

# Laborator 5

## Calculul valorilor proprii

### 5.1 Valori și vectori proprii

**Definiția 5.1** Fie  $A \in \mathbb{C}^{n \times n}$ . Un vector  $x \in \mathbb{C}^n$  este un *vector propriu* al matricei  $A$ , asociat *valorii proprii*  $\lambda \in \mathbb{C}$ , dacă sunt satisfăcute următoarele două proprietăți

$$\begin{aligned} a) \quad & x \neq 0 \\ b) \quad & Ax = \lambda x. \end{aligned} \tag{5.1}$$

Câteva dintre cele mai importante proprietăți ale valorilor proprii ale unei matrice (pătrate) sunt prezentate mai jos.

**Teorema 5.2** Fie  $A \in \mathbb{C}^{n \times n}$ . Numărul  $\lambda \in \mathbb{C}$  este o *valoare proprie* a lui  $A$  dacă și numai dacă matricea  $\lambda I - A$  este singulară. Mai mult, matricea  $A$  are exact  $n$  valori proprii, incluzând multiplicitățile, care sunt zerourile polinomului caracteristic

$$p(\lambda) = \det(\lambda I_n - A). \tag{5.2}$$

Dacă  $A \in \mathbb{R}^{n \times n}$ , atunci valorile proprii complexe apar în perechi conjugate.

Notând cu

$$\lambda(A) = \{\lambda_1, \lambda_2, \dots, \lambda_n\} = \{\lambda \in \mathbb{C} \mid \det(\lambda I - A) = 0\} \tag{5.3}$$

mulțimea tuturor valorilor proprii ale matricei  $A \in \mathbb{C}^{n \times n}$ . Mulțimea  $\lambda(A)$  se numește *spectrul (de valori proprii al) lui A*. Este util să definim transformările matriceale care conservă spectrul unei matrice date.

**Definiția 5.3** Două matrice  $A, B \in \mathbb{R}^{n \times n}$  se numesc *asemenea* dacă există o matrice nesingulară  $T \in \mathbb{R}^{n \times n}$  astfel încât

$$B = TAT^{-1}. \tag{5.4}$$

Dacă  $T$  este o matrice ortogonală, atunci  $A$  și  $B$  se numesc *ortogonal asemenea*.

**Teorema 5.4** Dacă  $A, B \in \mathbb{R}^{n \times n}$  sunt asemenea atunci ele au același spectru

$$\lambda(A) = \lambda(B). \quad (5.5)$$

Mai mult, dacă  $T$  este o transformare de asemănare ca în (5.4) și dacă  $x_A$  este un vector propriu al lui  $A$  asociat valorii proprii  $\lambda \in \lambda(A)$ , atunci vectorul

$$x_B = Tx_A \quad (5.6)$$

este un vector propriu al matricei  $B$ , asociat aceleiași valori proprii.

## 5.2 Forma Schur (reală)

Forma Schur (reală) joacă un rol central în calculul eficient al valorilor proprii ale unei matrice.

**Definiția 5.5** O matrice  $S \in \mathbb{R}^{n \times n}$  se numește *formă Schur reală (FSR)* dacă are structura

$$S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1q} \\ 0 & S_{22} & \cdots & S_{2q} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & S_{qq} \end{bmatrix}, \quad (5.7)$$

unde  $S_{ii} \in \mathbb{R}^{1 \times 1}$  sau  $S_{ii} \in \mathbb{R}^{2 \times 2}$  și toate blocurile diagonale  $2 \times 2$  au valori proprii complexe (mai precis, complex conjugate).

O consecință imediată a structurii Schur este dată în propoziția următoare.

**Propoziția 5.6** Fie  $S \in \mathbb{R}^{n \times n}$  în FSR. Atunci:

$$\lambda(S) = \cup_{i=1}^q \lambda(S_{ii}). \quad (5.8)$$

Un rezultat important în calculul valorilor proprii este dat de următoarea teoremă.

**Teorema 5.7** Pentru orice matrice pătrată  $A \in \mathbb{R}^{n \times n}$  există o matrice ortogonală  $Q \in \mathbb{R}^{n \times n}$  astfel încât

$$Q^T A Q = S \quad (5.9)$$

este FSR.

Calculul (exact) al formei Schur (reale) pentru matrice de ordin superior lui 4 nu poate fi realizat într-un număr finit de operații elementare pentru că altfel s-ar contrazice un rezultat fundamental din algebră conform căruia rezolvarea unei ecuații algebrice de grad superior lui 4 nu este posibilă într-un număr finit de operații elementare.

Pe de altă parte, dacă se dispune de un vector propriu  $x$ , calculul valorii proprii asociate  $\lambda$  este imediat folosind formula (numită *cât Rayleigh*)

$$\lambda = \frac{x^T A x}{x^T x}.$$

Mai mult, printr-o procedură de deflație, după determinarea unui vector propriu și a valorii proprii asociate, problema își reduce dimensiunea. De aceea, există metode de calcul iterativ al unui vector propriu în absența cunoașterii valorii proprii asociate, metode care stau la baza algoritmului QR, de calcul iterativ al formei Schur (reale), dar care au și aplicații directe. Prezentăm direct algoritmi corespunzători celor mai folosite astfel de metode: *metoda puterii* și *metoda puterii inverse*.

### 5.2.1 Metoda puterii

Presupunem că matricea  $A \in \mathbb{R}^{n \times n}$  are o valoare proprie dominantă, i.e. mai mare în modul decât toate celelalte, și fie aceasta  $\lambda_1$ . Metoda puterii se bazează faptul că șirul vectorial

$$y^{(k)} = A^k y_0, \quad k = 1, 2, \dots, \quad (5.10)$$

unde  $y_0$  este un vector aleator ales, tinde, ca direcție, către direcția vectorului propriu asociat valorii proprii dominante. Introducând și normarea vectorilor din acest șir, acesta poate fi definit recurent prin

$$y^{(k)} = \rho_k A y^{(k-1)}, \quad k = 1, 2, \dots \quad (5.11)$$

unde  $\rho_k = \frac{1}{\|A y^{(k-1)}\|}$  este un factor de scalare.

ALGORITM 5.8 (*Metoda puterii*) (Dată o matrice  $A \in \mathbb{C}^{n \times n}$ , un nivel de toleranță  $tol \in \mathbb{R}$ ,  $tol < 1$ , și un număr maxim admis *maxiter* de iterații, algoritmul calculează un vector propriu unitar  $y$  asociat valorii proprii dominante a matricei date sau tipărește un mesaj dacă obiectivul nu a fost atins în numărul de iterații admis.)

1. Se alege aleator un vector  $y \in \mathbb{R}^n$
2.  $y \leftarrow y / \|y\|$
3.  $i = 0, \quad e = 1$
4. **cât timp**  $e > tol$ 
  1. **dacă**  $i > maxiter$  **atunci**
    1. **tipărește** 'S-a atins numărul maxim de iterații fără a se fi obținut nivelul prescris al toleranței.'
    2. **stop**
  2.  $z = Ay$
  3.  $z \leftarrow z / \|z\|$
  4.  $e = |1 - |z^T y||$
  5.  $y \leftarrow z$
  6.  $i \leftarrow i + 1$

Dacă matricea nu are o valoare proprie net dominantă, viteza de convergență poate fi nesatisfăcătoare, iar în cazul absenței unei valori proprii dominante șirul poate fi divergent.

### 5.2.2 Metoda puterii inverse

Metoda puterii inverse pentru matricea  $A$  este, în fapt, metoda puterii pentru matricea  $(\mu I - A)^{-1}$ , unde  $\mu$  este un scalar adecvat ales la fiecare pas, numit deplasare.

Prin urmare, metoda puterii inverse construiește un șir de vectori cu relația de recurență

$$y^{(k)} = \rho_k(\lambda I - A)^{-1}y^{(k-1)}, \quad k = 1, 2, \dots \quad (5.12)$$

Desigur, mijlocul eficient de implementare a iterației (5.12) evită calculul matricei  $(\lambda I - A)^{-1}$  și apelează la rezolvarea unui sistem liniar, conform următoarei scheme.

1. Se rezolvă în raport cu  $y^{(k)}$  sistemul liniar  $(\lambda I - A)y^{(k)} = y^{(k-1)}$ .
2.  $y^{(k)} \leftarrow y^{(k)} / \|y^{(k)}\|$ .

Metoda puterii inverse este una dintre cele mai bune metode iterative de calcul al unui vector propriu a unei matrice.

Cea mai bună alegere a deplasării este dată de câtul Rayleigh

$$\lambda = \frac{(y^{(k-1)})^T A y^{(k-1)}}{\|y^{(k-1)}\|^2} = (y^{(k-1)})^T A y^{(k-1)}. \quad (5.13)$$

Utilizând același criteriu de oprire ca și la metoda puterii obținem următorul algoritm.

#### ALGORITM 5.9 (Metoda puterii inverse cu deplasare Rayleigh)

(Dată o matrice  $A \in \mathbb{R}^{n \times n}$ , un nivel de toleranță  $tol \in \mathbb{R}$ ,  $tol < 1$ , și un număr maxim admis *maxiter* de iterații, algoritmul calculează un vector propriu unitar  $y$  al matricei date sau tipărește un mesaj dacă obiectivul nu a fost atins în numărul admis de iterații.)

1. Se alege aleator un vector  $y \in \mathbb{R}^n$ .
2.  $y \leftarrow y / \|y\|$
3.  $i = 0, \quad e = 1$
4. **cât timp**  $e > tol$ 
  1. **dacă**  $i > maxiter$  **atunci**
    1. **tipărește** 'S-a atins numărul maxim de iterații fără a se fi obținut nivelul prescris al toleranței.'
    2. **stop**
  2.  $\mu = y^T A y$
  3. Se rezolvă sistemul liniar  $(\mu I_n - A)z = y$

4.  $z \leftarrow z/\|z\|$
5.  $e = |1 - |z^T y||$
6.  $y \leftarrow z$
7.  $i \leftarrow i + 1$

### 5.3 Algoritmul QR

Algoritmul **QR** este, în esență, o procedură de deflație iterativă care construiește (recurent) un șir de matrice ortogonal asemenea cu matricea inițială, șir care, în anumite condiții, este convergent către forma Schur (reală).

În vederea minimizării efortului de calcul, într-o fază preliminară, matricea dată este adusă, prin transformări de asemenea ce implică un număr (teoretic) *finit* și (practic) rezonabil de mic de operații, la cea mai apropiată structură posibilă de forma Schur (reală). Această structură este forma superior Hessenberg <sup>1</sup>. În continuare, structura Hessenberg este conservată de recurența fazei iterative a algoritmului. În acest fel, se obține o importantă reducere a complexității unei iterații **QR**, fapt esențial în economia algoritmului.

#### 5.3.1 Reducerea la forma superior Hessenberg

Posibilitatea reducerii unei matrice  $A \in \mathbb{R}^{n \times n}$  la forma superior Hessenberg, cu conservarea valorilor proprii, este dată de următorul rezultat.

**Teorema 5.10** *Oricare ar fi matricea  $A \in \mathbb{R}^{n \times n}$ , există o matrice ortogonală  $Q \in \mathbb{R}^{n \times n}$ , calculabilă printr-o secvență finită de operații aritmetice, astfel încât matricea*

$$H = Q^T A Q \quad (5.14)$$

*este superior Hessenberg.*

Concret, reducerea la forma superior Hessenberg este bazată pe următoarea schemă de calcul

- HQ 1. pentru  $k = 1 : n - 2$
1. Se calculează un reflector elementar  $U_{k+1}$  astfel încât  $(U_{k+1}A)(k+2 : n, k) = 0$ .
  2.  $A \leftarrow U_{k+1}A$
  3.  $A \leftarrow AU_{k+1}$ .

care suprascrie matricea  $A$  cu matricea Hessenberg rezultată

$$A \leftarrow H = U_{n-1} \cdots U_3 U_2 A U_2 U_3 \cdots U_{n-1} = Q^T A Q, \quad (5.15)$$

<sup>1</sup>Matricea  $H \in \mathbb{R}^{n \times n}$  este în formă superior Hessenberg dacă  $h_{ij} = 0, \forall i > j + 1$ .

unde

$$Q = U_2 U_3 \cdots U_{n-1}. \quad (5.16)$$

Avându-se în vedere că reflectorul  $U_{k+1}$  are structura

$$U_{k+1} = \begin{bmatrix} I_k & 0 \\ 0 & \bar{U}_1 \end{bmatrix},$$

unde  $\bar{U}_1$  este un reflector de ordin  $n - k$  și indice 1, rezultă imediat următoarele echivalențe calculatorii:

$$\begin{aligned} A \leftarrow U_{k+1} A &\equiv A(k+1 : n, :) \leftarrow \bar{U}_1 A(k+1 : n, :), \\ A \leftarrow A U_{k+1} &\equiv A(:, k+1 : n) \leftarrow A(:, k+1 : n) \bar{U}_1. \end{aligned}$$

Prin urmare, pentru obținerea formei superior Hessenberg este suficient să știm să ”operăm” numai cu reflectori *de indice 1*. În acest scop introducem următoarele proceduri:

**Z** (Procedură de calcul a elementelor definatorii  $u$  și  $\beta$  ale unui reflector  $U_1$  astfel încât  $(U_1 x)(2 : n) = 0$  și de calcul  $x \leftarrow U_1 x$ , unde  $x \in \mathbb{R}^n$  este un vector dat.)

1.  $\sigma = \text{sign}(x_1) \sqrt{\sum_{i=1}^n x_i^2}$
2.  $u_1 = x_1 + \sigma$
3. pentru  $i = 2 : n$ 
  1.  $u_i = x_i$
  2.  $(x_i = 0)$
4.  $\beta = u_1 \sigma$
5.  $x_1 \leftarrow -\sigma$

Sintaxa de apel a procedurii **Z** este

$$[u, \beta, x] = \mathbf{Z}(x).$$

**As** (Procedură de înmulțire la stânga a unei matrice  $A \in \mathbb{R}^{n \times p}$  cu un reflector  $U_1$ , dat prin elementele sale definatorii  $u$  și  $\beta$ ; i.e., se calculează  $A \leftarrow U_1 A$ .)

1. pentru  $j = 1 : p$ 
  1.  $\tau = (\sum_{i=1}^n u_i a_{ij}) / \beta$
  2. pentru  $i = 1 : n$ 
    1.  $a_{ij} \leftarrow a_{ij} - \tau u_i$

Sintaxa de apel a procedurii **As** este

$$A = \mathbf{As}(u, \beta, A).$$

**Ad** (Procedură de înmulțire la dreapta a unei matrice  $A \in \mathbb{R}^{m \times n}$  cu un reflector  $U_1$ , dat prin elementele sale definatorii  $u$  și  $\beta$ ; i.e., se calculează  $A \leftarrow AU_1$ .)

1. pentru  $i = 1 : m$ 
  1.  $\tau = \left( \sum_{j=1}^n a_{ij} u_j \right) / \beta$
  2. pentru  $j = 1 : n$ 
    1.  $a_{ij} \leftarrow a_{ij} - \tau u_j$

Sintaxa de apel a procedurii **Ad** este

$$A = \mathbf{Ad}(A, u, \beta).$$

Utilizând procedurile **Z**, **As** și **Ad**, se obține imediat următorul algoritm.

ALGORITM 5.11 (**HQ** – Reducerea la forma superior Hessenberg)  
 (Dată o matrice  $A \in \mathbb{R}^{n \times n}$ , algoritmul calculează o secvență de reflectori  $U_2, U_3, \dots, U_{n-1}$  astfel încât matricea (5.15) este superior Hessenberg. De asemenea se calculează matricea de transformare  $Q = U_2 U_3 \cdots U_{n-1}$ .)

1.  $Q = I_n$
2. pentru  $k = 1 : n - 2$ 
  1.  $[u, \beta, A(k+1 : n, k)] = \mathbf{Z}(A(k+1 : n, k))$
  2.  $A(k+1 : n, k+1 : n) = \mathbf{As}(u, \beta, A(k+1 : n, k+1 : n))$
  3.  $A(1 : n, k+1 : n) = \mathbf{Ad}(A(1 : n, k+1 : n), u, \beta)$
  4.  $Q(1 : n, k+1 : n) = \mathbf{Ad}(Q(1 : n, k+1 : n), u, \beta)$ .

Pentru apelul algoritmului **HQ** se utilizează sintaxa generală

$$[Q, H] = \mathbf{HQ}(A),$$

care exprimă posibilitatea de a memora rezultatele în alte tablouri decât cele inițiale deși calculele se fac cu suprascrierea internă a matricei inițiale.

### 5.3.2 Faza iterativă a algoritmului QR

Etapa iterativă a algoritmului **QR** utilizează, într-o manieră implicită, metodele puterii și puterii inverse pentru reducerea unei matrice la forma Schur (reală).

Precizăm că din motive de simplitate vom prezenta exclusiv algoritmul **QR** cu deplasare explicită cu pași simpli care "funcționează" numai pentru matrice reale cu spectru real.

Presupunem că matricea  $H \in \mathbb{R}^{n \times n}$  are o structură superior Hessenberg. Algoritmul **QR** cu deplasare explicită construiește un șir de matrice

$$H = H_1, H_2, \dots, H_k, H_{k+1}, \dots \quad (5.17)$$





```

7.  $Q(:, 1:m) \leftarrow Q(:, 1:m)\tilde{Q}$       % actualizare  $Q$ 
8.  $iter \leftarrow iter + 1$ 
9. tipărește  $iter$  și  $H$ 
10. dacă  $|H(m, m-1)| < tol$ 
    1.  $H(m, m-1) \leftarrow 0$ 
    2.  $m \leftarrow m - 1$                 % deflație

```

### Observații

1. Dacă nu aveți timp să scrieți o procedură eficientă de factorizare QR a unei matrice superior Hessenberg puteți folosi funcția `qr` din MATLAB pentru factorizarea QR a matricelor generale.

2. Dacă nu v-a reușit obținerea unui program de reducere la forma superior Hessenberg, în locul funcției `dv`. `HQ` puteți utiliza funcția `hess` din MATLAB.

3. La instrucțiunea 4.10, pentru neglijarea elementelor subdiagonale se preferă satisfacerea unei toleranțe relative, e.g. a unei condiții de forma

$$|H(k, k-1)| < tol (|H(k-1, k-1)| + |H(k, k)|).$$

## 5.4 Sarcini de lucru

### 5.4.1 A. In laborator

1. Se vor edita și testa programele MATLAB pentru implementarea metodelor puterii și puterii inverse.
2. Se va edita și testa un program MATLAB pentru implementarea algoritmului HQ de reducere la forma superior Hessenberg a unei matrice prin transformări ortogonale de asemănare. Se va compara rezultatul obținut cu cel oferit de funcția MATLAB `hess`.
3. Se va edita și testa un program MATLAB pentru implementarea algoritmului QR de reducere a unei matrice cu spectru real în forma superior Hessenberg la forma Schur (reală). Programul va afișa matricea curentă din șirul QR pentru a se putea urmări anularea asimptotică a elementelor subdiagonale.

Pentru aceeași matrice se vor utiliza mai multe niveluri de toleranță  $tol$  (e.g.  $1.e-15$ ,  $1.e-10$ ,  $1.e-5$ ) și se vor compara numerele de iterații necesare pentru aducerea la forma Schur.

**Observație.** Pentru crearea unei matrice dense cu spectru impus (în particular real) se poate proceda în felul următor: dacă  $L = \{l_1, l_2, \dots, l_n\}$  este spectrul (real) impus atunci următorul program MATLAB crează o matrice  $A$  densă cu spectrul  $L$ :

```

L=[ $l_1, l_2, \dots, l_n$ ];
A=diag(L);

```

```
T=randn(n);  
A=T*A/T;
```

Se recomandă impunerea unor valori proprii ușor de recunoscut (de exemplu, numere întregi).

#### 5.4.2 B. Acasă

- 1 Se va edita și testa un program MATLAB eficient de triangularizare ortogonală cu rotații a unei matrice superior Hessenberg și un program MATLAB pentru implementarea algoritmului **QR** cu deplasare explicită cu pași simpli, de reducere a unei matrice cu spectru real în forma superior Hessenberg la forma Schur (reală) fără a calcula explicit matricea  $Q$  din factorizarea **QR** a matricei  $H - \mu I$ .
- 2 Se va edita și testa un program MATLAB pentru implementarea algoritmului **QR** cu deplasare *implicită* cu pași simpli, de reducere a unei matrice cu spectru real în forma superior Hessenberg la forma Schur (reală). Programul va afișa matricea curentă din șirul **QR** pentru a se putea urmări anularea asimptotică a elementelor subdiagonale.