EUV IMAGING SPECTROMETER

Hinode

EIS SOFTWARE NOTE No. 17

Version 3.2 2 April 2013

Gaussian fitting examples using the EIS_AUTO_FIT routine

Peter Young
George Mason University
4400 University Drive
Fairfax, VA 22030
U.S.A.

pyoung9@gmu.edu

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, /correct_sensitivity

Update history

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
fit.refwvl=195.119
```

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)
vel=eis_get_fitdata(fit,/vel)
wid=eis_get_fitdata(fit,/wid)
!p.multi=[0,3,1]
loadct,3
plot_image,int
plot_vel_image,vel
loadct,3
plot_image,wid
!p.multi=0
```

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:

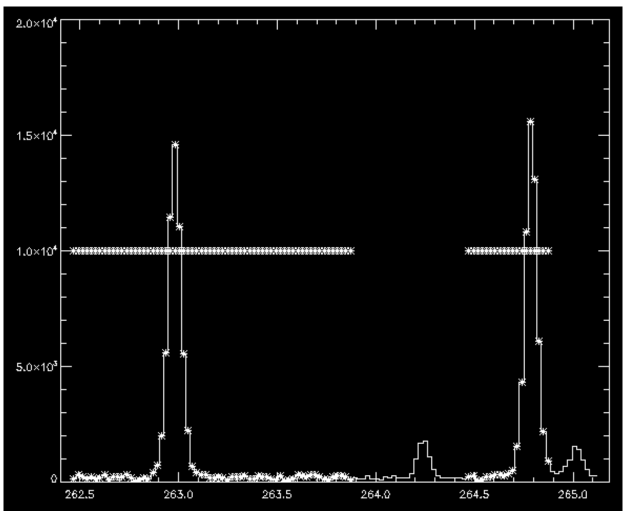
```
eis_fit_template, wd, template
```

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:

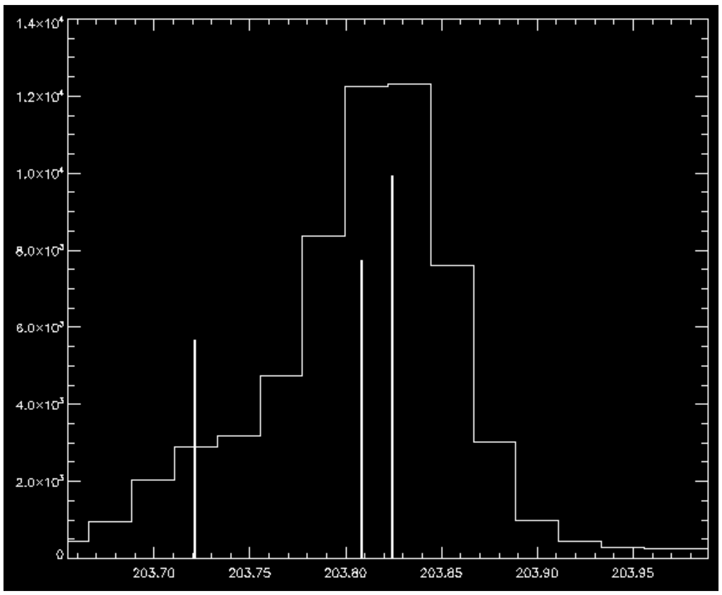
```
eis_auto_fit, wd, fit, wvl_select=wvl_select, template=template
eis_fit_viewer, wd, fit
```

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.

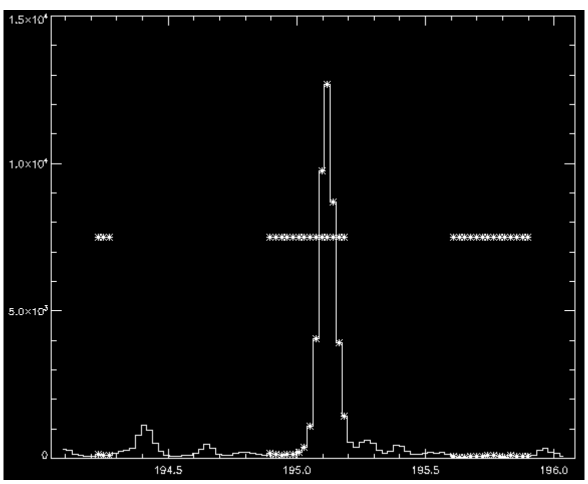
l1name='eis_l1_20070503_050629.fits'
wd=eis_getwindata(l1name,195.12,/refill)

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:

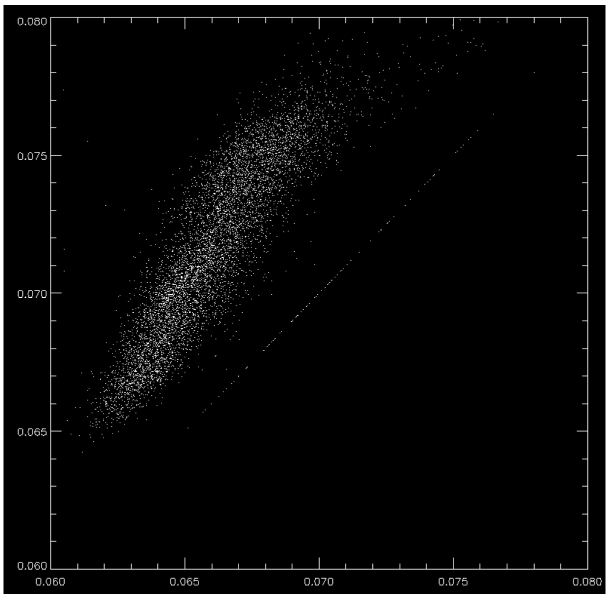
eis_auto_fit, wd, fit, wvl_select=wvl_select, template=template

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
```

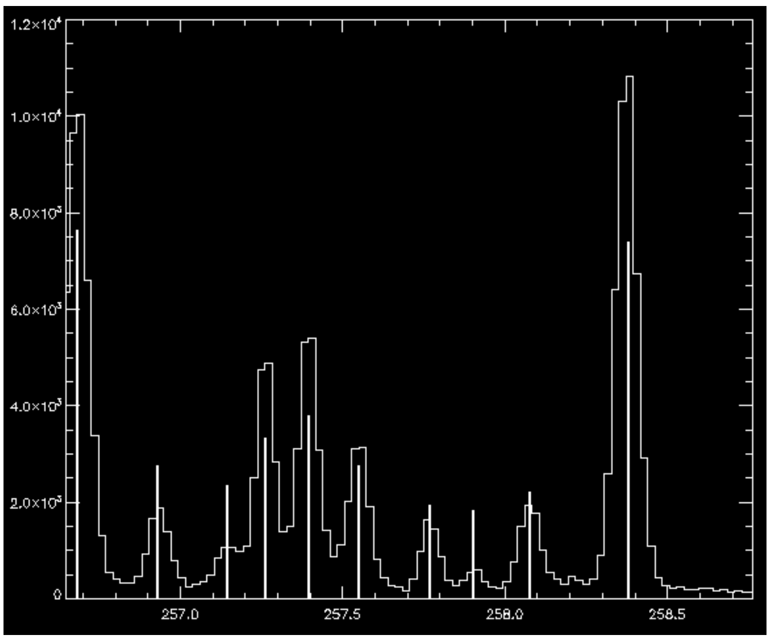
you will find that there are some pixels for which the 195.18 intensity is zero, thus the routine was not able to fit a second Gaussian to the line profile in this case.

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.

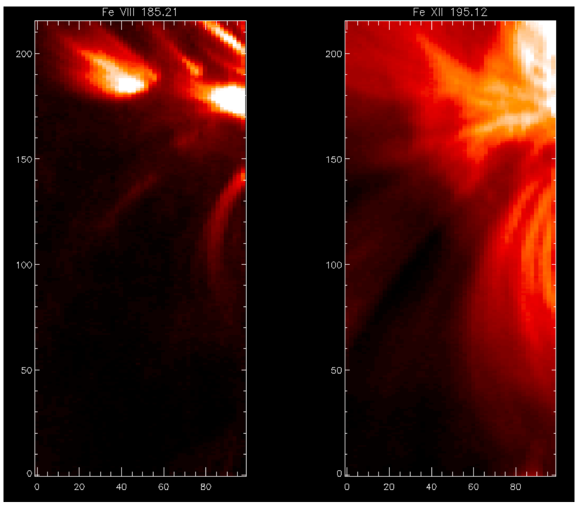
Example 7: using Fe VIII to obtain the orbit correction for Fe XII

The standard option for deriving an absolute wavelength scale for EIS data is to use the method described by Kamio et al. (2010, Sol. Phys., 266, 209). In this example we show how to determine the wavelength scale by using a portion of quiet Sun data in the science raster.

Firstly let’s view images in the Fe VIII 185 and Fe XII 195 lines from a raster showing loops a the periphery of an active region:

```
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. We will thus take the bottom of the raster in Fe VIII to represent quiet Sun. In this example we show an absolute wavelength scale determined from the Fe VIII line (in the quiet Sun area) can be extended to the Fe XII line, yielding absolute velocities for Fe XII.

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 to boost the signal-to-noise:

```
wd185=eis_getwindata(l1name,185.21,/refill)
wd185=eis_bin_windata(wd185,ybin=20)
```

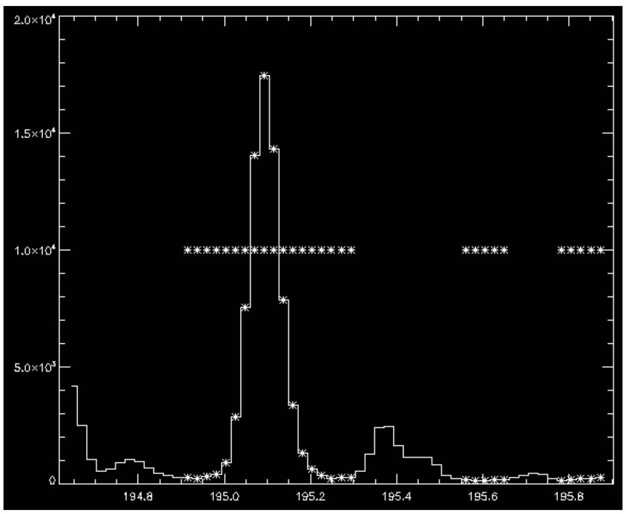
Now use eis_wvl_select to de-select two weak lines in the Fe VIII wavelength window, and then perform the fit.

```
eis_wvl_select,wd185,wvl_select
eis_auto_fit, wd185, fit185qs, wvl_select=wvl_select
```

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

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

This uses the quiet Sun centroid positions of the Fe VIII line to define a new 'offset' array that goes into newfit185qs.offset. In addition, the call above produces a 2D array called 'offset' that is the same size as the original raster (50x216). It has been constructed using the orbit variation from the Fe VIII 185 line in the quiet Sun region. This array can then be used for fitting other emission lines, instead of using the default windata.wave_corr array. The example below shows how this is done for Fe XII 195.12.



To fit the 195 line do:

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

(Note that of the 'offset' array is now being used as an input.)

For the wavelength selection the pixels I have chosen are shown in the figure.

The final step is to set the reference wavelength (fit195.refwvl) for the 195.12 line. If I first do:

```
print,newfit185qs.refwvl
```

I get a value of 185.193. This is the average wavelength of the Fe VIII 185 line in the quiet Sun region, which we assume corresponds to 0 km/s. Now, the laboratory wavelengths for the Fe VIII and Fe XII lines are 185.213 and 195.119 Å, respectively (the 'Lit.' column in Table 2 of Brown et al. 2008, ApJS, 176, 511). That is, they are separated by 9.906 Å. We thus set the reference wavelength of the 195 line to be:

```
fit195.refwvl=newfit185qs.refwvl+9.906
```

This tells the software that, on a wavelength scale where 185.193 corresponds to 0 km/s, the rest wavelength of the Fe XII line will be 195.099.

We can now view the fits for the Fe XII line:

```
eis_fit_viewer, wd195, fit195
```

Bear in mind that an additional uncertainty applies to the 195 velocity measurements, namely the uncertainty in the EIS relative wavelength scale. Based on the plots shown in Fig. 4 of Brown et al. (2008, ApJS, 176, 511) these are around ± 0.002 Å.

Three questions should be considered by the reader:

1. Is the region chosen as quiet Sun in the Fe VIII image *really* quiet Sun?
2. Is the wavelength offset between Fe VIII and Fe XII *really* 9.906 Å?
3. Is Fe VIII really at rest in the quiet Sun?

For the first question, we can 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 35, 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.

For the second question, an alternative way of deriving the wavelength offset is to measure the actual EIS offset of the two lines in the quiet Sun. Warren et al. (2011, ApJ, 727, 58) measured the two lines above the limb in a quiet Sun region and found an offset of 9.908 Å, so this may be a better offset value to use.

For the third question, we note that Fe VIII appears to be 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 −2.5 km/s (a blueshift). The method for specifying this shift in the fitdata structure is described in the next example.

Example 8: Deriving absolute velocities for Fe VIII 185.21

Here we will derive absolute velocities for Fe VIII 185.21 in a coronal loop structure by using a nearby patch of quiet Sun to set the absolute wavelength calibration. All (known) sources of error will be combined to yield a definitive accuracy for the velocity measurement.

We will repeat the initial steps from Example 7 up until the fit to the quiet Sun Fe VIII 185 line has been performed and the *fit185qs* structure created. The call to *eis_update_fitdata* is modified to:

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

the output *refpix* contains the Y-pixel number that represents the reference point for wavelength measurements, and *fit_error* is the error on the fit to the orbit variation. We will use these numbers later.

We will now re-fit the 185 line using the new offset array, but this time we will not bin the 185 data.

```
wd185=eis_get_windata(l1name,185.21,/refill)
eis_wvl_select,wd185,wvl_select,offset=offset
```

De-select a couple of weak lines in the wavelength window at this stage, and then do the line fitting:

```
eis_auto_fit, wd185, fit185, offset=offset, wvl_select=wvl_select
```

We now manually change the *refwvl* wavelength for *fit185*:

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

(Remember that *eis_auto_fit* always sets *refwvl* to be the average centroid in the raster and so this value generally should not be used for absolute velocity work.) This calibrates the velocities relative to the quiet Sun region. If you now create a velocity array:

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

then these velocities are actually *absolute* velocities (bearing in mind the various assumptions that have been made in getting to this point, though).

The errors on the velocity can be obtained by giving the optional input, error=

```
vel=eis_get_fitdata(fit185,/vel,error=velerr)
```

but the only uncertainties this contains are the Gaussian fitting errors. There are additional uncertainties to consider:

1. the uncertainty in the tilt of the slit relative to the detector;
2. the uncertainty in the determination of the orbital variation of the line centroid; and
3. the uncertainty in the assumption that the quiet Sun absolute velocity of Fe VIII is zero.

The tilt error can be obtained by doing:

```
tilterr=eis_tilt_error(fit185,refpix)
```

where *refpix* was obtained earlier. Note that *tilterr* is a 2D array with the spatial dimensions of the raster.

For the orbit variation, the quantity *fit_error* (derived earlier) contains this uncertainty.

For the absolute velocity uncertainty, this is somewhat subjective. We believe that Fe VIII is formed at the temperature of Ne VIII, although perhaps a little lower. Peter & Judge (1999, ApJ, 522, 1148) give a disk center quiet Sun velocity of -2.7 ± 1.0 km s^{-1} (this is a blueshift) for Ne VIII. There are some assumptions that go into deriving this velocity and the reader is referred to the Peter & Judge paper for more details. A conservative scientist may prefer to use a value such as 0 ± 5.0 km s^{-1} (see Fig. 6 of Peter & Judge).

We can combine these errors to produce a final velocity error array by doing:

```
newfit185=eis_update_cen_error(fit185, tilterr, fit_error, absvel=0, abserr=5)
```

or

```
newfit185=eis_update_cen_error(fit185, tilterr, fit_error, absvel=-2.7, abserr=1)
```

depending on what you want to do with the absolute quiet Sun uncertainty.

The final velocity error is then given by:

```
vel=eis_get_fitdata(fit185,/vel,error=velerr)
```

Note that the Gaussian fitting errors and the orbital variation errors will generally be around 5 km s^{-1} or more so the quiet Sun errors are not critical to the analysis.