Python – Übung 8

1 NumPy Arrays

☞ Erstellen Sie die folgenden NumPy-Arrays:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \boldsymbol{C} = \begin{bmatrix} 4 & 4 & 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 & 4 & 4 \end{bmatrix} \qquad \boldsymbol{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{E} = \begin{bmatrix} 1 & 2 & 3 & \cdots & 100 \end{bmatrix} \qquad \boldsymbol{F} = \underbrace{\begin{bmatrix} 0. & 0.01 & \cdots & 1. \end{bmatrix}}_{\text{len}(\boldsymbol{F})=101} \qquad \boldsymbol{G} = \begin{bmatrix} 100. & 99.5 & 99. & \cdots & 1. \end{bmatrix}$$

$$\boldsymbol{H} = \begin{bmatrix} \text{(10} \times \text{3) gleichverteilte} \\ \text{Zufallszahlen im Bereich } [0, 1) \end{bmatrix}, \text{ z.B. } \begin{bmatrix} 0.76738357 & 0.39524331 & 0.99835296 \\ \vdots & \vdots & \vdots \\ 0.35871752 & 0.44280981 & 0.01094926 \end{bmatrix}$$

$$\boldsymbol{J} = \begin{bmatrix} \text{(10} \times \text{3) normalverteilte} \\ \text{Zufallszahlen mit:} \\ \text{Standardabweichung } \sigma = 2.5 \\ \text{und Mittelwert } \mu = 9 \end{bmatrix}, \text{ z.B. } \begin{bmatrix} 6.52423225 & 11.66255392 & 8.47543261 \\ \vdots & \vdots & \vdots \\ 9.55731363 & 7.28759547 & 4.06290509 \end{bmatrix}$$

☞ Berechnen Sie die Dezibel-Werte des $\boldsymbol{E}$-Arrays, d.h. $\boldsymbol{E}_{\text{dB}} = 10 \log_{10} \boldsymbol{E}$.

☞ Berechnen Sie den Mittelwert des $\boldsymbol{H}$-Arrays.

☞ Berechnen Sie den quadratischen Mittelwert des $\boldsymbol{J}$-Arrays.

☞ Das Array $\boldsymbol{J}$ ist als Gruppe von zehn Vektoren im 3D-Raum zu interpretieren, d.h. jede Zeile repräsentiert ein einzelner 3D-Vektor. Addieren Sie zu jedem der zehn 3D-Vektoren den Vektor $\boldsymbol{r} = \begin{bmatrix} 1.5 & -9 & 0 \end{bmatrix}$ hinzu.

☞ Die Elemente eines NumPy-Arrays besitzen alle denselben Datentypen. Der Datentyp kann mit dem Attribut `dtype` ermittelt werden, z.B. `A.dtype`. Ermitteln Sie den Datentypen der Arrays $\boldsymbol{A}$ bis $\boldsymbol{J}$. Gibt es Unterschiede unter den Arrays? Wenn ja, wie kann man diese vermeiden?

2 Lineares Gleichungssystem

In einem Laden werden Äpfel und Birnen zu zwei unterschiedlichen Stückpreisen verkauft. Eine Person kauft 7 Äpfel und 4 Birnen und bezahlt total Fr. 6.50. Eine andere Person kauft 3 Äpfel und 5 Birnen und bezahlt total Fr. 5.25. Wieviel kostet ein Apfel bzw. eine Birne?

Daraus ergibt sich folgendes Gleichungssystem:

$$7x_1 + 4x_2 = 6.50 \tag{1}$$
$$3x_1 + 5x_2 = 5.25 \tag{2}$$

Dies kann auch als Matrizengleichung geschrieben werden:

$$\boldsymbol{M}\boldsymbol{x} = \boldsymbol{y} \; , \tag{3}$$

$$\begin{bmatrix} 7 & 4 \\ 3 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6.50 \\ 5.25 \end{bmatrix} \; . \tag{4}$$

Um die Matrizengleichung zu lösen, muss auf beiden Seiten von links her mit der Inversen $\boldsymbol{M}^{-1}$ multipliziert werden:

$$\boldsymbol{x} = \boldsymbol{M}^{-1}\boldsymbol{y} \tag{5}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 7 & 4 \\ 3 & 5 \end{bmatrix}^{-1} \begin{bmatrix} 6.50 \\ 5.25 \end{bmatrix} \; . \tag{6}$$

☞ Schreiben Sie ein Programm, welches die Matrizengleichung löst und somit den Apfel- und Birnenpreis ermittelt.

Hinweis: Benutzen Sie `np.linalg.solve()`[1].

3 Solarkollektor

Abb. 1 zeigt einen Solarkollektor. Es soll die Menge an Wärmeenergie berechnet werden, die von der Sonne abgestrahlt wird und auf den Solarkollektor während einer gewissen Zeit trifft. Die aktive Fläche des Solarkollektors ist $a \times b$ gross und deren Ausrichtung im Raum wird durch den Normalenvektor $\boldsymbol{n}$ angegeben. Der momentane Sonnenstand wird mit dem Einheitsvektor $\hat{\boldsymbol{s}}$ definiert, der vom Azimutwinkel (φ) und Höhenwinkel (θ) abhängig ist. Die Bestrahlungsstärke E gibt an, wieviel Leistung auf eine Oberfläche trifft, bezogen auf die Grösse der Fläche.

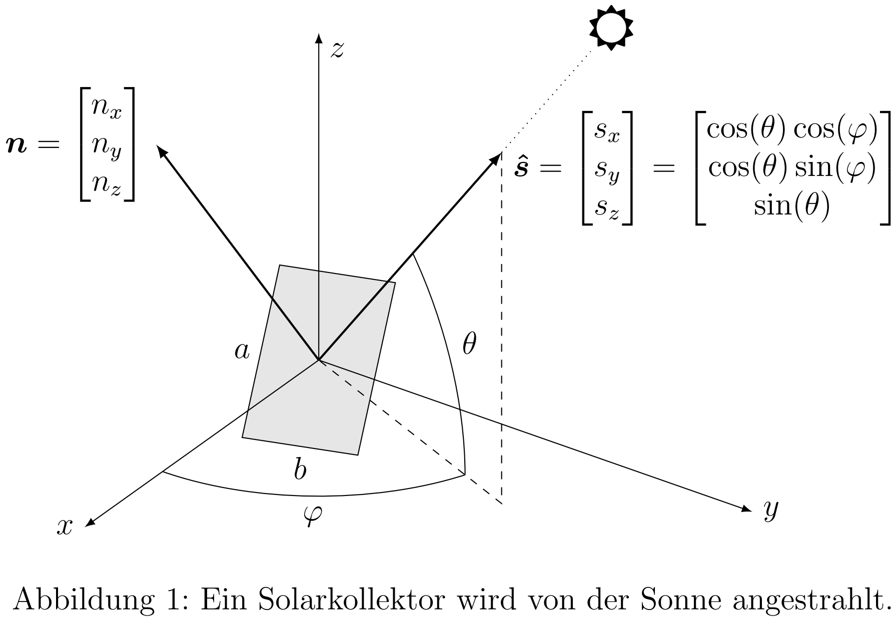

Abbildung 1: Ein Solarkollektor wird von der Sonne angestrahlt.

[1] `https://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.solve.html`

Die kumulierte Energie W, die während einer gewissen Zeit auf den Solarkollektor trifft, kann wie folgt berechnet werden:

$$W = abET \sum_{k=0}^{N-1} p_k \,, \tag{7}$$

$$p_k = \begin{cases} \hat{\boldsymbol{s}}_k \cdot \dfrac{\boldsymbol{n}}{\|\boldsymbol{n}\|} & \text{falls } \hat{\boldsymbol{s}}_k \cdot \dfrac{\boldsymbol{n}}{\|\boldsymbol{n}\|} > 0 \\ 0 & \text{sonst} \end{cases} \,, \tag{8}$$

wobei T der Zeitabstand zwischen den einzelnen Messungen des Sonnenstandes ist und $\|\boldsymbol{n}\| = \sqrt{n_x^2 + n_y^2 + n_z^2} = \sqrt{\boldsymbol{n} \cdot \boldsymbol{n}}$ der Betrag des Vektors $\boldsymbol{n}$ bezeichnet.

☞ Implementieren Sie die Funktion `solar_energy(a, b, n, T, E, phi, theta)`, welche die kumulierte Energie W (wie oben definiert) berechnet und als `float`-Zahl zurückgibt.

☞ Berechnen Sie mittels der obigen Funktion die Energie W, welche auf den Solarkollektor während einer bestimmten Zeit trifft. Der Solarkollektor hat die Masse $a = 2$ und $b = 1.5$ und seine Ausrichtung ist $\boldsymbol{n} = [0.5, 2, 10]$. Der Sonnenstand während der Messzeit von 12 Stunden ist in Abb. 2 dargestellt. Während der gesamten Messzeit herrscht eine konstante Bestrahlungsstärke von $E = 1000$. Wählen Sie eine Zeitauflösung von $T = 1/3600\,\mathrm{h}$, d.h. der Zeitabstand zwischen den Messungen ist eine Sekunde.

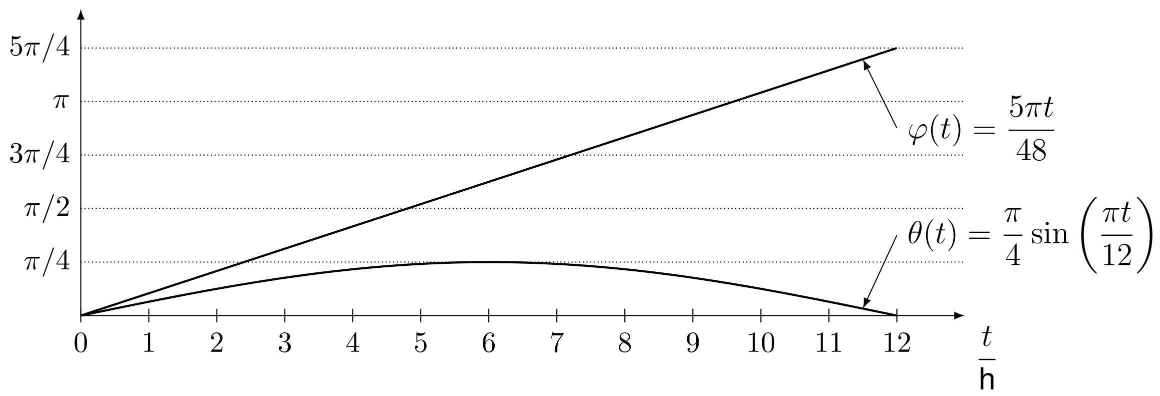

Abbildung 2: Azimuth- (φ) und Höhenwinkel (θ) während der Messzeit.

Hinweis: Benutzen Sie für die Lösungen geeignete NumPy-Funktionen und verzichten Sie gänzlich auf `for`-Schleifen und (List-)Comprehensions.