Richard Huber
richard.huber@uni-graz.at

Numerische Mathematik für LAK
4.Übungsblatt – 29ten Oktober, 2019

4. Aufgabenblatt

Aufgabe 1. *Gegeben sei die Funktion* $\varphi(x_1, x_2, x_3) = (x_1^3 - x_2 x_3, \sin(x_1 - x_3))^T$*, ein Vektor* $(x_1, x_2, x_3) = (1, 2, 2)^T$ *und* $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3) = (x_1, x_2, x_3) + \delta$ *für einen Vektor von Fehlern* $\delta = (\delta_{x_1}, \delta_{x_2}, \delta_{x_3})$ *sowie* $y = \varphi(x_1, x_2, x_3)$.

- ***a)*** *Bestimmen Sie* $\delta_y = \varphi(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3) - y$ *zunächst als lineare Funktion von* $(\delta_{x_1}, \delta_{x_2}, \delta_{x_3})$ *bis auf Terme der Ordnung* $o(|\delta_{x_1}| + |\delta_{x_2}| + |\delta_{x_3}|)$*. Geben Sie anschließend unter den Annahmen* $|\delta_{x_1}| \leq 0.1$*,* $|\delta_{x_2}| \leq 0.2$*, und* $|\delta_{x_3}| \leq 0.3$ *Abschätzungen für* $|\delta_{y_i}|$ *mit* $i = 1, 2$ *an.*

- ***b)*** *Berechnen Sie die Konditionszahlen* $K_{ik} = \frac{\partial \varphi_i}{\partial x_k}(x_1, x_2, x_3)\frac{x_k}{y_i}$ *für* $i \in \{1, 2\}$*,* $k \in \{1, 2, 3\}$ *und nutzen Sie diese, um für* $i = 1, 2$ *den relativen Fehler* ε_{y_i} *als lineare Funktion von* $(\varepsilon_{x_1}, \varepsilon_{x_2}, \varepsilon_{x_3})$ *unter Vernachlässigung Terme höherer Ordnung auszudrücken. Geben Sie anschließend unter den Annahmen* $|\delta_{x_1}| \leq 0.1$*,* $|\delta_{x_2}| \leq 0.2$*, und* $|\delta_{x_3}| \leq 0.3$ *eine Abschätzung für* $|\varepsilon_{y_i}|$ *mit* $i = 1, 2$ *an.*

Matlab-Aufgabe 2 (Programmierung elementarer Funktionen)**.** *Viele Softwareprogramme (wie Matlab) haben vorgefertigte Funktionen zur Auswertung elementarer Funktionen wie* e^x*,* $\log(x)$ *oder sqrt*(x)*. Dabei muss aber natürlich auf elementarere mathematische Operationen* $(+, -, \cdot, \div)$ *zurückgegriffen werden, was beispielsweise mittels Taylor-Entwicklung bewerkstelligt werden kann, aber einiges an analytischem Vorwissen benötigt.*

- ***a)*** *Für die Exponentialfunktion* $\exp(x) = e^x$ *gilt für* $x \in \mathbb{R}$ *die Potenzreihendarstellung*

$$\exp(x) = \lim_{N \to \infty} \sum_{n=0}^{N} \frac{x^n}{n!} := S_N(x).$$

  Schreiben Sie eine Matlab-Funktion [val,hist]=my_exp(x, N)*, welche für* $x \in \mathbb{R}$ *und* $N \in \mathbb{N}$ *die Potenzreihendarstellung nutzt um* exp *zu approximieren. Die Funktion gibt als val den Wert von* S_N *zurück, sowie einen Vektor hist=* $[S_1, \ldots, S_N]$.

- ***b)*** *Für den natürlichen Logarithmus* $\log(x)$ *gilt die Potenzreihendarstellung*

$$\log(x) = \lim_{N \to \infty} \sum_{n=1}^{N} (-1)^{n+1} \frac{(x-1)^n}{n} := L_N(x) \qquad \text{für } 0 < x \leq 2.$$

  Schreiben Sie eine Matlab-Funktion [val,hist]=my_log(x, N)*, welche für* $x > 0$ *und* $N \in \mathbb{N}$ *den natürlichen Logarithmus* $\log(x)$ *mit der bei* N *abgebrochenen Potenzreihe approximiert. Dabei entspricht val dem Wert von* L_N *und hist=*$[L_1, \ldots, L_N]$*. (Achtung, wie kann man den Fall* $x > 2$ *abdecken?)*

- ***c)*** *Die Quadratwurzel einer Zahl* $x > 0$ *kann mittels dem Heronschen Verfahren berechnet werden, welches iterativ mittels*

$$x_1 \in \mathbb{R} \setminus 0, \qquad x_{n+1} = \frac{x_n^2 + x}{2x_n} \quad \text{für } n = 1, 2, \ldots$$

  gegeben ist. Schreiben Sie eine Funktion [val,hist]=my_sqrt(x, x_0, N) *welche die Iteration bis zu* $n = N$ *ausführt, und val mit dem Wert* x_N *sowie hist* $= [x_1, \ldots, x_N]$ *zurückgibt.*

Richard Huber
richard.huber@uni-graz.at

Schreiben Sie auch ein Matlab-Skript, welches ihre Funktionen für vernünftige x, x_0 und N aufruft, und das Ergebnis mit dem der entsprechenden Funktionen exp, log*, sqrt vergleicht um sich von der Richtigkeit der Implementierung zu überzeugen.*

Hinweis. *Es gilt* $\log(x) = -\log(1/x)$ *für* $x > 0$.

Matlab-Aufgabe 3 (Auslöschung in Summation)**.** *Gegeben sei eine Zahl $N \in \mathbb{N}$, reelle Zahlen $(a_n)_{n=1}^N$ und $(\widetilde{a_n})_{n=1}^N$ mit $\delta_{a_n} = \widetilde{a_n} - a_n$ für $n = 1, \dots, N$, sowie Zahlen $(\tau_n)_{n=1}^N$. Man setze $\overline{S}_0 = \widetilde{S}_0 = S_0 = 0$ und betrachten*

$$S_n = S_{n-1} + a_n, \qquad \widetilde{S}_n = \overline{S}_{n-1} + \widetilde{a_n} \qquad \overline{S}_n = \widetilde{S}_n + \tau_n \quad \text{für } n = 1, \dots, N.$$

a) *Zeigen Sie, dass für $\delta_{S_N} = \overline{S_N} - S_N$ die Gleichung $\delta_{S_N} = \sum_{n=1}^N (\delta_{a_n} + \tau_n)$ gilt. Folgern Sie unter der Annahme $|\delta_{a_n}| \le \tau |a_n|$ und $|\tau_n| \le \tau |\widetilde{S}_n|$ für ein $\tau > 0$, dass*

$$|\varepsilon_{S_N}| := \left| \frac{\delta_{S_N}}{|S_N|} \right| \le \frac{\tau}{|S_N|} \left(\sum_{n=1}^N |a_n| + \sum_{n=1}^N |\widetilde{S}_n| \right)$$

b) *Schreiben Sie eine Matlab-Funktion $[\overline{S}, abs, rel]$ =my_errors(K, ε, τ), welche für die Zahlen $a = [1 + \varepsilon, 2, \dots, K, -K, \dots, -2, -1]$ den Wert $\overline{S} = \overline{S}_N$ $(N = 2K)$ berechnet, wobei die Fehler mit $\delta_{a_n} = \tau |a_n|$ und $\tau_n = \tau |\widetilde{S}_n|$ für $n = 1, \dots N$ gegeben sind. Dabei soll K eine natürliche Zahl, $\varepsilon > 0$ und $\tau > 0$ sein. Die Ausgabe enthält den berechneten Wert $\overline{S}$, sowie den absoluten Fehler δ_{S_N} und relativen Fehler ε_{S_N}.*

c) *Schreiben Sie ein Matlab-Skript, welches die Funktion my_errors für $K \in \{1, \dots, 10\}$ mit $\tau = 0.001$ und $\varepsilon \in \{0.1, 0.5, 0.9\}$ aufruft, und die jeweiligen Ergebnisse zwischenspeichert. Danach nutzen Sie die "bar" Funktion um einen Plot für abs und einen Plot für rel zu erzeugen, analog zu Abbildung 1. Sind die Resultate wie aus Punkt a) zu erwarten?*

Bemerkung. *In diesem Beispiel steht S_N für die Summe der Zahlen (a_n), wobei $(\delta_{a_n})_n$ Fehler in den Daten a_n sind und τ_n für Rundungsfehler stehen, d.h. wir simulieren Fehler die bei der Addition auftreten. Dabei wird zum vorherigen Wert $\overline{S_n}$ zunächst der Wert $\widetilde{a}_n$ hinzugefügt, was $\widetilde{S}_n$ ergibt. Anschließend wird das Ergebnis $\widetilde{S}_n$ aber noch gerundet, was dem Hinzufügen des Werts τ_n entspricht. Es ist eine klassische Annahme, dass $|\delta_{a_n}| \le \tau |a_n|$ und $|\tau_n| \le \tau |\widetilde{S}_n|$ – d.h. die Fehler sind auf Rundungsfehler der Daten bzw. Ergebnisse zurückzuführen. Insbesondere ist unsere Wahl von τ viel größer als die Maschinengenauigkeit, wodurch "echte" Rechenfehler des Computers nicht ins Gewicht fallen.*

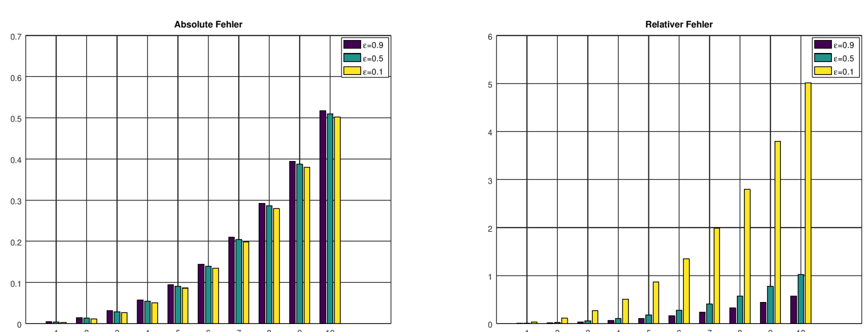

Abbildung 1: Bar-Plot absoluter und relativer Fehler