P1788: IEEE Standard For Interval Arithmetic Version 04.1

John Pryce and Christian Keil, Technical Editors

⚠ To the P1788 reader.

General. Passages in this color are my editorial comments: mostly asking for answers to or debate on a question; or giving my opinion; or noting changes made.

Ignore text like this: (some text)→*delete*[1]. It belongs in the full text but isn't relevant to the part currently under discussion, and it didn't seem worth rewriting the whole sentence. !

Definitions. Christian has done a lot of work on these. We believe they are now reasonably complete, and consistent with each other and the text. Only those definitions relevant to Level 1 are included in the current text.

Required and recommended operations. The lists in §5.6, 5.7 have not been voted on in their present form. I formed those in §5.6.1, 5.7.1 from the original lists of Jürgen Wolff von Gudenberg in Motion 10, much modified by subsequent discussion.

Maybe the group will accept them as they are; but if they prove controversial, we shall need separate motions to discuss them.

Provisional items. The subclauses §5.6.5 Constructors, §5.6.6 Numeric functions of intervals, and §5.6.7. Boolean functions of intervals, are purely provisional, and are not part of the present motion. I look forward to people starting discussion and submitting motions to decide these items.

[1] *CK*

3. Notation, abbreviations, definitions

3.1. Notation and abbreviations.

754	IEEE-Std-754-2008 "IEEE Standard for Floating-Point Arithmetic".
$\mathbb{R}$	the set of real numbers.
$\mathbb{R}^*$	the set of extended real numbers, $\mathbb{R} \cup \{-\infty, +\infty\}$.
$\mathbb{IR}$	the set of bounded, nonempty closed real intervals.
$\overline{\mathbb{IR}}$	the set of closed real intervals, including unbounded intervals and the empty set.
$\mathbb{F}$	the subset of $\mathbb{R}^*$ representable in a given floating point format.
$\mathbb{IF}$	the intervals of $\mathbb{IR}$ whose bounds are in $\mathbb{F}$.
$\overline{\mathbb{IF}}$	the intervals of $\overline{\mathbb{IR}}$ whose bounds are in $\mathbb{F}$.
Empty	the empty set.
Entire	the whole real line.
NaI	Not an Interval.
NaN	Not a Number.
qNaN	quiet NaN.
sNaN	signaling NaN.
$x, y, \ldots$ [resp. $f, g, \ldots$]	generic notation for a numeric value [resp. numeric function].
$\boldsymbol{x}, \boldsymbol{y}, \ldots$ [resp. $\boldsymbol{f}, \boldsymbol{g}, \ldots$]	generic notation for an interval value [resp. interval function].
$\mathsf{f}, \mathsf{g}, \ldots$	generic notation for an expression, producing a function by evaluation.
Domain(f)	domain of a point-function f.
Range($f \mid \boldsymbol{s}$)	range of a point-function f over a set $\boldsymbol{s}$. Same as image of $\boldsymbol{s}$ under f.

3.2. Definitions.

⚠ Definitions belonging to Levels 2 onward have been temporarily removed.

Dan Zuras notes that the Definitions subclause should be self-contained, i.e. one does not need to look outside it to understand a definition, at least in outline. Please check if this rule is flouted.

3.2.1. **arithmetic operation.** A function provided by an implementation (see Definition 3.2.8). It comes in three forms: the **point** operation, which is a mathematical real function of real variables such as $\log(x)$; one or more **interval versions**, each being an interval extension of the point operation. (; and one or more **decorated interval versions**, each being a decorated interval extension of the point operation)$\rightarrow$*delete*[2] !

Together with the interval non-arithmetic operations (§5.4.1), these form the implementation's **library**, which splits into the **point library** containing point functions and the **interval library** containing interval functions.

A **basic arithmetic operation** is one of the six functions $+$, $-$, $\times$, $\div$, fma and square root.

Constants such as 3 and π are regarded as arithmetic operations whose number of arguments is zero. Details in §5.4.

3.2.2. **box.** See Definition 3.2.10.

3.2.3. **domain.** For a function with arguments taken from some set, the **domain** comprises those points in the set at which the function has a value. The domain of an arithmetic operation is part of its definition. E.g., the (point) arithmetic operation of division x/y, in this standard, has arguments (x, y) in $\mathbb{R}^2$, and its domain is the set $\{\,(x,y) \in \mathbb{R}^2 \mid y \neq 0\,\}$. See also Definition 3.2.15.

3.2.4. **elementary function.** Synonymous with arithmetic operation.

3.2.5. **expression.** A symbolic object f that is either a symbolic variable or, recursively, of the form $\phi(\mathsf{g}_1, \ldots, \mathsf{g}_k)$, where ϕ is the name of a k-argument arithmetic or non-arithmetic operation and the g_i are expressions. It is an **arithmetic expression** if all its operations are arithmetic operations. Writing $\mathsf{f}(\mathsf{z}_1, \ldots, \mathsf{z}_n)$ makes f a **bound expression**, giving it an **argument list** comprising the variables z_i in that order, which must include all those that occur in f.

Details in §5.5.1. For the ways in which an expression defines a function, see §5.5.2, 5.5.3.

3.2.6. **fma.** Fused multiply-add operation computing $x \times y + z$ with only one rounding.

3.2.7. **hull.** (Full name: **interval hull**.) When not qualified by the name of a finite-precision interval type, the hull of a subset $\boldsymbol{s}$ of $\mathbb{R}$ is the tightest interval containing $\boldsymbol{s}$.

[2] *CK* 2011-10-15 Not in the motion.

3.2.8. **implementation.** When used without qualification, means a realization of an interval arithmetic conforming to the specification of this standard.

3.2.9. **inf-sup.** Describes a representation of an interval based on its lower and upper bounds.

3.2.10. **interval.** A closed connected set of real numbers, may be empty, bounded or unbounded. Belongs to the set of intervals $\overline{\mathbb{IR}}$.

A **box** or **interval vector** is an n-dimensional interval, i. e. a tuple $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ where the $\boldsymbol{x}_i$ are 1-dimensional intervals. Often identified with the cartesian product $\boldsymbol{x}_1 \times \ldots \times \boldsymbol{x}_n \subseteq \mathbb{R}^n$, it is empty if any of the $\boldsymbol{x}_i$ is empty. Details in §5.2.

3.2.11. **interval extension.** An interval extension of a point function f is a function $\boldsymbol{f}$ from intervals to intervals such that $f(x)$ belongs to $\boldsymbol{f}(\boldsymbol{x})$ whenever x belongs to $\boldsymbol{x}$ and $f(x)$ is defined. Details in §5.4.3.

3.2.12. **interval function, interval mapping.** A function from intervals to intervals is called an interval function if it is an interval extension of a point function, and an interval mapping otherwise. Details in §5.4.3.

3.2.13. **interval library.** See Definition 3.2.1.

3.2.14. **interval version.** See Definition 3.2.1.

3.2.15. **natural domain.** For an arithmetic expression $\mathsf{f}(\mathsf{z}_1, \ldots, \mathsf{z}_n)$, the natural domain is the set of $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ where the expression defines a value for the associated point function $f(x)$. See §5.5.

3.2.16. **number.** Any member of the set $\mathbb{R} \cup \{-\infty, +\infty\}$ of extended reals: a **finite number** if it belongs to $\mathbb{R}$, else an **infinite number**. See §5.1

3.2.17. **point function, point operation.** A mathematical function of real variables: that is, a map f from its domain, which is a subset of $\mathbb{R}^n$, to $\mathbb{R}^m$, where $n \geq 0, m > 0$. It is **scalar** if $m = 1$. Any arithmetic expression $\mathsf{f}(\mathsf{z}_1, \ldots, \mathsf{z}_n)$ defines a (usually scalar) point function, whose domain is the natural domain of f.

3.2.18. **point library.** See Definition 3.2.1.

3.2.19. **range.** The range, $\mathrm{Range}(f \mid \boldsymbol{s})$, of a point function f over a subset $\boldsymbol{s}$ of $\mathbb{R}^n$ is the set of all values that f assumes at those points of $\boldsymbol{s}$ where it is defined, i.e. $\{\, f(x) \mid x \in \boldsymbol{s} \text{ and } x \in \mathrm{Domain}\, f \,\}$.

3.2.20. **tightest.** Smallest in the set sense. The tightest set (unique, if it exists) with a given property is contained in every other set with that property.

Relationships between specification levels for interval arithmetic (for a given finite-precision interval type $\mathbb{T}$)		
Level 1	Number system $\mathbb{R}$. Set $\overline{\mathbb{IR}}$ of allowed intervals over $\mathbb{R}$. Principles of how $+, -, \times, \div$ and other arithmetic operations are extended to intervals.	Mathematical Model level.
	$\downarrow$ $\mathbb{T}$-*interval hull* total, many-to-one[a] *identity map* $\uparrow$ total, one-to-one[b]	
Level 2	A finite subset $\mathbb{T}$ of $\overline{\mathbb{IR}}$ —the $\mathbb{T}$-interval datums— and operations on them.	Interval datum level.
	"represents" $\uparrow$ partial, many-to-one, onto[c]	
Level 3	Representation of a $\mathbb{T}$-interval, e.g. by two floating point numbers.	Representations of interval data.
	"encodes" $\uparrow$ partial, many-to-one, onto[d]	
Level 4	Encodings `0111000...`	Bit strings.

TABLE 1. Specification levels for interval arithmetic

4. Structure of the standard in levels

4.1. Specification levels overview.

The standard is structured into four levels, summarized in Table 1, that match the levels defined in the 754 standard, see 754 Table 3.1.

Level 1, in Clause 5, defines the mathematical theory underlying the standard. The entities at this level are mathematical intervals and operations on them. Conforming implementations shall implement this theory.

Level 2, in Clause 6, is the central part of the standard. Here the mathematical theory is approximated by an implementation-defined finite set of entities and operations. A level 2 entity is called a *datum* (plural "datums" in this standard, since "data" is often misleading). In addition to an ordinary (bare) interval, this level defines a *decorated* interval, comprising a bare interval and a *decoration*. Decorations implement the P1788 exception handling mechanism.

Level 3, in Clause 7, is concerned with the representation of interval datums—usually but not necessarily in terms of floating point values. A level 3 entity is an *interval object*. The Level 3 requirements in this standard are few, and concern mappings from internal representations to external ones, such as interchange formats and I/O.

Level 4, in Clause 8, is concerned with the encoding of interval objects. A level 4 entity is a *bit string*. This standard makes no Level 4 requirements.

The arrows in the table denote mappings between levels. The phrases in italics name these mappings. Each phrase "total, many-to-one", etc., labeled with a letter [a] to [d], is descriptive of the mapping and is equivalent to the corresponding labeled fact below.

a. Each mathematical interval (indeed each subset of $\mathbb{R}$) has a unique interval datum as its $\mathbb{T}$-hull.
b. Each interval datum is a mathematical interval.
c. Not every interval object necessarily represents an interval datum, but when it does, that datum is unique. Each interval datum has at least one representation, and may have more than one.
d. Not every interval encoding necessarily encodes an interval object, but when it does, that object is unique. Each interval object has at least one encoding and may have more than one.

4.2. Conformance requirements.

⚠ Temporarily omitted since they are almost entirely about levels 2 onward.

5. Level 1 description

In this clause, subclauses §5.1 to §5.5 describe the theory of mathematical intervals and interval functions that underlies this standard. The relation between expressions and the point or interval functions that they define is specified, since it is central to the Fundamental Theorem of Interval Arithmetic. Subclauses §5.6, 5.7 list the required and recommended *arithmetic operations* (also called elementary functions) with their mathematical specifications. (Subclause 5.8 describes, at a mathematical level, the system of *decorations* that is used among other things for exception handling in P1788.)$\rightarrow$*delete*[3] !

5.1. Numbers. Following the terminology of 754 (e.g., 754§2.1.25), any member of the set $\mathbb{R} \cup \{-\infty, +\infty\}$ of extended reals is called a number: it is a **finite number** if it belongs to $\mathbb{R}$, else an **infinite number**.

5.2. Intervals. The set of mathematical intervals in $\mathbb{R}$, denoted $\overline{\mathbb{IR}}$, comprises[4] all closed intervals of real numbers

$$\boldsymbol{x} = [\underline{x}, \overline{x}] := \{\, x \in \mathbb{R} \mid \underline{x} \le x \le \overline{x} \,\},$$

where the bounds $\underline{x}, \overline{x}$ are extended-real numbers satisfying $-\infty \le \underline{x} \le \overline{x} \le +\infty$.
[*Notes.*

- *In particular, the empty interval* $[-\infty, -\infty] = [+\infty, +\infty] = \emptyset$ *belongs to* $\overline{\mathbb{IR}}$.
- *The above definition implies* $-\infty$ *and* $+\infty$ *can be bounds of an interval, but are never members of it.*
- *The round bracket or outward bracket notations for closed intervals in* $\mathbb{R}$ *with an infinite end point (e.g.,* $[2, +\infty)$ *or* $[2, +\infty[$ *instead of* $[2, +\infty]$*) are not used in this document but are not considered wrong.*

]

A **box** or **interval vector** is an n-tuple $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ whose components $\boldsymbol{x}_i$ are intervals, that is a member of $\overline{\mathbb{IR}}^n$. Usually $\boldsymbol{x}$ is identified with the cartesian product $\boldsymbol{x}_1 \times \ldots \times \boldsymbol{x}_n$ of its components, a subset of $\mathbb{R}^n$. In particular $x \in \boldsymbol{x}$, for $x \in \mathbb{R}^n$, means $x_i \in \boldsymbol{x}_i$ for all $i = 1, \ldots, n$; and $\boldsymbol{x}$ is empty if (and only if) any of its components $\boldsymbol{x}_i$ is empty.

5.3. Hull. The (interval) **hull** of an arbitrary subset $\boldsymbol{s}$ of $\mathbb{R}^n$, written hull($\boldsymbol{s}$), is the tightest member of $\overline{\mathbb{IR}}^n$ that contains $\boldsymbol{s}$. (The **tightest** set with a given property is the intersection of all sets having that property, provided the intersection itself has this property.)

5.4. Functions.

5.4.1. *Function terminology.* In this standard, operations are written as named functions; in a specific implementation they might be represented by operators (i.e., using some form of infix notation), or by families of type-specific functions, or by operators or functions whose names might differ from those in this standard.

The terms operation, function and mapping are broadly synonymous. The following summarizes the usage in this standard, with references in parentheses to precise definitions of terms.

- A *point function* (§5.4.2) is a mathematical real function of real variables. Otherwise, *function* is usually used with its general mathematical meaning.
- A (point) *arithmetic operation* (§5.4.2) is a mathematical real function for which an implementation provides versions in the implementation's *library* (§5.4.2).
- A *version* of a point function f means a function derived from f, such as an interval extension (§5.4.3) of it; usually applied to library functions.
- An *interval arithmetic operation* is an interval version of a point arithmetic operation (§5.4.3).
- An *interval non-arithmetic operation* is an interval-to-interval library function that is not an interval arithmetic operation (§5.4.3).
- A *constructor* is a function that creates an interval from non-interval data (§6.6.4).

[3] *CK* Decorations are not part of the motion

[4] The overline is for compatibility with older notation, which uses $\mathbb{IR}$ for the set of closed and *bounded* real intervals.

5.4.2. *Point functions.* A **point function** is a (possibly partial) multivariate real function: that is, a mapping f from a subset D of $\mathbb{R}^n$ to $\mathbb{R}^m$ for some integers $n \geq 0, m > 0$. The function is called a *scalar* function if $m = 1$, otherwise a *vector* function. When not otherwise specified, scalar is assumed. The set D where f is defined is its **domain**, also written Domain f. To specify n, call f an n-variable point function, or denote values of f as

$$f(x_1, \ldots, x_n). \tag{1}$$

The **range** of f over an arbitrary subset $\boldsymbol{s}$ of $\mathbb{R}^n$ is the set

$$\text{Range}(f \mid \boldsymbol{s}) := \{\, f(x) \mid x \in \boldsymbol{s} \text{ and } x \in \text{Domain}\, f \,\}. \tag{2}$$

Thus mathematically, when evaluating a function over a set, points outside the domain are ignored (e.g., $\text{Range}(\text{sqrt} \mid [-1, 1]) = [0, 1]$).

Equivalently, for the case where f takes separate arguments $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$, each being a subset of $\mathbb{R}$, the range is written as $\text{Range}(f \mid \boldsymbol{s}_1, \ldots, \boldsymbol{s}_n)$. This is an alternative notation when $\boldsymbol{s}$ is the cartesian product of the $\boldsymbol{s}_i$.

A (point) **arithmetic operation** is a function for which an implementation provides versions in a collection of user-available operations called its **library**. This includes functions normally written in operator form (e.g., $+$, $\times$) and those normally written in function form (e.g., exp, arctan). It is not specified how an implementation provides library facilities.

5.4.3. *Interval-valued functions.* A box is an interval vector $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \overline{\mathbb{IR}}^n$. It is usually identified with the cartesian product $\boldsymbol{x}_1 \times \ldots \times \boldsymbol{x}_n \subseteq \mathbb{R}^n$; however, the correspondence is one-to-one only when all the $\boldsymbol{x}_j$ are nonempty.

Given an n-variable scalar point function f, an **interval extension** of f is a (total) mapping $\boldsymbol{f}$ from n-dimensional boxes to intervals, that is $\boldsymbol{f} : \overline{\mathbb{IR}}^n \to \overline{\mathbb{IR}}$, such that $f(x) \in \boldsymbol{f}(\boldsymbol{x})$ whenever $x \in \boldsymbol{x}$ and $f(x)$ is defined, equivalently

$$\boldsymbol{f}(\boldsymbol{x}) \supseteq \text{Range}(f \mid \boldsymbol{x})$$

for any box $\boldsymbol{x} \in \overline{\mathbb{IR}}^n$, regarded as a subset of $\mathbb{R}^n$. The **natural interval extension** of f is defined by

$$\boldsymbol{f}(\boldsymbol{x}) := \text{hull}(\text{Range}(f \mid \boldsymbol{x})).$$

Equivalently, using multiple-argument notation for f, an interval extension satisfies

$$\boldsymbol{f}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \supseteq \text{Range}(f \mid \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n),$$

and the natural interval extension is defined by

$$\boldsymbol{f}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) := \text{hull}(\text{Range}(f \mid \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n))$$

for any intervals $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.

In some contexts it is useful for $\boldsymbol{x}$ to be a general subset of $\mathbb{R}^n$, or the $\boldsymbol{x}_i$ to be general subsets of $\mathbb{R}$; the definition is unchanged.

! The natural extension is automatically defined for all interval or set arguments. (The decoration system introduced below in §5.8 gives a systematic way of diagnosing when the underlying point function has been evaluated outside its domain.)$\rightarrow$*delete*[5]

When f is a binary operator $\bullet$ written in infix notation, this gives the usual definition of its natural interval extension as

$$\boldsymbol{x} \bullet \boldsymbol{y} = \text{hull}(\{\, x \bullet y \mid x \in \boldsymbol{x},\ y \in \boldsymbol{y}, \text{ and } x \bullet y \text{ is defined} \,\}).$$

[*Example. With these definitions, the relevant natural interval extensions satisfy* $\sqrt{[-1, 4]} = [0, 2]$ *and* $\sqrt{[-2, -1]} = \emptyset$*; also* $\boldsymbol{x} \times [0, 0] = [0, 0]$ *for any nonempty* $\boldsymbol{x}$*, and* $\boldsymbol{x}/[0, 0] = \emptyset$*, for any* $\boldsymbol{x}$.]

When f is a vector point function, a vector interval function with the same number of inputs and outputs as f is called an interval extension of f if each of its components is an interval extension of the corresponding component of f.

An interval-valued function in the library is called an **interval arithmetic operation** if it is an interval extension of a point arithmetic operation, and an **interval non-arithmetic operation** otherwise. Examples of the latter are interval intersection and union, $(\boldsymbol{x}, \boldsymbol{y}) \mapsto \boldsymbol{x} \cap \boldsymbol{y}$ and $(\boldsymbol{x}, \boldsymbol{y}) \mapsto \text{hull}(\boldsymbol{x} \cup \boldsymbol{y})$.

[5] *CK* Not part of the motion.

5.4.4. *Constants.* A real scalar function with no arguments—a mapping $\mathbb{R}^n \to \mathbb{R}^m$ with $n = 0$ and $m = 1$—is a **real constant**. Languages may distinguish between a literal constant (e.g., the decimal value defined by the string `1.23e4`) and a named constant (e.g., π) but the difference is not relevant on Level 1 (and easily handled by outward rounding on Level 2).

From the definition, an interval extension of a real constant is any zero-argument interval function that returns an interval containing c. The *natural extension* returns the interval $[c, c]$.

5.5. Expressions and the functions they define.

5.5.1. *Expressions.* A variable is a symbolic name. A scalar **expression** in zero or more (independent) variables is a symbolic object defined to be either one of those variables or, recursively, of the form $\phi(\mathsf{g}_1, \ldots, \mathsf{g}_k)$ where ϕ is a symbolic arithmetic or non-arithmetic *operation* that takes k arguments ($k \geq 0$) and returns a single result, and the g_i are expressions. Other syntax may be used, such as traditional algebraic notation and/or program pseudo-code. An **arithmetic expression** is one in which all the operations are arithmetic operations.

A vector-valued expression is regarded, for purposes of definition in this standard, as a tuple of scalar expressions (see second example below).

[*Examples.*

– *The object* f *where*

$$\mathsf{f} = (e^x + e^{-x})/2y$$

is an arithmetic expression in two variables x, y *that may be written in the above form as*

$$\mathrm{div}(\mathrm{plus}(\exp(x), \exp(\mathrm{uminus}(x))), \mathrm{times}(2(), y)),$$

where uminus, plus, times, div *and* exp *name the arithmetic operations "unary minus",* $+$, $\times$, $\div$ *and exponential function. The literal constant 2 is regarded as a zero-argument arithmetic operation* $2()$, *see* §*5.4.4.*

The expression may be split (e.g., by a compiler) into simple assignments each involving a single operation, that may be sequenced in several ways. In the example above, one of these is

$$\boxed{\begin{aligned} v_1 &= \exp(x) \\ v_2 &= \mathrm{uminus}(x) \\ v_3 &= \exp(v_2) \\ v_4 &= \mathrm{plus}(v_1, v_3) \\ v_5 &= 2() \\ v_6 &= \mathrm{times}(v_5, y) \\ f &= \mathrm{div}(v_4, v_6). \end{aligned}}$$

All these forms are regarded as defining the same expression f.

– *A vector expression that returns the two roots of* $ax^2+bx+c = 0$ *is regarded for definitional purposes as the two separate expressions* $(-b-\sqrt{b^2 - 4ac})/2a$ *and* $(-b+\sqrt{b^2 - 4ac})/2a$, *although in practical code it would be simplified so that common subexpressions are only evaluated once.*

– *An expression that uses the interval intersection or union operation is a non-arithmetic expression.*

]

To define functions of variables, an expression must be made into a **bound expression** by giving it a *formal argument list* with notation such as

$$f(z_1, \ldots, z_n) = \text{expression}.$$

This defines f to be the indicated expression with the formal argument list $z_1, \ldots, z_n$, which must include at least the variables that actually occur in the right-hand side. An expression without such an argument list is a **free expression**.

5.5.2. *Generic functions.* An arithmetic operation name such as $+$, or a bound arithmetic expression such as $f(x, y) = x + \sin y$, may be used to refer to different, related, functions: such operations and expressions, or the resulting functions, are called generic (or polymorphic). Whether generic function syntax can be used in program code is language-defined.

An implementation provides a number of arithmetic operation names ϕ, each representing various functions as follows.

(a) The point function of ϕ. This is unique, theoretical and generally non-computable in finite precision.

(b) Various **interval versions** of ϕ, among them in particular
 (b1) The natural interval extension of the point function. This also is unique, theoretical and generally non-computable.
! (b2) Computable interval extensions of the point function.
(c) (Decorated interval versions of ϕ, see §5.8.)→*delete*[6]

The implementation's library contains all computable versions of all provided arithmetic operations: see §5.6 for those that are required, and §5.7 for those recommended.

An arithmetic operation or expression may also denote a floating point function; this is not relevant to this standard.

5.5.3. *Point functions and interval versions of expressions.* A bound arithmetic expression
! $\mathsf{f} = \mathsf{f}(z_1, \ldots, z_n)$ represents various functions in the categories (a), (b1), (b2)(, (c))→*delete*[7] above.

The **point function** of (or defined by) f is a function $f : \operatorname{Domain} f \subseteq \mathbb{R}^n \to \mathbb{R}$. The **natural domain** $\operatorname{Domain} f$ of f is the set of all $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ where its value is defined according to the following rules:

– If f is the variable z_i, $f(x)$ is the number x_i. It is defined for all $x \in \mathbb{R}^n$.
– Recursively, if $\mathsf{f} = \phi(\mathsf{g}_1, \ldots, \mathsf{g}_k)$, $f(x)$ is the value of the point function of ϕ at $u = (u_1, \ldots, u_k)$ where u_i is the value of the point function of g_i at x. It is defined for those $x \in \mathbb{R}^n$ such that each of $u_1, \ldots, u_k$ is defined and the resulting u is in ϕ's domain $\operatorname{Domain}\phi$, which is part of its mathematical definition.

 In the recursive clause of this definition, ϕ can be a constant (a function with $k = 0$ arguments), so that $f(x) = \phi()$. Then, see §5.4.4, there are two cases: (a) ϕ has a real value c; then $f(x) = c$ for all $x \in \mathbb{R}^n$; (b) ϕ is the undefined function NaN, in which case $f(x)$ is not defined for any $x \in \mathbb{R}^n$.

[*Example. The specifications*

$$f(x, y) = (e^x + e^{-x})/(2y),$$
$$f(y, x) = (e^x + e^{-x})/(2y),$$
$$f(w, x, y, z) = (e^x + e^{-x})/(2y)$$

are all valid and different (they define different functions), while

$$f(y) = (e^x + e^{-x})/(2y)$$

is invalid since the argument list does not include the variable x, which occurs in f.]

[*Example. The natural domain of $f(x, y) = 1/(\sqrt{x-1} - y)$ is the set of (x, y) in the plane that do not cause either square root of a negative number or division by zero, i.e., where $x \geq 1$ and $y \neq \sqrt{x-1}$.*]

An **interval version** of $\mathsf{f}(z_1, \ldots, z_n)$ is a (total) function $f : \overline{\mathbb{IR}}^n \to \overline{\mathbb{IR}}$. Its value at an actual argument $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \overline{\mathbb{IR}}^n$, a box, is recursively constrained as follows:

– If f is the variable z_i, the value is some interval containing $\boldsymbol{x}_i$.
– If $\mathsf{f} = \phi(\mathsf{g}_1, \ldots, \mathsf{g}_k)$, the value is some interval extension of ϕ evaluated at $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k)$ where $\boldsymbol{u}_i$ is the value of some interval version of g_i at $\boldsymbol{x}$.

The **natural** interval version of f is the unique interval version such that when f is z_i, the value equals $\boldsymbol{x}_i$, and that uses the natural interval extension of each ϕ.

Moore's theorem states:

Theorem 5.1 (Fundamental Theorem of Interval Arithmetic, FTIA).
(i) Every interval version of an arithmetic expression $\mathsf{f}(z_1, \ldots, z_n)$ is an interval extension of the point function defined by the expression.
(ii) If every variable occurs at most once in f and, for some $\boldsymbol{x} \in \overline{\mathbb{IR}}^n$, all elementary operations occurring in f evaluate to their range, then $f(\boldsymbol{x}) = \operatorname{Range}(f \mid \boldsymbol{x})$.

[*Note. To part (ii). Using exact arithmetic, that is, the natural interval version of* f*, it is known that an*
! *elementary operation evaluates to its range if it is continuous on its input box. (This can be checked by the decoration system (§5.8), so the conditions of (ii) are computable. The result is that in finite*

[6] *CK* 2011-10-15 Not part of the motion
[7] *CK* 2011-10-15 Not part of the motion.

precision, it is often verifiable at run time that $f(\boldsymbol{x})$ *equals* Range$(f \mid \boldsymbol{x})$ *up to roundoff error.*)$\rightarrow$delete[8]
]

[8] *CK* 2011-10-15 Not part of the motion.

5.6. Required operations.

For the functions listed in this subclause, an implementation shall provide interval versions appropriate to its supported interval types. For the forward and reverse arithmetic operations in §5.6.1, 5.6.2, 5.6.3, each interval version shall be an interval extension of the corresponding point function. The required rounding behavior of these, and of the numeric functions of intervals in §5.6.6, is detailed in §6.5, 6.6.

5.6.1. *Forward-mode elementary functions.*

Table 2 on page 17 lists required arithmetic operations. The term *operation* includes functions normally written in function notation $f(x, y, \ldots)$, as well as those normally written in binary operator notation $x \bullet y$. The names of operations may not correspond to those that any particular language would use.

[*Note. The list includes all general-computational operations in 754§5.4 except* **convertFromInt**, *and some recommended functions in 754§9.2.*]

Proving the correctness of interval computations relies on the Fundamental Theorem of Interval Arithmetic, which in turn relies on the relation between a point-function and its interval versions. Thus the domain of each point-function and its value at each point of the domain are specified to aid a rigorous implementation. This is mostly straightforward but needs care for functions with discontinuities, such as pow() and atan2().

⚠ We probably should have a version of interval **division** $\boldsymbol{x}/\boldsymbol{y}$ that returns two intervals so that it doesn't lose information when 0 is an interior point of $\boldsymbol{y}$. Several solutions exist, e.g. Kulisch's; Kearfott's, which is similar; and Vienna proposal's "division with gap". A motion please!

5.6.2. *Case expressions and case function.*

Functions are often defined by conditionals: $f(x)$ equals $g(x)$ if some condition on x holds, and $h(x)$ otherwise. To handle interval extensions of such functions in a way that automatically conforms to the Fundamental Theorem of Interval Arithmetic, the ternary function $\mathtt{case}(c, g, h)$ is provided. To simplify defining its interval extension, the argument c specifying the condition is real (instead of boolean), and the condition means $c < 0$ by definition. That is,

$$\mathtt{case}(c, g, h) = \begin{cases} g & \text{if } c < 0, \\ h & \text{otherwise.} \end{cases}$$

Its natural interval extension is

$$\mathtt{case}(\boldsymbol{c}, \boldsymbol{g}, \boldsymbol{h}) = \begin{cases} \emptyset & \text{if } \boldsymbol{c} \text{ is empty} \\ \boldsymbol{g} & \text{elseif } \overline{c} < 0 \\ \boldsymbol{h} & \text{elseif } \underline{c} \geq 0 \\ \text{hull}(\boldsymbol{g} \cup \boldsymbol{h}) & \text{otherwise.} \end{cases} \tag{3}$$

for any intervals $\boldsymbol{c}, \boldsymbol{g}, \boldsymbol{h}$, where $\boldsymbol{c} = [\underline{c}, \overline{c}]$ when nonempty.

The function f above may be encoded as $f(x) = \mathtt{case}\big(c(x), g(x), h(x)\big)$. Then, if $\boldsymbol{c}$, $\boldsymbol{g}$, $\boldsymbol{h}$ are interval extensions of c, g and h, the function

$$\boldsymbol{f}(\boldsymbol{x}) = \mathtt{case}\big(\boldsymbol{c}(\boldsymbol{x}), \boldsymbol{g}(\boldsymbol{x}), \boldsymbol{h}(\boldsymbol{x})\big) \tag{4}$$

is automatically an interval extension of f.

[*Notes.*

1. *This method is less awkward than using interval comparisons as a mechanism for handling such functions. However, the resulting interval function is usually not the tightest extension of the corresponding point function. E.g., the absolute value $f(x) = |x|$ may be defined by*

$$f(x) = \mathtt{case}(x, -x, x).$$

   Then it is easy to see that formula (4), applied to a nonempty $\boldsymbol{x} = [\underline{x}, \overline{x}]$, gives the exact range $\{\,|x| \mid x \in \boldsymbol{x}\,\}$ when $\overline{x} < 0$ or $0 \leq \underline{x}$, but the poor enclosure $(-\boldsymbol{x}) \cup \boldsymbol{x}$ when $\underline{x} < 0 \leq \overline{x}$.
2. $\mathtt{case}(c, g, h)$ *is equivalent to the C expression $(c < 0 \,?\, g : h)$.*
3. *Compound conditions may be expressed using the* `max` *and* `min` *operations: e.g., a real function $f(x, y)$ that equals $\sin(xy)$ in the positive quadrant of the plane, and zero elsewhere, may be written*

$$f(x, y) = \mathtt{case}(\min(x, y), 0, \sin(xy)),$$

   since $\min(x, y) < 0$ is equivalent to ($x < 0$ or $y < 0$).

TABLE 2. Required forward elementary functions.

Name	Definition	Point function domain	Point function range	Note
`add`(x, y)	$x + y$	$\mathbb{R}^2$	$\mathbb{R}$	
`sub`(x, y)	$x - y$	$\mathbb{R}^2$	$\mathbb{R}$	
`mul`(x, y)	xy	$\mathbb{R}^2$	$\mathbb{R}$	
`div`(x, y)	x/y	$\mathbb{R}^2 \setminus \{y = 0\}$	$\mathbb{R}$	a
`Rev`(x)	$1/x$	$\mathbb{R} \setminus \{0\}$	$\mathbb{R} \setminus \{0\}$	
`sqrt`(x)	$\sqrt{x}$	$[0, \infty)$	$[0, \infty)$	
`hypot`(x, y)	$\sqrt{x^2 + y^2}$	$\mathbb{R}^2$	$[0, \infty)$	
`case`(b, g, h)		See §5.6.2.		
`sqr`(x)	x^2	$\mathbb{R}$	$[0, \infty)$	
`pown`(x, p)	x^p, $p \in \mathbb{Z}$	$\begin{cases}\mathbb{R} \text{ if } p \geq 0 \\ \mathbb{R}\backslash\{0\} \text{ if } p < 0\end{cases}$	$\begin{cases}\mathbb{R} \text{ if } p > 0 \text{ odd} \\ [0, \infty) \text{ if } p > 0 \text{ even} \\ \{1\} \text{ if } p = 0 \\ \mathbb{R}\backslash\{0\} \text{ if } p < 0 \text{ odd} \\ (0, \infty) \text{ if } p < 0 \text{ even}\end{cases}$	b
`pow`(x, y)	x^y	$\{x{>}0\} \cup \{x{=}0, y{>}0\}$	$[0, \infty)$	c, a
`exp,exp2,exp10`(x)	b^x	$\mathbb{R}$	$(0, \infty)$	d
`log,log2,log10`(x)	$\log_b x$	$(0, \infty)$	$\mathbb{R}$	d
`sin`(x)		$\mathbb{R}$	$[-1, 1]$	
`cos`(x)		$\mathbb{R}$	$[-1, 1]$	
`tan`(x)		$\mathbb{R}\backslash\{(k + \frac{1}{2})\pi \mid k \in \mathbb{Z}\}$	$\mathbb{R}$	
`asin`(x)		$[-1, 1]$	$[-\pi/2, \pi/2]$	e
`acos`(x)		$[-1, 1]$	$[0, \pi]$	e
`atan`(x)		$\mathbb{R}$	$(-\pi/2, \pi/2)$	e
`atan2`(y, x)		$\mathbb{R}^2 \setminus \{(0, 0)\}$	$(-\pi, \pi]$	f, e
`sinh`(x)		$\mathbb{R}$	$\mathbb{R}$	
`cosh`(x)		$\mathbb{R}$	$[1, \infty)$	
`tanh`(x)		$\mathbb{R}$	$(-1, 1)$	
`asinh`(x)		$\mathbb{R}$	$\mathbb{R}$	
`acosh`(x)		$[1, \infty)$	$[0, \infty)$	
`atanh`(x)		$(-1, 1)$	$\mathbb{R}$	
`sign`(x)		$\mathbb{R}$	$\{-1, 0, 1\}$	g
`ceil`(x)		$\mathbb{R}$	$\mathbb{Z}$	h
`floor`(x)		$\mathbb{R}$	$\mathbb{Z}$	h
`round`(x)		$\mathbb{R}$	$\mathbb{Z}$	h
`trunc`(x)		$\mathbb{R}$	$\mathbb{Z}$	h
`abs`(x)	$\lvert x \rvert$	$\mathbb{R}$	$[0, \infty)$	
`min`$(x_1, \ldots, x_k)$		$\mathbb{R}^k$ for $k = 2, 3, \ldots$	$\mathbb{R}$	i
`max`$(x_1, \ldots, x_k)$		$\mathbb{R}^k$ for $k = 2, 3, \ldots$	$\mathbb{R}$	i

Notes to Table 2

a. In describing the domain, notation such as $\{y = 0\}$ is short for $\{\, (x, y) \in \mathbb{R}^2 \mid y = 0 \,\}$, and so on.
b. Regarded as a family of functions parameterized by the integer argument p.
c. Defined as $e^{y \ln x}$ for real $x > 0$ and all real y, and 0 for $x = 0$ and $y > 0$, else undefined.
d. $b = e, 2$ or 10, respectively.
e. The ranges shown are the mathematical range of the point function. To ensure containment, an interval result may include values just outside the mathematical range.
f. atan2(y, x) is the principal value of the argument (polar angle) of (x, y) in the plane.
g. `sign`(x) is -1 if $x < 0$; 0 if $x = 0$; and 1 if $x > 0$.
h. `ceil`(x) is the smallest integer $\geq x$. `floor`(x) is the largest integer $\leq x$. `round`(x) is the nearest integer to x, rounding halfway cases away from zero. `trunc`(x) is the nearest integer to x in the direction of zero. (As defined in the C standard §7.12.9.)
i. Smallest, or largest, of its real arguments. Regarded as a family of functions parameterized by the arity k.

Table 3. Required reverse elementary functions.

Unary	Binary
`sqrRev`(x)	`mulRev`(x, y)
`invRev`(x)	`divRev`(x, y)
`absRev`(x)	`powRev`(x, y)
`pownRev`(x, p)	`atan2Rev`(x, y)
`sinRev`(x)	
`cosRev`(x)	
`tanRev`(x)	
`coshRev`(x)	

]

5.6.3. *Reverse-mode elementary functions.*

Constraint-satisfaction algorithms use the functions in this subclause for iteratively tightening an enclosure of a solution to a system of equations.

Given a unary arithmetic operation ϕ, a **reverse interval extension** of ϕ is a binary interval function ϕRev such that

$$\phi\mathrm{Rev}(\boldsymbol{z}, \boldsymbol{x}) \supseteq \{\, x \in \boldsymbol{x} \mid \phi(x) \text{ is defined and in } \boldsymbol{z} \,\}, \tag{5}$$

for any intervals $\boldsymbol{z}, \boldsymbol{x}$.

Similarly, a binary arithmetic operation $\bullet$ has two forms of reverse interval extension, which are ternary interval functions $\bullet\mathrm{Rev}_1$ and $\bullet\mathrm{Rev}_2$ such that

$$\bullet\mathrm{Rev}_1(\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{x}) \supseteq \{\, x \in \boldsymbol{x} \mid b \in \boldsymbol{b} \text{ exists such that } x \bullet b \text{ is defined and in } \boldsymbol{c} \,\}, \tag{6}$$

$$\bullet\mathrm{Rev}_2(\boldsymbol{a}, \boldsymbol{c}, \boldsymbol{x}) \supseteq \{\, x \in \boldsymbol{x} \mid a \in \boldsymbol{a} \text{ exists such that } a \bullet x \text{ is defined and in } \boldsymbol{c} \,\}. \tag{7}$$

If $\bullet$ is commutative then $\bullet\mathrm{Rev}_1$ and $\bullet\mathrm{Rev}_2$ agree and may be implemented simply as $\bullet$Rev.

In each of (5, 6, 7), the unique **natural reverse interval extension** is the one whose value is the interval hull of the right-hand side. Clearly, any reverse interval extension encloses this hull.

The last argument $\boldsymbol{x}$ in each of (5, 6, 7) is optional, with default $\boldsymbol{x} = \mathbb{R}$ if absent.

[*Note. The argument $\boldsymbol{x}$ can be thought of as giving prior knowledge about the range of values taken by a point-variable x, which is then sharpened by applying the reverse function: see the example below.*]

Reverse operations shall be provided as in Table 3. Note `pownRev`(x, p) is regarded as a family of unary functions parametrized by p.

[*Example.*

– *Consider the function $sqr(x) = x^2$. Evaluating $sqr\mathrm{Rev}([1, 4])$ answers the question: given that $1 \le x^2 \le 4$, what interval can we restrict x to? Using the natural reverse extension, we have*

$$sqr\mathrm{Rev}([1, 4]) = \mathsf{hull}\{\, x \in \mathbb{R} \mid x^2 \in [1, 4] \,\} = \mathsf{hull}([-2, -1] \cup [1, 2]) = [-2, 2].$$

– *If we can add the prior knowledge that $x \in \boldsymbol{x} = [0, 1.2]$, then using the optional second argument gives the tighter enclosure*

$$sqr\mathrm{Rev}([1, 4], [0, 1.2]) = \mathsf{hull}\{\, x \in [0, 1.2] \mid x^2 \in [1, 4] \,\} = \mathsf{hull}\big([0, 1.2] \cap ([-2, -1] \cup [1, 2])\big) = [1, 1.2].$$

– *One might think it suffices to apply the operation without the optional argument and intersect the result with $\boldsymbol{x}$. This is less effective because "hull" and "intersect" do not commute. E.,g., in the above, this method evaluates*

$$sqr\mathrm{Rev}([1, 4]) \cap \boldsymbol{x} = [-2, 2] \cap [0, 1.2] = [0, 1.2],$$

so no tightening of the enclosure $\boldsymbol{x}$ is obtained.

]

5.6.4. *Non-arithmetic operations.*

The following operations shall be provided, the arguments and result being intervals.

Name	Definition
`intersection`$(\boldsymbol{x}, \boldsymbol{y})$	$\boldsymbol{x} \cap \boldsymbol{y}$
`convexHull`$(\boldsymbol{x}, \boldsymbol{y})$	interval hull of $\boldsymbol{x} \cup \boldsymbol{y}$

TABLE 4. Required numeric functions of an interval $\boldsymbol{x} = [\underline{x}, \overline{x}]$.

Name	Definition
`inf`$(\boldsymbol{x})$	$\begin{cases}\text{lower bound of } \boldsymbol{x} \text{ if } \boldsymbol{x} \text{ is nonempty}\\ \text{NaN if } \boldsymbol{x} \text{ is empty}\end{cases}$
`sup`$(\boldsymbol{x})$	$\begin{cases}\text{upper bound of } \boldsymbol{x} \text{ if } \boldsymbol{x} \text{ is nonempty}\\ \text{NaN if } \boldsymbol{x} \text{ is empty}\end{cases}$
`mid`$(\boldsymbol{x})$	$\begin{cases}\text{midpoint } (\underline{x}+\overline{x})/2 \text{ if } \boldsymbol{x} \text{ is nonempty bounded}\\ \text{absolutely smallest member of } \boldsymbol{x} \text{ if } \boldsymbol{x} \text{ is unbounded}\\ \text{NaN if } \boldsymbol{x} \text{ is empty}\end{cases}$
`diam`$(\boldsymbol{x})$	$\begin{cases}\text{diameter } \overline{x}-\underline{x} \text{ if } \boldsymbol{x} \text{ is nonempty}\\ \text{NaN if } \boldsymbol{x} \text{ is empty}\end{cases}$
`rad`$(\boldsymbol{x})$	$\begin{cases}\text{radius } (\overline{x}-\underline{x})/2 \text{ if } \boldsymbol{x} \text{ is nonempty}\\ \text{NaN if } \boldsymbol{x} \text{ is empty}\end{cases}$
`midRad`$(\boldsymbol{x})$	returns the pair (mid$(\boldsymbol{x})$, rad$(\boldsymbol{x})$)
`mag`$(\boldsymbol{x})$	$\begin{cases}\text{magnitude } \sup\{\,\lvert x\rvert \mid x \in \boldsymbol{x}\,\} \text{ if } \boldsymbol{x} \text{ is nonempty}\\ \text{NaN if } \boldsymbol{x} \text{ is empty}\end{cases}$
`mig`$(\boldsymbol{x})$	$\begin{cases}\text{mignitude } \inf\{\,\lvert x\rvert \mid x \in \boldsymbol{x}\,\} \text{ if } \boldsymbol{x} \text{ is nonempty}\\ \text{NaN if } \boldsymbol{x} \text{ is empty}\end{cases}$

⚠ The "absolutely smallest member" part of the mid() definition is from the Vienna proposal. George Corliss is skeptical. So am I on KISS grounds, and would prefer NaN in this case. George also questions whether infinite values of inf(), rad(), etc. should be supported. I do not: I think this is exactly the sort of thing the extended reals were invented for.
⚠ It may be decided that NaN is not a desirable Level 1 concept, in which case the functions could be made "undefined" instead of taking the value NaN.

5.6.5. *Constructors.*
⚠ This is provisional since we have not had a motion on them at the time of writing. The text roughly follows the Vienna proposal.

The following operations shall be provided, that create an interval from non-interval data.

The operation `nums2interval`(l, u), where l and u are extended-real values, returns the set $\{\, x \in \mathbb{R} \mid l \le x \le u \,\}$. If (see §5.2) the conditions $l \le u$, $l < +\infty$ and $u > -\infty$ hold, this set is the nonempty interval $[l, u]$ and the operation is said to *succeed*. Otherwise the operation is said to *fail*, and returns Empty.

Success and failure are used in specifying how decorations are set by the corresponding constructors of decorated intervals at Level 2.

The operation `num2interval`(x) is equivalent to `nums2interval`(x, x). It fails if and only if x is infinite.

The operation `text2interval`(t) succeeds and returns the interval denoted by the text string t, if t denotes an interval. Otherwise, it fails and returns Empty.
[*Note. Since Level 1 is mainly for human-human communication, any understandable t is acceptable, e.g.* "`[3.1,4.2]`" *or* "`[`2π`,`∞`]`"*. Rules for the strings t accepted at an implementation level are given in the Level 2 Subclause 6.11 on I/O and may optionally be followed.*]

5.6.6. *Numeric functions of intervals.*
The operations in Table 4 shall be provided, the argument being an interval and the result a number, which for some of the operations may be infinite.
⚠ The Vienna proposal has other operations: smallest$(\boldsymbol{x})$, zerolength$(\boldsymbol{x})$, sign2$(\boldsymbol{x})$, signedDistance$(x, \boldsymbol{y}$ distance$(\boldsymbol{x}, \boldsymbol{y})$, innerDistance$(\boldsymbol{x}, \boldsymbol{y})$. Will someone speak up for the usefulness—or not—of these?

5.6.7. *Boolean functions of intervals.*
⚠ This includes interval comparisons, together with `areDisjoint`$(\boldsymbol{x}, \boldsymbol{y})$, `containedIn`$(\boldsymbol{x}, \boldsymbol{y})$ and `containedInInterior`$(\boldsymbol{x}, \boldsymbol{y})$. Possibly some others. Details still to be finalized by the group.

5.6.8. *Dot product function.*
The function `dotProduct`(x, y) is provided, where x and y are real vectors with the same number of elements. It returns $\sum_i x_i y_i$.

TABLE 5. Recommended elementary functions.

Name	Definition	Point function domain	Point function range	Note
`rootn`(x,q)	real $\sqrt[q]{x}$, $q \in \mathbb{Z} \setminus \{0\}$	$\begin{cases} \mathbb{R} \text{ if } q>0 \text{ odd} \\ [0,\infty) \text{ if } q>0 \text{ even} \\ \mathbb{R}\backslash\{0\} \text{ if } q<0 \text{ odd} \\ (0,\infty) \text{ if } q<0 \text{ even} \end{cases}$	same as domain	a
`powr`(x,p,q)	real $x^{p/q}$	$\begin{cases} \mathbb{R} \text{ if } r \ge 0,\ s \text{ odd} \\ [0,\infty) \text{ if } r>0,\ s \text{ even} \\ \mathbb{R}\backslash\{0\} \text{ if } r<0,\ s \text{ odd} \\ (0,\infty) \text{ if } r<0,\ s \text{ even} \end{cases}$	$\begin{cases} \mathbb{R} \text{ if } r>0,\ rs \text{ odd} \\ [0,\infty) \text{ if } r>0,\ rs \text{ even} \\ \{1\} \text{ if } r=0 \\ \mathbb{R}\backslash\{0\} \text{ if } r<0,\ rs \text{ odd} \\ (0,\infty) \text{ if } r<0,\ rs \text{ even} \end{cases}$	a
	$p \in \mathbb{Z}$, $q \in \mathbb{Z} \setminus \{0\}$;	r/s is p/q in lowest terms, see Note c.		
$\begin{cases} \texttt{expm1}(x) \\ \texttt{exp2m1}(x) \\ \texttt{exp10m1}(x) \end{cases}$	$b^x - 1$	$\mathbb{R}$	$(-1,\infty)$	b, d
$\begin{cases} \texttt{logp1}(x) \\ \texttt{log2p1}(x) \\ \texttt{log10p1}(x) \end{cases}$	$\log_b(x+1)$	$(-1,\infty)$	$\mathbb{R}$	b, d
`compoundm1`(x,y)	$(1+x)^y - 1$	$\{x>-1\} \cup \{x=-1, y>0\}$	$[0,\infty)$	d, e
`rSqrt`(x)	$1/\sqrt{x}$	$(0,\infty)$	$(0,\infty)$	
`sinPi`(x)	$\sin(\pi x)$	$\mathbb{R}$	$[-1,1]$	f
`cosPi`(x)	$\cos(\pi x)$	$\mathbb{R}$	$[-1,1]$	f
`tanPi`(x)	$\tan(\pi x)$	$\mathbb{R}\backslash\{k+\frac{1}{2} \mid k \in \mathbb{Z}\}$	$\mathbb{R}$	f
`asinPi`(x)	$\arcsin(x)/\pi$	$[-1,1]$	$[-1/2,1/2]$	f
`acosPi`(x)	$\arccos(x)\pi$	$[-1,1]$	$[0,1]$	f
`atanPi`(x)	$\arctan(x)/\pi$	$\mathbb{R}$	$[-1/2,1/2]$	f
`atan2Pi`(y,x)	$\text{atan2}(y,x)/\pi$	$\mathbb{R} \times \mathbb{R}$	$(-1,1]$	f

Notes to Table 5

a. Regarded as a family of functions parameterized by the integer arguments q, or r and s.
b. $b = e, 2$ or 10, respectively.
c. Defined as $(\sqrt[s]{x})^r$ where r/s is the reduced fraction equal to p/q, where integers r and s have no common factor, $s > 0$. E.g., `powr`$(x,-2,-6) = x^{(-2)/(-6)}$ reduces to $\sqrt[3]{x}$ and `powr`$(x,4,-6) = x^{4/(-6)}$ to $(\sqrt[3]{x})^{-2}$.
d. Mathematically unnecessary, but included to let implementations give better numerical behavior for small values of the arguments.
e. In describing domains, notation such as $\{y = 0\}$ is short for $\{\,(x,y) \in \mathbb{R}^2 \mid y = 0\,\}$, and so on.
f. These functions avoid a loss of accuracy due to π being irrational, cf. Table 2, note e.

This is a point-function; an interval extension of it is not *required.* The most significant part of its definition concerns its rounding properties in finite precision, which are specified in §6.6.

5.7. Recommended operations (informative).

Language standards should define interval versions of some or all functions in this subclause, and some or all supported types, as is most appropriate to the language. Each interval version provided is required to be an interval extension of the point function.

5.7.1. *Forward-mode elementary functions.*

The list of recommended functions is in Table 5.

5.7.2. *Slope functions.*

The functions in Table 6 are the commonest ones needed to efficiently implement improved range enclosures via *first- and second-order slope* algorithms. They are analytic at $x = 0$ after filling in the removable singularity there, where each has the value 1.

Table 6. Recommended slope functions.

Name	Definition	Point function domain	Point function range	Note
`expSlope1`(x)	$\frac{1}{x}(e^x - 1)$	$\mathbb{R}$	$(0, \infty)$	
`expSlope2`(x)	$\frac{2}{x^2}(e^x - 1 - x)$	$\mathbb{R}$	$(0, \infty)$	
`logSlope1`(x)	$\frac{2}{x^2}(\log(1+x) - x)$	$\mathbb{R}$	$(0, \infty)$	
`logSlope2`(x)	$\frac{3}{x^3}(\log(1+x)-x+\frac{x^2}{2})$	$\mathbb{R}$	$(0, \infty)$	
`cosSlope2`(x)	$-\frac{2}{x^2}(\cos x - 1)$	$\mathbb{R}$	$[0, 1]$	
`sinSlope3`(x)	$-\frac{6}{x^3}(\sin x - x)$	$\mathbb{R}$	$(0, 1]$	
`asinSlope3`(x)	$\frac{6}{x^3}(\arcsin x - x)$	$[-1, 1]$	$[1, 3\pi - 6]$	
`atanSlope3`(x)	$-\frac{3}{x^3}(\arctan x - x)$	$\mathbb{R}$	$(0, 1]$	
`coshSlope2`(x)	$\frac{2}{x^2}(\cosh x - 1)$	$\mathbb{R}$	$[1, \infty)$	
`sinhSlope3`(x)	$\frac{3}{x^3}(\sinh x - x)$	$\mathbb{R}$	$[\frac{1}{2}, \infty)$	