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

WiSe 2023/2024
12.11.2023

Angewandte Numerik

Übungsblatt 04

Besprechung in den Tutorien von Montag, 10.11.2023 bis Freitag, 24.11.2023

Für dieses Übungsblatt gibt es 15 Theorie- und 21 Matlab-Punkte, sowie 9 Theorie- und 13 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 70-Prozent-Grenzen liegen aktuell (inklusive Blatt 04) bei 53,9 Theorie- und 49,7 Matlab-Punkten.

Aufgabe 9 *(Programmieraufgabe zweiter Teil: LR-Zerlegung mit Spalten-Pivotisierung)*
(4M+(2M+2M+2M*+4M*+2T*) Punkte)

e) Schreiben Sie eine MATLAB-Funktion `inversA = inversMat(A)`, die, ohne einschlägige MATLAB-interne Funktionen zu verwenden, mit den von Ihnen erstellten Funktionen effizient die Inverse der Matrix A berechnet.

   Geben Sie den Aufwand des von Ihnen verwendeten Algorithmus in $\mathcal{O}$-Notation an.

   Testen Sie Ihre MATLAB-Funktion `inversA = inversMat(A)` an verschiedenen Beispielen.

f) In dieser Teilaufgabe wollen wir die Laufzeiten verschiedener Implementierungen der LR-Zerlegung mit Spalten-Pivotisierung vergleichen.

   i) Vermutlich haben Sie Ihre MATLAB-Funktion `[A,p] = lrPivotZP(A)` aus Aufgabenteil a) so implementiert, dass die Matrix `A` zeilenweise durchlaufen wird. Falls nicht, ist diese Teilaufgabe entsprechend anzupassen, sodass Sie ebenfalls alle drei Varianten implementieren.

      Implementieren Sie zwei Varianten `[A,p] = lrPivotSP(A)` und `[A,p] = lrPivotVP(A)` Ihrer MATLAB-Funktion `[A,p] = lrPivotZP(A)` aus Aufgabenteil a), bei denen die Matrix `A` spaltenweise durchlaufen wird bzw. die beiden inneren Schleifen vektorisiert sind.

      Bei allen drei Varianten sollen die Zeilen der Matrix `A` nicht vertauscht, sondern für den Zugriff auf die Zeilen der Matrix `A` ein Permutationsvektor `p` verwendet werden.

   ii) Nun wollen wir drei weitere Varianten `[A,p] = lrPivotZT(A)`, `[A,p] = lrPivotST(A)` und `[A,p] = lrPivotVT(A)` der MATLAB-Funktion `[A,p] = lrPivotZP(A)` aus Aufgabenteil a) implementieren, bei denen wieder die Matrix `A` spaltenweise bzw. zeilenweise durchlaufen wird bzw. die beiden inneren Schleifen vektorisiert sind.

      Allerdings soll jetzt für den Zugriff auf die Zeilen der Matrix `A` kein Permutationsvektor verwendet werden, sondern die Zeilen der Matrix `A` sollen tatsächlich vertauscht werden.

   iii) Erweitern Sie Ihr MATLAB-Skript `testLrPivot`, sodass Sie auch die neuen fünf Varianten der LR-Zerlegung mit Spaltenpivotisierung an geeigneten Beispielen testen können.

iv) Schreiben Sie ein MATLAB-Skript `laufzeiten`, in dem Sie für $n \in \{2^i | i = 3, \dots, 12\}$ jeweils eine reguläre Matrix $A \in \mathbb{R}^{n \times n}$ aufstellen. Berechnen Sie mit Ihren sechs verschiedenenen Implementierungen jeweils eine LR-Zerlegung mit Spaltenpivotisierung der Matrix A.

Plotten Sie die zur Berechnung benötigten Zeiten über die Dimension `n` der Matrix 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 längere Zeit brauchen wird.

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

Aufgabe 10 *(Voraussetzungen für die Durchführbarkeit der Cholesky-Zerlegung)* (8T Punkte)

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

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

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 11 *(Programmieraufgabe: Choleskyzerlegung)* (4M+2M+3M Punkte)

a) Schreiben Sie eine MATLAB-Funktion `L = cholesky(A)`, die die Cholesky-Zerlegung $A = L\,L^\top$ einer symmetrisch positiv definiten Matrix `A` nach dem Algorithmus von Folie 43 (Skript Seite 40) berechnet. Dabei sind `L` die linke untere Dreiecksmatrix 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 10 und weiteren Beispielen.

c) Vektorisieren Sie Ihre MATLAB-Funktion `L = cholesky(A)`, sodass sie nur noch die äußere Schleife (über k) enthält. Bezeichnen Sie Ihre vektorisierte MATLAB-Funktion mit `choleskyV`.

Testen Sie Ihre vektorisierte MATLAB-Funktion `choleskyV` entsprechend Aufgabenteil b).

Aufgabe 12 *(Cholesky-Zerlegung durch Koeffizientenvergleich)* (5T+2T*+1T*+2T*+4M* Punkte)

a) Berechnen Sie durch Koeffizientenvergleich die Cholesky-Zerlegung der Matrix

$$A = \begin{pmatrix} 1 & -1 & 1 & 0 \\ -1 & 2 & 0 & 1 \\ 1 & 0 & 6 & -3 \\ 0 & 1 & -3 & 30 \end{pmatrix}.$$

Gesucht ist also eine linke untere Dreiecksmatrix

$$L = \begin{pmatrix} l_{1,1} & 0 & 0 & 0 \\ l_{2,1} & l_{2,2} & 0 & 0 \\ l_{3,1} & l_{3,2} & l_{3,3} & 0 \\ l_{4,1} & l_{4,2} & l_{4,3} & l_{4,4} \end{pmatrix},$$

so dass $A = L\,L^T$.

Bestimmen Sie die Koeffizienten der linken unteren Dreiecksmatrix L sukzessive durch Vergleich der Koeffizienten der beiden Matrizen A und $L * L^\top$. Verwenden Sie bei allen Rechnungen ausschließlich Brüche und keine Dezimalzahlen und geben Sie alle Zwischenschritte an.

b) Handelt es sich bei der Matrix A um eine Bandmatrix? Falls ja: Welche Bandbreite hat A?

c) Ist die Cholesky-Zerlegung einer symmetrischen positiv definiten Matrix eindeutig? Begründen Sie Ihre Aussage.

d) Unterscheidet sich die Reihenfolge, in der Sie die einzelnen Koeffizienten berechnet haben, von der Reihenfolge des in der Vorlesung auf Folie 43 vorgestellten und auf Seite 40 des Skripts angegebenen Algorithmus für das Cholesky-Verfahren?

Falls ja: Worin? Falls nein: Können die Koeffizienten auch in einer anderen Reihenfolge berechnet werden? In welcher?

e) Implementieren Sie eine MATLAB-Funktion `L = choleskyZ(A)`, die die Koeffizienten einer Cholesky-Zerlegung $A = L\,L^\top$ einer symmetrisch positiv definiten Matrix `A` in einer zum Algorithmus von Folie 43 (Skript Seite 40) alternativen Reihenfolge berechnet.

Testen Sie Ihre MATLAB-Funktion `choleskyZ` entsprechend Aufgabe 11 b).

Aufgabe 13 *(Programmieraufgabe: Vergleich der Rechenzeiten)* (4M+2T+(3M*+2T*) Punkte)

In dieser Aufgabe wollen wir die Laufzeiten der verschiedenen Implementierungen der Cholesky-Zerlegung aus den vorigen Aufgaben 11 und 12 untereinander sowie mit den Laufzeiten der LR-Zerlegung aus Aufgabe 9 vergleichen.
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 analog zu Ihrem MATLAB-Skript `laufzeiten` aus Aufgabe 9 ein MATLAB-Skript `laufzeitenA13`, in dem Sie für $n \in \{2^i \,|\, i = 3, \ldots, 12\}$ jeweils die Matrix A aufstellen. Berechnen Sie mit Ihren MATLAB-Funktionen `cholesky` und `choleskyV` aus Aufgabe 11 sowie `choleskyZ` aus Aufgabe 12 jeweils eine Cholesky-Zerlegung der Matrix A.

   Plotten Sie die zur Berechnung der Cholesky-Zerlegungen benötigten Zeiten über die Dimension `n` der Matrix in einem Schaubild mit logarithmischen Achsen. Zeichnen Sie in Ihr Schaubild eine geeignete Steigungsgerade ein.

   Beachten Sie, dass Ihr MATLAB-Skript je nach Leistungsfähigkeit Ihres Computers zur Berechnung längere Zeit brauchen wird.

   Hinweis: Die MATLAB-Funktion `diag` könnte hilfreich sein.

b) Interpretieren Sie Ihr Schaubild.

c) Integrieren Sie auch einige Ihrer in Aufgabe 9 implementierten MATLAB-Funktionen zur Berechnung einer LR-Zerlegung mit Spaltenpivotisierung der Matrix A in Ihr Schaubild.

   Treffen Sie dabei eine sinnvolle Auswahl, in der zumindest Ihre schnellste Implementierung aus Aufgabe 9 enthalten ist.

   Haben Sie diese Ergebnisse erwartet? Warum?

Hinweise:

Teamarbeit ist bei der Bearbeitung der Übungsblätter erlaubt und erwünscht. Falls Sie die Aufgaben im Team lösen, muss jedes Teammitglied eine eigene Lösung votieren und abgeben. Geben Sie dann bitte auf Ihrer Lösung alle an der Aufgabe beteiligten Teammitglieder an.

Die Programmieraufgaben sind in MATLAB zu lösen. Der Source Code muss strukturiert und dokumentiert sein.

Speichern Sie Ihre Lösungen der Theorieaufgaben, Ihre MATLAB-Programme und Ihre Ergebnisse in einem Directory mit dem Namen **Blatt04_*Vorname_Nachname*** und verpacken Sie dieses in eine `.zip`-Datei. **Spätestens bis Sonntag, 19.11.2023, 23:59 Uhr** müssen Sie diese `.zip`-Datei in Moodle hochladen und für die Aufgabenteile, die Sie vorstellen können, votieren.

Aufgabe 10

$$A_1 = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 1 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$ -> nicht symmetrisch, daher existiert keine Cholesky-Zerlegung

$$A_2 = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 8 & 1 \\ 1 & 1 & -8 \end{pmatrix}$$

$\det(A - I\lambda) = -\lambda^3 + \lambda^2 + 68\lambda - 54$

$\Rightarrow \underline{\lambda_0 = -8{,}153}$; $\lambda_1 = 0{,}792$; $\lambda_3 = 8{,}361$

-> keine Cholesky-Zerlegung möglich, da A nicht positiv definit (alle Eigenwerte $\lambda > 0$)

$$A_3 = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 8 & 2 \\ 1 & 2 & 2 \end{pmatrix} = \begin{pmatrix} l_{11} & & 0 \\ l_{21} & l_{22} & \\ l_{31} & l_{32} & l_{33} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & l_{31} \\ & l_{22} & l_{32} \\ 0 & & l_{33} \end{pmatrix}$$

$$= \begin{pmatrix} l_{11} & l_{11}l_{21} & l_{11}l_{31} \\ l_{11}l_{21} & l_{21}^2 + l_{22}^2 & l_{21}l_{31} + l_{22}l_{32} \\ l_{11}l_{31} & l_{31}l_{21} + l_{32}l_{22} & l_{31}^2 + l_{32}^2 + l_{33}^2 \end{pmatrix} = \begin{pmatrix} 1 & & 0 \\ 2 & 2 & \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \\ & 2 & 0 \\ 0 & & 1 \end{pmatrix}$$

$$l_{11} = \sqrt{1^2} = 1 \qquad l_{22} = \sqrt{8 - l_{21}^2} = \sqrt{8 - 2^2} = \sqrt{4} = 2$$

$$l_{21} = 2 : 1 = 2 \qquad l_{32} = (2 - l_{31}l_{21}) : l_{22} = (2 - 2 \cdot 1) : 2 = 0 : 2 = 0$$

$$l_{31} = 1 : 1 = 1 \qquad l_{33} = \sqrt{2 - l_{31}^2 - l_{32}^2} = \sqrt{2 - 1^2 - 0^2} = 1$$

Aufgabe 12

a)

$$A = \begin{pmatrix} 1 & -1 & 1 & 0 \\ -1 & 2 & 0 & 1 \\ 1 & 0 & 6 & -3 \\ 0 & 1 & -3 & 30 \end{pmatrix}$$

$$A = LL^T$$

$$L = \begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & 0 \\ l_{31} & l_{32} & l_{33} & \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix}$$

$$LL^T = \begin{pmatrix} l_{11}^2 & & & \\ l_{11}l_{21} & l_{21}^2 + l_{22}^2 & & \\ l_{11}l_{31} & l_{21}l_{31} + l_{22}l_{32} & l_{31}^2 + l_{32}^2 + l_{33}^2 & \\ l_{11}\,{}_{41} & l_{21}l_{41} + l_{22}l_{42} & l_{31}l_{41} + l_{32}l_{42} + l_{33}l_{43} & l_{41}^2 + l_{42}^2 + l_{43}^2 + l_{44}^2 \end{pmatrix}$$

$l_{11} = \sqrt{1} = 1$

$l_{21} = {}^{-1}/_{1} = -1$

$l_{22} = \sqrt{2-1} = 1$

$l_{31} = {}^{1}/_{1} = 1$

$l_{32} = 0 - (-1) : 1 = 1$

$l_{33} = \sqrt{6-1-1} = \sqrt{4} = 2$

$l_{41} = {}^{0}/_{1} = 0$

$l_{42} = {}^{1-(0)}/_{1} = 1$

$l_{43} = (-3-1) : 2 = -4 : 2 = -2$

$l_{44} = \sqrt{30-1-4} = \sqrt{25} = 5$

$$\Rightarrow \quad L = \begin{pmatrix} 1 & & & 0 \\ -1 & 1 & & \\ 1 & 1 & 2 & \\ 0 & 1 & -2 & 5 \end{pmatrix}$$

b) die Bandbreite beträgt $m = 2$

c) Nein, die Cholesky Zerlegung ist nicht eindeutig, da das Vorzeichen der Diagonalelemente nicht exakt bestimmt wird $\Rightarrow LDL^T = \bar{L}\bar{L}^T$

$\rightarrow \bar{L} = LD^{1/2}$, $D^{1/2} = \mathrm{diag}(\pm\sqrt{d_i})$

Dadurch können mehrere Varianten entstehen

d) Hier wurden die Elemente zeilenweise bestimmt.
Bei dieser Art wird zuerst in der k-ten Zeile $l_{k,1}$ errechnet und aus diesem Ergebnis lassen sich dann nacheinander die Werte $l_{k,2}$ bis $l_{k,k}$ bestimmen.

Aufgabe 13

b) Die vektorisierte Cholesky Zerlegung arbeitet am schnellsten und für größere n wird die Zeilenbasierte Cholesky Zerlegung langsamer als die Spaltenbasierte.
Dies liegt vermutlich an der internen Speicherung der Matrizen von Matlab.

c) Wie erwartet arbeitet die schnellste LR-Zerlegung deutlich langsamer als alle Varianten der Cholesky-Zerlegung.