

1. Aufgabenblatt

Dieser erste Übungszettel ist noch auf freiwilliger Basis, wobei Sie bereits durch Vorrechnen Tafel-Punkte sammeln können. Das genaue Benotungssystem wird ebenfalls in der 1. Übungsstunde am 01.10.2019 besprochen.

Aufgabe 1 (Ein Rückblick in die lineare Algebra).

Gegeben sei die Matrix A und der Vektor x mit

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 4 \\ 3 & 0 & 2 \end{pmatrix} \quad \text{und} \quad x = \begin{pmatrix} 4 \\ -5 \\ 7 \end{pmatrix}.$$

- a) Berechnen Sie das Matrix-Vektor-Produkt Ax , sowie die Matrix-Produkt $A^2 = AA$ und A^2x .
- b) Bestimmen Sie (mittels händischer Rechnung) A^{-1} sowie die Determinante von A .
- c) Zeigen Sie für $N \in \mathbb{N}$ dass die Menge $\text{Reg}[N] = \{A \in \mathbb{R}^{N \times N} \mid A \text{ regulär}\} \subset \mathbb{R}^{N \times N}$ offen ist, d.h. für jede reguläre Matrix $A \in \text{Reg}[N]$ gibt es ein $\delta > 0$ sodass für alle Matrizen $B \in \mathbb{R}^{N \times N}$ mit $|B_{ij}| \leq \delta$ für $i, j \in \{1, \dots, N\}$ auch $A + B$ regulär ist. Folgern Sie daraus, dass die Abbildung $\mathcal{I}: \text{Reg}[N] \rightarrow \text{Reg}[N]$ mit $\mathcal{I}(A) = A^{-1}$ stetig ist, das heißt wenn $\lim_{n \rightarrow \infty} B_n = A$ dann auch $\lim_{n \rightarrow \infty} B_n^{-1} = A^{-1}$ wobei der Grenzwert eintragsweise zu verstehen ist. Hierbei ist $(B_n)_n$ eine Folge von Matrizen in $\text{Reg}[N]$.

Hinweis. Die Aufgaben a) und b) können mit linearer Algebra 1 gelöst werden. Für c) nutzen Sie, dass A regulär ist genau dann wenn $\det(A) \neq 0$, sowie die Cramersche Regel um Stetigkeit zu zeigen. Überzeugen Sie sich dafür insbesondere davon, dass $(A_{11}, A_{12}, \dots, A_{1N}, A_{21}, \dots, A_{NN}) \mapsto \det(A)$ stetig ist.

Bemerkung. Aufgabe 1 c) zeigt, dass das Lösen eines Gleichungssystems durch leichte Störung der Matrix A nicht zu stark beeinflusst wird. Dies garantiert, dass selbst wenn eine Matrix nicht absolut korrekt ist (Beispielsweise aufgrund von Rundungsfehlern beim Initialisieren), dennoch eine annähernd korrekte Lösung gefunden werden kann.

Aufgabe 2 (Binäre Zahlendarstellung). Diese Aufgabe beschäftigt sich mit der Darstellung von Zahlen im Dualsystem.

- a) Berechnen Sie für die Zahl 27 die Binärdarstellung, das heißt bestimmen Sie Zahlen $a_0, a_1, a_2, a_3, a_4 \in \{0, 1\}$ sodass $27 = a_0 2^0 + a_1 2^1 + a_2 2^2 + a_3 2^3 + a_4 2^4 = \sum_{i=0}^4 a_i 2^i$.
- b) Zeigen Sie, dass für jedes $a \in \mathbb{N}$ eine Binärdarstellung existiert, das heißt für $n \in \mathbb{N}_0$ sodass $2^n \leq a < 2^{n+1}$ existieren $a_i \in \{0, 1\}$ für $i \in \{0, \dots, n\}$ sodass $a = \sum_{i=0}^n a_i 2^i$.
- c) Gegeben seien $a = 101011_2 = 43$ und $b = 101_2 = 5$, also die Binärdarstellung von Zahlen a und b . Berechnen Sie $a + b$ und $a - b$ und a/b (als Division mit Rest) im Dualsystem ohne auf das Dezimalsystem zu wechseln.

Hinweis. Für b): Nutzen Sie dass $\sum_{i=0}^n 2^i = 2^{n+1} - 1$. Versuchen Sie anhand a) zu verstehen wie sich die a_i berechnen lassen. Ein leichter Beweis mittel vollständiger Induktion nach n löst dann b): Induktionsbehauptung für $a \in \mathbb{N}$ mit $2^n \leq a < 2^{n+1}$ gibt es eine Binärdarstellung. Für c): An den Grundrechenarten und deren Berechnung mit der Hand ändert sich nichts wenn eine andere Basis als 10 verwendet wird.

Bemerkung. Während wir es gewohnt sind mit Zahlen im Dezimalsystem – also zur Basis 10 – zu rechnen, lässt sich dies auch zu beliebigen anderen ganzzahligen Basen durchführen. Aufgrund der ("elektrischen") Funktionsweise von Computern, verwenden diese das Dualsystem für alle Berechnungen. Dieser Fakt ist insbesondere relevant, da Computer nach jeder Operation Zahlen runden, wobei die Wahl der Basis Einfluss auf das Ergebnis hat.

Matlab-Aufgabe 3 (Matlab Einleitung). Dieses Beispiel dient als sanfte Erinnerung an Matlab: Für geübte Programmierer ist diese Aufgabe in weniger als 5 Minuten lösbar, aber gerade wenn Sie mit Matlab Schwierigkeiten haben empfehle ich diese Aufgabe zu machen um Ihre Kenntnisse aufzufrischen.

- a) Schreiben Sie ein Matlab-Skript welches A aus Aufgabe 1 definiert. Berechnen Sie des weiteren in diesem Skript die zugehörigen Größen: das Matrixquadrat A^2 , die transponierte Matrix A^T sowie das eintragsweise Quadrat von A . Nutzen Sie "Indexing" um die 2.te Zeile von A^2 sowie eine Matrix $B = \begin{pmatrix} A_{11} & A_{13} \\ A_{31} & A_{33} \end{pmatrix}$ zu erhalten. Geben Sie alle Ergebnisse aus (beispielsweise via `disp('A');disp(A);`).
- b) Schreiben Sie eine Matlab-Funktion `[e]=ainb(a,b)` die für einen Zeilenvektoren b und eine Zahl a überprüft, ob ein Eintrag in b gleich a ist. Wenn dem so ist, soll der Wert $e = 1$ zurückgegeben werden, andernfalls soll $e = 0$ zurückgegeben werden. Um dies zu erreichen, setzen Sie am Anfang der Funktion $e=0$ und benutzen sie die "length" Funktion um n – die Länge des Vektors b – zu bestimmen. Nutzen Sie anschließend eine For-Schleife mit Variable $i = [1 : n]$ um jeden Eintrag in b zu betrachten. Überprüfen Sie mittels einer If-Abfrage ob $b(i) == a$, und setzen Sie in diesem Fall $e = 1$. Geben Sie am nachdem die For-Schleife zu ende ist mittels "return" den Wert von e zurück.

Hinweis. Links zu den Relevanten Funktionen und Operationen: Indexing Funktion, Grundlegende Matrix-Operationen, For-Schleifen, If-Abfrage, Vergleichsoperatoren, length

Bemerkung. Numerische Mathematik und Programmieren sind eng miteinander verwoben, da die meisten Berechnungen auf Computern numerische Mathematik benötigen. Andererseits entfalten viele numerische Methoden erst bei größeren Rechnungen (welche ohne Computer nicht umsetzbar wären) ihren vollen Effekt. Aus diesem Grund wird es in diesem Kurs mehrmals Programmier-Aufgaben geben, die Sie mit Matlab/Octave lösen sollten.