Universität des Saarlandes
Fachrichtung Mathematik
Dr. Christian Steinhart
Friedrich Günther

Übungen zur Vorlesung „Höhere Mathematik für Ingenieure IV A“
Sommersemester 2025

Blatt 4
Abgabe bis Dienstag, 6. Mai 2025, 20 Uhr

Aufgabe 1 (4+3=7 Punkte): Seien n eine natürliche Zahl und A in $\mathrm{GL}(n, \mathbb{R})$.

(i) Zeigen Sie, dass $\kappa(A) \geq 1$ und dass $\kappa(A^{-1}) = \kappa(A)$.

(ii) Ist A sogar orthogonal, d. h. $A^t = A^{-1}$, dann ist $\kappa_{\|\cdot\|_2}(A) = 1$.

Hierbei ist in (i) die Konditionszahl κ bezüglich irgendeiner submultiplikativen Matrixnorm gemeint, und in (ii) die Konditionszahl bezüglich der *Spektralnorm*

$$\|A\|_2 = \max\{\sqrt{\mu} \mid \mu \in \mathrm{Spec}(AA^t)\},$$

wobei $\mathrm{Spec}(AA^t)$ die Menge der Eigenwerte von AA^t ist. Die Spektralnorm ist die von der $\|\cdot\|_2$-Norm induzierte Matrixnorm.

Aufgabe 2 (3 Punkte): Seien A in $\mathbb{R}^{n\times m}$, b in $\mathbb{R}^n$ und x_0 in $\mathbb{R}^m$ mit $Ax_0 = b$. Seien x ein weiterer Vektor in $\mathbb{R}^m$ und $\varepsilon > 0$ eine Fehlerschranke. Können Sie ein Kriterium für $\|x - x_0\|$ geben, damit sicherlich $\|Ax - b\| < \varepsilon$ gilt?

Aufgabe 3 (2+3+3+2=10 Punkte): Anders als auf dem letzten Aufgabenblatt verwenden wir für die Programmieraufgaben im Folgenden die von `NumPy` zur Verfügung gestellten Datentypen für Matrizen und Vektoren, nämlich `numpy.array`. Damit stehen (schnelle und sichere) Funktionen für grundlegende Matrix-Vektor-Algebra zur Verfügung.

(i) Seien zunächst $A = \left(\begin{smallmatrix} 2 & 1 \\ 0 & 2 \end{smallmatrix}\right)$, $x = \left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right)$, $b = \left(\begin{smallmatrix} 3 \\ 2 \end{smallmatrix}\right)$ und $\varepsilon = 0.2$. Skizzieren Sie $B_{\|\cdot\|_\infty}(x, \varepsilon)$, $B_{\|\cdot\|_\infty}(b, \varepsilon)$ und $A(B_{\|\cdot\|_\infty}(x, \varepsilon))$. Was fällt Ihnen auf?

Hinweis: $B_{\|\cdot\|_\infty}(x, r) = \{y \in \mathbb{R}^2 \mid \|x - y\|_\infty < r\}$.

(ii) Schreiben Sie eine Funktion `check_dominance(A)`, die

$$L = \max_{1\leq i\leq n} \frac{1}{|a_{i,i}|}\left(\sum_{j=1}^{n}|a_{j,i}|\right) - 1$$

berechnet, und L zurückgibt, falls $L < 1$, sonst -1.

Hinweis: Die `.shape`-Funktion für `numpy.arrays`, die einen n-Tupel mit den Dimensionen des übergebenen Arrays übergibt, könnte praktisch sein. Auch könnte die Funktion `.sort()` für Listen bei der Implementierung helfen.

(iii) Schreiben Sie eine Funktion `Jacobi_iteration(A,b,s,e)`, die eine Matrix `A`, eine rechte Seite `b`, einen Startwert `s` und eine Fehlerschranke `e` einliest. Falls A strikt diagonaldominant ist, soll die Funktion eine näherungsweise Lösung `x*` berechnen, für die $\|\mathtt{Ax*} - \mathtt{b}\|_\infty < \mathtt{e}$ gilt.

(iv) Testen Sie ihre Funktionen mit

$$A = \begin{pmatrix} 3 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & 3 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}, \quad s = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad \varepsilon = 0.001.$$

Wie viele Iterationen sind nötig? Welche näherungsweise Lösung wird zurückgegeben?