



Algorithmische Mathematik I

Wintersemester 19/20
Prof. Dr. J. Gedicke
Johannes Rentrop und Jannik Schürg



Übungsblatt 11.

Abgabedatum: 13.01.2020–17.01.2020

Programmieraufgabe 1. (LR-Zerlegung)

- Schreiben Sie eine Funktion, die für eine gegebene reelle Matrix $A \in \mathbb{R}^{n \times n}$ die LR-Zerlegung $A = LR$ mit einer unteren Dreiecksmatrix L und einer oberen Dreiecksmatrix R ohne Pivotisierung berechnet (Der Fall einer Division durch Null muss also abgefangen werden). Die Funktion soll *in-place* arbeiten, d.h. die ursprüngliche Matrix mit der LR-Zerlegung überschreiben. Die Matrix soll als eindimensionales Array der Größe n^2 gespeichert werden.
- Schreiben Sie eine weitere Funktion, die für gegebene L und R ein Gleichungssystem der Form $Ax = b$ mit $x, b \in \mathbb{R}^n$ für beliebige b durch Vorwärts- und Rückwärtseinsetzen mittels $Ly = b$ und $Rx = y$ löst.
- In diesem Aufgabenteil soll motiviert werden, dass man in der Praxis immer auf fertige Bibliotheken zurückgreifen sollte, anstatt grundlegende Verfahren wie die LR-Zerlegung selbst zu implementieren. Die populärste Bibliothek für Routinen aus der linearen Algebra heißt LAPACK¹. Die Bedienung ist zwar etwas umständlich, dafür stehen hochoptimierte Implementierungen für alle gängigen Betriebssysteme und Architekturen bereit. Machen Sie sich für die Verwendung in C/C++ mit dem Wrapper LAPACKE² vertraut. Denken Sie daran, mit dem Flag `-llapacke` zu kompilieren.

Schreiben Sie nun ein Programm, das eine Matrix A und einen Vektor b mit zufälligen Einträgen (in double precision) erstellt und jeweils eine Kopie speichert. Die Dimensionen sollten ausreichend hoch gewählt werden, sodass folgende Berechnungen mindestens einige Minuten Laufzeit in Anspruch nehmen. Es sollen nun die LR-Zerlegung, sowie die Lösung des zugehörigen Gleichungssystems mit den Funktionen aus a) und b) berechnet und jeweils die Laufzeiten ausgegeben werden. Zusätzlich sollen beide Rechenschritte mit den analogen LAPACK-Routinen³ wiederholt und die Laufzeit sowie numerische Abweichungen verglichen werden. Was beobachten Sie?

(12 Punkte)

¹<http://www.netlib.org/lapack/>

²<http://www.netlib.org/lapack/lapacke.html>

³Suchen Sie in der Dokumentation nach `dgetrf()` und `dgetrs()`

Programmieraufgabe 2. (Cholesky-Zerlegung)

Implementieren Sie eine Funktion, die die Cholesky-Zerlegung $A = LL^T$ mit einer unteren Dreiecksmatrix L mit $l_{ii} > 0$, $i = 1, \dots, n$ für eine reelle symmetrische positiv definite Matrix $A \in \mathbb{R}^{n \times n}$ berechnet. Auch diese Funktion soll *in-place* arbeiten. Achten Sie dabei auf die Reihenfolge, in der die einzelnen Einträge von L berechnet werden.

Erstellen Sie eine zufällige symmetrische positiv definite Matrix von derselben Größenordnung wie in Aufgabe 1. Dazu können Sie direkt eine untere Dreiecksmatrix L mit der Bedingung $l_{ii} > \sum_{j=1}^{i-1} |l_{ij}|$ erzeugen, die zugleich die Cholesky-Darstellung darstellt und somit zur Überprüfung dienen kann, und $A = LL^T$ verwenden. Berechnen Sie nun die Cholesky-Zerlegung einmal mit Ihrer Funktion und einmal mit der entsprechenden LAPACK-Routine⁴ und vergleichen Sie auch hier die Laufzeit.

(8 Punkte)

⁴Verwenden Sie `dpotrf()`