

# **Math Library Manual**

---

# Math Library Manual

Version X.Y

---

# Table of Contents

1. Concepts .....	1
2. Reference .....	2
math reference .....	2
Exponential and logarithmic functions .....	2
Hyperbolic functions .....	3
LIB_VERSION .....	3
LIB_VERSION_TYPE .....	3
Nearest integer, absolute value, and remainder functions .....	4
Power functions .....	4
Trigonometric functions .....	5
struct exception .....	6
math_h_ .....	6
matherr .....	6

---

# Chapter 1. Concepts

---

# Chapter 2. Reference

## math reference

### Exponential and logarithmic functions

#### exp

```
double exp(double x);
```

Exponential function of X.

#### frexp

```
double frexp(double x, int* e);
```

Break VALUE into a normalized fraction and an integral power of 2.

#### ldexp

```
double ldexp(double x, int e);
```

X times (two to the EXP power).

#### log

```
double log(double x);
```

Natural logarithm of X.

#### log10

```
double log10(double x);
```

Base-ten logarithm of X.

#### modf

```
double modf(double x, double* iptr);
```

Break VALUE into integral and fractional parts.

#### exp10

```
double exp10(double x);
```

A function missing in all standards: compute exponent to base ten.

#### pow10

```
double pow10(double x);
```

Another name occasionally used.

## Hyperbolic functions

### cosh

```
double cosh(double x);
```

Hyperbolic cosine of X.

### sinh

```
double sinh(double x);
```

Hyperbolic sine of X.

### tanh

```
double tanh(double x);
```

Hyperbolic tangent of X.

### acosh

```
double acosh(double x);
```

Hyperbolic arc cosine of X.

### asinh

```
double asinh(double x);
```

Hyperbolic arc sine of X.

### atanh

```
double atanh(double x);
```

Hyperbolic arc tangent of X.

## LIB\_VERSION

```
LIB_VERSION_TYPE LIB_VERSION;
```

This variable can be changed at run-time to any of the values above to affect floating point error handling behavior (it may also be necessary to change the hardware FPU exception settings).

## LIB\_VERSION\_TYPE

```
enum LIB_VERSION_TYPE { IEEE=- 1, SVID, XOPEN, POSIX, ISOC};
```

## Nearest integer, absolute value, and remainder functions

### ceil

```
double ceil(double x);
```

Smallest integral value not less than X.

### fabs

```
double fabs(double x);
```

Absolute value of X.

### floor

```
double floor(double x);
```

Largest integer not greater than X.

### fmod

```
double fmod(double x, double y);
```

Floating-point modulo remainder of X/Y.

### isinf

```
int isinf(double x);
```

Return 0 if VALUE is finite or NaN, +1 if it is +Infinity, -1 if it is -Infinity.

### isfinite

```
int isfinite(double x);
```

Return nonzero if VALUE is finite and not NaN.

### isnan

```
int isnan(double x);
```

Return nonzero if VALUE is not a number.

## Power functions

### pow

```
double pow(double x, double y);
```

Return X to the Y power.

## sqrt

```
double sqrt(double x);
```

Return the square root of X.

## hypot

```
double hypot(double x, double y);
```

Return  $\sqrt{X^2 + Y^2}$ .

## cbrt

```
double cbrt(double x);
```

Return the cube root of X.

# Trigonometric functions

## acos

```
double acos(double x);
```

Arc cosine of X.

## asin

```
double asin(double x);
```

Arc sine of X.

## atan

```
double atan(double x);
```

Arc tangent of X.

## atan2

```
double atan2(double y, double x);
```

Arc tangent of Y/X.

## cos

```
double cos(double x);
```

Cosine of X.

## sin

```
double sin(double x);
```



Sine of X.

## tan

```
double tan(double x);
```

Tangent of X.

## struct exception

### type

```
int type;
```

### name

```
char * name;
```

### arg1

```
double arg1;
```

### arg2

```
double arg2;
```

### retval

```
double retval;
```

## math\_h\_

```
math_h_
```

## matherr

```
int matherr(exception* exc);
```

# Types

## E

exception  
exception, 6

## L

LIB\_VERSION\_TYPE  
LIB\_VERSION\_TYPE, 3

# Functions

## A

acos  
    acos, 5  
acosh  
    acosh, 3  
asin  
    asin, 5  
asinh  
    asinh, 3  
atan  
    atan, 5  
atan2  
    atan2, 5  
atanh  
    atanh, 3

## C

cbrt  
    cbrt, 5  
ceil  
    ceil, 4  
cos  
    cos, 5  
cosh  
    cosh, 3

## E

exp  
    exp, 2  
exp10  
    exp10, 2

## F

fabs  
    fabs, 4  
floor  
    floor, 4  
fmod  
    fmod, 4  
frexp  
    frexp, 2

## H

hypot  
    hypot, 5

## I

isfinite

- isfinite, 4
- isinf
  - isinf, 4
- isnan
  - isnan, 4

## **L**

- ldexp
  - ldexp, 2
- log
  - log, 2
- log10
  - log10, 2

## **M**

- matherr
  - matherr, 6
- modf
  - modf, 2

## **P**

- pow
  - pow, 4
- pow10
  - pow10, 2

## **S**

- sin
  - sin, 5
- sinh
  - sinh, 3
- sqrt
  - sqrt, 5

## **T**

- tan
  - tan, 6
- tanh
  - tanh, 3