Last Updated: 23 Sept 2012

Kast Reduction Guide

This is a detailed guide regarding data reduction for observations taken with the Shane 3-m telescope located at Lick Observatory atop Mount Hamilton near San Jose, CA. It is specific to spectroscopic observing with the Kast Double Spectrograph as well as to the University of California, Berkeley Supernova Research Group.

Contents

List of Figures

1 Transferring the Raw Data and Log File to Your Personal Computer

In a terminal, type `get_kast_data` from anywhere. After the data is finished downloading and unzipping you will be prompted by the program for your FlipperWiki username and password as well as the run code (in lowercase!) and the name of the wiki log page. This step will create a new folder in /Users/Kelsey/observing/kast/reduction/ with the name equal to the run code. Within the run folder there will be a folder named "rawdata" that will have the FITS files for that night as well as an text log file (named after the run code with a ".log" extension, e.g. "te.log").

2 Preparing for Extraction

2.1 Running `reduce` to Rename the Raw Files and Create Run-specific File Lists and IRAF Scripts

In the run folder (e.g. /Users/Kelsey/observing/kast/reduction/te), type `reduce`. The program will prompt you to enter the run id (in lowercase!) and the name of the logfile (e.g. "te.log"). This step will create several new files in the run folder which previously only contained the rawdata directory (note that this is also briefly explained in the program output):

First, three IRAF scripts will be created (startred.cl, arcs.cl, and dcorr.cl). These will be explained when we get to them later in the reduction process.

Next, in addition to copying the log file, it copies each raw data file to the run folder with a new naming convention. Blue side files that were named with a prefix of "b" followed by the observation number and the suffix ".fits" (e.g. "b1.fits" or "b44.fits") are now named with the run code as the prefix followed by "blue" and then the observation number padded with zeroes as necessary to be three digits and the suffix ".fits" (e.g. "teblue001.fits" or "teblue044.fits". Red side files are treated in the same manner with the obvious difference of the "r" prefix being replaced with "red" (e.g. "tered001.fits" or "tered044.fits").

Finally, several text files will be created:

- **all1** - a list of all blue side objects, arcs, and flats (in the new form, e.g. "teblue023.fits")
- **all2** - a list of all red side objects, arcs, and flats (in the new form, e.g. "tered023.fits")
- **f1list** - a list of all blue side flats
- **f?list** - a list of all red side flats for a single object (where ? is a number > 1 and can be one or more digits, e.g. "f3list" or "f12list")
- **o1list** - a list of all blue side objects as well as the blue side 0.5" R1 arc
- **o2list** - a list of all red side files for the first object and the corresponding 2.0" R2 arc as well as the red side 0.5" R1 and R2 arcs
- **o?list** - a list of all red side files for a single object and the corresponding 2.0" R2 arc (where ? is a number > 2 and can be one or more digits, e.g. "o3list" or "o12list")
- **obs1** - a list of filenames that will eventually be created and correspond to dispersion corrected and flattened files for all blue side objects (these files have an extra "df" prefix, e.g. "dfteblue034.ms.fits")
- **obs2** - a list of filenames that will eventually be created and correspond to dispersion corrected and flattened files for all red side objects (these files have an extra "df" prefix, e.g. "dftered034.ms.fits")

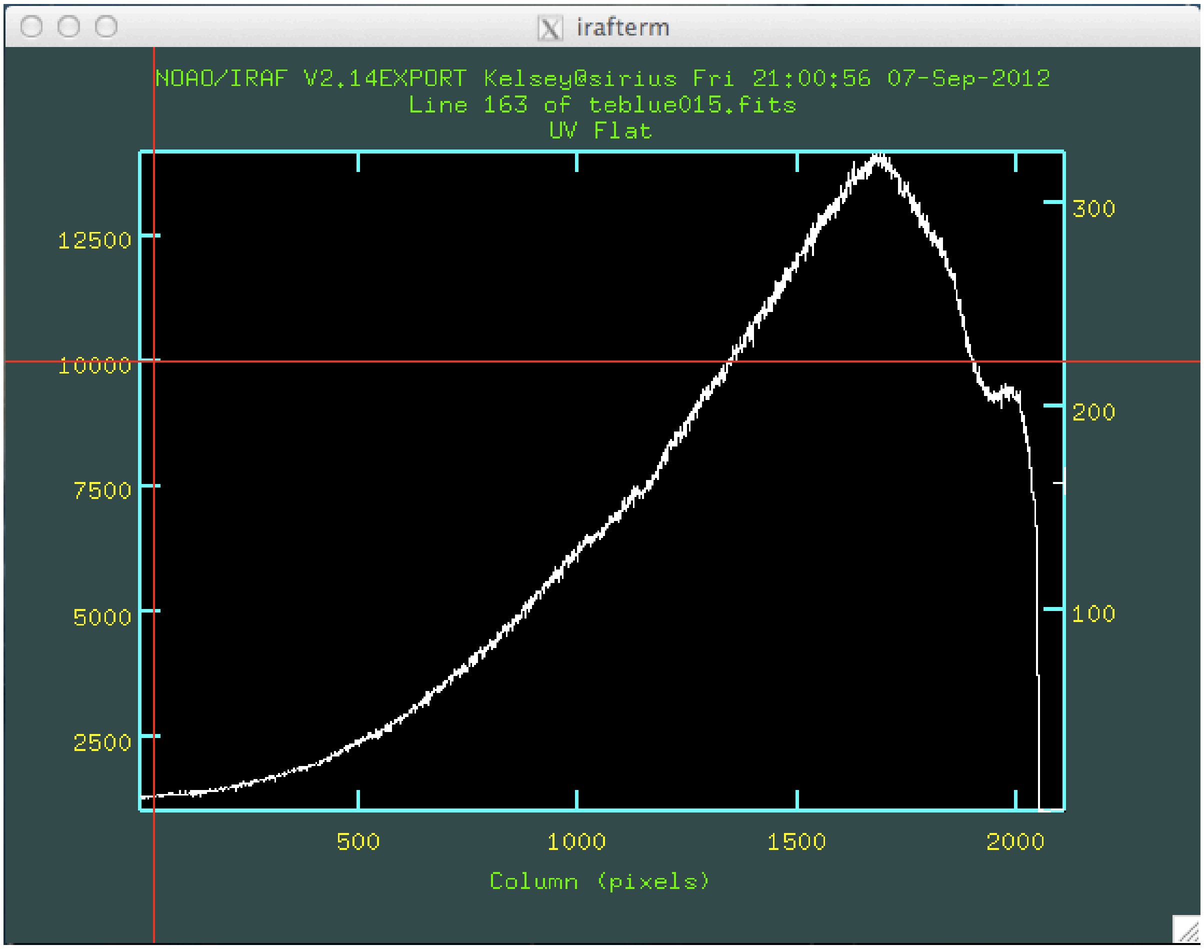

irafterm
NOAO/IRAF V2.14EXPORT Kelsey@sirius Fri 21:00:56 07-Sep-2012
Line 163 of teblue015.fits
UV Flat
12500
10000
7500
5000
2500
300
200
100
500
1000
1500
2000
Column (pixels)

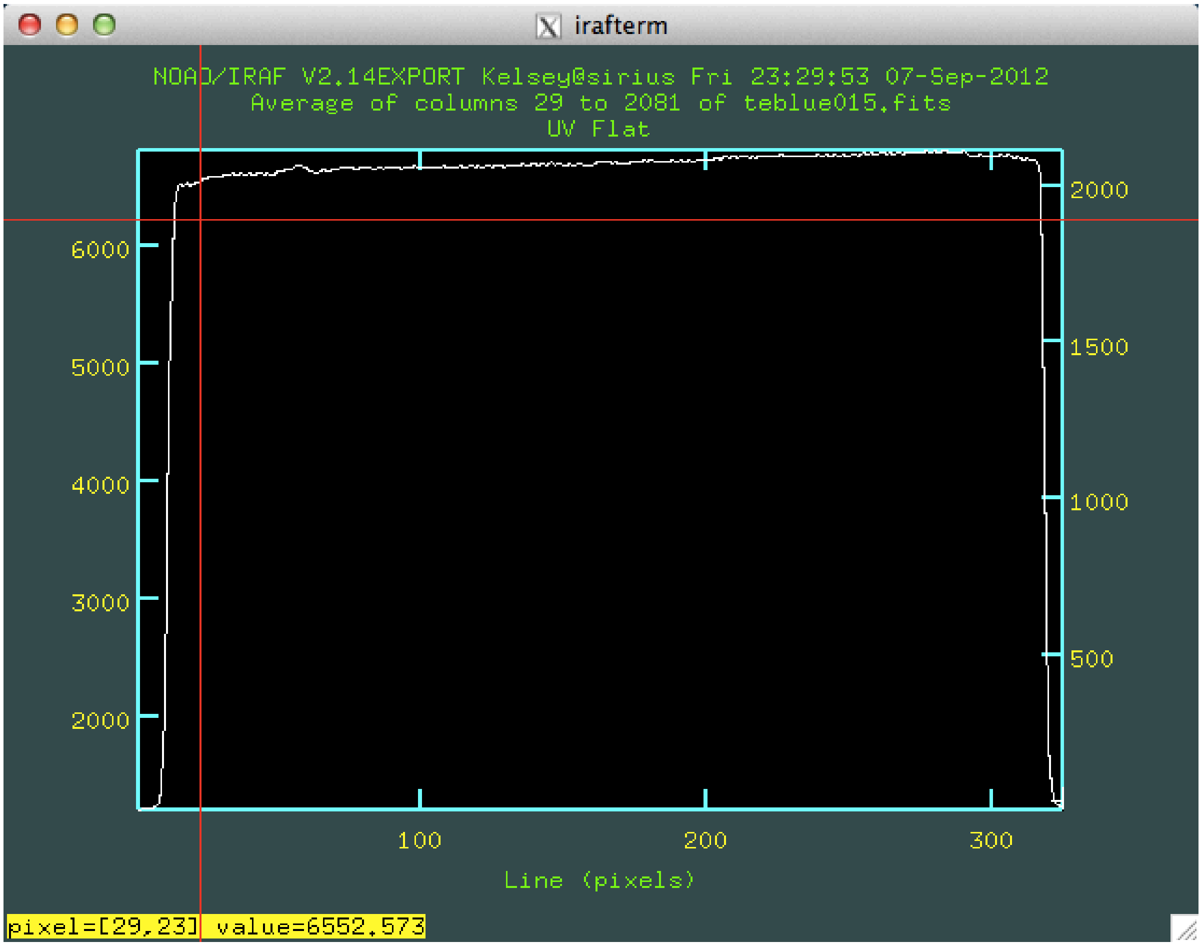
Figure 2.1: *An IRAF plot showing a row/line cut for one of the blue flats.*

2.2 Running `implot` to Determine the Trim Sections for the Blue and Red Flats

Open up an xgterm and start up IRAF (i.e., cd to your iraf folder and type `cl`). Change directories to the location of the run folder (e.g. `cd /Users/Kelsey/observing/kast/reduction/te/`). At the IRAF prompt, type `implot`. The program will prompt you for the name of the image to be plotted. We want to plot one of the blue flats, so enter the file name that corresponds to a blue flat (e.g. `teblue015.fits`). A window should pop up displaying a row cut of the blue flat (as seen in Figure 2.1 above).

Column values (in pixels) are given on the bottom axis. Row (called "line" in IRAF) values (in pixels) are given on the right axis (with the currently displayed row listed in the information above the plot and visually designated by a small, white dash on the right axis). (Note: If you want to plot a different row/line, hover over the plot so that the horizontal line that follows the cursor lines up with the row/line you want to select, then press `l` (for row/line cut).) The left axis is the number of counts. (Note: In Figure 2.1 above, the vertical red line is around a column value of 25 pixels and the horizontal red line is around row/line 213 pixels and at the level of 10,000 counts.)

We actually want to look at a column cut of the blue flat, so move the cursor so that the vertical red line lines up with nearly the peak of the flat response, then press `c` (for column cut). Notice now that the bottom and right axes have been interchanged, but the left is still the count level. Next, we want to


irafterm
NOAO/IRAF V2.14EXPORT Kelsey@sirius Fri 23:29:53 07-Sep-2012
Average of columns 29 to 2081 of teblue015.fits
UV Flat
6000
5000
4000
3000
2000
2000
1500
1000
500
100
200
300
Line (pixels)

```
pixel=[29,23] value=6552.573
```

Figure 2.2: *An IRAF plot showing a column cut (averaged over many columns) for one of the blue flats.*

display the same column plot, but averaged over many rows. We can do this by moving the cursor to the lower left corner of the plot and pressing `a` (which should result in a "again:" prompt in the lower left corner of the plot window) and then moving the cursor to the upper right corner of the plot and pressing `a` again. The top of the plotted blue flat should now be much smoother (as seen in Figure 2.2 above).

Next, we want to determine the minimum and maximum row/line values that span the useable portion of the blue flat. Do this by moving the cursor so that the red vertical line lines up with the row/line you want to designate as the minimum good value, then press `[spacebar]`. At the lower left corner of the plot window, you should see feedback in the form of "pixel=[#,#] value=#.#". The second number in the pixel array is the minimum row/line value for the blue flat (e.g. 23). Write this number down somewhere for future reference.

Now move the cursor so that the red vertical line lines up with the row/line you want to designate as the maximum good value, then press `[spacebar]`. The feedback should be updated to reflect this new cursor position and the second number in the pixel array is now the maximum row/line value for the blue flat (e.g. 310). Again, write this number down for reference. (Note: In Figure 2.2 above, the vertical red line is showing a possible minimum good value at a row/line value of 23 pixels.)

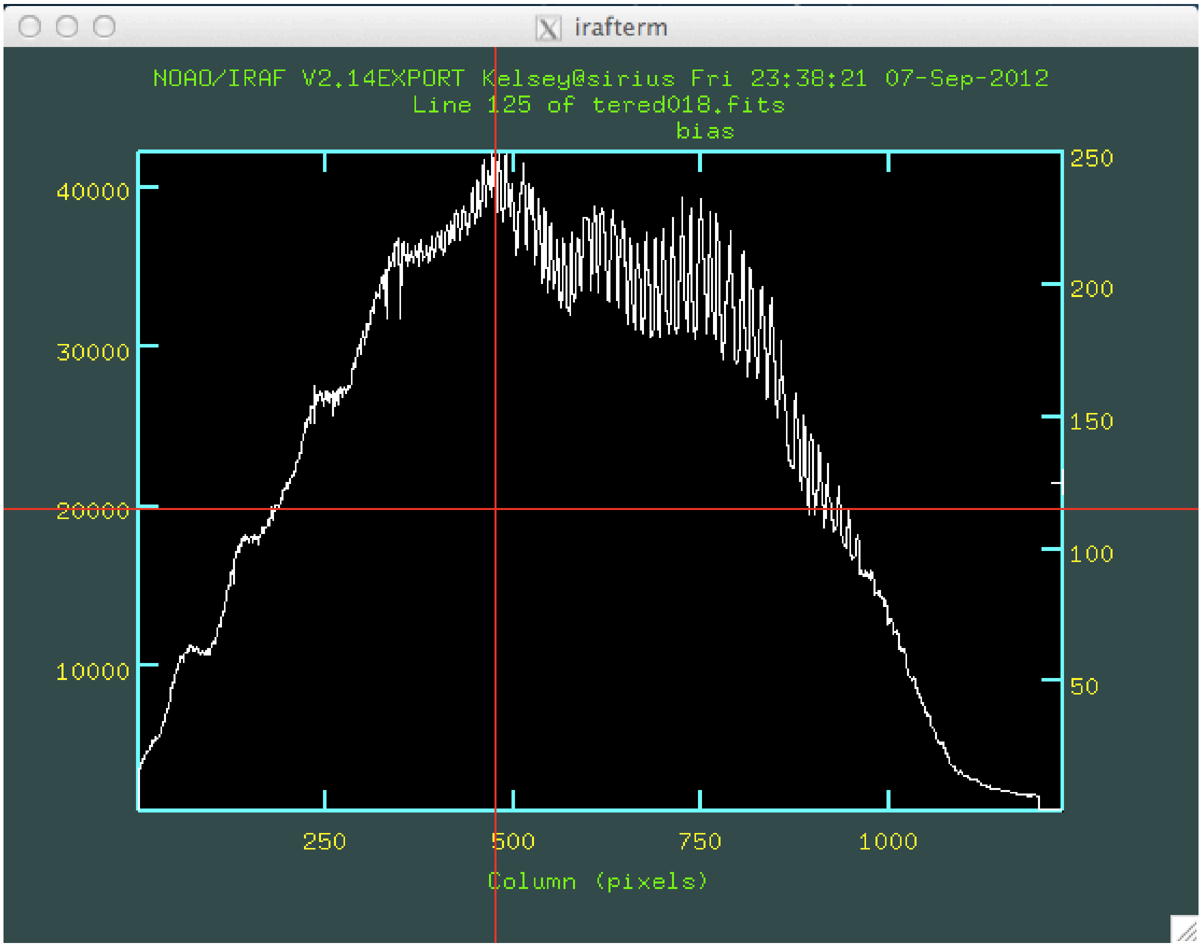


Figure 2.3: *An IRAF plot showing a row/line cut for one of the red flats. Notice the significant fringing of the red ccd!*

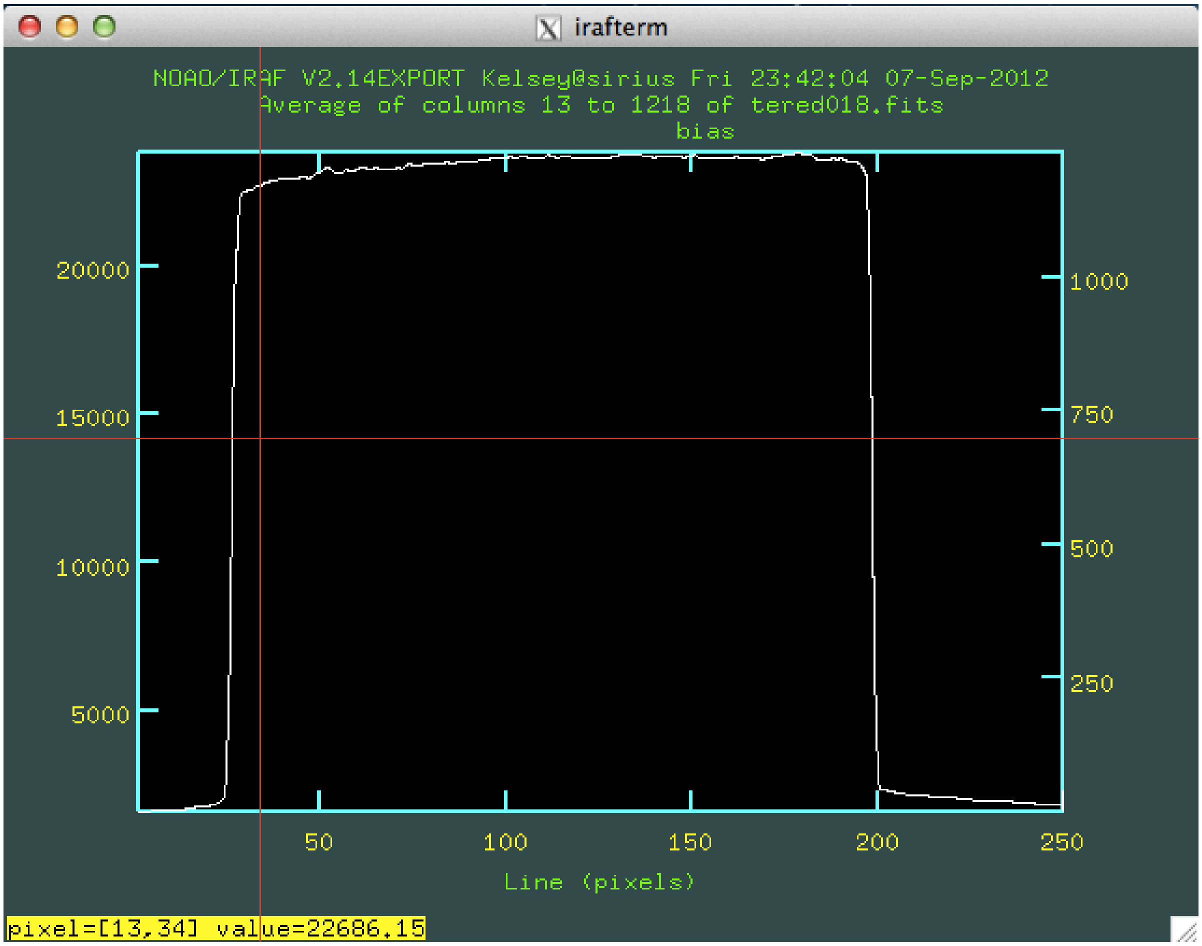
Now we need to quit out of the interaction with the blue flat by pressing `q` in the plot window and repeat the above steps, but this time for a red flat. When you plot the red flat, you'll immediately notice the fringing of the red ccd (as seen in Figure 2.3 above). Again, make sure to write the minimum and maximum good values down for future reference (note that the minimum good value is being designated as 34 in Figure 2.4 on page 8).

2.3 Running `multikastbias.pro` to Trim and Bias-correct the Raw Files

Now that we have the useable portion of the blue and red flats determined, we are ready to run an IDL program that will trim each blue and red flat to include only the useable portions of each as well as bias-correct the trimmed flats using the overscan region. The IDL program is called multikastbias.pro and should be run twice within the run folder as follows (where minblue, maxblue, minred, and maxred are the numbers you found with implot):

IDL> multikastbias,'all1',y1=minblue,y2=maxblue (e.g., `multikastbias,'all1',y1=23,y2=310`)
IDL> multikastbias,'all2',y1=minred,y2=maxred (e.g., `multikastbias,'all2',y1=34,y2=190`)

This step will create a new bias-corrected file for all blue objects, arcs, and flats as well as all red objects, arcs, and flats. These bias-corrected files will be located in the run folder and have a "b" prefix.



```
NOAO/IRAF V2.14EXPORT Kelsey@sirius Fri 23:42:04 07-Sep-2012
Average of columns 13 to 1218 of tered018.fits
bias
```

```
pixel=[13,34] value=22686.15
```

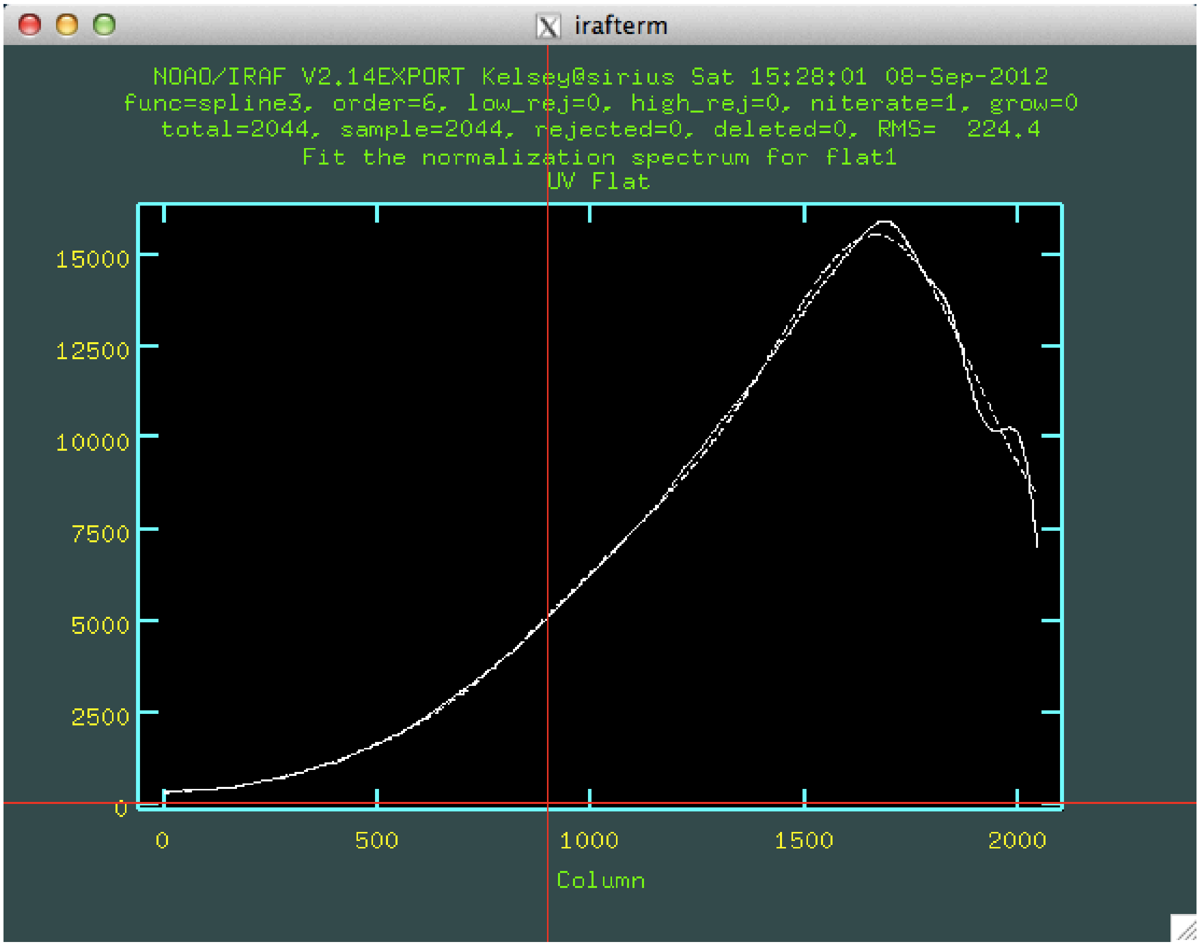
Figure 2.4: *An IRAF plot showing a column cut (averaged over many columns) for one of the red flats.*

2.4 Running `startred.cl` to Combine and Fit a Response Function to the Blue and Red Flats

We are ready to run the IRAF script startred.cl and can do this by typing `cl < startred.cl` in our IRAF window. This script will add some pertinent keywords to the FITS header of each bias-corrected file (such as identifying Lick as the observatory as well as the optimal PA). It will then calculate the airmass for the midpoint of each observation and store this in the FITS header of each bias-corrected file.

Next it will combine all of the blue flats and IRAF will prompt you with "Fit the normalization spectrum for flat1 interactively (yes):" to which you don't actually need to type anything. In the IRAF display window, a row/line cut of the combined blue flat will be displayed. The solid white line is the flat and the dashed white line is the current fit to the solid line. You will note that the fit doesn't follow the flat exactly, but rather gives a fit for the overall shape of the flat, this is OK (as seen in Figure 2.5 on page 9).

The first thing we need to check is that the flat does not have any pixels with negative values. The left axis shows the count level so there should be no pixels below 0 (note that the horizontal red line in Figure 2.5 on page 9 is at a count level of 0 and all pixel values are above the red line and, thus, positive-valued). If there are any negative-valued pixels, you'll need to zoom-in on the pixels, delete the pixels, and refit.



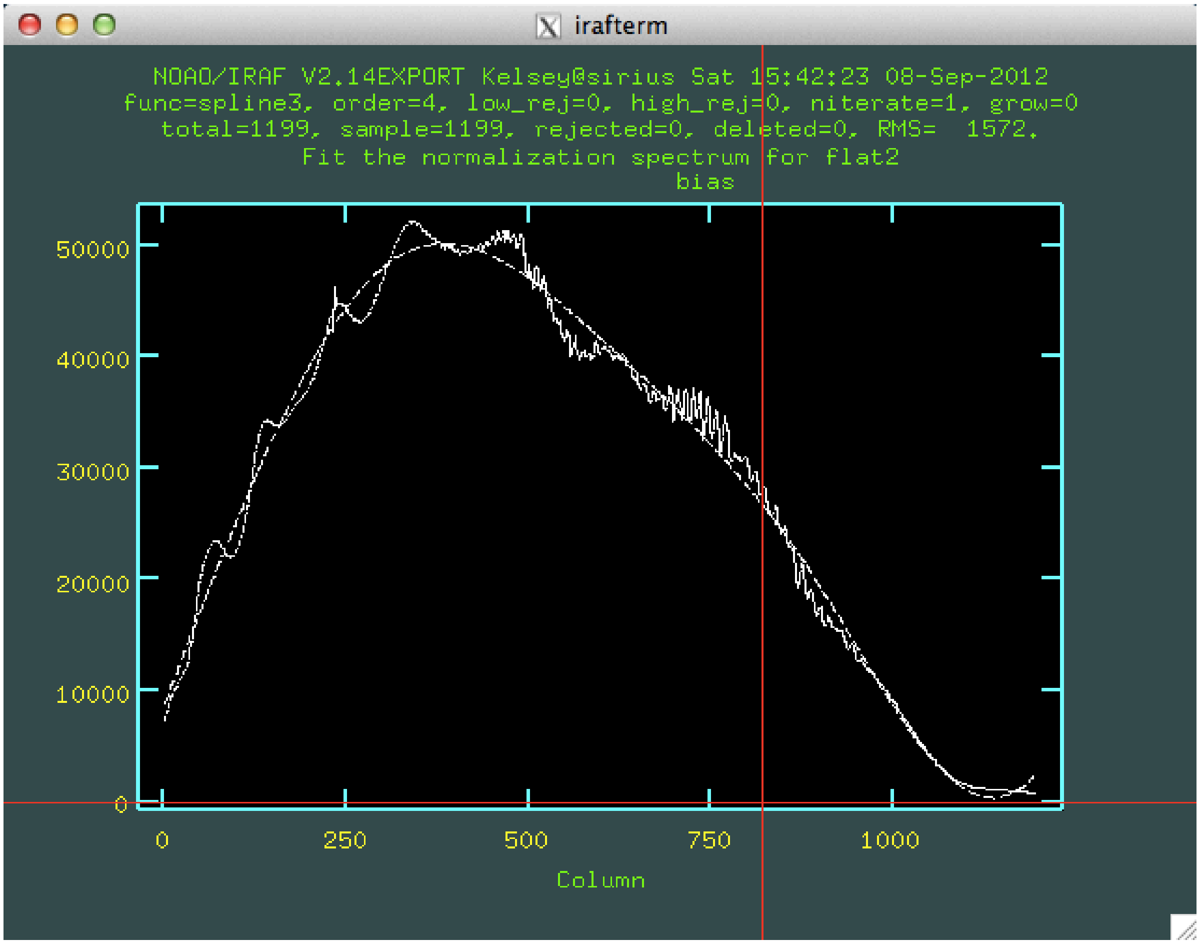
```
NOAO/IRAF V2.14EXPORT Kelsey@sirius Sat 15:28:01 08-Sep-2012
func=spline3, order=6, low_rej=0, high_rej=0, niterate=1, grow=0
total=2044, sample=2044, rejected=0, deleted=0, RMS=  224.4
Fit the normalization spectrum for flat1
UV Flat
```

Figure 2.5: *An IRAF plot showing a fit (dashed line) to the combined blue flat (solid line).*

You can zoom-in in IRAF by moving the cursor to the bottom left of the part of the plot you want to zoom-in on and press `w` then press `e` then move the cursor to the top right of the part you want to zoom-in on and press either `w` or `e` (to zoom back out to show the entire plot, press `w` then `a`). To delete a pixel, move the cursor so that it hovers over the pixel you want to delete and press `d` (if you want to delete several pixels in a row, pressing and holding `d` will delete the closest undeleted pixel to the cursor until you release `d`). If you want to undelete a pixel (or several nearby pixels), move the cursor so that it hovers over the pixel you want to delete and press `u` (or press and hold `u`).

Once the flat has only positive-valued pixels, the second thing we want to check is to look for bad pixels in the flat that could result in a poor fit. This is especially important at the left and right ends of the flat. To do this, zoom-in on small chunks of the flat and delete bad pixels as necessary. After deleting a bad pixel you can view the new fit by pressing `f` and determine if deleting the pixel improved the fit. Once you're happy with the fit, press `q` to save the fit and flatten each blue object and arc.

After a few seconds, the IRAF display window will update to display the combined red flat for the first group of red flats. As before with the red ccd, you'll notice some significant fringing in the combined red flats and, again, it is OK that the fit will only follow the overall shape of the flat, not every little wiggle (as seen in Figure 2.6 on page 10).



```
NOAO/IRAF V2.14EXPORT Kelsey@sirius Sat 15:42:23 08-Sep-2012
func=spline3, order=4, low_rej=0, high_rej=0, niterate=1, grow=0
total=1199, sample=1199, rejected=0, deleted=0, RMS=  1572.
Fit the normalization spectrum for flat2
bias
```

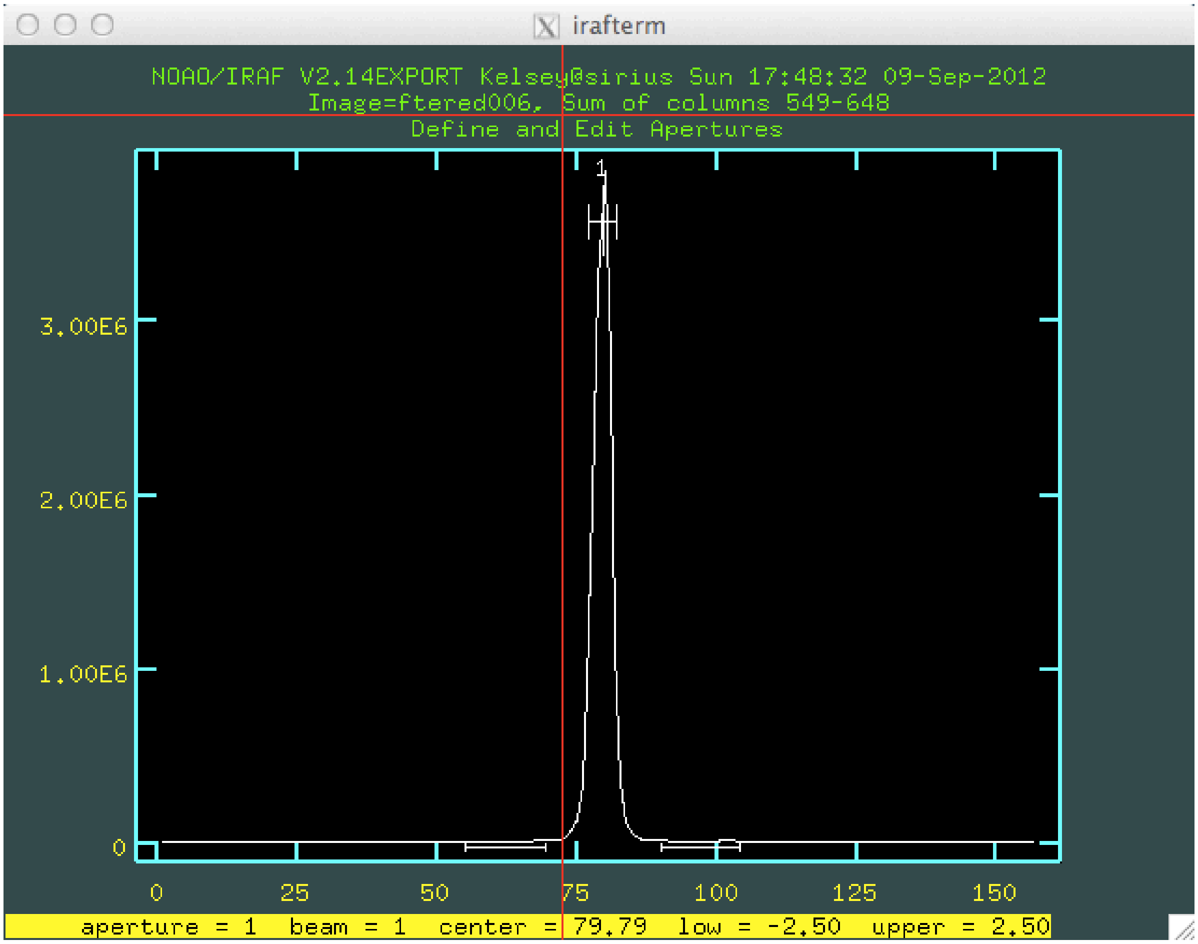
Figure 2.6: *An IRAF plot showing a fit (dashed line) to the combined red flat for the first group of red flats (solid line).*

As with the combined blue flat, you'll need to delete any negative-valued pixels and other bad pixels (especially on the left and right edges of the flat) and refit. Once you're satisfied with the fit, press `q` to save the fit and flatten the corresponding red object and arc. Since there is a group of red flats for each object, you'll need to repeat the same steps as above for each group of red flats (i.e., if you observed 15 objects, you'll review and save the fit for 15 separate combined red flats.

This step creates the following files:

- **logfile** - A text file containing the output of imcombine (the IRAF script that combines images)
- **flat1.fits** - The combined blue flat
- **flat?.fits** - The combined red flat for group ? (where ? is a number > 1)
- **flat1res.fits** - The fit to the combined blue flat
- **flat?res.fits** - The fit to the combined red flat for group ? (where ? is a number > 1)

It also creates a new flattened (and bias-corrected) file for all blue objects and arcs as well as all red objects and arcs. These flattened (and bias-corrected) files will be located in the run folder and have a "f" prefix (e.g. "ftebluе040.fits" or "ftered051.fits").



```
NOAO/IRAF V2.14EXPORT Kelsey@sirius Sun 17:48:32 09-Sep-2012
Image=ftered006, Sum of columns 549-648
Define and Edit Apertures
```

```
aperture = 1  beam = 1  center = 79.79  low = -2.50  upper = 2.50
```

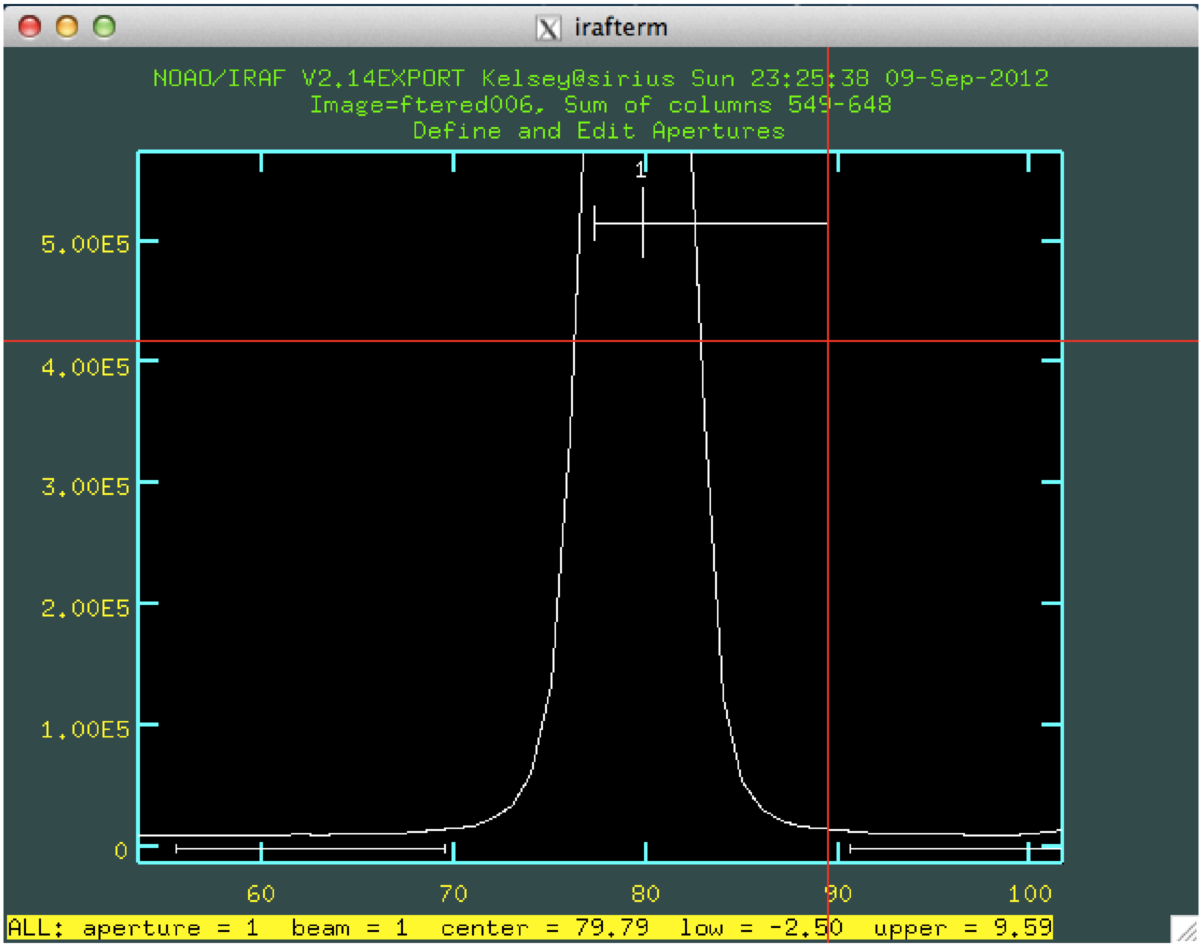
Figure 3.1: *An IRAF plot showing the initial aperture and background regions after running apall on the red side of the first object.*

3 Extracting the Spectra

3.1 Running `apall` to Extract the Red Side of Each Object

We are now ready to extract the red side of each object. First, you'll need to know the observation numbers for all of the red side objects. You can use the wiki log (or paper logbook) to look up the observation numbers or you can look at the text file "obs2" (but only note the numbers, not the actual file name). Next, in IRAF, run apall on the flattened red spectrum for the first object, e.g., `apall ftered006` (note that you don't need to include the ".fits" suffix). In response to IRAF's prompts, `yes` you want to find apertures, the number of apertures to be found automatically should be `1`, and `yes` you want to edit apertures.

The IRAF window should now be displaying the summed counts for the middle 100 columns vs. the row/line number (as seen in Figure 3.1 above). By plotting this, the background/sky counts are hopefully less than the object counts, so adding 100 larger numbers together compared to adding 100 smaller numbers together (i.e., effictively increasing the signal-to-noise ratio) should help make the counts associated with the object more easily discernable (this is especially the case for bright objects, like standards, as seen in Figure 3.1 above).



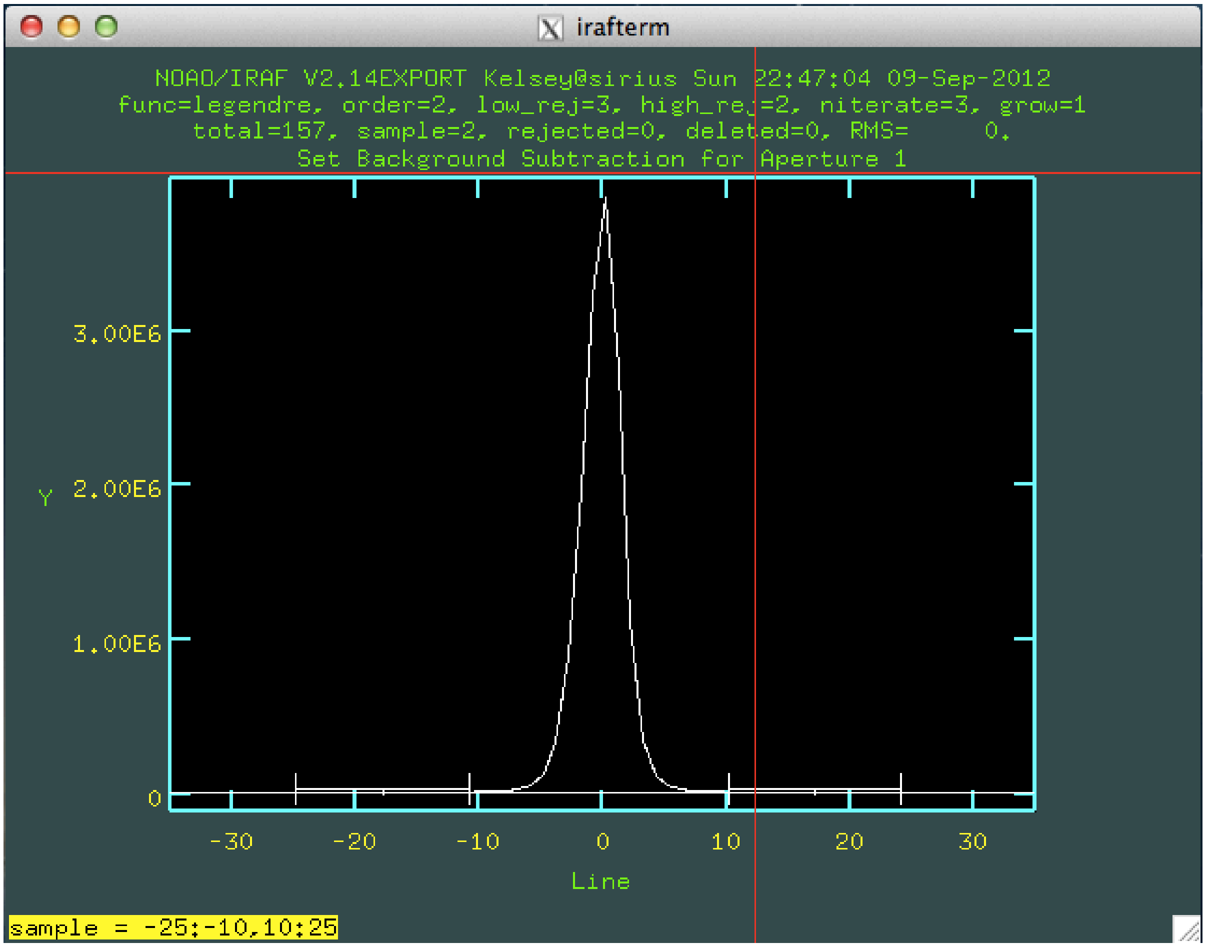
```
NOAO/IRAF V2.14EXPORT Kelsey@sirius Sun 23:25:38 09-Sep-2012
Image=ftered006, Sum of columns 549-648
Define and Edit Apertures
ALL: aperture = 1  beam = 1  center = 79.79  low = -2.50  upper = 9.59
```

Figure 3.2: *An IRAF plot showing that a right/upper edge of 9.59 pixels includes the majority of this red standard's signal without including the noise.*

3.1.1 Defining the Aperture

The first thing we need to do is to define the aperture (i.e., the full width of the object's signal above the background). By default, the initially chosen aperture is always ±2.5 pixels from the center of what IRAF thinks is the center of the object's point-spread function. Thus, if there is a spurious source that has fooled IRAF into thinking it's the object's signal, you'll need to shift the aperture to the object. To do this, move the cursor so that it lines up vertically with the location you want to shift the aperture to and the press `s`. You will be asked if you want IRAF to center the aperture automatically, to which you can type `yes` or `no` or press `[enter]` to choose the default (shown in parentheses). If the signal is significantly above the background, you'll probably want to let IRAF center the aperture automatically.

Once you're satisfied with the general location of the aperture, you'll want to zoom-in on the signal to determine the ideal width for the aperture. Next, you'll want to move the cursor so that the red vertical line lines up with where you'd like to set the right/upper edge of the new aperture then press `u`. The feedback in the yellow stripe at the bottom of the IRAF window tells you what the new upper value now is (e.g., 9.59 in Figure 3.2 above). You can also change the upper value to any value # by typing ":up #". If you think too few or too many pixels are being included in the aperture, use the same "u" and/or "up" commands to redefine the right/upper edge of the aperture until you are satisfied.



```
NOAO/IRAF V2.14EXPORT Kelsey@sirius Sun 22:47:04 09-Sep-2012
func=legendre, order=2, low_rej=3, high_rej=2, niterate=3, grow=1
total=157, sample=2, rejected=0, deleted=0, RMS=     0.
Set Background Subtraction for Aperture 1
```

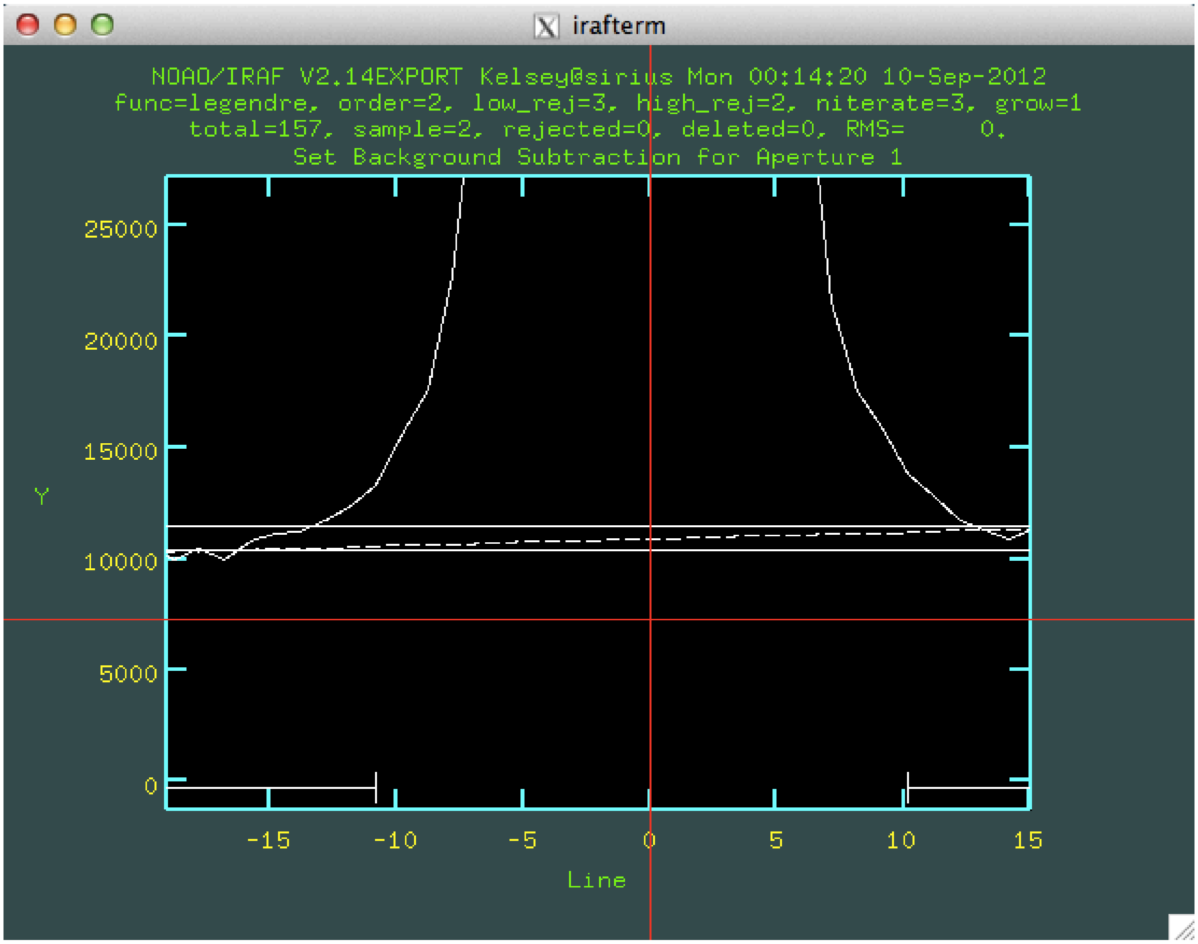
```
sample = -25:-10,10:25
```

Figure 3.3: *An IRAF plot showing the default background regions.*

Once you're satisfied with upper/right edge of the new aperture, you can type ":low -#", where # is the value of the upper/right edge of the new aperture (e.g. `:low -9.59`). (Note that, except for special circumstances, you'll want to use the same number you used for the right/upper edge in order to have a symmetric aperture.) You'll want to check the left/lower edge by moving the cursor so that the red vertical line lines up with the new left edge of the aperture. It's OK if it doesn't perfectly include all of the signal (especially for very high signal-to-noise spectra like standards or other bright science objects), but it should be very close. If a symmetrical aperture really doesn't cut it, you can change the left/lower edge with the "low" command and/or by hovering the cursor so that the red vertical line lines up with where you'd like the set the left/lower edge of the new aperture and then press `l`. When you've found a good aperture, make note of the lower and upper values for future reference.

3.1.2 Defining the Background

Now we are ready to define the background level that will be subtracted from the signal (since the signal also contains some background/sky counts). To do this, we first need to enter "background mode" by pressing `b`. You should see two background sections to the left and right of the aperture and the resulting fit as a dashed white line connecting the two background regions (as seen in Figure 3.3 above). To view the current background regions type `:sample` (note that IRAF defaults to a left background of -25:-10 and a right background of 10:25).

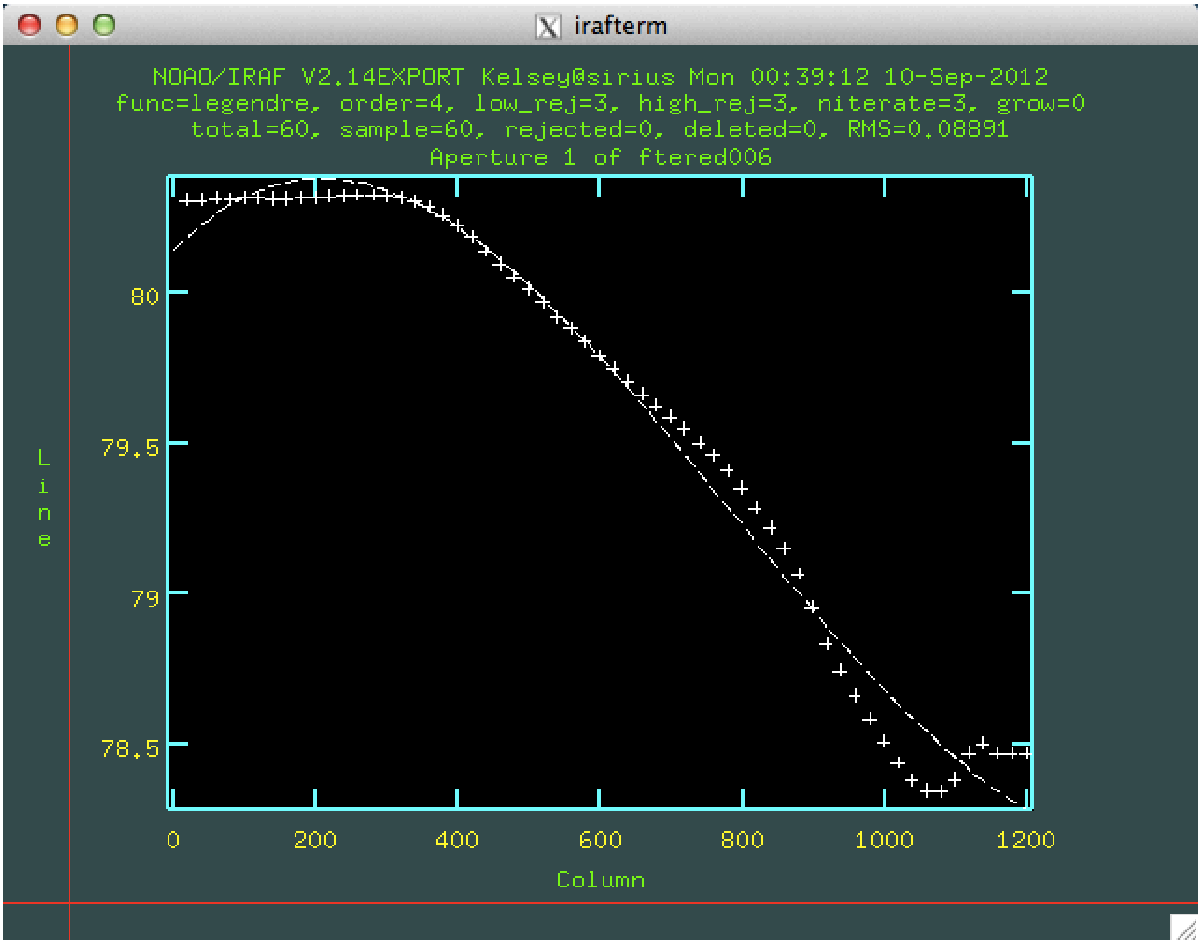


```
NOAO/IRAF V2.14EXPORT Kelsey@sirius Mon 00:14:20 10-Sep-2012
func=legendre, order=2, low_rej=3, high_rej=2, niterate=3, grow=1
total=157, sample=2, rejected=0, deleted=0, RMS=      0.
Set Background Subtraction for Aperture 1
```

Figure 3.4: *An IRAF plot showing the fit to the two background regions.*

If you're not satisfied with the default (and you probably shouldn't be most of the time), here are two ways to change the default background regions. The first way is to hover the cursor so that the red vertical line is within the current left background region and press `z` (for zap) to delete the current left background region and then do the same for the right background region. Then, you can hover the cursor so that the red vertical line is at the left edge of the new left background region you want to use and press `s` to set the left edge then hover the cursor so that the red vertical line is at the right edge of the new left background region you want to use and press `s` to set the right edge and then do the same for the new right background region. The second way is to type ":sample #:#,#:#", where the #s correspond to the left edge of the left region, the right edge of the left region, the left edge of the right region, and the right edge of the right region (e.g., `:sample -20:-15,15:-20`).

Each time you change a background region you'll need to press `f` to redo the fit. The fit is determined by finding the average count value within the left region, finding the average count value within the right region, "placing" those average values at the midpoint of the respective region and then fitting a straight line between those two points/values. As seen in Figure 3.4 above, the lower horizontal solid white line corresponds to the average count value of the left background region, the upper horizontal solid white line corresponds to the average count value of the right background region, and the dashed white line is the fitted background level (note that it has a slightly positive slope and that's perfectly OK).



```
NOAO/IRAF V2.14EXPORT Kelsey@sirius Mon 00:39:12 10-Sep-2012
func=legendre, order=4, low_rej=3, high_rej=3, niterate=3, grow=0
total=60, sample=60, rejected=0, deleted=0, RMS=0.08891
Aperture 1 of ftered006
```

Figure 3.5: *An IRAF plot showing the fit to the object's trace along the red CCD.*

In general, background regions should be between 5 and 10 pixels wide and it is imperative that the background regions do not intersect with the aperture. To check this, and to write the background region values down for future reference, type `:sample`. When you're satisfied with background, press `q` to quit out of "background mode" and return to "aperture mode". If you need to modify the aperture, do so now, otherwise press `q` to save the aperture and background regions. In response to the IRAF prompt at the bottom of the plot window, type `yes` to trace the aperture, `yes` to fit the trace interactively, and `yes` to fit the curve interactively. A new plot should appear and now we're ready to fit the object's trace.

3.1.3 Fitting the Object's Trace along the CCD

The IRAF plot window should now show a row/line vs. column plot of what it believes to be the object's trace along the CCD. You'll notice it is not a perfectly horizontal or sloped line, but it does only span a relatively few number of rows/lines (about two pixels worth for the plot show in Figure 3.5 above). You'll want to use a 4th order Legendre fit (unless the points are nearly monotonic, then you can try a 3rd order Legendre fit). If necessary, you can change the function to Legendre by typing `:func legendre` and/or change the order by typing ":ord #" (e.g., `:ord 4`). Since the vertical axis is only changing by at most a few pixels, the initial fit (as seen in Figure 3.5 above) doesn't look to follow the trace that well, however, it only differs by half of a pixel at most and our aperture is several pixels wide, so this is OK.

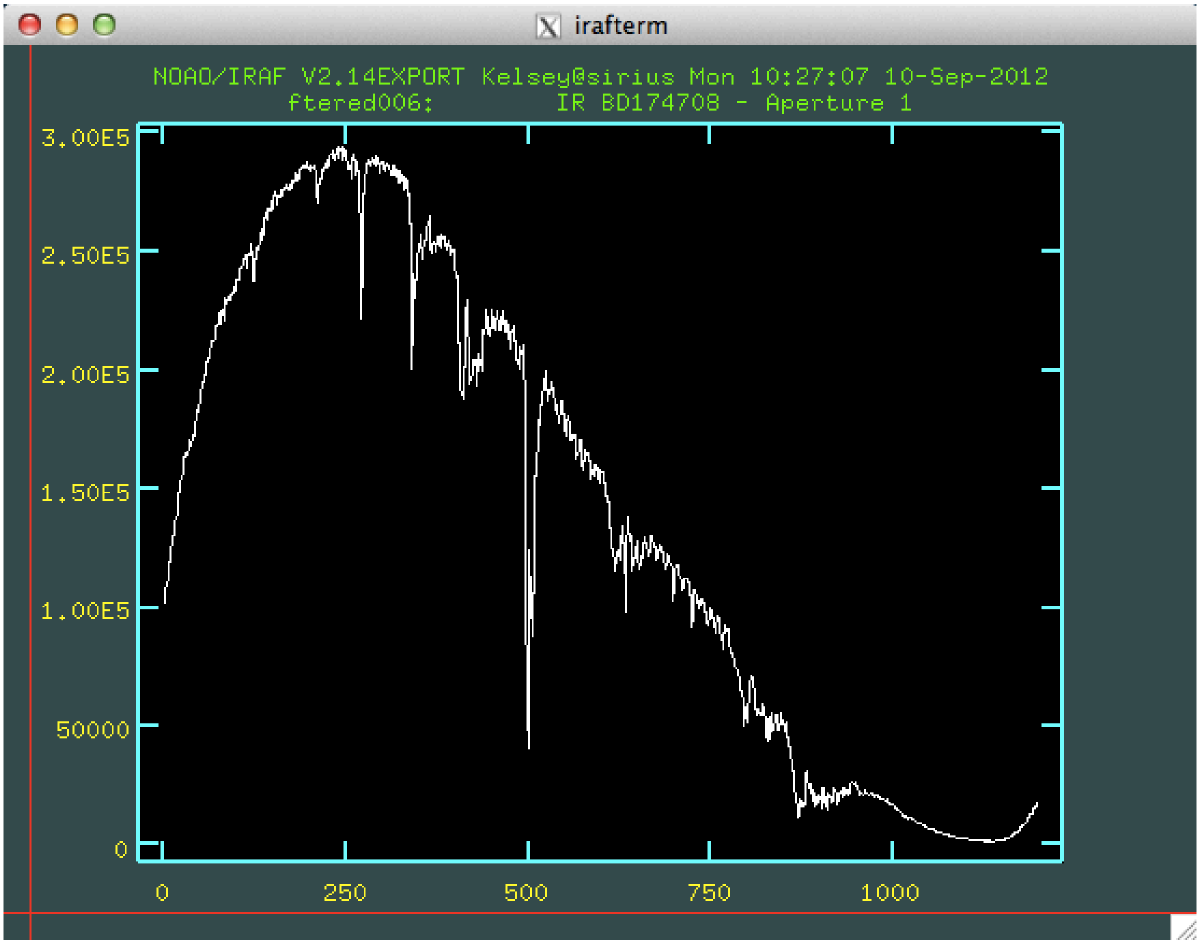

irafterm
NOAO/IRAF V2.14EXPORT Kelsey@sirius Mon 10:27:07 10-Sep-2012
ftered006: IR BD174708 - Aperture 1
3.00E5
2.50E5
2.00E5
1.50E5
1.00E5
50000
0
0
250
500
750
1000

Figure 3.6: *An IRAF plot showing an extracted spectrum for a red standard.*

If you want to see the values for the trace and the fit for a single point, you can hover the cursor to line up vertically with the point you want to check and type `c`. The feedback in the yellow stripe at the bottom of the IRAF plot window will give you the x/column value, y/row/line value, and fit value for the nearest point to the cursor. If it looks like some points are throwing off the fit, you can hover the cursor over the point you want to remove from the fit and press `d` to delete the point from the fit. You'll need to press `f` to refit the remaining points and, if you're not happy with the result, you can hover over the deleted point and type `u` to undelete the point and type `f` to return to the previous fit.

Once you're satisfied with the fit to the trace along the CCD, type `q` to quit and save the fit to the trace. In response to the IRAF prompt at the bottom of the plot window, type `yes` to write the aperture to the database, `yes` to extract the spectrum, `yes` to review the extraction, and `yes` to review the extraction for aperture 1 (if you're re-running apall on an observation, you'll also need to type `yes` to clobber the existing output image). IRAF will now use the fit to the trace for each column pixel as the center of the aperture for that column pixel and sum the counts along the rows/lines that are within the defined aperture size. The final result will be a single count value for a single column pixel and a plot of the counts vs. column pixel for the extracted spectrum should now appear (as seen in Figure 3.6 above). If all looks good, you can type `q` to quit and save the extraction.

This step updates the logfile with some minor information from apall, creates a database folder that stores a new text file with detailed results from apall for each extracted spectrum named with a prefix of "ap" added to the input filename (e.g. "apftered006"), copies the most recently added database text file to a file named "aplast", and, finally, saves the one-dimensional extracted spectrum to a new file with a ".ms." inserted after the input filename and before the ".fits" suffix (e.g. "ftered006.ms.fits"). (Note: If you want to view a .ms.fits file, DS9 is not very useful. Instead, you can use implot in IRAF or readfits in IDL, but the file is a [#ofcolumns] by 1 by 4 array, so you'll want to plot it with "plot,im[*,*,1]" (where im=readfits('filename') if using IDL.)

Now you should repeat the above steps for the red side of each object. If you have an object with two or more consecutive observations, run apall on the first observation as outlined above. Then, for each additional observation, you can reference the first observation in the call to apall (e.g., type `apall ftered027 ref=ftered026`).

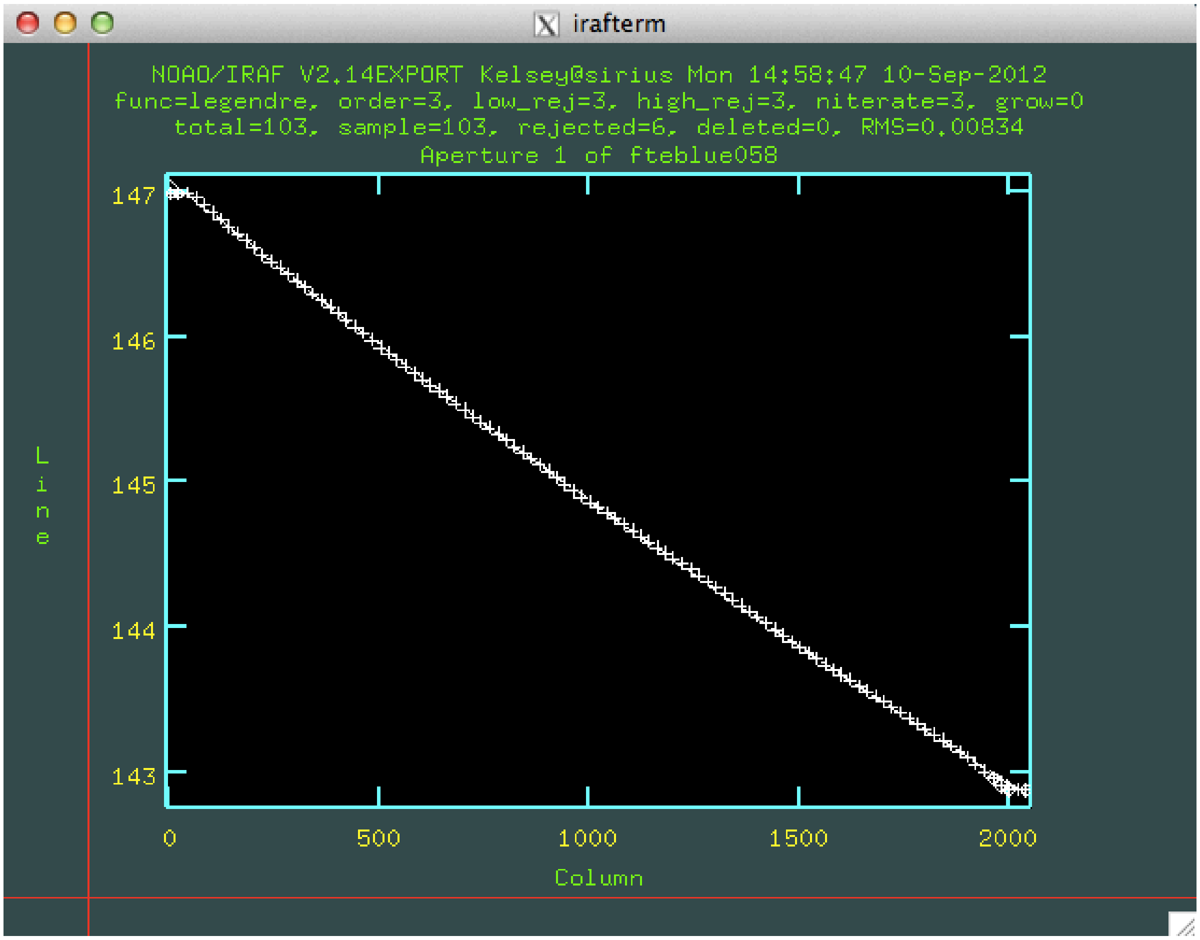
3.2 Running `apall` to Extract the Blue Side of Each Object

We are now ready to extract the blue side of each object (except we don't need to extract the blue side of the red standards since we won't ever be using them). Again, you'll first need to know the observation numbers that correspond to the blue side of each object. Start by running apall just as we did for the red side on the low and high airmass blue standards.

Now, for every blue side object except the blue standards, we want to run apall as usual, but instead of defining an aperture and background regions as we did with the red side, we will want to use the same aperture and background regions as the red side of the corresponding object. The important caveat here is that the plate scale in the spatial/vertical direction is different for both CCDs. The blue CCD has a spatial plate scale of 0.798"/pixel, while the red CCD has a spatial plate scale of 0.43"/pixel. Thus, the blue CCD has a plate scale that is 1.8558 times larger than the red CCD. The result of this is that the red side aperture and background region values will need to be multiplied by 1.8558 in order to encompass the desired aperture and background regions on the blue side.

So, referring back to the red side aperture and background region values you wrote down for future reference, run apall on the blue side observation of one of the objects (by typing `apall fteblue040`, e.g.). Then type `:low -#`, where -# is the lower/left edge of the red side aperture for the corresponding object multiplied by 1.8558, then type `:up #`, where # is the upper/right edge of the red side aperture for the corresponding object multiplied by 1.8558 (note that you can round to the nearest .10, if desired). If IRAF centers the aperture on a spurious source you will need to shift the aperture to the actual object as previously described.

Similarly for the background regions, after pressing `b` to enter "background mode", type `:sample -#:-#,#:#`, where the numbers are the background region values for the red side multiplied by 1.8558. Don't forget to press `f` to refit the new background regions. If either the aperture and/or the background regions look significantly erroneous, go ahead and edit them as needed (but this usually shouldn't be the case, so double check your math). When your blue side aperture and background regions are scaled to the red side aperture and background regions, you can press `q` as needed to save them and move on to fitting the trace. Fitting the trace is the same as it was on the red side, but you'll likely notice a more monotonic trace that spans several pixels (instead of just a few) on the blue side (as seen in Figure 3.7 on page 18).



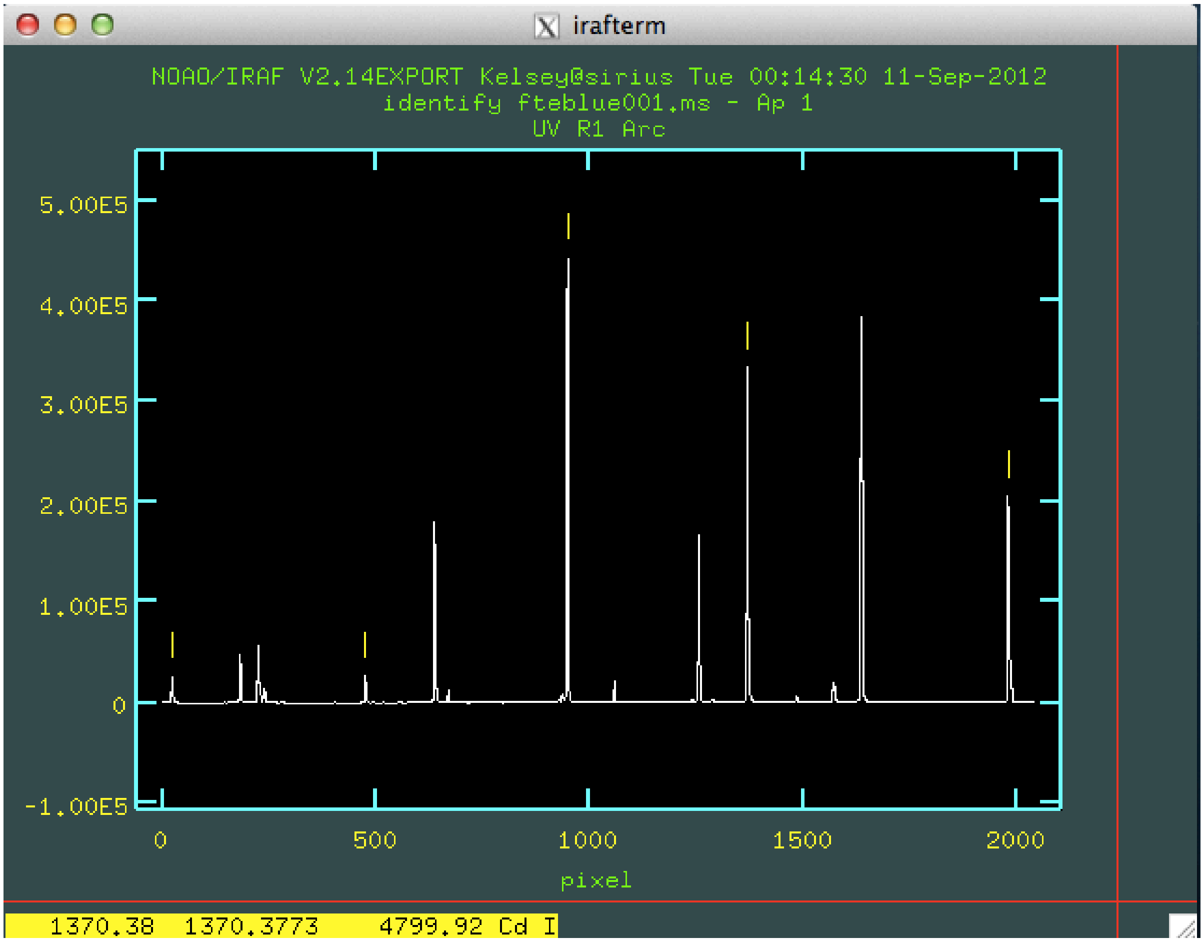
NOAO/IRAF V2.14EXPORT Kelsey@sirius Mon 14:58:47 10-Sep-2012
func=legendre, order=3, low_rej=3, high_rej=3, niterate=3, grow=0
total=103, sample=103, rejected=6, deleted=0, RMS=0.00834
Aperture 1 of fteblue058

Figure 3.7: *An IRAF plot showing the fit to the object's trace along the blue CCD.*

Now you should repeat the above steps for the blue side of each object (except the blue standards). As with the red side, if you have two or more consecutive observations of the same object, you can run apall on the first observation as just described (i.e. using the scaled red side values). Then, for each additional observation, you can reference the first observation in the call to apall (e.g. type `apall fteblue043 ref=fteblue042`). As it did for the red side objects, this step will again update the logfile, store a summary of the apall results in the database folder, and save the newly extracted, one-dimensional object spectrum with a ".ms" after the input filename and before the ".fits" suffix.

3.3 Running `arcs.cl` to Extract All Blue and Red Arcs

Now we are ready to run the night-specific IRAF script arcs.cl and can do this by typing `cl < arcs.cl` in our IRAF window. This script will automatically (as opposed to interactively) extract all blue and red arcs. For the blue side 0.5" arc, we will reference the blue side observation of the first blue standard. For the red side 0.5" arcs, we will reference the red side observation of the first red standard. For each red side 2.0" arc, we will reference the associated red side observation. We also specifically do not allow IRAF to recenter the aperture, fit a trace, or do a background subtraction. The logfile and database folder will be updated and the newly extracted, one-dimensional arc spectrum with a ".ms" after the input filename and before the ".fits" suffix.


irafterm
NOAO/IRAF V2.14EXPORT Kelsey@sirius Tue 00:14:30 11-Sep-2012
identify fteblue001.ms - Ap 1
UV R1 Arc
5.00E5
4.00E5
3.00E5
2.00E5
1.00E5
0
-1.00E5
0
500
1000
1500
2000
pixel

```
1370.38  1370.3773     4799.92 Cd I
```

Figure 4.1: *An IRAF plot showing five manually marked lines for the 0.5" blue side arc.*

4 Solving for a Wavelength Solution

The next task is to determine the wavelength associated with each column pixel so that we can eventually make a flux vs. wavelength plot of each spectra. To do this we will use IRAF to identify the lines in each arc spectrum taken throughout the night. Let's start with the blue side.

4.1 Running `id` to Identify the Lines in the Blue Side 0.5" Arc

Each night, we take a 15-second blue side exposure through the 0.5" slit with the R1 set of lamps (Ar, He, Hg-Cd). This is our only blue side arc for the night. To begin the line identification process, run id on the blue side 0.5" arc, i.e., type `id fteblue001.ms`, if the 0.5" R1 arc was the first blue side exposure of the night (otherwise change "001" to the appropriate blue side observation number). A plot of the 0.5" blue side arc should now appear in the IRAF plot window. Now you're ready to identify a few of the lines manually. We'll want to choose about five lines that span the entirety of the blue ccd (as shown in Figure 4.1 above) to help IRAF find the most robust fit to these manually specified lines.

There are five suggested lines to manually mark on the blue side. The first obvious line is the 5460 Hg I line that we use to determine the desired wavelength coverage at the beginning of the night. It should be the strong, right most line on the blue side arc (currently near column pixel 1980). To identify this line manually, hover the cursor so that it lines up vertically with the 5460 line, press `m` to mark the line (a vertical yellow line will appear above the line in question), type `5460` in response the the IRAF feedback in the yellow stripe at the bottom of the plot window, and check that IRAF found the correct line in it's line list (e.g. 5460.7348 Hg I).

The next line usually manually identified is the left-most (and somewhat weak) line. This is the 3466 line (currently near column pixel 25). Mark the line as before and verify that IRAF found it to be 3466.55 Cd I. The next line to mark is the strong line in the middle, 4358 (which is currently near column pixel 950). IRAF should find this line to be 4358.3277 Hg I. Next, you can mark the 3888 line (currently near column pixel 475). IRAF should find it to be 3888.649 He I. And, finally, you can mark the 4799 line (currently near column pixel 1370). IRAF should find it to be 4799.92 Cd I.

Now that you've manually marked a set of lines for the blue side 0.5" arc, you'll want to press `f` to have IRAF find a fit to these lines and apply the fit across the entire blue CCD. The IRAF plot window will update to show the residuals of the fit it found. If the RMS is larger than about 0.1, you should try a higher order fit (e.g., type `:ord 4` to change to a 4th order fit, which is usually the highest used for the blue side arc). After changing the order, you'll need to refit (i.e., press `f`). Increase the order number as needed up to a maximum of 4. If the RMS is larger than about 0.1 with a 4th-order Legendre function, you should press `q` to return to the spectrum and double-check that you haven't made an error (if you have, you can hover the cursor over the incorrectly marked line and press `d` to delete the marking and re-mark). If all marked lines check out, you should consult the Kast blue side reference spectrum and mark (an) additional line(s) until you find a satisfactory fit.

Once you're satisfied with the RMS of the residuals, press `q` to return to the spectrum then press `l` to identify all of the lines. Nearly all of the lines should now be identified with a yellow line. Next, you'll want to look at every single one of the identified lines and make sure they look like real lines and not noise, aren't significantly blended (the 3610.51 Cd I line, which actually says "(blended)" will definitely need to be deleted), and aren't saturated. To do this, hover the cursor so that the red vertical line is to the left of the 3466 line (the first line) and press `z` to zoom-in on the identified line nearest the cursor. If the identified line looks bad, press `d` to delete it, otherwise, if the identified line looks good, press `n` to skip to the next identified line. Once you've reached the last identified line, press `p` to pan back out to view the entire spectrum and all identified lines.

Now we're ready to check the fit again, so press `f`. Now you should delete any obvious outliers. If the RMS is still larger than about 0.1, increase the order up to a maximum of 4. If the RMS is still too high, delete additional outlying points as needed. Once satisfied with the residuals, press `q` to return to the spectrum plot, and press `q` again to quit out of the identification process. Make sure to answer `yes` to the prompt in the IRAF terminal to write the feature data to the database. A new text file with the prefix "id" attached to the 0.5" blue arc filename is created in the database folder as a result of this step.

4.2 Running `id` to Identify the Lines in the Combined Red Side 0.5" Arc

Each night, we take a 15-second red side exposure through the 0.5" slit with the R1 set of lamps (Ar, He, Hg-Cd). We also take a 2-second red side exposure through the 0.5" slit with the R2 set of lamps (He, Hg-A, Ne). The reason why we can't just take a single 0.5" arce with the R1+R2 sets of lamps is that the Hg lamp is faint enough to require a 15 second exposure to achieve a strong signal, while the Ne lamp

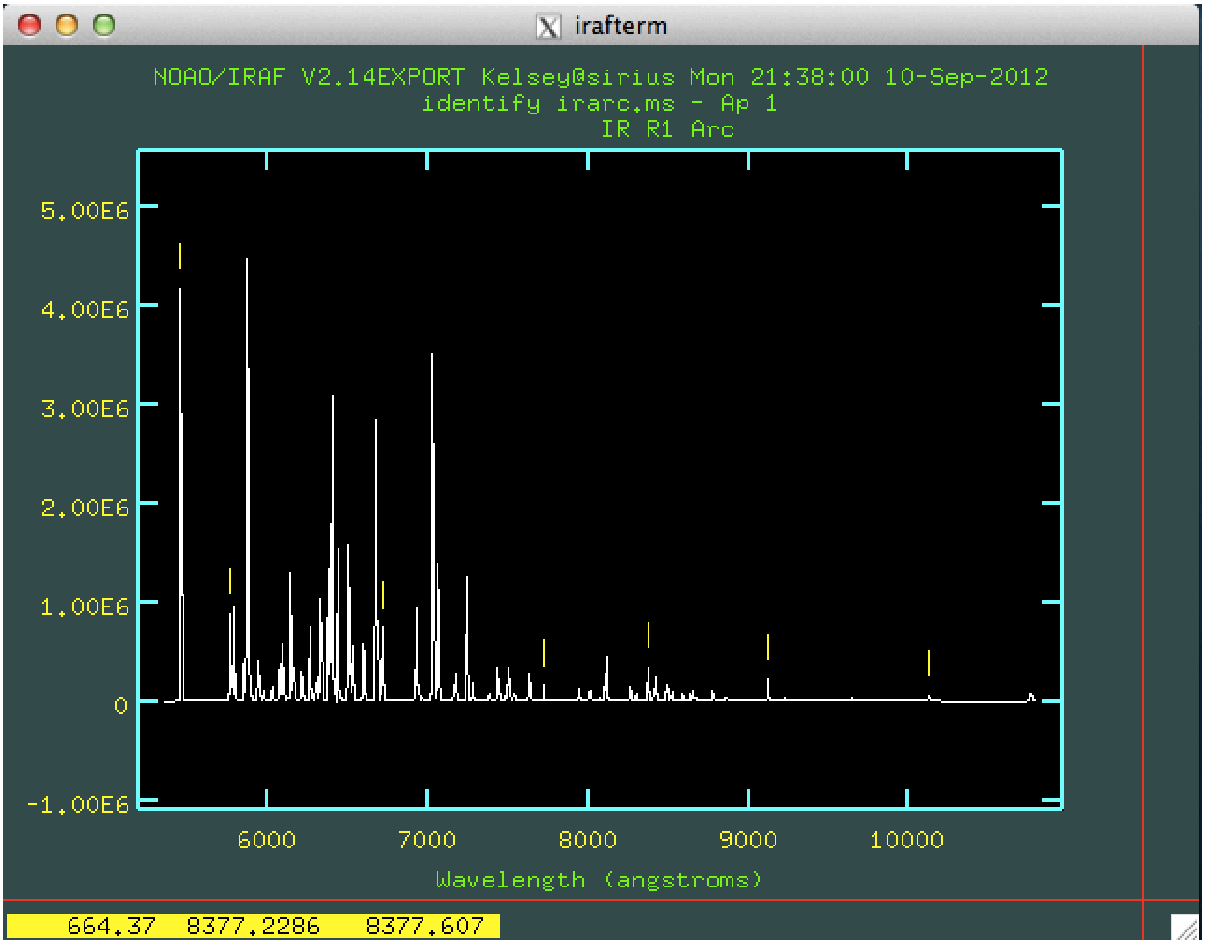


Figure 4.2: *An IRAF plot showing seven manually marked lines for the combined 0.5" red side arc.*

will saturate with an exposure longer than just 2 seconds. Our first step, then, is to combine these two red side 0.5" arcs into a single red side 0.5" arc. To do this simply type "imarith [0.5" IR arc R1 filename] + [0.5" IR arc R2 filename] [output filename]" (e.g. `imarith ftered002.ms + ftered003.ms irarc.ms`). This results in the creation of the newly combined 0.5" red side arc (likely named "irarc.ms.fits").

Next, in IRAF, run id on the combined 0.5" red side arc (e.g. type `id irarc.ms`, if you called the combined file "irarc.ms"). A plot of the combined 0.5" red side arc should now appear in the IRAF plot window. Now you're ready to identify a few of the lines manually. We'll want to choose 5-10 lines that span the entirety of the red ccd (as shown in Figure 4.2 above) to help IRAF find the most robust fit to these manually specified lines.

There are seven suggested lines to manually mark on the red side. The first obvious line is the 5460 Hg I line that we use to determine the desired wavelength coverage at the beginning of the night. It should be the strong, left most line on the red side arc (currently between column pixels 20-30). To identify this line manually, hover the cursor so that it lines up vertically with the 5460 line, press `m` to mark the line (a vertical yellow line will appear above the line in question), type `5460` in response the the IRAF feedback in the yellow stripe at the bottom of the plot window, and check that IRAF found the correct line in it's line list (e.g. 5460.7348 Hg I).

The next line usually manually identified is the first line to the right of 5460. It's the first line after the small gap to the right of 5460 and it's a somewhat weak and blended line, so you'll want to zoom-in on it to make sure you're choosing the correct line (it's currently between column pixels 90-100). This is the 5769 line. Mark the line as before and verify that IRAF found it to be 5769.5982 Hg I. The next line to mark is the last line before the next significant gap, 6717 (which is also somewhat weak and crowded, so you'll want to zoom-in before marking and it's currently between column pixels 300-310). IRAF should find this line to be 6717.0428 Ne I.

The next line to mark is the last line before the next significant gap, 7723 (currently between column pixels 520-530). IRAF should find this line to be 7723.7599 A I. Next, 8377 (which, is currently between column pixels 660-670). IRAF should find this line to be 8377.607 (no chemical element). Next, the most obvious, non-crowded line toward the right edge (which I guess you could say is the line after the next significant gap currently between column pixels 820-830) is 9122 and found to be 9122.966 (no chemical element designated). Finally, the last (barely discernable) line is 10123 (currently between column pixels 1045-1035), which IRAF should find as 10123.77 (no chemical element designated).

Now that you've manually marked a set of lines for the red side 0.5" arc, you'll want to press `f` to have IRAF find a fit to these lines and apply the fit across the entire red CCD. The IRAF plot window will update to show the residuals of the fit it found. If the RMS is larger than about 0.4, you should try a higher order fit (e.g., type `:ord 4` to change to a 4th order fit). After changing the order, you'll need to refit (press `f`). Increase the order number to a maximum of 6.

If the RMS is larger than about 0.4 with a 6th-order Legendre function, you should press `q` to return to the spectrum and double-check that you haven't made an error (if you have, you can hover the cursor over the incorrectly marked line and press `d` to delete the marking). If all marked lines check out, you should consult the Kast red side reference spectrum and mark (an) additional line(s) until you find a satisfactory fit.

Once you're satisfied with the RMS of the residuals, press `q` to return to the spectrum then press `l` to identify all of the lines. There should now be significantly more lines identified with a yellow line. Next, you'll want to look at every single one of the identified lines and make sure they look like real lines and not noise, aren't significantly blended, and aren't saturated.

To do this, hover the cursor so that the red vertical line is to the left of the 5460 line (the first line) and press `z` to zoom-in on the identified line nearest the cursor. If the identified line looks bad, press `d` to delete it, otherwise, if the identified line looks good, press `n` to skip to the next identified line. Once you've reached the last identified line, press `p` to pan back out to view the entire spectrum and all identified lines.

Now we're ready to check the fit again, so press `f`. Now you should delete any obvious outliers. If the RMS is still larger than about 0.3, increase the order up to a maximum of 6. If the RMS is still too high, delete any and all points larger than ± 1 Å, possibly even ± 0.5 Å. Once satisfied with the residuals, press `q` to return to the spectrum plot, and press `q` again to quit out of the identification process. Make sure to answer `yes` to the prompt in the IRAF terminal to write the feature data to the database. A new text file with the prefix "id" attached to the combined 0.5" red side arc filename is created in the database folder.

4.3 Running `reid` to Reidentify the Red Side 2.0" Arcs

4.3.1 For the First Red Side 2.0" Arc of the Night Only

Now that we have identified the lines in the combined 0.5" red side arc, we can use this result to reidentify the 2.0" red side arcs. For the first 2.0" red side arc of the night, type "reid [combined 0.5" red side arc filename] [first 2.0" red side arc filename]" (e.g. `reid irarc.ms ftered007.ms`). Since the combined 0.5" red side arc has lines from both the R1 and R2 sets of lamps, but the 2.0" red side arcs have only the R2 set of lines, you definitely want to respond `yes` to the IRAF terminal prompt of whether to fit the dispersion function interactively. The IRAF plot window should now show the identified lines for the first 2.0" red side arc of the night.

(Warning: We do not want to refit at all during this process, so be very careful to not press f at any time.) We do want to look through each individual line to make sure they are real and not significantly blended nor non-linear/saturated (flat-topped) as well as checking that the identified wavelength matches closely to the assigned wavelength (which will be displayed in the yellow stripe at the bottom of the plot window). So, hover the cursor to the left of the first identified line and press `z` to zoom-in and delete or move to the next line. (Some Tips: Usually the 4th line is off by more than a few Å, so it should be deleted. There may be a line at 6163 that is somewhat blended and off by 1.5 Å, so it should be deleted.)

Now press `p` to pan back out to the entire spectrum. If any of the non-linear/saturated lines are still marked delete them now. When satisfied that only good lines are marked, press `q` to quit and save the reidentification. Check that the RMS is less than about 0.5 (0.2-0.3 is ideal). If it's higher, you'll want to go through the re-reid the first 2.0" red side arc (which requires deleting the file created in the database folder described next). This step updated the logfile with the output of the reid program and creates a new text file with the prefix "id" attached to the first 2.0" arc filename.

4.3.2 For All but the First Red Side 2.0" Arc of the Night

Now that we have identified the lines in the first 2.0" red side arc of the night, we can use this result to reidentify the rest of the 2.0" red side arcs. First you'll need a list of the observation numbers for all of the rest of the 2.0" red side arcs. For each of the rest of the 2.0" red side arcs, type "reid [first 2.0" red side arc filename] [non-first 2.0" red side arc filename]" (e.g. `reid ftered007.ms ftered016.ms`).

Since we are now comparing arcs with identical setups, you only need to respond "yes" to the IRAF terminal prompt of whether to fit the dispersion function interactively if the RMS is greater than about 0.6 (ideally, it will be nearly the same as the RMS of the first 2.0" red side arc). Otherwise, you can simply type "no" and a new text file with the prefix "id" attached to the inputted 2.0" red side arc filename is created in the database folder.

4.4 Running `dcorr.cl` to Dispersion Correct All Objects

Now we are ready to run the night-specific IRAF script dcorr.cl and can do this by typing `cl < dcorr.cl` in our IRAF window. Check the output in the IRAF terminal for any error messages, such as that it could not find the reference file (which can happen if you forget to extract one of the red side arcs and/or blue side objects, etc.). This step creates a new file with a "df" prefix that is the bias-corrected, flattened, extracted, and dispersion-corrrected red or blue side spectrum for every standard, supernovae, and any other objects observed. And we are now done with the IRAF portion of the reduction process.

5 Flux-Calibrating and Telluric-Correcting the Object Spectra

5.1 Running `make_final_input_lists.py` to associate a standard with each object

You will likely have four standard star observations: a low airmass blue, a low airmass red, a high airmass blue, and a high airmass red. You will also likely have red and blue side observations of objects at a range of airmasses. We need to associate a standard with each object. Fortunately, there is a python program that will do this automatically. This program is called make_final_input_lists.py and should be located in the Kast tools folder (note: you will need to edit this program to point to the location of the Kast tools folder on your computer). Run this program within the run folder e.g., by typing `python /Users/Kelsey/observing/kast/tools/make_final_input_lists.py`.

This program will check all the dispersion corrected spectra to determine which are the standard stars. If necessary, it will the take the average of the airmasses for the low blue standard and the high blue standard and will separate the blue sides of each object into a low blue or high blue group (and do the same for the red standard(s) and object(s) as well, of course). So, it is actually possible that the blue side of an object can be considered to be at a high airmass, while the red side of the same object is considered to be at a low airmass.

The program currently assumes a uvir setup (i.e. the standard blue and red side settings) and that there are either one or two blue standards and one or two red standards. It will also only use the lowest and highest airmass blue and red standards. You should double-check the program's output in the terminal window and verify the name and observation number for each standard (the final column is airmass).

This step will create up to four text files within the run folder:

- **lowuv** - a list of the filenames for the low blue standard and all low airmass blue side objects
- **lowir** - a list of the filenames for the low red standard and all low airmass red side objects
- **hiuv** - a list of the filenames for the high blue standard and all high airmass blue side objects
- **hiir** - a list of the filenames for the high red standard and all high airmass red side objects

5.2 Running `cal.pro` for the low airmass blue group

Two specific things will make life significantly easier when fitting the standard with cal.pro. The first is, unless you're more comfortable with three-button mouse emulations with your keyboard and/or trackpad, you will benefit greatly by acquiring an actual mouse with three-button capability. Second, if you usually use Terminal.app on a Mac, you will want to run cal.pro in an X11.app window instead (this is because there is a significant amount of interaction between the terminal and the graphics window and, if you use Terminal.app, you'll have to click on the terminal window several times to respond to a prompt and then click on the graphics window to interact with the graph; this extra clicking is not necessary if you use an X11 terminal instead).

To run cal.pro, open up an X11 terminal window, cd to the location of the run folder, start up IDL, and (assuming the Kast tools folder is in your IDL path) simply type `cal`. Type `y` to select an interactive session. Provide the grating code (i.e., type `uv` if you're running cal on the low airmass blue group). Provide the dispersion-corrected filename for the appropriate standard star (e.g, type `dfteblue058.ms.fits`).

Note that this filename is also the first filename in the lowuv text file. Type `y` when asked if you want to use the same star as the b-star. Finally, provide the appropriate name of the text file that was created by make_final_input_lists.py (i.e., type `lowuv` if you're running cal on the low airmass blue group). Now a window should pop up with a plot of the appropriate standard star (the low airmass blue, in this case).

The plot should actually be showing two different extractions overplotted: An "optimal" (variance-weighted) extraction in white and a "normal" extraction in red (if your plot isn't displaying the correct colors, i.e., white text on a black background with both a white and red version of the standard star extraction plotted on the same plot, then you'll need to modify realmkfluxstar.pro, final.pro, and wombat.pro to redefine the color scheme that will work on your computer, i.e., the "col = [...]" lines). Type `o` to use the optimal extraction (unless the normal is significantly smoother, then you would type `n`). A list of standard star names should now appear and you'll want to enter the number corresponding to the standard star that is currently plotted (e.g. type `21` if the blue standard G191B2B is being plotted).

5.2.1 Fitting a Spline to the Continuum of the Low Airmass Blue Standard

It is now time to fit the continuum manually. The plot should now be showing the standard star plotted in white and the current spline fit as a green line. (Note that, on the red side, some segments of the standard star will be plotted in purple. These are the segments that contain telluric absorption, but there is no telluric absorption within the wavelength range of the blue side for Kast uvir observations.)

The program will ask you if the current fit is OK. Type `n`. It should then ask you if you want to change the y-scale (so that you can zoom-in on smaller segments and adjust the fit). Type `y`. The program now instructs you to mark the corners of your box. To do this, hover the cursor over the location within the plot window that you want to designate as the bottom left of the segment to zoom-in on. A red cross should appear at the location on the plot that you clicked. Now, hover the cursor over the location within the plot window that you want to designate as the top right of the segment to zoom-in on. The plot should now update to display only the zoomed-in segment.

Your job now is to add and/or delete red points (that are used to constrain the spline fit) in order to produce a smooth spline fit to the continuum of this standard star. As the program reminds you, you can add a point by hovering the cursor over the location at which you want to add a point and left-click (or it's non-mouse equivalent). A middle-click will delete the existing red point nearest to the cursor and subsequently replace it with a green cross. Finally, after you've added and/or deleted red points, you can right-click to refit with the current set of red points (the new fit will be plotted over the previous fit in the same color, so you'll probably want to type `n` when prompted if the current fit is OK and `n` when asked if you want to change the y-scale so that you can more clearly see the new fit at the same scale).

You'll want to repeat the process of zooming-in on smaller segments of the spectrum and adding and/or deleting points and refitting until the fit looks good across the entire spectrum. This is a somewhat subjective process and will likely take some practice to get good at knowing what a good fit looks like. An example of a good fit is shown in Figure 5.1 on page 26. When you are satisfied with the fit, you can type `y` when prompted if the current fit is OK. The program will save the fit as "fluxstar" plus the grating code you specified and the ".fits" suffix (e.g., "fluxstaruv.fits", if you just ran cal on the low blue standard). It will then automatically flux-calibrate the files that were assigned to this standard star's group (i.e. the filenames listed in the "lowuv" text file, if you just ran cal on the low blue standard). These flux-calibrated files will have a "c" prefix added to the input filename (e.g., "cdfteblue040.ms.fits").

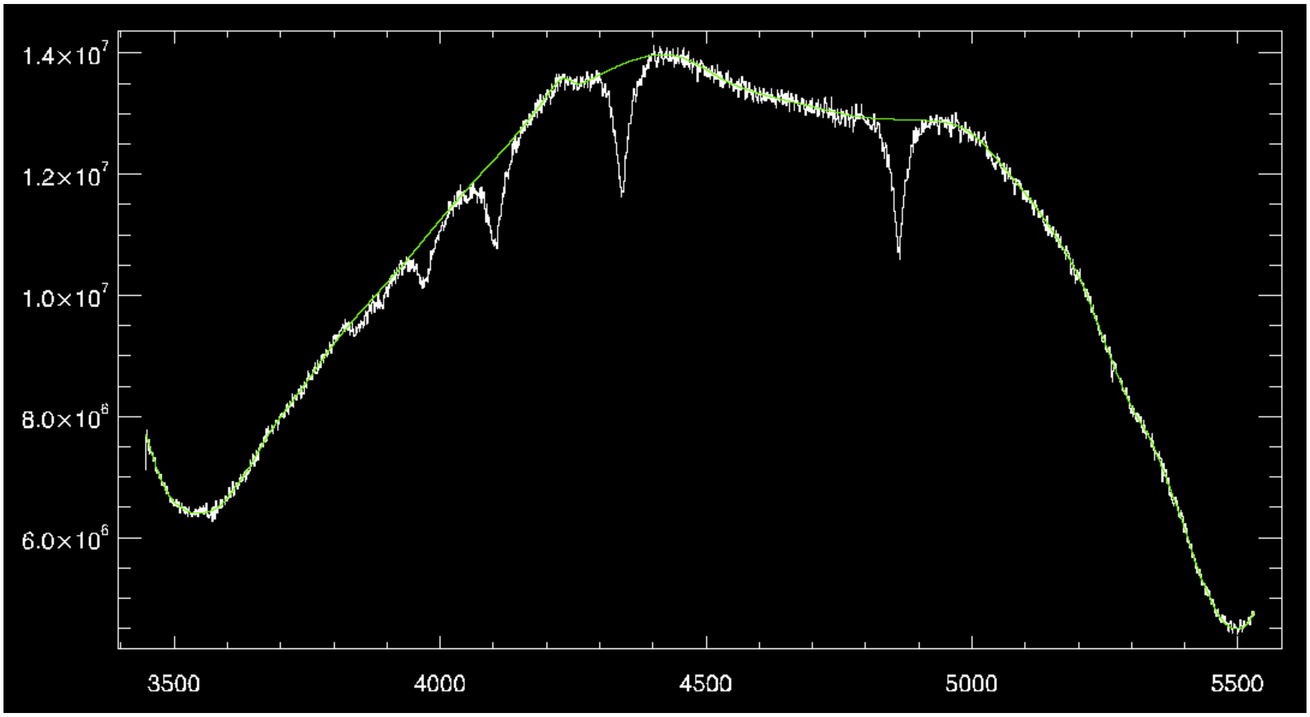


Figure 5.1: *An IDL plot showing a spline fit (in green) to the continuum of a low blue airmass standard (in white).*

5.2.2 Finishing the Reduction for the Low Airmass Blue Standard

The plot window should now be plotting the flux-calibrated standard star (optimal in white and normal in red). You are asked to choose which version you want to use and you should use the same version that you used in the previous step. Now you are asked above what airmass is considered high. This is no longer actually used, so just type `9`. Now we are prompted to fit the flux-calibrated continuum manually. Since the blue side of the Kast uvir setup has no telluric absorption, this step is unnecessary. Thus, even if the fit is bad, you can type `y` to the prompt "Is this ok? (y/n)". This step will create a file called "bstar" plus the grating code you specified and the ".fits" suffix (e.g., "bstaruv.fits" if you just ran cal on the low blue standard). We are now given the option of choosing the optimal or normal extraction and, again, you should choose the optimal unless the normal is significantly smoother. Now, you should see a plot of the master sky in white and the standard's sky in red. Since the standard has such a high signal-to-noise ratio, we don't care if the sky is lined up exactly or not, so type `y` to tell the program this is OK.

Now we're asked to specify how many Å/pixel we want to rebin to, always rebin to `2` Å/pixel. Next, we need to specify how much we want to clip off of the wavelength range. The plot window should now be showing the first and last 100 pixels of the spectrum. You want to clip off a few Å (maybe 5-20 or so) on each end and you always want to choose an even wavelength value (since we are binning to 2 Å/pixel). So, type the desired wavelength value for the left edge and the right edge (e.g., type `3450 5510`).

The flux vs. wavelength will now be plotted for the low airmass blue standard and you are prompted to give the object name for the final FITS file. It's common practice to use all lower case letters, so, e.g., type `g191b2b` or the name of the low blue standard you just fit if it wasn't G191B2B. This will create the final reduced FITS file for the blue side of the low blue standard (named, e.g., "g191b2b-20120907.545-uv.ms.fits").

5.2.3 Finishing the Reduction for the Low Airmass Blue Side Objects

The program will now go through the steps to finish the reduction for each low airmass blue side object observation. It will start by plotting the optimal and normal extractions for the first low airmass blue side object spectrum (as always, use the optimal unless the normal is significantly smoother). Next, you'll see a plot of the master sky spectrum in white and this object's sky spectrum shifted to match the master sky in red. Check that the two sky spectra are well matched across the wavelength span. If it doesn't look well matched, type `n` to manually shift the object's sky spectrum until they match well. If the two match well, type `y` to specify that this shift is OK.

Now you're asked to specify a value to rebin to. It should show you that your previous choice for the low blue standard star was 2 Å/pixel, so you can press `enter` to again rebin to 2 Å/pixel. Next, you're asked to specify a starting and ending wavelength. Again, it will show you your previous choice for the low blue standard star and it will even plot these values on each plot in red. If this still looks OK, press `enter` to choose the same wavelength range or type new starting and ending wavelength values.

The flux vs. wavelength will now be plotted for the first low airmass blue side object spectrum and you are prompted to give the object name for the final FITS file. It's common practice to use all lower case letters and you must include the full four digit year if it's an "SN" prefix object (e.g., "sn2012ea"). This will create the final reduced FITS file for the blue side of the object you just finished (with a name of the format "[objectname]-[utdate]-uv.ms.fits"). The program will repeat for each low airmass blue side object observation.

5.3 Running `cal.pro` for the low airmass red group

Now we're ready to run cal.pro for the low airmass red group. Type `cal` at the IDL prompt. Type `y` to select an interactive session. Provide the grating code (i.e., type `ir` if you're running cal on the low airmass red group). Provide the dispersion-corrected filename for the appropriate standard star (e.g, type `dftered080.ms.fits`). Note that this filename is also the first filename in the lowuv text file. Type `y` when asked if you want to use the same star as the b-star. Finally, provide the appropriate name of the text file that was created by make_final_input_lists.py (i.e., type `lowir` if you're running cal on the low airmass red group). Now a window should pop up with a plot of the appropriate standard star (the low airmass red, in this case).

The plot should actually be showing two different extractions overplotted: An "optimal" (variance-weighted) extraction in white and a "normal" extraction in red (if your plot isn't displaying the correct colors, i.e., white text on a black background with both a white and red version of the standard star extraction plotted on the same plot, then you'll need to modify realmkfluxstar.pro, final.pro, and wombat.pro to redefine the color scheme that will work on your computer, i.e., the "col = [...]" lines). Type `o` to use the optimal extraction (unless the normal is significantly smoother, then you would type `n`). A list of standard star names should now appear and you'll want to enter the number corresponding to the standard star that is currently plotted (e.g. type `1` if the red standard HD19445 is being plotted).

5.3.1 Fitting a Spline to the Continuum of the Low Airmass Red Standard

It is now time to fit the continuum manually. The plot should now be showing the standard star plotted in white and the current spline fit as a green line. You should also notice that some segments of the standard star will be plotted in purple. These are the segments that contain telluric absorption.

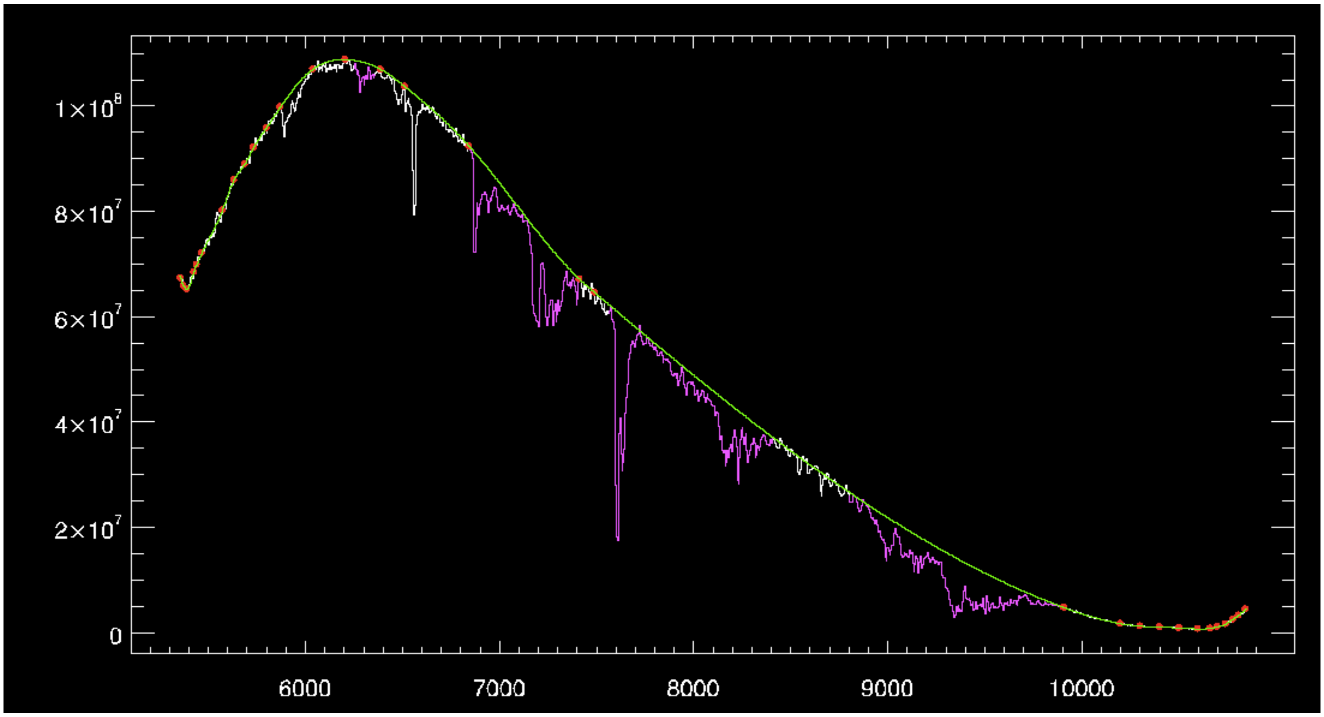


Figure 5.2: *An IDL plot showing a spline fit (in green) to the continuum of a low red airmass standard (segments with telluric absorption are plotted in purple, non-telluric segments are plotted in white).*

The program will ask you if the current fit is OK. Type `n`. It should then ask you if you want to change the y-scale (so that you can zoom-in on smaller segments and adjust the fit). Type `y`. The program now instructs you to mark the corners of your box. To do this, hover the cursor over the location within the plot window that you want to designate as the bottom left of the segment to zoom-in on. A red cross should appear at the location on the plot that you clicked. Now, hover the cursor over the location within the plot window that you want to designate as the top right of the segment to zoom-in on. The plot should now update to display only the zoomed-in segment.

Your job now is to add and/or delete red points (that are used to constrain the spline fit) in order to produce a smooth spline fit to the continuum of this standard star. As the program reminds you, you can add a point by hovering the cursor over the location at which you want to add a point and left-click (or it's non-mouse equivalent). A middle-click will delete the existing red point nearest to the cursor and subsequently replace it with a green cross. Finally, after you've added and/or deleted red points, you can right-click to refit with the current set of red points (the new fit will be plotted over the previous fit in the same color, so you'll probably want to type `n` when prompted if the current fit is OK and `n` when asked if you want to change the y-scale so that you can more clearly see the new fit at the same scale).

You should aim to smoothly interpolate over the telluric segments (in purple) and try not to place points within the telluric segments. You'll want to repeat the process of zooming-in on smaller segments of the spectrum and adding and/or deleting points and refitting until the fit looks good across the entire spectrum. This is a somewhat subjective process and will likely take some practice to get good at knowing what a good fit looks like. An example of a good fit is shown in Figure 5.2 above.

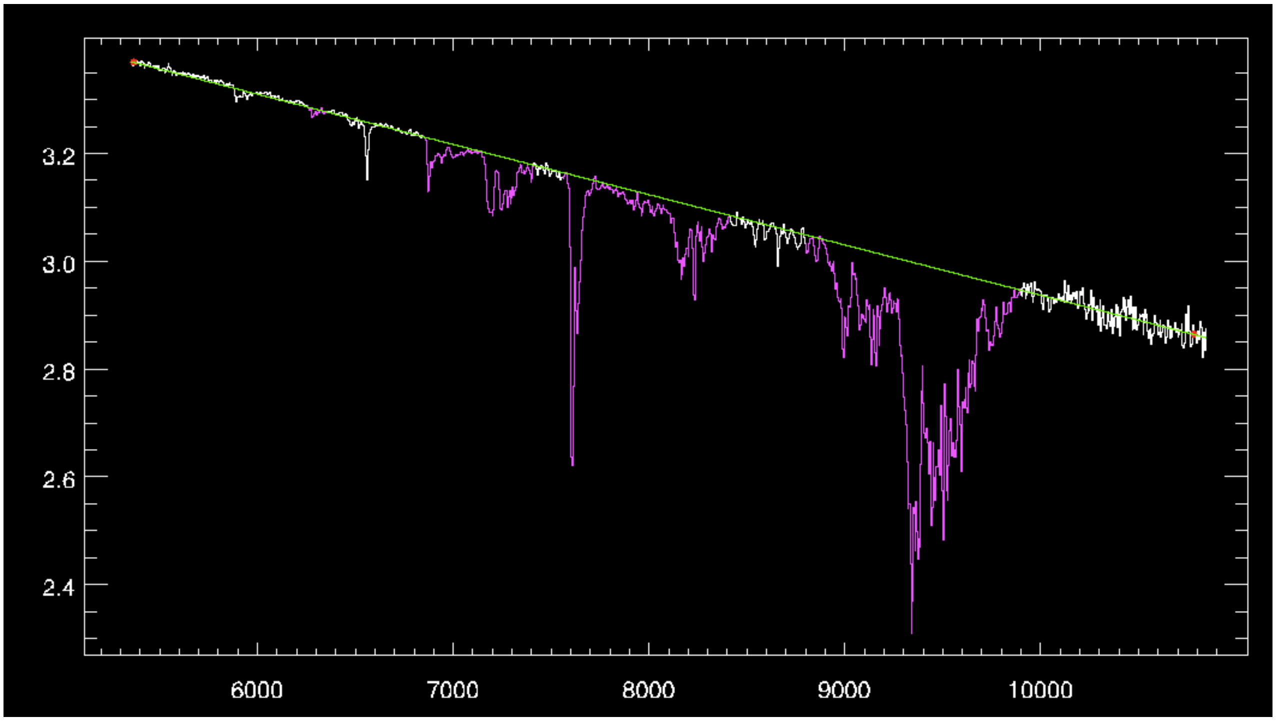


Figure 5.3: *An IDL plot showing a straight line fit (in green) to the flux-calibrated continuum of a low red airmass standard (segments with telluric absorption are plotted in purple, non-telluric segments are plotted in white).*

When you are satisfied with the fit, you can type `y` when prompted if the current fit is OK. The program will save the fit as "fluxstar" plus the grating code you specified and the ".fits" suffix (e.g., "fluxstarir.fits", if you just ran cal on the low red standard). It will then automatically flux-calibrate the files that were assigned to this standard star's group (i.e. the filenames listed in the "lowir" text file, if you just ran cal on the low red standard). These flux-calibrated files will have a "c" prefix added to the input filename (e.g., "cdftered080.ms.fits").

5.3.2 Finishing the Reduction for the Low Airmass Red Standard

The plot window should now be plotting the flux-calibrated standard star. Again, there are two versions: the "optimal" (variance-weighted) in white and the "normal" in red. You are asked to choose which version you want to use and you should use the same version that you used in the previous step. So, type `o` to use the optimal version or type `n` to use the normal version. Now you are asked above what airmass is considered high. This is no longer actually used, so just type `9`.

Now we are ready to fit the flux-calibrated continuum manually. The flux-calibrated continuum should be a straight line, so, unless the default fit is miraculously already a straight line, type `n` to the prompt "Is this ok? (y/n)". Then, you'll want to adjust the fit as before until you have a straight line going through the middle of all the noise within the white segments and skimming the top of the purple segments. A good way to start here is to delete all default points that are obviously high or low then, if needed, add points as necessary to produce a straight line fit. An example of a good fit is shown in Figure 5.3 above.

When you are satisfied with the fit, you can type `y` when prompted if the current fit is OK. The pro-

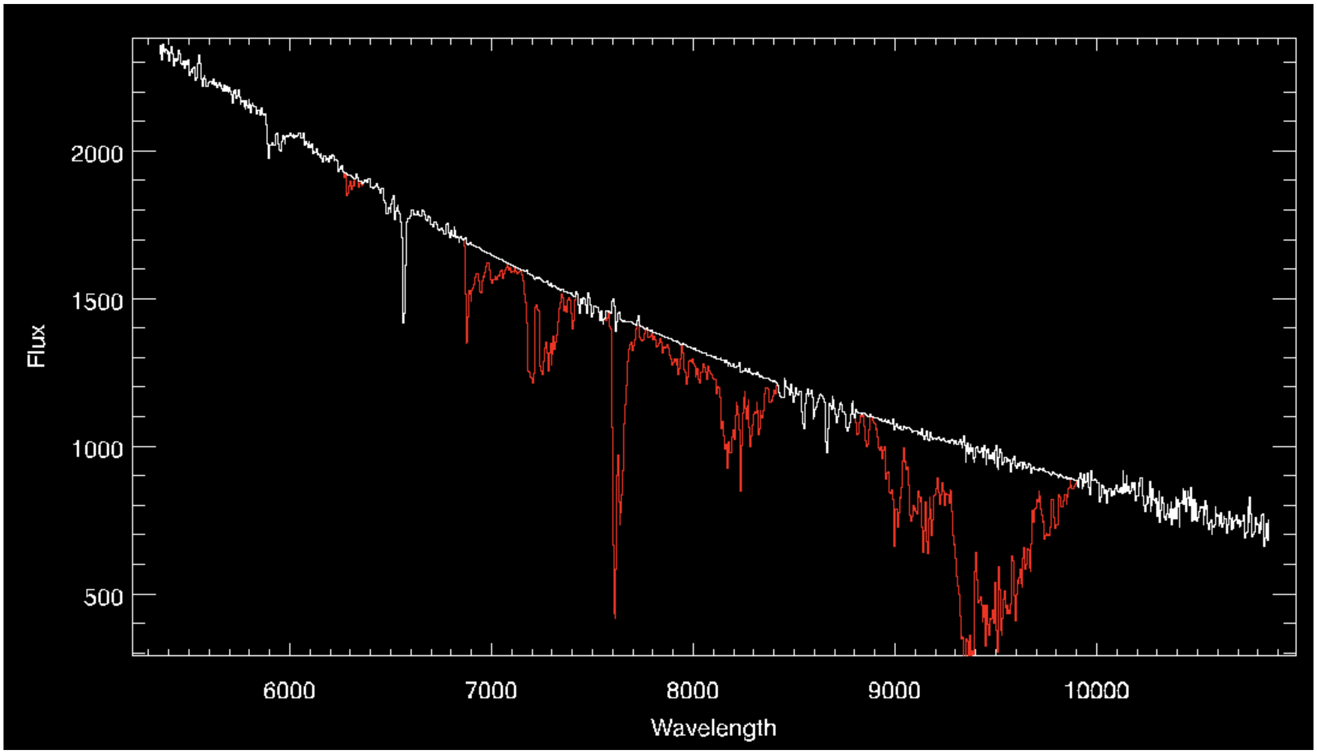


Figure 5.4: *An IDL plot showing the spectrum of the telluric-corrected low red standard (in white) overplotted on the spectrum of the low red standard before telluric correction (in red).*

gram will save the fit and create a file called "bstar" plus the grating code you specified and the ".fits" suffix (e.g., "bstarir.fits" if you just ran cal on the low red standard). This is how we can correct for telluric absorption. For each column pixel within a telluric (purple) segment, the ratio of the data value to value of the straight line fit will be calculated and multiplied back in to all of the object spectra, in essence adding back in all of the flux that was absorbed by the atmosphere before reaching the CCD.

We are now given the option of choosing the optimal or normal extraction and, again, you should choose the optimal unless the normal is significantly smoother (so type `o` or `n`). Now, you should see a plot of the master sky in white and the standard spectrum's sky in red. Since the standard has such a high signal-to-noise ratio, we don't care if the sky is lined up exactly or not, so type `y` to tell the program this is OK. Now we're shown the result of the telluric correction. The non-telluric-corrected spectrum is plotted in red and the telluric-corrected spectrum is overplotted in white (as seen in Figure 5.4 above). This should always look OK since we used the standard to correct for the telluric absorption, so type `y` to tell the program this is OK.

Now we're asked to specify how many Å/pixel we want to rebin to, always rebin to `2` Å/pixel. Next, we need to specify how much we want to clip off of the wavelength range. The plot window should now be showing the first last 500 pixels of the spectrum. You want to clip off a few Å (maybe 5-20 or so) on the left end and up to 50 or so on the right end. You also always want to choose an even wavelength value (since we are binning to 2 Å/pixel). So, type the desired wavelength value for the left edge and the right edge (e.g., type `5362 10800`).

The flux vs. wavelength will now be plotted for the low airmass red standard and you are prompted to give the object name for the final FITS file. It's common practice to use all lower case letters, so, e.g., type `hd19445` or the name of the low blue standard you just fit if it wasn't HD19445. This will create the final reduced FITS file for the red side of the low red standard (named, e.g., "hd19445-20120907.539-ir.ms.fits").

5.3.3 Finishing the Reduction for the Low Airmass Red Side Objects

The program will now go through the steps to finish the reduction for each low airmass red side object observation. It will start by plotting the optimal and normal extractions for the first low airmass red side object spectrum (as always, use the optimal unless the normal is significantly smoother). Next, you'll see a plot of the master sky spectrum in white and this object's sky spectrum shifted to match the master sky in red. Check that the two sky spectra are well matched across the wavelength span. If it doesn't look well matched, type `n` to manually shift the object's sky spectrum until they match well. If the two match well, type `y` to specify that this shift is OK.

Now we're shown the result of the telluric correction. The non-telluric-corrected spectrum is plotted in red and the telluric-corrected spectrum is overplotted in white. (Note that, unlike for the standard, the tellurically-corrected segments likely won't be flat.) If it doesn't look like the telluric-correction was done correctly, you can answer "no" this isn't OK and you should be able to manually choose a different shift value. Otherwise, if the correction looks good, you can type `y` to tell the program this is OK.

Now you're asked to specify a value to rebin to. It should show you that your previous choice for the low red standard star was 2 Å/pixel, so you can press `enter` to again rebin to 2 Å/pixel. Next, you're asked to specify a starting and ending wavelength. Again, it will show you your previous choice for the low red standard star and it will even plot these values on each plot in red. If this still looks OK, press `enter` to choose the same wavelength range or type new starting and ending wavelength values.

The flux vs. wavelength will now be plotted for the first low airmass red side object spectrum and you are prompted to give the object name for the final FITS file. It's common practice to use all lower case letters and you must include the full four digit year if it's an "SN" prefix object (e.g., "sn2012ea"). This will create the final reduced FITS file for the red side of the object you just finished (with a name of the format "[objectname]-[utdate]-ir.ms.fits"). The program will repeat for each low airmass red side object observation.

5.4 Running `cal.pro` for the high airmass blue and red groups

The procedure for running cal.pro for the high airmass blue and red groups is nearly exactly the same as for the low airmass blue and red groups. The first thing you need to be aware of is, if you reuse the "uv" and "ir" grating codes, the files corresponding to the low airmass versions of fluxstaruv.fits, fluxstarir.fits, bstaruv.fits, and bstarir.fits will be overwritten. It is possible you'll want to reuse these files, so before running cal.pro on the high airmass groups, you should create a new directory in the run folder (call it "lowairmass_fluxstar", e.g.) and move those four files into this new folder.

After you've run cal.pro for the high airmasss blue and red groups, it's a good idea to go ahead and create a new directory in the run folder (call it "highairmass_fluxstar", e.g.) and move the high airmass versions of fluxstaruv.fits, fluxstarir.fits, bstaruv.fits, and bstarir.fits into this new folder.

6 Putting the Spectra into Final Reduced Form

6.1 Running `wombat.pro` to Combine Multiple Blue or Red Sides of an Object

In the case for which you have observed a faint object and obtained more than one blue side (and/or red side) exposure, you'll first want to co-add the two or more blue side (and/or red side) exposures. We can use the IDL program wombat.pro to do this. Make sure you're in the run folder, start IDL, then type `wombat`. First, read in the ".ms.fits" file for the first exposure. To do this, type `rerrfits`, then type the name of the first file (e.g., `psn23+14-20120907.372-ir.ms.fits`). Now, we need to store this data into a hopper (a container to store data). To do this, type `hop`, then press `1` to store the first exposure in hopper 1. Now type `rerrfits` again, then type the name of the second file (e.g., `psn23+14-20120907.393-ir.ms.fits`). Now, store this data into a hopper by typing `hop`, then press `2` to store the second exposure in hopper 2.

To co-add two exposures, type `com`. You will be prompted for the hopper numbers of the two spectra you would like to add and then prompted to store the resulting co-addd spectrum in yet another hopper. You can only co-add two exposures at a time. Thus, if you have more than two exposures, you'll have to co-add each additional exposure with the previously co-added exposures. Make sure to only co-add blue side exposures with blue side exposures, etc. You'll want to co-add any and all exposures for a single object before you attempt to combine the blue and red sides for that object.

6.2 Running `wombat.pro` to Combine the Blue and Red Sides of Each Object

We'll use the IDL program wombat.pro to combine the blue and red sides of each object using the overlap region. Make sure you're in the run folder, start IDL, then type `wombat`. First, we want to read in the ".ms.fits" file for the blue side. To do this, first type `rerrfits`, then type the name of the blue side file (e.g., `ptf12gzk-20120907.203-uv`). Now, we need to store this data into a hopper (a container to store data). To do this, type `hop`, then press `1` to store the blue side in hopper 1. Next, we want to read in the ".ms.fits" file for the red side. To do this, first type `rerrfits`, then type the name of the red side file (e.g., `ptf12gzk-20120907.203-ir`). Now, we need to store this data into a hopper. To do this, type `hop`, then press `2` to store the red side in hopper 2.

Now, we want to combine the blue and red sides for this object using the overlap region. To do this, first type `cat`, then press `m` to use the mouse to define the overlap region, then press `1` to use hopper 1 (the blue side), then press `2` to use hopper 2 (the red side). The graphics window should now be plotting flux vs. wavelength for the blue side in blue and the red side in red. If you can see the blue side, but not the red side (or want to change the y-scale for a different reason), press `y` to change the y-scale, otherwise press `n` to leave the y-scale as is.

You are now prompted to mark the left and right edges of the portion of the overlap region within which the blue and red sides seem to line up (the blue flux goes up as the red flux goes up and vice versa). (Note: This wavelength range will only be used to compute an average, not actually define the overlap region.) To do this, hover the mouse such that it lines up vertically with what you want to select as the left edge and click. A small red dot should appear at the location of your click. Now, hover the mouse such that it lines up vertically with what you want to select as the right edge and click. Another small red dot should appear at the location of your click. If these points are as you wanted, press `y`.

Now you're asked whether you want to scale to the blue side or the red side. Unless the blue side looks significantly better, press `r` to scale to the red side. The program takes the average of the blue flux within the wavelength range you chose and the average of the red flux within the wavelength range you chose and then replots the overlap region scaling the blue flux to the red flux. Press `n` to leave the y-scale as is (or press `y` to change the y-scale).

You are now prompted to mark the left and right edges of the portion of the overlap region within which the blue and red sides seem to line up (the blue flux goes up as the red flux goes up and vice versa). (Note: This wavelength range will actually be the region used to combine the blue and red side together.) To do this, hover the mouse such that it lines up vertically with what you want to select as the left edge and click. A small red dot should appear at the location of your click. Now, hover the mouse such that it lines up vertically with what you want to select as the right edge and click. Another small red dot should appear at the location of your click. If these points are as you wanted, press `y`.

Now you're asked whether you want to combine the overlap regions equally, with weights, or by variance. Unless one side is significantly noisier than the other, press `e` to combine the blue and red sides equally. The graphics window should now be plotting flux vs. wavelength for the (small) overlap region in the top portion (blue side in blue, red side in red, and the average of the two in white) and flux vs. wavelength for the combined red and blue sides in the bottom portion (overlap region in red).

Press `3` to store the combined blue and red sides into hopper 3. Now, we want to do a quick check by plotting the combined spectrum and then overplotting the blue side and the red side. To do this, press `p` to plot the current hopper (hopper 3, the combined spectrum). Press `n` if the y-scale is fine as is. Type `rh` to read in the data from a different hopper. Press `2` to read in the data in hopper 2 (the red side). Type `apl` to overplot hopper 2 (the red side) onto hopper 3 (the combined spectrum). (Note: I'm overplotting the red side first because the default parameters will overplot the first overplot in red and the second overplot in blue, thus the colors will match the sides! Also, if you scaled to the red side, the red side should simply plot directly over the combine spectrum.) Type `rh` to read in the data from a different hopper. Press `1` to read in the data in hopper 1 (the blue side). Type `apl` to overplot hopper 1 (the blue side) onto hopper 3 (the combined spectrum).

If there are any big cosmic rays and/or bad pixels, you can press `b` to enter the blotch routine. Use the mouse to select the region to zoom in on and use the mouse to click the edges of the bad region. You can blotch more ranges if needed. You might want to store the data in a new hopper after blotching.

Type `rh` and press `3` (or the appropriate hopper number) to read the combined spectrum into the current hopper. Press `p` to plot the current hopper and verify it's the combined spectrum (as seen in Figure 6.1 on page 34), then press `n` to leave the y-scale as is. Finally, type `wpl` to write out a three-column text file (wavelength, flux, and uncertainty in the flux) of the current hopper (which should be the combined spectrum). You'll be asked to provide a name for this new file and the convention is objectname-utdate-ui.flm, where the utdate is the same number that was in the original .ms.fits file name for the blue and red side (note that this is actually the fractional day in UT at the midpoint of the observation). So, e.g., type `ptf12gzk-20120907.203-ui.flm`.

Now, you just need to repeat the steps in this section for each of the objects that were observed!

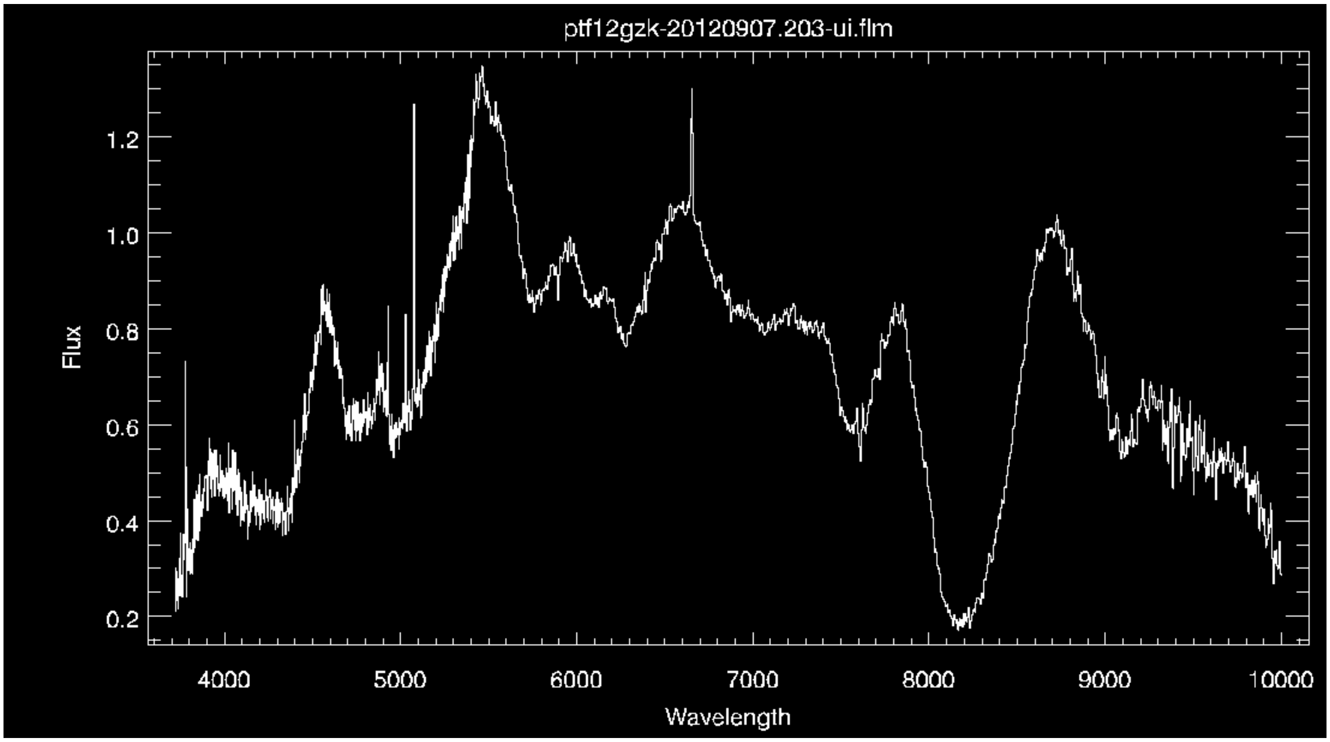


Figure 6.1: *An IDL plot showing the final reduced spectrum for PTF 12gzk.*

6.3 Making a PostScript File of the Final Reduced Spectrum for Each Object

It is helpful to create a .ps file of the final reduced spectrum for each object so that collaborators can easily view the final reduced spectrum. To do this, start IDL in the run folder. Read in the .flm file with rdfloat (e.g., type `rdfloat,'ptf12gzk-20120907.203-ui.flm',wav,flux`). Next, use spec.pro to display the spectrum (e.g., type `spec,wav,spec,title='PTF 12gzk',xtit='Wavelength',ytit='Flux'`). This will cause an IDL GUI to appear on the screen displaying the spectrum and giving several options at the top of the GUI. Click on "Make PS Plot" to write out a .ps version of the displayed spectrum. By default this will create a new file called "temp.ps" so you'll need to rename that file appropriately (e.g., type `$mv temp.ps ptf12gzk-20120907.203-ui.ps`). Now, you just need to repeat these few steps for each of the objects that were observed!