Arithmetic operations for floating-point intervals

Author: Ulrich Kulisch, Proposer: Bo Einarsson,
Seconder: John Pryce.

The author owes thanks to Gerd Bohlender, Bo Einarsson, Arnold Neumeier, John Pryce, Juergen Wolff von Gudenberg, and others.

Denotations

$\mathbb{R}$: the set of real numbers, $\mathbb{R}^* := \mathbb{R} \cup \{-\infty, +\infty\}$,
$\mathbb{F}$: the set of floating-point numbers of a given format and encoding,
$\mathbb{IR}$: the set of nonempty, closed, and bounded real intervals,
$\mathbb{IF}$: the intervals of $\mathbb{IR}$ whose bounds are in $\mathbb{F}$,
$\overline{\mathbb{IR}}$: the set of closed real intervals, $\varnothing \in \overline{\mathbb{IR}}$,[1]
$\overline{\mathbb{IF}}$: the intervals of $\overline{\mathbb{IR}}$ whose bounds are in $\mathbb{F}$, $\varnothing \in \overline{\mathbb{IF}}$,
$\triangledown$, $\triangle$: the roundings downwards and upwards,
$\overset{\triangledown}{+}$, $\overset{\triangledown}{-}$, $\overset{\triangledown}{*}$, $\overset{\triangledown}{/}$: the operations for elements of $\mathbb{F}$ with rounding downwards,
$\overset{\triangle}{+}$, $\overset{\triangle}{-}$, $\overset{\triangle}{*}$, $\overset{\triangle}{/}$: the operations for elements of $\mathbb{F}$ with rounding upwards.

Interval arithmetic over the real numbers $\mathbb{R}$ deals with closed and connected sets of real numbers. An interval is denoted by an ordered pair $[a, b]$. The first element is the lower bound and the second is the upper bound. The lower bound shall not be greater than the upper bound. If a bound is $-\infty$ or $+\infty$ the bound is not an element of the interval. Such intervals may also be written as $(-\infty, a], [b, +\infty)$ or $(-\infty, +\infty)$ with $a, b \in \mathbb{R}$. They are also closed[2] intervals.

For intervals $\boldsymbol{a}, \boldsymbol{b} \in \overline{\mathbb{IR}}$ arithmetic operations are defined as set operations in $\mathbb{R}$ by:

(R) $\boldsymbol{a} \circ \boldsymbol{b} := \{a \circ b \mid a \in \boldsymbol{a} \wedge b \in \boldsymbol{b} \wedge \boldsymbol{a} \circ \boldsymbol{b}$ is defined$\}$, for all $\boldsymbol{a}, \boldsymbol{b} \in \overline{\mathbb{IR}}$ and $\circ \in \{+, -, *, /\}$.

If $0 \notin \boldsymbol{b}$ in case of division then for all $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{IR}$ also $\boldsymbol{a} \circ \boldsymbol{b} \in \mathbb{IR}$.

On the computer a real number or an interval over the real numbers is mapped onto the smallest floating-point interval that contains the number or interval respectively. This mapping $\diamondsuit : \overline{\mathbb{IR}} \to \overline{\mathbb{IF}}$ is characterized by the following properties:

(R1) $\diamondsuit\, \boldsymbol{a} = \boldsymbol{a}$, for all $\boldsymbol{a} \in \overline{\mathbb{IF}}$,
(R2) $\boldsymbol{a} \subseteq \boldsymbol{b} \Rightarrow \diamondsuit\, \boldsymbol{a} \subseteq \diamondsuit\, \boldsymbol{b}$, for all $\boldsymbol{a}, \boldsymbol{b} \in \overline{\mathbb{IR}}$,
(R3) $\boldsymbol{a} \subseteq \diamondsuit\, \boldsymbol{a}$, for all $\boldsymbol{a} \in \overline{\mathbb{IR}}$,
(R4) $\diamondsuit\, (-\boldsymbol{a}) = -\diamondsuit\, \boldsymbol{a}$, for all $\boldsymbol{a} \in \overline{\mathbb{IR}}$.

With the mapping $\diamondsuit : \overline{\mathbb{IR}} \to \overline{\mathbb{IF}}$ binary arithmetic operations in $\mathbb{IF}$ are uniquely defined by:

(RG) $\boldsymbol{a} \diamondsuit \boldsymbol{b} := \diamondsuit\, (\boldsymbol{a} \circ \boldsymbol{b})$, for all $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{IF}$ and all $\circ \in \{+, -, *, /\}$.

Here for division we assume again that $0 \notin \boldsymbol{b}$.

For intervals $\boldsymbol{a} = [a_1, a_2]$, $\boldsymbol{b} = [b_1, b_2] \in \mathbb{IF}$ these operations $\diamondsuit$, $\circ \in \{+, -, *, /\}$, in $\mathbb{IF}$ have the property

$$\boldsymbol{a} \diamondsuit \boldsymbol{b} = \left[\min_{i,j=1,2} (a_i \overset{\triangledown}{\circ} b_j), \max_{i,j=1,2} (a_i \overset{\triangle}{\circ} b_j) \right],$$

or with the monotone roundings $\triangledown$ and $\triangle$

$$\boldsymbol{a} \diamondsuit \boldsymbol{b} = \left[\triangledown \min_{i,j=1,2} (a_i \circ b_j),\ \triangle \max_{i,j=1,2} (a_i \circ b_j) \right].$$

[1] $\overline{\mathbb{IR}}$ is the set of bounded and unbounded real intervals. $\{\overline{\mathbb{IR}}, \subseteq\}$ is a complete lattice, i.e., every subset has an infimum and a supremem. $\varnothing$ is the least and $\mathbb{R} = (-\infty, +\infty)$ is the greatest element.

[2] A subset of $\mathbb{R}$ is called closed if its complement is open.

The unary operator $-\boldsymbol{a}$ is defined by $-\boldsymbol{a} := (-1) \diamondsuit \boldsymbol{a}$.

The definition of the arithmetic operations in $\mathbb{IF}$ given on page 1 and the above formulas may be not well suited for implemention of the arithmetic on the computer. The unary operation $-\boldsymbol{a}$ and the binary operations addition, subtraction, multiplication, and division can be expressed by more explicit formulas as shown in the following tables. There the operator symbols for intervals are simply denoted by $+, -, *$, and $/$. These formulas are much more suited for implementation on the computer.

Minus operator $\quad -\boldsymbol{a} = [-a_2, -a_1]$.

Addition $\quad [a_1, a_2] + [b_1, b_2] = [a_1 \overset{*}{\nabla} b_1, a_2 \overset{*}{\Delta} b_2]$.

Subtraction $\quad [a_1, a_2] - [b_1, b_2] = [a_1 \nabla b_2, a_2 \Delta b_1]$.

Multiplication $[a_1, a_2] * [b_1, b_2]$	$[b_1, b_2]$ $b_2 \leq 0$	$[b_1, b_2]$ $b_1 < 0 < b_2$	$[b_1, b_2]$ $b_1 \geq 0$
$[a_1, a_2], a_2 \leq 0$	$[a_2 \overset{*}{\nabla} b_2, a_1 \overset{*}{\Delta} b_1]$	$[a_1 \overset{*}{\nabla} b_2, a_1 \overset{*}{\Delta} b_1]$	$[a_1 \overset{*}{\nabla} b_2, a_2 \overset{*}{\Delta} b_1]$
$a_1 < 0 < a_2$	$[a_2 \overset{*}{\nabla} b_1, a_1 \overset{*}{\Delta} b_1]$	$[min(a_1 \overset{*}{\nabla} b_2, a_2 \overset{*}{\nabla} b_1),$ $max(a_1 \overset{*}{\Delta} b_1, a_2 \overset{*}{\Delta} b_2)]$	$[a_1 \overset{*}{\nabla} b_2, a_2 \overset{*}{\Delta} b_2]$
$[a_1, a_2], a_1 \geq 0$	$[a_2 \overset{*}{\nabla} b_1, a_1 \overset{*}{\Delta} b_2]$	$[a_2 \overset{*}{\nabla} b_1, a_2 \overset{*}{\Delta} b_2]$	$[a_1 \overset{*}{\nabla} b_1, a_2 \overset{*}{\Delta} b_2]$

Division, $0 \notin \boldsymbol{b}$ $[a_1, a_2]/[b_1, b_2]$	$[b_1, b_2]$ $b_2 < 0$	$[b_1, b_2]$ $b_1 > 0$
$[a_1, a_2], a_2 \leq 0$	$[a_2 \overset{/}{\nabla} b_1, a_1 \overset{/}{\Delta} b_2]$	$[a_1 \overset{/}{\nabla} b_1, a_2 \overset{/}{\Delta} b_2]$
$[a_1, a_2], a_1 < 0 < a_2$	$[a_2 \overset{/}{\nabla} b_2, a_1 \overset{/}{\Delta} b_2]$	$[a_1 \overset{/}{\nabla} b_1, a_2 \overset{/}{\Delta} b_1]$
$[a_1, a_2], 0 \leq a_1$	$[a_2 \overset{/}{\nabla} b_2, a_1 \overset{/}{\Delta} b_1]$	$[a_1 \overset{/}{\nabla} b_2, a_2 \overset{/}{\Delta} b_1]$

In real analysis division by zero is not defined. In interval arithmetic, however, the interval in the denominator of a quotient may contain zero. So this case has to be considered also.

An important application is the extended interval Newton method. With it Newton's method reaches its ultimate elegance and strength. It computes all (single) zeros in a given domain. If a function has several zeros in a given interval its derivative becomes zero in that interval also. Thus division by an interval that contains zero is required.

In interval arithmetic the result of an operation is a set. Since in real analysis division by zero is not defined, the result of division by the interval $\boldsymbol{b} = [0, 0]$ can only be the empty set $\varnothing$. This means, the element 0 in the denominator of an interval division does not contribute to the solution set. So it can be excluded without changing the solution set.

So the general rule for computing the set $\boldsymbol{a}/\boldsymbol{b}$ with $0 \in \boldsymbol{b}$ is to remove its zero from the interval $\boldsymbol{b}$ and perform the division with the remaining set.[3] Whenever the zero in $\boldsymbol{b}$ coincides with a bound of the interval $\boldsymbol{b}$ the result of the division can directly be obtained from the above table for division with $0 \notin \boldsymbol{b}$ by the limit process $b_1 \to 0$ or $b_2 \to 0$ respectively. The results are shown in the following table. Here, the parentheses stress that the bounds $-\infty$ and $+\infty$ are not elements of the interval.

Whenever zero is an interior point of the denominator the following consideration leads to the correct answer.

[3]This is in full accordance with section 3.5.4 of the motion 6 position paper: When evaluating a function over a set, points outside its domain are simply ignored. See also [1], [2].

Division, $0 \in \boldsymbol{b}$	$\boldsymbol{b} =$	$[b_1, b_2]$	$[b_1, b_2]$
$[a_1, a_2]/[b_1, b_2]$	$[0, 0]$	$b_1 < b_2 = 0$	$0 = b_1 < b_2$
$[a_1, a_2] = [0, 0]$	$\varnothing$	$[0, 0]$	$[0, 0]$
$[a_1, a_2], a_1 < 0, a_2 \leq 0$	$\varnothing$	$[a_2 \mathbin{\overset{\triangledown}{\triangledown}} b_1, +\infty)$	$(-\infty, a_2 \mathbin{\overset{\triangle}{\triangle}} b_2]$
$[a_1, a_2], a_1 < 0 < a_2$	$\varnothing$	$(-\infty, +\infty)$	$(-\infty, +\infty)$
$[a_1, a_2], 0 \leq a_1, 0 < a_2$	$\varnothing$	$(-\infty, a_1 \mathbin{\overset{\triangle}{\triangle}} b_1]$	$[a_1 \mathbin{\overset{\triangledown}{\triangledown}} b_2, +\infty)$

A basic concept of mathematics is that of a function or mapping. A function consists of a pair (f, D_f). It maps each element x of its domain of definition D_f on a single element y of the range R_f of f, $f : D_f \to R_f$. A rational function $y = f(x)$ where the denominator is zero for $x = c$ is not defined for $x = c$; i.e., c is not an element of the domain of definition D_f. Since the function $f(x)$ is not defined at $x = c$ it does not have any value or property there. In this strict mathematical sense, division by an interval $[b_1, b_2]$ with $b_1 < 0 < b_2$ is not well posed. The interval $[b_1, b_2]$ overflows the range of definition of the function $f(x)$. For division the set $b_1 < 0 < b_2$ devolves into the two distinct sets $[b_1, 0)$ [4] and $(0, b_2]$ and division by the set $b_1 < 0 < b_2$ actually means two divisions. The results of the two divisions are already shown in the table for divison with $0 \in \boldsymbol{b}$. It is highly desirable to perform the two divisions consecutively.

In the user's program, however, the two divisions appear as a single operation, as division by an interval $[b_1, b_2]$ with $b_1 < 0 < b_2$. So an arithmetic operation in the user's program delivers two distinct results. This is an unusual phenomenon in digital computing,[5] but it can be handled.

A solution to the problem would be for the computer to provide a flag for *distinct intervals*. The situation occurs if the divisor is an interval that contains zero as an interior point. In this case the flag would be raised and signaled to the user. The user may then apply a routine of his choice to deal with the situation as is appropriate for his application.

This routine could be: return the entire set of real numbers $(-\infty, +\infty)$ as result and continue the computation, or continue the computation with one of the sets and ignore the other one, or put one of the sets on a list and continue the computation with the other one, or modify the operands and recompute, or stop computing, or some other action.

An alternative would be to provide a second division which in case of division by an interval that contains zero as an interior point generally delivers the result $(-\infty, +\infty)$. Then the user can decide when to use which division in his program.

Thus only four kinds of result come from division by an interval of $\mathbb{IF}$ that contains zero:

$$\varnothing, \quad (-\infty, a], \quad [b, +\infty), \quad \text{and} \quad (-\infty, +\infty).$$

We call such elements extended intervals.

The union of the set of closed and bounded intervals of $\mathbb{IR}$ with the set of extended real intervals is denoted by $\overline{\mathbb{IR}}$. Intervals of $\overline{\mathbb{IR}}$ and of $\overline{\mathbb{IF}}$ are sets of real numbers. $-\infty$ and $+\infty$ are not elements of these intervals. Arithmetic operations for extended intervals of $\overline{\mathbb{IF}}$ are now to be defined.

The first rule is that any operation with the empty set $\varnothing$ has the empty set as its result.

Arithmetic operations for unbounded intervals of $\overline{\mathbb{IF}}$ can be performed on the computer by using the above formulas for bounded intervals of $\mathbb{IF}$ if in addition a few formal rules for operations with $-\infty$ and $+\infty$ are applied. These rules are shown in the following tables.

[4] Since division by zero does not contribute to the solution set it does not matter whether a parenthesis or bracket is used here.

[5] Fixed point division has always yielded two results.

Addition	$-\infty$	b	$+\infty$
$-\infty$	$-\infty$	$-\infty$	
a	$-\infty$		$+\infty$
$+\infty$		$+\infty$	$+\infty$

Subtraction	$-\infty$	b	$+\infty$
$-\infty$		$-\infty$	$-\infty$
a	$+\infty$		$-\infty$
$+\infty$	$+\infty$	$+\infty$	

Multiplication	$-\infty$	$b < 0$	0	$b > 0$	$+\infty$
$-\infty$	$+\infty$	$+\infty$	0	$-\infty$	$-\infty$
$a < 0$	$+\infty$				$-\infty$
0	0				0
$a > 0$	$-\infty$				$+\infty$
$+\infty$	$-\infty$	$-\infty$	0	$+\infty$	$+\infty$

Division	$-\infty$	$+\infty$
a	0	0

These rules are not new in principle. They are well established in real analysis and **IEEE 754 provides them anyway.** The only rule that goes beyond IEEE 754 is

$$0 * (-\infty) = (-\infty) * 0 = 0 * (+\infty) = (+\infty) * 0 = 0.$$

This rule follows quite naturally from the definition of unbounded intervals. However, it should not be taken as a new mathematical law. It is just a short cut to easily compute the bounds of the result of an operation on unbounded intervals.

The calculus in $\overline{\mathbb{IF}}$ as defined in this document is free of exceptions. If the operations are hardware supported interval arithmetic is almost as fast as simple floating-point arithmetic. See [2].

References

[1] Ulridh Kulisch: *Complete Interval Arithmetic and its Implementation on the Computer*, position paper and the Dagstuhl 2008 proceedings.

[2] Ulrich Kulisch: *Computer Arithmetic and Validity - Theory, Implementation, and Applications*, de Gruyter, Berlin, New York, 2008.

[3] Arnold Neumaier: *Vienna Proposal for Interval Standardization.*

[4] John Pryce: *Text and Rationale for Motion 6: Multi-Format Support.*