PROF. DR. BERND SIMEON
CHRISTIAN GOBERT
THOMAS MÄRZ

TECHNISCHE UNIVERSITÄT MÜNCHEN

Numerik

SS 2009

Musterlösung zu Hausaufgabenblatt 1

Aufgabe 1 *Links- und Rechtseigenpaare* (20 P.)
Definition:
Sei $A \in \mathbb{C}^{n \times n}$.
Ein Vektor $0 \neq x \in \mathbb{C}^n$ heißt (Rechts-)eigenvektor von A, falls es ein $\lambda \in \mathbb{C}$ gibt mit $Ax = \lambda x$. In diesem Fall heißt (λ, x) (Rechts-)eigenpaar.
Ein Vektor $0 \neq y \in \mathbb{C}^n$ heißt Linkseigenvektor von A, falls es ein $\mu \in \mathbb{C}$ gibt mit $y^* A = \mu y^*$, dabei bedeutet $y^* = y^H = \overline{y}^T$. In diesem Fall heißt (μ, y) Linkseigenpaar.

(a) Zu jedem Rechtseigenpaar (λ, x) gibt es ein Linkseigenpaar (λ, y) und umgekehrt. Warum?

$\langle \cdot, \cdot \rangle$ bezeichnet im Folgenden das euklid. Skalarprodukt.

(b) Sei (λ, y) ein Linkseigenpaar und (μ, x) ein Rechtseigenpaar von A. Zeigen Sie: $x \perp y$, also $\langle x, y \rangle = 0$, falls $\lambda \neq \mu$.

(c) Sei λ ein einfacher Eigenwert von A. (λ, x) ein Rechts- und (λ, y) ein Linkseigenpaar. Zeigen Sie: $\langle x, y \rangle \neq 0$.

(d) Ist A obere Dreiecksmatrix und normal, so ist A eine Diagonalmatrix.

(e) Sei (λ, x) ein Rechtseigenpaar von A. Ist A normal, dann ist (λ, x) auch ein Linkseigenpaar.

Lösung
(a, 2P.) Rechtseigenvektoren sind Lösung der Gleichung

$$(A - \lambda I)x = 0,$$

also aus dem Kern von $A - \lambda I$. Linkseigenvektoren sind Lösung der Gleichung

$$y^*(A - \mu I) = (A - \mu I)^* y = 0,$$

also aus dem Kern von $(A - \mu I)^*$. $\boxed{\text{1P.}}$
Da der Zeilenrang einer Matrix gleich ihr Spaltenrang ist $\boxed{\text{1P.}}$, gilt die zu zeigende Aussage.
(b, 4P.)

$$\begin{aligned}(\mu - \lambda)\langle y, x \rangle &= \langle \overline{\mu} y, x \rangle - \langle y, \lambda x \rangle = \\ &= \langle A^* y, x \rangle - \langle y, Ax \rangle = \langle A^* y, x \rangle - \langle A^* y, x \rangle = 0\end{aligned}$$

Wegen $\lambda - \mu \neq 0$ muss also $\langle x, y \rangle = 0$ gelten $\boxed{\text{4P.}}$.

(c, 5P.) Wir betrachten die Jordan-Normalform $\boxed{\text{1P.}}$ von A (geht auch mit Schurzerlegung). Sei also $S \in \mathbb{C}^{n\times n}$ invertierbar mit

$$S^{-1}AS = T = \left(\begin{array}{c|c} \lambda & \\ \hline & \hat{T} \end{array}\right).$$

Da λ einfach ist, gibt es nur einen 1×1 Jordan-Block für den Eigenwert λ und diesen können wir o. E. links oben platzieren. Nun gilt

$$\begin{aligned} Ax = STS^{-1}x &= \lambda x \quad &\Rightarrow \quad T(S^{-1}x) = \lambda \underbrace{(S^{-1}x)}_{=:\tilde{x}} \boxed{\text{1P.}} \\ y^*A = y^*STS^{-1} &= \lambda y^* \quad &\Rightarrow \quad (S^*y)^*T = \lambda \underbrace{(S^*y)^*}_{=:\tilde{y}}. \boxed{\text{1P.}} \end{aligned}$$

Wegen

$$\langle y, x\rangle = y^*x = y^*SS^{-1}x = (S^*y)^*S^{-1}x = \tilde{y}^*\tilde{x} = \langle \tilde{y}, \tilde{x}\rangle$$

reicht es, zu zeigen, dass $\langle \tilde{y}, \tilde{x}\rangle \neq 0$ $\boxed{\text{1P.}}$.
(λ, e_1) ist Rechtseigenpaar von T und da λ einfach ist, ist $\tilde{x} = \alpha e_1$, dabei können wir o. E. $\alpha = 1$ annehmen. Wäre $\langle \tilde{y}, \tilde{x}\rangle = \langle \tilde{y}, e_1\rangle = 0$, so hätte $\tilde{y}$ die Form $\tilde{y} = (0, \hat{y})^T$ mit $\hat{y} \in \mathbb{C}^{n-1}$. So würde aus

$$\tilde{y}^*T = \lambda\tilde{y}^*$$

folgen, dass $\hat{y}^*\hat{T} = \lambda\hat{y}^*$ gelten würde und damit $(\lambda, \hat{y})$ ein Linkseigenpaar von $\hat{T}$ wäre. Dies ist ein Widerspruch, denn $\hat{T}$ hat λ nicht als Eigenwert $\boxed{\text{1P.}}$.

(d, 4P.) Wir partitionieren T $\boxed{\text{1P.}}$ wie folgt

$$T = \left(\begin{array}{c|c} \alpha & v^* \\ \hline & \hat{T} \end{array}\right), \qquad T^* = \left(\begin{array}{c|c} \overline{\alpha} & \\ \hline v & \hat{T}^* \end{array}\right)$$

Wegen $TT^* = T^*T$ $\boxed{\text{1P. für Definition 'normal'}}$ muss gelten $e_1^*TT^*e_1 = |\alpha|^2 + \|v\|^2 = e_1^*T^*Te_1 = |\alpha|^2$, also $v = 0$ $\boxed{\text{1P.}}$. Rekursive anwendung liefet T diagonal $\boxed{\text{1P.}}$.

(e, 5P.) Normale Matrizen sind unitär diagonalisierbar $\boxed{\text{1P.}}$, d. h. es gibt zu A eine unitäre Matrix U, so dass

$$U^*AU = D \qquad \boxed{\text{1P.}}$$

mit D diagonal. Damit ergibt sich einmal $AU = UD$ $\boxed{\text{1P.}}$ und alle Spalten von U sind Rechtseigenvektoren von A $\boxed{\text{1P.}}$. Genauer: Die Spalten von U bilden eine Rechtseigenbasis von C^n. Außerdem ergibt sich $U^*A = DU^*$, womit die Spalten von U ebenso eine Linkseigenbasis von C^n sind $\boxed{\text{1P.}}$. Kurz: die Links- und Rechtseigenräume für einen Eigenwert λ stimmen überein, da Rechts- und Linkseigenvetkoren zu verschiedenen Eigenwerten senkrecht stehen.

Aufgabe 2 *Givens-Rotationen* (20 P.)
Givens-Rotationen sind orthogonale Transformationen, die sukzessive die QR-Zerlegung einer Matrix $A \in \mathbb{R}^{nxn}$ berechnen. Givens-Rotationen werden als Reduktionsalgorithmen zur Lösung von Eigenwertproblemen herangezogen. Die Idee ist ähnlich der Reduktion durch Householder-Reflexionen (vgl. Skript). In dieser Aufgabe wird die Technik der Givens-Rotationen erarbeitet.

a) Es sei $A \in \mathbb{R}^{2\times 2}$. Bestimmen Sie eine orthogonale Matrix $R \in \mathbb{R}^{2\times 2}$ so, dass RA obere Dreiecksmatrix ist.

b) Es sei $A \in \mathbb{R}^{n\times n}$ der Form

$$A = \begin{pmatrix} a_{11} & * & * & * & * \\ 0 & * & * & * & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & * & * & * & * \\ a_{k,1} & * & * & * & * \\ 0 & * & * & * & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & * & * & * & * \end{pmatrix}$$

Bestimmen Sie eine orthogonale Matrix $R \in \mathbb{R}^{n\times n}$ so, dass RA die folgende Gestalt hat:

$$RA = \begin{pmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & * & * & * & * \end{pmatrix}$$

c) Es sei $A \in \mathbb{R}^{n\times n}$ der Form

$$A = \begin{pmatrix} a_{11} & * & * & * & * \\ 0 & \ddots & * & * & * \\ \vdots & \ddots & a_{k,k} & \vdots & \vdots \\ & & 0 & * & * \\ & & \vdots & \vdots & \vdots \\ & & 0 & * & * \\ & & a_{k+l,k} & * & * \\ & & 0 & * & * \\ \vdots & & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & * & * \end{pmatrix}$$

Bestimmen Sie eine orthogonale Matrix $R \in \mathbb{R}^{n\times n}$ so, dass RA die folgende Gestalt hat:

$$RA = \begin{pmatrix} * & * & * & * & * \\ 0 & \ddots & * & * & * \\ \vdots & \ddots & * & \vdots & \vdots \\ & & 0 & * & * \\ & & \vdots & \vdots & \vdots \\ & & 0 & * & * \\ & & 0 & * & * \\ & & 0 & * & * \\ \vdots & & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & * & * \end{pmatrix}$$

d) Die Transformationen aus den Aufgaben a) bis c) werden Givens-Rotationen genannt. Formulieren Sie einen Algorithmus, der für eine Matrix $A \in \mathbb{R}^{n\times n}$ durch Givens-Rotationen sukzessive die QR-Zerlegung berechnet.

e) Wieso heißen diese Transformationen Rotationen?

Lösung

a, 4P.)

$$(RA)_{2,1} = 0$$

O.E. sei $a_{22} \neq 0$. Der Ansatz $\boxed{\text{1P.}}$

$$R = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix}$$

liefert

$$\begin{aligned} -a_{11}\sin\phi + a_{21}\cos\phi &= 0 && (1) \\ \frac{\cos\phi}{\sin\phi} &= \frac{a_{11}}{a_{21}} \quad \boxed{\text{1P.}} && (2) \end{aligned}$$

Mit $\cos^2\phi = 1 - \sin^2\phi$ erhält man z.B. (auch andere Wahl der Vorzeichen ist möglich)

$$R = \begin{pmatrix} \gamma & \sigma \\ -\sigma & \gamma \end{pmatrix}$$

mit

$$\gamma = \frac{a_{11}}{\sqrt{a_{11}^2 + a_{21}^2}} \boxed{\text{1P.}} \qquad \sigma = \frac{a_{21}}{\sqrt{a_{11}^2 + a_{21}^2}} \boxed{\text{1P.}}$$

b, 3P.) Mit dem Ergebnis aus a) lässt sich die Lösung direkt aufschreiben:

$$R = \begin{pmatrix} \gamma & & & & \sigma & & & \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & 1 & & & & \\ -\sigma & & & & \gamma & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{pmatrix} \begin{array}{l} \\ \\ \\ \\ \leftarrow \text{Zeile } k \\ \\ \\ \\ \end{array} \qquad \boxed{\text{1P.}}$$

mit

$$\gamma = \frac{a_{1,1}}{\sqrt{a_{1,1}^2 + a_{k,1}^2}} \ \boxed{\text{1P.}} \qquad \sigma = \frac{a_{k,1}}{\sqrt{a_{1,1}^2 + a_{k,1}^2}} \ \boxed{\text{1P.}}$$

c, 3P.) Wie oben:

$$R = \begin{array}{c} \begin{pmatrix} 1 & & & & & & & & & & \\ & \ddots & & & & & & & & & \\ & & 1 & & & & & & & & \\ & & & \gamma & & & & \sigma & & & \\ & & & & 1 & & & & & & \\ & & & & & \ddots & & & & & \\ & & & & & & 1 & & & & \\ & & & -\sigma & & & & \gamma & & & \\ & & & & & & & & 1 & & \\ & & & & & & & & & \ddots & \\ & & & & & & & & & & 1 \end{pmatrix} \end{array} \begin{array}{l} \\ \\ \\ \leftarrow \text{ Zeile } k \\ \\ \\ \\ \leftarrow \text{ Zeile } k+l \\ \\ \\ \\ \end{array} \qquad \boxed{\text{1P.}}$$

mit

$$\gamma = \frac{a_{k,k}}{\sqrt{a_{k,k}^2 + a_{k+l,k}^2}} \ \boxed{\text{1P.}} \qquad \sigma = \frac{a_{k+l,k}}{\sqrt{a_{k,k}^2 + a_{k+l,k}^2}} \ \boxed{\text{1P.}}$$

d, 8P.) In jeder Spalte wird eine Matrix R (siehe oben) sukzessive angewendet, so dass in dieser Spalte alle Elemente unterhalb der Diagonalen eliminiert werden. In jedem Schritt ändern sich dabei nur die Zeilen k und $k+l$. Für die Spalte k liefert das der folgende Algorithmus: $\boxed{\text{2P.}}$ + $\boxed{\text{1P. für Effizienz (keine volle Matrix-Matrix Multiplikation)}^{1}}$

```
R=A;
n=size(A,1);
for l=1:(n-k)
gamma=R(k,k)/sqrt(R(k,k)^2 + R(k+l,k)^2);
sigma=R(k+l,k)/sqrt(R(k,k)^2 + R(k+l,k)^2);
R([k, k+l],:)=[gamma;-sigma]*R(k,:)+[sigma;gamma]*R(k+l,:);
end
```

Angewendet Spalte für Spalte:

```
R=A;
n=size(A,1);
for k=1:n
for l=1:(n-k)
gamma=R(k,k)/sqrt(R(k,k)^2 + R(k+l,k)^2);
sigma=R(k+l,k)/sqrt(R(k,k)^2 + R(k+l,k)^2);
R([k, k+l],:)=[gamma;-sigma]*R(k,:)+[sigma;gamma]*R(k+l,:);
end
end
```

So wird die Matrix R der QR-Zerlegung bestimmt. Es muss noch (gleichzeitig) die Matrix Q berechnet werden. Es gilt

$$...R_3 R_2 R_1 A = R$$

dabei sind die R_i Rotationsmatrizen von oben, also orthogonal. Deshalb ist

$$A = \underbrace{I R_1^T R_2^T R_3^T ...}_{=Q} R \qquad \boxed{\text{2P.}}$$

Die Rotationsmatrizen müssen also (transpoiert) von links auf die Einheitsmatrix multipliziert werden, um Q zu erhalten. Das bedeutet, dass sich jetzt in jedem Schritt nur die Spalten k und $k+l$ ändern. Man erhält den folgenden Algorithmus: 2P. + 1P. für Effizienz[1]

```
R=A;
n=size(A,1);
Q=eye(n,n);
for k=1:n
for l=1:(n-k)
gamma=R(k,k)/sqrt(R(k,k)2̂ + R(k+l,k)2̂);
sigma=R(k+l,k)/sqrt(R(k,k)2̂ + R(k+l,k)2̂);
R([k, k+l],:)=[gamma;-sigma]*R(k,:)+[sigma;gamma]*R(k+l,:);
Q(:,[k, k+l])=Q(:,k)*[gamma -sigma]+Q(:,k+l)*[sigma gamma];
end
end
```

e, 2P.) Die Matrizen R sind Rotationsmatrizen, die Givens-Transformation ist eine Serie von Rotationen. 2P.

[1] In der Angabe wurde nicht nach einem effizienten Algorithmus gefragt, dieser Punkt ist also ein Zusatzpunkt