

**Lösung 4 zur Vorlesung Numerische Mathematik
Sommersemester 2020**

Abgabe bis 15. Mai, 23:59

Zerlegungsmethoden zur Eigenwertbestimmung Man lese Abschnitt 5.4 *Zerlegungsverfahren...* im Buch, zunächst bis einschließlich *Beispiel 5.20 Cholesky-Verfahren*. In diesem Abschnitt ist Hilfsatz 5.17 wesentlich für den späteren Beweis von Satz 5.19 (Konvergenz der Cholesky-Methode). Der Beweis von Satz 5.17 (insbesondere (ii)) ist weder besonders schön und es ist sinnvoller, den Beweis zu Satz 5.19 näher zu studieren. Man gebe kurze Antworten auf die folgenden Fragen:

1. Was ist der grundsätzliche Vorteil der Cholesky-Iteration im Vergleich zur Potenzmethode oder zur inversen Iteration?
-

Die Cholesky-Iteration liefert direkt alle Eigenwerte der Matrix. Die Potenzmethode gibt immer nur den betragsgrößten, die Inverse Iteration den kleinsten, bzw. bei Verwendung eines Shifts einen speziellen, der dem Shift am nächsten ist.

2. Man gebe die Iterationsvorschrift der Cholesky-Iteration an.
-

Ausgehen von $A^{(0)} := A$ iteriert man die folgenden beiden Schritte für $i = 1, 2, \dots$

$$\text{a) } L^{(i)}L^{(i),T} := A^{(i-1)}, \quad \text{b) } A^{(i)} := L^{(i),T}L^{(i)}$$

3. Unter welchen Bedingungen an die Matrix $A \in \mathbb{R}^{n \times n}$ konvergiert das Cholesky-Verfahren?
-

Die Cholesky-Iteration konvergiert, wenn die Matrix A symmetrisch positiv definit ist.

4. Was ist der Aufwand pro Iteration für das Cholesky-Verfahren (in Bezug auf die Matrixgröße n)?
-

Die Erstellung der Cholesky-Zerlegung mit der direkten Methode benötigt

$$\frac{n^3}{6} + \mathcal{O}(n^2)$$

Iterationen. Im Anschluss muss das Produkt zweier Dreiecksmatrizen erstellt werden. Wir wissen, dass das Ergebnis LL^T symmetrisch ist. Daher muss nur die Hälfte der Elemente berechnet werden, d.h:

$$(LL^T)_{ij} = \sum_k L_{ik}L_{jk} \text{ für } i = 1, \dots, n \text{ und } j = 1, \dots, i$$

Also ist immer $j \leq i$ und die Summe kürzt sich zu

$$(LL^T)_{ij} = \sum_{k=1}^j L_{ik}L_{jk} \text{ für } i = 1, \dots, n \text{ und } j = 1, \dots, i,$$

da $L_{jk} = 0$ für $k > j$. Der Aufwand ist somit (in Multiplikationen) noch einmal

$$\sum_{i=1}^n \sum_{j=1}^i j = \sum_{i=1}^n \frac{i(i+1)}{2} = \frac{1}{2} \sum_{i=1}^n i + i^2 = \frac{\frac{n(n+1)}{2} + \frac{n(n+1)(2n+1)}{6}}{2} = \frac{n^3}{6} + \mathcal{O}(n^2).$$

Ein Schritt hat demnach den Aufwand $n^3/3 + \mathcal{O}(n^2)$.

Die QR-Methode Man lese bis zum Ende des Abschnittes und gebe kurze Antworten:

1. Was ist der Vorteil der QR-Iteration gegenüber der Cholesky-Methode?
-

Die Anwendbarkeit auf beliebige (also auch nicht positiv definite) Matrizen.

2. Was bedeutet "QR-Iteration mit Shift"?
-

In jedem Schritt wird nicht einfach die QR-Zerlegung der Matrix $A^{(i)}$ erstellt sondern von $A^{(i)} - \mu^{(i)}I$ mit dem Shift $\mu^{(i)}$. Bei guter Wahl kann die Konvergenz so wesentlich beschleunigt werden. Optimal gewählt konvergiert die QR-Iteration (bei Matrizen mit separierten Eigenwerten) mit dritter Ordnung!

3. Was ist der Aufwand für einen Schritt der QR-Methode, wenn die QR-Zerlegung mit Householder-Transformationen berechnet wird?
-

Der Aufwand für die Multiplikation RQ ist $n^3/2 + (n^2)$. Kann wie oben berechnet werden. R ist eine Dreiecksmatrix, daher reduziert sich der Aufwand im Vergleich zur vollen Matrix-Matrix Multiplikation leicht.

Die QR-Zerlegung kann mit Householder-Transformationen in $2/3n^3 + \mathcal{O}(n^2)$ Operationen erstellt werden. Siehe Satz 4.15. Insgesamt ergibt sich damit $7/6n^3 + \mathcal{O}(n^2)$ also etwa 3.5 mal mehr Aufwand als bei der Cholesky-Iteration. Dafür ist das Verfahren universeller einsetzbar.

Üblicherweise wird die Matrix A zunächst auf *Hessenbergform* reduziert. Im Anschluss kann die QR-Iteration viel schneller durchgeführt werden. Dies ist dann das leistungsfähigste Verfahren zur Bestimmung von Eigenwerten.

Reduktionsmethoden Man lese (sehr grob) Abschnitt 5.3 *Reduktionsmethoden*. Hier geht es um eine Transformation der Matrix A auf eine Gestalt, bei der sich nachfolgend die Eigenwerte einfacher bestimmen lassen. Die Transformation soll dabei eine Ähnlichkeitstransformation sein, also

$$A \rightarrow \tilde{A} := S^{-1}AS$$

damit A und \tilde{A} die gleichen Eigenwerte haben. Optimal wäre eine Transformation auf Dreiecksgestalt oder Diagonalgestalt, dann könnten wir die Eigenwerte ablesen. Dies gelingt im allgemeinen jedoch nicht. Stattdessen wird die Hessenberg-Form eingeführt: eine Dreiecksmatrix mit einer zusätzlichen Zeile neben der Diagonalen. Es zeigt sich, dass die Hessenberg-Form mit Hilfe von Householder-Transformationen erzeugt werden kann. Im Anschluss kann die QR-Methode sehr effizient zur Berechnung der Eigenwerte durchgeführt werden. Man gebe kurz Antworten

1. Man skizziere die Hessenberg-Form einer Matrix.

Dies ist eine Dreiecksmatrix (links unten oder rechts oben ist egal). Zusätzlich ist eine Nebendiagonale der Matrix mit Einträgen belegt.

2. Was ist der Aufwand zur Berechnung der QR-Zerlegung einer $n \times n$ Hessenberg Matrix?

Hat A Hessenbergform so kann die QR-Zerlegung in $2n^2 + \mathcal{O}(n)$ Operationen durchgeführt werden. Der Grund liegt in der Form der Spiegelungsvektoren v . Üblicherweise hat dieser Vektor im i -ten Schritt die Länge $n - i$. Bei Hessenberg-Matrizen werden immer nur 2 von Null verschiedene Einträge verwendet.

3. Was ist der grundlegende Unterschied bei der Konstruktion der Spiegelungsvektoren bei der Erstellung der QR-Zerlegung und der Erstellung der Hessenberg-Form?

Es reicht, den ersten Schritt zu betrachten. Bei Transformation zu einer Dreiecksmatrix wird v so gewählt, dass die Matrix $S = I - 2vv^T$ den ersten Spaltenvektor auf ein Vielfaches des ersten Einheitsvektors abbildet. Bei Transformation auf Hessenberg-Form wählt man v nur so, dass $S = I - 2vv^T$ auf in den Raum $\text{span}(e_1, e_2)$ abbildet. Der Vektor v wird dann so gewählt, dass der erste Eintrag 0 ist. Dies

sorgt dafür, dass S in der ersten Zeile und Spalte (bis auf die Diagonale) nur Null-Einträge hat. Hieraus folgt, dass die Multiplikation von rechts mit S die erste Spalte nicht ändert. Die durch Multiplikation mit S von links erreichte Form bleibt also erhalten.

Übungsaufgaben

Aufgabe 4.1

Im Skript wird in Satz 5.23 die QR-Iteration mit Shift vorgestellt. Man zeige, dass auch die Cholesky-Iteration mit Shift eine Folge von ähnlichen Matrizen erzeugt. D.h. Mit $A^{(0)} := A$ und $\mu \in \mathbb{R}$ iteriere man für $l = 1, 2,$

1. Erstelle die Cholesky-Zerlegung

$$L^{(l)}[L^{(l)}]^T := A^{(l-1)} - \mu I$$

2. Setze

$$A^{(l)} := [L^{(l)}]^T L^{(l)} + \mu I$$

Es ist sei $LL^T = A - \mu I$. Dann ist

$$\tilde{A} := L^T L + \mu I = L^{-1}(LL^T)L + \mu I = L^{-1}(A - \mu I)L + \mu I = L^{-1}AL$$

also sind A und \tilde{A} ähnlich.

Programmieraufgabe 4.2

a) Man implementiere die Cholesky-Zerlegung und verwende hierfür das direkte Verfahren, Algorithmus 3.5 im Buch. Für die angegebenen Test-Instanzen teste man das Verfahren. Die korrekte Lösung ist in `template_04.py` vorgegeben. Man darf ausnutzen, dass alle Matrizen tridiagonal sind, d.h. $A_{ij} = 0$ für $|i - j| > 2$.

b) Man implementiere die Cholesky-Iteration zur Approximation aller Eigenwerte der Matrix. Man breche die Iteration ab, wenn die maximale Abweichung der Diagonalelemente zweier Iterationen kleiner als 10^{-3} ist, d.h. falls

$$\max_{l=1, \dots, n} |A_{ll}^{(k+1)} - A_{ll}^{(k)}| < 10^{-3}$$

Für $n = 2, 4, 8, 16, 32, \dots$ teste man durch Zeitmessung den Gesamtaufwand zum Erreichen der Toleranz. Mit welcher Ordnung (in n) wächst der Aufwand?

c) (**Zusatz**) Man versuche eine effiziente Implementierung von Cholesky-Zerlegung und Cholesky-Iteration und nutze aus, dass nur Tridiagonalmatrizen auftauchen, also Bandmatrizen mit Bandbreite 1. Wie wächst nun der Aufwand in Bezug auf n ?

Abgabe per Mail an Henry von Wahl (henry.vonwahl@ovgu.de). Abgabe der Kurzfragen in 2er Gruppen, der Übungsaufgaben in den 5er Gruppen. Die theoretische Gruppenaufgabe ist freiwillig.