Vorlesung Numerische Mathematik 1
Kapitel 2: Rechnerarithmetik
Studiengang Informatik

14. September 2017

Zürcher Hochschule
für Angewandte Wissenschaften

Gliederung des Kapitels 2

1. Geschichte der Zahlendarstellung
2. Maschinenzahlen
3. Umrechnung zwischen Basen
   - ... ins Dezimalsystem
   - ... in andere Zahlensysteme
4. Approximations- und Rundungsfehler
   - Rundungsfehler und Maschinengenauigkeit
   - Fehlerfortpflanzung / Konditionierung

Lernziele

- Sie verstehen die Definition der maschinendarstellbaren Zahlen und können Gleitpunktzahlen zwischen verschiedenen Basen umrechnen.
- Sie können die Fehler, die beim Abbilden von reellen Zahlen auf Maschinenzahlen entstehen, sowie die Maschinengenauigkeit berechnen.
- Sie können die Fortpflanzung von Fehlern bei Funktionsauswertungen abschätzen und die Konditionszahl berechnen.

Geschichte der Zahlendarstellung

- In den frühen Hochkulturen entwickelten sich unterschiedliche Konzepte zur Darstellung von Zahlen, die nach Art der Zusammenstellung und der Anordnung der Ziffern in Additionssysteme und Positionssysteme (auch Stellenwertsysteme genannt) einteilbar sind:
  - Additionssysteme ordnen jeder Ziffer eine bestimmte Zahl zu.
  - Im Gegensatz dazu ordnen Positions- oder Stellenwertsysteme jeder Ziffer aufgrund der relativen Poistion zu anderen Ziffern eine Zahl zu: 25 -> 52

Geschichte der Zahlendarstellung

- Alle Zahlensysteme bauen dabei auf einer sogenannten ganzzahligen Grundzahl $B > 1$, auch Basis genannt, auf.
- Vor allem wurden die Zahlen 2, 5, 10, 12, 20 und 60 benutzt. Die wohl wichtigsten Grundzahlen sind 2 und 10. Von besonderem Interesse für die Babylonier war die Zahl 60, da sie zugleich die Zahl 30, also ungefähr die Anzahl Tage in einem Monat, als auch die Zahl 12, die Anzahl Monate in einem Jahr, als Teiler besitzt.

Geschichte der Zahlendarstellung

- Ägyptisches Additionssystem (ca. 3000 v. Chr.)

Abbildung: Symbole zum Darstellen von Zahlen bei den antiken Ägyptern: Ein Strich war ein Einer, ein umgekehrtes U ein Zehner, die Hunderter wurden durch eine Spirale, die Tausender durch die Lotusblüte mit Stil und die Zehntausender druch einen oben leicht angewinkelten Finger dargestellt. Dem Hunderttausender entsprach eine Kaulquappe mit hängendem Schwanz. Ergänzend ohne Bild hier: Die Millionen wird durch einen Genius, der die Arme zum Himmel erhebt, repräsentiert (aus [7]).

Geschichte der Zahlendarstellung

- Babylonisches Positionssystem (ca. 2000 v. Chr.)

Abbildung: Babylonische Form der Zahl
$46821 = 13 \cdot 60^2 + 0 \cdot 60^1 + 21 \cdot 60^0$ als $13 \mid 0 \mid 21$ (aus [7]).

Geschichte der Zahlendarstellung

- Sollen die Ziffern unabhängig von der Position verwendet werden, kommen wir also um die Darstellung der Ziffer Null nicht herum.
- Wir sind dann z.B. in der Lage, die beiden Zahlen $701 = 7 \cdot 10^2 + 0 \cdot 10^1 + 1 \cdot 10^0$ und $71 = 7 \cdot 10^1 + 1 \cdot 10^0$ zu unterscheiden.
- Die Ziffer 0 deutet das Auslassen einer "Stufenzahl" B^i an und ermöglicht eine übersichtlichere Darstellung in der modernen Nomenklatur

$$z = \sum_{i=0}^{n} z_i B^i.$$

Geschichte der Zahlendarstellung

- Indisch-Arabisches Zahlensystem (ca. 3. Jhr.v.Chr. bis 5. Jhr. n.Chr.)

Abbildung: Arabische und indische Symbole zum Darstellen von Zahlen: In der ersten Zeile sehen wir die indischen Ziffern des 2. Jahrhunderts n.Chr. Diese bildhaften Ziffern wurden erst von den Arabern übernommen (zweite Zeile) und später von den Europäern (dritte bis sechste Zeile: 12., 14, 15. und 16. Jhr.) immer abstrakter dargestellt (aus [7]).

Geschichte der Zahlendarstellung

- In Europa galt die Ziffer Null lange als Teufelswerk und findet sich erstmalig in einer Handschrift von 976.
- Bis ins Mittelalter wurden in Europa Zahlen in lateinischen Grossbuchstaben geschrieben.
- Im römischen Zahlensystem standen I, V, X, L, C, D und M für 1, 5, 10, 50, 100, 500 und 1000.
- Beispiel: MMMDCCCLXXVI = 3876.
- Zum Rechnen war dieses Zahlensystem allerdings kaum geeignet und wurde im Laufe des 13. Jahrhunderts von den arabischen Ziffern abgelöst.

Geschichte der Zahlendarstellung

- Das neue Zahlensystem mit der Null ermöglichte auch das Automatisieren von Rechenschritten.
- Im Jahre 1671 entwickelte Gottfried Wilhelm Leibniz seine Rechenmaschine, die REPLICA, die bereits alle vier Grundrechenarten beherrschte.
- Kurz darauf beschrieb er das binäre (oder auch duale) Zahlensystem, ohne das die heutige elektronische Datenverarbeitung kaum vorstellbar wäre.
- Auf dieses und weitere Zahlensysteme und die Implikationen auf arithmetische Operationen wollen wir im weiteren näher eingehen.

Maschinenzahlen
Mantisse und Exponent

- Die Menge der reellen Zahlen $\mathbb{R}$ hat unendlich viele Elemente.
- Jede Rechenmaschine ist aber ein endlicher Automat, d.h. er kann aufgrund der beschränkten Stellenzahl nicht alle Zahlen exakt darstellen und nur endlich viele Operationen ausführen.
- Für eine gegebene Basis $B \in \mathbb{N}$ kann jede reelle Zahl $x \in \mathbb{R}$ aber als

$$x = m \times B^e$$

dargestellt werden, wobei $m \in \mathbb{R}$ die **Mantisse** und $e \in \mathbb{Z}$ der **Exponent** genannt wird.

Maschinenzahlen
Verschiedene Basen

- Computerintern wird üblicherweise die Basis $B = 2$ verwendet (als Binär- od. Dualzahlen benannt), dies als direkte Folge der Zustände 'Strom' / 'kein Strom' (bzw. 1 und 0) von mikroelektronischen Schaltungen.
- Man spricht hier von einem *Bit* ('binary digit')[1]
- Weitere Basen sind $B = 8$ (Oktalz.), $B = 10$ (Dezimalz.) und $B = 16$ (Hexadez.).
- Für letztere benötigt man 16 verschiedene Zeichen und verwendet die Ziffern 0,1,...,9 sowie $A,...,F$ (wobei $A \triangleq 10$, $B \triangleq 11$ etc., auch Kleinbuchstaben sind erlaubt).

[1] Gemäss [7] wurde der Begriff *bit* das erste Mal wahrscheinlich von John Tukey (amerikanischer Mathematiker, 1915 - 2000, Träger der IEEE 'Medal of Honor') verwendet, als kürzere Alternative zu *bigit* oder *binit.* Das Wort digit kommt aus dem Lateinischen und bedeutet Finger.

Maschinenzahlen

Aufgabe 2.1:

1. Überlegen Sie sich: wie viele verschiedene Möglichkeiten gibt es, mit Binärzahlen ein Byte zu füllen?
2. Wieviele Ziffern bräuchten Sie im Hexadezimalsystem, um die gleiche Anzahl Möglichkeiten zu erhalten?
3. Was folgern Sie daraus bzgl. der Vorteile des Hexadezimalsystems?

Maschinenzahlen

Definition 2.1: Maschinenzahlen / Gleitpunktzahlen

- Im Rechner stehen nun nur endlich viele Stellen für m und e zur Verfügung, z.B. n Stellen für m und l Stellen für e.
  - Wir schreiben $m = \pm 0.m_1 m_2 \ldots m_n$ und $e = \pm e_1 e_2 \ldots e_l$. Dabei gilt $m_i,\ e_i \in \{0,1,\ldots,\ B\text{-}1\}$.
  - Unter der zusätzlichen Normierungs–Bedingung $m_1 \neq 0$ (für $x \neq 0$) ergibt sich eine eindeutige Darstellung der sogenannten ***maschinendarstellbaren Zahlen***

$$M = \left\{ x \in \mathbb{R} \,|\, x = \pm 0.m_1 m_2 m_3 \ldots m_n \cdot B^{\pm e_1 e_2 \ldots e_l} \right\} \cup \{0\}$$

    Der Wert (im Dezimalsystem) einer solchen Zahl, ist $\sum_{i=1}^{n} m_i B^{e-i}$ (e hier im Dezimalsystem).
  - Man spricht dann auch von einer ***n*-stelligen Gleitpunktzahl zur Basis *B***.
- Zahlen, die nicht in dieser Menge M liegen, müssen durch Rundung in eine maschinendarstellbare Zahl umgewandelt werden.

Maschinenzahlen
Bemerkungen

- Der Exponent $e \in \mathbb{Z}$ definiert, wie wir es vom Dezimalsystem kennen, die Position des Dezimalpunktes, also z.B. $x = 112.78350 = 112.78350 \cdot 10^0 = 1127835.0 \cdot 10^{-4} = 0.11278350 \cdot 10^3$.
- Um Missverständnisse zu vermeiden, kann die Basis explizit als Index zu einer Mantisse in Klammern angegeben werden. Wird kein Exponent angegeben, ist das gleichbedeutend mit $e = 0$, z.B. $(1011100.111)_2 = 1011100.111 \cdot 2^0 = 0.1011100111 \cdot 2^7 = (0.1011100111)_2 \cdot 2^7$.
- Die in Definition 2.1 gewählte Normierungsbedingung, kann auch durch andere Normierungsbedingungen ersetzt werden. Was wären weitere Möglichkeiten? Weshalb normiert man überhaupt?

Maschinenzahlen

Beispiele 2.1: Normierte Gleitpunktzahlen

1. Normierte Gleitpunktzahlen (gemäss Definition 2.1):
   1. $x_1 = -0.2345 \cdot 10^3$ ist eine vierstellige Gleitpunktzahl im Dezimalsystem mit dem Wert
   $$-\sum_{i=1}^{4} m_i \cdot 10^{3-i} = -(2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0 + 5 \cdot 10^{-1}) = -234.5$$
   2. $x_2 = 0.111 \cdot 2^3$ ist eine dreistellige Gleitpunktzahl im Binär-/Dualsystem mit dem Wert
   $$\sum_{i=1}^{3} m_i \cdot 2^{3-i} = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 7 \ (= 0.7 \cdot 10^1)$$
   3. $x_3 = 0.1001 \cdot 2^{-3}$ ist eine vierstellige Gleitpunktzahl im Binär-/Dualsystem mit dem Wert
   $$\sum_{i=1}^{4} m_i \cdot 2^{-3-i} = 2^{-4} + 2^{-7} = 0.0703125 \ (= 0.703125 \cdot 10^{-1})$$

Maschinenzahlen

Beispiele 2.1: Normierte Gleitpunktzahlen

(d) $x_4 = 0.71537 \cdot 8^2$ ist eine fünfstellige Gleitpunktzahl im Oktalsystem mit dem Wert

$$\begin{aligned} \sum_{i=1}^{5} m_i \cdot 8^{2-i} &= 7 \cdot 8^1 + 1 \cdot 8^0 + 5 \cdot 8^{-1} + 3 \cdot 8^{-2} + 7 \cdot 8^{-3} \\ &= 57.685546875 \; (= 0.57685546875 \cdot 10^2) \end{aligned}$$

(e) $x_5 = 0.AB3C9F \cdot 16^4$ ist eine sechsstellige Gleitpunktzahl im Hexadezimalsystem mit dem Wert

$$\begin{aligned} \sum_{i=1}^{6} m_i \cdot 16^{4-i} &= 10 \cdot 16^3 + 11 \cdot 16^2 + 3 \cdot 16^1 + 12 \cdot 16^0 + 9 \cdot 16^{-1} + 15 \cdot 16^{-2} \\ &= 43836.62109375 \, (= 0.4383662109375 \cdot 10^5) \end{aligned}$$

2. Nicht normierte Gleitpunktzahlen:
   - Die obigen Beispiele $x_1 - x_5$ sind alle gemäss Definition 2.1 normiert, d.h. die erste Ziffer vor dem Punkt ist Null, die erste Ziffer nach dem Punkt ist ungleich Null für $x \neq 0$.
   - Damit sind ihre Mantisse und der Exponent eindeutig definiert.
   - Die folgenden Beispiele geben zur Illustration nicht normierte Varianten für x_1 und x_2. Es ist offensichtlich, dass eine nicht normierte Darstellung bzgl. Mantisse und Exponent nicht mehr eindeutig ist (auch wenn der Wert immer gleich bleibt). Dies ist ein Zustand, der bei der Speicherung vermieden werden muss.

$$\tilde{x}_1 = -0.002345 \cdot 10^5 = -23.45 \cdot 10^2 = -234500 \cdot 10^{-3} = \ldots$$

$$\tilde{x}_2 = 0.0111 \cdot 2^4 = 1.11 \cdot 2^2 = 111 \cdot 2^0 = 11100 \cdot 2^{-2} = \ldots$$

3. IEC/IEEE - Gleitpunktzahlen mit $B = 2$
   - single format: Gesamtlänge der Zahl ist 32 Bit, wobei 1 Bit für das Vorzeichen, 23 Bit für die Mantisse, und 8 Bit für den Exponenten
     - double format: Gesamtlänge der Zahl ist 64 Bit, wobei 1 Bit für das Vorzeichen, 52 Bit für die Mantisse, und 11 Bit für den Exponenten

   Das Vorzeichenbit $v \in \{0,1\}$ erzeugt das Vorzeichen über den Faktor $(-1)^v$, d.h. $v = 0$ entspricht einem positiven, $v = 1$ einem negativen Vorzeichen. Ausserdem wird ein von der Anzahl Bits abhängiger Biaswert im Exponent subtrahiert, womit der Exponent kein eigenes Vorzeichenbit benötigt.

Maschinenzahlen
Beispiele 2.1: IEC / IEEE

`single`: (32 bit)

```
   V EEEEEEEE MMMMMMMMMMMMMMMMMMMMMMM
   0 1       8 9                     31
```

0	10000000	00000000000000000000000	$=$	$+1.0 * 2^{128-127} = 2$
0	10000001	10100000000000000000000	$=$	$+1.101 * 2^{129-127} = 6.5$
1	10000001	10100000000000000000000	$=$	$-1.101 * 2^{129-127} = -6.5$
0	00000001	00000000000000000000000	$=$	$+1.0 * 2^{1-127} = 2^{-126} = x_{\min}$
0	00000000	10000000000000000000000	$=$	$+0.1 * 2^{-126} = 2^{-127}$
0	00000000	00000000000000000000001	$=$	$+0.0\ldots01 * 2^{-126} = 2^{-149}$
			$=$	kleinste darstellbare Zahl

`double`: (64 bit)

```
V EEEEEEEEEEE MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
0 1          11 12                                                63
```

- Wieso rechnet man eigentlich mit Gleitpunktzahlen? Nimmt man, abgesehen vom Vorzeichen, an, dass 8 dezimale Speichereinheiten zur Verfügung stehen, so liessen sich damit in den folgenden Systemen die folgenden positiven Zahlen darstellen:
- Ganzzahlsystem:
  1. kleinste darstellbare positive Zahl:00000001
  2. grösste darstellbare positive Zahl: 99999999
     Es lassen sich also im positiven Bereich alle ganzen Zahlen zwischen 1 und 99999999 $(= 10^8 - 1)$ hinterlegen. Der Abstand zwischen aufeinanderfolgenden Zahlen ist konstant gleich 1.

Maschinenzahlen

Beispiele 2.1: Weshalb Gleitpunktzahlen

- Festpunktsystem mit 4 Dezimalen:
  1. kleinste darstellbare positive Zahl:0000.0001
  2. grösste darstellbare positive Zahl: 9999.9999

  Es lassen sich im positiven Bereich Zahlen zwischen 0.0001 und 9999.9999 $(= 10^4 - 10^{-4})$ darstellen. Der Abstand zwischen aufeinanderfolgenden Zahlen ist konstant gleich 10^{-4}.
- Normalisiertes Gleitpunktsystem mit 6 Mantissen- und 2 Exponenentenziffern (mit Bias -50)
  1. kleinste darstellbare positive Zahl:$0.100000 \cdot 10^{-50}$
  2. grösste darstellbare positive Zahl: $0.999999 \cdot 10^{49}$

  Es lassen sich im positiven Bereich Zahlen zwischen 10^{-51} und fast10^{49} darstellen. Der Abstand zwischen aufeinanderfolgenden Zahlen ist allerdings variabel, wie in Kap. 2.4 gezeigt.

Maschinenzahlen
Aufgaben 2.2

1. Wie viele Stellen benötigt man, um die folgenden Zahlen als n-stellige Gleitpunktzahlen im Dezimalsystem darzustellen? $x_1 = 0.00010001$, $x_2 = 1230001$, $x_3 = \frac{4}{5}$, $x_4 = \frac{1}{3}$
2. Bestimmen Sie alle dualen 3-stelligen positiven Gleitpunktzahlen mit einstelligem positiven binären Exponenten sowie ihren dezimalen Wert.
3. Wie viele verschiedene Maschinenzahlen gibt es auf einem Rechner, der 20-stellige Gleitpunktzahlen mit 4-stelligen binären Exponenten sowie dazugehörige Vorzeichen im Dualsystem verwendet? Wie lautet die kleinste positive und die größte Maschinenzahl?
4. Verstehen Sie den folgenden 'Witz'?
   Es gibt 10 Gruppen von Menschen: diejenigen, die das Binärsystem verstehen, und die anderen.

Umrechnung zwischen den Basen

- Für das gegenseitige Konvertieren von Zahlen mit unterschiedlichen Basen reicht es, als ein Bezugssystem das Dezimalsystem zu nehmen.
- Wir müssen also im Grunde zwei Richtungen betrachten:
  - die Umrechnung einer Gleitkommazahl mit Basis $B \neq 10$ ins Dezimalsystem und
  - die Umrechnung vom Dezimalsystem in eine beliebige andere Basis[2]

[2] Wir wollen an dieser Stelle nicht auf alle möglichen Spielarten eingehen, ausführlich erläutert werden die Umrechnungsarten z.B. auf der Webseite
http://www.arndt-bruenner.de/mathe/scripts/Zahlensysteme.htm#1

Umrechnung zwischen den Basen

Umrechnung von einer beliebigen Basis ins Dezimalsystem

- Die Umwandlung einer Gleitkommazahl mit Basis B in die zugehörige Dezimalzahl ist nichts anderes als die Berechnung des Wertes gemäss Definition 2.1, doch empfiehlt es sich, den ganzzahligen Anteil und den Dezimalteil (nach dem Dezimalpunkt) getrennt als eigenständige Polynome zu behandeln.

Umrechnung zwischen den Basen
Beispiel 2.2

- Die (unnormierte) Binärzahl $(x)_2 = 11001.1011$ soll ins Dezimalsystem umgerechnet werden. Gemäss Definition 2.1 gilt

$$\begin{aligned}(x)_{10} &= \sum_{i=1}^{n} m_i B^{e-i} \\ &= \underbrace{1\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0}_{\text{ganzzahliger Anteil}} \\ &\quad \underbrace{+1\cdot 2^{-1} + 0\cdot 2^{-2} + 1\cdot 2^{-3} + 1\cdot 2^{-4}}_{\text{Dezimalanteil}} \\ &= 25.6875\end{aligned}$$

- Die sequentielle Summation dieses langen Ausdrucks ist nicht sehr effizient.
- Das Horner-Schema erlaubt die Auswertung dieser beiden Polynome an den Stellen $x = 2$ bzw. $x = \frac{1}{2}$ sehr effizient.

Umrechnung zwischen den Basen
Beispiel 2.2

- Ganzzahliger Anteil: der Faktor 2 wird fortlaufend mit den Koeffizienten m_i $(i = 1, 2, ..., e)$ multipliziert und 'von links nach rechts' aufsummiert (mit Start beim innersten Klammerausdruck):

$$\underrightarrow{11001} = \underrightarrow{(((\mathbf{1}\cdot 2+\mathbf{1})\cdot 2+\mathbf{0})\cdot 2+\mathbf{0})\cdot 2+\mathbf{1}} = 25$$
$$(= 1\cdot 2^4+1\cdot 2^3+0\cdot 2^2+0\cdot 2^1+1)$$

oder analog mit dem Schema

	2^4	2^3	2^2	2^1	2^0
	1	1	0	0	1
$B = 2$	$\downarrow$	2	6	12	24
	1	3	6	12	**25**

Umrechnung zwischen den Basen
Beispiel 2.2

- Dezimalanteil: der Faktor 2^{-1} bzw. $\frac{1}{2}$ wird fortlaufend mit den Koeffizienten m_i $(i = e+1, e+2, \ldots, n)$ multipliziert und von 'rechts nach links' aufsummiert (mit Start beim innersten Klammerausdruck):

$$.\underleftarrow{1011} = \underleftarrow{\frac{1}{2}\left(\mathbf{1}+\left(\frac{1}{2}\right)\left(\mathbf{0}+\left(\frac{1}{2}\right)\left(\mathbf{1}+\mathbf{1}\cdot\left(\frac{1}{2}\right)\right)\right)\right)} = 0.6875$$

$$(= 1\cdot 2^{-1} + 0\cdot 2^{-2} + 1\cdot 2^{-3} + 1\cdot 2^{-4})$$

oder analog mit dem Schema (Erklärung folgt im Unterricht)

	$(\frac{1}{2})^4$	$(\frac{1}{2})^3$	$(\frac{1}{2})^2$	$(\frac{1}{2})^1$	$(\frac{1}{2})^0$
	1	1	0	1	0
$B = \frac{1}{2}$	$\downarrow$	0.5	0.75	0.375	0.6875
	1	1.5	0.75	1.375	**0.6875**

Umrechnung zwischen den Basen
Aufgabe 2.3

- Konvertieren Sie die folgenden Zahlen mit dem Horner-Schema ins Dezimalsystem:

1. $(110001110.00101)_2$
2. $(111110101.1101)_2$
3. $(122102.102)_3$
4. $(345.2114)_6$
5. $(AFDE.BB1C)_{16}$

Umrechnung zwischen den Basen

Umrechnung vom Deziamlsystem in andere Basen

- Die Umkehrung des Horner-Schemas erlaubt es, den ganzzahligen Anteil und den Dezimalanteil je für sich durch durch fortlaufendes Ausklammern der neuen Basis B (bzw. $\frac{1}{B}$) zu berechnen.
- Wir illustrieren dies beispielhaft anhand der Umrechnung der Zahl $(1006.687)_{10}$ ins Binär-, Octal- und Hexadezimalsystem.

Umrechnung zwischen den Basen
Vom Dezimalsystem ins Dualsystem

Die Zahl 1006.687 soll vom 10er-System ins 2er-System umgewandelt werden. Ganzzahliger Anteil und Dezimalteil werden jeweils getrennt behandelt.

- Zunächst die Umwandlung des ganzzahligen Teils. Gehen Sie nach folgendem Verfahren vor:
  1. Teilen Sie die Zahl durch 2 und notieren sich den Rest (0 oder 1).
  2. Nehmen Sie das Resultate der vorherigen Division und wiederholen den Vorgang bis die Division durch 2 Null ergibt.
  3. Die Ziffernfolge für den Rest ergibt (von "unten nach oben") die Binärdarstellung der Zahl.

Umrechnung zwischen den Basen
Vom Dezimalsystem ins Dualsystem

$$
\begin{aligned}
1006{:}2 &= 503 \quad \text{Rest: } 0\\
503{:}2 &= 251 \quad \text{Rest: } 1\\
251{:}2 &= 125 \quad \text{Rest: } 1\\
125{:}2 &= 62 \quad \text{Rest: } 1\\
62{:}2 &= 31 \quad \text{Rest: } 0\\
31 : 2 &= 15 \quad \text{Rest: } 1\\
15 : 2 &= 7 \quad \text{Rest: } 1\\
7 : 2 &= 3 \quad \text{Rest: } 1\\
3 : 2 &= 1 \quad \text{Rest: } 1\\
1 : 2 &= 0 \quad \text{Rest: } 1
\end{aligned}
$$

Das Resultat ist: 1111101110

Umrechnung zwischen den Basen
Vom Dezimalsystem ins Dualsystem

- Nun die Umwandlung der Nachkommastellen. Gehen Sie nach folgendem Verfahren vor:
  1. Multiplizieren Sie die Zahl mit der Basis 2 und notieren Sie sich die Zahl vor dem Komma
  2. Falls diese 1 wird, schneiden Sie sie weg bis der Rest 0 ist, der Rest sich wiederholt oder die gewünschte Genauigkeit erreicht ist.
  3. Die Ziffernfolge für den Rest ergibt (von "oben nach unten") die Binärdarstellung der Zahl.

Umrechnung zwischen den Basen

Vom Dezimalsystem ins Dualsystem

$$
\begin{array}{lll}
2 \cdot 0{,}687 = 1{,}374 & \text{Ziffer} & 1 \\
2 \cdot 0{,}374 = 0{,}748 & \text{Ziffer} & 0 \\
2 \cdot 0{,}748 = 1{,}496 & \text{Ziffer} & 1 \\
2 \cdot 0{,}496 = 0{,}992 & \text{Ziffer} & 0 \\
2 \cdot 0{,}992 = 1{,}984 & \text{Ziffer} & 1 \\
2 \cdot 0{,}984 = 1{,}968 & \text{Ziffer} & 1 \\
2 \cdot 0{,}968 = 1{,}936 & \text{Ziffer} & 1 \\
2 \cdot 0{,}936 = 1{,}872 & \text{Ziffer} & 1 \\
2 \cdot 0{,}872 = 1{,}744 & \text{Ziffer} & 1 \\
\vdots & \vdots & \vdots
\end{array}
$$

Das Resultat ist: 0.10101111...

Umrechnung zwischen den Basen

Vom Dezimalsystem ins Dualsystem

- Die beiden Teile zusammen ergeben also das Resultat

$$1006.687 = 1111101110.10101111...$$

in unnormierter Darstellung.

- Wollen wir noch normieren, z.B. auf die Mantisselänge $n = 13$, so erhalten wir

$$1006.687 \approx 0.1111101110101 \cdot 2^{10}$$

- Dabei haben wir von 1111101110.10101111... die ersten 13 Ziffern genommen und noch ein '0.' vorangestellt. Der Wert des Exponents ergibt sich aus der Anzahl Ziffern für den ganzzahligen Anteil (nämlich 10).

Umrechnung zwischen den Basen
Vom Dezimalsystem ins Dualsystem

- Bei begrenzter Mantisselänge lässt es sich nicht verhindern, dass wegen dem 'Abschneiden' der binären Zahl ein Fehler gemacht wird, denn der Wert der binären Zahl $0.1111101110101 \cdot 2^{10}$ ist

$$\begin{aligned} &1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 \\ &+1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \\ &= 1006.625 \end{aligned}$$

- Deshalb wird bei jedem Rechner jeweils ein Fehler auftreten, da die Mantisselänge immer begrenzt ist.
- Konkret ergibt sich in diesem Beispiel bei der Abbildung ins Dualsystem der absolute Fehler (definiert als Betrag der Differenz zwischen dem Näherungswert und dem exakten Wert):

$$|\, 1006.625 - 1006.687 \,| = 0.0620$$

Umrechnung zwischen den Basen
Vom Dezimalsystem ins Dualsystem

- Auf die verschiedenen Fehlerarten kommen wir in Kap. 2.4 noch zu sprechen.
- Dort werden wir zeigen, dass einfaches Abschneiden i.d.R. kein gutes Verfahren ist, um eine reelle Zahl auf die Menge der Maschinenzahlen abzubilden.

Umrechnung zwischen den Basen
Vom Dezimalsystem ins Oktalsystem

- Das Vorgehen ist analog zur Umrechnung ins Dualsystem, nur das als Divisor bzw. Multiplikator nun statt der Basis 2 die Basis 8 verwendet wird.
- Für den ganzzahligen Anteil erhalten wir:

$$\begin{aligned} 1006{:}8 &= 125 \quad \text{Rest: } 6 \\ 125{:}8 &= 15 \quad \text{Rest: } 5 \\ 15{:}8 &= 1 \quad \text{Rest: } 7 \\ 1{:}8 &= 0 \quad \text{Rest: } 1 \end{aligned}$$

- Also $(1006)_{10} = (1756)_8$. Hier haben wir die Basis zur Verdeutlichung als Index angegeben.

Umrechnung zwischen den Basen

Vom Dezimalsystem ins Oktalsystem

- Für den Nachkommateil haben wir dann

$$\begin{aligned}
8 \cdot 0{,}687 &= 5{,}496 \quad \text{Ziffer} \quad 5 \\
8 \cdot 0{,}496 &= 3{,}968 \quad \text{Ziffer} \quad 3 \\
8 \cdot 0{,}968 &= 7{,}744 \quad \text{Ziffer} \quad 7 \\
8 \cdot 0{,}744 &= 5{,}952 \quad \text{Ziffer} \quad 5 \\
8 \cdot 0{,}952 &= 7{,}616 \quad \text{Ziffer} \quad 7 \\
8 \cdot 0{,}616 &= 4{,}928 \quad \text{Ziffer} \quad 4 \\
8 \cdot 0{,}928 &= 7{,}424 \quad \text{Ziffer} \quad 7 \\
8 \cdot 0{,}424 &= 3{,}392 \quad \text{Ziffer} \quad 3 \\
8 \cdot 0{,}392 &= 3{,}136 \quad \text{Ziffer} \quad 3 \\
&\vdots \qquad\qquad \vdots \quad \vdots
\end{aligned}$$

- Zusammen erhalten wir die unnormierte Darstellung

$$(1006.687)_{10} = (1756.537574733\ldots)_8$$

Umrechnung zwischen den Basen
Vom Dezimalsystem ins Oktalsystem

- Für die Mantissenlänge $n = 13$ ergibt sich die normierte Darstellung

$$1006.687 \approx 0.1756537574733 \cdot 8^4$$

- Der Wert dieser Oktalzahl ist

$$\begin{aligned} &1 \cdot 8^3 + 7 \cdot 8^2 + 5 \cdot 8^1 + 6 \cdot 8^0 + 5 \cdot 8^{-1} + 3 \cdot 8^{-2} + 7 \cdot 8^{-3} \\ &+5 \cdot 8^{-4} + 7 \cdot 8^{-5} + 4 \cdot 8^{-6} + 7 \cdot 8^{-7} + 3 \cdot 8^{-8} + 3 \cdot 8^{-9} \\ &= 1006.686999998987... \end{aligned}$$

- Für den absoluten Fehler erhalten wir

$$|\,1006.686999998987... - 1006.687\,| \approx 1.0133 \cdot 10^{-9}$$

- Offensichtlicht ist der absolute Fehler hier deutlich kleiner als in der Binärdarstellung.

Umrechnung zwischen den Basen
Vom Dezimalsystem ins Hexadezimalsystem

- Als Divisor bzw. Multiplikator verwenden wir nun die Basis 16.
- Wir verwenden die Ziffern 0,1,...,9 sowie $A,...,F$ (wobei $A \triangleq 10$, $B \triangleq 11$, $C \triangleq 12$, $D \triangleq 13$, $E \triangleq 14$, $F \triangleq 15$ in Gross- oder Kleinbuchstaben). Für den ganzzahligen Anteil erhalten wir:

$$\begin{aligned} 1006{:}16 &= 62 \quad \text{Rest: } 14 \rightarrow \text{Ziffer: E} \\ 62{:}16 &= 3 \quad \text{Rest: } 14 \rightarrow \text{Ziffer: E} \\ 3{:}16 &= 0 \quad \text{Rest: } 3 \rightarrow \text{Ziffer: 3} \end{aligned}$$

- Also $(1006)_{10} = (3\text{EE})_{16}$.

Umrechnung zwischen den Basen
Vom Dezimalsystem ins Hexadezimalsystem

- Für den Nachkommateil haben wir dann

$$\begin{aligned}
16 \cdot 0{,}687 &= 10{,}992 \quad \text{Ziffer} \quad A \\
16 \cdot 0{,}992 &= 15{,}872 \quad \text{Ziffer} \quad F \\
16 \cdot 0{,}872 &= 13{,}952 \quad \text{Ziffer} \quad D \\
16 \cdot 0{,}952 &= 15{,}232 \quad \text{Ziffer} \quad F \\
16 \cdot 0{,}232 &= 3{,}712 \quad \text{Ziffer} \quad 3 \\
16 \cdot 0{,}712 &= 11{,}392 \quad \text{Ziffer} \quad B \\
16 \cdot 0{,}392 &= 6{,}272 \quad \text{Ziffer} \quad 6 \\
16 \cdot 0{,}272 &= 4{,}352 \quad \text{Ziffer} \quad 4 \\
16 \cdot 0{,}352 &= 5{,}632 \quad \text{Ziffer} \quad 5 \\
16 \cdot 0{,}632 &= 10{,}112 \quad \text{Ziffer} \quad A \\
&\vdots \qquad\qquad \vdots \quad \vdots
\end{aligned}$$

und zusammen erhalten wir die unnormierte Darstellung

$$(1006.687)_{10} = (3EE.AFDF3B645A\ldots)_{16}$$

Umrechnung zwischen den Basen
Vom Dezimalsystem ins Hexadezimalsystem

- Für die Mantisselänge $n = 13$ ergibt sich die normierte Darstellung

$$1006.687 \approx 0.3\text{EEAFDF3B645A} \cdot 16^3$$

- Der Wert dieser Hexadezimalzahl ist

$$\begin{aligned} &3 \cdot 16^2 + 14 \cdot 16^1 + 14 \cdot 16^0 + 10 \cdot 16^{-1} + 15 \cdot 16^{-2} + 13 \cdot 16^{-3} \\ &+15 \cdot 16^{-4} + 3 \cdot 16^{-5} + 11 \cdot 16^{-6} + 6 \cdot 16^{-7} + 4 \cdot 16^{-8} \\ &+5 \cdot 16^{-9} + 10 \cdot 16^{-10} = 1006.68699999999898... \end{aligned}$$

- Für den absoluten Fehler erhalten wir

$$|\,1006.68699999999898... - 1006.687\,| \approx 1.1369 \cdot 10^{-13}$$

Umrechnung zwischen den Basen
Aufgabe 2.4

1. Konvertieren Sie die Dezimalzahl 2678.317 in die Basis $B = 5$
2. Normieren Sie Ihr Resultat auf eine Mantissenlänge von $n = 12$ und passen Sie den Exponenten entsprechend an.
3. Was für einen absoluten Fehler machen Sie durch das Abschneiden bei dieser Normierung?
4. Schreiben Sie in MATLAB eine Funktion dec_to_bin, die eine beliebige Dezimalzahl inklusive Nachkommastellen (Input) in ihre Binärzahl (Output) mit wählbarer ganzzahliger Mantisselänge (Input) und genügend grossem Exponenten berechnet. -> Kommt in einer Übungsserie

- Die Maschinenzahlen sind nicht gleichmässig verteilt.
- Ein Beispiel für alle binären normalisierten Gleitpunktzahlen mit 4-stelliger Mantisse und 2-stelligem Exponenten ist auf der nächsten Seite dargestellt.
- Zwangsläufig gibt es bei jedem Rechner eine grösste (x_{max}) und kleinste positive Maschinenzahl (x_{min}).
- Dabei gilt für normalisierte Gleitpunktzahlen:

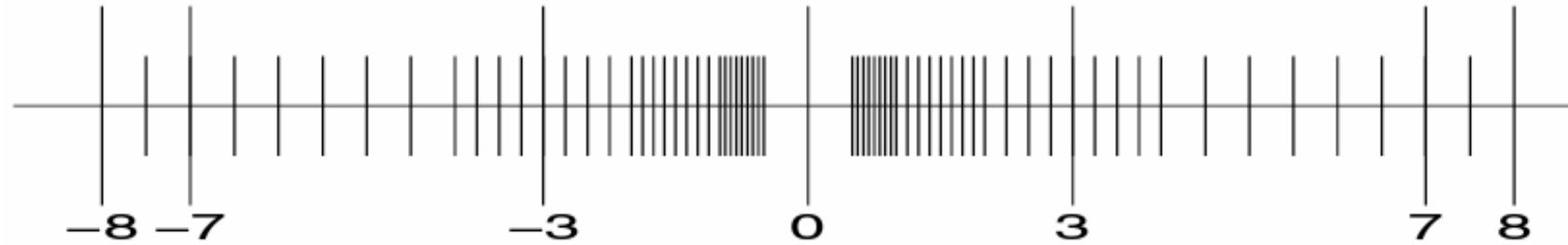
$$\begin{aligned} x_{max} &= B^{e_{max}} - B^{e_{max}-n} = (1 - B^{-n})B^{e_{max}} \\ x_{min} &= B^{e_{min}-1} \end{aligned}$$

- Wird auf die Normalisierung der Mantisse ($m_1 \neq 0$) verzichtet, führt dies zu sgnt. subnormalen Zahlen, die bis B^{m-n} hinunter reichen (IEEE Standard 754).

Approximations- und Rundungsfehler

Ungleichmässige Verteilung der Maschinenzahlen

- Alle binären Maschinenzahlen mit $n = 4$ und $0 \leq e \leq 3$ (Abbildung entnommen aus Knorrenschild)



Approximations- und Rundungsfehler
Aufgabe 2.5

- Schreiben Sie die kleinste und grösste binäre positive Maschinenzahl für die vorhergehende Abbildung ($n = 4$ und $0 \leq e \leq 3$) explizit auf und berechnen Sie deren Wert.
- Stimmt das mit x_{max} und x_{min} überein?

- Zahlen, die ausserhalb des Rechenbereichs $[-x_{max}, x_{max}]$ liegen, sind im *Überlaufbereich (overflow)* und führen zum Abbruch der Rechnung (mit IEEE 754 konforme Systeme geben die Bitsequenz *inf* aus).
- Zahlen ungleich 0, die innerhalb des Bereichs $[-x_{min}, x_{min}]$ liegen, führen zu einem *Unterlauf (underflow)*. Dann ist es sinnvoll, die Rechnung mit 0 weiterzuführen.
- Offensichtlich ist die Anzahl n der Mantissestellen von entscheidender Bedeutung für den Bereich der Zahlen, die abgebildet werden können. Dies wird eindrücklich illustriert in folgendem Beispiel:

Approximations- und Rundungsfehler
Beispiel 2.3

Am 4. Juni 1996 startete zum ersen Mal eine Ariane 5-Rakete der ESA von Französisch Guyana aus. Die unbemannte Rakete hatte vier Satelliten an Bord. 36.7 Sekunden nach dem Start wurde in einem Programm versucht, den gemessenen Wert der horizontalen Geschwindigkeit von 64 Bit Gleitpunktdarstellung in 16 Bit Ganzzahldarstellung (signed Integer) umzuwandeln. Da die entsprechende Masszahl grösser war als $2^{15} = 32768$, wurde ein Überlauf erzeugt. Das Lenksystem versagte daraufhin seine Arbeit und gab die Kontrolle an eine zweite, identische Einheit ab. Diese produzierte folgerichtig ebenfalls einen Überlauf. Da der Flug der Rakete instabil wurde und die Triebwerke abzubrechen drohten, zerstörte sich die Rakete selbst. Es entstand ein Schaden von ca. 500 Millionen Dollar durch den Verlust der Rakete und der Satelliten. Die benutzte Software stammte vom Vorgängermodell Ariane 4. Die Ariane 5 flog schneller und offensichtlich wurde dies bei der Repräsentation der Geschwindigkeit nicht beachtet.

Rundungsfehler und Maschinengenauigkeit

Definition 2.2: Absoluter / Relativer Fehler

- Jede reelle Zahl, die von einem Rechner verwendet werden soll, aber selber keine Maschinenzahl ist, muss also durch eine solche ersetzt werden.
- Dabei entstehen Fehler, wie wir gesehen haben.

Definition 2.2:

- Hat man eine Näherung $\widetilde{x}$ zu einem exakten Wert x, so ist der Betrag der Differenz $|\widetilde{x} - x|$ der **absolute Fehler**.
- Falls $x \neq 0$, so ist $|\frac{\widetilde{x}-x}{x}|$ bzw. $\frac{|\widetilde{x}-x|}{|x|}$ der **relative Fehler** dieser Näherung. In der Numerik ist der relative Fehler der wichtigere.

Rundungsfehler und Maschinengenauigkeit

- Natürlich sollte die Maschinenzahl dabei so gewählt werden, dass sie möglichst nahe bei der reellen Zahl liegt.
- Einfaches Abschneiden ist dazu nicht geeignet. Ein besseres Verfahren ist die Rundung.
- Beim Runden einer Zahl x wird eine Näherung unter den Maschinenzahlen gesucht, die einen minimalen absoluten Fehler $| rd(x) - x |$ aufweist.

Rundungsfehler und Maschinengenauigkeit

- Eine n-stellige dezimale Gleitpunktzahl $\widetilde{x} = 0.m_1 m_2 m_3 \ldots m_n \cdot 10^e = rd(x)$, die durch die Rundung eines exakten Wertes x entstanden ist, hat also einen absoluten Fehler von höchstens

$$| rd(x) - x | \leq 0.\underbrace{00\ldots005}_{n} \cdot 10^e = 0.5 \cdot 10^{e-n},$$

  wobei die 5 an der Stelle $n+1$ nach dem Dezimalpunkt auftritt.
- Für eine beliebige Basis gilt analog

$$| rd(x) - x | \leq 0.\underbrace{00\ldots00}_{n}\frac{B}{2} \cdot B^e = \frac{B}{2} \cdot B^{e-n-1},$$

- Für die Berechnungen bedeutet das, dass jede einzelne Operation (+, -, *, ...) auf $n+1$ genau gerechnet wird und das Ergebnis auf n Stellen gerundet wird (*n-stellige*

Rundungsfehler und Maschinengenauigkeit

Beispiel 2.4

- Sei $x = 180.1234567 = 0.1801234567 \cdot 10^3$. Gerundet auf eine siebenstellige Mantisse ($n = 7$) erhält man $rd(x) = 0.1801235 \cdot 10^3$ und es gilt wegen $e = 3$

$$| rd(x) - x | = 0.\underbrace{0000000}_{n=7}433 \cdot 10^3 = 0.433 \cdot 10^{-4} \leq 0.5 \cdot 10^{-4}$$

Rundungsfehler und Maschinengenauigkeit
Aufgabe 2.6

1. Vergewissern Sie sich anhand einfacher Zahlenbeispiele, dass die Rundung ein besseres Verfahren für die Abbildung einer rellen Zahl auf eine Maschinenzahl darstellt als einfaches Abschneiden der überzähligen Ziffern, wie in den früheren Beispielen in Kap. 2.3.2. Was ist der maximale Fehler, der durch das Abschneiden auftreten kann?
2. Wir kennen die Rundungsregeln für das Dezimalsystem. Verallgemeinern sie diese für eine beliebige Basis B. Runden Sie anschliessend die folgenden Zahlen auf eine vierstellige Mantisse, berechnen Sie den absoluten Fehler der Rundung und vergewissern Sie sich, dass $| rd(x) - x | \leq \frac{B}{2} \cdot B^{e-n-1}$:

   a) $(11.0100)_2$ b) $(11.0110)_2$ c) $(11.111)_2$

   d) $(120.212)_3$ e) $(120.212)_3$ f) $(0.FFFFF)_{16}$

Rundungsfehler und Maschinengenauigkeit

n-stellige Gleitpunktarithmethik

- Für die Berechnungen bedeutet Rundung, dass jede einzelne Operation (+, -, *, ...) auf n+1 genau gerechnet wird und das Ergebnis auf n Stellen gerundet wird (*n-stellige Gleitpunktarithmetik*).
- Jedes Zwischenergebnis wird also gerundet, nicht erst das Endergebnis.
- Das bedeutet auch, dass die einzelnen Rundungsfehler durch die Rechnung weitergetragen werden und allenfalls das Endergebnis verfälschen können.

Rundungsfehler und Maschinengenauigkeit

Definition 2.3: Maschinengenauigkeit

- Für den maximal auftretenden relativen Fehler bei der Rundung kann bei n-stelliger Gleitpunktarithmetik im Dezimalsystem also schreiben:

$$\left| \frac{rd(x) - x}{x} \right| \leq 5 \cdot 10^{-n} \, (da \, x \geq 10^{e-1}).$$

Definition 2.3:

- Die Zahl $eps := 5 \cdot 10^{-n}$ heisst **Maschinengenauigkeit**. Bei allgemeiner Basis B gilt $eps := \frac{B}{2} \cdot B^{-n}$.
- Alternative Definition: Die Maschinengenauigkeit ist die kleinste positive Maschinenzahl, für die auf dem Rechner $1 + eps \neq 1$ gilt.

- Bemerkung: der Begriff Maschinengenauigkeit impliziert nicht, dass der Rechner nicht mit deutlich kleineren Zahlen $x < eps$ noch 'genau' rechnen kann!

Rundungsfehler und Maschinengenauigkeit
Beispiel 2.5

Am Freitag, dem 25. November 1983, schloss der Aktienindex von Vancouver bei 524.811 Punkten und eröffnete am folgenden Montag, dem 28. November, bei 1098.892 Punkten. Was war passiert? Seit dem Start im Januar 1982 bei 1000 Punkten war der Aktienindex kontinuierlich gefallen, trotz florierendem Handel und guter Wirtschaftslage. Der Index wurde ca. 3000 mal am Tag neu berechnet, jeweils auf vier Dezimalstellen genau. Doch statt auf drei Dezimalstellen zu runden, wurde die vierte Dezimalstelle einfach abgeschnitten. Der dabei maximal mögliche Fehler von 0.0009 mutet zwar klein an, doch bei 3000 Wiederholungen pro Tag konnte sich dieser Abschneidefehler auf bis zu $0.0009 \cdot 3000 = 2.7$ Punkte pro Tag aufsummieren. Über die Zeitspanne von fast zwei Jahren verlor der Index so fast die Hälfte seines Wertes. Dies wurde am 28. November basierend auf korrekter Rundung korrigiert. Grössere Auswirkungen hatte diese Korrektur offenbar nicht, da zum damaligen Zeitpunkt das Volumen an Derivaten gering war.

Rundungsfehler und Maschinengenauigkeit
Aufgabe 2.7

1. Gesucht ist eine Näherung $\widetilde{x}$ zu $x = \sqrt{2} = 1.414213562...$ mit einem absoluten Fehler von höchstens 0.001.
2. Es soll $2590 + 4 + 4$ in 3-stelliger Gleitpunktarithmetik gerechnet werden (im Dezimalsystem), einmal von links nach rechts und einmal von rechts nach links. Wie unterscheiden sich die Resultate?

   Was lernen wir daraus? Beim Addieren sollte man die Summanden in der Reihenfolge aufsteigender Beträge sortieren
3. Berechnen Sie $s_{300} := \sum_{i=1}^{300} \frac{1}{i^2}$ sowohl auf- als auch absteigend, je einmal mit 3-stelliger und 5-stelliger Gleitpunktarithmetik (in MATLAB können Sie eine Zahl x auf 3 Stellen reduzieren z.B. mit dem Befehl string2num(num2string(x,3)))

Rundungsfehler und Maschinengenauigkeit
Aufgabe 2.7

- Es ist $\lim_{n\to\infty}(1+\frac{1}{n})^n = e$. Erstellen Sie eine Tabelle mit ihrem Rechner oder MATLAB für n = 1, 10, 100, ... für den Ausdruck $(1+\frac{1}{n})^n$sowie den absoluten und relativen Fehler. Erklären Sie Ihre Beobachtungen.

n	$(1+\frac{1}{n})^n$	absoluter Fehler	relativer Fehler
10^0			
10^2			
10^3			
10^4			
10^5			
10^6			
10^8			
10^9			
10^{10}			
10^{15}			

Fehlerfortpflanzung bei Funtionsauswertungen

- Wir haben gesehen, dass ein Rundungsfehler durch die Abbildung einer reellen Zahl x auf ihre Maschinenzahl $\widetilde{x}$ in die Berechnungen einfliesst.
- Soll nun eine Funktion f an der Stelle x ausgewertet werden, wird ein zusätzlicher Fehler dadurch generiert, dass nicht $f(x)$ sondern $f(\widetilde{x})$ berechnet wird.
- Für den fehlerbehafteten Wert $\widetilde{x}$ können wir den Fehler quantifizieren als $\Delta x = \widetilde{x} - x$ (vgl. Def. 2.2) oder

$$\widetilde{x} = x + \Delta x$$

- Nun wollen wir den absoluten Fehler $| f(\widetilde{x}) - f(x) |$ und den relativen Fehler $\frac{|f(\widetilde{x}) - f(x)|}{|f(x)|}$ dieser Funktionsauswertung berechnen.

Fehlerfortpflanzung bei Funtionsauswertungen

- Aus der allg. Taylor-Reihe (bekannt aus der Analysis) einer Funktion $f(x)$ um den Entwicklungspunkt x_0

$$f(x) = \sum_{i=0}^{\infty} \frac{f^{(i)}(x_0)}{i!}(x - x_0)^i$$

erhalten wir für die Entwicklung von $f(\widetilde{x})$ um den Entwicklungspunkt x

$$\begin{aligned} f(\widetilde{x}) &= f(x + \Delta x) = \sum_{i=0}^{\infty} \frac{f^{(i)}(x)}{i!}(\Delta x)^i \\ &= f(x) + f'(x)\Delta x + \frac{f''(x)}{2}(\Delta x)^2 + \ldots \end{aligned}$$

wobei wir in der Taylor-Reihe x durch $\tilde{x}$ und x_0 durch $x$ersetzt haben.

Fehlerfortpflanzung bei Funtionsauswertungen

- Unter der Annahme $\Delta x \ll 1$ können die höheren Fehlerterme $(\Delta x)^n$ für $n \geq 2$ vernachlässigt werden und es ergibt sich die folgende Näherung

$$\begin{aligned} f(\widetilde{x}) - f(x) &\approx f'(x)\Delta x \\ &\approx f'(x)(\tilde{x} - x) \end{aligned}$$

bzw. bei beidseitiger Division durch $f(x)$ und rechtseitiger Mulitplikation mit $\frac{x}{x}$:

$$\frac{f(\widetilde{x}) - f(x)}{f(x)} \approx \frac{f'(x) \cdot x}{f(x)} \cdot \frac{\tilde{x} - x}{x}$$

Fehlerfortpflanzung bei Funtionsauswertungen

Wir erhalten also die folgenden Näherungen:

- Näherung für den **absoluten Fehler bei Funktionsauswertungen**:

$$|f(\tilde{x}) - f(x)| \approx |f'(x)| \cdot |\tilde{x} - x|$$

- Näherung für den **relativen Fehler bei Funktionsauswertungen**:

$$\frac{|f(\tilde{x}) - f(x)|}{|f(x)|} \approx \frac{|f'(x)| \cdot |x|}{|f(x)|} \cdot \frac{|\tilde{x} - x|}{|x|}$$

Fehlerfortpflanzung bei Funtionsauswertungen

Definition 2.4: Konditionszahl

- Den Faktor

$$K := \frac{|\, f'(x) \,| \cdot |\, x \,|}{|\, f(x) \,|}$$

  nennt man **Konditionszahl**.

- Man unterscheidet **gut konditionierte Probleme**, d.h. die Konditionszahl ist klein, und **schlecht konditionierte Probleme** (ill posed problems) mit grosser Konditionszahl. Bei gut konditionierten Problemen wird der relative Fehler durch die Auswertung der Funktion nicht grösser.

Fehlerfortpflanzung bei Funtionsauswertungen

Bemerkungen:

- $\widetilde{x}$ kann generell als fehlerbehafteter Näherungswert für x angesehen werden. Ob der Fehler nun durch Rundung oder andere Effekte verursacht wird (z.B. durch fehlerhafte Messungen) ist hierbei nicht von Belang.
- Bei Funktionsauswertungen pflanzt sich der absolute Fehler in x näherungsweise mit dem Faktor $f'(x)$ fort. Falls $|f'(x)| > 1$ wird der absolute Fehler grösser, falls $|f'(x)| < 1$ kleiner.
- Bei Funktionsauswertungen pflanzt sich der relative Fehler in x näherungsweise mit der Konditionszahl fort.

Fehlerfortpflanzung bei Funtionsauswertungen
Beispiele 2.5

- Was lässt sich über die Fehlerfortpflanzung des absoluten Fehlers für die Funktion $f(x) = \sin(x)$ aussagen?
  Da$| f'(x) | = | \cos(x) | \leq 1$, folgt dass der absolute Fehler in den Funktionswerten nicht grösser sein kann als in den $x-$Werten sondern eher kleiner.
- Bei der Funktion $f(x) = 1000 \cdot x$ wird wegen $f'(x) = 1000$ der absolute Fehler in der Funktionsauswertung um den Faktor 1000 grösser.
- Die Konditionszahl für das Quadrieren, also $f(x) = x^2$, ist $K = \frac{|2x| \cdot |x|}{|x^2|} = 2$, d.h. der relative Fehler verdoppelt sich in etwa. Dies ist aber noch keine schlechte Konditionierung.

Fehlerfortpflanzung bei Funtionsauswertungen
Beispiele 2.5

- Das Polynom

$$P(x) = (x-1)^3 = x^3 - 3x^2 + 3x - 1$$

hat die dreifache Nullstelle $x_1 = x_2 = x_3 = 1$. Das nahe bei P liegende Polynom

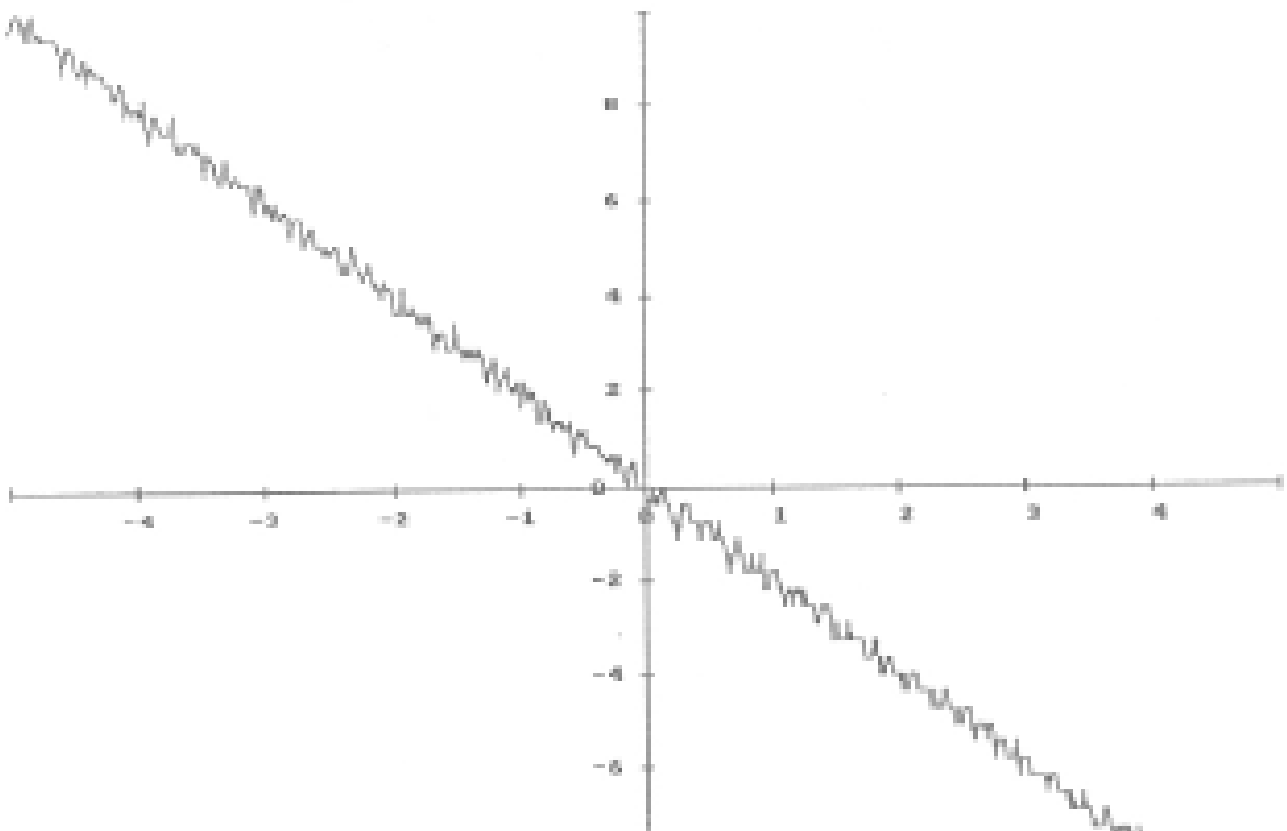
$$Q(x) = x^3 - 3.000001x^2 + 3x - 0.999999$$

hat die Nullstellen $x_1 = 1$, $x_2 \backsimeq 1.001414$, $x_3 \backsimeq 0.998586$.

- Während die Koeffizienten von P um 10^{-6} gestört wurden, haben sich die Nullstellen um 10^{-3}verändert, d.h. die Störung wurde um einen Faktor 1000 verstärkt.

Fehlerfortpflanzung bei Funtionsauswertungen
Beispiele 2.5

- Die Nullstellen von P sind schlecht konditioniert. Schaut man sich den Graphen von Q in der Umgebung der Nullstelle an, sieht man, wie die Rundungsfehler der Maschinenzahlen zu einer Zackenlinie führen.



Fehlerfortpflanzung bei Funtionsauswertungen

Fehlerfortpflanzung der Summation

- Betrachten wir nun die relativen Fortpflanzungsfehler für die grundlegenden arithmeitschen Operationen.
- Für

$$f(x) = x + c \ (c \in \mathbb{R})$$

haben wir für die Ableitung $f'(x) = 1$ und damit

$$\frac{|f(\tilde{x}) - f(x)|}{|f(x)|} \approx \frac{|x|}{|x+c|} \cdot \frac{|\tilde{x} - x|}{|x|}$$

bzw.

$$K = \frac{|x|}{|x+c|}$$

- Wenn x und die Konstante c gleiches Vorzeichen haben, gilt $K \leq 1$ dann haben wir also ein gut konditioniertes Problem.
- Was passiert aber, wenn x und c entgegengesetzte Vorzeichen haben und betragsmässig fast gleich gross sind? Dann wird $|x+c|$ sehr klein und somit K sehr gross, die Addition (bzw. Subtraktion) ist dann schlecht konditioniert.
- Dieses Phänomen nennt man auch ***Auslöschung***. Es tritt immer dann auf, wenn ungefähr gleich grosse fehlerbehaftete Zahlen voneinander abgezogen werden und das Resultat anschliessend normiert wird.

Fehlerfortpflanzung bei Funtionsauswertungen
Beispiel 2.6

- Für die beiden reellen Zahlen $r = \frac{3}{5}$ und $s = \frac{4}{7}$ mit den normierten gerundeten Räpresentationen mit fünfstelliger Mantisse, also $\tilde{r} = (0.10011)_2$ und $\tilde{s} = (0.10010)_2$, berechnen wir die Differenz $r - s = \frac{1}{35}$ näherungsweise als

$$0.10011 \cdot 2^0 - 0.10010 \cdot 2^0 = 0.00001 \cdot 2^0 = 0.10000 \cdot 2^{-4} = \frac{1}{32}.$$

Für den relativen Fehler erhalten wir

$$\frac{\frac{1}{32} - \frac{1}{35}}{\frac{1}{35}} = 0.0938 \approx 9.4\%$$

was viel ist (zum Vergleich, dies ist rund dreimal grösser als die Maschinengenauigkeit $2^{-5} = 0.0313$). Für die Berechnung mit dreistelliger Mantisse erhalten wir

$$0.101 - 0.101 = 0$$

und damit einen Fehler von 100%.

Fehlerfortpflanzung bei Funtionsauswertungen
Beispiel 2.7

- Gegeben seien die drei Werte

$$\begin{aligned} x_1 &= 123.454 \cdot 10^9 \\ x_2 &= 123.446 \cdot 10^9 \\ x_3 &= 123.435 \cdot 10^9 \end{aligned}$$

- Legt man eine 5-stellige dezimale Gleitpunktarithmetik zugrunde, so wird durch Rundung

$$\begin{aligned} \tilde{x}_1 &= 0.12345 \cdot 10^{12} \\ \tilde{x}_2 &= 0.12345 \cdot 10^{12} \\ \tilde{x}_3 &= 0.12344 \cdot 10^{12} \end{aligned}$$

Fehlerfortpflanzung bei Funtionsauswertungen

Beispiel 2.7

- Man erhält statt

$$\begin{aligned} x_1 - x_2 &= x_1 + (-x_2) = 8 \cdot 10^6 \\ x_1 - x_3 &= x_1 + (-x_3) = 19 \cdot 10^6 \end{aligned}$$

die fehlerhaften Werte

$$\begin{aligned} \tilde{x}_1 - \tilde{x}_2 &= 0 \\ \tilde{x}_1 - \tilde{x}_3 &= 10 \cdot 10^6 \end{aligned}$$

- Dies zeigt, dass die Subtraktion ein schlecht konditioniertes Problem darstellt, wenn x_1 und x_2 nahe beieinander liegen.

Fehlerfortpflanzung bei Funtionsauswertungen
Aufgabe 2.8

1. Untersuchen Sie, ob die Multiplikation und die Division zweier Zahlen gut oder schlecht konditionierte Funktionsauswertungen sind.