TU Chemnitz – Professur Numerische Mathematik
Prof. Dr. Oliver Ernst
Dipl.-Math. Björn Sprungk

16. April 2015

Numerische Mathematik

Sommersemester 2015

3. Übung

Hinweis: Programmieraufgaben sind mit (P) gekennzeichnet. Sie können vor der Übung per E-Mail abgegeben werden. Zur erfolgreichen Abgabe gehört dabei auch die Beantwortung evtl. Teilfragen der Aufgabe (z.B. in einem gesonderten Textdokument oder mittels Kommentaren in den MATLAB - Skripten).

Aufgabe 1

(a) Was ist das Ergebnis des arithmetischen Ausdrucks

```
u = abs( (4.0/3 - 1)*3 - 1 )
```

in MATLAB und warum?

(b) In den 50er Jahren war bei Division durch Null in Gleitpunktsystemen die Konvention verbreitet, anstelle eines Symbols ∞ als Ergebnis $N_{\max}$ zu liefern. Inwiefern wäre dies bei der Auswertung des Ausdrucks $(1/0)/10000000$ problematisch ? Was liefert IEEE-Arithmetik in diesem Fall ?

Aufgabe 2

Es sei

$$x_n = \left(1 + \frac{1}{n}\right)^n, \quad n = 1, 2, \ldots$$

Bekanntlich ist $\lim_{n\to\infty} x_n = e$.

a) Zeigen Sie

$$|e - x_n| = \mathcal{O}(n^{-1}) \text{ für } n \to \infty.$$

b) Berechnen Sie mit MATLAB den Fehler $|e - x_n|$ für $n = 10^m$, $m = 0, 1, \ldots, 18$, und interpretieren Sie Ihre Ergebnisse.

Aufgabe 3

Wir betrachten die Folge

$$I_n = \int_0^1 x^n\, e^x dx, \quad (n = 0, 1, 2, \ldots).$$

(a) Beweisen Sie, dass $0 < I_{n+1} < I_n < e/(n+1)$ für alle $n \in \mathbb{N}_0$ gilt.

(b) Zeigen Sie, dass I_n der folgenden Rekursionsformel genügt:

$$I_0 = e - 1,\ I_n = e - nI_{n-1} \quad (n = 1, 2, \ldots).$$

(c) Wertet man diese Rekursionsformel für $n = 0, 1, \ldots, 19$ auf einem Rechner mit Maschinengenauigkeit $\texttt{eps} = 2^{-52}$ aus, so erhält man das (unsinnige) Ergebnis $I_{19} = -13.2742$. Erklären Sie dieses Resultat (der alleinige Hinweis auf Rundungsfehler ist keine akzeptable Antwort).

(d) Berechnet man $\widetilde{I}_{19}$ (auf dem gleichen Rechner) durch

$$\widetilde{I}_{40} = 0,\ \widetilde{I}_{n-1} = \frac{1}{n}\,e - \frac{1}{n}\widetilde{I}_n \quad (n = 40, 39, \ldots, 20),$$

so ergibt sich $\widetilde{I}_{19} = 0.1297$. Schätzen Sie $|I_{19} - \widetilde{I}_{19}|$ ab.

Aufgabe 4 (P)

Die folgende Aufgabe soll zeigen, dass Rundungsfehler nicht zufällig sind. Werten Sie in MATLAB die rationale Funktion

$$r(x) = \frac{622 - x(751 - x(324 - x(59 - 4x)))}{112 - x(151 - x(72 - x(14 - x)))}$$

in der angegebenen Klammerung (*Horner-Schema*) für

$$x = x_k = 1.606 + (k-1)2^{-52}, \quad k = 1, 2, \ldots, 361,$$

das sind 361 aufeinander folgende Maschinenzahlen, aus und stellen Sie Ihre Ergebnisse grafisch dar. Die Funktion r ist im fraglichen Bereich (nahezu) konstant mit

$$r(x_k) \in [8.7523765807784881\ldots, 8.7523765807784891\ldots].$$

Plotten Sie $k-1$ gegen $10^{12}(r(x_k) - 8.75237658077848)$, um die Rundungsfehler sichtbar zu machen.

Aufgabe 5

(a) Es seien $f, g\colon \mathbb{R} \to \mathbb{R}$ hinreichend glatte Funktionen (z.B. zweimal differenzierbar) und $h(x) := g(f(x))$. Zeigen Sie, dass dann für die relative Konditionszahl von h folgendes gilt:

$$c_h(x) = c_g(f(x)) \cdot c_f(x).$$

Machen Sie sich klar, wie sich dies auf Funktionen $f : \mathbb{R}^n \to \mathbb{R}^n$ und $g : \mathbb{R}^n \to \mathbb{R}$ erweitern lässt.

(b) Wir betrachten nun die Berechnung der Nullstellen des Polynoms $x^2 - bx + c$. Konkret sei

$$h(b, c) := \frac{b - \sqrt{b^2 - 4c}}{2}.$$

Schreiben Sie h geeignet als eine Verkettung von Funktionen $f\colon \mathbb{R}^2 \to \mathbb{R}^2$ und $g : \mathbb{R}^2 \to \mathbb{R}$ und berechnen Sie mit Hilfe des obigen Ergebnisses die Konditionszahlen von h.

Für welche Punkte $(b, c) \in \mathbb{R}^2$ ist die Auswertung der Funktion h schlecht konditioniert?

Aufgabe 6 (P)

Der Mittelwert und die Varianz eines Vektors $x = (x_1, \ldots, x_N) \in \mathbb{R}$ ist gegeben durch

$$\mu_N := \frac{1}{N}\sum_{i=1}^{N} x_i, \qquad \sigma_N^2 := \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_N)^2.$$

Um beide Größen mit nur **einem** Durchlauf durch die Elemente des Vektors zu ermitteln werden zwei Algorithmen vorgeschlagen:

- Nutze die Beziehungen

  $$\mu_N = \frac{1}{N}\sum_{i=1}^{N} x_i, \qquad \sigma_N^2 = \frac{1}{N}\sum_{i=1}^{N} x_i^2 - \mu_N^2,$$

  d.h., bilde mittels eines Durchlaufs die Summen $\sum_{i=1}^{N} x_i$ und $\sum_{i=1}^{N} x_i$ und berechne μ_N und σ_N^2 entsprechend.

- Nutze die rekursiven Beziehungen

  $$\mu_n := \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{n-1}{n}\mu_{n-1} + \frac{x_n}{n},$$
  $$\sigma_n^2 := \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_n)^2 = \frac{n-1}{n}\sigma_{n-1}^2 + \frac{(x_n - \mu_n)^2}{n-1}.$$

  zur rekursiven Berechnung von μ_N und σ_N^2 mittels eines Durchlaufs.

Implementieren Sie beide Verfahren in MATLAB als Funktionen. Testen Sie diese an einem Vektor x erzeugt durch

```
x = 100 * ones(1,N) + 1e-5 * (rand(1,N) - 0.5)
```

für z.B. $N = 1000$ und vergleichen Sie die Ergebnisse der beiden Verfahren mit denen der MATLAB - Funktionen `mean` und `var`.

Welcher Algorithmus ist stabiler? Warum ist der andere Algorithmus instabil?