Číslo podmíněnosti matice. Číslo podmíněnosti $\kappa_2(\mathbf{A})$ matice $\mathbf{A}$ (v dalším stručně označované jako $\kappa(\mathbf{A})$) definujeme jako podíl největšího a nejmenšího singulárního čísla matice $\mathbf{A}$,

$$\kappa(\mathbf{A}) = \sigma_{max}(\mathbf{A})/\sigma_{min}(\mathbf{A})\,.$$

Má-li matice $\mathbf{A}$ plnou sloupcovou hodnost, tj. když $h(\mathbf{A}) = n$, pak

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}^+\|\,, \tag{2.12}$$

jak jsme uvedli v kapitole 2.1. Když má matice $\mathbf{A}$ neúplnou sloupcovou hodnost, tj. když $h(\mathbf{A}) < n$, pak $\sigma_{min}(\mathbf{A}) = 0$, takže $\kappa(\mathbf{A}) = \infty$.

V kapitole 2.2 jsme také uvedli, že

$$\kappa(\mathbf{A}^T\mathbf{A}) = \kappa^2(\mathbf{A})\,. \tag{2.13}$$

To je ale přímý důsledek toho, že singulární čísla matice $\mathbf{A}^T\mathbf{A}$ jsou kvadráty singulárních čísel matice $\mathbf{A}$.

Řešení soustavy lineárních rovnic s minimální euklidovskou normou. Pomocí singulárního rozkladu dostaneme

$$\begin{aligned}\|\mathbf{r}\|^2 =& \|\mathbf{U}^T\mathbf{r}\|^2 = \|\mathbf{U}^T(\mathbf{b} - \mathbf{A}\mathbf{x})\|^2 = \|\mathbf{U}^T(\mathbf{b} - \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{x})\|^2 = \|\mathbf{\Sigma}\mathbf{V}^T\mathbf{x} - \mathbf{U}^T\mathbf{b}\|^2 = \\ =& \sum_{i=1}^{r}(\sigma_i\alpha_i - \mathbf{u}_i^T\mathbf{b})^2 + \sum_{i=r+1}^{m}(\mathbf{u}_i^T\mathbf{b})^2\,,\end{aligned}$$

kde $\boldsymbol{\alpha} = \mathbf{V}^T\mathbf{x}$. Minimální reziduum dostaneme, když $\alpha_i^* = \mathbf{u}_i^T\mathbf{b}/\sigma_i$ pro $i = 1, 2, \ldots, r$. Jestliže zvolíme $\alpha_i^* = 0$ pro $i = r{+}1, r{+}2, \ldots, n$, pak také euklidovská norma $\|\mathbf{x}^*\| = \|\boldsymbol{\alpha}^*\|$ MNČ řešení $\mathbf{x}^*$ je minimální. Přitom

$$\mathbf{x}^* = \sum_{i=1}^{r}\alpha_i^*\mathbf{v}_i = \sum_{i=1}^{r}\mathbf{u}_i^T\mathbf{b}/\sigma_i\mathbf{v}_i = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^T\mathbf{b} = \mathbf{A}^+\mathbf{b}\,.$$

Existují spolehlivé stabilní algoritmy pro výpočet singulárního rozkladu, viz např. [14], a jejich kvalitní implementace. V MATLABu vypočteme singulární rozklad pomocí funkce `pinv`, příkaz `x=pinv(A)*b` dodá MNČ řešení s minimální euklidovskou normou.

2.4. QR algoritmy

V této kapitole uvedeme tři často používané QR algoritmy založené na

- Householderových reflexích
- Givensových rovinných rotacích
- Gramově – Schmidtově ortogonalizaci

Nejlepší je Householderův algoritmus, který proto rozebíráme poměrně podrobně. Zbývající dva algoritmy jsou pro některé typy matic také velmi užitečné. Uvádíme je mimo jiné proto, že Givensovy matice rovinných rotací i Gramův – Schmidův ortonormalizační proces jsou významné nástroje numerické matematiky, které nacházejí uplatnění nejen v metodě nejmenších čtverců.

2.4.1. Householderův QR algoritmus

je založen na použití *Householderových reflektorů*, což jsou matice tvaru

$$\mathbf{H} = \mathbf{I} - 2\mathbf{w}\mathbf{w}^T, \tag{2.14}$$

kde $\mathbf{w}$ je vektor jednotkové délky, tj. $\|\mathbf{w}\| = 1$. Protože $\mathbf{H}^2 = \mathbf{I}$, je $\mathbf{H} = \mathbf{H}^{-1} = \mathbf{H}^T$, takže $\mathbf{H}$ ortonormální. Geometricky vektor $\mathbf{Hx}$ reprezentuje zrcadlový obraz $\mathbf{x}$ vzhledem k nadrovině $\mathrm{span}(\mathbf{w})^\perp$ kolmé k $\mathbf{w}$. To je zřejmé z toho, že střed $\mathbf{y} = \frac{1}{2}(\mathbf{x}+\mathbf{Hx})$ úsečky spojující vzor $\mathbf{x}$ a obraz $\mathbf{Hx}$ leží v $\mathrm{span}(\mathbf{w})^\perp$:

$$\mathbf{w}^T\mathbf{y} = \mathbf{w}^T(\mathbf{I} - \mathbf{w}\mathbf{w}^T)\mathbf{x} = \mathbf{w}^T\mathbf{x} - \mathbf{w}^T\mathbf{w}\mathbf{w}^T\mathbf{x} = \mathbf{o}.$$

Pro daný vektor $\mathbf{x} \neq \mathbf{o}$ se vektor $\mathbf{w}$ Householderovy transformace (2.14) zvolí tak, aby

$$\mathbf{Hx} = \alpha\mathbf{e}_1\,,$$

kde α je skalár a $\mathbf{e}_1$ je první sloupec jednotkové matice. Transformace $\mathbf{Hx}$ tedy anuluje všechny složky $\mathbf{x}$ s výjimkou prvé. V tom je podstata Householderovy transformace! Z rovnice $(\mathbf{I} - 2\mathbf{w}\mathbf{w}^T)\mathbf{x} = \alpha\mathbf{e}_1$ plyne

$$(2\mathbf{w}^T\mathbf{x})\mathbf{w} = \mathbf{x} - \alpha\mathbf{e}_1\,.$$

To ale znamená, že $\mathbf{w}$ je násobkem vektoru $\mathbf{x} - \alpha\mathbf{e}_1$, takže

$$\mathbf{w} = \pm\frac{\mathbf{x} - \alpha\mathbf{e}_1}{\|\mathbf{x} - \alpha\mathbf{e}_1\|}\,.$$

Protože $\mathbf{H}$ je ortonormální, $|\alpha| = \|\mathbf{Hx}\| = \|\mathbf{x}\|$, tj. $\alpha = \pm\|\mathbf{x}\|$. Abychom se vyhnuli tomu, že vektor $\mathbf{x} - \alpha\mathbf{e}_1$ bude malý, zvolíme $\alpha = -\mathrm{sign}(x_1)\|\mathbf{x}\|$ a

$$\mathbf{w} = \frac{\mathbf{x} + \beta\mathbf{e}_1}{\|\mathbf{x} + \beta\mathbf{e}_1\|}\,, \qquad \text{kde} \quad \beta = -\alpha = \mathrm{sign}(x_1)\|\mathbf{x}\|\,.$$

Všimněte si, že když chceme vypočítat $\mathbf{Hx}$, nemusíme matici $\mathbf{H}$ vůbec sestavovat, neboť

$$\mathbf{Hx} = (\mathbf{I} - 2\mathbf{w}\mathbf{w}^T)\mathbf{x} = \mathbf{x} - \lambda\mathbf{w}\,, \qquad \text{kde} \quad \lambda = 2\mathbf{w}^T\mathbf{x}\,.$$

Věnujme se nyní už popisu vlastního Householderova QR algoritmu. V prvním kroku anulujeme poddiagonální prvky prvního sloupce $\mathbf{a}_1 = (a_{11}, a_{21}, \ldots, a_{m1})^T$ matice $\mathbf{A}$. Pokud $a_{21} = a_{31} = \cdots = a_{m1} = 0$, položíme $\mathbf{H}_1 = \mathbf{I}$, $\mathbf{A}_1 = \mathbf{H}_1\mathbf{A} = \mathbf{A}$ a první krok je hotov. Je-li však alespoň jeden z prvků $a_{21}, a_{31}, \ldots, a_{m1}$ nenulový, anulujeme poddiagonální prvky prvního sloupce matice $\mathbf{A}$ pomocí Householderovy matice

$$\mathbf{H}_1 = \mathbf{I} - 2\mathbf{w}_1\mathbf{w}_1^T, \qquad \text{kde} \quad \mathbf{w}_1 = \frac{\mathbf{a}_1 + \beta_1\mathbf{e}_1}{\|\mathbf{a}_1 + \beta_1\mathbf{e}_1\|}\,, \quad \beta_1 = \mathrm{sign}(a_{11})\|\mathbf{a}_1\|\,.$$

Pak v matici $\mathbf{A}_1 = \mathbf{H}_1\mathbf{A}$ je $a_{11}^{(1)} = -\beta_1$ a zbývající prvky prvního sloupce jsou nulové. Abychom nekomplikovali značení, budeme prvky $a_{ij}^{(1)}$ matice $\mathbf{A}_1$ zase značit jako a_{ij}, tj. horní index vypustíme.

Předpokládejme, že matice $\mathbf{A}$ byla transformována v $k-1$ krocích na tvar

$$\mathbf{A}_{k-1} = \mathbf{H}_{k-1}\mathbf{H}_{k-2}\cdots\mathbf{H}_1\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & \cdots & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & \cdots & \cdots & a_{2n} \\ 0 & 0 & a_{33} & \cdots & \cdots & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & a_{kk} & \cdots & a_{kn} \\ 0 & 0 & 0 & \cdots & a_{k+1,k} & \cdots & a_{k+1,n} \\ \vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{mk} & \cdots & a_{mn} \end{pmatrix}.$$

Tato matice je horní trojúhelníková až do sloupce $k-1$. V $k-$tém kroku transformujeme matici $\mathbf{A}_{k-1}$ na matici

$$\mathbf{A}_k = \mathbf{H}_k\mathbf{A}_{k-1}$$

tak, aby $\mathbf{A}_k$ měla měla nulové poddiagonální prvky v prvních k sloupcích. Pokud $\mathbf{A}_{k-1}$ už v k-tém sloupci pod diagonálou samé nuly má, položíme $\mathbf{H}_k = \mathbf{I}$ a $k-$tý krok je hotov. Je-li však některý z prvků $a_{k+1,k}, a_{k+2,k}, \ldots, a_{m,k}$ nenulový, zvolíme za $\mathbf{H}_k$ Householderovu matici

$$\mathbf{H}_k = \mathbf{I} - 2\mathbf{w}_k\mathbf{w}_k^T\,,$$

v němž

$$\mathbf{w}_k = \frac{\mathbf{z}_k}{\|\mathbf{z}_k\|} \tag{2.15}$$

a složky vektoru $\mathbf{z}_k$ jsou

$$z_i^{(k)} = \begin{cases} 0 & \text{pro } i < k, \\ a_{kk} + \beta_k & \text{pro } i = k, \\ a_{ik} & \text{pro } i > k, \end{cases} \qquad \beta_k = \operatorname{sign}(a_{kk})\left(\sum_{i=k}^{m} a_{ik}^2\right)^{1/2}. \tag{2.16}$$

Není obtížné prověřit, že matice $\mathbf{A}_{k-1}$ a $\mathbf{A}_k$ mají prvních $k-1$ řádků a sloupců stejných. Nuly pod diagonálou v prvních $k-1$ sloupcích tedy zůstávají zachovány. V $k-$tém sloupci matice $\mathbf{A}_k$ dostaneme na diagonále $-\beta_k$ a pod diagonálou samé nuly.

Nechť $q = \min(m-1, n)$. Pak po provedení q kroků dostaneme horní trojúhelníkovou matici

$$\mathbf{R} = \mathbf{H}_q\mathbf{H}_{q-1}\cdots\mathbf{H}_1\mathbf{A} \equiv \mathbf{H}\mathbf{A}\,, \qquad \text{kde} \quad \mathbf{H} = \mathbf{H}_q\mathbf{H}_{q-1}\cdots\mathbf{H}_1\,,$$

a odtud

$$\mathbf{A} = \mathbf{Q}\mathbf{R}\,, \qquad \text{kde} \quad \mathbf{Q} = \mathbf{H}^T = \mathbf{H}_1\mathbf{H}_2\cdots\mathbf{H}_q\,.$$

Matici $\mathbf{H}$ můžeme počítat tak, že položíme $\mathbf{H} = \mathbf{I}$ a v k-tém kroku provedeme $\mathbf{H} := \mathbf{H}_k\mathbf{H}$. Jestliže má matice $\mathbf{A}$ plnou hodnost, tj. když $h(\mathbf{A}) \equiv r = p$, kde $p = \min(m, n)$, pak také $h(\mathbf{R}) = p$ a matice $\mathbf{R}$ má na hlavní diagonále nenulové prvky. Jestliže však $\mathbf{A}$ plnou hodnost nemá, tj. když $r < p$, pak také $h(\mathbf{R}) = r$, a to znamená, že na hlavní diagonále matice $\mathbf{R}$ jsou i nulové prvky. Vhodným přeuspořádáním sloupců matice $\mathbf{A}$ lze docílit toho, že diagonální prvky r_{ii}, $i = 1, 2, \ldots, r$, budou nenulové, a že řádky $r+1, r+2, \ldots, m$ matice $\mathbf{R}$ budou nulové.

Pivotování po sloupcích. Začneme tím, že si připravíme permutační matici $\mathbf{P} = \mathbf{I}$, kde $\mathbf{I}$ je jednotková matice řádu n, do níž budeme zaznamenávat provedená prohození sloupců. V $k-$tém kroku pak určíme index s *pivotního sloupce* tak, že

$$\gamma_s = \max_{k \le j \le n} \gamma_j \,, \qquad \text{kde} \quad \gamma_j = \sum_{i=k}^{m} a_{ij}^2 \,. \tag{2.17}$$

Jestliže $\gamma_s = 0$, je $h(\mathbf{A}) = k-1$ a QR algoritmus končí. Pokud ale $\gamma_s > 0$, prohodíme $k-$tý a $s-$tý sloupec matic $\mathbf{A}_{k-1}$ a $\mathbf{P}$ a teprve pak provedeme Householderovu transformaci. Tak nakonec dostaneme

$$\mathbf{AP} = \mathbf{QR} \,. \tag{2.18}$$

Čísla γ_j se nemusejí počítat v každém kroku znovu, využijeme-li vlastnost:

$$\text{pokud} \quad \mathbf{Qv} = \begin{pmatrix} \alpha \\ \mathbf{w} \end{pmatrix} , \quad \text{pak} \quad \|\mathbf{w}\|^2 = \|\mathbf{v}\|^2 - \alpha^2 \,,$$

která platí pro každou ortonormální matici $\mathbf{Q}$. Nyní již můžeme uvést

Algoritmus HQR (Householder QR)

1. $\mathbf{H} := \mathbf{I}$, $\mathbf{p} := (1, 2, \ldots, n)^T$, $p := \min(m, n)$, $r := p$
2. **for** $j := 1, 2, \ldots, n$ **do** $\gamma_j := \sum_{i=1}^{m} a_{ij}^2$ **end**
3. **for** $k := 1, 2, \ldots, p$ **do**
4. určíme s a γ_s tak, že $\gamma_s = \max_{k \le j \le n} \gamma_j$
5. **if** $\gamma_s = 0$, $r := k - 1$, **goto** 13, **end**
6. **if** $s \neq k$, prohodíme $\gamma_k \leftrightarrows \gamma_s$, $\mathbf{a}_k \leftrightarrows \mathbf{a}_s$ a $p_k \leftrightarrows p_s$, **end**
7. **if** $k = m$ **goto** 13
8. vypočteme $\mathbf{w}_{k[k:m]}$ podle (2.15) – (2.16)
9. $\mathbf{A}_{[k:m,k:n]} := \mathbf{A}_{[k:m,k:n]} - 2\mathbf{w}_{k[k:m]}(\mathbf{w}_{k[k:m]}^T \mathbf{A}_{[k:m,k:n]})$
10. $\mathbf{H}_{[k:m,1:m]} := \mathbf{H}_{[k:m,1:m]} - 2\mathbf{w}_{k[k:m]}(\mathbf{w}_{k[k:m]}^T \mathbf{H}_{[k:m,1:m]})$
11. **for** $j := k + 1, k + 2, \ldots, n$ **do** $\gamma_j := \gamma_j - a_{kj}^2$ **end**
12. **end**
13. $\mathbf{R} := \mathbf{A}$, $\mathbf{Q} := \mathbf{H}^T$, $\mathbf{P} := (\mathbf{e}_{p_1}, \mathbf{e}_{p_2}, \ldots, \mathbf{e}_{p_n})$

K algoritmu HQR připojíme několik poznámek.

1. V řádku 1 je $\mathbf{I}$ jednotková matice řádu m.

2. Označili jsme $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n)$.

3. $\mathbf{A}_{[k:m,k:n]}$ je submatice $\mathbf{A}$ tvořená řádky $k, k+1, \ldots, m$ a sloupci $k, k+1, \ldots, n$, podobně $\mathbf{H}_{[k:m,1:m]}$ resp. $\mathbf{w}_{k[k:m]}$ je submatice $\mathbf{H}$ resp. subvektor $\mathbf{w}_k$.

4. V řádku 7 při výpočtu $\mathbf{w}_k$ využijeme v (2.16) toho, že $\beta_k = \mathrm{sign}(a_{kk})\gamma_k^{1/2}$.

5. V řádku 13 $\mathbf{e}_{p_j}$ je p_j-tý sloupec jednotkové matice řádu n.

6. $r = h(\mathbf{A})$ je hodnost matice $\mathbf{A}$.

Redukovaný Householderův QR rozklad. Nechť $n < m$, $\mathbf{Q}_1$ je matice tvořená prvními n sloupci matice $\mathbf{Q}$ a $\mathbf{R}_1$ je čtvercová matice tvořená prvními n řádky matice $\mathbf{R}$, pak

$$\mathbf{AP} = \mathbf{Q}_1\mathbf{R}_1\,. \tag{2.19}$$

Rozklad (2.19) bývá v anglicky psané literatuře označován jako „economy size QR decomposition“, tj. ekonomický (tedy úsporný, redukovaný) QR rozklad. Toto pojmenování se používá také třeba v MATLABu: příkazem `[Q,R,P]=qr(A)` dostaneme úplný rozklad (2.18), zatímco příkazem `[Q1,R1,P]=qr(A,0)` dostaneme redukovaný rozklad (2.19).

HQRe algoritmus (Householder QR economy)

dostaneme z HQR algoritmu náhradou řádků 1, 10 a 13,

1. $\mathbf{p} := (1, 2, \ldots, n)^T$, $p := n$, $r := n$
10. $\mathbf{q}_k := \mathbf{e}_k$, **for** $j := k, k-1, \ldots, 1$ **do** $\mathbf{q}_{k[j:m]} := \mathbf{q}_{k[j:m]} - (2\mathbf{w}_{j[j:m]}^T\mathbf{q}_{k[j:m]})\mathbf{w}_{j[j:m]}$ **end**
13. $\mathbf{R}_1$ je prvních n řádků matice $\mathbf{A}$, $\mathbf{Q}_1 = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n)$

přičemž $\mathbf{e}_k$ je $k-$tý sloupec jednotkové matice řádu m. Rozdíl oproti algoritmu HQR spočívá v tom, že v modifikovaném řádku 9 vytváříme $k-$tý sloupec $\mathbf{q}_k$ matice $\mathbf{Q}_1$. Generují se tedy jen ty ortonormální vektory, které jsou v $\mathbf{Q}_1$! Abychom pochopili řádek 9, musíme si uvědomit, že

$$\mathbf{H}_n\mathbf{H}_{n-1}\cdots\mathbf{H}_1\mathbf{AP} = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n)\mathbf{R}_1,$$

kde $\mathbf{e}_j$, $j = 1, 2, \ldots, n$, je $j-$tý sloupec jednotkové matice řádu m, takže

$$\mathbf{AP} = \mathbf{H}_1\mathbf{H}_2\cdots\mathbf{H}_n(\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n)\mathbf{R}_1 \equiv \mathbf{Q}_1\mathbf{R}_1,$$

a že přitom

$$\mathbf{H}_1\mathbf{H}_2\cdots\mathbf{H}_n\mathbf{e}_k = \mathbf{H}_1\mathbf{H}_2\cdots\mathbf{H}_k\mathbf{e}_k\,, \qquad k = 1, 2, \ldots, n.$$

K výpočtu $\mathbf{q}_k$ podle řádku 9 potřebujeme mít k dispozici vektory $\mathbf{w}_{j[j:m]}$ pro $j = 1, 2, \ldots, k$. K úschově $\mathbf{w}_{j[j+1:m]}$ lze využít poddiagonální pozice matice $\mathbf{A}$, v nichž by jinak byly nuly, $\mathbf{w}_{j[j:j]}$ je třeba uložit zvlášť.

Pokud QR algoritmus používáme jen k nalezení MNČ řešení SLR, v algoritmu HQR změníme řádky 1,10 a 13,

1. $\mathbf{d} := \mathbf{b}$, $\mathbf{p} := (1, 2, \ldots, n)^T$, $p := \min(m, n)$, $r := p$
10. $\mathbf{d}_{[k:m]} := \mathbf{d}_{[k:m]} - (2\mathbf{w}_{k[k:m]}^T \mathbf{d}_{[k:m]})\mathbf{w}_{k[k:m]}$
13. $\mathbf{R}_1 := \mathbf{A}_{[1:r,1:n]}$, $\mathbf{d}_1 := \mathbf{d}_{[1:r]}$, $\mathbf{P} := (\mathbf{e}_{p_1}, \mathbf{e}_{p_2}, \ldots, \mathbf{e}_{p_n})$

a MNČ řešení soustavy $\mathbf{Ax} = \mathbf{b}$ dostaneme jako klasické řešení soustavy $\mathbf{R}_1\mathbf{P}^T\mathbf{x} = \mathbf{d}_1$, viz kapitola 2.2.

Závěr. Householderův QR algoritmus je považován mezi QR algoritmy za nejlepší. Používá ho také funkce `qr` v MATLABu. Pro speciální typy matic však může být některý z dalších známých QR algoritmů efektivnější.

2.4.2. Givensův QR algoritmus

Všimněte si, že matice

$$\mathbf{G} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}$$

je ortonormální, když $c^2 + s^2 = 1$. Tato matice se nazývá *matice rovinné rotace*, protože čísla c a s jsou kosinus a sinus nějakého úhlu, o který se vektor $\mathbf{x}$ rovinně otočí, jestliže ho maticí $\mathbf{G}$ vynásobíme. Buď $\mathbf{x} = (x_1, x_2)^T \neq \mathbf{o}$. Jestliže položíme $c = x_1/\|\mathbf{x}\|$, $s = x_2/\|\mathbf{x}\|$, splňují čísla c a s potřebnou rovnost a platí

$$\mathbf{Gx} = \begin{pmatrix} x_1^2/\|\mathbf{x}\| + x_2^2/\|\mathbf{x}\| \\ -x_1x_2/\|\mathbf{x}\| + x_1x_2/\|\mathbf{x}\| \end{pmatrix} = \begin{pmatrix} \|\mathbf{x}\| \\ 0 \end{pmatrix}.$$

Právě v tom je smysl použití matice rovinné rotace, totiž anulovat druhou složku vektoru.

Buď $\mathbf{a}_1$ první sloupec matice $\mathbf{A}$. Chceme nejdříve vynulovat jeho druhý prvek čili prvek a_{21} matice. Je-li již $a_{21} = 0$, neprovádíme nic. V opačném případě pro $d = \sqrt{a_{11}^2 + a_{21}^2}$ položíme

$$\mathbf{G}_{21} = \begin{pmatrix} a_{11}/d & a_{21}/d & 0 & \ldots & 0 \\ -a_{21}/d & a_{11}/d & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix}$$

a provedeme součin $\mathbf{G}_{21}\mathbf{A}$. Protože $\mathbf{G}_{21}\mathbf{a}_1$ je vektor, jehož druhá složka je nulová, anulovali jsme prvek a_{21}.

Jako další budeme anulovat třetí prvek prvního sloupce matice $\mathbf{G}_{21}\mathbf{A}$. Její prvky označíme zase jen a_{ij}, abychom nekomplikovali označení. Je-li $a_{31} \neq 0$, položíme nyní $d = \sqrt{a_{11}^2 + a_{31}^2}$ a zavedeme další ortonormální matici

$$\mathbf{G}_{31} = \begin{pmatrix} a_{11}/d & 0 & a_{31}/d & 0 & \ldots & 0 \\ 0 & 1 & 0 & 0 & \ldots & 0 \\ -a_{31}/d & 0 & a_{11}/d & 0 & \ldots & 0 \\ 0 & 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & \ldots & 1 \end{pmatrix}.$$

Po vynásobení touto maticí bude v poloze $(3,1)$ nulový prvek, a jak snadno zjistíme, v předchozím kroku anulovaný prvek v poloze $(2,1)$ zůstane nulový. Tak pokračujeme, až vynulujeme všechny prvky prvního sloupce.

Podobně anulujeme prvky pod diagonálou v dalších sloupcích, obecně prvek $a_{ik} \neq 0$ anulujeme vynásobením maticí $\mathbf{G}_{ik}$, kterou dostaneme z jednotkové matice $\mathbf{I}$ řádu m takto:

$$\mathbf{G}_{ik} := \mathbf{I}\,, \quad \text{pak v } \mathbf{G}_{ik} = \{g_{j\ell}\}_{j,\ell=1}^{m} \quad \text{změníme} \qquad \begin{aligned} g_{kk} &:= c_{ik}, \quad g_{ki} := s_{ik}, \\ g_{ik} &:= -s_{ik}, \quad g_{ii} := c_{ik}, \end{aligned}$$

přičemž $c_{ik} = a_{kk}/d$, $s_{ik} = a_{ik}/d$ a $d = \sqrt{a_{kk}^2 + a_{ik}^2}$. Důležité je, že jednou anulované pozice zůstavají nulové. Pokud $a_{ik} = 0$, klademe $\mathbf{G}_{ik} = \mathbf{I}$.

Matice $\mathbf{G}_{ik}$ jsou známé jako *Givensovy matice* rovinných rotací. Označíme-li

$$\mathbf{G}_k = \mathbf{G}_{mk}\mathbf{G}_{m-1,k}\ldots\mathbf{G}_{k+1,k}, \quad k = 1,2,\ldots,n\,,$$

můžeme psát

$$\mathbf{G}_n\mathbf{G}_{n-1}\ldots\mathbf{G}_1\mathbf{A} = \mathbf{R}\,.$$

Přitom $\mathbf{R}$ je horní trojúhelníková matice a $\mathbf{G} = \mathbf{G}_n\mathbf{G}_{n-1}\ldots\mathbf{G}_1$ je matice ortonormální. Označíme-li $\mathbf{Q} = \mathbf{G}^T$, pak z rovnosti $\mathbf{GA} = \mathbf{R}$ plyne $\mathbf{A} = \mathbf{G}^T\mathbf{R} = \mathbf{QR}$.

Pivotování. Při anulování prvků k-tého sloupce je vhodné provádět pivotování: zvýšíme tím odolnost QR algoritmu vůči zaokrouhlovacím chybám a zajistíme, aby $r_{kk} \neq 0$ pro $k = 1,2,\ldots,r$, kde $r = h(\mathbf{A})$ je hodnost matice $\mathbf{A}$. K dosažení tohoto cíle použijeme sloupcové pivotování. Postupujeme analogicky jako při pivotování v Householderově metodě, viz (2.17), (2.18).

Závěr. Givensův QR algoritmus je výhodný v případě, když matice $\mathbf{A}$ má pod hlavní diagonálou málo nenulových prvků v předem známých pozicích (i,j) a neprovádí se pivotování. Pak totiž stačí použít jen „cílené“ Givensovy rotace $\mathbf{G}_{ij}$. Je-li však $\mathbf{A}$ plná matice, dáme přednost Householderově QR metodě.

2.4.3. Gramův-Schmidtův QR algoritmus

Nechť $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ jsou lineárně nezávislé vektory. Naším cílem je setrojit ortonormální vektory $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$ tak, aby $\operatorname{span}(\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n) = \operatorname{span}(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n)$. $\mathbf{q}_1$ dostaneme normalizací $\mathbf{a}_1$, tj. $\mathbf{q}_1 = \mathbf{a}_1/\|\mathbf{a}_1\|$. Předpokládejme, že jsme už sestrojili ortonormální vektory $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{k-1}$ s vlastností $\operatorname{span}(\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{k-1}) = \operatorname{span}(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{k-1})$. Nejdříve najdeme vektor $\hat{\mathbf{q}} = \mathbf{a}_k - \sum_{j=1}^{k-1} r_{jk}\mathbf{q}_j$, který je ortogonální ke každému z vektorů $\mathbf{q}_i$, $i = 1,2,\ldots,k-1$. Z podmínky $(\hat{\mathbf{q}}, \mathbf{q}_i) = 0$ dostaneme $r_{ik} = (\mathbf{a}_k, \mathbf{q}_i)$. $\mathbf{q}_k$ obdržíme normalizací $\hat{\mathbf{q}}$, tj. $\mathbf{q}_k = \hat{\mathbf{q}}/\|\hat{\mathbf{q}}\|$. To je

Klasický GSQR algoritmus (Gram-Schmidt QR)

1. $r_{11} := \|\mathbf{a}_1\|$, $\mathbf{q}_1 := \mathbf{a}_1/r_{11}$
2. **for** $k := 1,2,\ldots,n$ **do**
3. **for** $i := 1,2,\ldots,k-1$ **do** $r_{ik} := (\mathbf{a}_k, \mathbf{q}_i)$

4. $\hat{\mathbf{q}} := \mathbf{a}_k - \sum_{i=1}^{k-1} r_{ik}\mathbf{q}_i$
5. $r_{kk} := \|\hat{\mathbf{q}}\|$, $\mathbf{q}_k := \hat{\mathbf{q}}/r_{kk}$
6. **end**

Z řádků 4 a 5 plyne

$$\mathbf{a}_k = \sum_{i=1}^{k} r_{ik}\mathbf{q}_i, \quad k = 1, 2, \ldots, n.$$

Jestliže $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n)$, $\mathbf{Q}_1 = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n)$ a jestliže $\mathbf{R}_1$ je horní trojúhelníková matice řádu n, jejíž naddiagonální prvky r_{ik} jsou definovány GSQR algoritmem, pak zřejmě $\mathbf{A} = \mathbf{Q}_1\mathbf{R}_1$ je redukovaný QR rozklad matice $\mathbf{A}$.

GSQR je standardní Gramův – Schmidtův algoritmus. Existují alternativní formulace, které mají lepší numerické vlastnosti. Nejznámější z nich je modifikovaný Gramův-Schmidtův algoritmus.

GSMQR algoritmus (Gram-Schmidt modified QR)

1. $r_{11} := \|\mathbf{a}_1\|$, $\mathbf{q}_1 := \mathbf{a}_1/r_{11}$
2. **for** $k := 1, 2, \ldots, n$ **do**
3. $\quad\hat{\mathbf{q}} := \mathbf{a}_k$
4. $\quad$**for** $i := 1, 2, \ldots, k-1$ **do**
5. $\quad\quad r_{ik} := (\hat{\mathbf{q}}, \mathbf{q}_i)$
6. $\quad\quad\hat{\mathbf{q}} := \hat{\mathbf{q}} - r_{ik}\mathbf{q}_i$
7. $\quad$**end**
8. $\quad r_{kk} := \|\hat{\mathbf{q}}\|$
9. $\quad\mathbf{q}_k := \hat{\mathbf{q}}/r_{kk}$
10. **end**

Podstatou modifikace je využití vztahu $r_{ik} = (\mathbf{a}_k, \mathbf{q}_i) = (\mathbf{a}_k - \sum_{j=1}^{i-1} r_{jk}\mathbf{q}_j, \mathbf{q}_i)$.

Pokud matice $\mathbf{A}$ nemá plnou sloupcovou hodnost, můžeme pomocí sloupcového pivotování algoritmus upravit tak, aby $r_{kk} \neq 0$ pro $k = 1, 2, \ldots, r$, kde $r = h(\mathbf{A})$ je hodnost matice $\mathbf{A}$.