Universität zu Köln
Dr. M. Lanser

Wintersemester 2016/2017

14. Dezember 2016

9.Übung zur Algorithmischen Mathematik und Programmieren

Hinweis: Beachten Sie alle Abgabeformalitäten, die auf dem ersten Übungszettel angegeben wurden.

Aufgabe 1: (6 Punkte)
Lösen Sei das lineare Gleichungssystem $Ax = b$ mit

$$A = \begin{pmatrix} 1 & 1 & 2 \\ 2 & 2 & -1 \\ 2 & 3 & 2 \end{pmatrix}$$

und $b = (4, -2, 3)^T$. Bringen Sie dazu das Gleichungssystem auf obere Dreiecksgestalt und lösen Sie mit Rückwärts-Substitution. Gehen Sie dabei **streng nach dem Algorithmus** der Gauß-Elimination in der kij-Variante **mit Spalten-Pivotsuche** vor. Würde die Elimination ohne Pivotsuche dür dieses Beispiel funktionieren?

Aufgabe 2: (2 + 2 + 2 + 2 = 8 Punkte)
Es seien $A, B \in \mathbb{R}^{n \times n}$ reguläre Matrizen und $x \in \mathbb{R}^n$.

a) Zeigen Sie, dass die Komplexität (die Anzahl der benötigten Rechenoperationen) der Gauß-Elimination für eine Matrix A dem Wert $\frac{2}{3}n^3 + O(n^2)$ entspricht.

b) Leiten Sie die Komplexität der Rückwärts-Substitution her.

c) Leiten Sie die Komplexität der Matrix-Vektor-Multiplikation $A \cdot x$ her.

d) Leiten Sie die Komplexität der Matrix-Matrix-Multiplikation $A \cdot B$ her.

Aufgabe 3: (3 + 3 = 6 Punkte)
Gegeben sei eine **Bandmatrix** $A \in \mathbb{R}^{n \times n}$ mit halber Bandbreite $r < n$. Das heißt, es gilt für alle alle Einträge a_{ij} mit $|i - j| > r$, dass $a_{ij} = 0$ ist.

a) Entwickeln Sie eine effiziente Gauß-Elimination ohne Pivotsuche für Bandmatrizen, die nur die nötigen Operationen ausführt. Überlegen Sie sich dazu, welche Einträge von L und R auf Grund der Bandstruktur nicht berechnet werden müssen. Geben Sie den Algorithmus in Form eines Pseudocodes an.

b) Bestimmen Sie die Komplexität des von Ihnen vorgeschlagenen Algorithmus. Ziehen Sie einen Vergleich zur Komplexität der vollständigen Gauß-Elimination (s. Aufgabe 2 a)) für den Fall $r \ll n$.

Aufgabe 4: (4 Punkte)
Formulieren Sie eine kji-Variante der Gauß-Elimination in Form eines Pseudocodes. Das heißt, dass im k-ten Eliminationsschritt die Einträge $a_{ij}^{(k+1)}$ mit $i, j > k$ nicht zeilenweise (erst Zeile $k+1$, dann Zeile $k+2$ usw.) berechnet werden, sondern spaltenweise - beginnend mit Spalte $k+1$.

Anmerkung für Interessierte: Eine in MATLAB implementierte kji-Variante, in der die innere Schleife durch Vektoroperationen ersetzt wird (vgl. dazu Übung 8), wird schneller sein als eine entsprechende kij-Variante, da Matrizen in MATLAB spaltenweise abgespeichert werden. Greift man in seinem Programm auf eine Spalte zu, muss der Rechner die Spalte aus dem Arbeitsspeicher laden, um damit arbeiten zu können. Das gelingt, grob gesagt, schneller, wenn die Spalte zusammenhängend im Speicher liegt. Das ist in MATLAB gegeben, da Matrizen eben spaltenweise gespeichert werden. Im Gegensatz zu MATLAB ist z.B. C/C++ zeilenorientiert, d.h. kij-Varianten wären hier schneller.

Abgabedatum: 22.12.2016, 12 Uhr in den Kasten im Mathematischen Insitut