NMB - Oefenzitting 1:
QR-factorisatie en kleinste kwadratenproblemen

Hendrik Speleers

1 Pen en papier

Opgave 1. Hoe ziet de QR-factorisatie van een bandmatrix met bovenbandbreedte q en onderbandbreedte p eruit? Hoeveel bewerkingen vraagt het berekenen van een Householder respectievelijk een Givens QR-factorisatie? Je mag veronderstellen dat $p, q \ll n$.

Oplossing. *R is bovendriehoeks met bovenbandbreedte $p+q$. De 3 methodes vragen $\mathcal{O}(p(p+q)n)$ bewerkingen, maar voor Givens QR zijn $\mathcal{O}(pn)$ vierkantswortels nodig. Een iets gedetailleerder overzicht:*

	$+-$	$*/$	$\surd$
Householder	$2pn$	$3pn$	$\mathbf{2n}$
	$2p(p+q)n$	$\mathbf{2p(p+q)n}$	
Givens	pn	$4pn$	$\mathbf{pn}$
	$2p(p+q)n$	$\mathbf{4p(p+q)n}$	

De eerste rij toont telkens het aantal bewerkingen nodig voor het opbouwen van de transformaties; de tweede toont deze nodig voor het toepassen van de transformaties. De eerste rij is van een lagere orde, maar als $p, q \ll n$ dan kan het toch een rol gaan spelen.

Opgave 2. Gegeven de QR-factorisatie

$$QR = A = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}, \quad a_i \in \mathbb{R}^m$$

Hoe zou je te werk gaan om de QR-factorisaties van $\tilde{A}$ te berekenen in de volgende gevallen?

a) kolom k verwijderd

$$\tilde{A} = \begin{bmatrix} a_1 & \cdots & a_{k-1} & a_{k+1} & \cdots & a_n \end{bmatrix} \in \mathbb{R}^{m\times(n-1)}$$

Oplossing. *$Q^T\tilde{A} = H$ waarbij H kan bekomen worden door kolom k uit R te verwijderen. H is een boven-Hessenberg matrix met onderdiagonaalelementen vanaf de weggelaten kolom. Deze elementen kunnen nul gemaakt worden met Givenstransformaties : $G_{n-1}^T \cdots G_k^T H = R_1$. Stellen we $Q_1 = QG_k \cdots G_{n-1}$ dan hebben we de gevraagde QR-factorisatie $\tilde{A} = Q_1R_1$. Voorbeeld:*

$$H = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

b) een kolom $z \in \mathbb{R}^m$ toegevoegd

$$\tilde{A} = \left[\begin{array}{ccccccc} a_1 & \cdots & a_k & z & a_{k+1} & \cdots & a_n \end{array}\right] \in \mathbb{R}^{m\times(n+1)}$$

Oplossing. *$Q^T\tilde{A} = H$ waarbij H kan bekomen worden door een kolom $w = Q^T z$ aan R toe te voegen. De elementen onder de diagonaal kunnen weer met Givenstransformaties op nul gezet worden. Als je dit van onder naar boven doet, dan worden enkel op de diagonaal niet-nul elementen geïntroduceerd. De transformaties worden dus bepaald zodat $G_{k+1}^T \cdots G_{m-1}^T w = \begin{bmatrix} \tilde{w} \\ 0 \end{bmatrix}$, $\tilde{w} \in \mathbb{R}^{k+1}$. Met $Q_1 = QG_{m-1}\cdots G_{k+1}$ en $R_1 = G_{k+1}^T \cdots G_{m-1}^T H$ vinden we de gevraagde QR-factorisatie $\tilde{A} = Q_1R_1$.*
Voorbeeld:

$$H = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & 0 & \times \\ 0 & 0 & 0 & \times & 0 & 0 \\ 0 & 0 & 0 & \times & 0 & 0 \end{bmatrix}$$

c) een rij $w^T \in \mathbb{R}^n$ toegevoegd

$$\tilde{A} = \begin{bmatrix} w^T \\ A \end{bmatrix} \in \mathbb{R}^{(m+1)\times n}$$

Oplossing. *Stel $Q_d = \begin{bmatrix} 1 & \\ & Q \end{bmatrix}$, dan geldt dat $Q_d\tilde{A} = \begin{bmatrix} w^T \\ R \end{bmatrix} = H$ is boven-Hessenberg. We kunnen dus Givenstransformaties bepalen zodat $G_n^T \cdots G_1^T H = R_1$ bovendriehoeks is. De gewenste QR-factorisatie is dan $\tilde{A} = Q_1R_1$, met $Q_1 = Q_dG_1\cdots G_n$.*
Voorbeeld:

$$H = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

2 Op de computer

2.1 Householdertransformatie

Opgave 3. Maak de volgende matrix

```
X = rand(5,2); X(2:5,1) = X(2:5,1)*1.e-8
```

Pas de Householderspiegelingen

$$H_k = I - 2u_k u_k^T/(u_k^T u_k) \qquad k = 1, 2$$

toe met

$$\begin{aligned} u_1 &= X(:,1) + \|X(:,1)\|_2 e_1 \\ u_2 &= X(:,1) - \|X(:,1)\|_2 e_1 \end{aligned}$$

Bereken H_1X en H_2X. Wat zie je? En waarom?

Oplossing. *Tip: het commando* `format short e` *kan handig zijn als je matrices van getallen met sterk verschillende grootte-orde moet bestuderen.*
Met

$$u = X(:,1) + \mathrm{sign}(X(1,1))\|X(:,1)\|_2 e_1$$

hebben we het minst last van afrondingsfouten. In dit geval is u_1 dus de juiste keuze. We zien inderdaad dat voor $Y_1 = H_1X$ de elementen $Y_1(2:5,1)$ van de grootte-orde 10^{-25} zijn, ruim beneden de machineprecisie. Voor $Y_2 = H_2X$ zijn de elementen $Y_2(2:5,1)$ echter van de orde 10^{-8}, een veelvoud van de machineprecisie. Op deze elementen zit dus een niet verwaarloosbare fout.
Dit komt omdat u_2 het verschil is van twee vectoren van dezelfde grootte-orde zodat er een verlies aan beduidende cijfers optreedt. In plaats van te kiezen voor u_1 kan je u_2 ook als volgt berekenen

```
uu2(1) = -sum(X(2:5,1).^2)/(X(1,1)+norm(X(:,1)))
uu2(2:5) = X(2:5,1)
```

De vector `(u2-uu2)./uu2` *(meer bepaald het eerste element) bevestigt dat er een beduidend verlies aan nauwkeurigheid is bij de berekening van u_2.*

2.2 Kleinste kwadratenproblemen: normaalvergelijkingen versus QR-factorisatie

Opgave 4. Construeer een voorbeeld waarvoor de normaalvergelijkingen falen en Householder niet. Kies een $A \in \mathbb{R}^{3\times 2}$ bovendriehoeks. Werk zoveel mogelijk op papier, dan maak je geen afrondingsfouten.

Oplossing. *Bijvoorbeeld*

$$A = \begin{bmatrix} 1 & 1 \\ 0 & \varepsilon \\ 0 & 0 \end{bmatrix}$$

met $\varepsilon = 10^{-12}$.

2.3 Snelle Givenstransformatie

Met snelle Givenstransformaties kan je een (M, D)-representatie van Q berekenen. Meer bepaald geldt voor een matrix $A \in \mathbb{R}^{m\times n}$ dat $M^T A = T$ met T een bovendriehoeksmatrix en $M^T M = D$ diagonaal. Bovendien geldt dat $Q = MD^{-1/2}$ een orthogonale matrix is en

$$Q^T A = D^{-1/2} T \equiv R.$$

Gegeven $x \in \mathbb{R}^2$ en positieve $d \in \mathbb{R}^2$. De routine `fastgivens1.m` berekent een 2×2 snelle Givenstransformatie M zodat de tweede component van $M^T x$ nul is en $M^T D M = D_1$ diagonaal is met $D = \mathrm{diag}(d_1, d_2)$. In de implementatie wordt enkel gebruik gemaakt van de vorm

$$\begin{bmatrix} \beta & 1 \\ 1 & \alpha \end{bmatrix}.$$

De routine `fastgivensQR1.m` maakt gebruik van routine `fastgivens1.m` om een matrix $A \in \mathbb{R}^{m \times n}$ te transformeren naar een bovendriehoeksmatrix T en geeft de bijbehorende matrix D terug.

Opgave 5.

a) Ga de groei na van een element van D (bijvoorbeeld het laatste) voor een willekeurige matrix. Gebruik de grafische mogelijkheden van MATLAB om een kwalitatief idee te krijgen. Wat verwacht je? Komt dit overeen met je experimenten?

> **Oplossing.** *We verwachten dat de elementen groot zullen worden. Zie* `fastgivensQR1d.m`.

b) Wat is een manier om de groei te beperken? Pas de MATLAB-routines aan zodat ze deze oplossing gebruiken. (Test de correctheid van je implementatie!)

> **Oplossing.** *Een gepaste keuze van type 1 of type 2 snelle Givensrotatie zal de groei beperken. Zie* `fastgivens.m`, `fastgivensQR.m` *en* `fastgivensQRd.m`.

c) Vergelijk de groei van een element van D met deze oplossing ten opzichte van de orginele. Verbetering?

> **Oplossing.** *Er is een verbetering.*

d) Kunnen er zich met deze nieuwe implementatie nog problemen voordoen? Wat zou je kunnen doen om ze te verhelpen?

> **Oplossing.** *Er kan zelfs met de voorgaande verbetering nog een aanzienlijke groei optreden. Het is nodig om de matrices M, D en T te observeren en een herschaling toe te passen als overflow dreigt op te treden. Verder zal er, net als bij QR-factorisatie, een probleem optreden bij matrices die niet van volle rang zijn. Dit kan opgelost worden via pivotering.*

e) Genereer een willekeurige matrix `A(1:m,1:n)` en een willekeurige kolomvector `b(1:m)` en gebruik de snelle Givensfactorisatie om een kleinste kwadratenoplossing te vinden voor het stelsel `A*x = b`.

> **Oplossing.**
>
> ```
> A = rand(m,n);
> b = rand(m,1);
> [d,T] = fastgivensQR([A,b]);
> x = T(1:n,1:n)\T(1:n,n+1);
> norm(A*x - b) - norm(A*(A\b) - b)
> ```