Heinrich Heine Universität Düsseldorf

MATHEMATISCHES INSTITUT
PROF. DR. ACHIM SCHÄDLE
MELINDA HAGEDORN

5.12.2024

Computergestützte Mathematik zur linearen Algebra – 9. Übungsblatt

Aufgabe 33: (*Vorwärtssubstitution*)

Gegeben sei eine linke untere Dreiecksmatrix $L = (\ell_{ij})_{i,j=1}^n \in \mathbb{R}^{n\times n}$ und ein Vektor $b = (b_i)_{i=1}^n \in \mathbb{R}^n$.

(a) Schreiben Sie eine Funktion `vorwaerts(L,b)`, welche das lineare Gleichungssystem $Lx = b$ durch Vorwärtssubstitution löst und den Vektor x ausgibt. Benutzen Sie dabei nicht den `solve`-Befehl, sondern berechnen Sie x wie folgt:

$$x_1 = \frac{b_1}{\ell_{11}} \text{ und } x_i = \frac{1}{\ell_{ii}}\left(b_i - \sum_{k=1}^{i-1} x_k \ell_{ik}\right) \text{ für } i = 2, \dots, n$$

(b) Wie erkennt man, ob L singulär ist? Erweitern Sie Ihre Funktion, so dass mit Hilfe von `assert` eine sinnvolle Fehlermeldung ausgegeben wird, sollte L nicht invertierbar sein. Berechnen Sie dazu NICHT die Determinante der Matrix.

(c) Testen Sie Ihre Vorwärtssubstitution mit

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2 & 4 & 0 & 0 \\ 4 & -5 & 6 & 0 \\ -7 & 8 & -9 & 3 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} 3 \\ -6 \\ 6 \\ 0 \end{pmatrix}$$

und überprüfen Sie das Ergebnis mit Hilfe der Befehle `numpy.linalg.solve` und `numpy.allclose`.

Aufgabe 34: (*Gleichungssysteme stören*)

Gegeben seien die Matrizen

$$A = \begin{pmatrix} 1 & 10^{-11} \\ 0 & 2 \end{pmatrix} \text{ und } B = \begin{pmatrix} 10^{-11} & 1 \\ 0 & 2 \end{pmatrix} \text{ und der Vektor } b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

(a) Lösen Sie die Gleichungssysteme $Ax_A = b$ und $Bx_B = b$.

(b) Erstellen Sie nun die gestörten Matrizen $\hat{A}$ und $\hat{B}$, indem Sie bei A und B jeweils zum Eintrag unten rechts 10^{-7} hinzuaddieren.

(c) Lösen Sie die Gleichungssysteme $\hat{A}\hat{x}_A = b$ und $\hat{B}\hat{x}_B = b$.

(d) Vergleichen Sie x_A und x_B mit $\hat{x}_A$ und $\hat{x}_B$. Was fällt Ihnen auf? Verwenden Sie den `cond`-Befehl aus dem `np.linalg`-Modul (was tut der Befehl überhaupt?) und andere Matrizen um ein Muster zu erkennen. Wieso ist das für die numerische Mathematik wichtig?

b.w.

Aufgabe 35: (*LR-Zerlegung / Fehler im Code*)

Gegeben ist folgende platzsparende, aber nicht ganz fehlerfreie Version einer LR-Zerlegung der nicht-singulären Matrix $A \in \mathbb{R}^{n\times n}$ *mit* Pivotisierung, die Sie auch auf der Vorlesungshomepage als `LR_kompakt_mit_Fehlern.py` finden.

```
1  import numpy as np
2
3  def LR_kompakt(A)
4      A = A.astype(’float’).copy()
5      n = A.shape[0]
6      p = arange(n)
7      for kk in range(n):
8          ll = kk + np.flatnonzero[A[kk:, kk]][0]
9          A[[ kk, ll ]] = A[[ ll, kk ]]
10         p[[ kk, ll ]] = p[[ kk, ll ]]
11         A[kk+1:, kk] /= A[kk, kk]
12         A[kk+1:, kk+1:] -= A[kk+1:, [ kk ]] @ A[[ kk ], kk+1:]
13     return p
```

(a) Finden und korrigieren Sie die **sechs Fehler** im Quelltext und überprüfen Sie die Funktion mit geeigneten Tests.
Tipp: Es ist $P \cdot A =$ `A[p,:]`.

(b) Kommentieren Sie wichtige Stellen im Quelltext sinnvoll.

(c) Was passiert, wenn Sie Zeile 8 durch `ll = kk + np.argmax(abs(A[kk:, kk]))` ersetzen? Was ist der Unterschied?

(d) Testen Sie Ihre Funktion an der Matrix

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{pmatrix},$$

indem Sie überprüfen, ob $PA = LR$ gilt, und indem Sie die Determinante von A mithilfe der pivotisierten LR-Zerlegung von A berechnen.

Aufgabe 36: (*Aufwand Determinanten-Berechnung*)

Sie haben verschiedene Möglichkeiten kennengelernt, die Determinante einer $(n \times n)$-Matrix zu berechnen:

- rekursiv mithilfe der Laplace-Entwicklung wie in Aufgabe 29,
- mithilfe von `numpy.linalg.det`,
- mithilfe einer LR-Zerlegung wie in Aufgabe 35.

Erstellen Sie für $n \in \{1, \ldots, 10\}$ zufällige $(n \times n)$-Matrizen, deren Einträge gleichverteilt auf $[0, 1)$ sind, und berechnen sie auf alle drei Weisen die Determinanten Ihrer Zufallsmatrizen. Messen Sie die Zeit, die Ihr Computer für die Berechnung jeweils benötigt. Erstellen Sie einen Plot, in dem Sie die benötigte Zeit gegen die Dimension n jeweils logarithmisch auftragen. Was fällt Ihnen dabei auf?

Besprechung in den Übungen vom 09.12.2024 bis 13.12.2024.