Astronomy Camp Data Reduction of Long-slit spectra using iraf

Created: June 16th, 2023
Updated: June 22nd, 2024

Iraf is the software used by most astronomers to reduce long-slit spectroscopic observations. In this manual, instructions are based on data taken at the Kuiper telescope and edited from reduction protocols which have been used for published astronomical results.

However, iraf was first created in 1993 and still works like it was created in 1993. Therefore, there are a few quirks you should know about before embarking on the journey of spectroscopic reduction.

Important to Note:

- **NEVER just click the X button** on an interactive window to close it. **Press `q` FIRST** then X.
    - If you accidentally do click the X, keyboard slam for ~3 seconds. If the window does not pop-back up, give the computer a minute or two and ask for help.
- You will not be able to backspace. This is extremely annoying but unavoidable. We recommend copy and pasting directly from a document to avoid having to retype things over and over again.
- Copy-pasting with ctrl-c and ctrl-v will not work. You will need to right click to copy and paste.
    - You may need to use a center mouse button, if you find this to be the case, find a counselor to see if they have one
    - **OR** update your settings so you can copy-paste using the trackpad
        - Go into `XQuartz > Preferences > Input > Emulate three button mouse`
        - To paste it will be hold option then click trackpad.

- Please **ask for help** if you run into any problems. This software is much older than you are, expect it to behave like a senile computer program.

0. Install

Iraf is currently installed on the computers, you should not need to worry about this.

For future reference (from Austin):

Installed miniconda using this: https://docs.anaconda.com/miniconda/
`conda list`
outputs correctly.

Add 32 bit dependencies for ubuntu

```
sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get install libc6:i386 libz1:i386
libncurses5:i386 libbz2-1.0:i386 libuuid1:i386
libxcb1:i386 libxmu6:i386
```

Followed this guide for IRAF:
https://astroconda.readthedocs.io/en/latest/installation.html#iraf-install

IRAF is once again supported by NOIRLab details on the newest version can be found here:
https://iraf.noirlab.edu/

1. Starting up

To start up iraf, you need to type the following:

```
xgterm &
```

This should pop up a *white* terminal window. In that window type:

```
conda activate iraf27
mkiraf
cl
```

You should get a message letting you know that iraf has started up.
To load all the packages you will need to use type the following:

```
mscred
imred
ccdred
onedspec
twodspec
longslit
apextract
```

If this worked it should say “apextract” in your input box.

Before you start reducing your spectra, open up your spectrum file and take a look at it. Note down anything that looks interesting about it.

To do this type **in your purple terminal:**

```
conda activate iraf27
ds9 <your_spectrum_filename> &
```

This may take a minute but should pop up a window with your raw spectrum image in it. So you can see everything, go to scale on the toolbar above the image and press zscale.

There should be 3 different rows of light, two of them are the same and mirrored on either side of the science data. These are arc spectra, note down the x coordinates at about the middle of one of the arc spectra. You will use this in step 3.

Leave the window with your spectrum open in case you need to reference it while reducing your spectrum.

2. Zero and Flat correction

In order to account for noise from the telescope detector, we apply flat and zero corrections.
Zeros (aka biases) are zero second exposures which give a measure of the inherent noise of the detector (read noise). Flats (aka continuum flats) in contrast are several second exposures, generally using a white continuum lamp in the case of long-slit spectroscopy, used to determine and correct for the optics of the telescope which can flex and contact with temperature and moisture changes.

First we bias correct *all* the images (copy paste the following into your terminal):

```
zerocombine.input = "zero.*"
zerocombine.output = "bias"
zerocombine.combine = "average"
zerocombine.reject = "minmax"
zerocombine.ccdtype = " "
zerocombine.process = no
zerocombine.delete = no
zerocombine.clobber = no
zerocombine.scale = "mode"
zerocombine.statsec = ""
zerocombine.nlow = 1
zerocombine.nhigh = 1
zerocombine.nkeep = 1
zerocombine.mclip = yes
zerocombine.lsigma = 3.
zerocombine.hsigma = 3.
zerocombine.rdnoise = "3.5"
zerocombine.gain = "GAIN"
zerocombine.snoise = "0."
zerocombine.pclip = -0.5
```

```
zerocombine.blank = 1.
zerocombine.mode = "al"
zerocombine
```

Now we apply it to all the images:

```
ccdproc.images = "Change this to include ALL your fits images except your zeros"
ccdproc.output = ""
ccdproc.ccdtype = " "
ccdproc.max_cache = 0
ccdproc.noproc = no
ccdproc.fixpix = no
ccdproc.overscan = no
ccdproc.trim = no
ccdproc.zerocor = yes
ccdproc.darkcor = no
ccdproc.flatcor = no
ccdproc.illumcor = no
ccdproc.fringecor = no
ccdproc.readcor = no
ccdproc.scancor = no
ccdproc.readaxis = "line"
ccdproc.fixfile = ""
ccdproc.biassec = ""
ccdproc.trimsec = ""
ccdproc.zero = "bias.fits"
ccdproc.dark = ""
ccdproc.flat = ""
ccdproc.illum = ""
ccdproc.fringe = ""
ccdproc.minreplace = 1.
ccdproc.scantype = "shortscan"
ccdproc.nscan = 1
ccdproc.interactive = no
ccdproc.function = "legendre"
```

```
ccdproc.order = 1
ccdproc.sample = "*"
ccdproc.naverage = 1
ccdproc.niterate = 1
ccdproc.low_reject = 3.
ccdproc.high_reject = 3.
ccdproc.grow = 0.
ccdproc.mode = "al"
ccdproc
```

Now we do a similar thing for all our continuum flats:
Change the input file names so they include all your flats.

```
flatcombine.input = "FLAT*"
flatcombine.output = "master_flat"
flatcombine.combine = "average"
flatcombine.reject = "crreject"
flatcombine.ccdtype = " "
flatcombine.process = no
flatcombine.subsets = no
flatcombine.delete = no
flatcombine.clobber = no
flatcombine.scale = "mode"
flatcombine.statsec = ""
flatcombine.nlow = 1
flatcombine.nhigh = 1
flatcombine.nkeep = 1
flatcombine.mclip = yes
flatcombine.lsigma = 3.
flatcombine.hsigma = 3.
flatcombine.rdnoise = "3.5"
flatcombine.gain = "GAIN"
flatcombine.snoise = "0."
flatcombine.pclip = -0.5
flatcombine.blank = 1.
flatcombine.mode = "al"
```

```
flatcombine

response.calibration = "master_flat"
response.normalizatio = "master_flat"
response.response = "normal_flat"
response.interactive = yes
response.threshold = INDEF
response.sample = "*"
response.naverage = 1
response.function = "spline3"
response.order = 40
response.low_reject = 3.
response.high_reject = 3.
response.niterate = 1
response.grow = 0.
response.graphics = "stdgraph"
response.cursor = ""
response.mode = "al"
response
```

This will pop up an interactive window. If you want, you can press `d` while hovering over any points that don't follow the dotted line, then press `f`. Press `q` and `enter` on any dialogue boxes that pop up, until your cursor is once again writing in the terminal. Now you can press the X on the window to make it disappear.

Now we apply the flat correction:

```
ccdproc.images = "Change to be all science files"
ccdproc.flat = "normal_flat"
ccdproc.flatcor = yes
ccdproc
```

Congratulations all your science files are now bias and flat corrected!

3. Wavelength Calibration

a. Identifying lines

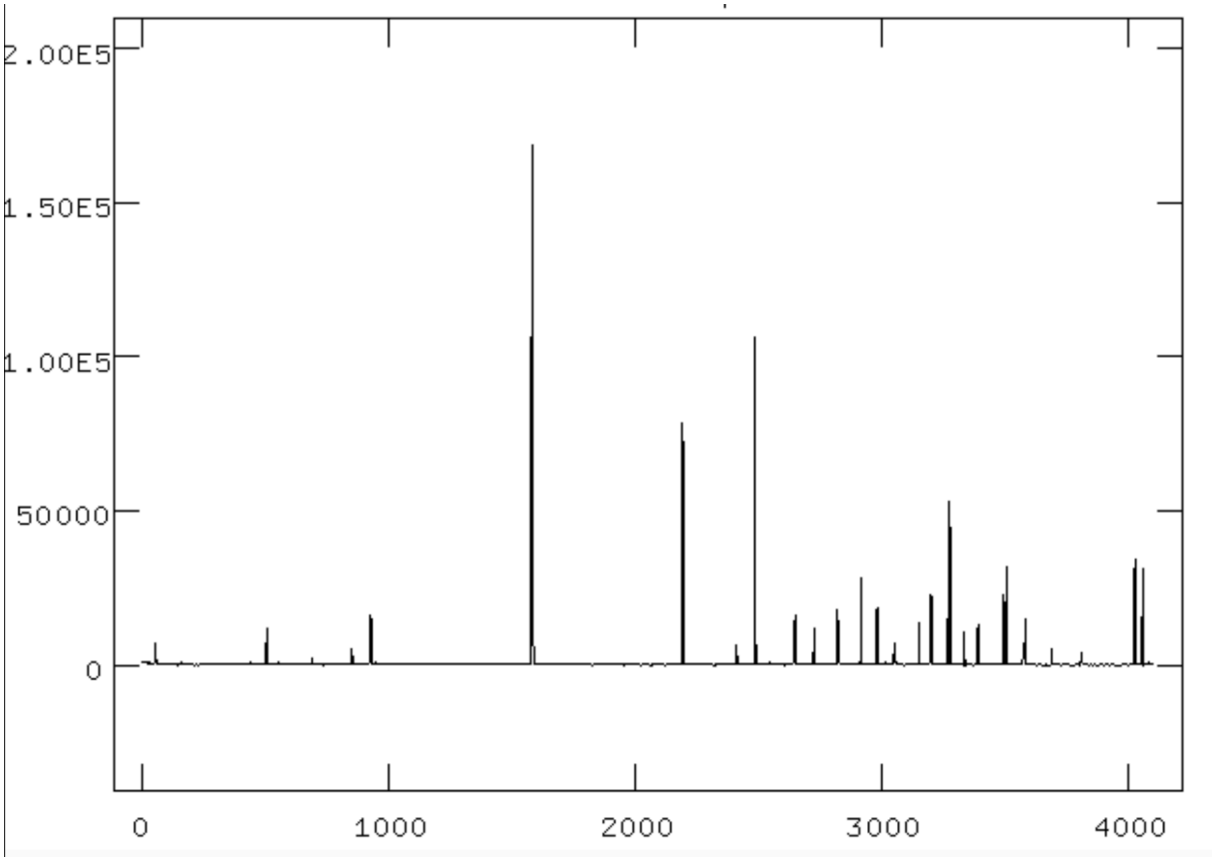
The image we took has lines in pixel space but we need them in terms of wavelength.

To identify the emission lines, and determine the dispersion relation, i.e. the relation between position along the x-axis and wavelength, use the identify task:

```
identify.images = "Your science image"
identify.section = "line your_arc_spectrum_x_coord"
identify.database = "database"
identify.coordlist = "linelists$idhenear.dat"
identify.units = ""
identify.nsum = "3"
identify.match = -3.
identify.maxfeatures = 50
identify.zwidth = 100.
identify.ftype = "emission"
identify.fwidth = 4.
identify.cradius = 5.
identify.threshold = 0.
identify.minsep = 2.
identify.function = "spline3"
identify.order = 1
identify.sample = "*"
identify.niterate = 0
identify.low_reject = 3.
identify.high_reject = 3.
identify.grow = 0.
identify.autowrite = no
identify.graphics = "stdgraph"
identify.cursor = ""
identify.aidpars = ""
```

```
identify.mode = "al"
identify
```

This should pop up a window that looks something like this (the lines will not be the same):

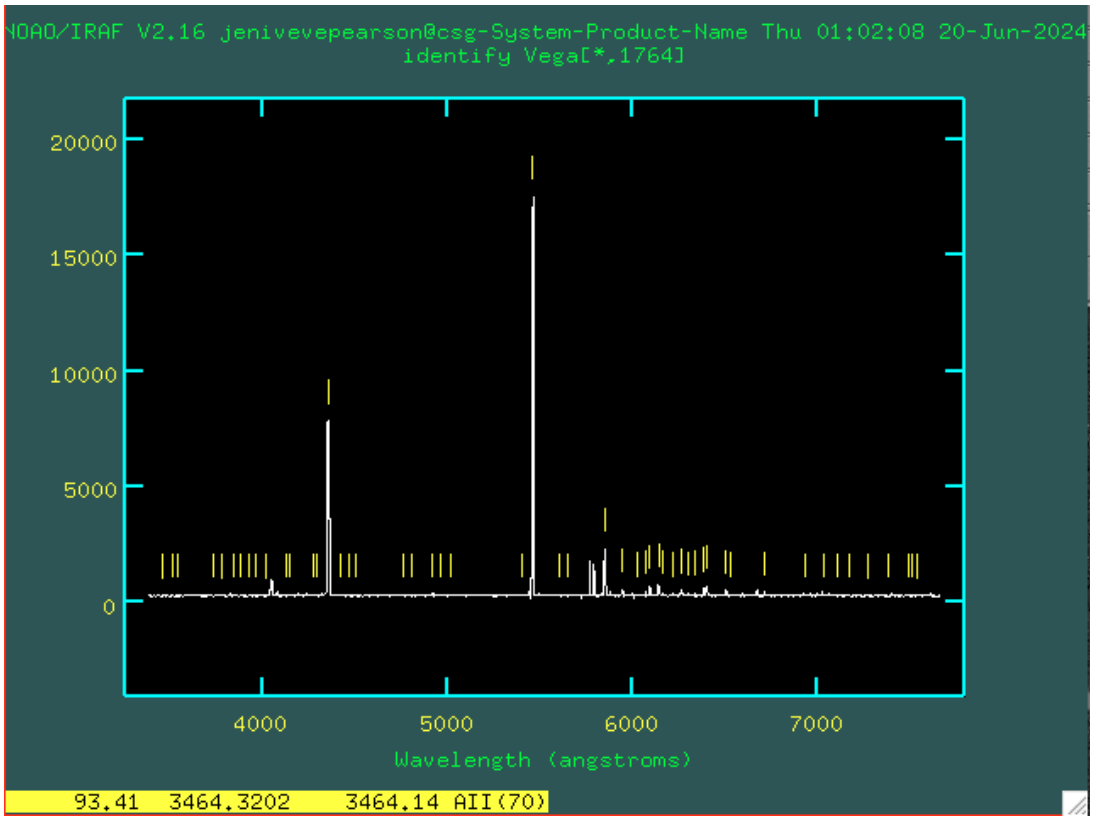


The lines you see should be from the following list:

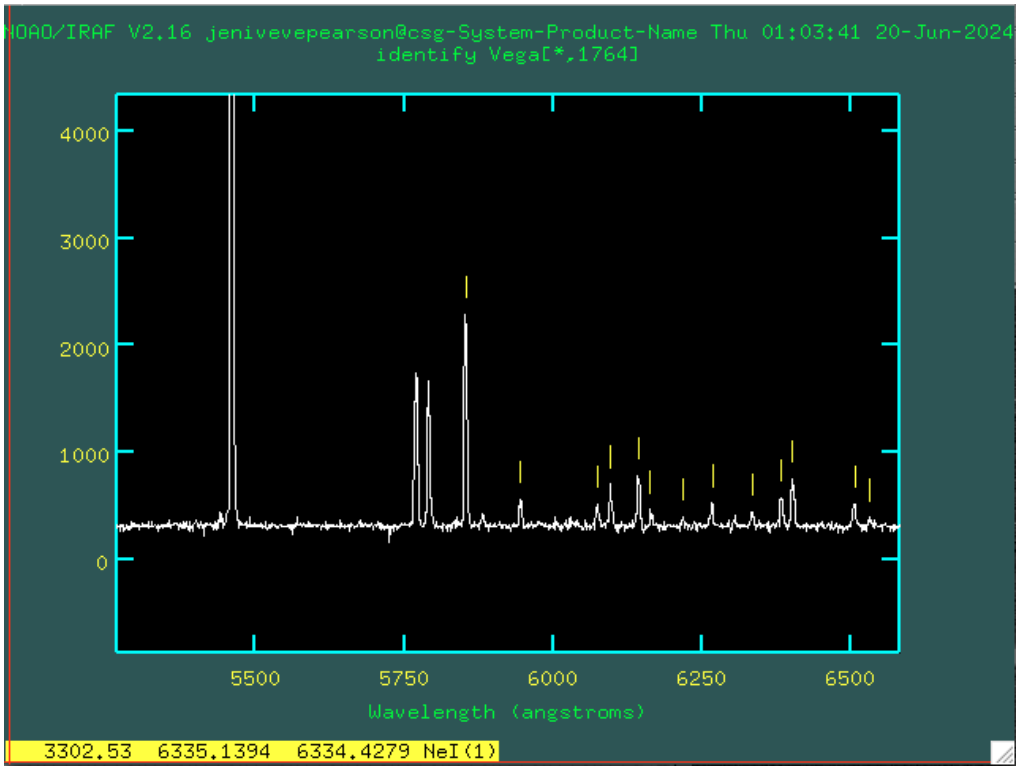
Wavelength (A)	Pixel (x axis)
4047	788
4077	819
4358	1134
5461	2351
5770	2688
5790	2717
5852	2781
6402	3378

You need to map the pixel values (where the lines in the window currently are) to their associated wavelength values. Pick only 3-4 to use. We recommend determining which lines are which before you officially assign them.

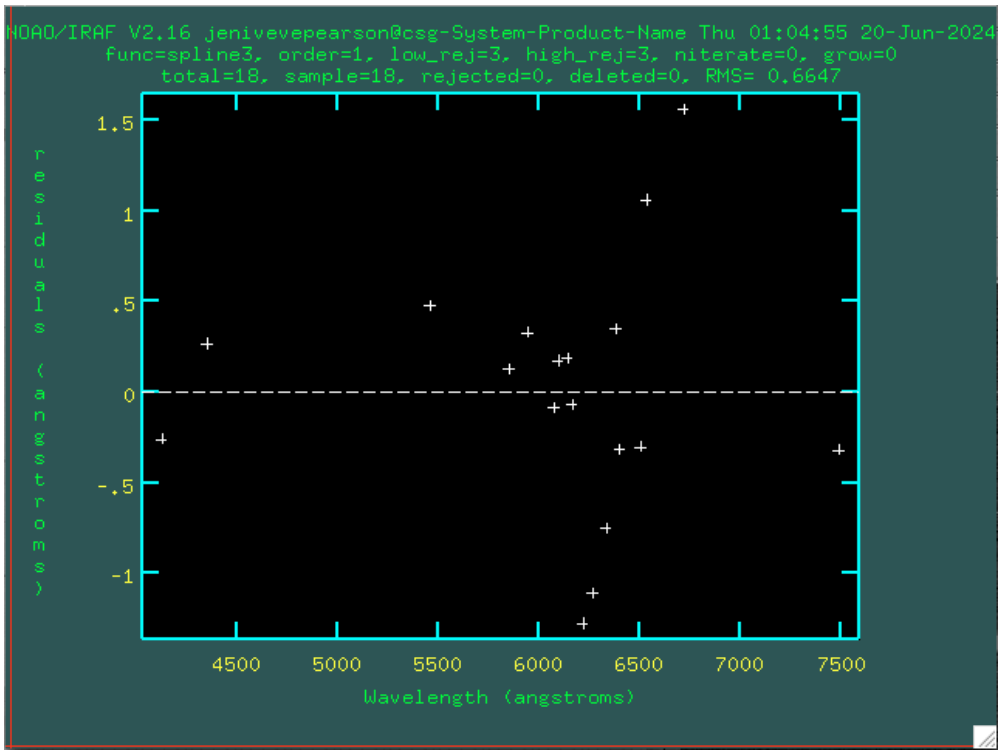
First, mark 3-4 lines by placing the cursor on them, hitting "`m`" and entering the wavelength of the line. Type "`l`" to automatically identify more lines. Your window should now look like this:



CHECK that these are correct using the line list. Delete any identification that are just noise (i.e. have no clear bump associated) using "`d`". Zoom in using `w e` (lower left corner you want) and `e` again (upper right corner you want), reset the zoom by using `w a`. The lines with yellow ticks below are NOT noise, keep lines that look like these ones.



Hit “f” to fit the wavelength solution, the screen should now look like this.



You will probably need to delete points in this window that are too scattered. If the point furthest away (in y) from the fit line is at y of more (in an absolute sense) than +/-0.5, type `d` to remove it. Now type `f` again. Continue to do this for the next farest away point until all points lie within +/-0.5.

Type "`q`" to exit the fit display and return to your spectrum - note how the x-axis has changed to Angstroms. You can type `f` to return to the fitting if need be. When you're happy with the fit, type "`q`" to exit and write the solution to the database by typing `yes` and/or pressing `enter`.

Make sure the cursor is once again writing in the terminal, now you can press the X on the window to make it disappear.

b. Applying the Wavelength Calibration

```
refspec.input = "Your science image"
refspec.answer = "YES"
refspec.references = "Your science image"
refspec.apertures = ""
refspec.refaps = ""
refspec.ignoreaps = yes
refspec.select = "interp"
refspec.sort = " "
refspec.group = " "
refspec.time = no
refspec.timewrap = 17.
refspec.override = yes
refspec.confirm = no
refspec.assign = yes
refspec.logfiles = "STDOUT,logfile"
refspec.verbose = no
refspec.mode = "al"
```

```
refspec

dispcor.input = "Your science image"
dispcor.output = "Your science image(without .fit)_wc"
dispcor.linearize = yes
dispcor.database = "database"
dispcor.table = ""
dispcor.w1 = INDEF
dispcor.w2 = INDEF
dispcor.dw = INDEF
dispcor.nw = INDEF
dispcor.log = no
dispcor.flux = yes
dispcor.blank = 0.
dispcor.samedisp = no
dispcor.global = no
dispcor.ignoreaps = no
dispcor.confirm = no
dispcor.listonly = no
dispcor.verbose = yes
dispcor.logfile = ""
dispcor.mode = "al"
dispcor
```

If you have a longer spectrum, you want to make sure the 5577A and 6300A sky lines are accurate and strong.
If you have a short spectrum (i.e. a super bright star) you will not see these lines, so this step is just to check you have a wavelength solution.
`w e` and `e` again to zoom, `w a` to return to zoomed out view

`splot` science_image_name(without .fit)`_wc.fits`
It will prompt you with:
Image line/aperture to plot (0:) (175): `100`

Your window should look like either a super fuzzy flat line or a fuzzy flat line with a few spikes, two of which will be at 5577 and 6300. If it doesn't look like this ASK FOR HELP.

Congratulations, your spectrum is now wavelength calibrated!

4. Extracting the spectra

Now we are ready to extract the spectrum! "Extracting" the spectra means to sum up the flux of the target object from those rows with object flux to create a 1-dimensional spectrum. Or in simpler terms make the 2-D spectrum a 1-D spectrum we can read. In the same step, we will estimate and subtract the background flux from the part of the slit that observed "empty" sky. The task apall does all of these at the same time.

Before you copy-paste the below, find a y value in your ds9 window where the science trace (the middle bright line) is especially bright and there are no arc lines above or below it. Note down that y value.

```
apall.input = "Your science image(without .fit)_wc"
apall.nfind = 1
apall.output = ""
apall.apertures = ""
apall.format = "onedspec"
apall.references = ""
apall.profiles = ""
apall.interactive = yes
apall.find = yes
apall.recenter = no
apall.resize = no
apall.edit = yes
apall.trace = yes
apall.fittrace = yes
apall.extract = yes
apall.extras = no
```

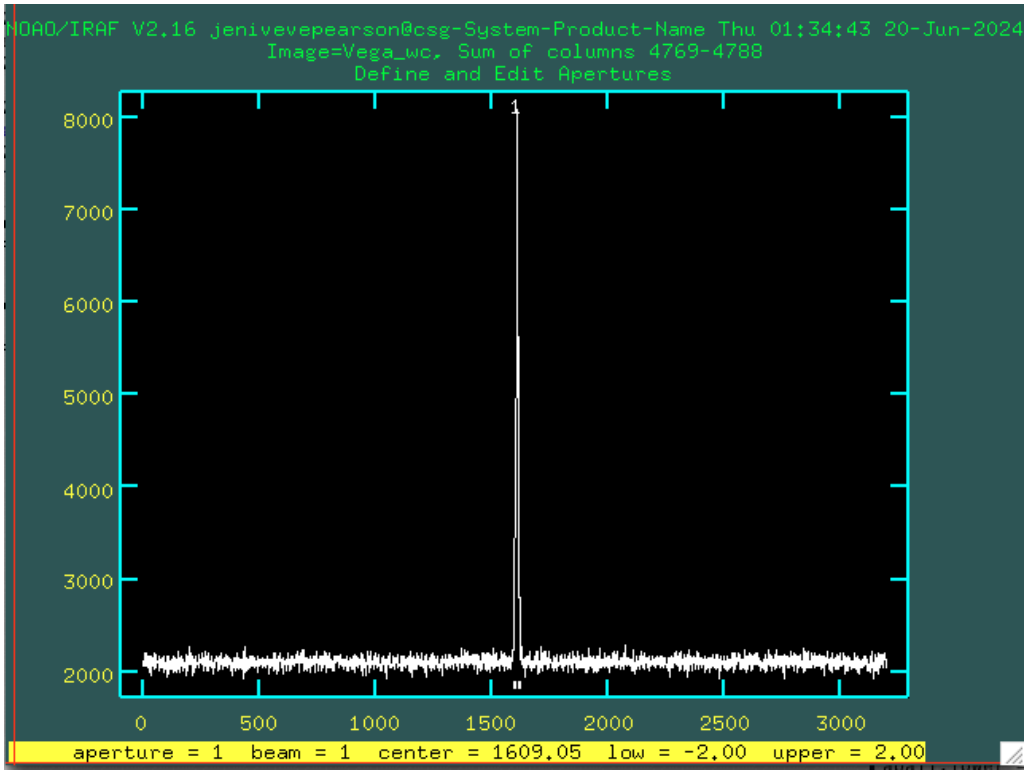
```
apall.review = yes
apall.line = Your bright y value
apall.nsum = 20
apall.lower = -2.
apall.upper = 2.
apall.apidtable = ""
apall.b_function = "chebyshev"
apall.b_order = 1
apall.b_sample = "-10:-6,6:10"
apall.b_naverage = -3
apall.b_niterate = 0
apall.b_low_reject = 3.
apall.b_high_rejec = 3.
apall.b_grow = 0.
apall.width = 5.
apall.radius = 10.
apall.threshold = 0.
apall.minsep = 5.
apall.maxsep = 100000.
apall.order = "increasing"
apall.aprecenter = ""
apall.npeaks = INDEF
apall.shift = yes
apall.llimit = INDEF
apall.ulimit = INDEF
apall.ylevel = 0.1
apall.peak = yes
apall.bkg = yes
apall.r_grow = 0.
apall.avglimits = no
apall.t_nsum = 10
apall.t_step = 10
apall.t_nlost = 3
apall.t_function = "legendre"
apall.t_order = 3
```

```
apall.t_sample = "*"
apall.t_naverage = 1
apall.t_niterate = 0
apall.t_low_reject = 3.
apall.t_high_rejec = 3.
apall.t_grow = 0.
apall.background = "fit"
apall.skybox = 1
apall.weights = "none"
apall.pfit = "fit1d"
apall.clean = no
apall.saturation = INDEF
apall.readnoise = "3.5"
apall.gain = "GAIN"
apall.lsigma = 4.
apall.usigma = 4.
apall.nsubaps = 1
apall.mode = "al"
apall
```

apall shows a view along a column, i.e. at a fixed wavelength - it shows the spatial information along the slit. In the example, we have specified to show the data centered at row 360; specifying a position is useful e.g. when viewing an emission-line object. Answer the following prompts with their defaults by just pressing `enter` or by typing `yes`:

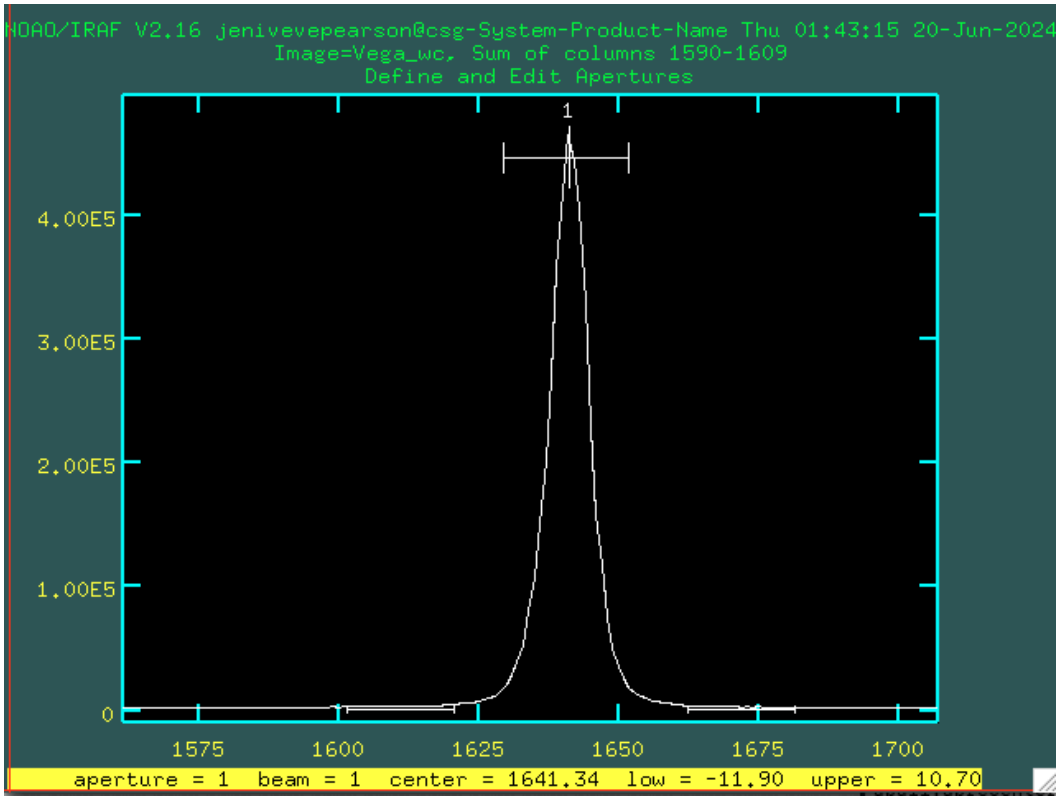
Find apertures for science 1.CF? ('yes'): `yes`
Number of apertures to be found automatically (1): `1`
Resize apertures for science 1.CF? ('yes'): `yes`
Edit apertures for science 1.CF? ('yes'): `yes`

You should now see a profile of the flux along the slit in the display window. Make sure you see the flux of the object; if not, choose a different central line and call over a counselor if that doesn't work. It should look like the figure below:



You should see an automatically placed aperture (ideally on the object), as well as an aperture to place the background. You will probably have to adjust both of these to capture most of the object flux, as well as to select a wide background region that is free of object flux. To adjust the object aperture, move the cursor to where you want to place the lower aperture limit and hit "`l`". Repeat for the upper limit with "`u`".

To place the background region, first hit "`b`". To delete the existing background region, hit "`z`". Hit "`s`" to place one side of the background region, and "`s`" again to place the other side. You should have 2 background regions, one on each side of your aperture (the big spike). Your window should now look something like this:



CHECK THIS WITH A COUNSELOR BEFORE MOVING ON!

Hit "`q`" when you're done to return to the object aperture extraction mode, and "`q`" again to finish the task. It will then ask you:

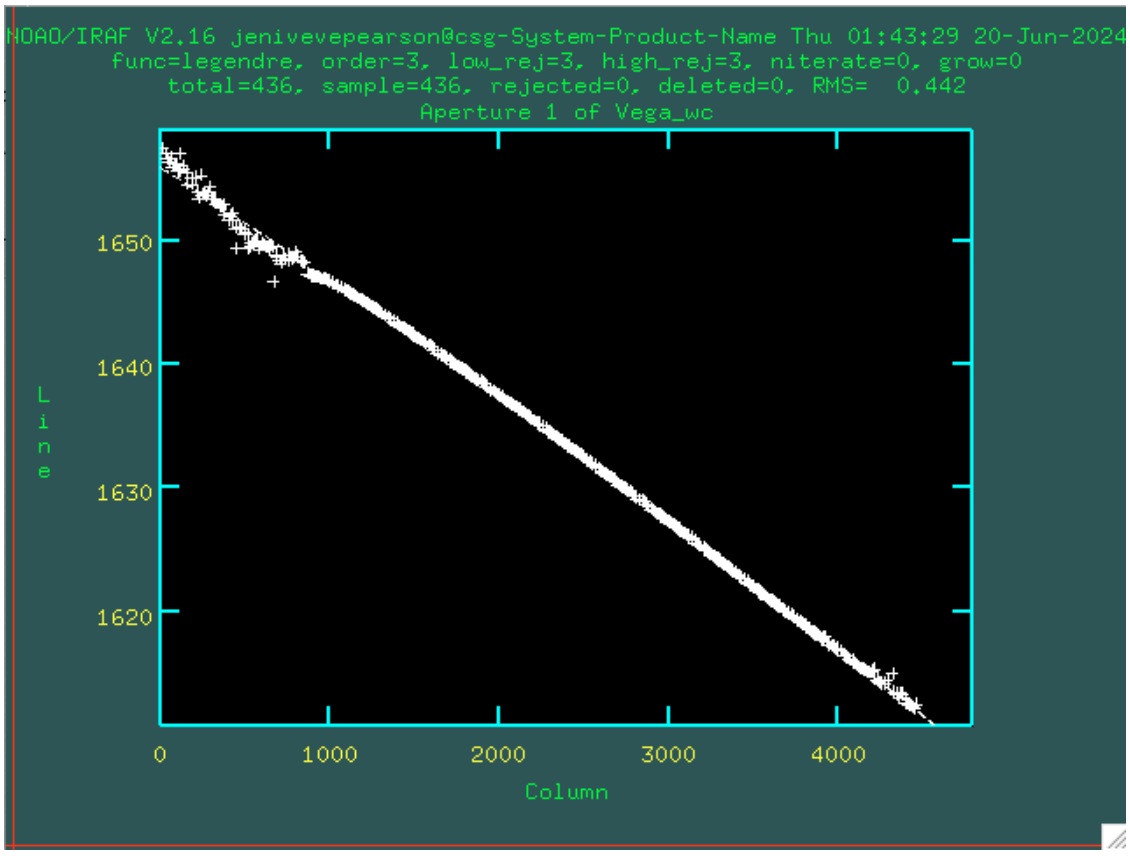
Trace apertures for science 1.CF?

We want to do this. Press `enter`.

Answer the following prompts with their default values or by pressing enter:

Write apertures for science 1.CF to database? `yes`
Extract aperture spectra for science 1.CF? `yes`
Review extracted spectra from science 1.CF? `yes`
Review extracted spectrum for aperture 1 from science 1.CF? `yes`

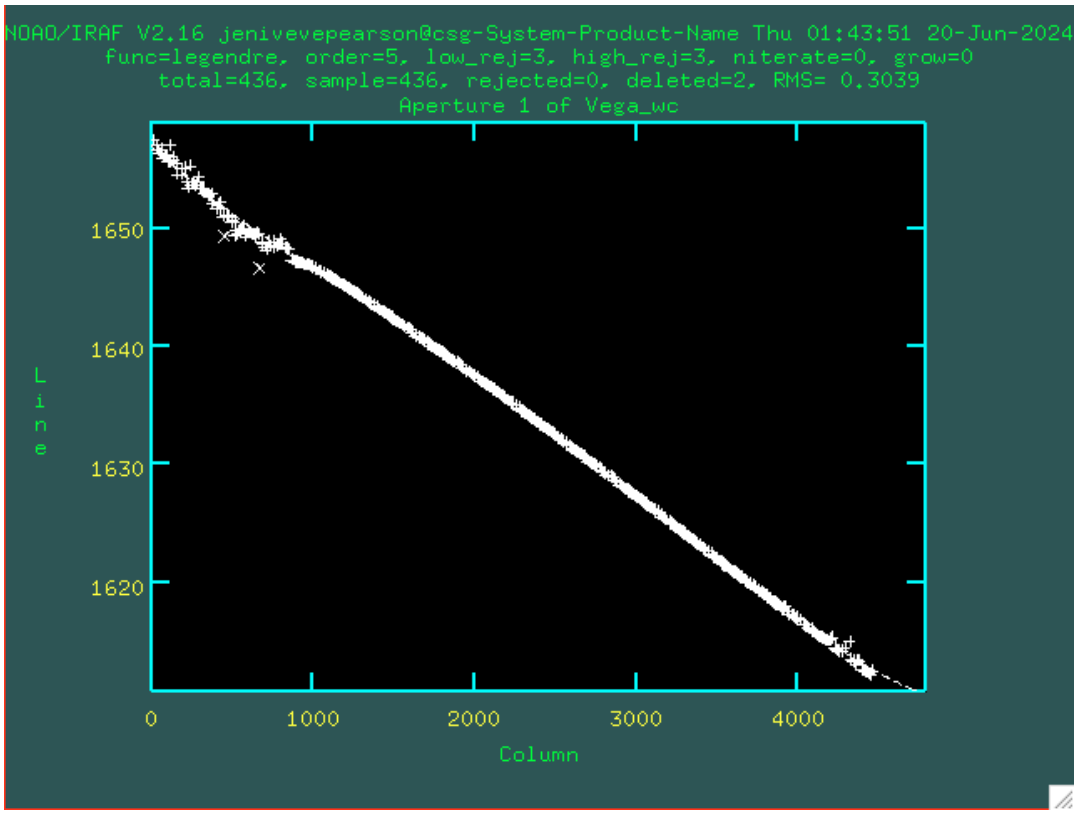
This should pop-up a window that looks like this:



Delete any points that do not follow the trend by a lot by typing `d` while hovering over the point. Do not delete all the scattered points, see below for an example of which points in the example above you might want to delete.

You might also want to change the order of the fit line (white dashed line). Do this by typing `:order` <number you want>, the order is currently 3. You want to avoid adding additional wiggles that aren't real, do not use an order >10 if you decided to mess with it.

You screen should now look something like the image below



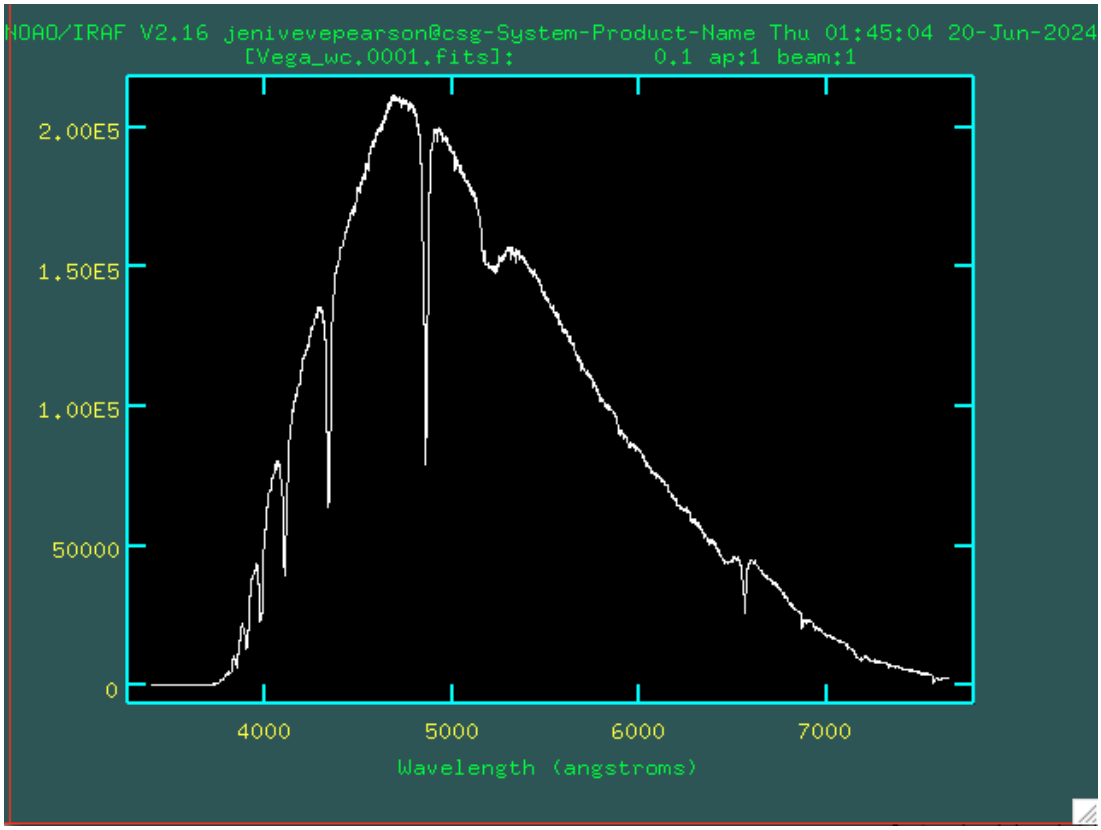
CHECK YOUR WINDOW WITH A COUNSELOR BEFORE PROCEEDING

Type `q` and `enter` on any prompts that come up. Your cursor should now be typing in the terminal. If it is not, type `q` again. Once your cursor is typing in the terminal press X on the upper corner of the interactive window to close it.

You have now successfully reduced the spectrum!!!

Type: `splot` Your_science_image(without .fit)`_wc.0001.fits` to take a look at your spectrum

If your target is a star the spectrum should look something like:



5. Combining multiple exposures

Sometimes when you have a fainter source, you will take multiple exposures and stack them so you can get higher signal.

A question to think about: Why can't you just take one really long exposure?

If you have multiple exposures you'll need to do the below steps to combine them all. Otherwise, you can proceed to the next section.

You'll want to make sure all your science images you want to combine are named something similar so you can add them all together easily. If they aren't, let a counselor know and we can help you.

If you have <5 images you will combine them using "`average`", if you have ≥5 images you will combine them using "`median`"

```
scombine.input = "Your_science_images(without
.fit)*_wc.0001.fits"
scombine.output = "Your_targets_name_wc.0001.fits"
scombine.logfile = "STDOUT"
scombine.apertures = ""
scombine.group = "apertures"
scombine.combine = "median"
scombine.reject = "avsigclip"
scombine.first = no
scombine.w1 = “”
scombine.w2 = “”
scombine.dw = INDEF
scombine.nw = INDEF
scombine.log = no
scombine.blank = 0.
scombine.scale = "none"
scombine.zero = "none"
scombine.weight = "none"
scombine.sample = ""
scombine.lthreshold = INDEF
scombine.hthreshold = INDEF
scombine.nlow = 1
scombine.nhigh = 1
scombine.nkeep = 1
scombine.mclip = yes
scombine.lsigma = 3.
scombine.hsigma = 3.
scombine.rdnoise = "3.5"
scombine.gain = "GAIN"
scombine.snoise = "0."
scombine.sigscale = 0.1
scombine.pclip = -0.5
scombine.grow = 2.
scombine.mode = "al"
scombine
```

You should now have one spectrum that is a combination of all your exposures!

Type: `splot Your_targets_name_wc.0001.fits` to take a look at your spectrum

6. Flux Calibration

Note: This step is difficult for *everyone*, even people who get paid to reduce spectra. The below copy-paste information is designed for using Vega as your standard star (if you are not using Vega just replace it with the name of your standard). If that is ***not*** the case, ask a counselor to help you edit the code for your standard.

To calibrate the sensitivity as a function of wavelength, we compare the observed spectrum of a star to its known reference spectrum. To do so, first extract the standard star spectrum and apply the appropriate wavelength calibration as above (**i.e. follow steps 3-5 for your standard star image files before continuing**).

Now that you have a reduced standard star spectrum, run the standard task:

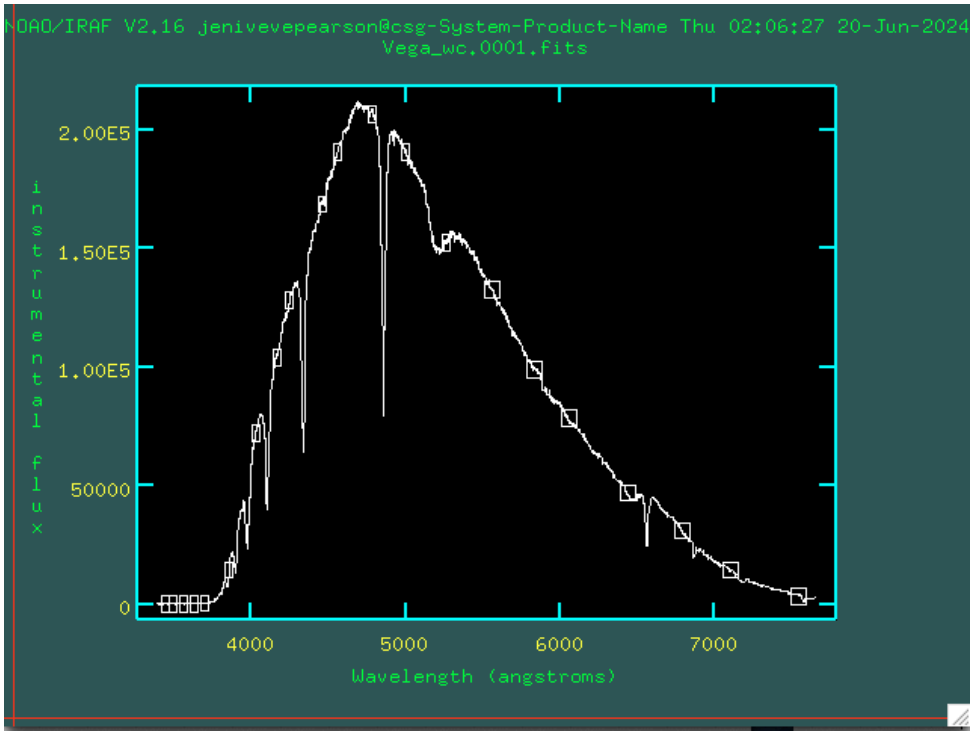
```
standard.input = "Vega_wc.0001.fits"
standard.output = "std_fake"
standard.star_name = "HR7001"
standard.answer = "yes"
standard.samestar = yes
standard.beam_switch = no
standard.apertures = ""
standard.bandwidth = INDEF
standard.bandsep = INDEF
standard.fnuzero = 3.6800000000000E-20
standard.interact = yes
standard.graphics = "stdgraph"
```

```
standard.cursor = ""
standard.caldir = "onedstds$bstdscal/"
standard.mode = "al"
standard
```

This may ask you a few questions.

If it asks you which telescope site type `mtbigelow`.
If it asks you what airmass, input what airmass you took your observation at (if you noted it down). This value should be between 1 and 2.5ish. If you don't know just use 1.

This will bring up a window like this:

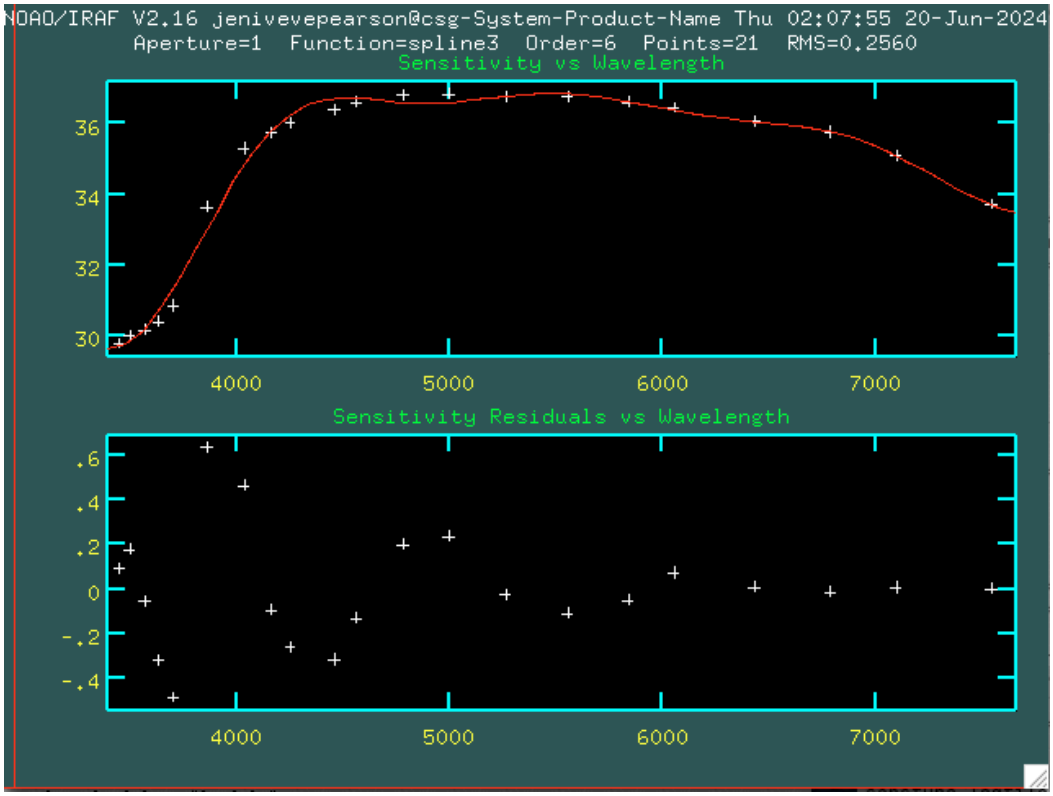


Make sure there are no boxes on the narrow lines (the sharp dips). If there are some boxes on lines, press `d` while hovering over the boxes in question. Press `q` once you are done.

```
sensfunc.standard = "std"
sensfunc.sensitivity = "sens"
sensfunc.answer = "yes"
sensfunc.apertures = ""
sensfunc.ignoreaps = yes
sensfunc.logfile = "logfile"
sensfunc.newextinctio = "extinct.dat"
sensfunc.function = "spline3"
sensfunc.order = 6
sensfunc.interactive = yes
sensfunc.graphs = "sr"
sensfunc.marks = "plus cross box"
sensfunc.colors = "2 1 3 4"
sensfunc.cursor = ""
sensfunc.device = "stdgraph"
sensfunc.mode = "al"
sensfunc
```

This is the hardest part, **do not proceed without having a counselor check your work.**

The window that popped-up should look like this:

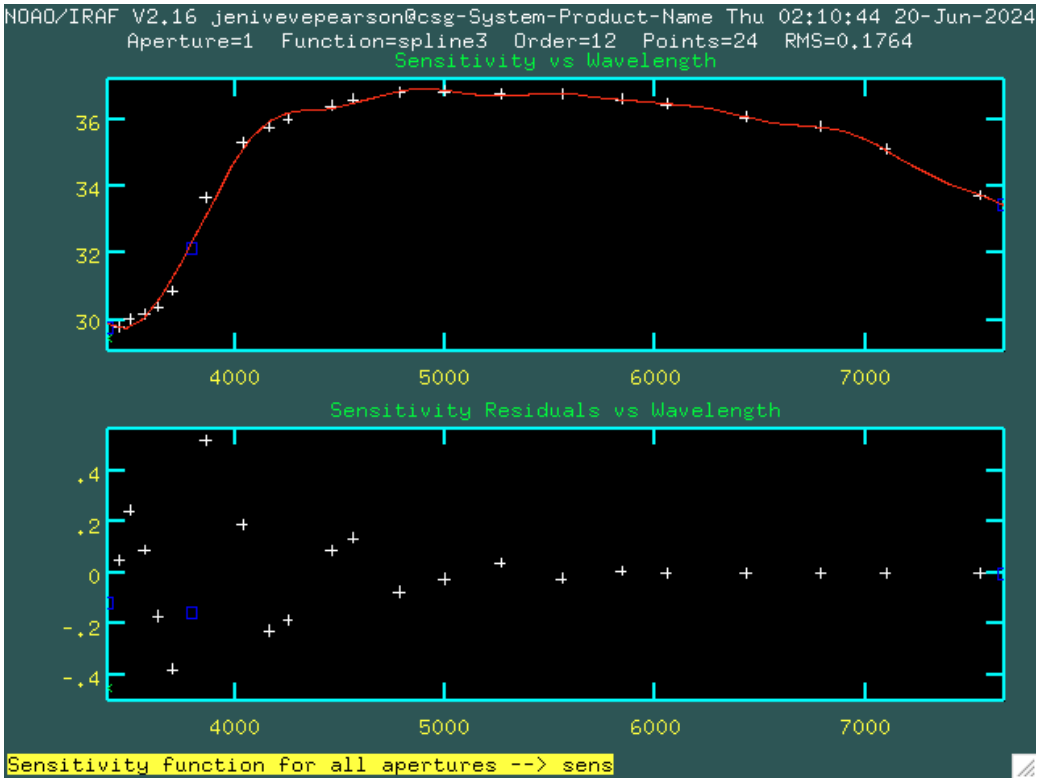


You want to make the bottom panel have no obvious wiggles. Do this by messing with the order and the function. Change the order by typing `:order` <number you want to use> and then click `enter` then `f` and `r`. Increase the number until the wiggles in the bottom panel disappear OR until the red line in the top panel starts to wiggle too much.

You can also try typing `:func le` then `enter` to change the function. Type `:func spline3` to return to the default fitting function if you don't like the other one.

You might need to add points at the beginning and end of the top curve to stop the red line from going crazy at the edges. To do this hover over the *exact* place you want your new point and type `a p` and then `f r`. If you mess up, type `d` while hovering over the messed up point.

When you finish, **CHECK YOUR WORK WITH A COUNSELOR**. It should look something like this:



Once a counselor has checked your work, type `q` and press `enter` to save your sensitivity function (the thing you just fit). Make sure your cursor can type in the terminal before pressing x to close the interactive window.

Now we can apply this solution to our science spectrum!

```
calibrate.input = "Vega_wc.0001.fits"
calibrate.output = "Whatever you want to call your final spectrum.fits"
calibrate.extinct = yes
calibrate.flux = yes
calibrate.extinction = "onedstds$kpnoextinct.dat"
calibrate.ignoreaps = yes
calibrate.sensitivity = "sens"
calibrate.fnu = no
calibrate.mode = "al"
calibrate
```

Congratulations, your spectrum is finally flux calibrated!
Now we just need to make sure it doesn't look crazy.

Google (or have someone else google) a spectrum of what your object should look like.

`splot your_final_spectrum.fits`

Compare your final spectrum side by side with the reference spectrum.
If there are a bunch of crazy wiggles in your spectrum that aren't in the reference, ask a counselor to look at it.

If they look similar enough, or a counselor tells you it is close enough, **you have a final, flux calibrated spectrum!!!**

Take a screenshot of the spectrum so you can show everyone how cool you are!

7. Exporting the spectra to ASCII files

You can use the task wspectext to output the 1D spectra to a ASCII file so you can plot it in python if you want:

```
wspectext your_final_spectrum.fits your_final_spectrum.txt
header=no
```

8. Measuring line fluxes (Optional)

The splot task, which we have used for plotting spectra before, is a convenient tool for measuring line fluxes. After bringing up a spectrum in splot, use the window keystrokes (`w y` to zoom in by a factor of 2 in y; `w x` to zoom in by a factor of 2 in x) to zoom in on a line / a set of lines. You can

also use `w e` with your cursor in the lower left corner of where you want to zoom and `e` again in the upper right corner of where you want to zoom to zoom around a particular line. Type `w a` to return to the zoomed out view.

To measure a single line, hit "`k`" on the left side of the line at the position where you want to set the background; hit "`k`" again on the right side (you might have to hit "`k`" three times to tell it that you want to fit a Gaussian). Splot will return the line center, line flux, and equivalent width. Try this a few times on the same line to estimate the uncertainty.

If lines are blended or nearby, you can use "`d`" to fit multiple lines at the same time. Again, hit "`d`" on the left side and on the right side of the area you are trying to fit. Then, mark the lines to be fit with "`g`". Hit "`q`" to indicate you are done with marking lines. Answer the following prompts. By default, you should try to fit all positions and widths, but keep the background fixed (it is determined by the first two "`d`" points). splot reports the measured fluxes one-by-one; use the "`+`" and "`−`" keys to move to the next line. Again, do this a few times to estimate the uncertainty.

Note that splot can smooth the spectrum (keystroke "`s`"), which may help bring out faint lines. Also note that to revert to the unsmoothed spectrum, you have to exit splot ("`q`").