

Programmierpraktikum Nr. 2 zur Vorlesung Einführung in die Numerik, Sommer 2019

Aufgabe 2.1: LR-Zerlegung für Bandmatrizen

Eine Bandmatrix der Bandbreite $m < n$ ist eine Matrix $A \in \mathbb{R}^{n \times n}$, für die gilt:

$$a_{ij} = 0 \quad \text{für} \quad |i - j| > m.$$

Bandmatrizen haben also die spezielle Form

$$\begin{pmatrix} a_{11} & \cdots & a_{1(m+1)} & 0 & \cdots & 0 \\ \vdots & \ddots & & \ddots & & \vdots \\ a_{(m+1)1} & & & & & \\ 0 & \ddots & & & & \\ \vdots & & & & \ddots & 0 \\ & & & & & a_{(n-m)n} \\ & & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{n(n-m)} & \cdots & a_{nn} \end{pmatrix}$$

- (a) Implementieren Sie die LR-Zerlegung (ohne Pivotierung) für Bandmatrizen. Nutzen Sie dabei die spezielle Struktur der Bandmatrizen, um Speicherplatz und Operationen einzusparen. Schreiben Sie das Programm so, dass man eine beliebige Matrix A übergeben kann und die LR-Zerlegung zurückbekommt.
- (b) Berechnen Sie die LR-Zerlegung der $(n \times n)$ -Matrix

$$A_n = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$$

für $n = 4, 7, 10$ mithilfe des Programmes aus (a). Vergleichen Sie die Struktur der Matrizen A , L und R mit der Struktur der Inversen von A , die Sie z.B. mit der Python-Funktion `numpy.linalg.inv(A)` berechnen können. Begründen Sie Ihre Beobachtungen.

- (c) Schreiben Sie ein weiteres Programm, das durch Vorwärts- und Rückwärtseinsetzen ein lineares Gleichungssystem $Ax = b$ mithilfe der vorher berechneten LR-Zerlegung der Matrix A löst.
- (d) Lösen Sie das lineare Gleichungssystem

$$A_n x = b_n, \quad \text{wobei} \quad b_n = (1, \dots, 1)^T \in \mathbb{R}^n,$$

für $n = 4, 7, 10$ mit Hilfe der Programme aus (a) und (c).

- (e) Überlegen Sie sich Strategien, wie Sie Ihren Code auf Richtigkeit überprüfen können. Verifizieren Sie anschließend Ihren Code.

Zusatz: Finden Sie heraus, wie groß Sie n in Ihrem Programm maximal wählen können. Was sind limitierende Faktoren? Haben Sie Lösungsvorschläge?