

Tema nr. 3

Se dau n dimensiunea sistemului, ε - precizia calculelor, matricea pătratică $A \in \mathbb{R}^{n \times n}$ și un vector $s \in \mathbb{R}^n$.

1. Să se calculeze vectorul $b \in \mathbb{R}^n$ astfel:

$$b_i = \sum_{j=1}^n s_j \cdot a_{ij} \quad , \quad i = 1, \dots, n$$

2. Să se implementeze descompunerea QR a matricei A folosind algoritmul lui Householder.
3. Să se rezolve sistemul liniar:

$$Ax = b \quad ,$$

folosind descompunerea QR din una din bibliotecile menționate în pagina laboratorului (se obține soluția x_{QR}) și descompunerea QR calculată la punctul 2. (se obține soluția $x_{Householder}$). Calculați și afișați:

$$\|x_{QR} - x_{Householder}\|_2 \cdot$$

4. Să se calculeze și să se afișeze următoarele erori ($\|\cdot\|_2$ este norma euclidiană):

$$\|A^{init} x_{Householder} - b^{init}\|_2 \quad ,$$

$$\|A^{init} x_{QR} - b^{init}\|_2 \quad ,$$

$$\frac{\|x_{Householder} - s\|_2}{\|s\|_2} \quad ,$$

$$\frac{\|x_{QR} - s\|_2}{\|s\|_2} \quad .$$

(aceste valori ar trebui să fie mai mici decât 10^{-6})

5. Să se calculeze inversa matricei A folosind descompunerea QR calculată la punctul 2. și să se compare cu inversa calculată folosind funcția din bibliotecă, afișând valoarea următoarei norme:

$$\|A_{Householder}^{-1} - A_{bibl}^{-1}\| \quad .$$

6. Scrieți programul și cu inițializare random a datelor de intrare (astfel încât programul vostru să poată fi rulat cu orice dimensiune n a datelor).

Bonus (15pt): Considerați o matrice $A \in \mathbb{R}^{n \times n}$ simetrică. Calculați limita aproximativă a următorului șir de matrice:

- 1) $k=0, A^{(0)}=A$;
- 2) Pentru $A^{(k)}$ se calculează descompunerea QR cu algoritmul Householder, $A^{(k)} = Q^*R$;
- 3) $A^{(k+1)} = R^*Q$;
- 4) Limita aproximativă a acestui șir de matrice este matricea $A^{(k+1)}$ care satisface proprietatea:

$$\|A^{(k+1)} - A^{(k)}\| \leq \varepsilon.$$

Ce formă are matricea $A^{(k+1)}$ și ce reprezintă elementele acestei matrice pentru matricea inițială A ?

Restricții de calcul:

- a) Folosiți un număr cât mai mic de matrice la construcția șirului $A^{(k)}$;
- b) La înmulțirea matricială R^*Q folosiți doar elementele din partea superior triunghiulară a matricei R (r_{ij} cu $i \leq j$), și toate elementele matricei Q , fără a utiliza instrucțiuni „if-then-else”.

Rezolvarea sistemelor liniare folosind descompunerea QR

Fie A o matrice reală pătratică de dimensiune n . Presupunem că pentru matricea A avem o descompunere de forma:

$$A=Q^*R$$

unde Q este matrice ortogonală ($Q^TQ = QQ^T = I_n$) iar R este matrice superior triunghiulară. Având o asemenea descompunere pentru matricea A , rezolvarea sistemului $Ax=b$ se reduce la rezolvarea sistemului superior triunghiular $Rx=Q^Tb$.

$$Ax=b \leftrightarrow QRx = b \leftrightarrow Q^TQRx = Q^Tb \leftrightarrow Rx=Q^Tb$$

$$Ax=b \Leftrightarrow Rx=Q^Tb$$

Algoritmul lui Householder

Pentru a aduce sistemul $Ax=b$ la forma $Rx = Q^T b$ se folosesc matricele de reflexie.

O matrice de reflexie $P = (p_{ij})_{i,j=1,n}$ are următoarea formă:

$$P = I_n - 2vv^T, \quad v \in \mathbb{R}^n, \quad \|v\|_2 = |v| = \sqrt{\sum_{i=1}^n v_i^2} = 1.$$

(unde cu I_n am notat matricea unitate de dimensiune n)

Se poate arăta că matricele de reflexie sunt simetrice și ortogonale :

$$P = P^T, \quad P^2 = I_n.$$

Algoritmul Householder de calcul al descompunerii QR se desfășoară în $(n-1)$ pași. La pasul r se transformă coloana r a matricei A în formă superior triunghiulară fără a modifica primele $(r-1)$ coloane. La acest pas se obține coloana r a matricei R . Calculul matricei R se poate face direct în matricea A (de fapt se transformă matricea A într-o matrice superior triunghiulară). În același timp, se fac transformările necesare asupra vectorului termenilor liberi și se pot efectua transformări pentru a calcula matricea Q^T . Pentru a calcula matricea Q^T , se inițializează o matrice $\bar{Q} = I_n$ și se fac aceleași transformări asupra matricei \bar{Q} ca și cele făcute asupra matricei A .

Pasul r ($r=1,2,\dots,n-1$)

La acest pas, matricea A are primele $(r-1)$ coloane în formă superior triunghiulară și se folosește o matrice de reflexie pentru a transforma coloana r a matricei A în formă superior triunghiulară (fără a le modifica pe primele $(r-1)$).

Pasul r constă în următoarele calcule :

$$A = P_r A$$

$$b = P_r b$$

$$\bar{Q} = P_r \bar{Q}$$

unde matricea P_r se construiește astfel :

$$P_r = I_n - \frac{1}{\beta} u u^T$$

$$\beta = \sigma - k a_{rr}, \quad \sigma = \sum_{j=r}^n a_{jr}^2, \quad k = -\text{semn}(a_{rr}) \sqrt{\sigma}, \quad \text{semn}(x) = \begin{cases} +1 & \text{dacă } x \geq 0, \\ -1 & \text{dacă } x < 0. \end{cases}$$

$$u = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ a_{rr} - k \\ a_{r+1r} \\ \vdots \\ a_{nr} \end{pmatrix}, \quad \begin{aligned} u_i &= 0, i = 1, \dots, r-1, \\ u_r &= a_{rr} - k, \\ u_i &= a_{ir}, i = r+1, \dots, n \end{aligned}$$

Matricea $V = u u^T \in \mathbb{R}^{n \times n}$, $V = (v_{ij})_{i,j=1,\dots,n}$ are următoarele elemente:

$$v_{ij} = \begin{cases} 0 & \text{pentru } i = 1, \dots, r-1, j = 1, \dots, n \\ 0 & \text{pentru } i = r, \dots, n, j = 1, \dots, r-1 \\ u_i u_j & \text{pentru } i = r, \dots, n, j = r, \dots, n \end{cases}$$

Prin urmare, matricea de reflexie $\mathbf{P}_r = (p_{ij})_{i,j=1,n}$ are următoarele elemente:

$$p_{ij} = \begin{cases} 0 & \text{pentru } i = 1, \dots, r-1, j = 1, \dots, n, i \neq j \\ 1 & \text{pentru } i = 1, \dots, r-1, j = i \\ 0 & \text{pentru } i = r, \dots, n, j = 1, \dots, r-1 \\ -f u_i u_j & \text{pentru } i = r, \dots, n, j = r, \dots, n, i \neq j \\ 1 - f u_i^2 & \text{pentru } i = r, \dots, n, j = i \end{cases}, \quad f = \frac{1.0}{\beta}$$

La sfârșitul algoritmului de mai jos, matricea superior triunghiulară \mathbf{R} se găsește în matricea \mathbf{A} , iar transpusa matricei \mathbf{Q} în matricea $\bar{\mathbf{Q}}$. În vectorul \mathbf{b} vom avea $\mathbf{Q}^T \mathbf{b}^{init}$. Verificarea faptului că matricea inițială nu este singulară, se face testând dacă toate elementele de pe diagonala matricei \mathbf{R} (sau \mathbf{A}) sunt nenule ($|r_{ii}| \leq \varepsilon, \forall i$).

Algoritmul Householder pentru factorizarea $A=QR$

```

 $\tilde{Q} = I_n$ ;
for  $r = 1, \dots, n-1$ 
    // construcția matricei  $P_r$  – constanta  $\beta$  și vectorul  $u$ 
    •  $\sigma = \sum_{i=r}^n a_{ir}^2$ ;
    • if (  $\sigma \leq \varepsilon$  ) break ; //  $r = r + 1 \leftrightarrow P_r = I_n$  ( $A$  singulară)
    •  $k = \sqrt{\sigma}$ ;
    • if (  $a_{rr} \geq 0$  )  $k = -k$ ;
    •  $\beta = \sigma - k a_{rr}$ ;
    •  $u_r = a_{rr} - k$ ;  $u_i = a_{ir}$ ,  $i = r + 1, \dots, n$ ;
    //  $A = P_r * A$ 
    // transformarea coloanelor  $j = r + 1, \dots, n$ 
    • for  $j = r + 1, \dots, n$ 
        *  $\gamma = (\gamma_j / \beta) = (Ae_j, u) / \beta = (\sum_{i=r}^n u_i a_{ij}) / \beta$ ;
        * for  $i = r, \dots, n$ 
             $a_{ij} = a_{ij} - \gamma * u_i$ ;
    // transformarea coloanei  $r$  a matricei  $A$ 
    •  $a_{rr} = k$ ;  $a_{ir} = 0$ ,  $i = r + 1, \dots, n$ ;
    //  $b = P_r * b$ 
    •  $\gamma = (\gamma / \beta) = (b, u) / \beta = (\sum_{i=r}^n u_i b_i) / \beta$ ;
    • for  $i = r, \dots, n$   $b_i = b_i - \gamma * u_i$ ;
    //  $\tilde{Q} = P_r * \tilde{Q}$ 
    • for  $j = 1, \dots, n$ 
        *  $\gamma = (\tilde{Q}e_j, u) / \beta = (\sum_{i=r}^n u_i \tilde{q}_{ij}) / \beta$ ;
        * for  $i = r, \dots, n$ 
             $\tilde{q}_{ij} = \tilde{q}_{ij} - \gamma * u_i$ ;

```

Calculul unei aproximări a inversei unei matrice

Dacă se cunoaște o metodă numerică de rezolvare a sistemelor liniare (în cazul de față se va folosi descompunerea QR calculată folosind algoritmul lui Householder), coloanele matricei inverse se pot aproxima rezolvând n sisteme liniare.

Coloana j a matricei A^{-1} se aproximează rezolvând sistemul liniar:

$$Ax = e_j, \quad e_j = (0, 0, \dots, \underset{\text{poziția } j}{1}, \dots, 0)^T, \quad j = 1, \dots, n.$$

Procedura de calcul a matricei $A_{Householder}^{-1}$ este următoarea:

- Se calculează descompunerea QR a matricei A folosind algoritmul Householder
- Dacă matricea A este singulară ($\det A = 0 \leftrightarrow \exists |r_{ii}| < \varepsilon$), STOP, inversa nu se poate calcula.
- Altfel, se calculează coloanele matricei $A_{Householder}^{-1}$ astfel:

for $j=1, \dots, n$

1. Se inițializează vectorul $b = Q^T e_j =$ coloana j a matricei Q^T sau linia j a matricei Q ;
(se folosește inițializarea cu coloana j a matricei Q^T sau linia j a matricei Q în funcție de matricea returnată de algoritmul Householder)
2. Se rezolvă sistemul superior triunghiular $Rx=b$, folosind metoda substituției inverse. Se obține soluția x^* (x^* este și soluția sistemului liniar $Ax=e_j$);
3. Se memorează x^* în coloana j a matricei $A_{Householder}^{-1}$.

Procedura descrisă mai sus detaliază rezolvarea numerică, folosind o descompunere QR , a ecuației matriceale:

$$AX=I_n.$$

Exemplu

$$A = \begin{pmatrix} 0 & 0 & 4 \\ 1 & 2 & 3 \\ 0 & 1 & 2 \end{pmatrix}, \quad s = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}$$

$$\begin{aligned} 4x_3 &= 4 \\ x_1 + 2x_2 + 3x_3 &= 10 \\ x_2 + 2x_3 &= 4 \end{aligned}$$

$$A = \begin{pmatrix} 0 & 0 & 4 \\ 1 & 2 & 3 \\ 0 & 1 & 2 \end{pmatrix} = QR = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} * \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 4 \end{pmatrix}, \quad b = A * s = \begin{pmatrix} 4 \\ 10 \\ 4 \end{pmatrix}$$

$$Ax = b \Leftrightarrow Rx = Q^T b \Leftrightarrow \begin{aligned} x_1 + 2x_2 + 3x_3 &= 10 \\ x_2 + 2x_3 &= 4 \\ 4x_3 &= 4 \end{aligned} \Rightarrow x^* = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}$$

Prima coloană a matricei $A_{Householder}^{-1}$

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 0 \\ x_2 + 2x_3 &= 0 \\ 4x_3 &= 1 \end{aligned} \Rightarrow x^* = \begin{pmatrix} 0.25 \\ -0.5 \\ 0.25 \end{pmatrix}$$

Coloana a 2-a a matricei $A_{Householder}^{-1}$

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 1 \\ x_2 + 2x_3 &= 0 \\ 4x_3 &= 0 \end{aligned} \Rightarrow x^* = \begin{pmatrix} 1.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

Coloana a 3-a a matricei $A_{Householder}^{-1}$

$$\begin{array}{rcrcrcrcrcrcl} x_1 & + & 2x_2 & + & 3x_3 & = & 0 \\ & & x_2 & + & 2x_3 & = & 1 \\ & & & & 4x_3 & = & 0 \end{array} \Rightarrow x^* = \begin{pmatrix} -2.0 \\ 1.0 \\ 0.0 \end{pmatrix}$$

$$A_{Householder}^{-1} = \begin{pmatrix} 0.25 & 1.0 & -2.0 \\ -0.5 & 0.0 & 1.0 \\ 0.25 & 0.0 & 0.0 \end{pmatrix}$$