



## Einführung in die Programmierung für Studierende der Naturwissenschaften

Blatt 11 – 23.07.2020

Abgabe: Bis Donnerstag, 30.07.2020, 12 Uhr, per E-Mail an Ihren Tutor.

Notwendiges Wissen: Kursskript bis einschl. Kapitel 6

---

*Hinweis: Alle Aufgaben auf diesem Blatt sind Bonusaufgaben.*

**Aufgabe 1** (5 Bonuspunkte). Schreiben Sie ein C++-Programm, das ein Sudoku einliest und in einem Array

```
unsigned int Sudoku [9] [9];
```

speichert. Das einzulesende Sudoku soll in einer Text-Datei gespeichert sein. Jede Zeile soll eine Zeile des Sudokus enthalten. Wenn eine Zeile Leerzeichen enthält, sollen diese ignoriert werden. Wenn eine Zeile ein # - Zeichen enthält, soll alles, was in der Zeile auf das Zeichen folgt, ignoriert werden. Sie dürfen annehmen, dass die Gesamtzahl der Einträge korrekt ist, also dass keine Zeile zu wenig oder zu viele Zahlen enthält. Eine Beispieldatei `sudoku.txt` finden Sie auf der Vorlesungswebseite.

**Aufgabe 2** (5 Bonuspunkte). Schreiben Sie ein C++-Programm, das ein ausgefülltes Sudoku-Rätsel auf Richtigkeit überprüft. Ein Sudoku können Sie in einem zweidimensionalen Array

```
unsigned int Sudoku [9] [9];
```

speichern. Die Sudoku-Regeln finden Sie unter <https://de.wikipedia.org/wiki/Sudoku>. Für diese Aufgabe müssen Sie das Sudoku nicht aus einer Datei einlesen. Sie können es entweder direkt im Code speichern oder Ihre Lösung aus Aufgabe 1 verwenden.

**Aufgabe 3** (5 Bonuspunkte). Das Bisektionsverfahren zur Approximation einer Nullstelle einer gegebenen Funktion  $f: \mathbb{R} \rightarrow \mathbb{R}$  ist wie folgt definiert:

- (1) Wähle eine Genauigkeit  $\varepsilon > 0$  und zwei Startwerte  $a_0, b_0 \in \mathbb{R}$ , sodass  $a_0 < b_0$  und  $f(a_0)$  und  $f(b_0)$  unterschiedliches Vorzeichen haben. Setze  $k = 0$ .
- (2) Falls  $b_k - a_k < \varepsilon$ , so ist  $[a_k, b_k]$  das gesuchte Lösungsintervall: Breche das Verfahren mit Ausgabe  $\frac{a_k + b_k}{2}$  ab.
- (3) Falls  $f(\frac{a_k + b_k}{2})$  das gleiche Vorzeichen wie  $f(a_k)$  hat, setze  $a_{k+1} = \frac{a_k + b_k}{2}$  und  $b_{k+1} = b_k$ . Andernfalls setze  $a_{k+1} = a_k$  und  $b_{k+1} = \frac{a_k + b_k}{2}$ .
- (4) Erhöhe  $k$  um 1 und gehe zu (2).

Implementieren Sie dieses Verfahren in MATLAB als Funktion

```
function nst = bisektion(f,a,b,eps),
```

wobei die als Parameter übergebene Funktion  $f$  ein sogenannter *function handle* ist, d.h. eine Variable, welche den Verweis auf eine Funktion speichert. Die Syntax zur Definition solcher

auch als *anonyme Funktion* bezeichneter Variablen ist dabei wie im folgenden Beispiel, in dem die Funktion  $f(x)$  den Wert von  $x^2 - 1$  berechnet:

```
f = @(x) x*x-1;
```

Der Rückgabewert `nest` Ihrer Funktion sollte der Mittelpunkt des Lösungsintervalls sein. Testen Sie Ihr Programm, indem Sie damit die Nullstelle der Funktion  $x \mapsto x^3 - 2$  approximieren.

**Aufgabe 4** (5 Bonuspunkte). Implementieren Sie das *Mergesort* Verfahren (siehe Algorithmus 5.10 im Kursskript) in MATLAB.