Prof. Dr. Karsten Urban
Dipl.-Math. oec. Klaus Stolle
Institut für Numerische Mathematik
Universität Ulm

SoSe 2014
Blatt 2
16.05.2014

Angewandte Numerik 1

Abgabetermin: Freitag, 30.05.2014, **vor der Übung**

Dieses Übungsblatt hat eine Bearbeitungszeit von zwei Wochen.
Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.

Aufgabe 8 *(Induzierte Matrixnormen)* (3+3 Punkte)

Wir betrachten den Vektorraum $X := \mathbb{R}^n$, $n \in \mathbb{N}$ und $x \in X$. Eine Matrixnorm $\|A\|_*$ heißt durch die Vektornorm $\|x\|_*$ induziert, falls gilt

$$\|A\|_* = \max_{x \neq 0} \frac{\|Ax\|_*}{\|x\|_*}.$$

Zeigen Sie:

a) Die Spaltensummennorm

$$\|A\|_1 := \max_j \sum_{k=1}^n |a_{kj}|$$

ist von der Betragsnorm $\|x\|_1 := \sum_{i=1}^n |x_i|$ induziert.

b) Die Zeilensummennorm

$$\|A\|_\infty := \max_i \sum_{k=1}^n |a_{ik}|$$

ist von der Maximumsnorm $\|x\|_\infty := \max_{i=1,\dots,n} |x_i|$ induziert.

Tipp: Zeigen Sie zuerst die Verträglichkeit der Matrixnorm mit der Vektornorm. Beachten Sie die äquivalenten Formulierungen $\|A\|_* = \max_{x\neq 0} \frac{\|Ax\|_*}{\|x\|_*} = \max_{\|x\|_*=1} \|Ax\|_*$.

Aufgabe 9 *(Konditionszahl von Matrizen)* (2+2 Punkte)

a) Berechnen Sie für die Matrix

$$A = \begin{pmatrix} 5 & -2 \\ -6 & 4 \end{pmatrix}.$$

die Konditionszahlen

$$\kappa_*(A) = \|A\|_* \cdot \|A^{-1}\|_*$$

bezüglich der Spaltensummennorm $\|\cdot\|_1$, der Zeilensummennorm $\|\cdot\|_\infty$ und der Frobeniusnorm $\|\cdot\|_F$. Die Frobeniusnorm ist für eine $m \times n$-Matrix definiert als

$$\|\cdot\|_F := \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{i,j}|^2}.$$

Die Frobeniusnorm ist verträglich mit der euklidischen Vektornorm, wird aber nicht durch diese induziert.

b) Überprüfen Sie Ihre Ergebnisse aus a) mit Matlab. Lesen Sie dazu in der Matlab-Hilfe nach, mit welchem Matlab-Befehl man die Konditionszahl einer Matrix berechnen kann.

Berechnen Sie mit Matlab auch die Konditionszahl der Matrix A bezüglich der Spektralnorm. Die Spektralnorm

$$\|A\|_2 := \sqrt{\lambda_{max}(A^TA)}$$

ist die von der Euklidischen Vektornorm $\|x\|_2 := \sqrt{\sum_{i=1}^n |x_i|^2}$ induzierte Matrixnorm. Hierbei bezeichne $\lambda_{max}(A^TA)$ den maximalen Eigenwert von A^TA.

Aufgabe 10 *(Rechenaufwand bei Dreiecksmatrizen)* (2 Punkte)

Sei $R \in \mathbb{R}^{n\times n}$ eine reguläre obere Dreiecksmatrix und $x \in \mathbb{R}^n$. Berechnen Sie den Rechenaufwand FLOP(Rx) der Matrix-Vektor-Multiplikation. Gehen Sie dabei davon aus, dass jede skalare Multiplikation und jede skalare Addition jeweils eine Gleitkommaoperation (FLOP) ist.

Aufgabe 11 *(Programmieraufgabe: LR-Zerlegung ohne Pivotisierung)* (4+4 Punkte)

a) Schreiben Sie eine Matlab-Funktion `[L,R] = lr(A)` zur Berechnung der LR-Zerlegung einer Matrix $A \in \mathbb{R}^{n\times n}$. Programmieren Sie speicherplatzsparend. Formen Sie A schrittweise um und speichern Sie die linke untere Dreiecksmatrix L in den frei werdenden Bereichen von A. Ihre Funktion soll die berechneten Matrizen L und R zurück liefern. Ist die LR-Zerlegung ohne Pivotsuche nicht möglich oder nicht sinnvoll, soll Ihr Programm eine Warnung ausgeben.

b) Schreiben Sie eine Matlab-Funktion `x = solve(L,R,b)`, die ein Gleichungssystem $LRx = b$ mittels Vorwärts- und Rückwärtseinsetzen löst. Schreiben Sie zum Test Ihrer Matlab-Funktionen ein Matlab-Skript `testLR`, welches mindestens das Gleichungssystem

$$Ax = b \quad \text{mit} \quad A = \begin{pmatrix} 6 & -4 & 7 \\ -12 & 5 & -12 \\ 18 & 0 & 22 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} \frac{41}{12} & \frac{-22}{3} & \frac{29}{2} \end{pmatrix}^T.$$

numerisch löst.

Aufgabe 12 *(LR-Zerlegung mit Spalten-Pivotisierung)* (3+3+2+2 Punkte)

Gegeben seien

$$A = \begin{pmatrix} 2 & 3 & -1 & 0 \\ -6 & -5 & 0 & 2 \\ 2 & -5 & 6 & -6 \\ 4 & 6 & 2 & -3 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} 20 \\ -33 \\ -43 \\ 49 \end{pmatrix}.$$

a) Bestimmen Sie, falls existent, mit der Spaltenpivotstrategie die Matrizen P, L und R der Zerlegung $PA = LR$. Hierbei bezeichne P die Permutationsmatrix.

b) Lösen Sie mit Hilfe der Zerlegung $PA = LR$ das Gleichungssystem $Ax = b$.

c) Berechnen Sie die Determinante der Matrix A.

Hinweise: Verwenden Sie hierbei, dass nach dem Determinanten-Multiplikationssatz für zwei Matrizen A und B die Gleichung $\det(AB) = \det(A)\det(B)$ gilt. Überlegen Sie sich, wie die Determinante $\det(P)$ der Permutationsmatrix P lautet.

d) Schätzen Sie für den allgemeinen Fall einer Matrix $A \in \mathbb{R}^{n \times n}$ den Rechenaufwand (in Abhängigkeit von n) zur Berechnung der Determinante $det(A)$ mit der LR-Zerlegung grob ab. Vergleichen Sie dieses Ergebnis mit dem bei Verwendung des Laplaceschen Entwicklungssatzes benötigten Aufwand. Wie lange würde ein 100 Mflops-Rechner (100 Mflop pro Sekunde) für eine 20×20-Matrix in beiden Fällen brauchen?

Aufgabe 13 *(Programmieraufgabe: LR-Zerlegung mit Spalten-Pivotisierung)* (6+2+3*+4*+2* Punkte)

a) Erweitern Sie Ihre Matlab-Funktion `[L,R] = lr(A)` aus Aufgabe 11 a) zu einer Matlab-Funktion `[L,R,P] = lrPivot(A)` zur Berechnung der LR-Zerlegung mit Spaltenpivotsuche. Ihre Funktion soll die Dreiecks-Matrizen L und R sowie die volle Permutationsmatrix P zurück geben.

b) Erweitern Sie Ihre Funktion `x = solve(L,R,b)` aus Aufgabe 11 b) zu einer Matlab-Funktion `x = solvePivot(L,R,P,b)`, die mit Hilfe Ihrer Funktion `lrPivot` ein Gleichungssystem $Ax = b$ löst.

c) Lösen Sie mit ihren Routinen `solve` und `solvePivot` jeweils die folgenden Gleichungssysteme $Ax = b$ numerisch. Hierbei seien

(i)

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

(ii)

$$A = \begin{pmatrix} 11 & 44 & 1 \\ 0.1 & 0.4 & 3 \\ 0 & 1 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

(iii)

$$A = \begin{pmatrix} 0.001 & 1 & 1 \\ -1 & 0.004 & 0.004 \\ -1000 & 0.004 & 0.000004 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Vergleichen Sie ihre Ergebnisse hinsichtlich Robustheit und Genauigkeit der Verfahren `lr` und `lrPivot`. Als Referenzlösung können Sie das Ergebnis von `A\b` betrachten.

d) Wenden Sie Ihre Funktion `lrPivot` auf Bandmatrizen an. Welche Form haben die resultierenden Matrizen L und R? Wie erklären Sie sich Ihre Beobachtung?

Tipp: Sie können beispielsweise mit

```
A = diag(ones(n,1)) + diag(2*ones(n-1,1),1) + diag(2*ones(n-1,1),-1) ...
      + diag(4*ones(n-2,1),2) + diag(4*ones(n-2,1),-2);
```

eine Bandmatrix der Dimension n erzeugen.

e) Was beobachten Sie, wenn Ihre Bandmatrix zusätzlich in der linken unteren Ecke ein betragsmäßig großes Element enthält. Verfolgen Sie die einzelnen Schritte Ihrer LR-Zerlegung.

Aufgabe 14 *(Cholesky-Zerlegung)* (6* Punkte)

Berechnen Sie die Cholesky-Zerlegung von

$$A = \begin{pmatrix} 1 & -1 & 1 & 0 \\ -1 & 2 & 0 & 1 \\ 1 & 0 & 6 & -3 \\ 0 & 1 & -3 & 30 \end{pmatrix}.$$

Aufgabe 15 *(Programmieraufgabe: Cholesky-Zerlegung)* (4+4+2 Punkte)

a) Schreiben Sie eine Matlab-Funktion `L = cholesky(A)`, welche zu einer symmetrisch positiv definiten Matrix $A \in \mathbb{R}^{n\times n}$ die Cholesky-Zerlegung bestimmt, also eine untere Dreieckmatrix L berechnet mit $A = LL^T$. Testen Sie ihre Funktion mit Hilfe des Skriptes `testCholesky.m`, welches auf der Homepage verfügbar ist.

b) Schreiben Sie eine Matlab-Funktion `L = choleskyBand(A,m)`, welche zu einer symmetrisch positiv definiten Matrix $A \in \mathbb{R}^{n\times n}$ mit einer **Bandmatrixstruktur** und Bandbreite $m \in \mathbb{N}$, $m < n$ die Cholesky-Zerlegung bestimmt, also eine untere Dreieckmatrix L berechnet mit $A = LL^T$. Testen Sie ihre Funktion mit Hilfe des Skriptes `testCholeskyBand.m`, welches auf der Homepage verfügbar ist.

c) Führen Sie das bereitgestellte Skript `testRuntime.m` aus und interpretieren Sie das Ergebnis. Was beobachten Sie hinsichtlich der Aufwandsabschätzungen aus den Sätzen 3.6.9 und 3.7.5? Beachten Sie, dass das Schaubild doppelt logarithmisch geplottet wird.

Hinweise:
Die Programmieraufgaben sind in Matlab zu erstellen. Senden Sie alle Files in einer E-mail mit dem Betreff **Loesung-Blatt02** an **angewandte.numerik@uni-ulm.de** (Abgabetermin jeweils wie beim Theorieteil). Drucken Sie zusätzlich allen Programmcode sowie die Ergebnisse aus und geben Sie diese vor der Übung ab. Der Source Code sollte strukturiert und, wenn nötig, dokumentiert sein.