



Serie 5.

Woche: 19.10 – 22.10.2026

Aufgabe 1 (Vektornormen | 2 Punkte).

- (a) Schreibe eine Funktion `n = vectorNorm2(x)` welche die euklidische (oder l_2 -) Norm eines Spaltenvektors $\mathbf{x} \in \mathbb{R}^n$ nach folgender Definition berechnet:

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}.$$

- (b) Schreibe eine Funktion `n = vectorNormInf(x)` welche die Maximum-Norm (oder l_∞ -Norm) eines Vektors $x \in \mathbb{R}^n$ berechnet:

$$\|\mathbf{x}\|_\infty = \max_i |x_i|.$$

Verwende dabei eine for-Schleife und benutze keine eingebauten Befehle wie `sum`, `max` oder `norm`. Teste deine Funktion für $\mathbf{x} = (1, 2, 3, 4)^\top$ und vergleiche das Ergebnis mit den Matlab-Befehlen `norm(x)` und `norm(x, inf)`.

Hinweis: Der Spaltenvektor $\mathbf{x} = (1, 2, 3, 4)^\top$ kann in MATLAB mit `x = [1; 2; 3; 4]` definiert werden. Auf das i -te Element eines Vektors $\mathbf{x}$ kann in MATLAB mit `x(i)` zugegriffen werden.

Aufgabe 2 (Fibonacci-Zahlen | 2 Punkte).

Die Fibonacci-Zahlen F_n , $n \geq 0$, sind nach dem italienischen Mathematiker Leonardo Pisano (Spitzname Fibonacci, 1170–1250, Pisa) benannt. Sie sind rekursiv gegeben durch

$$\begin{aligned} F_0 &= 0, \\ F_1 &= 1, \\ F_n &= F_{n-1} + F_{n-2}, \quad n \geq 2. \end{aligned}$$

- (a) Benutze eine for-Schleife, um die 35. Fibonacci-Zahl zu berechnen. Schreibe dazu ein Skript.
- (b) Schreibe eine MATLAB-Funktion `function Fn = Fibonacci(n)`, die n als Eingabe nimmt und F_n ausgibt. Was ist F_{35} ?
- (c) Leonhard Euler hat 1765 eine explizite Formel für die n -te Fibonacci-Zahl gefunden. Diese berechnet sich wie folgt:

$$\begin{aligned} F_n &= \frac{1}{\sqrt{5}} (\varphi^n - \hat{\varphi}^n), \quad \text{mit} \\ \varphi &= \frac{1 + \sqrt{5}}{2} \quad (\text{der Goldene Schnitt}), \\ \hat{\varphi} &= -\frac{1}{\varphi}. \end{aligned}$$

Schreibe eine MATLAB-Funktion `function F_n = Fibonacci_exact(n)`, die für die Eingabe n die zugehörige Fibonacci-Zahl exakt berechnet. Kontrolliere dann, ob deine Programme in (a) und (b) richtig funktionieren.

- (d) Alle Variablen, welche im Command Window von MATLAB definiert sind, werden im Fenster “Workspace” aufgelistet. Beobachtest du einen Unterschied beim Ausführen der Programme in (a) bzw. (b)? Hierfür ist es hilfreich, jede Teilaufgabe in einem eigenen Skript zu schreiben bzw. die Funktionen aufzurufen und mit `clear` jeweils zuerst alle Variablen zu löschen.

Aufgabe 3 (HOTPO-Folge | 2 Punkte).

Während `for`-Schleifen hauptsächlich verwendet werden, wenn schon im Voraus bekannt ist, wieviele Schleifendurchläufe notwendig sind, werden in den anderen Fällen meist `while`-Schleifen verwendet.

- (a) Schreibe eine Funktion `function NumberOfIterations = HOTPO(a_0)`, welche die HOTPO-Folge (Half Or Triple Plus One)

$$a_{n+1} = \begin{cases} a_n/2, & \text{falls } a_n \text{ gerade,} \\ 3a_n + 1, & \text{falls } a_n \text{ ungerade,} \end{cases}$$

solange berechnet, bis $a_m = 1$ ist. Der Startwert $a_0 \in \mathbb{N}$ wird der Funktion als Parameter übergeben. Die Funktion soll den ersten Index m mit $a_m = 1$ zurückgeben. Teste dein Programm für verschiedene Startwerte $a_0 \in \mathbb{N}$.

Hinweis: Der Befehl `mod` ist hilfreich.

- (b) Ändere deine Funktion zu `function NumberOfIterations = HOTPO(a_0, flag)` ab. Der zusätzliche Parameter `flag` soll den Wert 0 oder 1 haben. Für `flag = 1` soll die erzeugte HOTPO-Folge ausgegeben werden, für `flag = 0` nicht.

Schreibe zusätzlich ein Skript, das die Funktion für einen gegebenen Startwert a_0 aufruft und sowohl die Anzahl der Iterationen als auch, für `flag = 1`, die erzeugte Zahlenfolge ausgibt.