CURS 3

METODE NUMERICE PENTRU SISTEME DE ECUAŢII LINIARE

Partea I

Metode directe:

1 Gauss
2 LU
3 Cholesky

0. INTRODUCERE

Sistemele de ecuaţii liniare apar în modelarea unor probleme ştiinţifice, precum şi în metodele numerice de rezolvare a acestora. Exemple: sisteme de ecuaţii neliniare, aproximarea funcţiilor, probleme la limită pentru ecuaţii diferenţiale, ecuaţii cu derivate parţiale, optimizare, etc.

Găsirea soluţiei unui sistem de ordin mare (sau inversarea unei matrici de ordin mare), poate fi o sarcină dificilă în practică, datorită:

(1) Numărului mare de operaţii aritmetice necesare pentru găsirea soluţiei.

(2) Erorilor propagate, într-un şir lung de operaţii cu numere reprezentate în virgulă flotantă.

Dificultatea (1) face nepractică rezolvarea manuală, iar (2) este inerentă în utilizarea calculatorului.

Analiza unei metode cuprinde problemele:

- Numărul de operaţii aritmetice necesare pentru a găsi soluţia.
- Estimarea, înainte de calculare, a preciziei soluţiei (estimarea *a priori* a erorii).
- Verificarea preciziei soluţiei calculate (estimarea *a posteriori* a erorii).

Problemele practice se pot clasifica după ordinul sistemului (numărul de ecuaţii) şi tipul matricii sistemului. Matricea sistemului se zice *densă*, dacă majoritatea elementelor sale sunt diferite de zero, şi *rară* – dacă majoritatea elementelor sunt zero. Astfel, avem categoriile:

I. Sisteme de ordin moderat, cu matrice densă:

Acestea se rezolvă în memorie, şi ordinul sistemului este limitat numai de memoria disponibilă. Uzual, ordinul sistemului poate ajunge la sute de ecuaţii. Majoritatea algoritmilor se bazează pe eliminarea Gauss, cu variante pentru clase speciale de matrici.

II. Sisteme de ordin mare, cu matrice rară:

Asemenea sisteme pot ajunge la sute de mii de ecuaţii. Eliminarea Gauss nu mai este convenabilă, şi se utilizează *metode iterative*.

III. Sisteme de tip II – rezolvare prin utilizarea de fişiere disc

Memoria fiind insuficientă, se folosesc fişiere pe disc pentru generarea şi procesarea matricii. Lucrul cu matricea sistemului se face pe 'blocuri': de exemplu, matricea este generată pe disc; succesiv, blocurile sunt citite, procesate în memorie şi rescrise pe disc. Un exemplu este cel al sistemelor cu matrice bandă simetrică, sisteme care apar în problemele de analiză statică a structurilor ■

Un sistem de ordinul n se va nota prin:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ \cdots \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = b_n \end{cases}$$

Matriceal, sistemul se scrie:

$$\begin{bmatrix} a_{11} & \cdots & \cdots & a_{1n} \\ a_{21} & \cdots & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & \cdots & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{1}$$

Sau:

$$\mathbf{Ax} = \mathbf{b}, \tag{1'}$$

în care **A** este matricea sistemului ($n \times n$), iar **x** şi **b** vectorul necunoscutelor şi, respectiv, vectorul termenilor liberi (matrici coloană cu n elemente):

$$\mathbf{A} = [a_{ij}]_{i,\,j=1,n} \qquad \mathbf{x} = [x_1 \,...\, x_n]^T, \qquad \mathbf{b} = [b_1 \,...\, b_n]^T$$

I. METODE DIRECTE

1 Metoda Gauss

1.1 Metoda

Sistemul dat se transformă în sistemul echivalent $\mathbf{Ux} = \mathbf{g}$, cu matricea **U** superior triunghiulară (v. ecuația (2), mai jos), cum urmează. Sistemul dat (1) se notează

$$\mathbf{A}^{(1)}\mathbf{x} = \mathbf{b}^{(1)}.$$

La pasul curent k ($k = 1, 2, \ldots, n-1$), sistemul este

$$\mathbf{A}^{(k)}\mathbf{x} = \mathbf{b}^{(k)}.$$

Se lucrează cu liniile $i = k, \ldots, n$, din $\mathbf{A}^{(k)}$ şi $\mathbf{b}^{(k)}$. (Liniile $1, \ldots, k-1$, procesate anterior, rămân neschimbate). Concret, se operează cu sub-matricea $\mathbf{A}^{(k)}(k:n,\, k:n)$ şi sub-coloana $\mathbf{b}^{(k)}(k:n)$ a termenilor liberi.

$$\mathbf{A}^{(k)} = \left[\begin{array}{ccccccccc}
a_{11}^{(1)} & a_{12}^{(1)} & \ldots & a_{1k}^{(1)} & . & & . & & a_{1n}^{(1)} \\
0 & a_{22}^{(2)} & \ldots & a_{2k}^{(2)} & . & & . & & a_{2n}^{(2)} \\
. & 0 & \ddots & & . & & . & & \\
. & . & \ldots & a_{k-1,k-1}^{(k-1)} & . & & . & & a_{k-1,n}^{(k-1)} \\
\hline
0 & . & \ldots & 0 & a_{kk}^{(k)} & . & a_{kj}^{(k)} & . & a_{kn}^{(k)} \\
. & . & \ldots & . & . & . & . & . & . \\
. & . & \ldots & . & a_{ik}^{(k)} & . & a_{ij}^{(k)} & . & a_{in}^{(k)} \\
. & . & \ldots & . & . & . & . & . & . \\
0 & 0 & \ldots & 0 & a_{nk}^{(k)} & . & . & . & a_{nn}^{(k)}
\end{array}\right]
\begin{array}{l}
\\ \\ \\ \\
\times(-m_{ik}); \\
\quad\downarrow \\
\oplus \hookleftarrow \\
\\ \\
\end{array}$$

$$\mathbf{b}^{(k)} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \\ b_{k-1,k-1}^{(k-1)} \\ \text{-- -- -- -- --} \\ b_k^{(k)} \\ . \\ b_i^{(k)} \\ . \\ b_n^{(k)} \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \times(-m_{ik}) \\ \downarrow \\ \oplus \hookleftarrow \\ \\ \\ \end{matrix}$$

Elementul diagonal $a_{kk}^{(k)}$ se numeşte *pivot* (la pasul *k*). Presupunem $a_{kk}^{(k)} \neq 0$ – v. mai jos pivotare, şi definim **multiplicatorii** de la pasul *k*:

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}; \qquad i = k+1, \ldots, n$$

Pentru $i = k+1, \ldots, n$: linia *k* se înmulţeşte cu $-m_{ik}$, şi se adună la linia *i*. Rezultă elemente nule în coloana *k*, sub elementul diagonal $a_{kk}^{(k)}$. Linia *k* rămâne neschimbată. Noii coeficienţi şi termeni liberi vor fi:

$$\left.\begin{array}{l} a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} \cdot a_{kj}^{(k)}; \quad j = k+1, \ldots, n \\ b_i^{(k+1)} = b_i^{(k)} - m_{ik} \cdot b_k^{(k)} \end{array}\right\} \quad i = k+1, \ldots, n$$

La pasul *n* se notează, pentru convenienţă, $\mathbf{U} = \mathbf{A}^{(n)}, \quad \mathbf{g} = \mathbf{b}^{(n)}$, şi sistemul devine $\mathbf{Ux} = \mathbf{g}$. Explicit:

$$\begin{bmatrix} u_{11} & . & . & . & . & u_{1n} \\ 0 & u_{22} & . & . & . & u_{2n} \\ & & . & . & . & . \\ 0 & & 0 & u_{kk} & . & u_{kn} \\ & & & & . & . \\ 0 & & & & 0 & u_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_k \\ . \\ x_n \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ . \\ g_k \\ . \\ g_n \end{bmatrix} \tag{2}$$

Observaţi că $u_{kk} = a_{kk}^{(k)}$.

Acest sistem se rezolvă prin substituţie înapoi:

$$x_n = \frac{g_n}{u_{nn}};$$

$$x_k = \frac{g_k - \sum_{j=k+1}^{n} u_{kj} \cdot x_j}{u_{kk}}; \qquad k = n-1, n-2, \dots, 1$$

■

1.2 Factorizarea (descompunerea) LU

Multiplicatorii m_{ik} (unde $i = k+1, \dots, n$) din eliminarea Gauss, se pot reţine pentru a rezolva $\mathbf{Ax} = \mathbf{b}$ cu diferiţi termeni liberi **b**. Introducem atunci, matricea inferior-triunghiulară (cu 1 pe diagonală, şi multiplicatorii m_{ik} în sub-coloanele $k+1:n$):

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & . & . & . & 0 \\ m_{21} & 1 & 0 & & & 0 \\ m_{31} & m_{32} & 1 & 0 & & 0 \\ . & . & . & . & . & . \\ . & . & . & . & 1 & 0 \\ m_{n1} & m_{n2} & . & . & m_{n,n-1} & 1 \end{bmatrix} \qquad (3)$$

Avem următoarea propoziţie:

Propoziţia 1

Dacă, în eliminarea Gauss, *la fiecare pas k* pivotul $a_{kk}^{(k)} \neq 0$, atunci $\mathbf{A} = \mathbf{LU}$ ■

L este matricea inferior triunghiulară a multiplicatorilor (3), iar **U** matricea superior triunghiulară din (2) ■

***Consecinţă – Calculul determinantului*:**

Dacă la fiecare pas *k* pivotul $a_{kk}^{(k)} \neq 0$, atunci:

$$\det(\mathbf{A}) = u_{11} \cdot u_{22} \cdot \ldots \cdot u_{nn}$$

Observaţi că $u_{kk} = a_{kk}^{(k)}$, astfelcă valoarea determinantului este *produsul pivoţilor* ■

Într-adevăr: $\det(\mathbf{A}) = \det(\mathbf{L}) \cdot \det(\mathbf{U})$, iar det(**L**) =1.

Pentru calculul determinatului în cazul când se schimbă liniii în matricea **A** (se pivotează), v. mai jos.

1.3 Număr de operaţii

Pentru n = ”mare”, numărul de operaţii în eliminarea Gauss este:

$$NOP_{Guass} \approx \frac{n^3}{3}$$

(Operaţie = adunare sau scădere; în calcule, acestea sunt urmate de obicei de o inmulţire sau împărţire.)

■

1.4 Pivotare

La fiecare pas *k*, *pivotul* $a_{kk}^{(k)} \neq 0$ apare la numitorul multiplicatorilor m_{ik}. Astfel, avem condiţiile:

Pivotul nu trebuie să fie nul, şi nici ”foarte mic”:

- Dacă pivotul este nul, eliminarea Gauss nu poate continua. În acest caz, matricea este singulară.
- Dacă pivotul este ”foarte mic”: rezultă multiplicatori m_{ik} mari, iar erorile de rotunjire în valoarea lui m_{ik} conduc la erori mari în calculaţiile următoare. Matricea este ”aproape singulară”.

Practic, se testează dacă pivotul este mai mic decât un *prag* ales.
V. Exemplul de mai jos.

Procedeul de alegere a unui pivot ≠ 0 se zice *pivotare*. Aceasta constă în:

- Căutarea elementului de modul maxim, în sub-matricea cu liniile şi coloanele $k, \ldots, n$;
- Aducerea lui pe poziţia de pivot – poziţia (k, k) – prin schimbări de linii, sau schimbări de linii şi coloane.

  Schimbarea de linii din **A**, se face şi în vectorul **b**.

Dacă pivotul este mai mic decât pragul, eliminarea Gauss se opreşte.

Strategii de pivotare:

1) Pivotarea parţială:

Se caută elementul de modul maxim în sub-coloana $a(k:n,k)$. Acesta devine pivot, aducându-l în poziţia (k,k). Fie pivotul găsit în linia $I \geq k$. Dacă pivotul este mai mare decât pragul şi $I > k$, atunci se schimbă liniile *k* şi *I* (în $\mathbf{A}^{(k)}$ şi în $\mathbf{b}^{(k)}$).

$$
\begin{array}{cc}
\left[\begin{array}{c:cc}
a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\
\vdots & & \vdots \\
a_{Ik}^{(k)} & \cdots & . \\
\vdots & & \vdots \\
a_{nk}^{(k)} & \cdots & a_{nn}^{(k)}
\end{array}\right] &
\begin{array}{c} (k) \\ \updownarrow \\ (I) \\ \\ \\ \end{array}
\end{array}
\qquad\qquad
\begin{array}{cc}
\left[\begin{array}{c}
b_{k}^{(k)} \\ \vdots \\ b_{I}^{(k)} \\ \vdots \\ b_{n}^{(k)}
\end{array}\right] &
\begin{array}{c} (k) \\ \updownarrow \\ (I) \\ \\ \\ \end{array}
\end{array}
$$

2) Pivotarea completă:

Se caută elementul de modul maxim în sub-matricea $a(k:n,\, k:n)$. Dacă maximul este atins pentru linia $I > k$ şi coloana $J > k$, atunci se permută liniile *k* şi *I* şi coloanele *k* şi *J* - în **A**, şi liniile *k* şi *I* - în **b** ■

$$
\begin{array}{cc}
\left[\begin{array}{ccccc}
a_{kk}^{(k)} & \cdots & . & \cdots & a_{kn}^{(k)} \\
\vdots & & & & \vdots \\
. & & a_{IJ}^{(k)} & & . \\
\vdots & & & & \vdots \\
a_{nk}^{(k)} & \cdots & . & \cdots & a_{nn}^{(k)}
\end{array}\right] &
\begin{array}{c} (k) \\ \updownarrow \\ (I) \\ \\ \\ \end{array} \\
(k) \quad \leftrightarrow \quad (J) &
\end{array}
\qquad\qquad
\begin{array}{cc}
\left[\begin{array}{c}
b_{k}^{(k)} \\ \vdots \\ b_{I}^{(k)} \\ \vdots \\ b_{n}^{(k)}
\end{array}\right] &
\begin{array}{c} (k) \\ \updownarrow \\ (I) \\ \\ \\ \end{array}
\end{array}
$$

Observaţii

- Prin pivotare rezultă, la fiecare pas, multiplicatori de modul ≤ 1:

  $|m_{ik}| \leq 1, \quad i = k+1, \ldots, n$.

Acesta previne generarea în $\mathbf{A}^{(k+1)}$, de elemente de mărime foarte diferită, care ar putea duce la erori de pierdere de semnificaţie.

- La pivotarea completă, datorită permutării de coloane, ordinea necunoscutelor se schimbă. Ea trebuie refăcută la sfârşitul eliminării.

Teoretic, pivotarea completă duce la o precizie mai mare a soluţiei; în practică însă, diferenţa este mică faţă de pivotarea parţială. În plus, pivotarea completă cere mai mult timp-calculator. Din aceste motive, majoritatea algoritmilor utilizează pivotarea parţială. În ceea ce urmează vom considera numai pivotarea parţială, numită pe scurt, pivotare ■

Observaţie

Dacă nu se pivotează, atunci avem **LU** = **A** (Propoziţia 1).

Dacă se pivotează, atunci avem $\mathbf{LU} = \mathbf{A}'$, unde $\mathbf{A}' = \mathbf{PA}$, iar **P** este o matrice de permutare de linii. Matricea $\mathbf{A}'$ se obţine din **A**, permutând liniile în aceeaşi ordine în care se permută la pivotare. Sistemul care se rezolvă este $\mathbf{A}'\mathbf{x} = \mathbf{b}'$, unde $\mathbf{b}'$ este **b** cu liniile permutate în ordinea de la pivotare ■

Calculul determinantului:

Determinantul matricii $\mathbf{A}' = \mathbf{LU}$ este $\det(\mathbf{A}') = u_{11}u_{22}\ldots u_{nn}$.

Determinantul matricii **A** va fi

$$\det(\mathbf{A}) = (-1)^{n\_l}\det(\mathbf{A}'),$$

unde *n_l* este numărul de schimbări de linii în cursul pivotării.

■

Exemplu – Pivot ”foarte mic”

Considerăm sistemul $\mathbf{Ax} = \mathbf{b}$, unde:

$$\mathbf{A} = \begin{bmatrix} 5 & 6 & -1 & 1 \\ -1 & 0 & -1 & 1 \\ 2 & 2 & 1 & 6 \\ 4 & 2 & 3 & 4 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$$

În calculul exact, **A** este singulară. Rezolvând în simplă precizie, cu programul Gauss 2007, cu un prag de $1E-7$, se obţin următoarele rezultate:

```
Permutare, Pivot:
1       5.000000
2<->4  -2.800000
3      0.8571427
4      9.5367432E-07

 * Solutia pentru termenii liberi nr. 1
 x(1) =    7340030.
 x(2) =   -7340031.
 x(3) =   -6291456.
 x(4) =    1048576.


 * Proba

 Diferente (A*x-b)
           1  -1.000000
           2  0.5000000
           3  -2.000000
           4  -5.000000

    Diferenta (A*x-b), maxima in modul:  -5.000000    ; ecuatia 4
```

Adică, soluţia nu verifică, şi este inutil a o calcula.

Erorile în soluţie sunt produse de pivotul de la pasul 4, care este "foarte mic".

Cu un prag de $1E-6$, programul se opreşte cu mesajul:

```
Linia 4: Pivot =  9.5367432E-07

* Pivot <1.00E-06 !
* Sistemul nu se poate rezolva.
```

În calculul în dublă precizie, pivotul din linia 4 rezultă `0.000000000000000`.

■

2 Descompunerea LU

2.1 Calculul direct al factorilor L şi U

Problema este calculul direct al factorilor **L** şi **U**, astfel ca $\mathbf{A} = \mathbf{LU}$.

Elementele lui **L** şi **U** se găsesc din ecuaţiile $a_{ij} = linia\ i_{\mathbf{L}} \times coloana\ j_{\mathbf{U}}$:

$$a_{ij} = \begin{bmatrix} l_{i1} & l_{i2} & \dots & l_{ii} & \vdots & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} u_{j1} \\ u_{j2} \\ \vdots \\ u_{jj} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \sum_{k=1}^{\min(i,j)} l_{ik} u_{kj}$$

În particular, înmulţind linia *i* din **L** cu coloana *i* din **U**, rezultă:

$$a_{ii} = \sum_{k=1}^{i-1} l_{ik} u_{ki} + l_{ii} u_{ii}$$

Există o neunicitate a descompunerii **LU**, care provine din alegerea arbitrară a lui l_{ii} sau u_{ii}, $i = 1, \dots, n$. Două metode se pot considera:

l_{ii} = arbitrar $\neq 0$;

u_{ii} = arbitrar $\neq 0$.

Prezentăm, pentru exemplificare, formulele generale pentru prima metodă.

l_{ii} = ales arbitrar (nenul), $i = 1, 2, .., n$

$$u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} l_{ik} \cdot u_{kj}}{l_{ii}}, \qquad j = i,\ i+1, \dots, n \qquad \text{(a)}$$

Dacă $u_{ii} \neq 0$, $i = \overline{1, n-1}$, rezultă:

$$l_{ji} = \frac{a_{ji} - \sum_{k=1}^{i-1} l_{jk} \cdot u_{ki}}{u_{ii}}, \qquad j = i+1, \dots, n \qquad \text{(b)}$$

Elementele lui **L** si **U** se determină în următoarea secvenţă, conform schemei de mai jos (se calculează elementele din linia *i* din **U,** şi coloana *i* din **L**).

$$
\begin{array}{c|l}
l_{ii} & \rightarrow \quad u_{i,i} \quad u_{i,i+1} \quad \ldots \quad u_{i,n} \\
\hline
\downarrow & \\
l_{i+1,i} & \\
\vdots & \\
l_{n,i} &
\end{array}
$$

În particular, la $i = n$, se calculează numai u_{nn} - din (a).

Numărul de operaţii este acelaşi ca în eliminarea Gauss ($\approx n^3/3$).

Două metode se utilizează în practică:

l_{ii} = 1 ... Metoda *Doolittle* (descompunerea **LU** revine la cea din eliminarea Gauss).

u_{ii} = 1 ... Metoda *Crout*.

Posibilitatea descompunerii LU

Propoziţie

Descompunerea **LU** există dacă şi numai dacă toate sub-matricile *principale* $\mathbf{A}(1:i,1:i),\ i = \overline{1,n-1}$ sunt nesingulare ■

2.2 Pivotare la descompunerea LU

Pivotare parţială: A pivota în descompunerea **LU** inseamnă ca, la fiecare pas *i*, să căutm *pivotul* u_{ii} în coloana *i* a lui **A**, în liniile $i, i+1, \ldots, n$, şi apoi să permutăm două linii.

Pivotarea trebuie făcută înainte de calculul elementelor lui **U** şi **L**. Adică, pivotul u_{ii} trebuie calculat şi testat *înainte* de aplicarea formulelor (a) şi (b).

Pentru metoda Doolittle, pivotul este dat de (cf. a):

$$u_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik} \cdot u_{ki}$$

Aşa cum s-a arătat la metoda Gauss, trebuie pivotat şi dacă pivotul este foarte mic. Practic, se testează dacă pivotul este mai mic decât un *prag*.

Observaţia de la pivotarea în eliminarea Gauss se aplică).

Determinantul matricii **A** (în metoda Doolittle) este

$$\det(\mathbf{A}) = (-1)^{n\_l}\ u_{11}u_{22}\ldots u_{nn},$$

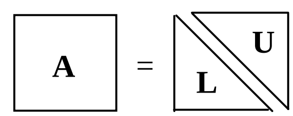
unde *n_l* este numărul de schimbări de linii în cursul pivotării.

■

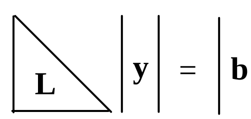
2.3 Metoda

Rezolvarea sistemului prin descompunerea **LU**, constă în paşii:

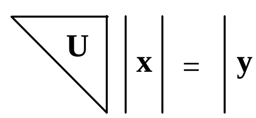
1. Factorizare $\mathbf{A} = \mathbf{LU}$, cu pivotare parţială.



2. Rezolvarea sistemului $\mathbf{Ly} = \mathbf{b}$, prin *substituţie înainte*; rezultă **y**.



3. Rezolvarea sistemului $\mathbf{Ux} = \mathbf{y}$, prin *substituţie înapoi*; rezultă **x.**



Observaţii

- Legătura cu eliminarea Gauss este **y** = **g.**
- Pentru noi termeni liberi **b**, se repetă numai paşii 2, 3 (n^2 operaţii suplimentare pentru fiecare **b**)
- Numărul de operaţii este acelaşi ca în eliminarea Gauss ($\approx n^3/3$).

■

3 Metoda Cholesky

Metoda Cholesky se aplică pentru un sistem cu matrice *simetrică şi pozitiv definită*.

Simetrie: $\mathbf{A}^T = \mathbf{A} \quad \Leftrightarrow \quad a_{ij} = a_{ji}, \quad i, j = \overline{1,n}$

Definire pozitivă: $\forall \mathbf{x} \neq \mathbf{0} \quad \mathbf{x}^T \mathbf{A}\mathbf{x} = \sum_i \sum_j a_{ij} x_i x_j > 0$

3.1 Proprietăţi ale matricilor simetrice şi pozitiv definite

Pentru o matrice *pozitiv definită*, au loc proprietăţile:

(1) Matricea este nesingulară.

(2) Elementele diagonale sunt pozitive: $a_{kk} > 0$. Mai mult, sub-matricile principale sunt pozitiv definite.

Urmează că a_{11} poate fi luat ca pivot în eliminarea Gauss, la pasul 1.

Pentru o matrice *simetrică şi pozitiv definită*, au loc şi proprietăţile:

(3) Sub-matricile $\mathbf{A}^{(k)}(k:n, k:n), \quad k \geq 2$, rezultate din eliminarea Gauss sunt simetrice şi pozitiv definite.

De aici, rezultă că există pivoţi $a_{kk}^{(k)} > 0$, la fiecare pas *k*.

(4) Valorile proprii ale lui **A** sunt reale şi pozitive.

Mai întâi, se arată că dacă **A** este simetrică, valorile proprii sunt reale – v. Cap. 5. Apoi, fie λ o valoare proprie şi **x** vectorul propriu asociat, atunci $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$. Inmulţind la stânga cu $\mathbf{x}^T$ şi ţinând cont de definirea pozitivă a matricii **A**, rezultă $\lambda \mathbf{x}^T \mathbf{x} > 0$, din care urmează că λ este pozitiv.

Observaţie

Reciproc, dacă **A** este simetrică şi valorile proprii sunt pozitive, atunci **A** este pozitiv definită. Demonstraţia cere rezultate de algebră matriceală care sunt în afara scopului acesui paragraf. V. de exemplu, R. Bellman, “Introducere in analiza matriceală”, E.T., Bucureşti, 1969.

(5) Determinatul matricii este pozitiv. Mai mult, toţi determinaţii principali sunt pozitivi.

Întrucât determinatul matricii este $\det(\mathbf{A}) = \lambda_1 \lambda_2 \ldots \lambda_n$ - v. Cap. 5, urmează că avem $\det(\mathbf{A}) > 0$. A doua afirmaţie decurge din faptul ca sub-matricile principale sunt şi ele pozitiv definite – conform Proprietăţii 2.

Observaţie

Reciproc, dacă matricea este simetrică şi toţi minorii principali sunt pozitivi, matricea este pozitiv definită. Pentru demonstraţie – v. Wilkinson-2.

Cu aceasta, rezultă:

Dacă **A** *este simetrică, o condiţie necesară şi suficientă ca* **A** *să fie pozitiv definită, este ca toţi determinaţii principali să fie pozitivi.*

■

3.2 Metoda Cholesky

Propoziţie

Dacă **A** este simetrică şi pozitiv definită, atunci există descompunerea

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T, \tag{1}$$

unde **L** este o matrice *inferior* triunghiulară, cu elemente diagonale *pozitive*.

■

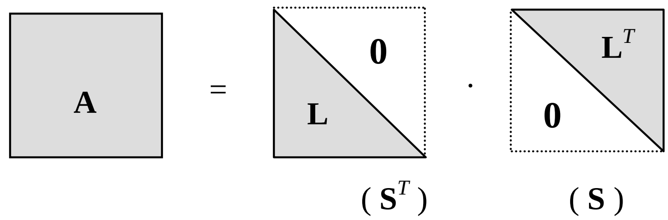
Cu alte cuvinte, în descompunerea **LU** se poate alege $\mathbf{U} = \mathbf{L}^T$. Notând

$$\mathbf{S} = \mathbf{L}^T,$$

unde **S** este *superior* triunghiulară, avem şi descompunerea

$$\mathbf{A} = \mathbf{S}^T\mathbf{S} \tag{2}$$

Cele două forme (1) şi (2) reprezintă aceeaşi descompunere, luând ca referinţă una sau alta din cele două matrici triunghiulare. Ele sunt ilustrate mai jos. $\mathbf{0}$ desemnează elementele nule.



■

Exemplu – Evaluarea lui **L**, pentru n = 3:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}$$

Ecuaţii: Lucrăm cu triunghiul *inferior* al lui **A**, luând elementele *pe coloane*:

$$a_{11} = l_{11}{}^2, \quad a_{21} = l_{21} \cdot l_{11}, \quad a_{31} = l_{31} \cdot l_{11}$$

$$a_{22} = l_{21}{}^2 + l_{22}{}^2, \quad a_{32} = l_{31} \cdot l_{21} + l_{32} \cdot l_{22}$$

$$a_{33} = l_{31}{}^2 + l_{32}{}^2 + l_{33}{}^2$$

Rezultă elementele lui **L**, *pe coloane*:

$$l_{11} = \sqrt{a_{11}}; \qquad l_{21} = \frac{a_{21}}{l_{11}}, \qquad l_{31} = \frac{a_{31}}{l_{11}}.$$

$$l_{22} = (a_{22} - l_{21}{}^2)^{1/2}; \qquad l_{32} = \frac{a_{32} - l_{31} \cdot l_{21}}{l_{22}}$$

$$l_{33} = (a_{33} - l_{31}{}^2 - l_{32}{}^2)^{1/2}$$

Analog, se poate lucra cu triunghiul superior al lui **A** şi determina elementele lui **L**, pe linii.

Exempu numeric:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 13 & 18 \\ 3 & 18 & 50 \end{bmatrix}; \qquad \mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 3 & 0 \\ 3 & 4 & 5 \end{bmatrix}$$ ■

Paşii rezolvării sunt aceeaşi ca la descompunerea **LU**, anume:

- Cu matricea **L** (unde $\mathbf{U} = \mathbf{L}^T$):

1) *Factorizare* (descompunere): $\mathbf{A} = \mathbf{L}\mathbf{L}^T$
2) *Substituţie înainte* (rezultă **y**): $\mathbf{L}\mathbf{y} = \mathbf{b}$
3) *Substituţie înapoi* (rezultă **x**): $\mathbf{L}^T\mathbf{x} = \mathbf{y}$.

- Cu matricea **S** (unde $\mathbf{U} = \mathbf{S}$; $\mathbf{L} = \mathbf{S}^T$):

1) $\mathbf{A} = \mathbf{S}^T\mathbf{S}$
2) $\mathbf{S}^T\mathbf{y} = \mathbf{b}$
3) $\mathbf{S}\mathbf{x} = \mathbf{y}$

■

3.3 Numărul de operaţii

Numărul de operaţii (adunări/scăderi), pentru *n* = 'mare', este

$$NOP_{Cholesky} \approx \frac{n^3}{6}$$

Acesta este cca. **½ din numărul de operaţii cerute eliminarea Gauss (sau LU)**.

(În afară de adunări/scăderi, se mai calculează *n* rădăcini pătrate.)

Alte avantaje, în raport cu descompunerea generală **LU**:

- Matricea **A** fiind simetrică, se poate stoca numai triunghiul inferior (sau superior), cerând numai $\frac{1}{2}n(n+1)$ locaţii de memorie. În aceste locaţii se stochează matricea **L** (sau **S**).
- Nu este necesară pivotarea – conform proprietăţilor (2, 3).

Observație

Rădăcinile pătrate – care consumă timp-calculator – pot fi evitate printr-o uşoară modificare a descompunerii **LU**, şi anume: căutăm o matrice inferior triunghiulară $\tilde{\mathbf{L}}$ cu 1 pe diagonală, şi o *matrice diagonală* **D**, astfel ca:

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

Într-adevăr: dacă avem $\mathbf{A} = \mathbf{L}\mathbf{L}^T$, fie $\tilde{\mathbf{L}}$ matricea obţinută din **L** punînd pe diagonală $\tilde{l}_{ii} = 1$ (în rest, $\tilde{l}_{ij} = l_{ij}$), şi $\mathbf{D}' = diag(l_{ii})$. Avem $\mathbf{L} = \tilde{\mathbf{L}}\mathbf{D}'$, şi $\mathbf{L}\mathbf{L}^T = \tilde{\mathbf{L}}\mathbf{D}'\mathbf{D}'\tilde{\mathbf{L}}^T$. Astfel, matricea $\mathbf{D} = \mathbf{D}'\mathbf{D}' = diag(l_{ii}^2)$.

Această factorizare se poate face cu aproximativ acelaşi număr de operaţii ca în metoda Cholesky, şi fără calcul de rădăcini pătrate ■