Serie 8.

Woche: 09.11 – 12.11.2026

Aufgabe 1 (Verschiedene Lösungswege | 1 Punkt).

In dieser Aufgabe sehen wir, dass es für eine Problemstellung ganz verschiedene Lösungen geben kann. Dazu wollen wir uns anschauen, wie man alle geraden Zahlen von 1 bis 100 aufsummieren kann.

a) Verwende die Gaußsche Summenformel, d.h. es gilt:

$$\sum_{k=1}^{n} 2k = n(n+1).$$

b) Verwende eine `for`-Schleife.

c) Verwende eine `while`-Schleife.

d) Verwende den Matlab-Befehl `sum`.

e) Verwende eine geschickte Multiplikation mit einem `ones`-Vektor.

f) Du darfst dir gerne noch weitere Möglichkeiten überlegen.

Aufgabe 2 (Glücksspiel | 2 Punkte).

Betrachte folgendes Glücksspiel: Zu Beginn zahlst du dem Veranstalter einmal einen Einsatz von 10 Franken. Der Veranstalter wirft nun so lange mit einer fairen Münze, bis zum ersten Mal Kopf erscheint. Sei k die Anzahl der dazu notwendigen Versuche. Der Veranstalter zahlt dir nun 2^{k-1} Franken aus.

(a) Simuliere dieses Spiel in einem Matlab-Programm. Um den einzelnen Münzwurf zu simulieren, kannst du den Befehl `randi` benutzen. Verwende eine `while`-Schleife, welche die Anzahl der Versuche k zählt und berechne schliesslich Auszahlung und Gewinn.

(b) Nun wiederhole das Spiel 10^7 mal (`for`-Schleife). Zeichne sowohl die ausbezahlte Summe pro Spiel als auch den Gesamtgewinn in eine Abbildung. Dabei entspricht der Gesamtgewinn vom i-ten Spiel dem addierten Gewinn der Spiele 1 bis i. Was siehst du?

Aufgabe 3 (Skalarprodukt | 3 Punkte).

Häufig führt die Benutzung von vielen `for`-Schleifen in Matlab zu einem Zeitverlust. Deshalb ist es vorteilhaft, wenn man gewisse Schleifen durch Operationen mit Vektoren ersetzt.

(a) Das Skalarprodukt zweier Spaltenvektoren $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ ist definiert als

$$\langle \mathbf{x}, \mathbf{y} \rangle := \mathbf{x}^\top \mathbf{y} = \sum_{i=1}^{n} x_i y_i.$$

Schreibe eine Funktion `xy = ScalarProductFor(x,y)`, welche das Skalarprodukt zweier allgemeiner Spaltenvektoren $\mathbf{x}, \mathbf{y}$ der Länge n berechnet. Teste es für

$$\mathbf{x}, \mathbf{y} \in \mathbb{R}^{11} \quad \mathbf{x} = (1, 1.1, 1.2, \ldots, 1.9, 2)^\top, \quad \mathbf{y} = (-1, 1, -1, , \ldots, 1, , -1)^\top,$$

(Resultat: -1.5).

(b) Schreibe eine Funktion `xy = ScalarProduct(x,y)`, welches das Skalarprodukt aus Aufgabe (i) berechnet, aber dieses Mal ohne eine `for`- oder `while`-Schleife oder die Befehle `dot` und `sum` zu verwenden. Teste auch hier für die beiden Vektoren aus (i), ob dein Programm das Richtige tut.

(c) Mit den Befehlen `tic` und `toc` kann man in Matlab die Zeit für den Durchlauf eines Programms messen. Zum Beispiel wird für

```
tic
(1:1000).^2
time = toc
```

die Zeit in Sekunden gemessen, wie lange es dauert, den Vektor $(1, 2, \ldots, 1000)$ eintragsweisse zu quadrieren. Die verstrichene Zeit wird in `time` gespeichert.

Vergleiche mit Hilfe dieser Befehle die Laufzeiten der beiden Programme zur Berechnung des Skalarprodukts (d.h. einmal mit und einmal ohne `for`-Schleife) für die Vektoren $\mathbf{x} = (1, 2, \ldots, n)^\top$ und $\mathbf{y} = 3\mathbf{x}$, wobei $n = 10^3, 10^4, \ldots, 10^8$, indem du die Zeiten für die verschiedenen n in eine Abbildung zeichnest. Benutze den Befehl `semilogx` und die Plot-Optionen `'-o'` und `'-*'`. Füge einen Titel, Achsenbeschriftungen und eine Legende hinzu, welche die Grafik sinnvoll beschreiben. Was stellst du fest?