Calcul Numeric

Cursul 5

2022

Anca Ignat

Algoritmul de eliminare Gauss fără schimbare de ecuații $\Leftrightarrow$ descompunere LU

Presupunem că la fiecare pas al algoritmului de eliminare Gauss pivotul este nenul ($a_{rr}^{(r-1)} \neq 0$), deci nu e nevoie de schimbare de ecuații.

Algoritmul se poate scrie astfel:

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

$$\quad \textbf{for } i = r+1, \ldots, n$$

$$\bullet\; f = -\frac{a_{ir}}{a_{rr}};$$

$$// \; E_i = E_i + f * E_r$$

$$\bullet\; \textbf{for } j = r+1, \ldots, n$$

$$\qquad a_{ij} = a_{ij} + f * a_{rj};$$

$$\bullet\; a_{ir} = 0;$$

$$\bullet\; b_i = b_i + f * b_r;$$

Considerăm vectorul și matricea:

$$\boldsymbol{t}^{(r)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ t_{r+1}^{(r)} \\ \vdots \\ t_n^{(r)} \end{pmatrix} \in \mathbb{R}^n \quad , \quad \boldsymbol{T}_r := \boldsymbol{I}_n + \boldsymbol{t}^{(r)} \boldsymbol{e}_r^T \in \mathbb{R}^{n \times n}$$

$$\boldsymbol{t}^{(r)}\boldsymbol{e}_r^T = \begin{pmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \boldsymbol{t}_{r+1}^{(r)} \\ \vdots \\ \boldsymbol{t}_n^{(r)} \end{pmatrix} \begin{pmatrix} \mathbf{0} & \cdots & \mathbf{1} & \mathbf{0} \cdots \mathbf{0} \end{pmatrix} = \begin{array}{cc} \begin{array}{cccccc} \phantom{\mathbf{0}} & \phantom{\mathbf{0}} & & \text{col r} & & \phantom{\mathbf{0}} \end{array} & \\ \begin{pmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & & & & & \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{t}_{r+1}^{(r)} & \cdots & \mathbf{0} \\ \vdots & & & & & \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{t}_n^{(r)} & \cdots & \mathbf{0} \end{pmatrix} & \begin{matrix} \\ \\ \\ -\,\mathrm{lin}\,(\boldsymbol{r}+\mathbf{1}) \\ \\ \\ \end{matrix} \end{array}$$

Matricea $\boldsymbol{T_r}$ este matrice triunghiulară inferior cu ***1*** pe diagonala principală:

$$\boldsymbol{T_r} = \begin{array}{c} \begin{array}{cccccc} & & & \textbf{col } \boldsymbol{r} & & \end{array} \\ \begin{pmatrix} \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & & & & & \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{t_{r+1}^{(r)}} & \cdots & \mathbf{0} \\ \vdots & & & & & \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{t_{n}^{(r)}} & \cdots & \mathbf{1} \end{pmatrix} \end{array}$$

Inversa matricei $\boldsymbol{T_r}$ este

$$\boldsymbol{T_r^{-1} = I_n - t^{(r)} e_r^T}\,.$$

$$\boldsymbol{T_r\, T_r^{-1} = (I_n + t^{(r)} e_r^T)(I_n - t^{(r)} e_r^T) = I_n - t^{(r)} e_r^T + t^{(r)} e_r^T - t^{(r)} e_r^T t^{(r)} e_r^T =}$$

$$\boldsymbol{= I_n - t^{(r)}\, t_r^{(r)} e_r^T = I_n \quad (0 = t_r^{(r)} = e_r^T t^{(r)})}$$

Dacă $\boldsymbol{A}$ este o matrice oarecare, vrem să vedem cum se poate construi matricea $\boldsymbol{B=T_r\, A}$ fără a face înmulțire matricială. Vom studia legătura între liniile matricelor $\boldsymbol{A}$ și $\boldsymbol{B}$.

$$\boldsymbol{e}_i^T \boldsymbol{B} = \boldsymbol{e}_i^T (\boldsymbol{T}_r \boldsymbol{A}) = \boldsymbol{e}_i^T (\boldsymbol{I}_n + \boldsymbol{t}^{(r)} \boldsymbol{e}_r^T) \boldsymbol{A} = \boldsymbol{e}_i^T \boldsymbol{A} + \boldsymbol{e}_i^T \boldsymbol{t}^{(r)} \boldsymbol{e}_r^T \boldsymbol{A} =$$
$$= \boldsymbol{e}_i^T \boldsymbol{A} + t_i^{(r)} (\boldsymbol{e}_r^T \boldsymbol{A})$$

Linia $\boldsymbol{i}$ a noii matrice $\boldsymbol{B}$ se obține din linia $\boldsymbol{i}$ a matricei $\boldsymbol{A}$ la care se adaugă linia $\boldsymbol{r}$ a matricei $\boldsymbol{A}$ înmulțită cu factorul $t_i^{(r)}$.

$$\boldsymbol{e}_i^T \boldsymbol{B} = \begin{cases} \boldsymbol{e}_i^T \boldsymbol{A} & i = 1,\ldots,r \ (t_i^{(r)} = 0) \\ \boldsymbol{e}_i^T \boldsymbol{A} + t_i^{(r)} (\boldsymbol{e}_r^T \boldsymbol{A}) & i = r+1,\ldots,n \end{cases}.$$

Operația $\boldsymbol{T_r A}$ descrie Pasul $\boldsymbol{r}$ al algoritmului de eliminare Gauss dacă:

$$t_{r+1}^{(r)} = -\frac{a_{r+1r}^{(r)}}{a_{rr}^{(r)}}, \cdots, t_i^{(r)} = -\frac{a_{ir}^{(r)}}{a_{rr}^{(r)}}, \cdots, t_n^{(r)} = -\frac{a_{nr}^{(r)}}{a_{rr}^{(r)}}.$$

Algoritmul de eliminare Gauss fără schimbare de ecuații poate fi descris astfel:

$$T_{n-1} \cdots T_2 T_1 A = U \quad \text{cu} \quad T_r = I_n + t^{(r)} e_r^T,$$

$$t^{(r)} = \left(0 \ \cdots 0 \ \left(-\frac{a_{r+1r}^{(r)}}{a_{rr}^{(r)}}\right) \ \cdots \left(-\frac{a_{ir}^{(r)}}{a_{rr}^{(r)}}\right) \ \cdots \left(-\frac{a_{nr}^{(r)}}{a_{rr}^{(r)}}\right) \right)^T.$$

Avem:

$$A = T_1^{-1}T_2^{-1}\cdots T_{n-1}^{-1}\,U = LU \quad , \quad L := T_1^{-1}T_2^{-1}\cdots T_{n-1}^{-1}$$

$$T_1^{-1}T_2^{-1} = (I_n - t^{(1)}e_1^T)(I_n - t^{(2)}e_2^T) = I_n - t^{(1)}e_1^T - t^{(2)}e_2^T + t^{(1)}e_1^T t^{(2)}e_2^T =$$
$$= I_n - t^{(1)}e_1^T - t^{(2)}e_2^T + t^{(1)}t_1^{(2)}e_2^T = I_n - t^{(1)}e_1^T - t^{(2)}e_2^T \quad (t_1^{(2)} = 0)$$

Prin inducție se arată că:

$$L = T_1^{-1}T_2^{-1}\cdots T_{n-1}^{-1} = I_n - t^{(1)}e_1^T - t^{(2)}e_2^T - \cdots - t^{(n-1)}e_{n-1}^T$$

$$L=\begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ \frac{a_{21}}{a_{11}} & 1 & \cdots & 0 & \cdots & 0 \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}^{(1)}}{a_{22}^{(1)}} & \cdots & 0 & \cdots & 0 \\ \vdots & & & & & \\ \frac{a_{r1}}{a_{11}} & \frac{a_{r2}^{(1)}}{a_{22}^{(1)}} & \cdots & 1 & \cdots & 0 \\ \frac{a_{r+11}}{a_{11}} & \frac{a_{r+12}^{(1)}}{a_{22}^{(1)}} & \cdots & \frac{a_{r+1r}^{(r-1)}}{a_{rr}^{(r-1)}} & \cdots & 0 \\ \vdots & & & & & \\ \frac{a_{n1}}{a_{11}} & \frac{a_{n2}^{(1)}}{a_{22}^{(1)}} & \cdots & \frac{a_{nr}^{(r-1)}}{a_{rr}^{(r-1)}} & \cdots & 1 \end{pmatrix}$$

Descompuneri *QR*

Definiție

Se numește ***matrice ortogonală***, o matrice $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ care satisface relația:

$$\boldsymbol{Q^T Q = Q Q^T = I_n} \quad (\boldsymbol{Q^{-1} = Q^T}) \, .$$

Matricele ortogonale au următoarele proprietăți:

- Dacă $\boldsymbol{Q}$ este matrice ortogonală atunci şi matricea transpusă $\boldsymbol{Q^T}$ este ortogonală.

$$\boldsymbol{Q^T Q = Q^T (Q^T)^T = Q Q^T = (Q^T)^T Q^T = I_n}$$

- Dacă Q_1 şi Q_2 sunt matrice ortogonale atunci Q_1Q_2 este tot matrice ortogonală.

$$(Q_1Q_2)^T(Q_1Q_2) = Q_2^TQ_1^TQ_1Q_2 = Q_2^T I\, Q_2 = I_n$$

$$(Q_1Q_2)(Q_1Q_2)^T = Q_1Q_2Q_2^TQ_1^T = Q_1^T I\, Q_1 = I_n$$

- Dacă $Q \in \mathbb{R}^{n\times n}$ este matrice ortogonală şi $x \in \mathbb{R}^n$ atunci $\|Qx\|_2 = \|x\|_2$.

$$\|Qx\|_2^2 = \left(Qx, Qx\right) = \left(x, Q^TQx\right) = \left(x, x\right) = \|x\|_2^2, \quad \|\cdot\|_2 \geq 0 \Rightarrow$$

$$\|Qx\|_2 = \|x\|_2$$

Fie $\boldsymbol{A}$ o matrice reală pătratică de dimensiune $\boldsymbol{n}$. Pp. că avem:

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$$

unde $\boldsymbol{Q}$ este o matrice ortogonală iar $\boldsymbol{R}$ este o matrice superior triunghiulară.

$$\boldsymbol{Ax} = \boldsymbol{b} \quad \Leftrightarrow \quad \boldsymbol{QRx} = \boldsymbol{b} \quad \Leftrightarrow \quad \boldsymbol{Q}^T\boldsymbol{QRx} = \boldsymbol{Q}^T\boldsymbol{b} \quad \Leftrightarrow \quad \boldsymbol{Rx} = \boldsymbol{Q}^T\boldsymbol{b}$$

Algoritmul lui Householder

***Matrice de reflexie* - $P \in \mathbb{R}^{n\times n}$** de forma:

$$P = I_n - 2vv^T \ , \quad v \in \mathbb{R}^n \ , \quad \| v \|_2 = \sqrt{\sum_{j=1}^{n} | v_j |^2} = 1$$

$$vv^T = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} (v_1, v_2, \ldots, v_n) = \begin{pmatrix} v_1^2 & v_1 v_2 & \ldots & v_1 v_n \\ v_2 v_1 & v_2^2 & \ldots & v_2 v_n \\ \vdots & & & \\ v_n v_1 & v_n v_2 & \ldots & v_n^2 \end{pmatrix}.$$

Matricele de reflexie sunt :

simetrice - $\boldsymbol{P = P^T}$ şi

ortogonale - $\boldsymbol{PP^T = P^T P = P^2 = I_n}$.

$$\boldsymbol{P^T = (I_n - 2vv^T)^T = I_n - 2(vv^T)^T = I_n - 2(v^T)^T v^T = I_n - 2vv^T = P}$$

$$P^2 = (I_n - 2vv^T)(I_n - 2vv^T) = I_n - 2vv^T - 2vv^T + 4(vv^T)(vv^T) =$$

$$= I_n - 4vv^T + 4v(v^T v)v^T = I_n - 4vv^T + 4v\|v\|_2^2 v^T =$$

$$= I_n - 4vv^T + 4vv^T = I_n \quad (\|v\|_2 = 1)$$

$$n = 2, \quad v = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - 2\begin{pmatrix} 1 \\ 0 \end{pmatrix}(1 \quad 0) = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \; x \in \mathbb{R}^2, \; y = Px$$

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -x_1 \\ x_2 \end{pmatrix}$$

Vectorul $y=Px$ este reflectatul vectorului x în raport cu axa Ox_2.

Algoritmul care folosește matricele de reflexie pentru a obține o descompunere $\boldsymbol{QR}$ pentru o matrice $\boldsymbol{A} \in \mathbb{R}^{\boldsymbol{n \times n}}$ a fost descris de Alston S. Householder în articolul "*Unitary triangularization of a nonsymmetric matrix*" apărut în Journal of the Assoc. of Computing Machinery 5 (1958), 339-342.

Transformarea matricei $\boldsymbol{A}$ într-una superior triunghiulară se face în ***(n-1)*** pași, la fiecare pas folosindu-se o matrice de reflexie.

Pas 1: $\boldsymbol{A^{(1)}} = \boldsymbol{P_1 A}$ (matricea $\boldsymbol{P_1}$ se alege astfel încât col. ***1*** să fie transformată în formă superior triunghiulară)

Pas 2: $\boldsymbol{A^{(2)}} = \boldsymbol{P_2 A^{(1)}} = \boldsymbol{P_2 (P_1 A)}$ ($\boldsymbol{P_2}$ transformă col. ***2*** în formă superior triunghiulară, fără să schimbe col. ***1***)

Pas *r*: $\boldsymbol{A^{(r)}} = \boldsymbol{P_r A^{(r-1)}} = \boldsymbol{P_r (P_{r-1} \ldots P_1 A)}$ (se transformă col ***r*** în formă superior triunghiular fără să schimbe primele ***(r-1)*** coloane)

Descompunerea $\boldsymbol{QR}$ construită cu algoritmul Householder este următoarea:

$$\boldsymbol{P_{n-1}\cdots P_r\cdots P_2P_1A=\tilde{Q}A=R}$$

unde

$$\boldsymbol{\tilde{Q}=P_{n-1}\cdots P_r\cdots P_2P_1}$$

$\boldsymbol{\widetilde{Q}}$ este matrice ortogonală ca produs de matrice ortogonale.

$$\boldsymbol{\tilde{Q}A=R \quad\Leftrightarrow\quad \tilde{Q}^TQA=\tilde{Q}^TR \quad\Leftrightarrow\quad A=\tilde{Q}^TR=QR}$$

$$\boldsymbol{Q=\tilde{Q}^T=(P_{n-1}\cdots P_r\cdots P_2P_1)^T=P_1P_2\cdots P_r\cdots P_{n-1}}$$

Pasul *r*

La intrarea în pasul $\boldsymbol{r}$ matricea $\boldsymbol{A}$ are forma:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1r} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2r} & \cdots & a_{2n} \\ \vdots & & & & & \\ 0 & 0 & \cdots & a_{rr} & \cdots & a_{rn} \\ 0 & 0 & \cdots & a_{r+1r} & \cdots & a_{r+1n} \\ \vdots & & & & & \\ 0 & 0 & \cdots & a_{ir} & \cdots & a_{in} \\ \vdots & & & & & \\ 0 & 0 & \cdots & a_{nr} & \cdots & a_{nn} \end{pmatrix}$$

Pasul $\boldsymbol{r}$ constă în:

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

unde vectorul $\boldsymbol{v}^{\,\boldsymbol{r}}$ se alege astfel ca matricea $\boldsymbol{A}$ să aibă şi coloana $\boldsymbol{r}$ în formă superior triunghiulară:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1r} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2r} & \cdots & a_{2n} \\ \vdots & & & & & \\ 0 & 0 & \cdots & a_{rr} & \cdots & a_{rn} \\ 0 & 0 & \cdots & 0 & \cdots & a_{r+1n} \\ \vdots & & & & & \\ 0 & 0 & \cdots & 0 & \cdots & a_{in} \\ \vdots & & & & & \\ 0 & 0 & \cdots & 0 & \cdots & a_{nn} \end{pmatrix}$$

Calculul matricei $\boldsymbol{P_r}$

Pentru simplitate vom nota $\boldsymbol{P_r = P}$, $\boldsymbol{v^r = v}$.

$$Ae_r = \begin{pmatrix} a_{1r} \\ a_{2r} \\ \vdots \\ a_{r-1r} \\ a_{rr} \\ a_{r+1r} \\ \vdots \\ a_{ir} \\ \vdots \\ a_{nr} \end{pmatrix} \rightarrow (PA)e_r = \bar{A}e_r = \begin{pmatrix} \bar{a}_{1r} = a_{1r} \\ \bar{a}_{2r} = a_{2r} \\ \vdots \\ \bar{a}_{r-1r} = a_{r-1r} \\ \bar{a}_{rr} = k \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Aplicând proprietatea matricelor ortogonale :

$$Q \in \mathbb{R}^{n\times n},\ x \in \mathbb{R}^{n},\ \| Qx \|_2 = \| x \|_2$$

pentru matricea $Q=P$ şi vectorul $x=Ae_r$ avem:

$$\| PAe_r \|_2^2 = a_{1r}^2 + a_{2r}^2 + \cdots + a_{r-1r}^2 + k^2 =$$

$$\| Ae_r \|^2 = a_{1r}^2 + a_{2r}^2 + \cdots + a_{r-1r}^2 + a_{rr}^2 + a_{r+1r}^2 + \cdots + a_{ir}^2 + \cdots + a_{nr}^2$$

Din relația de mai sus rezultă:

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

Determinarea vectorului ***v*** ce definește matricea ***P:***

$$(\boldsymbol{PA})\boldsymbol{e}_r = (\boldsymbol{I}_n - 2\boldsymbol{v}\boldsymbol{v}^T)(\boldsymbol{A}\boldsymbol{e}_r) = \boldsymbol{A}\boldsymbol{e}_r - 2(\boldsymbol{v}\boldsymbol{v}^T)(\boldsymbol{A}\boldsymbol{e}_r) =$$

$$= \boldsymbol{A}\boldsymbol{e}_r - 2\boldsymbol{v}(\boldsymbol{v}^T(\boldsymbol{A}\boldsymbol{e}_r)) = \boldsymbol{A}\boldsymbol{e}_r - (2\alpha)\boldsymbol{v} = \boldsymbol{A}\boldsymbol{e}_r - \boldsymbol{u}$$

unde cu α și ***u*** am notat:

$$\alpha := v^T(Ae_r) = \big((Ae_r), v\big)_{\mathbb{R}^n} = \left(\begin{pmatrix} a_{1r} \\ a_{2r} \\ \vdots \\ a_{ir} \\ \vdots \\ a_{nr} \end{pmatrix}, \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_i \\ \vdots \\ v_n \end{pmatrix} \right)_{\mathbb{R}^n} =$$

$$= a_{1r}v_1 + a_{2r}v_2 + \cdots + a_{ir}v_i + \cdots + a_{nr}v_n$$

$$u := (2\alpha)\, v = Ae_r - (PA)e_r = \begin{pmatrix} a_{1r} \\ a_{2r} \\ \vdots \\ a_{r-1r} \\ a_{rr} \\ a_{r+1r} \\ \vdots \\ a_{ir} \\ \vdots \\ a_{nr} \end{pmatrix} - \begin{pmatrix} a_{1r} \\ a_{2r} \\ \vdots \\ a_{r-1r} \\ k \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ a_{rr} - k \\ a_{r+1r} \\ \vdots \\ a_{ir} \\ \vdots \\ a_{nr} \end{pmatrix}$$

Cu aceste notații, matricea $\boldsymbol{P}$ devine:

$$\boldsymbol{P} = \boldsymbol{I}_n - 2\left(\frac{1}{2\alpha}\right)\boldsymbol{u}\left(\frac{1}{2\alpha}\right)\boldsymbol{u}^T = \boldsymbol{I}_n - \left(\frac{1}{2\alpha^2}\right)\boldsymbol{u}\boldsymbol{u}^T = \boldsymbol{I}_n - \frac{1}{\beta}\boldsymbol{u}\boldsymbol{u}^T$$

$$\beta := 2\alpha^2$$

Pentru a cunoaște matricea $\boldsymbol{P}$ trebuie să mai determinăm constanta β. Din condiția:

$$\|\boldsymbol{v}\|_2^2 = 1 \Rightarrow \left\|\frac{1}{2\alpha}\boldsymbol{u}\right\|_2^2 = 1 \Rightarrow \frac{1}{4\alpha^2}\|\boldsymbol{u}\|_2^2 = 1 \Rightarrow 2\beta = \|\boldsymbol{u}\|_2^2$$

$$\| u \|_2^2 = \left\| \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ a_{rr} - k \\ a_{r+1r} \\ \vdots \\ a_{ir} \\ \vdots \\ a_{nr} \end{pmatrix} \right\|_2^2 = (a_{rr} - k)^2 + a_{r+1r}^2 + \cdots + a_{ir}^2 + \cdots + a_{nr}^2 =$$

$$= a_{rr}^2 + a_{r+1r}^2 + \cdots + a_{ir}^2 + \cdots + a_{nr}^2 - 2ka_{rr} + k^2 =$$

$$= \sigma - 2ka_{rr} + \sigma = 2(\sigma - ka_{rr})$$

de unde obținem:

$$\boldsymbol{\beta = \sigma - k a_{rr}}$$

Vom alege semnul constantei $\boldsymbol{k}$ astfel încât $\boldsymbol{\beta}$ să fie cât mai mare posibil deoarece constanta $\boldsymbol{\beta}$ apare în operația de împărțire. Avem:

$$\boldsymbol{\beta \quad \text{"}\mathit{mare}\text{"} \;\rightarrow\; \beta = \sigma - k a_{rr} \geq \sigma \quad (\sigma \geq 0) \quad \rightarrow}$$

$$\mathbf{semn}\, \boldsymbol{k} = -\mathbf{semn}\, \boldsymbol{a_{rr}}$$

Ce înseamnă $\boldsymbol{\beta = 0}$?

$$\boldsymbol{\beta} = \frac{1}{2}\|\boldsymbol{u}\|_2^2 = 0 \to \|\boldsymbol{u}\|_2^2 = 0 \to \boldsymbol{u} = 0 \to$$

$$\to \boldsymbol{a_{rr}} = \boldsymbol{k}, \boldsymbol{a_{r+1r}} = 0, \ldots, \boldsymbol{a_{ir}} = 0, \ldots, \boldsymbol{a_{nr}} = 0$$

Cum $\boldsymbol{a_{rr}}=\boldsymbol{k}$ şi **semn** $\boldsymbol{k}$ **= -semn** $\boldsymbol{a_{rr}}$ obținem:

$$\boldsymbol{a_{ir}} = 0, \quad \forall \boldsymbol{i} = \boldsymbol{r}, \ldots, \boldsymbol{n}$$

adică avem coloana $\boldsymbol{r}$ deja în formă superior triunghiulară, se poate trece la pasul următor. În acest caz matricea $\boldsymbol{A}$ este singulară.

Ne interesează cum se efectuează operația ***A=P_rA*** fără a face înmulțire matricială. Vom pune în evidentă schimbările în raport cu coloanele.

$$(PA)e_j = \text{ noua col. } j \text{ a matr. } A = (I_n - \frac{1}{\beta}uu^T)(Ae_j) =$$

$$= Ae_j - \frac{1}{\beta}(uu^T)(Ae_j) = Ae_j - \frac{1}{\beta}u(u^T(Ae_j)) =$$

$$= Ae_j - \frac{\gamma_j}{\beta}u$$

$$\gamma_j := u^T(Ae_j) = (\begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{rj} \\ a_{r+1j} \\ \vdots \\ a_{nj} \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \vdots \\ a_{rr} - k \\ a_{r+1r} \\ \vdots \\ a_{nr} \end{pmatrix})_{\mathbb{R}^n} =$$

$$= a_{rj}(a_{rr} - k) + \cdots + a_{ij}a_{ir} + \cdots + a_{nj}a_{nr} = \sum_{i=1}^{n} u_i a_{ij} = \sum_{i=r}^{n} u_i a_{ij}$$

$$(u_i = 0, i = 1, \ldots, r-1, u_r = a_{rr} - k, u_i = a_{ir}, i = r+1, \ldots, n)$$

Noua coloană ***j*** se obține din vechea coloană ***j*** din care scădem vectorul ***u*** înmulțit cu constanta $\boldsymbol{\gamma_j}$. Ne interesează ca primelor ***(r-1)*** coloane să nu li se schimbe forma superior triunghiulară deja obținută.

Pentru ***j=1,…,(r-1)*** avem:

$$\gamma_j := u^T(Ae_j) = \left(\begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{jj} \\ a_{j+1j} = 0 \\ \vdots \\ a_{rj} = 0 \\ \vdots \\ a_{nj} = 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ a_{rr} - k \\ a_{r+1r} \\ \vdots \\ a_{nr} \end{pmatrix} \right)_{\mathbb{R}^n} =$$

$$= a_{1j}0 + \cdots + a_{jj}0 + \cdots + 0(a_{rr} - k) + \cdots + 0a_{ir} + \cdots + 0a_{nr} = 0$$

Din faptul că $\gamma_j = 0,\ j = 1,\ldots,r-1$ rezultă că primele ***(r-1)*** coloane ale matricei $\boldsymbol{A}$ nu se schimbă când facem operația $\boldsymbol{A}=\boldsymbol{P_r A}$, rămân în formă superior triunghiulară.

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

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

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

$$u_i = 0, \quad i = 1,\ldots,r-1, \quad u_r = a_{rr} - k, \qquad u_i = a_{ir}, \quad i = r+1,\ldots,n$$

Operaţia de transformare a vectorului termenilor liberi

$b := P_r b$:

$$P_r b = (I_n - \frac{1}{\beta}(uu^T))b = b - \frac{1}{\beta}(uu^T)b = b - \frac{1}{\beta}u(u^T b) = b - \frac{\gamma}{\beta}u$$

$$\gamma = u^T b = (b, u)_{\mathbb{R}^n} = \sum_{i=r}^{n} u_i b_i$$

Algoritmul lui Householder

$$\widetilde{Q} = I_n;$$

$$\textbf{for } r = 1, \cdots, n-1$$

$$\begin{cases} \textbf{calculează matricea } P_r \textbf{ (constanta } \beta \textbf{ şi vectorul } u\textbf{)} \\ A = P_r * A; \\ b = P_r * b; \\ \widetilde{Q} = P_r * \widetilde{Q}; \end{cases}$$

La sfârșitul acestui algoritm, în matricea A vom avea matricea superior triunghiulară R, în vectorul b vom avea $Q^T b^{\mathrm{init}}$, b^{init} este vectorul inițial al termenilor liberi, iar matricea $\widetilde{Q}$ va conține matricea Q^T din factorizarea QR a matricei A.

Algoritmul Householder detaliat

$\widetilde{Q} = I_n;$

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

// **construcţia matricii** P_r – **constanta** β **şi vectorul** u

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

$// \; A = P_r * A$

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

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

$$* \gamma = (\gamma_j / \beta) = (Ae_j, u)/\beta = \left(\sum_{i=r}^{n} u_i a_{ij}\right)/\beta;$$

$$* \text{for } i = r, \ldots, n$$

$$a_{ij} = a_{ij} - \gamma * u_i;$$

$//$ **transformarea coloanei** r **a matricii** A

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

$// \; b = P_r * b$

- $\gamma = (\gamma / \beta) = (b, u)/\beta = \left(\sum_{i=r}^{n} u_i b_i\right)/\beta;$

- **for** $i = r, \ldots, n \quad b_i = b_i - \gamma * u_i;$

$$//\ \widetilde{Q} = P_r * \widetilde{Q}$$

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

$$* \ \gamma = (\widetilde{Q}e_j, u) / \beta = \Big(\sum_{i=r}^{n} u_i \tilde{q}_{ij}\Big) / \beta;$$

$$* \ \textbf{for} \ \ i = r, \ldots, n$$

$$\tilde{q}_{ij} = \tilde{q}_{ij} - \gamma * u_i;$$

Numărul de operații efectuate:

A (adunări, scăderi):

$$(n-1)+2n(n-1)+\frac{n(n-1)(2n-1)}{3}=\frac{(n-1)(n+1)(2n+3)}{3}=$$

$$=\frac{2}{3}n^3+O(n^2)$$

M (înmulțiri, împărțiri):

$$4(n-1)+3n(n-1)+\frac{n(n-1)(2n-1)}{6}=\frac{2}{3}n^3+O(n^2)$$

R (radicali): ***(n-1)***

Algoritmul lui Givens

Fie A o matrice reală pătratică de dimensiune n. Pp. că avem:

$$A = QR$$

unde Q este o matrice ortogonală iar R este o matrice superior triunghiulară.

$$Ax = b \quad \Leftrightarrow \quad QRx = b \Leftrightarrow Q^T QRx = Q^T b \quad \Leftrightarrow Rx = Q^T b$$

În cazul algoritmului Givens, pentru a aduce sistemul $Ax=b$ la forma $Rx = Q^T b$ se folosesc matricele de rotație. O matrice de rotație $R_{pq}(\theta) = (r_{ij})_{i,j=1,n}$ are următoarea formă :

$$
R_{pq}(\theta) =
\begin{array}{c}
\begin{array}{cccccccc}
 & & & p & & q & &
\end{array} \\
\begin{pmatrix}
1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
\vdots & & & & & & & \\
0 & 0 & \cdots & c & \cdots & s & \cdots & 0 \\
\vdots & & & & 1 & & & \\
0 & 0 & \cdots & (-s) & \cdots & c & \cdots & 0 \\
\vdots & & & & & & & \\
0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1
\end{pmatrix}
\end{array}
\begin{array}{c}
\\ \\ \\ \\ p \\ \\ q \\ \\ \\
\end{array}
$$

$$r_{ij} = \begin{cases} \mathbf{1} & \textbf{pentru } \boldsymbol{i = j, \quad i \neq p, i \neq q} \\ \boldsymbol{c} & \textbf{pentru } \boldsymbol{i = j, \quad i = p, i = q} \\ \boldsymbol{s} & \textbf{pentru } \boldsymbol{i = p, j = q} \\ \boldsymbol{-s} & \textbf{pentru } \boldsymbol{i = q, j = p} \\ \mathbf{0} & \text{în rest} \end{cases}$$

unde $\boldsymbol{p, q \in \{1,...,n\}}$ iar $\boldsymbol{c}$ şi $\boldsymbol{s}$ sunt două numere reale care satisfac relația $\boldsymbol{c^2+s^2=1}$. Constantele $\boldsymbol{c}$ şi $\boldsymbol{s}$ pot fi alese astfel încât $\boldsymbol{c=\cos\theta, s=\sin\theta}$. Se arată uşor, folosind relația $\boldsymbol{c^2+s^2=1}$, că matricea $\boldsymbol{R_{pq}(\theta)}$ este ortogonală:

$$\boldsymbol{R_{pq}(\theta)\, R_{pq}^T(\theta) = R_{pq}^T(\theta) R_{pq}(\theta) = I_n}$$

Calculul matricei:

$$B = R_{pq}(\theta)A\ ,$$

B se obține din A modificând doar liniile p şi q.
Fie

$$A_i = e_i^T A, \text{ - linia } i \text{ a matricei } A$$

$$B_i = e_i^T B \quad \text{ - linia } i \text{ a matricei } B.$$

Liniile matricei B:

$$B_i = A_i \ , \quad i = 1,\ldots,n\ , \quad i \neq p\ , i \neq q$$

$$B_p = cA_p + sA_q$$

$$B_q = -sA_p + cA_q$$

$$b_{pj} = c\,a_{pj} + s\,a_{qj} \;\;,\;\; j = 1,\dots,n$$

$$b_{qj} = -\,s\,a_{pj} + c\,a_{qj} \;\;,\; j = 1,\dots,n$$

$$b_{ij} = a_{ij} \quad \textbf{în rest}$$

Calculul matricei :

$$D = A\,R^T_{pq}(\theta)\,,$$

D se obține din ***A*** modificând doar coloanele ***p*** și ***q***.

Notăm $\boldsymbol{Ae_j}$, $\boldsymbol{De_j}$ – coloana j a matricei A şi respectiv D.

Coloanele matricei ***D***:

$$
\begin{aligned}
&D_j = A_j \;,\quad j = 1,\ldots,n\,,\quad j \neq p\,,\, j \neq q\\
&De_p = cAe_p + sAe_q\\
&De_q = -sAe_p + cAe_q\\
&d_{ip} = c\,a_{ip} + s\,a_{iq}\;,\; i = 1,\ldots,n\\
&d_{iq} = -\,s\,a_{ip} + c\,a_{iq}\;,\, i = 1,\ldots,n\\
&d_{ij} = a_{ij}\quad \text{în rest}
\end{aligned}
$$

Algoritmul lui Givens se desfăşoară în ***(n-1)*** paşi - la pasul ***r*** se transformă coloana ***r*** a matricei ***A*** în formă superior triunghiulară fără a modifica primele ***(r-1)*** coloane.

Pasul 1

Intrare: matricea $\boldsymbol{A}$, vectorul $\boldsymbol{b}$

Ieșire: matricea $\boldsymbol{A}^{(1)}$ (cu prima coloană în formă superior triunghiulară), $\boldsymbol{b}^{(1)}$

Se efectuează următoarele operații de înmulțire cu matrice de rotație:

$$\boldsymbol{R}_{1n}(\boldsymbol{\theta}_{1n})\cdots \boldsymbol{R}_{13}(\boldsymbol{\theta}_{13})\boldsymbol{R}_{12}(\boldsymbol{\theta}_{12})\,\boldsymbol{A} = \boldsymbol{A}^{(1)}$$

$$\boldsymbol{R}_{1n}(\boldsymbol{\theta}_{1n})\cdots \boldsymbol{R}_{13}(\boldsymbol{\theta}_{13})\boldsymbol{R}_{12}(\boldsymbol{\theta}_{12})\,\boldsymbol{b} = \boldsymbol{b}^{(1)}$$

Unghiurile $\boldsymbol{\theta_{1i}}$ (constantele $\boldsymbol{c_{1i}}$ și $\boldsymbol{s_{1i}}$) se aleg astfel ca elementul de pe poziția ***(i,1)*** din matricea rezultat să devină ***0***.

Pasul r

Intrare: matricea $\boldsymbol{A^{(r-1)}}$ (are primele ***(r-1)*** coloane în formă superior triunghiulară), $\boldsymbol{b^{(r-1)}}$

Ieșire: matricea $\boldsymbol{A^{(r)}}$ (cu primele ***r*** coloane în formă superior triunghiulară), $\boldsymbol{b^{(r)}}$

La acest pas matricea $\boldsymbol{A^{(r-1)}}$ şi vectorul $\boldsymbol{b^{(r-1)}}$ se transformă astfel:

$$\boldsymbol{R_{r\,n}(\theta_{r\,n})\cdots R_{r\,i}(\theta_{r\,i})\cdots R_{r\,r+1}(\theta_{r\,r+1})\,A^{(r-1)} = A^{(r)}}$$

$$\boldsymbol{R_{r\,n}(\theta_{r\,n})\cdots R_{r\,i}(\theta_{r\,i})\cdots R_{r\,r+1}(\theta_{r\,r+1})\,b^{(r-1)} = b^{(r)}}$$

unde elementele $\boldsymbol{c = c_{r\,i}}$ şi $\boldsymbol{s = s_{r\,i}}$ din matricele de rotație se aleg astfel ca după înmulțirea cu $\boldsymbol{R_{r\,i}(\theta_{r\,i})\,, i = r+1,\ldots,n}$ elementul ***(i,r)*** să devină ***0***.
Considerăm operația $\boldsymbol{B = R_{r\,i}(\theta_{r\,i})A}$, unde $\boldsymbol{\theta_{r\,i}}$ se alege astfel ca $\boldsymbol{b_{i\,r} = 0}$:

$$\boldsymbol{b_{rj} = c\,a_{rj} + s\,a_{ij}} \;\; ,$$

$$\boldsymbol{b_{ij} = -\,s\,a_{rj} + c\,a_{ij}} \;\; , \boldsymbol{j = 1,\ldots,n}$$

$$(\boldsymbol{b_{kl} = a_{kl}} \quad \text{în rest})$$

$$(\boldsymbol{j = r}) \quad \boldsymbol{b_{ir} = -s\,a_{rr} + c\,a_{ir}}$$

Cea mai simplă alegere pentru c și s astfel ca să obținem $b_{ir}=0$ este:

$$c = f\ a_{rr} \quad , \quad s = f\ a_{ir} \quad ,$$

$$f \ \text{ ales astfel ca } c^2 + s^2 = 1 \quad \Rightarrow \quad f = \frac{1}{\sqrt{a_{rr}^2 + a_{ir}^2}}$$

$$c = \frac{a_{rr}}{\sqrt{a_{rr}^2 + a_{ir}^2}} \quad , \quad s = \frac{a_{ir}}{\sqrt{a_{rr}^2 + a_{ir}^2}}$$

$$\sqrt{a_{rr}^2 + a_{ir}^2} = 0 \ \Rightarrow a_{rr} = a_{ir} = 0 \ \Rightarrow \ c = 1\ ,\ s = 0$$

$$(R_{ir}(\theta) = I_n)$$

Deci elementul de pe poziția ***(i,r)*** este deja nul. Nu avem ce schimba în matricea ***A.***

Operația $\boldsymbol{B = R_{r\,i}(\theta_{r\,i})A}$, nu afectează forma superior triunghiulară a primelor ***(r-1)*** coloane. În matricea ***B*** aceste coloane vor continua să fie în formă superior triunghiulară.

$$\boldsymbol{b_{rj} = c\,a_{rj} + s\,a_{ij}} = 0\,\boldsymbol{,}\quad \boldsymbol{b_{ij} = -\,s\,a_{rj} + c\,a_{ij}} = 0\ \boldsymbol{,\ j = 1,\ldots,r-1}$$

$$\textbf{deoarece}\quad \boldsymbol{a_{rj} = a_{ij} = 0}$$

Înmulțirea $\boldsymbol{B = R_{r\,i}(\theta_{r\,i})A}$ nu schimbă decât liniile ***r*** și ***i*** ale matricei ***B***. În concluzie, operația $\boldsymbol{B = R_{r\,i}(\theta_{r\,i})A}$ nu schimbă elementele nule deja obținute, ci doar face ca elementul de pe poziția ***(i,r)*** să devină ***0***.

Algoritmul lui Givens poate fi descris astfel:

$$\boldsymbol{R}_{n-1n}(\boldsymbol{\theta}_{n-1n})\cdots \boldsymbol{R}_{rn}(\boldsymbol{\theta}_{rn})\cdots \boldsymbol{R}_{r\,r+1}(\boldsymbol{\theta}_{r\,r+1})\cdots \boldsymbol{R}_{1n}(\boldsymbol{\theta}_{1n})\cdots \boldsymbol{R}_{12}(\boldsymbol{\theta}_{12})\,\boldsymbol{A}=\boldsymbol{R}$$

$$\boldsymbol{R}_{n-1n}(\boldsymbol{\theta}_{n-1n})\cdots \boldsymbol{R}_{rn}(\boldsymbol{\theta}_{rn})\cdots \boldsymbol{R}_{r\,r+1}(\boldsymbol{\theta}_{r\,r+1})\cdots \boldsymbol{R}_{1n}(\boldsymbol{\theta}_{1n})\cdots \boldsymbol{R}_{12}(\boldsymbol{\theta}_{12})\boldsymbol{b}=\bar{\boldsymbol{b}}=\boldsymbol{Q}^T\boldsymbol{b}$$

Notăm cu $\widetilde{\boldsymbol{Q}}$ următoarea matrice:

$$\widetilde{\boldsymbol{Q}}=\boldsymbol{R}_{n-1n}(\boldsymbol{\theta}_{n-1n})\cdots \boldsymbol{R}_{rn}(\boldsymbol{\theta}_{rn})\cdots \boldsymbol{R}_{r\,r+1}(\boldsymbol{\theta}_{r\,r+1})\cdots \boldsymbol{R}_{1n}(\boldsymbol{\theta}_{1n})\cdots \boldsymbol{R}_{12}(\boldsymbol{\theta}_{12})$$

Matricea $\widetilde{\boldsymbol{Q}}$ este matrice ortogonală ca produs de matrice ortogonale. Descompunerea $\boldsymbol{QR}$ a matricei $\boldsymbol{A}$ este următoarea:

$$\widetilde{\boldsymbol{Q}}\,\boldsymbol{A} = \boldsymbol{R} \quad (\widetilde{\boldsymbol{Q}}^{-1} = \widetilde{\boldsymbol{Q}}^{T}) \Rightarrow \boldsymbol{A} = \widetilde{\boldsymbol{Q}}^{T}\boldsymbol{R} = \boldsymbol{QR}$$

$$\boldsymbol{Q} = \widetilde{\boldsymbol{Q}}^{T} = \left(\boldsymbol{R}_{n-1n}(\theta_{n-1n})\cdots\boldsymbol{R}_{rn}(\theta_{rn})\cdots\boldsymbol{R}_{r\,r+1}(\theta_{r\,r+1})\cdots\boldsymbol{R}_{1n}(\theta_{1n})\cdots\boldsymbol{R}_{12}(\theta_{12})\right)^{T}$$

$$= \boldsymbol{R}_{12}^{T}(\theta_{12})\cdots\boldsymbol{R}_{1n}^{T}(\theta_{1n})\cdots\boldsymbol{R}_{r\,r+1}^{T}(\theta_{r\,r+1})\cdots\boldsymbol{R}_{rn}^{T}(\theta_{rn})\cdots\boldsymbol{R}_{n-1n}^{T}(\theta_{n-1n})$$

Pe scurt, ***algoritmul lui Givens*** este următorul:

$$\widetilde{Q} = I_n;$$

$$\textbf{for } r = 1, \cdots, n-1$$

$$\quad \textbf{for } i = r+1, \cdots, n$$

$$\left\{ \begin{array}{l} A = R_{ri}(\theta_{ri}) * A; \\ b = R_{ri}(\theta_{ri}) * b; \\ \widetilde{Q} = R_{ri}(\theta_{ri}) * \widetilde{Q}; \end{array} \right.$$

$\widetilde{Q} = I_n;$

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

for $i = r+1,\ldots,n$

// **construcţia matricii** $R_{ri}(\theta_{ri})$ – **constantele** c **şi** s

- $f = \sqrt{a_{rr}^2 + a_{ir}^2};$
- **if** $(f \leq \varepsilon)\{c = 1; s = 0;\}$ // $R_{ri}(\theta_{ri}) = I$

  else $\{c = a_{rr} / f; s = a_{ir} / f;\}$

// $A = R_{ri}(\theta_{ri}) * A$

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

$$\begin{cases} a_{rj} = c * a_{rj} + s * a_{ij}; \\ a_{ij} = -s * a_{rj}^{vechi} + c * a_{ij}; \end{cases}$$

- $a_{ir} = 0;\ a_{rr} = f$;

$// \ b = R_{ri}(\theta_{ri}) * b$

- $b_r = c * b_r + s * b_i;$
- $b_i = - s * b_r^{vechi} + c * b_i;$

$// \ \widetilde{Q} = R_{ri}(\theta_{ri}) * \widetilde{Q}$

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

$$\begin{cases} \tilde{q}_{rj} = c * \tilde{q}_{rj} + s * \tilde{q}_{ij}; \\ \tilde{q}_{ij} = - s * \tilde{q}_{rj}^{vechi} + c * \tilde{q}_{ij}; \end{cases}$$

La sfârșitul acestui algoritm, în matricea $\boldsymbol{A}$ vom avea matricea superior triunghiulară $\boldsymbol{R}$, în vectorul $\boldsymbol{b}$ vom avea $\widetilde{\boldsymbol{Q}}\boldsymbol{b}^{\mathbf{init}}$($\boldsymbol{b}^{\mathbf{init}}$ - vectorul termenilor liberi inițial), iar matricea $\widetilde{\boldsymbol{Q}}$ va conține matricea $\boldsymbol{Q}^T$ din factorizarea $\boldsymbol{QR}$ a matricei $\boldsymbol{A}$.
Numărând operațiile efectuate (exceptând calculul matricei $\widetilde{\boldsymbol{Q}}$) obținem:

$$\frac{n(n-1)}{2} \ \textbf{radicali}$$

$$\frac{n(n-1)(4n+7)}{6} = \frac{2}{3}n^3 + O(n^2) - \textbf{adunări/scăderi}$$

$$\frac{2n(n-1)(2n+5)}{3} = \frac{4}{3}n^3 + O(n^2) \ \textbf{înmulţiri/împărţiri}$$

QR – algoritmul Gram Schmidt

$$A \in \mathbb{R}^{n\times n} \quad \text{cu} \quad \det A \neq 0$$

$$A=QR$$

$$\begin{bmatrix} a^1 & a^2 & \cdots & a^n \end{bmatrix} = \begin{bmatrix} q^1 & q^2 & \cdots & q^n \end{bmatrix} \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & & & \\ 0 & 0 & \cdots & r_{nn} \end{pmatrix} \tag{1}$$

$a^j = Ae_j$ – coloana j a matricei A

$q^j = Qe_j$ – coloana j a matricei Q

Relația (1) poate fi rescrisă astfel:

$$\begin{cases} \boldsymbol{r}_{11}\boldsymbol{q}^1 = \boldsymbol{a}^1 \\ \boldsymbol{r}_{12}\boldsymbol{q}^1 + \boldsymbol{r}_{22}\boldsymbol{q}^2 = \boldsymbol{a}^2 \\ \vdots \\ \boldsymbol{r}_{1p}\boldsymbol{q}^1 + \cdots + \boldsymbol{r}_{jp}\boldsymbol{q}^j + \cdots + \boldsymbol{r}_{pp}\boldsymbol{q}^p = \boldsymbol{a}^p \\ \vdots \\ \boldsymbol{r}_{1n}\boldsymbol{q}^1 + \cdots + \boldsymbol{r}_{jn}\boldsymbol{q}^j + \cdots + \boldsymbol{r}_{nn}\boldsymbol{q}^n = \boldsymbol{a}^n \end{cases} \tag{2}$$

Algoritmul de calcul al descompunerii ***QR*** cu metoda Gram-Schmidt se desfăşoară în ***n*** paşi, la fiecare pas calculându-se:

- coloana ***p*** din matricea ***R***
- coloana ***p*** din matricea ***Q***

Avem:

$$\mathbf{det}\, A \neq 0,\ A = QR,\ Q - \ \mathbf{ortogonal\breve{a}} \Rightarrow\ \mathbf{det}\, R \neq 0\ (r_{ii} \neq 0\ \forall i)$$

Pasul 1

Se folosește prima ecuație a sistemului (2)

$$r_{11}q^1 = a^1$$

Se face produsul scalar al acestei ecuații cu vectorii q^1 şi a^1**.** Se folosește proprietatea coloanelor matricelor ortogonale:

$$\left(q^i, q^j\right)_{\mathbb{R}^n} = \begin{cases} \| q^i \|_2^2 = 1 & \textbf{pentru} \quad i = j \\ 0 & \textbf{pentru} \quad i \neq j \end{cases}$$

$$q^1 = \frac{1}{r_{11}} a^1 \quad \textbf{şi} \quad \left\| q^1 \right\|_2 = 1$$

$$r_{11} = \pm \left\| a^1 \right\|_2 \;, \quad q^1 = \frac{1}{r_{11}} a^1 \ (r_{11} \neq 0 \textbf{ deoarece } \det A \neq 0)$$

Pasul p
Se folosește ecuația $\boldsymbol{p}$ sistemului (2):

$$r_{1p}q^{1} + \cdots + r_{jp}q^{j} + \cdots + r_{pp}q^{p} = a^{p}$$

La acest pas se cunosc deja coloanele $q^{1}, q^{2}, \ldots, q^{p-1}$. Se face, pe rând, produsul scalar al acestei ecuații cu vectorii:

$$q^{1}, q^{2}, \ldots, q^{p-1}$$

$$q^{p} \text{ şi } a^{p}.$$

$$\left(\sum_{k=1}^{p} r_{kp} q^{k}, q^{j} \right)_{\mathbb{R}^{n}} = \left(a^{p}, q^{j} \right)_{\mathbb{R}^{n}} \quad j = 1, \ldots, p-1$$

$$\left(\sum_{k=1}^{p} r_{kp} q^k, q^j\right)_{\mathbb{R}^n} = \sum_{\substack{k=1 \\ k\neq j}}^{p} r_{kp}\left(q^k, q^j\right)_{\mathbb{R}^n} + r_{jp}\left(q^j, q^j\right)_{\mathbb{R}^n} = r_{jp}$$

$$r_{jp} = \left(a^p, q^j\right)_{\mathbb{R}^n} \quad j = 1, \dots, p-1$$

Avem:

$$q^p = \frac{1}{r_{pp}}\left(a^p - r_{1p} q^1 - \cdots - r_{p-1\,p} q^{p-1}\right)$$

$$\| q^p \|_2^2 = 1 = \frac{1}{r_{pp}^2} \| \left(a^p - r_{1p} q^1 - \cdots - r_{p-1\,p} q^{p-1}\right) \|_2^2$$

Obținem:

$$r_{pp} = \pm \left\| a^p - r_{1p} q^1 - \cdots - r_{p-1\,p} q^{p-1} \right\|_2$$

$$q^p = \frac{1}{r_{pp}} \left(a^p - r_{1p} q^1 - \cdots - r_{p-1\,p} q^{p-1} \right) = \frac{1}{r_{pp}} \left(a^p - \sum_{j=1}^{p-1} r_{jp} q^j \right)$$

Algoritmul Gram Schmidt modificat

for ***i = 1,n***

  $v^i = a^i;$

for ***i = 1, n***

  $r_{ii} = \|v^i\|_2;$

  $q^i = (1/ r_{ii}) v^i;$

  for $j = (i + 1), n$

    $r_{ij} = (q^i, v^j);$

    $v^j = v^j - r_{ij} q^i;$

$$\text{M: } \frac{(n^3 + 3n^2)}{2} \qquad \text{A: } \frac{(n^3 + n^2 - 2)}{2}$$