Aufgabenblatt 3.

Abgabedatum: **11.11.2015.**

3. Version, Aufgabe 4b) und Aufgabe 3b) modifiziert.

Aufgabe 1. (Vermeidung von Auslöschung)

Wir haben gelernt, dass mit *Auslöschung* eine inakzeptable Vergrößerung des relativen Eingabefehlers bezeichnet wird. Ferner haben wir gelernt, dass die Hauptquelle für Auslöschung die Subtraktion von betragsmäßig nahezu gleich großen Zahlen ist.

Schreiben Sie folgende Ausdrücke so um, dass für die angegebenen Argumente Auslöschung vermieden wird.

a) $\sqrt[3]{1+x}-1$ für $x \approx 0$

b) $\frac{1-\cos x}{\sin x}$ für $x \approx 0$

c) $\frac{1}{x-\sqrt{x^2-1}}$ für $x \gg 1$ (Erinnerung: d.h. x wesentlich größer als 1)

d) $x^3\left(\frac{x}{x^2-1}-\frac{1}{x}\right)$ für $x \gg 1$

Hinweis. Durch Erweiterung mit geeignet gewählten Termen $\frac{a}{a}$ mit $a \neq 0$ können Differenzen wegfallen. Beachten Sie auch die Identität $\cos^2 x + \sin^2 x = 1$.

(6 Punkte)

Aufgabe 2. (Datenstabilität)

a) Wir betrachten die Berechnung der kleineren Lösung der quadratischen Gleichung $y^2 - 10^6 y + q = 0$ unter der Voraussetzung, dass $q \leq 10^{12}/4$. Für welche Werte von q ist die relative Kondition kleiner als 10?

b) Zeigen Sie, dass rückwärtsstabile Algorithmen auch vorwärtsstabil sind.

(4 Punkte)

Aufgabe 3. (Stabilitätsanalyse)

Seien Maschinenzahlen $a_1, \dots, a_n$ gegeben und sei $f(a_1, \dots, a_n) = \sum_{i=1}^n a_i$ zu berechnen. Der Computer habe die Maschinengenauigkeit eps. Nach jedem Zwischenergebnis muss gerundet werden, so dass die tatsächliche Implementierung mit

$$\boxed{\mathrm{f}}(a_1, \dots, a_n) = (\cdots((a_1 \boxplus a_2) \boxplus a_3) \boxplus \cdots) \boxplus a_n$$

gegeben ist.

a) Zeigen Sie, dass f rückwärtsstabil ist, d.h. betrachten Sie

$$\boxed{\mathrm{f}}(a_1, \dots, a_n) = f(a_1(1+\delta_1), \dots, a_n(1+\delta_n))$$

und zeigen Sie, dass diese relativen Eingabefehler δ_i sich wie $|\delta_i| \leq C\mathrm{eps}$ mit einer von a_i unabhängigen Konstanten C abschätzen lassen.

b) Gegeben seien **positive** Zahlen $a_1, \ldots, a_n > 0$. Rechnen Sie nach, dass f vorwärtsstabil ist, d.h. zeigen Sie, dass

$$\left| \frac{\boxed{\mathrm{f}}(a_1, \ldots, a_n) - f(a_1, \ldots, a_n)}{f(a_1, \ldots, a_n)} \right| \leq C(a_1, \ldots, a_n)\, \mathrm{eps}$$

mit einer geeigneten Konstanten C, die von $a_1, \ldots, a_n$ abhängt, gilt.

c) Zeigen Sie, dass für den absoluten Fehler zwischen $\boxed{\mathrm{f}}$ und f gilt:

$$|\boxed{\mathrm{f}}(a_1, \ldots, a_n) - f(a_1, \ldots, a_n)| \leq \mathrm{eps} \sum_{i=1}^{n} |n - i + 1|\, |a_i|$$

d) In welcher Reihenfolge sollte summiert werden, um den Fehler zu minimieren?

Kleine Terme der Größenordnungen $\mathrm{eps}^2, \mathrm{eps}^3, \ldots$ sollen vernachlässigt werden.

(5 Punkte)

Aufgabe 4. (3-Term Rekursion)

Eine rekursiv definierte Folge

$$x_n = a_n x_{n-1} + b_n x_{n-2} + c_n \qquad \text{für} \quad n = 2, 3, \ldots$$

mit gegebenen Startwerten $x_0, x_1 \in \mathbb{R}$ und Koeffizienten $a_k, b_k, c_k \in \mathbb{R}$ mit $b_k \neq 0$ heißt 3-*Term Rekursion*. Betrachten Sie die 3-Term Rekursion

$$x_n = 2x_{n-1} - \frac{2q-1}{q^2} x_{n-2}, \quad q > 0,\, q \neq 1.$$

Zeigen Sie (per Induktion), dass

a) $x_n = C_1 q^{-n} + C_2 \left(\frac{2q-1}{q}\right)^n$ mit $C_1, C_2 \in \mathbb{R}$.

b) Falls $x_1 = 1$, dann folgt

$$x_n = \frac{(2q-1)x_0 - q}{2q-2} q^{-n} + \frac{q - x_0}{2q-2} \left(\frac{2q-1}{q}\right)^n.$$

c) Falls $x_1 = 1$ und $x_0 = q$, dann folgt $x_n = q^{1-n}$.

(5 Punkte)

Programmieraufgabe 1. (Harmonische Reihe)

Die *harmonische Reihe* $s = \lim_{n\to\infty} s_n$ mit

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

ist wohl die bekannteste divergente $s = \infty$ Reihe in der Mathematik.

a) Schreiben Sie ein C/C++ Programm welches für gegebenes $n \in \mathbb{N}$ den Wert von `summe` $= s_n$ bestimmt. Dabei soll `summe` im Format `float` berechnet werden. Testen Sie Ihr Programm für $n = 2^i$ mit $i = 5, 10, \ldots, 30$. Was fällt Ihnen hierbei auf?

b) Modifizieren Sie ihr Programm so, dass Sie das erste Folgeglied als Startwert für `summe` setzen und dann jeweils zuerst blockweise die Summe der Folgeglieder für Blöcke der Größe 2^k für $k = 0, 1, 2, \ldots, m$ berechnet und diese danach zu `summe` addiert, d.h.

$$\texttt{summe} = 1 + \underbrace{\frac{1}{2}}_{2^0\text{Terme}} + \underbrace{\left(\frac{1}{3} + \frac{1}{4}\right)}_{2^1\text{Terme}} + \underbrace{\left(\frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8}\right)}_{2^2\text{Terme}} + \ldots.$$

Beachten Sie, dass zwischen m und n die Relation $n = 2^{m+1}$ besteht und testen Sie ihr Programm wiederum für $m = 4, 9, \ldots, 29$. Vergleichen Sie die Ergebnisse mit den Ergebnissen aus Teil a). Was fällt Ihnen auf und wie erklären Sie sich das unterschiedliche Verhalten?

Bemerkung. Das exakte Ergebnis von s_n lässt sich approximieren durch $s_n \approx \log(n) + 0.5772156649$.

(6 Punkte)

Die Programmieraufgabe wird in der Woche vom 23.11-27.11 im Cip-Pool Endenicher Allee oder im Cip-Pool Wegelerstraße abgegeben/vorgestellt. In der Woche vom 16.11-20.11 werden in den Cip-Pools Listen für die Abgabe aushängen.

Programmieraufgabe 2. (Präsenzübung: 3-Term Rekursion)

Wir betrachten die 3-Term Rekursion aus Aufgabe 4. Schreiben Sie ein C/C++ Programm welches für gegebenes x_0, x_1, q und für $n \in \mathbb{N}$ die Folgeglieder $x_2, x_3, \ldots, x_n$ bestimmt. Testen Sie ihr Programm für $n = 500$ und

a) $x_0 = 3$, $x_1 = 1$ und $q = 3$.

b) $x_0 = 2$, $x_1 = 1$ und $q = 2$.

Wie lautet das korrekte Ergebnis und was liefert Ihr Programm für $n = 10, 100, 200, 500$? Können Sie sich das Ergebnis erklären?

Hinweis. Schauen Sie sich Aufgabe 4b) für die verwendeten Werte von q an.

Die Präsenzübung wird in der Woche 9.-13.11 in den Programmiertutorien besprochen.