

Übung 1 Schurkomplement

Sei $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$. Gegeben sei eine Blockpartitionierung einer Matrix $A \in \mathbb{K}^{n \times n}$ der Form

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

wobei die Teilmatrix $A_{11} \in \mathbb{K}^{p \times p}$ ($1 \leq p \leq n$) invertierbar sei. Für die anderen Teilmatrizen gilt $A_{12} \in \mathbb{K}^{p \times (n-p)}$, $A_{21} \in \mathbb{K}^{(n-p) \times p}$ und $A_{22} \in \mathbb{K}^{(n-p) \times (n-p)}$. Die Block LR-Zerlegung von A lässt sich dann darstellen als

$$A = \begin{pmatrix} Id & 0 \\ A_{21}A_{11}^{-1} & Id \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ 0 & S \end{pmatrix}$$

mit entsprechend dimensionierten Einheitsmatrizen Id . Die Teilmatrix $S \in \mathbb{K}^{(n-p) \times (n-p)}$ wird Schurkomplement von A_{11} in A genannt.

a) Zeigen Sie

$$S = A_{22} - A_{21}A_{11}^{-1}A_{12}.$$

b) Zeigen Sie: Wenn A hermitesch positiv definit ist, dann sind auch A_{11} und S hermitesch positiv definit.

(3 Punkte)

Übung 2 LR-Zerlegung tridiagonaler Matrizen

Gegeben sei eine Tridiagonalmatrix

$$A = \begin{pmatrix} a_1 & b_1 & & 0 \\ c_2 & a_2 & b_2 & \\ & \ddots & \ddots & \ddots \\ & & \ddots & \ddots & b_{n-1} \\ 0 & & & c_n & a_n \end{pmatrix}$$

mit

$$\begin{aligned} |a_1| &> |b_1| > 0 \\ |a_j| &\geq |b_j| + |c_j| > 0, \quad b_j, c_j \neq 0, \quad j \in \{2, \dots, n-1\} \\ |a_n| &\geq |c_n| > 0. \end{aligned}$$

Zeigen Sie: Der Algorithmus

$$\begin{aligned} r_1 &:= a_1 \\ l_j &:= c_j / r_{j-1} \quad j \in \{2, \dots, n\} \\ r_j &:= a_j - l_j b_{j-1} \quad j \in \{2, \dots, n\} \end{aligned}$$

ist durchführbar (d.h. $r_1, \dots, r_n \neq 0$) und liefert die LR-Zerlegung

$$A = \begin{pmatrix} 1 & & & 0 \\ l_2 & 1 & & \\ & \ddots & \ddots & \\ 0 & & l_n & 1 \end{pmatrix} \begin{pmatrix} r_1 & b_1 & & 0 \\ & r_2 & \ddots & \\ & & \ddots & b_{n-1} \\ 0 & & & r_n \end{pmatrix}$$

Zeigen Sie, dass $\det(A) \neq 0$ gilt (woraus die Invertierbarkeit von A folgt).

(5 Punkte)

Übung 3 LR-Zerlegung im Besonderen

Sei $A \in \mathbb{R}^{n \times n}$ gegeben durch

$$a_{ij} = \begin{cases} +1 & \text{wenn } i = j \text{ oder } j = n \\ -1 & \text{wenn } i > j \\ 0 & \text{sonst.} \end{cases}$$

- a) Begründen Sie, dass die LR-Zerlegung ohne Pivotisierung von A die Eigenschaften

$$|l_{ij}| \leq 1 \quad \text{und} \quad r_{nn} = 2^{n-1}$$

erfüllt.

- b) Zeigen Sie, dass eine LR-Zerlegung von A mit Zeilen- und Spaltenvertauschung existiert mit

$$|r_{nn}| = 2 = \max_{i,j=1..n} \{r_{ij}\}.$$

Tipp: Betrachten Sie zunächst $n = 4$ um die beste Vertauschungsstrategie zu finden.

(5 Punkte)

Übung 4 LR-Zerlegung für symmetrisch positiv definite Matrizen (Praktische Übung)

In dieser praktischen Übung sollen Sie die LR-Zerlegung ohne Zeilenvertauschung auf eine symmetrisch positiv definite Matrix anwenden und die dabei auftretenden Pivotelemente untersuchen.

- a) Schreiben Sie eine neue Headerdatei `lr_no_pivot.hh`, welche die Template-Funktion

```
template<class T>
void lr_no_pivot (hdnum::DenseMatrix<T>& A, hdnum::Vector<T>& pivotElements)
{...}
```

enthält.

Die Funktion `void lr_no_pivot(...)` soll die LR-Zerlegung ohne Zeilenvertauschung von A berechnen und die Ergebnisse L und R wiederum in die Matrix A speichern. In dem Vektor `pivotElements` sollen die Pivotelemente gespeichert werden.

Mit dieser Headerdatei können Sie das auf der Vorlesungsseite zur Verfügung gestellte Hauptprogramm `lr_sym_pos_def.cc` ausführen. Dort wird für $n = 15$ die Matrix $A \in \mathbb{R}^{n \times n}$ des Gleichungssystems zum Rohrleitungsnetzwerk aus den früheren Übungen aufgestellt. Wenden Sie ihre Funktion `void lr_no_pivot(...)` auf diese Matrix an und geben Sie die Pivotelemente aus.

- b) Fügen Sie den Vektor $b \in \mathbb{R}^n$ zum Programm hinzu mit $(b)_i = 0$ für $0 < i < n$ und $(b)_n = 10$. Dieser Vektor entspricht gerade der rechten Seite des Gleichungssystems $Ax = b$ aus der Rohrleitungsaufgabe für die Flussrate $q_P = 10$.

Lösen Sie das lineare Gleichungssystem $Ax = b$ mithilfe Ihrer LR Zerlegung aus der ersten Teilaufgabe. Für die Vorwärts- und Rückwärtssubstitution dürfen Sie die Funktionen

```
//! Assume L = lower triangle of A with l_ii=1, solve L x = b
template<class T>
void solveL (const DenseMatrix<T>& A, Vector<T>& x, const Vector<T>& b)
{...}
```

und

```
//! Assume R = upper triangle of A and solve R x = b
template<class T>
void solveR (const DenseMatrix<T>& A, Vector<T>& x, const Vector<T>& b)
{...}
```

verwenden, welche bereits in HDNum implementiert sind.

c) Überprüfen Sie ihr Ergebnisse, indem Sie den relativen Fehler des Residuums

$$r = \frac{\|A\hat{x} - b\|_2}{\|b\|_2}$$

ausrechnen und ausgeben. Dabei sei \hat{x} die numerische Lösung des Gleichungssystems.

Vorsicht: Durch das Aufstellen der LR-Zerlegung und durch die Vorwärts-/Rückwärtssubstitution wurden die Matrix A und der Vektor b überschrieben. Das müssen Sie beim ausrechnen von r beachten!

(5 Punkte)