Prof. Dr. Dirk Lebiedz
M.Sc. Pascal Heiter
Dipl.-Math. oec. Klaus Stolle

Universität Ulm
Institut für Numerische Mathematik
Wintersemester 2013/14

Numerische Lineare Algebra - Übungsblatt 2

(Abgabe: Mittwoch, 13. November 2013)

Aufgabe 5 *(Komplexität)* (2+2+2+2=8 Punkte)

Bei der Aufwandsabschätzung von Algorithmen spielt die Landau-Symbolik eine wichtige Rolle, um die Komplexität des Problems und das Verhalten des Algorithmus, z.B. in Abhängigkeit der Dimension des Problems, zu untersuchen. Sei (a_n) eine reellwertige Folge und

$$\mathcal{O}(a_n) := \{(b_n) \subset \mathbb{R} \,:\, \exists\, C > 0\; \exists\, n_0 \in \mathbb{N} : |b_n| \leq C\,|a_n|\;\; \forall\, n \geq n_0\}.$$

Man schreibt $b_n = \mathcal{O}(a_n)$, was eigentlich $b_n \in \mathcal{O}(a_n)$ bedeutet, wenn demnach die Folge (b_n) von gleicher oder niedrigerer $\mathcal{O}$rdnung als (a_n) ist.

Seien $(a_n), (b_n)$ und (c_n) reelle Zahlenfolgen. Beweisen oder widerlegen Sie

a) $b_n = \mathcal{O}(a_n) \quad \Rightarrow b_n + b_n = \mathcal{O}(a_n)$

b) $b_n = \mathcal{O}(a_n) \quad \Rightarrow b_n \cdot b_n = \mathcal{O}(a_n)$

c) $b_n = \mathcal{O}(a_n)$ und $c_n = \mathcal{O}(b_n) \quad \Rightarrow b_n + c_n = \mathcal{O}(a_n)$

d) (b_n) beschränkt $\quad \Rightarrow a_n \cdot b_n = \mathcal{O}(a_n)$

Aufgabe 6 *(Aufwand bei elementaren Operationen)* (3+3+3+3=12 Punkte)

Wir wollen nun die Komplexität elementarer Matrix-Operationen in Abhängigkeit der Dimension anschauen. Berechnen Sie die Anzahl der Floating Point Operationen (=FLOP) der folgenden Berechnungen, wobei wir hier annehmen, dass eine skalare Multiplikation und Addition jeweils eine Operation benötigt. Geben Sie dabei sowohl die genaue Anzahl der FLOP als auch den Aufwand in $\mathcal{O}$-Notation an.

a) $A \cdot B$ für vollbesetzte Matrizen $A \in \mathbb{R}^{n\times m}, B \in \mathbb{R}^{m\times k}$

b) $R \cdot S$ für linke untere Dreiecksmatrizen $R, S \in \mathbb{R}^{n\times n}$

c) $A \cdot x$ für eine vollbesetzte Matrix $A \in \mathbb{R}^{n\times n}$ und $x \in \mathbb{R}^n$

d) $T \cdot x$ für eine Tridiagonalmatrix $T \in \mathbb{R}^{n\times n}$ und $x \in \mathbb{R}^n$.

- Bitte wenden -

Aufgabe 7 *(Cramersche Regel)* (4+3+3=10 Punkte)

Eine Möglichkeit zur direkten Lösung linearer Gleichungssysteme liefert die Cramersche Regel (vgl. Skript S.30ff). Dabei kommt es - in Bezug auf den numerischen Aufwand - auf die Wahl der Methode zur Berechnung der Determinane an.

(a) Sei

$$A = \begin{pmatrix} 2 & 1 & -2 \\ -1 & 0 & 1 \\ 3 & 8 & 4 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}.$$

Lösen Sie das lineare Gleichungssystem $Ax = b$ mit Hilfe der Cramerschen Regel.

(b) Es sei nun $A \in \mathbb{R}^{5\times 5}$ und $b \in \mathbb{R}^5$ gegeben. Bestimmen Sie die Anzahl der FLOP für das Lösen des linearen Gleichungssystems $Ax = b$ mittels Cramerscher Regel. Verwenden Sie für die Methode der Determinantenbestimmung den Laplaceschen Entwicklungssatz (vgl. Skript, S. 31). Dabei dürfen Sie annehmen, dass die Bestimmung von $(-1)^{j+k}$ keine Rechenoperation benötigt.

(c) Der Aufwand für das Lösen eines linearen Gleichungssystem per Cramersche Regel beträgt $\sim (n^2 \cdot n!)$ FLOP. Der derzeit beste Supercomputer **Tianhe-2** in China (Stand 10/2013) rechnet mit etwa 33,8 PetaFLOP pro Sekunde (=FLOPS), d.h. $\sim 33,8 \cdot 10^{15}$ FLOPS. Berechnen Sie die Zeit [in Jahren], wie lange dieser Supercomputer benötigt, um ein lineares Gleichungssystem der Dimension $n = 25$ mittels Cramerschen Regel unter Verwendung des Laplaceschen Entwicklungssatzes zu lösen.

Hinweis: Ein Jahr hat 365,24219052 Tage.

Aufgabe 8 *(LR-Zerlegung, LaTeX-Aufgabe)* (7+3=10 Punkte)

Sei

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

a) Bestimmen Sie die LR-Zerlegung von A und geben Sie die Zwischenschritte an.

b) Lösen Sie damit durch Vorwärts- und Rückwärtseinsetzen das Gleichungssystem

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