

Übungsblatt 2

(Abgabe Fr. 10.05. 2013 um 10 Uhr **vor** der Übung.)
Nächste Woche (3.5.2013) findet **keine** Übung statt!

Aufgabe 6 (Laudau-Symbolik)

(6 Punkte)

Es sei (a_n) eine reellwertige Folge. Dann definiert man

$$\mathcal{O}((a_n)) = \{(b_n) \text{ reellwertige Folge} : \exists C > 0 \exists n_0 \in \mathbb{N} \forall n \geq n_0 : |b_n| \leq C \cdot |a_n|\}.$$

Man schreibt für $(b_n) \in \mathcal{O}((a_n))$ auch etwas salopp $b_n = \mathcal{O}(a_n)$. Zeigen Sie für reellwertige Folgen $(a_n), (b_n)$ folgende Eigenschaften:

- (a) $a_n = \mathcal{O}(a_n)$
- (b) $a_n = c \cdot \mathcal{O}(b_n), c \in \mathbb{R} \Rightarrow a_n = \mathcal{O}(b_n)$
- (c) $b_n = \mathcal{O}(a_n) \iff \mathcal{O}(b_n) \subset \mathcal{O}(a_n)$
- (d) $c_n = \mathcal{O}(b_n), b_n = \mathcal{O}(a_n) \Rightarrow c_n = \mathcal{O}(a_n)$
- (e) $a_n + b_n = \mathcal{O}(\max(|a_n|, |b_n|))$
- (f) Man gebe zwei Folgen $(a_n), (b_n)$ an, für die weder $\mathcal{O}(b_n) \subset \mathcal{O}(a_n)$ noch $\mathcal{O}(a_n) \subset \mathcal{O}(b_n)$ gilt.

Lösung (Je 1 Punkt)

- (a) Die Wahl von $C = 1$ liefert das Gewünschte.
- (b) $a_n = c \cdot \mathcal{O}(b_n), c \in \mathbb{R} \Rightarrow \exists C > 0 \exists n_0 \in \mathbb{N} \forall n \geq n_0 : |a_n| \leq C \cdot |c \cdot b_n| = |c| \cdot C \cdot |b_n|$
Also gilt mit $\tilde{C} = |c| \cdot C$:
 $\exists \tilde{C} > 0 \exists n_0 \in \mathbb{N} \forall n \geq n_0 : |a_n| \leq \tilde{C} \cdot |b_n|$
und somit $a_n = \mathcal{O}(b_n)$.
- (c) " $\Rightarrow$ " Sei $b_n = \mathcal{O}(a_n)$ und $c_n = \mathcal{O}(b_n)$. Dann folgt:

$$\begin{aligned} \exists C_0 > 0 \exists n_0 \in \mathbb{N} \forall n \geq n_0 : |b_n| &\leq C_0 \cdot |a_n| \\ \exists C_1 > 0 \exists n_1 \in \mathbb{N} \forall n \geq n_1 : |c_n| &\leq C_1 \cdot |b_n| \end{aligned}$$

Daraus folgt: $\forall n \geq \max\{n_0, n_1\} : |c_n| \leq C_1 \cdot C_0 \cdot |a_n|$. Also gilt $c_n = \mathcal{O}(a_n)$ und damit $\mathcal{O}(b_n) \subset \mathcal{O}(a_n)$.
" $\Leftarrow$ " Es gilt $\mathcal{O}(b_n) \subset \mathcal{O}(a_n)$. Also genügt es zu zeigen, dass $b_n = \mathcal{O}(b_n)$. Dies ist klar, da $|b_n| \leq 1 \cdot |b_n|$.

- (d) $c_n = \mathcal{O}(b_n) \Rightarrow \exists C_1 > 0 \exists n_1 \in \mathbb{N} \forall n \geq n_1 : |c_n| \leq C_1 \cdot |b_n|$
 $b_n = \mathcal{O}(a_n) \Rightarrow \exists C_2 > 0 \exists n_2 \in \mathbb{N} \forall n \geq n_2 : |b_n| \leq C_2 \cdot |a_n|$
Also gilt $\exists C = C_1 \cdot C_2 > 0 \exists n_0 = \max\{n_1, n_2\} \in \mathbb{N} \forall n \geq n_0 : |c_n| \leq C \cdot |a_n|$ und somit $c_n = \mathcal{O}(a_n)$
- (e) Aus der Dreiecksungleichung folgt:

$$|a_n + b_n| \leq |a_n| + |b_n| \quad \forall n \in \mathbb{N}$$

Diesen Term kann man weiter abschätzen:

$$|a_n| + |b_n| \leq 2 \cdot \max(|a_n|, |b_n|) \quad \forall n \in \mathbb{N}$$

- (f) Wähle $a_n = 1$ und $b_n = n \cdot |\sin(\frac{\pi \cdot n}{2})|$

Da

$$\frac{|b_n|}{|a_n|} = n \cdot \left| \sin\left(\frac{\pi \cdot n}{2}\right) \right|$$

unbeschränkt ist gilt: $b_n \notin \mathcal{O}(b_n)$. Daraus folgt wiederum: $\mathcal{O}(b_n) \not\subseteq \mathcal{O}(a_n)$

Andererseits gilt für gerades n :

$$1 \not\leq C \cdot 0$$

weshalb $a_n \notin \mathcal{O}(b_n)$. Daraus folgt wiederum: $\mathcal{O}(a_n) \not\subseteq \mathcal{O}(b_n)$

Aufgabe 7 (LR-Zerlegung)

(6 Punkte)

(a) Gegeben sei die Tridiagonalmatrix

$$A = \begin{pmatrix} a_1 & c_1 & & & \\ b_1 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-2} & a_{n-1} & c_{n-1} \\ & & & b_{n-1} & a_n \end{pmatrix} \in \mathbb{R}^{n \times n}$$

Leiten Sie für den Fall, dass die LR-Zerlegung

$$A = LR, \quad L = \begin{pmatrix} 1 & & & & \\ d_1 & 1 & & & \\ & \ddots & \ddots & & \\ & & d_{n-1} & 1 & \end{pmatrix}, \quad R = \begin{pmatrix} e_1 & f_1 & & & \\ & e_2 & f_2 & & \\ & & \ddots & \ddots & \\ & & & e_{n-1} & f_{n-1} \\ & & & & e_n \end{pmatrix},$$

existiert, Rekursionsformeln für d_j, e_j und f_j her. Welchen Aufwand hat die Berechnung der LR-Zerlegung mittels dieser Rekursion?

Lösung Durch Multiplizieren der Matrizen L und R ergibt sich

$$L \cdot R = \begin{pmatrix} e_1 & f_1 & 0 & \dots & 0 \\ d_1 e_1 & d_1 f_1 + e_2 & f_2 & & \\ 0 & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & f_{n-1} \\ 0 & \dots & d_{n-1} e_{n-1} & d_{n-1} f_{n-1} + e_n \end{pmatrix}$$

Es ergibt sich sofort, dass $f_j = c_j, \forall j = 1, \dots, n-1$ und $e_1 = a_1$ gilt. Die Einträge e_j und d_j lassen sich dann Spaltenweise berechnen, d.h. mit e_1 kann d_1 berechnet werden, mit d_1 kann e_2 berechnet werden, usw. (4 Punkte).

Für den Aufwand ergibt sich also:

Für die Bestimmung der Einträge $e_j, j = 2, \dots, n$ 1 Multi+1 Add = 2 FLOP's

Für die Bestimmung der Einträge $d_j, j = 1, \dots, n-1$ 1 Div = 1 FLOP

Insgesamt ergibt sich $3(n-1) = \mathcal{O}(n)$, also linearer Aufwand. (2 Punkte)

Aufgabe 8 (Schachbrett-Matrix)

(3+3 Punkte)

Gegeben sei eine Matrix $A \in \mathbb{R}^{n \times n}$ (n gerade), die ein Schachbrett-Muster aufweist:

$$A = \begin{pmatrix} a_{1,1} & 0 & a_{1,3} & 0 & \cdots & a_{1,n-1} & 0 \\ 0 & a_{2,2} & 0 & a_{2,4} & \cdots & 0 & a_{2,n} \\ a_{3,1} & 0 & a_{3,3} & 0 & \cdots & a_{3,n-1} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n-1,1} & 0 & a_{n-1,3} & 0 & \cdots & a_{n-1,n-1} & 0 \\ 0 & a_{n,2} & 0 & a_{n,4} & \cdots & 0 & a_{n,n} \end{pmatrix}$$

Um beim Lösen des Gleichungssystems $Ax = b$ sowohl Speicher als auch Rechenaufwand zu sparen, kann wie folgt verfahren werden:

1. Spalte die Matrix auf in zwei Teilmatrizen:

$$A_1 = \begin{pmatrix} a_{1,1} & a_{1,3} & \cdots & a_{1,n-1} \\ a_{3,1} & a_{3,3} & \cdots & a_{3,n-1} \\ \vdots & \vdots & & \vdots \\ a_{n-1,1} & a_{n-1,3} & \cdots & a_{n-1,n-1} \end{pmatrix}, \quad A_2 = \begin{pmatrix} a_{2,2} & a_{2,4} & \cdots & a_{2,n} \\ a_{4,2} & a_{4,4} & \cdots & a_{4,n} \\ \vdots & \vdots & & \vdots \\ a_{n,2} & a_{n,4} & \cdots & a_{n,n} \end{pmatrix}$$

2. Löse die Gleichungssysteme

$$A_1(x_1, x_3, \dots, x_{n-1})^T = (b_1, b_3, \dots, b_{n-1})^T, \quad A_2(x_2, x_4, \dots, x_n)^T = (b_2, b_4, \dots, b_n)^T.$$

Bearbeiten Sie die folgenden Aufgaben:

- (a) Berechnen Sie den Rechenaufwand für obige Methode zur Lösung eines Gleichungssystems mit Schachbrett-Matrix und vergleichen Sie ihn mit dem Aufwand zum Lösen des vollen Systems mit einer einfachen LR Zerlegung und Vorwärts-Rückwärtseinsetzen für $n = 2, 4, 6, 20$. Um welchen Faktor ist die angepasste Methode asymptotisch schneller als die Standard-LR Zerlegung mit Vorwärts-Rückwärtseinsetzen?
- (b) Lösen Sie das Gleichungssystem $Ax = b$ mit

$$A = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 3 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \quad b = (1 \ 2 \ 3 \ 1)^T.$$

mit dem oben beschriebenen Algorithmus.

Lösung

- (a) Der Rechenaufwand zum Lösen des Gleichungssystems $Ax = b$ ($A \in \mathbb{R}^{n \times n}$) mit LR Zerlegung beträgt

$$\frac{4n^3 + 9n^2 - n}{6} = \frac{4}{6}n^3 + \mathcal{O}(n^2).$$

Mit der angegebenen Methode müssen zwei Gleichungssysteme der Größe $\frac{n}{2}$ gelöst werden. Dies entspricht einem Aufwand von

$$2 \cdot \frac{4\left(\frac{n}{2}\right)^3 + 9\left(\frac{n}{2}\right)^2 - \frac{n}{2}}{6} = \frac{2n^3 + 9n^2 - 2n}{12} = \frac{1}{6}n^3 + \mathcal{O}(n^2).$$

Folgende Tabelle zeigt den Rechenaufwand für das Lösen des Vollen bzw. der beiden kleinen Systeme.

n	Volles System	2 kleinere Systeme
2	11	4
4	66	22
6	197	62
10	815	240
20	5930	1630

Für große n hat der neue Algorithmus ca. 4mal weniger Aufwand als der Standard Algorithmus.

(b) Wir lösen die beiden Systeme

$$\begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} \quad \begin{pmatrix} 3 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_2 \\ x_4 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

und erhalten

$$\begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ -1 \end{pmatrix} \quad \begin{pmatrix} x_2 \\ x_4 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$$

Aufgabe 9 (MATLAB)

(4+4+2 Punkte)

- (a) Schreiben Sie eine Matlab-Funktion `LR = lr(A)` zur Berechnung der LR -Zerlegung von A . Schreiben sie die Funktion so, dass **kein** zusätzlicher Speicherplatz benötigt wird und nur eine Matrix zurückgegeben wird, in der die Matrix R und L gemeinsam gespeichert sind.
Ist die LR -Zerlegung ohne Pivotsuche nicht möglich, soll das Programm eine Fehlermeldung ausgeben.
- (b) Schreiben Sie eine Funktion `x = solve(LR,b)`, die die Matrix aus Aufgabenteil (a) übergeben bekommt und dann mittels Vorwärts- und Rückwärtseinsetzen das Gleichungssystem $Ax = b$ löst.
- (c) Schreiben Sie ein Skript `main.m`, das ein beliebiges (nichtriviales) Gleichungssystem der Größe 5 löst und vergleichen Sie das Ergebnis mit dem Ergebnis des MATLAB `'\'`-Operators.

Drucken sie den Code sowie das Schaubilder aus und geben Sie diese mit ab. Senden Sie außerdem alle Funktionen in einer Email an folgende Adresse

angewandtenumerik@gmail.com
Betreff: Blatt01, Name1 Vorname1, Name2 Vorname2