



#### **Aufgabe 4—1** (Quellcodebereinigung)

Laden Sie sich den zum Übungsblatt gehörigen Quellcode und die zugehörigen Dateien von der Internetseite<sup>1</sup> des Methodenkurses herunter und öffnen Sie ihn in MATLAB mit dem MATLAB-Editor.

Wie Sie sehen, haben Sie es mit unstrukturiertem, undokumentiertem „Spaghetti-Code“ zu tun. Nutzen Sie zunächst die MATLAB-internen Möglichkeiten der Code-Formatierung (automatische Einrückung) sowie die Hinweise zur Formatierung von Quellcode aus der Lösung zum letzten Aufgabenblatt und beseitigen Sie in einem zweiten Schritt alle Warnungen, die Ihnen der MATLAB-Editor anzeigt.

#### **Aufgabe 4—2** (Quellcode gliedern und dokumentieren)

Versuchen Sie, das zum Übungsblatt gehörende Code-Beispiel in allen Details zu verstehen, gliedern Sie den Code gemäß der einzelnen Aufgaben in Blöcke und dokumentieren Sie im Code in wenigen Worten, was in welchem Block gemacht wird.

#### **Aufgabe 4—3** (Quellcode in Funktionen aufteilen)

Überlegen Sie, wie Sie die einzelnen Teile des Ihnen als Beispiel gegebenen Quellcodes sinnvoll in Funktionen aufteilen können<sup>2</sup>. Entwerfen Sie die einzelnen Funktionen, implementieren Sie sie und schreiben Sie das Codebeispiel entsprechend um, so dass die einzelnen Funktionen aufgerufen werden.

Achten Sie beim Design Ihrer Funktionen darauf, möglichst allgemeingültigen und wiederverwendbaren Code zu schreiben. Verwenden Sie sprechende Namen für Ihre Funktionen und achten Sie auf nutzerfreundliche Schnittstellen.

Jede Ihrer Funktionen sollte mit einem sinnvollen und für den Nutzer hilfreichen Kommentarkopf ausgestattet sein, der durch den MATLAB-Befehl `help` aufgerufen werden kann.

#### **Aufgabe 4—4** (Bisher geschriebene Funktionen dokumentieren)

Versehen Sie die bisher von Ihnen erstellten Funktionen `gaussian` und `rotationMatrix` mit einem sinnvollen und für den Nutzer hilfreichen Kommentarkopf, der durch den MATLAB-Befehl `help` aufgerufen werden kann.

---

<sup>1</sup><https://www.till-biskup.de/de/lehre/matlab/ss2019/material/06/>

<sup>2</sup>Folgen Sie dabei der Unix-Philosophie: „eine Aufgabe, eine Funktion“ und achten Sie darauf, dass Sie die Aufgaben nicht zu weit, aber auch nicht zu eng fassen.