SCAN 2008 El Paso

Staggered Correction Computations with Enhanced Accuracy and Extremely Wide Exponent Range

University of Wuppertal - Germany
Frithjof Blomquist

Summary

1. Classic staggered correction arithmetic
2. Disadvantages of this arithmetic
   - Restricted exponent range
   - Low accuracy near the underflow range
3. New extended staggered interval arithmetic
4. Numerical examples

There are two possibilities for implementing an interval arithmetic in higher precision:

1. **Multiprecision-arithmetic** with base $B = 2^{32}$ and integer-valued mantissas. $\leadsto$ high memory requirements and an enlarged runtime, caused by the simulation of the basic operations by software.
2. **Classic Interval staggered arithmetic**

$$\boldsymbol{x} := \sum_{i=1}^{p-1} x_i + [x_p, x_{p+1}] = [\mathtt{Inf}(\boldsymbol{x}), \mathtt{Sup}(\boldsymbol{x})],$$

$$\mathtt{Inf}(\boldsymbol{x}) := \sum_{i=1}^{p-1} x_i + x_p, \ \mathtt{Sup}(\boldsymbol{x}) := \sum_{i=1}^{p-1} x_i + x_{p+1};$$

x_i : IEEE double numbers; $p \geq 1$: precision.

The basic arithmetic operations are based on the data type `dotprecision` and the x_i are generated by reading out the accumulator. So the x_i are in general non-overlapping IEEE numbers.

$$\boldsymbol{x} := \sum_{i=1}^{p-1} x_i + [x_p, x_{p+1}] = [\mathtt{Inf}(\boldsymbol{x}), \mathtt{Sup}(\boldsymbol{x})],$$

$$\mathtt{Inf}(\boldsymbol{x}) := \sum_{i=1}^{p-1} x_i + x_p, \ \mathtt{Sup}(\boldsymbol{x}) := \sum_{i=1}^{p-1} x_i + x_{p+1};$$

x_i : IEEE double numbers; $p \geq 1$: precision.

$\mathtt{Inf}(\boldsymbol{x}) \sim 10^k$:

10^k	10^{k-16}	10^{k-32}	$\ldots$	$10^{k-16(p-1)}$
x_1	x_2	x_3		x_p

Precision: $(16 \cdot p)$ decimal digits;
In case of $x_p = x_{p+1}$ ($\boldsymbol{x}$ is a point interval) the maximum accuracy: $(16 \cdot p)$ decimal digits.
Thus, precision and accuracy seem to be unbounded.

IEEE-double number 10^k: $-308 \leq k \leq +308$.
$10^{k-16(p-1)} \geq 10^{-308} \iff p \leq 20 + k/16$.

Restriction of the exponent: $-308 \leq k \leq +308.$

Restriction of p: $1 \leq p \leq 20 + k/16 =: p_{max}$

Some examples for $p_{max} = p_{max}(k)$:

k	$1 \leq p \leq p_{max}$	p_{max} · 16 dec. digits
$+308$	$p_{max} = 39$	624
$+150$	$p_{max} = 29$	464
0	$p_{max} = 20$	320
-308	$p_{max} = 1$	16

Thus, using the classic staggered interval arithmetic intermediate results near the underflow range must be avoided!

The next example will demonstrate how to manage the described difficulties near the underflow range:

$\boldsymbol{x} \sim 10^{-150}$, $\boldsymbol{y} \sim 10^{-150}$,
Product: $\boldsymbol{x} \cdot \boldsymbol{y} \sim 10^{-300}$, (only 20 dec. digits)

$\widetilde{\boldsymbol{x}} := 10^{300} \cdot \boldsymbol{x} \sim 10^{150}$, $\widetilde{\boldsymbol{y}} := 10^{300} \cdot \boldsymbol{y} \sim 10^{150}$;
$\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}} \sim 10^{150} \rightsquigarrow$ max. accuracy: 464 decimal digits.
Product: $\widetilde{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{y}} \sim 10^{300}$, without overflow!

$\boldsymbol{x} = 10^{-300} \cdot \widetilde{\boldsymbol{x}}, \qquad \boldsymbol{y} = 10^{-300} \cdot \widetilde{\boldsymbol{y}}.$
$\boldsymbol{x} \cdot \boldsymbol{y} = (10^{-300} \cdot \widetilde{\boldsymbol{x}}) \cdot (10^{-300} \cdot \widetilde{\boldsymbol{y}}) = 10^{-600} \cdot (\widetilde{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{y}}).$

Now the scaled intervals $\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}}$ can be multiplied in high accuracy and the above factor 10^{-600} or better -600 should be stored in some way.

Extended Staggered Arithmetic in C-XSC

Actual implementation in C-XSC: `lx_interval`

$$\boldsymbol{X} = 2^{m} \cdot \boldsymbol{x}, \quad m : \textit{double}, \quad \boldsymbol{x} : \texttt{l\_interval};$$

$$-9007199254740991 \leq m \leq +9007199254740991;$$

Multiplication:

$$\begin{aligned} \boldsymbol{X} \cdot \boldsymbol{Y} &= (2^{m_x} \cdot \boldsymbol{x}) \cdot (2^{m_y} \cdot \boldsymbol{y}) \\ &= (2^{m_x - s_x}\{2^{s_x} \cdot \boldsymbol{x}\}) \cdot (2^{m_y - s_y}\{2^{s_y} \cdot \boldsymbol{y}\}) \end{aligned}$$

Condition to get maximum accuracy:

$$\boldsymbol{u} := 2^{s_x} \cdot \boldsymbol{x} \sim 10^{150}, \quad \boldsymbol{v} := 2^{s_y} \cdot \boldsymbol{y} \sim 10^{150},$$
$$\rightsquigarrow \boldsymbol{u} \cdot \boldsymbol{v} \sim 10^{300}, \text{ no overflow,}$$

and a maximum accuracy of about 620 decimal digits.

$$\boldsymbol{X} \cdot \boldsymbol{Y} = (2^{m_u} \cdot \boldsymbol{u}) \cdot (2^{m_v} \cdot \boldsymbol{v}) = 2^{m_u + m_v}\{\boldsymbol{u} \cdot \boldsymbol{v}\};$$

Division:

$$\begin{aligned} \boldsymbol{X}/\boldsymbol{Y} &= (2^{m_x}\cdot\boldsymbol{x})/(2^{m_y}\cdot\boldsymbol{y}) \\ &= (2^{m_x-s_x}\{2^{s_x}\cdot\boldsymbol{x}\})/(2^{m_y-s_y}\{2^{s_y}\cdot\boldsymbol{y}\}) \end{aligned}$$

Condition to get maximum accuracy:

$$\boldsymbol{u} := 2^{s_x}\cdot\boldsymbol{x} \sim 10^{300}, \quad \boldsymbol{v} := 2^{s_y}\cdot\boldsymbol{y} \sim 10^{150},$$
$$\rightsquigarrow \boldsymbol{u}/\boldsymbol{v} \sim 10^{150}, \text{ no overflow,}$$

and a maximum accuracy of about 470 decimal digits.

$$\boldsymbol{X}/\boldsymbol{Y} = (2^{m_u}\cdot\boldsymbol{u})/(2^{m_v}\cdot\boldsymbol{v}) = 2^{m_u-m_v}\{\boldsymbol{u}/\boldsymbol{v}\};$$

Addition:

$$\begin{aligned} \boldsymbol{X}+\boldsymbol{Y} &= (2^{m_x}\cdot\boldsymbol{x})+(2^{m_y}\cdot\boldsymbol{y}) \\ &= (2^{m_x-s_x}\{2^{s_x}\cdot\boldsymbol{x}\})+(2^{m_y-s_y}\{2^{s_y}\cdot\boldsymbol{y}\}) \end{aligned}$$

1. condition: $\boldsymbol{u} := 2^{s_x}\cdot\boldsymbol{x} \sim 10^{+300}$;
2. condition: $\boldsymbol{v} := 2^{s_y}\cdot\boldsymbol{y}$; $m_y - s_y \stackrel{!}{=} m_x - s_x =: n$;

$$\boldsymbol{X}+\boldsymbol{Y} = (2^n\cdot\boldsymbol{u})+(2^n\cdot\boldsymbol{v}) = 2^n\cdot(\boldsymbol{u}+\boldsymbol{v});$$

1. Example

Two complex numbers:

$$z = a + i \cdot b, \quad w = c + i \cdot d, \quad i := \sqrt{-1};$$

$$\Im(z/w) = \frac{b \cdot c - a \cdot d}{c^2 + d^2},$$

$$a = b = 3; \quad c = 10^{300}; \quad d = 10^{300} - 1;$$

$$\begin{aligned} \Im(z/w) &= \frac{3}{2 \cdot 10^{600} - 2 \cdot 10^{300} + 1} \\ &\approx 1.5 \cdot 10^{-600}; \end{aligned}$$ 476 dec. digits.

2. Example

$$z = a + i \cdot b, \quad w = c + i \cdot d, \quad i := \sqrt{-1};$$

$$\Im(z/w) = \frac{b \cdot c - a \cdot d}{c^2 + d^2},$$

$$a = 2 \cdot 10^{5000}; \quad b = 2 \cdot a; \quad c = d = 10^{-5000};$$

$$\begin{aligned} \Im(z/w) &= \frac{2}{2 \cdot 10^{-10000}} \\ &= 1.0 \cdot 10^{+10000}; \end{aligned}$$ 474 decimal digits.

Elementary Interval Functions in C-XSC

$\|\boldsymbol{x}\|$	$\log_{10}(\boldsymbol{x})$
$\boldsymbol{x}^2$	$\sin(\boldsymbol{x})$
$\sqrt{\boldsymbol{x}}$	$\cos(\boldsymbol{x})$
$\sqrt[n]{\boldsymbol{x}}$	$\tan(\boldsymbol{x})$
$\sqrt{1+\boldsymbol{x}}-1$	$\cot(\boldsymbol{x})$
$\sqrt{1+\boldsymbol{x}^2}$	$\arcsin(\boldsymbol{x})$
$\sqrt{1-\boldsymbol{x}^2}$	$\arccos(\boldsymbol{x})$
$\sqrt{\boldsymbol{x}^2-1}$	$\arctan(\boldsymbol{x})$
$\boldsymbol{x}^n, p \in \mathbb{Z}$	$\operatorname{arccot}(\boldsymbol{x})$
$\boldsymbol{x}^p, p:$ `lx_interval`	$\sinh(\boldsymbol{x})$
$e^{\boldsymbol{x}}$	$\cosh(\boldsymbol{x})$
$e^{\boldsymbol{x}}-1$	$\tanh(\boldsymbol{x})$
$2^{\boldsymbol{x}}$	$\coth(\boldsymbol{x})$
$10^{\boldsymbol{x}}$	$\operatorname{arsinh}(\boldsymbol{x})$
$\ln(\boldsymbol{x})$	$\operatorname{arcosh}(\boldsymbol{x})$
$\ln(1+\boldsymbol{x})$	$\operatorname{artanh}(\boldsymbol{x})$
$\log_2(\boldsymbol{x})$	$\operatorname{arcoth}(\boldsymbol{x})$

Example Exponential function:

IEEE: $x = 709.5 \quad \mapsto \quad e^x = 1.35\ldots \cdot 10^{+308}$.

Inclusion with an **extended** staggered interval of type `lx_interval`

$$x = \overbrace{6243314768166065}^{\text{16 decimal digits}}$$

$$e^x \in 10^{\overbrace{+271437152599601}^{\text{16 decimal digits}}} \cdot \underbrace{9.90\ldots663}_{\text{451 dec. digits}} \begin{matrix} 83\ldots \\ 05\ldots \end{matrix}$$

Mathematica:

$$\texttt{Exp}[\overbrace{1488521748}^{\text{10 dec. digits}}] \quad \rightsquigarrow \quad \text{Overflow message!}$$

Next application **Taylor Arithmetic**

In C-XSC a complete Taylor arithmetic, based on the class `lx_interval`, is implemented.

$$f(x) = x^4 \cdot \sin(x^4);$$

Task:

Find a guaranteed inclusion of the fiftieth derivative for $x = 10^5$,

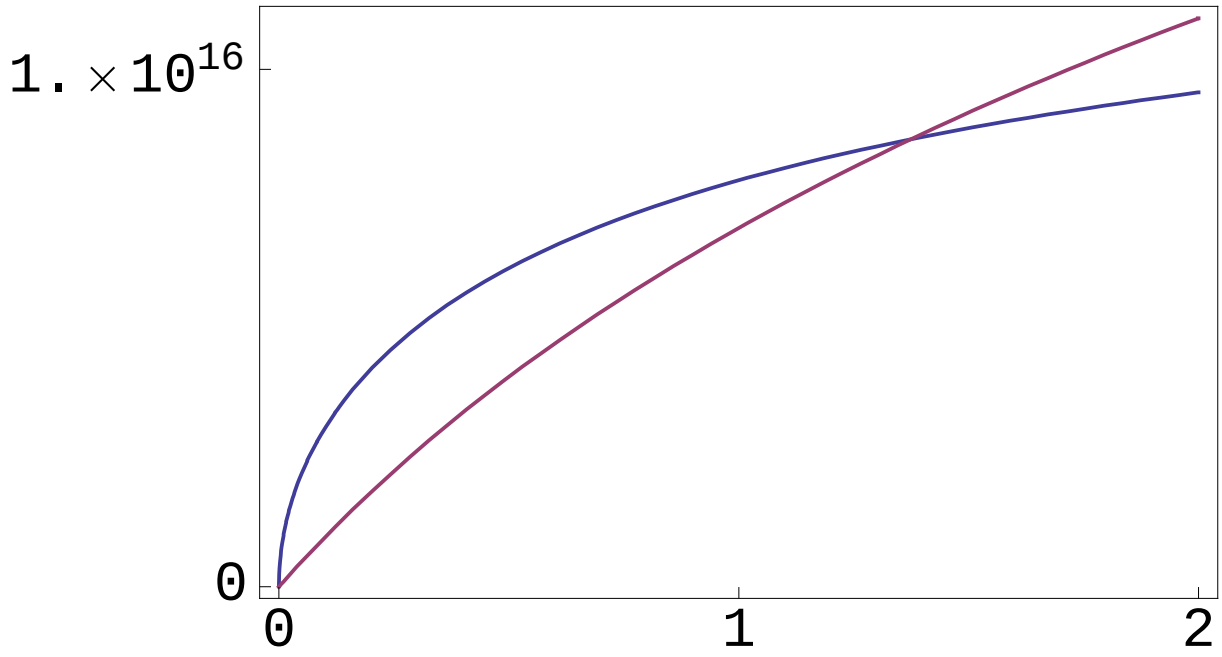
$$f^{(50)}(10^5) \in ?$$

$$f^{(50)}(10^5) \in 10^{799} \cdot \underbrace{8.179531\ldots70843}_{461 \text{ dec. digits}} \begin{matrix} 44\ldots \\ 20\ldots \end{matrix}$$

The inclusion interval lies deep in the IEEE overflow range and so the inclusion can only be calculated with the **extended** interval staggered arithmetic!

Next application: **Inclusion of (all) zeros**

$$g(x) := 10^{16} \cdot \arctan(\sqrt{x}) - 10^{16} \cdot \ln(1+x);$$



$$f(x) := 10^{16} \cdot \arctan(\sqrt{x}) - 10^{0} \cdot \ln(1+x).$$

$$f(x_0) = 0, \qquad x_0 \in 10^{44707} \cdot \underbrace{7.66\ldots564}_{\text{471 dec. digits}} \begin{matrix} 96\ldots \\ 80\ldots \end{matrix}$$

Complex-valued Interval Functions in C-XSC

$z = x + i \cdot y, \quad z$: lx_cinterval	
$\lvert z \rvert$	arctan(z)
z^2	arccot(z)
$\sqrt{z}$	sinh(z)
$\sqrt[n]{z}$	cosh(z)
arg(z)	tanh(z)
ln(z)	coth(z)
$z^n, n \in \mathbb{Z}$	arsinh(z)
z^p, p : lx_interval	arcosh(z)
z^w, w : lx_cinterval	artanh(z)
e^z	arcoth(z)
sin(z)	$\sqrt{1+z} - 1$
cos(z)	$\sqrt{1+z^2}$
tan(z)	$\sqrt{1-z^2}$
cot(z)	$\sqrt{z^2-1}$
arcsin(z)	$e^z - 1$
arccos(z)	ln($1+z$)