Heinrich Heine Universität Düsseldorf

Mathematisches Institut
Prof. Dr. Achim Schädle
Melinda Hagedorn

28.11.2024

Computergestützte Mathematik zur linearen Algebra – 8. Übungsblatt

Aufgabe 29: (*Determinante*)

(a) Schreiben Sie eine Funktion `mydet(A)`, welche rekursiv mit Hilfe des Laplaceschen Entwicklungssatzes die Determinante einer $(n \times n)$-Matrix A berechnet. Falls A nicht quadratisch ist, soll eine aussagekräftige Fehlermeldung ausgegeben werden. Falls benötigt, dürfen Sie zusätzliche Hilfsfunktionen schreiben, jedoch keine bereits vorhandene Python-Methode zur Berechnung einer Determinante verwenden.

(b) Testen Sie Ihre Funktion an folgenden Matrizen. Vergleichen Sie Ihr Ergebnis mit dem des Befehls `numpy.linalg.det`.

$$A = \begin{pmatrix} 1 & 3 & 2 & 0 \\ 1 & 1 & 2 & 3 \\ 4 & 0 & 8 & 3 \\ 2 & 5 & 4 & 5 \end{pmatrix}, \quad B = \begin{pmatrix} 4 & 5 & 6 & 6 \\ 7 & 7 & 9 & -3 \\ 8 & 4 & 0 & -7 \\ -5 & 3 & -6 & -4 \end{pmatrix}, \quad C = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

Aufgabe 30: (*Cramersche Regel*)

(a) Mithilfe der *Cramerschen Regel* kann man lineare Gleichungssysteme lösen.
Schreiben Sie eine Funktion `x=solveCramer(A, b)`, die genau das tut.

Cramersche Regel:
Seien eine invertierbare Matrix $A \in \mathbb{R}^{n\times n}$ und ein Vektor $b \in \mathbb{R}^n$ gegeben. Dann löst x, mit $x_k = \frac{\det A_k}{\det A}$, das LGS $Ax = b$. Hierbei ist $\det A$ die Determinante von A. A_k ist die Matrix, bei der die k-te Spalte von A durch b ersetzt wurde.

Hinweise: Für die Berechnung der Determinanten können Sie Ihre Funktion `mydet` aus Aufgabe 29 oder `numpy.linalg.det` verwenden. Beachten Sie, dass die Eingabematrix A durch Ihre Funktion NICHT verändert werden soll!

(b) Testen Sie Ihre Funktion anhand einer Zufallsmatrix Z der Dimension 10×10 und eines Zufallsvektors v passender Länge. Berechnen Sie hierfür die euklidische Norm des Residuums $Zx - v$. Vergleichen Sie Ihr Ergebnis mit der *numpy*-Funktion `numpy.linalg.solve`

b.w.

Aufgabe 31: (*Zeilenstufenform*)

(a) Gegeben sei eine Matrix $A \in \mathbb{R}^{m\times n}$. Schreiben Sie eine Funktion `zeilenstufen(A)`, die die Matrix A in Zeilenstufenform bringt. Sie können die notwendigen Umformungen entweder mit Hilfe von Multiplikationen von links mit Elementarmatrizen vornehmen oder direkt auf der Matrix A ausführen. Achten Sie darauf, dass Ihr Algorithmus mit Matrixeinträgen vom Typ `float` rechnet.

(b) Wie erhalten Sie den Rang der Matrix aus der Zeilenstufenform? Erweitern Sie Ihre Funktion so, dass sie neben der umgeformten Matrix auch deren Rang ausgibt.

(c) Testen Sie Ihre Funktion an den Matrizen

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -2 & 2 & -2 \\ 2 & -1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 3 & 1 & 2 & 3 \\ 1 & 2 & 4 & 0 \\ 2 & -1 & -2 & 4 \end{pmatrix} \quad \text{und} \quad C = \begin{pmatrix} 0 & 0 & 4 \\ 0 & 1 & 5 \\ 0 & 1 & 5 \end{pmatrix}.$$

Aufgabe 32: (*Matrix-Normen*)

Die Spektralnorm einer Matrix $M \in \mathbb{R}^{m\times n}$ kann geschrieben werden als

$$\|M\|_2 = \max\{\sqrt{|\lambda|} \mid \lambda \in \mathbb{R},\ M^{\mathrm{T}}Mx = \lambda x,\ x \in \mathbb{R}^n \setminus \{0\}\}.$$

(a) Schreiben Sie eine eigene Funktion, die für eine gegebene $(m \times n)$-Matrix M die Spektralnorm von M berechnet. Lassen Sie Ihre Funktion aussagekräftige Warnungen mithilfe von `assert` ausgeben, wenn M keine reelle Matrix ist.

Hinweis: Die Funktion `numpy.linalg.eig` könnte nützlich für Sie sein.

(b) Finden Sie heraus, wie man mithilfe von `numpy.linalg.norm` die Spektralnorm von M berechnen kann.

(c) Erstellen Sie sich eine zufällige Matrix $A \in \mathbb{N}^{300\times 200}$ und berechnen Sie mithilfe von (a) und (b) jeweils die Spektralnorm von A. Welche der beiden Methoden benötigt dafür mehr Zeit? Wie weit sind die beiden Ergebnisse voneinander entfernt?

(d) Was bedeutet der Wert, der ausgegeben wird, wenn man `numpy.linalg.norm(A)` eingibt?

Besprechung in den Übungen vom 02.12.2024 bis 06.12.2024.