Davis and Palamadai Natarajan (2010), suitesparse.com **LDL** Davis (2005), suitesparse.com **MA38** Davis and Duff (1997), www.hsl.rl.ac.uk **MA41** Amestoy and Duff (1989), www.hsl.rl.ac.uk **MA42**, **MA43** Duff and Scott (1996), www.hsl.rl.ac.uk. Successor to **MA32**. **HSL MP42**, **HSL MP43** Scott (2001a) (2001b) (2003), www.hsl.rl.ac.uk. Also **MA52** and **MA72**. **MA46** Damhaug and Reid (1996), www.hsl.rl.ac.uk **MA47** Duff and Reid (1996b), www.hsl.rl.ac.uk **MA48**, **HSL MA48** Duff and Reid (1996a), www.hsl.rl.ac.uk. Successor to **MA28**. **HSL MP48** Duff and Scott (2004), www.hsl.rl.ac.uk **MA49** Amestoy et al. (1996b), www.hsl.rl.ac.uk **MA57**, **HSL MA57** Duff (2004), www.hsl.rl.ac.uk **MA62**, **HSL MP62** Duff and Scott (1999), Scott (2003), www.hsl.rl.ac.uk **MA67** Duff et al. (1991), www.hsl.rl.ac.uk **HSL MA77** Reid and Scott (2009b), www.hsl.rl.ac.uk **HSL MA78** Reid and Scott (2009a), www.hsl.rl.ac.uk **HSL MA86**, **HSL MA87** Hogg et al. (2010)   85/91

Notes

Referințe

Mathematica Wolfram, Inc., www.wolfram.com **MATLAB** Gilbert et al. (1992), www.mathworks.com **Meschach** Steward and Leyk, www.netlib.org/c/meschach **MUMPS** Amestoy et al. (2000), Amestoy et al. (2001a), Amestoy et al. (2006), www.enseeiht.fr/apo/MUMPS **NAG** www.nag.com **NSPIV** Sherman (1978b) (1978a), www.netlib.org/toms/533 **Oblio** Dobrian, Kumfert and Pothen (2000), Dobrian and Pothen (2005), www.cs.purdue.edu/homes/apothen **PARDISO** Schenk and Gartner (2004), Schenk, Gartner and Fichtner (2000), www.pardiso-project.org **PaStiX** Henon et al. (2002), www.labri.fr/ramet/pastix **QR MUMPS** Buttari (2013), buttari.perso.enseeiht.fr **PSPASES** Gupta et al. (1997), www.cs.umn.edu/mjoshi/pspases **Quern** Bridson, www.cs.ubc.ca/~rbridson/quern **S+** Fu et al. (1998), Shen et al. (2000), www.cs.ucsb.edu/projects/s+ **Sparse 1.4** Kundert (1986), sparse.sourceforge.net **SPARSPAK** Chu et al. (1984), George and Liu (1979a) (1981) (1999), www.cs.uwaterloo.ca/~jageorge **SPOOLES** Ashcraft and Grimes (1999), www.netlib.org/linalg/spooles **SPRAL** SSIDS Hogg et al. (2016), www.numerical.rl.ac.uk/spral **SuiteSparseQR** Yeralan et al. (2016), Foster and Davis (2013), suitesparse.com **SuperLLT** Ng and Peyton (1993a), <http://crd.lbl.gov/EGNg> **SuperLU** Demmel et al. (1999a), crd.lbl.gov/xiaoye/SuperLU **SuperLU DIST** Li and Demmel (2003), crd.lbl.gov/xiaoye/SuperLU **SuperLU** MT Demmel et al. (1999b), crd.lbl.gov/xiaoye/SuperLU        86/91

Notes

Referințe

TAUCS Rotkin and Toledo (2004), www.tau.ac.il/~stoledo/taucs **UMFPACK** Davis (2004b) Davis and Duff (1997) (1999), suitesparse.com **WSMP** Gupta (2002a), Gupta et al. (1997), www.cs.umn.edu/~agupta/wsm **Y12M** Zlatev, Wasniewski and Schaumburg (1981), www.netlib.org/y12m **YSMP** Eisenstat et al. (1977) (1982), Yale Librarian, New Haven, CT

Notes

Referințe

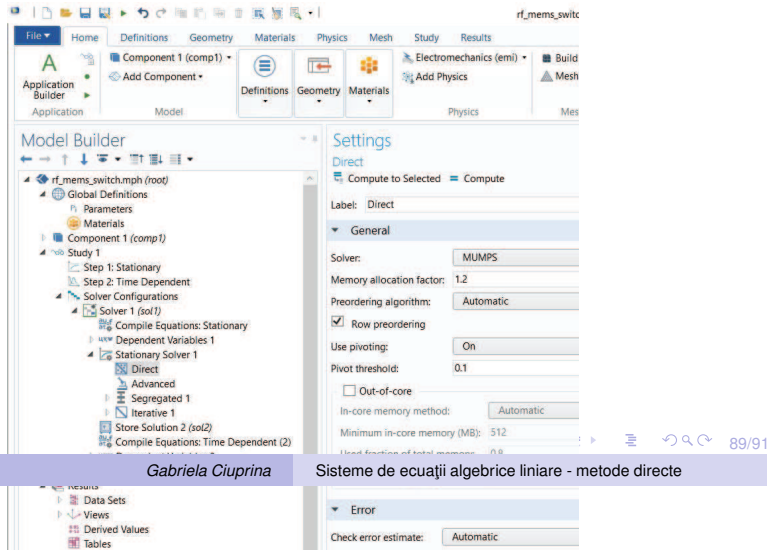
42 de cursuri pe youtube ale lui T. Davis, primul este aici <https://www.youtube.com/watch?v=1dGRTOWBkQs>



Notes

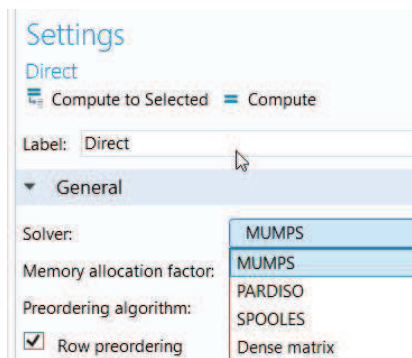
Pe scurt

Fiți atenți la astfel de informații (capturi din COMSOL)



Pe scurt

Fiți atenți la astfel de informații



Notes

Notes
