Igor
Technical Notes

WaveMetrics Technical Support

#020: Peak Measurement and Fitting

Written by: James Prouty
Oct 1990
Modified: James Prouty
Sep 1993

This revision addresses Igor Pro compatibility.

This Technical Note explains how to use the Peak Measurement Experiment. This experiment was created to address some of the common needs of spectroscopists. It might be of use to chromatographers, too.

The experiment requires 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.

This experiment relies on one or two XOP files, 'FindPeaks' and 'Execute' from the 'Sample XOPs Folder' of the 'Igor Extras Disk'.

Accompanying files:

'Peak Measurement'	-- Sample experiment.
'InitPeakMacros'	-- File containing macro to initialize the global variables in the 'Peak Measurement' Experiment (included primarily for purposes of documentation.)

Supporting files (from the 'Sample XOPs Folder' of the 'Igor Extras Disk'.)**:**

'FindPeaks'	-- XOP containing the BoxSmooth, FindAPeak, and EstimatePeakSizes operations.
'Execute'	-- XOP containing the Execute operation. **The 'Execute' XOP is not needed and is not compatible with Igor Pro**.

Related files:

'TN020-A Custom Peak Measurement'	-- Sample experiment (Fits Voigt and Template peaks).
'TN020-B Peak Areas'	-- Sample experiment (baseline fit, piecewise-linear baseline estimation, measures many peaks, no peak fitting).

The “Peak Measurement” experiment is intended to inspire the engineers and scientists to consider Igor for analyzing simple spectra. The experiment features:

- Full support of equally-spaced or X-Y data representation
- Baseline (background) estimation and removal

  Any built-in Igor function may be fit to the baseline at multiple areas indicated with the cursors.
- Integration of data over a cursor-specified area

  Rectangular integration
  Trapezoidal integration
  Counts
- Peak Fitting

  Manual or Automatic Initial Guesses (“Peak Identification”).
  Simultaneous fit to Single or Multiple Gaussian or Lorentzian Peaks plus an optional Baseline function.
  Display of individual fit peaks.
  Report of amplitudes, positions, FWHMs, and areas of fit peaks.

We would like to better support spectroscopists and chromatographers. Please give us your comments on how we could make this experiment more useful.

Installing the Supporting Files

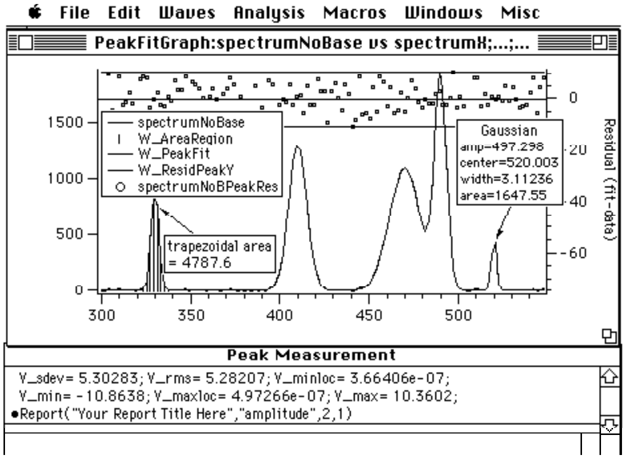
Igor Pro users must copy the ‘FindPeaks’ XOP into the ‘Igor Extensions’ folder which can be found in the folder containing the ‘Igor Pro’ application.

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.)

A Quick Look at the Peak Measurement Experiment

The experiment is thoroughly documented below but to start off, let’s step through a simple example.

After installing the XOP(s), launch Igor. Then open the ‘Peak Measurement’ experiment. At first, you see a sample graph that shows a spectrum with it’s baseline and numerous other waves.



```
 V_sdev= 5.30283; V_rms= 5.28207; V_minloc= 3.66406e-07;
 V_min= -10.8638; V_maxloc= 4.97266e-07; V_max= 10.3602;
•Report("Your Report Title Here","amplitude",2,1)
```

If you're running on system with large screen, select ‘Tile’ from the ‘Windows’ menu or arrange the graph and command windows manually.

Run the ‘Initialize Everything’ macro. This lets you identify the wave containing the spectrum's Y values, the wave containing it's X values and the wave containing the baseline, if any.

Select the “spectrum” wave from the “peak data wave, including baseline” popup.

Click ‘Continue’ to accept the parameters shown.

To clean up the graph, run the ‘Show Only Data and Base’ macro. This gets rid of the auxiliary waves, leaving only the spectrum to be examined.

Select ‘Init Baseline Fit’ from the ‘Baseline’ submenu of the ‘Macros’ menu. Click ‘Continue’.

Position the cursors on a region of the spectrum that look like the baseline and run ‘Add Region to Fit’ from the ‘Baseline’ submenu. Repeat this for each of the regions that look like baseline regions.

Select ‘Fit Baseline at Regions’ from the ‘Baseline’ submenu. Click ‘Continue’ to accept the default parameters. This does a fit to the baseline regions and appends a wave showing the resulting baseline.

Select ‘Subtract Baseline’ from the ‘Baseline’ submenu. Click ‘Continue’ to accept the default parameters. This adds a new wave to the graph showing the spectrum with the baseline removed.

Run the ‘Show Only Data and Base’ macro again to get rid of everything but the baseline-subtracted spectrum.

Select ‘Init Identify Peaks’ from the ‘Fit Peaks’ submenu of the ‘Macros’ menu. Click ‘Continue’ to accept the default parameters.

Run the ‘Auto Identify Peaks’ macro from the ‘Fit Peaks’ submenu. Click ‘Continue’ to accept the default parameters.

Run the ‘Fit Peaks and Baseline’ macro from the ‘Fit Peaks’ submenu. Click ‘Continue’ to accept the default parameters. Now sit back and relax. This will take a while.

Finally, run the ‘Show Fitted Peaks’ macro from the ‘Fit Peaks’ submenu.

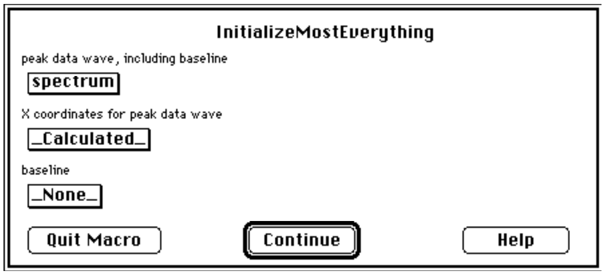
That’s a quick look at the ‘Peak Measurement’ experiment. The rest of this Tech Note describes it in more detail.

Supported Data Types and Formats

The experiment uses double-precision real-valued waves. Your data can be converted to double precision using the ‘Initialize Most Everything’ macro.

The data may be equally-spaced or X-Y pairs.

In Igor-terminology, “equally-spaced” means that the data is represented by a single wave which contains the data y values, which we expect to be spectral amplitudes. The x coordinate for each data y value is calculated from the wave’s x scaling, which may be set with the SetScale command or the ‘Change Wave Scaling…’ menu item. The experiment’s macros have wave list popup menus for “X coordinates for peak data wave”. When the data is equally-spaced, the “_Calculated_” item should be selected:



InitializeMostEverything

peak data wave, including baseline: spectrum

X coordinates for peak data wave: _Calculated_

baseline: _None_

Data which is comprised of X and Y pairs is represented in Igor as a pair of waves. The manual uses the term “parametric” when discussing graphs of this kind of data. This means that corresponding elements of the x wave and the y wave are identified by a common parameter: the point number. With this kind of data, the waves’ x scaling is irrelevant. In this experiment, X-Y data is handled by selecting the wave providing the x values in the “X coordinates for peak data wave” popup menu.

The only waves that appear in the “X coordinates…” popup are those whose names end with “X” or “x”, so the name of your data’s x wave should, too! You can rename a wave by typing “Rename oldWaveName, newWaveName” at the command line.

The utility of the baseline wave is explained below. At this point all we want to point out is that the **waves which appear in baseline popup menus must have the word “base”**

somewhere in their name.

The x, y, and baseline waves must have the same number of points.

Baselines and Backgrounds

Spectra may be considered to be composed of three major components:

1 the peaks, which represent the response or composition of the item under investigation,

2 noise, and

3 the background response, which is the response obtained when the item under investigation is absent.

This experiment refers to this background response as the “baseline”.

If you have collected a background response, it can be subtracted from spectra containing the background and the peak responses, leaving just the peak responses.

If your data contains both peaks and background, this experiment gives you two ways to deal with it:

1 estimate the background and remove it, or

2 fit a baseline function at the same time as fitting peaks to the data.

A few important side notes about baseline waves:

The ‘Peak Measurement’ experiment assumes that baseline waves have the same number of points as the spectral peaks wave.

Equally-spaced data representations also assume that the x-scaling of the baseline wave is the same as the spectral peaks wave.

X-Y data representations assume that the x coordinates for the spectral peaks wave are appropriate for the baseline wave. In fact, the experiment assumes that one x coordinates wave is sufficient for all waves related to, or derived from, the spectral peaks wave.

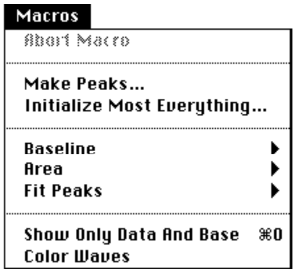
If your data does not conform to these assumptions, you may need to interpolate your other waves using Igor’s “interp” function, or with the Interpolate XOP[3].

Starting To Use The Peak Measurement Experiment

This experiment relies on two XOP files, “FindPeaks” and “Execute” from the ‘Sample XOPs Folder’ of the ‘Igor Extras Disk’. Place these files in the folder which contains the Igor application, launch Igor, and then open the ‘Peak Measurement’ experiment.

A graph with a sample spectrum will appear. You can use this data to become familiar with the experiment, or you can import your own using the standard techniques; use ‘Browse Waves’ or see the manual. 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 spectral peaks to demonstrate what this experiment can do. The peaks are defined by entering values into three waves which can be edited using the ‘DefinePeaks’ table macro under the ‘Window’ menu. Enter the peak’s center x coordinates, their amplitudes, and widths at 50% (FWHMs), then invoke the ‘Make Peaks…’ macro to choose the other peak parameters (peak type, 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.

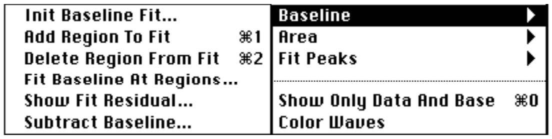
Initialize Most Everything…

This macro announces to the experiment which waves are your spectral data and their representation (equally-spaced or X-Y). The baseline wave is also announced here. See the dialog above. The waves are converted to double precision, various global variables are reset, and results waves are initialized. If the waves are not currently displayed in the target graph, they are appended. If no graph exists, a new one is created.

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:



The Baseline: To Extract or Not To Extract?

Perhaps the simplest way to deal with data containing both peaks and background is to *estimate* the background component by fitting an appropriate function to portions of the spectra which appear to contain only the background. The estimate may be subtracted from the peak data wave, or used only as an initial guess when fitting a sum of peak and background functions to the original spectra.

The Macros in the ‘Baseline’ submenu handle both cases by fitting the standard Igor functions to the baseline at ***regions*** interactively defined by the user, producing a result wave named “W_BaselineFit”.

Init Baseline Fit…

To begin the process of selecting regions of the peak data wave to which the baseline will be fit, select the ‘Init Baseline Fit…’ macro. The “W_BaseRegion” wave will be appended to the target graph, which is expected to contain the peak data wave. The cursors are enabled. You should drag them onto the peak data wave so that the two cursors bracket a “region” which you consider be the baseline (“background”) of the spectra.

Add Region To Fit

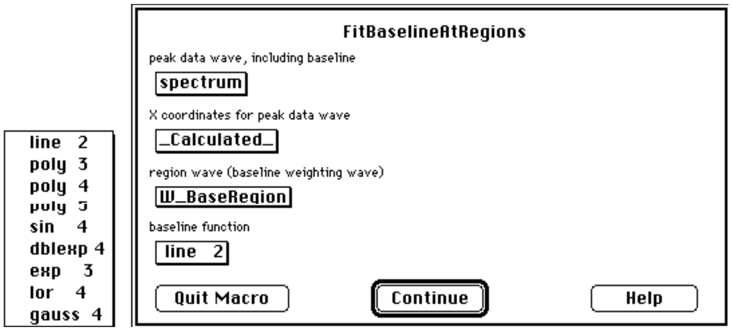
This macro adds the region between the cursors to the portion of the peak data wave to which the baseline will be fit. It can be invoked by typing command-1.

Delete Region From Fit

This macro deletes the region between the cursors from the portion of the peak data wave to which the baseline will be fit. It can be invoked by typing command-2.

Fit Baseline at Regions…

This macro selects a function to be fit to the peak data wave at the regions selected by the two macros above:



FitBaselineAtRegions

peak data wave, including baseline: spectrum

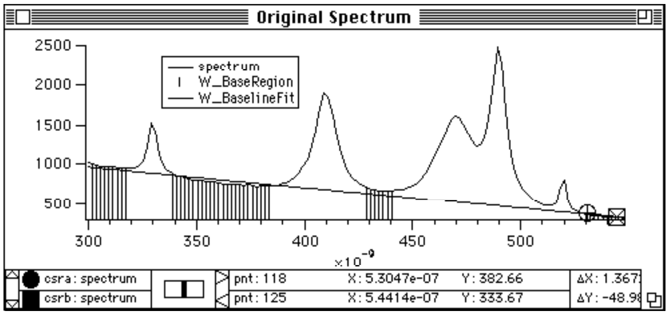
X coordinates for peak data wave: _Calculated_

region wave (baseline weighting wave): W_BaseRegion

baseline function: line 2

(The numbers which follow the function names in the “baseline function” popup menu are used by the experiment’s macros to determine the number of terms the function has; don’t modify the global string which defines that popup menu.)

The resulting fit is used to create the wave “W_BaselineFit”, which is appended to the graph:



```
csra: spectrum   pnt: 118   X: 5.3047e-07   Y: 382.66   ΔX: 1.367
csrb: spectrum   pnt: 125   X: 5.4414e-07   Y: 333.67   ΔY: -48.9
```

‘Fit Baseline’ also stores the results of the fit in a pair of coefficients waves: “W_BaseCoefs” and “W_BasePM” (the “PM” stands for “PolyMorph”, the name of the function used for function fitting in this experiment). When fitting the sum of a baseline function and a number of peak functions to the original data with the ‘Fit Peaks And Baseline…’ macro, these waves will provide the initial baseline guesses if you choose “_From Fit Baseline_” in the baseline popup menu of that macro The waves are also used by the ‘Subtract Baseline…’ macro to compute the baseline when “_From Fit Baseline_” is selected in the baseline popup menu.

Problems?

Sometimes fitting a function to the baseline results in this 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. (The baseline is actually exponential, and fitting to that function works fine, even with this x range.)

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
```

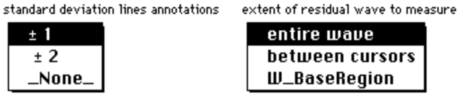
which rescales, in this case, from meters to nanometers.

When using calculated x scaling, use the SetScale command:

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

Show Fit Residual…

This macro displays the difference between the fitted wave and the original data. The difference is displayed against the right axis because of the different vertical scaling required to show the differences. Optional standard deviation lines are displayed if you select them in the “standard deviation lines annotation” popup menu.


standard deviation lines annotations
± 1
± 2
None
extent of residual wave to measure
entire wave
between cursors
W_BaseRegion

Use the “extent” popup menu to limit how much of the difference wave is displayed.

Choosing “W_BaseRegion” limits the residual displayed to those regions that were used to fit the baseline; other points are ignored (set to NaNs). This is the most appropriate setting for the “extent” popup. The other settings are more appropriate when this macro is selected from the ‘Fit Peaks’ hierarchical menu.

Choosing “between cursors” displays the residual only in the region between the two cursors, and choosing “entire wave” displays the residual for, naturally, the entire wave.

Statistics will be printed to the command history for the portions of the residual wave selected by the “extent” popup. The deviation lines are based on those statistics.

Subtract Baseline…

This macro computes a new or selected wave which is the difference between the peak data wave and the baseline wave. The baseline wave is usually "W_BaselineFit", or your acquired background data.

If you select "_From Fit Peaks_" as the baseline, the subtracted baseline wave is computed from the baseline coefficients in the "W_PeakFitCoefs" and "W_PeakPM" waves generated by the 'Fit Peaks And Baseline…' macro.

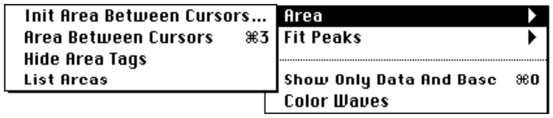
The difference wave is then automatically designated as the new peak data wave, and the experiment is told that there is no longer any baseline (background) wave to take into account during subsequent processing.

Do not subtract the baseline if you intend to use the baseline fit as an initial guess for simultaneous baseline and peak fitting with the 'Fit Peaks And Baseline…' macro later on.

Helpful Hint: To remove waves and stuff that the experiment appends to your graphs, peek ahead to the 'Show Only Data and Base' macro, or simply type command-0 (zero, not oh). Only the newly designated peak wave will be displayed (after you run 'Subtract Baseline…' the experiment declares that there is no baseline wave).

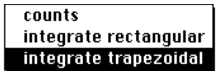
Area

This hierarchical menu contains the macros related to computing areas over a single region of the peak data wave, and the baseline wave if there is one. The cursors are used to select the region.



Init Area Between Cursors…

This macro selects the type of area calculation to be used in subsequent invocations of the 'Area Between Cursors…' macro:



It also gives you the option of having the area calculation(s) displayed on the graph with tags.

Area Between Cursors

This macro computes the area of the region between the cursors using the method selected in 'Init Area Between Cursors…' macro. The result is printed to the command history window, and is stored in the "Areas Between Cursors" table. The macro can be invoked by typing command-3.

If there is a baseline wave, its area is also computed and printed to the command history. The area is subtracted from the peak data wave's and printed to the command history. In this case, the area in the

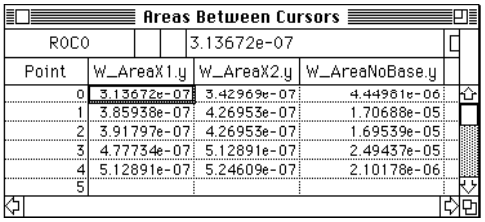
“AreaReport” table and optionally displayed in the graph tag is the baseline-corrected area.

Hide Area Tags

This macro hides (kills, actually) any tags created by the ‘Area Between Cursors’ macro.

List Areas

This is the macro which displays the table showing the current list of areas, and the x ranges over which they were computed:

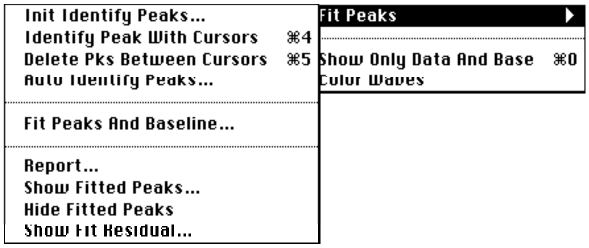


Point	W_AreaX1.y	W_AreaX2.y	W_AreaNoBase.y
0	3.13672e-07	3.42969e-07	4.44981e-06
1	3.85938e-07	4.26953e-07	1.70688e-05
2	3.91797e-07	4.26953e-07	1.69539e-05
3	4.77734e-07	5.12891e-07	2.49437e-05
4	5.12891e-07	5.24609e-07	2.10178e-06
5			

The baseline areas, if any, have been subtracted from the peak areas. The list is sorted by the starting x coordinates stored in “W_AreaX1”.

Fit Peaks

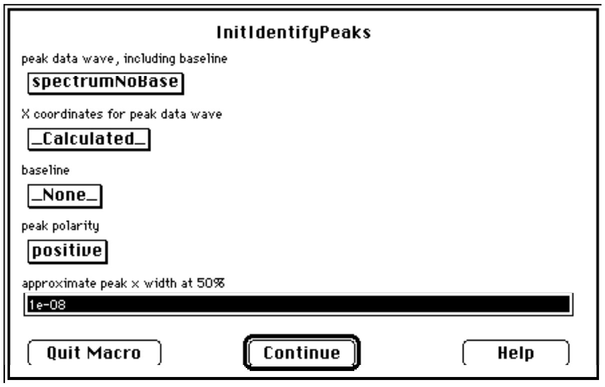
This hierarchical menu contains the macros related to fitting the sum of an optional baseline and one or more peaks simultaneously to the peak data wave. You can employ the cursors to identify the approximate location and size of peaks to provide initial guesses to the peak fit, or you can use the ‘Auto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

Two of the parameters in the ‘Init Identify Peaks’ dialog deserve some explanation:



InitIdentifyPeaks

peak data wave, including baseline: spectrumNoBase

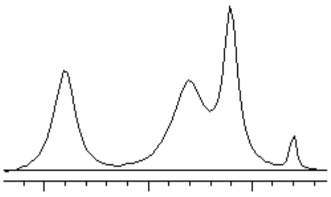
X coordinates for peak data wave: _Calculated_

baseline: _None_

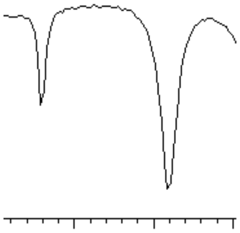
peak polarity: positive

approximate peak x width at 50%: 1e-08

The “peak polarity” popup should be set to describe the type of peaks contained in the peak data wave.



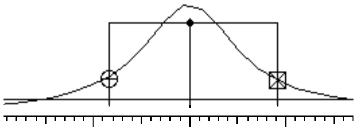
positive pea



negative pe

The “approximate peak x width at 50%” is an estimate of the Full Width at Half Max of the narrowest peak. This value is used to determine the amount of smoothing that may be used on the data without obscuring the narrowest peaks, which suffer the most degradation from smoothing. If noise causes misidentification of a peak’s center or size estimates, try increasing this value. Too large a value will result in low peak amplitude estimates due to excessive smoothing.

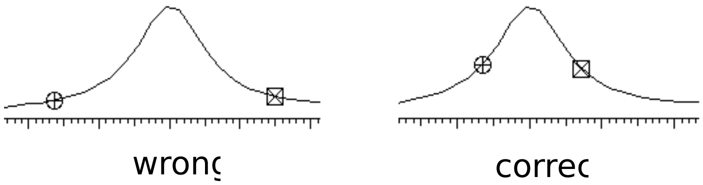
When a peak is identified, the experiment automatically draws annotations which show the peak’s approximate smoothed amplitude (with any baseline amplitude removed), width, and center locations:



This macro also ensures that a graph with the name of “FitPeaksGraph” exists, which displays the peak data wave and baseline selected in this macro’s dialog. This graph is used in the layout created by the ‘Report…’ macro.

Identify Peak With Cursors

This macro identifies the region currently between the cursors as a peak. The center of the peak is determined by the location of the local maxima between the cursors, if “positive” peaks are expected, or the minima location if “negative” peaks are expected. The estimate of peak’s width is the difference between the cursors, so the cursors should be placed at the peak’s 50% points:



This macro can be invoked by typing command-4.

Delete Pks Between Cursors

This macro un-identifies all peaks whose centers are within the region between the cursors. This macro can be invoked by typing command-5.

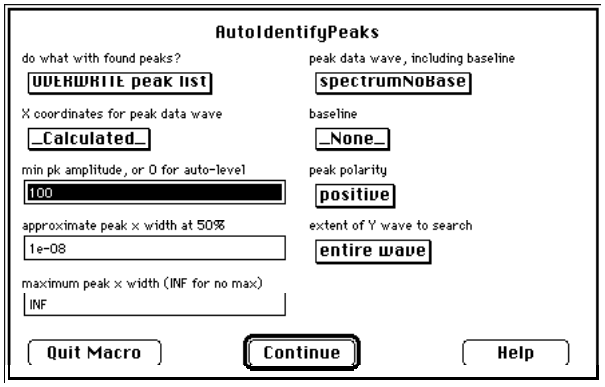
Auto Identify Peaks…

The macro scans the peak data wave and identifies any peaks it finds. A simple smoothed-derivative algorithm is used.

Derivatives are quite sensitive to noise, however, and 'Auto Identify Peaks…' may identify noise spikes as peaks. If your peak data wave is very noisy, it may find many more peaks than actually exist. In fact, it might run virtually forever, finding billions and billions of peaks, if the data is very noisy. In that case, abort the macro with command-period, and either try increasing the value of "approximate peak x width at 50%" to filter out the noise, or use the 'Identify Peak with Cursors" macro to manually identify peaks.

Extra peaks may be removed with the 'Delete Pks Between Cursors' macro.

This macro's dialog has several non-obvious parameters:



AutoIdentifyPeaks

do what with found peaks? OVERWRITE peak list

peak data wave, including baseline: spectrumNoBase

X coordinates for peak data wave: _Calculated_

baseline: _None_

min pk amplitude, or 0 for auto-level: 100

peak polarity: positive

approximate peak x width at 50%: 1e-08

extent of Y wave to search: entire wave

maximum peak x width (INF for no max): INF

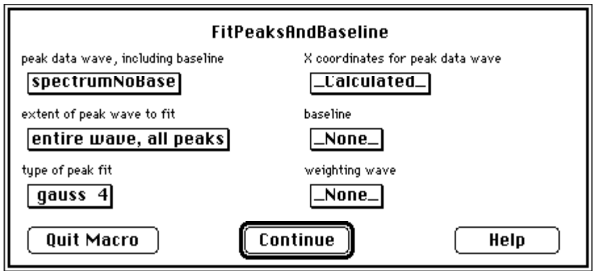
"do what with found peaks?" is a popup menu which allows you to either add the found peaks to the current peak list, or to overwrite the peak list. If you have already manually identified other peaks using the 'Identify Peak With Cursors' macro, set this item to "ADD to existing peak list". Otherwise, choose "OVERWRITE peak list".

"min pk amplitude, or 0 for auto-level" is a threshold value which is used to exclude the usually noisy data near the baseline. The peak finding algorithm excludes peak data values deviation from the baseline is less than this min pk amplitude value. Note that for negative peak data, it is assumed that the baseline is generally above the peak data, with the negative peaks much below the baseline.

"maximum peak x width (INF for no max)" sets an upper bound on the width of identified peaks.

Fit Peaks and Baseline…

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



FitPeaksAndBaseline

peak data wave, including baseline: spectrumNoBase

X coordinates for peak data wave: _Calculated_

extent of peak wave to fit: entire wave, all peaks

baseline: _None_

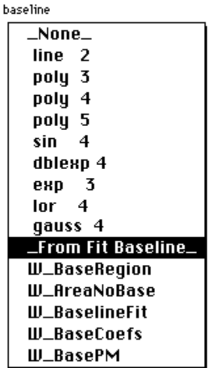
type of peak fit: gauss 4

weighting wave: _None_

"extent of peak wave to fit" is a popup which selects either the entire wave or the region of the wave between the cursors for the fit. If the region is selected, only those peaks whose centers lie between the cursors will be used in the fit. Otherwise, all peaks are used to fit the entire wave.

"type of peak fit" selects whether gaussian or lorentzian peaks are fit to the data. The initial guesses for the peak coefficients are derived from the peak list, which can be viewed in the "PeakEstimates" table.

"baseline" is a popup menu:



"_None_" means that no baseline function is added to the peak functions. Well, actually, a constant K0 term is added, because each peak's individual constant term K0 is factored out. For instance, if you select gaussian peaks, the fit equation will be:

$$W\_PeakFit = K0 \sum_{i=1}^{N} K1_i * \epsilon^{-\left(\frac{x-K2_i}{K3_i}\right)^2}$$

“line 2”, “poly 3”, etc select a built-in function for the baseline fit. For instance, “line 2” adds a K1*x term before the summation symbol in the equation shown above.

"_From Fit Baseline_" selects the baseline fit function selected in 'Fit Baseline At Regions…' macro, and uses the coefficients in "W_BaseCoefs" and "W_BasePM" to provide the initial guess for the baseline fit here.

Choosing an existing wave for a baseline results in fitting to the difference of the peak data wave and the baseline wave; no K0 term is used in the fit (the baseline wave is supposed to remove that).

"weighting wave" is a popup menu which selects an optional curve-fit weighting wave. Selecting "_None_" gives all points equal weighting. If you select a weighting wave, you should be sure that you have set its values, and it must have the same number of points as 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.

If you examine the "W_PeakFitCoefs" wave, you will discover that the first coefficient is NaN (Not a Number). This is not a mistake; the format of this coefficient wave permits fitting a single parameter baseline, K0, when the selected baseline is "_None_". The NaN is a place-holder, since waves have a minimum length of two points.

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 peaks which may be fit at one time is 9, providing no baseline fitting is done at the same time. With the worst case baseline (poly 5), 7 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 co-processor (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 source of this problem is described in the 'Fit Baseline At Regions…' macro section, above.

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. 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, and area 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

This macro removes any waves and tags appended by ‘Show Fitted Peaks…’, and also kills the waves.

Show Fit Residual

This macro displays the difference between the fitted wave and the original data. The difference is displayed against the right axis because of the different vertical scaling required to show the differences. Optional standard deviation lines are displayed if you select them in the “standard deviation lines annotation” popup menu.. (This is the same macro found under the ‘Baseline’ hierarchical menu.)

The “extent” popup menu selection should **not** be “W_BaseRegion”, but rather should be “between cursors” or “entire wave”.

Show Only Data and Base

This macro removes all waves from the target window, except the current peak data and baseline waves. (It also keeps all waves whose names end in “X”, because it is impolite to remove x waves from underneath y waves.) The removed waves are not killed. This macro is invoked by typing command-0 (zero, not oh).

If you have another wave you want to keep on the graph, set the global string g_keep to the name of that wave before invoking the macro.

Color Waves

This macro (which also appears in the ‘Graph Macros’ menu) is used by the experiment to color the waves in each graph to aid in distinguishing one wave from another. It also applies other style preferences such as minor ticks, curve smoothing, and legends.

If you don’t use color with Igor, or prefer a different graph presentation (such as turning off axis standoff), you should change this macro to make waves distinguishable from one another. The experiment calls this macro when it appends a wave, so you’ll get your own presentation.

Modifications You’ll Probably Want to Make

The ‘ColorWaves’ Macro

This macro is invoked when the experiment appends waves to or removes waves from a graph. It sets the color of the waves to help differentiate them. This is, of course, only of benefit to users with color displays. It may be improved adding macro commands to modify the line style of the waves.

Adding line styles is most easily accomplished by creating a graph with many waves (or just display one wave, and append it over and over again to the graph). Manually set the line styles using the ‘Modify Waves Display…’ dialog. Then select ‘Control…’ from the ‘Windows’ menu to create a style macro. Open the procedure window, find the newly created style macro (at the end of the procedure window), and copy the Modify lstyle[i]=n lines into the ‘ColorWaves’ macro. Then delete the style macro you created.

The ‘ColorWaves’ macro is also where those legends get added to the graphs. If they annoy you, remove the Legend command from the macro. The macro also does things like turning on smoothing, minor ticks, etc. Make any changes that please you.

The ‘Make Peaks’ Macro

This macro is really only for demonstration purposes. Just remove the MakePeaks macro, and change the following lines in the procedure window from:

```
Menu "Macros"
   m_mp
   m_so
   "-"
…
```

to:

```
Menu "Macros"
   m_so
   "-"
…
```

Tag texts

The ‘Area Between Cursors’ tags include the method of integration in the tags that annotate the area values. You can remove the text explaining the method by changing the ‘Area Between Cursors’ macro code from:

```
if(mir_anno!=1)
 Tag/C/N=$("area"+num2istr(g_atgn))/A=MB $w,(pnt2x($w,s)+pnt2x($w,en))/2, text+"\r= "+num2str(wi)
…
```

to:

```
if(mir_anno!=1)
 Tag/C/N=$("area"+num2istr(g_atgn))/A=MB $w,(pnt2x($w,s)+pnt2x($w,en))/2, num2str(wi)
…
```

Waves and their Uses

spectrum	Sample spectra data, aka “the peak data wave”.
spectrumX	X coordinates of sample spectra data.
W_AreaNoBase	Area values, less baseline area. Part of the list of areas computed by ‘Area Between Cursors’.
W_AreaRegion	Annotation wave showing which area was measured by ‘Area Between Cursors’.

W_AreaX1 Starting x coordinate of area computation. Part of the list of areas computed by ‘Area Between Cursors’.

W_AreaX2 Ending x coordinate of area computation. Part of the list of areas computed by ‘Area Between Cursors’.

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.

Revisions:

V1.2 Fixed data precision transfer bugs in Proc Dup and Proc SameLen, excessive averaging when displaying Y vs X with X decreasing in AutoIdentifyPeaks. (5/91)

References:

[1] TN002: FWHM & Area for Gaussians Lorentzians
[2] TN018: Area & Integration
[3] Interpolate XOP (in the ‘Sample XOPs’ folder on the ‘Igor Extras’ disk)
[4] TN020-A: Custom Peak Measurement
[5] TN020-B: Peak Areas