Algorithmische Mathematik I

Winter Semester 2015 / 2016
Prof. Dr. Sven Beuchler
Markus Siebenmorgen

universitätbonn

Aufgabenblatt 2.

Abgabedatum: **04.11.2015**.

2. Version, Aufgabe 2b) modifiziert.

Aufgabe 1. (Gleitkommazahlen)

Wir statten einen Rechner mit einer Zwei-Byte (16 Bit) Gleitkommaarithmetik

$$\pm(.1\, m_1\, m_2 \ldots)_2 \cdot 2^e$$

aus. Die Zahldarstellung hat ein Vorzeichenbit, ist normalisiert und die führende 1 wird nicht gespeichert (hidden Bit). Für die Zahl 0 wird das Bitmuster $000\ldots0$ speziell verwendet. Der Exponent soll im Bereich $-7 \leq e \leq 7$ möglich sein und wird in sogenannter Exzess- (Bias-) Darstellung gespeichert, d.h. $e = \tilde{e} - \text{Bias}$ wobei die Dualdarstellung von $\tilde{e}$ gespeichert wird. In unserem Fall werden also 4 Bits für den Exponenten verwendet mit einem Bias von 8. Außer der Spezialbehandlung für die 0 soll nur noch die spezielle Exponentenbitfolge $\tilde{e} = 0$ zur Kennzeichnung von "Underflow" reserviert werden — sie steht also nicht für die Zahldarstellung zur Verfügung.

a) Welche Darstellung haben die Zahlen 13, 42.125 und 0.8? Für den Fall, daß eine Zahl nicht exakt darstellbar ist, werden die überzähligen Stellen einfach abgeschnitten.

b) Ermitteln Sie die zu der Bitfolge

$$1\,|\,01101011100\,|\,1101$$

gehörige Dezimalzahl.

c) Wie viele Zahlen können in diesem Gleitkomma–Format dargestellt werden? Die Bitkombinationen für Underflow seien zu vernachlässigen.

d) Geben Sie die größte darstellbare Zahl $z_{\max}$, die kleinste darstellbare Zahl $z_{\min}$ sowie die betragsmäßig kleinste Zahl $\bar{z} \neq 0$ an. Geben Sie jeweils auch die zugehörigen Bitmuster an.

(6 Punkte)

Aufgabe 2. (Darstellbarkeit von Gleitkommazahlen)

a) Zeigen sie, dass die Zahl $1/10$ keine t-stellige Gleitkommazahl zur Basis $B = 2$ ist.

Hinweis. Sie dürfen ohne Beweis verwenden, dass 2^k für $k \in \mathbb{N}$ nicht durch 10 teilbar ist.

b) Gegeben sei eine Menge von Gleitkommazahlen $G_{B,t,e_{\min},e_{\max}}$ und eine reele Zahl x mit $|x| \in [R_{\min}, R_{\max}]$. Zeigen Sie, dass für die Rundung $\mathrm{rd}(x)$, gilt

$$\mathrm{rd}(x) = x(1 + \varepsilon_x) \quad \text{mit} \quad |\varepsilon_x| \leq \frac{1}{2}\varepsilon,$$

wobei $\varepsilon = B^{1-t}$ die relative Maschinengenauigkeit bezeichnet.

(4 Punkte)

Aufgabe 3. (Fehlerfortpflanzung)

Wir wollen nun die Fortpflanzung von Fehlern bei der Durchführung der vier arithmetischen Grundoperationen $(+, -, \cdot, /)$ betrachten. Die Zahlen x und y seien mit Fehlern Δx und Δy behaftet, wobei $|\frac{\Delta x}{x}|$ und $|\frac{\Delta y}{y}|$ weit kleiner als 1 seien.[1] Zeigen Sie, dass selbst bei exakter Rechnung (also ohne weitere Rundungsfehler) für die relativen Fehler der Ergebnisse die folgenden Aussagen gelten:

a) $$\frac{(x+\Delta x)+(y+\Delta y)-(x+y)}{x+y} = \frac{x}{x+y}\cdot\frac{\Delta x}{x} + \frac{y}{x+y}\cdot\frac{\Delta y}{y}$$

b) $$\frac{(x+\Delta x)-(y+\Delta y)-(x-y)}{x-y} = \frac{x}{x-y}\cdot\frac{\Delta x}{x} - \frac{y}{x-y}\cdot\frac{\Delta y}{y}$$

c) $$\frac{(x+\Delta x)\cdot(y+\Delta y)-(x\cdot y)}{x\cdot y} \approx \frac{\Delta x}{x} + \frac{\Delta y}{y}$$

d) $$\frac{(x+\Delta x)/(y+\Delta y)-(x/y)}{x/y} \approx \frac{\Delta x}{x} - \frac{\Delta y}{y}$$

e) Falls der relative Fehler des Ergebnisses sehr viel größer ist, als der relative Fehler der Eingabedaten, so spricht man von Auslöschung. In welchen der obigen Fälle kann Auslöschung auftreten?

Hinweis. Das "$\approx$" heißt hier, dass die sehr kleinen Produkte $\Delta x \cdot \Delta y$ weggelassen werden können.

(5 Punkte)

Aufgabe 4. (Rundung)

a) Zeigen Sie, dass die folgenden Ausdrücke mathematisch äquivalent sind:

- $((a+b)(a-b))^2$
- $(a^2+b^2)^2 - 4(ab)^2$
- $(a^2-b^2)^2$

b) Seien nun $a = 10^6+1$ und $b = 10^6-2$. Multiplizieren Sie damit obige Ausdrücke aus. *Jedes* Zwischenergebnis, das nicht mit 10 gültigen Stellen dargestellt werden kann, soll auf 10 Stellen gerundet werden.

c) Berechnen Sie jeweils den relativen Fehler der Resultate (2 gültige Ziffern genügen). Was ist der Grund für dieses Verhalten?

(5 Punkte)

[1] Man schreibt dies üblicherweise als $|\frac{\Delta x}{x}| \ll 1$

Programmieraufgabe 1. (Berechnung der Maschinengenauigkeit)

In der Praxis lässt sich die Maschinengenauigkeit ε als die kleinste positive Gleitkommazahl ε ermitteln, so dass $\mathrm{rd}(1+\varepsilon) > 1$. Verwenden Sie dieses Verfahren um je ein C/C++ Programm zu schreiben, dass die Maschinengenauigkeit `epsilon` des `float`-, `double`- bzw. `long double`-Gleitkommasystems ermittelt.

(4 Punkte)

Programmieraufgabe 2. (Summen und Reihen)

a) Schreiben Sie ein C/C++ Programm welches für gegebenes $n \in \mathbb{N}$ den Wert der Summe

$$s_n = \sum_{k=1}^{n} \frac{1}{k^2}$$

ausgibt. Überlegen Sie sich wie groß n gewählt werden muss damit alle Summanden für $k > n$ kleiner als 10^{-6} sind.

b) Der Grenzwert für $n \to \infty$ der Summe aus Aufgabenteil a), die *Reihe* $s = \sum_{k=1}^{\infty} \frac{1}{k^2}$, ist endlich und hat den Wert $\frac{\pi^2}{6}$. Verändern Sie ihr Programm aus Aufgabenteil a) nun so, dass eine Approximation an diese Reihe bestimmt wird, die alle Summanden die kleiner sind als eine vorgegebene Genauigkeit ε vernachlässigt. Wie groß ist der Fehler im Vergleich zur Genauigkeit $\varepsilon = 10^{-6}$ und $\varepsilon = 10^{-10}$?

c) Schreiben Sie ein C/C++ Programm welches den Wert der Reihe

$$\sum_{k=0}^{\infty} \frac{x^k}{k!}$$

ausgibt, wobei alle Summanden die kleiner sind als eine vorgegebene Genauigkeit ε vernachlässigt werden sollen. Vergleichen Sie diesen Wert für $x \in \{1, 2, 3\}$ und für die Genauigkeit $\varepsilon = 10^{-6}$ mit dem Wert der Exponentialfunktion $\exp(x)$. Diese Funktion steht in C/C++ zur Verfügung, sobald Sie den header `math.h` eingebunden haben.

(6 Punkte)

Programmieraufgabe 3. (Präsenzübung: Komplexe Zahlen)

Schreiben Sie ein C/C++ Programm welches

a) den Real- und Imaginärteil einer komplexen Zahl c einliest und anschließend c in der Form $c = a + ib$ ausgibt.

b) die Addition, Subtraktion, Multiplikation und Division zweier komplexer Zahlen c und d durchführt.

c) die komplexe Zahl c in der Standardform $c = a + ib$ in die trigonometrische Form $c = re^{i\phi}$ überführt.

d) die komplexe Zahl $c = re^{i\phi}$ in trigonometrischer Form in die Standardform $c = a + ib$ überführt.

Die Präsenzübung wird in der Woche 2.-6.11 in den Programmiertutorien besprochen.