Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät
Institut für Mathematik

pictures/husiegel_bw.pd

Berichts-Handout

Untersuchung der harmonischen Reihe

Arbeitsblatt 3

Chantal Gerth, Alina Apel
635838, 614787

Gruppe 22

Inhaltsverzeichnis

1 Einführung und Motivation

Im heutigen Zeitalter werden Computer in fast allen Lebensbereichen genutzt. Daher ist es besonders wichtig, dass der Computer korrekt rechnet. Leider tut er das nicht immer. Ein anschauliches Beispiel dafür ist die harmonische Reihe. Deswegen geht es in diesem Experiment darum, die Konvergenz der harmonischen Reihe mit verschiedenen Algorithmen und Datentypen zu untersuchen und dies mit dem analytischen Ergebnis zu vergleichen.

2 Theoretische Grundlagen

Die harmonische Reihe ist eine unendliche Reihe der Form

$$\sum_{n=1}^{\infty} \frac{1}{n},$$

die aus den Kehrwerten natürlicher Zahlen besteht. Mathematisch gesehen divergiert die harmonische Reihe, was bedeutet, dass die Summe der Reihe gegen Unendlich strebt, wenn die Anzahl der Terme zunimmt. Beim Berechnen der Partialsummen der harmonischen Reihe durch einen Computer kann es allerdings zu Rundungsfehlern kommen, die durch die begrenzte Genauigkeit der Fließkommazahlen verursacht werden. Je nach Datentyp wird n so groß und damit der auf die vorherige Partialsumme aufzuaddierende Summand so klein, dass die Bitmuster der entsprechenden Datentypen nicht mehr ausreichen, um diese numerisch kleine Zahl darzustellen. Der entsprechende Summand wird deshalb auf 0 gerundet. Für höhere n wird dann nur noch 0 aufaddiert, sodass die Folge der Partialsummen zu konvergieren scheint. Die Genauigkeit der Ergebnisse hängt dabei vom verwendeten Datentyp und Algorithmus ab. Je weniger Nachkommastellen das Bitmuster eines Datentyps darstellen kann, desto früher tritt das Problem auf. Schlussendlich trat in allen von uns untersuchten Beispielen diese Konvergenz auf, obwohl die Reihe selbst divergiert.

3 Experiment

Unser Programm `main.py` ist ein Experimentierskript für den Nutzer. Es gibt die Möglichkeit, eine gewisse Anzahl an Partialsummen zu berechnen, grafisch darzustellen und zu speichern/abzurufen. Dabei kann der Nutzer Standardparameter wählen oder eigene Parameter eingeben.

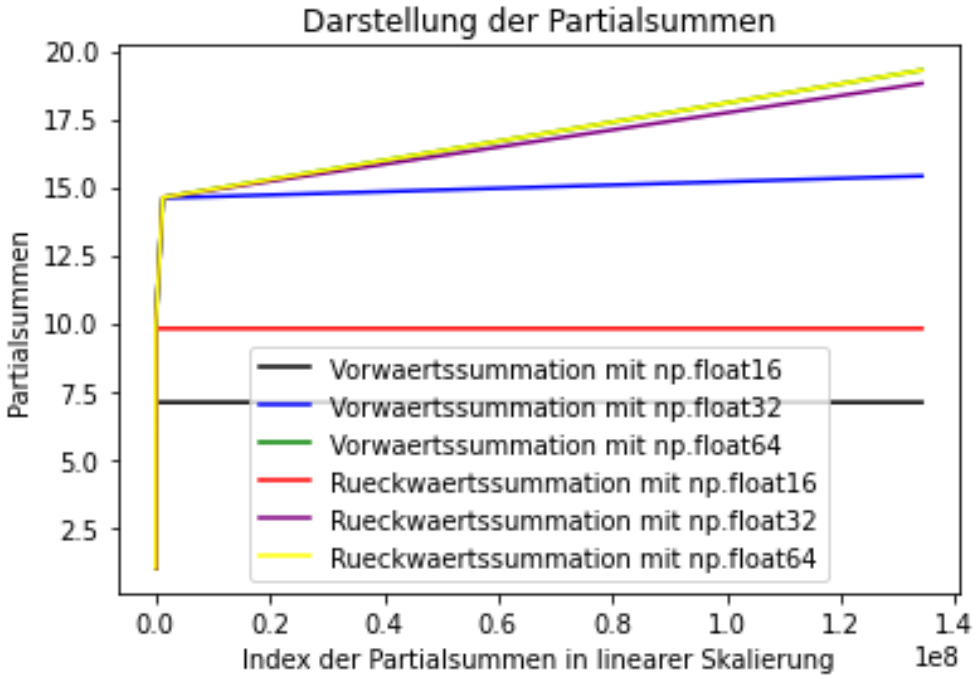

Abbildung 1: Basis 8, Startwert 0, Endwert 9, Num 5

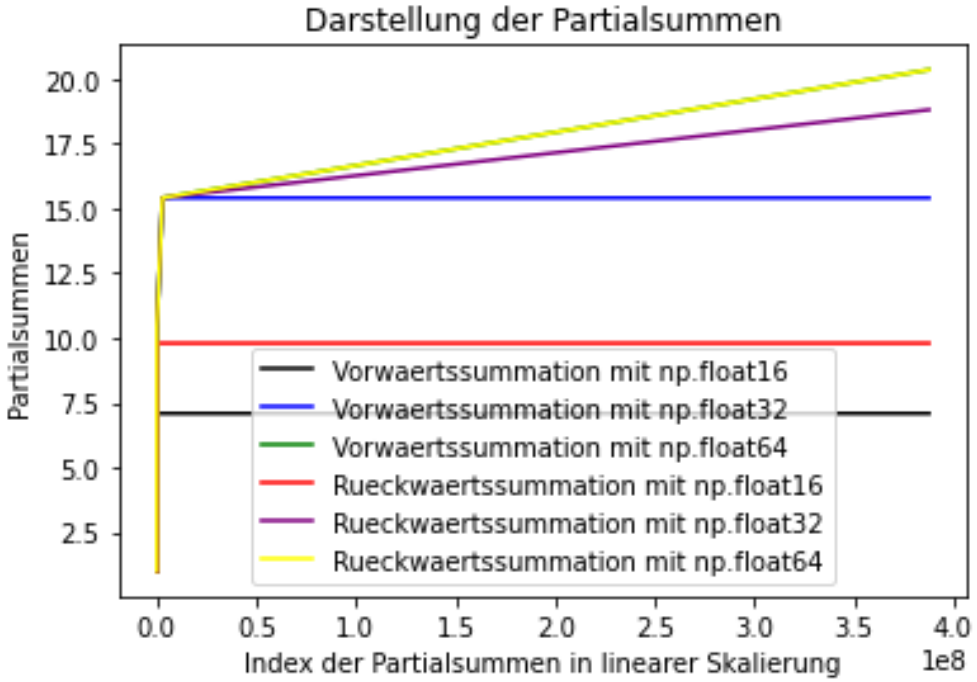

Abbildung 2: Basis 9, Startwert 0, Endwert 9, Num 5

Für die Berechnung wird ein Anfangswert und ein Endwert, sowie eine Basis benötigt. Dann kann der Nutzer wählen, wie viele Partialsummen er berechnet haben will. Die Eingabe der Parameter erfolgt über den Terminal. Dafür wird die Funktion read_number aus dem Modul tools_read_save (**??**) genutzt. Wählt der Nutzer die vordefinierten Parameter, wird mit `start = 1, stop = 5, basis = 10, num = 10` fortgefahren.

Anschließend berechnet das Programm durch Vorwärtssummation und Rückwärtssummation mit den Datentypen `np.float16`, `np.float32` und `np.float64` die Partialsummen und stellt die Ergebnisse graphisch dar. Hat der Nutzer angegeben, dass die berechneten Partialsummen gespeichert werden sollen, werden die Ergebnisse mit der Funktion save_data aus dem Modul tools_read_save (**??**) in eine CSV-Datei exportiert.

Entsprechend dieses Arbeitsablaufs wurden für die Untersuchung der Konvergenz der harmonischen Reihe folgende Berechnungen durchgeführt:

- Basis 8, Startwert 0, Endwert 9, 5 Partialsummen
- Basis 9, Startwert 0, Endwert 9, 5 Partialsummen

Die Ergebnisse sind in **??** und **??** graphisch dargestellt. Eine Wahl höherer Parameter war aufgrund der Laufzeit des Programms nicht möglich. In den Grafiken ist dennoch das theoretisch zu erwartende Problem, welches in **??** beschrieben wurde, ersichtlich: Die Reihe scheint für die Datentypen `np.float16`, `np.float32` zu konvergieren.

4 Schlussfolgerung

Basierend auf den Ergebnissen unseres Experiments lässt sich feststellen, dass die harmonische Reihe aufgrund von Rundungsfehlern und begrenzter Genauigkeit der Fließkommazahlen auf Computern scheinbar konvergiert, obwohl sie theoretisch divergiert. Insbesondere zeigt sich, dass die Genauigkeit der Berechnung von der Wahl des Datentyps und des verwendeten Algorithmus abhängt. Während die Vorwärtssummation mit dem Datentyp `np.float32` die geringste Genauigkeit aufweist, liefert die Rückwärtssummation mit dem Datentyp `np.float64` das genaueste Ergebnis. Diese Erkenntnisse regen dazu an, weitere Optimierungen zu untersuchen, um möglicherweise eine Divergenz der harmonischen Reihe auf dem Computer nachzuweisen.

5 Python-Dokumentation tools_read_save

Das Modul `tools_read_save.py` dient der Modularisierung häufig verwendeter Funktionen zur Nutzerinteraktion. Das Programm implementiert drei Funktionen: `read_number` zum Einlesen von Nutzereingaben, `save_data` zum Abspeichern von Listen in .csv-Dateien, und `load_data` zum Einlesen von Daten aus .csv-Dateien.

5.1 Schnittstellendokumentation

read_number

Liest eine Zahl vom Benutzer ein, die innerhalb der angegebenen Grenzen liegt und in den gewünschten Datentyp konvertiert werden kann.

Parameter:

- `question` (str): Eingabeaufforderung an den Benutzer.
- `data_type` (type): Gewünschter Datentyp der Eingabe.
- `lower_limit` (float): Untere Grenze der gültigen Eingabe (Standardwert: -∞).
- `upper_limit` (float): Obere Grenze der gültigen Eingabe (Standardwert: ∞).

Rückgabewert:

- Die eingegebene Zahl im gewünschten Datentyp.

Ausnahmen:

- `ValueError`: Wenn eine leere Eingabe gemacht wird.

Beispiel:
Nachfolgend wird die verwendung der Funktion im Experimentierskript zur Konvergenz der harmonischen Reihe (**??**) gezeigt:

```
num = read_number(question = "Anzahl der zu berechnenden Partialsummen: ",
                  data_type = int,
                  lower_limit = 2)
```

Programmablauf:
Im nachfolgenden Fließdiagramm ist dargestellt, wie die Funktion sicherstellt, dass die Eingabe des Nutzers den gewünschten Anforderrungen entspricht, bevor der eingelesene Wert zurückgegeben wird.

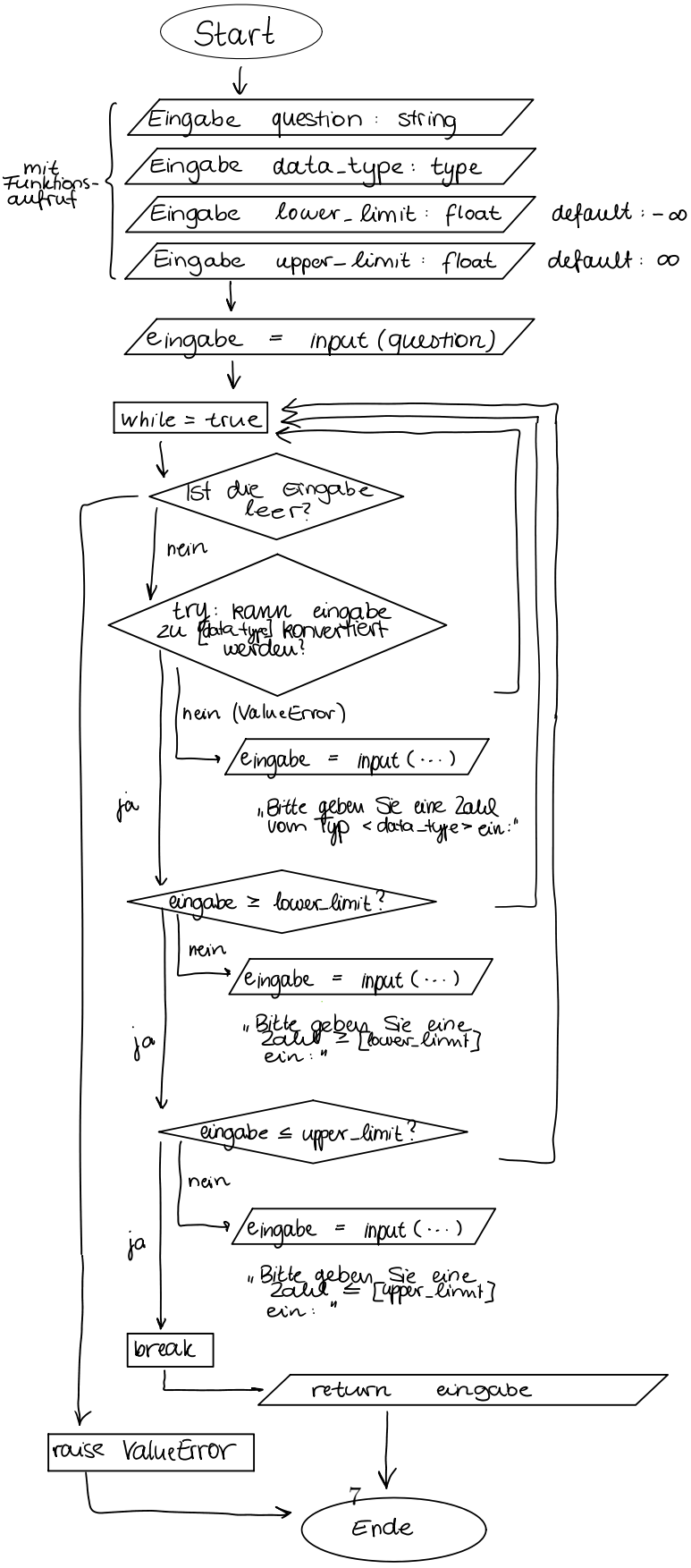

Abbildung 3: Fließdiagramm zum Ablauf von read_number

save_data

Speichert eine Liste von Zahlen in eine CSV-Datei mit `","` als Trennzeichen.

Parameter:

- `data`: Eine Liste von Zahlen.
- `filepath` (str): Pfad der zu speichernden Datei.

Ausnahmen:

- `RuntimeError`: Wenn das Speichern fehlschlägt.

Beispiel:
Nachfolgend wird die verwendung der Funktion im Experimentierskript zur Konvergenz der harmonischen Reihe (**??**) gezeigt:

```
save_data(result_vorwaerts_float16,
          "convergence_data_export/result_vorwaerts_float16.csv")
```

read_data

Liest eine Liste von Zahlen aus einer CSV-Datei mit `","` als Trennzeichen ein.

Parameter:

- `filepath` (str): Pfad der zu lesenden Datei.

Rückgabewert:

- Eine Liste von Zahlen.

Ausnahmen:

- `RuntimeError`: Wenn das Einlesen fehlschlägt.

Beispiel:
Nachfolgend wird die verwendung der Funktion im Experimentierskript zur Konvergenz der harmonischen Reihe (**??**) gezeigt:

```
result_vorwaerts_float16 = load_data("convergence_data_import/
                                      result_vorwaerts_float16.csv")
```

5.2 Nutzungshinweise und Hauptprogramm

In tools_read_save.py ist eine main() Funktion implementiert. Sie dient der Demonstration der enthaltenen Funktionen und wird nur ausgeführt, wenn tools_read_save.py direkt mittels python3 tools_read_save.py gestartet wird.

Dabei wird der Nutzer zuerst aufgefordert, eine ganze Zahl zwischen 3 und 7 in den Terminal einzugeben, um read_number zu testen. Eine leere Eingabe beendet das gesamte Testprogramm.

```
def main():
    """Anwednungsbeispiele"""
    # read_number()
    print("Zun chst wird die Funktion load_data() getestet.")
    print("Eine leere Eingabe erm glicht den Abbruch und f hrt
          zum Test der n chsten Funktion.")

    anfrage = "Bitte geben Sie eine ganze Zahl x mit 3 <= x <= 7 ein."

    try:
        eingabe_zahl = read_number(anfrage, int, lower_limit=2)
        print("")
        print("Die Funktion gibt zur ck: "
              + str(eingabe_zahl)
              + ", Datentyp: "
              + str(type(eingabe_zahl)))
    except ValueError:
        print("Abbruch des Tests.")
        exit()
```

Anschließend wird save_data durch Speichern einer vordefinierten Liste demonstriert:

```
    # save_data()
    print("")
    print("Nun wird eine Beispielliste erstellt und exportiert:
          [1.1117634, 2.55, 3.3, 144.0]")
    liste = [1.1117634, 2.55, 3.3, 144.0]
    try:
        save_data(liste, "test.csv")
    except RuntimeError:
```

```
        print("Speichern der Liste fehlgeschlagen.")
```

Zuletzt wird read_data demonstriert, indem die zuvor gespeicherte Liste wieder eingelesen wird:

```
# load_data()
print("")
print("Dieselbe Liste wird nun wieder eingelesen und ausgegeben.")
try:
    print(load_data("test.csv"))
except RuntimeError:
    print("Einlesen der Datei fehlgeschlagen.")
```

Literaturverzeichnis

[1] NumPy Developers, *NumPy documentation* (2022). https://numpy.org/doc/1.26/

[2] *Harmonische Reihe.* Wikipedia [online] https://de.wikipedia.org/wiki/Harmonische_Reihe

[3] Scott Pakin, *The Comprehensive LaTeX Symbol List* (2024). https://tug.ctan.org/info/symbols/comprehensive/symbols-a4.pdf

[4] Hella Rabus, *Einführung in das Wissenschaftliche Rechnen, Skript zur Vorlesung* (2024). https://moodle.hu-berlin.de/mod/folder/view.php?id=4331460

Selbstständigkeitserklärung

Ich versichere, dass ich in dieser schriftlichen Studienarbeit alle von anderen Autor:innen wörtlich übernommenen Stellen wie auch die sich an die Gedankengänge anderer Autoren:innen eng anlehnenden Ausführungen meiner Arbeit besonders gekennzeichnet und die entsprechenden Quellen angegeben habe. Zusätzlich führe ich den Einsatz von IT-/KI-gestützten Schreibwerkzeugen zur Anfertigung dieser Arbeit im Abschnitt „Übersicht verwendeter Hilfsmittel“ im Anhang des eingereichten pdf-Dokumentes vollständig auf. Hierin habe ich die Verwendung solcher Tools vollständig durch Angabe ihres Produktnamens, meiner Bezugsquelle (z.B. URL) und Angaben zu genutzten Funktionen der Software sowie zum Nutzungsumfang dokumentiert. Bei der Erstellung dieser Studienarbeit habe ich durchgehend eigenständig und beim Einsatz IT-/KI-gestützter Schreibwerkzeuge steuernd gearbeitet.

Chantal Gerth, Berlin, 21.05.2024

Alina Apel, Berlin, 21.05.2024