Parameter w[0] sets the vertical offset, w[1] sets the peak area, w[2] sets the location of the peak, w[3] gives the Gaussian component's full width at half max and w[4] is the ratio of the Lorentzian width to the Gaussian width.

After the fit, assuming you used a coefficient wave named voigtCoefs, you can calculate the width of the full Voigt peak as follows:

```
Variable/G wl = voigtCoefs[4]*voigtCoefs[3]
Variable/G wg = voigtCoefs[3]
Variable wv = wl/2 + sqrt( wl^2/4 + wg^2)
```

References

The code used to compute the VoigtFunc was written by Steven G. Johnson of MIT. You can learn more about it at http://ab-initio.mit.edu/Faddeeva.

See Also

VoigtPeak, **Faddeeva**, **Built-in Curve Fitting Functions** on page III-206

VoigtPeak

VoigtPeak(*w*,*x*)

The VoigtPeak function returns a value from a Voigt peak shape defined by coefficients in wave *w* at location *x*. It was added in Igor Pro 8.00.

The Voigt peak shape is defined as a convolution of a Gaussian and a Lorentzian peak. We use an approximation that is described by the author as having "accuracy typically at least 13 significant digits". This function is equivalent to the built-in Voigt fitting function. See **Built-in Curve Fitting Functions** on page III-206.

The coefficients are:

w[0]:	Vertical offset.
w[1]:	Peak area.
w[2]:	Peak center location.
w[3]:	Gaussian component width expressed as Full Width at Half Max (FWHM).
w[4]:	Ratio of Lorentzian component width to the Gaussian component width. For w[4]=0, the peak shape is purely Gaussian, as w[4] → ∞, the peak shape become purely Lorentzian. A value of 1 results in Gaussian and Lorentzian components of equal width.

References

The code used to compute VoigtPeak was written by Steven G. Johnson of MIT. You can learn more about it at http://ab-initio.mit.edu/Faddeeva.

See Also

VoigtFunc, **Faddeeva**, **Built-in Curve Fitting Functions** on page III-206.

WAVE

WAVE [**/C**][**/T**][**/WAVE**][**/DF**][**/Z**][**/ZZ**][**/SDFR**=*dfr*] ***localName*** [**=*pathToWave***][, ***localName1*** [**=*pathToWave1***]]…

WAVE is a declaration that identifies the nature of a user-defined function parameter or creates a local reference to a wave accessed in the body of a user-defined function.

The optional parameter /SDFR flag and *pathToWave* parameter are used only in the body of a function, not in a parameter declaration.

The WAVE declaration is required when you use a wave in an assignment statement in a function. At compile time, the WAVE statement specifies that the local name references a wave. At runtime, it makes the connection between the local name and the actual wave. For this connection to be made, the wave must exist when the WAVE statement is executed.

The WAVE declaration is also required if you use a wave name as a parameter to an operation or function if **rtGlobals**=3 is in effect which is the usual case.