Calculul iterativ al unui vector propriu.
Metoda puterii. Metoda puterii inverse

Fie matricea $\mathbf{A} \in \mathbf{R}^{n\times n}$ având spectrul de valori proprii

$$\lambda(\mathbf{A}) = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$$

şi fie

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$$

un set de vectori proprii, de normă euclidiană unitară, ai matricei $\mathbf{A}$ (i.e. $\mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{x}_i$, $\|\mathbf{x}_i\| = 1$, $\mathbf{i = 1 : n}$). Proprietățile din următoarea propoziție se vor dovedi utile în demersul calculatoriu al valorilor şi vectorilor proprii.

Propoziție. Cu notațiile de mai sus avem

a) pentru orice $\beta, \mu \in \mathbf{C}$

$$\lambda(\mathbf{A}-\mu\mathbf{I}) = \{\lambda_1-\mu, \lambda_2-\mu, \dots, \lambda_n-\mu\};$$

b) pentru orice $\mathbf{k} \in \mathbf{N}$

$$\lambda(\mathbf{A}^k) = \{\lambda_1^k, \lambda_2^k, \dots, \lambda_n^k\};$$

c) pentru orice $\mu \in \mathbf{C} \setminus \lambda(\mathbf{A})$

$$\lambda\left((\mathbf{A} - \mu\mathbf{I})^{-1}\right) = \left\{\frac{1}{\lambda_1 - \mu}, \frac{1}{\lambda_2 - \mu}, \dots, \frac{1}{\lambda_n - \mu}\right\}$$

şi, în acelaşi timp, vectorii îşi păstrează semnificația, adică sunt vectori proprii pentru matricele $\mathbf{A}-\mu\mathbf{I}$, $\mathbf{A}^k$, $(\mathbf{A}-\mu\mathbf{I})^{-1}$ respectiv

$$(\mathbf{A}-\mu\mathbf{I})\mathbf{x}_i = (\lambda_i-\mu)\mathbf{x}_i$$

$$\mathbf{A}^k\mathbf{x}_i = \lambda_i^k\mathbf{x}_i$$

$$(\mathbf{A} - \mu\mathbf{I})^{-1}\mathbf{x}_i = \frac{1}{\lambda_i - \mu}\mathbf{x}_i$$

pentru toți $\mathbf{i = 1 : n.}$

Demonstrația este imediată prima relație este evidentă, cea de-a doua se obține prin inducție, iar ultima se obține, pentru $\mathbf{A - \mu I}$ nesingulară.

Presupunem acum că mulțimea este ordonată în sensul descrescător al valorilor absolute şi că matricea $\mathbf{A}$ are o valoare proprie dominantă, i.e.

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n| \quad .$$

Dacă $\mathbf{y} \in \mathbf{C}^n$ este un vector de normă euclidiană unitară având o componentă nenulă pe direcția vectorului propriu $\mathbf{x}_1 \in \mathbf{X}$, definim şirul vectorial $\left(\mathbf{y}^{(k)}\right)_{k\in\mathbf{N}}$ şi şirul numeric $\left(\lambda^{(k)}\right)_{k\in\mathbf{N}}$ prin relațiile

1. $\mathbf{y}^{(0)} = \mathbf{y}$ % inițializare
2. Pentru $\mathbf{k = 1,2,}$ … % iterare
   1. $\mathbf{z} \leftarrow \mathbf{A} \cdot \mathbf{y}^{(k-1)}$

2. $\mathbf{y}^{(k)} \leftarrow \dfrac{\mathbf{z}}{\|\mathbf{z}_2\|_2}$

3. $\lambda^{(k)} \leftarrow \left(\mathbf{y}^{(k)}\right)^T \mathbf{A}\mathbf{y}^{(k)}$.

care definesc metoda puterii.

$$\mathbf{y}^{(k)} = \rho_k \mathbf{A}^k \mathbf{y}^{(0)}, \qquad k = 1,2,\ldots$$

unde $\rho_k = \dfrac{1}{\left\|\mathbf{A}^k \mathbf{y}^{(0)}\right\|}$ este un factor scalar de normare. Dacă

$$\mathbf{y}^{(0)} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i$$

cu $\alpha_1 \neq 0$, atunci

$$\mathbf{y}^{(k)} = \rho_k \alpha_1 \lambda_1^k \left[\mathbf{x}_1 + \sum_{i=2}^{n} \frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k \mathbf{x}_i \right] .$$

Datorită condiției de dominanță avem

$$\lim_{k \to \infty} \left(\frac{\lambda_i}{\lambda_1} \right)^k = 0, \qquad i = 2 : n,$$

ceea ce asigură convergența şirului vectorial $\mathbf{y}^{(k)}$ către vectorul propriu unitar asociat valorii proprii dominante

$$\mathbf{y}^{(\infty)} = \lim_{k \to \infty} \mathbf{y}^{(k)} = \mathbf{x}_1 .$$

Viteza de convergență este cu atât mai mare cu cât raportul $\dfrac{|\lambda_2|}{|\lambda_1|}$ este mai mic, metoda puterii fiind performantă în cazul matricelor cu o valoare proprie net dominantă. Şirul numeric $\left(\lambda^{(k)}\right)_{k \in \mathbf{N}^*}$, converge către valoarea proprie dominantă. În principiu, în cadrul metodei puterii, nu este necesară construcția explicită a şirului $\left(\lambda^{(k)}\right)_{k \in \mathbf{N}^*}$, valoarea proprie λ_1 putând fi calculată cu

$$\lambda_1 = \left(\mathbf{y}^{(\infty)}\right)^H \mathbf{A}\mathbf{y}^{(\infty)} = \mathbf{x}_1^H \mathbf{A}\mathbf{x}_1 .$$

În schimb o astfel de evaluare devine foarte benefică în cadrul unei variante a metodei puterii inverse.

Metoda puterii inverse de determinare iterativă a unui vector propriu se bazează pe relația conform căreia matricea $\mathbf{B} = (\mathbf{A} - \mu \mathbf{I})^{-1}$ va avea o valoare proprie net dominantă dacă, "deplasarea" μ este o aproximație, chiar şi grosieră, a unei valori proprii distincte a matricei $\mathbf{A}$. Metoda puterii inverse, este, în fapt metoda puterii aplicată matricei $\mathbf{B}$ de mai sus şi se defineşte

1. $\mathbf{y}^{(0)} = \mathbf{y}$ % inițializare

2. pentru $\mathbf{k} \leftarrow \mathbf{1,2,}$ … % iterare

1. Se rezolvă sistemul liniar $(\mathbf{A} - \mu\mathbf{I})\mathbf{z} = \mathbf{y}^{(k-1)}$

2. $\mathbf{y}^{(k)} \leftarrow \dfrac{\mathbf{z}}{\|\mathbf{z}\|}$

3. $\lambda^{(k)} \leftarrow \left(\mathbf{y}^{(k)}\right)^T \mathbf{A}\mathbf{y}^{(k)}$.

Conform celor arătate la metoda puterii, şirul $\left(\mathbf{y}^{(k)}\right)_{k\in\mathbf{N}}$ este convergent către vectorul propriu al matricei $\mathbf{A}$ asociat valorii proprii celei mai apropiate de deplasarea μ. Proprietăţi de convergenţă deosebite se obţin în varianta în care deplasarea μ este modificată de la o iteraţie la alta utilizând aproximaţia curentă $\mu_k = \lambda^{(k)}$ a unei valori proprii a matricei $\mathbf{A}$. Această variantă, cunoscută sub denumirea de *iterarea câtului Rayleigh*, este definită de vectorul iniţial de normă unitară $\mathbf{y}$, deplasarea iniţială

$$\mu = \mu_0 = \mathbf{y}^T\mathbf{A}\mathbf{y}$$

şi procedura în care μ se înlocuieşte cu $\mu_k = \lambda^{(k)}$. Dacă iniţializarea $\mathbf{y}$ nu este ortogonală pe direcţia vectorului propriu asociat valorii proprii celei mai apropiate de μ_0 (condiţie generic satisfăcută de o alegere aleatoare a lui $\mathbf{y}$), atunci se poate arăta că şirul are o convergenţă pătratică către vectorul propriu corespunzător, adică există o constantă $\rho > 0$ astfel încât

$$\left\|\mathbf{y}^{(k)} - \mathbf{y}^{(\infty)}\right\| \le \rho \cdot \left\|\mathbf{y}^{(k-1)} - \mathbf{y}^{(\infty)}\right\|^2,$$

ceea ce înseamnă o viteză de convergenţă foarte bună. Mai mult, pentru matrice hermitice (simetrice) se poate demonstra că se asigură chiar o convergenţă cubică.

Aceste excelente proprietăţi de convergenţă sunt moderate de efortul de calcul sporit, în raport cu metoda puterii, necesare pentru rezolvarea unui sistem liniar la fiecare iteraţie. Totuşi, dacă matricea $\mathbf{A}$ are o structură particulară, cum ar fi de exemplu forma Hessenberg, metoda puterii inverse, cu alegerea deplasării egală cu câtul Rayleigh, este considerată ca fiind cea mai performantă. Deplasări de forma $\mathbf{A} \rightarrow \mathbf{A} - \mu\mathbf{I}$, destinate sporirii vitezei de convergenţă, pot fi utilizate, în principiu şi în cazul metodei puterii, însă cu o eficienţă mult mai redusă.

Formele concrete ale algoritmilor pentru implementarea metodelor puterii şi puterii inverse se obţin prin completarea schemelor de calcul (76), (83) cu trunchierea şirurilor respective pe baza unor criterii de oprire a iteraţiilor cum ar fi

$$\left| 1 - \left| \left(\mathbf{y}^{(k-1)}\right)^T \mathbf{y}^{(k)} \right| \right| < \varepsilon ,$$

ε este o toleranţă dată, care consideră aproximaţia drept satisfăcătoare atunci când doi vectori succesivi sunt "suficient de coliniari" (realizarea trunchierii când norma diferenţei a două aproximaţii succesive scade sub o toleranţă dată poate să nu dea rezultate: dacă sensul vectorului se schimbă la fiecare pas acestă normă tinde către 2). Dacă acest criteriu nu este atins după un număr maxim admis de iteraţii, se consideră că metoda nu asigură convergenţa dorită.

Cu aceste precizări putem să prezentăm algoritmii formali de implementare a metodei puterii şi a metodei puterii inverse.

Metoda puterii directe

```
[λ,x,OK]=MPD(A, y, maxit, ε)
% Intrări:
%     A = matricea dată;
%     y = aproximaţia iniţială a vectorului propriu;
```

```
%     ε = toleranța impusă;
%     maxit = numărul maxim de iterații.
% Ieşiri:
%   λ = valoarea proprie dominantă;
%     x = vectorul propriu asociat.
%   OK = indicator succes / esec
  OK=1;
  k=0;
  y=y/norm(y);
  e=2*ε;
  while e>=ε
     k=k+1;
     if k>maxit
        OK=0;
        error('nu s-a atins precizia impusa');
     end
     x=A*y;
     x=x/norm(x);
     e=abs(1-abs(y'*x));
     y=x;
  end
  λ=x'*A*x;
```

Metoda puterii inverse cu deplasare Rayleigh

```
[λ,x,OK]=MPI(A,y,maxit,ε)
% Intrări:
%     A = matricea dată;
%     y = aproximația inițială a vectorului propriu;
%     ε = toleranța impusă;
%     maxit = numărul maxim de iterații.
% Ieşiri:
%   λ = valoarea proprie dominantă;
%     x = vectorul propriu asociat.
%   OK = indicator succes / esec
  k=0;
  y=y/norm(y);
  λ=y'*A*y;
  E=2*ε;
  while e>=ε
     k=k+1;
     if k>maxit
        OK=0;
        error('nu s-a atins precizia impusa');
     end
     x=(A-λ)\y;
     x=x/norm(x);
     e=abs(1-abs(y'*x));
     y=x;
     λ=x'*A*x;
  end
```

Pentru a asigura eficiența necesară algoritmului, rezolvarea sistemului liniar trebuie să exploateze eventuale particularități structurale ale matricei sistemului.
Algoritmii prezentați sunt destinați calculului unui singur vector propriu şi al valorii proprii asociate. Dacă se doreşte calculul mai multor valori şi vectori proprii pentru o matrice dată, acești algoritmi trebuie completați cu o procedură de deflație. Principiile unei astfel de proceduri sunt date în demonstrația teoremei 9.4 şi sunt utilizate implicit în cadrul algoritmului **QR** prezentat ulterior.
În încheierea acestui paragraf, considerăm necesar să subliniem faptul că cea mai bună metodă de calcul a unui vector propriu, în absența oricărei informații inițiale despre valoarea proprie asociată, este metoda puterii inverse cu deplasarea dată de câtul Rayleigh.

Deflația valorilor proprii.

Presupunem că am localizat o pereche dominantă $(\lambda_1, \mathbf{x}_1)$ şi dorim să continuăm procesul pentru celelalte perechi proprii, presupunând că acestea sunt distincte. Situația este aplicabilă matricelor simetrice cu elemente reale.

$\lambda(A)=\{\lambda_1,\lambda_2,...,\lambda_n\}$
$X(A)=\{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_n\}$

Dacă formăm matricea:

$\mathbf{B}=(\mathbf{I}_n-\mathbf{x}_1\mathbf{y}^T)\mathbf{A}$, cu $\mathbf{y}^T\mathbf{x}_1=1$

atunci:

$\lambda(B)=\{0,\lambda_2,...,\lambda_n\}$
$X(B)=\{\mathbf{x}_1,\mathbf{z}_2,...,\mathbf{z}_n\}$

în care:

$\mathbf{z}_i=\mathbf{x}_i-\mathbf{x}_1\mathbf{y}^T\mathbf{x}_i=\mathbf{x}_i-\mathbf{y}^T\mathbf{x}_i\mathbf{x}_1, \quad i=2:n$

Demonstrație: Mai întâi vom alege vectorul $\mathbf{y}$ coliniar cu $\mathbf{x}$ (sunt şi alte opțiuni posibile).

$$\mathbf{y}=\alpha.\mathbf{x}, \qquad \mathbf{y}^T\mathbf{x} = \alpha\cdot\mathbf{x}^T\mathbf{x} = \alpha\cdot\|\mathbf{x}\|_2^2 = 1 \Rightarrow \alpha = \frac{1}{\|\mathbf{x}\|_2^2} \Rightarrow \mathbf{y} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2^2}$$

$$\mathbf{B}.\mathbf{x}_1=(\mathbf{I}_n-\mathbf{x}_1\mathbf{y}^T)\mathbf{A}.\mathbf{x}_1=\lambda_1\mathbf{x}_1-\lambda_1\mathbf{x}_1\mathbf{y}^T\mathbf{x}_1=0$$

ceea ce arată că $\mathbf{B}$ are valoarea proprie $\mathbf{0}$ şi vectorul propriu corespunzător $\mathbf{x}_1$.

$$\mathbf{B}.\mathbf{x}_i=\mathbf{A}.\mathbf{x}_i-\mathbf{A}.\mathbf{x}_1.\mathbf{y}^T.\mathbf{x}_i=\lambda_i\mathbf{x}_i-\lambda_1\mathbf{x}_1\mathbf{y}^T\mathbf{x}_i$$

$$\mathbf{B}\cdot\mathbf{x}_i = \lambda_i\underbrace{\left(\mathbf{x}_i - \mathbf{x}_1\mathbf{y}^T\mathbf{x}_i\right)}_{\mathbf{z}_i} = \lambda_i\mathbf{z}_i$$

$$\mathbf{B}\cdot\mathbf{z}_i = \mathbf{B}\cdot\left(\mathbf{x}_i - \mathbf{x}_1\mathbf{y}^T\mathbf{x}_i\right) = \mathbf{B}\cdot\mathbf{x}_i - \underbrace{\left(\mathbf{B}\cdot\mathbf{x}_1\right)}_{0}\mathbf{y}^T\mathbf{x}_i = \mathbf{B}\cdot\mathbf{x}_i$$

Din ultimele două relații rezultă că: $\mathbf{B}.\mathbf{z}_i = \lambda_i\mathbf{z}_i$, ceea ce arată că matricea $\mathbf{B}$ are valorile proprii λ_i şi vectorii proprii $\mathbf{z}_i$, cu $\mathbf{i=2:n}$.

Altă alternativă de deflație consideră: $\mathbf{B}=\mathbf{A}.(\mathbf{I}_n-\mathbf{x}_1\mathbf{y}^T)$, cu $\mathbf{y}^T\mathbf{x}_1=1$
Vom arăta că:
$\lambda(B)=\{0,\lambda_2,...,\lambda_n\}$
$X(B)=\{\mathbf{x}_1,\mathbf{z}_2,...,\mathbf{z}_n\}$

în care:

$$z_i = x_i - \frac{\lambda_1}{\lambda_i}\left(y^T x_i\right)x_1, \qquad i = 2 : n$$

În mod asemănător:

$$B \cdot x_1 = A \cdot x_1 - A \cdot x_1 \cdot \underbrace{\left(y^T \cdot x_1\right)}_{1} = 0$$

$$B.x_i = A.x_i - A.x_1 y^T x_i = \lambda_i x_i - \lambda_1 x_1 y^T x_i$$

$$B \cdot x_i = \lambda_i\left(x_i - \frac{\lambda_1}{\lambda_i}\left(y^T x_i\right) \cdot x_1\right) = \lambda_i z_i$$

$$B \cdot z_i = B \cdot \left(x_i - \frac{\lambda_1}{\lambda_i}\left(y^T x_i\right) \cdot x_1\right) = B \cdot x_i - \frac{\lambda_1}{\lambda_i}\left(y^T x_i\right)\underbrace{\left(B \cdot x_1\right)}_{0} = B \cdot x_i$$

de unde rezultă $B.z_i = \lambda_i z_i$.

Metoda de deflație *Wielandt* consideră:

$$y^T = \frac{1}{\lambda_1 x_{1i}}\begin{bmatrix} A_{i1} & A_{i2} & \cdots & A_{in} \end{bmatrix}$$

Matricea $B = A - \lambda_1 x_1 y^T$ are:

$$\lambda(B) = \{0, \lambda_2, \ldots, \lambda_n\}$$
$$X(B) = \{x_1, z_2, \ldots, z_n\}$$

în care:

$$z_i = x_i - x_1 \cdot \frac{x_{ii}}{x_{1i}}, \qquad i = 2 : n$$

$$\left(A \cdot x_1\right)_i = \sum_{j=1}^{n} A_{ij} x_{1j} = \left(\lambda_1 x_1\right)_i = \lambda_1 x_{1i}$$

$$y^T x_1 = \frac{1}{\lambda_1 x_{1i}} \sum_{j=1}^{n} A_{ij} x_{1j} = 1$$

$$y^T x_i = \frac{1}{\lambda_1 x_{1i}} \underbrace{\sum_{j=1}^{n} A_{ij} x_{1j}}_{(Ax_i)_i} = \frac{\lambda_i x_{ii}}{\lambda_1 x_{1i}}$$

$$B \cdot x_1 = \left(A - \lambda_1 x_1 y^T\right) \cdot x_1 = A \cdot x_1 - \lambda_1 x_1 \underbrace{\left(y^T x_1\right)}_{1} = 0$$

$$B.x_i = (A - \lambda_1 x_1 y^T) x_1 = A.x_i - \lambda_1 x_1 (y^T x_i)$$

$$B \cdot x_i = \lambda_i x_i - \lambda_1 x_1 \frac{\lambda_i x_{ii}}{\lambda_1 x_{1i}} = \lambda_i\left(x_i - x_1 \frac{x_{ii}}{x_{1i}}\right) = \lambda_i z_i$$

$$B \cdot z_i = B \cdot \left(x_i - x_1 \frac{x_{ii}}{x_{1i}}\right) = B \cdot x_i - \underbrace{B \cdot x_1}_{0} \cdot \frac{x_{ii}}{x_{1i}} = B \cdot x_i$$

$$B.z_i = \lambda_i . z_i$$

Se ştie că $\mathbf{A}$ şi $\mathbf{A}^T$ au acelaşi spectru, iar vectorii proprii corespunzători a două valori proprii diferite sunt ortogonali.

Fie matricea $\mathbf{B} = \mathbf{A} - \lambda_1 \frac{\mathbf{x}_1\mathbf{y}_1^T}{\mathbf{y}_1^T\mathbf{x}_1}$ în care $\mathbf{x}_1$ şi $\mathbf{y}_1$ sunt vectorii proprii dominanți din $\mathbf{A}$ şi $\mathbf{A}^T$

$\lambda(\mathbf{B})=\{0,\lambda_2,\ldots,\lambda_n\}$
$\mathbf{X}(\mathbf{B})=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}$

$$\mathbf{B}\mathbf{x}_1 = \mathbf{A}\mathbf{x}_1 - \lambda_1\mathbf{x}_1 \frac{\mathbf{y}_1^T\mathbf{x}_1}{\mathbf{y}_1^T\mathbf{x}_1} = \mathbf{0}$$

$$\mathbf{B}\mathbf{x}_i = \mathbf{A}\mathbf{x}_i - \lambda_1 \frac{\mathbf{x}_1\overbrace{\left(\mathbf{y}_1^T\mathbf{x}_i\right)}^{0}}{\mathbf{y}_1^T\mathbf{x}_1} = \mathbf{A}\mathbf{x}_i \text{ c.c.t.d.}$$

Reducerea la forma Hessenberg

Am văzut că aducerea unei matrice $\mathbf{A} \in \mathbf{R}^{n\times n}$ date la forma Schur reală prin transformări (ortogonale) de asemănare este, în absența unor informații apriorice privind valorile şi vectorii proprii, un proces, în mod necesar, infinit. Se ridică întrebarea naturală privitoare la structura cea mai apropiată de forma Schur reală, care poate fi obținută printr-un proces de calcul finit.

Teorema 1. Oricare ar fi $\mathbf{A} \in \mathbf{R}^{n\times n}$ există o matrice ortogonală $\mathbf{U} \in \mathbf{R}^{n\times n}$, calculabilă printr-o procedură directă (i.e. printr-un număr finit de operații aritmetice elementare, inclusiv extrageri de radical) astfel încât

$$\mathbf{H} = \mathbf{U}^T\mathbf{A}\mathbf{U}$$

este superior Hessenberg (adică $\mathbf{H(i,j)} = \mathbf{0}$ pentru toți $\mathbf{i > j + 1}$).

Demonstrație. Vom da o demonstrație constructivă care va indica modul de obținere efectivă a matricei $\mathbf{U}$ ca un produs de reflectori elementari (transformări Householder)

$$\mathbf{U} = \mathbf{U}_2\mathbf{U}_3 \cdots \mathbf{U}_{n-1}$$

conform schemei de calcul

1. **pentru k=1:n-2**
   1. Se determină reflectorul $\mathbf{U}_{k+1}$ asfel încât $(\mathbf{U}_{k+1}\mathbf{A})(\mathbf{k+2:n},\ \mathbf{k})=\mathbf{0}$
   2. $\mathbf{A} \leftarrow \mathbf{U}_{k+1}\mathbf{A}$
   3. $\mathbf{A} \leftarrow \mathbf{A}\mathbf{U}_{k+1}$.

care suprascrie matricea $\mathbf{A}$ cu matricea $\mathbf{H}$. În primul rând să observăm că reflectorul elementar $\mathbf{U}_{k+1}$ care anulează elementele $\mathbf{k + 2 : n}$ din coloana $\mathbf{k}$ a matricei $\mathbf{A}$ există şi are structura

$$\mathbf{U}_{k+1} = \mathbf{diag}\left(\mathbf{I}_k, \overline{\mathbf{U}}_{k+1}\right),$$

unde $\overline{\mathbf{U}}_{k+1}$ este un reflector de ordinul $\mathbf{n - k}$ şi indice 1. În consecință, la primul pas ($\mathbf{k = 1}$) premultiplicarea matricei $\mathbf{A}$ cu $\mathbf{U}_2$ anulează elementele $\mathbf{A(3 : n, 1)}$, iar postmultiplicarea cu $\mathbf{U}_2 = \mathbf{U}_2^T$, datorită structurii nu afectează prima coloană a matricei $\mathbf{U}_2\mathbf{A}$, respectiv conservă zerourile create. Prin urmare, am adus matricea $\mathbf{A}$ la forma superior Hessenberg în prima coloană.
Presupunem acum că după $\mathbf{k-1}$ paşi matricea

$$\mathbf{A} \leftarrow \mathbf{U}_k \ldots \mathbf{U}_2 \mathbf{A} \mathbf{U}_2 \ldots \mathbf{U}_k$$

este superior Hessenberg în primele **k-1** coloane. Acum, premultiplicarea cu reflectorul $\mathbf{U}_{k+1}$, adecvat calculat, nu modifică primele **k** linii şi (în plus, datorită structurii din acest moment a lui **A**) primele **k-1** coloane (conservând astfel zerourile obținute în paşii precedenți) şi anulează elementele **A(k + 2 : n, k)**. Postmultiplicarea cu $\mathbf{U}_{k+1}$ nu modifică primele **k** coloane conservând zerourile create la paşii precedenți şi cele create prin premultiplicarea de la pasul curent.

Deci procesul de reducere la forma Hessenberg prin transformări Householder de asemănare poate fi inițializat şi apoi continuat, într-un număr finit de **n-2** paşi, fiecare pas având un număr finit de operații, până la capăt, obținându-se în final matricea

$$\mathbf{A} \leftarrow \mathbf{H} = \mathbf{U}_{n-1} \cdots \mathbf{U}_3 \mathbf{U}_2 \mathbf{A} \mathbf{U}_2 \mathbf{U}_3 \cdots \mathbf{U}_{n-1}$$

în formă superior Hessenberg. Teorema este demonstrată.

Elementele definitorii $\mathbf{v}_k \in \mathbf{R}^n$, $\beta_k = \frac{1}{2}\|\mathbf{v}_k\|^2$ ale reflectorilor elementari

$$\mathbf{U}_k = \mathbf{I}_n - \frac{\mathbf{v}_k \mathbf{v}_k^T}{\beta_k}, \qquad \mathbf{k = 2 : n - 1}$$

pot fi memorate în matricea $\mathbf{V} \in \mathbf{R}^{n \times (n-2)}$ ($\mathbf{v}_k$ este coloana **k-1** a matricei **V**) şi vectorul $\mathbf{b} \in \mathbf{R}^{n-2}$ $\left(\beta_k = \mathbf{b}_{k-1},\ \mathbf{k = 2:n-2}\right)$.

Algoritmul de implementare a schemei de calcul de reducere la forma superior Hessenberg, la care adăugăm acumularea opțională a transformărilor (i.e. calculul matricei **U**) va utiliza intensiv rezultatele relative la transformările Householder prezentate în capitolul 2.

Concret, se presupune cunoscută procedura de calcul fiabil şi economic al elementelor definitorii ale unui reflector ce anulează anumite componente ale unui vector dat, procedură care va fi utilizată explicit în scrierea algoritmilor din acest capitol. Amintim în acest sens că fiind dat un vector nenul $\mathbf{x} \in \mathbf{R}^n$, elementele definitorii $\mathbf{v}_k \in \mathbf{R}^n$, β_k ale reflectorului (91) care asigură anularea componentelor **k + 1 : n** ale lui **x**, i.e. $(\mathbf{U}_k\mathbf{x})(\mathbf{k+1:n}) = \mathbf{0}$, se calculează economic cu relațiile

1. $\mathbf{v}_k(1 : k-1) = 0$
2. $\sigma = \mathbf{sign}(\mathbf{x}(k)) \cdot \|\mathbf{x}(k : n)\|_2$
3. $\mathbf{v}_k(k) = \mathbf{x}(k) + \sigma$
4. $\beta_k = \mathbf{x}(k) \cdot \sigma$
5. $\mathbf{v}_k(k + 1 : n) = \mathbf{x}(k + 1 : n)$.

În ceea ce priveşte procedurile de premultiplicare şi de postmultiplicare ale unei matrice cu un reflector, pentru comoditatea cititorului, preferăm să le prezentăm detaliat aici. Astfel este suficient să dispunem de procedurile de premultiplicare (eventual cu suprascriere) $A \leftarrow B = UA$ a unei matrice $\mathbf{A} \in \mathbf{R}^{n \times p}$ şi de postmultiplicare (eventual cu suprascriere) $\mathbf{A} \leftarrow \mathbf{B} = \mathbf{AU}$ a unei matrice $\mathbf{A} \in \mathbf{R}^{m \times n}$ cu un reflector $\mathbf{U} = \mathbf{I}_n - \frac{\mathbf{u}\mathbf{u}^T}{\beta}$ de ordin **n** si indice 1. Precizăm că în variantele cu suprascriere aceasta se efectuează intern, element cu element, fără a se crea matricea intermediară **B**.

Premultiplicarea economică $\mathbf{A \leftarrow B = UA}$ foloseşte partiția pe coloane a matricelor $\mathbf{A, B}$. Notând $\mathbf{a_j = A(:, j),\ b_j = B(:, j),\ j = 1 : p}$ schema de calcul este

1. **pentru** $\mathbf{j = 1 : p}$
   1. $\mathbf{a_j \leftarrow b_j = Ua_j}$

în care produsul $\mathbf{a_j \leftarrow b_j = Ua_j = (I_n - \frac{uu^T}{\beta})a_j = a_j - \tau u}$ se efectuează economic calculând mai întâi scalarul $\mathbf{\tau = \frac{u^T a_j}{\beta} = \frac{\sum_{i=1}^{n} u_i a_{ij}}{\beta}}$. Rezultă următorul algoritm.

```
[B]=refmat(u,β,A)
% Premultiplicarea unei matrice cu un reflector elementar de indice 1.
% Intrări: u ∈ R^n, β ∈ R = elementele definitorii ale reflectorului;
%          A ∈ R^(n×p) = matricea dată.
%Ieşiri:   B = UA.
[n,p]=size(A);
for j=1:p
  τ = u(1:n)*A(1:n,j)/β;
  B(1:n,j)=A(1:n,j)- τ*u(1:n);
end
```

Postmultiplicarea economică $\mathbf{A \leftarrow B = AU}$ foloseşte, absolut analog, partiția pe linii a matricelor $\mathbf{A,\ B}$. Notând $\mathbf{a_i^T = A(i, :),\ b_i^T = B(i, :),\ i = 1 : m}$ schema de calcul este

1. **pentru** $\mathbf{i = 1 : m}$
   1. $\mathbf{a_i^T \leftarrow b_i^T = a_i^T U}$

în care produsul $\mathbf{a_i^T \leftarrow b_i^T = a_i^T U = a_i^T(I_n - \frac{uu^T}{\beta}) = a_i^T - \tau u^T}$ se efectuează economic calculând mai întâi scalarul $\mathbf{\tau = \frac{a_i^T u}{\beta} = \frac{\sum_{j=1}^{n} a_{ij} u_j}{\beta}}$. Rezultă următorul algoritm.

```
[B]= matref(A,u, β)
% Postmultiplicarea unei matrice cu un reflector elementar de indice 1.
% Intrări: A ∈ R^(m×n) = matricea dată;
%          u ∈ R^n, β ∈ R = elementele definitorii ale reflectorului.
%Ieşiri:   B = AU
  [m,n]=size(A);
  for i=1:m
```

```
    τ = A(i,1:n)*u(1:n)/β;
    B(i,1:n)=A(i,1:n)- τ*u(1:n);
  end
```

Utilizând relațiile de calcul ale unui reflector elementar ce anulează componentelele dorite ale unui vector dat precum şi funcțiile **refmat** şi **matref** algoritmul de reducere a unei matrice la forma superior Hessenberg prin transformări ortogonale de asemănare poate fi scris într-o formă mai concisă cum este cea prezentată în continuare.

```
% Reducerea la forma Hessenberg
[A,U,V,b]=HQ(A)
% Intrări:
%         A = matricea dată;
%Ieşiri:  A ← H = U^T AU = matricea superior Hessenberg ortogonal asemenea cu A;
%        U ∈ R^{n×n} = matrice ortogonală de transformare
%        V ∈ R^{n×(n-2)} = matrice cu coloane vectorii definitori ai transformărilor;
%        b ∈ R^{n-2} = vector conținând scalarii definitori ai transformărilor.
  [n,n]=size(A);
  U=eye(n,n);
  for k=1:n-2
    % Determinarea lui U_{k+1}, i.e. v_{k+1} = V(:,k), ρ_{k+1} = b(k)
    V(1 : k, k) = 0
    σ = sign(A(k + 1, k)) · ‖A(k + 1 : n, k)‖
    V(k + 1, k) = A(k + 1, k) + σ
    b(k) = V(k + 1, k) · σ
    V(k + 2 : n, k) = A(k + 2 : n, k);
    % Calculul A ← U_{k+1}A
    A(k + 1, k) = −σ;
    A(k + 2 : n, k) = 0;
    [A(k + 1 : n, k + 1 : n)] =
         refmat(V(k + 1 : n, k), b(k), A(k + 1 : n, k + 1 : n));
    % Calculul A ← AU_{k+1}
    [A(:,k+1:n)]=matref(A(;,k+1:n),V(k+1:n,k),b(k));
    % Acumularea transformărilor (i.e. calculul A ← AU_{k+1})
    [U(:, k + 1 : n)] =
         matref(U(:, k + 1 : n), V(k + 1 : n, k), b(k));
```

Complexitatea algoritmului este $O(n^3)$ fiind necesare $N_{op} \cong \frac{10}{3} n^3$ operații elementare în format virgulă mobilă. Dacă se calculează şi matricea de transformare U, atunci trebuie executate încă $N'_{op} \cong \frac{4}{3} n^3$ operații.

Algoritmul 9.3 este un algoritm numeric stabil, existând demonstrații pentru faptul că matricea Hessenberg calculată într-o aritmetică de virgulă mobilă este un rezultat exact pentru o matrice inițială foarte puțin perturbată.

Observație: Este posibilă reducerea matricei $\mathbf{A}$ la forma superior Hessenberg prin *transformări de asemănare neortogonale* cu un efort de calcul redus aproximativ la jumătate din cel cerut de algoritmul 9.3. O astfel de posibilitate o oferă construcția transformării sub forma unui produs de transformări gaussiene elementare stabilizate (deci cu permutări de linii) $(\mathbf{P_k M_k})$, $\mathbf{k = 2 : n-1}$ cu $\mathbf{M_k = I_n - m_k e_k^T}$ și $\mathbf{P_k}$ o matrice de permutare a liniei $\mathbf{k}$ cu o linie $\mathbf{i_k \geq k}$ (vezi problema rezolvată R9.17). Motivele pentru care, totuşi, sunt preferate transformările ortogonale se referă la faptul că acestea din urmă conservă condiționarea numerică a problemei de calcul a valorilor şi vectorilor proprii şi asigură o stabilitate numerică mult mai bună.
Matricea superior Hessenberg ortogonal asemenea cu matricea dată obținută cu algoritmul de mai sus nu este unica matrice Hessenberg care satisface (86). Într-adevăr dacă $\mathbf{A}$ este matricea dată şi $\mathbf{Q_0 \in R^{n\times n}}$ este o matrice ortogonală arbitrară atunci aplicând algoritmul matricei $\mathbf{Q_0^T A Q_0}$ obținem

$$\mathbf{H' = U^T Q_0^T A Q_0 U = Q^T A Q}$$

cu $\mathbf{Q = Q_0 U}$ şi $\mathbf{H'}$ superior Hessenberg, în general diferită de matricea $\mathbf{H}$. Se observă totuşi că

$$\mathbf{U e_1 = U_2 U_3 \dots U_{n-1} e_1 = e_1}$$

şi deci

$$\mathbf{Q e_1 = Q_0 U e_1 = Q_0 e_1} .$$

De aici se poate extrage ideea că determinantă pentru identitatea transformării este prima sa coloană. Această observație are o importanță capitală în dezvoltarea procedurală a algoritmului $\mathbf{QR}$ cu deplasare implicită prezentat într-o secțiune ulterioară şi care reprezintă un rezultat de importanță recunoscută în calculul numeric matriceal. Contextul în care acționează algoritmul $\mathbf{QR}$ cu deplasare implicită este cel al matricelor superior Hessenberg *ireductibile* i.e. al matricelor superior Hessenberg cu *toate* elementele de pe prima subdiagonlă nenule. Acum transpunem observația de mai sus într-un cadru formal dat de următoarea teoremă.

Teorema 2. Fie $\mathbf{A \in R^{n\times n}}$ şi matricele ortogonale $\mathbf{U, V \in R^{n\times n}}$ astfel încât matricele

$$\mathbf{H = U^T A U}, \qquad \mathbf{G = V^T A V}$$

sunt ambele superior Hessenberg ireductibile, i.e.

$$\mathbf{h_{j+1,j} \neq 0}, \qquad \mathbf{g_{j+1,j} \neq 0}, \qquad \mathbf{j = 1 : n-1} .$$

Dacă matricele $\mathtt{U}$ şi $\mathtt{V}$ au aceeaşi primă coloană

$$\mathbf{U(:,1) = V(:,1)}$$

atunci între celelalte coloane există relațiile

$$\mathbf{U(:,j) = \pm V(:,j)}, \qquad \mathbf{j = 2 : n} .$$

Mai mult, avem cu necesitate

$$\mathbf{h_{ij} = \pm g_{ij}}, \qquad \forall\ \mathbf{i,j} .$$

Demonstrație. Matricea ortogonală $\mathbf{Q = V^T U}$ are ca primă coloană vectorul $\mathbf{q_1 = Q(:,1) = V^T U(:,1) = V^T V(:,1) = e_1}$. Celelalte coloane ale matricei Q pot fi exprimate recurent în raport de precedentele în modul următor. Rezultă $\mathbf{A = UHU^T = VGV^T}$, i.e. $\mathbf{QH = GQ}$, relație care, scrisă pe coloane, ne conduce la

$$GQ(:,j) = QH(:,j) = \sum_{k=1}^{j} Q(:,k)h_{k,j} + Q(:,j+1)h_{j+1,j} \ .$$

Acum, notând $q_j = Q(:,j)$, avem

$$q_{j+1} = \frac{Gq_j - \sum_{k=1}^{j} q_k h_{kj}}{h_{j+1,j}} \ .$$

Este uşor de observat că această recurență induce, în mod necesar, matricei Q o structură superior triunghiulară. Într-adevăr, G fiind superior Hessenberg iar q_1 având numai primul element nenul rezultă că q_2 are numai primele două elemnte nenule etc. Inversa unei matrice superior triunghiulare este superior triunghiulară. Dar Q fiind ortogonală rezultă că este simultan superior şi inferior triunghiulară, i.e. Q este o matrice diagonală. Cum elementele diagonale ale unei matrice ortogonale diagonale nu pot fi decât egale cu ± 1, rezultă

$$U(:,j) = VQ(:,j) = V \cdot (\pm e_j) = \pm V(:,j) \ .$$

De asemenea din $QH = GQ$ avem $H = QGQ$, de unde rezultă că teorema este demonstrată.

Miezul teoremei constă în faptul că transformările ortogonale U şi V de reducere a aceleiaşi matrice la o formă superior Hessenberg ireductibilă sunt esențial determinate (mai puțin un semn la fiecare coloană) de prima lor coloană. Dacă prima coloană din cele două matrice de transformare este identică, atunci şi rezultatul reducerii este esențial acelaşi (mai puțin semnul unor elemente). Aici termenii *"esențial aceleaşi"* semnifică faptul că cele două matrice superior Hessenberg sunt "la fel de departe" de o formă Schur a matricei inițiale, al carei calcul este obiectivul principal al problemei valorilor şi vectorilor proprii.

CALCULUL VALORILOR PROPRII ŞI VECTORILOR PROPRII PENTRU MATRICE NESIMETRICE

Sistemele dinamice liniare formează o clasă importantă, cu aplicaţii largi în domenii variate ale ştiinţei şi tehnicii. Acestea sunt reprezentate numeric de un set de matrice, iar proprietăţile lor sunt exprimate, din punct de vedere cantitativ, în termenii elementelor acestor matrice. Astfel stabilitatea sistemelor liniare este condiţionată exclusiv de plasarea valorilor proprii ale matricei de stare a sistemului.

În acest capitol sunt prezentate, în principal, cele mai performante metode de calcul ale valorilor proprii ale matricelor reale dense nesimetrice care se întâlnesc cel mai des în aplicaţii. Din motive de eficienţă aceste metode fac apel, practic exclusiv, la o aritmetică reală deşi valorile proprii ale unei matrice reale pot fi şi complexe. Din motive de condiţionare numerică calculul vectorilor proprii se face cu mai puţină acurateţe şi, din acest motiv, practica numerică recomandă utilizarea în aplicaţii în locul lor a vectorilor Schur, definiţi în acest capitol. Cazul matricelor complexe poate fi redus la cazul matricelor reale de ordin dublu.

Valori şi vectori proprii

Prezentăm aici conceptele de valoare proprie şi vector propriu ale unei matrice şi principalele lor proprietăţi. De asemenea, având în vedere faptul că valorile proprii, ca şi elementele vectorilor proprii, ale unei matrice reale pot fi numere complexe, reamintim câteva noţiuni referitoare la spaţiul vectorial C^n.

În acest capitol, prin vector complex *n*-dimensional *x* vom înţelege un vector coloană cu *n* elemente complexe

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \qquad x_i \in C, \quad i = 1:n.$$

Mulţimea tuturor vectorilor complecşi *n*-dimensionali va fi notată C^n. Evident, orice vector $x \in C^n$ poate fi scris, în mod unic, sub forma

$$x = u + iv, \quad u, v \in R^n$$

unde *i* este unitatea imaginară. Mulţimea tuturor matricelor cu elemente complexe având *m* linii şi *n* coloane va fi notată $C^{m\times n}$. Deşi spaţiile liniare R^n şi C^n au foarte multe proprietăţi comune, induse de axiomele comune ale corpurilor de numere complexe şi reale, există şi diferenţe cum sunt cele datorate definirii diferite a produsului scalar standard a doi vectori şi a normei vectoriale induse de acesta. Pentru evidenţierea unora dintre aceste diferenţe vom nota conjugatul transpusului (i.e. conjugatul hermitic al) unui vector $x \in C^n$ prin

$$x^H = \bar{x}^T,$$

unde bara superioară semnifică operaţia de conjugare complexă iar indicele ^{T} transpunerea.

Dacă $x, y \in C^n$, definim produsul scalar al celor doi vectori prin numărul complex

$$\langle x, y \rangle = y^H x.$$

Evident, avem

$$\langle y, x \rangle = \overline{\langle x, y \rangle}.$$

Cu ajutorul produsului scalar putem defini conceptul de ortogonalitate în C^n. Vom spune că doi vectori sunt ortogonali dacă produsul lor scalar este nul, i.e.

$$x^H y = y^H x = 0.$$

De asemenea, plecând de la observaţia că scalarul $x^H x$ este un număr real pozitiv, oricare ar fi vectorul nenul $x \in C^n$, se poate defini următoarea normă pe C^n

$$\|\cdot\|_2 : C^n \to R_+, \quad \|x\|_2 = \sqrt{x^H x} \ ,$$

numită normă euclidiană.

Mai general, foarte multe rezultate referitoare la norme pe R^n pot fi extinse la norme corespunzătoare pe C^n.

Rolul matricelor reale simetrice este jucat în $C^{n\times n}$ de matricele *hermitice*. O matrice $A \in C^{n\times n}$ se numeşte *hermitică* dacă

$$A^H = A.$$

Dacă $A \in C^{n\times n}$ este hermitică, atunci scalarul $\alpha = x^H A x$ este real, oricare ar fi $x \in C^n$, fapt care ne permite definirea matricelor pozitiv-definite în $C^{n\times n}$. O matrice hermitică $A \in C^{n\times n}$ este *pozitiv-definită* dacă

$$x^H A x > 0 \ \ \forall x \in C^n, \quad x \neq 0 .$$

În sfârşit, în analogie cu matricele reale ortogonale, vom spune despre o matrice $Q \in C^{n\times n}$ că este *unitară* dacă are coloanele ortogonale şi de normă euclidiană (7) unitară, respectiv dacă

$$Q^H Q = I_n \ .$$

Vom introduce acum conceptele de *valoare proprie* şi *vector propriu asociat*. Deşi vom considera mai ales cazul matricelor reale, aceste noţiuni pot fi definite în cadrul mai general al matricelor complexe.

Definiţia 9.1 **Fie $A \in C^{n\times n}$. Un număr $\lambda \in C$ se numeşte *valoare proprie* a matricei A dacă există un vector nenul $x \in C^n$, numit *vector propriu asociat* valorii proprii $\lambda \in C$, astfel încât**

$$Ax = \lambda x.$$

Sistemul liniar omogen (11) admite soluţii nenule dacă şi numai dacă

$$p(\lambda) = \det(\lambda I_n - A) = 0 .$$

Polinomul monic p(λ) de gradul n se numeşte *polinom caracteristic* al matricei $E(\lambda) \subset C^n$, iar ecuaţia se numeşte *ecuaţie caracteristică*.

Prin urmare, valorile proprii ale unei matrice sunt zerourile polinomului caracteristic. Dacă luăm în considerare multiplicităţile, numărul valorilor proprii este egal cu ordinul matricei. Mulţimea

$$\lambda(A) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$$

a valorilor proprii ale matricei A poartă numele de *spectrul* (de valori proprii al) matricei A. Numărul real

$$\rho(A) = \max_{\lambda_i \in \lambda(A)} \left(|\lambda_i| \right)$$

se numeşte *raza spectrală* a matricei A.

Dacă A este o matrice reală atunci valorile proprii complexe apar în perechi complex conjugate.

Valorile proprii ale unei matrice satisfac relaţiile

$$\sum_{i=1}^{n}\lambda_i = \sum_{i=1}^{n} A(i,i) = \mathrm{tr}(A),$$

$$\prod_{i=1}^{n}\lambda_i = \det(A),$$

unde tr*(A)* este, prin definiţie, *urma* matricei A (vezi problema rezolvată R9.1).

Dacă $x \in C^n$ este un vector propriu asociat valorii proprii $\lambda \in \lambda(A)$ atunci oricare ar fi $0 \neq \alpha \in C$ vectorul $y = \alpha x$ este de asemenea un vector propriu al matricei A asociat aceleiaşi valori proprii. În consecinţă, vectorii proprii asociaţi unei valori proprii nu sunt unic determinaţi decât ca direcţii. Mai mult, se arată că mulţimea vectorilor proprii asociaţi unei valori proprii $\lambda \in \lambda(A)$ generează un *subspaţiu liniar* $E(\lambda) \subset C^n$ numit *subspaţiul propriu* al valorii proprii λ. Subspaţiile proprii sunt subspaţii A*-invariante* în sensul următoarei definiţii.

Definiţia 9.2 **Fie o matrice $A \in C^{n\times n}$. Un subspaţiu liniar $V \subset C^n$ se numeşte subspaţiu -*invariant* al matricei A sau, pe scurt, subspaţiu A-*invariant* dacă**

$$AV \subset V, \quad \text{i.e.} \quad Ax \in V \ \ \forall x \in V.$$

O matrice $A \in C^{n\times n}$ care admite n vectori proprii liniar independenţi se numeşte *simplă*. Se arată că o matrice care are cele n valori proprii distincte este simplă, dar reciproca nu este, în general, adevărată. Cei n vectori proprii liniar independenţi ai unei matrice simple formează o bază a spaţiului C^n, numită *baza proprie* asociată matricei A.

Ordinul de multiplicitate n_i al rădăcinii λ_i a polinomului caracteristic se numeşte *multiplicitate algebrică* a valorii proprii $\lambda_i \in \lambda(A)$. Dacă $n_i = 1$ valoarea proprie se numeşte *simplă*. Dimensiunea $\dim E(\lambda_i)$ a subspaţiului propriu al valorii proprii $\lambda_i \in \lambda(A)$ se numeşte *multiplicitatea geometrică* a valorii proprii λ_i. În general, avem

$$\dim E(\lambda_i) \leq n_i .$$

Încheiem acest paragraf cu un set de proprietăţi ale subspaţiilor invariante şi, în particular, ale subspaţiilor proprii ale unei matrice.

Teorema 1 **Fie matricea $A \in C^{n\times n}$.**

a) Dacă $x_1, x_2, \ldots, x_p$ sunt vectori proprii ai matricei A şi notăm $X = \left[x_1\ x_2 \ldots x_p\right] \in C^{n\times p}$, atunci subspaţiul liniar

$$S = \mathrm{Im}\, X = \left\{ y \in C^n \;\middle|\; \exists z \in C^n \ \text{astfel incat} \ y = Xz \right\}$$

este A-invariant, i.e. orice subspaţiu generat de vectori proprii ai unei matrice este un subspaţiu invariant al acelei matrice.

b) Dacă $S \subset C^n$ este un subspaţiu A-invariant p-dimensional şi vectorii $x_1, x_2, \ldots, x_p$ formează o bază a lui S, atunci există o matrice $B \in C^{p\times p}$ cu

$$\lambda(B) \subset \lambda(A)$$

astfel încât matricea $X = \left[x_1\ x_2 \ldots x_p\right] \in C^{n\times p}$ satisface condiţia

$$AX = XB.$$

Matricea B se numeşte restricţia matricei A la subspaţiul A-invariant S şi se notează $B = A|S$.

c) Dacă are loc o relaţie de forma (20) **atunci $S = \mathrm{Im}\, X$ este un subspaţiu A-invariant.**

d) Complementul ortogonal $T = S^{\perp} \subset C^n$ al unui subspaţiu A-invariant S este un subspaţiu A^H-invariant.

Demonstraţie. a) Pentru orice vector $y \in S$ există un vector $z \in C^p$ astfel încât $y = Xz$ şi, deci,

$$Ay = AXz = \left[Ax_1\ Ax_2 \dots Ax_p\right]z = \left[\lambda_1 x_1\ \lambda_2 x_2 \dots \lambda_p x_p\right]z = \left[x_1\ x_2 \dots x_p\right]\cdot\begin{bmatrix}\lambda_1 z_1\\ \lambda_2 z_2\\ \vdots\\ \lambda_p z_p\end{bmatrix} = Xw \in S,$$ unde $\lambda_i, i = 1:p$ sunt valorile proprii ale matricei A asociate vectorilor proprii $x_i, i = 1:p$. Prin urmare, subspaţiul S este A-invariant. b) Coloanele matricei X formând o bază a subspaţiului S, orice vector $x \in S$ se scrie sub forma $x = Xy,\ y \in C^p$. Pe de altă parte S fiind un subspaţiu A-invariant rezultă că $Ax \in S, \forall x \in S$, în particular pentru toţi $j \in 1:p$ avem $Ax_j \in S$, i.e. există vectorii $b_j \in C^p,\ j = 1:p$ astfel încât $Ax_j = Xb_j$. Rezultă că matricea $B = \left[b_1\ b_2\ \dots b_p\right] \in C^{p\times p}$ satisface egalitatea (20). Mai mult, dacă $z \in C^p$ este un vector propriu al matricei B asociat valorii proprii $\mu \in \lambda(B)$ atunci din (20) avem $AXz = XBz = \mu Xz$. Cum $z \neq 0$ iar X este o matrice monică (i.e. cu coloanele liniar independente) rezultă că $y = Xz \neq 0$, adică y este vector propriu al matricei A (conţinut în S) şi, deci, $\mu \in \lambda(A)$. Rezultă $\lambda(B) \subset \lambda(A)$. c) Presupunem că are loc (20) pentru o matrice $X \in C^{n\times p}$ şi o matrice $B \in C^{p\times p}$. Atunci avem $AXz = XBz,\ \forall z \in C^p$, i.e. $Ax = Xy \in \operatorname{Im} X, \forall x = Xz \in \operatorname{Im} X$, ceea ce înseamnă că $S = \operatorname{Im} X$ este un subspaţiu A-invariant. d) Fie $x \in S$ şi $y \in T = S^{\perp}$ doi vectori arbitrari. Subspaţiul S fiind A-invariant avem $Ax \in S$ şi, prin urmare, $y^H Ax = (A^H y)^H x = 0$. Cum x este un vector arbitrar din S, rezultă $(A^H y) \perp S$, respectiv $A^H y \in T$ pentru toţi $y \in T$, adică subspaţiul T este A^H-invariant.

Transformări de asemănare. Forma Jordan

Vom fi interesaţi în transformări ale matricei A care conservă spectrul de valori proprii. Astfel de transformări sunt *transformările de asemănare*.

Vom spune că două matrice A şi B sunt *asemenea* dacă există o matrice nesingulară $T \in C^{n\times n}$ astfel încât

$$B = TAT^{-1}.$$

Dacă matricea de transformare T este unitară (în cazul real ortogonală) atunci matricele A şi B se numesc *unitar (ortogonal) asemenea*.

Dacă $A,B \in C^{n\times n}$ sunt asemenea atunci au acelaşi spectru de valori proprii

$$\lambda(A) = \lambda(B),$$

iar dacă $x \in C^n$ este un vector propriu al matricei A asociat valorii proprii λ, atunci vectorul $y \in C^n$, definit de

$$y = Tx,$$

este un vector propriu al matricei B din (21) asociat aceleiaşi valori proprii λ.

Evident, valorile proprii ale unei matrice triunghiulare (în particular diagonale) sunt date de elementele sale diagonale. De aceea suntem interesaţi de situaţiile în care o matrice dată este asemenea cu o matrice având o astfel de structură. O matrice asemenea cu o matrice diagonală se numeşte *diagonalizabilă*.

Dacă pentru toate valorile proprii distincte ale unei matrice date $A \in C^{n\times n}$ avem

$$\dim E(\lambda_i) = n_i,$$

atunci matricea A este diagonalizabilă, respectiv există o matrice nesingulară $T = X^{-1} \in C^{n\times n}$ astfel încât

$$X^{-1}AX = \Lambda = \text{diag}\left(\lambda_1, \lambda_2, \ldots, \lambda_n\right).$$

Într-un astfel de caz matricea A este simplă, coloanele matricei de transformare X formând un set de n vectori proprii liniar independenţi ai matricei A.
În cazul general, cea mai simplă structură care poate fi obţinută prin transformări de asemănare este aşa numita *formă canonică Jordan* definită în următoarea teoremă.

Teorema 2 **Oricare ar fi matricea $A \in C^{n\times n}$ există o matrice nesingulară $X \in C^{n\times n}$ astfel încât**

$$X^{-1}AX = \text{diag}\left(J_1, J_2, \ldots, J_q\right),$$

unde

$$J_i = \begin{bmatrix} \lambda_i & 1 & 0 & \ldots & 0 \\ 0 & \lambda_i & 1 & \ldots & 0 \\ & & \ldots & & \\ 0 & 0 & 0 & \lambda_i & 1 \\ 0 & 0 & 0 & \ldots & \lambda_i \end{bmatrix} \in C^{n_i \times n_i}$$

cu $\lambda_i \in \lambda(A)$ şi $\sum_{i=1}^{q} n_i = n$. Matricele J_i se numesc *blocuri Jordan*.

Numărul şi dimensiunile blocurilor Jordan asociate fiecărei valori proprii distincte din spectrul matricei A sunt unice, dar ordonarea blocurilor în (26) poate fi arbitrară.
Forma canonică Jordan conţine maximul de informaţie structurală privitoare la o matrice dată. Totuşi structura Jordan (respectiv numărul şi dimensiunile blocurilor) este foarte sensibilă la perturbaţiile numerice în elementele matricei şi, din acest motiv, calculul numeric al formei canonice Jordan întâmpină dificultăţi serioase (vezi [5]) şi nu este recomandat pentru calculul valorilor proprii într-o aritmetică aproximativă.

9.1.3. Localizarea valorilor proprii. Cercurile lui Gershgorin

In unele aplicaţii o localizare a valorilor proprii într-un domeniu al planului complex este suficientă pentru a evidenţia anumite proprietăţi. Amintim, în acest context, problema aprecierii stabilităţii sistemelor dinamice liniare. De asemenea, o localizare a valorilor proprii permite stabilirea unor aproximaţii ale acestora utile pentru iniţializările necesare în diverse metode iterative.
Există mai multe rezultate privitoare la localizarea valorilor proprii dintre care cel mai cunoscut este dat de *teorema cercurilor lui Gershgorin.*

Teorema 3 **Spectrul de valori proprii al unei matrice $A \in C^{n\times n}$ satisface condiţia**

$$\lambda(A) \subset \bigcup_{i=1}^{n} D_i,$$

unde D_i sunt discurile (sau cercurile) lui Gershgorin definite de $D_i = \left\{ z \in C \;\middle|\; |z - a_{ii}| \le \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \right\}.$

Demonstraţie. Fie $\lambda \in \lambda(A)$ şi x un vector propriu asociat lui λ. Fie x_i componenta de modul maxim a lui x, i.e. $\frac{|x_j|}{|x_i|} \le 1$ pentru toţi $j \in 1:n$. Atunci linia i a relaţiei de definiţie $Ax = \lambda x$ se poate scrie sub forma $(\lambda - a_{ii})x_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}x_j$, de unde rezultă imediat inegalităţile $|\lambda - a_{ii}| \le \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \cdot \frac{|x_j|}{|x_i|} \le \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$, q.e.d.

Se poate arăta că dacă unul din discurile lui Gershgorin este izolat de celelalte, atunci el conţine o valoare proprie a matricei A şi numai una, oferind un mijloc de separare a valorilor proprii.

Alte metode de localizare a valorilor proprii ale unei matrice, în general mai fine dar mult mai puţin utilizate în practica curentă, fac obiectul unor probleme rezolvate sau propuse (vezi secţiunile 9.8 şi 9.9).

9.1.4. Condiţionarea numerică a valorilor şi vectorilor proprii

Aşa cum s-a menţionat în primul capitol al lucrării, acurateţea rezultatelor obţinute prin rezolvarea unei probleme de calcul numeric într-un mediu de calcul aproximativ depinde esenţial de doi factori. Pe de o parte precizia obţinută depinde de "calitatea numerică a problemei", i.e. de sensibilitatea rezultatelor la variaţiile datelor problemei în ipoteza unor calcule exacte. Este vorba de aşa numita *condiţionare numerică a problemei*. Pe de altă parte calitatea rezultatelor depinde de calitatea algoritmului utilizat, exprimată prin aşa numita *stabilitate numerică a algoritmului* respectiv. Întrucât condiţionarea numerică a valorilor proprii este independentă de modul de calcul al acestora, vom prezenta aici rezultatele esenţiale privitoare la aceasta, urmând ca discuţia despre calităţile algoritmilor propuşi să aibă loc în finalul capitolului.

Aprecierea condiţionării numerice se face uzual prin stabilirea unor margini superioare pentru variaţiile rezultatelor în raport cu variaţiile datelor de intrare. Chiar dacă, de multe ori, aceste margini sunt supraevaluate ele oferă posibilitatea detectării situaţiilor critice în care erorile au tendinţa de a ieşi de sub control. Evaluarea sensibilităţii valorilor şi vectorilor proprii se bazează pe proprietăţile de continuitate ale acestora în raport cu elementele matricei. Precizăm de la început că ne vom ocupa în exclusivitate de cazul valorilor proprii simple. Valorile proprii multiple sunt, în general, mult mai sensibile decât cele simple, iar studiul analitic al sensibilităţii lor ridică dificultăţi importante.

A. Condiţionarea numerică a valorilor proprii

Fie o matrice $A \in C^{n \times n}$, o valoare proprie simplă $\lambda \in \lambda(A)$ şi vectorii proprii de normă euclidiană unitară $x, y \in C^n$ la dreapta, respectiv la stânga, asociaţi valorii proprii λ. Prin urmare avem $Ax = \lambda x$ şi, respectiv, $y^H A = \lambda y^H$. Considerăm acum o perturbaţie $E = \varepsilon G$ cu $\|G\| = 1$ a matricei A. Vom fi interesaţi de evaluarea valorii proprii $\lambda(\varepsilon)$ şi a vectorului $x(\varepsilon)$ a matricei perturbate $\tilde{A} = A + E$. Conform unor rezultate clasice [10], valoarea proprie $\lambda(\varepsilon)$ şi vectorul propriu $x(\varepsilon)$ admit dezvoltări în serie de puteri

$$\lambda(\varepsilon) = \lambda + \mu_1\varepsilon + \mu_2\varepsilon^2 + \dots$$
$$x(\varepsilon) = x + v_1\varepsilon + v_2\varepsilon^2 + \dots$$

convergente într-o vecinătate a punctului $\varepsilon = 0$. Evident, avem $\lambda(0) = \lambda$ şi $x(0) = x$ iar funcţiile $\lambda(\varepsilon)$ şi $x(\varepsilon)$ sunt continue şi derivabile în domeniul de convergenţă. Vom considera în cele ce urmează că parametrul ε este suficient de mic pentru a putea neglija puterile superioare ale acestuia şi a ne mărgini la aproximaţiile liniare ale funcţiilor $\lambda(\varepsilon)$ şi $x(\varepsilon)$. În aceste condiţii putem aprecia că sensibilitatea valorii proprii $\lambda(\varepsilon)$ pe "direcţia" definită de matricea G este dată de $|\mu_1|$ şi că pentru toate direcţiile de perturbare sensibilitatea sa poate fi definită prin numărul de condiţionare

$$\kappa_\lambda = \max_{\substack{G\in C^{n\times n}\\ \|G\|=1}} |\mu_1| .$$

Pentru a găsi o evaluare a acestui număr de condiţionare considerăm relaţia $(A+\varepsilon G)x(\varepsilon) = \lambda(\varepsilon)x(\varepsilon)$ care derivată în raport cu ε ne conduce, pentru $\varepsilon = 0$, la

$$Gx + Av_1 = \mu_1 x + \lambda v_1 .$$

Înmulţind această relaţie la stânga cu y^H rezultă $y^H Gx + y^H Av_1 = \mu_1 y^H x + \lambda y^H v_1$, i.e. $y^H Gx = \mu_1 y^H x$ întrucât $y^H A = \lambda y^H$. Ţinând seama acum de faptul că pentru valorile proprii simple avem $y^H x \neq 0$ (vezi problema rezolvată R9.5) obţinem

$$|\mu_1| = \frac{|y^H Gx|}{|y^H x|} \le \frac{\|y\|\cdot\|G\|\cdot\|x\|}{|y^H x|} = \frac{1}{|y^H x|} .$$

Prin urmare, numărul de condiţionare al unei valori proprii simple este

$$\kappa_\lambda = \max_{\substack{G\in C^{n\times n}\\ \|G\|=1}} |\mu_1| = \frac{1}{|y^H x|} .$$

Valoarea minimă a numărului de condiţionare este 1 şi se obţine atunci când vectorii proprii la stânga şi la dreapta, asociaţi valorii proprii de interes, sunt coliniari (aceasta se întâmplă, de exemplu, la matricele hermitice şi, în particular, la matricele reale simetrice). Conform dezvoltărilor de mai sus, condiţionarea κ_λ indică nivelul de amplificare al erorilor absolute din datele iniţiale care se regăsesc în erorile absolute ale valorilor proprii. Prin urmare, pentru valori proprii cu acelaşi număr de condiţionare erorile *relative* vor fi cu atât mai mari cu cât valorile proprii sunt de modul mai mic.

În aplicaţii prezintă interes de multe ori condiţionarea unui grup de valori proprii sau a întregului spectru al unei matrice date. O cale de a defini condiţionarea unui grup de valori proprii este de a considera o normă a vectorului condiţionărilor numerice a valorilor proprii individuale care fac parte din grup. O altă cale face apel la conceptul de *proiector spectral* care exprimă condiţionarea unui grup de valori proprii prin poziţia relativă a subspaţiilor A-invariante şi A^H-invariante generate de vectorii proprii la dreapta şi la stânga asociaţi valorilor proprii din grup. Pentru detalii recomandăm consultarea, de exemplu, a lucrărilor [13], [29], [30].

Încheiem consideraţiile noastre privitoare la condiţionarea numerică a valorilor proprii prin prezentarea teoremei Bauer-Fike care reprezintă un rezultat important ce permite exprimarea condiţionării numerice a spectrului unei matrice simple într-un context general.

Teorema 4 **Considerăm o matrice $A\in C^{n\times n}$ diagonalizabilă şi fie $X\in C^{n\times n}$ o matrice nesingulară de vectori proprii ai matricei A. Dacă $E\in C^{n\times n}$ este o matrice de perturbaţie arbitrară şi $\mu\in\lambda(A+E)$ este o valoare proprie a matricei perturbate $\tilde{A} = A+E$, atunci**

$$e(\mu) = \min_{\lambda\in\lambda(A)} |\lambda-\mu| \le \kappa(X)\cdot\|E\|$$

unde

$$\kappa(X) = \|X\|\cdot\|X^{-1}\|$$

este numărul de condiţionare la inversare al matricei X a vectorilor proprii iar $\|\cdot\|$ este orice normă matriceală consistentă care satisface condiţia

$$\|\operatorname{diag}(\alpha_1,\alpha_2,\ldots,\alpha_n)\| = \max_{i=1:n}(|\alpha_i|)$$

(în particular, $\|\cdot\|$ poate fi oricare din normele $\|\cdot\|_p$, $p = 1,2,\infty$).

Demonstraţia teoremei poate fi găsită în [13]. Aici ne vom mărgini la prezentarea unor succinte comentarii. În primul rând rezultatele specificate în teoremă sunt valabile nu numai pentru perturbaţii mici ci pentru orice nivel al perturbaţiilor. În al doilea rând, trebuie evidenţiat rolul jucat de numărul de condiţionare la inversare al matricei vectorilor proprii în caracterizarea condiţionării valorilor proprii. Interpretând raportul $\frac{e(\mu)}{\|E\|}$ drept sensibilitate (condiţionare) numerică a valorii proprii a matricei A pentru care se atinge minimul din (36) rezultă că numărul de condiţionare la inversare al matricei vectorilor proprii este o margine superioară pentru numerele de condiţionare ale tuturor valorilor proprii şi, în consecinţă, poate fi folosit drept măsură pentru condiţionarea întregului spectru. Se pot face şi alte conexiuni între rezultatele din teorema Bauer-Fike şi numerele de condiţionare definite anterior [13]. Încheiem aceste scurte comentarii cu remarcarea faptului că teorema 9.4 confirmă observaţia că matricele hermitice (simetrice în cazul real) au un spectru perfect condiţionat numeric întrucât, după cum vom vedea în capitolul 11, acestea au un set complet de vectori proprii ortogonali, i.e. matricea vectorilor proprii este unitară (ortogonală) şi, deci, are în raport cu norma $\|\cdot\|_2$ numărul de condiţionare la inversare egal cu 1 (cel mai mic posibil).

B. Condiţionarea numerică a vectorilor proprii

Fie o matrice $A \in C^{n\times n}$ cu valorile proprii simple λ_k, $k=1:n$ şi vectorii proprii asociaţi $x_k \in C^n$ de normă euclidiană unitară. Considerăm matricea perturbată $\tilde{A} = A + E$, unde $E = \varepsilon G$ cu $\|G\| = 1$ şi fie $\lambda_k(\varepsilon)$, $x_k(\varepsilon)$, $k=1:n$ valorile proprii, respectiv vectorii proprii asociaţi ai matricei perturbate. La fel ca în cazul valorilor proprii, pe baza dezvoltării în serie de puteri (31) a funcţiei vectoriale $x(\varepsilon)$, în ipoteza unor variaţii ε mici, putem aprecia că sensibilitatea vectorului propriu $x(\varepsilon)$ pe "direcţia" definită de matricea G este dată de $\|v_1\|$ şi că pentru toate direcţiile de perturbare sensibilitatea sa poate fi definită prin numărul de condiţionare

$$\kappa_x = \max_{\substack{G\in C^{n\times n}\\ \|G\|=1}} \|v_1\|.$$

Adaptând notaţiile la noul context, relaţia (33) se poate scrie sub forma

$$Gx_k + Av_1^{(k)} = \mu_1^{(k)} x_k + \lambda_k v_1^{(k)}.$$

Deoarece, în ipotezele menţionate, vectorii proprii x_k, $k=1:n$ formează o bază a spaţiului C^n, rezultă că putem exprima vectorul $v_1^{(k)}$ în raport cu această bază prin relaţia $v_1^{(k)} = \sum_{i=1}^{n} \gamma_i^{(k)} x_i$ care, introdusă în (39), produce

$$\sum_{\substack{i=1\\ i\neq k}}^{n} \gamma_i^{(k)} (\lambda_k - \lambda_i) x_i = (G - \mu_k I_n) x_k.$$

Înmulţind la stânga ultima relaţie cu y_i^H, unde y_i este vectorul propriu la stânga al matricei A asociat valorii proprii λ_i şi, ţinând seama (vezi problema rezolvată R9.5) de faptul că $y_j^H x_i = 0, \forall i \neq j$ şi $y_i^H x_i \neq 0,$ obţinem

$$\gamma_i^{(k)} = \frac{y_i^H G x_k}{(\lambda_k - \lambda_i) y_i^H x_k}, \quad i = 1:n, \quad i \neq k, \tag{42}$$

cu care numărul de condiţionare (39) devine

(43) $$\kappa_{x_k} = \max_{\substack{G\in C^{n\times n}\\ \|G\|=1}} \left\| v_1^{(k)} \right\| = \max_{\substack{G\in C^{n\times n}\\ \|G\|=1}} \left\| \sum_{\substack{i=1\\ i\neq k}}^{n} \frac{y_i^H G x_k}{(\lambda_k-\lambda_i) y_i^H x_i} x_i \right\| \le \sum_{\substack{i=1\\ i\neq k}}^{n} \frac{1}{\left|\lambda_k-\lambda_i\right|\cdot\left|y_i^H x_i\right|}.$$

Se observă din evaluarea (43) că sensibilitatea unui vector propriu la variaţiile elementelor matricei depinde atât de sensibilităţile tuturor celorlalte valori proprii cât şi de depărtarea valorii proprii asociate de celelalte valori proprii. În consecinţă, mai ales în cazul existenţei unor valori proprii apropiate, condiţionarea numerică a vectorilor proprii este mult mai rea decât cea a valorilor proprii. De aceea în literatura de specialitate se recomandă evitarea calculului vectorilor proprii. Din fericire, în majoritatea aplicaţiilor, aceştia pot fi înlocuiţi cu succes de vectorii Schur (vezi secţiunea ce urmează).
Problema condiţionării unor grupuri de vectori proprii se formulează mai general sub forma condiţionării subspaţiilor invariante. Condiţionarea subspaţiilor invariante este influenţată decisiv de localizarea valorilor proprii asociate. Este însă posibil ca un subspaţiu invariant generat de vectori proprii rău condiţionaţi să aibă o condiţionare foarte bună dacă grupul corespunzător de valori proprii este bine separat de restul valorilor proprii. Pentru detalii privitoare la cuantificarea condiţionării subspaţiilor invariante se poate consulta lucrarea [29].

9.2. Forma Schur reală. Tehnici de deflaţie

O modalitate de abordare a calculului valorilor proprii ce pare a fi naturală, respectiv determinarea polinomului caracteristic şi rezolvarea ecuaţiei caracteristice, este practic abandonată în calculul numeric actual pentru dimensiuni de matrice cât de cât semnificative, pe de o parte din cauza sensibilităţii ridicate a rădăcinilor în raport cu perturbaţiile numerice în coeficienţii ecuaţiilor algebrice, iar pe de altă parte din cauza complexităţii relativ ridicate a algoritmilor disponibili pentru calculul coeficienţilor polinomului caracteristic. Din aceste motive, metodele numerice cele mai performante de calcul a valorilor proprii fac apel la structuri (cvasi)triunghiulare, cum este *forma Schur (reală).*
Baza teoretică a calculului performant al valorilor proprii ale unei matrice este dată de următoarea teoremă.

Teorema 4. **Pentru orice matrice** $A\in C^{n\times n}$ **(în particular** $A\in R^{n\times n}$**) există o matrice unitară** $\tilde{Q}\in C^{n\times n}$ **astfel încât matricea unitar asemenea cu** A

(44) $$\tilde{S} = \tilde{Q}^H A \tilde{Q} \in C^{n\times n}$$

este superior triunghiulară.

În cazul real, pentru orice matrice $A\in R^{n\times n}$ **există o matrice ortogonală** $Q\in R^{n\times n}$ **astfel încât matricea ortogonal asemenea cu** A

(45) $$S = Q^T A Q = S \in R^{n\times n}$$

are o structură cvasisuperior triunghiulară

(46) $$S = \begin{bmatrix} S_{11} & S_{12} & \dots & S_{1q} \\ 0 & S_{22} & \dots & S_{2q} \\ & & \dots & \\ 0 & 0 & \dots & S_{qq} \end{bmatrix}$$

unde blocurile diagonale $S_{ii}, i=1:q$ **sunt matrice** 1×1 **sau** 2×2**, cele de dimensiune** 2×2 **având valorile proprii complexe.**

Matricea $\tilde{S}$ din (27) poartă numele de *formă Schur* (**FS**) a matricei A dacă A este complexă şi de *formă Schur complexă* (**FSC**) a lui A dacă A este reală. Matricea S din (45), (46) poartă numele de *formă Schur reală* (**FSR**) a matricei *A*. Coloanele $\tilde{q}_j = \tilde{Q}e_j,\ j=1:n$ ale matricei $\tilde{Q}$ (respectiv coloanele $q_j = Qe_j,\ j=1:n$ ale matricei *Q*) se numesc *vectori Schur* ai lui A şi formează o bază unitară a spaţiului C^n (respectiv, o bază ortogonală a spaţiului R^n) numită *baza Schur* asociată lui A. În multe aplicaţii

vectorii Schur pot înlocui cu succes vectorii proprii. Evident, elementele diagonale ale matricei $\tilde{S}$ sunt valorile proprii ale matricei A, iar în cazul formei Schur reale avem

(47) $$\lambda(A)=\lambda(S)=\bigcup_{i=1}^{q}\lambda(S_{ii})$$

şi, în consecinţă, în acest caz calculul spectrului de valori proprii se reduce la rezolvarea unui set de ecuaţii algebrice de grad cel mult doi.

Demonstraţia teoremei9.4. Această demonstraţie pune în evidenţă o tehnică procedurală numită *deflaţie*, care stă la baza algoritmului fundamental de reducere a unei matrice reale date la forma Schur reală printr-un şir de transformări ortogonale de asemănare.

a) Mai întâi arătăm cum se pune în evidenţă, printr-un pas de deflaţie, o valoare proprie reală. Dacă $\lambda\in\lambda(A)$ este reală, fie $x_1\in R^n$ un vector propriu asociat. Fără restrângerea generalităţii putem presupune $\|x_1\|_2=1$. Considerăm o matrice ortogonală Q_1 a cărei primă coloană este x_1, i.e.

(48) $$Q_1=\begin{bmatrix}x_1 & Y\end{bmatrix},\quad Y\in R^{n\times(n-1)}$$

(o astfel de matrice există întotdeauna: de exemplu Q_1 poate fi un reflector elementar care anulează elementele $2:n$ ale vectorului x_1). Evident, avem

(49) $$x_1^TY=0$$

şi, ţinând seama de faptul că $Ax_1=\lambda_1x_1$, rezultă

(50) $$x_1^TAx_1=\lambda_1x_1^Tx_1=\lambda_1,\qquad Y^TAx_1=\lambda_1Y^Tx_1=0,$$

de unde obţinem matricea

(51) $$A_1=Q_1^TAQ_1=\begin{bmatrix}\lambda_1 & x_1^TAY\\ 0 & Y^TAY\end{bmatrix}=\begin{bmatrix}\lambda_1 & g_1^T\\ 0 & B\end{bmatrix}$$

care este în formă Schur reală în prima coloană.

b) Un pas de deflaţie corespunzător unei perechi de valori proprii complex conjugate

(52) $$\lambda_{1,2}=\alpha\pm i\beta,\quad \beta\neq0$$

pune în evidenţă un bloc diagonal 2×2 din forma Schur reală. Într-adevăr nu este greu de arătat că vectorii proprii asociaţi valorilor proprii (52) pot fi aleşi complex conjugaţi

(53) $$x_{1,2}=u\pm iv$$

cu $u,v\in R^n$ liniar independenţi. Din relaţia de definiţie

(54) $$A(u\pm iv)=(\alpha\pm i\beta)\cdot(u\pm iv)$$

rezultă imediat

(55) $$A\cdot\begin{bmatrix}u & v\end{bmatrix}=\begin{bmatrix}u & v\end{bmatrix}\cdot\begin{bmatrix}\alpha & \beta\\ -\beta & \alpha\end{bmatrix}.$$

Vectorii u şi v fiind liniar independenţi generează un subspaţiu liniar $S=\mathrm{Im}\begin{bmatrix}u & v\end{bmatrix}$ bidimensional. Fie $y_1,y_2\in S$ o bază ortonormală a acestui subspaţiu şi $Z\in R^{n\times(n-2)}$ o completare până la o matrice ortogonală a matricei $Y=\begin{bmatrix}y_1 & y_2\end{bmatrix}\in R^{n\times2}$ respectiv astfel încât matricea

(56) $$Q_1=\begin{bmatrix}Y & Z\end{bmatrix}$$

este ortogonală. În raport cu baza de mai sus vectorii $u, v \in S$ se pot scrie

(57) $u = t_{11}y_1 + t_{21}y_2, \quad v = t_{12}y_1 + t_{22}y_2,$
respectiv

(58) $[u\ v] = YT$

unde $T \in R^{2\times 2}$ este nesingulară, întrucât *u* şi *v* sunt liniar independenţi. În consecinţă, din (55) avem

(59) $$AY = YT \cdot \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} \cdot T^{-1} = YS_{11}$$

unde matricea

(60) $$S_{11} = T \cdot \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} \cdot T^{-1} \in R^{2\times 2}$$

are valorile proprii complex conjugate $\lambda_{1,2}$. Prin urmare din (56), (59) avem

(61) $$\begin{cases} Y^T AY = S_{11} \\ Z^T AY = 0 \end{cases},$$

de unde

(62) $$A_1 = Q_1^T A Q_1 = \begin{bmatrix} S_{11} & Y^T AZ \\ 0 & Z^T AZ \end{bmatrix} = \begin{bmatrix} S_{11} & G \\ 0 & B \end{bmatrix}$$

iniţiindu-se reducerea la forma Schur reală prin punerea în evidenţă a unui bloc diagonal 2×2.
Procesul de deflaţie iniţializat în (51) sau (62) poate fi continuat acţionând ca mai sus asupra matricei *B*, ceea ce este echivalent cu acţiunea asupra matricei A_1 prin transformări ortogonale de asemănare definite de matrice de forma $Q_2 = \text{diag}(I, \overline{Q}_2)$. În final, după *q* paşi de deflaţie se obţine forma Schur reală. Matricea cumulată a transformărilor ortogonale este dată de

(63) $Q = Q_1 Q_2 \ldots Q_q$.

Teorema 9.4 este demonstrată.

În continuare ne vom limita universul de investigaţie la clasa de matrice predominantă în practica numerică şi anume cea a matricelor reale. Dacă în cazul complex apar numai deflaţii de tipul a), în cazul real avantajul principal constă în faptul că forma Schur reală a unei matrice reale poate fi calculată utilizând exclusiv o aritmetică reală. În acest fel, în procesul de calcul al valorilor proprii ale unei matrice reale numerele complexe apar numai în final, la calculul valorilor proprii ale blocurilor diagonale 2×2.

Aplicarea procedurii de deflaţie presupune cunoaşterea la fiecare pas al procedurii, a unui vector propriu de normă unitară (respectiv a unei perechi complex conjugate de vectori proprii) asociat valorii proprii corespunzătoare. Calculul unui vector propriu, în absenţa informaţiei privitoare la valoarea proprie asociată, nu este posibil, pentru matrice de ordin superior lui patru, printr-o secvenţă finită de operaţii elementare, pentru că s-ar contrazice un rezultat fundamental din algebră conform căruia rezolvarea ecuaţiilor de grad superior lui patru nu este posibilă printr-un număr finit de operaţii aritmetice elementare şi extrageri de radical.
De aceea procedura de deflaţie trebuie să fie completată cu o procedură de determinare a unui vector propriu, fără cunoaşterea valorii proprii asociate. O astfel de procedură va furniza, inerent, o aproximaţie mai mult sau mai puţin precisă a vectorului propriu, atât datorită erorilor de trunchiere, cât şi erorilor de rotunjire. În acest context este important de ştiut care este cea mai bună aproximaţie a valorii proprii asociate unui vector propriu cunoscut aproximativ. Fie $x \in R^n$ un astfel de vector propriu aproximativ şi

considerăm drept cea mai bună aproximaţie a valorii proprii asociate soluţia în sensul celor mai mici pătrate a sistemului de n ecuaţii şi necunoscuta scalară μ

(64) $x\mu = Ax$,

i.e. acel μ care minimizează norma euclidiană a reziduului de ecuaţie $r(\mu) = Ax - \mu x$. Conform celor arătate în capitolul 3 avem

$$\mu = x^{+}Ax = (x^T x)^{-1} x^T \cdot Ax .$$

Numărul

(65) $$\mu = \frac{x^T Ax}{x^T x},$$

se numeşte *câtul Rayleigh* asociat perechii (A, x) şi joacă un rol esenţial în calculul (aproximativ) al valorilor proprii. Pentru început să observăm că, aplicarea procedurii de deflaţie, cu transformarea (48), în care x_1 este un vector propriu aproximativ, de normă unitară, conduce la

(66) $$A_1 = Q_1^T A Q = \begin{bmatrix} \mu & g_1^T \\ h & B \end{bmatrix}$$

având μ dat de (65) şi vectorul $h = Y^T Ax_1$ de normă euclidiană "mică". Prin urmare, aplicarea transformării (48) reprezintă, în condiţiile date, un pas de deflaţie optim întrucât μ este cea mai bună aproximaţie de care dispunem. O concluzie asemănătoare se poate stabili şi în cazul unei perechi de valori proprii complex conjugate.

Observaţiile de mai sus evidenţiază necesitatea unei proceduri de calcul a unui vector propriu. Dintre metodele cele mai folosite în acest scop sunt tehnicile iterative cunoscute sub denumirile de *metoda puterii* şi *metoda puterii inverse*.

Algoritmul QR cu deplasare explicită

Algoritmul **QR**, considerat drept unul dintre rezultatele cele mai remarcabile ale calculului numeric matriceal (elaborat, în forma sa cea mai evoluată, independent, de către V.N. Kublanovskaia [20] şi J.G.F. Francis [9]), este, în esenţă, o procedură de construcţie iterativă a unui şir de matrice ortogonal asemenea cu matricea iniţială şi rapid convergent către forma Schur reală.

Aşa cum am mai precizat, ne vom ocupa practic în exclusivitate de calculul valorilor proprii pentru matrice reale din trei motive:

- matricele reale sunt cel mai des întâlnite în aplicaţii;
- cazul matricelor complexe poate fi redus la cel al matricelor reale de ordin dublu (vezi problema rezolvată R9.13.);
- versiunea complexă necesită introducerea mijloacelor adecvate (transformări unitare, reflectori şi rotaţii complexe etc.) pentru care nu dispunem de spaţiul tipografic necesar; precizăm totuşi că ideile sunt absolut similare iar cititorul interesat poate consulta [5], [13], [29].

Toate versiunile algoritmului **QR** sunt organizate în două etape:

1. *etapa directă*, de reducere a matricei date la forma superior Hessenberg prin transformări ortogonale de asemănare (algoritmul **HQ** din secţiunea precedentă);
2. *etapa iterativă*, de construcţie recurentă a unui şir de matrice convergent către forma Schur reală.

În cele ce urmează vom considera parcursă prima etapă şi numai la asamblarea întregului algoritm vom evidenţia în mod explicit reducerea la forma superior Hessenberg.

Partea iterativă a algoritmului **QR** prezintă patru versiuni şi anume

a) cu deplasare explicită cu paşi simpli,
b) cu deplasare explicită cu paşi dubli,
c) cu deplasare implicită cu paşi simpli,
d) cu deplasare implicită cu paşi dubli,

dintre care numai ultima este utilizată ca bază a programelor profesionale de calcul al valorilor şi vectorilor proprii. Motivele constau în faptul că versiunile cu pas simplu nu asigură convergenţa în prezenţa valorilor proprii complexe iar versiunea cu paşi dubli cu deplasare explicită este ineficientă, ascunzând, într-un mod cu totul insidios (!), calcule redundante. Totuşi, pentru claritatea expunerii (de exemplu, pasul dublu este, în esenţă, concatenarea a doi paşi simpli succesivi), vom parcurge, la nivelul ideilor, relaţiilor şi al eventualelor scheme de calcul, toate versiunile; scrierea detaliată a algoritmului formal o vom face numai pentru versiunea cea mai evoluată şi anume cea cu deplasare implicită cu paşi dubli.

Algoritmul QR cu deplasare explicită cu paşi simpli

Fie matricea superior Hessenberg ireductibilă $\mathbf{H} \in \mathbf{R}^{n\times n}$ (dacă matricea $\mathbf{H}$ este reductibilă, atunci problema fie îşi reduce dimensiunea, fie se sparge în două subprobleme de dimensiuni mai mici). Algoritmul **QR** cu deplasare explicită cu pas simplu este definit de următoarea schemă de calcul

1. $\mathbf{H}_0 = \mathbf{H}$
2. pentru $\mathbf{k = 0, 1, 2, \ldots}$
    1. $\mathbf{H_k - \mu_k I_n = Q_k R_k}$
    2. $\mathbf{H_{k+1} = R_k Q_k + \mu_k I_n}$

care construieşte iterativ şirul matriceal

$$H_0, H_1, \ldots, H_k, H_{k+1}, \cdots,$$

numit şirul **QR**. Instrucţiunile din ciclu definesc o iteraţie sau *un pas* **QR**. În această secţiune vom descrie succint proprietăţile şirului **QR** şi vom da o justificare calităţilor sale excepţionale de convergenţă.

După cum se observă, un pas **QR** este format dintr-o factorizare **QR** (vezi capitolul 2) a matricei curente H_k având elementele diagonale "deplasate" cu scalarul μ_k şi apoi matricea succesor H_{k+1} se obţine din multiplicarea, în ordine inversă, a rezultatelor factorizării, urmată de refacerea deplasării. Principalele proprietăţi ale acestor transformări sunt sintetizate în următoarea teoremă.

Teorema 1 **a) Matricele H_k şi H_{k+1} şi, prin urmare, toate matricele şirului QR, sunt ortogonal asemenea.**
b) Un pas QR conservă structura superior Hessenberg. Prin urmare dacă matricea iniţială este superior Hessenberg, atunci toate matricele şirului QR sunt superior Hessenberg.

Demonstraţie. a) Din prima relaţie a pasului **QR** avem $R_k = Q_k^T(H_k - \mu_k I_n)$, de unde

$$H_{k+1} = R_k Q_k + \mu_k I_n = Q_k^T(H_k - \mu_k I_n)Q_k + \mu_k I_n = Q_k^T H_k Q_k.$$

În consecinţă, toate matricele H_k din şirul **QR** sunt ortogonal asemenea cu matricea iniţială H.

$$H_k = \tilde{Q}_k^T H \tilde{Q}_k, \qquad k = 0, 1, 2, \ldots$$

unde matricea cumulată a transformărilor ortogonale

$$\tilde{Q}_k = Q_0 Q_1 \cdots Q_{k-1}$$

se poate obţine iterativ cu relaţiile

$$\tilde{Q}_0 = Q_0 = I_n, \qquad \tilde{Q}_{k+1} = \tilde{Q}_k Q_k, \qquad k = 0, 1, 2, \ldots.$$

b) Dacă H_k este o matrice superior Hessenberg, atunci în factorizarea QR din instrucţiunea 2.1 matricea ortogonală Q_k rezultă superior Hessenberg (demonstraţi!). Cum produsul unei matrice superior triunghiulare cu o matrice superior Hessenberg este superior Hessenberg (demonstraţi!), rezultă că H_{k+1} este o matrice superior Hessenberg. Teorema este demonstrată.
Faptul că toate matricele din şirul **QR** sunt superior Hessenberg are urmări benefice pentru eficienţa algoritmului **QR** (complexitatea unei iteraţii se reduce de la $O(n^3)$ la $O(n^2)$.
În continuare expunem o proprietate extrem de importantă a şirului **QR**, proprietate care, nefiind adevărată întotdeauna, nu este un fapt matematic şi, prin urmare, nu a putut să fie inclusă în teorema de mai sus. Proprietatea s-a dovedit adevărată practic în toate aplicaţiile concrete dar a fost invalidată pe exemple special construite în acest scop. Remarcăm această situaţie paradoxală în care un rezultat fără valoare universală stă la baza celei mai performante metode de calcul al valorilor proprii ale unei matrice. Enunţul proprietăţii este următorul.

c) Printr-o alegere adecvată a deplasării μ_k şirul QR converge, *aproape întotdeauna,* către forma Schur a matricei H.

Justificare. Vom arăta mai mult, şi anume că, printr-o alegere corespunzătoare a deplasărilor μ_k, această convergenţă poate fi făcută extrem de rapidă. În fapt, trebuie arătat că unele din elementele subdiagonale ale matricei H_k se anulează asimptotic când $k \to \infty$. Pentru simplificare, considerăm că matricea H are numai valori proprii reale, respectiv trebuie anulate asimptotic toate elementele subdiagonale (în caz contrar trebuie să luăm în consideraţie deplasări complexe).

$$\mathbf{R_k = Q_k^T\left(H_k - \mu_k I_n\right) = Q_k^T\left(\tilde{Q}_k^T H \tilde{Q}_k - \mu_k I_n\right) = \tilde{Q}_{k+1}^T\left(H - \mu_k I_n\right)\tilde{Q}_k},$$

de unde avem

$$\mathbf{\tilde{Q}_{k+1} R_k = \left(H - \mu_k I_n\right) \cdot \tilde{Q}_k}\ .$$

Ţinând seama de faptul că matricea R_k este superior triunghiulară, prima coloană a egalităţii devine (în cazul $\mu_k \notin \lambda(H)$)

$$\tilde{q}_1^{(k+1)} = \frac{1}{r_{11}^{(k)}}\left(H - \mu_k I_n\right) \cdot \tilde{q}_1^{(k)}\ ,$$

unde $\tilde{q}_1^{(k)} = \tilde{Q}_k e_1$. Relaţia arată că prima coloană a matricei ortogonale $\tilde{Q}_k$ se supune iteraţiei metodei puterii şi, în consecinţă, tinde către vectorul propriu unitar asociat valorii proprii dominante a matricei $A - \mu_\infty I_n$ (dacă iniţializarea $\tilde{q}_1^{(0)}$ are componentă nenulă pe această direcţie, ceea ce, generic, este satisfăcut), respectiv

$$\tilde{Q}_\infty = \lim_{k\to\infty} \tilde{Q}_k = \left[\ x_1 \quad Y\ \right].$$

Conform procedurii de deflaţie rezultă că

$$\lim_{k\to\infty} h_{21}^{(k)} = 0,$$

adică matricea H_k tinde către forma Schur în prima coloană punând în evidenţă pe poziţia (1,1), o valoare proprie (reală) a matricei H.

Transpunând relaţia avem $R_k^T \tilde{Q}_{k+1}^T = \tilde{Q}_k^T\left(H^T - \mu_k I_n\right)$, sau $\tilde{Q}_k R_k^T = \left(H^T - \mu_k I_n\right)\tilde{Q}_{k+1}$. Luând egalitatea ultimelor coloane din ultima relaţie obţinem

$$\left(H^T - \mu_k I_n\right) \cdot \tilde{q}_n^{(k+1)} = r_{nn}\tilde{q}_n^{(k)},$$

care arată că ultima coloană a matricei $\tilde{Q}_k$ se supune iteraţiei puterii inverse pentru matricea H^T cu deplasarea curentă de accelerare a convergenţei μ_k. În consecinţă, coloana $\tilde{q}_n^{(k)}$ tinde către un vector propriu al matricei H^T şi anume către cel asociat valorii proprii $\lambda = \mu_\infty = \lim_{k\to\infty} \mu_k$, cea mai apropiată de aproximaţia iniţială. Această convergenţă este cu atât mai rapidă cu cât deplasarea curentă este mai apropiată de o valoare proprie a matricei H. În consecinţă, analog cu (110) şi simultan cu ea, avem

$$\tilde{Q}_\infty = \lim_{k\to\infty} \tilde{Q}_k = \left[\ Z \quad u_n\right],$$

unde u_n este un vector propriu unitar al matricei H^T. Rezultă că matricea limită a şirului **QR** este:

$$H_\infty = \tilde{Q}_\infty^T H \tilde{Q}_\infty = \left(\tilde{Q}_\infty^T H^T \tilde{Q}_\infty\right)^T.$$

Dar, întrucât $\lambda(H^T) = \lambda(H)$ şi $H^T u_n = \lambda_n u_n$, rezultă relaţia

$$\tilde{Q}_\infty^T H^T \tilde{Q}_\infty = \begin{bmatrix} Z^T \\ u_n^T \end{bmatrix} \cdot A^T \cdot \begin{bmatrix} Z & u_n \end{bmatrix} = \begin{bmatrix} Z^T H^T Z & 0 \\ u_n^T H^T Z & \lambda_n \end{bmatrix},$$

care pune în evidenţă o deflaţie în ultima coloană a matricei H^T sau, echivalent, o deflaţie în ultima linie a matricei H. Ţinând seama de faptul că procesul iterativ are loc cu conservarea structurii superior Hessenberg rezultă că în matricea H_k din şirul **QR** avem

$$\lim_{k\to\infty} h_{n,n-1}^{(k)} = 0$$

cu o viteză mare, caracteristică metodei puterii inverse. Din (115) reiese că

$$\lim_{k\to\infty} h_{n,n}^{(k)} = \lambda_n$$

ceeace înseamnă că $h_{nn}^{(k)}$ reprezintă o aproximare din ce în ce mai bună a valorii proprii λ_n. În consecinţă, alegerea deplasării

$$\mu_k = h_{n,n}^{(k)}$$

asigură o convergenţă pătratică (adică excelentă) pentru anularea elementului $h_{n,n-1}$.

Am arătat deci că şirul **QR** asigură o deflaţie iterativă în prima coloană şi, cu alegerea (117) a deplasării, una foarte rapidă în ultima linie. În practică se constată chiar mai mult, şi anume, reducerea simultană, cu viteze diferite, a tuturor modulelor elementelor subdiagonale, fapt care explică remarcabilele proprietăţi de convergenţă ale algoritmului. Cu toate acestea există situaţii "patologice", dificil de caracterizat matematic, în care şirul **QR** nu converge (vezi [5]). Cu această observaţie încheiem justificarea proprietăţii c).

Trunchierea procesului iterativ al şirului **QR** se face progresiv, anulându-se efectiv elementele subdiagonale convergente către zero în momentul satisfacerii unui criteriu de forma

$$\left| h_{p,p-1}^{(k)} \right| \le \varepsilon \cdot \left\| H_k \right\|$$

sau, mai simplu:

$$\left| h_{p,p-1}^{(k)} \right| \le \varepsilon \cdot \left(\left| h_{p-1,p-1}^{(k)} \right| + \left| h_{pp}^{(k)} \right| \right),$$

unde ε este o toleranţă precizată, iar $\|\cdot\|$ o normă matriceală oarecare. Anularea unuia sau mai multor elemente subdiagonale reduce, evident, dimensiunea problemei de deflaţie iterativă, spărgând eventual problema curentă în două sau mai multe probleme de aceeaşi natură.

Întrucât, aşa cum s-a mai menţionat, implementările profesionale ale algoritmului **QR** merg pe o cale specifică (prezentată mai departe), recomandăm cititorului, ca un exerciţiu util, să elaboreze algoritmul de implementare a unui pas **QR** definit în (101) cu alegerea (117) a deplasării. Se recomandă efectuarea calculelor pe loc, în tabloul matricei iniţiale $H \leftarrow H_k$. Pentru factorizarea QR a matricei $H - \mu I_n$ din (101) se recomandă utilizarea unei secvenţe de rotaţii plane. Calculul efectiv al matricei de transformare curente $Q = Q_k$ nu este necesar putându-se utiliza forma sa factorizată

$$Q = P_{12}\, P_{23} \cdots P_{n-1,n}.$$

Astfel schema de calcul propusă pentru suprascrierea matricei $H = H_k$ cu matricea succesor H_{k+1} din şirul **QR** are următoarea formă.

1. $\mu = h_{nn}$
2. $H \leftarrow H - \mu I_n$
3. Pentru $j = 1:n-1$
    1. Se determină rotaţia plană $P_{j,j+1}$ astfel încât
    $(P_{j,j+1}^T H)_{j+1,j} = 0$.
    2. $H \leftarrow P_{j,j+1}^T H$
4. Pentru $j = 1:n-1$
    1. $H \leftarrow H P_{j,j+1}$
5. $H \leftarrow H + \mu I_n$

Algoritmul **QR** cu deplasare explicită funcţionează foarte bine în cazul matricelor cu spectru real, existenţa valorilor proprii complexe crează probleme serioase în evidenţierea asimptotică a blocurilor 2×2 din forma Schur reală.

9.5.2. Algoritmul QR cu deplasare explicită cu paşi dubli

Pentru depăşirea dificultăţilor legate de absenţa convergenţei şirului **QR** creat de utilizarea paşilor simpli atunci când matricea are valori proprii complexe se adoptă aşa numita strategie a *paşilor dubli* **QR** care comprimă într-o singură iteraţie doi paşi simpli **QR** succesivi. Pentru a deduce procesarea aferentă unui pas dublu curent observăm că din ecuaţiile a doi paşi simpli **QR** consecutivi descrişi, conform relaţiilor (101), de

$$\begin{aligned}
&\mathbf{H_k - \mu_k I_n = Q_k R_k}\\
&\mathbf{H_{k+1} = R_k Q_k + \mu_k I_n}\\
&\mathbf{H_{k+1} - \mu_{k+1} I_n = Q_{k+1} R_{k+1}}\\
&\mathbf{H_{k+2} = R_{k+1} Q_{k+1} + \mu_{k+1} I_n}
\end{aligned}$$

rezultă mai întâi $\mathbf{H_{k+1} = Q_k^T H_k Q_k}$, relaţie care, introdusă în ecuaţia a treia din (122), conduce la $\mathbf{Q_k^T H_k Q_k - \mu_{k+1} I_n = Q_{k+1} R_{k+1}}$. Înmulţind această relaţie la stânga cu Q_k şi la dreapta cu R_k obţinem $\mathbf{H_k Q_k R_k - \mu_{k+1} Q_k R_k = Q_k Q_{k+1} R_{k+1} R_k}$, ecuaţie care permite obţinerea lui H_{k+2} direct din H_k pe baza relaţiilor

$$\begin{aligned}
&\mathbf{(H_k - \mu_{k+1} I_n)(H_k - \mu_k I_n) = Q_k Q_{k+1} R_{k+1} R_k}\\
&\mathbf{H_{k+2} = (Q_k Q_{k+1})^T H_k Q_k Q_{k+1}}
\end{aligned}$$

care, la rândul lor, se mai scriu sub forma

$$\begin{aligned}
&\mathbf{M = H_k^2 - s_k H_k + p_k I_n = \bar{Q}_k \bar{R}_k}\\
&\mathbf{H_{k+2} = \bar{Q}_k^T H_k \bar{Q}_k},
\end{aligned}$$

unde $\mathbf{s_k = \mu_k + \mu_{k+1}}$ şi $\mathbf{p_k = \mu_k \mu_{k+1}}$. Evident, matricea $\mathbf{\bar{Q}_k = Q_k Q_{k+1}}$ este ortogonală iar matricea $\mathbf{\bar{R}_k = R_{k+1} R_k}$ este superior triunghiulară.

Ceea ce este important constă în faptul că într-un pas dublu **QR** apare ca deosebit de naturală alegerea celor două deplasări $\mathbf{\mu_k}$, $\mathbf{\mu_{k+1}}$ egale cu cele două valori proprii (posibil complex conjugate) ale blocului 2×2 din dreapta jos a matricei curente $\mathbf{H_k}$, i.e. ale matricei

$$\mathbf{G_k} = \begin{bmatrix} \mathbf{h}^{(k)}_{n-1,n-1} & \mathbf{h}^{(k)}_{n-1,n} \\ \mathbf{h}^{(k)}_{n,n-1} & \mathbf{h}^{(k)}_{n,n} \end{bmatrix}.$$

Aceste deplasări apar în expresia matricei $\mathbf{M}$ numai în forma sumei şi produsului

$$\mathbf{s_k} = \mu_k + \mu_{k+1} = \mathbf{h}^{(k)}_{n-1,n-1} + \mathbf{h}^{(k)}_{n,n}$$
$$\mathbf{p_k} = \mu_k\mu_{k+1} = \mathbf{h}^{(k)}_{n-1,n-1}\mathbf{h}^{(k)}_{n,n} - \mathbf{h}^{(k)}_{n,n-1}\mathbf{h}^{(k)}_{n-1,n},$$

care sunt *reale,* inclusiv în cazul în care blocul $\mathbf{G_k}$ are valori proprii *complexe.* În consecinţă, matricea $\mathbf{M}$ este reală, produsul $\bar{\mathbf{Q}}_\mathbf{k}\bar{\mathbf{R}}_\mathbf{k}$ reprezintă factorizarea **QR** a matricei $\mathbf{M}$ cu $\bar{\mathbf{Q}}_\mathbf{k}$ şi $\bar{\mathbf{R}}_\mathbf{k}$ reale şi deci toate calculele aferente unui pas dublu **QR** pot fi efectuate în exclusivitate în aritmetică reală. Mai mult, matricea $\mathbf{H_{k+2}}$ din (124) fiind aceeaşi cu matricea H_{k+2} din (122) pasul dublu **QR** conservă şi el structura superior Hessenberg.
În concluzie, algoritmul **QR** cu deplasare explicită cu pas dublu are la bază următoarea schemă de calcul

1. Pentru $k = 1, 2, \ldots$
    1. $s = h_{n-1,n-1} + h_{nn}$
    2. $p = h_{n-1,n-1}h_{nn} - h_{n,n-1}h_{n-1,n}$
    3. Se calculează $M = H^2 - sH + pI_n$
    4. Se factorizează $M = QR$
    5. $H \leftarrow Q^T HQ$

care, în forma implementabilă, se completează cu procedura de trunchiere constând în anularea efectivă a elementelor subdiagonale ce satisfac o condiţie de tipul (118) sau (119) şi monitorizarea elementelor subdiagonale nule pentru detectarea terminării (testul de terminare ar putea fi inexistenţa a două elemente subdiagonale nenule consecutive).
Schema asigură o convergenţă rapidă a matricei *H* către o structură cvasisuperior triunghiulară cu blocuri diagonale de dimensiune cel mult 2×2 care, fie că este chiar forma Schur reală urmărită, fie că permite calculul direct şi imediat al acestei forme prin triangularizarea, cu ajutorul unor transformări de asemănare ortogonală, a blocurilor 2×2 cu valori proprii reale.
Motivul pentru care schema (127) nu constituie suportul implementărilor profesionale ale algoritmului **QR** este legat de slaba sa eficienţă. Într-adevăr este uşor de văzut că un pas simplu **QR** are o complexitate $O(n^2)$, datorată în principal structurii Hessenberg cu care operează. Doi paşi simpli **QR** vor avea aceeaşi complexitate. În schimb un pas dublu, în varianta (127) necesită calculul explicit al matricei *M,* crescând complexitatea la $O(n^3)$. Acest spor de efort de calcul nu are o fundamentare teoretică ci se datorează mai curând inabilităţii noastre de a organiza calculele în cadrul unui pas dublu **QR** în aşa fel încât să păstrăm complexitatea $O(n^2)$. Aşa cum se va vedea în secţiunea următoare, abilitatea necesară pentru evitarea creşterii complexităţii este departe de a fi banală.

9.6. Algoritmul QR cu deplasare implicită

Baza teoretică a elaborării unui algoritm eficient pentru un pas dublu **QR** este dată de teorema 6 conform căreia reducerea unei matrice la forma superior Hessenberg prin transformări ortogonale de asemănare are matricea ortogonală de transformare determinată, în esenţă, de prima sa coloană.

Concret, putem reduce complexitatea unui pas dublu **QR**, la $\mathbf{O(n^2)}$ procedând în modul următor:

1. Se calculează prima coloană a matricei M, i.e.
   $m_1 = M(:,1)$.
2. Se determină reflectorul elementar U_1 astfel ca
   $(U_1 m_1)(2:n) = 0$.
   adică
   $U_1 m_1 = -\text{sign}(m_{11}) \| m_1 \| e_1 = \rho e_1$
3. Se calculează
   $N = U_1^T H_k U_1$
4. Se aplică matricei *N* algoritmul **HQ** de aducere la forma superior Hessenberg:
   $\tilde{H} = Q^T N Q$.

Procedura de mai sus calculează forma superior Hessenberg

$$\tilde{\mathbf{H}} = (\mathbf{U_1 Q})^{\mathbf{T}} \mathbf{H_k U_1 Q} = \tilde{\mathbf{Q}}^{\mathbf{T}} \mathbf{H_k} \tilde{\mathbf{Q}}$$

cu matricea de transformare $\tilde{Q}$ având proprietatea

$$\tilde{\mathbf{Q}}(:, 1) = \mathbf{U_1 Q e_1} = \mathbf{U_1 e_1} = \frac{1}{\rho} \mathbf{m_1},$$

penultima egalitate datorându-se faptului că $Q = \text{diag}(1, \bar{Q})$ (vezi algoritmul **HQ**), iar ultima rezultă din.

Pe de altă parte, din prima relaţie (124), datorită formei superior triunghiulare a matricei $\bar{R}_k$, rezultă

$$\bar{\mathbf{Q}}_{\mathbf{k}} \mathbf{e_1} = \frac{1}{\mathbf{r}_{11}^{(k)}} \cdot \mathbf{m_1}.$$

În concluzie, matricele ortogonale de transformare $\tilde{Q}$ şi $\bar{Q}$ au aceeaşi primă coloană şi deci, conform teoremei 9.6, dacă matricele $\tilde{H}$ din (113) şi H_{k+2} din (124) sunt ambele *ireductibile* atunci ele sunt, esenţial, aceleaşi. Pentru a ne convinge de complexitatea $O(n^2)$ a procedurii vom explicita relaţiile (128) - (132). Mai întâi vectorul m_1 are expresia

$$\mathbf{m_1} = \begin{bmatrix} \mathbf{h_{11}^2 + h_{12} h_{21} - s h_{11} + p} \\ \mathbf{h_{21}(h_{11} + h_{22} - s)} \\ \mathbf{h_{21} h_{32}} \\ \cdots \\ \mathbf{0} \end{bmatrix}$$

în care am omis scrierea indicelui *k* asociat matricei H_k. Vectorul m_1 având numai trei componente nenule, calculul său necesită un număr de operaţii independent de n, deci are o o complexitate $O(1)$. În plus, reflectorul U_1 din (129) are structura

$$\mathbf{U_1} = \mathbf{diag}(\bar{\mathbf{U}}_1, \mathbf{I_{n-3}})$$

cu $U_1 \in R^{3\times 3}$ a cărui determinare are, de asemenea, o complexitate $O(1)$.

În calculul matricei N sunt afectate numai primele trei linii şi primele trei coloane ale matricei $H_k = H$. Într-adevăr, dacă partiţionăm

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix}, \qquad \mathbf{H}_{11} \in \mathbf{R}^{3\times 3}$$

atunci

$$\mathbf{N} = \mathbf{U}_1^{\mathrm{T}}\mathbf{H}\mathbf{U}_1 = \begin{bmatrix} \bar{\mathbf{U}}_1^{\mathrm{T}}\mathbf{H}_{11}\bar{\mathbf{U}}_1 & \bar{\mathbf{U}}_1^{\mathrm{T}}\mathbf{H}_{12} \\ \mathbf{H}_{21}\bar{\mathbf{U}}_1 & \mathbf{H}_{22} \end{bmatrix}.$$

În plus, blocul H_{21} are structura

$$\mathbf{H}_{21} = \begin{bmatrix} 0 & 0 & \mathbf{h}_{43} \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix}$$

şi, prin urmare,

$$\mathbf{H}_{21}\bar{\mathbf{U}}_1 = \begin{bmatrix} \begin{bmatrix} 0 & 0 & \mathbf{h}_{43} \end{bmatrix}\bar{\mathbf{U}}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \times & \times & \times \\ 0 & & \end{bmatrix} \begin{matrix} 1 \\ \mathbf{n-4} \end{matrix}$$

adică transformarea (131) alterează structura Hessenberg a matricei H numai în poziţiile (3,1), (4,1) şi (4,2). Prin urmare, aplicarea algoritmului **HQ** are ca obiectiv anularea, prin transformări ortogonale de asemănare, a celor trei elemente nenule ale matricei N alterante ale structurii superior Hessenberg. Pe de altă parte, această particularitate structurală (numai trei elemente în afara structurii Hessenberg în matricea iniţială) obligă la adaptarea algoritmului **HQ** astfel încât să se obţină o eficacitate maximă. Schema de calcul a acestei particularizări a algoritmului **HQ** este următoarea.

1. Pentru $j = 1:n-2$
   1. Se determină reflectorul elementar U_{j+1} astfel încât
      $(U_{j+1}N)(i,j) = 0$ pentru $i = j+2:j+3$ şi $i \le n$.
   2. $N \leftarrow U_{j+1}N$
   3. $N \leftarrow NU_{j+1}N$

cu precizarea că diferenţa faţă de algoritmul **HQ** standard constă în faptul că de data aceasta, la fiecare pas se anulează numai două elemente (la ultimul, unul singur), celelalte elemente fiind nule din start.

SNu este greu de constatat că reflectorii elementari utilizaţi în schema (142) au structura

$$\mathbf{U}_{j+1} = \mathbf{diag}\left(\mathbf{I}_j, \bar{\mathbf{U}}_{j+1}, \mathbf{I}_{n-j-3}\right), \qquad \mathbf{j = 1 : n-3}$$
$$\mathbf{U}_{n-1} = \mathbf{diag}\left(\mathbf{I}_{n-2}, \bar{\mathbf{U}}_{n-1}\right),$$

unde $\bar{\mathbf{U}}_{j+1} \in \mathbf{R}^{3\times 3}$, $\bar{\mathbf{U}}_{n-1} \in \mathbf{R}^{2\times 2}$ sunt reflectorii elementari de ordinul 3, respectiv 2. Pe această bază rezultă imediat că numărul asimptotic de operaţii aritmetice necesar pentru execuţia schemei (142) este $\mathbf{N}_{op} \cong \mathbf{10n}^2$, la care se adaugă $n-2$ extrageri de radicali. Dacă se doreşte acumularea transformărilor, respectiv calculul matricei Q din relaţia (132)

$$\mathbf{Q} = \mathbf{U}_1\mathbf{U}_2 \ldots \mathbf{U}_{n-1},$$

sunt necesare $\mathbf{N}'_{op} \cong \mathbf{10n}^2$ operaţii suplimentare.

Exemplificăm, pentru clarificare, evoluţia structurală datorată execuţiei schemei (142), pentru cazul particular $n=5$.

$$N=\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \otimes & \times & \times & \times & \times \\ \otimes & \otimes & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix} \Rightarrow$$

$$N \leftarrow U_2 N = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \varnothing & \times & \times & \times & \times \\ \varnothing & \otimes & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix} \Rightarrow N \leftarrow NU_2 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \otimes & \times & \times & \times \\ 0 & \otimes & \otimes & \times & \times \end{bmatrix} \Rightarrow N \leftarrow U_3 N = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \varnothing & \times & \times & \times \\ 0 & \varnothing & \otimes & \times & \times \end{bmatrix} \Rightarrow$$

$$N \leftarrow NU_3 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \otimes & \times & \times \end{bmatrix} \Rightarrow N \leftarrow U_4 N = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \varnothing & \times & \times \\ 0 & 0 & \varnothing & \times & \times \end{bmatrix} \Rightarrow N \leftarrow NU_4 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}.$$

În diagramele de mai sus simbolul $\otimes$ marchează elementele care alterează structura Hessenberg, simbolul $\varnothing$ marchează elementele anulate la pasul curent, iar încadrările marchează elementele afectate de multiplicarea matriceală curentă.

În concluzie, complexitatea pasului dublu **QR** cu deplasare implicită, definit de (128)-(132) şi prezentat prima dată de J. Francis [9] şi V.N. Kublanovskaia [20] în 1961 este $O(n^2)$, această formă reprezentând baza implementărilor performante ale algoritmului **QR** de reducere iterativă a unei matrice la forma Schur reală.

În scrierea efectivă a algoritmului de implementare a unui pas dublu **QR** cu deplasare implicită, din motive de concizie, vom utiliza funcţiile **refmat** şi **matref** de premultiplicare, respectiv de postmultiplicare cu un reflector elementar (vezi algoritmii 9.3 şi 9.4), al căror apel ţine seama de structura specială din acest caz. De asemenea, nu se realizează calculul efectiv al matricei de transformare dar vor fi furnizate elementele definitorii ale reflectorilor utilizaţi. Concret, pentru calculul matricei Q este suficientă cunoaşterea reflectorilor de ordinul 3 şi 2

$$\bar{U}_j = I_3 - \frac{u_j u_j^T}{\beta_j}, \qquad u_j \in R^3, \qquad j = 1 : n-2$$

$$\bar{U}_{n-1} = I_2 - \frac{u_{n-1} u_{n-1}^T}{\beta_{n-1}}, \qquad u_{n-1} \in R^2,$$

ale căror elemente definitorii vor fi memorate într-o matrice $V \in R^{3\times(n-1)}$ astfel încât $V(:,j) = u_j, \quad j = 1 : n-2$, $V(:,n-1) = \begin{bmatrix} u_{n-1} \\ 0 \end{bmatrix}$ şi un vector $b \in R^{n-1}$ al scalarilor $\beta_j, \quad j = 1 : n-1.$

Cu aceste precizări, prezentăm mai jos algoritmul de implementare a unui pas dublu **QR** cu deplasare implicită, cu comentarii care să permită identificarea simplă a consideraţiilor teoretice de mai sus.

Algoritmul 6

$[H,V,b] = \mathbf{QR2}(n,H)$

/* Implementarea unui pas dublu **QR** cu deplasare implicită.
/* *Intrări*: n = ordinul matricei Hessenberg ireductibile de intrare;
/* H = matricea reală superior Hessenberg ireductibilă
/* *Ieşiri*: $H \leftarrow Q^T HQ$ - suprascrierea matricei de intrare cu matricea
/* succesor din şirul **QR** cu paşi dubli;
/* $V \in R^{3\times(n-1)}$ matrice cu elementele vectorilor de ordinul 3 ce
/* definesc reflectorii utilizaţi;
/* $b \in R^{n-1}$ vectorul ce conţine scalarii β_j .

1./* Calculul sumei şi produsului deplasărilor (126)
1. $s = H(n-1,n-1) + H(n,n)$
2. $p = H(n-1,n-1)\cdot H(n,n) - H(n-1,n)\cdot H(n,n-1)$

2. /* Calculul elementelor nenule din m_1 cu (136)
1. $m_1(1) = (H(1,1))^2 + H(1,2)\cdot H(2,1) - sH(1,1) + p$
2. $m_1(2) = H(2,1)\cdot\big(H(1,1) + H(2,2) - s\big)$
3. $m_1(3) = H(2,1)\cdot H(3,2)$

3. /* Calculul reflectorului elementar U_1 din (137)
1. $\sigma = \text{sign}(m_1(1))\cdot\|m_1\|_2$
2. $u(1) = m_1(1) + \sigma$
3. $u(2:3) = m_1(2:3)$
4. $V(:,1) = u$
5. $b(1) \leftarrow \beta_1 = u(1)\cdot\sigma$

4. /*Calculul matricei $H \leftarrow N = U_1^T H U_1$ din (131)
1. $[H(1:3,:)] = \mathbf{refmat}(u, b(1), H(1:3,:))$
2. $[H(1:4,1:3)] = \mathbf{matref}(H(1:4,1:3), u, b(1))$

5. /*Readucerea lui H la forma superior Hessenberg
1. Pentru $j = 1:n-3$
1. /* Calculul reflectorului U_{j+1}
1. $\sigma = \text{sign}(H(j+1,j))\cdot\|H(j+1:j+3,j)\|_2$
2. $u(1) = H(j+1,j) + \sigma$
3. $u(2:3) = H(j+2:j+3,j)$
4. $V(:,j+1) = u$
5. $b(j+1) \leftarrow \beta_{j+1} = u(1)\cdot\sigma$
2. /* Calcul $H \leftarrow U_{j+1}^T H U_{j+1}$
1. $H(j+1,j) = -\sigma$
2. $H(j+2:j+3,j) = 0$
3. $[H(j+1:j+3,j+1:n)] =$
$\mathbf{refmat}(u, b(j+1), H(j+1:j+3,j+1:n))$
4. $r = \min(j+4,n)$
5. $[H(1:r,j+1:j+3)] =$

matref $(H(1:r, j+1: j+3), u, b(j+1))$

2. /* Calculul reflectorului U_{n-1}
   1. $\sigma = \text{sign}(H(n-1,n-2)) \cdot \|H(n-1:n,n-2)\|_2$
   2. $u(1) = H(n-1,n-2) + \sigma$
   3. $u(2) = H(n,n-2)$
   4. $u(3) = 0$
   5. $V(:,n-1) = u$
   6. $b(n-1) \leftarrow \beta_{n-1} = u(1) \cdot \sigma$
3. /* Calculul $H \leftarrow U_{n-1}^T H U_{n-1}$
   1. $H(n-1,n-2) = -\sigma$
   2. $H(n,n-2) = 0$
   3. $[H(n-1:n,n-1:n)] =$
      refmat $(u(1:2), b(n-1), H(n-1:n,n-1:n))$
   4. $[H(1:n,n-1:n)] =$
      matref $(H(1:n,n-1:n), u(1:2), b(n-1))$

Algoritmul 5 de reducere la formă superior Hessenberg şi algoritmul 9.6 de iterare în construcţia şirului **QR** reprezintă baza programelor profesionale de calcul a valorilor proprii (cum ar fi cele din sistemele de programe ştiinţifice EISPACK, LAPACK, MATLAB, MATEMATICA).

Pentru a descrie întregul proces de calcul, cunoscut sub numele de algoritmul **QR**, trebuie să controlăm procesul iterativ prin anularea efectivă a elementelor subdiagonale la scăderea lor sub un prag precizat, monitorizarea elementelor subdiagonale nule şi, pe această cale, creşterea eficienţei prin reducerea progresivă a dimensiunii.

Pentru implementarea acestor aspecte vom utiliza faptul că deflaţia iterativă are loc, cel mai rapid, în raport cu colţul din dreapta jos al matricei H şi, cu o viteză de convergenţă ceva mai redusă, în raport cu colţul din stânga sus. În consecinţă, se impune, în monitorizarea zerourilor subdiagonale, partiţia, la fiecare iteraţie, a matricei curente H din şirul **QR** în forma

$$\mathbf{H} = \begin{bmatrix} \mathbf{H_{11}} & \mathbf{H_{12}} & \mathbf{H_{13}} \\ \mathbf{0} & \mathbf{H_{22}} & \mathbf{H_{23}} \\ \mathbf{0} & \mathbf{0} & \mathbf{H_{33}} \end{bmatrix}, \qquad \mathbf{H_{11}} \in \mathbf{R^{p \times p}}, \quad \mathbf{H_{33}} \in \mathbf{R^{q \times q}},$$

astfel încât blocul diagonal H_{33} să aibă o structură cvasisuperior-triunghiulară (adică o structură cu blocuri diagonale de dimensiuni cel mult 2×2) de dimensiune maximă, iar blocul H_{22} să fie superior Hessenberg *ireductibil*, de asemenea de dimensiune maximă (ceea ce implică *p* minim).

Cu această partiţie iteraţia curentă se materializează în aplicarea pasului dublu **QR** submatricei H_{22}

$$\mathbf{H_{22}} \leftarrow \mathbf{Q_{22}^T H_{22} Q_{22}},$$

ceea ce derivă din aplicarea transformării ortogonale de asemănare globale

$$\mathbf{H} \leftarrow \mathbf{diag}\left(\mathbf{I_p}, \mathbf{Q_{22}^T}, \mathbf{I_q}\right) \cdot \mathbf{H} \cdot \mathbf{diag}\left(\mathbf{I_p}, \mathbf{Q_{22}}, \mathbf{I_q}\right),$$

cu următoarele efecte asupra celorlalte blocuri ale partiţiei

$$\mathbf{H_{12}} \leftarrow \mathbf{H_{12}} \cdot \mathbf{Q_{22}}, \qquad \mathbf{H_{23}} \leftarrow \mathbf{Q_{22}^T} \cdot \mathbf{H_{23}}.$$

Algoritmul îşi epuizează funcţia în momentul în care întreaga matrice H devine cvasisuperior triunghiulară, respectiv parametrul q atinge dimensiunea n a matricei.

Rezultatul procesului iterativ, cu monitorizarea structurală de mai sus, este o structură a matricei H având blocuri diagonale de dimensiune 1×1 sau 2×2. Pentru a obţine forma Schur reală, care este obiectivul declarat al algoritmului **QR**, trebuie triangularizate blocurile 2×2 cu valori proprii reale. Pentru a vedea cum se poate realiza această operaţie, vom proceda la deflaţia (vezi 9.2) unei matrice $B \in R^{2\times 2}$ cu

$$b_{21} \neq 0$$

şi satisfăcând condiţia

$$\Delta = (b_{11} - b_{22})^2 + 4b_{12}b_{21} \geq 0,$$

de existenţă a valorilor proprii reale

$$\lambda_1 = \frac{b_{11} + b_{22} - \sqrt{\Delta}}{2}, \qquad \lambda_2 = \frac{b_{11} + b_{22} + \sqrt{\Delta}}{2}.$$

Determinând un vector propriu unitar asociat lui λ_1 şi completându-l până la o matrice ortogonală, este uşor de văzut că transformarea ortogonală de asemănare definită de matricea de rotaţie

$$P = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

cu

$$c = \frac{\lambda_1 - b_{22}}{\sqrt{b_{21}^2 + (\lambda_1 - b_{22})^2}}, \qquad s = \frac{-b_{21}}{\sqrt{b_{21}^2 + (\lambda_1 - b_{22})^2}}$$

conduce la realizarea triangularizării urmărite

$$C = P^T B P = \begin{bmatrix} \lambda_1 & b_{12} - b_{21} \\ 0 & \lambda_2 \end{bmatrix}.$$

La nivelul matricei cvasisuperior triunghiulare **H**, dacă blocul diagonal 2×2 cu valori proprii reale se află pe liniile şi coloanele **k, k + 1**, atunci pentru triangularizarea lui se aplică o transformare de forma

$$H \leftarrow \text{diag}\,(I_{k-1}, P^T, I_{n-k-1}) \cdot H \cdot \text{diag}\,(I_{k-1}, P, I_{n-k-1})$$

cu P calculat ca mai sus.

La fel ca şi în cazul reflectorilor, pentru descrierea algoritmică a transformării (156) vom introduce două proceduri. Prima procedură, numită **rotmat**, va realiza premultiplicarea, cu suprascriere, $A \leftarrow PA$ a unei matrice $A \in R^{2\times p}$ cu o rotaţie $P \in R^{2\times 2}$ având expresia (153) şi dată de parametri scalari c şi s.

Algoritmul 7.

$[A] = \textbf{rotmat}\,(c, s, A)$

/* Premultiplicarea unei matrice cu o rotaţie.

/* *Intrări*: c, s = parametri definitori ai rotaţiei;

/* $A \in R^{2\times p}$ = matricea dată.

/* *Ieşiri*: $A \leftarrow PA$.

1. Pentru $j = 1 : p$
   1. $\alpha = cA(1, j) + sA(2, j)$
   2. $A(2, j) \leftarrow -sA(1, j) + cA(2, j)$

3. $A(1, j) = \alpha$

Evident, un apel $[A] = \textbf{rotmat}(c, -s, A)$ va realiza suprascrierea $A \leftarrow P^T A$.

A doua procedură, numită **matrot**, va realiza postmultiplicarea, cu suprascriere, $A \leftarrow AP$ a unei matrice $A \in R^{m\times 2}$ cu o rotaţie $P \in R^{2\times 2}$ având expresia (153) şi dată de parametri scalari c şi s.

Algoritmu 8.

$[A] = \textbf{matrot}(A, c, s)$
/* Postmultiplicarea unei matrice cu o rotaţie.
/* *Intrări*: c, s = parametri definitori ai rotaţiei;
/* $A \in R^{m\times 2}$ = matricea dată.
/* *Ieşiri*: $A \leftarrow AP$.

1. Pentru $i = 1 : m$
   1. $\alpha = A(i,1) \cdot c - A(i,2) \cdot s$
   2. $A(i,2) \leftarrow A(i,1) \cdot s + A(i,2) \cdot c$
   3. $A(i,1) = \alpha.$

Suntem în măsură acum să prezentăm, în toată plenitudinea sa, algoritmul **QR**. Toate matricele şirului **QR** vor suprascrie matricea iniţială dată. În mod facultativ, pe baza variabilei de opţiune opt algoritmul calculează matricea de transformare ortogonală cumulată Q. Pentru concizie, produsele matriceale nu au fost explicitate, dar se atrage atenţia că ele trebuie făcute cât mai economicos, exploatând particularităţile structurale ale matricelor factor.

Algoritmul 9.

$[A, Q] = \textbf{QR}(n, A, \varepsilon, \text{opt})$
/* Algoritmul **QR** de reducere a unei matrice reale la forma Schur
/* reală prin transformări ortogonale de asemănare.
/* *Intrări*: n = ordinul matricei iniţiale A;
/* A = matricea dată;
/* ε = toleranţa pentru detectarea elementelor de modul neglijabil;
/* opt = opţiunea de calcul a matricei de transformare ('Y' sau 'N').
/* *Ieşiri*: $A \leftarrow S = Q^T AQ$ - matricea redusă la forma Schur reală;
/* Q = matricea ortogonală de transformare.

1. /* Iniţializarea matricei Q
   1. $Q = I_n$
2. /* Reducerea matricei A la formă superior Hessenberg
   1. $[A, Q, V, b] =$ **HQ** (n, A, opt)
3. /* Reducerea matricei superior Hessenberg la formă cvasisuperior triunghiulară
   1. $p = 0$
   2. $q = 0$
   3. Cât timp $q < n$
      1. /* Anularea elementelor subdiagonale neglijabile
         1. Pentru $i = p+1 : n-q-1$
            1. Dacă $|A(i+1,i)| \le \varepsilon \cdot (|A(i,i)| + |A(i+1,i)|)$ atunci

1. $A(i+1,i)=0$

2./* Stabilirea valorii maxime pentru q şi a valorii minime pentru p astfel încât blocul superior Hessenberg A_{22} din (122) să fie ireductibil de dimensiune maximă.

1. test = TRUE
2. Cât timp test = TRUE şi $q<n$
   1. Dacă $A(n-q,n-q-1)=0$ atunci
      1. $q \leftarrow q+1$

      altfel
      1. Dacă $q<n-1$ şi $A(n-q-1,n-q-2)=0$ atunci
         1. $q \leftarrow q+2$

         altfel
         1. test = FALSE

3. Dacă $q<n-2$ atunci
   1. $p=n-q-1$
   2. Cât timp $A(p,p-1)\neq 0$
      1. $p \leftarrow p-1$

   altfel
   1. /* Matricea A are formă cvasisuperior triunghiulară.
      /*Triangularizarea blocurilor diagonale 2×2 cu valori proprii reale
      1. $i=1$
      2. Cât timp $i<n$
         1. Dacă $A(i+1,i)\neq 0$ atunci
            1. $\delta=\left(A(i,i)-A(i+1,i+1)\right)^2+4A(i+1,i)A(i,i+1)$
            2. Dacă $\delta\geq 0$ atunci
               1. $\sigma=A(i,i)+A(i+1,i+1)$
               2. $\delta \leftarrow \sqrt{\delta}$
               3. $\lambda_1=\dfrac{\sigma-\delta}{2}$
               4. $\lambda_2=\dfrac{\sigma+\delta}{2}$
               5. $\rho=\sqrt{\left(A(i+1,i)\right)^2+\left(\lambda_1-A(i+1,i+1)\right)^2}$
               6. $c=\dfrac{\lambda_1-A(i+1,i+1)}{\rho}$, $\quad s=\dfrac{-A(i+1,i)}{\rho}$
               7. $A(i,i)=\lambda_1, \quad A(i+1,i+1)=\lambda_2$
               8. $A(i,i+1)\leftarrow A(i,i+1)-A(i+1,i)$
               9. $A(i+1,i)=0$
               10. Dacă $i>1$ atunci
                  1. $A(1:i-1,i:i+1)=$ **matrot** $(A(1:i-1,i:i+1),c,s)$
               11. Dacă $i+1<n$ atunci
                  1. $A(i:i+1,i+1:n)=$ **rotmat** $(c,-s,A(i:i+1,i+1:n))$

12. Dacă opt ='Y' atunci
1. $Q(:,i:i+1) = \mathbf{matrot}(Q(:,i:i+1),c,s)$
13. $i \leftarrow i+2$
altfel
1. $i \leftarrow i+1$
3. /* Terminarea algoritmului
1. RETURN
4. /* Aplicarea unui pas dublu **QR** cu deplasare implicită blocului diagonal A_{22}
1. $[A(p+1:n-q,p+1:n-q),V,b] =$
$\mathbf{QR2}(n-p-q,A(p+1:n-q,p+1:n-q))$
2. Pentru $i = 1:n-p-q-2$
1. $v = V(1:3,i)$
2. $m = p+i$
3. Dacă $p > 0$ atunci
1. $A(1:p,m:m+2) = \mathbf{matref}(A(1:p,m:m+2),v,b(i))$
4. Dacă $q > 0$ atunci
1. $A(m:m+2,n-q+1:n) =$
$\mathbf{refmat}(v,b(i),A(m:m+2,n-q+1:n))$
5. Dacă opt = 'Y' atunci
1. $Q(:,m:m+2) = \mathbf{matref}(Q(:,m:m+2),v,b(i))$
3./* Idem pentru $i = n-p-q-1$
1. $m = n-q-1$
2. $v = V(1:2,n-p-q-1)$
3. Dacă $p > 0$ atunci
1. $A(1:p,m:m+1) =$
$\mathbf{matref}(A(1:p,m:m+1),v,b(i))$
4. Dacă $q > 0$ atunci
1. $A(m:m+1,m:m+1) =$
$\mathbf{refmat}(v,b(i),A(m:m+1,m:m+1))$
5. Dacă opt = 'Y' atunci
1. $Q(:,m:m+1) =$
$\mathbf{matref}(Q(:,m:m+1),v,b(n-p-q-1))$

Evaluarea complexităţii algoritmului **QR** nu poate fi făcută decât în mod empiric, datorită prezenţei procesului iterativ. Statisticile şi numeroasele experimente numerice arată că, în medie, sunt suficiente două iteraţii **QR** pentru anularea convenţională a unui element subdiagonal. În această ipoteză, pentru calculul formei Schur reale sunt necesare $N'_{op} \cong 10n^3$ operaţii aritmetice, iar pentru calculul matricei de transformare, alte $N''_{op} \cong 15n^3$ operaţii, ceea ce încadrează algoritmul **QR** în categoria algoritmilor cu complexitate $O(n^3)$.

Observaţii. Aşa cum s-a prezentat în secţiunea 9.2 forma Schur reală a unei matrice este determinată până la o ordonare a blocurilor diagonale. Forma Schur reală calculată de algoritmul 9.9 nu satisface nici un criteriu, a priori fixat, privitor la ordinea blocurilor diagonale.
În momentul în care o formă Schur a unei matrice date satisface un criteriu precizat de ordine a blocurilor diagonale, este numită în mod uzual, *formă Schur reală ordonată*. Pentru a putea

ordona după dorinţă blocurile diagonale este suficient să dispunem de un mijloc de permutare a două elemente (blocuri) diagonale vecine. În cazul nostru, un astfel de instrument trebuie să fie o transformare de asemănare ortogonală. Propunem cititorului ca exerciţiu, găsirea transformărilor ortogonale de asemănare care permută două blocuri diagonale vecine în toate situaţiile structurale posibile, respectiv: a) două blocuri scalare; b) două blocuri 2×2; c) un bloc 1×1 cu un bloc 2×2 şi d) un bloc 2×2 cu un bloc 1×1. Dacă apar dificultăţi consultaţi, de exemplu, lucrarea [5].

Ordonarea formei Schur (reale) permite evidenţierea unor subspaţii invariante asociate unui grup de valori proprii şi calculul unor baze ortogonale pentru acestea. Într-adevăr, dacă $\mathbf{S = Q^T A Q}$ este forma Schur reală ordonată astfel încât în partiţia

$$\mathbf{S} = \begin{bmatrix} \mathbf{S_{11}} & \mathbf{S_{12}} \\ \mathbf{0} & \mathbf{S_{22}} \end{bmatrix}, \qquad \mathbf{S_{11}} \in \mathbf{R^{p\times p}},$$

blocul S_{11} conţine valorile proprii de interes, atunci este uşor de văzut că subspaţiul liniar $\mathbf{S_1 = Im\ Q_1}$, unde $\mathbf{Q_1 = Q(:, 1:p)}$, este un subspaţiu $\mathbf{A}$-invariant, asociat grupului de valori proprii $\boldsymbol{\lambda}(\mathbf{S_{11}}) \subset \boldsymbol{\lambda}(\mathbf{A})$, iar coloanele matricei $\mathbf{Q_1}$ formează o bază ortogonală a acestui subspaţiu. De aceea, în [29] este folosit pe scară largă conceptul de *sistem propriu* al unei matrice, definit de o pereche (grup de valori proprii, subspaţiu invariant asociat), ca o generalizare naturală a conceptului (valoare proprie, vector (direcţie) proprie asociat(ă)).

Calculul vectorilor proprii

Algoritmul 9 calculează forma Schur reală $\mathbf{S = Q^T A Q}$ a unei matrice $\mathbf{A} \in \mathbf{R^{n\times n}}$ date. Conform cu cele prezentate în secţiunea 9.1, dacă $\mathbf{y_k}$ este un vector propriu al matricei *S*, asociat valorii proprii $\boldsymbol{\lambda_k} \in \boldsymbol{\lambda}(\mathbf{S}) = \boldsymbol{\lambda}(\mathbf{A})$, respectiv

$$\mathbf{S y_k = \lambda_k y_k},$$

atunci

$$\mathbf{x_k = Q y_k}$$

este un vector propriu al matricei A asociat aceleaşi valori proprii λ_k. În consecinţă, dacă algoritmul $\mathbf{QR}$ calculează matricea de transformare Q, calculul vectorilor proprii asociaţi valorilor proprii reale se reduce la rezolvarea sistemelor liniare (166) şi multiplicărilor (167). Dacă λ_k este o valoare proprie distinctă, rezolvarea sistemului omogen (166) nu ridică probleme. În acest caz sistemul (166) are forma

$$\begin{bmatrix} \mathbf{S_{11} - \lambda_k I_{k-1}} & \mathbf{S_{12}} & \mathbf{S_{13}} \\ \mathbf{0} & \mathbf{0} & \mathbf{S_{23}} \\ \mathbf{0} & \mathbf{0} & \mathbf{S_{33} - \lambda_k I_{n-k}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{y'_k} \\ \mathbf{y''_k} \\ \mathbf{y'''_k} \end{bmatrix} = \mathbf{0}$$

cu $\mathbf{y''_k = y_k(k)}$ şi cu blocurile $\mathbf{S_{11} - \lambda_k I_{k-1}}$ şi $\mathbf{S_{33} - \lambda_k I_{n-k}}$ nesingulare. De aici rezultă $\mathbf{y'''_k = 0}$ şi deci vectorul propriu are structura

$$\mathbf{y_k} = \boldsymbol{\alpha} \begin{bmatrix} -\left(\mathbf{S_{11} - \lambda_k I_{k-1}}\right)^{-1}\mathbf{S_{12}} \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix} \begin{matrix} \}\mathbf{k-1} \\ \}\mathbf{1} \\ \}\mathbf{n-k} \end{matrix}$$

cu $\boldsymbol{\alpha} \in \mathbf{R} \setminus \{\mathbf{0}\}$ arbitrar. În concluzie, calculul lui $\mathbf{y_k}$ se reduce la rezolvarea unui sistem liniar cvasitriunghiular de ordinul $\mathbf{k-1}$. În cazul valorilor proprii reale multiple (identitatea a două valori proprii calculate este practic imposibilă datorită aritmeticii aproximative) apar

dificultăţi importante datorită proastei condiţionări numerice a matricelor $\mathbf{S_{11} - \lambda_k I_{k-1}}$ şi/sau $\mathbf{S_{33} - \lambda_k I_{n-k}}$ şi de aceea se recomandă utilizarea unor procedee speciale.
Pentru valori proprii complex conjugate, distincte,

$$\lambda_{k,k+1} = \alpha_k \pm i\beta_k = \lambda(S_{22}) \subset \lambda(S) = \lambda(A)$$

vectorii proprii asociaţi matricei S pot avea forma

$$y_{k,k+1} = u_k \pm iv_k, \qquad u_k, v_k \in R^n .$$

Vectorii reali u_k, v_k se pot calcula prin rezolvarea sistemului omogen singular $2n$ dimensional

$$\begin{cases} Su_k = \alpha_k u_k - \beta_k v_k \\ Sv_k = \beta_k u_k + \alpha_k v_k \end{cases}$$

Cum $\beta_k \neq 0$

$$\begin{cases} u_k = \dfrac{1}{\beta_k}(S - \alpha_k I_n)v_k \\ (S^2 - 2\alpha_k S + (\alpha_k^2 + \beta_k^2)I_n)v_k = 0 . \end{cases}$$

Pe baza partiţiei

$$S = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ 0 & S_{22} & S_{23} \\ 0 & 0 & S_{33} \end{bmatrix}, \qquad S_{11} \in R^{(k-1)\times(k-1)},$$

$$S_{22} \in R^{2\times 2},\ S_{33} \in R^{(n-k-1)\times(n-k-1)}$$

acceptate deja, a matricei S, a doua ecuaţie se scrie

$$\begin{bmatrix} S_{11}^2 - 2\alpha_k S_{11} + (\alpha_k^2 + \beta_k^2)I_{k-1} & S_{11}S_{12} + S_{12}S_{22} & \times \\ 0 & 0 & \times \\ 0 & 0 & S_{33}^2 - 2\alpha_k S_{33} + (\alpha_k^2 + \beta_k^2)I_{n-k} \end{bmatrix} \begin{bmatrix} v_k' \\ v_k'' \\ v_k''' \end{bmatrix} = 0$$

unde s-a ţinut seama de faptul că blocul S_{22} este anulator pentru propriul polinom caracteristic (Cayley-Hamilton), iar $\times$ marchează blocurile neinteresante.
Dacă valorile proprii sunt distincte de celelalte valori proprii ale matricei S, atunci matricele

$$T_{11} = S_{11}^2 - 2\alpha_k S_{11} + (\alpha_k^2 + \beta_k^2)I_{k-1} \qquad \text{şi}$$

$$T_{33} = S_{33}^2 - 2\alpha_k S_{33} + (\alpha_k^2 + \beta_k^2)I_{n-k-1}$$

sunt nesingulare şi, prin urmare, ecuaţia (175) are ca soluţie un vector v_k având

$$\begin{cases} v_k''' = 0 \\ v_k'' = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \neq 0 \quad \text{arbitrar} \\ v_k' = -T_{11}^{-1}(S_{11}S_{12} + S_{12}S_{22})v_k'' \end{cases}$$

u_k obţinându-se din prima relaţie. Vectorii proprii ai matricei iniţiale A asociate aceloraşi valori proprii şi având structura

$$x_{k,k+1} = z_k \pm iw_k, \qquad z_k, w_k \in R^n$$

se calculează separat părţile reală şi imaginară

$$\begin{cases} z_k = Qu_k \\ w_k = Qv_k . \end{cases}$$

O alternativă pentru tehnicile de calcul de mai sus a vectorilor proprii, recomandată mai ales în cazurile în care se cere un set redus de vectori proprii, evită acumularea transformărilor în matricea **Q** şi apelează la metoda puterii inverse conform schemei:

1. Reducerea matricei la forma superior Hessenberg cu algoritmul **HQ**

$$A \leftarrow H = U^T AU.$$

2. Se aplică iterarea **QR** cu pas dublu cu deplasare implicită, fără acumularea transformărilor până la evidenţierea valorii proprii λ, al cărei vector propriu asociat dorim să-l calculăm.

3. Se aplică algoritmul 9.2 (metoda puterii inverse) matricei **H** cu deplasarea iniţială $\mu = \lambda$ care produce un vector **y** astfel încât $Hy = \lambda y$.

4. Se calculează vectorul propriu căutat $x = Uy$.

Dacă se doresc mai mulţi vectori proprii, se repetă paşii 2-4 de mai sus. Iterarea inversă pentru o matrice superior Hessenberg reprezintă o soluţie foarte economică întrucât practic s-a constatat că este suficientă o singură iteraţie (de complexitate $O(n^2)$), pentru calculul vectorului propriu **y**. În acest fel, partea cea mai complexă a schemei de mai sus este reducerea la forma superior Hessenberg şi evidenţierea iterativă a valorii proprii dorite. Dacă numărul vectorilor proprii doriţi este mare, (de exemplu mai mare decât $\frac{n}{4}$) atunci iterarea **QR** devine comparabilă cu cea din algoritmul 9.9 şi avantajul schemei de mai sus nu mai este evident.

În încheiere dorim să menţionăm din nou faptul că în multe aplicaţii vectorii proprii pot fi înlocuiţi cu succes cu vectorii Schur, respectiv de coloanele matricei de transformare **Q**, al căror calcul opţional este luat în considerare de către algoritmul 9.9 şi se efectuează cu o mare acurateţe.