EUV IMAGING SPECTROMETER

Hinode

EIS SOFTWARE NOTE No. 17

Version 4.1 22 August 2020

Gaussian fitting examples using the EIS_AUTO_FIT routine

Peter Young
NASA Goddard Space Flight Center
Greenbelt, MD 20771
U.S.A.

This document contains a number of examples for fitting EIS emission lines using the eis_auto_fit suite of routines. The following data-sets are used:

eis_l1_20061209_113031.fits
eis_l1_20070503_050629.fits
eis_l1_20070117_184227.fits

so the user should first download these files and process them with eis_prep:

```
eis_prep, filename, /default, /quiet, /save, /retain
```

Update history

Version 4.1: added Example 10.

Version 4.0: major updates to Examples 7 and 8; added Example 9.

Version 3.3: performed some minor updates to Examples 1, 7 & 8. Updated recommendation for eis_prep.

Version 3.2: modified Example 7 to better explain the offset= output; modified Example 1 to use plot_vel_image.

Version 3.1: added /correct_sensitivity to the recommended call to eis_prep.

Version 3: for many of the examples the routine eis_wave_corr no longer needs to be called, and the offset= input no longer needs to be specified, so these have been removed. Some of the text has been tidied up because of this. The document has been renamed as EIS Software Note #17.

Example 1: basic single Gaussian fit

This is the simplest case of all: fitting a strong line with a single Gaussian.

```
l1name=’eis_l1_20061209_113031.fits’
wd=eis_getwindata(l1name,195.12,/refill)
eis_auto_fit, wd, fit
```

View the fits with:

```
eis_fit_viewer, wd, fit
```

Extract the intensity, velocity and line width arrays and plot them in an IDL window:

```
int=eis_get_fitdata(fit,/int,/map,calib=3)
vel=eis_get_fitdata(fit,/vel,/map)
wid=eis_get_fitdata(fit,/wid,/map)

p=plot_map_obj(int,rgb_table=3,layout=[3,1,1])
q=plot_map_obj(vel,/vel,layout=[3,1,2],/current)
r=plot_map_obj(wid,layout=[3,1,3],/current,rgb_table=4)
```

Note that I use the calib=3 option for eis_get_fitdata, which corresponds to the Warren et al. (2014) intensity calibration. The EIS Wiki has further details about this, in the section about *eis_prep*.

The following examples introduce additional features that often needed to get better fits.

Example 2: single Gaussian, restricted wavelength range

Here there are four lines in the wavelength window and we are going to restrict the wavelength range to only fit one of the lines. The four lines in the window are: Fe XVI λ262.98, S X λ264.23, Fe XIV λ264.79 and Fe XVI λ265.00. The line we are going to fit is the shortest wavelength line, Fe XVI λ262.98,

```
l1name='eis_l1_20070503_050629.fits'
wd=eis_getwindata(l1name,262.98,/refill)
eis_wvl_select, wd, wvl_select
```

At this point a widget pops up. Click on the image and you will see spectrum appear on the right-hand side. Try clicking on various features in the image and you will see how the spectrum changes. For each pixel in the spectrum, a * indicates that that pixel is selected to be included in the fit. To de-select pixels, click-and-drag the cursor on the spectrum. A rubber-band box will appear. When you release the mouse button, all pixels within the X-range of the box will be de-selected (don't worry about the Y-size of the box). For this example click at about 263.5 Å and extend the box to the right-hand edge of the spectrum to de-select all pixels from 263.5 Å upwards. Now exit the widget.

You can view the resulting wvl_select structure by doing:

```
help,wvl_select,/str
```

read the header of the eis_wvl_select routine to find out more information about this structure (if you're interested!).

Now fit the emission line:

```
eis_auto_fit, wd, fit, wvl_select=wvl_select
```

and view the fits:

```
eis_fit_viewer, wd, fit
```

Try selecting the 'Pixel' option and clicking on a few spatial pixels to view the quality of the fits. On the left side of the spectrum plot (bottom left), under the 'X-range options' click on 'Fitted pixels'. The X-range will change to the wavelength region you selected with eis_wvl_select. Clicking on 'Selected line' will show the region ± 0.25 Å around the line.

Example 3: two Gaussian fit

Using the same data window as the last example, we will now perform a two Gaussian fit to Fe XVI λ262.98 and Fe XIV λ264.79. This requires specifying a fit template:

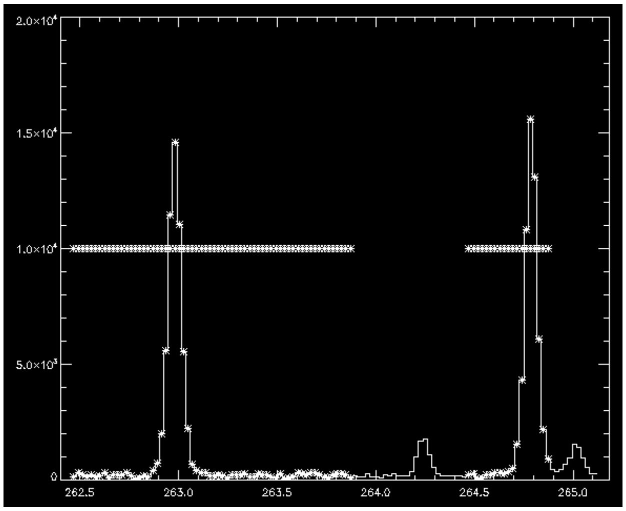
```
eis_fit_template, wd, template
```

A widget similar to eis_wvl_select pops up. On the image, click-and-drag to draw a small rubber-band box somewhere on the image. Typically the user should draw a box about 5-10 pixels on a side. Upon releasing the mouse button a spectrum appears on the right side of the widget. Click on the button 'Choose lines' and click once at the peak of the Fe XVI line (at 263 Å) and once at the peak of the Fe XIV line (264.8 Å). Click on 'End selection', and then click on Exit.

Now choose the wavelength region for the fit:

```
eis_wvl_select, wd, wvl_select
```

The aim is to de-select the S X λ264.23 and Fe XVI λ265.00 lines. See the example below.



Now perform the fit and view the results:

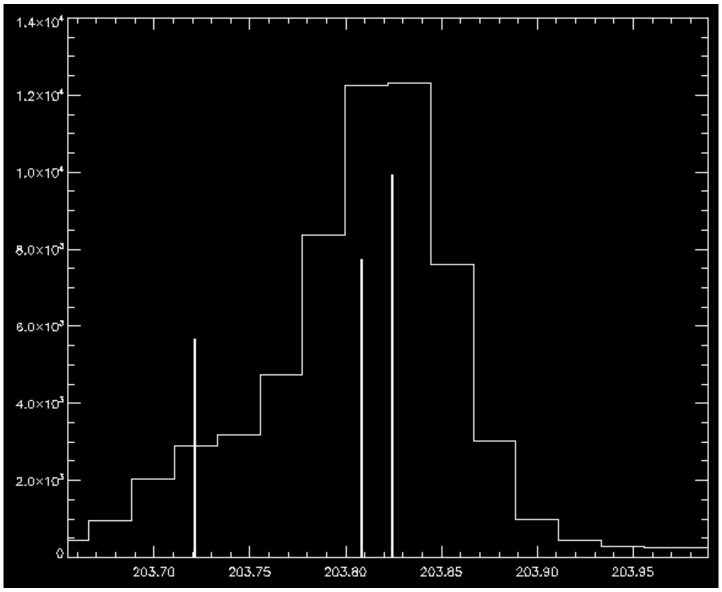
```
eis_auto_fit, wd, fit, wvl_select=wvl_select, template=template
eis_fit_viewer, wd, fit
```

Below the 'Unzoom' button there is now an option to choose which of the two lines' fits you would like to view. Clicking on a pixel in one of the images will show the spectrum fit in the bottom left graphic window. The fit profile for the selected line is shown in white. The full fit function is shown in yellow. Try playing with the X-range options mentioned in Example 2. Under the Y-range options click on 'Selected line' and you will see that the Y-range adjusts to show the selected line better.

Example 4: Fe XIII 203.82 – a parameter-tying example

Fe XIII λ203.82 is an important density diagnostic line, but unfortunately it is partly blended with a Fe XII line. In addition, the Fe XIII line is actually a self-blend. Young et al. (2009, A&A, 495, 587) suggested a prescription for fitting the feature that involves parameter-tying. This can be implemented using the 'template' structure.

```
l1name='eis_l1_20061209_113031.fits'
wd=eis_getwindata(l1name,203.82,/refill)
eis_fit_template,wd,template
```



The figure above shows a suggested template for the fit. I will assume that the three lines are stored in the template file in reverse wavelength order, i.e., line 0 is 203.83, line 1 is 203.79 and line 2 is 203.72.

We now perform parameter-tying using the prescription from Young et al. (2009, A&A, 495, 587)

```
template.lines[1].cen_tie=0
template.lines[1].cen_tie_val=-0.031
template.lines[1].wid_tie=0
template.lines[1].peak_tie=0
template.lines[1].peak_tie_val=0.40
template.lines[2].cen_tie=0
template.lines[2].cen_tie_val=-0.10
template.lines[2].wid_tie=0
```

For the present case the wavelength window is very narrow (16 pixels) and so the background is set to be uniform rather than linear. This is set by doing:

```
template.nback=1
```

i.e., the background is described by 1 parameter rather than 2.

The fit is then performed with:

```
eis_auto_fit, wd, fit, template=template
```

To save the template structure for use with other data-sets in the future do:

```
eis_write_template, ‘template_fe13_203.txt’, template
```

and to read it back into IDL do:

```
template=eis_read_template(‘template_fe13_203.txt’)
```

Example 5: fitting Fe XII λ195.12 + λ195.18

Young et al. (2009, A&A, 495, 587) highlighted the weak blending line on the red side of Fe XII λ195.12 and suggested a prescription for fitting the feature. This example shows how to implement this with eis_auto_fit.

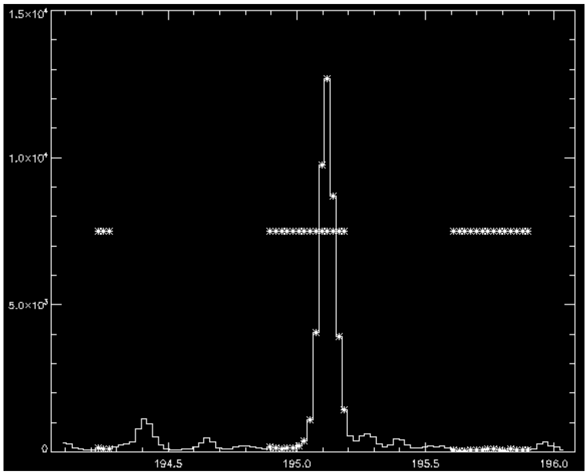
```
l1name=’eis_l1_20070503_050629.fits’
wd=eis_getwindata(l1name,195.12,/refill)
```

Now create a template for the fit. Ignore the other weak lines in the wavelength window – these will be de-selected with eis_wvl_select. Click once near the peak of the strong 195 line, and then click again in the long wavelength wing of the line to represent the weaker 195.18 component. Note that you can’t actually see this line as a distinct feature, it is only apparent as an asymmetry in the 195.12 profile.

```
eis_fit_template, wd, template
```

Now we restrict the wavelength range. The window is very broad and contains several other lines and the pixels I used are indicated in the figure.

```
eis_wvl_select, wd, wvl_select
```



Now we need to tie the line parameters of the 195.18 line to those of 195.12. I assume that 195.12 is represented by Gaussian 0 (i.e., the first line clicked in eis_fit_template):

```
template.lines[1].wid_tie=0
template.lines[1].cen_tie=0
template.lines[1].cen_tie_val=0.06
```

This forces the 195.18 line to have the same width as 195.12 and to have a fixed offset of +0.06 Å relative to it.

We now do the fit:

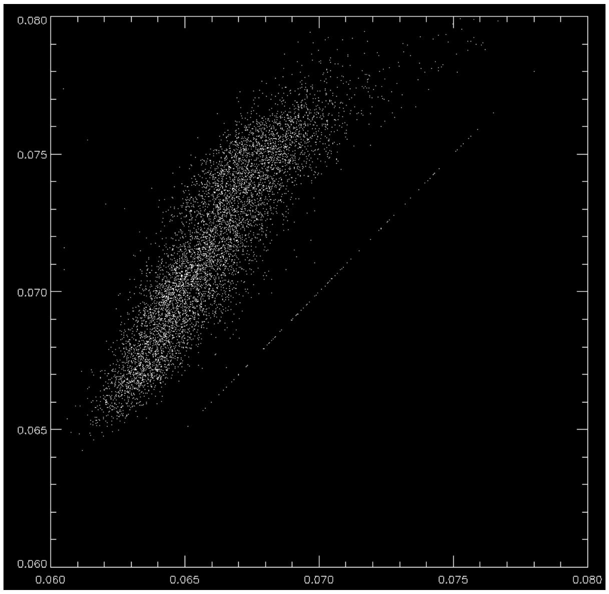
```
eis_auto_fit, wd, fit, wvl_select=wvl_select, template=template
```

We can now compare the width of 195.12 from this two Gaussian fit with that if we'd done a one Gaussian fit. We can use the same 'wvl_select' input, but omit 'template':

```
eis_auto_fit, wd, fit2, wvl_select=wvl_select

wid1=eis_get_fitdata(fit,/wid,line=0)
wid2=eis_get_fitdata(fit2,/wid)

plot,wid1,wid2,psym=3,xra=[0.060,0.080],yra=[0.060,0.080]
```



You will see that the single Gaussian fit widths are larger than those from the two Gaussian fit. Note that there are a few pixels where there is exact agreement. If you inspect the two Gaussian fits:

```
eis_fit_viewer, wd, fit
```

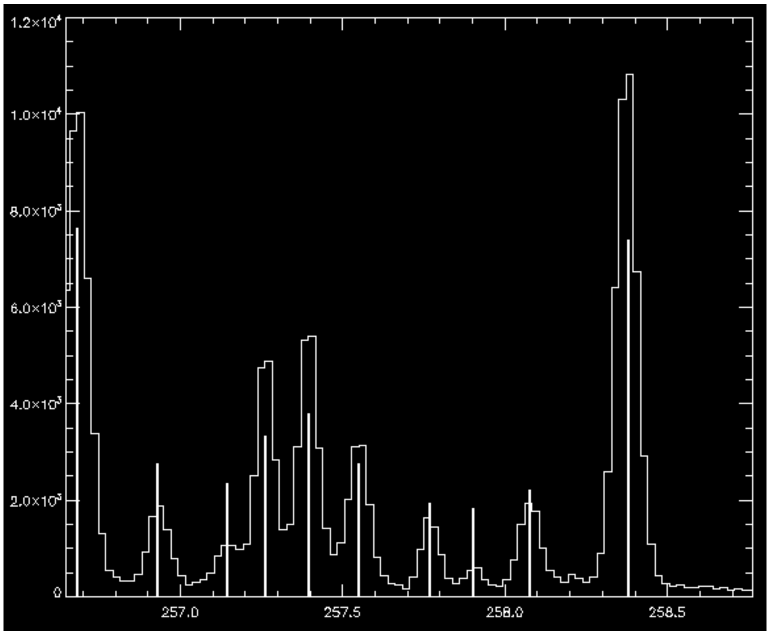
you will find that there are some pixels for which the 195.18 intensity is zero, thus the routine was not able to fit a second Gaussian to the line profile in this case.

Example 6: how many lines can you handle?!?

Here we take an extreme example of a multi-Gaussian fit. 3x3 pixel binning is employed in order to improve photon statistics for the weak lines.

```
l1name=’eis_l1_20070503_050629.fits’
wd=eis_getwindata(l1name, 257.3,/refill)
wd=eis_bin_windata(wd,xbin=3,ybin=3)
eis_fit_template, wd, template
```

After selecting a spatial region you will see there are a large number of lines in this window. We are going to try and automatically fit all of these lines! Select initial parameters for each line (10 in all). The lines I selected are shown below:



Now perform the fit and view the results:

```
eis_auto_fit, wd, fitdata, template=template
eis_fit_viewer, wd, fitdata
```

Try clicking through the 10 different lines using the button widgets and see how the image morphology changes (due to the different formation temperatures of the lines). Check the Brown et al. (2008, ApJS) spectral atlas paper to get the identifications for each of the lines.

Choosing one of the weaker lines in the spectrum, go to the profile plot (bottom-left) and click through the ‘X-range’ and ‘Y-range’ options and see the effect they have on the plot.

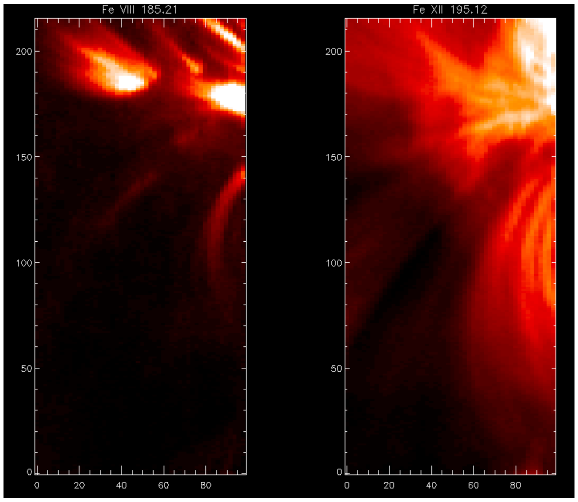
Example 7: an absolute velocity calibration for Fe VIII

Young et al. (2012, ApJ, 744, 14) argued that the only reliable way to obtain absolute velocity measurements from EIS active region data is to make use of quiet Sun regions that are observed simultaneously with the active region. Examples 7 and 8 go through the steps in obtaining absolute velocities for Fe VIII λ185.21 and Fe XII λ195.12, and Example 9 explains how to compute the uncertainties on these velocities.

The data-set is a 2” slit raster of the periphery of an active region obtained on 17-Jan-2007 at 18:42 UT. Firstly we view the Fe VIII 185 and Fe XII 195 images using eis_raster_browser:

```
l1name=’eis_l1_20070117_184227.fits’
eis_raster_browser, l1name
```

Use the widget tools to select the two lines. You should see images similar to the ones below.



The Fe XII image shows extended loop structures and, although the signal gets weak towards the bottom of the raster, by playing with the image scaling you will see that loop structures extend to the bottom of the raster, even in the dark lane. There is thus no quiet Sun at all in the raster for Fe XII. For Fe VIII it is a different situation. Only the lower parts of the loop structures are bright and, by playing with the image scaling, you will see that there are no loop structures at the bottom of the raster. This was the point made by Young et al. (2012): for coronal ions such as Fe XII the extent of the active region is so large that it is usually not possible to access quiet Sun regions that can be used for a velocity calibration. For a cooler ion such as Fe VIII, the active region structures are more compact and it is possible to access quiet Sun from the same data-set.

For the present Example, we assume that the bottom 40 pixels of the Fe VIII raster is quiet Sun and use this region to set an absolute velocity for Fe VIII. (Is this region actually quiet Sun? – see the end of this example for a discussion of this point.)

The first task is to fit the Fe VIII line in the quiet region. Since the Fe VIII 185 line is weak in the quiet regions, it is necessary to perform binning in the Y-direction (20 pixels in this case) to boost the signal-to-noise:

```
wd185=eis_getwindata(l1name,185.21,/refill)
```

```
wd185qs=eis_bin_windata(wd185,ybin=20)
```

(Why not use a y-bin of 40 since the quiet Sun region is 40 pixels? The signal in this data-set is good enough to use a y-bin of 20, so I then average the bottom two rows of the binned raster later. You should find using a y-bin of 40 gives approximately the same result, though.)

Now use eis_fit_template to select the 185.21 line (which is on the left side of the wavelength window), and eis_wvl_select to de-select two weak lines in the Fe VIII wavelength window, and then perform the fit. Note that I force the spectrum background to be flat by setting template.nback=1. This usually helps to avoid bad fits.

```
eis_fit_template, wd185qs, template
template.nback=1
eis_wvl_select,wd185qs,wvl_select
eis_auto_fit, wd185qs, fit185qs, wvl_select=wvl_select, template=template
```

You can use eis_fit_viewer to browse the fit data, paying particular attention to the bottom two rows that we consider to be quiet Sun.

Now update the fit structure by using the bottom two pixels (corresponding to pixels 0 to 39 in the original, unbinned data):

```
newfit185qs=eis_update_fitdata(fit185qs,yrange=[0,1],offset=offset)
```

This routine uses the centroids in rows 0 and 1 to (a) define a new orbit correction (which is plotted to the screen), and (b) compute a new *refwvl* for the emission line, which is stored in newfit185qs.refwvl. The new orbit correction replaces the default one from the Kamio et al. method (see EIS Software Note No. 5), and it goes into the *offset* array which is used here (and also in Example 8).

Now the *refwvl* value is calculated assuming that the average Doppler shift of Fe VIII in the quiet Sun (rows 0 and 1) is 0 km/s. Fe VIII is formed at logT=5.8 in the quiet Sun (see Brooks et al. 2011, ApJ, 730, 85). Peter & Judge (1999, ApJ, 522, 1148) found that the average quiet Sun Doppler shift at this temperature is about −2.6 km/s (a blueshift). Therefore, we should manually adjust the *refwvl* for Fe VIII to yield a Doppler shift of -2.6 km/s. This can be done with:

```
newfit185qs.refwvl= newfit185qs.refwvl - v2lamb(-2.6,185.21)
```

The next step is to fit the original unbinned Fe VIII λ185 data stored in wd185. Note that we are giving the optional input *offset*, which comes from the quiet Sun fit (see above).

```
eis_fit_template, wd185, template, offset=offset
template.nback=1
eis_wvl_select,wd185,wvl_select, offset=offset
eis_auto_fit, wd185, fit185, wvl_select=wvl_select, template=template, offset=offset
```

(Note that you need to generate *template* and *wvl_select* again as the previous ones correspond to the binned data.)

The final step is to change the *refwvl* value to the one computed above:

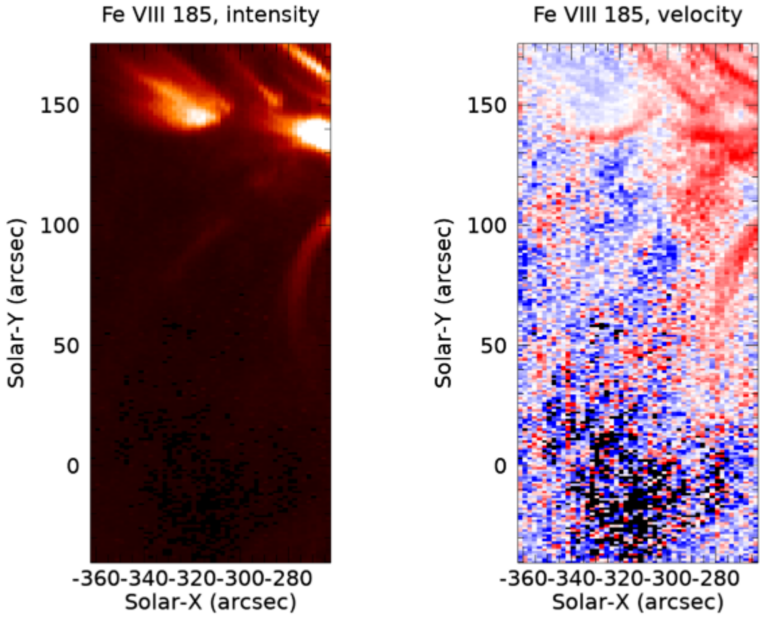
```
fit185.refwvl=newfit185qs.refwvl
```

If you now generate a velocity map with

```
vel=eis_get_fitdata(fit185,/vel,/map)
```

then it will be an *absolute* velocity map (although bear in mind the assumptions made). Example 9 will show how the uncertainties on this velocity map are computed.

The plots below show the Fe VIII intensity and velocity maps (scaled between ± 20 km/s) for the complete raster. Note the loop footpoints at the top of the raster are mainly redshifited.



Earlier in this example we queried whether the bottom 40 pixels really do correspond to quiet Sun. If we do:

```
IDL> fe8_int=eis_get_fitdata(newfit185qs,/int)
IDL> plot, fe8_int[*,0]
```

This plots the Fe VIII 185 intensity from the bottom row of the binned data. It varies from around 10 to 40, with a larger peak at the right-side of the raster. Brooks et al. (2009, ApJ, 705, 1522) give average quiet Sun intensities of a range of emission lines. For Fe VIII 185.21 they find a value of 19.7 erg cm^{-2} s^{-1} sr^{-1}, thus the measured intensities are consistent with this value apart from the peak at the right-side of the raster. It thus seems reasonable to assume that quiet Sun is being observed, except for the right-side of the raster which may be affected by a bright point or loop structure. (Note that Brooks et al. used the original laboratory intensity calibration, and so I use the same for the Fe VIII data-set, i.e., I do not set the *calib* keyword.)

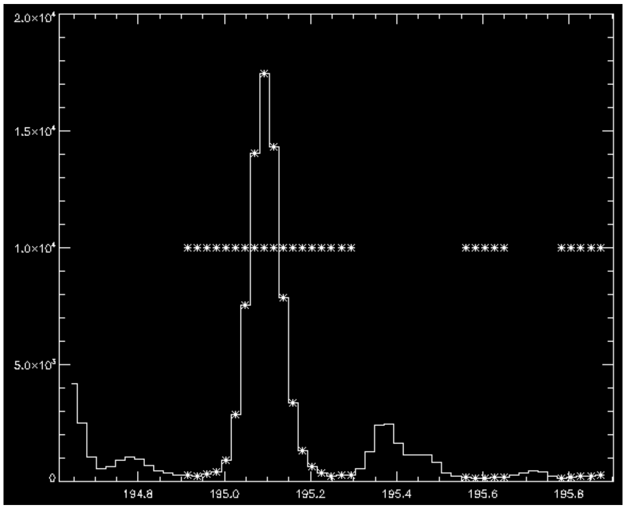
Example 8: extending the Fe VIII velocity calibration to Fe XII

Example 7 showed how to derive an absolute velocity calibration for Fe VIII λ185, taking advantage of the fact that the active region structures in Fe VIII are compact, allowing a section of quiet Sun to be observed. This Example shows how to extend this calibration to Fe XII λ195. Note that you will need some of the data products from Example 7.

First, we fit the λ195 line using the usual routines. Note, however, the use of the *offset* array from Example 7 as an optional input.

```
wd195=eis_getwindata(l1name,195.12,/refill)
eis_fit_template,wd195,template,offset=offset
eis_wvl_select,wd195,wvl_select,offset=offset
eis_auto_fit, wd195, fit195, offset=offset, wvl_select=wvl_select, template=template
```

For the template, simply select a single Gaussian for the strong Fe XII line (although you could also try the double-Gaussian fit from Example 5). For the wavelength selection (eis_wvl_select) the pixels I have chosen are shown in the figure below.



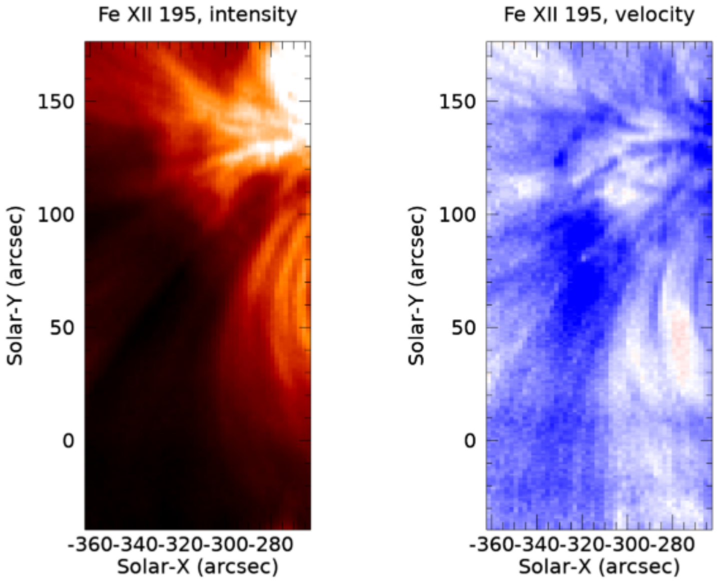
We now have to set the reference wavelength (*fit195.refwvl)* and for this we use the Fe VIII results from Example 7. The absolute Fe VIII λ185 wavelength is stored in *newfit185qs.refwvl.* Table 2 of Warren et al. (2011, ApJ, 727, 58) gives wavelengths of most of the strong EIS lines measured above the limb in a quiet Sun region. If we assume this plasma has zero Doppler shift (which is reasonable given there is unlikely to be bulk motion along the line-of-sight above the limb) then we can use the measured separation of the Fe VIII and Fe XII lines (9.908 Å) to infer the rest wavelength of the Fe XII line in our active region raster. That is:

```
fit195.refwvl= newfit185qs.refwvl + 9.908
```

With this simple change we now have an absolute velocity calibration for Fe XII! You can derive the velocity map in the usual way:

```
vel=eis_get_fitdata(fit195,/vel,/map)
```

As the image below shows, most of the raster area is blue-shifted for Fe XII (I've scaled between ± 15 km/s).



Example 9: Velocity uncertainties

The convoluted process by which we arrive at an absolute velocity means that there are several sources of uncertainties that combine to give the final velocity uncertainty. This Example continues on from Examples 7 and 8 to show how these uncertainties are obtained for the Fe VIII and Fe XII lines, and so the user should have the data products from these examples available. The procedure follows that described by Young et al. (2012, ApJ, 744, 14).

Eq. 5 of Young et al. (2012, ApJ, 744, 14) gives the expression for the centroid uncertainty of Fe VIII λ185 (the centroid uncertainty can be easily converted to the velocity uncertainty—see below). There are four terms. $\sigma_{\rm loop}$ is the measurement error for a particular pixel in the raster, which is obtained by doing:

```
cen=eis_get_fitdata(fit185,/cen,err=err)
```

Here *err* is the value of $\sigma_{\rm loop}$ for each pixel in the raster. It comes from the Gaussian fitting.

The tilt uncertainty, $\sigma_{\rm tilt}$, is obtained as follows:

```
newfit185qs=eis_update_fitdata(fit185qs,yrange=[0,1],offset=offset,refpix=refpix,fit_error=fit_er
ror)
tilterr=eis_tilt_error(fit185,refpix)
```

(The call to eis_update_fitdata is the same as that used in Example 7, only there are two additional optional outputs here.) The *tilterr* output is a 2D array of same size as the raster, giving the tilt error in angstroms at each position (note that *tilterr*=0 for Y=*refpix*).

$\sigma_{\rm qs}$ from Young et al. (2012) is the uncertainty arising from the orbit correction done in the quiet Sun region, and is contained in the optional output *fit_error* returned by *eis_update_fitdata*. It's a scalar in Angstroms.

The final uncertainty term is $\sigma_{\rm PJ}$ which comes from the Peter & Judge (1999) paper mentioned in Example 7. Fe VIII corresponds to Ne VIII in this work, and the uncertainty for this ion is ± 1.8 km/s. This can be converted to a wavelength for λ185 using w=v2lamb(1.8,185.21).

Adding the four uncertainty terms together in quadrature gives the overall centroid uncertainty for Fe VIII λ185 at each pixel in the raster. This can be converted to a velocity uncertainty using the *lamb2v* routine: sigma_v=lamb2v(sigma_cen,185.21).

For Fe XII λ195, there is an additional uncertainty coming from the relative wavelength separation of the Fe VIII and Fe XII lines. We use the uncertainties given in column 4 of Table 1 of Warren et al. (2012, ApJ, 727, 58), and add them in quadrature to give 0.51 mÅ. This is then added in quadrature to the Fe VIII uncertainty to give the final uncertainty for the Fe XII line.

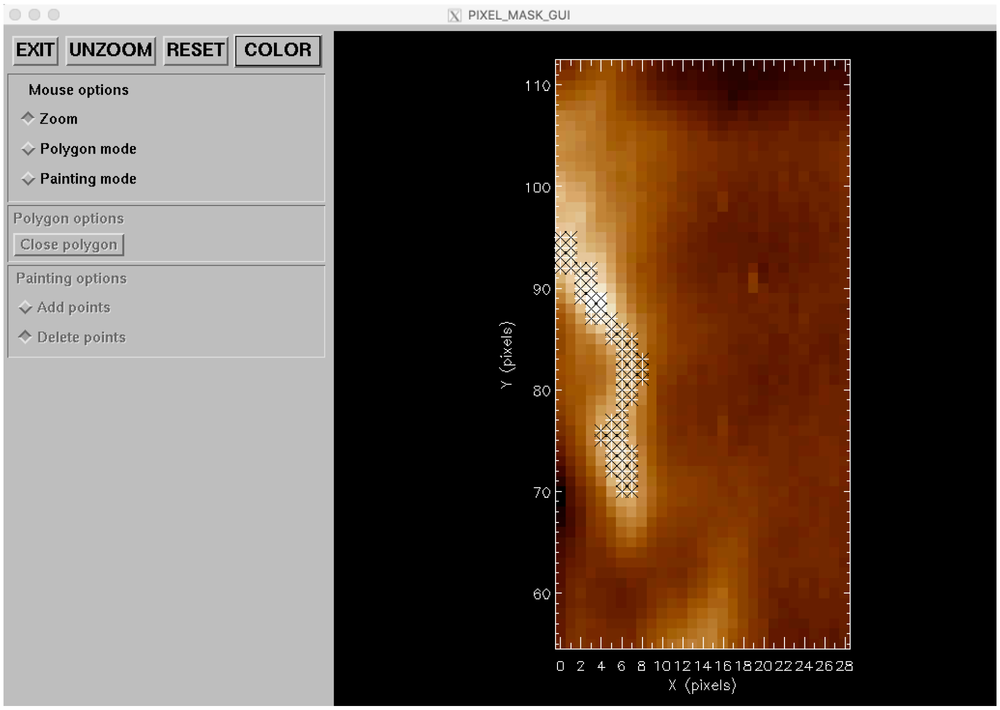
Example 10: Applying an eis_auto_fit template to a mask spectrum

Here we create a fit template for the 3-May-2007 data-set and then apply it to a mask spectrum from the same data-set.

Repeat Example 3 to obtain the template and wvl_select structures. You should have selected the Fe XVI λ262.98 and Fe XIV λ264.79 lines.

Now we create a pixel mask from the Fe XVI line:

```
eis_make_image, l1name, 262.98, map, /map
mask=pixel_mask_gui(map)
```



In the image above, the thin, wavy structure has been selected at the left side of the raster (note that the image shows a sub-region of the raster).

Now create the mask spectrum. Note the use of the /shift keyword and the calib=3 option (this selects the Warren et al. 2014 calibration).

```
eis_mask_spectrum, l1name, mask, lwspec=lwspec, /shift, calib=3, /refill
```

To fit the spectrum using the template, do

```
output=eis_mask_auto_fit(wd, lwspec, template, wvl_select=wvl_select)
```

where wd is the windata structure from Example 3. The output is a structure containing the fits to the two lines in the mask spectrum. The fits can also be sent to a text file by using outfile=. The file and the structure are in the same formats as produced by the spec_gauss_eis routine (see EIS Software Note 16).