
*Full credit is given to the above companies including the OS that this PDF file was generated!*

## *Red Hat Enterprise Linux Release 9.2 Manual Pages on 'erf.3p' command*

***$ man erf.3p***

ERF(3P) POSIX Programmer's Manual ERF(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

erf, erff, erfl ? error functions

SYNOPSIS

```
#include <math.h>
double erf(double x);
float erff(float x);
long double erfl(long double x);
```

DESCRIPTION

The functionality described on this reference page is aligned with the ISO C standard. Any conflict between the requirements described here and the ISO C standard is unintentional. This volume of POSIX.1?2017 defers to the ISO C standard.

These functions shall compute the error function of their argument x, defined as:

__?e^ ?t^2 dt

An application wishing to check for error situations should set errno to zero and call feclearexcept(FE_ALL_EXCEPT) before calling these

functions. On return, if errno is non-zero or fetestexcept(FE_INVALID |

FE_DIVBYZERO | FE_OVERFLOW | FE_UNDERFLOW) is non-zero, an error has

occurred.

RETURN VALUE

Upon successful completion, these functions shall return the value of

the error function.

If x is NaN, a NaN shall be returned.

If x is ?0, ?0 shall be returned.

If x is ?Inf, ?1 shall be returned.

If the correct value would cause underflow, a range error may occur,

and erf(), erff(), and erfl() shall return an implementation-defined

value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, re?

spectively.

If the IEC 60559 Floating-Point option is supported, 2 * x/sqrt(?)

should be returned.

ERRORS

These functions may fail if:

Range Error The result underflows.

If the integer expression (math_errhandling & MATH_ERRNO)

is non-zero, then errno shall be set to [ERANGE]. If the

integer expression (math_errhandling & MATH_ERREXCEPT) is

non-zero, then the underflow floating-point exception shall

be raised.

The following sections are informative.

EXAMPLES

Computing the Probability for a Normal Variate

This example shows how to use erf() to compute the probability that a

normal variate assumes a value in the range [x1,x2] with x1?x2.

This example uses the constant M_SQRT1_2 which is part of the XSI op?

tion.

```
#include <math.h>
double
Phi(const double x1, const double x2)
```

```
{
    return ( erf(x2*M_SQRT1_2) - erf(x1*M_SQRT1_2) ) / 2;
}
```

APPLICATION USAGE

Underflow occurs when |x| < DBL_MIN * (sqrt(?)/2).

On error, the expressions (math_errhandling & MATH_ERRNO) and (math_er?rhandling & MATH_ERREXCEPT) are independent of each other, but at least one of them must be non-zero.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

erfc(), feclearexcept(), fetestexcept(), isnan()

The Base Definitions volume of POSIX.1?2017, Section 4.20, Treatment of Error Conditions for Mathematical Functions, <math.h>

COPYRIGHT