Institut für Informationsverarbeitung
Leibniz Universität Hannover
http://www.tnt.uni-hannover.de

Scientific Computing I - Übung 4

1 Theorie

a) Sei $N_i = \max\{k \mid \forall 1 \leq j \leq k : a_{ij} = 0\}$ wobei a_{ij} Einträge in einer Matrix $A \in \mathbb{R}^{m \times n}$ sind. Die Zahl N_i gibt also an, wie viele führende Nullen in einer Zeile i sind. A (mit $m \geq n$) ist in Zeilenstufenform genau dann wenn gilt: $N_1 < N_2 < \cdots < N_k < N_{k+1} = N_{k+2} = \cdots = N_m = n$, d.h. die Anzahl der führenden Nullen von Zeile zu Zeile streng monoton steigt, bis nach der k-ten Zeile alle Einträge Null sind. Eine obere Dreiecksmatrix ist also ein Spezialfall einer Matrix in Zeilenstufenform, bei der gilt: $m = n$ und $N_i = i - 1$. Formt man eine Matrix A mit Hilfe der Gaußschen Elimination (inkl. Pivotieren) in Zeilenstufenform um, so gibt k (bestimmt durch die letzte Zeile k, die von Null verschiedene Einträge hat) den Rang der Matrix A an. Bestimmen Sie den Rang folgender Matrix:

$$A = \begin{bmatrix} 4 & 3 & 7 \\ 2 & 1 & 1 \\ 6 & 0 & 5 \\ 0 & 0 & 1 \end{bmatrix}$$

b) Lösen Sie das Gleichungssystem $Ax = b$ wobei $LU = A$ eine Cholesky-Zerlegung mit $L = \begin{bmatrix} 3 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 4 & 1 \end{bmatrix}$ und $b = \begin{bmatrix} 21 \\ 15 \\ 24 \end{bmatrix}$ ist.

c) Berechnen Sie das Inverse A^{-1} der Matrix $A = \begin{bmatrix} 2 & 3 & 2 \\ 4 & 2 & 3 \\ 9 & 6 & 7 \end{bmatrix}$. Formen Sie dazu A zur Einheitsmatrix I um und wenden gleichzeitig alle verwendeten elementaren Umformungen auch auf I an. Durch dieses Vorgehen formen Sie I Schritt für Schritt zu A^{-1} um.

Hinweis: In Aufgabe 3c) werden Sie diesen Ansatz in MATLAB umsetzen. Falls Sie bei der Berechnung auf Papier Probleme haben, lohnt es sich die Aufgabenstellung zu 3c) anzuschauen, da sie bewußt eine andere Formulierung des selben Sachverhalts verwendet. Vielleicht liegt Ihnen diese Formulierung mehr.

2 Cholesky-Zerlegung

Unter bestimmten Bedingungen kann die Lösung mittels LU-Zerlegung vereinfacht werden. Welche Bedingungen sind dies? Erweitern Sie ihre LU-Faktorisierung von Übungsblatt 3, Aufgabe 3 derart, dass für diese Fälle die schnellere Cholesky-Zerlegung durchgeführt wird.

Warum ist nun der Backslashoperator von MATLAB (Übungsblatt 3, Aufgabe 2) im Allgemeinen die bessere Wahl zur Lösung eines linearen Gleichungssystems? Dieser berechnet die Lösung mittels der LU-Faktorisierung (nebst Spezialfälle). Dieser Rechenweg ist in der Regel schneller und genauer als die explizite Berechnung inversen Matrix

3 Gauß-Jordan-Elimination

a) Implementieren Sie eine Funktion `function x = mylinsolve(A, b)`, welche das lineare Gleichungssystem $Ax = b$ mit Hilfe der Gauß-Jordan-Elimination (Kapitel 2, Folie 66) löst. Die Funktion `mylinsolve` soll eine Fehlermeldung zurück geben, falls A nicht quadratisch ist oder es keine eindeutige Lösung gibt.

b) Die Gleichung $A \cdot A^{-1} = I$ hat fast die Form eines linearen Gleichungssystems, wie wir sie bisher betrachtet haben. An Stelle des Lösungsvektors x, steht die Matrix A^{-1} und an Stelle des Vektors b, steht die Einheitsmatrix I. Implementieren Sie eine Funktion `function Ainv = myinv(A)`, welche die Matrix A durch sukzessives Lösen mehrerer linearen Gleichungssystem invertiert und zurück gibt. Falls A nicht invertierbar ist, soll eine Fehlermeldung ausgegeben werden.
Tipp: Gehen Sie spaltenweise vor.

c) Lässt sich eine quadratische Matrix A durch elementare Zeilen- bzw. Spaltenumformungen M_i, wie z.B. das Multiplizieren einer Zeile mit einem Skalar ungleich 0, in die Einheitsmatrix I umformen, so dass also die Gleichung $M_k \cdot ... \cdot M_2 \cdot M_1 \cdot A = I$ gilt, dann gilt $M_k \cdot ... \cdot M_2 \cdot M_1 = A^{-1}$. Verbessern Sie ihre Implementierung von `myinv` mit Hilfe der Gauß-Jordan-Eliminationsmatrizen, welche selbst aus elementaren Umformungen zusammengesetzt sind.
Tipps: $M_k \cdot ... \cdot M_2 \cdot M_1 = M_k \cdot ... \cdot M_2 \cdot M_1 \cdot I$. Nachdem man aus A eine Diagonalmatrix gemacht hat, muss man noch die Einträge auf der Diagonalen auf 1 bringen.

4 Matrix-Invertierung

Auf Übungsblatt 2 haben Sie bereits Minoren kennengelernt, d.h. Determinanten von Untermatrizen (vgl. D_{ij} in Aufgabe 2 von Übungsblatt 2). Mit Hilfe dieser Minoren kann man auch das Inverse einer Matrix A berechnen. Implementieren

Sie eine Funktion `function Ainv = myinv(A)`, welche mit Hilfe der Minoren die Matrix A invertiert und das Ergebnis zurückgibt. Gehen Sie dabei wie folgt vor:

a) Ihre Funktion soll zunächst die Matrix K der Kofaktoren berechnen. Die Einträge k_{ij} von K berechnen man durch $k_{ij} = -1^{i+j} \cdot D_{ij}$.

b) Als nächstes soll die Funktion mit Hilfe des Entwicklungssatzes von LaPlace und K die Determinate d von A berechnen. Ihre Funktion soll anschließend mit einem Fehler abbrechen, falls A nicht invertierbar ist.

c) Abschließend kann A^{-1} durch $A^{-1} = \frac{1}{d} \cdot K^T$ bestimmt werden.

d) Welches Verfahren zur Invertierung, das aus Aufgabe 3c) oder das aus Aufgabe 4a-c), ist genauer? Gehen Sie für die Entscheidung ähnlich wie in Übungsblatt 3, Aufgabe 2 vor: bestimmen Sie 10000 zufällige 10×10-Matrizen A und berechnen deren Inverse jeweils mit beiden Verfahren. Bestimmen Sie die Genauigkeit, indem Sie $B = A \cdot A^{-1} - I$ berechnen und anschließend den Fehler als $\sum_{i,j} |b_{ij}|$ berechnen. Durch bilden des Quotienten der beiden Fehler und auswerten aller 10000 Quotienten können Sie eine Entscheidung treffen.

e) Vergleichen Sie nun das genauere Verfahren aus 4d) mit der MATLAB–Funktion `inv`. Gehen Sie dabei wieder so vor wie in 4d) und entscheiden Sie, welches Verfahren am genauesten ist.