Dr. Gerhard Baur
Manuel Bernhard
Wintersemester 2011/2012

Übungsblatt 13

Lineare Algebra für Informatiker und Ingenieure

(Abgabe ist zu **zweit** am 6.2.2012 um 12:10 Uhr in H22 **vor** der Übung)

Aufgabe 1 (*Gram-Schmidt*) (4)

Man orthonormalisiere die Vektoren

$$v_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 4 \\ 0 \\ 1 \\ 2 \end{pmatrix}, \quad v_3 = \begin{pmatrix} 3 \\ 8 \\ 1 \\ -2 \end{pmatrix}$$

mit dem Gram-Schmidt-Verfahren.

Aufgabe 2 (*Ähnlichkeit von Matrizen*) (3+1+1+3*)

Man nennt zwei Matrizen $A, B \in K^{n,n}$ ähnlich und schreibt $A \sim B$, wenn es eine reguläre Matrix $S \in \mathrm{GL}_n(K)$ mit

$$A = S^{-1}BS$$

gibt. Man zeige:

(a) Es ist $\sim$ eine Äquivalenzrelation auf $K^{n,n}$.

(b) Es gilt $\det A = \det B$, falls A und B ähnlich sind.

(c) Wenn $A \sim B$ gilt, so haben A und B dieselben Eigenwerte. Ist weiter λ ein Eigenwert von A und B, dann sind die algebraischen Vielfachheiten identisch.

(d) Sind wieder A und B ähnlich, dann sind die geometrischen Vielfachheiten eines Eigenwertes λ von A und B identisch.

Aufgabe 3 (*Matrixpotenzen*) (5+2)

(a) Man berechne die Potenzen A^m, B^m, C^m und D^m (für $m \in \mathbb{N}$) der folgenden Matrizen (Für A, B, und D kann man die Ergebnisse von Blatt 12 Aufgabe 3 benutzen).

$$A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 3 \end{pmatrix} \quad C = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ -2 & 0 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 1 & 0 & 1 \\ -1 & 2 & 1 \\ -2 & 0 & 4 \end{pmatrix}$$

(b) Man finde eine Matrix $M \in \mathbb{C}^{n,n}$ mit der Eigenschaft

$$M^2 = \begin{pmatrix} 0 & 3 & -1 \\ 1 & 2 & -1 \\ 0 & 3 & -1 \end{pmatrix}.$$

Bitte wenden!

Aufgabe 4 (*Fehlererkennung über verrauschten Kanal*) (4+1*)
Wir stellen uns vor, dass wir Daten über einen verrauschten Kanal übermitteln (Internet, Lesen einer DVD,...). Oft werden für solche Leseoperationen geschickte Kodierungsalgorithmen verwendet. Diese können sehr effizient Fehler erkennen und diese auch korrigieren. In vielen Anwendungen (so beispielsweise auf einer Festplatte) benutzt man einfachere Verfahren, welche aber nur Fehler erkennen können. Ein Beispiel ist das Paritätsbit p, welches für ein Wort $w_1...w_n$ bestehend aus 0'en und 1'en über

$$p = \sum_{i=1}^{n} w_i \bmod 2$$

berechnet wird. Dieses wird dann mit übertragen. Tritt nur ein Übertragungsfehler (genauer ein Bitflip) auf, so kann dies erkannt werden. Eine gute Eigenschaft des Paritätsbit ist, dass für Worte der Länge n genau die Hälfte zu $p = 1$ und die andere Hälfte zu $p = 0$ führt. Das bedeutet hier, dass bei stärkeren Fehlern (so, dass viele nahe beieinander liegende Bits zufällig abgeändert wurden - Man denke hier an das Beispiel der Festplatte!) dies mit Wahrscheinlichkeit $1/2$ erkannt werden kann.
Der Vorteil einer solchen Methode ist, dass dies kaum Aufwand erfordert. Das Ziel der Aufgabe ist, dass wir den Aufwand noch verringern. Aber dafür muss man, wie wir sehen werden, auf einer anderen Ebene Abstriche machen. Dazu geben wir uns zwei Bitworte $x_1...x_n$ und $y_1...y_n$ vor. Wir berechnen nun für die beiden Worte nur ein Check-Bit

$$c = \sum_{i=1}^{n} x_i y_i \bmod 2.$$

Die Frage, die uns nun interessiert ist, wie viele solcher Wortpaare zu $c = 1$ (Diese Zahl nennen wir a_n) und wie viele zu $c = 0$ führen (Diese Zahl bezeichnen wir mit b_n).

(a) Man berechne a_n und b_n in Abhängigkeit von $n \in \mathbb{N}$.
(b) Für große n nähert sich der Wert $4^{-n}a_n$ und $4^{-n}b_n$ (= Anteil der Wortpaare mit $c = 1$ bzw. $c = 0$) dem Wert $1/2$ an. Man überträgt nun mit den beiden Worten auch c. Man vergleiche die beiden Methoden (p- oder c-Bit), wenn nur **ein Übertragungsfehler** auftritt und wenn (hier denke man an die Festplatte) es auch recht häufig vorkommt, dass **mehrere nahe beieinander liegenden Bits** betroffen sind (Sagen wir beim p-Bit die Bits $w_l...w_{l+2m}$ und beim c-Bit die Bits $x_k...x_{k+m}$ und $y_k...y_{k+m}$).