Abteilung für
Angewandte Mathematik

SoSe 2019
Christian Palus, Steve Wolff-Vorbeck

# Einführung in die Programmierung für Studierende der Naturwissenschaften

Blatt 9 – 01.07.2019

Abgabe: Briefkästen RZ/E-Mail bis Montag, den 08.07.2019, 16:00 Uhr

---

**Aufgabe 1** (2+2+1 Punkte)**.** Das aus der Vorlesung bekannte Knobelspiel „Die Türme von Hanoi" geht vermutlich auf den französischen Mathematiker Édouard Lucas (1883) zurück. Er überlegte sich dazu die Geschichte, dass indische Mönche im großen Tempel zu Benares, im Mittelpunkt der Welt, einen Turm aus 64 goldenen Scheiben versetzen müssen. Wenn ihnen das gelungen sei wäre das Ende der Welt gekommen.
(i) Wie viele Scheiben müssen in dem Algorithmus zur Lösung des Knobelspiels mit $n \geq 1$ versetzt werden? Begründen Sie ihre Amtwort.
(ii) Begründen Sie warum diese Anzahl an Versetzungen optimal ist, d.h. kein schnelleres Verfahren existieren kann, das mit weniger Versetzungen auskommt.
(iii) Wie viele Jahre bräuchten die indischen Mönche zur Lösung des Spiels mit 64 Scheiben, wenn sie pro Sekunde eine Scheibe versetzen?

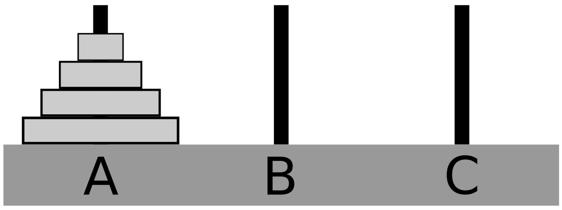


Abbildung 1. Knobelspiel „Die Türme von Hanoi" mit $n = 4$ Scheiben in der Ausgangsstellung.

**Aufgabe 2** (5 Punkte)**.** Die Räuber-Beute-Gleichungen sind ein System aus gekoppelten, nichtlinearen Differentialgleichungen und beschreiben die Wechselwirkung einer Räuber- und Beutepopulation. Sie sind gegeben durch:

$$\begin{aligned} B'(t) &= B(t)(\epsilon_B - \gamma_B R(t)), \\ R'(t) &= -R(t)(\epsilon_R - \gamma_R B(t)). \end{aligned}$$

Dabei beschreiben $B', R'$ die zeitlichen Änderungen der Beutepopulation $B$ sowie der Räuberpopulation $R$. Dabei sind die Reproduktionsrate $\epsilon_B$ der Beute ohne Räuber, die Sterberate $\epsilon_R$ der Räuber ohne Beute, die Sterberate $\gamma_B$ der Beute pro Räuber sowie die Reproduktionsrate $\gamma_R$ der Räuber pro Beutelebewesen gegebene Parameter. Gesucht sind in diesem Modell jeweils die Anzahl an Räubern $R(t)$ sowie die Anzahl an Beute $B(t)$ zu einem Zeitpunkt $t$, wobei die Anfangspopulationen $R(0), B(0)$ gegeben sind. Eine Möglichkeit die Ableitungen $B'(t), R'(t)$ approximativ zu berechnen ist die Verwendung des Differenzenquotienten

$$B'(t) \approx \frac{B_{i+1} - B_i}{\Delta t}$$

mit gegebener Zeitschrittweite $\Delta t$, wobei $B_i = B(i\Delta t)$. Bringen Sie mit Hilfe des Differenzenquotienten die Gleichungen auf die Form

$$B_{i+1} = B_i + \Delta t F_1(B_i, R_i, t_i),$$
$$R_{i+1} = R_i + \Delta t F_2(B_i, R_i, t_i).$$

Formulieren Sie darauf aufbauend einen Algorithmus (Pseudo-Code) zur approximativen Berechnung der Populationen zu einem Zeitpunkt $t = i\Delta t$.

**Aufgabe 3** (2+2+1 Punkte)**.** (i)Verwenden Sie den Befehl `plot()`, um die Graphen der folgenden Funktionen zu zeichnen:

$$f_1(x) = \log(x), \quad f_2(x) = x, \quad f_3(x) = x^2, \quad f_4(x) = e^x.$$

Dabei sollen alle Graphen im selben Koordinatensystem dargestellt werden. Benutzen Sie die Befehle `xlabel()` und `ylabel()`, um die Koordinatenachsen zu beschriften, den Befehl `xlim()`, um den dargestellten Bereich der $x$-Achse auf einen sinnvollen Wert zu ändern, sowie den Befehl `legend()`, um eine Legende der Funktionsgraphen zu erstellen. Geben Sie einen Ausdruck des Plots sowie der von Ihnen verwendeten Befehle ab.
(ii) Plotten Sie sowohl den Graphen als auch die Höhenlinien der Funktion

$$(x, y) \mapsto (x^2 + y^2)^{\frac{1}{2}} + \sin(x^2 + y^2) + \cos(x^2)$$

für $-3 \leq x, y \leq 3$ mit Hilfe der Matlab-Routinen `surf` und `contour`.
(iii) Laden Sie mit dem Befehl `A=load(temperaturen.txt)` die Jahresdurchschnittstemperaturen von Blatt 7. Erstellen Sie einen Plot, in welchem Sie die Temperaturen ($y$-Achse) gegen die Jahre ($x$-Achse) plotten. Nutzen Sie dabei den Befehl `A(:,1)` bzw. `A(:,2)` um auf die beiden Spalten der Matrix $A$ zuzugreifen.

**Aufgabe 4** (5 Punkte)**.** Eine Variable `x` vom Typ **`double`** kann in C++ mit Hilfe der Zuweisung `x=double(rand())/RAND_MAX;` mit einer Zufallszahl zwischen $0$ und $1$ belegt werden. Schreiben Sie ein Programm (in C++), welches eine vorgegebene Anzahl $N$ an Paaren $(x_n, y_n)_{1\leq n\leq N}$ von Zufallszahlen erzeugt und für jedes Zahlenpaar prüft, ob der Punkt $(x_n, y_n)$ innerhalb des Viertelkreises

$$\{(x, y) : 0 \leq x,\, 0 \leq y,\, x^2 + y^2 \leq 1\}$$

liegt. Der Anteil der Punkte innerhalb des Viertelkreises liefert für große $N$ eine Approximation an $\pi/4$. Für $1 \leq n \leq N$ bezeichne $\pi_n$ das $4-$fache des Anteils der $n$ Punkte $(x_i, y_i)_{i\leq n}$, die innerhalb des Viertelkreises liegen. Wählen Sie für $N$ nacheinander die Werte $10^3$, $10^4$, $10^5$ und $10^6$ und speichern Sie die Werte $(n, \pi_n)_{1\leq n\leq N}$ aus Ihrer Berechnung jeweils in eine Datei. Lesen Sie diese Dateien anschließend mit dem Befehl `load('DATEINAME')` in Matlab ein und erstellen Sie jeweils einen Plot der Daten, der die Konvergenz $\pi_n \to \pi$ veranschaulicht.