Laborator 3

Calculul factorizărilor LU. Calculul inversei şi al determinantului

3.1 Tema

Calculul factorizărilor LU ale unei matrice şi utilizarea lor în rezolvarea numerică a sistemelor liniare determinate a calculului inversei şi a determinantului. Calculul factorizării Cholesky pentru cazul particular, dar important, al matricelor simetrice pozitiv definite.

3.2 Preliminarii

3.2.1 Factorizări LU

În multe situaţii, se dovedeşte a fi convenabil de a exprima o matrice dată sub forma unui produs de două matrice triunghiulare.

Definiţia 3.1 Fie dată $A \in \mathbf{R}^{n \times n}$. Dacă există o matrice inferior triunghiulară $L \in \mathbf{R}^{n \times n}$ şi o matrice superior triunghiulară $U \in \mathbf{R}^{n \times n}$ astfel încât

$$A = LU,$$

atunci expresia de mai sus se numeşte factorizarea LU a matricei A.

Definiţia 3.2 Dacă $A \in \mathbf{R}^{n \times n}$, atunci expresia

$$A = LDU,$$

unde $L \in \mathbf{R}^{n \times n}$ este inferior triunghiulară *unitară*, $U \in \mathbf{R}^{n \times n}$ este superior triunghiulară *unitară* şi $D \in \mathbf{R}^{n \times n}$ este o matrice diagonală, se numeşte factorizarea LDU a matricei A.

Condiţile de existenţă şi unicitate a factorizării LDU sunt date de:

Teorema 3.3 *O matrice $A \in \mathbf{R}^{n\times n}$ admite factorizare LDU unică dacă şi numai dacăf toate submatricele lider principale $A^{[k]}$, $k = 1 : n-1$, sunt nesingulare.*

Factorizările practice LU asociază matricea diagonală D din factorizarea LDU:

- cu matricea U, definind aşa numita factorizare *Doolittle* $A = LU$, unde L este inferior triunghiulară *unitară* şi U este superior triunghiulară;
- sau cu matricea L, definind factorizarea *Crout* $A = LU$, unde L este inferior triunghiulară şi U superior triunghiulară *unitară.*

Din teorema 3.3 rezultă că ambele factorizări, Doolittle şi Crout, există şi sunt unice dacă şi numai dacă submatricele lider principale $A^{[k]}$, $k = 1 : n-1$, sunt toate nensingulare.

Întrucât algoritmul de triangularizare prin eliminare gaussiană fără pivotare calculează implicit şi factorizarea Doolittle, mai jos va fi prezentat numai algoritmul Crout de factorizare LU.

3.2.2 Factorizarea Cholesky

Un caz special important de factorizare LU care poate fi calculată cu un algoritm eficient şi sigur se referă la matrice pozitiv definite.

O matrice simetrică ($A = A^T$) A se numeşte *pozitiv definită* dacă pentru orice vector nenul $x \in \mathbf{R}^n$ avem $x^T Ax > 0$.

Orice submatrice principală a unei matrice pozitiv definite este pozitiv definită. Prin urmare, toate submatricele lider principale ale unei matrice simetrice pozitiv definite şi matricea însăşi sunt nesingulare.

Baza teoretică pentru descompunerea Cholesky este dată de următorul rezultat.

Teorema 3.4 *Pentru orice matrice simetrică pozitiv definită $A \in \mathbf{R}^{n\times n}$, există o matrice inferior triunghiulară unică $L \in \mathbf{R}^{n\times n}$ cu toate elementele diagonale pozitive astfel încât:*

$$A = LL^T,$$

care se numeşte factorizarea Cholesky a matricei A.

Factorizările LU sunt utile în multe probleme de calcul, cum sunt, de exemplu, rezolvarea sistemelor liniare, calculul inversei sau al determinantului unei matrice.

Presupunând că matricea dată $A \in \mathbf{R}^{n\times n}$ admite o factorizare LU (altfel se impune utilizarea strategiilor de pivotare) vom indica aici numai modalitatea de rezolvarea a sistemelor liniare $Ax = b$. Celelalte probleme sunt tratate în secţiuni distincte.

Soluţia sistemului liniar $Ax = b$ se poate calcula folosind următoarea schemă de calcul:

1. $A = LU$ (se calculează factorizarea LU)
2. Se rezolvă sistemul inferior triunghiular $Ly = b$
3. Se rezolvă sistemul superior triunghiular $Ux = y$

3.2.3 Algoritmi

ALGORITM 3.5 *(CROUT – factorizarea Crout)* (Dată matricea $A \in \mathbf{R}^{n\times n}$ cu toate submatricele lider principale $A^{[k]}$, $k = 1 : n-1$, nesingulare, acest algoritm calculează factorizarea Crout $A = LU$. Elementele matricelor L şi U suprascriu elementele corespondente ale lui A (mai puţin elementele diagonale ale lui U, care sunt egale cu 1).)

(0. $a_{i1} \leftarrow l_{i1} = a_{i1}, \quad i = 1 : n$)
1. **pentru** $j = 2 : n$
   1. $a_{1j} \leftarrow u_{1j} = a_{1j}/l_{11}$
2. **pentru** $k = 2 : n$
   1. **pentru** $i = k : n$
      1. $a_{ik} \leftarrow l_{ik} = a_{ik} - \sum_{l=1}^{k-1} l_{il}u_{lk}$
   2. **dacă** $k = n$ **atunci stop**
   3. **pentru** $j = k+1 : n$
      1. $a_{kj} \leftarrow u_{kj} = \left(a_{kj} - \sum_{l=1}^{k-1} l_{kl}u_{lj}\right)/l_{kk}$

Pentru a evita necesitatea ca $A^{[k]}$, $k = 1 : n-1$, să fie nensingulare, se poate utiliza numai pivotarea liniilor care pot fi introduse după instrucţtiunea 2.1 a algoritmului de mai sus.

ALGORITM 3.6 *(CHOL – factorizarea Cholesky)* (Dată o matrice simetrică $A \in \mathbf{R}^{n\times n}$, acest algoritm stabileşte dacă matricea este pozitiv definită. Dacă da, atunci suprascrie triunghiul inferior al lui A cu matricea L din factorizarea Cholesky $A = LL^T$.)

1. **dacă** $a_{11} \leq 0$ **then**
   1. Print('A nu este pozitiv definită')
   2. **stop**
2. $a_{11} \leftarrow l_{11} = \sqrt{a_{11}}$
3. **pentru** $i = 2 : n$
   1. $a_{i1} \leftarrow l_{i1} = a_{i1}/l_{11}$
4. **pentru** $k = 2 : n$
   1. $\alpha \leftarrow a_{kk} - \sum_{l=1}^{k-1} l_{kl}^2$
   2. **dacă** $\alpha \leq 0$ **atunci**
      1. Print('A nu este pozitiv definită')
      2. **stop**
   3. $a_{kk} \leftarrow l_{kk} = \sqrt{\alpha}$
   4. **dacă** $k = n$ **atunci stop**
   5. **pentru** $i = k+1 : n$
      1. $a_{ik} \leftarrow l_{ik} = \left(a_{ik} - \sum_{l=1}^{k-1} l_{il}l_{kl}\right)/l_{kk}$

Algoritm de mai sus furnizează cel mai bun test al pozitiv definirii unei matrice.

3.2.4 Inversarea matricelor nesingulare

Un mod eficient de calcul al inversei unei matrice nesingulare apelează la eliminarea gaussiană cu pivotare parţială (algoritmul GPP). Un alt mod, la fel de eficient de a calcula inversa constă în utilizarea unei factorizări LU (eventual cu pivotare). De asemenea, pentru calculul inversei avem nevoie de algoritmii de inversare a matricelor triunghiulare.

Inversarea matricelor triunghiulare

Se poate arăta că inversa unei matrice triunghiulare nesingulare este o matrice triunghiulară de acelaşi tip iar calculul ei se poate face rezolvând un set de sisteme triunghiulare de diferite dimensiuni dar de acelaşi tip. Dacă se respectă o ordine adecvată de calcul elementele matricei inverse pot suprascrie elementele matricei date pe măsură ce sunt calculate. Trimiţând studentul la cursul predat sau publicat pentru detalii, ne mărginim să prezentăm direct algoritmii de inversare.

ALGORITM 3.7 *(LINV)* (Dată o matrice inferior triunghiulară nesingulară $L \in \mathbf{R}^{n \times n}$, acest algoritm suprascrie L cu inversa sa.)

1. **pentru** $j = 1 : n$
   1. $l_{jj} \leftarrow x_{jj} = 1/l_{jj}$
   2. **dacă** $j = n$ **atunci stop**
   3. **pentru** $i = j + 1 : n$
      1. $l_{ij} \leftarrow x_{ij} = -\left(\sum_{k=j}^{i-1} l_{ik} x_{kj}\right)/l_{ii}$

Numărul de flopi necesar este $N_{LINV} =\approx \frac{n^3}{3}$, iar memoria ocupată este $M_{LINV} \approx n^2/2$ locaţii în format virgulă mobilă.

ALGORITM 3.8 *(UINV)* (Dată o matrice superior triunghiulară nesingulară $U \in \mathbf{R}^{n \times n}$, acest algoritm suprascrie U cu inversa sa.)

1. **pentru** $j = n : -1 : 1$
   1. $u_{jj} \leftarrow x_{jj} = 1/u_{jj}$
   2. **dacă** $j = 1$ **atunci stop**
   3. **pentru** $i = j - 1 : -1 : 1$
      1. $u_{ij} \leftarrow x_{ij} = -\left(\sum_{k=i+1}^{j} u_{ik} x_{kj}\right)/u_{ii}$

Ca şi în cazul inferior triunghiular, $N_{UINV} \approx n^3/3$ şi $M_{UINV} \approx n^2/2$.

Inversarea matricelor utilizând GPP

Dacă A este o matrice reală nesingulară $n \times n$, atunci eliminarea gaussiană cu pivotare parţială produce o matrice superior triunghiulară U:

$$M_{n-1}P_{n-1}\dots M_1P_1A = U,$$

unde M_k sunt matrice inferior triunghiulare elemntare şi P_k sunt matrice de permutare elementare. Avem

$$X \equiv A^{-1} = U^{-1}M_{n-1}P_{n-1}\dots M_1P_1$$

şi, deci, inversa matricei A poate fi calculată cu schema

1. $[U, M, p] = GPP(A)$
2. $X = UINV(U)$
3. **pentru** $k = n-1 : -1 : 1$
   1. $X \leftarrow XM_k$
   2. $X \leftarrow XP_k$

careia îi corespunde următorul algoritm ce calculează inversa pe loc, peste matricea A.

Algoritm 3.9 *(INV_GPP)* (Dată o matrice nesingulară $A \in \mathbf{R}^{n\times n}$, acest algoritm suprascrie A cu inversa sa, A^{-1}.)

1. $[U, M, p] = GPP(A)$ $\{U$ şi M suprascriu $A\}$
2. $A \leftarrow X = UINV(U)$ {se modifică numai triunghiul superior al lui A}
3. **pentru** $k = n-1 : -1 : 1$
   1. **for** $i = k+1 : n$
      1. $m_i \leftarrow \mu_{ik}$ {salvarea multiplicatorilor}
   2. **pentru** $i = 1 : k$
      1. $a_{ik} \leftarrow a_{ik} - \sum_{l=k+1}^{n} a_{il}m_l$
   3. **for** $i = k+1 : n$
      1. $a_{ik} \leftarrow -\sum_{l=k+1}^{n} a_{il}m_l$
   4. **dacă** $p(k) \neq k$ **atunci**
      1. **pentru** $i = 1 : n$
         1. $a_{ik} \leftrightarrow a_{i,p(k)}$

Cea de a doua metodă de inversare pe care o prezentăm se bazează pe factorizarea LU.

Inversarea matricelor utilizând factorizări LU

Dacă matricea nesingulară A admite o factorizare LU, atunci

$$A^{-1} = U^{-1}L^{-1},$$

deci o procedură de inversare poate fi următoarea:

1. $A = LU$ (se calculează factorizarea LU)
2. $U \leftarrow X = U^{-1}$
3. $L \leftarrow Y = L^{-1}$
3. $A^{-1} \leftarrow XY$

3.2.5 Calculul determinantului

Algoritmii de eliminare gauşiană cu pivotare GPP ssii GCP sunt utile pentru un calcul eficient al determinantului unei matrice. Baza este dată de cunoscuta proprietate conform căreia

$$\det(A \cdot B) = \det(A) \cdot \det(B)$$

care rămâne adevărată pentru orice număr finit de matrice pătrate. De asemenea, dacă $A \in \mathbf{R}^{n\times n}$ este o matrice triunghiulară, atunci

$$\det(A) = \prod_{i=1}^{n} a_{ii}.$$

În sfârşit permutarea a două linii sau a două coloane schimbă semnul determinantului sau, echivalent, dacă P este o matrice de permutare elementară, atunci

$$\det(P) = -1.$$

Matricele inferior triunghiulare elementare M_k, $k = 1 : n-1$, au $\det(M_s) = 1$ şi, deci,

$$\det(A) = (-1)^k \det(U) = (-1)^k \prod_{i=1}^{n} u_{ii},$$

unde $k \leq n-1$ este numărul matricelor de permutare elementare proprii (i.e. $P \neq I_n$). Algoritmul corespunzător este:

ALGORITM 3.10 *(DET)* (Dată o matrice $A \in \mathbf{R}^{n\times n}$, acest algoritm calculează determinantul matricei folosind reducerea gaussiană cu pivotare parţială la forma triunghiulară.)

1. $[U, M, p] = GPP(A)$
2. $det \leftarrow 1$
3. **pentru** $k = 1 : n$
    1. $det \leftarrow det \cdot u_{kk}$
4. **pentru** $k = 1 : n-1$
    1. **dacă** $p(k) \neq k$ **atunci**
        1. $det \leftarrow -det$

Dacă apelăm la o factorizare LU, atunci avem, evident,

$$\det(A) = \det(LU) = \det(L) \cdot \det(U) = \left(\prod_{i=1}^{n} l_{ii}\right)\left(\prod_{i=1}^{n} u_{ii}\right).$$

Astfel, folosind, de exemplu factorizarea Crout, determinantul $\delta = det(A)$ se poate calcula cu schema:

1. $A = LU$ (se calculează factorizarea Crout)
2. $\delta = \prod_{i=1}^{n} l_{ii}$

3.3 Sarcini de lucru

3.3.1 A. In laborator

1. Se elaborează şi editează programele MATLAB pentru implementarea algoritmilor de factorizare Crout şi Cholesky şi se testează corectitudinea algoritmilor şi programelor pentru matrice cu elemente aleatoare de ordin 7 şi 25.

2. Se scriu programele de rezolvare a unui sistem liniar cu ajutorul factorizării Crout şi, în cazul unei matrice a coeficienţilor pozitiv definite, cu ajutorul factorizării Cholesky, folosind funcţiile proprii de rezolvare a sistemelor triunghiulare. Se testează aceste programe pentru sisteme de 7 şi 25 de ecuaţii cu 7, respectiv 25, necunoscute.

3. Se editează programul MATLAB pentru implementarea algoritmului de inversare a unei matrice folosind GPP-ul propriu, se testează corectitudinea rezultatelor obţinute pentru matrice cu elemente aleatoare de ordin 7 şi 25, comparând cu rezultatul oferit de funcţia MATLAB `inv`.

Indicaţii. Pentru crearea unor matrice pozitiv definite se poate folosi secvenţa.

1. $A = \text{rand}(n)$
2. $A = A' * A$

3.3.2 B. Acasă

1. Se va scrie un program MATLAB pentru calculul determinantului unei matrice folosind GPP şi factorizarea LU şi compararea rezultatelor oferite de cele două metode pentru matrice cu elemente aleatoare de ordin 7 şi 100. Comparaţi rezultatele dv. cu rezultatul oferit de funcţia MATLAB `det`.

2. Se consideră dată o matrice superior Hessenberg H. Se cer programe MATLAB eficiente pentru calculul factorizării Crout, calculul inversei şi al determinantului lui H.

3. Se consideră date o matrice tridiagonală pozitiv definită $T \in \mathbf{R}^{n\times n}$ şi un vector $b \in \mathbf{R}^n$. Se cer programe MATLAB eficiente pentru calculul factorizării Cholesky, calculul inversei şi al determinantului lui T, precum şi pentru rezolvarea sistemului $Tx = b$.