1 Purpose

The purpose of this document is to describe how to determine the properties of line emission in an astrophysical source. These properties include

- Line Flux
- Line Width
- Line Position
- Multi-Order Fitting

and an example of each type of analysis will be presented for the ISIS/Guide analysis software.

2 Introduction

Spectral line analysis can proceed either by using an RMF (redistribution matrix) and forward folding to fit a model to the data, or with high resolution spectral data using line profile to fit individual emission lines. Here we will discuss using the line profile for the analysis of HETG data. First we will present an overview of the analysis path and the programs used in the analysis. Below we will assume that the data has be processed and the spectra extracted using methods discussed elsewhere (e.g. CIAO threads). These examples will be done using ISIS which is a prototype of the **Guide** software which is scheduled to appear in CIAO 2.1. An ISIS manual and overview can be found at `http://space.mit.edu/ASC/ISIS/`.

3 Data Reduction

It is assumed here that the data has been reduced and a spectrum has been extracted. This extracted spectrum can be either a type I spectrum or a type II spectrum (which contains multiple spectra). Below we assume that a type II spectrum has been extracted.
For a more complete discussion on extracting spectra from Chandra data see
`http://asc.harvard.edu/ciao/threads/threads.html`.

3.1 Overview

To load the data into ISIS
`load_data("filename");`
For a type II pha file choose which dataset to work with
Load the ARF/EEARF Flux correct the data Display the data
`load_arf("arf_file");`
Plot the data
`plot_data(10);`
Select a line to work with
Generate a line profile eg. external program `lsf_parm`
Load the line profile
`import lsf.isis`
Fit the line profile
Determine the flux

3.2 Loading the Data into ISIS

Once a data file has been created and ISIS has been started the data can be loaded into ISIS,
`isis>load_data("acisf1235_000N001_pha2.fits");`
which results in

```
Reading: ............
*** invalid uncertainties replaced

Current Spectrum List:
 id  instrument   m prt  src    use/nbins   A      totcts   exp(ksec) clr  target
  1   HETG-ACIS  -3  1   1    6972/ 6972   -  6.1000e+01   15.800   -   CAPELLA
  2   HETG-ACIS  -2  1   1    6972/ 6972   -  1.1100e+02   15.800   -   CAPELLA
  3   HETG-ACIS  -1  1   1    6972/ 6972   -  3.0010e+03   15.800   -   CAPELLA
  4   HETG-ACIS   1  1   1    6972/ 6972   -  2.5470e+03   15.800   -   CAPELLA
  5   HETG-ACIS   2  1   1    6972/ 6972   -  1.1300e+02   15.800   -   CAPELLA
  6   HETG-ACIS   3  1   1    6972/ 6972   -  6.3000e+01   15.800   -   CAPELLA
  7   HETG-ACIS  -3  2   1    6972/ 6972   -  6.1400e+02   15.800   -   CAPELLA
  8   HETG-ACIS  -2  2   1    6972/ 6972   -  8.8900e+02   15.800   -   CAPELLA
  9   HETG-ACIS  -1  2   1    6972/ 6972   -  1.3810e+04   15.800   -   CAPELLA
 10   HETG-ACIS   1  2   1    6972/ 6972   -  8.1050e+03   15.800   -   CAPELLA
 11   HETG-ACIS   2  2   1    6972/ 6972   -  3.6200e+02   15.800   -   CAPELLA
 12   HETG-ACIS   3  2   1    6972/ 6972   -  2.4800e+02   15.800   -   CAPELLA

Integer_Type[12]
```

being output to the screen. In this example we have both the HEG spectra, prt 1, and the MEG spectra, prt 2, for orders -3,-2,-1,1,2,3.

3.3 Selecting the data set to use

In this example we will use the, MEG +1 spectrum (prt = 2 and m=1) which is spectrum id = 10.

> *If we know which spectrum we wish to work with in advance then we could load only that spectrum using*
>
> `load_data("acisf1235_000N001_pha2.fits",10);`
>
> *to load the tenth spectrum (MEG order +1). In this case the id for the spectrum would be 1 instead of 10.*

3.4 The ARF (ancillary response file)

Once an arf has been created (see the CIAO threads on how to generate an arf) we can load the arf and create a fluxed spectrum.
`isis> load_arf("Capella_MEG_garf.fits");`

```
*** Encountered ARF uncertainties <= 0;  Reset those to 1 ***

Current ARF List:
 id grating detector   m prt src   nbins  exp(ksec)  target
  1    HETG     ACIS   1  2   0     8192    88.88  CAPELLA
```

Which tells us we have loaded a MEG arf for order +1.
Use the assign_arf command to associate the arf we just read in to spectrum 10.
`isis> assign_arf(1,10);`
Next we need to create a fluxed spectrum - that is a spectrum that has been divided by the arf and the exposure time. This results in a spectrum that is an approximation to the source spectrum blurred by the instrumental line spread function (lsf). Note that this calculation assumes that the effective area function is slowly varying so that it can be moved in front of the integral over wavelengths contributing to each histogram bin; this approximation may break down near instrumental edges and other sharp features in the effective-area function.
`isis> flux_corr(10,1);`
This assigns the arf to spectrum 10. The second argument applies a S/N threshold so that data in low count bins do not dominate the plotted spectrum.

3.5 Examine the data

At this point the data has been corrected for exposure time and the variations from the effective area of the telescope detector combination the next step is to examine the spectrum. This is done using the plot_data command in ISIS. First we select the flux corrected dataset,
`isis> use_flux;`
Using flux-corrected data
and next we plot the data.
`isis> plot_data(10);`

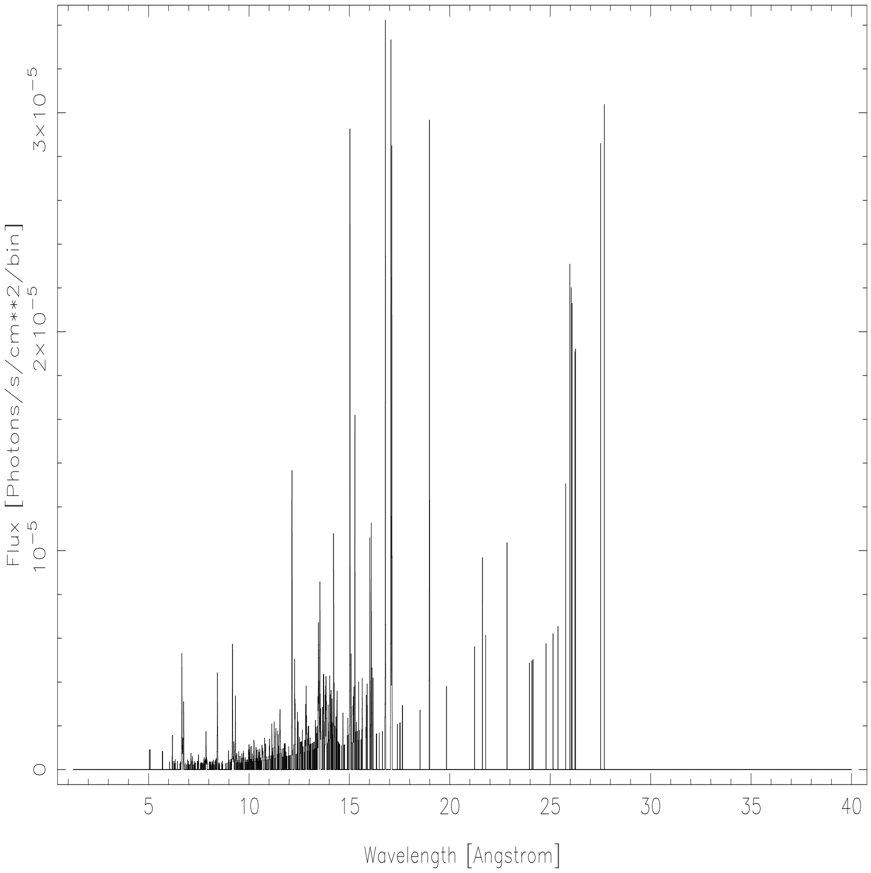

Figure 1: The fluxed spectrum of Capella

3.6 Select a Line to Work with

For this example we will work with the Fe line seen at 15A. First lets enlarge the region near the line.
`isis> xrange(14.99,15.07);`
`isis> plot_data(10);`
and now we see the data as plotted in figure 2.

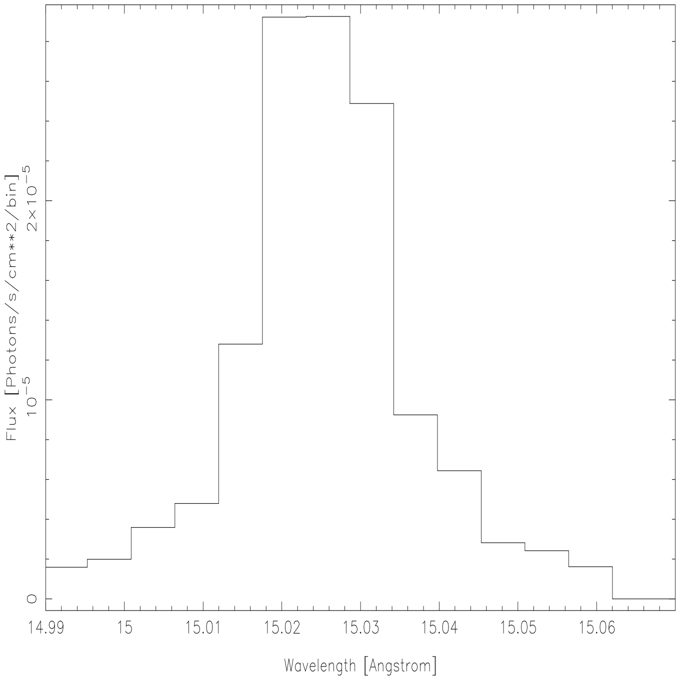

Figure 2: The 15 ÅFe line in the Capella spectrum

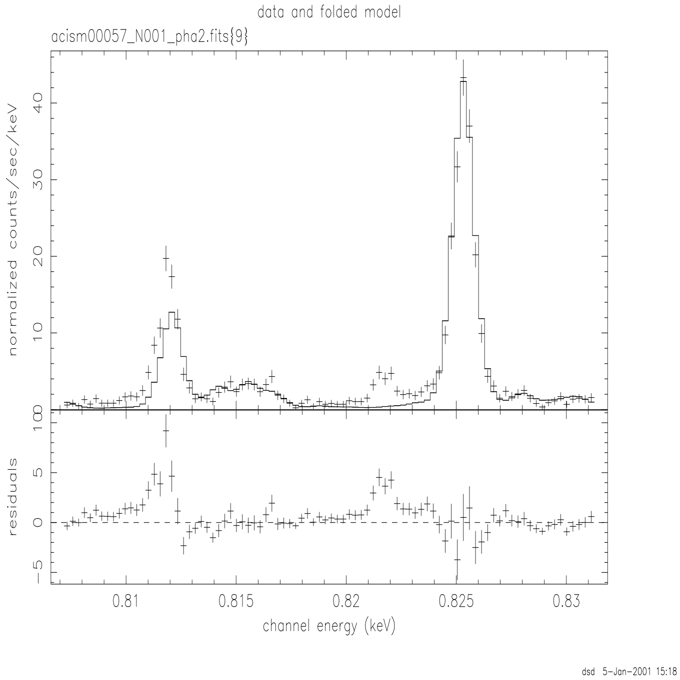

Figure 3: The 15 ÅFe line in the Capella spectrum with the fitted lsf shown in red.

4 Determining a Line Flux

There are several methods that can be used to determine the flux in a line for high resolution spectra. We can

- 1) sum the counts in the region of interest
- 2) we can fit the line with a function
- 3) we can use an RMF to forward fold a model

4.1 Region of Interest

The simplest method to find a flux is to sum the counts in the region of interest
`isis> p = region_sum (10, 14.99, 15.05);`
`isis> p.sum;`
0.000126637
Where again the units are Photons s^{-1} cm^{-2}. This works best for lines with little continuum or where there are nearby line free regions so that the continuum may be determined and removed from the line emission. The continuum level can be measured using the same method that we used for the line flux.
To be of practical use at least two conditions must be met. The first is that the arf for that region must be fairly constant across the region of interest. This assumption can be checked using the method described in

the section on forward folding. The second condition is that the continuum must be fairly well determined. For a weak continuum errors in determining the background level will not affect the line flux much. If the continuum is fairly flat then measuring its contribution off the line and assuming that that is a reasonable estimator for the continuum under the line may be adequate. Finally in some regions the continuum may not be evident, eg. in line rich regions. In this case the continuum may be determined from other methods, for example a theoretical model. In each of these cases the user must determine if this is adequate for the science that they are trying to achieve.

4.2 Profile Fitting

A more sophisticated method of determining the line flux is to fit a functional form to the line and the continuum. This has the advantage of simultaneously determining the line and continuum flux, being able to fit blended lines and being able to determine errors for the fitted parameters.
After loading the spectrum as shown above then to fit the function first ignore all the other spectra.
`isis>ignore([1:9]);`
`isis>ignore([11:12]);`
and then ignore the data outside our selected window.
`isis>xnotice(10,14.99,15.07);`
now we can fit that one line.
We will fit a function that represents the measured line spread function and an estimator for the continuum. The LSF may be represented my a Gaussian for the core and a Lorentzian for the wings and the background will be modeled with a polynomial. We will use the `ifit_fun` of ISIS which allows us to determine the initial parameters using a plot and the mouse.
`isis> ifit_fun("lorentz(1) + gauss(2) + poly(1)");`
now to check that the initial guess is reasonable
`plot_data(10); eval_fun; oplot_model;`
which will overplot the inital guess on the plotted data. If this does not seem reasonable then we can re-initialize the parameters with the `ifit_fun` or use `edit_par` to change the parameters. Once the inital values are O.K. then we can fit the function
`isis>fit;`
and again check the fit
`plot_data(10); oplot_model;`
to check the fit. This results in a model fit to the line seen in figure 3, where the white line is the data and the red line is the fit.
To find the flux in the model,

```
isis> (lo, hi, dat, dat_err) = get_data(10);
isis> (lo, hi, model) = get_model(10);
isis> limits;
isis> xrange(14.99,15.07);
isis> hplot (lo, hi,model);
isis>print,sum(model);
0.000113267
```

Where the units are the units of the model (photons s^{-1} cm^{-2}).
To find the flux in the line,
`isis> p = region_sum (10, 14.99, 15.05);`
`isis> p.sum;`
0.000126637
Where again the units are photons s^{-1} cm^{-2}.

4.3 Creating an RMF

In general the spectra generated from the pipeline products are adequate for most analysis. This means that custom RMFs are not needed and the off-the-shelf RMFs provided in the *Chandra* CALDB are suitable for

spectral analysis. Custom RMFs should be generated for spectra extracted with non-standard extraction regions, with non-standard gridding, when analyzing higher order spectra (order > 3) or if the observation is more than $\sim 90'$ off-axis.
Mkgrmf program generates an OGIP style RMF appropriate for spectral analysis of grating observations. The RMF is a representation of how the incident source photons are redistributed onto the detector, thus the grating RMF contains the HRMA + grating line spread function (LSF) vs energy for a given detector and grating combination.
mkgrmf calculates the RMF on user defined set of grid points, e.g. wavelength channel. To create a third order grating rmf

```
mkgrmf grating_arm="HEG" order=3 outfile="heg3.rmf" \
obsfile="./acis_f1235_000N002_pha2.fits" \
wvgrid_arf="compute"\
wvgrid_chan="compute"\
src_id=1
```

Computes a third order RMF for the HEG, writes the output to heg3.rmf and uses the default grids for the rmf. Note setting the parameter wvgrid_arf to "1.0:21.48:#8192" is equivalent to setting it to compute. In addition to creating rmfs using the standard grid a custom grid can be generated

```
mkgrmf grating_arm="LEG" order=1 outfile="leg1.rmf" \
obsfile="./hrcf00058N002_pha2.fits" \
wvgrid_arf="1.0:205.8:0.01"\
wvgrid_chan="1.0:205.8:0.01"
```

Computes a first order RMF for the LEG, writes the output to leg1.rmf and uses non-standard grids for the rmf.

4.4 Applying an RMF

In the above example we were fitting a line where the ARF was constant across the region of interest. At the other extreme we need to fit lines where the arf is changing rapidly. In this case we need to use an ARF and RMF to convolve the model and fit to the data to get an accurate estimate of the line flux.
For this example we need to load an ARF and assign it to a spectrum
`isis>assign_arf(1,10);`
Similarly, we need to load an rmf and assign it to a spectrum
`isis> assign_rmf("acismeg1D1999-09-22rmfN0000.fits",10);`
Get the ebounds array from the data
`isis>(lo, hi, , ) = get_data(10);`
Initialize the plasma database
`isis> plasma ("aped");`
Load a predefined model
`isis> load_model("model.dat");`

```
# id Temp Dens Abund Norm Vturb redshift Nh
#     (K) (cm^-3) (km/s) (cm^-2)
  1  2.0000e+06 1.0000e-03 1.0000e+00 1.0000e+00 2.0000e+02 0.0000e+00 1.0000e+20
```

and finally compute spectral model on the same grid as the data
`isis> sp = model_spectrum (lo, hi);`
Once we have loaded and assigned an ARF and RMF the fitting proceeds as above but this time the model is folded through the arf and rmf before being compared to the data.
To find the flux in the model,

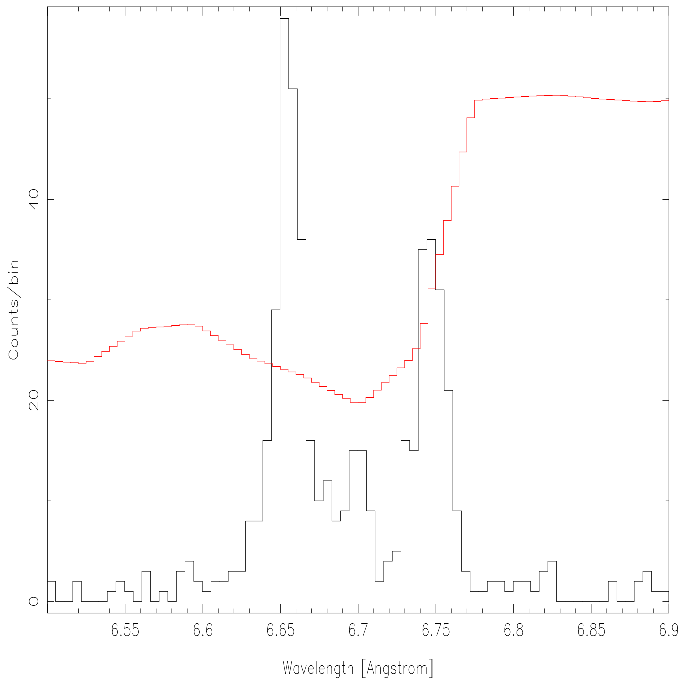

Figure 4: The Si line regin in the Capella spectrum with the arf (in arbitrary units) overplotted on the spectrum.

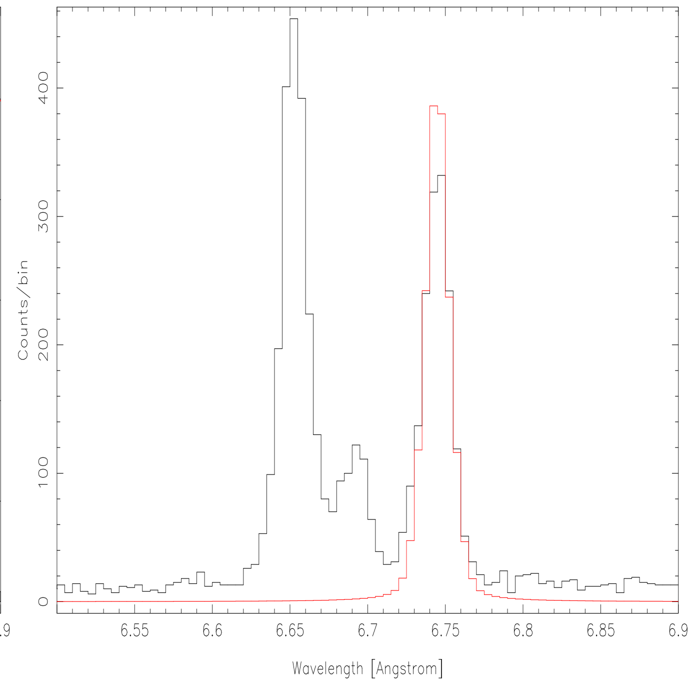

Figure 5: The Si line at 6.75 Åin the Capella spectrum with the fitted lsf shown in red.

```
isis> (lo, hi, dat, dat_err) = get_data(10);
isis> (lo, hi, model) = get_model(10);
isis> limits;
isis> xrange(6.5,6.9);
isis> hplot (lo, hi,model);
isis>print,sum(model);
0.00001527
```

5 Line Widths

5.1 Fitted LSF Parameters

The simulated line profiles, from MARX, are fit with a Gaussian plus Lorentzian profile.

$$lsf(r) = \frac{1}{2\pi} e^{-\frac{(r-r_0)^2}{2\sigma^2}} + \frac{a_0 \Gamma}{2\pi((r-r_0')^2 + (\Gamma/2)^2)} \tag{1}$$

where r_0 is the peak of the Gaussian, σ is the Gaussian width and r is the distance from the peak. For the Lorentzian component Γ is the Lorentzian width, r_0' is the Lorentzian center not necessarily the same as the Gaussian peak, and a−0 is the relative normalization with respect to the Gaussian. Figure 6 shows the variation of the Gaussian σ and relative Lorentzian normalization as a function of wavelength while figures 7 and 8 shows the same for the HEG and LEG spectra.
In these fits the Gaussian peak is set to 1.0 and the Lorentzian Γ is fixed at 0.012 Å.
As can be seen in figure 6 the Gaussian width between about 10 and 40 Årises from a low of ∼0.007 to a high of about 0.0105 Å. The right panel shows the Lorenzian amplitude (relative to the Gaussian) as a function of wavelength. At long wavelengths the Lorentzian is a weak component but at short wavelengths the Lorentzian component dominates over the Gaussian component. This effect is expected and is mostly

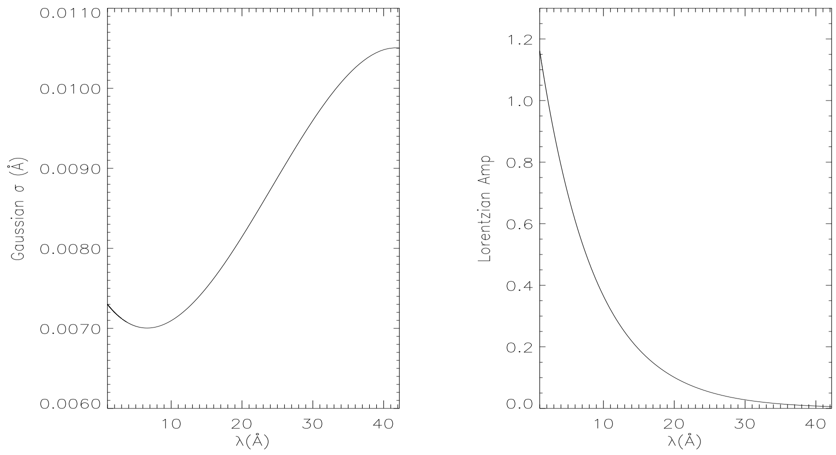

Figure 6: MEG Gaussian Width and Lorentzian amplitude vs wavelength

due to mirror scattering (see POG figure 4.6) and so the scattered component, represented by the Lorentzian, dominates at short wavelength (high energy).

6 Line Position

Line positions are one of the fundamental parameters for any line emitting plasma and is one thing that high resolution spectra can determine accurately.
In general line fitting will determine the line position along with the other parameters needed to fit the data. To determine the line position accurately some of the same considerations for determining the line flux come into play. In the case that the instrumental arf has significant structure or a slope then forward folding a model is the most accurate method to determine the line position. If the spectral data has a significant continuum slope then the line centroids from fitting or from forward folding may be systematically shifted without an accurate model for the background. The level at which the user needs to determine the accuracy of the line position determines the degree to which these concerns need to be addressed.
Once the line is fit using either RMF or without, using a simple Gaussian for example, the line center is given in the fit parameters. To determine the uncertainty for the line center,

```
isis> (low, high) = vconf(param_index);
isis> print,low,hi;
```

where the `param_index` depends on the order in which the fit function is set up. A simple way to determine this is to use the isis command

```
isis> list_par;
```

7 LETG and Higher Orders

The LETG has significant contributions of higher orders (orders>1) in the extracted spectrum. These contributions are most noticeable at large dispersion distances (long wavelengths) and at wavelengths above ∼155 Åcan dominate over the first order spectra. This can be seen in figure 9 which plots the effective area of the HRMA + LETG + HRC-S vs dispersion distance.

7.1 LETG Multi-order RSP Files

Making LETG + HRC-S RSP matrices

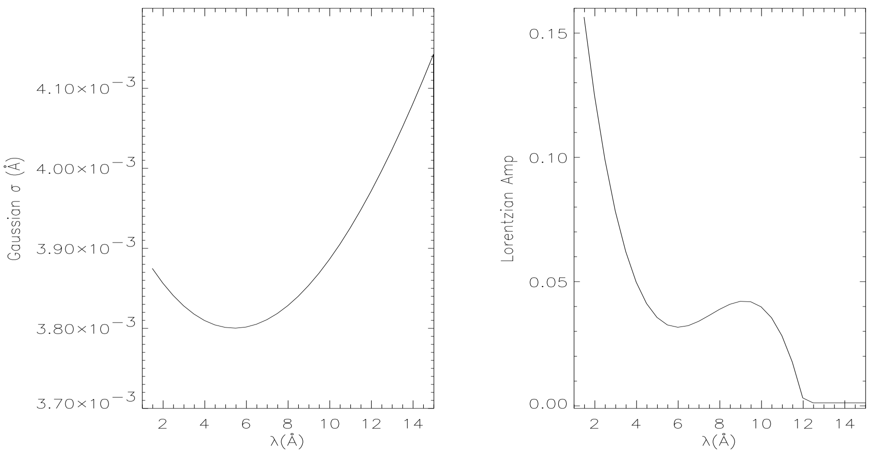

Figure 7: HEG Gaussian Width and Lorentzian
amplitude vs wavelength

The generation of an RSP matrix requires as in input an rmf and an arf. The rmf is supplied as an off-the-shelf product and is not intended for general analysis - for instance the LSF is not quite correct so that the analysis of a spectral line will yeild incorrect results. The arfs must be generated for the observation that is to be analyzed. I used the script mkgarf_hrcs that is available from

`http://asc.harvard.edu/ciao/download/scripts/mkgarf_hrcs`

to generate the arfs. Once the arfs have been generated for the rmf files then I use the ftools marfrmf and addrmf. marfrmf multiplies the arf and rmf to make the rsp files and addrmf combines the rsp files for different orders. I wrote a simple script to do this.

```
#! /bin/tcsh -f
# -*- sh -*-

source ftools


marfrmf "leg1_test.rmf[1]"  "leg1_test_arfLEG_1_garf.fits" leg_1_test.rsp
marfrmf "leg2_test.rmf[1]"  "leg1_test_arfLEG_2_garf.fits" leg_2_test.rsp

addrmf leg_1_test.rsp,leg_2_test.rsp 1.0,1.0 leg_mult.rsp

#exit
```

Where the "source ftools" is just to setup my version of ftools. The example above only sums the first and second order rsp files but in principle any number of orders may be summed.

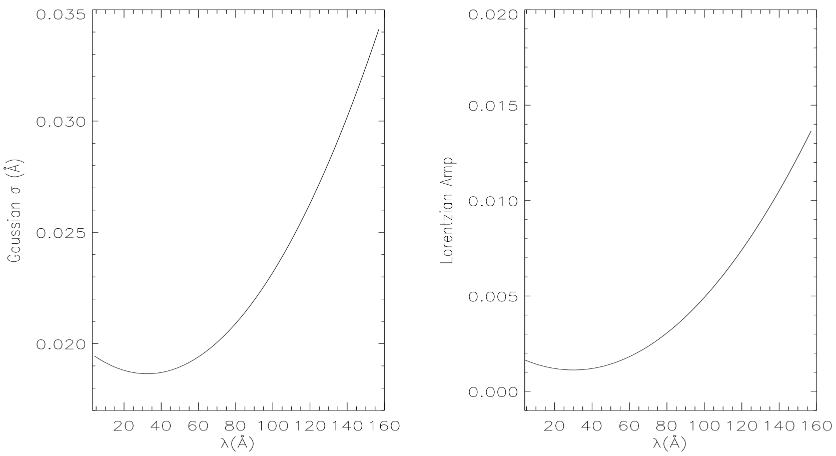

Figure 8: LEG Gaussian Width and Lorentzian amplitude vs wavelength

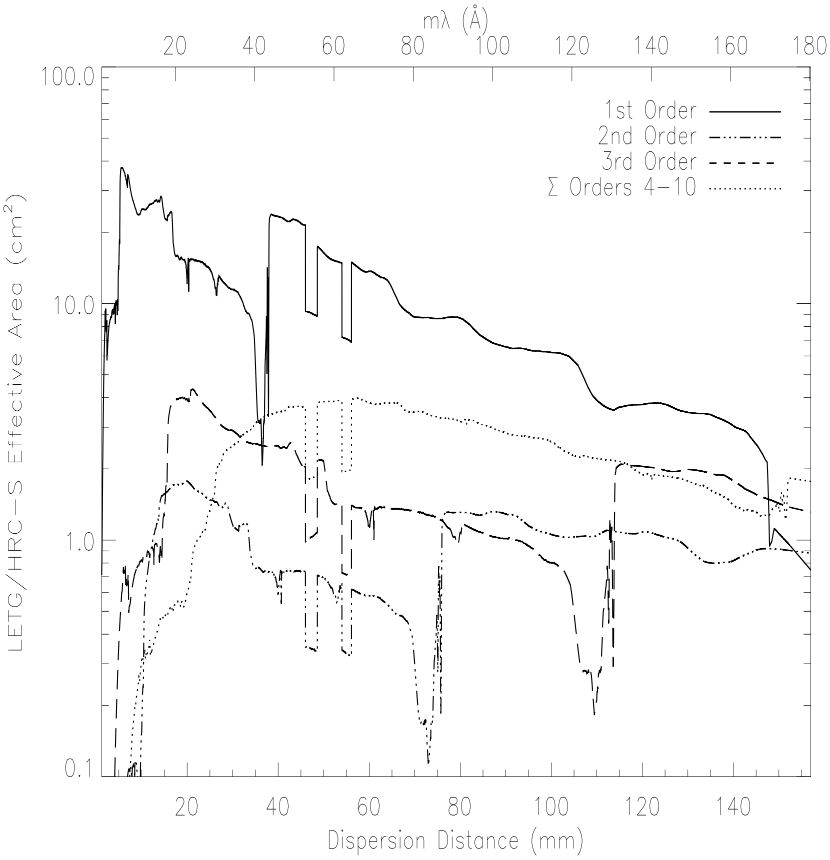

Figure 9: LETG Effective Areas