$$\operatorname{erfcw}(z) = \exp[-z^2]\operatorname{erfc}(-iz),$$

where

$$\operatorname{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_z^\infty \exp[-t^2]dt.$$

The function is computed with accuracy of 0.5e-10. It is particularly useful for large |z| where the computation of erfc(*z*) starts encountering numerical instability.

References

1. http://en.wikipedia.org/wiki/Error_function

2. W. Gautschi, "*Efficient Computation of the Complex Error Function*", SIAM J. Numer. Anal. Vol. 7, No. 1, March 1970.

See Also

erf, **erfc**, **erfcx**, **inverseErfc**, **dawson**

erfcx

erfcx(*num*)

The erfcx function returns the scaled complementary error function of num:

```
f(x) = exp(x2)*erfc(x)
```

Details

This implementation is for real numbers only. If you need complex input and output, use:

```
erfcx = Faddeeva(ix)
```

This method has poorer accuracy for negative X and a more limited region of numerical validity than this real implementation.

Computing erfcx using the mathematical definition has a very limited range in which it is accurate, or even valid.

References

Our implementation comes from **this Stack Overflow post**.

The code in that post is based on this paper:

M. M. Shepherd and J. G. Laframboise, "*Chebyshev Approximation of (1 + 2 x) exp(x2) erfc x in 0 ≤ x < ∞.*", Mathematics of Computation, Volume 36, No. 153, January 1981, pp. 249-253

See Also

erf, **erfc**, **erfcw**, **inverseErfc**, **dawson**, **Faddeeva**

ErrorBars

ErrorBars [*flags*] *traceName*, *mode* [*errorSpecification*]

The ErrorBars operation adds or removes error bars to or from the named trace in the specified graph and modifies error bar settings.

The “error bars” are lines that extend from each data point to “caps”. The length of the line (or “bar”) is usually used to bracket a measured value by the amount of uncertainty, or “error” in the measurement.

Parameters

traceName is the name of a trace on a graph (see **Trace Names** on page II-282).

A string containing *traceName* can be used with the $ operator to specify *traceName*.

mode is one of the following keywords:

OFF	No error bars.
X	Horizontal error bars only.
Y	Vertical error bars only.
XY	Horizontal and vertical error bars.
BOX [=*fillColor*]	Box error bars, optionally with fill color expressed as (r,g,b) or (r,g,b,a). If no color is specified, boxes are not filled. The *fillColor* parameter was added in Igor Pro 8.00.

ELLIPSE={mode, p, alpha}

Plots error ellipses.

You must provide error values via a three-column wave using the ewave=*ew* error specification described below.

mode=0: *ew* contains the standard deviation in X, the standard deviation in Y, and the correlation between X and Y.

mode=1: *ew* contains the variance in X, the variance in Y, and the covariance of X and Y.

p is the probability level represented by the error ellipses. *p*=0.6837, *p*=0.95, and *p*=0.997 respresent one, two, and three standard deviations respectively. Larger values of *p* result in larger ellipses corresponding to higher confidence levels.

alpha is an opacity value in the range of 0 (fully transparent) to 65535 (fully opaque). See **Error Ellipse Color** on page II-305 for details.

The ellipse mode was added in Igor Pro 9.00.

See **Error Ellipses** on page II-305 for further discussion.

NOCHANGE — This mode allows you to programmatically change aspects of error bars using flags (see ***Flags*** below) without affecting the mode or error specification.

The NOCHANGE mode was added in Igor Pro 9.00.

SHADE={*options*, *fillMode*, *fgColor*, *bkColor* [, *negFillMode*, *negFgColor*, *negBkColor*]}

SHADE was added in Igor Pro 7.00.

options is reserved for future use and must be zero.

fillMode sets the fill pattern.

n=0:	No fill.
n=1:	Erase.
n=2:	Solid black.
n=3:	75% gray.
n=4:	50% gray.
n=5:	25% gray.
n>=6:	See **Fill Patterns** on page III-498.

fgColor is (r,g,b) or (r,g,b,a). If all zeros including alpha, i.e., (0,0,0,0), then the actual color will be the trace color with an alpha of 20000.

bkColor is used for patterns only and can be simply (0,0,0) for solid fills.

negFillMode is the same as *fillMode* but for negative error shading.

negFgColor is the same as fgColor but for negative error shading.

negBkColor is the same as bkColor but for negative error shading.

The *errorSpecification* , described below, affects only the Y amplitude of error shading.

The X values of the trace to which shading is applied must be monotonic. Results with non-monotonic X values are undefined.

See **Error Shading** on page II-305 for more information and examples. See **Color Blending** on page III-498 for information on the alpha color parameter.

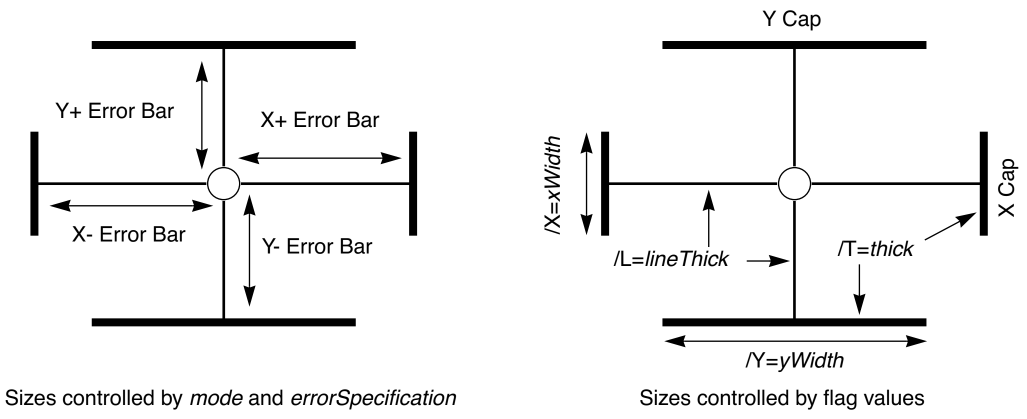
Error Specification Parameters

For any mode other than OFF and NOCHANGE, you must provide an error specification which consists of a keyword followed by zero or more parameters:

pct=*value*	*value* is the length of an error bar expressed as a percentage of the trace X or Y value.
sqrt	The error bar length is the square root of the trace X or Y value.
const=*value*	*value* is the length of an error bar independent of the trace X or Y value.
ewave=ew	*ew* is a three-column wave with error ellipse parameters. Each row represents information for one trace data point. The interpretation of the columns of ew depends on the mode parameter provided with the ELLIPSE keyword: *mode*=0: *ew* contains the standard deviation in X, the standard deviation in Y, and the correlation between X and Y. *mode*=1: *ew* contains the variance in X, the variance in Y, and the covariance of X and Y. ew may include a subrange specification as long as it results in effectively a 2D wave with three columns and a row for each trace data point. See **Subrange Display Syntax** on page II-321. See **Error Ellipses** on page II-305 for further discussion.
wave=(*w1*,*w2*)	Each error bar length is obtained from the corresponding point of wave *w1* for negative bars and from wave *w2* for positive bars. If you omit *w1* no negative bars are drawn. If you omit *w2* no positive bars are drawn. You can use subrange syntax for *w1* and *w2*. See **Subrange Display Syntax** on page II-321.
mulWave = *ew*	Each error bar length is computed from a multiplier taken from the corresponding point of a wave *ew*. If your trace wave is tw, the error bar lengths are computed as: positive error bar length = tw * (ew - 1) negative error bar length = tw * (1 - 1/ew) This is appropriate if you have computed geometric mean and standard deviation, as would be appropriate for errors with lognormal distribution. Such error bars appear symmetrical on a log axis. The values in *e* must be greater than or equal to 1. Values less than 1 result in the error bar not be displayed for that data point. An error value of exactly 1 results in a zero-length error bar. The mulWave keyword was added in Igor Pro 9.00.
nochange	Don't change *errorSpecification* from existing values. nochange allows you to, for instance, change the fill color for the box mode without having to repeat *errorSpecification* for the X and Y errors. nochange was in Igor Pro 8.00.

See the ***Examples*** below for examples using *mode* and *errorSpecification*.

mode and *errorSpecification* control only the lengths of the horizontal and vertical lines (the “bars”) to the “caps”. All other sizes and thicknesses are controlled by the flags.



XY *mode* Error Bars

Flags

/CLIP=*clip* /CLIP={*clipH*, *clipV*}	Sets a distance in points by which error bars are allowed to extend beyond the plot area. If you use the first form, the vertical and horizontal values are both set to clip. Use the second form to set horizontal and vertical clipping independently. clipH and clipV default to 2 points. Values may be negative to restrict error bar drawing to an area inside the plot area. /CLIP was added in Igor Pro 9.00.
/L=*lineThick*	Specifies the thickness of both the X and Y error bars drawn from the point on the wave to the caps. If you use the ELLIPSE keyword to draw error ellipses, lineThick sets the thickness of the ellipse outline.
/RGB=*strokeColor*	Sets the color of the lines used to draw the error bars. *strokeColor* is expressed as (r,g,b) or (r,g,b,a). The default color, used if you omit /RGB, is the same as the color of the trace to which the error bars are attached. /RGB was added in Igor Pro 8.00. If you use the ELLIPSE keyword, strokeColor sets the color of the ellipse outline.
/T=*thick*	Specifies the thickness of both the X and Y error bar "caps". In box mode, /T sets the thickness of the box outline.
/W=*winName*	Changes error bars in the named graph window or subwindow. When omitted, action will affect the active window or subwindow. This must be the first flag specified when used in a Proc or Macro or on the command line. When identifying a subwindow with *winName*, see **Subwindow Syntax** on page III-92 for details on forming the window hierarchy.
/X=*xWidth*	Specifies the width (height, actually) of the caps to the left or right of the point.
/Y=*yWidth*	Specifies the width of the caps above or below the point.

The thicknesses and widths are in units of points. The thickness parameters need not be integers. Although only integral thicknesses can be displayed exactly on a standard resolution screen, nonintegral thicknesses are produced properly on high resolution devices. Use /T=0 to completely suppress the caps and /L=0 to completely suppress the lines to the caps.

Details

If a wave specifying error values for *traceName* is shorter than the wave displayed by *traceName* then the last value of the error wave is used for the unavailable points. If a point in an error wave contains NaN (Not a Number) then the half-bar associated with that point is not shown.

Examples

```
// X 10% of wave1, Y is 5% of wave1
ErrorBars wave1, XY pct=10, pct=5

// X error bars only, square root of wave1
ErrorBars wave1,X sqrt
```