Heinrich Heine Universität Düsseldorf

MATHEMATISCHES INSTITUT
PROF. DR. ACHIM SCHÄDLE
MELINDA HAGEDORN

23.01.2025

Computergestützte Mathematik zur linearen Algebra – 14. Übungsblatt

Aufgabe 53: (*Lineare Ausgleichsrechnung I*)

Gegeben seien Punkte (z. B. Messdaten) $(t_i, f_i)_{i=1,\dots,9}$ in der Datei `a53.dat`. Die Datei enthält neun Zeilen. Jede Zeile enthält zwei durch eine Leerzeichen getrennte Werte. Die j-te Zeile enthält zuerst den Wert t_j und als zweites den Wert f_j. Es wird angenommen, dass für reelle Zahlen x_1, x_2, x_3, x_4 der folgende Zusammenhang besteht:

$$f(t) = x_1 \,\mathrm{e}^t + x_2 \,\mathrm{e}^{-t} + x_3 \,t^2 + x_4$$

Gesucht sind die Parameter x_1, x_2, x_3, x_4, für die die Funktion f die Werte f_i in den Punkten t_i im Sinne der kleinsten Fehlerquadrate bestmöglich approximiert.

(a) Formulieren Sie das entsprechende lineare Ausgleichsproblem, d.h. stellen Sie eine Matrix $A \in \mathbb{R}^{9\times 4}$ und einen Vektor $b \in \mathbb{R}^9$ auf, so dass die Parameter $x_1, x_2, x_3, x_4 \in \mathbb{R}$ von f die eindeutige Lösung des Problems $\min_{x\in\mathbb{R}^4} \|b - Ax\|_2$ sind.

(b) Lösen Sie das lineare Ausgleichsproblem mit Hilfe der kompakten QR-Zerlegung aus Aufgabe 49 von A und unter Verwendung der Funktion `QTprodb` aus Aufgabe 50 (beide Musterlösungen sind auf der Homepage). Geben Sie für die Lösung x^* die euklidische Norm $\|b - Ax^*\|_2$ aus.

(c) Erstellen Sie folgende Graphik auf dem Intervall $[-2, 2]$:
   (i) Stellen Sie die Datenpunkte $(t_i, f_i)_{i=1,\dots,9}$ als magenta-farbene Rauten (♦) dar.
   (ii) Verbinden Sie die Datenpunkte mit einem orange gestrichelten Polygonzug.
   (iii) Stellen Sie die Funktion f zu den berechneten Parametern als schwarze Kurve dar.

Aufgabe 54: (*Lineare Ausgleichsrechnung II*)

In folgendem Modell wird angenommen, dass zur Zeit t die Position $q(t) \in \mathbb{R}^3$ eines Planeten im Sonnensystem durch

$$q(t) = \begin{pmatrix} q_0(t) \\ q_1(t) \\ q_2(t) \end{pmatrix} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} \sin(t) + \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} \cos(t) + \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix}$$

beschrieben wird. Auf der Homepage der Vorlesung finden Sie eine Datei `Planet.dat`, die in jeder Zeile das Ergebnis einer Positionsbestimmung enthält. Eine Zeile besteht immer aus den vier Werten $t, q_0(t), q_1(t), q_2(t)$, in der angegebenen Reihenfolge.

(a) Stellen Sie ein lineares Ausgleichsproblem zur Bestimmung der neun Werte $a_0, a_1, a_2, b_0, b_1, b_2, c_0, c_1$ und c_2 auf.

(b) Lösen Sie das lineare Ausgleichsproblem mit einer der folgenden Methoden: Normalengleichung, QR-Zerlegung über `scipy.linalg.qr` oder mit Hilfe der kompakten QR-Zerlegung wie in Aufgabe 53(b).

(c) Stellen Sie die Werte aus der Datei `Planet.dat` und die Umlaufbahn des Planeten graphisch dar.

b.w.

Aufgabe 55: (*Approximation des betragsgrößten Eigenwerts, Vektoriteration/Potenzmethode*)

Sei $m \geq 1$ und $A \in \mathbb{R}^{m\times m}$ symmetrisch. Gesucht ist der betragsmäßig größte Eigenwert von A und ein zugehöriger Eigenvektor. Dazu dient der folgende Algorithmus:

> Wähle eine Toleranz tol > 0 und einen zufälligen Startvektor $v_0 \in \mathbb{R}^m$.
> **for** $n = 1, 2, \ldots$ **:**
> Setze $v_n = Av_{n-1}/\|Av_{n-1}\|_2$ `# n-te Schätzung eines normierten Eigenvektors`
> Setze $\lambda_n = (v_n)^T Av_n$ `# n-te Schätzung eines betragsgrößten Eigenwerts`
> Brich ab, falls $\|Av_n - \lambda_n v_n\|_2 <$ tol

(a) Schreiben Sie eine Funktion `ev,ew = findeEW(A, tol, N)`, die den obigen Algorithmus implementiert. `A` ist die Matrix A, `tol` die Toleranz für das Abbruchkriterium und `N` die Maximalanzahl an Iterationen. `ev` und `ew` sind Schätzungen für den Eigenwert bzw. den dazugehörigen Eigenvektor. Überprüfen Sie, ob `A` wirklich symmetrisch ist und geben Sie andernfalls eine Fehlermeldung aus.

(b) Setzen Sie `N` $=$ `20`, `tol` $=$ `10`$^{-5}$ und testen Sie Ihre Implementierung an den Matrizen `np.diag([10]+[1]*(N-1))`, `np.diag([10,9]+[1]*(N-2))` und einer symmetrischen 100×100 Zufallsmatrix `Z`. Vergleichen Sie das Ergebnis mit `numpy.linalg.eig()`.

Aufgabe 56: (*Gershgorin-Kreise*)

Auf der Kurs-Website finden Sie die Datei `aufg56_mit_fehlern.py`. Sie enthält die Lösung zu Aufgabenteil (a) dieser Übungsaufgabe, allerdings auch fünf Fehler. Korrigieren Sie die Fehler, erklären Sie den Code und implementieren Sie Aufgabenteil (b).

(a) Schreiben Sie eine Funktion `Gershgorin(A)`, welche für die Matrix $A \in \mathbb{C}^{n\times n}$ die Ränder der Gershgorin-Kreise

$$\mathcal{D}_i = \left\{ z \in \mathbb{C} \;\middle|\; |z - a_{ii}| \leq \sum_{j=1,\ j\neq i}^{n} |a_{ij}| \right\}, \qquad i = 1, 2, \ldots, n,$$

und die Eigenwerte in der komplexen Ebene zeichnet. Die Eigenwerte können Sie über den Befehl `numpy.linalg.eig` berechnen.

(b) Testen Sie die Funktion `Gershgorin` mit der Matrix

$$A = \begin{pmatrix} 6 & 0 & 1 & 2 \\ 3 & -6 & 2 & 1 \\ 3 & 6 & 12+6i & 0 \\ 0 & 6 & \frac{3}{\sqrt{2}}\cdot(1+i) & -6i \end{pmatrix}.$$

Besprechung in den Übungen vom 27.01.2025 bis 31.01.2025.