Wissenschaftliches Rechnen – Paralleles Höchstleistungsrechnen

Christian Engwer

`http://wwwmath.uni-muenster.de/num/Vorlesungen/WissenschaftlichesRechnen_SS11/`

Abgabe 27. 4. 2011. Abgabe der Programmieraufgaben bis per Email an `christian.engwer@uni-muenster.de`, schriftliche Abgabe Donnerstags in der Vorlesung.

- Alle Programmierübungen müssen per Email und in ausgedruckter Form abgegeben werden.
- Achten sie darauf, ihr Programm ordentlich zu formatieren und gut zu kommentieren.

Gegeben beliebige Zahlen $x, y \in \mathbb{R}$, sowie die arithmetischen Operationen $* \in \{+, -, \cdot, /\}$.

Um die Zahlen in Fließkommazahlen zu verwandeln benötigen wir eine Abbildung $\mathrm{rd} : D \to \mathbb{F}$. Hierbei fordert man

$$|x - \mathrm{rd}(x)| \leq \min_{y \in \mathbb{F}} |x - y| \quad \forall x \in D. \tag{1}$$

In Analogie zu den Operationen auf dem Körper $\mathbb{R}$ definieren wir die entsprechende *Maschinenoperationen* $\circledast : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ für $\circledast \in \{\oplus, \ominus, \odot, \oslash\}$, dabei soll $\oplus$ dem $+$ Operator entsprechen, $\ominus$ dem $-$, usw. Wenn $x, y \in \mathbb{F}$ folgt daraus *nicht*, dass $x * y \in \mathbb{F}$, sondern es ist eventuell eine Rundung erforderlich.

Man fordert für die Maschinenoperationen folgende Eigenschaft:

$$x \circledast y = \mathrm{rd}(x * y) \quad \forall x, y \in \mathbb{F}.$$

Man sagt $\circledast$ ist *exakt gerundet*. *Anm:* Effizient ist dies möglich, indem der Prozessor die Operationen mit zwei Stellen höherer Genauigkeit durchführt, als die Daten gespeichert werden.

Übung 1 Rundung

Das gängige Verfahren zum Runden von Zahlen sind das Aufrunden (natürliche Rundung), bei Fließkomma Zahlen mit geradem β wird jedoch ein anderes Verfahren verwendet, die gerade Rundung. Wenn x eine auf r Stellen zu rundende Zahl ist und $\mathrm{left}(x) = \max\{y \in \mathbb{F} \mid y \leq x\}$ sowie $\mathrm{right}(x) = \min\{y \in \mathbb{F} \mid y \geq x\}$ dann gilt beim Aufrunden:

$$rd(x) = \begin{cases} \mathrm{left}(x) & \text{falls} \quad 0 \leq m_{r+1} < \beta/2 \\ \mathrm{right}(x) & \text{falls} \quad \beta/2 \leq m_{r+1} < \beta \end{cases}$$

Beim geraden Runden ist dagegen:

$$rd(x) = \begin{cases} \mathrm{left}(x) & \text{falls} \quad |x - \mathrm{left}(x)| < |x - \mathrm{right}(x)| \vee \\ & \qquad (|x - \mathrm{left}(x)| = |x - \mathrm{right}(x)| \wedge m_r \text{gerade}) \\ \mathrm{right}(x) & \text{sonst} \end{cases}$$

Dabei ist m_i jeweils die i-te Stelle von x

- Berechnen Sie die Folge von Fließkommazahlen $x_0 = x$, $x_1 = (x_0 \ominus y) \oplus y$, $x_n = (x_{n-1} \ominus y) \oplus y$ mit $x = 1.56$ und $y = -0.555$. Dabei seinen x, x_i und y Fließkommazahlen in der Darstellung $\mathbb{F}(10, 3, 1)$. Welche Ergebnisse erhalten Sie für die ersten 10 Folgenglieder mit Aufrunden bzw. mit gerader Rundung?
- Diskutieren Sie die Ergebnisse.
- Warum wird bei Fließkommazahlen das gerade Runden verwendet?

4 Punkte

Übung 2 Zahlendarstellung

Sie haben in der Vorlesung die allgemeine Darstellung von Fließkommazahlen als $\mathbb{F}(\beta, r, s)$ kennengelernt. Dabei ist β die Basis, r die Anzahl Stellen der Mantisse und s die Anzahl Stellen des Exponenten.

1. Bilden Sie $x_0 = \frac{1}{11}$ nach $\mathbb{F}(2, 10, 3)$ ab, d.h. geben Sie rd $\left(\frac{1}{11}\right)$ mit rd: $\mathbb{R} \to \mathbb{F}(2, 10, 3)$.
2. Stellen Sie $x_1 = \frac{1}{2048}$ in $\mathbb{F}(10, 5, 1)$ dar.

3 Punkte

Übung 3 Fliesskommaarithmetik

Die Fließkommaarithmetik hat einige überraschende Eigenschaften. Zwar gilt in $\mathbb{F}$ das Kommutativgesetz:

$$x \oplus y = x \oplus x \qquad \text{für } x, y \in \mathbb{F},$$

Assoziativ- und Distributivgesetz gelten im allgemeinen nicht, d.h. es ist für $x, y, z, \in \mathbb{F}$:

$$(x \oplus y) \oplus z \neq x \oplus (y \oplus z), \qquad (x \oplus y) \odot z \neq (x \odot z) \oplus (y \odot z)$$

Ein Beispiel, für welches das Distributivgesetz fehlschlägt ist $x, y, z \in \mathbb{F}(10, 8, 1)$ mit $x = 6,0000003, y = -6, z = 20000$.

- Rechnen Die für gegebenes Beispiel die Ungleicheit von $(x \oplus y) \odot z \neq (x \odot z) \oplus (y \odot z)$ nach
- Konstruieren Sie eigene Beispiel, in denen das Assoziativ-, bzw. Distributivgesetz nicht erfüllt sind.
- Beschreiben Sie, wie es zu diesem Verhalten kommt. Welche Eigenschaft der Fließkommazahlen sorgt für das beobachtete Phänomen?

3 Punkte

Übung 4 Programmieraufgabe – Debugging

Das Programm soll alle Zahlen von 1 bis 10 aufsummieren

```
#include <iostream>

// summiert alle Zahlen im Intervall [a,b]
int summieren (int a, int b)
{
  int summe;
  for (int i=a; i<=b; i++)
  {
    int summe = summe + i;
  }
  return 0;
}

int main ()
{
  std::cout << summieren(1,10) << std::endl;
  return 0;
}
```

Obwohl das Programm syntaktisch korrekt ist, rechnet es falsch. Der Code enthält drei Fehler.

Finden Sie die Fehler und korrigieren Sie diese. *3 Punkte*