

Cap. 4 Calculul valorilor și vectorilor proprii

4.1 Formularea problemei

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

Definiție:

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

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

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

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

$\lambda \cdot I_n - A$ - singulară

ecuație caracteristică

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

polinom caracteristic

$$p_n(\lambda) = \lambda^n + \alpha_1 \cdot \lambda^{n-1} + \dots + \alpha_{n-1} \cdot \lambda + \alpha_n \quad \alpha_i \in \mathfrak{R}, i = 1, \dots, n$$

Teoremă de existență:

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



Orice matrice $A \in \mathbb{R}^{n \times n}$ are cel puțin un vector propriu.

- Nu se recomandă calculul numeric al valorilor proprii prin rezolvarea ecuației caracteristice deoarece:
 - rezolvarea ecuațiilor polinomiale este o problemă prost condiționată;
 - coeficienții polinomului caracteristic □ volum mare de calcule □ erori
- Metodele practice pentru calculul numeric al valorilor proprii □ proceduri iterative
 - matricea A adusă la formă canonică Schur prin transformări ortogonale de asemănare

4.2 Forma canonică Schur

Definiție:

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

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

□ Proprietăți:

❶ Matricile ortogonal asemenea au aceleași valori proprii:

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

❷ Relația dintre vectorii proprii ai două matrici ortogonal asemenea:

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

Definiție:

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

$$T = \begin{bmatrix} T_{11} & T_{12} & \boxtimes & T_{1m} \\ \mathbf{0} & T_{22} & \boxtimes & T_{2m} \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \mathbf{0} & T_{mm} \end{bmatrix}, \quad T_{ii} \in \mathbb{R}^{p_i \times p_i} \quad T_{ii}, i = 1, \dots, m \text{ - matrici pătrate}$$

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

Teoremă de existență:

Oricare ar fi matricea $A \in \mathbb{R}^{n \times n}$, există o matrice ortogonală $\tilde{Q} \in \mathbb{R}^{n \times n}$, astfel încât matricea:

$$\text{forma canonică Schur reală a matricei } A \longrightarrow S = \tilde{Q}^T \cdot A \cdot \tilde{Q}$$

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

Observații:

- ❶ coloanele matricii $\tilde{Q} \in \mathbb{R}^{n \times n}$ $\underline{\tilde{q}}_k, k = 1, \dots, n$ $\square$ vectori Schur ai matricii A
- ❷ Demonstrația teoremei este constructivă $\square$ **algoritmul QR**

4.3 Algoritmul QR pentru calculul formei canonice Schur

$\square$ **Principiu:** construcția unui șir de matrici ortogonal asemenea convergent la forma canonică Schur

$$\begin{array}{ccc}
 A_0 = A, A_1, \dots, A_k, \dots & & A_k \xrightarrow{k \rightarrow \infty} S \\
 \downarrow & & \\
 A_k = [a_{ij}^{[k]}]_{1 \leq i, j \leq n} & & a_{ij}^{[k]} \xrightarrow{k \rightarrow \infty} 0 \quad i \geq j + 2 \\
 & & \exists a_{i+1,i}^{[k]} \xrightarrow{k \rightarrow \infty} 0 \quad i = 1, \dots, n-1
 \end{array}$$

□ se definește șirul de matrici $\{A_k\}_{k \geq 0}$, $A_0 = A$

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

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

$\mu_k \in \mathbb{R}$ - deplasare (cu rol de accelerare a convergenței)

Propoziție:

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

□ **Observație:**

- algoritmul original QR □ consumator de timp □ se folosește o formă optimizată

Forma implementabilă a algoritmului QR cu deplasare explicită

□ parcurge două faze de lucru:

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

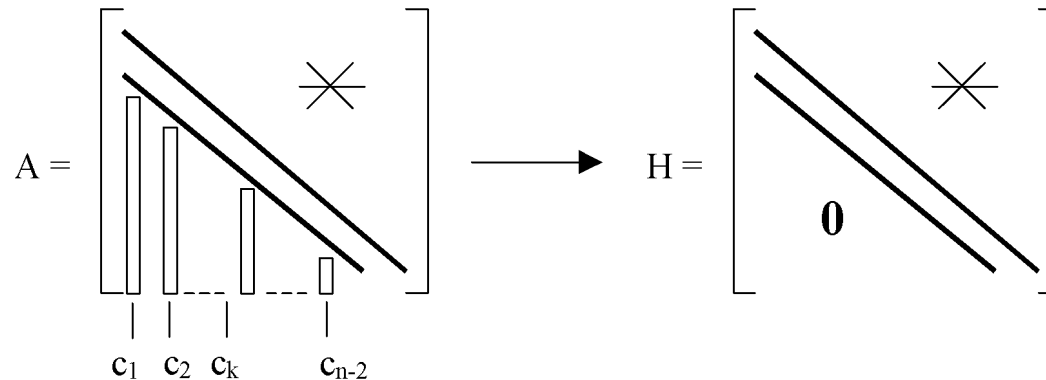
❷ faza 2 – aplicare algoritm QR matricii obținută la faza 1



se obține 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
- algorithm:

```

atribuie  $A_1 \leftarrow A$ 
pentru  $k = 1, n - 2$  execută
[
    * determinare matrice Householder astfel încât:
    |
    |  $(U_{k+1} \cdot A_k)_{i,k} = 0, i = k + 2, \dots, n$ 
    |
    | 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}

$$U_{k+1} = I_n - (\underline{u}_{k+1} \cdot \underline{u}_{k+1}^T / \beta_{k+1})$$

$$\underline{u}_{k+1} = [0 \quad \boxtimes \quad 0 \quad u_{k+1,k+1} \quad \boxtimes \quad u_{n,k+1}]^T$$

$$\sigma_{k+1} = \text{sign}(a_{k+1,k}^{[k]}) \cdot \sqrt{\sum_{i=k+1}^n (a_{i,k}^{[k]})^2}, \quad u_{k+1,k+1} = a_{k+1,k}^{[k]} + \sigma_{k+1}$$

$$u_{i,k} = a_{i,k}^{[k]}, \quad i = k+2, \dots, n, \quad \beta_{k+1} = \sigma_{k+1} \cdot u_{k+1,k+1}$$

- tabloul general al transformărilor:

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

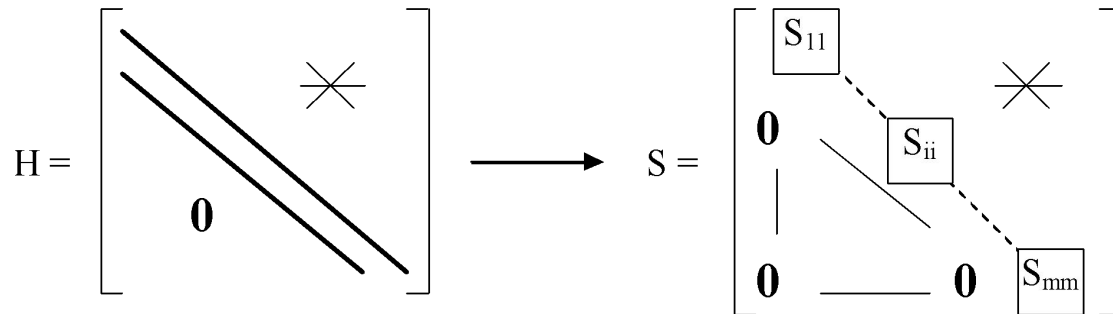


$$H = U \cdot A \cdot U^T$$

- sunt parcurse exact $(n - 2)$ iterații

□ Faza 2

- procedură iterativă de construcție a unui șir QR pornind de la matricea H
- scopul: anularea unora din elementele de pe sub-diagonala principală a matricei H



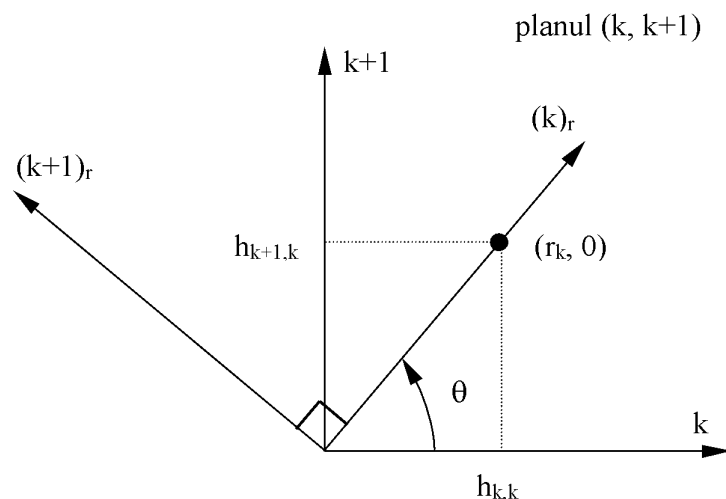
-algoritmul QR cu deplasare explicită parcurge următoarele etape:

- pas QR cu deplasare explicită*
1. determinare deplasare , μ ;
 2. atribuire $H \leftarrow H - \mu \cdot I_n$;
 3. descompunere $H = Q \cdot R$, 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. corecție matrice H (efectuare test de decuplare);
 7. atribuire $S \leftarrow H$ și efectuare teste reluare algoritm QR (de la etapa 1).
- (n-1) iterații

👉 Detaliere

etape:

- ❶ Etapa 1 - determinare deplasare μ
 - viteza maximă de convergență 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 obținută la etapa 2
 - se zerorizează elementele de pe sub-diagonala principală (n – 1 elemente)
 - se utilizează matrici de rotație plană Givens (ortogonale, nesimetrice)



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

$$R = \begin{bmatrix} c_k & d_k \\ -d_k & c_k \end{bmatrix} \begin{cases} c_k = \cos(\theta) = \frac{h_{k,k}}{r_k} \\ d_k = \sin(\theta) = \frac{h_{k+1,k}}{r_k} \end{cases}$$

- matricea de rotație Givens:

$$P_k = \begin{bmatrix} I_{k-1} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \mathbf{0}_{(k-1) \times 2} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \mathbf{0}_{(k-1) \times (n-k-1)} \\ \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} \\ \mathbf{0}_{2 \times (k-1)} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & R & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \mathbf{0}_{2 \times (n-k-1)} \\ \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} \\ \mathbf{0}_{(n-k-1) \times (k-1)} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \mathbf{0}_{(n-k-1) \times 2} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & I_{n-k-1} \end{bmatrix}_{n \times n}$$

- se poate demonstra:

$$P_k \cdot P_k^T = \begin{bmatrix} I_{k-1} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \mathbf{0}_{(k-1) \times 2} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \mathbf{0}_{(k-1) \times (n-k-1)} \\ \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} \\ \mathbf{0}_{2 \times (k-1)} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & Q & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \mathbf{0}_{2 \times (n-k-1)} \\ \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} \\ \mathbf{0}_{(n-k-1) \times (k-1)} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & \mathbf{0}_{(n-k-1) \times 2} & \begin{smallmatrix} \times \\ \times \end{smallmatrix} & I_{n-k-1} \end{bmatrix}_{n \times n} \quad \Rightarrow \quad P_k \cdot P_k^T = I_n$$

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

- tabloul transformărilor de la etapa 3:

$$P_{n-1} \cdot \begin{bmatrix} \times & & \\ & \times & \\ & & \times \end{bmatrix} \cdot P_1 \cdot H = R$$

$$Q^T = P_{n-1} \cdot \begin{bmatrix} \times & & \\ & \times & \\ & & \times \end{bmatrix} \cdot P_1 \Rightarrow Q = P_1^T \cdot \begin{bmatrix} \times & & \\ & \times & \\ & & \times \end{bmatrix} \cdot P_{n-1}^T$$

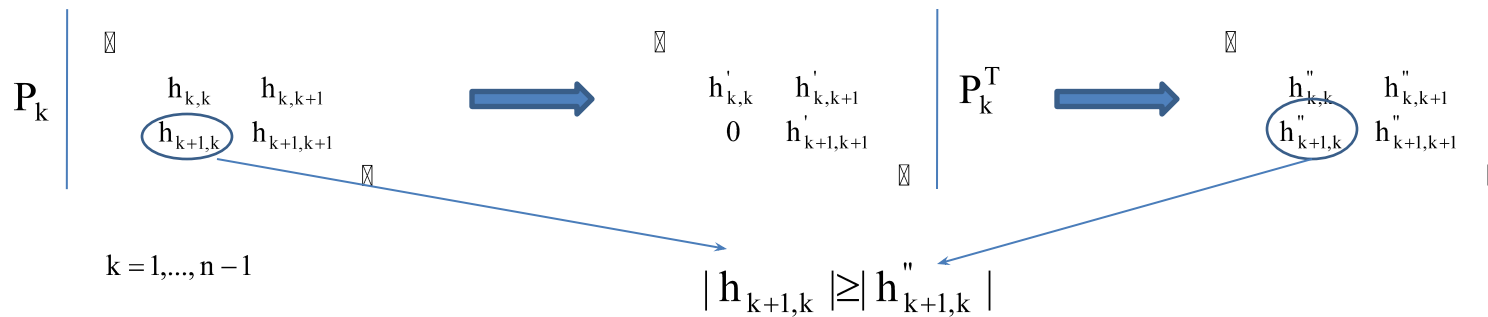
R – matrice superior triunghiulară

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

$$H = R \cdot P_1^T \cdot \begin{bmatrix} \times & & \\ & \times & \\ & & \times \end{bmatrix} \cdot P_{n-1}^T$$

⑤ Etapa 5 – se adună deplasarea la elementele de pe diagonala principală a matricii H

⑥ Etapa 6 – test de decuplare



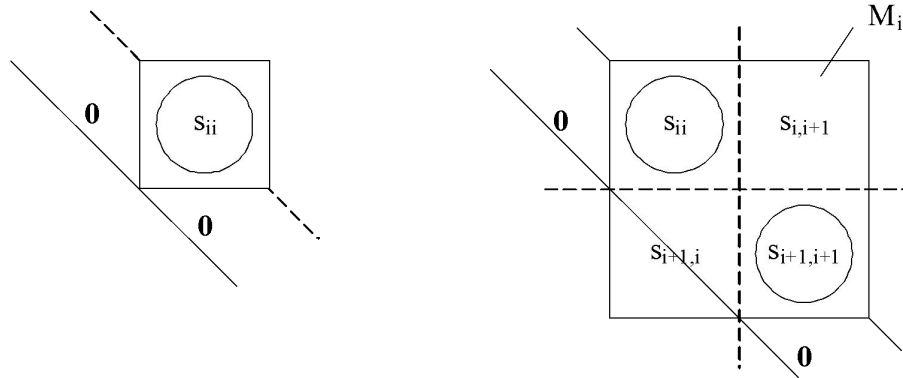
- inegalitatea strictă → elementul din poziția (k+1, k) devine zero în forma canonică Schur

- rol de decuplare a blocurilor diagonale din forma canonică Schur $\rightarrow$ test de decuplare:

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

[atribuie $h''_{k+1,k} \leftarrow 0$
•

7 Etapa 7 – testele care condiționează reluarea sau nu a algoritmului QR



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) dacă $\exists i \in \{1, 2, \dots, n-1\}$ pentru care $s_{i+1,i} \neq 0$ atunci
 matricea S nu este în forma canonică Schur (reluare de la etapa 1)

(T2) dacă ~~nu~~ să satisfacă (T1), dar $\exists i \in \{1, 2, \dots, n-1\}$ 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 atunci
 matricea S nu este în forma canonică Schur (reluare de la etapa 1)

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

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



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

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

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

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


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

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

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

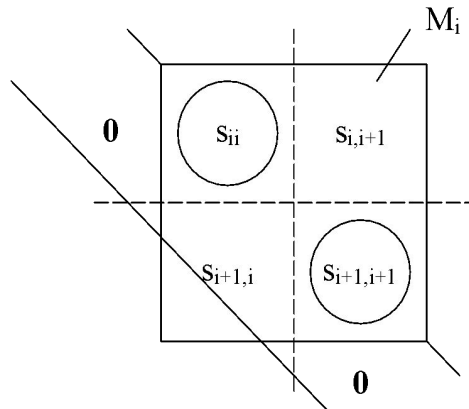
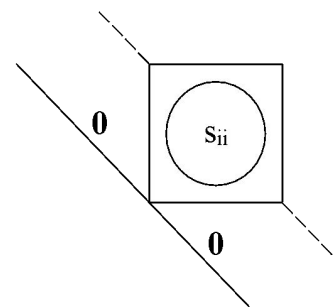
exemplu

4.4 Calculul valorilor și vectorilor proprii

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



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



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



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

atribuie $i \leftarrow 1$

cât timp $(i \leq n - 1)$

dacă $s_{i+1,i} = 0$ atunci

atribuie $\lambda_i \leftarrow s_{i,i}$

atribuie $i \leftarrow i + 1$



dacă $s_{i+1,i} \neq 0$ atunci

atribuie $\lambda_{i,i+1} \leftarrow$ valorile proprii ale blocului:

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

atribuie $i \leftarrow i + 2$



👉 calculul vectorilor proprii

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

👉 în practică $\rightarrow$ vectorii Schur ai matricei A $\rightarrow$ vectorii coloană ai matricei $\tilde{Q}$

$$\tilde{Q} = [\underline{\tilde{q}}_1 \quad \underline{\tilde{q}}_2 \quad \dots \quad \underline{\tilde{q}}_n] \quad \left\{ \begin{array}{l} \underline{\tilde{q}}_i, i=1, \dots, n \quad - \text{ortogonali} \\ \|\underline{\tilde{q}}_i\|_2^2 = 1, \quad i=1, \dots, n \end{array} \right.$$

• dacă $\lambda_i = s_{ii} \in \mathbb{R}$ ($s_{i+1,i} = 0$) atunci $\underline{x}_i \rightarrow \underline{\tilde{q}}_i$

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