

# Leitfaden zur Bewertung und Abgabe der Übungsaufgaben in Numerik

Prof. M. Grote, Y. Gleichmann, C. Santos

## 1 Theoretische Aufgaben

Pro Aufgabe ist maximal 1 Punkt erreichbar. Es wird zwischen 1,  $\frac{1}{2}$  und 0 Punkten wie folgt unterschieden:

(i) 1 Punkt bei

- korrekter Bearbeitung der Aufgabe,
- falls die Beweisidee stimmt,
- bei kleinen Flüchtigkeitsfehlern, z.Bsp. Vorzeichenfehler.

(ii)  $\frac{1}{2}$  Punkte falls

- Ansätze der korrekten Beweisidee vorhanden sind,
- Flüchtigkeitsfehler, die Rechnungen oder den Beweis erheblich vereinfachen,
- grobe Fahrlässigkeit, z.Bsp. das "finden" einer Wohlordnung auf den komplexen Zahlen ( $i < \pi$ ).

(iii) 0 Punkte bei

- Nichtbearbeitung,
- Abgabe eines offensichtlich falschen Beweises oder Rechenwegs,
- Wiederholung von Fehlern aus früheren Übungsaufgaben,
- grossen Unklarheiten oder einem verwirrenden Beweisstil.

## 2 Programmieraufgaben

Pro Aufgabe ist maximal 1 Punkt erreichbar. Es wird zwischen 1,  $\frac{1}{2}$  und 0 Punkten wie folgt unterschieden:

(i) 1 Punkt falls

- das Programm fehlerfrei ausführbar ist und das korrekte Ergebnis erzeugt wird, vorausgesetzt es entspricht dem Matlab *style guide*, siehe unten,
- oder das Programm ohne Fehlermeldungen ausführbar ist und es keine signifikante Abweichung zum korrekten Ergebnis gibt, vorausgesetzt es entspricht dem Matlab *style guide*, siehe unten.

(ii)  $\frac{1}{2}$  Punkt falls

- die Kriterien unter (i) erfüllt sind, sich jedoch nicht (zur Gänze) am *style guide* gehalten wurde.

(iii) 0 Punkte falls

- die Aufgabe nicht bearbeitet wurde,
- das Programm Warnungen oder Bugs ausgibt,
- das korrekte Ergebnis in keinster Weise ersichtlich ist,
- der *style guide* missachtet wurde.

## 3 Matlab 'style guide'

### 3.1 Kommentare

fügen zusätzliche wichtige Informationen hinzu, zum Beispiel um den Code zu erklären oder zu kommentieren, wo man etwas verbessern könnte. Es ist von Vorteil, die Kommentare während des Programmierens hinzuzufügen. Triviale Codezeilen sollten nicht kommentiert werden, dies erhöht nicht die Lesbarkeit:

```
m = 1; % setze m auf 1;
```

Jede nichttriviale Funktion sollte eine Beschreibung haben wie z.Bsp. Input, Output und eine grobe Beschreibung des Codes.

Der Autor und das Datum des Codes sollten in jeder Funktion enthalten sein. Falls es nicht Ihr Code ist, Sie aber trotzdem ihren Namen als Autor eintragen, ist dies ein Plagiat.

```
author: Yannik Gleichmann
```

```
last modified: 21.06.1992
```

### 3.2 Plots

sind ein sehr wichtiger Bestandteil bei der Abgabe von Programmieraufgaben. Falls das Programm einwandfrei läuft, Plots jedoch nicht korrekt erstellt werden, gibt es nur  $\frac{1}{2}$  Punkt.

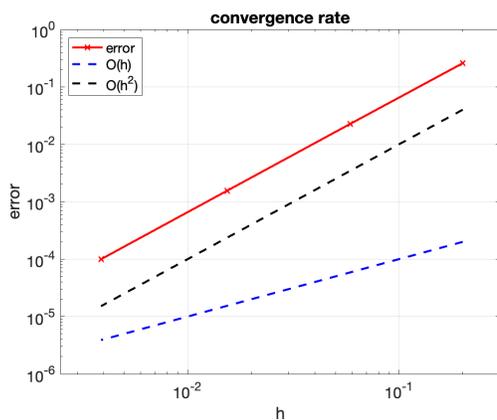
Plots brauchen zwingend

- einen Titel, (lesbare) Achsenbeschriftung und eine sinnvolle Skalierung.

Zudem auch noch

- eine Legende (falls nötig), eine genügend grosse Schriftart und Marker, um Datenpunkte zu markieren. Dies aber nur, wenn es nicht zu viele Datenpunkte sind.

**Gut:**



**Schlecht:**

