Technische Universität Berlin **Wintersemester 2018/2019**
Fakultät II – Mathematik und Naturwissenschaften
Institut für Mathematik
Reinhold Schneider, Benjamin Kutschan

Numerische Mathematik I

1. Übungsblatt: Rechnerarithmetik und Rundungsfehler

Hausaufgaben: (Abgabe Mittwoch 31. Oktober 16:10 - 16:15 in Raum A 053)

Aufgabe 1: **(2+2+2 Punkte)**

Die Maschinenoperation $\oplus$ können wir durch die Gleichung

$$x \oplus y = (x + y)(1 + \varepsilon)$$

mit der gewöhnlichen Addition in Beziehung setzen. Dabei bezeichnet ε den relativen Rundungsfehler. Handelsübliche Prozessoren sorgen dafür, dass dieser relative Rundungsfehler ε unabhängig von x und y beschränkt bleibt. Es gibt also ein ε^* so dass die Abschätzung

$$|\varepsilon| \leq \varepsilon^*$$

erfüllt wird.
(Wie groß dieses ist, wird in der ersten Programmieraufgabe bestimmt. Siehe auch erste Übung und Aufgabe 3 dazu. Das Doppelte, also $2\varepsilon^*$ wird übrigens als Rechnergenauigkeit oder Maschinenepsilon bezeichnet.)
Wir wollen nun n Zahlen summieren. Seien dazu Maschinenzahlen $x_1, \ldots, x_n$ gegeben und sei durch

$$\tilde{s}_1 = x_1, \quad \tilde{s}_n = \tilde{s}_{n-1} \oplus x_n$$

ein Algorithmus zur Berechnung der Summe $s_n = \sum_{i=1}^n x_i$ definiert.
Wir wollen in dieser Aufgabe den Fehler exakt abgeschätzten. ε^2 darf deshalb nicht gekürzt werden! Benutzen Sie bitte keine O-Notation. Führen Sie auch alle kleinen Zwischenschritte recht gewissenhaft aus. Man kann sich bei dieser Aufgabe leicht vertun.

(a) Berechnen Sie für $n = 1, ..., 4$ den absoluten Fehler $f_n := \tilde{s}_n - s_n$.

(b) Zeigen Sie: Zerlegt man $\tilde{s}_n = s_n + f_n$, so gilt für den absoluten Fehler

$$|f_n| \leq [(1 + \varepsilon^*)^n - 1] \sum_{i=1}^n |x_i|.$$

Diskutieren Sie den relativen Fehler von $\tilde{s}_n$.

(c) In dieser Aufgabe ist zu zeigen, dass die berechnete Lösung gleich der exakten Lösung von leicht gestörten Eingabedaten $x_i \cdot (1 + \delta_i)$ ist. Zeigen Sie also, dass

$$\tilde{s}_n = \sum_{i=1}^n x_i(1 + \delta_i) \quad \text{mit} \quad (1 - \varepsilon^*)^n - 1 \leq \delta_i \leq (1 + \varepsilon^*)^n - 1$$

und dass falls $n\varepsilon^* < 1$ ist, die Abschätzung

$$|\delta_i| \leq \frac{n\varepsilon^*}{1 - n\varepsilon^*}, \quad i = 1, \ldots, n$$

gilt.

Aufgabe 2: **(2+2 Punkte)**

Stellen Sie ein Neuntel als Binärzahl dar. Stellen Sie dazu zunächst neun im Binärsystem dar und führen Sie dann die schriftliche Division im Binärsystem durch, um das Ergebnis zu erlangen. Benutzen Sie keine Software dafür und nur einen Divisionsalgorithmus, den Sie verstehen. Nehmen Sie diese Aufgabe recht ernst, denn im weiteren Verlauf der Vorlesung (und in der Algebra) benötigen Sie ein gutes Verständnis des Divisionsalgorithmus. Schreiben Sie nun ein Neuntel im IEEE-754-Format für einfache Genauigkeit (single precision). Sie finden dazu Literatur auf der Kurswebseite.

Aufgabe 3: **(0 Punkte)**

Gilt für jede Norm $||\cdot||$, dass $||v+w|| \geq |||v|| - ||w|||$?

Programmieraufgabe 1: (Abgabe in den Rechnersprechstunden bis zum 2. November)
Schreiben Sie Matlab-Programme

```
e=meps(),    m=minimum(),    m=maximum()
```

die den maximalen relativen Rundungsfehler ε^* (siehe Aufgaben 1 und 3), die kleinste darstellbare positive Zahl x_{min} bzw. die grösste darstellbare Zahl x_{max} berechnen. Dabei soll ε^* als die kleinste Zahl ε definiert werden, für die $1 \oplus \varepsilon > 1$ gilt.
In den Programmen soll nur benutzt werden, dass intern eine Gleitkommadarstellung basierend auf dem Dualsystem verwendet wird, nicht jedoch die Länge explizit im Code oder als Variable einfließen.
Sie können zum Beispiel mit `if(1+y==1)` testen, ob auf 1 gerundet wird und dann `y` systematisch verkleinern.
Erkennen Sie anhand der Ausgaben von `meps()` und `maximum()` wie groß die Mantisse und der Exponent sind?
Vergleichen Sie ihre Ergebnisse mit denen der entsprechenden MATLAB-Funktionen (`eps,realmin,realmax`).
Interpretieren Sie die Unterschiede.