MOSIM WS 19/20

Dozent: Prof. Dr. Wolfgang V. Walter

23. März 2020

Inhaltsverzeichnis

Vorwort

Kapitel I

Introduction

14.10.2019

- Example:

  $$e^{-20} = 1 - 20 + \frac{400}{2} - \frac{8000}{6} + .. + -.. \frac{x^n}{n!}$$

  is problematic when you perform addition!

- Here comes the tableau
- IEEE
  - Standards: example: 754 (1985,2008,2018)
  - conference ARITH
  - Institute of Electrical and Electronics Engineers
  - Standard 1788 - for intervals/ interval arithmetic
- Scientific Computation (step by step)
  1. definition of problem
  2. <u>simplification</u>
  3. physical problem
  4. <u>model error</u>
  5. mathematical modeling
  6. <u>approximation error</u>
  7. mathematical approximation
  8. <u>rounding error</u>
  9. computation
  10. <u>error analysis</u>

  So we have 4 possible sources of errors!

- numerical differentiation (derivatives)

  $$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} \quad \text{for small } h, h > 0$$

  With smaller h we get better results, <u>but</u> do not make h to small. Then it is going to be even worse, because we get rounding - off error. **15.10.2019**

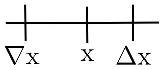
- Scientific software in trouble

mathematics/brain	**app/software**	**Hardware/Comp.**
real, complex numbers	floating point numbers	FPN
dense continuum	discrete grid	less fixed NFs
strong structure	weak algebraic structure	arithemtic structure
order relation (OR) $\leqslant$	OR $\leqslant$	directed rounding
OR $\subseteq$	OR $\subseteq$	directed rounding
intervals, sets	guaranteed inclusions	directed rounding
function	function value inclusions	directed rounding
math. notation	math. notation	machine code
math. objects	data abstraction, OOP	memory management
math. operations	operator overload	memory management
sequential methods	parallelization	multicore/ M-processor

  → Willian Kahan (“directed roundings”)

- Number formats (NFs)
  1. single 32 bit
  2. double 64 bit
  3. extended 80 bit
  4. quadruple 128 bit

- Rounding



Interval $X = (x, x), \diamond X = [\nabla x, \Delta x]$, but then we have errors in PC $(10^{20} + 13) - 10^{20} \to 10^{20} - 10^{20} \to 0$ instead of 13 (Here rounding - off error is missing)

Interval mathematics

→ Slogan: Interval-algorithm always possible close and mathematical correct enclosures of solutions.
If this is not possible, errors must be reported!

- control rounding errors
- enclosure approximation errors
- enclosure of derivatives and remainders
- guaranteed enclosures of the solution (sets)
- proof of existence of a solution in calculated interval
- If necessary, proof of uniqueness of the solution
- control of step size, order, accuracy of results
- sharp termination condition and reliable error detection
- no fantasy solutions or numerical artifacts

Tools:

- automatic differentation

  - LGS, NLGS
  - Fixed-point theorems (e.g. Brouwer)
  - integration ,..
- classic Newton method
  $x_0 =$ start value

$$x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})} \quad \text{with } x_0 = \text{ initial value}$$

- Interval Newton method

$$x_k = (| \; (x_{k-1}) - \frac{f(mid(x_{k-1})}{f'(x_{k-1})}) \cap x_{k-1} \quad \text{with } x_0 = \text{ initial interval}$$

  (has quadratic convergence)
  CISC → RISC → better pipelines (RISC is today in every processor!)

$$S_0 = 1 + 16^N - 16^N + 16^{N-1} \pm \cdots + 16^{-N} - 16^{-N} + 16^N - 16^{N-1} + 16^{N-1} - 16^{N-1}$$

  calculation with 4 pipelines we have here obliteration problems!
- Knuth - computer science pioneer, Turing, Wilkinson
- Requirements for programming systems
  - mathematical notation (simple program structure, clear, reusable, modulary, overloading, simple formulas)
  - Data abstraction (new data objects/ operations from existing operators combining
  - automatic memory management (dynamic objects variable size)
  - accuracy and inclusion

Kapitel II

Floating point numbers

28.10.2019

A floating point number is

$$x = \begin{cases} = (-1)^{S_x} \cdot m_x b^{e_x} = (S_x, m_x, e_x) \\ 0 \end{cases}$$

with sign bit (Vorzeichen-bit) $S_x \in \{0,1\}$, mantissa $m_x = 0.m_1 m_2..m_l$ and exponent $e_x \in \{\underline{e}, e + 1, \ldots, \overline{e}\}$ with $\underline{e} \approx -\overline{e}$. The mantissa digits are $m_i \in S - \{0, 1, .., b\}$ with $b := b - 1$ and mantissa length $l \in N^+$ (b = base, $\underline{e}$ = emin, $\overline{e}$ = emax and $\mathbb{N}^+ = \mathbb{N} \setminus \{0, 1\}$).

- Floating point format $R = R(b, l, \underline{e}, \overline{e}, \text{denorm})$ with

$$\text{denorm} = \begin{cases} \text{true} & \text{if denormalized FPN is allowed} \\ \text{false} \end{cases}$$

- normalized FPN $x \neq 1$: has as first mantissa digit $m_1 \neq 0$ ($b = 2 \Rightarrow m_1 = 1!$)
- unnormalized FPN $x \neq 0$ has $m_1 = 0$
- Normalization of a FPN $x \neq 0$: mantissa digits to push for k digits to the left (k = number of leading 0-digits) and subtracting k from e_x. $e_{x_n eu} = e_x - k$.
  Normalization is only possible if $e_{xneu} = e_x - k \geqslant e$
- denormalized FPN: $e_x = e$ and mantissa not normalized or can not be normalized
- number line is symmetrical to zero! Here its supposed to be the line.
- Wilkinson-epsilon

$$\varepsilon := b^{1-l} = \frac{1}{b^{l-1}}$$

  is the biggest relative gap between two neighbouring normalized FPN $\sim$ the biggest relative error in solving

- unit in last place ulp:

$$\text{ulp}(x) := \begin{cases} b^{e_x - l} & \text{if x is normal} \\ b^{e-l} & \text{if x is denormalized or 0} \end{cases}$$

  "unit in the last place" (at the mantissa of x.

$$x = 0.m_1 m_2 ... m_l \cdot b^{e_x} \text{ with } m_l = b^{e_x - l}$$

- successor ($k \in \mathbb{Z}$)

$$\text{succ}(x) := \begin{cases} x + \text{ulp}(x) & \text{if } x \neq -b^k \\ x + (\frac{\text{ulp}(x)}{b}) & \text{if } x = -b^k \end{cases}$$

- predecessor ($k \in \mathbb{Z}$)

$$\mathrm{pred}(x) := \begin{cases} x - \mathrm{ulp}(x) & \text{if } x \neq b^k \\ x - (\frac{\mathrm{ulp}(x)}{b}) & \text{if } x = b^k \end{cases}$$

x	$\mathrm{digits}_b(x)$
0	1
$b^0 = 1$	$\vdots$
$\vdots$	$\vdots$
$b - 1$	1
$b^1 = [10]_b$	2
$\vdots$	$\vdots$
$b^2 - 1$	2
$b^2 = [100]_b$	3

where we have base $b \in \mathbb{N} \setminus \{0, 1\}$ and $\mathrm{digits}_b(x) = \lfloor \log_b(x) \rfloor + 1$

Interval arithmetics (Wrap-around) in two-part complement

- exact result: $z^* = x \overset{\pm}{\underset{*}{}} y$, $x, y \in I_n$ (Integer with n bits inclusive sign)
- generated result:

$$z := ((z^* + z^{n-1}) \mod 2^n) - 2^{n-1} \in I_n$$

- for normal mantissa m_x it holds: $\frac{1}{b} \leqslant m_x < 1$
- for normal mantissa m_x it holds: $0 \leqslant m_x < \frac{1}{b}$
- 2 equivalent display possibilities for FPN $x \neq 0$:

$$x = (-1)^{S_x} \cdot m_x \cdot b^{e_x} = (-1)^{S_x} \cdot M_x \cdot b^{e_x - l}$$
$$M_x = m_x \cdot b^l \in \mathbb{N} \text{ with } b^{l-1} \leqslant M_x < b^l$$

  Optionally one could also add $M_x \Rightarrow |M_x| < b^l$, $M_x \in \mathbb{Z}$
  $\Rightarrow$ Every FPN x is an integer multiple of its ulp‘s.

- $\mathrm{ulp}(1) = \varepsilon$, since $1 = 0,10\ldots0 \cdot b^1$ and $\varepsilon = b^{1-l} = \mathrm{ulp}(1)$ $x \neq 0, x \in R \leftarrow$ grid:
  - $\mathrm{succ}(x) \leqslant x(1+\varepsilon) = x(1 + \mathrm{ulp}(1 == 1 \cdot \succ (1)$
  - $\mathrm{succ}(x) \geqslant x(1-\varepsilon) = x(1 - \mathrm{ulp}(1) = x \cdot \mathrm{pred}(1)$
- relative distance between 2 neighboring FPN:

Definition 0.1 (Rounding)
Rounding is a map $o : \mathbb{R} \to R$ FP-grid with the following properties

(i) $(R1)$ $ox = x, \forall x \in R$ (projection)

(ii) $(R2)$ $x, y \in R : x \leqslant y \Rightarrow ox \leqslant oy$ (monotonicity)

(iii) $(R3)$ $o(-x) = -ox, \forall x \in R$

▶ **Bemerkung 0.2**
An antisymmetric rounding i guess additionally satisfies also 0.1 (iii).

■ **Beispiel 0.3**
Established rounding possibilities are:

- $\square x$: ıto nearest floating point number →
  - error $\leqslant \frac{1}{2}$ ulp
  - (stochastically calculated: "round to even" = to next FPN with even mantissa, i.e. end digit = 0 (binary)
  - $\underline{\text{If}}$ the value which is to be rounded lays exactly in the middle between 2 FPN
  - satisfies 0.1 (iii)): $\square(-x) = -\square x$
- $\sqcup x$: truncation, rounding → 0
  - $|\sqcup x| \leqslant\leqslant |\sqcup x|$
  - maximal error < 1 ulp
  - satisfies 0.1 (iii)
- $\sqcap x$: augmentation(= the, in absolute value, biggest FPN),
  - $|x| \leqslant |\sqcap x|$
  - maximal error < 1 ulp
  - satisfies 0.1 (iii)
- Δx: upward rounding → $+\infty$
  - $x \leqslant \Delta x$
  - maximal error < 1 ulp
  - satisfies $\underline{\text{not}}$ 0.1 (iii)
- ∇x: downwards rounding → $-\infty$
  - $\nabla x \leqslant x$
  - maximum error < 1 ulp
  - satisfies $\underline{\text{not}}$ 0.1 (iii)
  - instead of 0.1 (iii) the antisymmetry holds:

$$\nabla(-x) = -\Delta x \Leftrightarrow \Delta(-x) = -\nabla x$$

Intervals

- $I\mathbb{R}\{X = [\underline{x}, \overline{x}] \mid \underline{x} \leqslant \overline{x} \text{ and } \underline{x}, \overline{x} \in \mathbb{R}\}$, i.e. set of the bounded, closed, real intervals
- Floating-point intervals: $IR = \{X = [\underline{x}, \overline{x}] \mid \underline{x} \leqslant \overline{x}; \underline{x}, \overline{x} \in R\}$ intervals with floating point boundaries

Definition 0.4
Interval rounding as map

$$\Diamond : \mathbb{R} \to R \text{ with } X = [\Delta x, \nabla x] \subseteq X = [\underline{x}, \overline{x}]$$

with properties

$R1$ $\Diamond X = X \quad \forall X \in \mathbb{R}$

$R2$ $X, Y \in R : X \subseteq Y \Rightarrow \Diamond X \leqslant \Diamond Y$ (Inclusion isotonic)

*R*3 $\diamond(-X) = -\diamond X$, hence $\Delta(-x) = -\nabla(x)$, $\nabla(-x) = -\Delta x \in R$

Then holds:

$$\begin{aligned} X + Y &= [\underline{x} + \underline{y}, \overline{x} + \overline{y}] \\ X - Y &= [\underline{x} - \overline{y}, \overline{x} - \underline{y}] \end{aligned} \quad \operatorname{diam}(X \pm Y) = \operatorname{diam} X + \operatorname{diam} Y$$

Anhang

Literaturverzeichnis

Index