Calcul Numeric

Cursul 8

2020

Anca Ignat

Valori şi vectori proprii
(eigenvalues, eigenvectors)

Definiţie

Fie $A \in \mathbb{R}^{n \times n}$. Numărul complex $\lambda \in \mathbb{C}$ se numeşte ***valoare proprie*** a matricei A dacă există un vector $u \in \mathbb{C}^n, u \neq 0$ astfel ca:

$$Au = \lambda u$$

Vectorul u se numeşte ***vector propriu*** asociat valorii proprii λ.

Pentru existenţa vectorului $u \neq 0$ este necesar şi suficient ca matricea $(\lambda I_n - A)$ să fie singulară , adică $\det(\lambda I_n - A) = 0$.

Polinomul de grad ***n***:

$$p_A(\lambda) = \det(\lambda I_n - A)$$

se numeşte ***polinom caracteristic*** al matricei ***A***.

Propoziţia 1

Fie rădăcinile polinomului caracteristic $\lambda_1, \lambda_2, \ldots, \lambda_n$ distincte, $\lambda_i \neq \lambda_j$ pentru $1 \leq i < j \leq n$ şi $u_1, u_2, \ldots, u_n$ vectorii proprii corespunzători. Atunci $u_1, u_2, \ldots, u_n$ sunt liniar independenţi. (demonstraţia se face prin inducţie)

Propoziţia 2

Fie valorile proprii λ_i ale matricei $A \in \mathbb{R}^{n\times n}$ distincte. Atunci există o matrice nesingulară T astfel ca:

$$T^{-1}AT = \mathbf{diag}[\lambda_1, \lambda_2, \ldots, \lambda_n].$$

Demonstraţie. Fie $u_1, u_2, \ldots, u_n$ vectorii proprii ai matricei A. Considerăm matricea T ale cărei coloane sunt vectorii proprii u_i, $\mathrm{T} = [u_1 \quad u_2 \quad \ldots \quad u_n]$. Deoarece vectorii proprii sunt liniar independenţi conform propoziţiei 1 rezultă că matricea T este nesingulară. Vom avea:

$$AT = \left[Au_1\ Au_2 \ldots Au_n\right] = \left[\lambda_1 u_1\ \lambda_2 u_2 \ldots \lambda_n u_n\right] = T.\mathbf{diag}\left[\lambda_1\ \lambda_2 \ldots \lambda_n\right]$$

Înmulţind la stânga cu T^{-1} obţinem concluzia propoziţiei 2.

Definiţie

Matricile ***A*** şi ***B*** sunt ***asemenea*** (notaţie ***A~B***) dacă şi numai dacă există o matrice nesingulară ***T*** (**det** ***T*** **≠ 0**) astfel ca:

$$\boldsymbol{A=T\,B\,T^{-1}}$$

Propoziţia 3

$$\boldsymbol{A \sim B \;\Rightarrow\; p_A(\lambda) = p_B(\lambda).}$$

Demonstraţie.

$$\boldsymbol{p_A(\lambda) = \det(\lambda I_n - A) = \det\left(\lambda I_n - TBT^{-1}\right) = \det\left(\lambda TT^{-1} - TBT^{-1}\right)}$$

$$\boldsymbol{= \det\left(T\left(\lambda I_n - B\right)T^{-1}\right) = \det(T)\det\left(\lambda I_n - B\right)\det\left(T^{-1}\right) = p_B(\lambda)}$$

Propoziţia 3 ne spune că matricele asemenea au același polinom caracteristic şi aceleași valori proprii.

Teorema lui Gershgorin

Fie $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ **şi** $\boldsymbol{\lambda} \in \mathbb{C}$ o valoare proprie oarecare a matricei $\boldsymbol{A}$. Atunci:

$$\exists i_0 \in \{1, 2, \ldots, n\} \text{ astfel încât } \left|\lambda - a_{i_0 i_0}\right| \le r_{i_0}\ ,\ \ r_{i_0} = \sum_{\substack{j=1 \\ j \neq i_0}}^{n} \left|a_{i_0 j}\right|.$$

(Valoarea proprie $\boldsymbol{\lambda}$ se află în cercul din planul complex de centru $\boldsymbol{a_{i_0 i_0}}$ **şi rază** $\boldsymbol{r_{i_0}}$.)

Demonstraţie. Fie $\lambda \in \mathbb{C}$ o valoare proprie a matricei A şi $u \neq 0$ un vector propriu asociat valorii proprii λ, $Au = \lambda u$. Avem:

$$\lambda u_i = a_{ii}u_i + \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}u_j \iff (\lambda - a_{ii})u_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}u_j, \; i = 1,\ldots,n.$$

Fie i_0 astfel ca $|u_{i_0}| = \|u\|_\infty = \max\{|u_k|; k = 1,\ldots,n\} > 0$ ($u \neq 0$).

Vom avea:

$$|\lambda - a_{i_0 i_0}| = \left|\sum_{\substack{j=1 \\ j\neq i_0}}^{n} a_{i_0 j}\frac{u_j}{u_{i_0}}\right| \leq \sum_{\substack{j=1 \\ j\leq i_0}}^{n} |a_{i_0 j}|\frac{|u_j|}{|u_{i_0}|} \leq r_{i_0} \text{ , ţinând seama că } \frac{|u_j|}{|u_{i_0}|} \leq 1.$$

Observaţie. Presupunem că matricea $\boldsymbol{A}$ are $\boldsymbol{n}$ vectori proprii liniar independenţi $\boldsymbol{u}^1, \boldsymbol{u}^2, \ldots, \boldsymbol{u}^n$ asociaţi valorilor proprii $\boldsymbol{\lambda_1}$, $\boldsymbol{\lambda_2}$, ..., $\boldsymbol{\lambda_n}$ Fie $\boldsymbol{U} = \left[\boldsymbol{u}^1\, \boldsymbol{u}^2 \ldots \boldsymbol{u}^n\right]$. Datorită independenţei vectorilor $\boldsymbol{u}^k$ rezultă că matricea $\boldsymbol{U}$ este nesingulară şi avem:

$$\Lambda = \operatorname{diag}\left[\lambda_1, \lambda_2, \ldots, \lambda_n\right] \qquad \boldsymbol{U^{-1}AU = \Lambda.}$$

Considerăm matricea perturbată:

$$\boldsymbol{A(\varepsilon) = A + \varepsilon B.}$$

$$\boldsymbol{U^{-1}A(\varepsilon)U = \Lambda + \varepsilon U^{-1}BU = \Lambda + \varepsilon C} \, .$$

$$\boldsymbol{A(\varepsilon) \sim U^{-1}A(\varepsilon)\, U} \Rightarrow \text{ au aceleaşi valori proprii } \boldsymbol{\lambda_i(\varepsilon)}$$

$$\left|\lambda(\varepsilon) - \lambda_i - \varepsilon c_{ii}\right| \le \varepsilon \sum_{\substack{j=1 \\ j \ne i}}^{n} \left|c_{ij}\right| \;\; \Rightarrow \;\; \left|\lambda(\varepsilon) - \lambda_i\right| = \mathcal{O}(\varepsilon).$$

Metoda puterii pentru matrice simetrice

Propoziţie

Fie $\boldsymbol{A} \in \mathbb{R}^{n\times n}, \boldsymbol{A} = \boldsymbol{A}^T$. Atunci toate valorile proprii ale matricei $\boldsymbol{A}$ sunt numere reale.

Demonstraţie. Fie $\lambda \in \mathbb{C}$ şi $\boldsymbol{u} \in \mathbb{C}^n, \boldsymbol{u} \neq \boldsymbol{0}$, $\boldsymbol{Au} = \lambda \boldsymbol{u}$.

Considerăm produsul scalar:

$$(\boldsymbol{Au}, \boldsymbol{u})_{\mathbb{C}^n} = \lambda (\boldsymbol{u}, \boldsymbol{u})_{\mathbb{C}^n} = \lambda \|\boldsymbol{u}\|_2^2 .$$

$$(\boldsymbol{Au}, \boldsymbol{u})_{\mathbb{C}^n} = \left(\boldsymbol{u}, \boldsymbol{A}^T \boldsymbol{u}\right)_{\mathbb{C}^n} = (\boldsymbol{u}, \boldsymbol{Au})_{\mathbb{C}^n} = \overline{(\boldsymbol{Au}, \boldsymbol{u})}_{\mathbb{C}^n} \Rightarrow (\boldsymbol{Au}, \boldsymbol{u})_{\mathbb{C}^n} \in \mathbb{R}$$

$$\lambda = \frac{(\boldsymbol{Au}, \boldsymbol{u})_{\mathbb{C}^n}}{\|\boldsymbol{u}\|_2^2} \in \mathbb{R}.$$

Propoziţie

Fie $A \in \mathbb{R}^{n\times n}$, $A = A^T$. Atunci există o bază ortonormată de vectori proprii ai matricei A, $\{u^1, u^2, \ldots, u^n\}$:

$$\left(u^i, u^j\right)_{\mathbb{R}^n} = \delta_{ij} = \begin{cases} 1 & \text{dacă } i = j \\ 0 & \text{dacă } i \neq j \end{cases}$$

Echivalent, putem scrie ca există vectori proprii $\{u^1, u^2, \ldots, u^n\}$ asociaţi valorilor proprii reale $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ atfel ca:

$$AU = U\Lambda \quad \Leftrightarrow \quad U^T A U = \Lambda$$

cu $\Lambda = \mathrm{diag}[\lambda_1, \lambda_2, \ldots, \lambda_n]$ şi $U = [u^1\, u^2\, \cdots\, u^n]$ matrice ortogonală.

Definiţie

Se numeşte *coeficient Rayleigh* al vectorului $u \in \mathbb{R}^n$ pentru matricea ***A*** următoarea mărime scalară:

$$r(u) = \frac{u^T A u}{u^T u} = \frac{(Au, u)_{\mathbb{R}^n}}{(u, u)_{\mathbb{R}^n}} = \frac{(Au, u)_{\mathbb{R}^n}}{\| u \|_2^2}$$

Se verifică uşor că dacă $u \in \mathbb{R}^n$ este vector propriu al matricei ***A*** asociat valorii proprii ***λ*** atunci ***r(u)= λ***.

Fie $A \in \mathbb{R}^{n \times n}, A = A^T$. Matricea are valori proprii reale λ_1, λ_2,..., λ_n. Presupunem în plus că:

$$|\lambda_1| > |\lambda_2| \geq \cdots |\lambda_n| \geq 0$$

Metoda puterii este un algoritm care aproximează valoarea proprie de modul maxim $\boldsymbol{\lambda_1}$ şi un vector propriu asociat.

Se porneşte de la un vector nenul de normă euclidiană 1, $\boldsymbol{u}^{(0)} \in \mathbb{R}^{n}$, $\| \boldsymbol{u}^{(0)} \|_2 = 1$ şi se construieşte următorul şir de vectori de normă euclidiană ***1***:

$$\boldsymbol{u}^{(0)},\ \boldsymbol{u}^{(1)} = \frac{1}{\| A\boldsymbol{u}^{(0)} \|_2} A\boldsymbol{u}^{(0)},\ \ \boldsymbol{u}^{(2)} = \frac{1}{\| A\boldsymbol{u}^{(1)} \|_2} A\boldsymbol{u}^{(1)}, \dots,$$

$$\boldsymbol{u}^{(k)} = \frac{1}{\| A\boldsymbol{u}^{(k-1)} \|_2} A\boldsymbol{u}^{(k-1)}, \dots$$

În anumite condiţii acest şir converge la un vector propriu asociat valorii proprii $\boldsymbol{\lambda_1}$, iar coeficienţii Rayleigh corespunzători converg către $\boldsymbol{\lambda_1}$.

Teoremă

Fie $\boldsymbol{A} \in \mathbb{R}^{n \times n}, \boldsymbol{A} = \boldsymbol{A}^T$ o matrice simetrică pentru care valorile proprii îndeplinesc condiţia:

$$|\lambda_1| > |\lambda_2| \geq \cdots |\lambda_n| \geq 0.$$

Dacă $\boldsymbol{u}^{(0)} \in \mathbb{R}^n$, $\|\boldsymbol{u}^{(0)}\|_2 = 1$, $(\boldsymbol{u}^{(0)}, \boldsymbol{u}^1)_{\mathbb{R}^n} \neq 0$ ($\boldsymbol{u}^1$ vector propriu asociat lui λ_1) atunci:

$$\boldsymbol{u}^{(k)} = \frac{1}{\|\boldsymbol{A}^k \boldsymbol{u}^{(0)}\|_2} \boldsymbol{A}^k \boldsymbol{u}^{(0)} \rightarrow \boldsymbol{u}^1 \text{ (vector propriu asociat lui } \lambda_1 \text{)}$$

$$r(\boldsymbol{u}^{(k)}) \rightarrow \lambda_1$$

Demonstraţie. Fie $\{u^1, u^2, \ldots, u^n\}$ vectori proprii asociaţi valorilor proprii $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ care formează o bază ortonormată în $\mathbb{R}^n$. Avem:

$$u^{(0)} = a_1 u^1 + a_2 u^2 + \cdots + a_n u^n \ , \ a_i \in \mathbb{R}$$

Deoarece $(u^{(0)}, u^1)_{\mathbb{R}^n} \neq 0$ rezultă că $a_1 \neq 0$. Din construcţia şirului $u^{(k)}$ deducem că există o constantă c_k astfel ca:

$$
\begin{aligned}
\boldsymbol{u}^{(k)} &= c_k A^k \boldsymbol{u}^{(0)} = \\
&= c_k A^k (a_1 \boldsymbol{u}^1 + a_2 \boldsymbol{u}^2 + \cdots + a_n \boldsymbol{u}^n) = \\
&= c_k (a_1 \lambda_1^k \boldsymbol{u}^1 + a_2 \lambda_2^k \boldsymbol{u}^2 + \cdots + a_n \lambda_n^k \boldsymbol{u}^n) = \\
&= c_k \lambda_1^k \left[a_1 \boldsymbol{u}^1 + a_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k \boldsymbol{u}^2 + \cdots + a_n \left(\frac{\lambda_n}{\lambda_1} \right)^k \boldsymbol{u}^n \right]
\end{aligned}
$$

Din această ultimă relaţie, din faptul că λ_1 este valoare proprie dominantă şi $a_1 \neq 0$ deducem că pentru k suficient de mare vectorul $\boldsymbol{u}^{(k)}$ se aliniază după vectorul propriu $\boldsymbol{u}^1$:

$$
\boldsymbol{u}^{(k)} \approx c_k \lambda_1^k a_1 \boldsymbol{u}^1
$$

Metoda puterii

$$u^{(0)} \in \mathbb{R}^n \,,\ \| u^{(0)} \|_2 = 1;$$

$$k = 0;$$

do

- $k++;$
- $w = Au^{(k-1)};$
- $u^{(k)} = \dfrac{1}{\| w \|_2} w \,;$
- $\lambda_k = r(u^{(k)}) = \left(Au^{(k)}, u^{(k)} \right)_{\mathbb{R}^n} ;$

$$\textit{while}\,(\| Au^{(k)} - \lambda_k u^{(k)} \| > \varepsilon \text{ şi } k \le k_{\max});$$

Metoda iteraţiei inverse

Considerăm o matrice simetrică $\boldsymbol{A} \in \mathbb{R}^{\boldsymbol{n \times n}}$, $\boldsymbol{A} = \boldsymbol{A}^{\boldsymbol{T}}$ şi $\boldsymbol{\mu} \in \mathbb{R}$ un număr real care nu este valoare proprie a matricei ***A***. Vom folosi metoda puterii pentru a aproxima valoarea proprie a matricei ***A*** care este cea mai apropiată de $\boldsymbol{\mu}$ şi un vector propriu asociat.

$\boldsymbol{\mu} \neq$ valoare proprie $\rightarrow$ $\mathbf{det}(\boldsymbol{A} - \boldsymbol{\mu I_n}) \neq \mathbf{0}$ $\rightarrow$ $\exists\, (\boldsymbol{A} - \boldsymbol{\mu I_n})^{-1}$

Fie $\{\boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}, \ldots, \boldsymbol{\lambda_n}\}$ valorile proprii reale ale matricei ***A***.

Valorile proprii ale matricei $(A-\mu I_n)^{-1}$ sunt:

$$\left\{\frac{1}{(\lambda_1-\mu)},\frac{1}{(\lambda_2-\mu)},\ldots,\frac{1}{(\lambda_n-\mu)}\right\}$$

Matricele A şi $(A-\mu I_n)^{-1}$ au aceiași vectori proprii. Să presupunem că λ_I este valoarea proprie cea mai apropiată de μ (şi singura). Atunci:

$$\frac{1}{|\lambda_I-\mu|} > \frac{1}{|\lambda_j-\mu|} \quad \forall j \neq I$$

Această relaţie sugerează ideea aplicării metodei puterii matricei $\boldsymbol{(A - \mu I_n)^{-1}}$ pentru a aproxima valoarea proprie $\boldsymbol{(\lambda_I - \mu)^{-1}}$ şi a unui vector propriu asociat. Algoritmul duce la aproximarea valorii proprii cea mai apropiată de $\boldsymbol{\mu}$**,** $\boldsymbol{\lambda_I}$ şi a unui vector propriu asociat acestei autovalori, $\boldsymbol{u^I}$.

Metoda iteraţiei inverse

$u^{(0)} \in \mathbb{R}^n\ ,\ \| u^{(0)} \|_2 = 1;$

$k = 0;$

do

- $k++;$
- Se rezolvă sistemul $(A - \mu I_n) w = u^{(k-1)}$;
- $u^{(k)} = \dfrac{1}{\| w \|_2} w$;
- $\lambda_k = r(u^{(k)}) = \left(A u^{(k)}, u^{(k)} \right)_{\mathbb{R}^n}$;

while $(\| A u^{(k)} - \lambda_k u^{(k)} \| > \varepsilon$ şi $k \le k_{\max});$

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 $\boldsymbol{A}$ oarecare într-o matrice Hessenberg superioară $\boldsymbol{H}$ care să aibă aceleaşi valori proprii:

$\boldsymbol{A \to H}$ **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

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

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

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

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

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

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

Algoritmul de trecere de la matricea ***A*** la matricea ***P***

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 $\boldsymbol{A:=AP_r}$ fără a face înmulţire matricială (matricea $\boldsymbol{A}$ este cea obţinută mai sus având primele $\boldsymbol{r}$ coloane în formă superior Hessenberg).

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

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

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

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:

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

Operaţia ***A:=APr*** nu modifică primele ***r*** coloane ale matricii ***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 $\boldsymbol{(a_{r+1r} > 0)\ k = -k;}$
- $\boldsymbol{\beta = \sigma - k\, a_{r+1r};}$
- $\boldsymbol{u_{r+1} = a_{r+1r} - k; u_i = a_{ir}\ , i = r+2,\ldots,n\,;}$

$// \ \boldsymbol{A} = \boldsymbol{P}_r * \boldsymbol{A}$

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

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

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

// transformarea coloanei $\boldsymbol{r}$ a matricii $\boldsymbol{A}$

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

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

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

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

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

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:

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

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. Şirul $A^{(k)}$ se construieşte astfel:

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

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

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

$\vdots$

$A^{(k)} = Q_k R_k$ (descomp. QR 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 $A^{(k)}$ şi $A^{(k+1)}$ sunt asemenea:

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

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

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

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

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

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

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

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) $\Rightarrow$ **?**

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

Avem:

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

Notăm cu:

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

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 $\overline{\boldsymbol{R}}$ va avea primele ***p*** coloane în formă Hessenberg iar restul coloanelor în formă superior triunghiulară. Operaţia $\overline{\boldsymbol{R}} := \tilde{\boldsymbol{R}}\, \boldsymbol{R}^T_{pp+1}(\boldsymbol{\theta}_{pp+1})$ presupune doar schimbarea elementelor coloanelor ***p*** şi ***p+1***:

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

.

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

Observăm din relaţia de mai sus că în matricea $\bar{\boldsymbol{R}}$ 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 $\bar{\boldsymbol{H}} = \boldsymbol{A}^{(\boldsymbol{k}+1)}$ 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

- $\boldsymbol{A = \bar{Q}\, A\, \bar{Q}^T;}$
- $\boldsymbol{k = 0;}$
- while **(** $\boldsymbol{A} \neq$ forma Schur reală **)**

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

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:

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

Algoritmului *QR* cu deplasare (“*shift*”) simplă

Algoritmul cu deplasare simplă este următorul:

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

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

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

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

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

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

Rapiditatea cu care $a_{p+1\,p}^{(k)} \to 0, k \to \infty$ este dată de rata de convergenţa a expresiei $\left|\dfrac{\lambda_{p+1} - d}{\lambda_p - d}\right|^k$. Dacă se alege $d \approx \lambda_n$ convergenţa $a_{n-1n}^{(k)} \to 0$ este rapidă. Avem următoarul 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 ***QR*** cu deplasare simplă găseşte valoarea proprie ***d*** într-un singur pas. Euristic s-a constatat că la fiecare pas, cea mai bună aproximare a unei valori proprii este $a_{nn}^{(k)}$.

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

Algoritmul QR cu deplasare simplă

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

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

Algoritmului *QR* cu deplasare (“*shift*”) dublă

În cazul când valorile proprii $\boldsymbol{a_1}$, $\boldsymbol{a_2}$ corespunzătoare blocului:

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

sunt complexe, $\boldsymbol{a_1, a_2} \in \mathbb{C}$, 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 :

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

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 Q care asigură trecerea de la matricea A la matricea A_2 este matricea ortogonală din descompunerea QR a matricii $M = \left(A - a_2\ I_n\right)\left(A - a_1\ I_n\right)$. Pasul 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

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

2) se calculează descompunerea ***QR*** a matricii ***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, $\boldsymbol{p_A(\lambda) \equiv p_B(\lambda)}$, 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:

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

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

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

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)

/ / se aduce matricea *A* la forma Hessenberg

$$\bullet\ A = \bar{Q}\, A\, \bar{Q}^T;$$

$$\bullet\ U = \bar{Q}^T;$$

$$\bullet\ k = 0;$$

$$\bullet\ \textbf{while}\ (\ A \neq \textbf{matrice diagonală}\ )$$

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