Einführung in die Grundlagen der Numerik

Winter semester 2019/2020

Prof. Dr. Marc Alexander Schweitzer
Denis Düsseldorf

universitätbonn

Übungsblatt 9.

Abgabe: 17.12.2019 vor der Vorlesung

Aufgabe 24. (Störungseigenschaft des Rayleigh-Quotient)

Es sei $A \in \mathbb{K}^{n\times n}$ für $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ und (λ, x) ein Eigenpaar von A, d.h.

$$Ax = \lambda x.$$

Zeigen Sie, dass für den Rayleigh-Quotienten

$$\mu_A(z) \coloneqq \frac{z^H Az}{z^H z}$$

folgende Störungseigenschaften gelten.

a) Es ist $|\mu_A(x+h) - \mu_A(x)| = \mathcal{O}(\|h\|)$ für $h \to 0$.

b) Es ist $|\mu_A(x+h) - \mu_A(x)| = \mathcal{O}(\|h\|^2)$ für $h \to 0$, falls A hermitesch ist.

(4 Punkte)

Aufgabe 25. (Generalisierter Rayleigh Quotient)

Es seien $A, B \in \mathbb{R}^{n\times n}$ zwei symmetrische, positiv-definite Matrizen und $x \in \mathbb{R}^n$. Der generalisierte Rayleigh-Quotient ist definiert als

$$R(A, B, x) \coloneqq \frac{x^T Ax}{x^T Bx}.$$

Zeigen Sie, dass das Maximum des generalisierten Rayleigh-Quotienten genau der größte Eigenwert des generalisierten Eigenproblems

$$Ax = \lambda Bx$$

ist, d.h.

$$\max_{x\in\mathbb{R}^n, x\neq 0} R(A, B, x) = \lambda_{max}(A, B).$$

Zeigen Sie außerdem, dass auch

$$\min_{x\in\mathbb{R}^{n}, x\neq 0} R(A, B, x) = \lambda_{min}(A, B).$$

(4 Punkte)

Aufgabe 26. (Absolute und relative Kondition / Verstärkungsfaktoren)

Es sei $y(x_1, \dots, x_n)$ eine zu berechnende Problemstellung. Die *absoluten Konditionszahlen* des Problems sind definiert als

$$\delta_i := \frac{\partial y(x_1, \dots, x_n)}{\partial x_i}.$$

Mit den absoluten Konditionszahlen können die *relativen Konditionszahlen* definiert werden als

$$\varrho_i = \left| \delta_i \frac{x_i}{y(x_1, \dots, x_n)} \right|.$$

Man spricht über die relativen Konditionszahlen auch als *Verstärkungsfaktoren des relativen Fehlers*. Sie geben an, wie sehr eine Störung in den Daten das Ergebnis der Berechnung der Problemstellung beeinflusst. Man spricht von einer *gut konditionierten Problemstellung*, wenn ϱ_i klein sind. Andernfalls ist das Problem *schlecht konditioniert*.

a) Es sei

$$A = \begin{bmatrix} a & -b & 0 \\ b & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

mit $a, b \in \mathbb{R} \setminus \{0\}$. Ist die Berechnung der Determinante von A gut konditioniert bzgl. den Parametern $a \in \mathbb{R}$ bzw. $b \in \mathbb{R}$?

b) Es sei $f : \mathbb{R} \to \mathbb{R}$, $f(x) := e^{3x^2}$. Wann ist diese Funktion gut und wann schlecht konditioniert?

(4 Punkte)

Programmieraufgabe 7. (Eigenwerte)

a) Implementieren Sie eine Funktion

```
def compute_gerschgorin_circles(A):
    # Input:
    #   A    Reelle n x n Matrix
    # Output:
    #   ms   Mittelpunkte
    #   rs   Radien

    # Ihr Code hier
```

die zu einer Matrix $A = [a_{i,j}]_{i,j=1}^n \in \mathbb{R}^{n\times n}$ die Gerschgorinkreise berechnet, d.h. deren Mittelpunkte und Radien. **Benutzen Sie weder für die Berechnung der Mittelpunkte, noch zur Berechnung der Radien eine `for` Schleife. Benutzen Sie stattdessen ausschließlich list comprehensions.**

Zur Erinnerung: Die Gerschgorinkreise sind Teilmengen der komplexen Zahlen, die durch einen Kreis beschrieben werden. Die Mittelpunkte der Gerschgorinkreise sind $m_i = a_{i,i}$ und die Radien sind $r_i = \sum_{j=1,j\neq i}^n |a_{i,j}|$ für $i = 1, \dots, n$.

b) Schreiben Sie eine Routine die eine reelle untere (obere) Schranke für den kleinsten (größten) reellen Eigenwert einer Inputmatrix berechnet. Nutzen Sie hierzu die Routine zur Berechnung der Gerschgorinkreise aus a).

c) Plotten Sie die Gerschgorinkreise für die Matrix `gerschgorin_50.txt`, die Sie als Textdatei auf der Webseite finden. Nutzen Sie hierfür die Darstellung von $\mathbb{C}$ als $\mathbb{R}^2$. Die Kreise sollten zur Unterscheidung alle eine andere Farbe haben. Die Skalierung der x- und y-Achse muss nicht gleich sein (die Mittelpunkte bewegen sich in einem weit größeren Bereich als die Radien) - Ihre Gerschgorinkreise können also im Plot wie Ellipsen aussehen.

Tipp: Importieren Sie `pyplot` mittels `import matplotlib.pyplot as plt`. Anschließend können Sie mittels `colors = plt.cm.rainbow(np.linspace(0,1,n))` eine Liste von n-verschiedenen Farben generieren (für jeden Kreis eine). Kreise können geplottet werden mittels

```
fig,ax = plt.subplots()
ax.add_artist(plt.Circle((x,y),r,color=c))
```

wobei (x, y) der Mittelpunkt und r der Radius des Kreises ist. Falls der Plot nicht korrekt dargestellt wird, kann es nötig sein die darzustellenden Intervalle auf der x und y Achse manuell zu definieren. Das geht mittels `plt.xlim((xmin,xmax))` bzw. `plt.ylim((ymin,ymax))`.

(5 Punkte)

Abgabe per Mail an die Tutoren.