Prof. Dr. Karsten Urban
Dipl.-Math. oec. Klaus Stolle
Institut für Numerische Mathematik
Universität Ulm

SoSe 2017
Blatt 06
27.05.2017

Angewandte Numerik 1

Besprechung in den Tutorien in der Woche vom 05.06.2017 bis 09.06.2017

Für dieses Übungsblatt gibt es 23 Theorie- und 12 Matlab-Punkte, sowie 8 Theorie- und 7 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 60-Prozent-Grenzen liegen aktuell (inklusive Blatt 06) bei 80,4 Theoriepunkten und 55,2 Matlabpunkten.

Aufgabe 25 *(Drehen und Drehmatrizen)* (1T+4T+4T* Punkte)

a) Im $\mathbb{R}^2$ kann ein Vektor durch Multiplikation von links mit der Drehmatrix

$$D_{12} = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}$$

in der x_1-x_2-Ebene entgegen dem Uhrzeigersinn um den Winkel φ gedreht werden. Geben Sie eine Drehmatrix D_{ij} an, mit der ein Vektor $x \in \mathbb{R}^n$ in der x_i-x_j-Ebene ($i, j \in \{1, \dots, n\}$) entgegen dem Uhrzeigersinn um den Winkel φ gedreht werden kann.

b) Die Drehmatrix D_{ij} soll einen Vektor $v = \big(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_{j-1}, x_j, x_{j+1}, \dots, x_n\big)^T$ auf einen Vektor $\tilde{v}$ der Form $\tilde{v} = \big(x_1, \dots, x_{i-1}, \tilde{x}_i, x_{i+1}, \dots, x_{j-1}, 0, x_{j+1}, \dots, x_n\big)^T$ drehen.

Bestimmen Sie die Matrix D_{ij}. Welche Werte kann $\tilde{x}_i$ annehmen? Geben Sie die Komponenten Ihrer Drehmatrix auch für die Spezialfälle $x_i = 0$ und bzw. oder $x_j = 0$ an.

Hinweis: Betrachten Sie zunächst den zweidimensionalen Fall und drücken Sie $\cos\varphi$ und $\sin\varphi$ in Abhängigkeit der Komponenten a und b des Vektors $v = \big(x_i, x_j\big)^T = \big(a, b\big)^T$ aus. Verallgemeinern Sie dann die zweidimensionale Drehmatrix auf den n-dimensionalen Fall.

c) Transformieren Sie die Matrix

$$A = \begin{pmatrix} 4 & 2 \\ 3 & -1 \\ 5 & -2 \end{pmatrix}$$

mit Hilfe sukzessiver Drehungen der Spalten von A auf die Matrix

$$\tilde{A} = \begin{pmatrix} * & * \\ 0 & * \\ 0 & 0 \end{pmatrix}.$$

Geben Sie dabei alle verwendeten Drehmatrizen an.

Aufgabe 26 *(Programmieraufgabe: Eine Ellipse als Lineares Ausgleichsproblem, Givens-Rotation)*
(3T+3M*+4T+5M Punkte)

Gegeben sind die vier Messwert-Paare

x_i	-1.6	-0.9	0.75	2.7
y_i	1.6	-0.9	1.0	-1.0

Es ist bekannt, dass diese Messwerte zu einer Ellipse der Form

$$\alpha x^2 + \beta xy + \gamma y^2 - 10 = 0 \tag{1}$$

mit unbekannten Parametern α, β und γ gehören.

a) Stellen Sie das lineare Ausgleichsproblem $\|Az-b\|_2 \to \min$ zur Bestimmung der unbekannten Parameter auf. Geben Sie A, z und b explizit an. Welche Dimensionen haben diese Größen?

b) Lösen Sie das lineare Ausgleichsproblem mittels Matlab. Fertigen Sie mit Matlab eine Skizze, in der die Ausgleichsellipse und die Messpunkte eingezeichnet sind. Beschriften Sie Ihre Zeichnung.

Hinweis: In der Form (1) ist die Ellipse als sogenannte *implizite Funktion*, die von zwei Parametern x und y abhängt, gegeben. Diese als implizite Funktion gegebene Ellipse können Sie beispielsweise mit der Matlab-Funktion `ezplot` zeichnen.

c) Die Behandlung des obigen Ausgleichsproblems $(A\,|\,b)$ für vier Messwerte führt bei der Lösung mittels orthogonaler Transformation auf ein oberes Dreieckssystem $(R\,|\,Q^T b)$.

Nun erhalten Sie eine weitere Messung (x_5, y_5). Das zugehörige Ausgleichsproblem unter Verwendung von $(R\,|\,Q^T b)$ sei dann

$$\left(\begin{array}{ccc|c} -7.8 & 3.2 & -1.9 & -14 \\ 0 & -2.2 & 0.74 & -4.4 \\ 0 & 0 & -2.2 & -13 \\ 0 & 0 & 0 & -0.26 \\ 5.85 & -2.4 & 0.98 & 10 \end{array}\right).$$

Lösen Sie dieses neue, erweiterte Ausgleichsproblem von Hand durch Anwendung von Givens-Rotationen. Geben Sie dabei die einzelnen Schritte, die resultierenden Matrizen sowie die Lösung $z = (z_1, z_2, z_3)^T$ mit $z_1 = \alpha$, $z_2 = \beta$ und $z_3 = \gamma$ und das unvermeidbare Residuum explizit an.

d) Kontrollieren Sie Ihr Ergebnis aus Aufgabenteil c) mit Matlab. Schreiben Sie hierzu eine Matlab-Funktion `[Q, R] = qrGivens(A)`, welche eine QR-Zerlegung mittels Givens-Rotationen berechnet. Verwenden Sie dabei nicht die Matlab-Funktion `givens`.

Hinweis: Überlegen Sie sich, welche Elemente von A und Q durch eine einzelne Rotation jeweils verändert werden. Wie können Sie in Matlab gezielt diesen Matrixelementen geänderte Werte zuweisen?

Aufgabe 27 *(Householder-Spiegelung)* (2T+2T+3T+3T+1T+4T* Punkte)

Sei $\omega \in \mathbb{R}^n \setminus \{0\}$. Dann heisst die Matrix

$$P := I - 2\frac{\omega\omega^T}{\omega^T\omega}$$

Householder-Matrix und ω der zugehörige Householder-Vektor.

a) Zeigen Sie: P ist symmetrisch und orthogonal.

b) Zeigen Sie, dass für $E := \{x \in \mathbb{R}^n : Px = x\}$ gilt: $E = \{x \in \mathbb{R}^n : \omega^T x = 0\}$.

c) Zeigen Sie, dass Px die Spiegelung von $x \in \mathbb{R}^n$ an der Hyperebene $E := \{x \in \mathbb{R}^n : Px = x\}$ ist, d. h. (vgl. Bemerkung 4.4.3)

$$x \in \mathbb{R}^n \quad \Longrightarrow \quad \exists\, t \in \mathbb{R} : \quad Px = x + t\omega \text{ und } x + \tfrac{t}{2}\omega \in E.$$

Hinweis: Um den zweiten Teil der Aussage zu zeigen, setzen Sie $z := x + \frac{t}{2}\omega$ und berechnen Sie Pz. Verwenden Sie hierbei den ersten Teil der Aussage.

d) Konstruieren Sie zum Vektor

$$x = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

die Householder-Matrix P, sodass x auf die x_1-Achse gespiegelt wird. Stellen Sie die Vektoren x, w und Px graphisch dar und erklären Sie die Bedeutung des Vektors ω.

e) Zeichnen Sie auch die Hyperebene E ein.

f) Berechnen Sie die QR-Zerlegung der Matrix A mit Householder-Spiegelungen

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 2 \\ 1 & 5 \end{pmatrix}.$$

Geben Sie dabei alle Transformations-Matrizen $P^{(i)}$ sowie die Matrix Q und R explizit an.

Aufgabe 28 *(Programmieraufgabe: Householder-Spiegelung)* (4M+3M*+3M+1M* Punkte)

a) Schreiben Sie eine Funktion `[Q, R] = qrHouseholder(A)`, die die QR-Zerlegung einer Matrix $A \in \mathbb{R}^{m \times n}$ mittels Householder-Spiegelungen berechnet.

b) Verifizieren Sie mit Ihrer Funktion `[Q, R] = qrHouseholder(A)` das in Aufgabe 26 c) angegebene obere Dreieckssystem $(R \,|\, Q^T b)$ und die von Ihnen auf dem Papier berechnete Lösung des erweiterten Ausgleichsproblems. Geben Sie in Ihrer Funktion geeignete Zwischenergebnisse aus, so dass Sie sich den Ablauf des Algorithmus verdeutlichen können. Was beobachten Sie? Stimmen die mittels Householder-Transformationen errechneten Ergebnisse mit den mittels Givens-Rotationen erzielten Ergebnissen überein?

c) Schreiben Sie ein Skript `testRuntimeQR.m`, in dem Sie sich beispielsweise für $n = 2^3, ..., 2^9$ eine quadratische Matrix mit Zufallszahlen anlegen (Matlab-Funktion `rand`). Berechnen Sie anschließend für jede dieser Matrizen die QR-Zerlegung mit Givens-Rotationen und Householder-Spiegelungen. Verwenden Sie dazu Ihre Funktionen `[Q, R] = qrHouseholder(A)` und `[Q, R] = qrGivens(A)`. Messen Sie dabei jeweils die Zeit und plotten Sie anschließend diese Zeiten in Abhängigkeit von n in doppelt-logarithmischer Skala. Erklären Sie Ihre Ergebnisse. Hinweis: Um vergleichbare Laufzeiten zu erhalten, sollten Sie jetzt auf die Ausgabe von Zwischenergebnissen verzichten.

d) Vergleichen Sie die Zeiten, die die von Ihnen geschriebenen Funktionen benötigen, auch mit den Zeiten, die die Matlab-Funktion `[Q, R] = qr(A)` benötigt.

Hinweise:
Die Programmieraufgaben sind in Matlab zu erstellen. Der Source Code muss strukturiert und dokumentiert sein. Senden Sie **spätestens 24 Stunden vor Ihrem Tutorium** alle Matlab-Files und alle Ergebnisse in einer E-mail mit dem Betreff **Loesung-Blatt06** an **angewandte.numerik@uni-ulm.de**.