Kapitel 1 : Fließkommazahlen

Im Computer gibt es verschiedene Typen zur Repräsentation von Zahlen. Etwa in C/C++ :

unsigned int	$\mathbb{N}_0$	(char, short, long)
int	$\mathbb{Z}$	(char, short, long)
~~float~~, double	$\mathbb{R}$	(float)
~~double~~	~~$\mathbb{R}$~~	
complex<double>	$\mathbb{C}$	(dito float)

int : exakt, aber endlicher Bereich

float, double, .. : Approximation, endlicher Bereich

Was hat das für Folgen?

Beispiel 1.1 (Potenzreihe für e^x). Auf Folie

1.1 Zahlendarstellung

Stellenwertsystem.

$$x = \pm \ldots m_n \beta^n + \ldots + m_1 \beta^1 + m_0 \beta^0 + m_{-1} \beta^{-1} + \ldots + m_{-k} \beta^{-k} + \ldots$$

$\pm$ ist das Vorzeichen (ein Bit genügt).

$\beta \in \mathbb{N}, \beta \geq 2$ heißt Basis.

$m_i \in \{0, 1, 2, \ldots, \beta - 1\}$ heißen Ziffern.

Jede reelle Zahl $x \in \mathbb{R}$ kann mit unendlich vielen Ziffern dargestellt werden.

Geschichte: Siehe [Knuth, Band 2, p. 194]

Babylonier 1750 v. Chr : $\beta = 60$

Basis 10 in Europa ab ca 1585

Blaise Pascal : Jedes $\beta \geq 2$ möglich

Festkommazahlen:

Wähle maximalen und minimalen Exponenten:

$$x = (-1)^s \sum_{i=-k}^{n} m_i \beta^i$$

Problem: Wissenschaftl. Anwendungen brauchen Zahlen sehr unterschiedlicher Größe

Plancksches Wirkungsquantum: $6.6260693 \cdot 10^{-34}$ Js

Ruhemasse Elektron: $9.11 \cdot 10^{-28}$ g

Avogadrokonstante: $6.0221415 \cdot 10^{23} \frac{1}{mol}$

$$\Rightarrow \beta = 2 \qquad 2^{n+k} \approx 10^{23+34} \rightarrow n+k = \frac{57}{\log_{10} 2} \approx \underline{190 \text{ Stellen}}!$$

Fließkommazahlen erlauben ~~effiziente~~ ausreichend genaue Darstellung unterschiedlich großer Zahlen. (im Sinne des relativen Fehlers)

Definition 1.2 (normierte Fließkommazahlen)

Sei $\beta, r, s \in \mathbb{N}$ und $\beta \geq 2$. $\mathbb{F}(\beta, r, s) \subset \mathbb{R}$ besteht aus den Zahlen mit folgenden Eigenschaften:

a) $\forall x \in \mathbb{F}(\beta, r, s)$ gilt $x = m(x)\, \beta^{e(x)}$ mit

$$m(x) = \pm \sum_{i=1}^{r} m_i \beta^{-i}, \qquad e(x) = \pm \sum_{j=0}^{s-1} e_j \beta^j$$

m heißt Mantisse, e Exponent.

b) $\forall x \in \mathbb{F}(\beta, r, s)$ gilt $x = 0 \vee m_1 \neq 0$.

$m_1 \neq 0$ heißt Normierung. Bedingung b) macht die Fließkommadarstellung eindeutig. ☒

Ist $x \in \mathbb{F}(\beta, r, s)$ und $x \neq 0$ dann gilt wegen b):

$$\beta^{-1} \leq |m(x)| < 1 \qquad \text{und damit } \beta^{e(x)-1} \leq \underset{\overset{\|}{|m(x)|\,\underbrace{|\beta^{e(x)}|}_{>0 \text{ also } |\beta^{e(x)}| = \beta^{e(x)}}}}{|x|} < \beta^{e(x)}. \quad (1.1)$$

Beispiel 1.3

a) $\mathbb{F}(10,3,1)$ besteht aus Zahlen der Form:

$$x = \pm(m_1\cdot 0.1 + m_2\, 0.01 + m_3\cdot 0.001)\cdot 10^{\pm e_0}$$

mit $m_1 \neq 0 \vee (m_1 = m_2 = m_3 = 0)$

z.B. $0.999\cdot 10^{1}$, $0.123\cdot 10^{-1}$, 0, aber

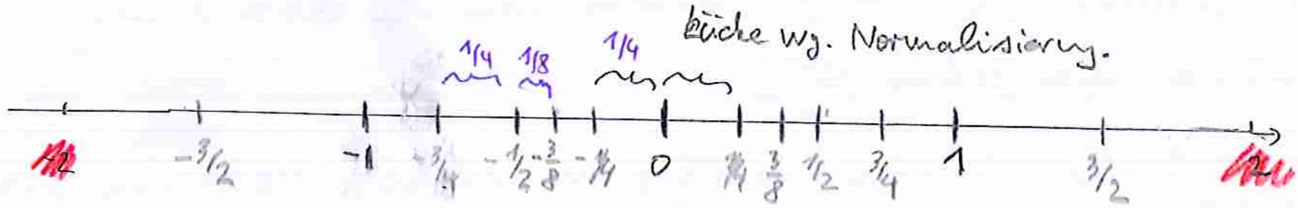
$0,\underline{000}\,\underline{000}\,\underline{000}\,014 = 0.140\cdot 10^{-10} \notin \mathbb{F}(10,3,1)$ da Exponent zu klein.

b) $\mathbb{F}(2,2,1)$ besteht aus Zahlen der Form

$$x = \pm\underbrace{\left(m_1\cdot\frac{1}{2} + m_2\cdot\frac{1}{4}\right)}_{\{0,\,\frac{1}{2},\,\frac{3}{4}\}}\cdot\underbrace{2^{\pm e_0}}_{\{2^{-1},\,2^{0},\,2^{1}\} = \{\frac{1}{2},\,1,\,2\}}$$

$m_1 = 1!$

$$\Rightarrow \mathbb{F}(2,2,1) = \Big\{\underbrace{-\frac{3}{2}, -1}_{\{-\frac{3}{4},-\frac{1}{2}\}\cdot 2}, \underbrace{-\frac{3}{4}, -\frac{1}{2}}_{\{-\frac{3}{4},-\frac{1}{2}\}\cdot 1}, \underbrace{-\frac{3}{8}, -\frac{1}{4}}_{\{-\frac{3}{4},-\frac{1}{2}\}\cdot\frac{1}{2}}, 0, \underbrace{\frac{1}{4}, \frac{3}{8}}, \underbrace{\frac{1}{2}, \frac{3}{4}}, \underbrace{1, \frac{3}{2}}\Big\}$$



Mögliche Lösg: Nehme für $|x| < \cancel{\beta^{-1}}\ \frac{1}{4}$ die nicht normalisierten Zahlen hinzu.

Zahlenbereich:

Größte / kleinste darstellbare Zahl in $\mathbb{F}(\beta,r,s)$

⇝ Betrag möglichst groß

$$X_{+/-} = \pm\underbrace{(\beta-1)\{\beta^{-1}+\dots+\beta^{-r}\}}_{=1-\beta^{-r}}\cdot\beta^{\overbrace{(\beta-1)\{\beta^{s-1}+\dots+\beta^{0}\}}^{=\beta^{s}-1}}$$

größte Ziffer

Wertigkeit der letzten Stelle

$$= \pm(1-\beta^{-r})\,\beta^{\beta^{s}-1}!$$

Kleinste positive / größte negative Zahl in $\mathbb{F}(\beta,r,s)$ ungleich Null:

⇝ Betrag mögl. klein

$$x_{+/-} = \pm\underbrace{\beta^{-1}}_{\text{kleinste Mantisse wg. Normierung}}\cdot\beta^{-\underbrace{(\beta-1)\{\beta^{s-1}+\dots+\beta^{0}\}}_{\beta^{s}-1}} = \pm\beta^{-\beta^{s}}.$$

negativ!

Damit

$$\mathbb{F}(\beta,r,s)\subset\mathbb{D}(\beta,r,s) = [X_-,x_-]\cup\{0\}\cup[x_+,X_+]\subset\mathbb{R}.$$

Abstände zwischen Fließkommazahlen

Festkommazahlen: Abstand zwischen Zahlen konstant β^{-k}.

Fließkommazahlen: Abstand von $x \in \mathbb{F}$ zum nächsten $x' \in \mathbb{F}$ hängt von $|x|$ ab
(Vorzeichen spielt keine Rolle)

$e-1 = z \Rightarrow e = z+1$

$$\beta^{e-1} = 0.\overset{m_1}{1}0\ldots\overset{m_r}{0} \cdot \beta^e$$

nächste Zahl: $0.10\ldots1 \cdot \beta^e$ } Abstand $\beta^{-r}\cdot\beta^e = \underline{\beta^{e-r}}$

$\vdots$

$\ldots\ldots\ldots(\beta-2)$ } Abstand β^{e-r} (da Exponent immer noch gleich)

$0.(\beta-1)(\beta-1)\ldots(\beta-1)\cdot\beta^e$

nächste Zahl $+\beta^{e-r}$ } Abstand β^{e-r}

Abstand springt!

$$\beta^e = 0.10\ldots0 \cdot \beta^{e+1}$$

nächste Zahl: $0.10\ldots1 \cdot \beta^{e+1}$ } Abstand $\beta^{-r}\cdot\beta^{e+1} = \underline{\beta^{e+1-r}}$

$\Rightarrow$ Im Intervall $[\beta^{e-1}, \beta^e]$ beträgt der absolute Abstand zwischen zwei Zahlen β^{e-r}.

← kein Rundungsfehler!

Sei $x' \in \mathbb{F}$ die nächstliegende Fließkommazahl zu $x \in \mathbb{F}$, $x>0$. Dann gilt

a) für $m(x) \neq \beta^{-1}$:

$$|x-x'| = \beta^{e(x)-r} \underset{\substack{\uparrow\\ x\neq0}}{=} \frac{|m(x)|\,\beta^{e(x)}}{|m(x)|}\cdot\beta^{-r} = \frac{|x|}{|m(x)|}\,\beta^{-r}$$

Da $\underbrace{\beta^{-1} \le |m(x)|}_{\frac{1}{m(x)} \le \beta} \underbrace{< 1}_{1 < \frac{1}{|m(x)|}}$ (nach (1.1)) folgt

groß, wenn $|m(x)|$ klein, d.h. β^{-1}

$$|x|\,\beta^{-r} < |x-x'| \le |x|\,\beta^{1-r}$$

b) $m(x) = \beta^{-1}$: $|x-x'| = \beta^{e-1-r} = \beta^{-1}\beta^{e(x)}\beta^{-r} = |x|\cdot\beta^{-r}$

Für den relativen Abstand gilt:

$$\beta^{-r} \le \frac{|x-x'|}{|x|} \le \beta^{1-r} \quad (= \beta\,\beta^{-r})$$

der Fall b

Er schwankt um den Faktor β je nach Größe von $|x|$. Dieser Faktor heißt wobble.

Übung:
β^{1-r}: Die kleinste Zahl $x \in \mathbb{F}$, so dass $1+x \neq 1$.
Schreibe Programm, das dieses x bestimmt.
Teste das für float, double.

Konsequenz: Kleine β sind besser!

IEEE 754 / IEC 559 Standard

Ziel: Portabilität von Programmen mit Fließkommazahlenarithmetik

Verabschiedet 1985.

$\beta = 2$ mit vier Genauigkeitsstufen und normierter Darstellung:

	Format single	single-ext	double	double-ext	
e_{max}	+127	≥ 1024	1023	≥ 16384	nicht sym.
e_{min}	−126	≤ -1021	−1022	≤ -16381	anders als in $\mathbb{F}(\beta,r,s)$
Bits Expon.	8	≤ 11	11	15	
Bits total	32	≥ 43	64	≥ 79	

Betrachte double genauer:

- total 64 bit
- 11 bit für Exponent. Dieser wird vorzeichenlos als Zahl $c \in [1, 2046]$ gespeichert.

  Setze $e = c - 1023 \Rightarrow e \in [\underset{1-1023}{-1022}, \underset{2046-1023}{1023}]$ ⤳ kein Vorzeichen extra notwendig!

  Die Werte $c \in \{0, 2047\}$ werden anderweitig genutzt:

  $c = 0 \wedge m = 0$ kodiert die Null
  $c = 0 \wedge m \neq 0$ kodiert denormalisierte Darstellung
  $c = 2047 \wedge m \neq 0$ kodiert NaN = „not a number" (z.B. Division durch Null)
  $c = 2047 \wedge m = 0$ kodiert ∞ (Überlauf).

- $64 - 11 = 53$ Bit Mantisse, 1 Bit für Vorzeichen, bleiben 52 Nachkommastellen

  Wg. $\beta = 2$ und Normierung ist immer $m_1 = 1$ und diese Ziffer wird nicht gespeichert. Heißt hidden bit.

  Somit gilt $\boxed{r = 53}$. (kleinste Wertigkeit)

- Der Standard definiert 4 Rundungsarten:

  nach $+\infty$, nach $-\infty$, nach $\emptyset$, nearest (natürliche Rundung)

1.2 Runden und Rundungsfehler

Um $x \in \mathbb{R}$ in $\mathbb{F}(\beta, r, s)$ zu approximieren brauchen wir

$$rd : \mathbb{D}(\beta, r, s) \longrightarrow \mathbb{F}(\beta, r, s) \qquad (1.2)$$

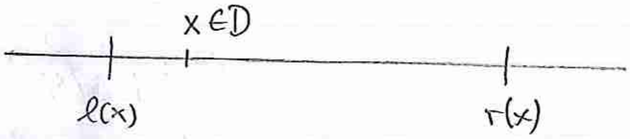
Achtung: rd setzt voraus, dass x im darstellbaren Bereich liegt!

Im Falle eines Über/Unterlaufes ist r, s zu ändern.

Sinnvollerweise soll für rd gelten:

$$|x - rd(x)| = \min_{y \in \mathbb{F}} |x - y| \quad \forall x \in \mathbb{D} \quad \text{(Bestapproximation)}.$$

Mit



$$l(x) = \max\{y \in \mathbb{F} \mid y \leq x\} \qquad r(x) = \min\{y \in \mathbb{F} \mid y \geq x\}$$

gilt dann

$$rd(x) = \begin{cases} x & l(x) = r(x),\ x \in \mathbb{F} \\ l(x) & |x - l(x)| < |x - r(x)| \\ r(x) & |x - l(x)| > |x - r(x)| \\ ? & |x - l(x)| = |x - r(x)| \end{cases}$$

Im letzten Fall ist eine Rundung erforderlich. Dafür gibt es verschiedene Möglichkeiten.

Sei $x = \operatorname{sign}(x) \cdot \left(\sum_{i=1}^{\infty} m_i \beta^{-i}\right) \beta^e$ die <u>normierte</u> Darstellung von $x \in \mathbb{D}$,
(alle Ziffern m_i sind korrekt (!), das ist der Unterschied zum TMD)

<u>Natürliche Rundung (das was jeder kennt)</u>

nächste Stelle

$$rd(x) = \begin{cases} l(x) = \operatorname{sign}(x) \left(\sum_{i=1}^{r} m_i \beta^{-i}\right) \beta^e & \text{falls } 0 \leq m_{r+1} < \beta/2 \\ r(x) = l(x) + \underbrace{\beta^{e-r}}_{\text{Wertigkeit der letzten Stelle}} & \text{falls } \beta/2 \leq m_{r+1} < \beta \end{cases}$$

Beispiel 1.4

$(x)_{10} = 110 \rightarrow x' \in \mathbb{F}(4,3,2)$

1) Exponent

$\log_4(110) = 3{,}39$ $\qquad 4^x = 110 \Rightarrow x \cdot \log 4 = \log 110$

$x = \frac{\log 110}{\log 4} = 3{,}39$

Aufrunden damit Mantisse < 1

$\Rightarrow$ Exponent $4 \ (10)$ d.h. $x = m \cdot 4^4 \Rightarrow m = \frac{x}{4^4}$

2) Mantisse

teile x durch 4^4 $\quad (m)_{10} = \frac{x}{4^4} = \frac{110}{256} = (0.4296875)_{10}$ (exakt!)

$y_1 = \frac{110}{4^4} = 0{,}4296875 = (m_1 \cdot 4^{-1} + m_2 \cdot 4^{-2} + m_3 \cdot 4^{-3} + \dots)$

$y_1 * 4 = 1{,}71875 \Rightarrow m_1 = 1$

$y_2 = (y_1 - m_1) * 4 = 2{,}875 \Rightarrow m_2 = 2$

$y_3 = (y_2 - m_2) * 4 = 3{,}5 \Rightarrow m_3 = 3$

$y_4 = (y_3 - m_3) * 4 = 2{,}0 \Rightarrow m_4 = 2$

3 Stellen ($r = 3$)

$(x)_4 = 0{,}1\,2\,3\,|\,2 \cdot 4^{10}$ (exakt)

② Runden mit gerader Rundung

$\Rightarrow (x')_4 = 1{,}30 \cdot 4^{10}$ (weil $m_3 = 3$, also ungerade)

Kontrolle: $(1{,}30 \cdot 4^{10})_4 = (112)_{10}$
$(1{,}23 \cdot 4^{10})_4 = (108)_{10}$ } das zeigt, dass x genau zwischen den beiden Zahlen ist.

① Natürliche Rundung

$m_{r+1} = m_4 = 2 = \frac{\beta}{2} \Rightarrow$ „aufrunden"

$rd(x) = (0{,}123 + 0{,}001) \cdot 4^{10} = 0{,}130 \cdot 4^{10}$ (alles zur Basis 4)

Gerade Rundung (β sei gerade)

$$rd(x) = \begin{cases} l(x) & (|x-l(x)| < |x-r(x)|) \vee \\ & (|x-l(x)| = |x-r(x)| \wedge m_r \text{ gerade}) \\ r(x) = l(x)+\beta^{e-r} & \text{sonst.} \end{cases}$$

Damit ist m_r in $rd(x)$ immer gerade, wenn gerundet wurde (dh. $|x-l(x)| = |x-r(x)|$)

- ist $rd(x) = l(x)$ so ist das per Def. so
- sonst ist $rd(x) = \underbrace{l(x)}_{m_r \text{ unger.}} + \underbrace{\beta^{e-r}}_{+1 \text{ in der letzten Stelle}}$ $\Rightarrow$ m_r gerade.

Diese Wahl vermeidet eine mögliche Drift beim Aufrunden. → Ü-Aufgabe.

Beispiel 1.4 → extra Blatt.

bis hier vorbereitet

Definition 1.5 (absoluter und relativer Fehler)

Sei $x' \in \mathbb{R}$ eine Näherung von $x \in \mathbb{R}$. Dann heißt

$$\Delta x = x' - x \qquad \text{absoluter Fehler}$$

und für $x \neq 0$

$$\varepsilon_{x'} = \frac{\Delta x}{x} \qquad \text{relativer Fehler.}$$

Umformen liefert:

$$x' = \underset{\substack{\uparrow \\ \text{abs. Fehler}}}{x + \Delta x} = x\left(1 + \frac{\Delta x}{x}\right) = x(1+\varepsilon_{x'})$$

Näherungswert = echter Wert $\cdot$ $(1+\varepsilon)$

Motivation:

Es sei $\Delta x = x' - x = 100$ km.

Für x = Entfernung Erde-Sonne $\approx 1.5 \cdot 10^8$ km ist

$$\varepsilon_{x'} = \frac{10^2 \text{ km}}{1{,}5 \cdot 10^8 \text{ km}} \approx 6.6 \cdot 10^{-7}$$

relativ klein. Für x = Entfernung Heidelberg-Paris ≈ 500 km ist

$$\varepsilon_{x'} = \frac{100 \text{ km}}{500 \text{ km}} = 0.2$$

dagegen relativ groß.

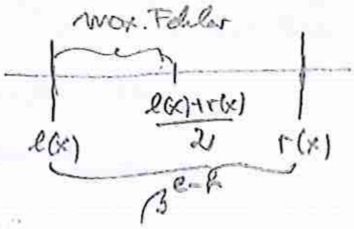
Lemma 1.5 (Rundungsfehler)

Bei der Rundung in $\mathbb{F}(\beta, r, s)$ gilt für den absoluten Fehler

$$|x - rd(x)| \le \frac{1}{2}\beta^{e(x)-r} \qquad (1.3)$$

und für den relativen Fehler $(x \neq 0)$

$$\frac{|x - rd(x)|}{|x|} \le \frac{1}{2}\beta^{1-r}$$



Beweis: Es sei $x = m(x)\beta^{e(x)}$ die normierte Darstellung von x.

Wie oben gezeigt gilt

$l = 0.m_1 \dots m_r \cdot \beta^e$ $\quad m_r < \beta - 1$

$r = 0.m_1 \dots (m_r+1) \cdot \beta^e$ $\quad$ siehe S. 4.

$|r - l| = \beta^{-r}\cdot\beta^e = \beta^{e-r}$

wg Fall b)

$$|\underbrace{r(x) - l(x)}_{\text{zwei aufeinander folg. Zahlen in } \mathbb{F}}| \le \beta^{e(l(x))-r}, \quad l(x) \in [\beta^{e(x)-1}, \beta^{e(x)}]$$

Der maximale Fehler ergibt sich für $x = \frac{l(x)+r(x)}{2}$, also

$$|x - rd(x)| \le \left|\frac{l(x)+r(x)}{2} - \underset{\substack{\uparrow \\ \text{egal}}}{l(x)}\right| = \frac{1}{2}|r(x) - l(x)| \le \frac{1}{2}\beta^{e(x)-r}.$$

Für den relativen Fehler $(x \neq 0)$ gilt

$$\frac{|x - rd(x)|}{|x|} \le \frac{\frac{1}{2}\beta^{e(x)-r}}{|m(x)|\beta^{e(x)}} = \frac{1}{2}\frac{1}{|m(x)|}\beta^{-r} \overset{|m(x)| \ge \beta^{-1}}{\le} \frac{1}{2}\beta^{1-r}. \qquad \blacksquare$$

Die Zahl $eps := \frac{1}{2}\beta^{1-r}$ heißt Maschinengenauigkeit. (Nennt Rannacher auch so).
Wird oft auch mit $\underline{u}$ (z.B. in MATLAB) abgekürzt.

Vorsicht: Die Bezeichnungen gehen in der Literatur durcheinander.

[Goldberg] nennt $\frac{1}{2}\beta^{1-r}$ machine epsilon.

[Quarteroni] nennt β^{1-r} machine epsilon und

$\frac{1}{2}\beta^{1-r}$ machine precision.

1.3 Fließkommaarithmetik

Wir benötigen eine Arithmetik auf $\mathbb{F}$:

$$\circledast : \mathbb{F} \times \mathbb{F} \to \mathbb{F} \quad \text{mit } \circledast \in \{\oplus, \ominus, \odot, \oslash\},$$

die den bekannten Operationen $* \in \{+, -, \cdot, /\}$ auf $\mathbb{R}$ entsprechen.

Problem: $x, y \in \mathbb{F} \Rightarrow x \circledast y \notin \mathbb{F}$ in der Regel!

Bsp Multiplikation
⇝ Verdopplung der Nachkommastellen.

Somit ist das Ergebnis $x \circledast y$ wieder zu runden, d.h. wir <u>definieren</u>

$$x \circledast y = rd(x * y) \quad \forall x, y \in \mathbb{F}. \qquad (1.4)$$

Man sagt $\circledast$ ist „exakt gerundet". Dies ist nicht trivial!
(implizite Annahme: $x * y \in \mathbb{D}$!)

<u>Beispiel 1.7 (Guard digit)</u>

Sei $\mathbb{F} = \mathbb{F}(10, 3, 1)$, $x = 0.215 \cdot 10^8$, $y = 0.125 \cdot 10^{-5}$. Wir betrachten die Subtraktion $x \ominus y = rd(x - y)$.

(1) $x - y$:

$x =$	0.215	000	000	000	000	0	$\cdot 10^8$
$y =$	$0.\underbrace{000}$	$\underbrace{000}$	$\underbrace{000}$	$\underbrace{000}_{13 \text{ Nullen!}}$	$0\,12$	5	$\cdot 10^8$
$x - y =$	0.214	999	999	999	$9\,87$	5	$\cdot 10^8$

← Schiebe y auf gleichen Exponenten.

$$y = 0.125 \cdot 10^{-5} = 0.125 \cdot \underbrace{10^{-13} \cdot 10^8}_{= 10^{-5}}$$

(2) Runden

$$x \ominus y = rd(0.2149\ldots \cdot 10^8) = 0.215 \cdot 10^8$$

Problem: Schritt 1 erfordert extrem hohe Stellenzahl $O(\beta^s)$!

Bemerkung zum Tabellenmacherdilemma.

Geht es einfacher? Z.B. Runde y schon nach dem Schieben.
Dies liefert im obigen Fall das gleiche Ergebnis.

Ü: Tabellenmacherdilemma. Tabellieren von transzendenten Funktionen.
$\mathbb{F} = \mathbb{F}(10, 4, 1)$, Erstelle Liste $y = \exp(x)$ $\forall x \in \mathbb{F}$ mit $y = rd(\exp(x))$

Problem: $\exp(1.626) = \underbrace{5.0835}$

runden auf 3 Stellen nach d. Komma 5 Stellen
genau $\exp(1.626) = 5.0835000\ldots$ die 5 ist noch [illegible]
↓ möglich nächste Stelle negativ
$5.08349999\ldots$

ich verstehe es nicht, wenn 5 eine <u>gültige</u> Ziffer ist, dann muss man aufrunden

Im allgemeinen ist das gefährlich:

$$\begin{array}{llll} x = 0.101 \cdot 10^1 & \longrightarrow & 0.101 & \cdot 10^1 \\ y = 0.993 \cdot 10^0 & \xrightarrow[\text{schieben}]{} & 0.099 & \cdot 10^1 \\ \hline & & 0.002 & \cdot 10^1 \end{array}$$

Schieben und runden

relativer Fehler im Ergebnis:

$$\frac{(x \ominus y) - (x-y)}{x-y} = \frac{0.02 - 0.017}{0.017} \approx 0.176 \approx 35 \text{ eps}$$

mit $\text{eps} = \frac{1}{2} 10^{1-3} = 0.005$.

→ Fehler ist 35 mal größer als erwartet.

Eine Stelle mehr im Addierer (also r+1) liefert das exakte Ergebnis!

Mit einer zusätzlichen Stelle erreicht man

$$\frac{(x \ominus y) - (x-y)}{x-y} \le 2 \text{ eps}.$$

Mit zwei Stellen erreicht man exakte Rundung!

Die zusätzlichen Stellen nennt man „guard digits".

- IEEE 754: $\oplus, \ominus, \otimes, \oslash$ sind exakt gerundet ebenso $\sqrt{x}$
- Tabellenmacherdilemma

Zusätzliche Probleme bei der Arithmetik $\circledast: \mathbb{F} \times \mathbb{F} \to \mathbb{F}$:

- Assoziativ und Distributivgesetz gelten nicht, es kommt also auf die Reihenfolge der Operationen an!
- $\exists\, y \in \mathbb{F}$ so dass $x \oplus y = x$ ← etwa ~~$(1+\varepsilon)-1$ bzw $(1-1)+\varepsilon$~~ $\underbrace{(\varepsilon+1)-1}_{=0}$ vs. $\underbrace{\varepsilon+(1-1)}_{=\varepsilon}$
- Allerdings gilt das Kommutativgesetz!
- Es gelten auch folg. einfache Regeln:

$(-x) \odot y = -(x \odot y)$, $\quad 1 \odot x = x$, $\quad x \odot y = 0 \Rightarrow x = 0 \vee y = 0$,

$x \odot z \le y \odot z$ falls $x \le y$ und $z > 0$.

1.4 Fehleranalyse

Fortpflanzung von Rundungsfehlern in Rechnungen.

~~Wie in Kap. 1 betrachten wir die Funktionsauswertung~~.

- Sei $F : \mathbb{R}^m \to \mathbb{R}^n$, in Komponenten

$$F(x) = \begin{pmatrix} F_1(x_1, \dots, x_m) \\ \vdots \\ F_n(x_1, \dots, x_m) \end{pmatrix}$$

- Zur Berechnung von F im Rechner nutze numerische Realisierung

$F' : \mathbb{F}^m \to \mathbb{F}^n$. F' wird durch einen Algorithmus realisiert, d.h. aus

-- endlich vielen (= Terminierung)

-- elementaren (= bekannte) Rechenoperationen (d.h. $\oplus, \ominus, \odot, \oslash$)

zusammengesetzt: $F'(x) = \varphi_\ell(\dots \varphi_2(\varphi_1(x))\dots)$.

Wichtig: i) Zu einem F gibt es i.d.R. viele Realisierungen im Sinne unterschiedlicher Reihenfolgen

$$a + b + c \approx (a \oplus b) \oplus c \neq a \oplus (b \oplus c) \,!$$

ii) Jedes φ_i steuert einen (unbekannten) Fehler bei

iii) Im Prinzip kann die Rechengenauigkeit beliebig gesteigert werden, d.h. eigentlich Folge $F^{(k)} : (\mathbb{F}^{(k)})^m \to (\mathbb{F}^{(k)})^n$.
Das machen wir aber nicht so formal.

~~Wie in Kap. 1~~ Nutze Aufspaltung:

$$\underbrace{F(x)}_{\substack{\text{exaktes}\\ \text{Ergebnis}}} \;\overset{\mathbb{F} \subseteq \mathbb{R}}{-}\; \underbrace{F'(\underbrace{rd(x)}_{\substack{\text{Eingabe}\\ \text{runden}}})}_{\substack{\text{numerische}\\ \text{Auswertung}}} = \underbrace{F(x) - F(rd(x))}_{\substack{①\ \text{Konditionsanalyse}\\ \text{von } F}} + \underbrace{F(rd(x)) - F'(rd(x))}_{\substack{②\ \text{Rundungsfehleranalyse}\\ \text{Unterschied } F, F' \text{ bei}\\ \text{gleicher Eingabe (aus } \mathbb{F})}} \qquad (1.5) \; \cancel{(2.4)}$$

$$-F(rd(x)) + F(rd(x)) = 0$$

von hier aus:

- Analyse „in erster Näherung"
- absolute/relative Fehler.
- Normen bilden, das lassen wir aber i.d.R. weg.

(1) Differentielle Konditionsanalyse

Wir nehmen an, dass $F: \mathbb{R}^m \to \mathbb{R}^n$ zweimal stetig differenzierbar.
Nach dem Satz von Taylor gilt für die Komponenten F_i:

$$F_i(x+\Delta x) = F_i(x) + \sum_{j=1}^{m} \frac{\partial F_i}{\partial x_j}(x)\, \Delta x_j + R_i^F(x;\Delta x) \qquad i = 1,\dots,n.$$

Für das Restglied gilt (unter diesen Voraussetzungen)

$$R_i^F(x;\Delta x) = O(\|\Delta x\|^2).$$

Definition ~~2.7~~ 1.8 (Landausche Symbole)

Man schreibt:

$$g(t) = O(h(t)) \qquad (t \to 0)$$

falls es $t_0 > 0$ und $c_0 \geq 0$ gibt so dass für alle $t \in (0, t_0]$ die Abschätzung

$$|g(t)| \leq c\, |h(t)|$$

gilt. Sprechweise: „$g(t)$ geht wie $h(t)$ gegen 0". Man will also quantifizieren „wie schnell" eine Funktion (mindestens) gegen $\emptyset$ geht.

Weiter bedeutet

$$g(t) = o(h(t)) \qquad (t \to 0),$$

dass es $t_0 > 0$ und eine Funktion $c(t)$, $\lim_{t \to 0} c(t) = 0$ gibt, sodass für alle $t \in (0, t_0]$ gilt

$$|g(t)| \leq c(t)\, |h(t)|.$$

Bedeutung: „$g(t)$ geht schneller als $h(t)$ gegen Null" (falls $h(t) \to 0$). ∎

Somit können wir die Taylorformel umformen:

$$F_i(x+\Delta x) - F_i(x) = \underbrace{\sum_{j=1}^{m} \frac{\partial F_i}{\partial x_j}(x)\, \Delta x_j}_{\text{„führende (erste) Ordnung"}} + \underbrace{O(\|\Delta x\|^2)}_{\text{höhere Ordng}} .$$

Oft lässt man die Terme höherer Ordnung weg und schreibt „$\doteq$" statt „$=$". Sprechweise: „ist in erster Näherung gleich".

Nun gehen wir zu relativen Fehlern über. Sei $F_i(x) \neq 0$ und $x_j \neq 0$

$$\frac{F_i(x+\Delta x) - F_i(x)}{F_i(x)} \doteq \sum_{j=1}^{m} \frac{\partial F_i}{\partial x_j}(x) \frac{\Delta x_j}{F_i(x)}$$

nur eine Komponente

$$\doteq \sum_{j=1}^{m} \underbrace{\left(\frac{\partial F_i}{\partial x_j}(x) \frac{x_j}{F_i(x)}\right)}_{\text{Verstärkungsfaktor } \bar{k}_{ij}(x)} \left(\frac{\Delta x_j}{x_j}\right) \qquad \text{1.6} \quad \cancel{(2.5)}$$

relativer Eingabefehler. $\leq$ eps!

Fasst man die Verstärkungsfaktoren in einer Matrix

$$(\bar{K}(x))_{ij} = \bar{k}_{ij}(x)$$

zusammen so kann man (für geeignete Normen) zeigen

$$\|\bar{K}(x)\| \leq K(x)$$

mit $K(x)$ der relativen Konditionszahl aus Kapitel 1.

Ü: Setze für $x = (x_1, \dots, x_m)^T$, $x_i \neq 0$ „$\frac{\Delta x}{x}$" $= \left(\frac{\Delta x_1}{x_1}, \dots, \frac{\Delta x_m}{x_m}\right)^T$

$$\frac{\|\Delta x\|_\infty}{\|x\|_\infty} \leq \left\| \text{„}\frac{\Delta x}{x}\text{"} \right\|_\infty \leq \frac{\|\Delta x\|_\infty}{\min_i x_i}$$

Kap 1

$$K(x) = \sup \left\{ \frac{\|F(x+\delta x) - F(x)\|}{\|\delta x\|} \cdot \frac{\|x\|}{\|F(x)\|} \right.$$

$$\Rightarrow \frac{\frac{\|F(x+\Delta x) - F(x)\|_\infty}{\|F(x)\|_\infty}}{\|\frac{\Delta x}{x}\|_\infty} \leq \frac{\cdot}{\frac{\|\Delta x\|_\infty}{\|x\|_\infty}} = K(x)$$

Definition ~~2.8~~ 1.9 Wir nennen die Auswertung $y = F(x)$ „schlecht konditioniert", im Punkt x, falls $|\kappa_{ij}(x)| \gg 1$, andernfalls „gut konditioniert".
$|\kappa_{ij}(x)| < 1$ heißt Fehlerdämpfung, $|\kappa_{ij}(x)| > 1$ Fehlerverstärkung. ◻

Warum relative Kondition?

Wegen Lemma ~~2.5~~ 1.6 gilt

$$\left|\frac{x - rd(x)}{x}\right| \le eps = \frac{1}{2}\beta^{1-r},$$

d.h. es gibt $\varepsilon \in \mathbb{R}$, $|\varepsilon| \le eps$, sodass

$$\frac{x - rd(x)}{x} = \varepsilon \iff x - rd(x) = \varepsilon x \iff rd(x) = x + \underbrace{\varepsilon x}_{=:\Delta x}$$

d.h. für die relativen Eingabefehler in ~~(2.5)~~ 1.6 gilt gerade

$$\frac{\Delta x_j}{x_j} = \frac{\varepsilon_j x_j}{x_j} = \varepsilon_j .$$

Beispiel ~~2.9~~ 1.10 Ü: Kondition von $F(x_1, x_2) = x \cdot y$, x/y, $F(x) = \sqrt{x}$

a) Addition. $F(x_1, x_2) = x_1 + x_2$, $\frac{\partial F}{\partial x_1} = 1$, $\frac{\partial F}{\partial x_2} = 1$. Nach obiger Formel ~~(2.5)~~ 1.6:

$$\frac{F(x_1+\Delta x_1, x_2+\Delta x_2) - F(x_1,x_2)}{F(x_1,x_2)} = \underbrace{1 \cdot \frac{x_1}{x_1+x_2}}_{=\kappa_1} \cdot \underbrace{\frac{\Delta x_1}{x_1}}_{|\cdot| \le eps} + \underbrace{1 \cdot \frac{x_2}{x_1+x_2}}_{=\kappa_2} \cdot \underbrace{\frac{\Delta x_2}{x_2}}_{|\cdot| \le eps}$$

$\Rightarrow$ Für $x_1 \to -x_2$ gehen beide Verstärkungsfaktoren gegen ∞ (sofern $|x_1|, |x_2| > \delta$).
Schlecht konditioniert!

b) $F(x_1, x_2) = x_1^2 - x_2^2$, $\frac{\partial F}{\partial x_1} = 2x_1$, $\frac{\partial F}{\partial x_2} = -2x_2$.

$$\frac{F(x_1+\Delta x_1, x_2+\Delta x_2) - F(x_1,x_2)}{F(x_1,x_2)} = \underbrace{2x_1 \cdot \frac{x_1}{x_1^2 - x_2^2}}_{\kappa_1 = 2\frac{x_1^2}{x_1^2-x_2^2}} \cdot \underbrace{\frac{\Delta x_1}{x_1}}_{|\cdot| \le eps} - \underbrace{2x_2 \cdot \frac{x_2}{x_1^2-x_2^2}}_{\kappa_2 = 2\frac{x_2^2}{x_2^2-x_1^2}} \cdot \underbrace{\frac{\Delta x_2}{x_2}}_{|\cdot| \le eps}$$

Schlecht konditioniert für $|x_1| \approx |x_2|$. ◻

② Rundungsfehleranalyse

d.h. Nach ~~2.4~~ 1.5: $F(x) - F'(x)$ mit $x \in \mathbb{F}^m$ Maschinenzahl.

· F' „zusammengesetzt" aus Einzeloperationen $\circledast \in \{\oplus, \ominus, \odot, \oslash\}$.

Wegen (~~2.3~~ 1.4) (exakt gerundete Arithmetik) und Lemma ~~2.5~~ 1.6 gilt

$$\frac{(x \circledast y) - (x * y)}{(x * y)} = \varepsilon \qquad \text{mit } |\varepsilon| \leq \text{eps}.$$

Vorsicht: ε ist abhängig von x und y, d.h. für jede Operation verschieden!

Und damit

$$x \circledast y = (x * y)(1+\varepsilon) \qquad \text{für ein } |\varepsilon(x,y)| \leq \text{eps}.$$

Analyse „in erster Näherung"

<u>Beispiel ~~2.10~~ 1.11</u> $F(x_1,x_2) = x_1^2 - x_2^2$ mit zwei Realisierungen

$$F_a(x_1,x_2) = (x_1 \odot x_1) \ominus (x_2 \odot x_2)$$

$$F_b(x_1,x_2) = (x_1 \ominus x_2) \odot (x_1 \oplus x_2)$$

a)
$$u = x_1 \odot x_1 = (x_1 \cdot x_1)(1+\varepsilon_1)$$
$$v = x_2 \odot x_2 = (x_2 \cdot x_2)(1+\varepsilon_2) \qquad \varepsilon_1 \neq \varepsilon_2 \text{ aber } |\varepsilon_i| \leq \text{eps}!$$

$$\begin{aligned}
F_a(x_1,x_2) = u \ominus v &= (u-v)(1+\varepsilon_3) \\
&= \left(x_1^2(1+\varepsilon_1) - x_2^2(1+\varepsilon_2)\right)(1+\varepsilon_3) \\
&= \left(x_1^2 + \varepsilon_1 x_1^2 - x_2^2 - \varepsilon_2 x_2^2\right)(1+\varepsilon_3) \\
&= \underbrace{x_1^2 - x_2^2}_{=F(x_1,x_2)} \underbrace{+\varepsilon_1 x_1^2 - \varepsilon_2 x_2^2 + \varepsilon_3 x_1^2 - \varepsilon_3 x_2^2}_{\text{erste Ordnung}} + \underbrace{\varepsilon_1\varepsilon_3 x_1^2 - \varepsilon_2\varepsilon_3 x_2^2}_{\text{zweite Ordnung}}
\end{aligned}$$

relativer Fehler

$$\frac{F_a(x_1,x_2) - F(x_1,x_2)}{F(x_1,x_2)} \overset{\circ}{=} \frac{x_1^2}{x_1^2 - x_2^2}(\varepsilon_1 + \varepsilon_3) + \frac{x_2^2}{x_2^2 - x_1^2}(\varepsilon_2 + \varepsilon_3)$$

in erster Näherung

<u>Bemerkung</u>: gleiche Verstärkungsfaktoren wie in Bsp. ~~2.9~~ 1.10!

b) $u = x_1 \ominus x_2 = (x_1 - x_2)(1+\varepsilon_1)$

$v = x_1 \oplus x_2 = (x_1 + x_2)(1+\varepsilon_2)$

$$F_B(x_1, x_2) = u \odot v = (u \cdot v)(1+\varepsilon_3)$$
$$= \big((x_1-x_2)(1+\varepsilon_1) \cdot (x_1+x_2)(1+\varepsilon_2) \big)(1+\varepsilon_3)$$
$$= \underbrace{(x_1-x_2)(x_1+x_2)}_{\circ} \underbrace{(1+\varepsilon_1)(1+\varepsilon_2)(1+\varepsilon_3)}_{\circ}$$
$$= (x_1^2 - x_2^2)(1 + \varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_1\varepsilon_2 + \ldots + \varepsilon_1\varepsilon_2\varepsilon_3)$$

$$\frac{F_B(x_1,x_2) - F(x_1,x_2)}{F(x_1,x_2)} \doteq \frac{x_1^2 - x_2^2}{x_1^2 - x_2^2}(\varepsilon_1 + \varepsilon_2 + \varepsilon_3) = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$$

$\Rightarrow$ Verstärkungsfaktor 1, besser als vorher.

Definition ~~2.11~~ 1.12

Wir nennen einen numerischen Algorithmus „numerisch stabil", wenn die im Lauf der Rechnung akkumulierten Rundungsfehler aus (2) den unvermeidbaren Problemfehler aus der Konditionsanalyse (1) nicht übersteigen.

a.w. Verstärkungsfaktoren aus Rundungsfehleranalyse $\leq$ denen aus Konditionsanalyse $\Rightarrow$ „numerisch stabil"

Beide Realisierungen a, b aus Bsp. ~~2.10~~ 1.11 sind numerisch stabil.

1.5 Auslöschung

Obige Beispiele 1.10, 1.11 ~~2.9, 2.10~~ enthalten das Phänomen der <u>Auslöschung</u>.

Dies tritt auf bei

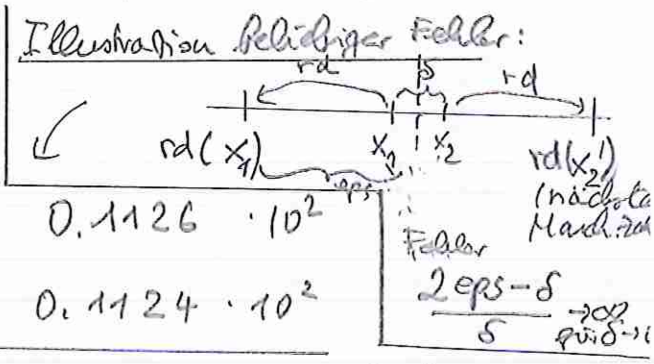
- Addition $x_1 + x_2$ mit $x_1 \approx -x_2$,
- Subtraktion $x_1 - x_2$ mit $x_1 \approx x_2$.

<u>Bemerkung 1.13 ~~2.12~~</u> Bei der Auslöschung werden <u>vor</u> der entsprechenden Addition bzw. Subtraktion eingeführte Fehler extrem verstärkt. ☒

Sind $x_1, x_2 \in \mathbb{F}$ <u>Maschinenzahlen</u>, so gilt wie oben gezeigt:

$$\left| \frac{(x_1 \ominus x_2) - (x_1 - x_2)}{x_1 - x_2} \right| \leq \text{eps}.$$

Also kein Problem. Das Problem tritt erst ein, wenn x_1, x_2 selbst schon mit Fehlern behaftet sind.



<u>Beispiel 1.14 ~~2.13~~</u> $\mathbb{F} = \mathbb{F}(10, 4, 1)$

$$x_1 = 0.11258762 \cdot 10^2 \quad \to \quad rd(x_1) = 0.1126 \cdot 10^2$$

$$x_2 = 0.11244891 \cdot 10^2 \quad \to \quad rd(x_2) = 0.1124 \cdot 10^2$$

$$x_1 \ominus x_2 = 0.00013871 \cdot 10^2 \qquad rd(x_1) - rd(x_2) = 0.0002 \cdot 10^2$$

$$= 0.13871 \cdot 10^{-1} \qquad = \underbrace{0.2 \cdot 10^{-1}}_{\text{keine gültige Ziffer!}}$$

~~überhaupt kein Fehler~~

relativer Fehler:

$$\frac{0.2 \cdot 10^{-1} - 0.13871 \cdot 10^{-1}}{0.13871 \cdot 10^{-1}} \approx 0{,}44 \approx 883 \cdot \underbrace{\tfrac{1}{2} 10^{-3}}_{= \text{eps}} \quad !$$ ☒

Hier: Rundung der Eingangsgrößen.

Herkunft der Fehler ist egal, tritt ebenso auf, wenn x_1, x_2 mit Fehlern vorhergehender Rechenschritte behaftet sind.

<u>Regel 1.15 ~~2.14~~</u> Setze potentiell gefährliche Operationen möglichst früh im Algorithmus ein. (Siehe Beispiel 1.19). ☒

1.6 Die quadratische Gleichung

Die Gleichung

$$y^2 - py + q = 0$$

hat für $p^2/4 > q \neq 0$ die beiden reellen und verschiedenen Lösungen

$$y_{1,2} = f_\pm(p,q) = \frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}\,. \qquad \text{(Definiert zwei } f\text{'s!)}$$

Die Konditionsanalyse liefert:

$$\frac{f(p+\Delta p, q+\Delta q) - f(p,q)}{f(p,q)} \doteq \left(1 \pm \frac{p}{2\underbrace{\sqrt{\frac{p^2}{4} - q}}_{(a)}}\right) \frac{p}{\underbrace{p \pm 2\sqrt{\frac{p^2}{4} - q}}_{(b)}} \frac{\Delta p}{p} \mp \frac{q}{\underbrace{\sqrt{\frac{p^2}{4} - q}}_{(a)} \underbrace{\left(p \pm 2\sqrt{\frac{p^2}{4} - q}\right)}_{(b)}} \frac{\Delta q}{q}$$

$\Rightarrow$
- Für $\frac{p^2}{4} \gg q$ und $p < 0$ ist $f_-(p,q) = \frac{p}{2} - \sqrt{\frac{p^2}{4} - q}$ gut konditioniert
- Für $\frac{p^2}{4} \gg q$ und $p > 0$ ist $f_+(p,q) = \frac{p}{2} + \sqrt{\frac{p^2}{4} - q}$ gut konditioniert
- für $\frac{p^2}{4} \approx q$ sind f_+ und f_- schlecht konditioniert ($(a) \to 0$)

Numerisch stabile Auswertung für den Fall $p^2/4 \gg q$.

$p < 0$: Berechne $y_2 = \underbrace{\frac{p}{2} - \sqrt{\underbrace{\frac{p^2}{4} - q}_{\text{keine Ausl. n. Vor.}}}}_{\text{keine Ausl. da } p<0}$ $\qquad y_1 = q/y_2$ nach Vieta:

$$p = y_1 + y_2$$
$$q = y_1 \cdot y_2$$

$p > 0$: Berechne $y_1 = \underbrace{\frac{p}{2} + \sqrt{\underbrace{\frac{p^2}{4} - q}}}$ $\qquad y_2 = q/y_1$.