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

Sommersemester 2009
Prof. Dr. Peter Benner
Dipl.-Math. Jens Saak
Dipl.-Ing. Martin Bernauer

Numerische Mathematik– 6. Hausaufgabe

Abgabetermin: 20./21.5.2009
(in der jeweiligen Übungsgruppe)

Theoretische Aufgaben

Aufgabe 1 (3 Punkte) (Cholesky-Zerlegung)

Zur zeilenweisen Berechnung des Cholesky-Faktors $R^\top$ der Zerlegung $A = R^\top R$ einer symmetrischen und positiv-definiten $n \times n$-Matrix A kann man die folgende Vorschrift (siehe Übung 5) benutzen:

$r_{11} = \sqrt{a_{11}}$

FOR $i = 2, \ldots, n$

$$r_{ij} = \Big(a_{ij} - \sum_{k=1}^{j-1} r_{ik} r_{jk}\Big)/r_{jj}, \qquad j = 1, \ldots, i-1$$

$$r_{ii} = \Big(a_{ii} - \sum_{k=1}^{i-1} r_{ik}^2\Big)^{1/2}$$

END

a) Zeigen Sie (z.B. mit Induktion):

$$\sum_{i=1}^{n} i^2 = \frac{1}{6}\, n\,(1+n)\,(1+2\,n).$$

b) Leiten Sie eine Abschätzung der Form $\mathcal{O}(f(n))$ für den Aufwand zur Durchführung (Anzahl der *flops*) der Cholesky-Zerlegung in Abhängigkeit von n her.

c) In welchem Verhältnis steht dieser Aufwand zum Aufwand der Durchführung einer LR-Zerlegung ohne Pivotisierung (siehe Vorlesung bzw. Programmieraufgaben)? Wie kann man diese Beziehung intuitiv erklären?

Aufgabe 2 (3 Punkte) (Kondition von Matrizen)

Zeigen Sie, dass für die Konditionszahl bezüglich einer submultiplikativen Matrixnorm bei regulären, quadratischen Matrizen A und B die Beziehungen

a) $\kappa(AB) \leq \kappa(A)\,\kappa(B)$,

b) $\kappa(c\,A) = \kappa(A)$ für $c \in \mathbb{R} \setminus \{0\}$ und

c) $\kappa(A) \geq 1$

gelten.

Aufgabe 3 **(2 Punkte)** **(Zeilenskalierung)**

a) Wie lautet die Systemmatrix DA nach Zeilenäquilibrierung (siehe Übung 6) für

$$A = \begin{bmatrix} 1 & 100 & 2 & 0 \\ 3 & 1000 & 1 & 4 \\ 10 & 1 & 9 & 20 \\ 6 & 0 & 8 & 6 \end{bmatrix} ?$$

Wie groß sind $\kappa(A)$ und $\kappa(DA)$?

b) Überprüfen Sie die Wirkung der Zeilenäquilibrierung auf die Konditionszahl $\kappa_\infty := \|A\|_\infty \cdot \left\|A^{-1}\right\|_\infty$ am Beispiel der Matrix

$$A = \begin{bmatrix} 4 & -1 \\ -1 & 1 \end{bmatrix}.$$

Welche Wirkung ergibt sich für $\kappa_2 := \|A\|_2 \cdot \left\|A^{-1}\right\|_2$? (Hinweis: Die Eigenwerte der Matrix $D\,A$ dürfen mit MATLAB berechnet werden.)

Programmieraufgaben

Aufgabe 1 **(2 Punkte)** **(Zeilenskalierung)**

Schreiben Sie ein MATLAB-Programm zur Zeilenäqulibrierung einer Matrix $A \in \mathbb{R}^{m\times n}$ mit der Schnittstelle

```
function [DA, d] = zeilenskalierung(A, p).
```

Dabei soll A die Eingabematrix sein und $p \in \{1, 2, \infty\}$ soll angeben, bezüglich welcher Norm die Kondition von A und der skalierten Matrix DA berechnet wird. Das Programm soll beide Konditionszahlen zur Information ausgeben. Der Vektor(!) d soll die Diagonaleinträge der Matrix D enthalten. Testen Sie das Programm mit den Matrizen aus Aufgabe 3 aus dem theoretischen Teil, sowie mit der Hilbert-Matrix (siehe Vorlesung) für die Werte $n \in \{10, 50, 100\}$.

Aufgabe 2 **(3 Punkte)** **(Aufwand bei Matrixzerlegungen)**

In dieser Aufgabe sollen verschiedene Matrixzerlegungen hinsichtlich des dafür benötigten Rechenaufwands verglichen werden. Insbesondere betrachten wir die Zerlegungen

$$A = Q \cdot R \quad \big(\mathcal{O}(8/3\,n^3)\big), \quad A = L \cdot R \quad \big(\mathcal{O}(2/3\,n^3)\big) \quad \text{und} \quad A = G^\top G \quad \big(\mathcal{O}(?)\big).$$

Den Aufwand zur Durchführung der Cholesky-Zerlegung entnehmen Sie Aufgabe 2 aus dem Theorieteil oder einer geeigneten Literaturquelle. Wir nehmen für diese Aufgabe auch an, dass die LR-Zerlegung ohne Pivotisierung durchgeführt werden kann.

Machen Sie sich mit den entsprechenden MATLAB-Routinen (`qr`, `lu` und `chol`) welche diese Zerlegungen implementieren vertraut. Schreiben Sie ein MATLAB-Skript, das anhand der Matrix

$$A = \begin{bmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n\times n} \tag{1}$$

den Aufwand der Routinen ermittelt. Wählen Sie z.B. $n \in \{25 : 25 : 1000\}$ und vergleichen Sie die tatsächlich ermittelten Laufzeiten mit dem theoretisch vorhergesagten Verhalten. Stellen Sie die Ergebnisse in geeigneter Weise (Achsenbeschriftungen, Legende, ...) grafisch dar. Verwenden Sie zur Speicherung der Matrizen kein spezielles `sparse`-Format (siehe Zusatz), sondern speichern Sie tatsächlich alle Matrixeinträge ab.

Hinweis: Eine Zeitnehmung kann in MATLAB mit den Befehlen `tic` und `toc` realisiert werden.

Zusatz **2 Punkte**

Matrizen in denen nur "wenige" Einträge ungleich Null sind (sogenannte "dünn besetzte" Matrizen) werden üblicherweise in einem speziellen `sparse`-Format gespeichert, d.h. man merkt sich nur diese Einträge und ihre Positionen anstatt die ganze Matrix zu speichern. MATLAB kann mit dünn besetzten Matrizen arbeiten; es gibt sogar einen Befehl, der Matrizen der Form (1) für beliebiges n im `sparse`-Format erzeugt. Die Umwandlung zwischen den Formaten erfolgt mit `F = full(S)` bzw. `S = sparse(F)`.

Was ändert sich am Laufzeitverhalten der obigen Zerlegungen, wenn man nicht die vollen Matrizen verwendet, sondern nur mit dünn besetzten Matrizen arbeitet? Wie ist dieses Verhalten zu erklären?