Übung 1 *Kondition der Standardoperationen*

Berechnen Sie die relativen Konditionszahlen (Verstärkungsfaktoren) der Standardoperationen *Multiplizieren, Dividieren* und *Wurzel ziehen*, also der Abbildungen:

$$F(x,y) = x \cdot y, \qquad F(x,y) = \frac{x}{y} \qquad F(x) = \sqrt{x}.$$

(1+1+1 Punkte)

Übung 2 *Landau Symbole*
Schreiben Sie die folgenden Ausdrücke in der Form $f(h) = \mathcal{O}(h^m)$ (für $h \to 0, h > 0$) mit einem möglichst großen $m \in \mathbb{N}$.

$$\begin{aligned} f(h) &= (ph^2 + h)^2 - p^2h^4 \qquad p \in \mathbb{N} \\ f(h) &= -\frac{h^2}{\ln(h)} \\ f(h) &= \frac{\sin(x+h) - 2\sin(x) + \sin(x-h)}{h^2} + \sin(x). \end{aligned}$$

und skizzieren Sie jeweils $f(h)$ zusammen mit dem jeweiligen $c \cdot h^m$ auf dem Intervall $(0,1)$.
(Hinweis: Für einen der Ausdrücke ist die Form $f(h) = o(h^m)$ vorzuziehen!)

(1+1+2 Punkte)

Übung 3 *Problematische Auswertung*
Betrachten Sie die Funktion

$$f(x) = \frac{1 - \cos x}{x}, \quad |x| \ll 1.$$

a) Für welche x ist die Auswertung von $f(x)$ gut bzw. schlecht konditioniert?

b) Zeigen Sie, dass der Algorithmus, diesen Ausdruck in der gegebenen Form für $|x| \ll 1$ zu berechnen, instabil ist. Führen Sie hierzu eine Stabilitätsanalyse (quantitativ) durch. Dabei sei angenommen, dass $\cos x$ mit Maschinengenauigkeit berechnet wird.

c) Finden Sie für $|x| \ll 1$ einen stabilen Algorithmus zur Berechnung von $f(x)$. Zeigen Sie in Analogie zu b) die Stabilität Ihres Algroithmus. Hinweis: Die Darstellung von f kann mit Hilfe der Rechenregeln für trigonometrische Funktionen umgeformt werden ($\sin^2 x + \cos^2 x = 1$).

(1+3+3 Punkte)

Übung 4 *Potenzreihe für die Exponentialfunktion (Praktische Übung)*
Die Exponentialfunktion e^x lässt sich für $x \in \mathbb{R}$ als Potenzreihe auffassen, wobei der Konvergenzradius unendlich ist. Siehe Beispiel 1.1 in der Vorlesung †.

Die rekursive Formel zur Berechnung der Potenzreihe lautet

$$\begin{aligned} y_1 &:= x, & f_1 &:= 1 + y_1, \\ y_n &:= \tfrac{x}{n} y_{n-1} & f_n &:= f_{n-1} + y_n. \end{aligned} \tag{1}$$

Schreiben Sie nun ein Programm `potenzreihe`, welches für gegebenes x und n die entsprechende Näherung der Exponentialfunktion berechnet. Der verwendete Datentyp soll dabei variabel sein. Die Benutzereingabe soll über die Kommandozeile

```
./potenzreihe <Zahl> <Iteration> <Datentyp>
```

möglich sein, d.h. der Benutzer des Programms `potenzreihe` gibt für `<Zahl>` eine beliebige Zahl x ein, für `<Iteration>` die maximale Anzahl der Iterationschritte und für `<Datentyp>` entweder `double` oder `float`.

a) Testen Sie das Programm mit $x = 5$ und $x = -10$ für 100 Iterationschritte und verschiedene Datentypen. Bilden Sie die Differenz zwischen dem exakten Wert von e^x und der Näherung f_n

$$e_n = |e^x - f_n|.$$

b) Insbesonders für die Werte $x \ll 0$ ist das Ergebnis um mehrere Größenordnungen daneben. Probieren Sie die Rekursionformel (1) so umzuschreiben, dass der Fehler kleiner wird. Testen Sie es mit $x = -20$ und `float`.
Hinweis: Man sollte die Auslöschung bei der Substraktion $x_1 - x_2$ mit $x_1 \approx x_2$ vermeiden.

Hinweise:

- Der exakte Wert von e^x kann man mit `long double` berechnen:

  ```
  long double exakt = std::exp(x);
  ```

  wobei die Funktion `exp` ist in der Header-Datei `cmath` definiert

  ```
  #include <cmath>
  ```

(4+2 Punkte)

---

† `http://conan.iwr.uni-heidelberg.de/teaching/numerik0_ws2015/num0_slides_2.pdf`