

Informatics V - Scientific Computing

Numerisches Programmieren

Tutorübung 1

Jürgen Bräckle, Nikola Tchipev

14.04. - 25.04.2014



Einführung in die Binärzahlen

Zahlendarstellung im Rechner

Rundungsfehler und ihre Folgen

Einführung in die Binärzahlen

Zahldarstellung im Rechner

Rundungsfehler und ihre Folgen

Zahldarstellung zur Basis B

- gewöhnlich mit $B \in \{2, 3, 10, 16\}$
- ganze Zahl x :

$$x = \sum_{i=0}^N r_i B^i$$

- mit Nachkommastellen:

$$x = \sum_{i=-\infty}^N r_i B^i$$


mit $r_i \in \{0, \dots, B-1\}$

Umrechnung von ganzen Zahlen

Umrechnung der Zahl 26 ins Binärsystem

- | Wert des Bits | 32 | 16 | 8 | 4 | 2 | 1 |
|---------------|----|----|---|---|---|---|
| Binärzahl | 0 | 1 | 1 | 0 | 1 | 0 |

- | | | | Rest |
|----|-------|----|------|
| 26 | : 2 = | 13 | 0 |
| 13 | : 2 = | 6 | 1 |
| 6 | : 2 = | 3 | 0 |
| 3 | : 2 = | 1 | 1 |
| 1 | : 2 = | 0 | 1 |



Umrechnung von Brüchen

am Beispiel $\frac{13}{4} = 111_3 : 11_3$ ins Trinärsystem.

Schriftliches Dividieren

$$\begin{array}{r} 111 : 11 = 10,\overline{02} \\ -11 \\ \hline 0100 \\ - 22 \\ \hline 001 \end{array}$$

Umrechnung von Brüchen

am Beispiel $\frac{13}{4} = 3 + \frac{1}{4} = 10_3 + \frac{1}{4}$ ins Trinärsystem.

Alternativer Weg

$$\begin{array}{cccc}
 & \cdot 3 & \cdot 3 & -2 \mid \cdot 3 \\
 \curvearrowright & & \curvearrowright & \curvearrowright \\
 \frac{1}{4} & \frac{3}{4} & \frac{9}{4} & \frac{3}{4} \\
 \hline
 0, & 0 & 2 & 0
 \end{array}$$

$$\Rightarrow \frac{13}{4} = 10, \overline{02}$$

- Man gibt immer den ganzzahligen Anteil an
- Dieser Anteil wird in jeder neuen Zeile abgezogen
- Jede neue Zeile wird mit der Basis (hier 3) multipliziert
- Bsp: $(\frac{9}{4} - 2) \cdot 3 = \frac{3}{4}$

Einführung in die Binärzahlen

Zahlendarstellung im Rechner

Rundungsfehler und ihre Folgen

2-Komplement für ganze Zahlen

- Darstellung von negativen Zahlen bei fester Bit-Anzahl n

Stelle des Bits	1	2	3	...	n
Wert des Bits	-2^{n-1}	2^{n-2}	2^{n-3}	...	2^0

- Beispiel: Darstellung der -5 mit $n = 4$

5 binär	0101
	↓
Invertiere	1010
	↓
Addiere 1	1011

Gleitkomma-Zahlen: 32-Bit IEEE-Standard

Ausgangspunkt:

- Festkommazahlen erlauben nur wenige darstellbare Zahlen
- Einführung einer normierten Exponentialdarstellung
(Beispiel: $-1.11001 \cdot 2^{-56}$)

32-Bit IEEE-Standard:



Gleitkomma-Zahlen: Korrektes Runden

Rundung der Zahl

$$x = 1, x_1 x_2 \dots x_{t-1} | x_t x_{t+1} \dots$$

Vergleich der Zahl nach der Rundungsstelle $|x_t x_{t+1} \dots$ mit der Wertigkeit der letzten verfügbaren Stelle $B^{-(t-1)}$:

- $|x_t x_{t+1} \dots > \frac{1}{2} \cdot B^{-(t-1)} \rightarrow$ Aufrunden (1.2|502)
- $|x_t x_{t+1} \dots < \frac{1}{2} \cdot B^{-(t-1)} \rightarrow$ Abrunden (1.2|317)
- $|x_t x_{t+1} \dots = \frac{1}{2} \cdot B^{-(t-1)} \rightarrow$ Uneindeutiger Fall (1.2|5)

Gleitkomma-Zahlen: Korrektes Runden

Analoges Vorgehen im IEEE-Standard:

- $x = 1,0|101 \rightarrow$ Aufrunden (1.1)
- $x = 1,0|011 \rightarrow$ Abrunden (1.0)
- Uneindeutiger Fall $|x_t x_{t+1} x_{t+2} \dots = 100 \dots$
 - $x = 1,1|100 \rightarrow$ Aufrunden (10.0)
 - $x = 1,0|100 \rightarrow$ Abrunden (1.0)

Fehler

- absoluter Fehler $f_{abs} := |x - \text{rd}(x)|$
- relativer Fehler $f_{rel} := \left| \frac{x - \text{rd}(x)}{x} \right|$

Fehler

- absoluter Fehler $f_{abs} := |x - \text{rd}(x)|$
- relativer Fehler $f_{rel} := \left| \frac{x - \text{rd}(x)}{x} \right|$

Maschinengenauigkeit ε_M

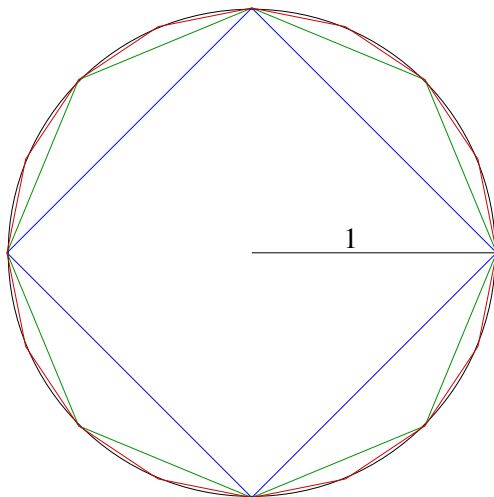
- bei Rundung mit t-stelliger Mantisse $\Rightarrow \varepsilon_M = 2^{-t}$
- größte positive Zahl ε_M mit $\text{rd}(1 + \varepsilon_M) = 1$

Einführung in die Binärzahlen

Zahldarstellung im Rechner

Rundungsfehler und ihre Folgen

Ermittlung von π nach Archimedes



Ermittlung von π nach Archimedes

