

```
/*
 * Copyright 07 Feb 1995 Sun Microsystems, Inc. All Rights Reserved.
 */

#ifndef _MATH_H
#define _MATH_H

#ifdef __cplusplus
extern "C" {
#endif

#pragma ident "@(#)math.h 2.5 95/02/07"

#ifdef _POSIX_SOURCE && !defined(_POSIX_C_SOURCE)
#define _POSIX_C_SOURCE 1
#endif

#ifdef __STDC__
#define __P(p) p
#else
#define __P(p) ()
#endif

/*
 * ANSI/POSIX
 */
typedef union _h_val {
    unsigned long _i[2];
    double _d;
} _h_val;

#ifdef __STDC__
extern const _h_val __huge_val;
#else
extern _h_val __huge_val;
#endif

#define HUGE_VAL __huge_val._d

#ifdef __EXTENSIONS__ || __STDC__ - 0 == 0 && \
    !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE)
/*
 * SVID & X/Open
 */
#define M_E 2.7182818284590452354
#define M_LOG2E 1.4426950408889634074
#define M_LOG10E 0.43429448190325182765
#define M_LN2 0.69314718055994530942
#define M_LN10 2.30258509299404568402
#define M_PI 3.14159265358979323846
#define M_PI_2 1.57079632679489661923
#define M_PI_4 0.78539816339744830962
#define M_1_PI 0.31830988618379067154
#define M_2_PI 0.63661977236758134308
#define M_2_SQRTPI 1.12837916709551257390
#define M_SQRT2 1.41421356237309504880
#define M_SQRT1_2 0.70710678118654752440

extern int signgam;

#define MAXFLOAT ((float)3.40282346638528860e+38)

#ifdef __EXTENSIONS__ || !defined(_XOPEN_SOURCE)
/*
 * SVID
 */
enum version {libm_ieee = -1, c_issue_4, ansi_1, strict_ansi};

#ifdef __STDC__
extern const enum version _lib_version;
#else
extern enum version _lib_version;
#endif
#endif

```

```
struct exception {
    int type;
    char *name;
    double arg1;
    double arg2;
    double retval;
};

#define HUGE          MAXFLOAT

#define _ABS(x)       ((x) < 0 ? -(x) : (x))

#define _REDUCE(TYPE, X, XN, C1, C2)    { \
    double x1 = (double)(TYPE)X, x2 = X - x1; \
    X = x1 - (XN) * (C1); X += x2; X -= (XN) * (C2); }

#define DOMAIN      1
#define SING        2
#define OVERFLOW    3
#define UNDERFLOW  4
#define TLOSS       5
#define PLOSS       6

#define _POLY1(x, c)    ((c)[0] * (x) + (c)[1])
#define _POLY2(x, c)    (_POLY1((x), (c)) * (x) + (c)[2])
#define _POLY3(x, c)    (_POLY2((x), (c)) * (x) + (c)[3])
#define _POLY4(x, c)    (_POLY3((x), (c)) * (x) + (c)[4])
#define _POLY5(x, c)    (_POLY4((x), (c)) * (x) + (c)[5])
#define _POLY6(x, c)    (_POLY5((x), (c)) * (x) + (c)[6])
#define _POLY7(x, c)    (_POLY6((x), (c)) * (x) + (c)[7])
#define _POLY8(x, c)    (_POLY7((x), (c)) * (x) + (c)[8])
#define _POLY9(x, c)    (_POLY8((x), (c)) * (x) + (c)[9])
#endif /* defined(__EXTENSIONS__) || !defined(_XOPEN_SOURCE) */
#endif /* defined(__EXTENSIONS__) || __STDC__ - 0 == 0 && ... */

/*
 * ANSI/POSIX
 */
extern double acos __P((double));
extern double asin __P((double));
extern double atan __P((double));
extern double atan2 __P((double, double));
extern double cos __P((double));
extern double sin __P((double));
extern double tan __P((double));

extern double cosh __P((double));
extern double sinh __P((double));
extern double tanh __P((double));

extern double exp __P((double));
extern double frexp __P((double, int *));
extern double ldexp __P((double, int));
extern double log __P((double));
extern double log10 __P((double));
extern double modf __P((double, double *));

extern double pow __P((double, double));
extern double sqrt __P((double));

extern double ceil __P((double));
extern double fabs __P((double));
extern double floor __P((double));
extern double fmod __P((double, double));

#if defined(__EXTENSIONS__) || __STDC__ - 0 == 0 && \
    !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE)
/*
 * SVID & X/Open
 */
extern double erf __P((double));
extern double erfc __P((double));
```

```
extern double gamma __P((double));
extern double hypot __P((double, double));
extern int isnan __P((double));
extern double j0 __P((double));
extern double j1 __P((double));
extern double jn __P((int, double));
extern double lgamma __P((double));
extern double y0 __P((double));
extern double y1 __P((double));
extern double yn __P((int, double));

#if defined(__EXTENSIONS__) || !defined(_XOPEN_SOURCE) || \
    defined(_XOPEN_SOURCE) && _XOPEN_SOURCE_EXTENDED - 0 == 1
/*
 * SVID & XPG 4.2
 */
extern double acosh __P((double));
extern double asinh __P((double));
extern double atanh __P((double));
extern double cbrt __P((double));
extern double logb __P((double));
extern double nextafter __P((double, double));
extern double remainder __P((double, double));
extern double scalb __P((double, double));

/*
 * XPG 4.2
 */
extern double expm1 __P((double));
extern int ilogb __P((double));
extern double loglp __P((double));
extern double rint __P((double));
#endif /* defined(__EXTENSIONS__) || !defined(_XOPEN_SOURCE) || ... */

#if defined(__EXTENSIONS__) || !defined(_XOPEN_SOURCE)
/*
 * SVID
 */
extern int matherr __P((struct exception *));

/*
 * IEEE Test Vector
 */
extern double significand __P((double));

/*
 * Functions callable from C, intended to support IEEE arithmetic.
 */
extern double copysign __P((double, double));
extern double scalbn __P((double, int));

/*
 * Reentrant version of gamma & lgamma; passes signgam back by reference
 * as the second argument; user must allocate space for signgam.
 */
#ifdef _REENTRANT
extern double gamma_r __P((double, int *));
extern double lgamma_r __P((double, int *));
#endif

/*
 * Orphan(s); frexp, ldexp, modf and modff are part of libc nowadays.
 */
extern float modff __P((float, float *));

#include <floatingpoint.h>
#endif /* defined(__EXTENSIONS__) || !defined(_XOPEN_SOURCE) */
#endif /* defined(__EXTENSIONS__) || __STDC__ - 0 == 0 && ... */

#ifdef __cplusplus
}
#endif
```

```
#endif /* !defined(_MATH_H) */
```