Institut für Analysis und Numerik 15.05.2020
Universität Magdeburg
Thomas Richter thomas.richter@ovgu.de
Gozel Judakova gozel.judakova@ovgu.de

Übung Nr. 5 zur Vorlesung Algorithmische Mathematik II
Sommersemester 2020

Abgabe bis zum 22. Mai 2020

Lineare Gleichungssysteme Thema der nächsten zwei bis drei Wochen ist das Lösen von linearen Gleichungssystemen. Es geht darum, die Methoden der linearen Algebra auf dem Computer umzusetzen. Mit dem Gauß'schen Eliminationsverfahren kennen wir eigentlich eine Methode, um Gleichungssystem zu lösen, um den Rang einer Matrix zu bestimmen, usw. Im Mittelpunkt der Betrachtung in der Vorlesung Algorithmische Mathematik stehen die folgenden Fragen:

- Was ist die Fehleranfälligkeit bei Umsetzung auf dem Computer mit nicht-exakter Arithmetik? Das bedeutet konkret: wie ist der Problem *Lösen eines LGS* konditioniert?
- Wie skaliert das Gauß'sche Verfahren wenn die Matrix $A \in \mathbb{R}^{n\times n}$ sehr groß ist, d.h. für $n = 100$, $n = 1\,000$ oder $n = 1\,000\,000$?
- Welche Alternative Ansätze gibt es? Diesen letzten Punkt werden wir nur streifen. Dies ist ein Teil der Vorlesung Numerische Mathematik im vierten Semester.

Auch wenn die Aufgabe *LGS Lösen* einfach wirkt, kommt ihr eine enorme Bedeutung zu. Lineare Gleichungssystem treten als Teilaufgabe in vielen Disziplinen auf, bei der Optimierung, bei der Diskretisierung von Differentialgleichungen, in der Algebra, der Geometrie und in vielen weiteren Gebieten. Die Gleichungssysteme können dabei sehr groß werden. In Klimasimulationen und für Wettervorhersagen müssen ständig LGS gelöst werden, bei denen die Matrixgröße $n \gg 1\,000\,000$ oft weit übersteigt. Ohne angepasste und sehr effiziente Verfahren ist dies nicht möglich.

numpy Zur praktischen Umsetzung werden wir das Python-Modul `numpy` verwenden. Für den Beginn brauchen wir nur sehr einfache Bestandteile und Grundfunktionen von `numpy`. Im Skript, gibt Kapitel 4 eine sehr kurze Einführung. Darüber hinaus wird derzeit ein Python-Kurs, abgestimmt auf die Vorlesung Algorithmische Mathematik, zusammengestellt. Der Kurs wird auf der Homepage (bis Sonntag, 17. Mai) verlinkt.

Wiederholung Lineare Algebra Man lese in Kapitel *7. Lineare Gleichungssysteme* bis einschließlich *Grundlagen der Linearen Algebra*. Das meiste sollte aus der Vorlesung Linearen Algebra gut bekannt sein, hier werden nur die wesentlichen Begriffe noch einmal zusammengefasst. Einige Aspekte sind für unsere Vorlesung nicht zentral, siehe auch den Abschnitt *Eigenwerte* unten. Ebenso ist der Abschnitt zur Orthogonalität, Definition 7.8 bis Beispiel 7.10, nicht zentral und kann schnell gelesen werden.
Für die numerische Umsetzung des Gaußschen Eliminationsverfahrens und dessen Analyse werden wir jedoch das Konzept der Norm etwas näher betrachten müssen:

- Wichtig ist der Begriff der *induzierten Norm*, also einer Norm, die mit einem Skalarprodukt zusammenhängt.
- Der Raum der Matrizen $A \in \mathbb{R}^{n\times n}$ kann als n^2-dimensionaler Vektorraum betrachtet werden. Man schreibt einfach alle Einträge untereinander. Dann ist die Addition von Matrizen $A + B$ und auch die skalare Multiplikation sA sinnvoll definiert. Wir kennen aber auch die Matrix-Matrix und die Matrix-Vektor Multiplikation, also zusätzliche Struktur. Wir bezeichnen den Vektorraum der Matrizen daher mit $\mathbb{R}^{n\times n}$ anstelle von $\mathbb{R}^{n^2}$. In der Vorlesung werden wir den Begriff der *Matrixnorm* kennenlernen. Eine Matrixnorm ist eine Norm, die jedoch noch weitere Eigenschaften hat.

Man beantworte die folgenden Fragen

1. Was ist die Lösung des linearen Gleichungssystem
$$\begin{pmatrix} 1 & -1 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$
2. Was ist die Dreiecksungleichung von Normen?
3. Es sei $A \in \mathbb{R}^{n\times n}$ eine Matrix, $x \in \mathbb{R}^n$ ein Vektor. Was ist der Aufwand, gemessen in $\mathcal{O}(n^\alpha)$ für die Matrix-Vektor Multiplikation
$$y = Ax \quad y_i = \sum_{j=1}^{n} A_{ij} x_j, \quad j = 1, \dots, n.$$
4. Was bedeutet es, dass drei Vektoren $v_1, v_2, v_3 \in \mathbb{R}^3$ bzgl. eines Skalarprodukts $(\cdot, \cdot)$ paarweise orthogonal sind?
5. Man nenne 2 Bedingungen die äquivalent dazu sind, dass eine Matrix $A \in \mathbb{R}^{n\times n}$ regulär ist.
6. Was heißt es, dass eine Matrixnorm und eine Vektornorm verträglich sind?
7. Es sei $\|\cdot\|$ eine Matrixnorm, die von einer beliebigen Vektornorm $\|\cdot\|$ induziert ist. Warum gilt dann für die Einheitsmatrix $I \in \mathbb{R}^{n\times n}$ immer
$$\|I\| = 1?$$

Eigenwerte Im Skript wird an vielen Stellen das Konzept der *Eigenwerte* angesprochen, die bei Ihnen in der Linearen Algebra noch nicht Thema waren. Die entsprechenden Stellen (Definition 7.13, Satz 7.16) können daher zunächst übersprungen werden. Auch die *Spektralnorm*, $\|\cdot\|_2$ für Matrizen hat mit Eigenwerten zu tun. Wir werden diese Norm jedoch verwenden können, ohne die exakten Hintergründe zu kennen. Es kann jedoch nicht schaden, Definition 7.13 zu überfliegen. Die Worte *Eigenwerte* und *Eigenvektoren* lassen sich bestimmt nicht komplett vermeiden. Es reicht aber zunächst zu wissen, dass für eine Matrix $A \in \mathbb{R}^{n\times n}$ ein Eigenwert $\lambda \in \mathbb{C}$ und ein zugehöriger Eigenvektor $v \in \mathbb{C}^n$ die Gleichung

$$Av = \lambda v$$

erfüllt. Ein Eigenvektor $v \in \mathbb{R}^n$ wird also durch Matrix-Vektor Multiplikation auf ein Vielfaches von sich selbst abgebildet.

Aufgabe 5.1:

Viel zu lesen, daher keine Theorieaufgabe.

Programmieraufgabe 5.2:

Man programmiere das Gaußsche Eliminationsverfahren zum Lösen eines linearen Gleichungssystems $Ax = b$. Dabei verwende man den folgenden Algorithmus:

```
Gegeben ist Matrix $A \in \mathbb{R}^{n \times n}$ und Vektor $b \in \mathbb{R}^n$
Für i von 1 bis n−1:    # Eliminieren
    Für j von i+1 bis n:
        $s = A_{ji}/A_{ii}$
        Für k von i+1 bis n:
            $A_{jk} = A_{jk} - sA_{ik}$
        $b_j = b_j - sb_i$
        $A_{ji} = 0$
Für i von n rueckwaerts bis 1:  # Rueckwaerts−Eliminieren
    $b_i = b_i/A_{ii}$
    $A_{ii} = 0$
    Für j von 1 bis i:
        $b_j = b_j - A_{ji}b_i$
        $A_{ji} = 0$
Rückgabe $b$
```

a) Man teste die Methode zur Lösung des linearen Gleichungssystems

$$\begin{pmatrix} 2 & -1 & 4 \\ -1 & 2 & -1 \\ 8 & -1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Die exakte Lösung ist $(x, y, z) = (0.125, 0.75, 0.375)$.
Falls die implementierte Methode nicht das korrekte Ergebnis liefert, so geben man die Matrix A und den Vektor b nach jedem Schritt aus und vergleiche mit einer Rechnung ohne Computer.

b) Man teste die Methode zur Lösung der linearen Gleichungssystem mit den vorgegebenen Testmatrizen bei großen Werten von $n \in \{2, 4, 8, 16, 32, 64, 128\}$ und bestimme die Zeit. Wie ist der Aufwand in Bezug auf die Problemgröße n?
Man gebe auch den Fehler zwischen der numerischen Lösung und der exakten Lösung aus. Diese ist in der Vorlage angegeben, auch einige Tipps zum Umgang mit numpy z.B. zur Berechnung von Fehlern.

c) (**Zusatz**) Man verwende den `slice`-Operator von `numpy` zur Beschleunigung der Gauselimination (die k-Schleife kann ersetzt werden) und wiederhole die Zeitmessung aus Aufgabe **b**). Wieviel schneller (relativ) wird das Verfahren?

Hinweis: Große Fehler für $n > 10$ bedeuten nicht, dass Sie etwas falsch programmiert haben! Das Lösen großer linearer Gleichungssysteme ist schwerer als gedacht!