Praktikum: "Mathematik am Computer" HS 2023
Prof. H. Harbrecht
M. Fallahpour, E. Gajendran, G. Ohlmann, M. Ramm Universität Basel

---

Serie 7

Matlab - Kontrollstrukturen III (while), Funktionen

zur 45. KW (6.11. – 9.11.2023)

Aufgabe 7.1 (1 Punkt): Schau dir die Folien genau an und löse dann folgende Aufgabe von Hand. Was würde Matlab für folgendes Code-Stück ausgeben?

```
x = 1;
s = 1;
while x<=4
     s = s*x;
     x = x+1;
end
s
```

Aufgabe 7.2 (1 Punkt): Schreibe die Produkte aus Aufgabe 5.3 a)–c) als Matlab-Funktionen, z.B. die Matrix-Multiplikation `function C = MatrixMult(A,B)`, die beliebige Matrizen **A** und **B** als Input-Variablen erhält und $\mathbf{C} = \mathbf{AB}$ ausgibt. Teste die Funktionen an den Beispielen aus Aufgabe 5.3. Dabei sollen die Funktionen so geschrieben sein, dass sie für allgemeine Matrizen bzw. Vektoren gültig sind. Dazu sind die Befehle `length` und `size` nützlich.

Aufgabe 7.3 (3 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

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

a) Benutze eine `for`-Schleife, um die 20. Fibonacci-Zahl ($F_{20} = 6765$) zu berechnen. Schreibe dazu ein Skript.

b) Schreibe eine MATLAB-Funktion `function F_n = Fibonacci(n)`, die n als Eingabe nimmt und F_n ausgibt. Was ist F_{40}?

c) Leonhard Euler hat 1765 eine explizite Formel für die n-te Fibonacci-Zahl gefunden. Diese berechnet sich wie folgt:

$$F_n = \frac{1}{\sqrt{5}} \left(\phi^n - \hat{\phi}^n \right), \quad \text{wobei} \quad \phi = \frac{1+\sqrt{5}}{2} \quad \text{und} \quad \hat{\phi} = -\frac{1}{\phi}.$$

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.
Bemerkung: Werden die Resultate aus c) von den Resultaten aus a), b) substrahiert, ist das Ergebnis klein, aber ungleich 0. Das ist bedingt durch vom Computer verursachte Rundungsfehler in der Auswertung der Funktion `Fibonacci_exact(n)`.

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, vor dem Skript- bzw. Funktionsaufruf die Funktion `clear` aufzurufen, um jeweils zuerst alle bereits gespeicherten Variablen zu löschen.

Aufgabe 7.4 (5 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)

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

solange berechnet und anzeigt, bis $a_m = 1$ ist. Der Startwert $a_0 \in \mathbb{N}$ wird der Funktion als Parameter übergeben. Diese 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 `HOTPO` so, dass sie einen zusätzlichen Parameter "Flag" als Funktionsargument erhält. Dieser Parameter akzeptiert zwei Werte:

- Wert 0 ("Fahne runter"): Die Funktion erzeugt die Zahlenfolge, aber sie wird nicht ausgegeben.
- Wert 1 ("Fahne hoch"): Die Funktion gibt die erzeugte Zahlenfolge aus.

c) Schreibe ein Skript, welches für die Anfangswerte $a_0 = 1, \ldots, 1000$ die Anzahl der benötigten Iterationen zeichnet.

Bemerkung: Weitere Einblicke in die HOTPO-Sequenz bietet das YouTube-Video "The Simplest Math Problem No One Can Solve – Collatz Conjecture" vom Kanal "Veritasium", das die Thematik anschaulich aufbereitet.
