Technische Universität Hamburg-Harburg
Institut für Numerische Simulation, E-10
Dr. Jens-Peter M. Zemke

Sommersemester 2008

Numerische Verfahren

Übungen und Lösungen, Blatt 6

Aufgabe 1: (Thema: Singulärwertzerlegung.)

Berechnen Sie eine Singulärwertzerlegung der Matrix

$$A = \begin{pmatrix} 1 & 2 & 2 \\ 2 & -2 & 1 \\ 2 & 1 & -2 \end{pmatrix}.$$

Lösung zu Aufgabe 1:

Da es sich bei der Matrix A um das Vielfache einer orthogonalen Matrix handelt, alle Zeilen und Spalten sind paarweise orthogonal und die Norm ist jeweils 3, lässt sich A schreiben als

$$A = \tfrac{1}{3}A \cdot 3E \cdot E =: U\Sigma V^T$$

wobei E die 3×3-Einheitsmatrix bezeichnet. Eine weitere Singulärwertzerlegung wäre z.B.

$$A = E \cdot 3E \cdot \tfrac{1}{3}A.$$

Aufgabe 2: (Thema: QR-Algorithmus.)

Konvergiert die Grundform des QR-Algorithmus für die Matrix A aus Aufgabe 1 gegen eine obere Dreiecksmatrix?

Lösung zu Aufgabe 2:

Nein. Die Matrix A ist eine um 3 skalierte orthogonale Matrix, wie bereits in Aufgabe 1 bemerkt. Damit ist die (eindeutige) QR-Zerlegung von A gegeben durch

$$A = \frac{1}{3}A \cdot 3E.$$

Folglich ergibt sich für A_1

$$A_1 = 3E \cdot \frac{1}{3}A = A$$

und per Induktion folgt $A_i = A$, $\forall i$. Damit ist dann aber auch der Limes gleich A und A ist *keine* obere Dreiecksmatrix. Allgemein gilt das für beliebige orthogonale (unitäre) Matrizen und deren Vielfache, sofern sie nicht (wie zum Beispiel die Identität) bereits obere Dreiecksmatrizen sind.

Aufgabe 3: (Thema: Inverse Iteration.)

Erzeugen Sie mit

```
I = speye(n);
E = sparse(2:n,1:n-1,1,n,n);
D = E + E' - 2*I;
A = kron(D,I) + kron(I,D);
A = -n^2*A;
```

eine $n^2 \times n^2$ -Matrix A.

Bestimmen Sie für $n = 100$ den kleinsten Eigenwert von A durch die folgenden Algorithmen (Bezeichnungen wie in Algorithmus 6.15):

a) Potenzmethode für A^{-1}

b) Inverse Iteration mit festen Shift

c) Inverse Iteration mit variablem Shift

$$\lambda_{m+1} = \lambda_m - 1/k_m$$

d) Inverse Iteration mit Rayleigh-Quotienten Shift:

$$\lambda_{m+1} = u_{m+1}^T A u_{m+1} / ||u_{m+1}||^2$$

Vergleichen Sie die Rechenzeiten.

Lösung zu Aufgabe 3:

Die inverse Iteration ist in dem beigefügten M–File `aufg06f01.m` implementiert. Die Art des Verfahrens wird über die Variable `var` gesteuert. Mögliche Werte für den Parameter sind:

`var` $= 0$: Es werden feste Shifts wie in Algorithmus 6.14 benutzt.

`var` $= 1$: Es werden variable Shifts wie in Algorithmus 6.15 benutzt.

`var` $= 2$: Es werden Rayleigh-Quotienten Shifts benutzt.

Die Potenzmethode für A^{-1} ist äquivalent zur inversen Iteration mit festen Shift 0. Die verschiedenen Aufrufe, wie auch die Generierung der Matrix A, geschehen im M–File `aufg06f02.m`. Nach Aufruf ohne Parameter werden der Reihe nach für die Toleranz 10^{-10} die einzelnen Verfahren angewendet und Daten, wie Anzahl der Schritte, Größe des Residuums, sowie die erzielte Eigenwertnäherung, ausgegeben.

a) Die Potenzmethode benötigt zum Berechnen des kleinsten Eigenwertes 18 Schritte, damit das Residuum $\|Ax - \lambda x\|/\|x\|$ kleiner als 1E-10 ist.

b) Die inverse Iteration mit einem festen Shift benötigt je nach Shift natürlich eine verschiedene Zahl an Schritten. Für den Shift 0 ist die inverse Iteration äquivalent zur Potenzmethode für A^{-1}. Für den Shift 18 werden noch 7 Schritte benötigt bis die Genauigkeit von 1E-10 erreicht ist. Wird der Shift auf 19,3 gelegt, was schon fast dem gesuchten Eigenwert entspricht, sind nur noch 4 Schritte notwendig. Daran lässt sich gut die theoretische Vorhersage bestätigen, dass die Konvergenz um so schneller ist, desto dichter der Shift an dem tatsächlichen Eigenwert liegt.

c) Mit variablen Shifts sieht das Bild anders aus. Bei einem Start-Shift von 0 wird die Toleranz 1E-10 nach 8 Schritten erreicht — eine deutliche Verbesserung gegenüber 18 Schritten bei der Potenzmethode für A^{-1}. Allerdings konvergiert das Verfahren gegen einen anderen als den kleinsten Eigenwert. Die Anzahl der Schritte verringert sich häufig, je näher der Start-Shift an einen tatsächlichen Eigenwert gelegt wird. Es ergibt sich leichter Mehraufwand (es müssen verschiedene Gleichungssysteme gelöst werden), der aber noch zu verschmerzen ist.

d) Rayleigh-Quotienten Shifts führen zur schnellsten Konvergenz. Tatsächlich lässt sich zeigen, dass die inverse Iteration mit Rayleigh-Quotienten Shift für symmetrische Matrizen *lokal kubisch* konvergiert. Bei einem Start-Shift von 0 braucht die inverse Iteration mit Rayleigh-Quotienten-Shifts auch nur noch 4 Schritte, um den kleinsten Eigenwert mit einer Genauigkeit des Residuums von 1E-10 zu erhalten. Hier kommmt noch der Mehraufwand pro Schritt hinzu, der in der Berechnung des Rayleigh-Quotienten liegt.

Aufgabe 4: (Thema: QR-Algorithmus.)

Führen Sie für die Matrix

```
A = rand(n);
A = A + A';
```

für $n = 10$ den QR-Algorithmus mit dem Shift $x = a_{nn}$ aus. Verkleinen Sie die Dimension der Matrix um 1, falls

$$\max_{i=1,\ldots,n-1} |a_{ni}| < 10^{-16}|a_{nn}|, \qquad i = 1, \ldots, n-1$$

gilt. Wie viele Schritte benötigt man im Mittel, um diese Genauigkeit zu erreichen? Können Sie die Wahl der Deflationsbedingung begründen?

Lösung zu Aufgabe 4:

Die Abbruchbedingung ergibt sich daraus, daß die Elemente der letzten Zeile beim QR-Algorithmus mit Ausnahme von a_{nn} gegen 0 konvergieren. Deflation ist dann sinnvoll, wenn die Einträge $a_{ni}, i = 1, \ldots, n-1$ im Verhältnis zu a_{nn} vernachlässigbar klein sind. Dies sagt gerade die Deflationsbedingung aus. Eine Implementierung des QR-Algorithmus findet sich im M–File `aufg06f04.m`. Nach jedem Schritt wird die Deflationsbedingung (implementiert im M–File `aufg06f03.m`) getestet und eine Deflation durchgeführt, wenn sie erfüllt ist. Ein Test mit 2.000 und 10.000 Durchläufen (erfolgt über das M–File `aufg06f05.m`) ergab im Mittel folgende Anzahl von Iterationen, bis die Deflationsbedingung bei der $k \times k$-Matrix erfüllt war:

2	3	4	5	6	7	8	9	10
0.5720	2.2910	2.5385	2.5895	2.7645	2.8870	2.9975	3.1735	5.0185
0.5722	2.2981	2.4886	2.6207	2.7662	2.8694	3.0074	3.1370	5.0284

Dabei braucht der QR-Algorithmus im Mittel immer weniger Schritte, je kleiner die Dimension wird. Dieses wird dadurch verständlich, dass die QR-Schritte für die Matrix der Dimension k auch schon eine Verbesserung für alle Hauptunterabschnittsmatrizen der Dimension j mit $j < k$ bringen.