Cap. 4 Calculul valorilor şi vectorilor proprii

4.1 Formularea problemei

- Se consideră o matrice reală, pătratică, de ordin *n*: $A \in \Re^{n \times n}$

Definiţie:

Oricare ar fi matricea $A \in \Re^{n \times n}$ *, un număr în general complex,* λ *, se numeşte* ***valoare proprie*** *a matricei A dacă există un vector,* $\underline{x} \in \mathbf{C}^{n \times 1}$*,* $\underline{x} \neq \underline{0}_n$ *astfel încât:*

$$A \cdot \underline{x} = \lambda \cdot \underline{x}$$

$\underline{x}$ – vector propriu al matricii A asociat valorii proprii λ

$$(\lambda \cdot I_n - A) \cdot \underline{x} = \underline{0}_n \qquad \underline{x} \neq \underline{0}_n$$

$$\Downarrow$$

$\lambda \cdot I_n - A$ - singulară

$$\Downarrow$$

ecuaţie caracteristică $\longrightarrow$ $p_n(\lambda) = \det(\lambda \cdot I_n - A) = 0$

polinom caracteristic $\longrightarrow$ $p_n(\lambda) = \lambda^n + \alpha_1 \cdot \lambda^{n-1} + \cdots + \alpha_{n-1} \cdot \lambda + \alpha_n \qquad \alpha_i \in \Re, i = 1, \ldots, n$

Teoremă de existenţă:

Orice matrice pătratică reală, de ordin n, *are exact* n *valori proprii, în general complexe şi nu neapărat distincte, care coincid cu rădăcinile polinomului caracteristic ataşat matricei. Dacă există valori proprii complexe, atunci acestea apar în perechi complex conjugate.*

Orice matrice $A \in \Re^{n \times n}$ *are cel puţin un vector propriu.*

□ Nu se recomandă calculul numeric al valorilor proprii prin rezolvarea ecuaţiei caracteristice deoarece:

- rezolvarea ecuaţiilor polinomiale este o problemă prost condiţionată;
- coeficienţii polinomului caracteristic □ volum mare de calcule □ erori

□ <u>Metodele practice</u> pentru calculul numeric al valorilor proprii □ <u>proceduri iterative</u>

- matricea A adusă la <u>*formă canonică Schur*</u> prin transformări ortogonale de asemănare

4.2 Forma canonică Schur

Definiţie:

Două matrici, $A, A' \in \Re^{n \times n}$ *, se numesc ortogonal asemenea, dacă există o matrice ortogonală ,* $Q \in \Re^{n \times n}$ *astfel încât:*

$$A' = Q^T \cdot A \cdot Q$$

- **Proprietăţi:**

❶ Matricile ortogonal asemenea au aceleaşi valori proprii:

$$\lambda_i(A) = \lambda'_i(A'), \quad i = 1, \dots, n$$

❷ Relaţia dintre vectorii proprii ai două matrici ortogonal asemenea:

$$\underline{x}_i(A) = Q \cdot \underline{x}'_i(A'), \quad i = 1, \dots, n$$

Definiţie:

O matrice se spune că este în formă bloc superior triunghiulară, dacă are următoarea structură:

$$T = \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1m} \\ \mathbf{0} & T_{22} & \cdots & T_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & T_{mm} \end{bmatrix}, \quad T_{ii} \in \Re^{p_i \times p_i}$$

$T_{ii}, i = 1, \dots, m$ - matrici pătratice

$p_i \in \{1, 2\}, \forall i = \overline{1, m}$ $\longrightarrow$ *formă cvasi-superior triunghiulară*

METODE NUMERICE – curs 6

Teoremă de existenţă:

Oricare ar fi matricea $A \in \Re^{n\times n}$ *, există o matrice ortogonală* $\widetilde{Q} \in \Re^{n\times n}$*, astfel încât matricea:*

<u>forma canonică Schur reală</u> a matricei A $\longrightarrow$ $S = \widetilde{Q}^T \cdot A \cdot \widetilde{Q}$

este în formă cvasi-superior triunghiulară. Blocurile diagonale de ordin întâi ale matricei S *reprezintă valorile proprii reale ale matricei* A *şi ale matricei* S*, iar blocurile diagonale de ordin doi au valori proprii complex conjugate reprezentând valori proprii complex conjugate ale matricelor* A *şi* S*.*

Observaţii:

❶ coloanele matricii $\widetilde{Q} \in \Re^{n\times n}$ $\underline{\widetilde{q}}_k, k = 1,\ldots,n$ <u>*vectori Schur ai matricii A*</u>

❷ Demonstraţia teoremei este constructivă **algoritmul QR**

4.3 Algoritmul QR pentru calculul formei canonice Schur

- **Principiu:** construcţia unui şir de matrici ortogonal asemenea convergent la forma canonică Schur

$$A_0 = A, A_1, \ldots, A_k, \ldots \qquad A_k \xrightarrow{k\to\infty} S$$

$$\Downarrow$$

$$A_k = \left[a_{ij}^{[k]}\right]_{1\le i,j\le n} \qquad a_{ij}^{[k]} \xrightarrow{k\to\infty} 0 \qquad i \ge j+2$$

$$\exists\ a_{i+1,i}^{[k]} \xrightarrow{k\to\infty} 0 \qquad i=1,\ldots,n-1$$

 se defineşte şirul de matrici $\{A_k\}_{k\geq 0}$, $A_0 = A$

pas QR cu deplasare explicită → $A_k - \mu_k \cdot I_n = Q_k \cdot R_k, \quad A_{k+1} = R_k \cdot Q_k + \mu_k \cdot I_n$

$Q_k \in \Re^{n\times n}$ - ortogonală; $R_k \in \Re^{n\times n}$ - superior triunghiulară

$\mu_k \in \Re$ - deplasare (cu rol de accelerare a convergenţei)

Propoziţie:

Matricele şirului QR sunt ortogonal asemenea: $A_{k+1} = Q_k^T \cdot A_k \cdot Q_k$

 Observaţie:

- algoritmul original QR consumator de timp se foloseşte o formă optimizată

Forma implementabilă a algoritmului QR cu deplasare explicită

 parcurge <u>două faze de lucru</u>:

❶ faza 1 – pregătitoare (zerorizare elemente de sub sub-diagonala principală)

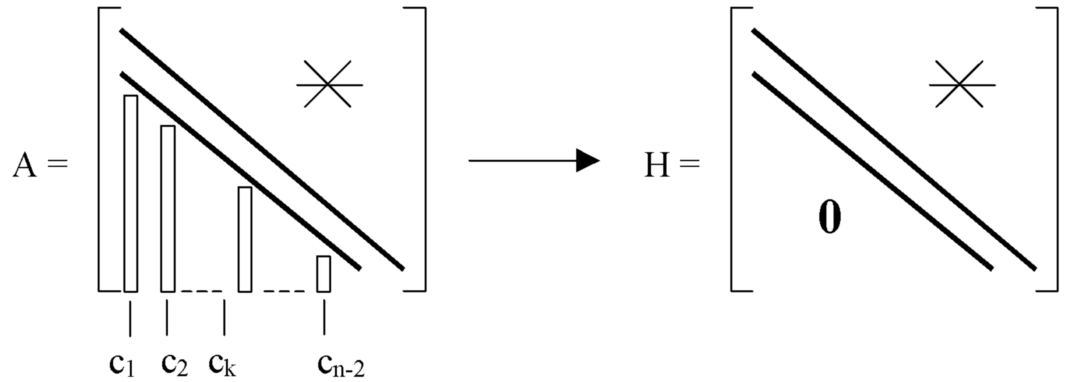
❷ faza 2 – aplicare algoritm QR matricii obţinută la faza 1

↓

se obţine forma canonică Schur

□ ***Faza 1***

- procedură directă de aducere a matricei A la forma superior Hessenberg (H)



- se folosesc matrici Householder în scopul anulării, coloană cu coloană, a elementelor matricei A situate sub sub-diagonala principală

- *algoritm*:

atribuie $A_1 \leftarrow A$
pentru $k = 1, n-2$ execută
⌈ * determinare matrice Householder astfel încât:
| $(U_{k+1} \cdot A_k)_{i,k} = 0, i = k+2, \ldots, n$
| atribuie $A'_{k+1} \leftarrow U_{k+1} \cdot A_k$
| atribuie $A_{k+1} \leftarrow A'_{k+1} \cdot U_{k+1}^T = A'_{k+1} \cdot U_{k+1}$
⌊ •
atribuie $H \leftarrow A_{n-1}$

METODE NUMERICE – curs 6

· sinteza matricii Householder, U_{k+1}

$$U_{k+1} = I_n - (\underline{u}_{k+1} \cdot \underline{u}_{k+1}^T / \beta_{k+1}) \quad \left\{ \begin{array}{l} \underline{u}_{k+1} = [0 \ \cdots \ 0 \ \ u_{k+1,k+1} \ \cdots \ u_{n,k+1}]^T \\ \sigma_{k+1} = \mathrm{sign}(a_{k+1,k}^{[k]}) \cdot \sqrt{\sum_{i=k+1}^{n} (a_{i,k}^{[k]})^2}, \ u_{k+1,k+1} = a_{k+1,k}^{[k]} + \sigma_{k+1} \\ u_{i,k} = a_{i,k}^{[k]}, \ i = k+2,\ldots,n, \ \beta_{k+1} = \sigma_{k+1} \cdot u_{k+1,k+1} \end{array} \right.$$

· tabloul general al transformărilor:

$$\underbrace{U_{n-1} \cdot \cdots \cdot U_2}_{U} \cdot A \cdot \underbrace{U_2 \cdot \cdots \cdot U_{n-1}}_{U^T} = H$$

$$\Downarrow$$

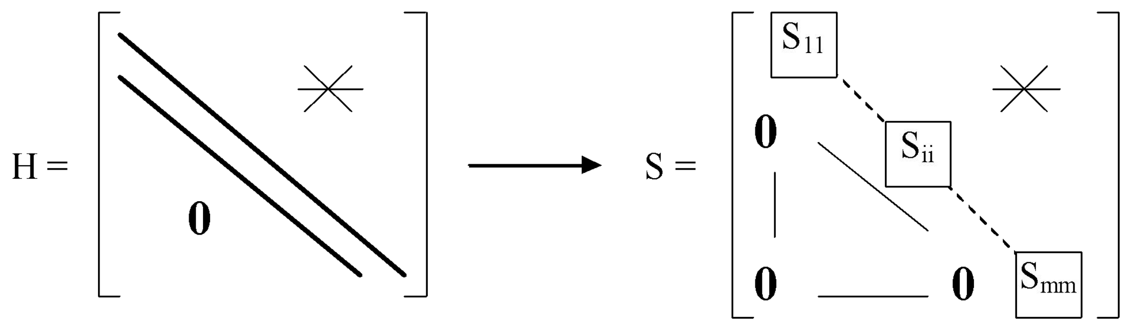
$$H = U \cdot A \cdot U^T$$

· sunt parcurse exact (n – 2) iteraţii

☐ ***Faza 2***

- procedură iterativă de construcţie a unui şir QR pornind de la matricea H

- scopul: anularea unora din elementele de pe sub-diagonala principală a matricei H



-algoritmul QR cu deplasare explicită parcurge următoarele etape:

pas QR cu deplasare explicită

1. determinare deplasare , μ;
2. atribuire $H \leftarrow H - \mu \cdot I_n$;
3. descompunere $\boxed{H = Q \cdot R}$, R - matrice superior triunghiulară, Q - matrice ortogonală;
4. atribuire $H \leftarrow R \cdot Q$;
5. refacere deplasare $H \leftarrow H + \mu \cdot I_n$;
6. corecţie matrice H (efectuare test de decuplare);
7. atribuire $S \leftarrow H$ şi efectuare teste reluare algoritm QR (de la etapa 1).

(n-1) iteraţii

👉 **Detaliere etape:**

❶ *Etapa 1* - determinare deplasare μ

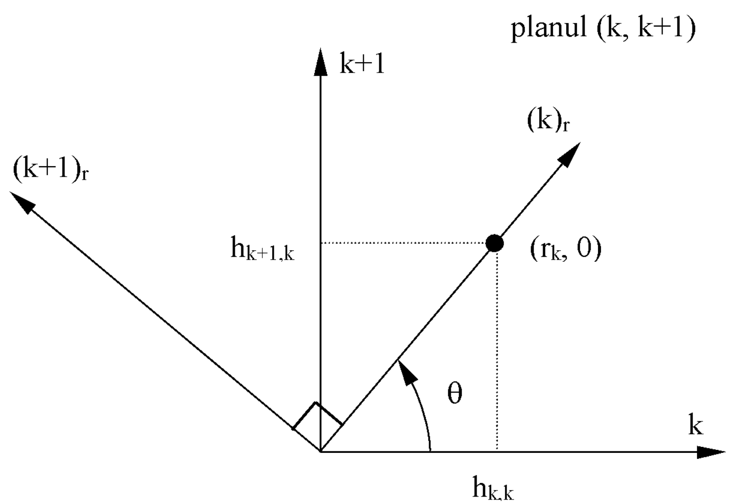
- viteza maximă de convergenţă pentru $\mu = h_{n,n}$

❷ *Etapa 2* - deplasarea μ se scade de pe diagonala principală a matricei H

- se lucrează în continuare cu această matrice

❸ *Etapa 3* – factorizare QR a matricei H obţinută la etapa 2

- se zerorizează elementele de pe sub-diagonala principală (n – 1 elemente)
- se utilizează *matrici de rotaţie plană Givens* (ortogonale, nesimetrice)



$$R \cdot \begin{bmatrix} h_{k,k} \\ h_{k+1,k} \end{bmatrix} = \begin{bmatrix} r_k \\ 0 \end{bmatrix}$$

$$R = \begin{bmatrix} c_k & d_k \\ -d_k & c_k \end{bmatrix}$$

$$c_k = \cos(\theta) = \frac{h_{k,k}}{r_k}$$

$$d_k = \sin(\theta) = \frac{h_{k+1,k}}{r_k}$$

- matricea de rotaţie Givens:

$$P_k = \left[\begin{array}{c|c|c} I_{k-1} & \mathbf{0}_{(k-1)\times 2} & \mathbf{0}_{(k-1)\times(n-k-1)} \\ \hline \mathbf{0}_{2\times(k-1)} & R & \mathbf{0}_{2\times(n-k-1)} \\ \hline \mathbf{0}_{(n-k-1)\times(k-1)} & \mathbf{0}_{(n-k-1)\times 2} & I_{n-k-1} \end{array}\right]_{n\times n}$$

- se poate demonstra:

$$P_k \cdot P_k^T = \left[\begin{array}{c|c|c} I_{k-1} & \mathbf{0}_{(k-1)\times 2} & \mathbf{0}_{(k-1)\times(n-k-1)} \\ \hline \mathbf{0}_{2\times(k-1)} & Q & \mathbf{0}_{2\times(n-k-1)} \\ \hline \mathbf{0}_{(n-k-1)\times(k-1)} & \mathbf{0}_{(n-k-1)\times 2} & I_{n-k-1} \end{array}\right]_{n\times n}$$

$$Q = \begin{bmatrix} c_k^2 + d_k^2 & 0 \\ 0 & c_k^2 + d_k^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\Rightarrow \quad P_k \cdot P_k^T = I_n$$

- tabloul transformărilor de la etapa 3:

$$P_{n-1} \cdot \cdots \cdot P_1 \cdot H = R$$

$$Q^T = P_{n-1} \cdot \cdots \cdot P_1 \;\Rightarrow\; Q = P_1^T \cdot \cdots \cdot P_{n-1}^T$$

R – matrice superior triunghiulară

❹ *Etapa 4* – matricea R se înmulţeşte la dreapta cu matricea Q, rezultatul fiind stocat în H:

$$H = R \cdot P_1^T \cdot \cdots \cdot P_{n-1}^T$$

❺ *Etapa 5* – se adună deplasarea la elementele de pe diagonala prinicipală a matricii H

❻ *Etapa 6* – test de decuplare

$$P_k \begin{vmatrix} \ddots & & & \\ & h_{k,k} & h_{k,k+1} & \\ & h_{k+1,k} & h_{k+1,k+1} & \\ & & & \ddots \end{vmatrix} \Rightarrow \begin{vmatrix} \ddots & & & \\ & h'_{k,k} & h'_{k,k+1} & \\ & 0 & h'_{k+1,k+1} & \\ & & & \ddots \end{vmatrix} P_k^T \Rightarrow \begin{vmatrix} \ddots & & & \\ & h''_{k,k} & h''_{k,k+1} & \\ & h''_{k+1,k} & h''_{k+1,k+1} & \\ & & & \ddots \end{vmatrix}$$

$k = 1, \dots, n-1$

$$| h_{k+1,k} | \geq | h''_{k+1,k} |$$

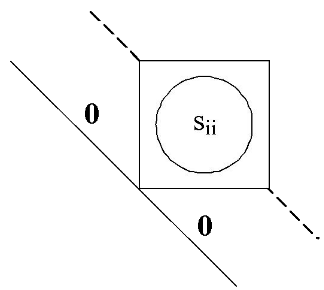
- inegalitatea strictă $\rightarrow$ elementul din poziţia (k+1, k) devine zero în forma canonică Schur

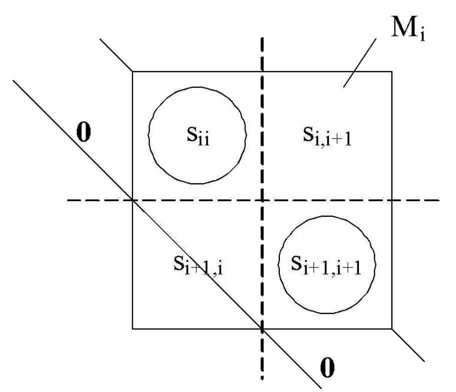
- rol de decuplare a blocurilor diagonale din forma canonică Schur ⟶ test de decuplare:

<u>dacă</u> $|h''_{k+1,k}| < \varepsilon \cdot (|h''_{k,k}| + |h''_{k+1,k+1}|)$ <u>atunci</u>

⌈ <u>atribuie</u> $h''_{k+1,k} \leftarrow 0$
⌊ •

❼ *<u>Etapa 7</u>* – *<u>testele care condiţionează reluarea sau nu a algoritmului QR</u>*





👉 **concluzii** privind structura formei canonice Schur:

- nu există pe sub-diagonala principală două elemente consecutive nenule
- blocuri de ordinul doi pe diagonala matricei S au valori proprii complex conjugate

👉 testele care se realizează la această etapă sunt:

(T1) <u>dacă</u> $\exists i \in \{1, 2, \ldots, n-2\}$ pentru care $s_{i+1,i} \neq 0$ și $s_{i+2,i+1} \neq 0$ <u>atunci</u>

- matricea S nu este în forma canonică Schur (reluare de la etapa 1)

(T2) <u>dacă</u> $\nexists i$ care să satisfacă (T1), dar $\exists i \in \{1, 2, \ldots, n-1\}$ pentru care

- blocul $M_i = \begin{bmatrix} s_{i,i} & s_{i,i+1} \\ s_{i+1,i} & s_{i+1,i+1} \end{bmatrix}$ are valori proprii reale <u>atunci</u>
- matricea S nu este în forma canonică Schur (reluare de la etapa 1)

☐ Tabloul general al transformărilor din faza a doua a algoritmului QR:

$$S = \underbrace{[P_{n-1}^{[s]} \cdot \cdots \cdot P_1^{[s]}] \cdot \cdots \cdot [P_{n-1}^{[1]} \cdot \cdots \cdot P_1^{[1]}]}_{P} \cdot H \cdot \underbrace{[(P_1^{[1]})^T \cdot \cdots \cdot (P_{n-1}^{[1]})^T] \cdot \cdots \cdot [(P_1^{[s]})^T \cdot \cdots \cdot (P_{n-1}^{[s]})^T]}_{P^T}$$

$$\Downarrow$$

$$P \cdot H \cdot P^T = S$$

👉 În urma aplicării ambelor faze ale formei implementabile a algoritmului QR rezultă:

$$P \cdot U \cdot A \cdot U^T \cdot P^T = S$$

$$\underbrace{\tilde{Q}^T = P \cdot U \longrightarrow \tilde{Q} = U^T \cdot P^T}$$

$$\Downarrow$$

$$S = \tilde{Q}^T \cdot A \cdot \tilde{Q}$$

👉 *forma implementabilă a algoritmului QR este o procedură stabilă numeric, iar forma canonică Schur calculată pentru o matrice* A*, notată prin , coincide cu forma canonică Schur exactă,* S*, a matricei* A *uşor perturbată,* $\hat{A} = A + E$ *:*

$$\hat{S} \equiv S_{A+E}, \quad \| E \|_\alpha << \| A \|_\alpha$$

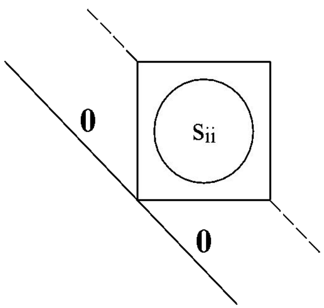
exemplu

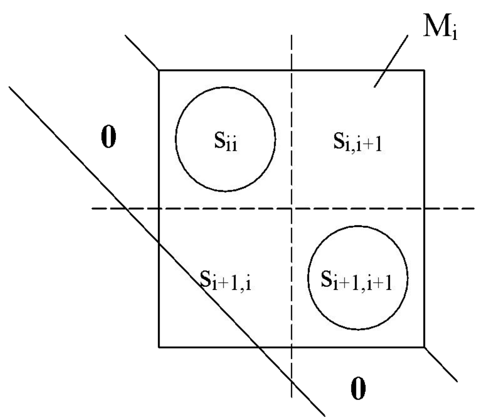
4.4 Calculul valorilor şi vectorilor proprii

👉 calculul **valorilor proprii** → inspecţia blocurilor diagonale ale formei canonice Schur, S

⬇

inspecţia formei canonice Schur → elementele aflate pe prima sub-diagonală a matricei S





<u>atribuie</u> i ← 1
<u>cât timp</u> (i ≤ n – 1)
  <u>dacă</u> $s_{i+1,i} = 0$ <u>atunci</u>
    <u>atribuie</u> $\lambda_i \leftarrow s_{i,i}$
    <u>atribuie</u> i ← i + 1
  □
  <u>dacă</u> $s_{i+1,i} \neq 0$ <u>atunci</u>
    <u>atribuie</u> $\lambda_{i,i+1}$ ← valorile proprii ale blocului:

$$M_i = \begin{bmatrix} s_{i,i} & s_{i,i+1} \\ s_{i+1,i} & s_{i+1,i+1} \end{bmatrix}$$

    <u>atribuie</u> i ← i + 2
  □
□

$$\det(\lambda \cdot I_2 - M_i) = 0$$

⬇

$$\lambda^2 - (s_{i,i} + s_{i+1,i+1}) \cdot \lambda + (s_{i,i} \cdot s_{i+1,i+1} - s_{i+1,i} \cdot s_{i,i+1}) = 0.$$

👉 calculul **<u>vectorilor proprii</u>**

$$\underline{x}_i = \tilde{Q} \cdot \underline{r}_i, \quad i = 1, \ldots, n \qquad \text{unde} \qquad S \cdot \underline{r}_i = \lambda_i \cdot \underline{r}_i, \quad i = 1, \ldots, n$$

👉 în practică ⟶ *<u>vectorii Schur</u>* ai matricei A ⟶ vectorii coloană ai matricei $\tilde{Q}$

$$\tilde{Q} = [\tilde{\underline{q}}_1 \quad \cdots \quad \tilde{\underline{q}}_i \quad \cdots \quad \tilde{\underline{q}}_n] \quad \begin{cases} \tilde{\underline{q}}_i,\ i = 1, \ldots, n & \text{- ortogonali} \\ \| \tilde{\underline{q}}_i \|_2^2 = 1,\ i = 1, \ldots, n \end{cases}$$

· dacă $\lambda_i = s_{ii} \in \Re \ (s_{i+1,i} = 0)$ atunci $\underline{x}_i \to \tilde{\underline{q}}_i$

· dacă $\lambda_i, \lambda_{i+1} \in \mathbf{C} \ (s_{i+1,i} \neq 0)$ atunci $\underline{x}_i \to \tilde{\underline{q}}_i + j \cdot \tilde{\underline{q}}_{i+1}, \quad \underline{x}_{i+1} \to \tilde{\underline{q}}_i - j \cdot \tilde{\underline{q}}_{i+1}$