Calcul Numeric

Cursul 8

2022

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:

$\boldsymbol{A \rightarrow H}$ **a.î.** $\mathbf{H} \sim \boldsymbol{A}$**,** $\boldsymbol{H = \tilde{P} A \tilde{P}^{-1}}$ **,** $\boldsymbol{\tilde{P}}$ **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 $\boldsymbol{A=P_1\ A\ P_1}$ (matricea $\boldsymbol{P_1}$ 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,…,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:

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

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

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

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

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

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

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

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

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

Algoritmul de trecere de la matricea $\boldsymbol{A}$ la matricea $\boldsymbol{P_r A}$ este următorul:

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

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

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

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,...,n*** avem:

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

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

unde

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

Elementele liniei $\boldsymbol{i}$ se schimbă astfel:

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

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

Algoritmul de obţinere a formei superior Hessenberg

for $\boldsymbol{r = 1,\ldots,n-2}$

// construcţia matricii $\boldsymbol{P_r}$ – constanta $\boldsymbol{\beta}$ şi vectorul $\boldsymbol{u}$

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

$// \; A = P_r * A$

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

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

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

$//$ transformarea coloanei r a matricii A

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

$// \; A = A * P_r$

// transformarea liniilor $i = 1, \ldots, n$

- for $i = 1, \ldots, n$

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

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

Prezentăm în continuare cel mai folosit algoritm de aproximare a valorilor proprii pentru matrice 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 $\boldsymbol{S}_{ii}$ sunt astfel ca:

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

Valorile proprii corespunzătoare blocului $\boldsymbol{S}_{ii} = \begin{pmatrix} \boldsymbol{a} & \boldsymbol{b} \\ \boldsymbol{c} & \boldsymbol{d} \end{pmatrix} \in \mathbb{R}^{2\times 2}$ 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. Șirul $A^{(k)}$ se construieşte astfel:

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

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

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

$$\vdots$$

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

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

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

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

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

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

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

Avem:

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

Notăm cu:

$$\bar{\boldsymbol{R}} = \boldsymbol{R}\, \boldsymbol{R}_{12}^T(\theta_{12})$$

pentru care avem:

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

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

La pasul ***p*** avem:

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

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

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

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

.

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

Observăm din relaţia de mai sus că în matricea $\overline{\boldsymbol{R}}$ coloana $\boldsymbol{p}$ are formă Hessenberg iar coloana $\boldsymbol{p+1}$ rămâne în formă superior triunghiulară (celelalte elemente din matrice nu se modifică).

Prin urmare după pasul ***n-1*** matricea $\overline{\boldsymbol{H}} = \boldsymbol{A}^{(\boldsymbol{k}+1)}$ este în formă superioară Hessenberg. Algoritmul $\boldsymbol{QR}$ de aproximare a valorilor proprii folosind descompunerea Givens păstrează forma Hessenberg.

Algoritmul QR pentru valori proprii

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

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

• $\boldsymbol{k = 0}$;

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

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

În practică se presupune că matricea $\boldsymbol{A}$ 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:

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

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

Algoritmul cu deplasare simplă este următorul:

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

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

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

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

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

$$\left|\lambda_1 - d\right| \geq \left|\lambda_2 - d\right| \geq \cdots \geq \left|\lambda_n - d\right|$$

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

Teoremă

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

- $\boldsymbol{A = \overline{Q}\ A\ \overline{Q}^T}$**;** / / 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:

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

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

$$\det(\lambda I_2 - G) = (\lambda - a_1)(\lambda - a_2) = (\lambda - a_{pp})(\lambda - a_{nn}) - a_{pn}a_{np} =$$
$$= \lambda^2 - (a_1 + a_2)\lambda + a_1a_2 = \lambda^2 - (a_{pp} + a_{nn})\lambda + a_{pp}a_{nn} - a_{pn}a_{np}$$

$$a_1 + a_2 = a_{pp} + a_{nn} = trace(G) \quad , \quad a_1a_2 = a_{pp}a_{nn} - a_{pn}a_{np} = \det G$$

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 :

$$M := (Q_1 Q_2)(R_2 R_1) = Q_1 (Q_2 R_2) R_1 = Q_1 (A_1 - a_2\ I_n) R_1 =$$
$$= Q_1 \left(Q_1^T A Q_1 - a_2\ I_n \right) R_1 = Q_1 Q_1^T A Q_1 R_1 - a_2 Q_1 R_1 =$$
$$= (A - a_2\ I_n) Q_1 R_1 = (A - a_2\ I_n)(A - a_1\ I_n)$$
$$M = (Q_1 Q_2)(R_2 R_1) = (A - a_2\ I_n)(A - a_1\ I_n) =$$
$$= A^2 - (a_1 + a_2) A + a_1 a_2\ I_n$$

Avem următoarele relaţii de asemănare:

$$A \sim A_1 = Q_1^T A Q_1 \sim A_2 = Q_2^T A_1 Q_2 = Q_2^T Q_1^T A Q_1 Q_2 = \left(Q_1 Q_2\right)^T A \left(Q_1 Q_2\right)$$

$$A_2 = \left(Q_1 Q_2\right)^T A \left(Q_1 Q_2\right) = Q^T A Q \quad , \quad Q := Q_1 Q_2$$

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 $M = \left(A - a_2\, I_n\right)\left(A - a_1\, I_n\right)$. Pasul $\boldsymbol{QR}$ cu deplasare dublă se face urmând etapele:

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

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

2) se calculează descompunerea ***QR*** a matricii ***M****;*

3) $A_2 := Q^T A Q$**.**

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, $\boldsymbol{p_A(\lambda) \equiv p_B(\lambda)}$, 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 $\boldsymbol{w}$?

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

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

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

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

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

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

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)

$$
\begin{aligned}
&/ / \text{ se aduce matricea } A \text{ la forma Hessenberg}\\
&\bullet\ A = \overline{Q}\, A\, \overline{Q}^T;\\
&\bullet\ U = \overline{Q}^{\,T};\\
&\bullet\ k = 0;\\
&\bullet\ \text{while } (\, A \neq \text{matrice diagonală }) \\
&\qquad \begin{cases} \bullet\ A = QR; \ / / \text{ se calculează cu algoritmul Givens} \\ \bullet\ A = RQ \text{ sau } Q^T \mathrm{AQ}; \\ \bullet U = UQ; \\ \bullet\ k = k+1; \end{cases}
\end{aligned}
$$

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 $\boldsymbol{r}$ este chiar rangul matricii $\boldsymbol{A}$, $\boldsymbol{r} = \mathbf{rang}(\boldsymbol{A})$.
Constantele $\boldsymbol{\sigma_1, \sigma_2, \ldots, \sigma_r}$ poartă numele de ***valori singulare*** ale matricii $\boldsymbol{A}$.
Folosind relaţia (1) avem:

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

$$\boldsymbol{AA^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T = U\Lambda_m U^T},$$

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

$$\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Sigma}^T\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \boldsymbol{V}\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\boldsymbol{V}^T = \boldsymbol{V}\boldsymbol{\Lambda}_n\boldsymbol{V}^T \quad ,$$

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

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

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

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

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

$$\boldsymbol{\left(AA^T\right)^T = \left(A^T\right)^T A^T = AA^T \quad , \quad \left(A^TA\right)^T = A^T\left(A^T\right)^T = A^TA}$$

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

$\boldsymbol{A} \in \mathbb{R}^{m\times n}$ astfel:

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

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

Pseudoinversa definită mai sus satisface următoarele proprietăţi:

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

$$\boldsymbol{A}\boldsymbol{A}^{I}\boldsymbol{A} = \boldsymbol{A} \qquad , \qquad \boldsymbol{A}^{I}\boldsymbol{A}\boldsymbol{A}^{I} = \boldsymbol{A}^{I}$$

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 ($\boldsymbol{m \neq n}$)

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

Problema celor mai mici pătrate

$$\boldsymbol{A} \in \mathbb{R}^{m\times n} \;,\; \boldsymbol{b} \in \mathbb{R}^{m} \;,\; \boldsymbol{Ax} = \boldsymbol{b} \;,\; \boldsymbol{x} \in \mathbb{R}^{n} \qquad \textbf{(1)}$$

$$\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\
&a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\
&\vdots \\
&a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i \\
&\vdots \\
&a_{m1}x_1 + a_{m2}x_2 \cdots + a_{mn}x_n = b_m
\end{aligned}$$

Sistemul are soluţii clasice dacă:

$$\textbf{rang}\ \boldsymbol{A} = \textbf{rang}\ [\boldsymbol{A} \mid \boldsymbol{b}]$$

- $\boldsymbol{m} < \boldsymbol{n}$ - o infinitate de soluţii
- $\boldsymbol{m} \geq \boldsymbol{n}$
  - dacă $\textbf{rang}\ \boldsymbol{A} = \textbf{rang}\ [\boldsymbol{A} \mid \boldsymbol{b}]$ soluţii clasice
  - dacă $\textbf{rang}\ \boldsymbol{A} \neq \textbf{rang}\ [\boldsymbol{A} \mid \boldsymbol{b}]$ soluţii în sensul celor mai mici pătrate

Vectorul reziduu:

$$\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{m}}$$

Vectorul $\boldsymbol{x} \in \mathbb{R}^{n}$ se numeşte ***soluţie în sensul celor mai mici pătrate*** pentru sistemul (1) dacă este soluţia următoarei probleme de optimizare:

$$\min\{\|r(x)\|_2^2 = \|b - Ax\|_2^2;\ x \in \mathbb{R}^n\} \qquad \textbf{\textit{(LSP)}}$$

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \in \mathbb{R}^{3\times 2} \quad , \quad b = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},\ m = 3,\ n = 2$$

rang A **= 2 ≠ rang** $[A \mid b]$ **= 3**

Sistemul:

$$\begin{aligned} \boldsymbol{x_1} + \mathbf{4}\boldsymbol{x_2} &= \mathbf{0} \\ \mathbf{2}\boldsymbol{x_1} + \mathbf{5}\boldsymbol{x_2} &= \mathbf{0} \\ \mathbf{3}\boldsymbol{x_1} + \mathbf{6}\boldsymbol{x_2} &= \mathbf{1} \end{aligned} \qquad \mathbf{(2)}$$

nu are soluţie clasică (nu există $\boldsymbol{x_1}$, $\boldsymbol{x_2}$ care să satisfacă toate cele 3 ecuaţii simultan). Vectorul reziduu are forma:

$$\boldsymbol{r(x)=b-Ax}=\begin{pmatrix}0\\0\\1\end{pmatrix}-\begin{bmatrix}1&4\\2&5\\3&6\end{bmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix}=\begin{pmatrix}-x_1-4x_2\\-2x_1-5x_2\\1-3x_1-6x_2\end{pmatrix}$$

Soluţia în sensul celor mai mici pătrate a acestui sistem este definită ca soluţia problemei de optimizare:

$$\min\{(-x_1-4x_2)^2+(-2x_1-5x_2)^2+\left(1-3x_1-6x_2\right)^2;\ x_1,x_2\in\mathbb{R}\}$$

$$\min\{1-6x_1-12x_2+64x_1x_2+14x_1^2+77x_2^2;\ x_1,x_2\in\mathbb{R}\}$$

Această problemă de minimizare are soluţia:

$$x_1 = \frac{13}{18}, \qquad x_2 = -\frac{2}{9}, \qquad \| r(x) \|_2^2 = \frac{1}{6}$$

şi este soluţia în sensul celor mai mici pătrate a sistemului (2).

$$\textbf{range}\left(\mathbf{A}\right) = \{\, \mathbf{y} \in \mathbb{R}^m ; y = a_1 A^1 + a_2 A^2 + \cdots + a_n A^n ,\, a_i \in \mathbb{R},\, i = \overline{1,n} \}$$

$A = \left[A^1 \ A^2 \cdots A^n \right]$, $A^i \in \mathbb{R}^m$ **sunt coloanele matricii** A

Teoremă

Fie $\boldsymbol{A} \in \mathbb{R}^{\boldsymbol{m \times n}}$ $(\boldsymbol{m} \geq \boldsymbol{n})$, $\boldsymbol{b} \in \mathbb{R}^{\boldsymbol{m}}$. Un vector $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$ minimizează norma euclidiană a vectorului reziduu $\|\boldsymbol{r(x)}\|_2 = \|\boldsymbol{b} - \boldsymbol{Ax}\|_2$, rezolvând problema ***(LSP)***, dacă şi numai dacă:

$$\boldsymbol{r}(\boldsymbol{x}) \perp \mathbf{range}(\boldsymbol{A}) \quad \Leftrightarrow \quad \boldsymbol{A}^{\boldsymbol{T}} \boldsymbol{r}(\boldsymbol{x}) = \mathbf{0}$$

sau echivalent

$$\boldsymbol{A}^{\boldsymbol{T}} \boldsymbol{A} \boldsymbol{x} = \boldsymbol{A}^{\boldsymbol{T}} \boldsymbol{b} \qquad \textbf{(3)}$$

Sistemul (3) poartă numele de sistemul de ***ecuaţii normale***.

Este un sistem pătratic de dimensiune $\boldsymbol{n}$, matricea sistemului $\boldsymbol{A}^T\boldsymbol{A} \in \mathbb{R}^{n\times n}$ este simetrică. Sistemul de ecuaţii normale (3) este nesingular dacă şi numai dacă **rang**$\boldsymbol{A} = \boldsymbol{n}$, în acest caz soluţia $\boldsymbol{x}$ a sistemului (3) este unică.

$$\det A^T A \neq 0 \iff \operatorname{rang} A = n$$

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}, \quad A^T A = \begin{bmatrix} 14 & 32 \\ 32 & 77 \end{bmatrix}, \; A^T b = \begin{pmatrix} 3 \\ 6 \end{pmatrix}$$

$$\begin{aligned} 14x_1 + 32x_2 &= 3 \\ 32x_1 + 77x_2 &= 6 \end{aligned} \quad \Rightarrow \; x_1 = \frac{13}{18}, \; x_2 = -\frac{2}{9}$$

Pseudo-inversa matricii *A*

Presupunem că $\boldsymbol{A}$ are **rang** $\boldsymbol{A = n}$. Atunci pseudo-inversa poate fi definită ca:

$$\boldsymbol{A^{+} = \left(A^{T} A\right)^{-1} A^{T}} \in \mathbb{R}^{\boldsymbol{n\times m}} \qquad (\boldsymbol{A^{+} = A^{I}} \quad ?)$$

$$\boldsymbol{A^{+} = \begin{bmatrix} 14 & 32 \\ 32 & 77 \end{bmatrix}^{-1} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}}$$

Rezolvarea sistemului de ecuaţii normale

1) Folosind factorizarea Cholesky (descompunere $\boldsymbol{LU}$) pentru matrici simetrice:

$$\boldsymbol{A^T A = LL^T}, \quad \boldsymbol{L} \in \mathbb{R}^{n\times n}$$ **matrice inferior triunghiulară**

- Se calculează matricea $\boldsymbol{A^T A}$ şi vectorul $\boldsymbol{A^T b}$;
- Se calculează factorizarea Cholesky a matricii $\boldsymbol{A^T A = LL^T}$;
- Se rezolvă sistemul inferior triunghiular $\boldsymbol{Ly = A^T b}$ pentru $\boldsymbol{y}$;
- Se rezolvă sistemul superior triunghiular $\boldsymbol{L^T x = y}$ pentru $\boldsymbol{x}$;

2) Se calculează descompunerea $\boldsymbol{QR}$ (cu algoritmul lui

Householder adaptat) pentru matricea $\boldsymbol{A}$:

$$\boldsymbol{A} = \boldsymbol{QR} \ , \ \boldsymbol{Q} \in \mathbb{R}^{m\times m} \textbf{ matrice ortogonală} \ , \ \boldsymbol{R} \in \mathbb{R}^{m\times n},$$

$$\boldsymbol{R} = \begin{bmatrix} \bar{\boldsymbol{R}} \in \mathbb{R}^{n\times n} \\ \boldsymbol{0}_{(m-n)\times n} \end{bmatrix} \quad , \quad \bar{\boldsymbol{R}} - \textbf{matrice superior triunghiulară}$$

- Se calculează factorizarea $\boldsymbol{QR}$ modificată a matricii $\boldsymbol{A}$;
- Se calculează vectorul $\boldsymbol{Q^T b}$;
- Se rezolvă sistemul sup. triunghiular $\bar{\boldsymbol{R}}\boldsymbol{x} = \left(\boldsymbol{Q}^T\boldsymbol{b}\right)_{i=1,n}$;

3) Se foloseşte desc. după valori singulare a matricii $\boldsymbol{A}$

$$\boldsymbol{A} = \boldsymbol{U\Sigma V^T}, \quad \boldsymbol{\Sigma} \in \mathbb{R}^{m\times n}, \boldsymbol{U} \in \mathbb{R}^{m\times m}, \boldsymbol{V} \in \mathbb{R}^{n\times n}$$

- Se calculează SVD pentru matricea $\boldsymbol{A=U\Sigma V^T}$;
- Se calculează vectorul $\boldsymbol{U^Tb}$;
- Se rezolvă sistemul diagonal $\boldsymbol{\Sigma w = U^Tb}$ pentru $\boldsymbol{w}$;
- Soluţia este $\boldsymbol{x=Vw}$;

1), 2) sau 3) ? → se recomandă **2)**