## NAME

**math** - floating-point mathematical library

## LIBRARY

Math Library (libm, -lm)

## SYNOPSIS

**#include <math.h>**

## DESCRIPTION

The math library includes the following components:

| | |
|---|---|
| <*math.h*> | basic routines and real-valued functions |
| <*complex.h*> | complex number support |
| <*tgmath.h*> | polymorphic (type-generic) versions of functions |
| <*fenv.h*> | routines to control rounding and exceptions |

The rest of this manual page describes the functions provided by <*math.h*>. Please consult complex(3), tgmath(3), and fenv(3) for information on the other components.

## LIST OF FUNCTIONS

Each of the following *double* functions has a *float* counterpart with an ‘f’ appended to the name and a *long double* counterpart with an ‘l’ appended. As an example, the *float* and *long double* counterparts of *double* **acos**(*double x*) are *float* **acosf**(*float x*) and *long double* **acosl**(*long double x*), respectively. The classification macros and silent order predicates are type generic and should not be suffixed with ‘f’ or ‘l’.

### Algebraic Functions

| *Name* | Description |
|---|---|
| cbrt | cube root |
| fma | fused multiply-add |
| hypot | Euclidean distance |
| sqrt | square root |

### Classification Macros

| *Name* | Description |
|---|---|
| fpclassify | classify a floating-point value |
| isfinite | determine whether a value is finite |
| isinf | determine whether a value is infinite |
| isnan | determine whether a value is NaN |
| isnormal | determine whether a value is normalized |

**Exponent Manipulation Functions**

| *Name* | Description |
|---|---|
| frexp | extract exponent and mantissa |
| ilogb | extract exponent |
| ldexp | multiply by power of 2 |
| logb | extract exponent |
| scalbln | adjust exponent |
| scalbn | adjust exponent |

**Extremum- and Sign-Related Functions**

| *Name* | Description |
|---|---|
| copysign | copy sign bit |
| fabs | absolute value |
| fdim | positive difference |
| fmax | maximum function |
| fmin | minimum function |
| signbit | extract sign bit |

**Not a Number Functions**

| *Name* | Description |
|---|---|
| nan | generate a quiet NaN |

**Residue and Rounding Functions**

| *Name* | Description |
|---|---|
| ceil | integer no less than |
| floor | integer no greater than |
| fmod | positive remainder |
| llrint | round to integer in fixed-point format |
| llround | round to nearest integer in fixed-point format |
| lrint | round to integer in fixed-point format |
| lround | round to nearest integer in fixed-point format |
| modf | extract integer and fractional parts |
| nearbyint | round to integer (silent) |
| nextafter | next representable value |
| nexttoward | next representable value |
| remainder | remainder |
| remquo | remainder with partial quotient |
| rint | round to integer |
| round | round to nearest integer |
| trunc | integer no greater in magnitude than |

The **ceil**(), **floor**(), **llround**(), **lround**(), **round**(), and **trunc**() functions round in predetermined directions, whereas **llrint**(), **lrint**(), and **rint**() round according to the current (dynamic) rounding mode. For more information on controlling the dynamic rounding mode, see fenv(3) and fesetround(3).

**Silent Order Predicates**

| *Name* | Description |
|---|---|
| isgreater | greater than relation |
| isgreaterequal | greater than or equal to relation |
| isless | less than relation |
| islessequal | less than or equal to relation |
| islessgreater | less than or greater than relation |
| isunordered | unordered relation |

**Transcendental Functions**

| *Name* | Description |
|---|---|
| acos | inverse cosine |
| acosh | inverse hyperbolic cosine |
| asin | inverse sine |
| asinh | inverse hyperbolic sine |
| atan | inverse tangent |
| atanh | inverse hyperbolic tangent |
| atan2 | atan(y/x); complex argument |
| cos | cosine |
| cosh | hyperbolic cosine |
| erf | error function |
| erfc | complementary error function |
| exp | exponential base e |
| exp2 | exponential base 2 |
| expm1 | exp(x)-1 |
| j0 | Bessel function of the first kind of the order 0 |
| j1 | Bessel function of the first kind of the order 1 |
| jn | Bessel function of the first kind of the order n |
| lgamma | log gamma function |
| log | natural logarithm |
| log10 | logarithm to base 10 |
| log1p | log(1+x) |
| log2 | base 2 logarithm |
| pow | exponential x**y |
| sin | trigonometric function |
| sinh | hyperbolic function |

| | |
|---|---|
| tan | trigonometric function |
| tanh | hyperbolic function |
| tgamma | gamma function |
| y0 | Bessel function of the second kind of the order 0 |
| y1 | Bessel function of the second kind of the order 1 |
| yn | Bessel function of the second kind of the order n |

The routines in this section might not produce a result that is correctly rounded, so reproducible results cannot be guaranteed across platforms. For most of these functions, however, incorrect rounding occurs rarely, and then only in very-close-to-halfway cases.

## SEE ALSO

complex(3), fenv(3), ieee(3), qmath(3), tgmath(3)

## HISTORY

A math library with many of the present functions appeared in Version 7 AT&T UNIX. The library was substantially rewritten for 4.3BSD to provide better accuracy and speed on machines supporting either VAX or IEEE 754 floating-point. Most of this library was replaced with FDLIBM, developed at Sun Microsystems, in FreeBSD 1.1.5. Additional routines, including ones for *float* and *long double* values, were written for or imported into subsequent versions of FreeBSD.

## BUGS

Many of the routines to compute transcendental functions produce inaccurate results in other than the default rounding mode.

On the i386 platform, trigonometric argument reduction is not performed accurately for huge arguments, resulting in large errors for such arguments to **cos**(), **sin**(), and **tan**().