Spectroscopic Data Reduction Example

Below is a tutorial on how to reduce spectroscopic data from the Kast instrument on the 3-m Shane telescope. This is not the only, nor the definitive, way to reduce the data, but will give you an idea how to use some of the tools in IRAF.

Example Data Logsheet

FileName	Object	ExpTime(s)	Grating/Grism	Slit(“)	PA(deg)	Comments
r100.fits – r110.fits	bias	0.0	600/7500	2.0	90.3	Tilt=11000
b100.fits – b110.fits	bias	0.0	600/4310	2.0	90.3	d55 dichroic
b112.fits	Arc HgCdHe	30	600/4310	0.5	90.3	
r112.fits	Arc NeHgAr	10	600/7500	0.5	90.3	
b113.fits – b132.fits	Dome flat	60	600/4310	2.0	90.3	blue lamp
r113.fits – r132.fits	Dome flat	15	600/7500	2.0	90.3	blue lamp
b133.fits	BD+284211	120	600/4310	2.0	119.2	Standard Star
r133.fits	BD+284211	120	600/7500	2.0	119.2	Standard Star
b136.fits	IAU2130+099	900	600/4310	2.0	192.2	airmass=1.13
r136.fits	IAU2130+099	900	600/7500	2.0	192.2	airmass=1.13

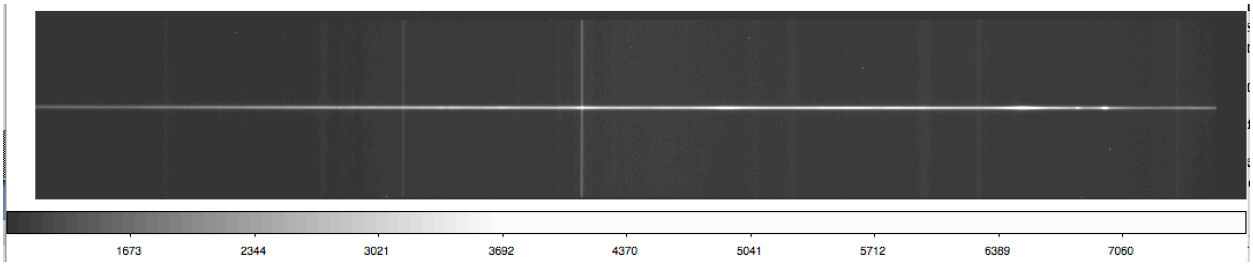

Figure 1: Raw data image, b136.fits.

Note, that starting 2016 September 20, the Kast Red detector was updated and now has the dispersion direction along columns rather than along rows (or lines in IRAF terminology). This will change some of the IRAF commands in the following tutorial that need to know the dispersion direction (e.g. response, identify, reidentify, transform, background). Sample images

from the red side are shown in Appendix A rather than in the main document due to the different orientation of the detector. Figure 1A shows raw data image r136.fits.

Overscan subtraction

For Kast spectra Lick Observatory provides a python script called overscanLickObs.py and an IDL script called kastbias that does overscan subtraction, which can properly overscan subtract Kast data when one or two amplifiers are used. Current versions of both scripts can be downloaded from the Mount Hamilton instrument web site.

Below are the commands used to overscan subtract the data files in IDL (this assumes you have the kastbias.pro program from Lick Observatory installed and IDL started). Syntax of kastbias is

kastbias,dataFileName,overscanSubtractedFileName

The overscanSubtractedFileName can be anything you wish, but in this case I'll append a tag of 'os' to the new file name to indicate it is overscan subtracted.

kastbias,'r100.fits','r100os.fits'

kastbias,'b100.fits','b101os.fits'

kastbias,'r101.fits','r101os.fits'

etc. for each file. It is possible to write scripts to loop through the files and batch process overscan subtraction in IDL, but that is outside the scope of this tutorial and left as an exercise for the interested student.

To use the overscanLickObs.py script, you must first create text file containing a list of input files, e.g. allfiles.list, and another containing a list of output file names, e.g. allfiles_os.list. This program is convenient because it batch processes all the files rather than operating on one file at a time like the kastbias IDL script. Syntax of overscanLickObs.py is

overscanLickObs.py –f –i inputFileList –o outputFileList

For example:

overscanLickObs.py –f –i allfiles.list –o allfiles_os.list

It is also possible to do overscan subtraction in IRAF using the colbias command, but Lick Observatory does not provide a prepared script to read the FITS headers and handle multiple amplifiers that are used with most Lick cameras. Hence, I will not cover that command in this tutorial.

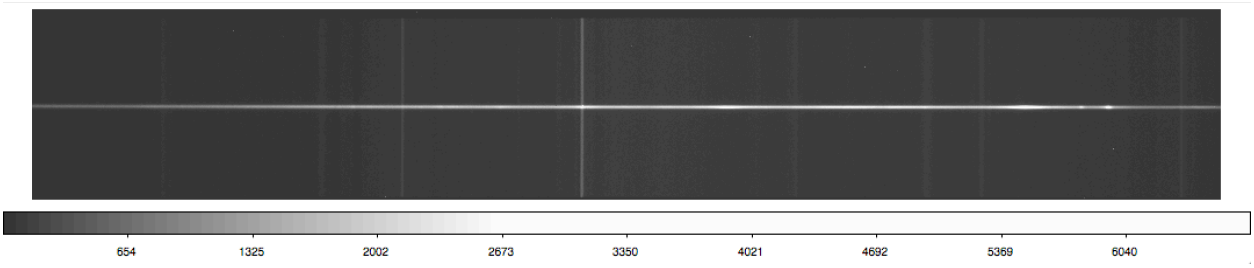

Figure 2: Overscan subtracted image, b136os.fits. Notice that the overscan region at the very right of Figure 1 is not present in the overscan subtracted data.

IRAF Basics

Note that from this point forward in the tutorial, IRAF will be used, so it is assumed that you have DS9 running for image display and cl (command language) running in an xgterm. How to start these programs depends somewhat on the computer system, so I can't give general instructions here. However, once in IRAF things work more or less the same regardless of operating system.

In IRAF there are packages which contain groups of procedures called tasks. Packages have names like noao or fitsutil and are differentiated from tasks with a trailing period when listed in cl, e.g. "noao.", as opposed to tasks which have no trailing period when listed in IRAF. To get a list of tasks in a package directory, type "?" at the cl prompt. To get help on a package or task type "help" and the package or task name, e.g. "help imcombine".

Each task has parameters that can be set or altered on the command line or via epar. To use epar just type "epar" and the task name, e.g., "epar imcombine". You can use the arrow keys to move up and down the list of parameters and set them appropriately. Parameters in parentheses are optional, while the others are not. If you do not set a required parameter, you must provide it on the command line or will be prompted for it at the time you run the task. You can exit epar by typing "Ctrl-d" after you are done setting parameters.

When a task can take multiple files as input or output you can either enter each file on the command line, or make a text file that contains the files names (one per line). To tell IRAF that the filename is a list of files, you use "@" before the file name. For example, the following two commands do exactly the same thing (NB: IRAF generally assumes that all files are FITS files, so that you need not include the .fits, or similar, extension).

imcomb r101,r102,r103,r104,r105 rcombined

imcomb @rfiles rcombined

where rfiles is a text file containing the following:
r101
r102
r103
r104

r105

Displaying your images after every analysis step is recommended to make sure that everything has run properly. This can be done directly from DS9 by selecting File - Open or from IRAF using the display command, e.g.

display r101 1

which will display file r101.fits to frame 1 in DS9.

IRAF is a very powerful tool and is feature rich. This can make it frustrating to use at times and it is not necessarily the most user-friendly software. However, working with experienced users and asking advice of colleagues and advisors can help you accomplish amazing data reduction tasks and help you get the most out of your data.

Bias Frame Subtraction

The bias frames, b100os.fits through b110os.fits (r100os.fits through r110os.fits for the red side), need to be combined to create a median bias frame that will be subtracted from all the data files. Even though the overscan has been subtracted, there may still be some residual bias structure that should be removed. Imcombine is the IRAF task suited to combining images. NB: In IRAF you don't need to write the full task name, but you need to use enough of the name to uniquely identify it. For this example the biases would be combined as follows:

imcomb b100os,b101os,b102os,b103os,b104os,b105os,b106os,b107os,b108os,b109os,b110os bbias combine=median reject=minmax

imcomb r100os,r101os,r102os,r103os,r104os,r105os,r106os,r107os,r108os,r109os,r110os rbias combine=median reject=minmax

First the bias files to be combined are listed, then the combined file base name is specified (in this case bias, so the output file name will be bias.fits).
Combine is the way you want to combine images, either median, average, or sum. Finally you can choose to discard bad data using the reject options: none, minmax, ccdclip, crreject, sigclip, avsigclip, or pclip. You should refer to the help documentation to determine which is best, but in general for bias subtraction minmax will adequately discard cosmic rays or other bad data points.

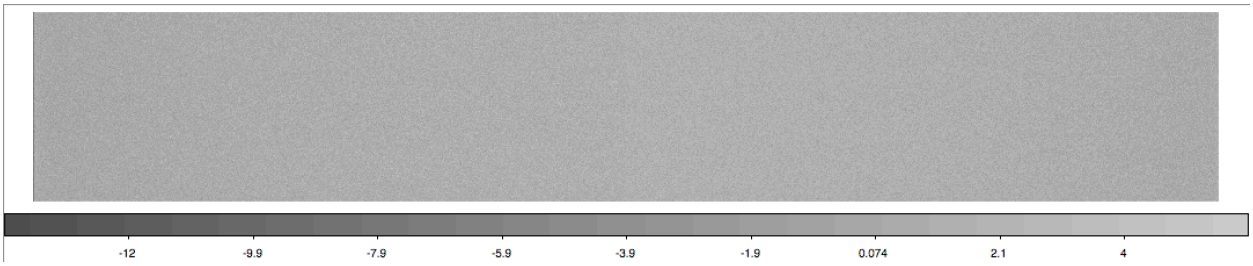

Figure 3: Combined blue bias frame, bbias.fits. For red bias frame, see Figure 3A.

Next the bias frame is subtracted from all the other frames. This is easily done using the imarith task. For these data the commands would be

imarith r136os - rbias r136bs

imarith b136os - bbias b136bs

imarith b112os - bbias b112bs
etc.,

where r136bs, b136bs, b112bs, etc. are the output bias subtracted images. You will want to make sure that all flat field frames, arc lamps, standard star, and data frames (as well as any other calibrations frames) are bias subtracted before continuing. You can also create text lists of input and output file names and use the following commands:

imarith @r_os.list – rbias @r_bs.list

imarith @b_os.list – bbias @b_bs.list

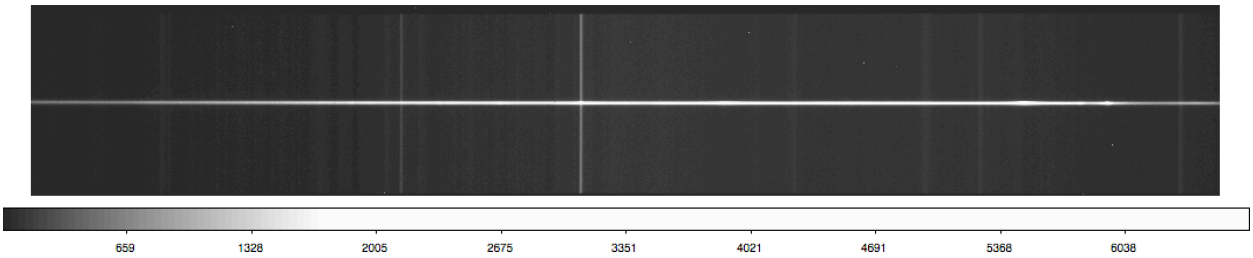

Figure 4: Bias subtracted blue data frame, b136bs.fits.

Create Dome Flat

In these data the dome flat field frames associated with the science data image are files b113bs.fits through b132bs.fits (red side dome flats are r113bs.fits through r132bs.fits). Note that we need to use the bias subtracted flat field frames. These need to be combined into a single image, so we'll once again use the task imcomb, Because there are so many flat field images, it is convenient to list them in a text file, e.g. bflat.list and rflat.list, rather than listing each file individually on the command line. We can also reject high and low pixel values in the average by setting reject=minmax.

imcomb @bflat.list bflat combine=average reject=minmax

imcomb @rflat.list rflat combine=average reject=minmax

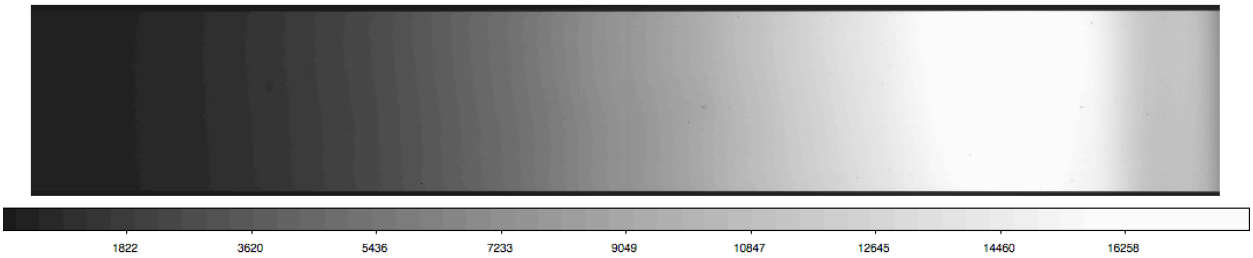

Figure 5: Combined blue flat field, bflat.fits.

Next the average spectrum will be fitted so that the spectral energy distribution (SED) of the lamp can be disentangled from the pixel response. The flat field frame will then be divided by the fitted SED and the data normalized. Notice that in these data there is atmospheric absorption in the red spectrum. When fitting a curve to the average spectrum in response, you must be careful not to do too high order a fit since you don't want to fit the absorption feature (or fringes, if they are present) but rather the SED of the flat field lamp. The IRAF task for creating the normalized flat field is called response. Response is in the noao set of packages, so you will have to load the appropriate packages so you can use the task. You load packages simply by typing their names, in this case the packages would be
noao
twodspec
longslit

The syntax for response is as follows:

response calibration normalization output

In most cases the calibration and normalization files are the same. In this case the commands are

response bflat bflat bflatn

or

response rflat rflat rflatn

which starts an interactive task. It is beyond the scope of this tutorial to go over everything interactive tasks can do, but the most important functions will be covered. In this case the following questions will be asked:

Fit the normalization spectrum for bflat interactively (yes):
Dispersion axis (1=along lines, 2=along columns, 3=along z) (1:3) (1):

In this case for the blue side data, you'll want to select Dispersion axis 1 (along lines). For red side data you'll want to select Dispersion axis 2 (along columns). A plot will be displayed in the graphics terminal showing the average flat field spectrum and the fit (see Figure 6).

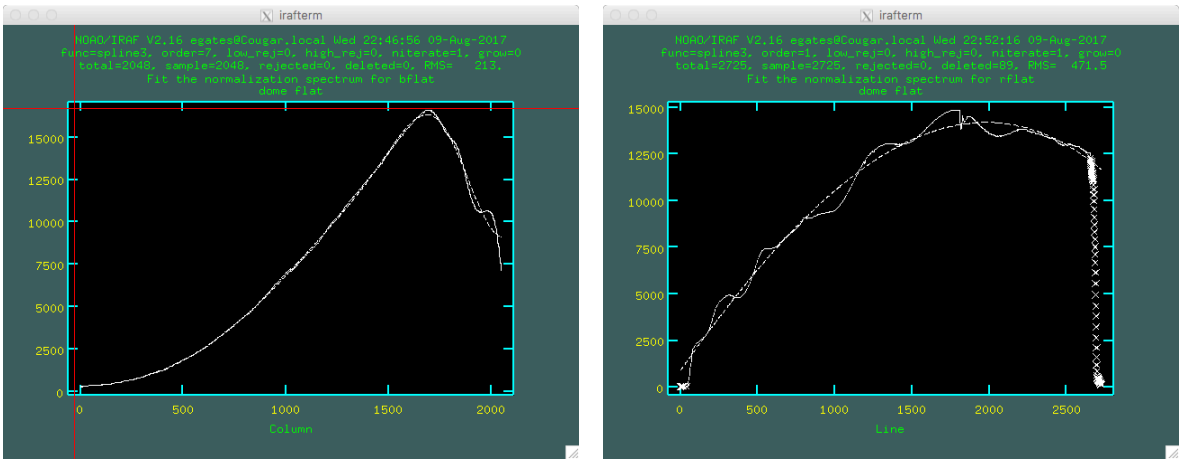

Figure 6: Response low order fit to SED, blue on left, red on right. Note that the red side has rejected data points at the extreme ends of the spectrum caused by vignetting by a mask in front of the detector, that are not included in the fit.

You can use colon commands when the cursor is in the window to change parameters of the fit. For example, :function chebyshev will change the fitting function from spline3 to chebyshev (the other option for function is legendre). :order will change the order of the fit, e.g. :order 7 will change the fit from order 1 (shown in Figure 6 above) to 7 for the blue side. To apply the new fit parameters type f in the window and you will see a new plot with the new fit. If there is a bad point in the data (in the red data the first and last data points are bad) you can place the cursor over it and press d to delete it from the fit (and u undeletes). There are many other commands, but these will be among the most frequently used. When you are happy with the fit type q to exit the interactive fitting mode. Figure 6 shows the final SED fits for the blue and red side flat fields. Figure 7 shows the normalized blue flat field (Figure 7A shows the red normalized flat field). The pixel values in the image should all be close to 1 after the normalization.

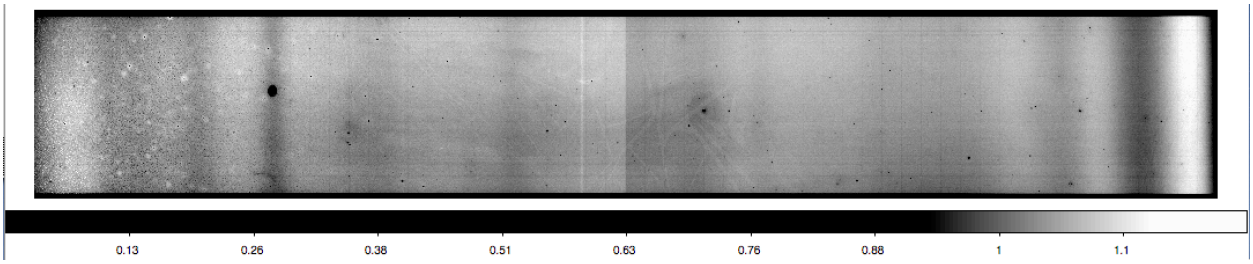

Figure 7: Normalized blue flat field frame from response, bflatn.fits.

Flat Fielding Data

Flat fielding data is straight forward as one just uses imarith to divide the data frame by the normalized flat field frame. In this case the IRAF commands would be (for the science target and standard star frames)

imarith b136bs / bflatn b136ff

imarith r136bs / rflatn r136ff

imarith b133bs / bflatn b133ff

imarith r133bs / rflatn r133ff

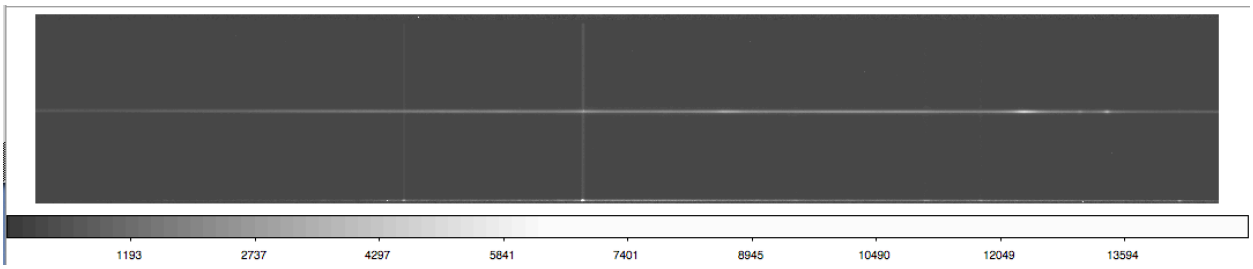

Figure 8: Flat fielded data frame, b136ff.fits.

Cosmic Ray Removal

Removing cosmic rays is optional, but can be important if a cosmic ray falls on the spectrum itself. There are a few tasks to remove cosmic rays, but we'll use task cosmicrays (in package noao.imred.crutil). For this example, the command is

cosmicrays b136ff b136cr

cosmicrays is another interactive command and you'll be asked the following question:

b136ff - Review parameters for a particular image (no|yes|NO|YES) (yes):

Answering yes will put it into interactive mode. In this case we'll say no and use the default parameters to remove cosmic rays. Cosmic ray rejection is rarely perfect, though many cosmicray events can be removed using this task.

Likewise, the commands for the red side science data and standard star are

cosmicrays r136ff r136cr

cosmicrays b133ff b133cr

cosmicrays r133ff r133cr

Wavelength Calibration

Wavelength calibration is a fairly involved process of identifying the arc lamp lines and fitting a function to associate each column of the detector with a wavelength. The task identify is often

used for creating a wavelength solution. Identify is a highly interactive routine with many commands of which we'll only touch on a few.

identify b112bs section="middle line" coordli=""

identify r112bs section="middle column" coordli=""

Identify will average the central 10 rows (blue side) or columns (red side) to create a mean spectrum which it will plot in the graphics window. To mark an arc line and input its wavelength put the cursor on the line and type m and you'll be prompted for the wavelength. If you make a mistake you can remove a misidentified line by positioning the cursor on it and typing d.

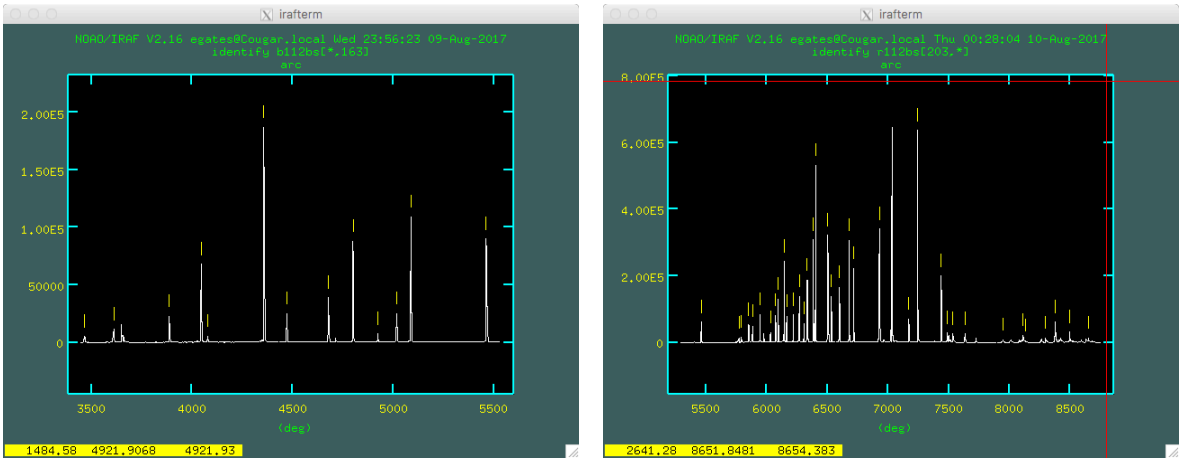

Figure 9: Identify with lines marked, blue on left, red on right.

Once the lines are identified you can fit a function to the wavelengths by typing f. The colon commands in the fitting function are similar to those in the response task above. You can delete outlying or bad points in the fit by typing d when the cursor is placed on it. The goal is to get a low RMS fit without using to high an order function. Once done with fitting, type q to get out of the interactive fitting routine. Type q again to exit identify. When queried

Write feature data to the database (yes)?

hit return or type yes to save the fit. The data describing the fit will be written to a subdirectory named database in a file named, in this case, idb112bs and idr112bs, for the blue and red detectors, respectively.

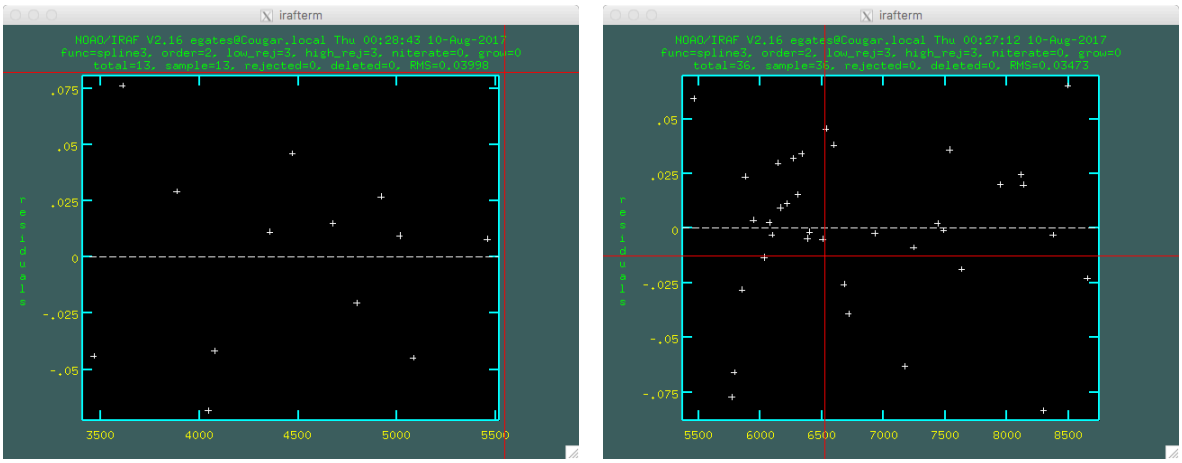

Figure 10: Identify task wavelength solution fit for blue (left) and red (right) detectors.

Now that the lines have been identified along the center rows of the 2-D spectrum, the lines need to be traced along the columns to account for any misalignments or curvature of the arc lines. The task to do this is called reidentify. The command for these data is

reidentify b112bs b112bs section="middle line" verbose=yes

Output looks like the following, where it identifies the arc lines every 10 rows through the image:

REIDENTIFY: NOAO/IRAF V2.16 egates@Cougar.local Thu 00:36:23 10-Aug-2017
Reference image = b112bs, New image = b112bs, Refit = yes

Image Data	Found	Fit Pix Shift	User Shift	Z Shift	RMS
b112bs[*,153]	13/13	-0.00386	-0.00515	-9.4E-7	0.0465
b112bs[*,143]	13/13	-0.0117	-0.0117	-2.8E-6	0.0434
b112bs[*,133]	13/13	-0.0149	-0.0154	-3.4E-6	0.0477
b112bs[*,123]	13/13	-0.0123	-0.0119	-3.0E-6	0.0421
b112bs[*,113]	13/13	-0.0141	-0.0143	-3.3E-6	0.0379
b112bs[*,103]	13/13	-0.0161	-0.016	-3.8E-6	0.0401

etc.

For the red side, since the dispersion direction is along columns, the command is

reidentify r112bs r112bs section="middle column" verbose=yes

Now a 2-D fit to the arc lines is needed so that a full wavelength solution for the full data image can be determined. The task fitcoords does this, for example,

fitcoords b112bs

where you'll be asked the following question:

Fit b112bs interactively (yes):

Respond yes and you'll be put into another interactive fitting window in the graphics terminal displaying the residuals of the fit on the y-axis and the X pixel coordinates on the x-axis.
To display the residuals versus the Y pixel coordinates type x to change the x coordinate axis and you'll be prompted for which data you want to plot (x, y, z, s, or r) and choose y. To see the new plot and fit the data type f. When you are happy with the fit type q to quit the task. At this point you'll be asked if you want to

Write coordinate map to the database (yes)?

to which you should respond yes.

Likewise, do the same for the red side:

fitcoords r112bs

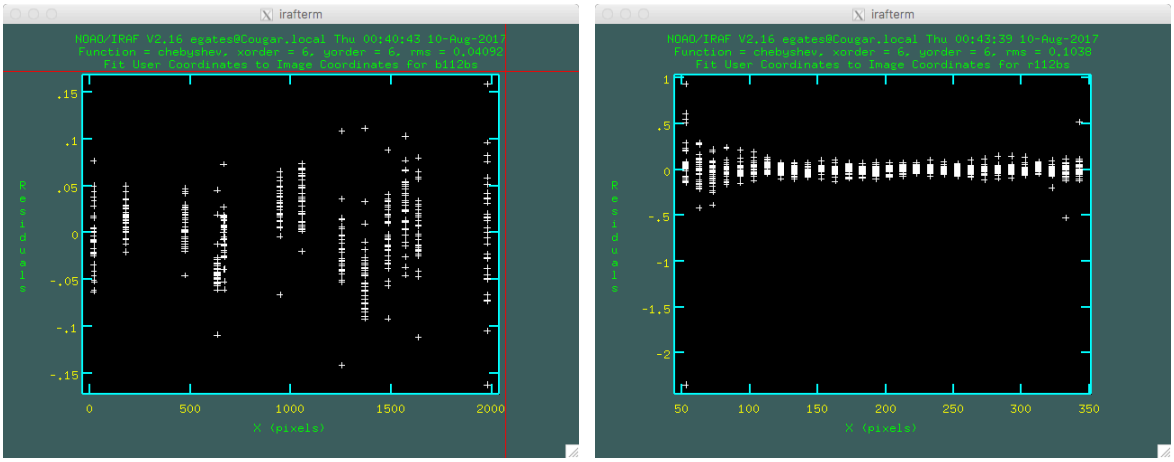

Figure 11: Fitcoords interactive fitting window, for blue (left) and red (right) sides.

Reduce Standard Star

We'll now start work on reducing the standard star so that it will be ready for fluxing the data when we get to that stage of the data reduction. The raw standard star data are images b133.fits and r133.fits, though if you have processed things correctly so far, you'll already have flat fielded and cosmic ray removed images, b133cr.fits and r133cr.fits.

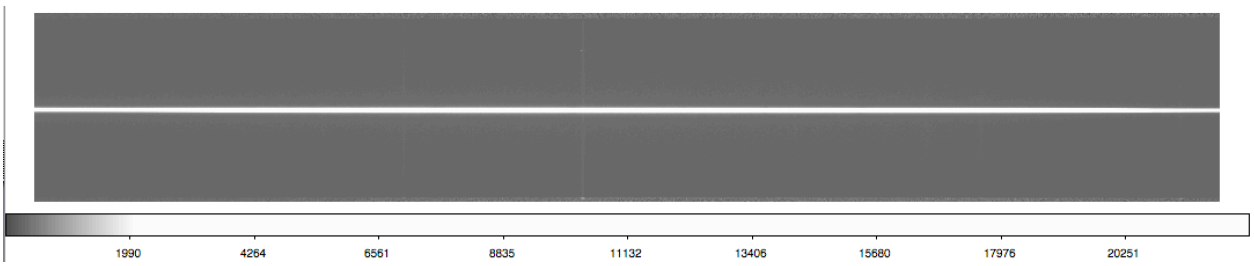

Figure 12: Flat fielded standard star, BD+28 4211 (b122cr.fits).

Because the standard star spectrum has high signal to noise, we can trace the spectrum easily across the chip. We'll do this using the identify task, on the blue side setting it to sum the 10 middle columns (instead of the 10 middle rows when identifying the arc lines):

identify b133cr section="middle column"

Mark the row which has the peak emission, but instead of labeling it with a wavelength input the row number (see Figure 13). Type q to quit out of the task and write the feature data to the database.

Similarly for the red side detector do

identify r133cr section="middle line"

Now reidentify the stellar spectrum and use fitcoords to model the path of the spectrum across the CCD:

reidentify b133cr b133cr section="middle column"
reidentify r133cr r133cr section="middle line"

fitcoords b133cr
fitcoords r133cr

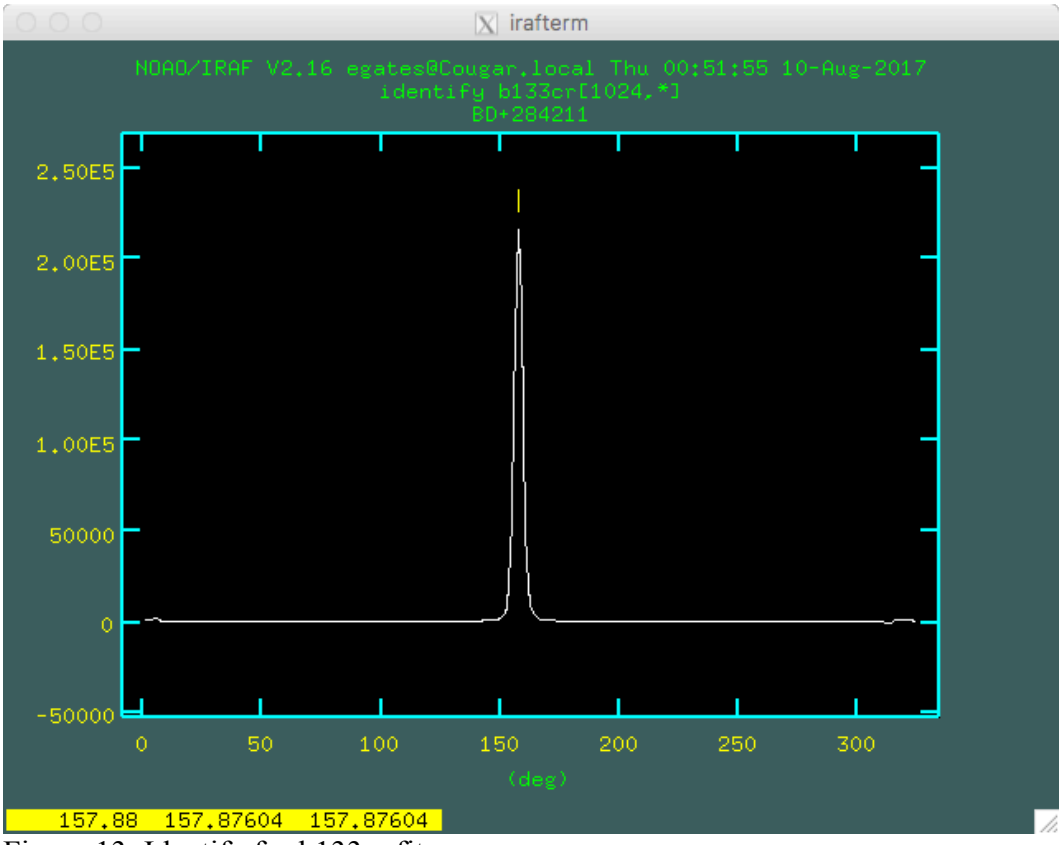

Figure 13: Identify for b133cr.fits.

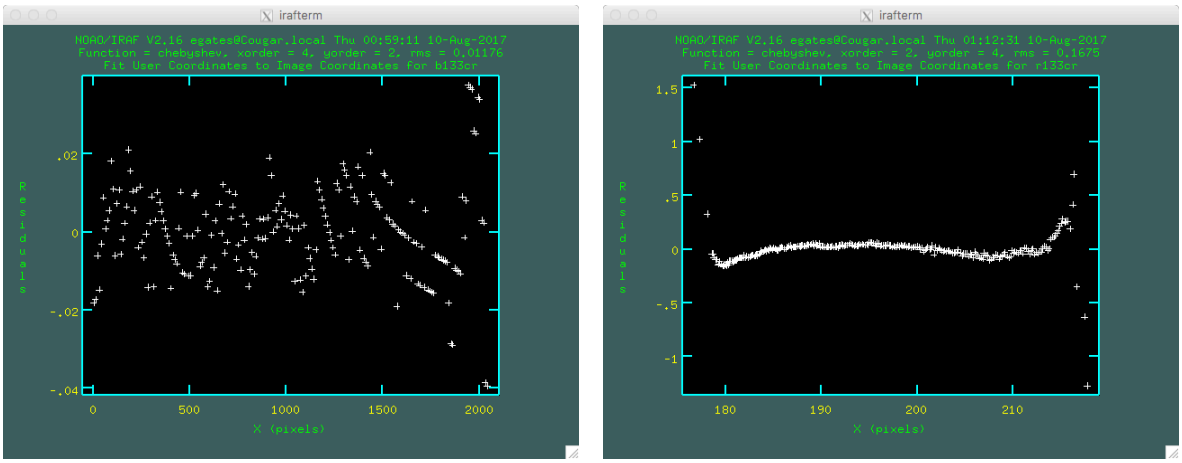

Figure 14: Fitcoords for b133cr.fits (left) and r133cr.fits (right).

Apply Wavelength Calibration

Next we apply the wavelength calibration and spectrum trace to the data using the task transform first on the standard star data:

transform b133cr b133tr b112bs,b133cr

where b133tr is the output file and b112bs and b133cr are the calibration files for which wavelength and spectrum trace have been done. You will be prompted with the following question:

Dispersion axis (1=along lines, 2=along columns, 3=along z) (1:3) (1):

In this case respond with 1 since the spectrum is aligned with the rows of the CCD for the blue side.

Likewise for the red side data do

transform r133cr r133tr r112bs,r133cr

and select Dispersion axis 2 (along columns).

You can also apply this wavelength solution to the data frames, b136cr and r136cr:

transform b136cr b136tr b112bs,b133cr
transform r136cr r136tr r112bs,r133cr

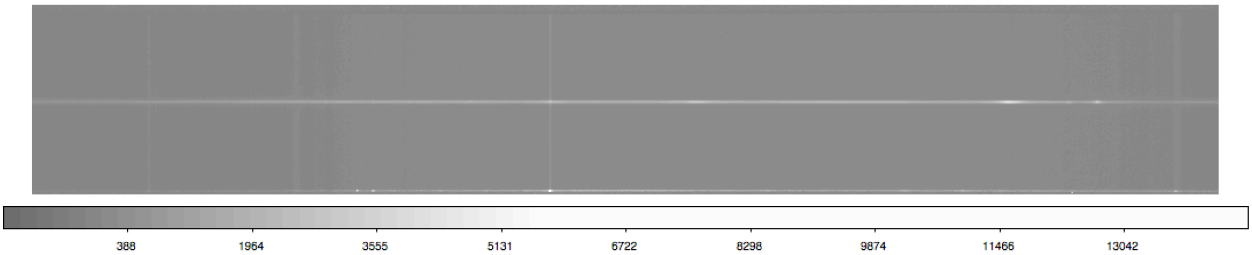

Figure 15: Transformed data, b136tr.fits (see Figure 15A for the red side transformed data).

Background Subtraction

Next the sky background, including sky lines, needs to be removed from the data frames. This is done with the task background, which is another interactive task. For example,

background b133tr b133bg axis=2

You will be prompted in the graphics terminal to enter which column you wish to use to set the background sample regions. It doesn't particularly matter which column you choose, but it is best to use one that doesn't have some obvious defects or other problem and where the object can be easily seen. If in doubt, just choose a column near the middle of the detector, e.g. 600 for the blue side.

The column will then be plotted in the graphics terminal. At this point you want to select appropriate regions for measuring the background. Position the cursor at the starting point of where you want to sample the background and type s. Reposition the cursor to the ending point of the background sample and type s again. You should see a solid line below the plot indicating the selected sample region. You may have multiple sample regions.

Once the background sample regions are selected, type f to fit a curve to the sample.

Type q to exit the sample selection and you will again be prompted for a fit column. Ignore this and just hit Return.

You can do the same operation for the science data, r129tr:

background b136tr b136bg axis=2

For the red side data, you'll want to set axis=1 and choose a row near the middle, e.g. 1350.

background r133tr r133bg axis=1

background r136tr r136bg axis=1

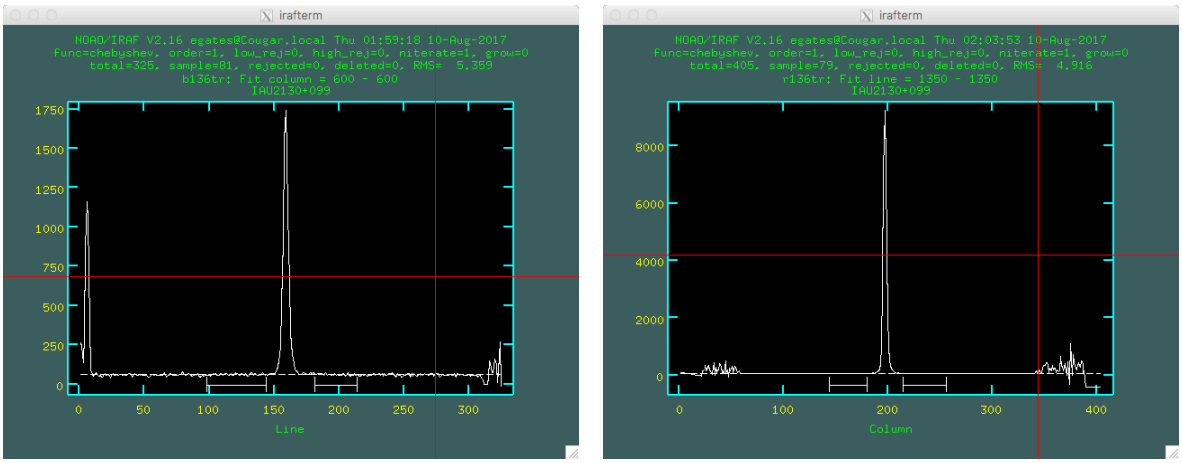

Figure 16: Background sample regions and fit for science data, b136tr.fits (left) and r136tr.fits(right).

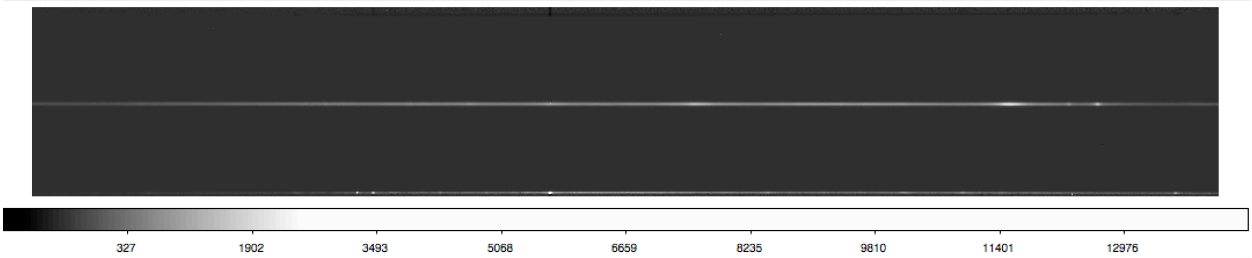

Figure 17: Background subtracted science data, b136bg.fits. (See Fig. 17A for r136bg.fits)

Extinction Correction

Atmospheric extinction needs to be corrected. IRAF comes with two extinction models, onedstds$kpnoextinct.dat and onedstds$ctioextinct.dat. These files are text files listing the wavelength in angstroms and the extinction coefficient. One can create a custom extinction file for the observatory the data was taken at, but often for lower altitude observatories the kpno file is sufficient and the ctio file is good for higher altitude observatories. As Lick Observatory is at about 1200 m altitude, we'll use the kpno extinction data.

extinction b136bg b136ex extinct="onedstds$kpnoextinct.dat"

The resulting image should look very similar to the background subtracted data, b136bg.fits.

Now do the same for the red side data image:

extinction r136bg r136ex extinct="onedstds$kpnoextinct.dat"

Flux Calibration

Flux calibration is a process of extracting the standard star data and modeling the sensitivity of the detector and instrument compared to the well know flux of the standard star as a function of wavelength. The science data are then fluxed using the sensitivity function.

First we need to extract a one dimensional spectrum from the two dimensional data. This can be done using the task apsum in the apextract package (which is in the twodspec package). Apsum is another highly interactive procedure using the graphics terminal. In this example the command would be

apsum b133bg

You will be prompted with the following questions (with suggested responses for this case shown):

Find apertures for b133bg? (yes): yes
Number of apertures to be found automatically: 1
Edit apertures for b133bg? (yes): yes

Figure 18 shows the plot of the average of the central 10 columns and default aperture and background sampling regions. Apsum usually does a good job of identifying the spectrum for bright objects, but can fail for faint objects and select a noise spike rather than the data. Often you will want to fine tune the aperture and background regions, which can be done with the following commands:

- To set the lower limit of the aperture, position the cursor in the desired position and type l.
- To set the upper limit of the aperture, position the cursor in the desired position and type u.
- To set the background, type b to get into background mode. Type t to set all points to sample region (this essentially resets the background sample).
- To select subregions for background sampling, position the cursor at one end of the desired sample region and type s. Then move the cursor to the other end of the sample region and type s again. You may select multiple background sample regions.
- To do a fit to the background over the sampled region(s), type f. At this point, you are back in the interactive curve fitting environment where you use commands such as :function and :order and refit the data using f.

When you are done setting apertures and satisfied with the background fit (see Figure 19), type q to quit the interactive fitting mode. Type q again to get out of the aperture and background setting mode. You will then be asked (with suggested answers shown for this case):

Trace apertures for b133bg? (yes): yes
Fit traced positions for b133bg interactively? (yes): yes
Fit curve to aperture 1 of b133bg interactively (yes): yes

At this point another plot will be displayed showing the row position of the spectrum for each 10 columns and a fit to the curve (see Figure 20). You will fit this curve using the interactive fitting commands such as :order and f until you are satisfied with the fit. Type q to quit the interactive fitting and you will be asked the following questions (shown with appropriate responses):

Write apertures for b133bg to database (yes): yes
Extract aperture spectra for b133bg? (yes): yes
Review extracted spectra from b133bg? (yes): yes

The extracted spectrum is shown in Figure 21. Type q to quit apsum when done reviewing the spectrum. The extracted spectrum file has the name b133bg.ms.fits.

Similarly do the same for the red side data (note that apsum knows the dispersion direction since the data have been transformed using the wavelength solution):

apsum r133bg

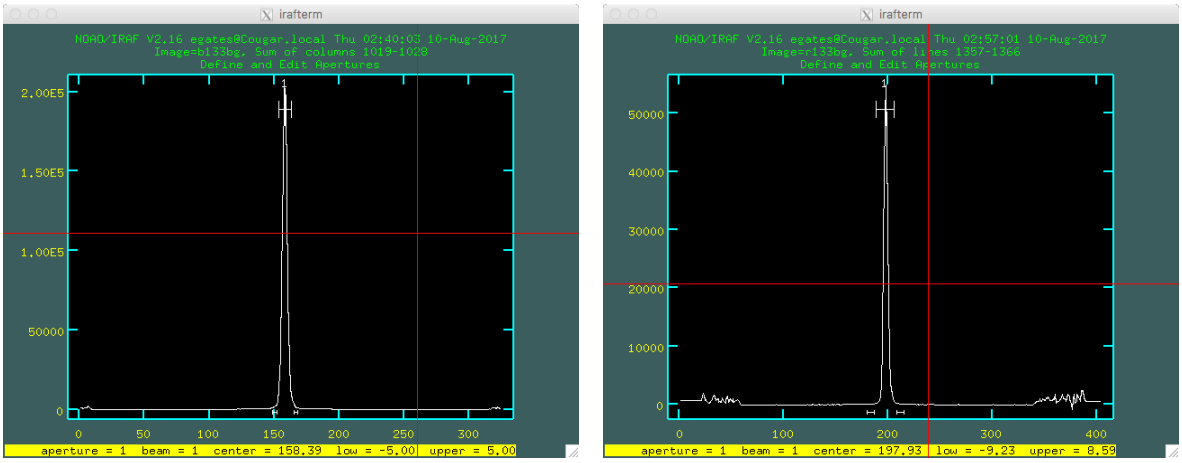

Figure 18: Apsum plot showing default aperture and background sampling regions for b133bg.fits (left) and r133bg.fits (right).

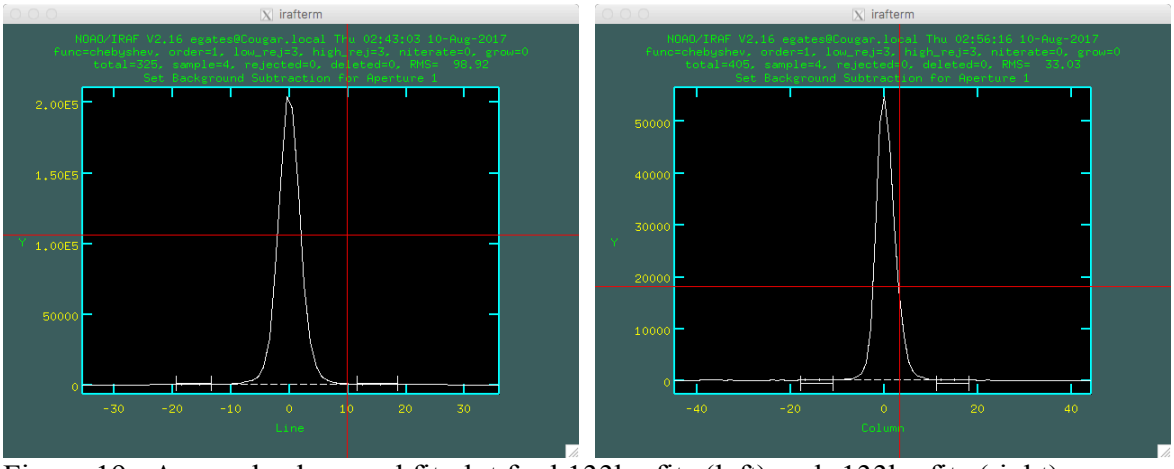

Figure 19: Apsum background fit plot for b133bg.fits (left) and r133bg.fits (right).

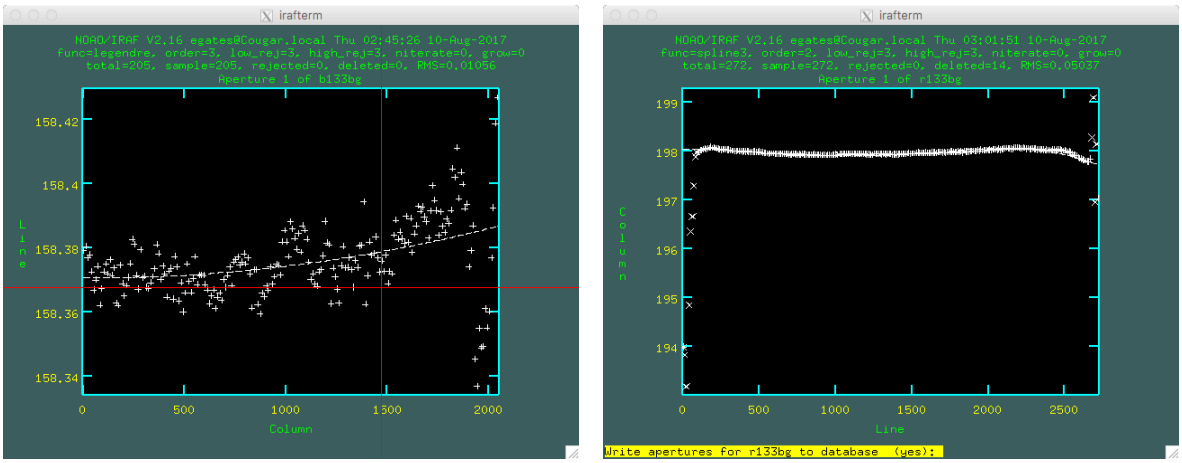

Figure 20: Apsum aperture trace fit for b133bg.fits (left) and r133bg.fits (right). Note that a number of bad pixels at the extreme ends of the spectrum were deleted from the fit to better fit the main spectrum.

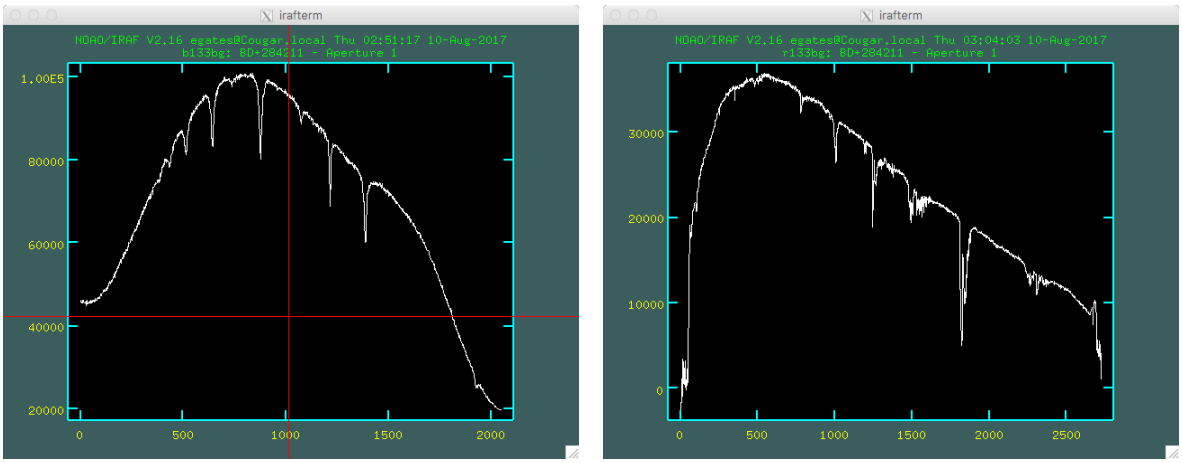

Figure 21: Extracted spectrum from apsum for BD+28 2411, b133bg.ms.fits (left) and r133bg.ms.fits (right).

Now the standard star flux must be modeled using the task standard. The command in this case is:

standard b133bg.ms b133std extinct="onedstds$kpnoextinct.dat" caldir="onedstds$irscal/" star_nam="bd284211"

There are many standard star flux data files available in IRAF, but not all standards are included. Check the onedstds subdirectories to see if the standard star data is available. If not, it you'll have to create a data file for use with IRAF for modeling the sensitivity. The star_name should match the name of the desired standard data file (without the extension).

You will be asked the following question (with appropriated responses shown):

r122bg.ms[1]: Edit bandpassses? (no|yes|NO|YES|NO!|YES!) (no): yes

Figure 22 shows the Feige 34 spectrum overlaid with boxes showing the fitting regions for the sensitivity function. Delete any regions that are contaminated with a strong absorption line, cosmic ray, etc, since the goal is to fit the continuum emission of the star, not the unique features. Individual fitting boxes can be deleted by positioning the cursor on them and typing d. You can add a new fitting band by typing a. When you are content with the bandpass selection, type q to quit the standard task.

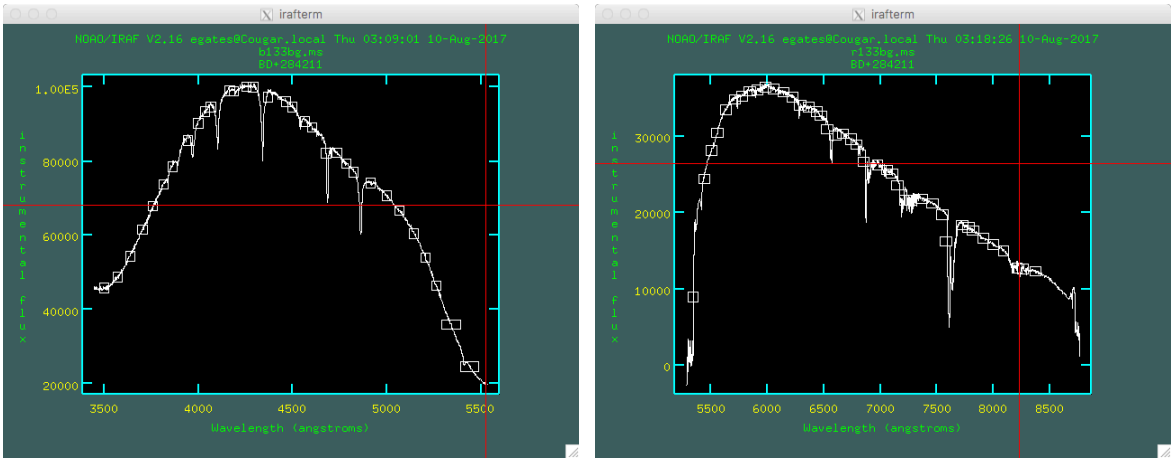

Figure 22: Standard task spectrum fitting bands, b133bg.ms.fits (left) and r133bg.ms.fits (right).

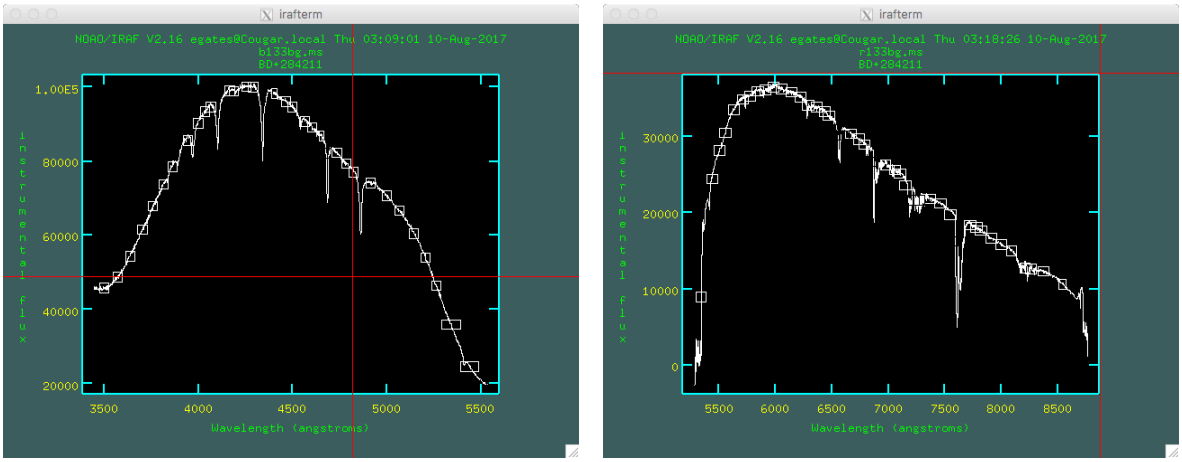

Figure 23: Fitting bands after editing, b133bg.ms.fits (left) and r133bg.ms.fits (right).

Run standard on the red side standard star data:

standard r133bg.ms r133std extinct="onedstds$kpnoextinct.dat" caldir="onedstds$irscal/" star_nam="bd284211"

Finally, the sensitivity function is actually fit the standard star with the task sensfunc. The command in this case is

sensfunc b133std b133sens

Any you will be prompted with the following question (which appropriate answers shown):
Fit aperture 1 interactively? (no|yes|NO|YES) (yes): yes

Figure 24 shows the sensitivity function plot in the top panel of the graphics terminal and the residuals of the fit in the bottom plot. The familiar interactive fitting commands are used here to adjust the fitting function and order of the fit as usual. Type q to quit sensfunc.

Run sensfunc for the red side standard star:

sensfunc r133std r133sens

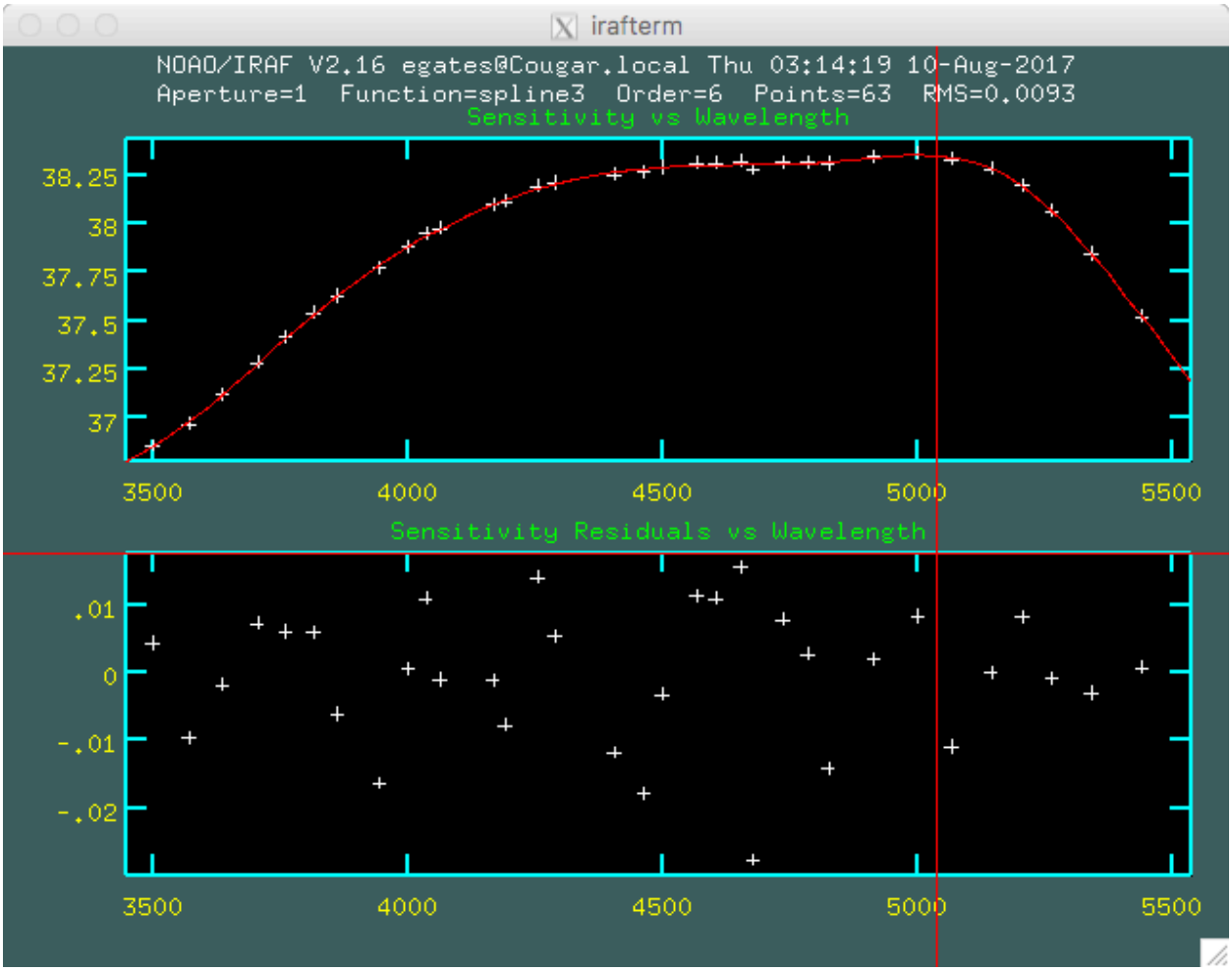

Figure 24: Sensitivity function plot from sensfunc for b133std.

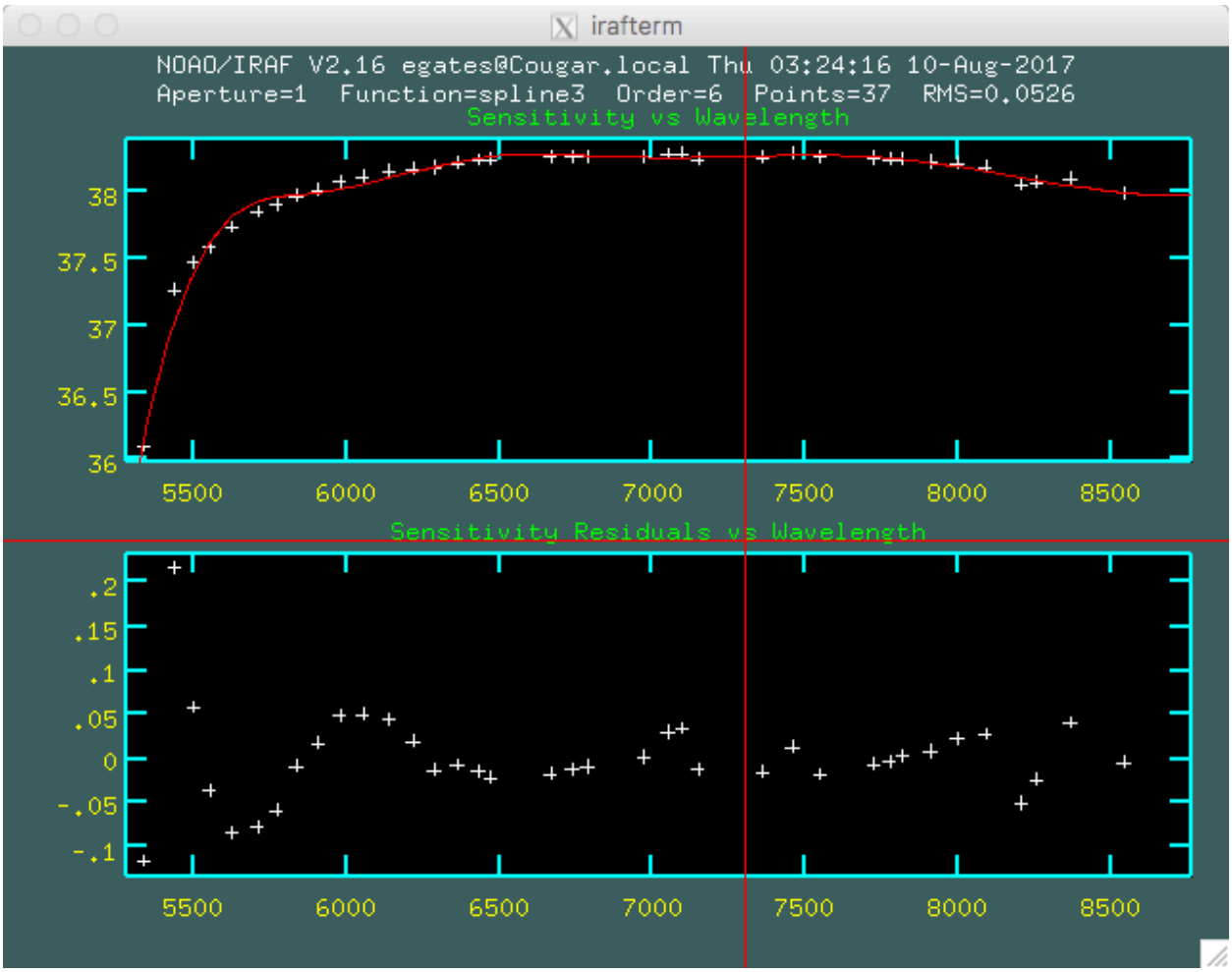

Figure 25: Sensitivity function plot from sensfunc for r133std.

Finally, we apply the sensitivity function to the data, b136ex.fits using the fluxcalib task. The command is

fluxcalib b136ex b136fl b133sens exposur="EXPTIME"

where exposur is the FITS header keyword containing the exposure time.
Likewise, for the red side data

fluxcalib r136ex r136fl r133sens exposur="EXPTIME"

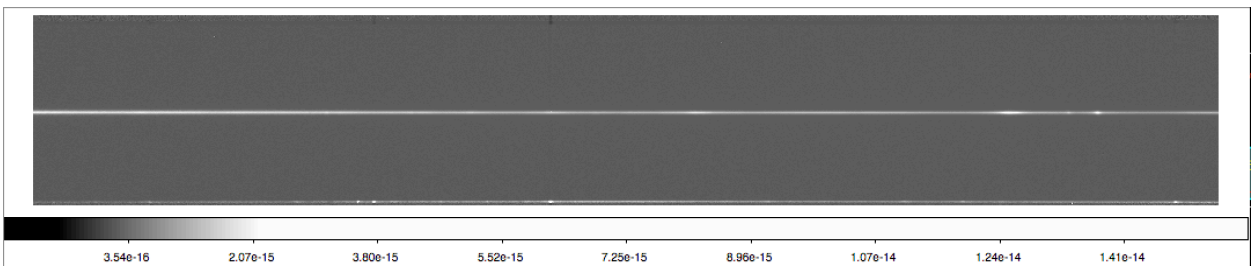

Figure 26: Flux calibrated 2D spectrum of science target, b136fl.fits.

Extract One Dimensional Spectrum for the Science Target

This is done with the apsum task, as described in the flux calibration section above. The command is

apsum b136fl

Figure 27 shows the initial plot for apsum identifying the aperture. In this case you can see that the identified aperture is for a noise spike rather than the target. There are actually two objects on the slit. The science target is the stronger signal at the center of the plot. The bad aperture needs to be deleted by typing d when the cursor is over the unwanted aperture, and a new aperture marked by typing m with the cursor over the desired object. Otherwise the apsum procedure is run in a similar manner to what was done for the standard star. Figures 28-30 show the apsum steps for the science target.

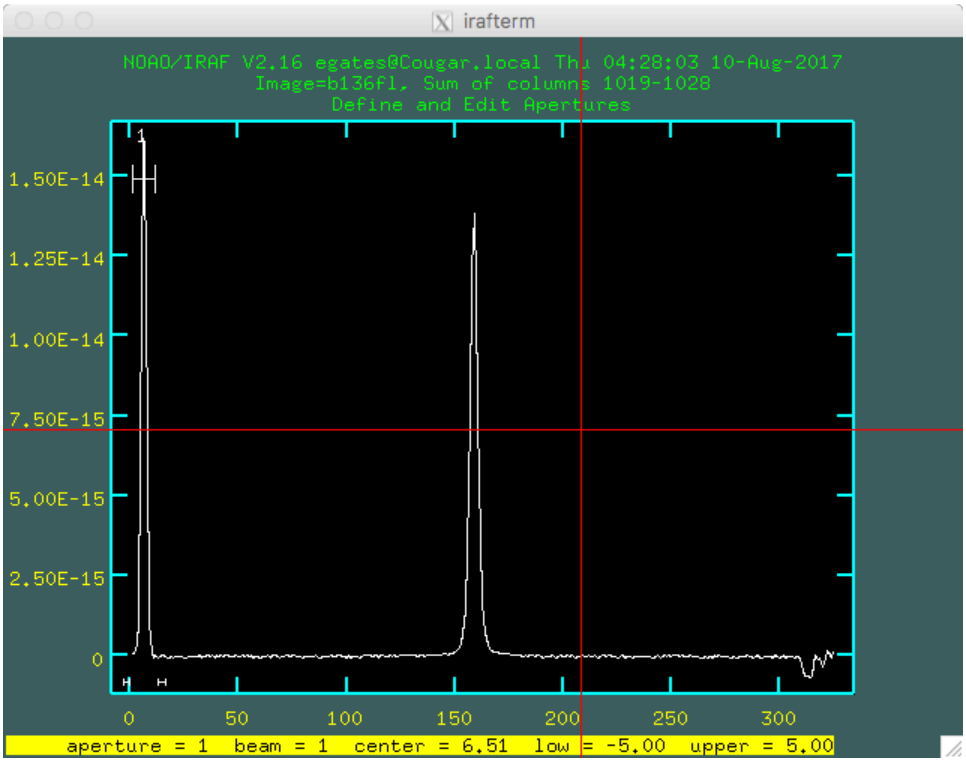

Figure 27: Apsum initial plot for b136fl.fits.

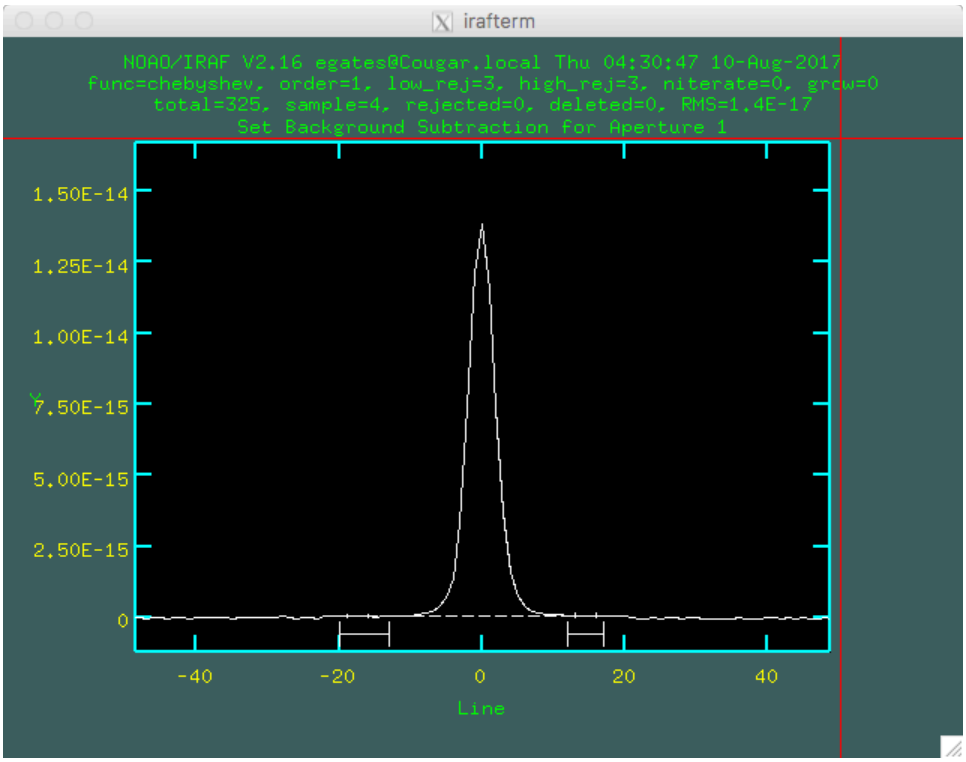

Figure 32: Apsum plot with correct aperture marked and background fitted for b136fl.fits.

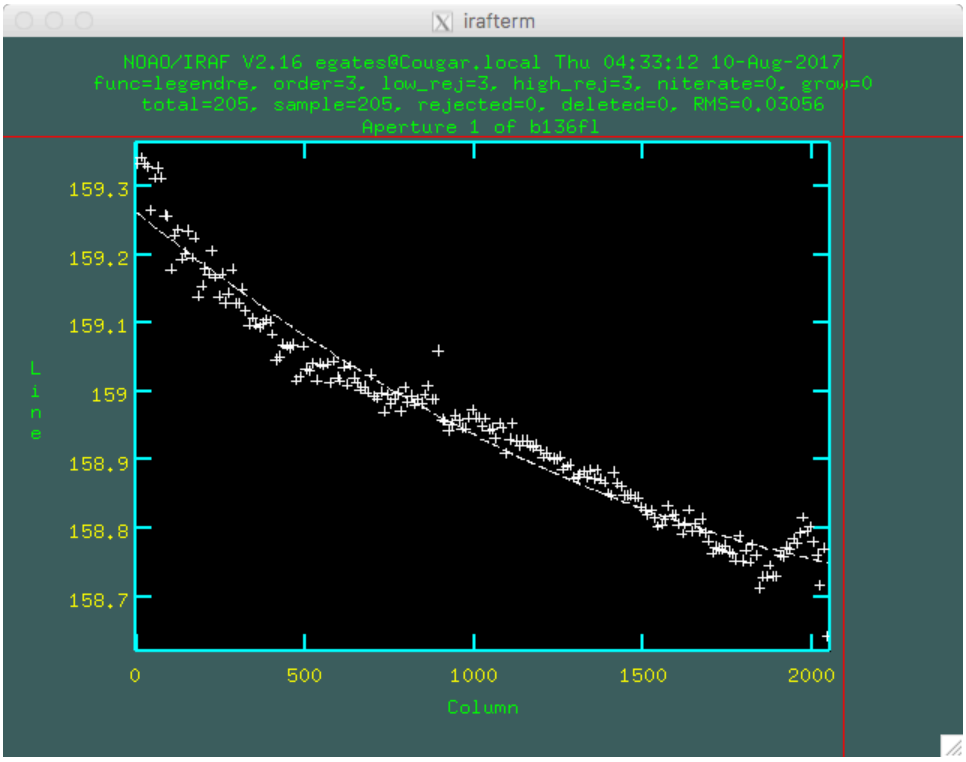

Figure 33: Apsum plot of traced spectrum with fit for b136fl.fits.

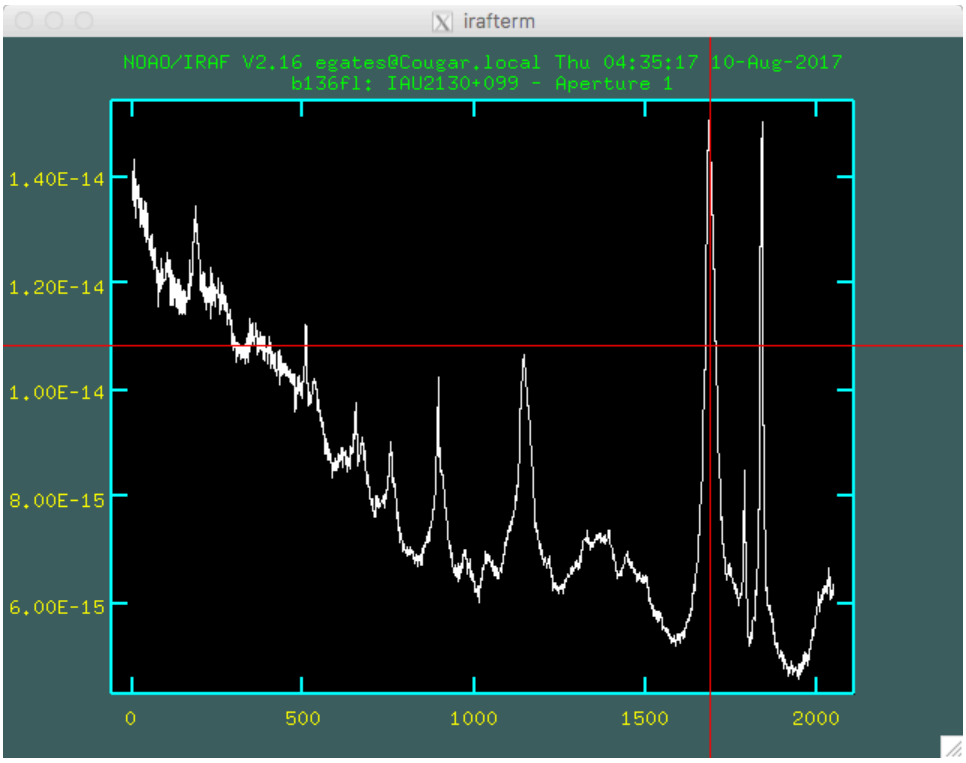

Figure 34: Extracted spectrum b136fl.ms.fits.

Do the same for the red side spectrum, r136fl.fits:
apsum r136fl

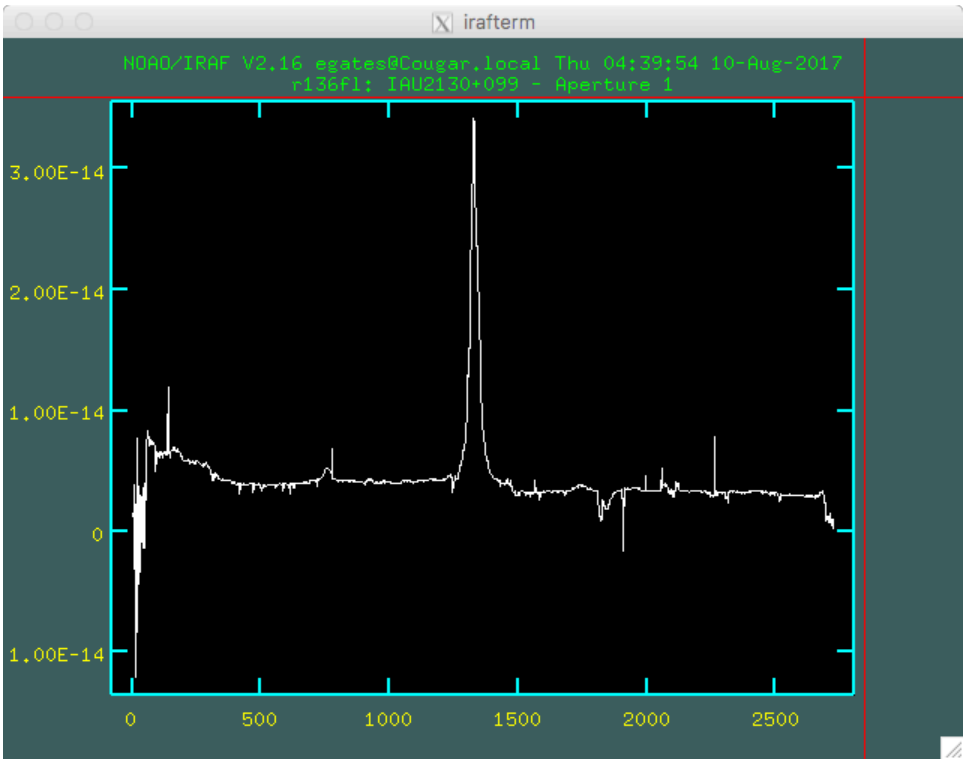

Figure 35: Extracted spectrum r136fl.ms.fits.

Analyzing Data

How you go about analyzing data depends on what information you are trying to extract. There are many tools out there, but I'll mention one tool here that is handy for doing a great number of analysis functions, splot. With splot you can examine the spectrum in detail, measure the FWHM, equivalent width, integrated flux, and center of lines, or deblend lines using profile models, as well as many other useful functions. Examining all these this is beyond the scope of this tutorial, but I will show an example of measuring the width and center of a line.

splot b136fl.ms

will show a plot similar to Figure 34. To examine a region of the spectrum in detail you can use the autoexpand command a. To use this, position the cursor at the left edge of the region you want to look at and type a, then position the cursor at the right edge of the region and type a again. Figure 36 shows the spectrum expanded around the region at 5000 Angstroms. There are a few emission lines in this region. Also shown in the figure is a fit to the leftmost emission line. The fit was done by typing e at the left edge of the line and typing e again at the right edge. The fit results are shown at the bottom of the window as well as recorded to the file splot.log.

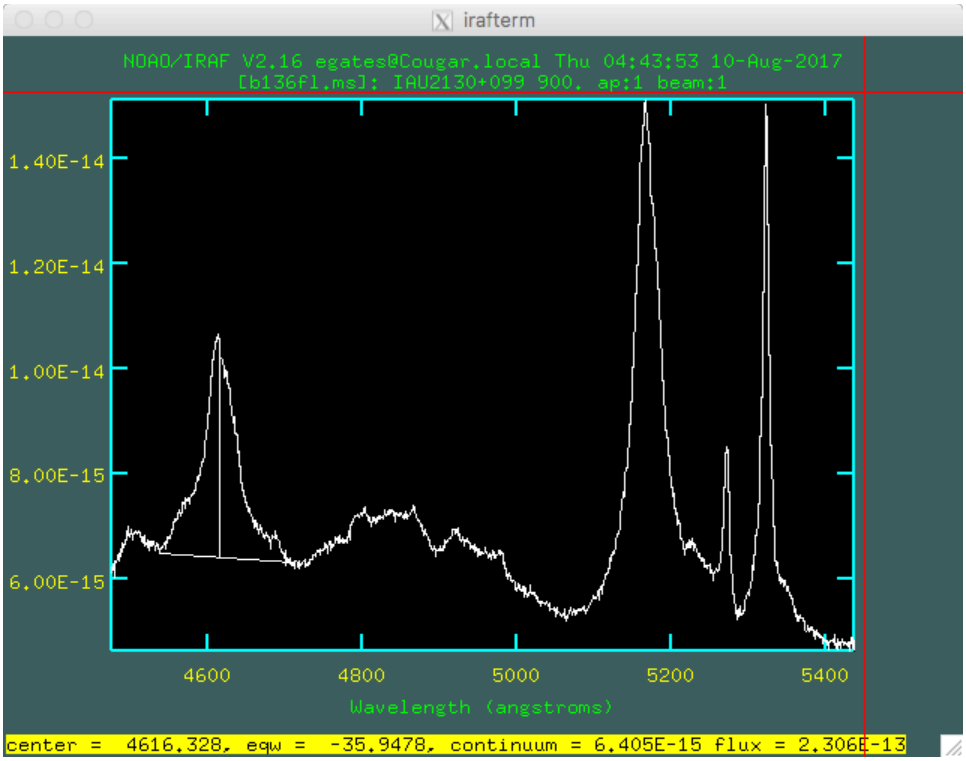

Figure 36: Spectrum expanded around 5000 Angstroms.

Appendix A

Images for the new red side detector are displayed here rather than in-line in the main document due to the dispersion being along columns rather than rows.

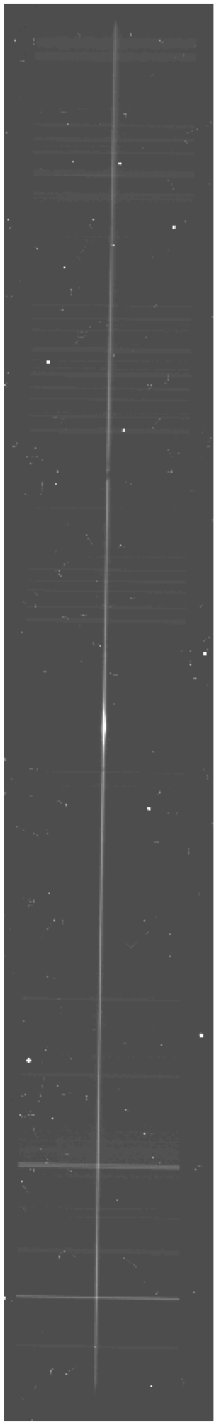

Figure 1A: Raw data, r136.fits

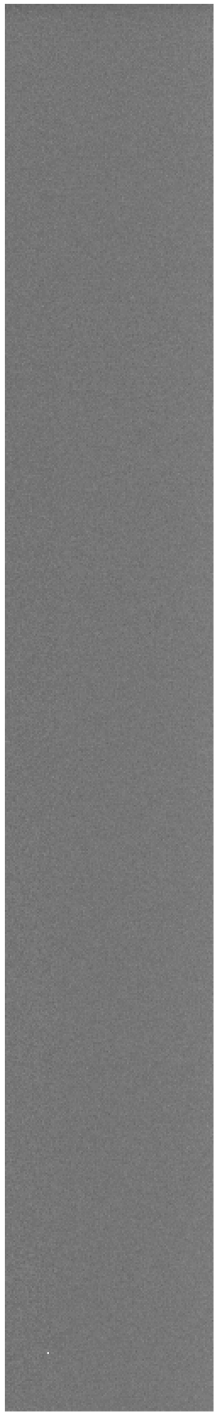

Figure 3A: Median combined red bias, rbias.fits

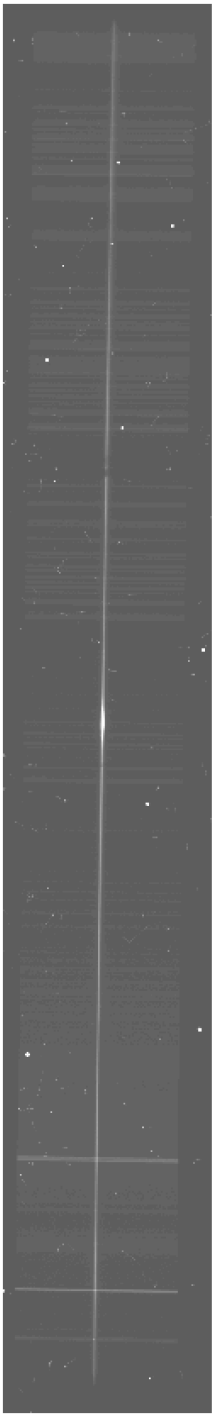

Figure 4A: Bias subtracted, r136bs.fits

Figure 5A: Flat field, rflat.fits

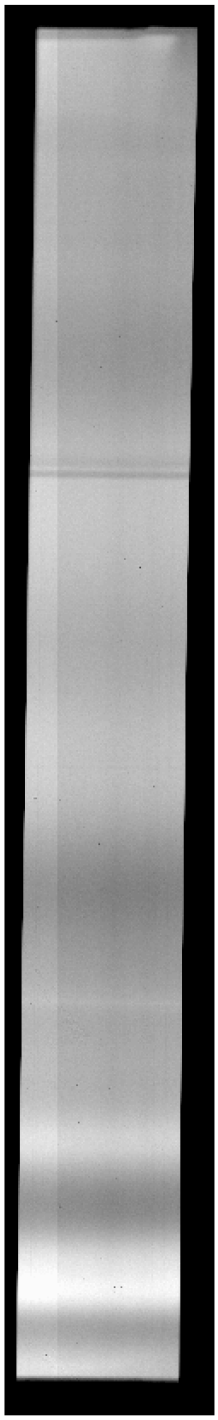

Figure 7A: Normalized flat field, rflatn.fits

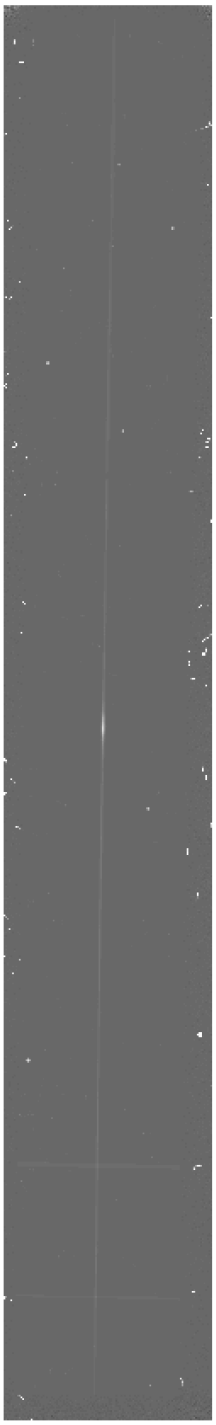

Figure 8A: Flat fielded data, r136ff.fits

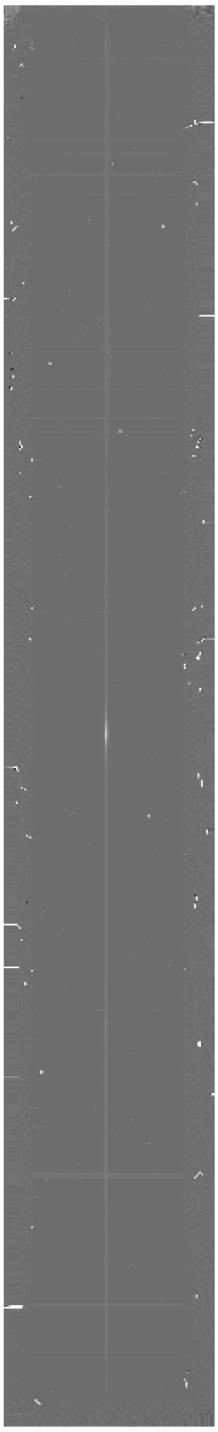

Figure 15A: Transformed data, r136tr.fits, Note spectrum now aligned with column

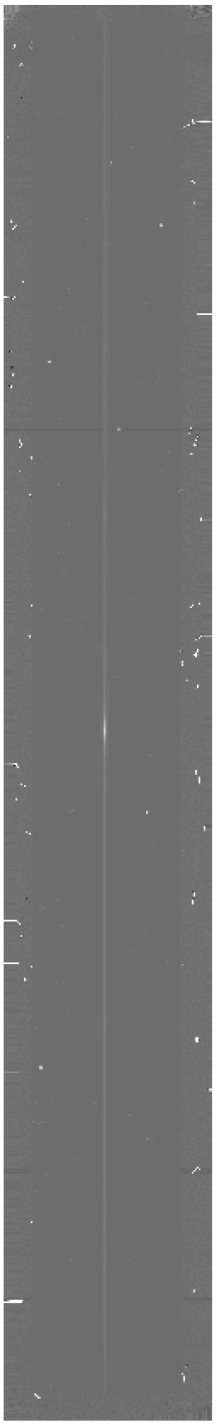

Figure 17A: Background subtracted data, r136bg.fits