

Igor Technical Notes

WaveMetrics Technical Support

#020-A: Custom Peak Measurement and Fitting

Written by:

James Prouty

Mar 1992

Modified:

James Prouty

Sep 1993

This revision addresses Igor Pro compatibility.

We started with the TN020 Peak Measurement experiment and created the Custom Peak Measurement experiment. The Custom Peak Measurement experiment is easier to modify for your own purposes than the TN020 Peak Measurement Experiment.

The Custom Peak Measurement experiment fits the Voigt Profile to peak data. See TN026, “The Voigt Profile” for more information. You can either use the Custom Peak Measurement experiment to fit to Voigt profiles or you can modify it to fit a different profile to your peaks.

This Technical Note explains how to use and modify the Custom Peak Measurement experiment. To show you how to modify it, we step you through the process we went through to change the Custom Peak Measurement experiment into the Template Peak Measurement experiment, which is also included as part of TN020-A. **Please read TN020, “Peak Measurement and Fitting” before making your own modified version of the Custom Peak Measurement experiment.**

The Template Peak Measurement experiment fits an arbitrary peak to peak data. A template of the peak is stored in the XY wave pair W_PeakTemplateX and W_PeakTemplate. Interpolated and scaled versions of the template peak are fitted to the data. The template originally contains a raised cosine peak. Though we don’t show you how, you can change the contents and size of the template. The necessary changes are in the same areas that are addressed in our example modification.

Accompanying files:

‘Custom Peak Measurement’	--	Sample experiment containing Voigt Peak. This is the experiment to modify if you are fitting to a new function.
‘Template Peak Measurement’	--	Sample experiment containing Template Peak. This currently fits to a sampled raised cosine peak.
‘TN020-A Voigt TEXT’	--	File containing macro to initialize the Voigt function’s global waves, and the Voigt function itself.

Supporting files (from the ‘Sample XOPs Folder’ of the ‘Igor Extras Disk’):

‘FindPeaks’	--	XOP containing the BoxSmooth, FindAPeak, and EstimatePeakSizes operations.
-------------	----	--

#020-A: Custom Peak Measurement and Fitting
'Execute'

2 of 18

-- XOP containing the Execute operation. **The 'Execute' XOP is not needed and is not compatible with Igor Pro.**

Related files:

'TN020 Peak Measurement'

-- Sample experiment (Fits Gaussian and Lorentzian peaks).

'TN020-B Peak Areas'

-- Sample experiment (baseline fit, piecewise-linear baseline estimation, measures many peaks, no peak fitting).

The Custom Peak Measurement experiment is intended to inspire the engineers and scientists to consider Igor for analyzing simple spectra (between 1 and about 9 peaks, depending on your patience and your machine's prowess).

The Custom Peak Measurement experiment features:

- Full support of equally-spaced or X-Y data representation.
- Baseline (background) estimation and removal.
 - Any built-in Igor function except gaussian or lorentzian may be fit to the baseline at multiple areas indicated with the cursors.
- Peak Fitting
 - Manual or automatic initial guesses ("Peak Identification").
 - Simultaneous fit to single or multiple Voigt Peaks plus an optional baseline function.
 - Display of individual fit peaks.
 - Report of amplitudes, positions, FWHMs, and approximate areas of fit peaks.

Choosing the Right Experiment

- Use TN020's Peak Measurement experiment for fitting to Gaussian or Lorentzian profiles.
- Use TN020-A's Custom Peak Measurement for fitting to the Voigt Profile, or as a starting point for your own peak profile function.
- Use TN020-A's Template Peak Measurement for fitting to a raised cosine profile, or as a starting point for your own peak template. This is useful when there is no closed-form equation for your peak.

Custom Peak Measurement Requirements

The experiments require Igor 1.24 or later. At least 1 megabyte of memory should be allocated to Igor 1.2. If you use color, allocate at least 1.5 megabytes. A floating point coprocessor is highly recommended.

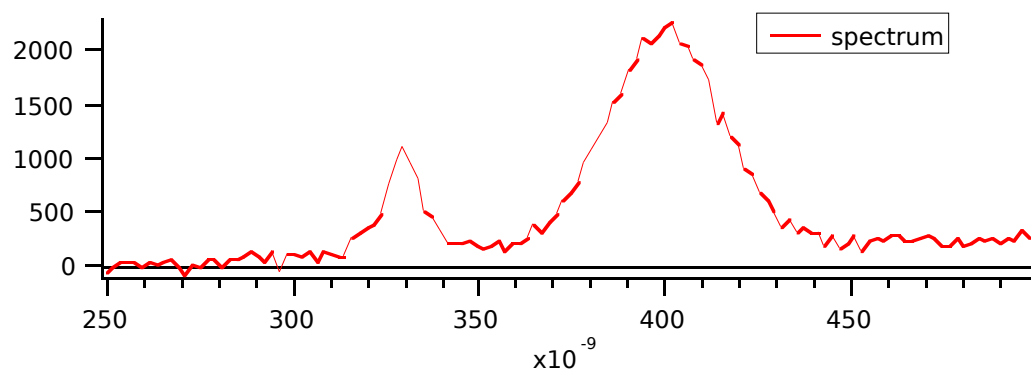
Installing the Supporting Files

Igor Pro users must copy the 'FindPeaks' XOP into the 'Igor Extensions' folder which can be found in the folder containing the 'Igor Pro' application.

Igor 1.2 users must copy both the 'FindPeaks' and 'Execute' XOPs into the folder containing the 'Igor' application. (Igor 1.2 doesn't use an 'Igor Extensions' folder.)

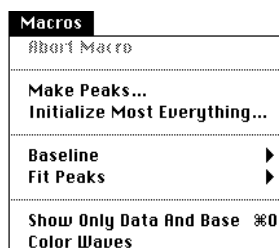
How To Use The Custom Peak Measurement Experiment

After installing the XOP(s), launch Igor. Then open the ‘Custom Peak Measurement’ experiment. A graph with a sample spectrum containing two noisy Voigt profiles on a linear baseline will appear.



You can use this data to become familiar with the experiment, or you can import your own using the standard techniques; see “loading” and “Browse Waves” in the manual’s index. If you import your own data, you should create a graph which displays it as equally-spaced or X-Y data. You might profitably display the background data in the same graph.

Pull down the Macros menu to see what this experiment contains:



Make Peaks...

This macro fabricates Voigt peaks to demonstrate what this experiment can do. The peaks are defined by entering values into four waves which can be edited in the ‘DefinePeaks’ table. If the table isn’t an open window, it may be created with the ‘DefinePeaks’ table macro found in the ‘Window’ menu. Enter the peak center X coordinates, amplitudes, widths at 50% (FWHMs), and shape factors (y). Invoke the ‘Make Peaks...’ macro to choose the other peak parameters (noise level) and wave parameters (range of X coordinates, number of points, etc.). It will graph the resulting wave. If a linear X scale was selected, the data is treated as equally-spaced; if a log X scale, it is treated as X-Y (“parametric”) data.

The Voigt Profile Function Parameters

The Custom Peak Measurement experiment is initially configured with the Voigt profile function. The

Voigt profile is used in spectroscopy when a given line shape is neither a Gaussian nor a Lorentzian but rather is a convolution of the two.

The curve-fitting function used by the experiment is “MyPeak”:

```
Function/D MyPeak(w,x) | Voigt, 4 terms
  Wave/D w
  Variable/D x

  return w[0]*voigt(w[1]*(x-w[2]),w[3])
End
```

Parameter w[0] affects the amplitude, w[1] affects the width, w[2] sets the location of the peak and w[3] is the shape factor y which adjusts the shape (and amplitude). Note that the MyPeak function does not incorporate any constant (“DC”) offset. That is taken care of separately, so that multiple peaks may be fit to the data.

The Voigt profile function header is:

```
Function/D voigt(x,y)
  Variable/D x,y
  ...
```

X is the normalized independent variable, and y is the shape parameter:

$$y = \sqrt{\frac{\lambda v^2}{\pi}} \cdot \frac{\Phi\Omega HM_{\text{λορεντζιαν}}}{\Phi\Omega HM_{\text{Γαουσιαν}}}$$

The Custom Peak Measurement experiment calculates values for w[0], w[1], w[2], and w[3] based on the values stored in the waves shown in the ‘DefinePeaks’ table. The w[i] values are calculated by the MyPeakSizesToCoefs function.

Initialize Most Everything...

This macro announces to the experiment which waves are your spectral data and their representation (equally-spaced or X-Y). Use this macro to analyze new data, or start over with old data.

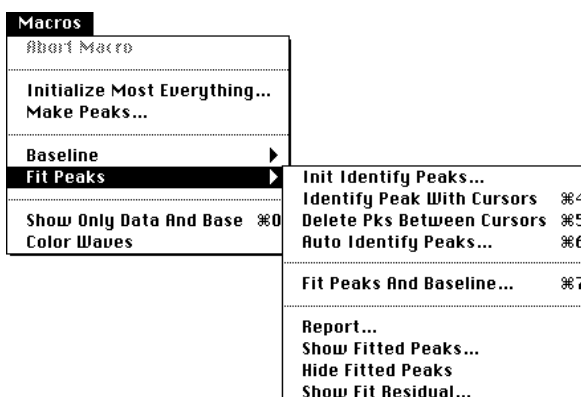
Baseline

This hierarchical menu contains everything related to baseline estimation and removal, and is the same as the Baseline menu in the Peak Measurement experiment:



Fit Peaks

This hierarchical menu contains the macros related to fitting the sum of an optional baseline and one or more peaks simultaneously to the peak data wave. You can employ the cursors to identify the approximate location and size of peaks to provide initial guesses to the peak fit, or you can use the ‘Auto Identify Peaks...’ macro to identify peaks automatically.

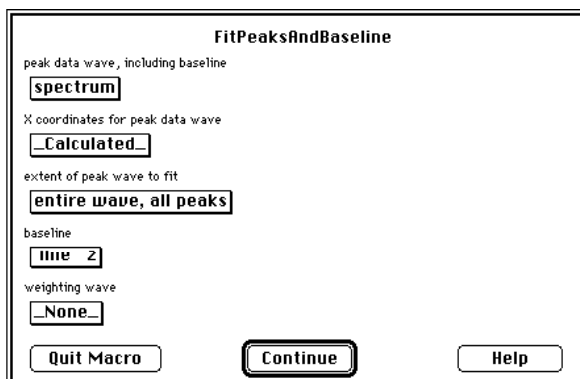


Init Identify Peaks...

This macro initializes the process of “identifying” peaks — manually or automatically — to provide initial guesses to the ‘Fit Peaks and Baseline...’ macro. The output of these four Identify macros is a related set of waves shown in the “Peak Estimates” table. These values are used to generate the initial guesses. The waves in this table describe all the peaks “identified” since the ‘Init Identify Peaks...’ or ‘Initialize Most Everything...’ macro was run, and is elsewhere referred to as “the peak list”.

Fit Peaks and Baseline...

This macro fits the sum of an optional baseline function or wave and identified peaks to the peak data wave:



The output coefficients of the peak fit are stored in two related waves, “W_PeakFitCoefs” and “W_PeakPM”. These are inputs to the ‘Report...’ macro and are used by the ‘Subtract Baseline’ macro when you select “_From Fit Peaks_” as the baseline popup menu selection.

The peak fit also generates an output wave, “W_PeakFit” which represents the fit of the sum of the baseline and peaks.

Since the maximum number of coefficients that can be used in a curve fit is 30, the maximum number of Voigt peaks which may be fit at one time is $(30-2)/4 \approx 7$, providing no baseline fitting is done at the same time. With the worst case baseline (poly 5), $(30-7)/4 \approx 5$ Voigt peaks may be fit concurrently.

WARNING: the peak fitting process can be rather slow when the number of peaks exceeds 5, even on a machine with a math coprocessor (5 peaks, 128 points in peak data wave took 2min 30secs on a Mac IICI).

Problems?

Sometimes fitting the peak data wave results in the very cryptic error message:

Singular matrix or other numerical error

The usual cause for this is fitting a higher polynomial function with X values whose magnitudes are not very close to unity. The data shown in the graph above, for example has X values whose magnitude is about $10e-7$. Trying to fit a third order polynomial (“poly 4”) will result in this error message.

To fit a poly 4 function, the data must be rescaled. We suggest keeping the magnitude of the X values between 10^{-3} and 10^{+3} .

When using an X coordinates wave, it is as simple as entering at the command line:

```
spectrumX *= 1e9
```

which rescales, in this case, from meters to nanometers.

When using calculated X scaling, use the SetScale command:

```
SetScale x, leftx(spectrum)*1e9, rightx(spectrum)*1e9, spectrum
```

Another common cause is an incorrectly implemented MyPeak function, such that varying one of the values in the coefficients wave has no effect on the returned value.

Yet another problem is one of the initial guesses being zero, which doesn't work.

Report...

This macro interprets the output of the ‘Fit Peaks and Baseline...’ macro, and creates a report of the fitted peaks. You choose whether the table of peak parameters is sorted by amplitude, center location, width (FWHM), or area. It does not sort by the shape factor y . The baseline coefficients are displayed in the report if you choose “include” from the “baseline info” popup menu. The graph showing the peak data and sum of baseline and peaks (the graph whose name – not title – is PeakFitGraph) is also displayed in the report.

Show Fitted Peaks...

This macro shows the amplitude, width, center position, area and shape factor of each peak fitted by the ‘Fit Peaks And Baseline...’ macro.

One popup menu selects whether replicas of each the fitted peaks are appended to the target graph, and another popup selects whether each peak's center, width, and area are annotated with a tag attached to the peak data wave.

Hide Fitted Peaks and Show Fit Residual

See TN020, “Peak Measurement and Fitting” for description.

Show Only Data and Base and Color Waves

See TN020, “Peak Measurement and Fitting” for description.

Modifications You Can Make to Fit to Your Special Peak

Okay, suppose your peaks are not Gaussian, Lorentzian, or Voigt. We’ll show you how to modify a copy of the Custom Peak Measurement experiment to fit to another type of peak. Follow along as we replace the Voigt peak with a tabulated peak function, or “template”:

$$y = K0 * W_PkT(K1*(x-K2))$$

where W_PkT is a wave containing a representative sampled peak.

The result of our modifications will be the Template Peak Measurement experiment. Your modifications will differ, and you should save your changed experiment under a different name.

NOTE: These modifications require changing the experiment’s procedure text. Comments containing “MyPeak” are peppered throughout the text where changes need to be made. You can use the “Find…” dialog to locate these occurrences, or follow the steps below. **You should be experienced in programming with Igor’s macro language before attempting this.**

1. Create a copy of the Custom Peak Measurement Experiment

Open the Custom Peak Experiment, and select ‘Save As…’ from the ‘File’ menu. Save the experiment under a new name. We chose “Template Peak Measurement”, but you should use another name.

2. Modify the ‘Initialize Most Everything’ Macro

Any global initialization should be performed here. For the Voigt peaks, we had added the following line:

```
Mk("W_MyPkCoefs",4) | MyPeak cw[0,3]
```

The wave named W_MyPkCoefs is required, and **its length must match the number of coefficients required for the peak function**. The experiment relies on the size of W_MyPkCoefs with statements such as:

```
Variable sz = numpnts(W_MyPkCoefs)
```

For the Voigt function, 4 terms were required, but for the template peak, we only use 3 terms (K0, K1, and K2):

```
Mk("W_MyPkCoefs",3) | MyPeak cw[0,2]
```

This is also a good place to create and initialize the template wave (or it could be loaded from a data file containing a representative peak). We added the following lines:

```

| Build XY Template
Variable n=101
Mk("W_PkT",n);Mk("W_PkTx",n);
SetScale/I x,-Pi,Pi,W_PkT,W_PkTx
W_PkTx= x
W_PkT=1+cos(x)           | Raised Cosine
S_PkT="Hanning"          | Peak type
| Template built, measure it
WaveStats/Q W_PkT
V_Tamp=1;W_PkT/=V_max    | this line normalizes template
V_Tamp=V_max             | this line doesn't
V_Tfwhm=FwhmTmpl()
V_Tarea=AreaXYTPt(W_PkTx,W_PkT,0,n-1)
V_TmaxX=W_PkTx[x2pnt(W_PkT,V_maxloc)]

```

The function `FwhmTmpl` assumes only that the peak center is at $X = 0$. We inserted the `FwhmTmpl`, `AreaXYTPt`, and the `BinaryLvlSearch` functions into the procedure window. See the actual Template Peak Measurement experiment for the implementation of these functions.

We also executed the following lines to create the global variables and strings used here. `V_Tamp`, `V_Tfwhm`, `V_Tarea`, and `V_TmaxX` are the amplitude, full width at half max, the area, and the maxima X location of the peak template. `S_PkT` will contain the peak type for use in reports and annotated peak tags:

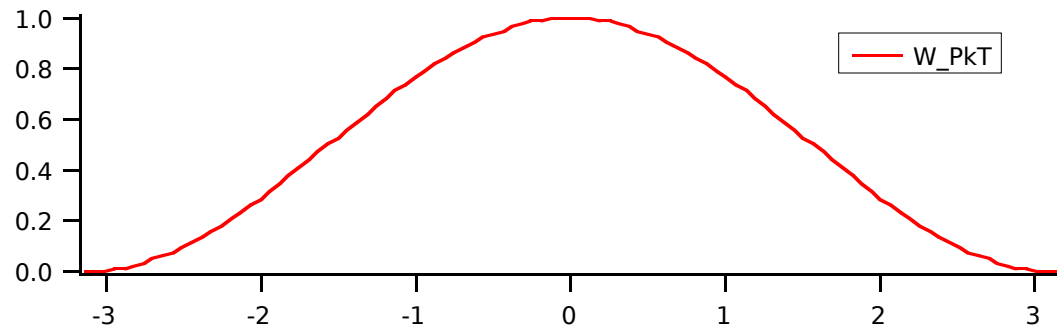
```

Variable/D/G V_Tamp,V_Tfwhm,V_Tarea,V_TmaxX
String/G S_PkT

```

Run the 'Initialize Most Everything' macro, and then execute:

Display `W_PkT` vs `W_PkTx`



3. Modify Function/D MyPeak

This is the function called to fit a single peak. The constant term is not included. For the Voigt profile, `MyPeak` was written as:

```

Function/D MyPeak(w,x) | Voigt, 4 terms
Wave/D w
Variable/D x

return w[0]*voigt(w[1]*(x-w[2]),w[3])
End

```

For the template peak, we will scale the amplitude and width, and offset the center using:

```
Function/D MyPeak(w,x) | Template, 3 terms
  Wave/D w
  Variable/D x
  x=limit(w[1]*(x-w[2]),W_PkTx[0],W_PkTx[numpts(W_PkTx)-1])
  return w[0]*interp(x,W_PkTx,W_PkT)
End
```

Interpolation is necessary to prevent a singular matrix from occurring during the curve fitting process; the returned value must vary when any of the input parameters change by a very small amount. If the function had used:

```
return w[0]*W_PkT[x2pnt(W_PkT,x)]
```

the output would not vary for small changes in $w[1]$ or $w[2]$ (the same template value would be returned as long as $x2pnt$ selected the same template wave y value), and a singular matrix would have occurred. I know, because I tried it!

4. Modify Function MyPeakSizesToCoefs

This function has two purposes. It is called by the MakePeaks macro to create peaks that conform to the requested peak center, amplitude, width, etc. It is also used by the FitPeaksAndBaseline macro to compute initial guesses for the curve fit of the peaks based on the peak estimates.

MyPeakSizesToCoefs converts peak center, amplitude, FWHM (full width at half maximum), and possibly other parameters into coefficients that when passed to the MyPeak function result in a peak that ideally fits the measured or desired peak.

For instance, for the Voigt profile:

```
Function MyPeakSizesToCoefs(ctr,amp,fwhm,y,cw) |output cw, for initial guesses
  Variable/D ctr,amp,fwhm,y
  Wave/D cw
  y=y+(y==0)*0.05 | initial guess of zero not allowed
  cw[0]=amp/voigt(0,y)
  cw[1]=1/fwhm*(y+sqrt(y*y+4*ln(2))) | approx
  cw[2]=ctr
  cw[3]=y
  return 4 | #of cw terms
End
```

The cw values are computed so that:

```
W_PeakFit= MyPeak(cw,x)
```

results in the wave $W_PeakFit$ containing a peak that has a peak amplitude of amp , is centered at ctr , has a width of $fwhm$, a shape factor of y , and a constant (or “DC”) term of zero. Note the fussing about with variable y . The ‘Identify Peaks’ macros provide estimates only for ctr , amp , and $fwhm$. Y is estimated to be zero, **but zero is not a legal initial guess for a user-defined curve fit**. The “ $y=y+ . . .$ ” statement replaces a zero value of y with 0.05, which **is** a legal initial guess.

For our the template peak, the function is:

```
Function MyPeakSizesToCoefs(ctr,amp,fwhm,cw) |output cw, for initial guesses
  Variable/D ctr,amp,fwhm
  Wave/D cw
  cw[0]=amp/V_Tamp
  cw[1]=V_Tfwhm/fwhm | exact
  cw[2]=ctr
  return 3      | #of cw terms
End
```

Note that MyPeakSizesToCoefs no longer expects or uses the y argument, that the width calculation is now exact (rather than the previous approximation), and that the return value is now 3.

5. Modify the MakePeaks and FitPeaksAndBaseline Macros

These macros call the MyPeakSizesToCoefs function we just changed.

For the Voigt profile, part of the MakePeaks macro was:

```
iterate (npks)
  n= MyPeakSizesToCoefs($ctrX[i],$ampY[i],$widthX[i],W_My[i],$cw)
  $py+=MyPeak($cw,$px[p])
loop
```

The Voigt peak required the shape parameter, which was supplied by the wave named “W_My”. This wave, and the parameter y it supplied, are no longer needed. Removing the W_My[i] parameter gives:

```
iterate (npks)
  n= MyPeakSizesToCoefs($ctrX[i],$ampY[i],$widthX[i],$cw)
  $py+=MyPeak($cw,$px[p])
loop
```

For the Voigt profile, part of the FitPeaksAndBaseline macro was:

```
Mk(tw0,sz)
n= MyPeakSizesToCoefs($ctrsX[i],$ampsY[i],$widsX[i],W_EstY[i],$tw0)
$pcw[st,st+sz-1]=$tw0[p-st]
```

Again, we eliminate the extra y term in the call to MyPeakSizesToCoefs, which was supplied by the wave named “W_EstY”:

```
Mk(tw0,sz)
n= MyPeakSizesToCoefs($ctrsX[i],$ampsY[i],$widsX[i],$tw0)
$pcw[st,st+sz-1]=$tw0[p-st]
```

6. Modify the Macros that refer to waves W_My, W_EstY, W_PkY

W_My was used to define the Voigt shape factor for the MakePeaks macro.

W_EstY was part of the peak list maintained by the Identify Peaks macros.

W_PkY was an output of the Report macro.

None of these waves are needed in the Template Peak Experiment, so we will be removing them from the experiment. However, do **not** modify references to the W_MyPkCoefs wave!

Note: If your peak requires 4 parameters, you can re-use these waves for the peak's fourth parameter. If your peak requires 5 or more parameters, you will need to modify the macros to create and use additional waves. Search the procedure window for "W_My", "W_EstY", and "W_PkY" and add the necessary code.

"W_My" shows up in the **DefinePeaks** the table macro; it can be deleted:

```
Window DefinePeaks() : Table
  PauseUpdate; Silent 1          | building window...
  Edit W_MakeCentersX.y,W_MakeAmpsY.y,W_MakeWidthsX.y as "DefinePeaks"
EndMacro
```

Remove W_My from the table window, if it exists.

"W_EstY" is created in the **InitIdentifyPeaks** macro, and used in the macros **IdentifyPeakWithCursors**, **DeletePksBetweenCursors**, **AutoIdentifyPeaks**, and **MergeEdges** and the in the table macro **PeakEstimates**. W_EstY can be deleted from all of them.

"W_PkY" is created in the **Report** macro, and used in the **PeakReportTable** macro; it can be deleted from these, too.

Execute the following command to delete these obsolete waves:

```
KillWaves/F W_My,W_EstY,W_PkY
```

7. Modify Function MyCoefsToPeakSizes

This function is called by the Report and ShowFittedPeaks macros to compute the peak center, amplitude, FWHM (Full Width at Half Maximum), and area from the peak coefficients. It returns the results in global variables. The macros place the results in a table, or in annotations of the fitted peaks.

For instance, for the Voigt profile:

```
Function MyCoefsToPeakSizes(cw)      | input cw, global outputs |MyPeak
  Wave/D cw
  |Variable/G/D V_ctr,V_amp,V_fwhm,V_area
  V_ctr=cw[2]
  V_amp=cw[0]*voigt(0,cw[3])
  V_fwhm=1/cw[1]*(cw[3]+sqrt(cw[3]*cw[3]+4*ln(2))) | +/- 1%
  V_area=cw[0]*sqrt(pi)/cw[1]
  return 0
End
```

```
Function MyCoefsToPeakSizes(cw) | input cw, global outputs |MyPeak
    Wave/D cw
    V_ctr=cw[2]
    V_amp=cw[0]*V_Tamp
    V_fwhm=V_Tfwhm/cw[1] | exact
    V_area=cw[0]/cw[1]*V_TArea
    return 0
End
```

8. Remove Function/D voigt

We no longer need the voigt function. Delete the voigt function from the procedure window.

9. Modify the Report and ShowFittedPeaks Macros

Hunt down these changes by searching for “Voigt”. Change them to use the global string S_PkT, which contains the name of your peak.

In Report:

```
text+=" Voigt Peaks" | MyPeak
```

becomes:

```
text+=" "+S_PkT+" Peaks" | MyPeak
```

and in **ShowFittedPeaks**:

```
if($pdcw[5]==7) |MyPeak
    typ="\CVoigt\r"
    Mk(tw2,sz)
    $tw2=$tcw[st+p]
    Variable n=MyCoefsToPeakSizes($tw2)
    y=$tcw[st+3]
else
    Abort "Unsupported peak type"
endif
sprintf text,"\L\Z09amp=%g\rcenter=%g\rwidth(+/-1%%)=%g\rarea=%g\ry=%g",
    V_amp,V_ctr,V_fwhm,V_area,y
```

becomes:

```
if($pdcw[5]==7) |MyPeak
    typ="\C"+S_PkT+"\r"
    Mk(tw2,sz)
    $tw2=$tcw[st+p]
    Variable n=MyCoefsToPeakSizes($tw2)
else
    Abort "Unsupported peak type"
endif
sprintf text,"\L\Z09amp=%g\rcenter=%g\rwidth=%g\rarea=%g", V_amp,V_ctr,V_fwhm,V_area
```

Note that we deleted the assignment of *y*, and its use in the `sprintf` statement. Since width is an exact calculation for our template peak, the +/-1% comment is also gone from the `sprintf` statement (which is actually one line in the procedure window; not two lines as shown here). The variable *y* is no longer used in `ShowFittedPeaks`, and its definition in the `ShowFittedPeaks` macro can be removed. After that is done the line reads:

```
Variable s=$pdcw[1],en=$pdcw[2],npks=$pdcw[0]-1,terms
```

10. Test the Modifications

The experiment should be ready. Close the procedure window. Fix any compilation errors, such as undefined global variables and waves in functions. Create them (as double precision is best). Select the 'PeakFitGraph' window. If you've made any other changes to the 'Initialize Most Everything' macro, run it again.

Then select the 'DefinePeaks' table, and enter parameters for one peak in the first row (Point 0), and set the remaining rows of `W_MakeCentersX` equal to NaN (empty) by selecting the cell(s), then double clicking the edit area, press the Delete key, and then the Return or Enter key:

Point	W_MakeCentersX.y	W_MakeAmps.y	W_MakeWidthsX.y
0	3.3e-07	1000	1e-08
1		0	0
2			

Run the 'MakePeaks' Macro, and enter start and end values that bracket the defined peak. Add a small amount of noise (0.1% is enough) so the curve fit will converge faster:

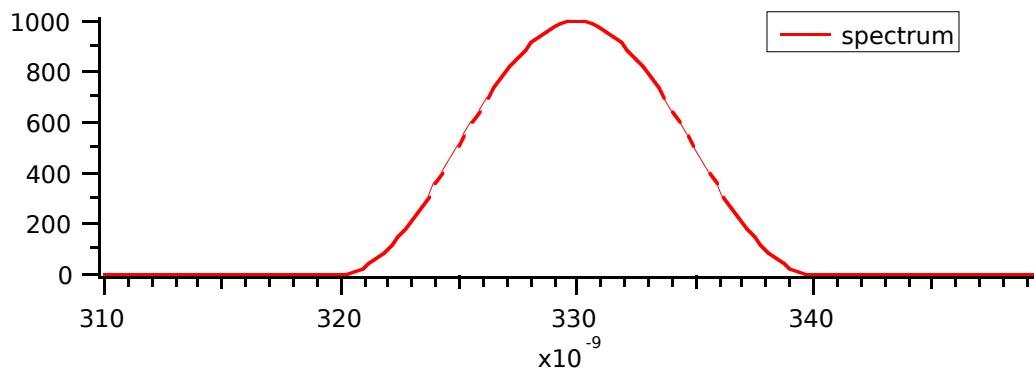
MakePeaks

Output wave name: peak wave noise (enose val):

starting x value: ending x value:

x wave increment: points in output waves:

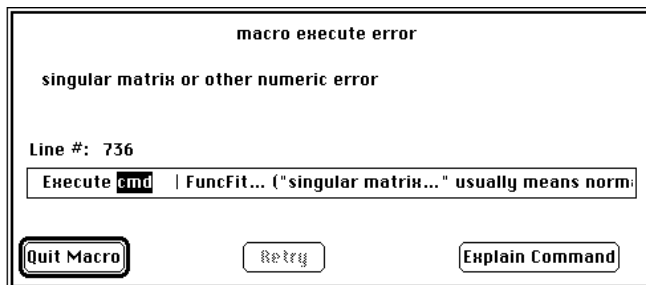
The result in our case:



Verify that the amplitude, center, and width are what you expect.

Run the **'InitIdentifyPeaks'** and **'AutoIdentifyPeaks'** macros, and the **'ShowOnlyDataAndBase'** macros.

Run the **'FitPeaksAndBaseline'** macro. If you get the following dialog:



it usually means that one of the functions was incorrectly modified, or that the value of MyPeak doesn't change when one of the coefficients is varied, or one of the initial guesses was zero. See the discussion related to "Function/D MyPeak" and interpolation on page 8, and "Problems?" on pages 5 and 6.

Also, check that the coefficients are getting set correctly by examining W_PM and W_PeakFitCoefs:

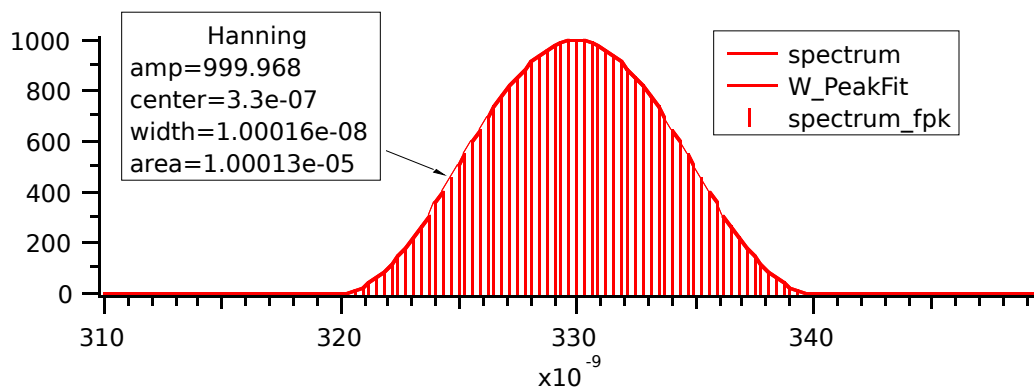
Edit W_PM, W_PeakFitCoefs

See the procedure window for a description of the format of W_PM and W_PeakFitCoefs.

If the fitting takes many passes, such as when "40 iterations with no convergence" is printed to the history window, there might be too little noise on the peak for the curve-fitting algorithm to find the minimum chi-square value. Try 'MakePeaks' again with more noise. If this doesn't correct the problem, look for other causes as described above.

Run the **'Report'** macro. Since there is only one peak, select ascending sort, or else the peak information will be sorted to the second row, and the first row will be empty. Check the center, amplitude, width, and area values shown in the PeakReportTable and Peak Report Layout for correctness. Adjust the table to show the desired number of columns and rows. Update the Layout window macro by closing the Layout window and selecting "Replace" from the resulting dialog.

Run the **'ShowFittedPeaks'** macro with both tags and peaks turned on:



Check that the center, amplitude, width, and area values shown in the tag are correct. Remember that the added noise will distort the measurements slightly.

11. You're Done!

If you get to this point, your experiment has been successfully changed to incorporate your peak. Clean up the windows, then save the experiment as stationery so that this valuable piece of work does not get undone. Make a backup copy on a floppy, too!

Waves and their Uses

spectrum	Sample spectra data, aka “the peak data wave”.
spectrumX	X coordinates of sample spectra data.
W_BaseCoefs	Baseline coefficients from ‘Fit Baseline at Regions’, this is passed to the PolyMorph function. This is paired with W_BasePM. See the procedure window for a description of the format of the coefficients. Note that, for implementation reasons, W_BaseCoefs[0] is NaN.
W_BaselineFit	Output wave of ‘Fit Baseline at Regions...’, this is that macro’s idea of the best fit of the baseline to the peak data wave at the regions identified by “W_BaseRegion”, or the whole wave. It has the same scaling and number of points as the peak data wave. After it is created, it becomes the default baseline wave.
W_BasePM	Baseline coefficients descriptor for PolyMorph (“PM”) from ‘Fit Baseline at Regions’, this is copied to the global wave W_PM for access by the PolyMorph function. This is paired with W_BaseCoefs.
W_BaseRegion	Annotation and weighting wave used in identifying regions of the peak data wave as belonging to the baseline of the data. Used by most of the ‘Baseline’ macros.
W_EstAmpsY	Output of the ‘Identify Peaks’ macros, input to ‘Fit Peaks and Baseline...’. Contains the estimated peak amplitudes, corrected for any baseline. Part of the “peak list”.
W_EstCentersP	Output of the ‘Identify Peaks’ macros, input to ‘Fit Peaks and Baseline...’. Contains the estimated peak centers, in point coordinates. Part of the “peak list”.
W_EstCentersX	Output of the ‘Identify Peaks’ macros, input to ‘Fit Peaks and Baseline...’. Contains the estimated peak centers, in X coordinates. Part of the “peak list”.

W_EstEdgesP	Output of the ‘Identify Peaks’ macros, input to ‘Fit Peaks and Baseline...’. Contains the estimated peak edge locations, in point coordinates. Part of the “peak list”, this wave is twice as long as the other W_Est... waves.
W_EstWidthsX	Output of the ‘Identify Peaks’ macros, input to ‘Fit Peaks and Baseline...’. Contains the estimated peak widths at 50% amplitude, in X coordinates. Part of the “peak list”.

W_MakeAmpsY	Input to the ‘Make Peaks...’ macro. Contains the desired peak amplitudes.
W_MakeCentersX	Input to the ‘Make Peaks...’ macro. Contains the desired peak center location, in X coordinates.
W_MakeWidthsX	Input to the ‘Make Peaks...’ macro. Contains the desired peak widths at 50%, in X coordinates.
W_PeakFit	Output of the ‘Fit Peaks and Baseline...’ macro, this is that macros idea of the best fit of the sum of the baseline and peaks to the peak data wave. It has the same scaling and number of points as the peak data wave.
W_PeakFitCoefs	Peak and baseline coefficients from ‘Fit Peaks And Baseline...’, this is passed to the PolyMorph function. This is paired with W_PeakPM. See the procedure window for a description of the format of the coefficients. Note that, for implementation reasons, W_PeakFitCoefs[0] is NaN.
W_PeakPM	Peak and baseline coefficients descriptor for PolyMorph (“PM”) from ‘Fit Peaks And Baseline...’, this is copied to the global wave W_PM for access by the PolyMorph function. This is paired with W_PeakFitCoefs.
W_PkAmps	Output of the ‘Report...’ macro, it contains the amplitudes of the peaks fit to the peak data wave by the ‘Fit Peaks And Baseline...’ macro, according to the values found in the “W_PeakFitCoefs” and “W_PeakPM” waves.
W_PkAreas	Output of the ‘Report...’ macro, it contains the areas of the peaks fit to the peak data wave by the ‘Fit Peaks And Baseline...’ macro, according to the values found in the “W_PeakFitCoefs” and “W_PeakPM” waves.
W_PkCenters	Output of the ‘Report...’ macro, it contains the center locations in X coordinates of the peaks fit to the peak data wave by the ‘Fit Peaks And Baseline...’ macro, according to the values found in the “W_PeakFitCoefs” and “W_PeakPM” waves.
W_PkFWHMs	Output of the ‘Report...’ macro, it contains the Full Width at Half Maximums of the peaks fit to the peak data wave by the ‘Fit Peaks And Baseline...’ macro, according to the values found in the “W_PeakFitCoefs” and “W_PeakPM” waves.
W_PM	Function coefficients descriptor used by PolyMorph (“PM”)

from 'Fit Peaks And Baseline...', this is usually a copy of either "W_BasePM" or "W_PeakPM".

W_ResidBaseX Annotation wave for 'Show Fit Residual...' when invoked from the 'Baseline' hierarchical menu, the X coordinates of the standard deviation lines.

W_ResidBaseY Annotation wave for 'Show Fit Residual...' when invoked from the 'Baseline' hierarchical menu, the y coordinates of the standard deviation lines.

W_ResidPeakX	Annotation wave for ‘Show Fit Residual...’ when invoked from the ‘Fit Peaks’ hierarchical menu, the X coordinates of the standard deviation lines.
W_ResidPeakY	Annotation wave for ‘Show Fit Residual...’ when invoked from the ‘Fit Peaks’ hierarchical menu, the y coordinates of the standard deviation lines.
W_ShowEstWidthsX	Annotation wave shows the estimated widths of identified peaks (X coordinates), used with the ‘Identify’ macros.
W_ShowEstWidthsY	Annotation wave shows the estimated widths of identified peaks (y coordinates), used with the ‘Identify’ macros.
W_ShowMinPkLvl	Annotation waves shows the where the minimum level intersects the peak data wave, used with the ‘Auto Identify Peaks...’ macro.
W_sigma	Curve fit output wave. See the Igor manual’s “Analysis” section.
W_tmp,W_tmp0	Temporary working waves.
W_tmp1,W_tmp2	Temporary working waves.

Waves specific to the Custom Peak Measurement Experiment (as configured for the Voigt Profile)

W_My	Input to the ‘Make Peaks...’ macro. Contains the shape factor $y = \sqrt{\ln(2)} * \text{Lorentzian width} / \text{Gaussian width}$.
W_EstY	Output of the ‘Identify Peaks’ macros, input to ‘Fit Peaks and Baseline...’. Contains the estimated shape factor y. Part of the “peak list”.
W_PkY	Output of the ‘Report...’ macro, it contains the shape factors y of the peaks fit to the peak data wave by the ‘Fit Peaks And Baseline...’ macro, according to the values found in the “W_PeakFitCoefs” and “W_PeakPM” waves.

Waves specific to the Template Peak Measurement Experiment

W_PkT	Peak Template Y values
W_PkTx	Peak Template X values
W_NBY2,W_RI	Voigt Profile waves. See [2].
W_HN,W_XN,W_YN	”
W_D4,W_D3	”

Revisions:

Custom v1.3	Fixed bug in ShowFittedPeaks that caused the fitted peak waves to be drawn in the wrong location (7/15/93)
Template v1.3	Added Initialize Template macro.

References:

- [1] TN020: Peak Measurement and Fitting
- [2] TN026: The Voigt Profile
- [3] Interpolate XOP (in the 'Sample XOPs' folder on the 'Igor Extras' disk)
- [4] Drayson, S.R. , Rapid Computation of the Voigt Profile, *J. Quant. Spectrosc. Radiat. Transfer*, Vol 16, pp. 611-614,1976
- [5] Armstrong, B.H. , Spectrum Line Profiles: The Voigt Function, *J. Quant. Spectrosc. Radiat. Transfer*, Vol 7, pp. 61-88,1967
- [6] Whiting, E.E. , An Empirical Approximation To The Voigt Profile, *J. Quant. Spectrosc. Radiat. Transfer*, Vol 8, pp. 1379-1384,1968