Complete Interval Arithmetic and its Implementation on the Computer

Gerd Bohlender, Ulrich Kulisch

Institut für Angewandte und Numerische Mathematik, Universität Karlsruhe

September 29, 2008

Table of Contents

Overview

History of IFIP proposal

- Proposal by IFIP WG 2.5, October 2007 (based on Kulisch/Kirchner paper)
- Intention: include interval arithmetic in revised IEEE 754 standard
- Discussion during conference at Dagstuhl castle, Germany, January 2008
- Formation of a new IEEE standardization group on interval arithmetic, authorized on June 11, 2008
- IEEE 754-2008 standard published on August 29, 2008, without interval arithmetic
- Revised Kulisch paper sent to all participants of scan 2008

Floating-point Arithmetic vs. Interval Arithmetic

Concentration on main mathematical properties of interval arithmetic

Floating point arithmetic:

- Approximation of mathematical results
- Exceptions lead to special values like $+\infty$, $-\infty$, NaN, $+0$, -0 (no real numbers)

Interval arithmetic:

- Result is always an enclosure of mathematical results, i.e. is always exact
- No exceptions can occur (depending on definition of division by intervals containing zero)

Interval Standard

Interval arithmetic can be realized via IEEE 754, but this leads to an unacceptable loss of efficiency.

- Switching of rounding modes is time consuming
- Case distinction for multiplication and division must be programmed in software
- Difficult to fully use parallelism (multithreading, SSE instructions) on current processors

With very little extra hardware, interval arithmetic can be made as fast as floating-point arithmetic.

Remarks on Floating-Point Numbers

Mathematical spaces:

- R set of real numbers (conditionally complete, linearly ordered field)
- $R^* := R \cup \{-\infty\} \cup \{+\infty\}$ (a complete lattice)
- F set of floating-point numbers
- $F^* := F \cup \{-\infty\} \cup \{+\infty\}$
- $-\infty$ and $+\infty$ are no elements of the field of real numbers
- The same holds for NaN, signed zeros, etc. of IEEE 754

We use the notation $\triangledown$, $\triangle$ for the directed roundings. Similarly for the rounded operations.

() denotes open interval bounds, [] denotes closed interval bounds, e.g. $[0, \infty)$ is the interval of all non-negative reals, including zero, not including ∞.

Overview

Arithmetic for Intervals of Real Numbers

Intervals *IR* (subset of the power set *PR*):
set of closed and bounded intervals over the real numbers *R*

- Definition and properties well known (provided that zero is not in the divisor)
- Essential properties: inclusion-isotony and inclusion-monotonicity
- Under the assumption $0 \notin$ *divisor* for division, the intervals of *IR* are an algebraically closed subset of the power set *PR*
- In floating-point arithmetic the crucial operation is division by zero. So we begin our study of extended interval arithmetic by defining division by an interval that contains zero.

Definition of Divison

Division in IR is defined by

$$\bigwedge_{A,B \in IR} A/B := \{a/b \mid a \in A \wedge b \in B\}. \tag{1}$$

The quotient a/b is defined as the inverse operation of multiplication, i.e., as the solution of the equation $b \cdot x = a$ (called backward calculation by Neumaier, reverse operation by Rump).

Thus (1) can be written in the form

$$\bigwedge_{A,B \in IR} A/B := \{x \mid bx = a \wedge a \in A \wedge b \in B\}. \tag{2}$$

For $0 \notin B$ (1) and (2) are equivalent.

Division by Interval Containing Zero: Case Distinction

For $A = [a_1, a_2]$ and $B = [b_1, b_2] \in IR$ with $0 \in B$ the following eight distinct cases can be set out:

1	$0 \in A$,	$0 \in B$.
2	$0 \notin A$,	$B = [0, 0]$.
3	$a_1 \leq a_2 < 0$,	$b_1 < b_2 = 0$.
4	$a_1 \leq a_2 < 0$,	$b_1 < 0 < b_2$.
5	$a_1 \leq a_2 < 0$,	$0 = b_1 < b_2$.
6	$0 < a_1 \leq a_2$,	$b_1 < b_2 = 0$.
7	$0 < a_1 \leq a_2$,	$b_1 < 0 < b_2$.
8	$0 < a_1 \leq a_2$,	$0 = b_1 < b_2$.

Division by Interval Containing Zero: Cases 1-2

Case 1: Since every $x \in R$ fulfills the equation $0 \cdot x = 0$ we obtain $A/B = R = (-\infty, +\infty)$. Here the parentheses indicate that the bounds are not included in the set.

Case 2: the set defined by (2) consists of all elements which fulfill the equation $0 \cdot x = a$ for $a \in A$. Since $0 \notin A$, there is no real number which fulfills this equation. Thus A/B is the empty set, i.e., $A/B = \varnothing$.

Cases 3-8: $0 \notin A$ also. We have already observed under case 2 that the element 0 in B does not contribute to the solution set. So it can be excluded without changing the set A/B.

Division by Interval Containing Zero: Cases 3-8

So the general rule for computing the set A/B by (2) is to remove its zero from the interval B and replace it by a small positive or negative number ϵ as the case may be.

The resulting set is denoted by B' and represented in column 4 of Table 1.

With this B' the solution set A/B' can now easily be computed by applying the rules for closed and bounded real intervals.

The results are shown in column 5 of Table 1. Now the desired result A/B as defined by (2) is obtained if in column 5 ϵ tends to zero.

Thus in the cases 3 to 8 the results are obtained by the limit process $A/B = \lim_{\epsilon \to 0} A/B'$.

Table 1: Division by Interval Containing Zero

case	$A = [a_1, a_2]$	$B = [b_1, b_2]$	B'	A/B'	A/B
1	$0 \in A$	$0 \in B$			$(-\infty, +\infty)$
2	$0 \notin A$	$B = [0, 0]$			$\varnothing$
3	$a_2 < 0$	$b_1 < b_2 = 0$	$[b_1, (-\epsilon)]$	$[a_2/b_1, a_1/(-\epsilon)]$	$[a_2/b_1, +\infty)$
4	$a_2 < 0$	$b_1 < 0 < b_2$	$[b_1, (-\epsilon)]$	$[a_2/b_1, a_1/(-\epsilon)]$	$(-\infty, a_2/b_2]$
			$\cup [\epsilon, b_2]$	$\cup [a_1/\epsilon, a_2/b_2]$	$\cup [a_2/b_1, +\infty)$
5	$a_2 < 0$	$0 = b_1 < b_2$	$[\epsilon, b_2]$	$[a_1/\epsilon, a_2/b_2]$	$(-\infty, a_2/b_2]$
6	$a_1 > 0$	$b_1 < b_2 = 0$	$[b_1, (-\epsilon)]$	$[a_2/(-\epsilon), a_1/b_1]$	$(-\infty, a_1/b_1]$
7	$a_1 > 0$	$b_1 < 0 < b_2$	$[b_1, (-\epsilon)]$	$[a_2/(-\epsilon), a_1/b_1]$	$(-\infty, a_1/b_1]$
			$\cup [\epsilon, b_2]$	$\cup [a_1/b_2, a_2/\epsilon]$	$\cup [a_1/b_2, +\infty)$
8	$a_1 > 0$	$0 = b_1 < b_2$	$[\epsilon, b_2]$	$[a_1/b_2, a_2/\epsilon]$	$[a_1/b_2, +\infty)$

Table: The eight cases of interval division A/B, with $A, B \in IR$, and $0 \in B$.

Division by Interval Containing Zero: Special Cases

The operands A and B of the division A/B in Table 1 are intervals of IR.
The results of the division shown in the last column, however, are no longer intervals of IR.
The result is now an element of the power set PR.
With the exception of case 2 the result is now a set which stretches continuously to $-\infty$ or $+\infty$ or both.

In two cases (rows 4 and 7 in Table 1) the result consists of the union of two distinct sets of the form $(-\infty, c_2] \cup [c_1, +\infty)$.
These cases can easily be identified by the signs of the bounds of the divisor before the division is executed.
For interval multiplication and division a case selection has to be done before the operations are performed anyhow, see [Kirchner].
In the two cases (rows 4 and 7 in Table 1) the sign of b_1 is negative and the sign of b_2 is positive.

Division by Interval Containing Zero: Cases 4 and 7

Division by zero does not contribute to the solution set.

Therefore, the set $b_1 < 0 < b_2$ devolves into the two distinct sets $[b_1, 0)$ and $(0, b_2]$ and division by the set $b_1 < 0 < b_2$ actually means two divisions.

The result of the two divisions consists of the two distinct sets shown in rows 4 and 7 of Table 1.

It is highly desirable to perform the two divisions sequentially. Then the two cases (rows 4 and 7) of Table 1 where an operation delivers two distinct results can be eliminated.

Table 2: Exclusion of Critical Cases

case	$A = [a_1, a_2]$	$B = [b_1, b_2]$	A/B
1	$0 \in A$	$0 \in B$	$(-\infty, +\infty)$
2	$0 \notin A$	$B = [0, 0]$	$\varnothing$
3	$a_2 < 0$	$b_1 < b_2 = 0$	$[a_2/b_1, +\infty)$
4	$a_2 < 0$	$0 = b_1 < b_2$	$(-\infty, a_2/b_2]$
5	$a_1 > 0$	$b_1 < b_2 = 0$	$(-\infty, a_1/b_1]$
6	$a_1 > 0$	$0 = b_1 < b_2$	$[a_1/b_2, +\infty)$

Table: The six cases of interval division with $A, B \in IR$, and $0 \in B$.

Extended Intervals

Thus only four kinds of result come from division by an interval of IR which contains zero:

$$\emptyset, \quad (-\infty, a], \quad [b, +\infty), \quad \text{and} \quad (-\infty, +\infty). \tag{3}$$

We call such elements extended intervals. The union of the set of closed and bounded intervals of IR with the set of extended intervals is denoted by (IR). The elements of the set (IR) are themselves simply called intervals. (IR) is the set of closed intervals of R. (A subset of R is called closed if its complement is open.)

Intervals of IR and of (IR) are sets of real numbers. $-\infty$ and $+\infty$ are not elements of these intervals.

Overview

Definition of Floating-Point Interval Operations

On the computer, arithmetic in IR is approximated by an arithmetic in IF.

IF is the set of closed and bounded intervals with bounds of F. An interval of IF represents the continuous set of real numbers between the floating-point bounds.

Arithmetic operations in IF are defined by those in IR with the lower bound of the result rounded downwards and the upper bound rounded upwards.

Table 3: The 6 Cases for Floating-Point Intervals

case	$A = [a_1, a_2]$	$B = [b_1, b_2]$	$A \oslash B$
1	$0 \in A$	$0 \in B$	$(-\infty, +\infty)$
2	$0 \notin A$	$B = [0, 0]$	$\varnothing$
3	$a_2 < 0$	$b_1 < b_2 = 0$	$[a_2 \triangledown b_1, +\infty)$
4	$a_2 < 0$	$0 = b_1 < b_2$	$(-\infty, a_2 \vartriangle b_2]$
5	$a_1 > 0$	$b_1 < b_2 = 0$	$(-\infty, a_1 \vartriangle b_1]$
6	$a_1 > 0$	$0 = b_1 < b_2$	$[a_1 \triangledown b_2, +\infty)$

Table: The six cases of interval division with $A, B \in IF$, and $0 \in B$.

Table 4: The 6 Cases in Another Layout

	$B = [0,0]$	$b_1 < b_2 = 0$	$0 = b_1 < b_2$
$a_2 < 0$	$\varnothing$	$[a_2 \bigtriangledown b_1, +\infty)$	$(-\infty, a_2 \bigtriangleup b_2]$
$a_1 \leq 0 \leq a_2$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$
$0 < a_1$	$\varnothing$	$(-\infty, a_1 \bigtriangleup b_1]$	$[a_1 \bigtriangledown b_2, +\infty)$

Table: The result of the interval division with $A, B \in IF$, and $0 \in B$.

We explicitly stress that the symbols $-\infty$, $+\infty$ are used here only to represent the resulting sets. These symbols are not elements of these sets.

Introduction of Extended Intervals

Table 3 and Table 4 show that division by an interval of IF which contains zero on the computer also leads to extended intervals as shown in (3) with $a, b \in F$.

The union of the set of closed and bounded intervals of IF with such extended intervals is denoted by (IF).

(IF) is the set of closed intervals of real numbers where all finite bounds are elements of F.

Except for the empty set, extended intervals also represent continuous sets of real numbers.

Flag and Handling Routine

Division by an interval $[b_1, b_2]$ with $b_1 < 0 < b_2$ actually consists of two divisions by the distinct sets $[b_1, 0)$ and $(0, b_2]$ the result of which again consists of two distinct sets.

In the user's program, however, the two divisions appear as one single operation, as division by an interval $[b_1, b_2]$ with $b_1 < 0 < b_2$.

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.

Options for Handling Distinct Intervals

A handling routine could perform one of the following operations:

- modify the operands and recompute,
- continue the computation with one of the sets and ignore the other,
- put one of the sets on a list and continue the computation with the other one,
- return the entire set of real numbers $(-\infty, +\infty)$ as result and continue the computation,
- stop computing,
- alternative approach: return an improper interval $[c_1, c_2]$ where the left hand bound is higher than the right hand bound $c_1 > c_2$, this represents the two distinct sets $(-\infty, c_2]$ and $[c_1, +\infty)$,
- any other action.

Solution by Parallelization

A somewhat natural solution would be to continue the computation on different processors, one for each interval.

But the situation can occur repeatedly. How many processors would we need? Future multicore units will provide a large number of processors. They will suffice for quite a while.

A similar situation occurs in global optimization using subdivision. After a certain test several candidates may be left for further investigation.

Overview

Arithmetic for Extended Intervals

On the computer, arithmetic in (IF) has to be provided. Threfore, we skip the arithmetic in (IR), which is similar.

First of all any operation with the empty set is again defined to be the empty set.

The general procedure for defining all other operations follows a continuity principle. Bounds like $-\infty$ and $+\infty$ in the operands A and B are replaced by a very large negative number ($-\Omega$) and a very large positive number ($+\Omega$) respectively. Then the basic rules for the arithmetic operations in IR and IF are applied. In the following tables these rules are repeated and printed in bold letters. In the resulting formulas a very large negative number is then shifted to $-\infty$ and a very large positive number to $+\infty$.

Simplification of Formulas

As a short cut for obtaining the resulting rules very simple and well established rules of real analysis like

- $\infty * x = \infty$ for $x > 0$,
- $\infty * x = -\infty$ for $x < 0$,
- $x/\infty = x/-\infty = 0$,
- $\infty * \infty = \infty$,
- $(-\infty) * \infty = -\infty$

can be applied together with variants obtained by applying the sign rules and the law of commutativity.

Two Special Cases

Two situations have to be treated separately. These are the cases shown in rows 1 and 2 of Table 1.

If $0 \in A$ and $0 \in B$ (row 1 of Table 1), the result consists of all the real numbers, i.e., $A/B = (-\infty, +\infty)$. This applies to rows 2, 5, 6 and 8 of Table 9.

If $0 \notin A$ and $B = [0, 0]$ (row 2 of Table 1), the result of the division is the empty set, i.e., $A/B = \varnothing$. This applies to rows 1, 3, 4 and 7 of column 1 of Table 9.

Algebraically Closed (After Splitting)

In summary it can be said that after splitting an interval $[b_1, b_2]$ with $b_1 < 0 < b_2$ into two distinct intervals $[b_1, 0)$ and $(0, b_2]$ the result of arithmetic operations for intervals of (IF) always leads to intervals of (IF) again.

Table 5: Addition

Addition	$(-\infty, b_2]$	$[b_1, b_2]$	$[b_1, +\infty)$	$(-\infty, +\infty)$
$(-\infty, a_2]$	$(-\infty, a_2 \triangle b_2]$	$(-\infty, a_2 \triangle b_2]$	$(-\infty, +\infty)$	$(-\infty, +\infty)$
$[a_1, a_2]$	$(-\infty, a_2 \triangle b_2]$	$[\boldsymbol{a_1} \triangledown \boldsymbol{b_1}, \boldsymbol{a_2} \triangle \boldsymbol{b_2}]$	$[a_1 \triangledown b_1, +\infty)$	$(-\infty, +\infty)$
$[a_1, +\infty)$	$(-\infty, +\infty)$	$[a_1 \triangledown b_1, +\infty)$	$[a_1 \triangledown b_1, +\infty)$	$(-\infty, +\infty)$
$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$

Table: Addition of extended intervals on the computer.

Table 6: Subtraction

Subtraction	$(-\infty, b_2]$	$[b_1, b_2]$	$[b_1, +\infty)$	$(-\infty, +\infty)$
$(-\infty, a_2]$	$(-\infty, +\infty)$	$(-\infty, a_2 \vartriangle b_1]$	$(-\infty, a_2 \vartriangle b_1]$	$(-\infty, +\infty)$
$[a_1, a_2]$	$[a_1 \triangledown b_2, +\infty)$	$[\boldsymbol{a_1} \triangledown \boldsymbol{b_2}, \boldsymbol{a_2} \vartriangle \boldsymbol{b_1}]$	$(-\infty, a_2 \vartriangle b_1]$	$(-\infty, +\infty)$
$[a_1, +\infty)$	$[a_1 \triangledown b_2, +\infty)$	$[a_1 \triangledown b_2, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$
$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$

Table: Subtraction of extended intervals on the computer.

Table 7: Multiplication

Multiplication	$[b_1, b_2]$ $b_2 \le 0$	$[b_1, b_2]$ $b_1 < 0 < b_2$	$[b_1, b_2]$ $b_1 \ge 0$	$[0, 0]$	$(-\infty, b_2]$ $b_2 \le 0$	$(-\infty, b_2]$ $b_2 \ge 0$	$[b_1, +\infty)$ $b_1 \le 0$	$[b_1, +\infty)$ $b_1 \ge 0$	$(-\infty, +\infty)$
$[a_1, a_2], a_2 \le 0$	$[\mathbf{a_2 \bigtriangledown b_2}, \mathbf{a_1 \bigtriangleup b_1}]$	$[\mathbf{a_1 \bigtriangledown b_2}, \mathbf{a_1 \bigtriangleup b_1}]$	$[\mathbf{a_1 \bigtriangledown b_2}, \mathbf{a_2 \bigtriangleup b_1}]$	$[0, 0]$	$[a_2 \bigtriangledown b_2, +\infty)$	$[a_1 \bigtriangledown b_2, +\infty)$	$(-\infty, a_1 \bigtriangleup b_1]$	$(-\infty, a_2 \bigtriangleup b_1]$	$(-\infty, +\infty)$
$a_1 < 0 < a_2$	$[\mathbf{a_2 \bigtriangledown b_1}, \mathbf{a_1 \bigtriangleup b_1}]$	$[\mathbf{min}(\mathbf{a_1 \bigtriangledown b_2}, \mathbf{a_2 \bigtriangledown b_1}), \mathbf{max}(\mathbf{a_1 \bigtriangleup b_1}, \mathbf{a_2 \bigtriangleup b_2})]$	$[\mathbf{a_1 \bigtriangledown b_2}, \mathbf{a_2 \bigtriangleup b_2}]$	$[0, 0]$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$
$[a_1, a_2], a_1 \ge 0$	$[\mathbf{a_2 \bigtriangledown b_1}, \mathbf{a_1 \bigtriangleup b_2}]$	$[\mathbf{a_2 \bigtriangledown b_1}, \mathbf{a_2 \bigtriangleup b_2}]$	$[\mathbf{a_1 \bigtriangledown b_1}, \mathbf{a_2 \bigtriangleup b_2}]$	$[0, 0]$	$(-\infty, a_1 \bigtriangleup b_2]$	$(-\infty, a_2 \bigtriangleup b_2]$	$[a_2 \bigtriangledown b_1, +\infty)$	$[a_1 \bigtriangledown b_1, +\infty)$	$(-\infty, +\infty)$
$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$
$(-\infty, a_2], a_2 \le 0$	$[a_2 \bigtriangledown b_2, +\infty)$	$(-\infty, +\infty)$	$(-\infty, a_2 \bigtriangleup b_1]$	$[0, 0]$	$[a_2 \bigtriangledown b_2, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, a_2 \bigtriangleup b_1]$	$(-\infty, +\infty)$
$(-\infty, a_2], a_2 \ge 0$	$[a_2 \bigtriangledown b_1, +\infty)$	$(-\infty, +\infty)$	$(-\infty, a_2 \bigtriangleup b_2]$	$[0, 0]$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$
$[a_1, +\infty), a_1 \le 0$	$(-\infty, a_1 \bigtriangleup b_1]$	$(-\infty, +\infty)$	$[a_1 \bigtriangledown b_2, +\infty)$	$[0, 0]$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$
$[a_1, +\infty), a_1 \ge 0$	$(-\infty, a_1 \bigtriangleup b_2]$	$(-\infty, +\infty)$	$[a_1 \bigtriangledown b_1, +\infty)$	$[0, 0]$	$(-\infty, a_1 \bigtriangleup b_2]$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$[a_1 \bigtriangledown b_1, +\infty)$	$(-\infty, +\infty)$
$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$[0, 0]$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$

Table: Multiplication of extended intervals on the computer.

Table 8: Division by Regular Interval

Division $\mathbf{0 \notin \boldsymbol{B}}$	$[b_1, b_2]$ $b_2 < 0$	$[b_1, b_2]$ $b_1 > 0$	$(-\infty, b_2]$ $b_2 < 0$	$[b_1, +\infty)$ $b_1 > 0$
$[a_1, a_2], a_2 \leq 0$	$[\boldsymbol{a_2 \triangledown b_1, a_1 \vartriangle b_2}]$	$[\boldsymbol{a_1 \triangledown b_1, a_2 \vartriangle b_2}]$	$[0, a_1 \vartriangle b_2]$	$[a_1 \triangledown b_1, 0]$
$[a_1, a_2], a_1 < 0 < a_2$	$[\boldsymbol{a_2 \triangledown b_2, a_1 \vartriangle b_2}]$	$[\boldsymbol{a_1 \triangledown b_1, a_2 \vartriangle b_1}]$	$[a_2 \triangledown b_2, a_1 \vartriangle b_2]$	$[a_1 \triangledown b_1, a_2 \vartriangle b_1]$
$[a_1, a_2], a_1 \geq 0$	$[\boldsymbol{a_2 \triangledown b_2, a_1 \vartriangle b_1}]$	$[\boldsymbol{a_1 \triangledown b_2, a_2 \vartriangle b_1}]$	$[a_2 \triangledown b_2, 0]$	$[0, a_2 \vartriangle b_1]$
$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$
$(-\infty, a_2], a_2 \leq 0$	$[a_2 \triangledown b_1, +\infty)$	$(-\infty, a_2 \vartriangle b_2]$	$[0, +\infty)$	$(-\infty, 0]$
$(-\infty, a_2], a_2 \geq 0$	$[a_2 \triangledown b_2, +\infty)$	$(-\infty, a_2 \vartriangle b_1]$	$[a_2 \triangledown b_2, +\infty)$	$(-\infty, a_2 \vartriangle b_1]$
$[a_1, +\infty), a_1 \leq 0$	$(-\infty, a_1 \vartriangle b_2]$	$[a_1 \triangledown b_1, +\infty)$	$(-\infty, a_1 \vartriangle b_2]$	$[a_1 \triangledown b_1, +\infty)$
$[a_1, +\infty), a_1 \geq 0$	$(-\infty, a_1 \vartriangle b_1]$	$[a_1 \triangledown b_2, +\infty)$	$(-\infty, 0]$	$[0, +\infty)$
$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$

Table: Division of extended intervals with $0 \notin B$ on the computer.

Table 9: Division by Interval Containing Zero

Division $\mathbf{0 \in B}$	$B = [0,0]$	$[b_1, b_2]$ $b_1 < b_2 = 0$	$[b_1, b_2]$ $0 = b_1 < b_2$	$(-\infty, b_2]$ $b_2 = 0$	$[b_1, +\infty)$ $b_1 = 0$	$(-\infty, +\infty)$
$[a_1, a_2], a_2 < 0$	$\varnothing$	$[\boldsymbol{a_2} \triangledown \boldsymbol{b_1}, +\infty)$	$(-\infty, \boldsymbol{a_2} \vartriangle \boldsymbol{b_2}]$	$[0, +\infty)$	$(-\infty, 0]$	$(-\infty, +\infty)$
$[a_1, a_2], a_1 \leq 0 \leq a_2$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$
$[a_1, a_2], a_1 > 0$	$\varnothing$	$(-\infty, \boldsymbol{a_1} \vartriangle \boldsymbol{b_1}]$	$[\boldsymbol{a_1} \triangledown \boldsymbol{b_2}, +\infty)$	$(-\infty, 0]$	$[0, +\infty)$	$(-\infty, +\infty)$
$(-\infty, a_2], a_2 < 0$	$\varnothing$	$[a_2 \triangledown b_1, +\infty)$	$(-\infty, a_2 \vartriangle b_2]$	$[0, +\infty)$	$(-\infty, 0]$	$(-\infty, +\infty)$
$(-\infty, a_2], a_2 > 0$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$
$[a_1, +\infty), a_1 < 0$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$
$[a_1, +\infty), a_1 > 0$	$\varnothing$	$(-\infty, a_1 \vartriangle b_1]$	$[a_1 \triangledown b_2, +\infty)$	$(-\infty, 0]$	$[0, +\infty)$	$(-\infty, +\infty)$
$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$

Table: Division of extended intervals with $0 \in B$ on the computer.

Simplification of Tables 5-9

On the computer actually only the basic rules for addition, subtraction, multiplication, and division for closed and bounded intervals of *IF* including division by an interval that includes zero need to be provided.

The remaining rules shown in the tables can automatically be produced out of these basic rules by the computer itself if a few well established rules for computing with $-\infty$ and $+\infty$ are formally applied. With $x \in F$ these rules are

$$
\begin{array}{ll}
\infty + x = \infty, & -\infty + x = -\infty, \\
-\infty + (-\infty) = (-\infty) \cdot \infty = -\infty, & \infty + \infty = \infty \cdot \infty = \infty, \\
\infty \cdot x = \infty \text{ for } x > 0, & \infty \cdot x = -\infty \text{ for } x < 0, \\
\frac{x}{\infty} = \frac{x}{-\infty} = 0, &
\end{array}
$$

together with variants obtained by applying the sign rules and the law of commutativity.

Multiplication of Zero and Infinity

If in an interval operand a bound is $-\infty$ or $+\infty$ the multiplication with 0 is performed as if the following rules would hold

$$0 \cdot (-\infty) = 0 \cdot (+\infty) = (-\infty) \cdot 0 = (+\infty) \cdot 0 = 0. \qquad (4)$$

These rules have no meaning otherwise.

We stress that (4) does not define new rules for the multiplication of 0 with $+\infty$ or $-\infty$. It just describes a short cut for applying the continuity principle mentioned earlier in this section.

Overview

Comparisons

Comparisons in (IF):

- equality,
- less than or equal,
- set inclusion

with bounds in $F^* := F \cup \{-\infty\} \cup \{+\infty\}$

Comparisons for the empty set $\varnothing$ are defined in a straightforward way.

$\{(IF), \leq\}$ is a lattice.

Lattice Operations

$$\begin{aligned} inf(A, B) &:= [max(a_1, b_1), min(a_2, b_2)] \quad \text{or the empty set } \varnothing, \\ sup(A, B) &:= [min(a_1, b_1), max(a_2, b_2)]. \end{aligned}$$

The intersection of an interval with the empty set is the empty set. The interval hull with the empty set is the other operand.

$\{(IF), \subseteq\}$ is also a lattice.

Overview

Evaluation of Functions

Interval evaluation of real functions fits smoothly into complete interval arithmetic as developed in the previous sections. Let f be a function and D_f its domain of definition. For an interval $X \subseteq D_f$, the range $f(X)$ of f is defined as the set of the function's values for all $x \in X$:

$$f(X) := \{f(x) | x \in X\}. \tag{5}$$

A function $f(x) = a/(x - b)$ with $D_f = R \setminus \{b\}$ is sometimes called singular or discontinuous at $x = b$. Both descriptions are meaningless in a strict mathematical sense. Since $x = b$ is not of the domain of f, the function cannot have any property at $x = b$.

Examples with Singular Points

In this strict sense a division $2/[b_1, b_2]$ by an interval $[b_1, b_2]$ that contains zero as an interior point, $b_1 < 0 < b_2$, means: $2/([b_1, 0) \cup (0, b_2]) = 2/[b_1, 0) \cup 2/(0, b_2] = (-\infty, 2/b_1] \cup [2/b_2, +\infty)$.

We give two examples:

$f(x) = 4/(x-2)^2, \quad D_f = R \setminus \{2\}, \quad X = [1, 4],$
$f([1, 2) \cup (2, 4]) = f([1, 2)) \cup f((2, 4]) = [4, +\infty) \cup [1, +\infty) = [1, +\infty).$

$g(x) = 2/(x-2), \quad D_g = R \setminus \{2\}, \quad X = [1, 3],$
$g([1, 2) \cup (2, 3]) = g([1, 2)) \cup g((2, 3]) = (-\infty, -2] \cup [2, +\infty),$

Here the flag *distinct intervals* should be raised and signaled to the user. The user may then choose a routine to apply which is appropriate for the application.

Alternative Definition

It has been suggested in the literature that the entire set of real numbers $(-\infty, +\infty)$ be returned as result in this case.

However, this may be a large overestimation of the true result and there are applications (Newton's method) which need the accurate answer.

To return the entire set of real numbers is also against a basic principle of interval arithmetic—to keep the sets as small as possible. So a standard should have the most accurate answer returned.

On the computer, interval evaluation of a real function $f(x)$ for $X \subseteq D_f$ should deliver a highly accurate enclosure of the range $f(X)$ of the function.

Evaluation Outside of the Domain

Evaluation of a function $f(x)$ for an interval X with $X \cap D_f = \varnothing$, of course, does not make sense, since $f(x)$ is not defined for values outside its domain D_f.

The empty set $\varnothing$ should be delivered and an error message may be given to the user.

Evaluation Outside of the Domain

There are, however, applications in interval arithmetic where information about a function f is useful when X exceeds the domain D_f of f. The interval X may also be the result of overestimation during an earlier interval computation.

In such cases the range of f can only be computed for the intersection $X' := X \cap D_f$:

$$f(X') := f(X \cap D_f) := \{f(x) | x \in X \cap D_f\}. \qquad (6)$$

To prevent the wrong conclusions being drawn, the user must be informed that the interval X had to be reduced to $X' := X \cap D_f$ to compute the delivered range.

A particular flag for *domain overflow* may serve this purpose. An appropriate routine can be chosen and applied if this flag is raised.

Examples

$l(x) := log(x), \quad D_{log} = (0, +\infty),$
$log((0, 2]) = (-\infty, log(2)].$
But also
$log([-5, 2]') = log((0, 2]) = (-\infty, log(2)].$
The flag *domain overflow* should be set. It informs the user that the function has been evaluated for the intersection
$X' := X \cap D_f = [-5, 2] \cap (0, +\infty) = (0, 2].$

$h(x) := sqrt(x), \quad D_{sqrt} = [0, +\infty),$
$sqrt([1, 4]) = [1, 2],$
$sqrt([4, +\infty)) = [2, +\infty).$
$sqrt([-5, -1]) = \emptyset$, an error message *sqrt* not defined for $[-5, -1]$, may be given to the user.
$sqrt([-5, 4]') = sqrt([0, 4]) = [0, 2].$
The flag *domain overflow* should be set. It informs the user that the function has been evaluated for the intersection
$X' := X \cap D_f = [-5, 4] \cap [0, +\infty) = [0, 4].$

Examples

$k(x) := sqrt(x) - 1, \quad D_k = [0, +\infty),$
$k([-4, 1]') = k([0, 1]) = sqrt([0, 1]) - 1 = [-1, 0].$
The flag *domain overflow* should be set. It informs the user that the function has been evaluated for the intersection
$X' := X \cap D_f = [-4, 1] \cap [0, +\infty) = [0, 1].$

Overview

Dot Product

For interval evaluation of an algorithm in the real number field, increasing the precision by k digits reduces the error bounds by b^{-k} [R. E. Moore]

Results can always be guaranteed to a number of correct digits by using variable precision interval arithmetic [Alefeld, Rump]

Variable length interval arithmetic can be made very fast by an exact dot product and complete arithmetic [Kulisch].

There is no way to compute a dot product faster than the exact result. By pipelining, it can be computed in the time the processor needs to read the data.

Conclusion

The tremendous progress in computer technology should be accompanied by extension of the mathematical capacity of the computer.

A balanced standard for computer arithmetic should require that the basic components of modern computing be provided by the computer's hardware:

- floating-point arithmetic,
- interval arithmetic,
- an exact dot product