Prof. Dr. Dirk Lebiedz
Dipl.-Math. oec. Klaus Stolle
Institut für Numerische Mathematik
Universität Ulm

Lösungsvorschlag Übungsblatt 06

Besprechung in den Tutorien in der Woche vom 08.06.2020 bis 12.06.2020

Für dieses Übungsblatt gibt es 16 Theorie- und 21 Matlab-Punkte, sowie 18 Theorie- und keine Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 70-Prozent-Grenzen liegen aktuell (inklusive Blatt 06) bei 107,8 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

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

existiert, Rekursionsformeln für $\ell_{j,j-1}$ $(j = 2, \ldots, n)$, $r_{j,j}$ $(j = 1, \ldots, n)$ und $r_{j,j+1}$ $(j = 1, \ldots, n-1)$ 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 $(\ell_{2,1}, \ldots, \ell_{n,n-1})$ der Matrix L, `rd` die Diagonale $(r_{1,1}, \ldots, r_{n,n})$ und `rnd` die obere Nebendiagonale $(r_{1,2}, \ldots, r_{n-1,n})$ der Matrix R, `aUnd` die untere Nebendiagonale $(a_{2,1}, \ldots, a_{n,n-1})$, `ad` die Diagonale $(a_{1,1}, \ldots, a_{n,n})$ und `aOnd` die obere Nebendiagonale $(a_{1,2}, \ldots, a_{n-1,n})$ der Matrix A.

d) Testen Sie Ihre MATLAB-Funktion `LR_TriDiag` an verschiedenen Beispielen.

Lösungsvorschlag:

a) Sei

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

eine Tridiagonalmatrix, für die die LR-Zerlegung

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

existiert.

Dann muss gelten

$$\begin{aligned}
&\begin{pmatrix} a_{1,1} & a_{1,2} & & & \\ a_{2,1} & a_{2,2} & a_{2,3} & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} \\ & & & a_{n,n-1} & a_{n,n} \end{pmatrix} \\
&\quad = A = L\,R \\
&\quad = \begin{pmatrix} 1 & & & & \\ \ell_{2,1} & 1 & & & \\ & \ddots & \ddots & & \\ & & \ell_{n-1,n-2} & 1 & \\ & & & \ell_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} r_{1,1} & r_{1,2} & & & \\ & r_{2,2} & r_{2,3} & & \\ & & \ddots & \ddots & \\ & & & r_{n-1,n-1} & r_{n-1,n} \\ & & & & r_{n,n} \end{pmatrix} \\
&\quad = \begin{pmatrix} r_{1,1} & r_{1,2} & & & \\ \ell_{2,1}r_{1,1} & \ell_{2,1}r_{1,2}+r_{2,2} & r_{2,3} & & \\ & \ddots & \ddots & \ddots & \\ & & \ell_{n-1,n-2}r_{n-2,n-2} & \ell_{n-1,n-2}r_{n-2,n-1}+r_{n-1,n-1} & r_{n-1,n} \\ & & & \ell_{n,n-1}r_{n-1,n-1} & \ell_{n,n-1}r_{n-1,n}+r_{n,n} \end{pmatrix}
\end{aligned}$$

Die Einträge von L und R lassen sich mittels Koeffizientenvergleich in der folgenden Reihenfolge berechnen:

$$\begin{aligned}
r_{1,1} &= a_{1,1} \\
r_{1,2} &= a_{1,2} \\
\ell_{2,1} &= \frac{a_{2,1}}{r_{1,1}} \\
r_{2,2} &= a_{2,2} - \ell_{2,1}r_{1,2} \\
r_{2,3} &= a_{2,3} \\
\ell_{3,2} &= \frac{a_{3,2}}{r_{2,2}} \\
r_{3,3} &= a_{3,3} - \ell_{3,2}r_{2,3} \\
&\vdots
\end{aligned}$$

$$
\begin{aligned}
r_{n-1,n} &= a_{n-1,n} \\
\ell_{n,n-1} &= \frac{a_{n,n-1}}{r_{n-1,n-1}} \\
r_{n,n} &= a_{n,n} - \ell_{n,n-1} r_{n-1,n}
\end{aligned}
$$

Damit ergibt sich für die Berechnung von $\ell_{j,j-1}\,(j = 2, \ldots, n)$, $r_{j,j}\,(j = 1, \ldots, n)$ und $r_{j,j+1}\,(j = 1, \ldots, n-1)$ der folgende Algorithmus

```
1: r_{1,1} = a_{1,1}
2: for j = 2, ..., n do
3:     r_{j-1,j} = a_{j-1,j}
4:     ℓ_{j,j-1} = a_{j,j-1} / r_{j-1,j-1}
5:     r_{j,j} = a_{j,j} - ℓ_{j,j-1} r_{j-1,j}
6: end for
```

Dieser Algorithmus entspricht dem „Thomas-Algorithmus zur Bestimmung der LR-Zerlegung“ von Folie 113.

b) Für Zuweisungen wird kein Aufwand berücksichtigt. Daher ergibt sich für den Aufwand des obigen Algorithmus: $(n-1)$ mal (wegen $j = 2, \ldots, n$) je eine Division, Subtraktion und Multiplikation.

   Werden bei der Berechnung des Aufwandes nur Multiplikationen und Divisionen berücksichtigt, so erhalten wir einen Gesamtaufwand von $2\,(n-1)$. Berücksichtigen wir auch Additionen, so ist der gesamte Aufwand $3\,(n-1)$. In $\mathcal{O}$-Notation ist der Aufwand in beiden Fällen $\mathcal{O}(n)$.

c) Siehe Moodle-Kurs im Abschnitt „Lösungsvorschläge“ unter „Blatt 06 Aufgabe 21 Matlab-Lösung“.

d) Siehe Moodle-Kurs im Abschnitt „Lösungsvorschläge“ unter „Blatt 06 Aufgabe 21 Matlab-Lösung“.

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 $n \in \{2^i \mid i = 3, \ldots, 14\}$ jeweils für die Matrix A den Vektor `ad` der Diagonalelemente, $ad = (2, 2, \ldots, 2)$, und den Vektor `nd` der Elemente der beiden Nebendiagonalen, $nd = (-1, -1, \ldots, -1)$ 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?

Lösungsvorschlag:

a) Siehe Moodle-Kurs im Abschnitt „Lösungsvorschläge“ unter „Blatt 06 Aufgabe 22 Matlab-Lösung“.

b) In Aufgabenteil a) haben wir das Schaubild der Abbildung 1 erhalten:

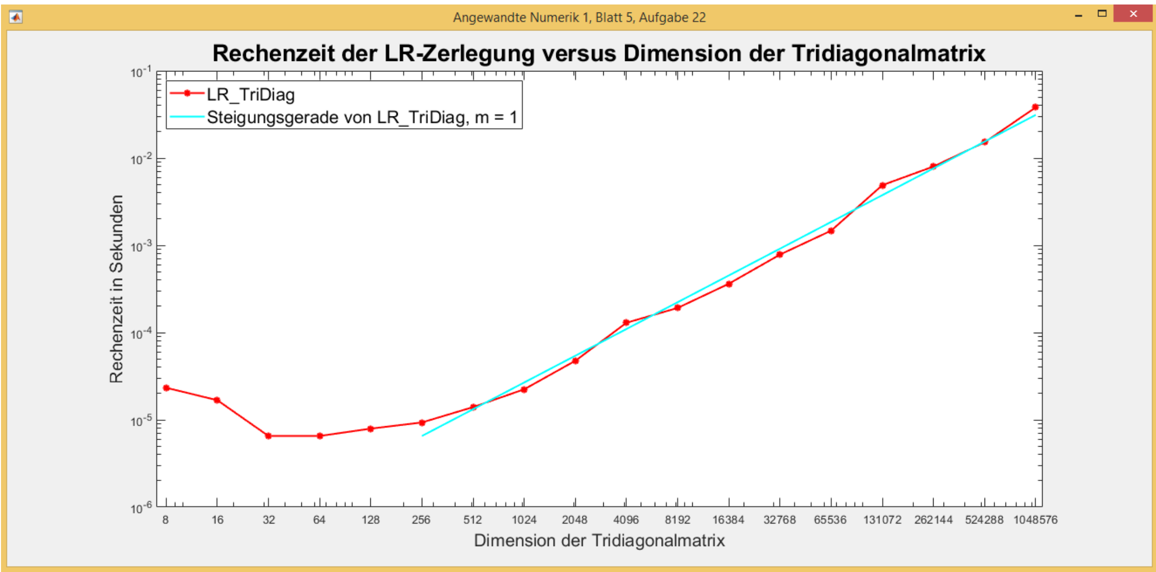

Abbildung 1: Laufzeiten von `LR_TriDiag` für Matrixdimensionen bis 2^{20}

   Die errechnete Steigungsgerade im doppelt logarithmischen Plot hat die Steigung $m = 1$. Das deckt sich mit unseren theoretischen Überlegungen aus Aufgabe 21, bei denen wir eine Aufwand von $\mathcal{O}(n)$ errechnet haben.

c) Siehe Moodle-Kurs im Abschnitt „Lösungsvorschläge“ unter „Blatt 06 Aufgabe 22 Matlab-Lösung“.

d) In Aufgabenteil c) haben wir das Schaubild der Abbildung 2 erhalten:

   Die verschiedenen Implementierungen der LR-Zerlegung für vollbesetzte Matrizen aus Aufgabe 13 haben im doppelt logarithmischen Plot alle eine Steigungsgerade mit einer Steigung von ungefähr $m = 3$ entsprechend einem Aufwand von $\mathcal{O}(n^3)$. Sie unterscheiden sich zwar in den Laufzeiten. Dieser Unterschied fällt aber für größere Matrizen im Vergleich zur Laufzeit mit Steigung ungefähr $m = 1$ des Tridiagonallösers aus der vorigen Aufgabe nicht ins Gewicht.

Aufgabe 23 *(Voraussetzungen für die Durchführbarkeit der Cholesky-Zerlegung)* (8T Punkte)
Berechnen Sie, falls möglich, für die folgenden Matrizen die Cholesky-Zerlegung per Hand:

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

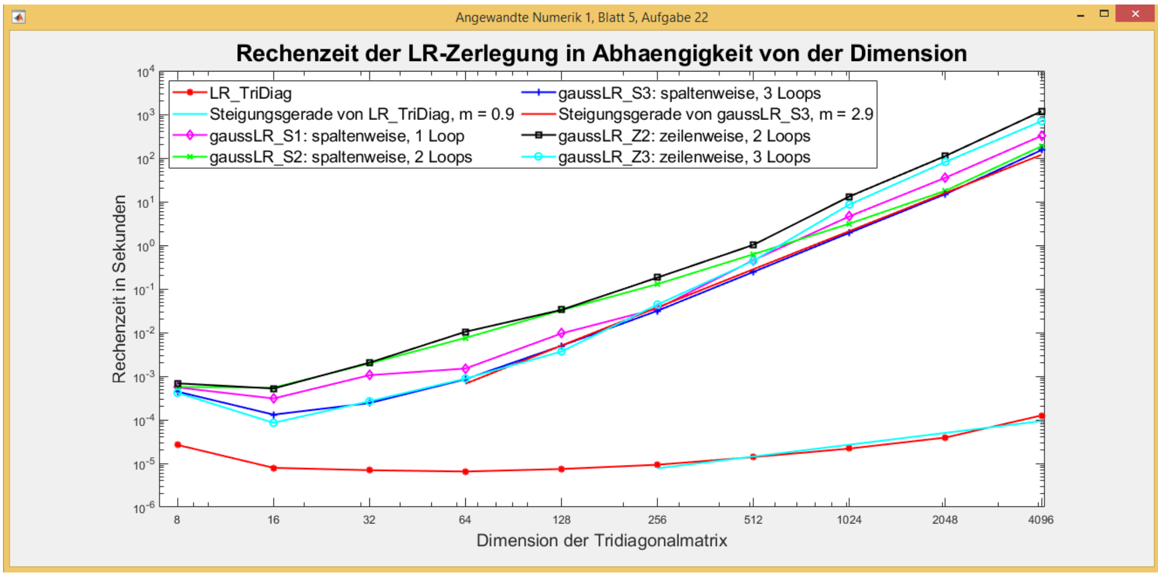

Abbildung 2: Laufzeiten verschiedener Implementierungen der LR-Zerlegung für vollbesetzte Matrizen und für Tridiagonalmatrizen für Matrixdimensionen bis 2^{12}

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.

Lösungsvorschlag:

A_1: Die Matrix A_1 ist nicht symmetrisch, da $a_{23} \neq a_{32}$.

A_2: Die Matrix A_2 ist nicht positiv definit.

Denn wäre A_2 positiv definit, müsste für alle $x \in \mathbb{R}^3$, $x \neq 0$ $x^T \cdot A_2 \cdot x > 0$ sein.

Wähle $x := \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$. Damit ist $x \neq 0$ und wir erhalten

$$x^T \cdot A_2 \cdot x = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \\ 1 & 1 & -2 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix} = -2 < 0.$$

Das ist ein Widerspruch zu $x^T \cdot A_2 \cdot x > 0$. Somit ist A_2 nicht positiv definit.

A_3: A_3 ist symmetrisch, da $a_{ij} = a_{ji}$ $\forall i, j = 1, 2, 3$ gilt.

Cholesky-Zerlegung über Zerlegung $A = LDL^T$ entsprechend der Folien 107 und 108:

$$\begin{array}{lllll}
j = 1: & (1,1)-\text{Element}: & d_{11} & = a_{11} = 1 & & \\
& (2,1)-\text{Element}: & l_{21}d_{11} & = 2 & \Longrightarrow & l_{21} = 2 \\
& (3,1)-\text{Element}: & l_{31}d_{11} & = 1 & \Longrightarrow & l_{31} = 1 \\
j = 2: & (2,2)-\text{Element}: & l_{21}^2 d_{11} + d_{22} & = a_{22} = 8 & \Longrightarrow & d_{22} = 8 - 4 \cdot 1 = 4 \\
& (3,2)-\text{Element}: & l_{31}d_{11}l_{21} + l_{32}d_{22} & = a_{32} = 2 & \Longrightarrow & l_{32} = (2 - 1 \cdot 1 \cdot 2)/4 = 0 \\
j = 3: & (3,3)-\text{Element}: & l_{31}^2 d_{11} + l_{32}^2 d_{22} + d_{33} & = a_{33} = 2 & \Longrightarrow & d_{33} = 2 - 1^2 - 0 \cdot 4 = 1
\end{array}$$

Man erhält

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \quad \text{und} \quad D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

L ist also eine linke untere normierte Dreiecksmatrix und D ist eine Diagonalmatrix mit positiven Diagonalelementen. Somit existiert die Cholesky-Zerlegung $A_3 = LDL^T$.

Als Probe erhalten wir

$$\begin{aligned} LDL^T &= \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 2 & 4 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 2 & 1 \\ 2 & 8 & 2 \\ 1 & 2 & 2 \end{pmatrix} = A_3 \end{aligned}$$

Aufgabe 24 *(Programmieraufgabe: Choleskyzerlegung)* (4M+2M+2M+1T Punkte)

a) Schreiben Sie eine MATLAB-Funktion `[L, D] = cholesky(A)`, die die Cholesky-Zerlegung $A = L\,D\,L^\top$ 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.

Lösungsvorschlag:

a) Siehe Moodle-Kurs im Abschnitt „Lösungsvorschläge“ unter „Blatt 06 Aufgabe 24 Matlab-Lösung“.

b) Siehe Moodle-Kurs im Abschnitt „Lösungsvorschläge“ unter „Blatt 06 Aufgabe 24 Matlab-Lösung“.

c) Siehe Moodle-Kurs im Abschnitt „Lösungsvorschläge“ unter „Blatt 06 Aufgabe 24 Matlab-Lösung“.

d) In Aufgabenteil c) haben wir das Schaubild der Abbildung 3 erhalten:

   Die Cholesky-Zerlegung benötigt im Vergleich zur LR-Zerlegung für vollbesetzte Matrizen etwas weniger Laufzeit. Sie hat aber im doppelt logarithmischen Plot ebenfalls eine Steigungsgerade mit einer Steigung von ungefähr $m = 3$ entsprechend einem Aufwand von $\mathcal{O}(n^3)$.

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: $\|Qx\|_2 = \|x\|_2$.

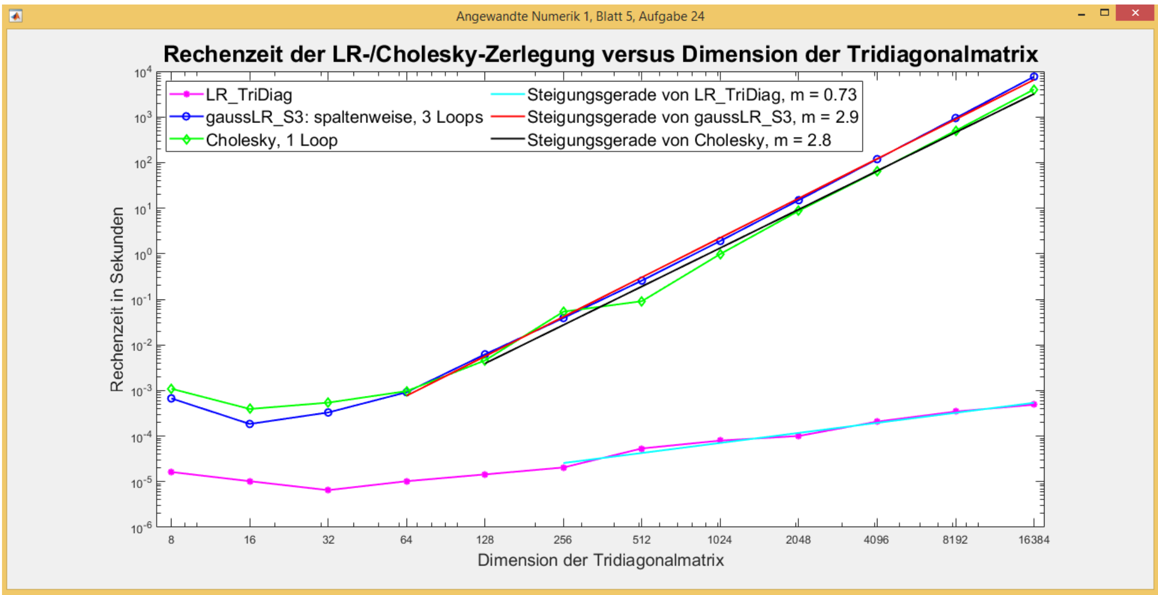

Abbildung 3: Laufzeiten der LR-Zerlegung und der Cholesky-Zerlegung für vollbesetzte Matrizen und der LR-Zerlegung für Tridiagonalmatrizen für Matrixdimensionen bis 2^{14}

c) $\kappa_2(Q) = 1$.

d) Sei $\tilde{Q}$ eine weitere orthogonale Matrix. Dann ist $Q\tilde{Q}$ orthogonal.

Lösungsvorschlag:

a) Aus der Orthogonalität von Q folgt $Q^T = Q^{-1}$ und damit $(Q^T)^T Q^T = QQ^T = QQ^{-1} = I$.

b) $||Qx||_2^2 = (Qx)^T Qx = x^T Q^T Qx = x^T Ix = x^T x = ||x||_2^2 \quad \Longrightarrow \quad ||Qx||_2 = ||x||_2$.

c) Aus b) folgt $||Q||_2 = \sup_{x \neq 0} \frac{||Qx||_2}{||x||_2} = \sup_{x \neq 0} \frac{||x||_2}{||x||_2} = 1$.

Da nach a) auch Q^T eine orthogonale Matrix ist, gilt auch $||Q^T||_2 = 1$ und somit $\kappa_2(Q) = ||Q||_2||Q^{-1}||_2 = ||Q||_2||Q^T||_2 = 1$.

d) $(\tilde{Q}Q)^T \tilde{Q}Q = Q^T \tilde{Q}^T \tilde{Q}Q = Q^T IQ = Q^T Q = I$.

Aufgabe 26 *(Eigenschaften der Householder-Transformation)* (2T*+2T*+2T*+2T*+2T* Punkte)

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

a) $Q_v = Q_v^T$.

b) $Q_v^2 = I$.

c) Für alle $\alpha \in \mathbb{R}$, $\alpha \neq 0$ gilt: $Q_{\alpha v} = Q_v$.

d) $Q_v y = y \quad \Longleftrightarrow \quad y^\top v = 0$.

e) $Q_v\, v = -v$.

Lösungsvorschlag:

a) Es gilt:

$$Q_v^T = (I - 2\frac{vv^T}{v^T v})^T = I^T - (2\frac{vv^T}{v^T v})^T = I - 2\frac{v^{T^T} v^T}{v^T v} = I - 2\frac{vv^T}{v^T v} = Q_v$$

b) Wegen $vv^Tvv^T = (v^Tv)(vv^T)$ gilt:

$$\begin{aligned} Q_v^2 =& I - 2\frac{vv^T}{v^Tv} - 2\frac{vv^T}{v^Tv} + 4\frac{vv^Tvv^T}{(v^Tv)^2} \\ =& I - 4\frac{vv^T}{v^Tv} + 4\frac{(v^Tv)(vv^T)}{(v^Tv)^2} = I \end{aligned}$$

c) Sei $\alpha \in \mathbb{R},\ \alpha \neq 0$. Dann gilt:

$$Q_{\alpha v} = I - 2\frac{\alpha v \alpha v^T}{\alpha v^T \alpha v} = I - 2\frac{\alpha^2(vv^T)}{\alpha^2(v^Tv)} = I - 2\frac{vv^T}{v^Tv} = Q_v$$

d) Wegen $vv^T \neq 0$ (also die Nullmatrix) gilt:

$$\begin{aligned} & Q_v y = (I - 2\frac{vv^T}{v^Tv})y = y - 2\frac{vv^T}{v^Tv}y = y \\ \iff\quad & 2\frac{vv^T}{v^Tv}y = 0 \\ \iff\quad & 2\frac{v}{v^Tv}y^Tv = 0 \\ \iff\quad & y^Tv = 0 \qquad \text{(da der Vektor } v \text{ ungleich dem Nullvektor ist)} \end{aligned}$$

e)

$$Q_v v = Iv - 2\frac{vv^Tv}{v^Tv} = v - 2v\frac{v^Tv}{v^Tv} = v - 2v = -v$$

Hinweise:
Die Lösungen der Theorieaufgaben und Ihren Text zu den Programmieraufgaben können Sie in 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.