Subsections

- Math Functions
- Math Constants

Mathematics: `<math.h>`

Mathematics is relatively straightforward library to use again. You **must** `#include <math.h>` and must **remember** to link in the math library at compilation:

```
cc mathprog.c -o mathprog -lm
```

A common source of error is in forgetting to include the `<math.h>` file (and yes experienced programmers make this error also). Unfortunately the C compiler does not help much. Consider:

```
double x;
x = sqrt(63.9);
```

Having not seen the prototype for `sqrt` the compiler (by default) assumes that the function returns an `int` and converts the value to a `double` with meaningless results.

Math Functions

Below we list some common math functions. Apart from the note above they should be easy to use and we have already used some in previous examples. We give no further examples here:

`double acos(double x)` -- Compute arc cosine of x.
`double asin(double x)` -- Compute arc sine of x.
`double atan(double x)` -- Compute arc tangent of x.
`double atan2(double y, double x)` -- Compute arc tangent of y/x, using the signs of both arguments to determine the quadrant of the return value.
`double ceil(double x)` -- Get smallest integral value that exceeds x.
`double cos(double x)` -- Compute cosine of angle in radians.
`double cosh(double x)` -- Compute the hyperbolic cosine of x.
`div_t div(int number, int denom)` -- Divide one integer by another.
`double exp(double x` -- Compute exponential of x
`double fabs (double x )` -- Compute absolute value of x.
`double floor(double x)` -- Get largest integral value less than x.
`double fmod(double x, double y)` -- Divide x by y with integral quotient and return remainder.
`double frexp(double x, int *expptr)` -- Breaks down x into mantissa and exponent of no.
`labs(long n)` -- Find absolute value of long integer n.
`double ldexp(double x, int exp)` -- Reconstructs x out of mantissa and exponent of two.
`ldiv_t ldiv(long number, long denom)` -- Divide one long integer by another.
`double log(double x)` -- Compute log(x).
`double log10 (double x )` -- Compute log to the base 10 of x.
`double modf(double x, double *intptr)` -- Breaks x into fractional and integer parts.
`double pow (double x, double y)` -- Compute x raised to the power y.
`double sin(double x)` -- Compute sine of angle in radians.
`double sinh(double x)` - Compute the hyperbolic sine of x.
`double sqrt(double x)` -- Compute the square root of x.
`void srand(unsigned seed)` -- Set a new seed for the random number generator (rand).
`double tan(double x)` -- Compute tangent of angle in radians.
`double tanh(double x)` -- Compute the hyperbolic tangent of x.

Math Constants

The `math.h` library defines many (often neglected) constants. It is always advisable to use these definitions:

`HUGE` -- The maximum value of a single-precision floating-point number.
`M_E` -- The base of natural logarithms (e).

`M_LOG2E` -- The base-2 logarithm of e.

`M_LOG10E` - The base-10 logarithm of e.

`M_LN2` -- The natural logarithm of 2.

`M_LN10` -- The natural logarithm of 10.

`M_PI` -- π.

`M_PI_2` -- $\pi/2$.

`M_PI_4` -- $\pi/4$.
`M_1_PI` -- $1/\pi$.

`M_2_PI` -- $2/\pi$.

`M_2_SQRTPI` -- $2/\sqrt{\pi}$.

`M_SQRT2` -- The positive square root of 2.

`M_SQRT1_2` -- The positive square root of 1/2.

`MAXFLOAT` -- The maximum value of a non-infinite single- precision floating point number.

`HUGE_VAL` -- positive infinity.

There are also a number a machine dependent values defined in `#include <value.h>` -- see `man value` or list `value.h` for further details.

Dave Marshall
1/5/1999