Technische Universität Chemnitz
Fakultät für Mathematik
Professur Mathematik in Industrie und Technik

Wintersemester 2007/2008
Dipl. Math. Jens Saak
Dipl. Math. Ulrike Baur

Numerische Lineare Algebra (Eigenwertprobleme) – 1. Hausaufgabe

Abgabetermin: 14.12.2007
(in der Vorlesung)

Theoretische Aufgaben

Aufgabe 1 (6 Punkte) (Normale Matrizen)

Zeigen Sie die Äquivalenz der folgenden Bedingungen für eine Matrix $A \in \mathbb{C}^{n \times n}$.

i) A is normal, d.h. $AA^* = A^*A$.

ii) U^*AU ist normal für jede unitäre Matrix $U \in \mathbb{C}^{n \times n}$.

iii) Für jede unitäre Matrix U mit $U^*AU = B = \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix}$, $B_{11} \in \mathbb{C}^{k \times k}$, gilt $B_{12} = 0$.

iv) Es gibt eine unitäre Matrix $U \in \mathbb{C}^{n \times n}$, so dass U^*AU diagonal ist.

v) Es gibt eine unitäre Matrix $U \in \mathbb{C}^{n \times n}$ mit $A^* = UA$.

vi) $\langle Ax, Ay \rangle = \langle A^*x, A^*y \rangle$ für alle $x, y \in \mathbb{C}^n$.

Zusatz: Zeigen Sie außerdem, dass i)-vi) zu folgender Bedingung äquivalent sind:

vii) $AA^* - A^*A$ ist positiv semidefinit (d.h. $x^*(AA^* - A^*A)x \geq 0$ für alle $x \in \mathbb{C}^n$).

Aufgabe 2 (4 Punkte) (Kronecker-Produkte)

Seien $Y \in \mathbb{C}^{j \times k}$ und $Z \in \mathbb{C}^{m \times n}$. Dann heißt die $mj \times kn$ Matrix

$$Y \otimes Z = \begin{bmatrix} y_{11}Z & y_{12}Z & \dots & y_{1k}Z \\ y_{21}Z & y_{22}Z & \dots & y_{2k}Z \\ \vdots & \vdots & \ddots & \vdots \\ y_{j1}Z & y_{j2}Z & \dots & y_{jk}Z \end{bmatrix}$$

das *Kronecker-Produkt* oder *Tensor-Produkt* von Y und Z.

a) Seien W, X, Y, Z Matrizen, so dass die Produkte WX und YZ definiert sind. Zeige $(W \otimes Y)(X \otimes Z) = (WX) \otimes (YZ)$.

b) Seien S, G nichtsinguläre Matrizen. Zeige, dass auch $S \otimes G$ nichtsingulär ist und dass $(S \otimes G)^{-1} = S^{-1} \otimes G^{-1}$.

c) Zeige, wenn A und B, sowie C und D ähnliche Matrizen sind, dann sind auch $A \otimes C$ und $B \otimes D$ ähnlich.

d) Seien $A \in \mathbb{C}^k$ und $B \in \mathbb{C}^m$. A habe die Eigenwerte $\lambda_1, \dots, \lambda_k$ und B habe die Eigenwerte $\mu_1, \dots, \mu_m$. Zeige

$$\Lambda(A \otimes B) = \{\lambda_i \mu_j \,|\, i = 1, \dots, k;\ j = 1, \dots, m\}.$$

Aufgabe 3 (5 Punkte) (Hessenberg-Matrizen und Givens-Rotationen)

Unter einer oberen Hessenberg-Matrix versteht man eine $n \times n$ Matrix $A = (a_{ij})_{i,j=1,\dots,n}$ für die gilt $a_{ij} = 0$ falls $i > j + 1$. Sei $H \in \mathbb{R}^{n \times n}$ obere Hessenberg Matrix und $b \in \mathbb{R}^n$. Geben Sie einen Lösungsalgorithmus zum Lösen eines Gleichungssystems

$$Hx = b$$

mittels Givens-Rotationen an. Berechnen Sie mit diesem Algorithmus die Lösung des Gleichungssystems zu den Daten

$$H = \begin{bmatrix} 3 & 0 & 19 \\ 4 & 5 & 17 \\ 0 & 4 & 35 \end{bmatrix} \qquad b = \begin{bmatrix} 22 \\ 26 \\ 39 \end{bmatrix}$$

Programmieraufgabe

Aufgabe 1 (5 Punkte) (Bilddatenkompression mit SVD)

Schreibe ein MATLAB-Programm, welches eine Matrix mit Bilddateninformation einliest, eine beste Approximation vom Rang k an diese Matrix berechnet, sowie den Approximationsfehler (in der 2-Norm), das Original und das komprimierte Bild darstellt (benutze `subplot`), sowie den für das Original und das komprimierte Bild benötigten Arbeitsspeicher ausgibt. Teste das Programm für die in MATLAB verfügbaren Bilder (`clown, gatlin, durer, mandrill, earth`) und verschiedene Werte von k. Ermittle (empirisch) für jedes Bild die beste Kompression, so dass das approximierte Bild noch optisch verlustfrei dargestellt werden kann.

Hinweis: Die MATLAB-Bilder erhält man mit dem `load`-Befehl. Z.B. erhält man mit

```
>> load clown
```

eine Bilddatenmatrix `X` (eine 200×320-Matrix in diesem Fall, also hat das Clown-Bild 200×320 Bildpunkte) sowie eine Matrix `map`, in der die Information über das verwendete Farbmodell gespeichert ist. Um nun das Clown-Bild anzuzeigen, kann man die Befehle

```
>> colormap(map)
>> image(X)
```

ausführen.