Fachbereich Mathematik & Informatik
Freie Universität Berlin
Prof. Dr. Ralf Kornhuber, Prof. Dr. Christof Schütte, Lasse Hinrichsen

3. Übung zur Vorlesung

Computerorientierte Mathematik I

WS 2020/2021

http://numerik.mi.fu-berlin.de/wiki/WS_2020/CoMaI.php

Abgabe: Do., 10. Dezember 2020, 12:15 Uhr

1. Aufgabe (8 TP)

Für $x, y, s \in \mathbb{R}$ mit $x, y, s > 0$ soll auf einem Rechner überprüft werden, ob die Gleichheit

$$x + y = s \tag{1}$$

gilt. Dabei ist zu beachten, dass im Rechner nur $\mathrm{rd}(x)$, $\mathrm{rd}(y)$, $\mathrm{rd}(s)$ darstellbar sind und $\mathrm{eps} \leq 0.5$ gilt.

a) ges. Bsp. sei $C = 1$ $y = 0{,}13$ $x = 0{,}17$ $y + x = 0{,}26 = 0{,}3$ $\mathrm{rd}(x) + \mathrm{rd}(y) = \mathrm{rd}(s)$ $= 0{,}1 + 0{,}1 = 0{,}2$

a) Zeigen Sie durch ein Beispiel, dass die Prüfung von

$$\mathrm{rd}(x) + \mathrm{rd}(y) = \mathrm{rd}(s) \tag{2}$$

nicht sinnvoll ist, da im Allgemeinen nicht „(1) ⇒ (2)“ gilt.

b) Zeigen Sie, dass die Abfrage

$$|\mathrm{rd}(x) + \mathrm{rd}(y) - \mathrm{rd}(s)| \leq 4\,|\mathrm{rd}(s)|\,\mathrm{eps} \tag{3}$$

in dem Sinne sinnvoll ist, dass „(1) ⇒ (3)“ gilt.

2. Aufgabe (8 PP + 2 TP Bonuspunkte)

a) Schreiben Sie eine Funktion `runden(x, L)`, die eine Eingabezahl x auf L Stellen rundet.

Zur Klarstellung: Dies soll gerade der Abbildung rd nach $\mathbb{G}(10, L)$ aus der Vorlesung mit kaufmännischem Runden entsprechen, also insbesondere *nicht* der Darstellung als Festkommazahl.

b) Schreiben Sie Funktionen `add(x, y, rd)` und `mult(x, y, rd)`. Hierbei sind `x` und `y` skalare Zahlen und `rd` ist eine Rundungsfunktion (wie beispielsweise `runden(·,L)` aus der vorangegangenen Unteraufgabe für festes L). Als Rückgabewert wird die

$|x + y - s| \leq 2|s|$

$\Rightarrow -2|s| \leq x + y - s \quad \wedge \quad x + y - s \leq 2|s|$

$\Rightarrow -x - y + s \leq -2s \leq x + y - s \quad \wedge \quad x + y - s \leq 2s \leq -x - y + s$

mit `rd` gerundete Summe bzw. Produkt der ebenfalls mit `rd` gerundeten Skalare `x` und `y` erwartet.

Anders formuliert: Implementieren Sie die Addition und Multiplikation im Sinne der Gleitkommaarithmetik entsprechend Vorlesung, wobei die zu verwendende Rundungsfunktion als zusätzlicher Parameter übergeben wird.

c) Schreiben Sie eine Funktion `c = binomA(a, b, rd)`, die die erste binomische Formel nach der Vorschrift

$$(\widetilde{a} + \widetilde{b})^2$$

auswertet und in der Variablen c zurückgibt, wobei $\widetilde{a}$ und $\widetilde{b}$ die mit rd gerundeten Werte von a und b sind und wobei die Ergebnisse von Addition und Multiplikation ebenso mit rd gerundet werden. Nutzen Sie hierfür die Funktionen `add` und `mult` aus der vorangegangenen Unteraufgabe.

d) Schreiben Sie analog zur vorangegangenen Unteraufgabe eine Funktion `c = binomB(a, b, rd)`, die nun jedoch die erste binomische Formel nach der Vorschrift

$$\widetilde{a}^2 + 2\widetilde{a}\widetilde{b} + \widetilde{b}^2$$

auswertet.

e) Nutzen Sie Ihre Funktionen `binomA` und `binomB` mit `rd = runden(·,L)`, um für $a = 0{,}012345$ und $b = -0{,}01234$ zu entscheiden, welche der beiden Darstellungen der binomischen Formel in diesem Fall die bessere ist. Betrachten Sie dabei verschiedene Werte für L und versuchen Sie das beobachtete Verhalten zu erklären. Schreiben Sie Ihre Antwort in eine Text-Datei mit dem Namen `beobachtungen.txt`.

Hinweis: Sie sollten Ihre Argumentation auf konkrete Daten stützen, die Sie ebenso in die Text-Datei aufnehmen.

f) (2 TP Bonuspunkte)

Vergleichen Sie die Ausgabe der folgenden Ausdrücke in Python:

- `print(0.1)` = 0,1
- `print(0.1*10 + 0.1*(-9))` = 1 + −0,9 = 0,1 erwartet — python = 0,0 9 8 (15 mal)
- `print(add(mult(0.1,10, rd), mult(0.1,-9, rd), rd))` = 0,1 — Bei L=16 kommt das gleiche Ergebnis wie bei phython

Hierbei soll `rd` das Runden in $\mathbb{G}(10, 5)$ mittels `runden(·, 5)` bezeichnen.

Welches Resultat erwarten Sie? Wie lassen sich die unterschiedlichen Ergebnisse erklären?

3. Bonusaufgabe (Quiz) (1 Bonus TP/PP)
Formulieren Sie eine Frage zur Vorlesung. Falls Sie die Antwort wissen, geben Sie die richtige Antwort und 3 falsche Antwortmöglichkeiten an.

Allgemeine Hinweise

Die Punkte unterteilen sich in Theoriepunkte (TP) und Programmierpunkte (PP). Bitte beachten Sie die auf der Vorlesungshomepage angegebenen Hinweise zur Bearbeitung und Abgabe der Übungszettel, insbesondere der Programmieraufgaben.

f) (2 TP Bonuspunkte)

Vergleichen Sie die Ausgabe der folgenden Ausdrücke in Python:

- `print(0.1)` = 0,1
- `print(0.1*10 + 0.1*(-9))` = 1 + −0,9 = 0,1 erwartet — python: 0,09 8 (15 mal)
- `print(add(mult(0.1,10, rd), mult(0.1,-9, rd), rd))` = 0,1

Bei L=16 kommt das gleiche Ergebnis wie bei phython

Hierbei soll `rd` das Runden in $\mathbb{G}(10, 5)$ mittels `runden(·, 5)` bezeichnen.

Welches Resultat erwarten Sie? Wie lassen sich die unterschiedlichen Ergebnisse erklären?

1. Für print(0.1) erwarten wir 0,1
2. Für print(0.1 • 10 + 0.1 • (-9)) erwarten wir 0,1, da
   1 + −0,9 = 0,1
3. Für print(add(mult(0.1,10,rd), mult(0.1,-9,rd),rd)) erwarten wir 0,1
   Man sollte auch meinen, dass runden(·,5) nichts bringt. Da die Zahlen 1 + −0,9 = 0,1 alle nur 1 "länge" benötigen

Ausgabe:

```
>>> print(0.1)
0.1
>>> print(0.1*10+0.1*(-9))
0.09999999999999998
>>> print(0.09999999999999998123)
0.09999999999999998
>>> L = 5
>>> print(add(mult(0.1,10,runden),mult(0.1,-9,runden),runden))
0.1
>>> L = 16
>>> print(add(mult(0.1,10,runden),mult(0.1,-9,runden),runden))
0.09999999999999998
```

2. Zeigt unerwartet die Zahl 0,09999999999999998 an. Interessant dabei ist, dass die Zahl in Gleitkommadarstellung die "länge" 16 hat. Also rechnet python3 mit der "länge" 16. Wenn man versucht eine Gleitkommazahl mit mehr als 16 Zeichen zu printen wird automatisch gerundet.

3. zeigt wie erwartet 0,1.

Das liegt daran, dass 3. eigentlich genau so rechnet wie 2. aber am Ende das Ergebnis auf die 5te Stelle aufrundet. Bei rd(·, 16) wäre das Ergebnis bei 2 und 3 gleich

Siehe am folgenden Beispiel:

(add (mult (0.1, 10, rd), mult (0.1, -9, rd), rd))

add (1.0 , -0.9 , rd)

add (1.0 , -0,9, rd)

⋮

rd (0,0999999999999998)

1 2 3 4 5 6

aufrunden

= 0,1