



Praktische Übungen zu Numerik I

Projekt 1 – 17.10.2022

Abgabe: per E-Mail bis Freitag, den 28.10.2022, 10:00 Uhr

Homepage zur Vorlesung:

<https://aam.uni-freiburg.de/agsa/lehre/ws22/num>

Hinweis: Der Praktikumsbetrieb findet in den ungeraden Semesterwochen statt und beginnt am 20.10. um 10 Uhr. Die Anmeldung zu den Praktikumsgruppen über HISinOne beginnt am Mittwoch den 19.10. um 10 Uhr.

Wir arbeiten in diesem Praktikum vor allem mit Matlab, da dies die einfache Implementierung von numerischen Algorithmen und graphischen Darstellung von Daten erlaubt. Matlab steht Ihnen auf den Rechnern im CIP-Pool zur Verfügung und eine Anleitung zum Arbeiten mit Matlab von zu Hause ist auf der Website der Vorlesung verlinkt.

Wenn Sie wollen, können Sie Ihre Programme aber auch in C++ oder in Python abgeben. Eine Einführung in C++ finden Sie unten auf der Korrekturseite des Buches 'Numerik 3x9', die auch auf der Website der Vorlesung verlinkt ist.

Für die Programme in Python empfehlen wir die Verwendung des Lineare Algebra Pakets `numpy`.

Erstellen eines ersten Programms. Das Programmpaket Matlab enthält eine Vielzahl numerischer Methoden und man kann eigene Programme schreiben. Wenn Sie Matlab starten, öffnet sich die Konsole, in der sie direkt mathematische Ausdrücke eingeben können, beispielsweise

```
1 >> A = [2, 1; 3, 4; 4, 2];
2 >> x = [1; 0];
3 >> A * x
```

Dieses Programm entspricht dem folgenden Matrix-Vektor-Produkt:

$$\begin{pmatrix} 2 & 1 \\ 3 & 4 \\ 4 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

A definiert also eine Matrix und x einen Vektor, wobei Einträge einer Zeile mit einem Komma und Zeilen mit einem Semikolon getrennt werden. Wenn am Ende einer eingegebenen Zeile kein Semikolon steht, so wird das definierte Objekt beziehungsweise das Resultat einer Operation in der Konsole ausgegeben.

Ein äquivalentes Programm in Python mit `numpy` könnte wie folgt aussehen:

```

1 import numpy as np
2 A = np.array([[2, 1], [3, 4], [4, 2]])
3 x = np.array([1, 0])
4 print(A @ x)

```

Hier entspricht also der Operator @ dem Matrix-Vektor-Produkt. Würde man den Operator * verwenden, so wäre das Resultat verschieden.

Welche Operation würde bei $A * x$ in Python ausgeführt werden?

Projekt 1 (2 Punkte). Zur Bestimmung der Rundungsgenauigkeit eines Rechners initialisieren wir $x = 1$ und ersetzen x durch $x/2$, solange der Ausdruck $1 + x > 1$ vom Rechner als wahr ausgewertet wird. Schreiben Sie ein Programm, das experimentell den Wert von x , für den dieses Vorgehen abbricht, bestimmt. Definieren Sie x dabei als Variable mit single precision bzw. double precision.

Hinweis: In Matlab und Python sind Numerische Variablen, bei denen der Typ nicht spezifiziert wird, standardmäßig double precision.

Um beispielsweise eine Variable y mit single precision mit dem Wert 5 zu erzeugen, können Sie folgende Befehle verwenden:

LISTING 1. Matlab

```

1 y = single(5)

```

LISTING 2. Python

```

1 import numpy as np
2 y = np.single(5)

```

Projekt 2 (3 Punkte). Die Funktionen $f, g: \mathbb{R}_{>0} \rightarrow \mathbb{R}$ definiert durch

$$f(x) = \frac{1}{x} - \frac{1}{x+1}, \quad g(x) = \frac{1}{x(x+1)}$$

stimmen überein, motivieren aber zwei unterschiedliche Verfahren zur numerischen Berechnung. Bestimmen Sie für $x_k = 10^k, k = 1, 2, \dots, 15$, den Ausdruck

$$\delta_k = \frac{|f(x_k) - g(x_k)|}{|g(x_k)|}$$

und tragen Sie die Ergebnisse in eine Tabelle ein. Was beobachten Sie und wie erklären sie die Beobachtungen?