Sisteme de ecuaţii algebrice liniare - metode directe

Prof.dr.ing. Gabriela Ciuprina

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

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

Cuprins

Formularea problemei

Sistem de n ecuaţii algebrice liniare cu n necunoscute:

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

Formularea problemei

Se dă matricea coeficienţilor

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

şi vectorul termenilor liberi

$$\mathbf{b} = \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix}^T \in \mathbb{R}^n, \tag{3}$$

se cere să se rezolve sistemul

$$\mathbf{Ax} = \mathbf{b}, \tag{4}$$

unde **x** este soluţia

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T \in \mathbb{R}^n. \tag{5}$$

Buna formulare matematică

Problema este bine formulată din punct de vedere matematic (soluţia există şi este unică)

$\Leftrightarrow$

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

Se scrie formal:

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

trebuie citită ca:

"**x** *este soluţia sistemului algebric liniar* $\mathbf{Ax} = \mathbf{b}$"

şi **NU** "*se calculează inversa matricei* **A** *care se înmulţeşte cu vectorul* **b**".

Condiţionarea problemei

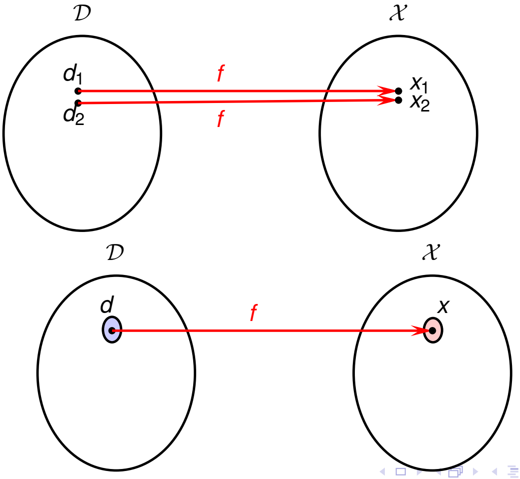
Condiţionarea

se referă la comportarea problemei matematice la perturbaţii ale datelor.

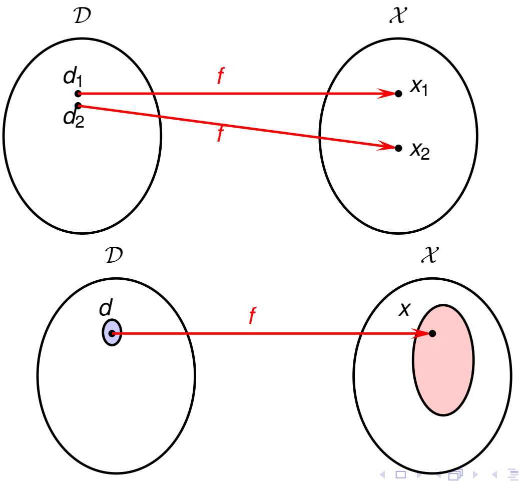
Problemă matematică f formulată explicit:

$$\text{Fie} \quad f : \mathcal{D} \to \mathcal{X} \quad \text{şi} \quad \mathbf{d} \in \mathcal{D}.$$
$$\text{Să se găsească} \quad \mathbf{x} \in \mathcal{X} \quad \text{astfel încât} \quad f(\mathbf{d}) = \mathbf{x}. \qquad (6)$$

Reprezentări intuitive - problemă bine condiţionată

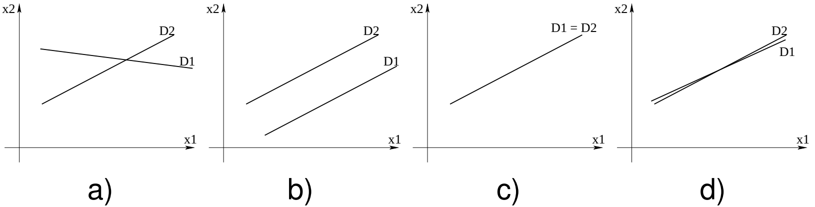


Reprezentări intuitive - problemă prost condiţionată



Condiţionarea - intuitiv ($n = 2$)

Nu orice problemă de rezolvare a unui sistem de ecuaţii algebrice liniare care este bine formulată matematic este şi bine condiţionată.



a) Problemă matematică bine formulată şi bine condiţionată. b) Problemă matematică prost formulată (nu există soluţie). c) Problemă matematică prost formulată (are o infinitate de soluţii). d) Problemă matematică bine formulată şi slab condiţionată.

Numărul de condiţionare

Fie

$$\mathbf{Ax} = \mathbf{b}, \tag{7}$$

unde $\mathbf{x}$ este soluţia exactă şi presupunem o perturbaţie a soluţiei $\mathbf{x} + \mathbf{e_x}$, corespunzătoare unei perturbaţii a datelor $\mathbf{b} + \mathbf{e_b}$:

$$\mathbf{A}(\mathbf{x} + \mathbf{e_x}) = \mathbf{b} + \mathbf{e_b}, \tag{8}$$

$\Rightarrow$

$$\mathbf{Ae_x} = \mathbf{e_b}. \tag{9}$$

Notăm erorile relativă a soluţiei şi a datelor:

$$\varepsilon_x = \frac{\|\mathbf{e_x}\|}{\|\mathbf{x}\|}, \quad \varepsilon_b = \frac{\|\mathbf{e_b}\|}{\|\mathbf{b}\|}. \tag{10}$$

$\Rightarrow$

$$\mathbf{e_x} = \mathbf{A}^{-1}\mathbf{e_b} \Rightarrow \|\mathbf{e_x}\| = \|\mathbf{A}^{-1}\mathbf{e_b}\| \leq \|\mathbf{A}^{-1}\|\|\mathbf{e_b}\|. \tag{11}$$

Numărul de condiţionare

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

Un majorant pentru eroarea asupra soluţiei

$$\varepsilon_x = \frac{\|\mathbf{e_x}\|}{\|\mathbf{x}\|} \leq \frac{\|\mathbf{A}^{-1}\|\|\mathbf{e_b}\|}{\frac{\|\mathbf{b}\|}{\|\mathbf{A}\|}} = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|\varepsilon_b. \tag{13}$$

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

număr de condiţionare la inversare al matricei **A.**

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

Numărul de condiţionare - altă demonstraţie

Pornind de la definiţia generală:

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

sau, scris mai simplu în ipoteza unor variaţii infinitezimale

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

Dacă *f* este derivabilă, atunci

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

Numărul de condiţionare - altă demonstraţie

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

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

Marginea inferioară pentru eroarea asupra soluţiei

$$\|\mathbf{e_b}\| = \|\mathbf{A}\mathbf{e_x}\| \leq \|\mathbf{A}\|\|\mathbf{e_x}\| \quad \Rightarrow \quad \|\mathbf{e_x}\| \geq \frac{\|\mathbf{e_b}\|}{\|\mathbf{A}\|}. \tag{20}$$

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

$\Rightarrow$

$$\varepsilon_x = \frac{\|\mathbf{e_x}\|}{\|\mathbf{x}\|} \geq \frac{\|\mathbf{e_b}\|}{\|\mathbf{A}\|\|\mathbf{A}^{-1}\|\|\mathbf{b}\|} = \frac{\varepsilon_b}{\kappa(\mathbf{A})}. \tag{22}$$

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

Numărul de condiţionare - proprietăţi

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

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

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

- Numărul de condiţionare este o proprietate a matricei şi nu are legătură nici cu metoda de rezolvare propriu-zisă, nici cu erorile de rotunjire care apar în mediul de calcul.
  În practică:
  Dacă $\kappa(\mathbf{A}) > 1/\mathtt{eps}$ problema se consideră slab condiţionată.

Perturbaţii în matricea coeficienţilor

$$(\mathbf{A} + \mathbf{e_A})(\mathbf{x} + \mathbf{e_x}) = \mathbf{b}. \tag{25}$$

$$\mathbf{A}\mathbf{e_x} = -\mathbf{e_A}(\mathbf{x} + \mathbf{e_x}), \tag{26}$$

$$\|\mathbf{e_x}\| = \| - \mathbf{A}^{-1}\mathbf{e_A}(\mathbf{x} + \mathbf{e_x})\| \leq \|\mathbf{A}^{-1}\|\|\mathbf{e_A}\|\|\mathbf{x} + \mathbf{e_x}\|. \tag{27}$$

$$\varepsilon_X \approx \frac{\|\mathbf{e_x}\|}{\|\mathbf{x} + \mathbf{e_x}\|} \leq \|\mathbf{A}^{-1}\|\|\mathbf{e_A}\| = \|\mathbf{A}^{-1}\|\|\mathbf{A}\|\frac{\|\mathbf{e_A}\|}{\|\mathbf{A}\|} = \kappa(\mathbf{A})\varepsilon_A. \tag{28}$$

Deoarece $\|\mathbf{x} + \mathbf{e_x}\| \leq \|\mathbf{x}\| + \|\mathbf{e_x}\|$, rezultă că $\|\mathbf{x}\| \geq \|\mathbf{x} + \mathbf{e_x}\| - \|\mathbf{e_x}\|$.

Perturbaţii în matricea coeficienţilor

Dacă presupunem că

$$\|\mathbf{x} + \mathbf{e_x}\| - \|\mathbf{e_x}\| > 0, \tag{29}$$

$\Rightarrow$

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

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

Clasificarea metodelor

1. **Metode directe** - găsesc soluţia teoretică a problemei într-un număr finit de paşi. (Gauss, factorizare LU)
2. **Metode iterative** - generează un şir de aproximaţii ale soluţiei care se doreşte a fi convergent către soluţia exactă.
   - **staţionare:** Jacobi, Gauss-Seidel, SOR, SSOR
   - **nestaţionare (semiiterative):** gradienţi conjugaţi (GC), reziduu minim (MINRES), reziduu minim generalizat (GMRES), gradienţi biconjugaţi (BiGC), etc.

Ideea metodei Gauss

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad \underbrace{\Leftrightarrow}_{\text{eliminare}} \quad \mathbf{U}\mathbf{x} = \mathbf{b}' \quad \underbrace{\Rightarrow}_{\text{subst.regresivă}} \quad \mathbf{x} = \mathbf{U}^{-1}\mathbf{b}'. \tag{31}$$

Un exemplu simplu

$$\left\{\begin{array}{rcrcrcr} x_1 & + & 2x_2 & - & x_3 & = & -1, \\ -2x_1 & + & 3x_2 & + & x_3 & = & 0, \\ 4x_1 & - & x_2 & - & 3x_3 & = & -2. \end{array}\right. \tag{32}$$

$$\left\{\begin{array}{rcrcrcr} x_1 & + & 2x_2 & - & x_3 & = & -1, \\ & & 7x_2 & - & x_3 & = & -2, \\ & & -9x_2 & + & x_3 & = & 2. \end{array}\right. \tag{33}$$

$$\left\{\begin{array}{rcrcrcr} x_1 & + & 2x_2 & - & x_3 & = & -1, \\ & & 7x_2 & - & x_3 & = & -2, \\ & & & - & 2/7x_3 & = & -4/7. \end{array}\right. \tag{34}$$

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

Algoritmul metodei Gauss - etapa de eliminare

$$\underset{\mathbf{A}_0}{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}} \rightarrow \underset{\mathbf{A}_1}{\begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{bmatrix}} \rightarrow \underset{\mathbf{A}_2}{\begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \end{bmatrix}} \rightarrow \underset{\cdots}{\cdots}$$

Eliminare în metoda Gauss: pentru un sistem de dimensiune n există $n-1$ sub-etape de eliminare. La final matricea este superior triunghiulară. Matricea iniţială este notată $\mathbf{A}_0$ iar matricea superior triunghiulară obţinută este notată $\mathbf{A}_{n-1}$. În realitate, transformările sunt memorate "în loc", în acelaşi tablou bidimensional.

Algoritmul metodei Gauss - etapa de eliminare

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

; anularea elementului a_{21}
$p = -a_{21}/a_{11}$; element de multiplicare
<u>pentru</u> $j = 1, n$; parcurge coloanele
    $a_{2j} = a_{2j} + pa_{1j}$
•
$b_2 = b_2 + pb_1$

$2 \rightarrow i$ inserată într-un ciclu cu contor

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} + p a_{1j}
      •
      b_i = b_i + p b_1
•
```

OBS: În ciclul în j contorul începe cu valoarea 2.
$1 \to k, 2 \to k+1$ inserate într-un ciclu cu contor.

Algoritmul metodei Gauss - etapa de eliminare

Secvenţa 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 + p a_kj
        •
        b_i = b_i + p b_k
    •
•
```

Algoritmul metodei Gauss - substituţie regresivă

$$\left\{\begin{array}{rrr} a_{11}x_1 + & a_{12}x_2 + \cdots + & a_{1n}x_n = b_1, \\ & a_{22}x_2 + \cdots + & a_{2n}x_n = b_2, \\ & \cdots & \\ & & a_{nn}x_n = b_n. \end{array}\right. \tag{36}$$

$$x_n = b_n / a_{nn}, \tag{37}$$

$$a_{ii}x_i + a_{i,i+1}x_{i+1} + \cdots + a_{in}x_n = b_i, \tag{38}$$

$\Rightarrow$

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

Algoritmul metodei Gauss - substituţie regresivă

$$x_n = b_n / a_{nn}, \tag{40}$$

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

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

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 coeficienţilor - indici de la 1
tablou real b[n]                  ; vectorul termenilor liberi
tablou real x[n]                  ; vectorul soluţie
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 + p a_kj
                •
                b_i = b_i + p b_k
        •
•
```

Algoritmul metodei Gauss

```
; etapa de retrosubstituţie
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 îmbunătăţit prin folosirea la fiecare etapă de eliminare a unei strategii de pivotare.

Evaluarea algoritmului

Din punct de vedere al timpului de calcul:

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

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

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

Din punct de vedere al necesarului de memorie:

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

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 obţine dacă se includ în algoritm strategii de pivotare.

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

Strategii de pivotare

Pivoţi

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

Determinantul sistemului = produsul pivoţilor.

$\Rightarrow$

Problema este bine formulată matematic $\Leftrightarrow$ toţi pivoţii sunt nenuli.

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

Pivotare

Operaţie de permutare care urmăreşte obţinerea valorilor nenule pentru pivoţi.

Trebuie făcută înainte de calculul multiplicatorului.

Strategii de pivotare

Strategii de pivotare:

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

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

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

Algoritmul pivotării pe linii

$p = 0$
<u>pentru</u> $i = k, n$; parcurge coloana k
    <u>dacă</u> $|a_{ik}| > p$ <u>atunci</u>
        $l = i$; memorează poziţia potenţialului pivot
        $p = |a_{ik}|$
    •
•
<u>dacă</u> $p = 0$ <u>atunci</u>
    <u>scrie</u> "problema este prost formulată matematic"
<u>altfel</u>
    <u>pentru</u> $j = k, n$; permută linia l cu linia k
        $p = a_{kj}$
        $a_{kj} = a_{lj}$
        $a_{lj} = p$
    •
    $p = b_k$; permută termenii liberi
    $b_k = b_l$
    $b_l = p$
•

Pivotarea parţială (pe linii)

Observaţii:

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 numeşte *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 parţială clasică. Pentru detalii, consultaţi [Cheney].

Avantajele pivotării

Pivotarea

1. necesară dacă pe parcursul algoritmului se întâlneşte un pivot nul.
2. efect benefic asupra stabilitătii şi acurateţii.

Avantajele pivotării

Exemplu

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

soluţia corectă $(x, y) \approx (-1, 1)$.
Gauss şi presupunem că eps = 10^{-16}:

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

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

Rezultatul final: $(x, y) = (0, 1)$ extrem de eronat.
Explicaţie: $\kappa(\mathbf{A}) \approx 2.6$, dar $\kappa(\mathbf{U}) = 10^{40}$!

Metoda Gauss - Concluzii

- Este o metodă directă - găseşte soluţia teoretică a problemei într-un număr finit de paşi.
- Calculele sunt afectate de erori de rotunjire $\Rightarrow$ nu se obţine soluţia exactă, ci o aproximare a ei.
- Se transformă sistemului de ecuaţii într-unul echivalent din punct de vedere al soluţiei (Δ sup.), mult mai uşor de rezolvat (subs. regr.).
- Pivotarea: esenţială pentru a asigura pivoţi nenuli; utilă pentru a creşte stabilitatea algoritmului şi acurateţea soluţiei.

Metoda Gauss - Concluzii

- Pivotarea parţială are un efort de implementare nesemnificativ.
- Pivotarea totală este rareori aplicată deoarece duce la o creştere semnificativă a timpului de calcul, nerealizând decât o îmbunătăţire nesemnificativă a acurateţii soluţiei.
- Dezavantajul metodei Gauss: în anumite situaţii, efortul de generare a problemei echivalente (eliminarea) este mare sau, necesarul de memorie poate deveni extrem de mare.

Formularea problemei

Fie m sisteme de ecuaţii algebrice liniare

$$\mathbf{A}\mathbf{x}^{(1)} = \mathbf{b}^{(1)}, \quad \mathbf{A}\mathbf{x}^{(2)} = \mathbf{b}^{(2)}, \quad \cdots, \quad \mathbf{A}\mathbf{x}^{(m)} = \mathbf{b}^{(m)}, \tag{47}$$

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

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

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

Se cere să se rezolve sistemul

$$\mathbf{A}\mathbf{X} = \mathbf{B}. \tag{50}$$

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.

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 coeficienţilor - indici de la 1
tablou real B[n][m]                ; matricea termenilor liberi
tablou real X[n][m]                ; matricea soluţie
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 = −a_ik / a_kk   ; element de multiplicare
                pentru j = k + 1, n  ; parcurge coloanele
                        a_ij = a_ij + p a_kj
                •
                pentru j = 1, m    ; parcurge coloanele termenilor liberi
                        b_ij = b_ij + p b_kj
                •
        •
•
```

Varianta II

```
; etapa de retrosubstituţie
pentru k = 1, m
        x_nk = b_nk / a_nn
        pentru i = n − 1, 1, −1
                s = 0
                pentru j = i + 1, n
                        s = s + a_ij x_jk
                •
                x_ik = (b_ik − s) / a_ii
        •
•
retur
```

Varianta II

Efort de calcul

$$\begin{aligned} T_e &= \sum_{k=1}^{n-1}[2(n-k)+2m+1](n-k) \approx \sum_{k=1}^{n-1}[2(n-k)^2+2m(n-k)] = \\ &= 2\frac{(n-1)n(2n-1)}{6} + 2m\frac{n(n-1)}{2} \approx \frac{2n^3}{3} + mn^2. \end{aligned} \tag{51}$$

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

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

Varianta III

Varianta III - rezolvarea succesivă a sistemelor folosind calculul inversei

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

Varianta III

```
funcţie invA(n, a)
; calculează inversa matricei a
întreg n                        ; dimensiunea matricei
tablou real a[n][n]             ; matricea, indici de la 1
; alte declaraţii
....
pentru i = 1, n
        pentru j = 1, n
                B_ij = 0
        •
        B_ii = 1
•
Gauss_multiplu(n, n, a, B, X)
întoarce X                      ; X este inversa matricei
```

Complexitatea calculului inversei: $2n^3/3 + 2mn^2 = 8n^3/3$

COSTISITOR!

Varianta III

```
funcţie produs_Mv (n, M, v)
; calculează produsul dintre o matrice pătrată M şi un vector coloană v
întreg 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 declaraţii
....
pentru i = 1, n
        p_i = 0
        pentru j = 1, n
                p_i = p_i + M_ij v_j
        •
•
întoarce p
```

Complexitatea inmulţirii dintre o matrice şi un vector: $2n^2$
Efortul total de calcul : $O(8n^3/3 + 2mn^2)$.
Există o variantă mai eficientă bazată pe factorizarea matricei coeficienţilor.

Ideea metodei

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

$$\mathbf{A} = \mathbf{LU}, \qquad \textcolor{red}{\text{factorizare}} \tag{54}$$

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

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

Ideea metodei

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

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

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

Notăm

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

(89) $\Leftrightarrow$

$$\begin{array}{lll} \mathbf{Ly} = \mathbf{b}, & \text{substituţie} & \text{progresivă} \\ \mathbf{Ux} = \mathbf{y}. & \text{substituţie} & \text{progresivă} \end{array} \tag{60}$$

Un exemplu simplu - pornind de la Gauss

$$\left\{\begin{array}{rcrcrcr} x_1 & + & 2x_2 & - & x_3 & = & -1, \\ -2x_1 & + & 3x_2 & + & x_3 & = & 0, \\ 4x_1 & - & x_2 & - & 3x_3 & = & -2. \end{array}\right. \tag{61}$$

$$\left\{\begin{array}{rcrcrcr} x_1 & + & 2x_2 & - & x_3 & = & -1, \\ & & 7x_2 & - & x_3 & = & -2, \\ & & -9x_2 & + & x_3 & = & 2. \end{array}\right. \tag{62}$$

$$\left\{\begin{array}{rcrcrcr} x_1 & + & 2x_2 & - & x_3 & = & -1, \\ & & 7x_2 & - & x_3 & = & -2, \\ & & & - & 2/7x_3 & = & -4/7. \end{array}\right. \tag{63}$$

$$\begin{array}{l} x_3 = (-4/7)/(-2/7) = 2, \\ x_2 = (-2 + x_3)/7 = 0, \\ x_1 = -1 - 2x_2 + x_3 = 1. \end{array} \tag{64}$$

Un exemplu simplu - pornind de la Gauss

Factorizare

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

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

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

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

Un exemplu simplu - pornind de la Gauss

Substituţie progresivă

$\mathbf{Ly} = \mathbf{b}$

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

$$\left\{ \begin{array}{rcrcrcr} y_1 & & & & & = & -1 \\ -2y_1 & + & y_2 & & & = & 0 \\ 4y_1 & - & 9/7y_2 & + & y_3 & = & -2 \end{array} \right. \tag{69}$$

$$\begin{array}{l} y_1 = -1 \\ y_2 = 2y_1 = -2 \\ y_3 = -2 - 4y_1 + 9/7y_2 = -4/7. \end{array} \tag{70}$$

Un exemplu simplu - pornind de la Gauss

Substituţie regresivă

$\mathbf{Ux} = \mathbf{y}$

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

$$\left\{ \begin{array}{rcl} x_1 + 2x_2 - x_3 & = & -1 \\ 7x_2 - x_3 & = & -2 \\ -2/7x_3 & = & -4/7. \end{array} \right. \tag{72}$$

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

Variante de factorizare

Factorizare nu este unică. Variante standard:

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

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

$$\left\{ \begin{array}{r} l_{11}u_{11} = 3 \\ l_{12}u_{12} = 2 \\ l_{21}u_{11} = 6 \\ l_{21}u_{12} + l_{22}u_{22} = 1 \end{array} \right. \tag{75}$$

Sistemul devine determinat doar dacă fixăm oricare două valori.

Variante de factorizare

Exemplu:

$$\left[\begin{array}{cc} 3 & 2 \\ 6 & 1 \end{array}\right] = \left[\begin{array}{cc} 1 & 0 \\ 2 & 1 \end{array}\right] \left[\begin{array}{cc} 3 & 2 \\ 0 & -3 \end{array}\right] = \left[\begin{array}{cc} 3 & 0 \\ 6 & -3 \end{array}\right] \left[\begin{array}{cc} 1 & 2/3 \\ 0 & 1 \end{array}\right]. \tag{76}$$

$$\left[\begin{array}{cc} 9 & 2 \\ 2 & 1 \end{array}\right] = \left[\begin{array}{cc} 3 & 0 \\ 2/3 & \sqrt{5}/3 \end{array}\right] \left[\begin{array}{cc} 3 & 2/3 \\ 0 & \sqrt{5}/3 \end{array}\right]. \tag{77}$$

Algoritmul variantei Doolittle

$$\mathbf{A}_0 = \mathbf{A}, \tag{78}$$

$$\begin{aligned} &\mathbf{A}_1 = \mathbf{E}_1\mathbf{A}_0, \\ &\mathbf{A}_2 = \mathbf{E}_2\mathbf{A}_1 = \mathbf{E}_2\mathbf{E}_1\mathbf{A}_0, \\ &\dots \\ &\mathbf{A}_{n-1} = \mathbf{E}_{n-1}\mathbf{A}_{n-2} = \mathbf{E}_{n-1}\mathbf{E}_{n-2}\cdots\mathbf{E}_2\mathbf{E}_1\mathbf{A}_0. \end{aligned} \tag{79}$$

$$\mathbf{U} = \mathbf{A}_{n-1}. \tag{80}$$

$$\mathbf{E} = \mathbf{E}_{n-1}\mathbf{E}_{n-2}\cdots\mathbf{E}_2\mathbf{E}_1, \tag{81}$$

$$\mathbf{U} = \mathbf{E}\mathbf{A}. \tag{82}$$

Dar **E** este nesingulară şi:

$$\mathbf{L} = \mathbf{E}^{-1}. \tag{83}$$

Algoritmul variantei Doolittle

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

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

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

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

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

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 = −a_ik / a_kk                   ; element de multiplicare
        pentru j = k + 1, n                ; parcurge coloanele
            a_ij = a_ij + p a_kj
        •
        a_ik = −p
    •
•

procedură factorizare_LU(n, a)
; factorizează "in loc" matricea a
; varianta Doolittle
; declaraţii
...
pentru k = 1, n − 1                        ; parcurge sub-etape ale eliminării
    pentru i = k + 1, n                    ; parcurge liniile
        a_ik = a_ik / a_kk                 ; element de multiplicare
        pentru j = k + 1, n                ; parcurge coloanele
            a_ij = a_ij − a_ik a_kj        ; Factorizare "pe loc" : "A = L + U − I"
        •
    •
•
retur
```

Calculul soluţiei după factorizare

$$\mathbf{LUx} = \mathbf{b}. \tag{89}$$

Notăm

$$\mathbf{y} = \mathbf{Ux}, \tag{90}$$

(89) $\Leftrightarrow$

$$\mathbf{Ly} = \mathbf{b}, \tag{91}$$

$$\mathbf{Ux} = \mathbf{y}. \tag{92}$$

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

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

Calculul soluţiei după factorizare

$$y_1 = b_1 / l_{11}, \tag{94}$$

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

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

$$x_n = y_n / u_{nn}, \tag{96}$$

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

Calculul soluţiei după factorizare

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

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 micşorate dacă se aplică strategii de pivotare.

Pivotare

Matrice de permutare:

- matrice care are exact un element egal cu 1 pe fiecare linie şi pe fiecare coloană, şi 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 înmulţirea la stânga cu o matrice de permutare notată **P**.

Pivotarea pe coloane

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

Pivotare

$$\mathbf{A}_0 = \mathbf{P}_1\mathbf{A}. \tag{98}$$

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

$$\begin{array}{l} \mathbf{A}_1 = \mathbf{E}_1\mathbf{A}_0 = \mathbf{E}_1\mathbf{P}_1\mathbf{A}, \\ \mathbf{A}_2 = \mathbf{E}_2\mathbf{P}_2\mathbf{A}_1 = \mathbf{E}_2\mathbf{P}_2\mathbf{E}_1\mathbf{P}_1\mathbf{A}, \\ \ldots \\ \mathbf{A}_{n-1} = \mathbf{E}_{n-1}\mathbf{P}_{n-1}\cdots\mathbf{E}_2\mathbf{P}_2\mathbf{E}_1\mathbf{P}_1\mathbf{A}. \end{array} \tag{99}$$

Pivotare

$$\mathbf{U} = \mathbf{E}_{n-1}\mathbf{P}_{n-1}\cdots\mathbf{E}_2\mathbf{P}_2\mathbf{E}_1\mathbf{P}_1\mathbf{A}. \tag{100}$$

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

Pivotare

Factorizarea cu pivotare pe linii:

$$\mathbf{PA} = \mathbf{LU}, \tag{102}$$

Factorizarea LU cu pivotare totală (rareori aplicată)

$$\mathbf{PAQ} = \mathbf{LU}, \tag{103}$$

Cazul sistemelor multiple

Rezolvate cu factorizare: $T = \mathrm{O}(2n^3/3 + 2mn^2)$, 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	$2n^3/3 + 2mn^2$

Varianta Cholesky

Dacă **A** este simetrică, atunci este de dorit ca

$$\mathbf{U} = \mathbf{L}^T. \tag{104}$$

Aceasta se poate realiza doar dacă **A** este pozitiv definită. Pp.

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T. \tag{105}$$

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

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

Varianta Cholesky

Teoremă:

Dacă **A** este o matrice simetrică şi pozitiv definită,
atunci factorizarea ei Cholesky există în mod unic, adică există în mod unic o matrice triunghiular inferioară **L** cu elementele diagonale pozitive, astfel încât

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

.

Modul de generare al matricei L

$$\left[\begin{array}{cc} \alpha & \mathbf{a}^T \\ \mathbf{a} & \mathcal{A} \end{array}\right] = \left[\begin{array}{cc} \lambda & \mathbf{0} \\ \mathbf{l} & \mathcal{L} \end{array}\right] \left[\begin{array}{cc} \lambda & \mathbf{l}^T \\ \mathbf{0} & \mathcal{L}^{\mathbf{T}} \end{array}\right]. \tag{106}$$

$$\lambda = \sqrt{\alpha} \tag{107}$$

$$\mathbf{l} = \mathbf{a}/\lambda \tag{108}$$

$$\mathcal{L}\mathcal{L}^T = \mathcal{A} - \mathbf{l}\mathbf{l}^T. \tag{109}$$

Complementul Schur al lui α:

$$\mathbf{S} = \mathcal{A} - \mathbf{l}\mathbf{l}^T = \mathcal{A} - \mathbf{a}\mathbf{a}^T/\alpha. \tag{110}$$

Se poate demonstra că **S** este simetrică şi pozitiv definită şi, în consecinţa $\mathcal{L}\mathcal{L}^T$ este factorizarea ei Cholesky.

Modul de generare al matricei L

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

Similar,

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

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

$$\mathbf{A} = \underbrace{\mathbf{L_1} \mathbf{L}_2 \cdots \mathbf{L}_n}_{\mathbf{L}} \underbrace{\mathbf{L}_n^T \mathbf{L}_{n-1}^T \cdots \mathbf{L}_1^T}_{\mathbf{L}^T} \tag{113}$$

Algoritm

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

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

Algoritm

```
procedură factorizare_LU_Cholesky(n, a, l)
; factorizează matricea a, presupusă simetrică şi pozitiv definită
; întoarce matricea triunghiular inferioară l
; varianta Cholesky
; declaraţii
...
pentru k = 1, n                        ; parcurge sub-etape ale eliminării
        pentru i = k, n                ; calculează coloana, sub diagonală
                s = a_ik
                pentru j = 1, k − 1
                        s = s − l_ij l_kj
                •
                dacă i = k
                        l_kk = √s
                altfel
                        l_ik = s/l_kk
                •
        •
•
retur
```

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 şi nu are nevoie de pivotare. Aceasta se datorează proprietăţilor speciale ale matricei **A**, care fiind pozitiv definită este şi diagonal dominantă.

Ce sunt matricele rare

Matrice rară = matrice care conţine un număr foarte mare de elemente nenule.
O matrice care nu este rară se numeşte matrice densă sau plină.
Densitatea unei matrice = raportul dintre numărul de elemente nenule şi numărul total de elemente al matricei.
Dacă, pentru o anumită matrice care are şi elemente nule, se poate elabora un algoritm care exploatează această structură şi care, este mai eficient decât algoritmul conceput pentru matricea plină, atunci aceasta este o matrice rară.

Formate de memorare a matricelor rare

Matrix Market: se memorează doar valorile nenule şi "coordonatele" lor în matrice. http://math.nist.gov/MatrixMarket
Exemplu: tablou bidimensional de dimensiune $m \times n$:
$M_{plin} = 8mn$ B
$M_{rar,coord} = 8 * n_{nz} + 4 * 2n_{nz} = 16n_{nz}$ B.

$$\mathbf{M} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 5 & 1 \\ 2 & 3 & 0 & 7 \end{bmatrix} \Rightarrow \begin{cases} val & = [\ 4 \quad 5 \quad 1 \quad 2 \quad 3 \quad 7\ ] \\ r\_idx & = [\ 1 \quad 2 \quad 2 \quad 3 \quad 3 \quad 3\ ] \\ c\_idx & = [\ 1 \quad 3 \quad 4 \quad 1 \quad 2 \quad 4\ ] \end{cases}$$

Formate de memorare a matricelor rare

Formatul Yale sau CRS - *Compressed Row Storage*:
$M_{rar,CRS} = 8n_{nz} + 4(m+1) + 4n_{nz} = 12n_{nz} + 4(m+1)$ 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 şi sum numele Harwell - Boeing.
$M_{rar,CCS} = 8n_{nz} + 4(n+1) + 4n_{nz} = 12n_{nz} + 4(n+1)$ B.

$$\mathbf{M} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 5 & 1 \\ 2 & 3 & 0 & 7 \end{bmatrix} \Rightarrow \begin{cases} val = [\ 4\ \ 2\ \ 3\ \ 5\ \ 1\ \ 7\ ] \\ c\_ptr = [\ 1\ \ 3\ \ 4\ \ 5\ \ 7\ ] \\ r\_idx = [\ 1\ \ 3\ \ 3\ \ 2\ \ 2\ \ 3\ ] \end{cases}$$

Formate de memorare a matricelor rare

Matricelor bandă (de exemplu matrice tridiagonală):

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

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

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

Metode directe pentru matrice rare

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

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

Un singur element de multiplicare $m = -p_{k+1}/q_k$.
Singura modificare suferind-o ecuaţia $k+1$:
$q_{k+1} = q_{k+1} + m * r_k$, şi temenul liber corespunzător.

Metode directe pentru matrice rare

Gauss pentru matrice tridiagonală, matricea după eliminare.

$$\left[\begin{array}{ccccccccccc}
q_1 & r_1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & q_2 & r_2 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
\ldots & & & & & & & & & \\
0 & 0 & 0 & q_k & r_k & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & 0 & q_{k+1} & r_{k+1} & 0 & \cdots & 0 & 0 \\
\ldots & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots q_{n-1} 0 & r_{n-1} & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & q_n
\end{array}\right]$$

Retrosubstituţie

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

$$q_i x_i + r_i x_{i+1} = b_i \quad \Rightarrow \quad x_i = (b_i - r_i x_{i+1}) / q_i, \quad i = n-1, \ldots, 1. \tag{117}$$

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" coeficienţilor - indici de la 1
tablou real b[n]                        ; vectorul termenilor liberi
tablou real x[n]                        ; vectorul soluţie
întreg i, k
; etapa de eliminare din metoda Gauss
pentru k = 1, n − 1                     ; parcurge sub-etape ale eliminării
        m = −p_{k+1}/q_k                ; element de multiplicare
        q_{k+1} = q_{k+1} + m r_k       ; modifică element în linia k + 1
        b_{k+1} = b_{k+1} + m b_k       ; modifică termenul liber al ecuaţiei k + 1
•
; etapa de retrosubstituţie
x_n = b_n/q_n
pentru i = n − 1, 1, −1
        x_i = (b_i − r_i x_{i+1})/q_i
•
retur
```

$$T = \mathrm{O}(8n),\ M = \mathrm{O}(5n).$$

Metode directe pentru matrice rare

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

Metode directe pentru matrice rare

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

$$\mathbf{A}_1 = \begin{bmatrix} * & 0 & \cdots & 0 & 0 & * \\ 0 & * & \cdots & 0 & 0 & * \\ \cdots & & & & & \\ 0 & 0 & \cdots & * & 0 & * \\ 0 & 0 & \cdots & 0 & * & * \\ * & * & \cdots & * & * & * \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} * & * & * & \cdots & * & * \\ * & * & 0 & \cdots & 0 & 0 \\ * & 0 & * & \cdots & 0 & 0 \\ \cdots & & & & & \\ * & 0 & \cdots & 0 & * & 0 \\ * & 0 & \cdots & 0 & 0 & * \end{bmatrix}$$

Matricea $\mathbf{A}_1$ are factorii LU rari, în timp ce matricea $\mathbf{A}_2$ are factorii LU plini.

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

Referinţe

- [Ciuprina13a] Gabriela Ciuprina - Algoritmi numerici pentru calcule ştiinţifice 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.

Referinţe

- [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

Referinţe

Pachete existente: ([Davis16])

BCSLIB-EXT Ashcraft (1995), Ashcraft et al. (1998), Pierce and Lewis (1997), aanalytics.com BSMP Bank and Smith (1987), www.netlib.org/linalg/bsmp.f CHOLMOD Chen et al. (2008), suitesparse.com CSparse Davis (2006), suitesparse.com DSCPACK 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)

Referinţe

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 G¨artner (2004), Schenk, G¨artner and Fichtner (2000), www.pardiso-project.org PaStiX H´enon et al. (2002), www.labri.fr/ ramet/pastix QR MUMPS Buttari (2013), buttari.perso.enseeiht.fr/qr mumps 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

Referinţe

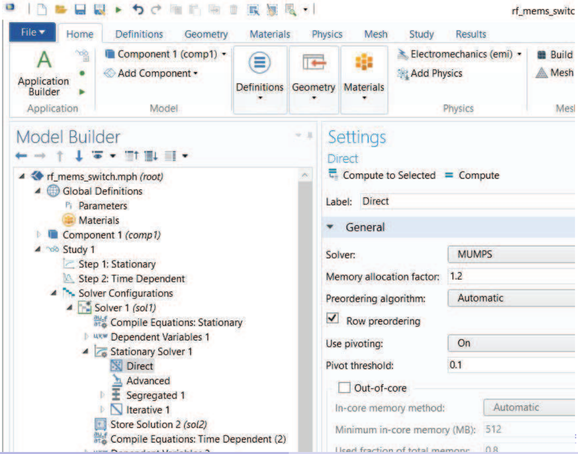
TAUCS Rotkin and Toledo (2004), www.tau.ac.il/ stoledo/taucs UMFPACK Davis (2004b) Davis and Duff (1997) (1999), suitesparse.com WSMP Gupta (2002a), Gupta et al. (1997), www.cs.umn.edu/ agupta/wsmp Y12M Zlatev, Wasniewski and Schaumburg (1981), www.netlib.org/y12m YSMP Eisenstat et al. (1977) (1982),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

Referinţe

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

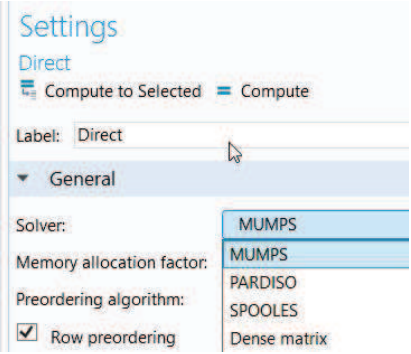
Pe scurt

Fiţi atenţi la astfel de informaţii (capturi din COMSOL)



Pe scurt

Fiţi atenţi la astfel de informaţii



Recomandare

Abonaţi-vă (cel puţin pe durata acestui semestru) la următoarele

1. NA Digest http://www.netlib.org/na-digest-html/
2. Computational Science Stack Exchange https://scicomp.stackexchange.com/ şi urmăriţi unul sau mai multe subiecte de interes.