

## Übungsblatt 06

**Besprechung** in den Tutorien in der Woche vom 08.06.2020 bis 12.06.2020

Für dieses Übungsblatt gibt es 34 Theorie- und 21 Matlab-Punkte.

Die 70-Prozent-Grenzen liegen aktuell (inklusive Blatt 06) bei 120,4 Theorie- und 68,6 Matlabpunkten.

**Aufgabe 21** (*Programmieraufgabe: LR-Zerlegung für Tridiagonalmatrizen*) (4T+1T+4M+2M Punkte)

a) Gegeben sei die Tridiagonalmatrix

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & & & \\ a_{2,1} & a_{2,2} & a_{2,3} & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} \\ & & & a_{n,n-1} & a_{n,n} \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

Leiten Sie für den Fall, dass die  $LR$ -Zerlegung

$$A = LR \quad \text{mit} \quad L = \begin{pmatrix} 1 & & & & \\ \ell_{2,1} & 1 & & & \\ & \ddots & \ddots & & \\ & & \ell_{n,n-1} & 1 & \end{pmatrix} \quad \text{und} \quad R = \begin{pmatrix} r_{1,1} & r_{1,2} & & & \\ & r_{2,2} & r_{2,3} & & \\ & & \ddots & \ddots & \\ & & & r_{n-1,n-1} & r_{n-1,n} \\ & & & & r_{n,n} \end{pmatrix}$$

existiert, Rekursionsformeln für  $\ell_{j,j-1}$  ( $j = 2, \dots, n$ ),  $r_{j,j}$  ( $j = 1, \dots, n$ ) und  $r_{j,j+1}$  ( $j = 1, \dots, n-1$ ) her.

- b) Welchen Aufwand hat die Berechnung der  $LR$ -Zerlegung mittels dieser Rekursion? Geben Sie den Aufwand auch in  $\mathcal{O}$ -Notation an.
- c) Schreiben Sie eine MATLAB-Funktion `[lnd, rd, rnd] = LR_TriDiag(aUnd, ad, aOnd)`, die die  $LR$ -Zerlegung einer quadratischen Matrix **A** mit Ihrem Algorithmus aus Aufgabenteil a) berechnet. Dabei sind **lnd** die untere Nebendiagonale ( $\ell_{2,1}, \dots, \ell_{n,n-1}$ ) der Matrix  $L$ , **rd** die Diagonale ( $r_{1,1}, \dots, r_{n,n}$ ) und **rnd** die obere Nebendiagonale ( $r_{1,2}, \dots, r_{n-1,n}$ ) der Matrix  $R$ , **aUnd** die untere Nebendiagonale ( $a_{2,1}, \dots, a_{n,n-1}$ ), **ad** die Diagonale ( $a_{1,1}, \dots, a_{n,n}$ ) und **aOnd** die obere Nebendiagonale ( $a_{1,2}, \dots, a_{n-1,n}$ ) der Matrix  $A$ .
- d) Testen Sie Ihre MATLAB-Funktion `LR_TriDiag` an verschiedenen Beispielen.

**Aufgabe 22** (Programmieraufgabe: Vergleich der Rechenzeiten)

(5M+1T+2M+1T Punkte)

In dieser Aufgabe wollen wir Ihre Überlegungen aus der vorherigen Aufgabe 21 zum Aufwand empirisch überprüfen. Wir betrachten dazu die Matrix

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

- a) Schreiben Sie ein Matlab-Skript `laufzeiten`, in dem Sie für  $n \in \{2^i \mid i = 3, \dots, 14\}$  jeweils für die Matrix  $A$  den Vektor `ad` der Diagonalelemente,  $ad = (2, 2, \dots, 2)$ , und den Vektor `nd` der Elemente der beiden Nebendiagonalen,  $nd = (-1, -1, \dots, -1)$  aufstellen und mit Ihrer MATLAB-Funktion `LR_TriDiag` aus Aufgabe 21 die  $LR$ -Zerlegung der Matrix  $A$  berechnen. Plotten Sie die zur Berechnung der  $LR$ -Zerlegungen benötigten Zeiten (Matlab-Befehle `tic` und `toc`) über die Dimension  $n$  des Gleichungssystems in einem Schaubild mit logarithmischen Achsen (Matlab-Befehl `loglog`). Zeichnen Sie in Ihr Schaubild eine geeignete Steigungsgerade ein.

Beachten Sie, dass Ihr MATLAB-Skript je nach Leistungsfähigkeit Ihres Computers zur Berechnung einige Minuten braucht. Um realistische Zeiten zu erhalten, sollten Sie Ihren Computer in der Zeit, in der die Berechnungen laufen, nicht anderweitig nutzen.

- b) Interpretieren Sie Ihr Schaubild. Decken sich Ihre numerischen Ergebnisse mit Ihren Überlegungen aus der vorigen Aufgabe 21?
- c) Nun möchten Sie sich auch den Laufzeit-Vorteil verdeutlichen, der aus der speziellen Bandstruktur der Matrix  $A$  resultiert. Erweitern Sie dazu Ihr Matlab-Skript `laufzeiten` und berechnen Sie die  $LR$ -Zerlegungen der Matrizen  $A$  auch mit Ihrer MATLAB-Funktion `gaussLR` aus Aufgabe 13 von Übungsblatt 04. Plotten Sie analog zu Aufgabenteil a) die hierfür benötigten Zeiten und zeichnen Sie wieder eine geeignete Steigungsgerade ein.

Falls Sie in Aufgabe 13 mehrere Varianten implementiert haben, können Sie Ihr Matlab-Skript `laufzeiten` um alle diese Varianten erweitern.

- d) Interpretieren Sie Ihr Schaubild. Haben Sie diese numerischen Ergebnisse erwartet?

**Aufgabe 23** (Voraussetzungen für die Durchführbarkeit der Cholesky-Zerlegung)

(8T Punkte)

Berechnen Sie, falls möglich, für die folgenden Matrizen die Cholesky-Zerlegung per Hand:

$$A_1 = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & -1 & 2 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \\ 1 & 1 & -2 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 8 & 2 \\ 1 & 2 & 2 \end{pmatrix}.$$

Falls die Cholesky-Zerlegung nicht existiert, geben Sie an, warum. Begründen Sie dabei Ihre Vermutungen. Verwenden Sie bei allen Rechnungen ausschließlich Brüche und keine Dezimalzahlen und geben Sie alle Zwischenschritte an.

**Aufgabe 24** (Programmieraufgabe: Choleskyzerlegung)

(4M+2M+2M+1T Punkte)

- a) Schreiben Sie eine MATLAB-Funktion `[L, D] = cholesky(A)`, die die Cholesky-Zerlegung  $A = L D L^T$  einer symmetrisch positiv definiten Matrix  $A$  berechnet. Dabei sind  $L$  die linke untere normierte Dreiecksmatrix und  $D$  die Diagonalmatrix mit positiven Diagonaleinträgen der Cholesky-Zerlegung sowie  $A$  die symmetrisch positiv definite Matrix, deren Cholesky-Zerlegung berechnet werden soll.

Ihre MATLAB-Funktion `cholesky` soll eine Fehlermeldung ausgeben, falls für die Matrix  $A$  keine Cholesky-Zerlegung existiert.

- b) Testen Sie Ihre MATLAB-Funktion `cholesky` an den drei Matrizen aus Aufgabe 23 und weiteren Beispielen.
- c) Untersuchen Sie nun auch das Laufzeitverhalten Ihrer MATLAB-Funktion `cholesky`. Erweitern Sie dazu Ihr Matlab-Skript `laufzeiten` und berechnen Sie die Cholesky-Zerlegungen der Matrizen  $A$ . Plotten Sie analog zu Aufgabe 23 a) die hierfür benötigten Zeiten und zeichnen Sie falls nötig eine weitere Steigungsgerade ein.
- d) Interpretieren Sie Ihr Schaubild.

**Aufgabe 25** (*Eigenschaften orthogonaler Matrizen*)

(2T+2T+2T+2T Punkte)

Sei  $Q \in \mathbb{R}^{n \times n}$  eine orthogonale Matrix. Zeigen Sie:

- a)  $Q^\top$  ist eine orthogonale Matrix.
- b) Für alle  $x \in \mathbb{R}^n$  gilt:  $\|Qx\|_2 = \|x\|_2$ .
- c)  $\kappa_2(Q) = 1$ .
- d) Sei  $\tilde{Q}$  eine weitere orthogonale Matrix. Dann ist  $Q\tilde{Q}$  orthogonal.

**Aufgabe 26** (*Eigenschaften der Householder-Transformation*)

(2T+2T+2T+2T+2T Punkte)

Sei  $v = (v_1, \dots, v_n)^\top \in \mathbb{R}^n$ ,  $v \neq 0$  und  $Q_v = I - 2 \frac{vv^\top}{v^\top v}$  die Householder-Transformation. Zeigen Sie:

- a)  $Q_v = Q_v^\top$ .
- b)  $Q_v^2 = I$ .
- c) Für alle  $\alpha \in \mathbb{R}$ ,  $\alpha \neq 0$  gilt:  $Q_{\alpha v} = Q_v$ .
- d)  $Q_v y = y \iff y^\top v = 0$ .
- e)  $Q_v v = -v$ .

**Hinweise:**

Die Lösungen der Theorieaufgaben und Ihren Text zu den Programmieraufgaben können Sie in  $\text{\LaTeX}$  erstellen oder handschriftlich aufschreiben und einscannen. Die Programmieraufgaben sind in Matlab zu lösen. Der Source Code muss strukturiert und dokumentiert sein.

Falls Sie die Aufgaben im Team lösen, geben Sie bitte auf allen Lösungen alle an der Aufgabe beteiligten Teammitglieder an. Jedes Teammitglied muss eine Lösung abgeben.

Speichern Sie Ihre Lösungen und Ihre Ergebnisse in einem Directory mit dem Namen **Blatt06\_Vorname\_Nachname** und verpacken Sie dieses in eine `.zip`-Datei. Laden Sie **spätestens 48 Stunden vor Ihrem Tutorium** diese `.zip`-Datei in Moodle hoch.