Übungsblatt 1 - Lösung

Aufgabe 1.1

Wir haben ein Gleitkomma-System $\mathbb{F}$ mit 4 Bits für den Exponenten e und $M = 12$ (signifikanten, d.h. 11 eigentlich abgespeicherten) Bits für die Mantisse gegeben.

(a) Was ist der größtmögliche relative Rundungsfehler?
(b) Was ist die kleinste positive (normalisierte) Zahl in $\mathbb{F}$?
(c) Was ist die größte Zahl in $\mathbb{F}$?
(d) Was ist (ungefähr) der größtmögliche absolute Rundungsfehler?
(e) Wieviele Zahlen lassen sich in $\mathbb{F}$ (theoretisch, d.h. ohne Platz für Sonderzahlen wie $\pm\infty$ und `NaN`) darstellen? (In anderen Worten: Was ist $|\mathbb{F}|$?)

Lösung

(a) Der größtmögliche relative Rundungsfehler ist immer $\varepsilon_{\text{mach}}/2 = 2^{-M}$. Also, hier $2^{-12} \approx 10^{-3.6}$. Somit ist eine Genauigkeit von mindestens 3 Dezimalstellen garantiert.

(b) Für die kleinstmögliche positive Zahl $x_{\min} \in \mathbb{F}$ setzen wir alle abgespeicherten, d.h. variablen, Mantissen-Bits auf 0 und wählen den kleinstmöglichen Exponenten $e_{\min}$. Für den Exponenten haben wir $2^4 = 16$ Möglichkeiten, wobei der theoretisch kleinste und größte Exponent für Sonderzahlen (subnormale Zahlen, $\pm\infty$, `NaN`) verwendet wird. Also gilt unter Berücksichtigung negativer Exponenten, dass $e \in [-6, 7]$. Also ist $e_{\min} = -6$ und somit $x_{\min} = 1 \cdot 2^{e_{\min}} = 2^{-6} \approx 10^{-1.8}$.

(c) Für die größtmögliche Zahl $x_{\max} \in \mathbb{F}$ setzen wir alle Mantissen-Bits auf 1 und wählen den größtmöglichen Exponenten, der nach (b) $e_{\max} = 7$ ist. Somit gilt $x_{\max} = (2 - 2^{-(M-1)}) \cdot 2^{e_{\max}} = (2 - 2^{-11}) \cdot 2^7 = 255.9375$.

(d) Der größtmögliche absolute Rundungsfehler ist (ungefähr) die größtmögliche Zahl multipliziert mit dem größtmöglichen relativen Rundungsfehler, also

$$x_{\max} \cdot \frac{\varepsilon_{\text{mach}}}{2} = 255.9375 \cdot 2^{-12} \approx 0.0625.$$

(e) Mit dem Bit für das Vorzeichen haben wir insgesamt 16 abgespeicherte, d.h. variable, Bits zur Verfügung. Also ist (theoretisch) $|\mathbb{F}| = 2^{16} = 65536$.

Aufgabe 1.2

Zur allgemeinen Approximation der Ableitung einer differenzierbaren Funktion f an der Stelle x_0 wird oft der (rechtsseitige) *Differenzenquotient*

$$\widetilde{f}'_r(x_0, h) := \frac{f(x_0 + h) - f(x_0)}{h} \approx f'(x_0)$$

für ein bestimmtes $h > 0$ verwendet.

Es sei nun $f(x) = x^2$ und $x_0 = 1$. Es gilt bekannterweise $f'(x_0) = 2$.

Plotte den Abfall des Fehlers $|\widetilde{f}'_r(x_0, h) - f'(x_0)|$ für kleiner werdendes $h > 0$ auf logarithmischen Achsen.

Finde eine plausible Erklärung für das Ergebnis.

(Mehr zur numerischen Ableitung gibt es in Kap. 8.)

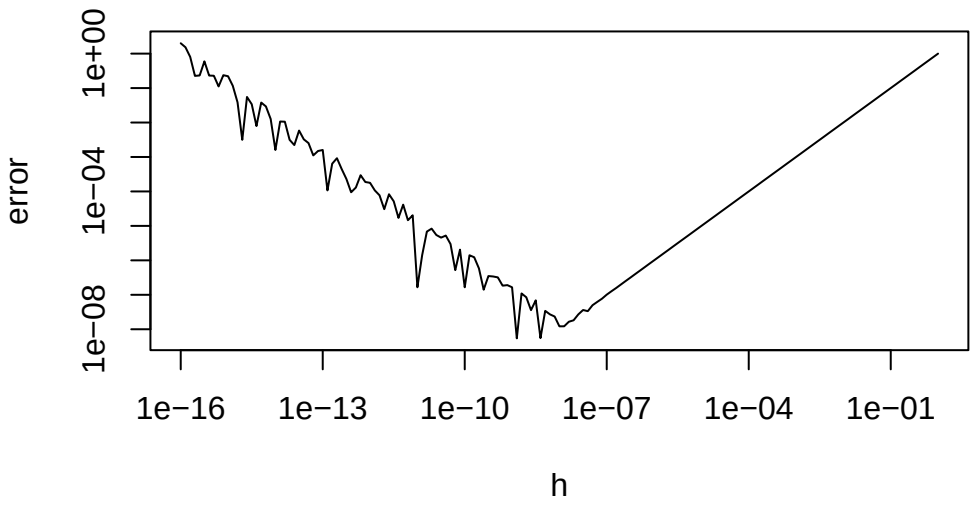
Lösung

```r
f <- \(x) x^2
f_p <- \(x) 2 * x
x_0 <- 1

err <- \(h) abs((f(x_0 + h) - f(x_0)) / h - f_p(x_0))

hs <- 10^(seq(-16, 0, 0.1))
errs <- err(hs)

plot(hs, errs, log = "xy", xlab = "h", ylab = "error", type = "l")
```



Erklärung: Der Gesamtfehler setzt sich aus zwei Fehlern, dem Approximationsfehler und dem Rundungsfehler, zusammen. Um den Approximationsfehler klein zu machen, muss man h möglichst klein wählen. Wird dieses h allerdings zu klein, setzt im Zähler des Differenzenquotienten Auslöschung ein, was den Rundungsfehler ansteigen lässt. Ein optimales h muss also beide Fehlerkomponenten ausbalancieren (ähnlich wie beim Bias-Variance Tradeoff im maschinellen Lernen). Man kann (für gut konditioniertes und stabil realisiertes f) zeigen, dass

$$h_{\text{opt}} \approx \sqrt{2 \left| \frac{f(x_0)}{f''(x_0)} \right| \varepsilon_{\text{mach}}}.$$

In unserem Fall wäre somit

$$h_{\text{opt}} \approx \sqrt{2 \cdot \frac{1}{2} \cdot 2.2 \cdot 10^{-16}} \approx 1.5 \cdot 10^{-8},$$

was mit dem Plot annähernd übereinstimmt.

Aufgabe 1.3

Wir möchten die quadratische Gleichung $x^2 + 2px + q = 0$ numerisch, d.h. auf einem Rechner, lösen. Die aus der Schule bekannte Mitternachtsformel liefert die zwei analytischen Lösungen

$$x_{1/2} = -p \pm \sqrt{p^2 - q}.$$

(a) Welche der beiden Lösungen ist für welche Werte von p und q aus numerischer Sicht von Auslöschung betroffen?

(b) Wie kann man mit dem Satz von Vieta einen Ausweg für den Problemfall der Auslöschung finden?

Hinweis: Der Satz von Vieta besagt, dass die zwei Lösungen folgende Gleichungen erfüllen:

$$x_1 + x_2 = -p$$
$$x_1 \cdot x_2 = q$$

(c) Implementiere die Mitternachtsformel und die Formel aus dem Satz von Vieta. Weise mit mehreren Werten für p und q den Unterschied der beiden Ansätze nach.

Lösung

(a) Wenn $p^2 \gg q$, subtrahieren wir bei $x_1 = -p + \sqrt{p^2 - q}$ zwei fast gleich große Zahlen $-p$ und $\sqrt{p^2 - q}$, was Auslöschung verursacht. Die zweite Lösung $x_2 = -p - \sqrt{p^2 - q}$ subtrahiert keine fast gleich großen Zahlen und ist somit nicht von Auslöschung betroffen.

(b) Die zweite Gleichung liefert

$$x_1 = \frac{q}{x_2} = \frac{q}{-p - \sqrt{p^2 - q}}.$$

Dieser Ausdruck ist nicht von Auslöschung betroffen.

(c)

```
mitternacht <- function(p, q) {
    x1 <- -p + sqrt(p^2 - q)
    x2 <- -p - sqrt(p^2 - q)
    return(c(x1, x2))
}

vieta <- function(p, q) {
    x2 <- -p - sqrt(p^2 - q)
    x1 <- q / x2
    return(c(x1, x2))
}
```

Um den Unterschied der beiden Ansätze nachzuweisen, verwenden wir verschiedene größere Werte für $p \in \{10^2, \ldots, 10^8\}$ und fixieren kleines $q = 1$, da diese Fälle laut Lösung zu (b) Auslöschung verursachen.

```r
ps <- 10^seq(2, 8, 1)
q <- 1

options(digits = 15)

for (p in ps) {
    cat("p:", p, " q:", q, "\n")

    m <- mitternacht(p, q)
    v <- vieta(p, q)

    print(m)
    print(v)
    cat("\n")
}
```

```
p: 100  q: 1
[1] -5.00012500624791e-03 -1.99994999874994e+02
[1] -5.00012500625039e-03 -1.99994999874994e+02

p: 1000  q: 1
[1] -5.00000125043698e-04 -1.99999949999987e+03
[1] -5.00000125000063e-04 -1.99999949999987e+03

p: 10000  q: 1
[1] -5.00000005558832e-05 -1.99999999500000e+04
[1] -5.0000000125e-05 -1.9999999950e+04

p: 1e+05  q: 1
[1] -4.99999441672117e-06 -1.99999999995000e+05
[1] -5.000000000125e-06 -1.999999999950e+05

p: 1e+06  q: 1
[1] -5.00003807246685e-07 -1.99999999999950e+06
[1] -5.00000000000125e-07 -1.99999999999950e+06

p: 1e+07  q: 1
[1] -5.02914190292358e-08 -1.99999999999999e+07
[1] -5.00000000000001e-08 -1.99999999999999e+07

p: 1e+08  q: 1
[1]  0e+00 -2e+08
```

```
[1] -5e-09 -2e+08
```

Man kann sehen, dass die Anzahl gleicher Dezimalstellen von Lösungen zu x_1 mit steigendem p abnimmt. Für x_2 stimmen die Lösungen miteinander überein, da beide Ansätze die gleiche Formel dafür verwenden.