Calcul Numeric

Cursul 10

2017

Anca Ignat

Forma superioară Hessenberg

Spunem că o matrice $\boldsymbol{H} \in \mathbb{R}^{n\times n}$ este în ***formă superioară Hessenberg*** dacă:

$$\boldsymbol{h_{ij} = 0\,,\, i = 1,\ldots,n\,,\, j = 1,\ldots,i-2}$$

O matrice în formă Hessenberg arată astfel:

$$\boldsymbol{H} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \cdots h_{1n-1} & h_{1n} \\ h_{21} & h_{22} & h_{23} \cdots h_{2n-1} & h_{2n} \\ 0 & h_{32} & h_{33} \cdots h_{3n-1} & h_{3n} \\ 0 & 0 & h_{43} \cdots h_{4n-1} & h_{4n} \\ \vdots & & & \\ 0 & 0 & 0 \cdots h_{nn-1} & h_{nn} \end{pmatrix}$$

Ne interesează un algoritm care să transforme o matrice pătratică ***A*** oarecare într-o matrice Hessenberg superioară ***H*** care să aibă aceleaşi valori proprii:

$$A \rightarrow H \quad \textbf{a.î.} \quad \mathrm{H} \sim A\,,\ H = \tilde{P} A\, \tilde{P}^{-1}\,,\ \tilde{P}\ \textbf{matrice nesingulară}$$

Algoritmul este o adaptare a algoritmului lui Housholder şi se desfăşoară în ***(n-2)*** paşi, folosind matricile de reflexie pentru a transforma matricea.

Pas 1

se efectuează operaţiile $A=P_1\, A\, P_1$ (matricea P_1 se alege astfel încât coloana 1 să fie transformată în formă superior Hessenberg)

Pas 2

$$A = P_2 A P_2 = P_2 (P_1 A^{init}) P_2$$

(***P₂*** transformă coloana 2 în formă superior Hessenberg fără să schimbe coloana 1)

Pas *r*

$$A = P_r A P_r = P_r (P_{r-1} \cdots P_1 A^{init} P_1 \cdots P_{r-1}) P_r$$

(se transformă coloana ***r*** în formă superior Hessenberg fără să schimbe primele ***(r-1)*** coloane)

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

La intrarea în pasul ***r*** matricea ***A*** are primele ***(r-1)*** coloane în formă superior Hessenberg. La ieşirea din pasul ***r*** matricea ***A*** va avea primele ***r*** coloane în formă superior Hessenberg:

$$A_{ies} = P_r\, A_{intr} P_r \;\;,\;\; A_{ies} \sim A_{intr}$$

$$P_r = I_n - 2v^r (v^r)^T \;,\quad v^r \in R^n \;,\quad \| v^r \|_2 = 1$$

Vectorul v^r se alege astfel ca matricea A_{ies} să aibă coloana ***r*** în formă superior Hessenberg şi să nu schimbe primele ***(r-1)*** coloane ale matricii A_{intr} .

Calculul matricii P_r

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

$$\beta = \sigma - k \cdot a_{r+1r}$$

$$k^2 = \sigma = a_{r+1r}^2 + \cdots + a_{ir}^2 + \cdots + a_{nr}^2 = \sum_{i=r+1}^{n} a_{ir}^2 \quad \Rightarrow \; k = \pm\sqrt{\sigma}$$

$$\text{semn } k = -\text{semn } a_{r+1r}$$

$$u := \begin{pmatrix} 0 \\ \vdots \\ 0 \\ a_{r+1r} - k \\ \vdots \\ a_{ir} \\ \vdots \\ a_{nr} \end{pmatrix}$$

$$\beta = 0 \rightarrow \; r = r + 1 \;\; (P = I_n)$$

Algoritmul de trecere de la matricea A la matricea $P_r A$ este următorul:

$$(P_r A)e_j = \begin{cases} Ae_j & \text{pentru } j = 1,\ldots,r-1 \\ (a_{1r}, a_{2r}, \ldots, a_{rr}, k, 0, \ldots, 0)^T & \text{pentru } j = r \\ Ae_j - \dfrac{\gamma_j}{\beta} u & \text{pentru } j = r+1,\ldots,n \end{cases}$$

$$\gamma_j = (Ae_j, u)_{\mathbb{R}^n} = \sum_{i=r+1}^{n} u_i a_{ij}$$

$$u_i = 0, \quad i = 1,\ldots,r \, , \; u_{r+1} = a_{r+1r} - k \, , \; u_i = a_{ir} \, , \quad i = r+2,\ldots,n$$

Vom descrie în continuare cum se efectuează operaţia ***A:=AP**$_r$* fără a face înmulţire matricială (matricea ***A*** este cea obţinută mai sus având primele ***r*** coloane în formă superior Hessenberg).

Vom arata că această operaţie nu schimbă forma superior Hessenberg obţinută. Vom pune în evidenţă transformările liniilor matricii A. Pentru $i=1,\dots,n$ avem:

$$e_i^T(AP) = \text{ noua linie } i \text{ a matricii } AP = (e_i^T A)(I_n - \frac{1}{\beta}uu^T) =$$

$$= e_i^T A - \frac{1}{\beta}(e_i^T A)u\, u^T = e_i^T A - \frac{\gamma_i}{\beta}u^T$$

unde

$$\gamma_i = (e_i^T A)u = a_{ir+1}u_{r+1} + \cdots + a_{in}u_n$$

Elementele liniei i se schimbă astfel:

$$a_{ij} = a_{ij} - \frac{\gamma_i}{\beta}u_j \;\; , j = r+1,\dots,n \, , \;\; i = 1,\dots,n$$

Operaţia $A{:=}AP_r$ nu modifică primele r coloane ale matricii A, ele rămânând în formă superior Hessenberg.

Algoritmul de obţinere a formei superior Hessenberg

for $r = 1,\ldots,n-2$

// construcţia matricii P_r – constanta β şi vectorul u

- $\sigma = \sum_{i=r+1}^{n} a_{ir}^2;$
- if $(\ \sigma \le \varepsilon\ )$ break ; $//\ r = r+1 \leftrightarrow P_r = I_n$
- $k = \sqrt{\sigma};$
- if $(\ a_{r+1r} > 0\ )\ k = -k;$
- $\beta = \sigma - k\ a_{r+1r};$
- $u_{r+1} = a_{r+1r} - k; u_i = a_{ir}\ , i = r+2,\ldots,n\ ;$

$// \ A = P_r * A$

$//$ transformarea coloanelor $j = r+1,\ldots,n$

- for $j = r+1,\ldots,n$

$$\begin{cases} \gamma = (\gamma_j / \beta) = (Ae_j, u) / \beta = (\sum_{i=r+1}^{n} u_i a_{ij}) / \beta; \\ \text{for } i = r+1,\ldots,n \\ \quad a_{ij} = a_{ij} - \gamma * u_i; \end{cases}$$

$//$ transformarea coloanei r a matricii A

- $a_{r+1r} = k$; $a_{ir} = 0, \quad i = r+2,\ldots,n$;

$// \ \boldsymbol{A = A * P_r}$

// transformarea liniilor $\boldsymbol{i = 1, \ldots, n}$

- for $\boldsymbol{i = 1, \ldots, n}$

$$
\begin{cases}
\boldsymbol{\gamma = (\gamma_i / \beta) = ((e_i^T A)u) / \beta = (\sum_{j=r+1}^{n} u_j a_{ij}) / \beta;} \\
\text{for } \boldsymbol{j = r+1, \ldots, n} \\
\qquad \boldsymbol{a_{ij} = a_{ij} - \gamma * u_j;}
\end{cases}
$$

Algoritmul *QR* de aproximare a valorilor proprii ale unei matrici oarecare

Prezentăm în continuare cel mai folosit algoritm de aproximare a valorilor proprii pentru matrici pătratice oarecare.

Spunem că o matrice $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ este în ***formă Schur reală*** dacă matricea $\boldsymbol{S}$ este în formă superior Hessenberg şi în plus este bloc-diagonală:

$$\boldsymbol{S} = \begin{pmatrix} \boldsymbol{S}_{11} & \boldsymbol{S}_{12} & \cdots & \boldsymbol{S}_{1p} \\ \boldsymbol{0} & \boldsymbol{S}_{22} & \cdots & \boldsymbol{S}_{2p} \\ \vdots & & & \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{S}_{pp} \end{pmatrix}$$

blocurile S_{ii} sunt astfel ca:

- $S_{ii} \in \mathbb{R}$ - este valoare proprie reală
- $S_{ii} \in \mathbb{R}^{2\times 2}$ - este bloc corespunzător valorilor proprii complexe

Valorile proprii corespunzătoare blocului $S_{ii} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathbb{R}^{2\times 2}$

sunt rădăcinile ecuaţiei:

$$\begin{vmatrix} \lambda - a & -b \\ -c & \lambda - d \end{vmatrix} = (\lambda - a)(\lambda - d) - bc = \lambda^2 - (a+d)\lambda + ad - bc = 0$$

Se presupune că această ecuaţie de gardul 2 are rădăcini complexe.

Algoritmul QR de aproximare a valorilor proprii construieşte un şir de matrici $A^{(k)} \in \mathbb{R}^{n\times n}$, matrici asemenea cu matricea ***A***, $A^{(k)} \sim A, \forall k$, şir care converge la o matrice în formă Schur reală, $A^{(k)} \to S, k \to \infty$. Matricea limită ***S*** este asemenea cu matricea ***A***, valorile prorii ale matricii ***S*** fiind uşor de calculat. Şirul $A^{(k)}$ se construieşte astfel:

$$A^{(0)} := A \;,\; A^{(0)} = Q_0 R_0 \text{ (descomp. } QR \text{ calc. pentru matricea } A^{(0)})$$

$$A^{(1)} := R_0 Q_0 \;,\; A^{(1)} = Q_1 R_1 \text{ (descomp. } QR \text{ calc. pentru matricea } A^{(1)})$$

$$A^{(2)} := R_1 Q_1$$

$$\vdots$$

$$A^{(k)} = Q_k R_k \text{ (descomp. } QR \text{ calc. pentru matricea } A^{(k)}),$$

$$A^{(k+1)} := R_k Q_k \;,\; k = 0,1,2,\ldots$$

Matricile $\boldsymbol{Q_k}$ sunt matrici ortogonale ($\boldsymbol{Q_k^{-1} = Q_k^T}$) iar matricile $\boldsymbol{R_k}$ sunt superior triunghiulare.

Matricile $\boldsymbol{A^{(k)}}$ şi $\boldsymbol{A^{(k+1)}}$ sunt asemenea:

$$\boldsymbol{Q_k^T * \mid A^{(k)} = Q_k R_k \;\Rightarrow R_k = Q_k^T A^{(k)}}$$

$$\boldsymbol{A^{(k+1)} = R_k Q_k = Q_k^T A^{(k)} Q_k \quad \Rightarrow A^{(k+1)} \sim A^{(k)}, \forall k}$$

Matricile şirului construit sunt toate asemenea, prin urmare au aceleaşi valori proprii, anume cele ale matricii iniţiale $\boldsymbol{A= A^{(0)}}$:

$$\boldsymbol{A = A^{(0)} \sim A^{(1)} \sim \cdots \sim A^{(k)} \sim \cdots \sim S}$$

Dacă matricea $A^{(k)}$ este în formă superioară Hessenberg, atunci descompunerea QR realizată cu algoritmul lui Givens se simplifică. Reamintim algoritmul lui Givens:

$$R_{n-1n}(\theta_{n-1n})\cdots R_{pn}(\theta_{pn})\cdots R_{pp+1}(\theta_{pp+1})\cdots R_{1n}(\theta_{1n})\cdots R_{12}(\theta_{12})\,A = R$$

Dacă matricea A este în formă Hessenberg în algoritmul lui Givens, din cele $\dfrac{n(n-1)}{2}$ înmulţiri cu matrici de rotaţie rămân doar $(n-1)$:

$$R_{n-1n}(\theta_{n-1n})\cdots R_{pp+1}(\theta_{pp+1})\cdots R_{23}(\theta_{23})R_{12}(\theta_{12})\,A = R.$$

Problema care se pune este dacă pornind cu o matrice în formă Hessenberg, toate matricile şirului rămân în formă Hessenberg:

$\boldsymbol{A^{(k)}}$ (în formă Hessenberg) $= \boldsymbol{H = QR}$ (cu Givens) $\Rightarrow$ **?**

$\boldsymbol{A^{(k+1)} = \overline{H} = RQ = Q^T A^{(k)} Q = Q^T H Q}$ – este tot în formă Hessenberg ?

Avem:

$$\boldsymbol{\overline{H} = Q^T H Q = R\, R_{12}^T(\theta_{12}) \cdots R_{rr+1}^T(\theta_{rr+1}) \cdots R_{n-1n}^T(\theta_{n-1n})}$$

Notăm cu:

$$\boldsymbol{\overline{R} = R\, R_{12}^T(\theta_{12})}$$

pentru care avem:

$$\boldsymbol{\begin{cases} \overline{r}_{i1} = c r_{i1} + s r_{i2} \quad , \forall i \\ \overline{r}_{i2} = -s r_{i1} + c r_{i2} \quad , \forall i \end{cases} + \begin{cases} r_{i1} = 0, i = 2, \ldots, n \\ r_{i2} = 0, i = 3, \ldots, n \end{cases} \Rightarrow \begin{cases} \overline{r}_{i1} = 0, i = 3, \ldots, n \\ \overline{r}_{i2} = 0, i = 3, \ldots, n \end{cases}}$$

deci coloana ***1*** se transformă în formă Hessenberg iar coloana ***2*** rămâne în formă suprior triunghiulară.

La pasul ***p*** avem:

$$\left(R\, R_{12}^T(\theta_{12})\cdots R_{p-1\,p}^T(\theta_{p-1\,p})\right) R_{pp+1}^T(\theta_{pp+1}) = \tilde{R}\, R_{pp+1}^T(\theta_{pp+1}) = \bar{R}\ ,$$

$$\tilde{R} = R\, R_{12}^T(\theta_{12})\cdots R_{p-1\,p}^T(\theta_{p-1\,p})$$

matricea $\widetilde{R}$ are primele ***(p-1)*** coloane în formă Hessenberg iar restul coloanelor sunt în formă superior triunghiulară. Vom arata că la acest pas matricea $\bar{R}$ va avea primele ***p*** coloane în formă Hessenberg iar restul coloanelor în formă superior triunghiulară. Operaţia $\bar{R} := \tilde{R}\, R_{pp+1}^T(\theta_{pp+1})$ presupune doar schimbarea elementelor coloanelor ***p*** şi ***p+1***:

$$\begin{cases} \bar{r}_{ip} = c\tilde{r}_{ip} + s\tilde{r}_{ip+1} \quad ,\forall i \\ \bar{r}_{ip+1} = -s\tilde{r}_{ip} + c\tilde{r}_{ip+1} \quad ,\forall i \end{cases} + \begin{cases} \tilde{r}_{ip} = 0,\ i = p+1,\ldots,n \\ \tilde{r}_{ip+1} = 0,\ i = p+2,\ldots,n \end{cases} \Rightarrow$$

$$\begin{cases} \bar{r}_{ip} = 0,\ i = p+2,\ldots,n \\ \bar{r}_{ip+1} = 0,\ i = p+2,\ldots,n \end{cases}$$

Observăm din relaţia de mai sus că în matricea $\bar{R}$ coloana p are formă Hessenberg iar coloana $p+1$ rămâne în formă superior triunghiulară (celelalte elemente din matrice nu se modifică). Prin urmare după pasul n-1 matricea $\bar{H} = A^{(k+1)}$ este în formă superioară Hessenberg. Algoritmul ***QR*** de aproximare a valorilor proprii folosind descompunerea Givens păstrează forma Hessenberg.

Algoritmul QR pentru valori proprii

/ / se aduce matricea $\boldsymbol{A}$ la forma Hessenberg

• $\boldsymbol{A} = \boldsymbol{\bar{Q}}\ \boldsymbol{A}\ \boldsymbol{\bar{Q}}^T$;

• $\boldsymbol{k} = \mathbf{0}$;

• while ($\boldsymbol{A} \neq$ forma Schur reală)

$$\begin{cases} \bullet\ \boldsymbol{A} = \boldsymbol{QR};\ / /\ \text{se calculează cu algoritmul Givens} \\ \bullet\ \boldsymbol{A} = \boldsymbol{RQ}\ \text{sau}\ \boldsymbol{Q}^T \boldsymbol{AQ}; \\ \bullet\ \boldsymbol{k} = \boldsymbol{k} + \mathbf{1}; \end{cases}$$

În practică se presupune că matricea A este în **formă Hessenberg neredusă**, adică:

$$a_{ii-1} \neq 0 \;\; \forall i = 2,\ldots,n$$

Dacă matricea nu este în formă neredusă, problema se decuplează:

$$A = \begin{array}{cc} \left[\begin{array}{cc} A_{11} & A_{12} \\ A_{21} & A_{22} \end{array}\right] & \begin{array}{c} p \\ n-p \end{array} \\ \begin{array}{cc} p & n-p \end{array} & \end{array} \quad , \;\; p = n-1 \text{ sau } n-2$$

Algoritmului *QR* cu deplasare ("*shift*") simplă

Algoritmul cu deplasare simplă este următorul:

- $A = \bar{Q}\, A\, \bar{Q}^T$; / / aducerea la forma Hessenberg neredusă
- $k = 0;$
- while ($A \neq$ forma Schur reală)

$$\begin{cases} \bullet\ A - d_k\, I_n = QR; \text{ / / se calc. cu alg. Givens} \\ \bullet\ A := RQ + d_k\, I_n; \\ \bullet\ k = k + 1; \end{cases}$$

$d_k \in \mathbb{R}$ sunt constantele de deplasare.

Dacă $\boldsymbol{A - d\, I_n = QR}$ $(A^{(k)})$ şi $\overline{\boldsymbol{A}} = \boldsymbol{RQ + d\, I_n}$ $(\boldsymbol{A^{(k+1)}})$, se pune problema dacă cele două matrici sunt asemenea ($\boldsymbol{A} \sim \overline{\boldsymbol{A}}$) (şirul de matrici construit cu pasul ***QR*** cu deplasare simplă au aceleaşi valori proprii).

$$\overline{\boldsymbol{A}} = \boldsymbol{Q^T QRQ + d\, Q^T Q = Q^T (QR + d\, I_n) Q = Q^T A Q} \Rightarrow \overline{\boldsymbol{A}} \sim \boldsymbol{A}$$

Varianta cu deplasare se efectuează pentru a accelera convergenţa algoritmului. Dacă $\boldsymbol{\lambda_1, \lambda_2, \ldots, \lambda_n}$ sunt valorile proprii ale matricii ***A*** ordonate astfel ca:

$$|\lambda_1 - \boldsymbol{d}| \geq |\lambda_2 - \boldsymbol{d}| \geq \cdots \geq |\lambda_n - \boldsymbol{d}|$$

Rapiditatea cu care $a_{p+1\,p}^{(k)} \to 0\,, k \to \infty$ este dată de rata de convergenţa a expresiei $\left| \frac{\lambda_{p+1} - d}{\lambda_p - d} \right|^k$. Dacă se alege $d \approx \lambda_n$ convergenţa $a_{n-1\,n}^{(k)} \to 0$ este rapidă. Avem următorul rezultat:

Teoremă

Fie d o valoare proprie a unei matrici Hessenberg nereduse H. Dacă $\overline{H} = RQ + d\, I_n$, cu $H - d\, I_n = QR$ descompunerea QR a matricii $H - d\, I_n = QR$. Atunci:

$$\overline{h}_{nn-1} = 0 \;\; , \;\; \overline{h}_{nn} = d$$

Algoritmul $\boldsymbol{QR}$ cu deplasare simplă găseşte valoarea proprie $\boldsymbol{d}$ într-un singur pas. Euristic s-a constatat că la fiecare pas, cea mai bună aproximare a unei valori proprii este $\boldsymbol{a_{nn}^{(k)}}$.

$$\boldsymbol{d_k = a_{nn}^{(k)}}$$

Algoritmul QR cu deplasare simplă

- $\boldsymbol{A = \bar{Q}\ A\ \bar{Q}^T}$**;** / / aducerea la forma Hessenberg neredusă
- $\boldsymbol{k = 0;}$
- while **(** $\boldsymbol{A} \neq$ forma Schur reală **)**

$$\begin{cases} \bullet\ \boldsymbol{A - a_{nn}\ I_n = QR;} \text{ / / se calc. cu algoritmul Givens} \\ \bullet\ \boldsymbol{A := RQ + a_{nn}\ I_n;} \\ \bullet\ \boldsymbol{k = k + 1;} \end{cases}$$

Algoritmului *QR* cu deplasare ("*shift*") dublă

În cazul când valorile proprii a_1, a_2 corespunzătoare blocului:

$$G = \begin{bmatrix} a_{pp} & a_{pn} \\ a_{np} & a_{nn} \end{bmatrix} \quad , \quad p = n - 1$$

sunt complexe, $a_1, a_2 \in \mathbb{C}$, abordarea cu deplasare simplă nu mai asigură accelerarea convergenţei. Avem:

$$\det(\lambda\, I_2 - G) = (\lambda - a_1)(\lambda - a_2) = (\lambda - a_{pp})(\lambda - a_{nn}) - a_{pn}a_{np} =$$

$$= \lambda^2 - (a_1 + a_2)\lambda + a_1 a_2 = \lambda^2 - (a_{pp} + a_{nn})\lambda + a_{pp}a_{nn} - a_{pn}a_{np}$$

$$a_1 + a_2 = a_{pp} + a_{nn} = trace(G) \quad , \quad a_1 a_2 = a_{pp}a_{nn} - a_{pn}a_{np} = \det G$$

Algoritmul ***QR*** cu deplasare dublă constă în trecerea de la matricea $A = A^{(k)}$ la matricea $A_2 = A^{(k+1)}$ realizând doi paşi cu deplasare simplă :

$A \rightarrow A_1$ (deplasare simplă a_1), $A_1 \rightarrow A_2$ (deplasare simplă a_2)

$$A - a_1 I_n = Q_1 R_1$$

$$A_1 = R_1 Q_1 + a_1 I_n$$

$$A_1 - a_2 I_n = Q_2 R_2$$

$$A_2 = R_2 Q_2 + a_2 I_n$$

Fie matricea :

$$M := \left(Q_1Q_2\right)\left(R_2R_1\right) = Q_1\left(Q_2R_2\right)R_1 = Q_1\left(A_1 - a_2\ I_n\right)R_1 =$$

$$= Q_1\left(Q_1^T AQ_1 - a_2\ I_n\right)R_1 = Q_1Q_1^T AQ_1R_1 - a_2Q_1R_1 =$$

$$= \left(A - a_2\ I_n\right)Q_1R_1 = \left(A - a_2\ I_n\right)\left(A - a_1\ I_n\right)$$

$$M = \left(Q_1Q_2\right)\left(R_2R_1\right) = \left(A - a_2\ I_n\right)\left(A - a_1\ I_n\right) =$$

$$= A^2 - (a_1 + a_2)A + a_1a_2\ I_n$$

Avem următoarele relaţii de asemănare:

$$A \sim A_1 = Q_1^T AQ_1 \sim A_2 = Q_2^T A_1Q_2 = Q_2^T Q_1^T AQ_1Q_2 = \left(Q_1Q_2\right)^T A\left(Q_1Q_2\right)$$

$$A_2 = \left(Q_1Q_2\right)^T A\left(Q_1Q_2\right) = Q^T AQ \quad , \quad Q := Q_1Q_2$$

Matricea $\boldsymbol{Q}$ care asigură trecerea de la matricea $\boldsymbol{A}$ la matricea $\boldsymbol{A_2}$ este matricea ortogonală din descompunerea $\boldsymbol{QR}$ a matricii $\boldsymbol{M} = \left(\boldsymbol{A} - \boldsymbol{a_2}\, \boldsymbol{I_n}\right)\left(\boldsymbol{A} - \boldsymbol{a_1}\, \boldsymbol{I_n}\right)$. Pasul $\boldsymbol{QR}$ cu deplasare dublă se face urmând etapele:

1) se calculează matricea $\boldsymbol{M} = \boldsymbol{A}^2 - \boldsymbol{s}\, \boldsymbol{A} + \boldsymbol{q I_n}$ cu

$$\boldsymbol{s = a_1 + a_2 = a_{pp} + a_{nn}} \quad , \quad \boldsymbol{q = a_1 a_2 = a_{pp}\, a_{nn} - a_{pn} a_{np}}\ ;$$

2) se calculează descompunerea $\boldsymbol{QR}$ a matricii $\boldsymbol{M}$;

3) $\boldsymbol{A_2 := Q^T A Q}$.

Vectori proprii

Considerăm două matrici asemenea $\boldsymbol{A}$ şi $\boldsymbol{B}$:

$$\boldsymbol{A} \sim \boldsymbol{B} \quad \Leftrightarrow \quad \boldsymbol{A} = \boldsymbol{P}\boldsymbol{B}\boldsymbol{P}^{-1} \;,\; \boldsymbol{P} \quad \text{matrice nesingulară}$$

Ştim că cele două matrici au acelaşi polinom caracteristic, $\boldsymbol{p}_A(\boldsymbol{\lambda}) \equiv \boldsymbol{p}_B(\boldsymbol{\lambda})$, deci au aceleaşi valori proprii. Ne interesează care este legătura între vectorii proprii asociaţi aceleiaşi valori proprii. Fie $\boldsymbol{u}$ vector propriu asociat valorii proprii $\boldsymbol{\lambda}$ pentru matricea $\boldsymbol{A}$ şi $\boldsymbol{w}$ vector propriu asociat valorii proprii $\boldsymbol{\lambda}$ pentru matricea $\boldsymbol{B}$. Care este relaţia între $\boldsymbol{u}$ şi $\boldsymbol{w}$?

$$Au = \lambda u \,, Bw = \lambda w \,,\; A = PBP^{-1} \;\Rightarrow\; PBP^{-1}u = \lambda u \;\Rightarrow$$

$$BP^{-1}u = \lambda\, P^{-1}u \;\Rightarrow\; w = P^{-1}u \quad,\; u = Pw$$

Dacă se aplică algoritmul ***QR*** unei matrici simetrice, forma Schur reală la care se ajunge este o matrice diagonală:

$$\boldsymbol{S} = \boldsymbol{\Lambda} = \mathbf{diag}[\lambda_1, \lambda_2, \ldots, \lambda_n]$$

Legătura dintre matricea simetrică iniţială A şi matricea diagonală este de forma:

$$\boldsymbol{S} = \boldsymbol{\Lambda} = \mathbf{diag}[\lambda_1, \lambda_2, \ldots, \lambda_n] = \boldsymbol{U}^T\boldsymbol{A}\,\boldsymbol{U}$$

unde ***U*** este o matrice ortogonală, coloanele matricii ***U*** fiind

vectori proprii asociaţi valorilor proprii reale $\lambda_1, \lambda_2, ..., \lambda_n$.
Matricea U se poate calcula astfel:

Algoritmul QR pentru matrici simetrice (valori +vectori proprii)

/ / se aduce matricea A la forma Hessenberg

- $A = \bar{Q}\, A\, \bar{Q}^T;$
- $U = \bar{Q}^T;$
- $k = 0;$
- **while (** $A \neq$ **matrice diagonală)**

$$\begin{cases} \bullet\ A = QR; \text{ / / se calculează cu algoritmul Givens} \\ \bullet\ A = RQ \text{ sau } Q^T AQ; \\ \bullet\ U = UQ; \\ \bullet\ k = k + 1; \end{cases}$$

Descompunerea după valori singulare
(**S**ingular **V**alue **D**ecomposition)

Teoremă

Fie $A \in \mathbb{R}^{m\times n}$. Atunci există o matrice ortogonală pătratică de dimensiune m, $U \in \mathbb{R}^{m\times m}$, o matrice ortogonală pătratică de dimensiune n, $V \in \mathbb{R}^{n\times n}$ şi constantele pozitive:

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0, \qquad r \leq \min\{m,n\} \text{ a.î.}$$

$$A = U\Sigma V^T, \quad \Sigma \in \mathbb{R}^{m\times n}, \Sigma = \begin{bmatrix} D & 0_{r\times(n-r)} \\ 0_{(m-r)\times r} & 0_{(m-r)\times(n-r)} \end{bmatrix}, \qquad (1)$$

$$D \in \mathbb{R}^{r\times r}, D = \mathbf{diag}\left[\sigma_1, \ldots, \sigma_r\right]$$

Constanta r este chiar rangul matricii A, r = **rang**(A). Constantele $\sigma_1, \sigma_2, \ldots, \sigma_r$ poartă numele de ***valori singulare*** ale matricii A.

Folosind relaţia (1) avem:

$$A^T = \left(U\Sigma V^T\right)^T = V\Sigma^T U^T,$$

$$AA^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T = U\Lambda_m U^T \quad ,$$

$$\Lambda_m = \Sigma\Sigma^T = \begin{bmatrix} D^2 & \mathbf{0}_{r\times(m-r)} \\ \mathbf{0}_{(m-r)\times r} & \mathbf{0}_{(m-r)\times(m-r)} \end{bmatrix} \in \mathbb{R}^{m\times m}$$

$$A^T A = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T\Sigma V^T = V\Lambda_n V^T \quad ,$$

$$\Lambda_n = \Sigma^T\Sigma = \begin{bmatrix} D^2 & \mathbf{0}_{r\times(n-r)} \\ \mathbf{0}_{(n-r)\times r} & \mathbf{0}_{(n-r)\times(n-r)} \end{bmatrix} \in \mathbb{R}^{n\times n}$$

Ţinând cont de ortogonalitatea matricilor ***U*** şi ***V***, putem rescrie relaţiile de mai sus astfel:

$$(AA^T)\,U = U\,\Lambda_m \quad , \; \Lambda_m = \mathbf{diag}\left[\sigma_1^2, \sigma_2^2, \ldots, \sigma_r^2, 0, \ldots, 0\right] \in \mathbb{R}^{m\times m}$$

$$(A^T A)\,V = V\,\Lambda_n \quad , \; \Lambda_n = \mathbf{diag}\left[\sigma_1^2, \sigma_2^2, \ldots, \sigma_r^2, 0, \ldots, 0\right] \in \mathbb{R}^{n\times n}$$

Din aceste relaţii deducem că $\sigma_1^2, \sigma_2^2, \ldots, \sigma_r^2$ sunt valorile proprii strict pozitive ale matricilor AA^T şi/sau A^TA iar matricile ***U*** şi ***V*** sunt matrici ale căror coloane sunt vectorii proprii asociaţi (cei ce formează baze ortonormate). Matricile AA^T şi A^TA sunt matrici simetrice:

$$\left(AA^T\right)^T = \left(A^T\right)^T A^T = AA^T \quad , \quad \left(A^T A\right)^T = A^T \left(A^T\right)^T = A^T A$$

şi au toate valorile proprii nenegative:

$$\left(AA^T\right)u = \lambda u \Rightarrow \left(AA^T u, u\right) = \left(\lambda u, u\right) \Rightarrow$$

$$\lambda = \frac{\left(A^T u, A^T u,\right)}{\left(u, u\right)} = \frac{\left\|A^T u\right\|_2^2}{\left\|u\right\|_2^2} \geq 0$$

Putem folosi descompunerea după valori singulare pentru a defini pseudo-inversa unei matrici oarecare $A \in \mathbb{R}^{m\times n}$ ($n \neq m$).

$$A = U\Sigma V^T \quad , \quad A^{-1} =_? \left(U\Sigma V^T\right)^{-1} = \left(V^T\right)^{-1} \Sigma_?^{-1} U^{-1} = V\Sigma_?^{-1} U^T$$

Rămâne de definit matricea $\Sigma_?^{-1}$. Urmând acest raţionament se defineşte ***pseudoinversa Moore-Penrose*** a unei matrici $A \in \mathbb{R}^{m\times n}$ astfel:

$$A^I = V\Sigma^I U^T \quad , \; A^I \in \mathbb{R}^{n\times m} \quad , \; \Sigma^I = \begin{bmatrix} D^{-1} & 0_{r\times(m-r)} \\ \mathbf{0}_{(n-r)\times r} & 0_{(n-r)\times(m-r)} \end{bmatrix} \in \mathbb{R}^{n\times m},$$

$$D^{-1} \in \mathbb{R}^{r\times r}, D^{-1} = \mathbf{diag}\left[\frac{1}{\sigma_1},\ldots,\frac{1}{\sigma_r}\right].$$

Pseudoinversa definită mai sus satisface următoarele proprietăţi:

$$\left(A^I\right)^I = A \;\; , \forall A \in \mathbb{R}^{m\times n} \;\; ; \; \left(A^T\right)^I = \left(A^I\right)^T, \; \forall A \in \mathbb{R}^{m\times n}$$

$$AA^I A = A \qquad , \qquad A^I AA^I = A^I$$

Există o proprietate care nu mai este satisfăcută de psudoinversă deşi este respectată de inversa clasică:

$$\exists\, \boldsymbol{A}, \boldsymbol{B} \;\; \textbf{a.}\text{î.} \quad \left(\boldsymbol{AB}\right)^{\boldsymbol{I}} \neq \boldsymbol{B}^{\boldsymbol{I}} \boldsymbol{A}^{\boldsymbol{I}}.$$

Descompunerea după valori singulare poate fi utilizată şi pentru rezolvarea sistemelor liniare cu matrici oarecare (***m≠n***)

$$\boldsymbol{Ax} = \boldsymbol{b} \quad , \quad \boldsymbol{A} \in \mathbb{R}^{\boldsymbol{m}\times\boldsymbol{n}} \;\; , \;\; \boldsymbol{b} \in \mathbb{R}^{\boldsymbol{m}} \quad , \quad \boldsymbol{x} := \boldsymbol{A}^{\boldsymbol{I}}\boldsymbol{b} \in \mathbb{R}^{\boldsymbol{n}}.$$