Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni preddiplomski studij matematike

Matea Ugrica

QR dekompozicija koristeći Givensove rotacije

Završni rad

Osijek, 20.9.2013.

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni preddiplomski studij matematike

Matea Ugrica

QR dekompozicija koristeći Givensove rotacije

Završni rad

Mentor:
doc.dr.sc. Zoran Tomljanović

Osijek, 20.9.2013.

Sažetak

U ovom radu ćemo se ukratko upoznati s problemom rješavanja sustava linearnih jednadžbi i QR dekompozicijom kao metodom rješavanja istog. Također ćemo se upoznati s matricama rotacije i njihovim svojstvima. Posebnu pozornost ćemo skrenuti na Givensove matrice rotacije jer njih možemo koristiti za dobivanje QR dekompozicije. Definirat ćemo i linearni problem najmanjih kvadrata (LPNK), te iskoristiti QR dekompoziciju za njegovo rješavanje. Na kraju rada navest ćemo neke primjere strukturiranih matrica, kao što su Hessenbergova i tridijagonalna forma. Uz to ćemo pokazati kako se upravo za takve matrice lako dobiva QR dekompozicija pomoću Givensovih rotacija. Uz rad su priloženi i Matlab kodovi za rješavanje sustava linearnih jednadžbi, pomoću QR dekompozicije dobivene Givensovim rotacijama.

Ključne riječi

sustav linearnih jednadžbi, QR dekompozicija, Givensove rotacije, linearni problem najmanjih kvadrata

Abstract

In this paper we will be introduced to the problem of solving system of linear equations and QR decomposition as one of the methods for solving it. Also we will be introduced to the rotation matrices and their properties. Particular attention will be given to the Givens matrices because they can be used for obtaining the QR decomposition. We will define linear least squares problem and use the QR decomposition for solving it. In the final chapter of the paper we will name few structured matrices, like Hessenberg and tridiagonal matrix, and demonstrate how to get QR decomposition using Givens matrices for those matrices. This paper also includes Matlab codes for solving the system of linear equations with QR decomposition obtained by Givens matrices.

Key words

system of linear equations, QR decomposition, Givens rotations, linear least squares problem

Sadržaj

1 Uvod

Rješavanje sustava linearnih jednadžbi jedan je od najstarijih matematičkih problema. Sustav n linearnih jednadžbi s n nepoznanica oblika

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
\vdots \qquad\qquad & \quad \vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
\end{aligned}$$

jednostavnije zapisujemo $\mathbf{Ax} = \mathbf{b}$, gdje je $\mathbf{A} \in \mathbb{R}^{n\times n}$, kvadratna matrica reda n, $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$, $n - dimenzionalni$ vektori. Dakle, sustav linearnih jednadžbi možemo zapisati kao produkt matrica

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

Sustavi linearnih jednadžbi imaju široku primjenu, neke od tih primjena su: obrada digitanog signala, aproksimacija nelinearnih problema u numeričkoj matematici, linearno programiranje i različite procjene i predviđanja. Algoritmi za rješavanje sustava linearnih jednadžbi su vrlo važan dio numeričke linearne algebre i imaju istaknutu ulogu u fizici, kemiji, inžinjerstvu, računarstvu i ekonomiji.

Postoje različite metode koje se koriste pri rješavanju sustava linearnih jednadžbi, kao što su Gaussove eliminacije, LU dekompozicija, QR dekompozicija, Cholesky dekompozicija, dekompozicija na singularne vrijednosti (SVD), iterativne metode (Gauss-Seidelova, Jacobijeva), o njima više u [3] i [4].

U ovom radu, koncentracija je na posebnom obliku QR dekompozicije, koja matricu $\mathbf{A} \in \mathbb{R}^{n\times n}$ rastavlja na produkt $\mathbf{QR}$, gdje je $\mathbf{Q} \in \mathbb{R}^{n\times n}$ ortogonalna (stoga vrijedi $\mathbf{Q^{-1}} = \mathbf{Q}^T$), a $\mathbf{R} \in \mathbb{R}^{n\times n}$ regularna, gornje trokutasta matrica. Tada se rješavanje sustava $\mathbf{Ax} = \mathbf{b}$ svodi na računanje $\mathbf{QR}$ faktorizacije $\mathbf{A} = \mathbf{QR}$ te rješavanje gornje trokutastog sustava

$$\mathbf{Rx} = \mathbf{Q}^T\mathbf{b}$$

koji se lako rješava metodom povratnih supstitucija (kod dodatno pogledati pod naslovom Prilog - kodovi).

QR dekompozicija može se izračunati na više različitih načina. Gram-Schmidtov postupak ortogonalizacije jedan je od tih načina, ali je numerički nestabilan, zato se za QR dekompoziciju najčešće koriste Householderove transformacije (za oba načina vidi [4]). Givensove rotacije su još jedna metoda dobivanja QR dekompozicije i upravo ta metoda biti će obrađena u ovom radu, zajedno sa njenom primjenom.

2 Givensove rotacije

2.1 Matrice rotacije

Matrica $\mathbf{A} \in \mathbb{R}^{2\times 2}$ je matrica rotacije u ravnini za koju vrijedi

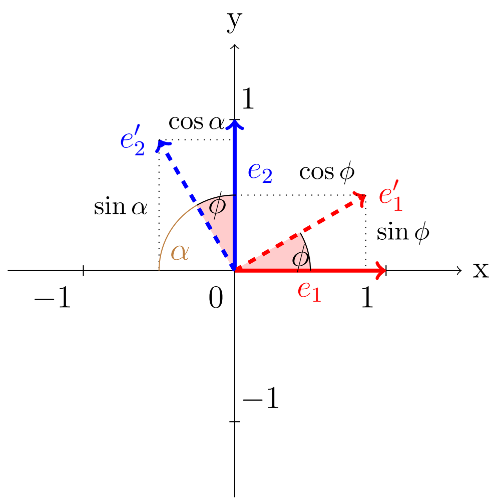
$$\mathbf{A} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix},$$

gdje je ϕ kut rotacije suprotan smjeru rotacije kazaljke na satu.

Pokažimo da je ovako zadana matrica $\mathbf{A}$ stvarno matrica rotacije. Neka je s

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

zadana matrica rotacije. Promotrimo što dobijemo rotiranjem vektora kanonske baze za kut ϕ u pravokutnom koordinatnom sustavu (Slika 1.).



Slika 1. Rotacija u koordinatnom sustavu

Primjetimo da će vektor e_2 nakon rotacije biti $e_2' = \begin{bmatrix} \cos\alpha & \sin\alpha \end{bmatrix}^T$, a da bi ga zapisali pomoću kuta ϕ koristimo adicijske formule i vrijedi sljedeće:

$$\begin{aligned} \cos\alpha &= \cos\left(\frac{\pi}{2} - \phi\right) = \cos\frac{\pi}{2}\cos(-\phi) + \sin\frac{\pi}{2}\sin(-\phi) = -\sin\phi \\ \sin\alpha &= \sin\left(\frac{\pi}{2} - \phi\right) = \cos\frac{\pi}{2}\sin(-\phi) + \sin\frac{\pi}{2}\cos(-\phi) = \cos\phi. \end{aligned}$$

Iz Slike 1. vidimo da rotacijom vektora $e_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$, dobivamo vektor $e_1' = \begin{bmatrix} \cos\phi & \sin\phi \end{bmatrix}^T$, a rotacijom vektora $e_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$, dobivamo vektor $e_2' = \begin{bmatrix} -\sin\phi & \cos\phi \end{bmatrix}^T$ iz čega slijedi da matrica $\mathbf{A}$ mora zadovoljavati sljedeće:

$$\begin{aligned} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} &= \begin{bmatrix} \cos\phi \\ \sin\phi \end{bmatrix} \\ \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} &= \begin{bmatrix} -\sin\phi \\ \cos\phi \end{bmatrix}. \end{aligned}$$

Pa lako vidimo da je $a_{11} = \cos\phi$, $a_{21} = \sin\phi$, $a_{12} = -\sin\phi$, $a_{22} = \cos\phi$, tj.

$$\mathbf{A} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}.$$

Provjerimo još je li operator koji određuje matricu $\mathbf{A}$ linearan. To je operator

$$\mathcal{A}(x) = \mathcal{A}(x_1e_1 + x_2e_2) = (x_1\cos\phi - x_2\sin\phi)e_1 + (x_1\sin\phi + x_2\cos\phi)e_2.$$

Neka su $x = x_1e_1 + x_2e_2, y = y_1e_1 + y_2e_2 \in \mathbb{R}^2$ proizvoljni vektori i neka su $\lambda, \mu \in \mathbb{R}$ proizvoljni skalari. Tada djelovanjem operatora $\mathcal{A}$ na linearnu kombinaciju $\lambda x + \mu y$ dobivamo

$$\begin{aligned} \mathcal{A}(\lambda x + \mu y) &= \mathcal{A}(\lambda(x_1e_1 + x_2e_2) + \mu(y_1e_1 + y_2e_2)) = \mathcal{A}((\lambda x_1 + \mu y_1)e_1 + (\lambda x_2 + \mu y_2)e_2) = \\ &= ((\lambda x_1 + \mu y_1)\cos\phi - (\lambda x_2 + \mu y_2)\sin\phi)e_1 + \\ &\quad +((\lambda x_1 + \mu y_1)\sin\phi + (\lambda x_2 + \mu y_2)\cos\phi)e_2 = \\ &= \lambda((x_1\cos\phi - x_2\sin\phi)e_1 + (x_1\sin\phi + x_2\cos\phi)e_2) + \\ &\quad +\mu((y_1\cos\phi - y_2\sin\phi)e_1 + (y_1\sin\phi + y_2\cos\phi)e_2) = \\ &= \lambda\mathcal{A}(x) + \mu\mathcal{A}(y), \end{aligned}$$

što znači da je operator linearan, pa je djelovanje matrice na proizvoljne vektore jedinstveno određeno djelovanjem na vektore baze, stoga će matrica $\mathbf{A}$ sve vektore rotirati za kut ϕ suprotno od smjera kazaljke na satu. Više o linearnim operatorima u [5].

Za matricu rotacije $\mathbf{A} \in \mathbb{R}^{2\times 2}$ vrijedi:

1. Matrica $\mathbf{A}$ je regularna.

$$\det\mathbf{A} = \begin{vmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{vmatrix} = \cos\phi\cos\phi - (-\sin\phi)\sin\phi = \cos^2\phi + \sin^2\phi = 1 \neq 0.$$

2. Matrica $\mathbf{A}$ je ortogonalna, tj. vrijedi $\mathbf{A}^{-1} = \mathbf{A}^T$.

$$\mathbf{A}\mathbf{A}^T = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{A}^T\mathbf{A}$$

3. Matrica $\mathbf{A}$ čuva $2-normu$ vektora.
   Neka je $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$, za koji je $||x||_2 = x_1^2 + x_2^2$. Djelovanjem matrice $\mathbf{A}$ na vektor x dobivamo vektor $Ax = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1\cos\phi - x_2\sin\phi \\ x_1\sin\phi + x_2\cos\phi \end{bmatrix}$, te vrijedi:

$$\begin{aligned}
||Ax||_2 &= (x_1\cos\phi - x_2\sin\phi)^2 + (x_1\sin\phi + x_2\cos\phi)^2 \\
&= x_1^2\cos^2\phi - 2x_1x_2\cos\phi\sin\phi + x_2^2\sin^2\phi + x_1^2\sin^2\phi + 2x_1x_2\sin\phi\cos\phi + x_2^2\cos^2\phi \\
&= x_1^2(\cos^2\phi + \sin^2\phi) + x_2^2(\cos^2\phi + \sin^2\phi) \\
&= x_1^2 + x_2^2 = ||x||_2.
\end{aligned}$$

4. Produkt matrica rotacije je opet matrica rotacije.
   Neka su $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{2\times 2}$ matrice rotacije. Pogledajmo kako će izgledati njihov produkt.

$$\begin{aligned}
\mathbf{AB} &= \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} = \\
&= \begin{bmatrix} \cos\phi\cos\psi - \sin\phi\sin\psi & -\cos\phi\sin\psi - \sin\phi\cos\psi \\ \sin\phi\cos\psi + \cos\phi\sin\psi & -\sin\phi\sin\psi + \cos\phi\cos\psi \end{bmatrix} = \\
&= \begin{bmatrix} \cos(\phi+\psi) & -\sin(\phi+\psi) \\ \sin(\phi+\psi) & \cos(\phi+\psi) \end{bmatrix}
\end{aligned}$$

   Dobivena matrica je matrica rotacije za kut $(\phi + \psi)$.

Jednako tako možemo pokazati da je matrica $\mathbf{M}(i,k,\phi) \in \mathbb{R}^{n\times n}$ matrica rotacije za kut ϕ u ravnini iOk ukoliko je zadana na sljedeći način:

$$\mathbf{M}(i,k,\phi) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cos\phi & \cdots & -\sin\phi & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \sin\phi & \cdots & \cos\phi & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix},$$

pri čemu se podmatrica $\begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$ nalazi na presjeku i-tog i k-tog retka odnosno stupca.

Za ovako zadanu matricu vrijede ista svojstva kao i za matrice rotacije iz $\mathbb{R}^{2\times 2}$.

2.2 Givensove rotacije

Givensove rotacije $\mathbf{G} \in \mathbb{R}^{2\times 2}$ su matrice rotacije koje rotiraju tako da ponište drugu komponentu vektora, tj. ako imamo vektor $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \in \mathbb{R}^2$ matrica $\mathbf{G}$ će na njega djelovati na sljedeći način:

$$\mathbf{G}x = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ 0 \end{bmatrix}.$$

Iz ovog sustava određujemo koliki su $\sin\phi$ i $\cos\phi$.

$$\begin{aligned} x_1\cos\phi - x_2\sin\phi &= y_1 && (1)\\ x_1\sin\phi + x_2\cos\phi &= 0 && (2) \end{aligned}$$

Iz $(1-2)$ slijedi da u slučaju kada je $x_2 = 0$, tada je matrica G jednaka jediničnoj matrici I, inače imamo da je

$$x_2\cos\phi = -x_1\sin\phi \tag{3}$$

i iskoristimo li jednu od osnovnih trigonometrijskih jednakosti $\sin^2\phi + \cos^2\phi = 1$ imamo:

$$x_2\cos\phi = -x_1\sin\phi = -x_1\sqrt{1-\cos^2\phi}.$$

Kvadriranjem dobivamo:

$$\begin{aligned} x_2^2\cos^2\phi &= x_1^2 - x_1^2\cos^2\phi \\ x_1^2 &= (x_1^2 + x_2^2)\cos^2\phi, \end{aligned}$$

te za $x_1^2 + x_2^2 \neq 0$, tj. $|| \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T || \neq 0$ vrijedi

$$\cos^2\phi = \frac{x_1^2}{x_1^2 + x_2^2}, \text{ tj. } \cos\phi = \frac{\pm x_1}{\sqrt{x_1^2 + x_2^2}}.$$

Ukoliko želimo da y_1 bude pozitivan uzimamo da je $\cos\phi = \dfrac{x_1}{\sqrt{x_1^2 + x_2^2}}$, a tada je iz (3) $\sin\phi = \dfrac{-x_2}{\sqrt{x_1^2 + x_2^2}}$, tj. $y_1 = \dfrac{x_1^2 + x_2^2}{\sqrt{x_1^2 + x_2^2}}$ i dobivamo vektor $y = \begin{bmatrix} y_1 & 0 \end{bmatrix}^T$.

Jednako tako i za matrice iz $\mathbb{R}^{n\times n}$ možemo konstruirati odgovarajuće Givensove rotacije. To bi bile matrice oblika

$$\mathbf{G}(i,k,\phi) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cos\phi & \cdots & -\sin\phi & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \sin\phi & \cdots & \cos\phi & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix},$$

pri čemu se podmatrica $\begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$ nalazi na presjeku i-tog i k-tog retka odnosno stupca, a u kojima se $\sin\phi$ i $\cos\phi$ računaju na prethodno opisani način, tj. za $x \in \mathbb{R}^n$

$$\begin{aligned} \cos\phi &= \frac{x_i}{\sqrt{x_i^2 + x_k^2}} \\ \sin\phi &= \frac{-x_k}{\sqrt{x_i^2 + x_k^2}}. \end{aligned} \tag{4}$$

Tada djelovanjem Givensove rotacije $\mathbf{G} \in \mathbb{R}^{n\times n}$ na vektor x dobivamo vektor $y \in \mathbb{R}^n$ oblika:

$$y_j = \begin{cases} x_j, & \text{za } j \neq i,\ j \neq k \\ 0, & \text{za } j = k \\ \dfrac{x_i^2 + x_k^2}{\sqrt{x_i^2 + x_k^2}}, & \text{za } j = i \end{cases},$$

odnosno poništavamo k-tu komponentu vektora x, j-tu mijenjamo dok ostale komponente ostaju nepromjenjene.

U praksi postoje i puno efikasniji načini računanja $\sin\phi$ i $\cos\phi$ od (4). Primjer je i sljedeći algoritam koji nam osigurava da neće doći do "overflow-a". Za više detalja vidjeti [4].

ALGORITAM 1. (za izračunavanje kuta rotacije)
ulaz: x_i, x_k
izlaz: $\cos\phi$, $\sin\phi$
1. **if** $x_k = 0$
2. **then** $\cos\phi = 1$, $\sin\phi = 0$
3. **else if** $|x_k| > |x_i|$
4. $\tau = \dfrac{-x_i}{x_k}$, $\sin\phi = \dfrac{1}{\sqrt{1+\tau^2}}$, $\cos\phi = \sin\phi\tau$
5. **else** $\tau = \dfrac{-x_k}{x_i}$, $\cos\phi = \dfrac{1}{\sqrt{1+\tau^2}}$, $\sin\phi = \cos\phi\tau$
6. **end**

Možemo primjetiti da se u algoritmu 1. zapravo $\cos\phi = \dfrac{x_i}{\sqrt{x_i^2 + x_k^2}}$ dijeli većim brojem između x_i i x_k kako bi smo osigurali da nećemo imati dijeljenje s nekim jako malim brojem.

Također primjetimo da prvi i drugi korak algoritma 1. obuhvaćaju slučaj kada je $x_k = 0$, tj. kada se nema što poništavati pa je matrica $\mathbf{G}$ jednaka jediničnoj matici $\mathbf{I}$.

2.2.1 Množenje matrica Givensovim rotacijama

Neka je $\mathbf{A} \in \mathbb{R}^{m\times n}$ i $\mathbf{G}_l(i,k,\phi) \in \mathbb{R}^{m\times m}$. Zbog jednostavne strukture matrice $\mathbf{G}$ vidimo da množenjem matrice $\mathbf{A}$ slijeva matricom $\mathbf{G}$ utječemo samo na dva retka

$$\mathbf{G}_l(i,k,\phi)\mathbf{A}([i,k],:) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{in} \\ a_{k1} & a_{k2} & \cdots & a_{kn} \end{bmatrix}$$

i to množenje odvija se po algoritmu 2.

Napomena 1. $\mathbf{A}([i,k],:)$ *je Matlab oznaka za i-ti i k-ti redak matrice A.*

Napomena 2. *S* $\mathbf{G}_l(i,k,\phi)$ *označavamo Givensovu matricu koja u l-tom stupcu poništava k-ti element, a samo i-ti mijenja u l-tom stupcu.*

ALGORITAM 2. (za množenje matrice A slijeva Givensovom matricom)
ulaz: $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{G}_l(i,k,\phi) \in \mathbb{R}^{n\times n}$ (gdje l označava stupac u kojem se poništava)
izlaz: $\mathbf{A}$ s poništenom odgovarajućom komponentom a_{kl} (vidi **Primjer 2.1.**)
1. **for** $j = 1 : n$
2. $\quad y_1 = \mathbf{A}(i,j)$
3. $\quad y_2 = \mathbf{A}(k,j)$
4. $\quad \mathbf{A}(i,j) = y_1\cos\phi - y_2\sin\phi$
5. $\quad \mathbf{A}(k,j) = y_1\sin\phi + y_2\cos\phi$
6. **end**

Ovaj algoritam zahtjeva $6n$ operacija. Imamo 2 retka sa n stupaca matrice $\mathbf{A}$ koji sudjeluju u množenju. U 4. redu algoritma imamo dva množenja i oduzimanje, a u 5. redu imamo dva množenja i zbrajanje, što je 6 operacija za svaki stupac. Djelovanjem na svih n stupaca dobijamo $6n$ operacija.

Primjer 2.1. *Neka su* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *neka općenita matrica i* $\mathbf{G}_i(i,k,\phi) \in \mathbb{R}^{m\times m}$ *Givensova matrica rotacije gdje je* ϕ *računat za zadane* a_{ii} *i* a_{ki}*, što znači da ćemo poništavanje u ovom primjeru vršiti u i-tom stupcu.*

$$\mathbf{G}_i(i,k,\phi)\mathbf{A} = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cos\phi & \cdots & -\sin\phi & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \sin\phi & \cdots & \cos\phi & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1k} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ii} & \cdots & a_{ik} & \cdots & a_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{ki} & \cdots & a_{kk} & \cdots & a_{kn} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mi} & \cdots & a_{mk} & \cdots & a_{mn} \end{bmatrix} =$$

$$
= \begin{bmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1k} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a'_{i1} & \cdots & a'_{ii} & \cdots & a'_{ik} & \cdots & a'_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a'_{k1} & \cdots & a'_{ki} & \cdots & a'_{kk} & \cdots & a'_{kn} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mi} & \cdots & a_{mk} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1k} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a'_{i1} & \cdots & a'_{ii} & \cdots & a'_{ik} & \cdots & a'_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a'_{k1} & \cdots & 0 & \cdots & a'_{kk} & \cdots & a'_{kn} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mi} & \cdots & a_{mk} & \cdots & a_{mn} \end{bmatrix},
$$

pri čemu su

$$
a'_{ij} = a_{ij} \cos\phi - a_{kj} \sin\phi, \quad j = 1, \ldots, n
$$

$$
a'_{kj} = a_{ij} \sin\phi + a_{kj} \cos\phi. \quad j = 1, \ldots, n
$$

Vidimo da se poništava komponenta koja se zadaje kao druga pri izračunavanju kuta ϕ.

Vrlo slično, množenjem matrice $\mathbf{A}$ zdesna s $\mathbf{G}_l^T(i,j,\phi)$ utječemo samo na dva stupca

$$
\mathbf{A}(:,[i,k])\mathbf{G}_l^T(i,j,\phi) = \begin{bmatrix} a_{1i} & a_{1k} \\ a_{2i} & a_{2k} \\ \vdots & \vdots \\ a_{mi} & a_{mk} \end{bmatrix} \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}
$$

i ono zahtjeva $6m$ operacija. Do broja računskih operacija dolazimo kao i u prethodnom algoritmu, samo što u ovom slučaju imamo 2 stupca s m redaka. Ovo množenje se odvija po algoritmu 3.

Napomena 3. $\mathbf{A}(:,[i,k])$ *je Matlab oznaka za i-ti i k-ti stupac matrice A.*

Napomena 4. *Matrica $\mathbf{G}_l^T(i,k,\phi)$ u l-tom retku poništavati k-ti element, a mijenjati i-ti.*

ALGORITAM 3. (za množenje matrice A zdesna Givensovom matricom)
ulaz: $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{G}_l^T(i,k,\phi) \in \mathbb{R}^{n\times n}$ (gdje l označava redak u kojem se poništava)
izlaz: $\mathbf{A}$ s poništenom odgovarajućom komponentom a_{lk} (vidi **Primjer 2.2.**)
1. **for** $j = 1 : m$
2. $\quad y_1 = \mathbf{A}(j,i)$
3. $\quad y_2 = \mathbf{A}(j,k)$
4. $\quad \mathbf{A}(j,i) = y_1 \cos\phi - y_2 \sin\phi$
5. $\quad \mathbf{A}(j,k) = y_1 \sin\phi + y_2 \cos\phi$
6. **end**

Primjer 2.2. *Neka su* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *neka općenita matrica i* $\mathbf{G}_i(i,k,\phi) \in \mathbb{R}^{n\times n}$ *Givensova matrica rotacije gdje je* ϕ *računat za zadane* a_{ii} *i* a_{ik}.

$$\mathbf{AG}^T = \begin{bmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1k} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ii} & \cdots & a_{ik} & \cdots & a_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{ki} & \cdots & a_{kk} & \cdots & a_{kn} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mi} & \cdots & a_{mk} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cos\phi & \cdots & -\sin\phi & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \sin\phi & \cdots & \cos\phi & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} a_{11} & \cdots & a'_{1i} & \cdots & a'_{1k} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a'_{ii} & \cdots & a'_{ik} & \cdots & a_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a'_{ki} & \cdots & a'_{kk} & \cdots & a_{kn} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a'_{mi} & \cdots & a'_{mk} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a'_{1i} & \cdots & a'_{1k} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a'_{ii} & \cdots & 0 & \cdots & a_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a'_{ki} & \cdots & a'_{kk} & \cdots & a_{kn} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a'_{mi} & \cdots & a'_{mk} & \cdots & a_{mn} \end{bmatrix},$$

pri čemu su

$$a'_{ji} = a_{ji}\cos\phi - a_{jk}\sin\phi, \quad j = 1,\ldots,m$$
$$a'_{jk} = a_{ji}\sin\phi + a_{jk}\cos\phi. \quad j = 1,\ldots,m$$

Možemo zaključiti da se množenjem slijeva poništavanje vrši po stupcima dok se množenjem zdesna poništavanje vrši po retcima.

Također primjetimo da smo uveli oznaku stupca u kojem poništavamo kada množimo slijeva (tj. retka kada množimo zdesna), s obzirom da imamo djelovanje na matricu, a ne na jedan vektor.

2.3 Egzistencija QR dekompozicije

Kako smo već spomenuli u uvodu, **QR** dekompozicija je rastav matrice **A** na dvije matrice $\mathbf{A} = \mathbf{QR}$, za koje vrijedi da je **Q** ortogonalna, a **R** gornja trokutasta.

Teorem 2.1. *Za bilo koju matricu* $\mathbf{A} \in \mathbb{R}^{m\times n}$, *gdje je* $m \geq n$ *postoje matrice* $\mathbf{Q} \in \mathbb{R}^{m\times m}$ *i* $\mathbf{R} \in \mathbb{R}^{m\times n}$ *takve da je* $\mathbf{A} = \mathbf{QR}$, *pri čemu je* **Q** *ortogonalna, a* **R** *gornja trokutasta matrica.*

Dokaz ovog teorema pomoću Gramm-Schimidtovog postupka ortogonalizacije možete naći u [3, str. 84], a ovdje ćemo primjerom pokazati kako izgleda QR dekompozicija kada se vrši pomoću Givensovih rotacija.

Primjer 2.3. *Neka su* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *neka općenita matrica, gdje je* $m \geq n$*. Rastavimo, pomoću Givensovih rotacija, matricu* $\mathbf{A}$ *na matrice* $\mathbf{Q} \in \mathbb{R}^{m\times m}$ *(ortogonalna) i* $\mathbf{R} \in \mathbb{R}^{m\times n}$ *(gornja trokutasta) tako da vrijedi* $\mathbf{A} = \mathbf{QR}$*.*

Kako Givensove rotacije poništavaju samo jedan element pa s $\mathbf{G}_l(i,k,\phi) \in \mathbb{R}^{m\times m}$ *matricu koja poništava element* a_{kl}*, a kut* ϕ *izračunava za zadane* a_{il} *i* a_{kl}*. Poništavanje ćemo vršiti po slijedećem algoritmu:*

ALGORITAM 4. *(QR dekompozicija pomoću Givensovih rotacija)*

ulaz: $\mathbf{A} \in \mathbb{R}^{m\times n}$

izlaz: $\mathbf{R} \in \mathbb{R}^{m\times n}$ *gornje trokutasta,* $\mathbf{Q} \in \mathbb{R}^{m\times m}$ *ortogonalna*

1. $\mathbf{R} = \mathbf{A}$, $\mathbf{Q} = \mathbf{I}$
2. ***for*** $l = 1 : n$
3.   ***for*** $i = m : -1 : l+1$
4.   *izračunati kut* ϕ *(tj. elemetne Givensove rotacije* $\mathbf{G}_l(i-1,i,\phi)$*)*
5. $\mathbf{R} = \mathbf{G}_l(i-1,i,\phi)\mathbf{R}$ *(ponišava element* a_{il}*)*
6. $\mathbf{Q} = \mathbf{Q}\mathbf{G}_l(i-1,i,\phi)^T$
7.   ***end***
8. ***end***

Koristeći ovaj algoritam za poništavanje elemenata u prvom stupcu matrice

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

potrebno je pomnožiti matricu $\mathbf{A}$ *redom matricama* $\mathbf{G}_1(m-1,m,\phi)$*,*$\mathbf{G}_1(m-2,m-1,\phi)$*,...,* $\mathbf{G}_1(1,2,\phi)$*. Na taj način ćemo dobiti matricu* $\mathbf{A}'$*.*

$$\mathbf{A}' = \mathbf{G}_1(1,2,\phi)\cdots\mathbf{G}_1(m-2,m-1,\phi)\mathbf{G}_1(m-1,m,\phi)\mathbf{A} = \begin{bmatrix} a'_{11} & a'_{12} & \cdots & a'_{1n} \\ 0 & a'_{22} & \cdots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a'_{m2} & \cdots & a'_{mn} \end{bmatrix}$$

Analogno, za poništavanje elemenata u j-tom stupcu množit ćemo prethodno dobivenu matricu redom matricama $\mathbf{G}_j(m-1,m,\phi)$*,*$\mathbf{G}_j(m-2,m-1,\phi)$*,...,* $\mathbf{G}_j(j,j+1,\phi)$*.*

U zadnjem koraku imamo poništavanje elemenata u zadnjem stupcu i to množeći matricama $\mathbf{G}_n(m-1,m,\phi)$*,* $\mathbf{G}_n(m-2,m-1,\phi)$*, ...,* $\mathbf{G}_n(n,n+1,\phi)$ *i na taj način smo dobili* *gornje trokutastu matricu* $\mathbf{R}$.

$$\begin{aligned}\mathbf{R} &= \mathbf{G}_n(n,n+1,\phi)\cdots\mathbf{G}_n(m-1,m,\phi)\cdots\mathbf{G}_j(j,j+1,\phi)\cdots\\ &\quad\mathbf{G}_j(m-1,m,\phi)\cdots\mathbf{G}_1(1,2,\phi)\cdots\mathbf{G}_1(m-1,m,\phi)\mathbf{A} =\end{aligned}$$

$$= \mathbf{Q}^T\mathbf{A} = \begin{bmatrix} a'_{11} & a'_{12} & \cdots & a'_{1,n-1} & a'_{1n} \\ 0 & a'_{22} & \cdots & a'_{2,n-1} & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a'_{n-1,n-1} & a'_{n-1,n} \\ 0 & 0 & \cdots & 0 & a'_{nn} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

S obzirom da su Givensove matrice rotacije ortogonalne matrice vidimo da je matrica $\mathbf{Q}$ *koja zadovoljava jednakost* $\mathbf{A} = \mathbf{QR}$ *jednaka*

$$\begin{aligned}\mathbf{Q} &= (\mathbf{G}_n(n,n+1,\phi)\cdots\mathbf{G}_n(m-1,m,\phi)\cdots\mathbf{G}_j(j,j+1,\phi)\cdots\\ &\quad\mathbf{G}_j(m-1,m,\phi)\cdots\mathbf{G}_1(1,2,\phi)\cdots\mathbf{G}_1(m-1,m,\phi))^T\\ &= \mathbf{G}_1(m-1,m,\phi)^T\cdots.\mathbf{G}_1(1,2,\phi)^T\cdots\mathbf{G}_j(m-1,m,\phi)^T\\ &\quad\cdots\mathbf{G}_j(j,j+1,\phi)^T\cdots\mathbf{G}_n(m-1,m,\phi)^T\cdots\mathbf{G}_n(n,n+1,\phi)^T\end{aligned}$$

Na taj način dobili smo tražene matrice $\mathbf{Q} \in \mathbb{R}^{m\times m}$ *i* $\mathbf{R} \in \mathbb{R}^{m\times n}$.

Napomena 5. *Svaka matrica* $\mathbf{G}_i$ *ima "svoj" kut* ϕ, *tj. ne radi se o istim kutu, ali radi jednostavnosti pisanja smo koristili istu oznaku.*

Izračunavanje matrice $\mathbf{R}$ u ovom primjeru zahtjeva $3n^2(\frac{m-n}{3})$ operacija, vidi [2] (bez obzira što u množenju sudjeluju samo dva stupca matrica koju poništavamo). Zbog toga se Givensove rotacije puno češće primjenjuju kada su u pitanju strukturirane matrice s puno nula ispod glavne dijagonale, ali o tome više u sljedećem poglavlju.

3 Primjena

3.1 Linearni problem najmanjih kvadrata

U ovom poglavlju objasnit ćemo ukratko što je problem najmanjih kvadrata te kako se za njegovo rješavanje koriste Givensove rotacije, tj. $\mathbf{QR}$ dekompozicija. Metoda najmanjih kvadrata se koristi kada u sustavu $\mathbf{Ax} = \mathbf{b}$ ima više jednadžbi nego nepoznanica.

Problem najmanjih kvadrata se često koristi u raznim tehničkim primjenama, kao i u elektrotehnici, fizici, geografiji, ekonomiji (linearna regresija).

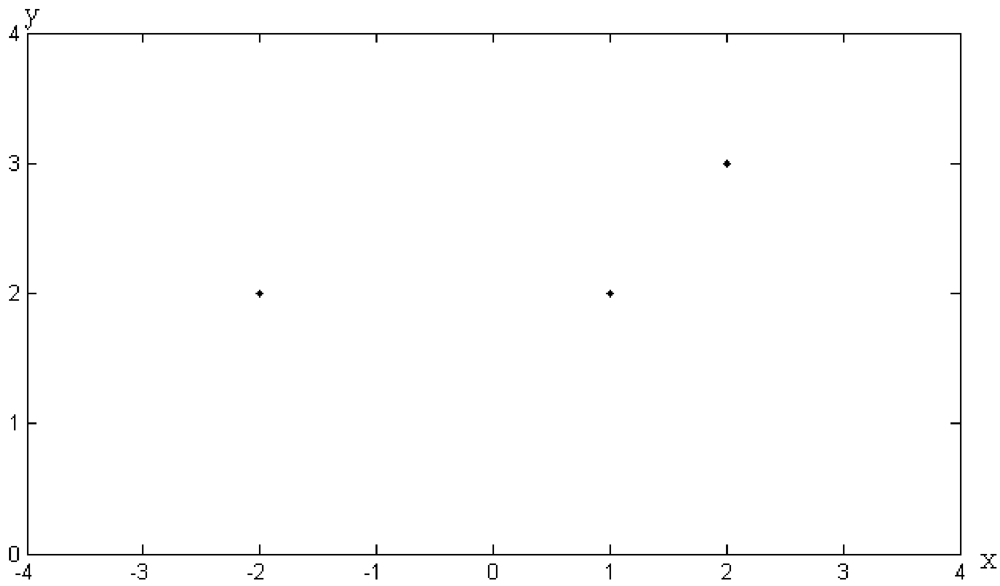
Upravo spomenutu linearnu regresiju najlakše je objasniti na primjeru.

Primjer 3.1. *Neka su zadane tri točke u ravnini*

x	-2	1	2
y	2	2	3

kao na slici 2. Kada bi imali jedinstveni pravac koji prolazi kroz sve zadane točke, tada bi za svaku točku (x_i, y_i), $i = 1, 2, 3$ vrijedilo

$$kx_i + l = y_i.$$



Slika 2. Zadane točke u ravnini

U navedenom primjeru imamo sljedeći sustav linearnih jednadžbi:

$$\begin{aligned} -2k + l &= 2 \\ 1k + l &= 2 \\ 2k + l &= 3 \end{aligned}$$

Ovaj sustav s tri jednadžbi i dvije nepoznanice k i l ima sljedeći matrični oblik

$$\begin{bmatrix} -2 & 1 \\ 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} k \\ l \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix}$$

odnosno imamo sustav $\mathbf{Ax} = \mathbf{b}$, *gdje je*

$$A = \begin{bmatrix} -2 & 1 \\ 1 & 1 \\ 2 & 1 \end{bmatrix}, \quad x = \begin{bmatrix} k \\ l \end{bmatrix}, \quad b = y = \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix}.$$

Kada bi ovaj sustav imao rješenje, tada bi vrijedio $Ax - b = 0$, tj. $||Ax - b||_2 = 0$. No, kako ovaj sustav nema rješenje cilj nam je minimizirati rezidual $r = Ax - b$, tj. minimizirati normu $||Ax - b||_2$ i na taj način dobiti pravac koji "najbolje" aproksimira zadane podatke.

Minimizacija norme $||Ax - b||_2$, ekvivalentna je minimizaciji kvadrata te iste norme, tj. minimiziramo $||Ax - b||_2^2 = \sum_{i=1}^{3}(kx_i + l - y_i)^2$. Naziv problem najmanjih kvadrata imamo jer minimiziramo sumu kvadrata razlike dobivene vrijednosti $(kx_i + l)$ i prave vrijednost (y_i).

Primjer ćemo riješiti u nastavku rada.

Problem iz prethodnog primjera može biti i višedimenzionalan. Tada imamo matricu $\mathbf{A} \in \mathbb{R}^{m\times n}$, takvu da je $m \geq n$ ("dugačka" matrica) i vektor $b \in \mathbb{R}^m$, a tražimo $x \in \mathbb{R}^n$ koji minimizira normu reziduala $r = Ax - b$, tj. normu $||Ax - b||_2$.

Postoje različiti načini rješavanja linearnog problema najmanjih kvadrata (LPNK). Neki od njih su sustav normalnih jednadžbi i rješavanje pomoću dekompozicije na singularne vrijednost (vidi [4]), ali u ovom radu ćemo pokazati kako se LPNK rješava pomoću QR dekompozicije.

3.2 Rješavanje LPNK pomoću QR dekompozicije

Kako smo već rekli, rješavanje LPNK svodi se na minimizaciju norme reziduala, tj. na minimizaciju norme $||Ax - b||_2$, gdje su $\mathbf{A} \in \mathbb{R}^{m\times n}$, gdje je $m \geq n$ i $b \in \mathbb{R}^m$. Neka je $\text{rang}(A) = r_A \leq n \leq m$ rang matrice A, te neka je $\mathbf{A} = \mathbf{QR}$, QR dekompozicija matrice A dobivena pomoću Givensovih rotacija, gdje je $\mathbf{Q} \in \mathbb{R}^{m\times m}$ ortogonalna matrica, a $\mathbf{R} \in \mathbb{R}^{m\times n}$ gornja trokutasta matrica.

Kako matrica $\mathbf{A}$ ne mora biti punog ranga stupaca ($r_A < n$), razlikujemo dva različita slučaja. U slučaj kada matrica nije punog ranga stupaca rješenje x LPNK nije jedinstveno, za više pogledati u [1], a u ovom ćemo se radu koncentrirat na slučaj kada je matrica $\mathbf{A}$ punog ranga stupaca.

Matrice $\mathbf{A}$ je punog ranga po stupcima, ali ne mora biti kvadratna, stoga za nju računamo reduciranu QR dekompoziciju

$$A = QR = [Q' \quad Q''] \begin{bmatrix} R' \\ 0 \end{bmatrix} = Q'R'.$$

To znači da smo za matricu $A \in \mathbb{R}^{m\times n}$, gdje je $m \geq n$, dobili $A = Q'R'$, gdje je $Q' \in \mathbb{R}^{m\times n}$ ortogonalna (sadrži prvih n stupaca matrice Q), a $R' \in \mathbb{R}^{n\times n}$ gornje trokutasta matrica (sadrži prvih n redaka matrice R).

Napomena 6. *U Matlabu bismo reducirane Q' i R' zapisali na sljedeći način:*

$$Q' = Q(:, 1:n)),$$

$$R' = R(1:n, 1:n).$$

Kako smo rekli da promatramo matrice koje su punog ranga po stupcima, neka je $r_A = n$ i $x \in \mathbb{R}^n$. Kako je Q matrica rotacije vrijedi svojstvo da čuva normu vektora, pa stoga možemo pisati

$$||r||_2 = ||Ax - b||_2 = ||QRx - b||_2 = ||Q(Rx - Q^T b)||_2 = ||Rx - Q^T b||_2.$$

Zapišimo vektor $Q^T b$ na sljedeći način:

$$Q^T b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix},$$

gdje je b_1 r_A dimenzionalni vektor, tj. $b_1 = Q'^T b$. Tada imamo

$$||Ax - b||_2^2 = ||Rx - Q^T b||_2^2 = || \begin{bmatrix} R'x \\ 0 \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} ||_2^2 = ||R'x - b_1||_2^2 + ||b_2||_2^2,$$

iz čega slijedi da će rezidual biti minimalan kada je $R'x = b_1 = Q'^T b$. Tada je rješenje x LPNK dobije rješavanjem trokutastog sustava $R'x = Q'^T b$.

Sljedeći algoritam služi za rješavanje LPNK u slučaju matrice punog ranga stupaca.
ALGORITAM 5.

ulaz: $A \in \mathbb{R}^{m \times n}$, $m \geq n$ i rang(A), $b \in \mathbb{R}^m$

izlaz: $x \in \mathbb{R}^n$ koji minimizira $||Ax - b||_2$

1. izračunaj reduciranu QR dekompoziciju matrice $A = Q'R'$
2. izračunaj $Q'^T b \in \mathbb{R}^n$
3. riješi $R'x = Q'^T b$ pomoću povratnih supstitucija.

Napomena 7. *U radu su u svim rješenjima brojevi zaokruženi na* 4 *decimale.*

Riješimo sada **Primjer 3.1.**
Imamo za riješiti sustav $\mathbf{Ax} = \mathbf{b}$, *gdje je*

$$A = \begin{bmatrix} -2 & 1 \\ 1 & 1 \\ 2 & 1 \end{bmatrix}, \quad x = \begin{bmatrix} k \\ l \end{bmatrix}, \quad b = y = \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix}.$$

Prvo ćemo poništiti element a_{13} i to pomoću matrice $G_1(2,3,\phi)$, koja glasi

$$G_1(2,3,\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.4472 & -0.8944 \\ 0 & 0.8944 & -0.4472 \end{bmatrix}.$$

Množenjem matrice A i vektora b s $G_1(2,3,\phi)$ dobijamo

$$A' = G_1(2,3,\phi)A = \begin{bmatrix} -2.0000 & 1.0000 \\ -2.2361 & -1.3416 \\ 0 & 0.4472 \end{bmatrix}, \quad b' = G_1(2,3,\phi)b = \begin{bmatrix} 2.0000 \\ -3.5777 \\ 0.4472 \end{bmatrix}.$$

Zatim želimo poništiti element a_{12} i za to poništavanje ćemo koristiti matricu $G_1(1,2,\phi)$, koja glasi

$$G_1(1,2,\phi) = \begin{bmatrix} -0.6667 & -0.7454 & 0 \\ 0.7454 & -0.6667 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

te dobijamo

$$A'' = G_1(1,2,\phi)G_1(2,3,\phi)A = \begin{bmatrix} 3.0000 & 0.3333 \\ 0 & 1.6398 \\ 0 & 0.4472 \end{bmatrix}, \quad b'' = G_1(1,2,\phi)G_1(2,3,\phi)b = \begin{bmatrix} 1.3333 \\ 3.8759 \\ 0.4472 \end{bmatrix}.$$

Preostalo nam je još jedno poništavanje i to elementa a_{23}. Givensova matrica koju ćemo koristiti za to poništavanje glasi

$$G_2(2,3,\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.9648 & 0.2631 \\ 0 & -0.2631 & 0.9648 \end{bmatrix},$$

te dobivamo

$$R = G_2(2,3,\phi)G_1(1,2,\phi)G_1(2,3,\phi)A = \begin{bmatrix} 3.0000 & 0.3333 \\ 0 & 1.6997 \\ 0 & 0 \end{bmatrix},$$

$$Q^T b = G_2(2,3,\phi)G_1(1,2,\phi)G_1(2,3,\phi)b = \begin{bmatrix} 1.3333 \\ 3.8570 \\ -0.5883 \end{bmatrix}.$$

Sada povratnim supstitucijama riješimo sustav $R'x = b_1 = Q'^T b$, gdje je

$$R' = \begin{bmatrix} 3.0000 & 0.3333 \\ 0 & 1.6997 \end{bmatrix}, \quad Q'^T b = \begin{bmatrix} 1.3333 \\ 3.8570 \end{bmatrix}$$

Traženo rješenje ovog LPNK je

$$x = \begin{bmatrix} 0.1923 \\ 2.2692 \end{bmatrix}.$$

U slučaju kada imamo kvadratnu matricu punog ranga $A \in \mathbb{R}^{n\times n}$ rješavamo sustav linearnih jednadžbi. Takav sustav imat će jedinstveno rješenje do kojeg se također može doći pomoću algoritma 5., iako se u tom slučaju neće računati reducirana QR dekompozicija, nego obična.

Primjetimo također da pri rješavanju sustava, formiranje matrice Q nije potrebno jer smo automatski množili i vektor b s Givensovim rotacijama (pogledati Matlab kod 4.).

Navedimo još na kraju ovog potpoglavlja jedan primjer LPNK s malo većom matricom.

Primjer 3.2. *Neka je matrica $A \in \mathbb{R}^{10\times 5}$ zadana na sljedeći način*

$$a_{ij} = \begin{cases} 2i & , za\ i{=}j \\ 100 & , inače \end{cases},$$

a vektor $b \in \mathbb{R}^{10}$, je jednak $b = [5 \quad 10 \quad 15 \quad 20 \quad 25 \quad 30 \quad 35 \quad 40 \quad 45 \quad 50]^T$. Riješite pripadni sustav $Ax = b$.

Koristeći Matlab i funkciju za određivanje QR dekompozicije pomoću Givensovih rotacija (vidi Prilog - kodovi) dobili smo reduciranu QR dekompoziciju, tj.

$$Q' = \begin{bmatrix} 0.0067 & 0.7353 & -0.4388 & -0.3126 & -0.2417 \\ 0.3333 & -0.6406 & -0.4479 & -0.3191 & -0.2467 \\ 0.3333 & 0.0782 & 0.7700 & -0.3259 & -0.2520 \\ 0.3333 & 0.0782 & -0.0448 & 0.8300 & -0.2575 \\ 0.3333 & 0.0782 & -0.0448 & -0.0298 & 0.8653 \\ 0.3333 & 0.0782 & -0.0448 & -0.0298 & -0.0209 \\ 0.3333 & 0.0782 & -0.0448 & -0.0298 & -0.0209 \\ 0.3333 & 0.0782 & -0.0448 & -0.0298 & -0.0209 \\ 0.3333 & 0.0782 & -0.0448 & -0.0298 & -0.0209 \\ 0.3333 & 0.0782 & -0.0448 & -0.0298 & -0.0209 \end{bmatrix}$$

$$R' = \begin{bmatrix} 300.0067 & 268.6607 & 269.3273 & 269.9940 & 270.6607 \\ 0 & 133.5568 & 64.7090 & 64.8654 & 65.0219 \\ 0 & 0 & -115.3756 & -38.8805 & -38.9700 \\ 0 & 0 & 0 & -106.9954 & -27.9489 \\ 0 & 0 & 0 & 0 & -101.5636 \end{bmatrix},$$

a zatim primjenom algoritma 5. dobijamo

$$x = [0.1692 \quad 0.1185 \quad 0.0655 \quad 0.0101 \quad -0.0478]^T$$

kao rješenje ovog linearnog problema najmanjih kvadrata.

3.3 Računanje QR dekompozicije strukturiranih matrica

U ovom poglavlju ćemo prokazati kako se Givensove rotacije koriste u izračunavanju QR dekompozicije strukturiranih matrica kao što su matrice u Hessenbergovoj formi i tridijagonalne matrice.

Definicija 1. *Za matricu $A \in \mathbb{R}^{n\times n}$ ćemo reći da je u gornjoj Hessenbergovoj formi ako vrijedi $a_{ij} = 0$, za $i - j \geq 2$, tj. matrica A je gornje trokutasta i ima još jednu sporednu dijagonalu ispod glavne.*

Ilustrirajmo primjerom ideju za izračunavanje QR dekompozicije.

Primjer 3.3. *Neka je $A \in \mathbb{R}^{5\times 5}$ matrica u gornjoj Hessenbergovoj formi. Tada je ona sljedećeg oblika*

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ 0 & a_{32} & a_{33} & a_{34} & a_{35} \\ 0 & 0 & a_{43} & a_{44} & a_{45} \\ 0 & 0 & 0 & a_{54} & a_{55} \end{bmatrix}.$$

S obzirom da trebamo poništiti samo elemente na sporednoj dijagonali ispod glavne, poništavanje ćemo vršiti dijagonalno, te na taj način izbjeći nepotrebna množenja.

Množenjem matrice A slijeva, redom Givensovim matricama $G_1(1,2,\phi)$ i $G_2(2,3,\phi)$ dobivamo matricu

$$G_2(2,3,\phi)G_1(1,2,\phi)A = \begin{bmatrix} a'_{11} & a'_{12} & a'_{13} & a'_{14} & a'_{15} \\ 0 & a'_{22} & a'_{23} & a'_{24} & a'_{25} \\ 0 & 0 & a'_{33} & a'_{34} & a'_{35} \\ 0 & 0 & a_{43} & a_{44} & a_{45} \\ 0 & 0 & 0 & a_{54} & a_{55} \end{bmatrix}.$$

Stoga množenjem još s matricama $G_3(3,4,\phi)$ i $G_4(4,5,\phi)$, dobivamo gornje trokutastu matricu

$$R = G_4(4,5,\phi)G_3(3,4,\phi)G_2(2,3,\phi)G_1(1,2,\phi)A = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} & r_{15} \\ 0 & r_{22} & r_{23} & r_{24} & r_{25} \\ 0 & 0 & r_{33} & r_{34} & r_{35} \\ 0 & 0 & 0 & r_{44} & r_{45} \\ 0 & 0 & 0 & 0 & r_{55} \end{bmatrix}.$$

Tada je matrica Q iz QR dekompozicije jednaka

$$\begin{aligned} Q &= (G_4(4,5,\phi)G_3(3,4,\phi)G_2(2,3,\phi)G_1(1,2,\phi))^T = \\ &= G_1(1,2,\phi)^T G_2(2,3,\phi)^T G_3(3,4,\phi)^T G_4(4,5,\phi)^T, \end{aligned}$$

te smo QR dekompoziciju dobili s četiri jednostavna množenja.

Možemo primjetiti razliku između ove strategije da dijagonalno poništavamo elemente i strategije primjenjene u algoritmu 4. gdje su se elementi poništavali po stupcima odozdo prema gore, počevši od zadnjeg elementa u prvom stupcu. Stoga u slučaju kada imamo matricu $A \in \mathbb{R}^{n\times n}$ u gornjoj Hessenbergovoj formu koristimo sljedeći algoritam za računanje QR dekompozicije:

ALGORITAM 5. (QR dekompozicija za matrice u gornjoj Hessenbergovoj formi)
ulaz: $\mathbf{A} \in \mathbb{R}^{n\times n}$ matrica u gornjoj Hessenbergovoj formi
izlaz: $\mathbf{R} \in \mathbb{R}^{n\times n}$ gornja trokutasta i $\mathbf{Q} \in \mathbb{R}^{n\times n}$ ortogonalna
1. $\mathbf{R} = \mathbf{A}$, $\mathbf{Q} = \mathbf{I}$
2. **for** $j = 1 : n-1$
3. izračunati ϕ za matricu $\mathbf{G}_j(j, j+1, \phi_j)$
4. $\mathbf{R} = \mathbf{G}_j(j, j+1, \phi_j)\mathbf{R}$
5. $\mathbf{Q} = \mathbf{Q}\mathbf{G}_j(j, j+1, \phi_j)^T$
6. **end**.

Primjer 3.4. *Matrica*

$$H = \begin{bmatrix} 0 & 12 & 5 & 3 & 0 \\ 1 & 3 & 9 & 0 & 31 \\ 0 & 4 & 4 & 7 & 17 \\ 0 & 0 & 3 & 8 & 5 \\ 0 & 0 & 0 & 6 & 11 \end{bmatrix}$$

je u gornjoj Hessenbergovoj formi. Prikažite je u QR dekompoziciji.

Koristeći funkciju za QR dekompoziciju matrica u gornjoj Hessenbergovoj formi uz pomoć Givensovih rotacija (vidi 4. u Prilog-kodovi) kao rezultat dobijemo $H = QR$, gdje su

$$Q = \begin{bmatrix} 0 & 0.9487 & 0.1878 & -0.0072 & -0.2544 \\ -1.0000 & 0 & 0 & 0 & 0 \\ 0 & 0.3162 & -0.5633 & 0.0216 & 0.7631 \\ 0 & 0 & -0.8047 & -0.0168 & -0.5935 \\ 0 & 0 & 0 & -0.9996 & 0.0283 \end{bmatrix}$$

$$R = \begin{bmatrix} -1.0000 & -3.0000 & -9.0000 & 0 & -31.0000 \\ 0 & 12.6491 & 6.0083 & 5.0596 & 5.3759 \\ 0 & 0.0000 & -3.7283 & -9.8169 & -13.5988 \\ 0 & 0 & -0.0000 & -6.0024 & -10.7127 \\ 0 & 0 & 0 & 0 & 10.3155 \end{bmatrix}$$

pri čemu smo zaokruživali na četiri decimale.

Osim određivanja QR dekompozicije, pomoću Givensovih rotacija općenite matrice se mogu svoditi na gornju Hessenbergovu formu. Tada u algoritmu 4. u 3. koraku umjesto

for $i = m : -1 : l + 1$ pišemo **for** $i = m : -1 : l + 2$ i na taj način će se poništavanje vršiti do elementra ispod dijagonale i matricu ćemo svesti na gornju Hessenbergovu formu.

Definicija 2. *Kažemo da je matrica* $T \in \mathbb{R}^{n\times n}$ *tridijagonalna ako je* $t_{ij} = 0$ *za* $|i - j| > 1$, *tj. matrica* T *ima glavnu i obje sporedne dijagonale.*

Primjer 3.5. *Matrica*

$$T = \begin{bmatrix} 1 & 12 & 0 & 0 & 0 \\ 8 & 2 & 9 & 0 & 0 \\ 0 & 4 & 3 & 7 & 0 \\ 0 & 0 & 3 & 13 & 5 \\ 0 & 0 & 0 & 5 & 11 \end{bmatrix}$$

je u tridijagonalnoj formi. Odredite QR dekompoziciju.

Primjenom istog koda kao u prethodnom primjeru, za matrice Q i R dobijamo

$$Q = \begin{bmatrix} -0.1240 & 0.9386 & -0.2349 & -0.1550 & 0.1564 \\ -0.9923 & -0.1173 & 0.0294 & 0.0194 & -0.0196 \\ 0 & 0.3245 & 0.6900 & 0.4554 & -0.4595 \\ 0 & 0 & 0.6840 & -0.5135 & 0.5182 \\ 0 & 0 & 0 & -0.7103 & -0.7039 \end{bmatrix}$$

$$R = \begin{bmatrix} -8.0623 & -3.4730 & -8.9305 & 0 & 0 \\ 0 & 12.3263 & -0.0824 & 2.2716 & 0 \\ 0 & 0 & 4.3863 & 13.7217 & 3.4198 \\ 0 & 0 & 0 & -7.0395 & -10.3807 \\ 0 & 0 & 0 & 0 & -5.1523 \end{bmatrix}$$

pri čemu smo zaokruživali na četiri decimale.

Primjetimo da se izračunavanje QR dekompozicije tridijagonalnih matrica provodi na isti način kao i kod Hessenbergovih matrica.

4 Literatura

[1] J. W. Demmel: Applied Numerical Linear Algebra, SIAM 1997.

[2] G.Golub, C.F.Van Loan, Matrix Computations, Johns Hopkins Univ Pr., 3rd edition, 1996.

[3] R. Scitovski: Numerička matematika, Sveučilište J. J. Strossmayera u Osijeku, Odjel za matematiku, Osijek, 2000.

[4] N. Truhar: Numerička linearna algebra, Sveučilište J. J. Strossmayera u Osijeku, Odjel za matematiku, Osijek, 2010.

[5] http://www.mathos.unios.hr/geometrija/Materijali/Geo_2.pdf (2012.)

[6] http://web.math.pmf.unizg.hr/∼drmac/NA1-2.pdf (22.12.2008.)

[7] http://web.math.pmf.unizg.hr/∼singer/num_mat/NM_0809/08.pdf (2009.)

5 Prilog - Matlab kodovi

1. Funkcija povratnih supstitucija

```
function [x] = povsup(U,b)
count = 0;
n = size(U,2);
x = zeros(n,1);
for j = n : -1 : 1
    suma = 0;
    for k = j+1 : n
        suma = suma + U (j,k) * x(k);
    end
    x(j) = (b(j) - suma) / U(j,j);
end
```

2. Funkcija za računanje elemenata Givensove matrice

```
function [c,s] = Givens(a,b)
r = 0;
if b == 0
    c = 1;
    s = 0;
   else if norm(b) > norm(a)
           r = - a / b;
           s = 1 / sqrt (1 + r * r);
           c = s * r;
              else r = - b / a;
                   c = 1 / sqrt ( 1 + r * r);
                   s = c * r;
        end
end
```

3. Funkcija za određivanje QR dekompozicije pomoću Givensovih rotacija

```
function [Q,A] = giv(A,b)
m = size(A,1);
n = size(A,2);
Q= eye(m);
for j = 1 : n
    for i = m : -1 : j + 1
        [c,s] = Givens ( A(i-1,j) , A(i,j) );
        A(i-1 : i , j : n) = [c s ; -s c] ' * A( i-1 : i, j : n);
        Q(1 : m , i-1 : i) = Q(1 : m , i-1 : i)*[c s;-s c];
    end
end
Q = Q( : , 1 : n)
A = A(1 : n , 1 : n)
```

4. Funkcija za rješavanje sustava linearnih jednadžbi pomoću Givensovih rotacija

```
function [x] = giv(A,b)
m = size(A,1);
n = size(A,2);
for j = 1 : n
    for i = m : -1 : j + 1
        [c,s] = Givens ( A(i-1,j) , A(i,j) );
        A(i-1 : i , j : n) = [c s ; -s c] ' * A( i-1 : i, j : n);
        b(i-1 : i) = [c s ; -s c] ' * b( i-1 : i);
    end
end
[x] = povsup(A,b)
```

5. Funkcija za QR dekompoziciju matrica u gornjoj Hessenbergovoj formi pomoću Givensovih rotacija

```
function [Q,A] = giv_hess(A)
n = size(A,2);
Q = eye(n);
for j = 1 : n-1
    [c,s] = Givens( A(j,j) , A(j+1,j) );
    A(j : j+1 , j : n) = [c s ; -s c] ' * A(j : j+1 , j : n);
    Q(1 : n , j : j+1) = Q(1 : n , j : j+1) * [c s ; -s c];
end
```