

Computergestützte Mathematik zur linearen Algebra – 5. Übungsblatt

Aufgabe 17: (*Matrizen und Arrays*)

Definieren Sie mit Hilfe des `numpy`-Moduls die (3×3) -Arrays

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

Definieren Sie außerdem einen Array `x` der Größe 3×1 gefüllt mit Einsen, einen Array `y` der Größe 1×3 mit den Einträgen 1, 2, 3 und einen eindimensionalen Array `z` der Größe 3 mit den Einträgen $-1, 0, 1$.

Welche der folgenden Operationen sind erlaubt und welche Ergebnisse sind mathematisch sinnvoll? Warum erzeugen manche Eingaben eine Fehlermeldung?

- (a) `A @ B` (d) `y * A` (g) `x @ A` (j) `z @ B` (m) `x * y` (p) `x * z` (s) `x @ z`
 (b) `A * B` (e) `A @ x` (h) `y @ A` (k) `B * z` (n) `y @ x` (q) `y * z` (t) `z @ y`
 (c) `A * y` (f) `A @ y` (i) `B @ z` (l) `z * B` (o) `x @ y` (r) `z @ x` (u) `y @ z`

Aufgabe 18: (*Arrays konstruieren*)

mögliche Befehle: `arange`, `zeros`, `len`, `hstack`, `vstack`, `reshape`, `T`, `transpose`, `flatten`, `kron`

- (a) Erstellen Sie einen Array `a` mit den Zahlen von 1 bis 9 (ohne die Zahlenfolge manuell einzutippen) und eine 2×2 Einheitsmatrix `I` ebenfalls vom Typ Array.

Wichtig: Lösen Sie die folgenden Teilaufgaben mit Ihren Arrays `a` und `I` (als Ganzes und nicht nur mit Teilen der Arrays), den oben genannten oder ähnlichen Befehlen und mathematischen Operationen. Jede Teilaufgabe soll ohne Speichern von Zwischenergebnissen in nur einer Zeile gelöst werden! Konstruieren Sie wie beschrieben die folgenden Matrizen:

(b) $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$

(d) $(1 \ 4 \ 7 \ 2 \ 5 \ 8 \ 3 \ 6 \ 9)$

(c) $\begin{pmatrix} 1 & 0 & 0 & 1 \\ 2 & 0 & 0 & 2 \\ \vdots & \vdots & \vdots & \vdots \\ 9 & 0 & 0 & 9 \end{pmatrix}$

(e) $\begin{pmatrix} 1 & 0 & 2 & 0 & 3 & 0 \\ 0 & 1 & 0 & 2 & 0 & 3 \\ 4 & 0 & 5 & 0 & 6 & 0 \\ 0 & 4 & 0 & 5 & 0 & 6 \\ 7 & 0 & 8 & 0 & 9 & 0 \\ 0 & 7 & 0 & 8 & 0 & 9 \end{pmatrix}$

(f) $\begin{pmatrix} 1 & 2 & 3 & 0 & 0 & 0 \\ 4 & 5 & 6 & 0 & 0 & 0 \\ 7 & 8 & 9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 4 & 5 & 6 \\ 0 & 0 & 0 & 7 & 8 & 9 \end{pmatrix}$

Aufgabe 19: (*Zugriff auf Untermatrizen*)

Matrizen lassen sich mit Hilfe von Listen in Listen auffassen lassen. Viel praktischer ist die Verwendung des `numpy`-Moduls. Erstellen Sie nun den folgenden (3×3) -Array

$$A = \begin{pmatrix} 1 & 0 & 2 \\ 1 & 9 & 4 \\ 3 & 2 & 5 \end{pmatrix}$$

und bestimmen Sie jeweils unter Ausnutzung Ihres Wissens über `numpy`-Arrays in einer möglichst kurzen Zeile den maximalen Eintrag

- (a) der ersten Zeile von `A`,
- (b) der zweiten Zeile von `A`,
- (c) der dritten Zeile von `A`,
- (d) der ersten Spalte von `A`,
- (e) der zweiten Spalte von `A`,
- (f) der dritten Spalte von `A`,
- (g) der Hauptdiagonalen von `A`,
- (h) von `A`.

Hierbei sollen die Aufrufe für jede Teilaufgabe so geschrieben werden, dass sie ohne großen Aufwand auf Arrays der Größe $n \times n$, $n \in \mathbb{N}$, angepasst werden können.

Aufgabe 20: (*Primitive Euler-Ziegel finden*)

Ein primitiver Euler-Ziegel ist wie in Aufgabe 14 definiert als ein Quader mit ganzzahligen Kantenlängen und ganzzahligen Flächendiagonalen, dessen drei Kantenlängen keinen gemeinsamen Teiler ungleich 1 haben. Schreiben Sie einen Python-Code, der sämtliche primitiven Euler-Ziegel mit Kantenlängen von höchstens 1000 findet. Den Versuch, von ChatGPT diese Aufgabe lösen zu lassen, finden Sie auf der Vorlesungsseite in der Datei `prim_euler.py`. Dort sind gute Ansätze erkennbar. Ihr Code soll in vertretbarer Zeit sämtliche primitiven Euler-Ziegel finden. Überlegen Sie sich gut, welche Fälle Sie wirklich überprüfen müssen, damit Sie Rechenzeit sparen.

Hinweis: Um den größten gemeinsamen Teiler zu bestimmen, dürfen Sie die Funktion `gcd` aus dem Paket `math` benutzen. Auch wenn es Ähnlichkeiten zu der Aufgabe 14 gibt, ist es nicht effizient, diese beiden Aufgaben zu vermischen. Sie können allerdings aus den Vorüberlegungen zu Aufgabe 14 profitieren.