Cap.2. Sisteme de ecuaţii algebrice liniare - metode directe (II)

Prof.dr.ing. Gabriela Ciuprina

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

Suport didactic pentru disciplina *Metode numerice*, 2017-2018

Cuprins

Formularea problemei

Sistem de n ecuaţii algebrice liniare cu n necunoscute:

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

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

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\| \tag{6}$$

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

$$\varepsilon_x \le \kappa(\mathbf{A}) \varepsilon_b, \tag{7}$$

- $\kappa(\mathbf{A}) \ge 1$:
  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.

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.

Formularea problemei

Fie m sisteme de ecuaţii algebrice liniare

$$\mathbf{A}\mathbf{x}^{(1)} = \mathbf{b}^{(1)}, \quad \mathbf{A}\mathbf{x}^{(2)} = \mathbf{b}^{(2)}, \quad \ldots, \quad \mathbf{A}\mathbf{x}^{(m)} = \mathbf{b}^{(m)}, \tag{8}$$

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

$$\mathbf{B} = [\mathbf{b}^{(1)} \quad \mathbf{b}^{(2)} \quad \ldots \quad \mathbf{b}^{(m)}] \quad \in \mathbb{R}^{n \times m} \tag{9}$$

$$\mathbf{X} = [\mathbf{x}^{(1)} \quad \mathbf{x}^{(2)} \quad \ldots \quad \mathbf{x}^{(m)}] \quad \in \mathbb{R}^{n \times m} \tag{10}$$

Se cere să se rezolve sistemul

$$\mathbf{A}\mathbf{X} = \mathbf{B}. \tag{11}$$

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

$$\begin{aligned} T_e &= \sum_{k=1}^{n-1}[2(n-k)+2m+1](n-k) \approx \sum_{k=1}^{n-1}[2(n-k)^2+2m(n-k)] = \\ &= 2\frac{(n-1)n(2n-1)}{6} + 2m\frac{n(n-1)}{2} \approx \frac{2n^3}{3} + mn^2. \end{aligned} \tag{12}$$

$$T_s = m\sum_{i=1}^{n-1}[2(n-i)+2] \approx m\sum_{i=1}^{n-1}[2(n-i)] = 2m\frac{n(n-1)}{2} \approx mn^2. \tag{13}$$

$T = \mathrm{O}(2n^3/3 + 2mn^2)$, 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 calcului 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

$$\mathbf{Ax} = \mathbf{b}, \tag{14}$$

$$\mathbf{A} = \mathbf{LU}, \qquad \text{factorizare} \tag{15}$$

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

$$\mathbf{LUx} = \mathbf{b}. \tag{16}$$

Ideea metodei

$$\mathbf{Ax} = \mathbf{b}, \tag{17}$$

$$\mathbf{A} = \mathbf{LU}, \qquad \text{factorizare} \tag{18}$$

$$\mathbf{LUx} = \mathbf{b}. \tag{19}$$

Notăm

$$\mathbf{y} = \mathbf{Ux}, \tag{20}$$

(50) $\Leftrightarrow$

$$\begin{array}{lll} \mathbf{Ly} = \mathbf{b}, & \text{substituţie} & \text{progresivă} \\ \mathbf{Ux} = \mathbf{y}. & \text{substituţie} & \text{progresivă} \end{array} \tag{21}$$

Un exemplu simplu - pornind de la Gauss

$$\left\{\begin{array}{rcrcrcr} x_1 & + & 2x_2 & - & x_3 & = & -1, \\ -2x_1 & + & 3x_2 & + & x_3 & = & 0, \\ 4x_1 & - & x_2 & - & 3x_3 & = & -2. \end{array}\right. \tag{22}$$

$$\left\{\begin{array}{rcrcrcr} x_1 & + & 2x_2 & - & x_3 & = & -1, \\ & & 7x_2 & - & x_3 & = & -2, \\ & & -9x_2 & + & x_3 & = & 2. \end{array}\right. \tag{23}$$

$$\left\{\begin{array}{rcrcrcr} x_1 & + & 2x_2 & - & x_3 & = & -1, \\ & & 7x_2 & - & x_3 & = & -2, \\ & & & - & 2/7x_3 & = & -4/7. \end{array}\right. \tag{24}$$

$$\begin{array}{l} x_3 = (-4/7)/(-2/7) = 2, \\ x_2 = (-2 + x_3)/7 = 0, \\ x_1 = -1 - 2x_2 + x_3 = 1. \end{array} \tag{25}$$

Un exemplu simplu - pornind de la Gauss

Factorizare

$$\mathbf{U} = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 7 & -1 \\ 0 & 0 & -2/7 \end{bmatrix}. \tag{26}$$

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ -2/1 & 1 & 0 \\ 4/1 & -9/7 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 4 & -9/7 & 1 \end{bmatrix}. \tag{27}$$

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

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & -1 \\ -2 & 3 & 1 \\ 4 & -1 & -3 \end{bmatrix}. \tag{28}$$

Un exemplu simplu - pornind de la Gauss

Substituţie progresivă

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

$$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 4 & -9/7 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ -2 \end{bmatrix}, \tag{29}$$

$$\left\{ \begin{array}{rcrcrcr} y_1 & & & & & = & -1 \\ -2y_1 & + & y_2 & & & = & 0 \\ 4y_1 & - & 9/7y_2 & + & y_3 & = & -2 \end{array} \right. \tag{30}$$

$$\begin{aligned} y_1 &= -1 \\ y_2 &= 2y_1 = -2 \\ y_3 &= -2 - 4y_1 + 9/7y_2 = -4/7. \end{aligned} \tag{31}$$

Un exemplu simplu - pornind de la Gauss

Substituţie regresivă

$\mathbf{U}\mathbf{x} = \mathbf{y}$

$$\begin{bmatrix} 1 & 2 & -1 \\ 0 & 7 & -1 \\ 0 & 0 & -2/7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \\ -4/7 \end{bmatrix}, \tag{32}$$

$$\begin{cases} x_1 + 2x_2 - x_3 & = & -1 \\ 7x_2 - x_3 & = & -2 \\ -2/7x_3 & = & -4/7. \end{cases} \tag{33}$$

$$\begin{aligned} x_3 &= (-4/7)/(-2/7) = 2, \\ x_2 &= (-2 + x_3)/7 = 0, \\ x_1 &= -1 - 2x_2 + x_3 = 1. \end{aligned} \tag{34}$$

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

$$\begin{bmatrix} 3 & 2 \\ 6 & 1 \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}. \tag{35}$$

$$\left\{ \begin{aligned} l_{11}u_{11} &= 3 \\ l_{12}u_{12} &= 2 \\ l_{21}u_{11} &= 6 \\ l_{21}u_{12} + l_{22}u_{22} &= 1 \end{aligned} \right. \tag{36}$$

Sistemul devine determinat doar dacă fixăm oricare două valori.

Variante de factorizare

Exemplu:

$$\left[\begin{array}{cc} 3 & 2 \\ 6 & 1 \end{array}\right] = \left[\begin{array}{cc} 1 & 0 \\ 2 & 1 \end{array}\right] \left[\begin{array}{cc} 3 & 2 \\ 0 & -3 \end{array}\right] = \left[\begin{array}{cc} 3 & 0 \\ 6 & -3 \end{array}\right] \left[\begin{array}{cc} 1 & 2/3 \\ 0 & 1 \end{array}\right]. \tag{37}$$

$$\left[\begin{array}{cc} 9 & 2 \\ 2 & 1 \end{array}\right] = \left[\begin{array}{cc} 3 & 0 \\ 2/3 & \sqrt{5}/3 \end{array}\right] \left[\begin{array}{cc} 3 & 2/3 \\ 0 & \sqrt{5}/3 \end{array}\right]. \tag{38}$$

Algoritmul variantei Doolittle

$$\mathbf{A}_0 = \mathbf{A}, \tag{39}$$

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{E}_1\mathbf{A}_0, \\ \mathbf{A}_2 &= \mathbf{E}_2\mathbf{A}_1 = \mathbf{E}_2\mathbf{E}_1\mathbf{A}_0, \\ &\dots \\ \mathbf{A}_{n-1} &= \mathbf{E}_{n-1}\mathbf{A}_{n-2} = \mathbf{E}_{n-1}\mathbf{E}_{n-2}\cdots\mathbf{E}_2\mathbf{E}_1\mathbf{A}_0. \end{aligned} \tag{40}$$

$$\mathbf{U} = \mathbf{A}_{n-1}. \tag{41}$$

$$\mathbf{E} = \mathbf{E}_{n-1}\mathbf{E}_{n-2}\cdots\mathbf{E}_2\mathbf{E}_1, \tag{42}$$

$$\mathbf{U} = \mathbf{E}\mathbf{A}. \tag{43}$$

Dar **E** este nesingulară şi:

$$\mathbf{L} = \mathbf{E}^{-1}. \tag{44}$$

Algoritmul variantei Doolittle

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & a'_{32} & a'_{33} \end{bmatrix} = \mathbf{E}_1 \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}. \quad (45)$$

$$\mathbf{E}_1 = \begin{bmatrix} 1 & 0 & 0 \\ -a_{21}/a_{11} & 1 & 0 \\ -a_{31}/a_{11} & 0 & 1 \end{bmatrix}. \quad (46)$$

$$\mathbf{E}_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ a_{21}/a_{11} & 1 & 0 \\ a_{31}/a_{11} & 0 & 1 \end{bmatrix}, \quad \mathbf{E}_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & a'_{32}/a'_{22} & 1 \end{bmatrix}. \quad (47)$$

$$\mathbf{E}^{-1} = \mathbf{E}_1^{-1}\mathbf{E}_2^{-1} \cdots \mathbf{E}_{n-2}^{-1}\mathbf{E}_{n-1}^{-1}. \quad (48)$$

$$\mathbf{E}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ a_{21}/a_{11} & 1 & 0 \\ a_{31}/a_{11} & a'_{32}/a'_{22} & 1 \end{bmatrix} = \mathbf{L}. \quad (49)$$

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

$$\mathbf{LUx} = \mathbf{b}. \tag{50}$$

Notăm

$$\mathbf{y} = \mathbf{Ux}, \tag{51}$$

(50) $\Leftrightarrow$

$$\mathbf{Ly} = \mathbf{b}, \tag{52}$$

$$\mathbf{Ux} = \mathbf{y}. \tag{53}$$

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

$$\begin{cases} l_{11}y_1 = b_1, \\ l_{21}y_1 + l_{22}y_2 = b_2, \\ \dots \\ l_{n1}y_1 + l_{n2}y_2 + \cdots l_{nn}y_n = b_n, \end{cases} \Rightarrow \begin{cases} y_1 = b_1/l_{11}, \\ y_2 = (b_2 - l_{21}y_1)/l_{22}, \\ \dots \\ y_n = (b_n - \sum_{k=1}^{n-1} l_{nk}y_k)/l_{nn}. \end{cases} \tag{54}$$

Calculul soluţiei după factorizare

$$y_1 = b_1 / l_{11}, \tag{55}$$

$$y_i = \left(b_i - \sum_{j=1}^{i-1} l_{ij} y_j \right) / l_{ii}, \quad i = 2, \ldots, n. \tag{56}$$

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

$$x_n = y_n / u_{nn}, \tag{57}$$

$$x_i = \left(y_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) / u_{ii}, \quad i = n-1, \ldots, 1. \tag{58}$$

Calculul soluţiei după factorizare

```
procedură 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 (55), unde l_11 = 1
pentru i = 2, n
        s = 0
        pentru j = 1, i − 1
                s = s + a_ij y_j; formula (56), unde L este memorat în a
        •
        y_i = b_i − s ; deoarece l_ii = 1
•
; substituţie regresivă
x_n = y_n / a_nn                 ; formula (57), unde 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
```

Evaluarea algoritmului

Complexitate:

- Factorizarea propriu-zisă a: $T_f = \mathrm{O}(2n^3/3)$
- Rezolvările: $T_s = \mathrm{O}(2n^2)$.
- 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.

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	$2n^3/3 + 2mn^2$

Cazul matricelor 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

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*):

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

Metode directe pentru matrice rare

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

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

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.

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

Retrosubstituţie

$$x_n = b_n/q_n, \tag{59}$$

$$q_i x_i + r_i x_{i+1} = b_i \quad \Rightarrow \quad x_i = (b_i - r_i x_{i+1})/q_i, \quad i = n-1, \dots, 1. \tag{60}$$

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

- Pag 51 - 56 din
  [3] Gabriela Ciuprina - Algoritmi numerici pentru calcule ştiintifice în ingineria electrică, Editura MatrixROM, 2013, disponibil la

  http://lmn.pub.ro/∼gabriela/books/AlgNr_MatrixRom2013.pdf

Notă. Metoda factorizării LU nu este parcursă în mod obligatoriu la laborator, dar intră în chestiunile care se vor examina în sesiune. Pentru înţelegerea profundă a lucrurilor, vă recomand să implementaţi, să testaţi şi să analizaţi toţi algoritmii care au descrişi în pseudcod în acest curs. Pentru bonus, puteţi redacta un raport dedicat acestei teme. Doritorii sunt rugaţi să îşi anunţe intenţia.