# 5 Requirements

## 5.1 General

The requirements defined in this SRS have the following associated meta-data:

- Requirement identifier: Unique identifier in the form of **REQ-BL-NNNN** where:
  - REQ stands for "Requirement",
  - BL stands for "Basic mathematical Library", and
  - NNNN represents a four digit unique number.
- Requirement justification or rationale: Trace to the reason for the existence of the requirement. This is usually one or multiple requirements from the E1356-GTD-TR-01 [RD07] , or the Statement of Work (SoW).
- Requirement validation method: T (Test), A (Analysis), I (Inspection), D (Demonstration), R (Review of design).
- A requirement may contain one or multiple remarks that are non normative. Remarks begin with *Remark:*
- All requirements end with the following marker: □
- The special cases table never belong to a single requirement but to the procedure as a whole, they are meant to better visualize all special cases of the procedure and are non normative.

In addition to the definitions and abbreviations defined in §3, this SRS follows these conventions:

- Not a Number (NaN) floating-point datums are represented with: `NaN`
- Infinite floating-point numbers are represented with: `Inf`
- Numeric values are represented in normal font: e.g. +1.0
- Mathematical expressions and constants are represented in cursive[1]: e.g. $x < 0$, $\pi$, $e$, &c.
- Programming language related elements are represented in typewriter font: e.g. `double sin(double x)`

[1]This statement is not exactly true. Mathematical expressions are represented in a combination of cursive and normal font; for example the arccosine function is represented as "$\arccos x$", where the $\arccos$ function name is represented in normal font and the $x$ function argument in cursive.

- Constants defined in a programming language are represented in typewriter font: e.g. `M_PI`, `M_E`, `FLT_MAX`, &c.
- Literal numeric values in a programming language are represented in typewriter font: e.g. `1.0f`

Thus, this SRS will define requirements for *procedures* like `sin(x)` that evaluates the mathematical *function* $\sin x$, and the mathematical transcendental constant $\pi$, which cannot be represented accurately in this text, is approximated in programming languages like C with constants such as `M_PI` (which have a finite precision value assigned like `3.14159265358979323846`).

Regarding the validation methods of the requirements we define them as follows:

- **T**est - Validated via Software Validation Specification (SVS) testing,
- **A**nalysis - Validated via manual analysis effort.
- **I**nspection - Validated via source code inspection.
- **D**emonstration - Validated via one-time display of an example execution.
- **R**eview of design - Validated via review of the Software Design Document (SDD).

## 5.2 Functional Requirements

This SRS describes the capabilities to be provided by the basic mathematical library.

**REQ-BL-0050**//GTD-TR-01-BL-0015/R

The Basic Library shall implement the following procedures of the ISO C99 standard [RD03] :

| function | description |
|---|---|
| `sin` | Procedure returning the trigonometric sine |
| `asin` | Procedure returning the trigonometric arcsine |
| `cos` | Procedure returning the trigonometric cosine |
| `acos` | Procedure returning the trigonometric arccosine |
| `tan` | Procedure returning the trigonometric tangent |
| `atan` | Procedure returning the trigonometric arctangent |
| `sqrt` | Procedure returning the square root |
| `atan2` | Procedure returning the trigonometric arctangent of $y/x$ |
| `exp` | Procedure returning the base `E` exponential of `x` |
| `pow` | Procedure returning `x` raised to the power of `y` |
| `log` | Procedure returning the natural logarithm |
| `log10` | Procedure returning the base 10 logarithm |
| `fabs` | Procedure returning the absolute value |

| function | description |
|---|---|
| `round` | Procedure for rounding to the nearest integer (Halfway values rounded away from 0) |
| `floor` | Procedure for rounding downwards to the nearest integer |
| `trunc` | Procedure for rounding towards 0 to the nearest integer |
| `ceil` | Procedure for rounding upwards to the nearest integer |
| `fmod` | Procedure returning the floating-point remainder of $x/y$ |
| `modf` | Procedure breaking x in its integral and fractional part |
| `fmin` | Procedure returning the smaller of two values |
| `fmax` | Procedure returning the larger of two values |
| `hypot` | Procedure returning the square root of $x^2 + y^2$ |
| `isfinite` | Procedure returning whether the value of the argument is finite or not (not ±`Inf` and not NaN) |
| `isinf` | Procedure returning whether the value of the argument is positive or negative Infinity or not |
| `isnan` | Procedure returning whether the argument is a not-a-number (NaN) floating-point value or not |
| `signbit` | Procedure returning whether the argument is negative or not |
| `copysign` | Procedure returning a floating-point number with the magnitude of x and the sign of y |

*Remark:* The description of these procedures are not to be understood as the requirements and specification of the procedures but as a description for identification purposes. For the exact specification of the procedure refer to the ISO C99 standard. □

**REQ-BL-0052**//GTD-TR-01-BL-0019, GTD-TR-01-BL-0020/R

The Basic Library shall implement the following additional procedures:

| function | description |
|---|---|
| `deg2rad` | Procedure returning the radians value of a provided degree value |
| `rem2pi` | Procedure returning the provided value reduced to the range $[+0, 2\pi]$ |

*Remark:* The description of these procedures are not to be understood as the requirements and specification of the procedures but as a description for identification purposes. □

**REQ-BL-0040**//GTD-TR-01-BL-0017/T

The Basic Library shall provide procedures for 32 bit (binary32) and 64 bit (binary64) precision floating-point datums.

*Remark:* This refers to the ISO C99 procedures with 32 bit floating-point arguments and return values and the procedures with 64 bit floating-point arguments and return values (e.g. `float sinf(float x)` and `double sin(double x)`). □

**REQ-BL-0051**//GTD-TR-01-BL-0018/I

The Basic Library shall provide the following constants:

| Name | description |
|---|---|
| `M_E` | Value of $e$ |
| `M_LOG2E` | Value of $log_{10}e$ |
| `M_LOG10E` | Value of $log_2e$ |
| `M_LN2` | Value of $log_e2$ |
| `M_LN10` | Value of $log_e10$ |
| `M_PI` | Value of $\pi$ |
| `M_PI_2` | Value of $\frac{\pi}{2}$ |
| `M_PI_4` | Value of $\frac{\pi}{4}$ |
| `M_1_PI` | Value of $\frac{1}{\pi}$ |
| `M_2_PI` | Value of $\frac{2}{\pi}$ |
| `M_2_SQRTPI` | Value of $\frac{2}{\sqrt{\pi}}$ |
| `M_SQRT2` | Value of $\sqrt{2}$ |
| `M_SQRT1_2` | Value of $\sqrt{\frac{1}{2}}$ |

| Name | description |
|---|---|
| `HUGE_VAL` | Value of `+Inf` (double) |
| `HUGE_VALF` | Value of `+Inf` (float) |
| `INFINITY` | Value of `+Inf` |
| `NAN` | Value of `NaN` |
| `MAXFLOAT` | Synonym of `FLT_MAX` |

*Remark:* Both `INFINITY` and `NAN` expand to floats or doubles depending on the context. □

**REQ-BL-1400**//GTD-TR-01-BL-0016/R

The Basic Library shall consider all arguments and return values that represent an angle to be in radians unless otherwise specified. □

### 5.2.1 Angle Degree to Radians Conversion

**REQ-BL-1420**//GTD-TR-01-BL-0019/T

The `deg2rad` and `deg2radf` procedures shall return the radians value equal to the argument `x` given in degrees. □

**REQ-BL-1421**//GTD-TR-01-BL-0019/T

The `deg2rad` and `deg2radf` procedures shall return `NaN` if the argument is `NaN`. □

**REQ-BL-1422**//GTD-TR-01-BL-0019/T

The `deg2rad` and `deg2radf` procedures shall return the value of the argument if the argument is ±0 or `±Inf`. □

| X | Result | Requirement |
|---|---|---|
| ±0 | X | REQ-BL-1422 |
| ±Inf | X | REQ-BL-1422 |
| NaN | NaN | REQ-BL-1421 |

Table 5.5: Special cases for deg2rad(X)

### 5.2.2 Angle Range Reduction

**REQ-BL-0150**//GTD-TR-01-BL-0020/T

The `rem2pi` and `rem2pif` procedures shall return the positive remainder of the argument `x` $n$-times divided by the value $2\pi$ such that:

$$+0 \leq x - n * 2\pi \leq 2\pi$$

*Remark:* The target range is $[+0, +2\pi]$. $n$ can be a positive or negative integer. □

**REQ-BL-0151**//GTD-TR-01-BL-0020/T

The `rem2pi` and `rem2pif` procedures shall return `NaN` if the argument is `NaN`. □

**REQ-BL-0152**//GTD-TR-01-BL-0020/T

The `rem2pi` and `rem2pif` procedures shall return the value of the argument if the argument is ±0. □

**REQ-BL-0153**//GTD-TR-01-BL-0020/T

The `rem2pi` and `rem2pif` procedures shall return `NaN` if the argument is `±Inf`. □

| X | Result | Requirement |
|---|---|---|
| ±0 | X | REQ-BL-0152 |
| ±Inf | NaN | REQ-BL-0153 |
| NaN | NaN | REQ-BL-0151 |

Table 5.6: Special cases for rem2pi(X)

### 5.2.3 SIN ($\sin x$)

**REQ-BL-0200**//GTD-TR-01-BL-0015/T

The `sin` and `sinf` procedures shall evaluate the `sine` of their argument `x` in radians. □

**REQ-BL-0203**//GTD-TR-01-BL-0015/R

The `sin` and `sinf` procedures shall use a minimax polynomial for the calculation. □

**REQ-BL-0210**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `sin` and `sinf` procedures shall return NaN if the argument is NaN. □

**REQ-BL-0220**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `sin` and `sinf` procedures shall return the value of the argument if the argument is ±0. □

**REQ-BL-0240**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `sin` and `sinf` procedures shall return NaN if `x` is ±`Inf`. □

| X | Result | Requirement |
|---|---|---|
| ±0 | X | REQ-BL-0220 |
| ±Inf | NaN | REQ-BL-0240 |
| NaN | NaN | REQ-BL-0210 |

Table 5.7: Special cases for sin(X)

### 5.2.4 ASIN ($\arcsin x$)

**REQ-BL-0250**//GTD-TR-01-BL-0015/T

The `asin` and `asinf` procedures shall evaluate the arcsine of their argument `x` in the output range $[-\frac{\pi}{2}, \frac{\pi}{2}]$ radians. □

**REQ-BL-0251**//GTD-TR-01-BL-0015/R

The `asin` and `asinf` procedures shall use a rational approximation for the calculation. □

**REQ-BL-0260**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `asin` and `asinf` procedures shall return `NaN` if the argument is `NaN`. ☐

**REQ-BL-0270**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `asin` and `asinf` procedures shall return the value of the argument if the argument is ±0. ☐

**REQ-BL-0280**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `asin` and `asinf` procedures shall return `NaN` if the argument is ±`Inf`. ☐

**REQ-BL-0281**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `asin` and `asinf` procedures shall return `NaN` if the argument is not in the range [-1, 1]. ☐

| X | Result | Requirement |
|---|---|---|
| ±0 | X | REQ-BL-0270 |
| ∉[-1, 1] | NaN | REQ-BL-0281 |
| ±Inf | NaN | REQ-BL-0250 |
| NaN | NaN | REQ-BL-0260 |

Table 5.8: Special cases for asin(X)

### 5.2.5 COS ($\cos x$)

**REQ-BL-0300**//GTD-TR-01-BL-0015/T

The `cos` and `cosf` procedures shall evaluate the `cosine` of their argument `x` in radians. ☐

**REQ-BL-0303**//GTD-TR-01-BL-0015/R

The `cos` and `cosf` procedures shall use a minimax polynomial for the calculation. ☐

**REQ-BL-0310**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `cos` and `cosf` procedures shall return `NaN` if the argument is `NaN`. ☐

**REQ-BL-0320**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `cos` and `cosf` procedures shall return `1.0` if the argument is ±0. ☐

**REQ-BL-0330**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `cos` and `cosf` procedures shall return `NaN`, if the argument is ±`Inf`. ☐

| X | Result | Requirement |
|---|---|---|
| ±0 | 1.0 | REQ-BL-0320 |
| ±Inf | NaN | REQ-BL-0330 |
| NaN | NaN | REQ-BL-0310 |

Table 5.9: Special cases for cos(X)

### 5.2.6 ACOS ($\arccos x$)

**REQ-BL-0450**//GTD-TR-01-BL-0015/T

The `acos` and `acosf` procedures shall evaluate the principal value of the arccosine of their argument `x` in the output range [0,$\pi$] radians. □

**REQ-BL-0451**//GTD-TR-01-BL-0015/R

The `acos` and `acosf` procedures shall use a rational approximation for the calculation. □

**REQ-BL-0460**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `acos` and `acosf` procedures shall return `NaN`, if the argument `x` is not in the range [-1,1]. □

**REQ-BL-0470**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `acos` and `acosf` procedures shall return `NaN` if the argument is `NaN`. □

**REQ-BL-0480**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `acos` and `acosf` procedures shall return `+0` if the argument is `+1`. □

**REQ-BL-0490**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `acos` and `acosf` procedures shall return `NaN` if the argument `x` is `±Inf`. □

| X | Result | Requirement |
|---|---|---|
| +1 | +0 | REQ-BL-0480 |
| $\notin$[-1, 1] | NaN | REQ-BL-0460 |
| ±Inf | NaN | REQ-BL-0490 |
| NaN | NaN | REQ-BL-0470 |

Table 5.10: Special cases for acos(X)

### 5.2.7 TAN ($\tan x$)

**REQ-BL-0500**//GTD-TR-01-BL-0015/T

The `tan` and `tanf` procedures shall evaluate the tangent of their argument x in radians. ☐

**REQ-BL-0503**//GTD-TR-01-BL-0015/R

The `tan` and `tanf` procedures shall use a minimax polynomial for the calculation. ☐

**REQ-BL-0520**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `tan` and `tanf` procedures shall return NaN if the argument is NaN. ☐

**REQ-BL-0530**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `tan` and `tanf` procedures shall return the value of the argument if the argument is ±0. ☐

**REQ-BL-0550**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `tan` and `tanf` procedures shall return NaN, if the argument is ±`Inf`. ☐

| X | Result | Requirement |
|---|---|---|
| ±0 | X | REQ-BL-0530 |
| ±Inf | NaN | REQ-BL-0550 |
| NaN | NaN | REQ-BL-0520 |

Table 5.11: Special cases for tan(X)

### 5.2.8 ATAN ($\arctan x$)

**REQ-BL-0600**//GTD-TR-01-BL-0015/T

The `atan` and `atanf` procedures shall calculate the arctangent of their argument x in the output range $[-\frac{\pi}{2},\frac{\pi}{2}]$ radians. ☐

**REQ-BL-0602**//GTD-TR-01-BL-0015/R

The `atan` and `atanf` procedures shall use a minimax polynomial for the calculation. ☐

**REQ-BL-0610**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `atan` and `atanf` procedures shall return NaN if the argument is NaN. ☐

**REQ-BL-0620**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `atan` and `atanf` procedures shall return the argument if the argument is ±0. ☐

**REQ-BL-0621**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `atan` and `atanf` procedures shall return $\pm\frac{\pi}{2}$ if the argument is `±Inf`. □

| X | Result | Requirement |
|---|---|---|
| ±0 | X | REQ-BL-0620 |
| -Inf | $-\frac{\pi}{2}$ | REQ-BL-0621 |
| +Inf | $+\frac{\pi}{2}$ | REQ-BL-0621 |
| NaN | NaN | REQ-BL-0610 |

Table 5.12: Special cases for atan(X)

### 5.2.9 ATAN2 ($\arctan y/x$)

**REQ-BL-0650**//GTD-TR-01-BL-0015/T

The `atan2` and `atan2f` procedures shall calculate the arctangent of the division $y/x$ of their arguments x and y in the output range [-$\pi$,$\pi$] radians. □

**REQ-BL-0652**//GTD-TR-01-BL-0015/R

The `atan2` and `atan2f` procedures shall use `atan` and `atanf` procedures. □

**REQ-BL-0660**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `atan2` and `atan2f` procedures shall return ±$\pi$, if the argument y is ±0 and the argument $x < 0$. □

**REQ-BL-0661**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `atan2` and `atan2f` procedures shall return ±0, if the argument y is ±0 and the argument $x > 0$. □

**REQ-BL-0662**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `atan2` and `atan2f` procedures shall return -$\frac{\pi}{2}$, if the argument y is < 0 and the argument x is ±0. □

**REQ-BL-0663**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `atan2` and `atan2f` procedures shall return $\frac{\pi}{2}$, if the argument y is > 0 and the argument x is ± 0. □

**REQ-BL-0670**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `atan2` and `atan2f` procedures shall return NaN if any argument is NaN. □

**REQ-BL-0680**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `atan2` and `atan2f` procedures shall return $\pm\pi$, if the argument y is $\pm 0$ and the argument x is -0. □

**REQ-BL-0681**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `atan2` and `atan2f` procedures shall return $\pm 0$, if the argument y is $\pm 0$ and the argument x is +0. □

**REQ-BL-0682**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `atan2` and `atan2f` procedures shall return $\pm\pi$, if the argument $\pm$y is finite and not 0, and the argument x is `-Inf`. □

**REQ-BL-0683**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `atan2` and `atan2f` procedures shall return $\pm 0$, if the argument $\pm$y is finite and not 0, and the argument x is `+Inf`. □

**REQ-BL-0684**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `atan2` and `atan2f` procedures shall return $\pm\frac{\pi}{2}$, if the argument y is $\pm$`Inf` and the argument x has a finite value. □

**REQ-BL-0685**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `atan2` and `atan2f` procedures shall return $\pm\frac{3\pi}{4}$, if the argument y is $\pm$`Inf` and the argument x is `-Inf`. □

**REQ-BL-0686**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `atan2` and `atan2f` procedures shall return $\pm\frac{\pi}{4}$, if the argument y is $\pm$`Inf` and the argument x is `+Inf`. □

| X | Y | Result | Requirement |
|---|---|---|---|
| <0 | -0 | $-\pi$ | REQ-BL-0660 |
| <0 | +0 | $+\pi$ | REQ-BL-0660 |
| -0 | -0 | $-\pi$ | REQ-BL-0680 |
| -0 | +0 | $+\pi$ | REQ-BL-0680 |
| ±0 | <0 | $-\frac{\pi}{2}$ | REQ-BL-0662 |
| ±0 | >0 | $+\frac{\pi}{2}$ | REQ-BL-0663 |
| +0 | -0 | -0 | REQ-BL-0681 |
| +0 | +0 | +0 | REQ-BL-0681 |
| >0 | -0 | -0 | REQ-BL-0661 |

| X | Y | Result | Requirement |
|---|---|---|---|
| >0 | +0 | +0 | REQ-BL-0661 |
| -Inf | <0 $\wedge \neq$ -Inf | -$\pi$ | REQ-BL-0682 |
| -Inf | >0 $\wedge \neq$ +Inf | +$\pi$ | REQ-BL-0682 |
| +Inf | <0 $\wedge \neq$ -Inf | -0 | REQ-BL-0683 |
| +Inf | >0 $\wedge \neq$ +Inf | +0 | REQ-BL-0683 |
| $\neq$ ±Inf | -Inf | -$\frac{\pi}{2}$ | REQ-BL-0684 |
| $\neq$ ±Inf | +Inf | +$\frac{\pi}{2}$ | REQ-BL-0684 |
| -Inf | -Inf | -$\frac{3\pi}{4}$ | REQ-BL-0685 |
| -Inf | +Inf | +$\frac{3\pi}{4}$ | REQ-BL-0685 |
| +Inf | -Inf | -$\frac{\pi}{4}$ | REQ-BL-0686 |
| +Inf | +Inf | +$\frac{\pi}{4}$ | REQ-BL-0686 |
| NaN | Any | NaN | REQ-BL-0670 |
| Any | NaN | NaN | REQ-BL-0670 |

Table 5.13: Special cases for atan2(Y, X)

### 5.2.10 SQRT ($\sqrt{x}$)

**REQ-BL-0700**//GTD-TR-01-BL-0015/T

The `sqrt` and `sqrtf` procedures shall calculate the square root of their argument x. □

**REQ-BL-0710**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `sqrt` and `sqrtf` procedures shall return NaN, if the argument x < -0. □

**REQ-BL-0720**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `sqrt` and `sqrtf` procedures shall return NaN if the argument x is NaN. □

**REQ-BL-0730**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `sqrt` and `sqrtf` procedures shall return the argument if the argument x is ±0 or `+Inf`. □

**REQ-BL-0740**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `sqrt` and `sqrtf` procedures shall return NaN, if the argument x is `-Inf`. □

| X | Result | Requirement |
|---|---|---|
| <-0 | NaN | REQ-BL-0710 |
| ±0 | X | REQ-BL-0730 |
| -Inf | NaN | REQ-BL-0740 |

| X | Result | Requirement |
|---|---|---|
| +Inf | +Inf | REQ-BL-0730 |
| NaN | NaN | REQ-BL-0720 |

Table 5.14: Special cases for sqrt(X)

### 5.2.11 EXP ($e^x$)

**REQ-BL-0800**//GTD-TR-01-BL-0015/T

The `exp` and `expf` procedures shall calculate the base $e$ exponential value of their argument `x`. □

**REQ-BL-0802**//GTD-TR-01-BL-0015/R

The `exp` and `expf` procedures shall use a minimax polynomial for the calculation. □

**REQ-BL-0831**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `exp` and `expf` procedures shall return `NaN` if the argument `x` is `NaN`. □

**REQ-BL-0832**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `exp` and `expf` procedures shall return 1 if the argument `x` is ±0. □

**REQ-BL-0833**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `exp` and `expf` procedures shall return +0 if the argument `x` is `-Inf`. □

**REQ-BL-0834**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `exp` and `expf` procedures shall return the argument if the argument `x` is `+Inf`. □

| X | Result | Requirement |
|---|---|---|
| ±0 | +1 | REQ-BL-0832 |
| -Inf | +0 | REQ-BL-0833 |
| +Inf | +Inf | REQ-BL-0834 |
| NaN | NaN | REQ-BL-0831 |

Table 5.15: Special cases for exp(X)

### 5.2.12 POW ($x^y$)

**REQ-BL-0850**//GTD-TR-01-BL-0015/T

The pow and powf procedures shall calculate the value of their argument x raised to the power of y. □

**REQ-BL-0851**//GTD-TR-01-BL-0015/R

The pow and powf procedures shall use a binary logarithm for the calculation. □

**REQ-BL-0860**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The pow and powf procedures shall return NaN if the argument x is < 0 and finite, and the argument y is a finite, non-integer value. □

**REQ-BL-0864**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The pow and powf procedures shall return
- ±Inf respectively if y is an odd integer, or
- +Inf if y is not an odd integer

if the argument y is < 0 and x is ±0. □

**REQ-BL-0870**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The pow and powf procedures shall return NaN if
- the argument y is NaN and x is not 1, or
- the argument x is NaN and y is not ±0. □

**REQ-BL-0885**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The pow and powf procedures shall return ±0 respectively, if the argument x is ±0 and the argument y is an odd integer > 0. □

**REQ-BL-0886**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The pow and powf procedures shall return +0, if the argument x is ±0 and the argument y is > 0 and not an odd integer. □

**REQ-BL-0871**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The pow and powf procedures shall return 1.0 if the argument x is +1. □

**REQ-BL-0872**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The pow and powf procedures shall return 1.0 if the argument y is ±0. □

**REQ-BL-0873**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `pow` and `powf` procedures shall return `1.0` if the argument `x` is `-1`, and the argument `y` is `±Inf`. □

**REQ-BL-0874**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `pow` and `powf` procedures shall return `+Inf` if $|x| < 1$ and the argument `y` is `-Inf`. □

**REQ-BL-0875**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `pow` and `powf` procedures shall return `+0`, if $|x| > 1$ and the argument `y` is `-Inf`. □

**REQ-BL-0876**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `pow` and `powf` procedures shall return `+0`, if $|x| < 1$ and the argument `y` is `+Inf`. □

**REQ-BL-0877**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `pow` and `powf` procedures shall return `+Inf`, if $|x| > 1$ and the argument `y` is `+Inf`. □

**REQ-BL-0878**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `pow` and `powf` procedures shall return `-0`, if the argument `x` is `-Inf` and the argument `y` is an odd integer < 0. □

**REQ-BL-0879**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `pow` and `powf` procedures shall return +0, if the argument `x` is `-Inf` and the argument `y` is < 0 and not an odd integer. □

**REQ-BL-0880**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `pow` and `powf` procedures shall return `-Inf`, if the argument `x` is `-Inf` and the argument `y` is an odd integer > 0. □

**REQ-BL-0881**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `pow` and `powf` procedures shall return `+Inf`, if the argument `x` is `-Inf` and the argument `y` is > 0 and not an odd integer. □

**REQ-BL-0882**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `pow` and `powf` procedures shall return +0, if the argument `x` is `+Inf` and the argument `y` < 0. □

**REQ-BL-0883**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `pow` and `powf` procedures shall return `+Inf`, if the argument `x` is `+Inf` and the argument `y` > 0. □

| X | Y | Result | Requirement |
|---|---|---|---|
| <0 | $\notin \mathbb{Z}$ | NaN | REQ-BL-0860 |
| -0 | $\{2k+1 : k \in \mathbb{Z}_{<0}\}$ | -Inf | REQ-BL-0864 |
| -0 | $\in \mathbb{R}_{<0} \setminus \{2k+1 : k \in \mathbb{Z}\}$ | +Inf | REQ-BL-0864 |
| +0 | <0 | +Inf | REQ-BL-0864 |
| ±0 | $\{2k-1 : k \in \mathbb{Z}_{>0}\}$ | X | REQ-BL-0885 |
| ±0 | $\in \mathbb{R}_{>0} \setminus \{2k-1 : k \in \mathbb{Z}\}$ | +0 | REQ-BL-0886 |
| +1 | Any | 1.0 | REQ-BL-0871 |
| Any | ±0 | 1.0 | REQ-BL-0872 |
| -1 | ±Inf | 1.0 | REQ-BL-0873 |
| ∈]-1, 1[ | -Inf | +Inf | REQ-BL-0874 |
| ∉[-1, 1] | -Inf | +0 | REQ-BL-0875 |
| ∈]-1, 1[ | +Inf | +0 | REQ-BL-0876 |
| ∉[-1, 1] | +Inf | +Inf | REQ-BL-0877 |
| -Inf | $\{2k+1 : k \in \mathbb{Z}_{<0}\}$ | -0 | REQ-BL-0878 |
| -Inf | $\in \mathbb{R}_{<0} \setminus \{2k+1 : k \in \mathbb{Z}\}$ | +0 | REQ-BL-0879 |
| -Inf | $\{2k-1 : k \in \mathbb{Z}_{>0}\}$ | -Inf | REQ-BL-0880 |
| -Inf | $\in \mathbb{R}_{>0} \setminus \{2k-1 : k \in \mathbb{Z}\}$ | +Inf | REQ-BL-0881 |
| +Inf | <0 | +0 | REQ-BL-0882 |
| +Inf | >0 | +Inf | REQ-BL-0883 |
| NaN | $\neq$±0 | NaN | REQ-BL-0870 |
| $\neq$1 | NaN | NaN | REQ-BL-0870 |

Table 5.16: Special cases for pow(X, Y)

### 5.2.13 LOG ($\ln x$)

**REQ-BL-0900**//GTD-TR-01-BL-0015/T

The `log` and `logf` procedures shall calculate the natural logarithm of their argument `x`. □

**REQ-BL-0910**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `log` and `logf` procedures shall return `-Inf` if the argument `x` is ±0. □

**REQ-BL-0920**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `log` and `logf` procedures shall return `NaN` if the argument `x` is finite and less than 0 or `x` is `-Inf`. □

**REQ-BL-0921**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `log` and `logf` procedures shall return `NaN`, if the argument `x` is `NaN`. □

**REQ-BL-0930**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `log` and `logf` procedures shall return +0, if the argument `x` is 1. □

**REQ-BL-0931**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `log` and `logf` procedures shall return `+Inf`, if the argument `x` is `+Inf`. □

| X | Result | Requirement |
|---|---|---|
| <0 | NaN | REQ-BL-0920 |
| ±0 | -Inf | REQ-BL-0910 |
| 1 | +0 | REQ-BL-0930 |
| +Inf | +Inf | REQ-BL-0931 |
| NaN | NaN | REQ-BL-0921 |

Table 5.17: Special cases for log(X)

### 5.2.14 LOG10 ($\log_{10} x$)

**REQ-BL-0950**//GTD-TR-01-BL-0015/T

The `log10` and `log10f` procedures shall calculate the base 10 logarithm of their argument `x`. □

**REQ-BL-0960**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `log10` and `log10f` procedures shall return `-Inf` if the argument `x` is ±0. □

**REQ-BL-0970**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `log10` and `log10f` procedures shall return `NaN` if the argument `x` is finite and less than 0 or `x` is `-Inf`. □

**REQ-BL-0971**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `log10` and `log10f` procedures shall return `NaN`, if the argument `x` is `NaN`. □

**REQ-BL-0980**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `log10` and `log10f` procedures shall return +0, if the argument x is 1. □

**REQ-BL-0981**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `log10` and `log10f` procedures shall return `+Inf`, if the argument x is `+Inf`. □

| X | Result | Requirement |
|---|---|---|
| <0 | NaN | REQ-BL-0970 |
| ±0 | -Inf | REQ-BL-0960 |
| 1 | +0 | REQ-BL-1280 |
| +Inf | +Inf | REQ-BL-0981 |
| NaN | NaN | REQ-BL-0971 |

Table 5.18: Special cases for log10(X)

### 5.2.15 FABS ($|x|$)

**REQ-BL-1000**//GTD-TR-01-BL-0015/T

The `fabs` and `fabsf` procedures shall calculate the absolute values of their argument x. □

**REQ-BL-1010**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `fabs` and `fabsf` procedures shall return NaN, if the argument x is NaN. □

**REQ-BL-1011**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `fabs` and `fabsf` procedures shall return +0, if the argument x is ±0. □

**REQ-BL-1012**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `fabs` and `fabsf` procedures shall return `+Inf`, if the argument x is `±Inf`. □

| X | Result | Requirement |
|---|---|---|
| ±0 | +0 | REQ-BL-1011 |
| ±Inf | +Inf | REQ-BL-1012 |
| NaN | NaN | REQ-BL-1010 |

Table 5.19: Special cases for fabs(X)

### 5.2.16 ROUND ($\lfloor x + 0.5 \rfloor, x \geqslant 0; \lceil x - 0.5 \rceil, x < 0$)

**REQ-BL-1020**//GTD-TR-01-BL-0015/T

The `round` and `roundf` procedures shall round their argument `x` to the nearest integer value, rounding halfway cases away from zero. □

**REQ-BL-1031**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `round` and `roundf` procedures shall return `NaN`, if the argument `x` is `NaN`. □

**REQ-BL-1032**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `round` and `roundf` procedures shall return the argument, if the argument `x` is `±0` or `±Inf`. □

| X | Result | Requirement |
|---|---|---|
| ±0 | X | REQ-BL-1032 |
| ±Inf | X | REQ-BL-1032 |
| NaN | NaN | REQ-BL-1031 |

Table 5.20: Special cases for round(X)

### 5.2.17 FLOOR ($\lfloor x \rfloor$)

**REQ-BL-1040**//GTD-TR-01-BL-0015/T

The `floor` and `floorf` procedures shall calculate the largest integral value not greater than their argument `x`. □

**REQ-BL-1051**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `floor` and `floorf` procedures shall return `NaN`, if the argument `x` is `NaN`. □

**REQ-BL-1052**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `floor` and `floorf` procedures shall return the argument, if the argument `x` is `±0` or `±Inf`. □

| X | Result | Requirement |
|---|---|---|
| ±0 | X | REQ-BL-1052 |
| ±Inf | X | REQ-BL-1052 |
| NaN | NaN | REQ-BL-1051 |

Table 5.21: Special cases for floor(X)

### 5.2.18 TRUNC ($\lfloor x \rfloor, x \geqslant 0; \lceil x \rceil, x < 0$)

**REQ-BL-1060**//GTD-TR-01-BL-0015/T

The `trunc` and `truncf` procedures shall round their argument `x` to the integer value nearest to but no larger in magnitude than the argument. □

**REQ-BL-1070**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `trunc` and `truncf` procedures shall return `NaN`, if the argument `x` is `NaN`. □

**REQ-BL-1071**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `trunc` and `truncf` procedures shall return the argument, if the argument `x` is `±0` or `±Inf`. □

| X | Result | Requirement |
|---|---|---|
| ±0 | X | REQ-BL-1072 |
| ±Inf | X | REQ-BL-1072 |
| NaN | NaN | REQ-BL-1071 |

Table 5.22: Special cases for trunc(X)

### 5.2.19 CEIL ($\lceil x \rceil$)

**REQ-BL-1080**//GTD-TR-01-BL-0015/T

The `ceil` and `ceilf` procedures shall compute the smallest integral value not less than argument `x`. □

**REQ-BL-1091**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `ceil` and `ceilf` procedures shall return `NaN`, if the argument `x` is `NaN`. □

**REQ-BL-1092**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `ceil` and `ceilf` procedures shall return the argument, if the argument `x` is `±0` or `±Inf`. □

| X | Result | Requirement |
|---|---|---|
| ±0 | X | REQ-BL-1092 |
| ±Inf | X | REQ-BL-1092 |
| NaN | NaN | REQ-BL-1091 |

Table 5.23: Special cases for ceil(X)

### 5.2.20 FMOD ($x \mod y$)

**REQ-BL-1100**//GTD-TR-01-BL-0015/T

The `fmod` and `fmodf` procedures shall calculate the floating-point remainder of the division of the argument `x` by the argument `y`.

*Remark:* The sign of the remainder is defined by the ISO C99 standard as the sign of the dividend (argument `x`). Therefore an argument `y` < 0 produces the same result as it's absolute value.

*Example:* Input of x = 3.456 and y = -2 results in a return value of 1.456. □

**REQ-BL-1120**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `fmod` and `fmodf` procedures shall return `NaN`, if any argument is `NaN`. □

**REQ-BL-1121**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `fmod` and `fmodf` procedures shall return `NaN`, if the argument `y` is 0. □

**REQ-BL-1122**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `fmod` and `fmodf` procedures shall return `NaN`, if the argument `x` is ±Inf. □

**REQ-BL-1130**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `fmod` and `fmodf` procedures shall return ±0, if the argument `x` is ±0 and the argument `y` is not zero. □

**REQ-BL-1131**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `fmod` and `fmodf` procedures shall return the argument `x`, if the argument `x` is not ±Inf and the argument `y` is `±Inf`. □

| X | Y | Result | Requirement |
|---|---|---|---|
| Any | ±0 | NaN | REQ-BL-1121 |
| ±Inf | Any | NaN | REQ-BL-1122 |
| ±0 | $\neq$ ±0 | X | REQ-BL-1130 |
| $\neq$ ±Inf | ±Inf | X | REQ-BL-1131 |
| NaN | Any | NaN | REQ-BL-1120 |
| Any | NaN | NaN | REQ-BL-1120 |

Table 5.24: Special cases for fmod(X, Y)

### 5.2.21 MODF

**REQ-BL-1200**//GTD-TR-01-BL-0015/T

The `modf` and `modff` procedures shall compute the integral and fractional part of the argument x.

*Remark:* The sign of both the integral and fractional part is defined by the ISO C99 standard as the sign of the argument x.

*Example:* Input of x = -3.456 results in a return value of -0.456 and sets the value pointed to by the argument $*iptr$ to -3.0. □

**REQ-BL-1201**//GTD-TR-01-BL-0015/T

The `modf` and `modff` procedures shall return the fractional part of the argument x and write the integral part of the argument x to the pointer provided by the argument $*iptr$. □

**REQ-BL-1210**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `modf` and `modff` procedures shall return NaN and set the argument $*iptr$ to NaN, if the argument x is NaN. □

**REQ-BL-1211**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `modf` and `modff` procedures shall return ±0 and set the argument $*iptr$ to `±Inf`, if the argument x is `±Inf`. □

| X | $*iptr$ | Result | Requirement |
|---|---|---|---|
| -Inf | -Inf | -0 | REQ-BL-1211 |
| +Inf | +Inf | +0 | REQ-BL-1211 |
| NaN | NaN | NaN | REQ-BL-1210 |

Table 5.25: Special cases for modf(X, $*iptr$)

### 5.2.22 FMIN ($min(x, y)$)

**REQ-BL-1220**//GTD-TR-01-BL-0015/T

The `fmin` and `fminf` procedures shall determine the minimum numeric value of the argument x and y.

*Remark:* This includes `±Inf`, with `-Inf` being lesser than any other numeric value and `+Inf` being greater than any other numeric value. □

**REQ-BL-1230**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `fmin` and `fminf` procedures shall return NaN, if the arguments x and y are NaN. □

**REQ-BL-1231**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `fmin` and `fminf` procedures shall return the one argument if only the other argument is NaN. □

**REQ-BL-1232**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `fmin` and `fminf` procedures shall return the y argument if both arguments are zero. □

| X | Y | Result | Requirement |
|---|---|---|---|
| -0 | -0 | -0 | REQ-BL-1232 |
| -0 | +0 | +0 | REQ-BL-1232 |
| +0 | -0 | -0 | REQ-BL-1232 |
| +0 | +0 | +0 | REQ-BL-1232 |
| NaN | $\neq$ NaN | Y | REQ-BL-1231 |
| $\neq$ NaN | NaN | X | REQ-BL-1231 |
| NaN | NaN | NaN | REQ-BL-1230 |

Table 5.26: Special cases for fmin(X, Y)

### 5.2.23 FMAX ($max(x, y)$)

**REQ-BL-1240**//GTD-TR-01-BL-0015/T

The `fmax` and `fmaxf` procedures shall determine the maximum numeric value of the argument `x` and `y`.

*Remark:* This includes ±`Inf`, with `-Inf` being lesser than any other numeric value and `+Inf` being greater than any other numeric value. □

**REQ-BL-1250**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `fmax` and `fmaxf` procedures shall return NaN, if the arguments `x` and `y` are NaN. □

**REQ-BL-1251**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `fmax` and `fmaxf` procedures shall return the one argument if only the other argument is NaN. □

**REQ-BL-1252**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `fmax` and `fmaxf` procedures shall return the y argument if both arguments are zero. □

| X | Y | Result | Requirement |
|---|---|---|---|
| -0 | -0 | -0 | REQ-BL-1252 |
| -0 | +0 | +0 | REQ-BL-1252 |

| X | Y | Result | Requirement |
|---|---|---|---|
| +0 | -0 | -0 | REQ-BL-1252 |
| +0 | +0 | +0 | REQ-BL-1252 |
| NaN | ≠ NaN | Y | REQ-BL-1251 |
| ≠ NaN | NaN | X | REQ-BL-1251 |
| NaN | NaN | NaN | REQ-BL-1250 |

Table 5.27: Special cases for fmax(X, Y)

### 5.2.24 HYPOT ($\sqrt{x^2 + y^2}$)

**REQ-BL-1260**//GTD-TR-01-BL-0015/T

The `hypot` and `hypotf` procedures shall compute the length of the hypotenuse of a rightangled triangle with sides of length `x` and `y`. □

**REQ-BL-1270**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `hypot` and `hypotf` procedures shall return `+Inf`, if one of the arguments `x` or `y` is `±Inf`. □

**REQ-BL-1271**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `hypot` and `hypotf` procedures shall return `NaN`, if one of the arguments `x` or `y` is `NaN` and the other is not `±Inf`. □

| X | Y | Result | Requirement |
|---|---|---|---|
| Any | ±Inf | +Inf | REQ-BL-1270 |
| ±Inf | Any | +Inf | REQ-BL-1270 |
| NaN | ≠ ±Inf | NaN | REQ-BL-1271 |
| ≠ ±Inf | NaN | NaN | REQ-BL-1271 |

Table 5.28: Special cases for hypot(X, Y)

### 5.2.25 ISFINITE

**REQ-BL-1300**//GTD-TR-01-BL-0015/T

The `isfinite` procedure shall return a non-zero value if the argument `x` has a finite value and is neither `NaN` nor `±Inf`. □

### 5.2.26 ISINF

**REQ-BL-1320**//GTD-TR-01-BL-0015/T

The `isinf` procedure shall return a non-zero value if and only if the argument `x` is `±Inf`. □

### 5.2.27 ISNAN

**REQ-BL-1340**//GTD-TR-01-BL-0015/T

The `isnan` procedure shall return a non-zero value if and only if the argument `x` is NaN. □

### 5.2.28 SIGNBIT

**REQ-BL-1360**//GTD-TR-01-BL-0015/T

The `signbit` procedure shall return a non-zero value if and only if the argument `x` is negative. □

### 5.2.29 COPYSIGN ($|x| \cdot sgn\ y$)

**REQ-BL-1380**//GTD-TR-01-BL-0015/T

The `copysign` and `copysignf` procedures shall compute a new number with the magnitude of the argument `x` and the the sign of `y`.

*Remark:* This includes NaN values for argument y, as NaN values have a sign just as any other value. □

**REQ-BL-1381**//GTD-TR-01-BL-0015, GTD-TR-01-BL-0026/T

The `copysign` and `copysignf` procedures shall return NaN if the argument `x` is NaN. □

| X | Y | Result | Requirement |
|---|---|---|---|
| NaN | Any | NaN | REQ-BL-1381 |

Table 5.29: Special cases for copysign(X, Y)

## 5.3 Performance Requirements

**REQ-BL-0061**//GTD-TR-01-BL-0029/T

The execution time of all procedures shall be bounded. □

**REQ-BL-0062**//GTD-TR-01-BL-0030/A

The execution time of all procedures shall be characterizable within subdomains which cover the complete domain of the procedure.

*Remark:* For example the sin procedure can be split into the subdomains (`-Inf`, $-2^{-27}$], ($-2^{-27}$, $+2^{-27}$) and [$+2^{-27}$, `+Inf`), figure 5.1 shows a plot of the execution time of sine for positive arguments. □

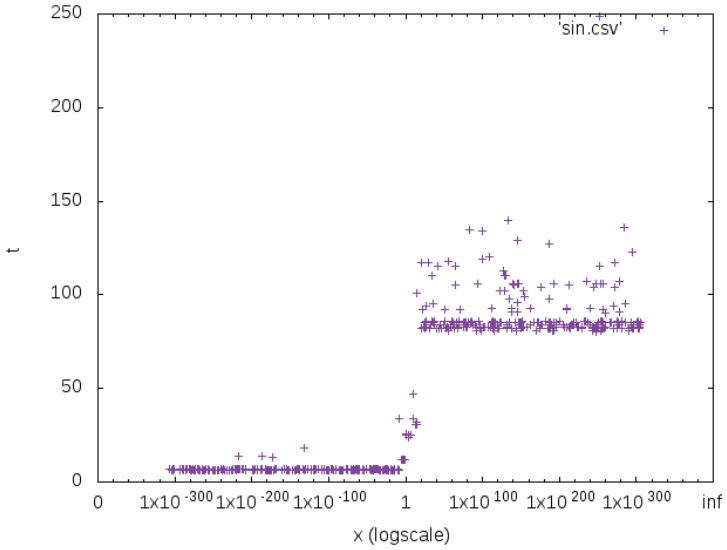


Figure 5.1: Execution time of sine for positive arguments

**REQ-BL-0060**//GTD-TR-01-BL-0030/A

The Worst Case Execution Time (WCET) of the procedures shall not deviate by more than 30% from the median execution times of the characterized ranges.

*Remark:* The median shall be defined with random arguments within the range. □

**REQ-BL-0070**//GTD-TR-01-BL-0033/T

The BL procedure results shall differ from theoretical results by less than 0.5 Units in the Last Place (ULPs). □

## 5.4 Interface Requirements

The requirement REQ-BL-0021 regarding compliance to the definitions in math.h of the ISO C99 standard [RD03] can be found in §5.9. □

**REQ-BL-0010**//GTD-TR-01-BL-0004/T

The Basic Library shall be compliant to IEEE 754-2008. □

**REQ-BL-0080**//GTD-TR-01-BL-0012/D

The Basic Library shall be compatible with code auto-generated with Matlab R2016a Simulink Coder.

*Remark:* The auto-generated code will require an include of `mlfs.h`.

*Remark:* All C definition and function signatures not present in `mlfs.h` need to be provided by the user. He will then have to link his own libm after linking with the MLFS. □

**REQ-BL-0120**//GTD-TR-01-BL-0013/D

The Basic Library shall provide best practices within the Software User Manual (SUM) to produce Ada wrappers to enable the use of the library from Ada83 and Ada95. ☐

**REQ-BL-0130**//GTD-TR-01-BL-0008, GTD-TR-01-BL-0010/D

The Basic Library shall provide best practices within the SUM to produce C-MEX function wrappers, to enable the use of the Basic Library in Matlab and Simulink Matlab function blocks. ☐

**REQ-BL-0131**//GTD-TR-01-BL-0009, GTD-TR-01-BL-0011/D

The Basic Library shall provide best practices within the SUM to produce S-function wrappers, to enable the use of the Basic Library in Simulink. ☐

## 5.5 Operational Requirements

Not applicable.

## 5.6 Resources Requirements

**REQ-BL-0180**//GTD-TR-01-BL-0001/T

The Basic Library shall run on x86-64 and SPARC V8 processor architectures. ☐

**REQ-BL-0181**//GTD-TR-01-BL-0031/D

The Basic Library shall contribute less than 100kB to the size of the final OBSW executable. ☐

## 5.7 Design Requirements and Implementation Constraints

**REQ-BL-0090**//GTD-TR-01-BL-0001/I

For the Basic Library it shall be selectable for which processor architecture it will be compiled. ☐

**REQ-BL-0100**//GTD-TR-01-BL-0007/D

For the Basic Library it shall be configurable which procedures of the library will be included in the linking step. ☐

**REQ-BL-0110**//GTD-TR-01-BL-0028/R

The Basic Library procedures shall be reentrant and thread-safe. ☐

**REQ-BL-0111**//GTD-TR-01-BL-0027/R

The Basic Library design shall define the behavior of the procedures in a tabular format, presenting the parameter domain decomposition and the corresponding output. □

**REQ-BL-0112**//GTD-TR-01-BL-0032/R

The Basic Library design shall justify every modification to be carried out on the reused library. □

## 5.8 Security and Privacy Requirements

Not applicable.

## 5.9 Portability Requirements

**REQ-BL-0020**//GTD-TR-01-BL-0002/I

The Basic Library shall be implemented in ISO C99. □

**REQ-BL-0021**//GTD-TR-01-BL-0016/R

The Basic Library shall be compliant to the math.h definitions as per ISO C99 standard [RD03] . □

**REQ-BL-0022**//GTD-TR-01-BL-0025/I

The Basic Library shall not provide error handling with the `errno` global variable.
*Remark:* All procedures that would have produced an error will instead return a defined value (see requirements for special cases), often NaN. □

**REQ-BL-0025**//GTD-TR-01-BL-0007/T

The Basic Library shall be compileable with GCC version 4.2.1 with Binutils version 2.18. □

**REQ-BL-0026**//GTD-TR-01-BL-0003/T

The Basic Library shall be compatible to RTEMS OS version 4.8 distributed by Edisoft. □

**REQ-BL-0027**//GTD-TR-01-BL-0021/I

The Basic Library shall not use Floating Point Unit (FPU) built in functions for elementary math function evaluation. □

**REQ-BL-0028**//GTD-TR-01-BL-0022/R

The Basic Library shall not rely on the presence of the Fused Multiply-Add (FMA) operation for algorithm optimization. □

**REQ-BL-0029**//GTD-TR-01-BL-0023/T

The Basic Library shall be able to convert to a modus that mirrors the FPU's behaviour regarding subnormal numbers in case the FPU does not handle subnormal numbers the same as normal numbers.

*Remark:* For example the FPU may simply throw a trap, or have the possibility to be set to Denormals are Zero (DAZ) and Flush to Zero (FTZ) behaviour. In both cases the library shall behave the same way as the FPU does. □

## 5.10 Software Quality Requirements

**REQ-BL-1500**//GTD-TR-01-BL-0006/A

The Basic Library source shall be compliant to the mandatory and required rules depicted in MISRA C:2012 [RD06] . □

**REQ-BL-1501**//GTD-TR-01-BL-0006/A

The Basic Library shall provide a justification for not respected required rules in MISRA C:2012 [RD06] . □

**REQ-BL-1502**//GTD-TR-01-BL-0005/A

The Basic Library shall comply to the mandatory metrics provided by ECSS-HB-Q-80 [RD05] with their proposed targets. □

**REQ-BL-1503**//GTD-TR-01-BL-0005/A

The Basic Library shall comply to the following project depending mandatory metrics proposed by ECSS-HB-Q-80 [RD05] :

| Metric name | Target |
|---|---|
| Statement Coverage (Source Code) | 1 |
| Modified Condition and Decision Coverage (Source Code) | 1 |
| Reuse modification rate | 0.3 |
| Code size stability | 1.3 |
| Requirement stability | 0.3 |

□

**REQ-BL-1504**//GTD-TR-01-BL-0005/A

The Basic Library shall comply to required metric 'User documentation completeness' proposed by ECSS-HB-Q-80 [RD05] with their proposed targets. ☐

## 5.11 Software Reliability Requirements

**REQ-BL-1600**//GTD-TR-01-BL-0014/T

The Basic Library shall return the exact same value whenever a procedure is called multiple times with the same argument(s). ☐

## 5.12 Software Maintainability Requirements

Upholding the requirements listed in 5.10 should produce a maintainable software. No additional maintainability requirements specified.

## 5.13 Software Safety Requirements

**REQ-BL-1800**//GTD-TR-01-BL-0001/R

The Basic Library shall never cause either CPU or FPU to stop. ☐

## 5.14 Software Configuration and Delivery Requirements

**REQ-BL-1900**//GTD-TR-01-BL-0002/I

The Basic Library source shall be delivered as a collection of C source files.
*Remark:* There will be neither a binary nor an executable delivered. ☐

**REQ-BL-1901**//GTD-TR-01-BL-0007/I

The Basic Library shall be delivered with a GNU Make makefile and associated scripts to configure, build and install the Basic Library. ☐

**REQ-BL-1902**//SoW §3.5/I

Each modified or newly created file, be it source or documentation, of the Basic Library shall contain a header including the ESA copyright notice:
Copyright European Space Agency, 20xx
*Remark:* xx corresponds to the applicable year. ☐

## 5.15 Data Definition and Database requirements

**REQ-BL-2000**//GTD-TR-01-BL-0017/R

The Basic Library shall use 32bit and 64bit IEEE 754 [RD01] floating-point datums as well as 32bit integers to exchange data. □

## 5.16 Human Factors Related Requirements

Not applicable.

## 5.17 Adaptation and Installation Requirements

The Basic Library can be installed using a GNU Make makefile, see REQ-BL-1901 in 5.14.