Spectroscopy - Data Reduction with IRAF

12 May 2025

Student: Thara Rubi Caba Pineda

Instructor: Prof. Dragana Ilić

The aim of this tutorial is to perform the data reduction steps with IRAF to arrive at a fully reduced one dimensional spectra. I performed the steps with the Copernico dataset.

Calibration

With calibration the raw images are corrected for bias, trimmed to useful areas on the detector, and normalized to the flat field lamp exposures taken in context of the observations.

Bias correction

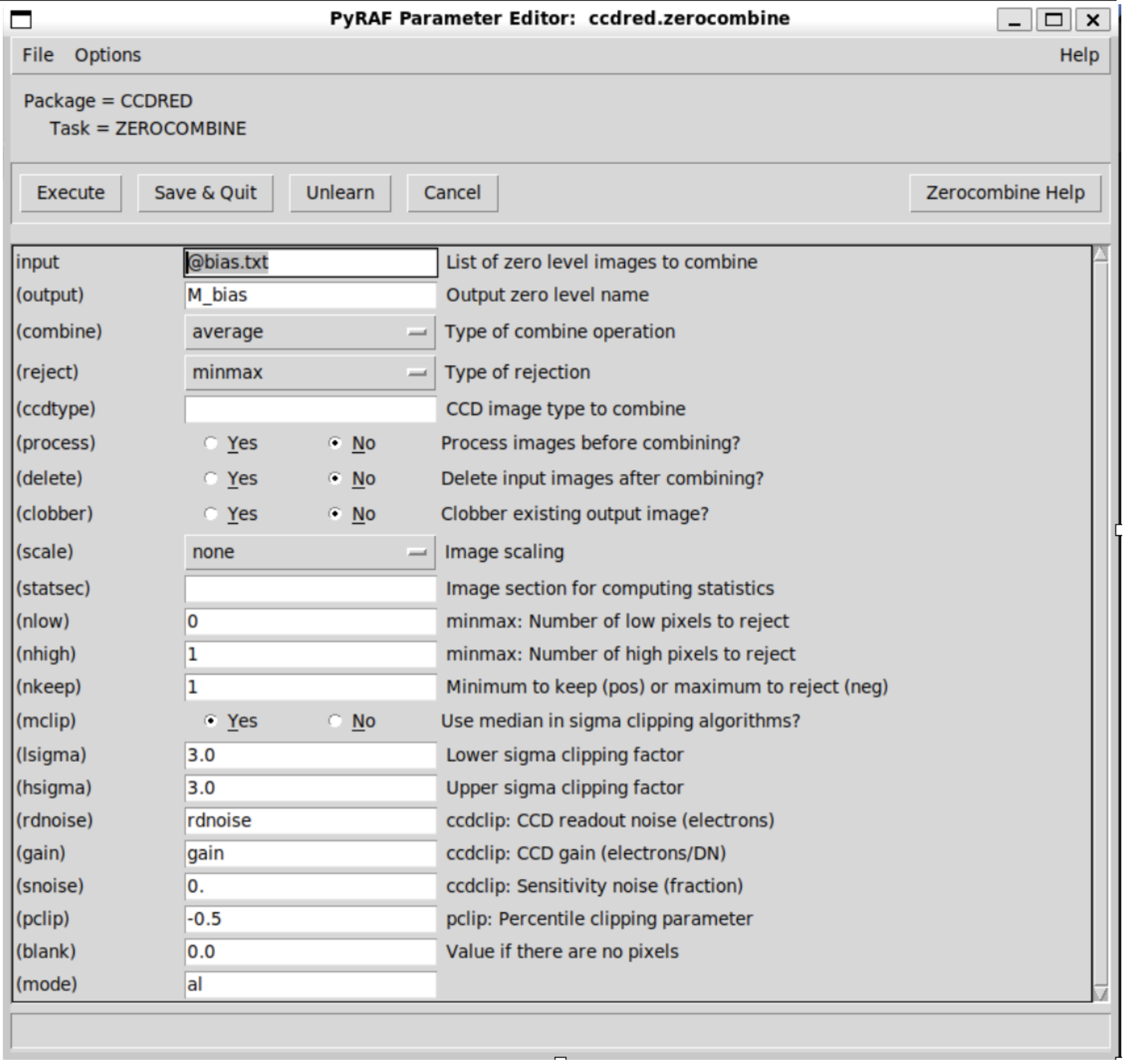
A bias is typically 0 s exposures with the shutters closed, which can be used to determine the bias level. We combine the bias images with the task:

```
noao
imred
ccdred
epar zerocombine
```

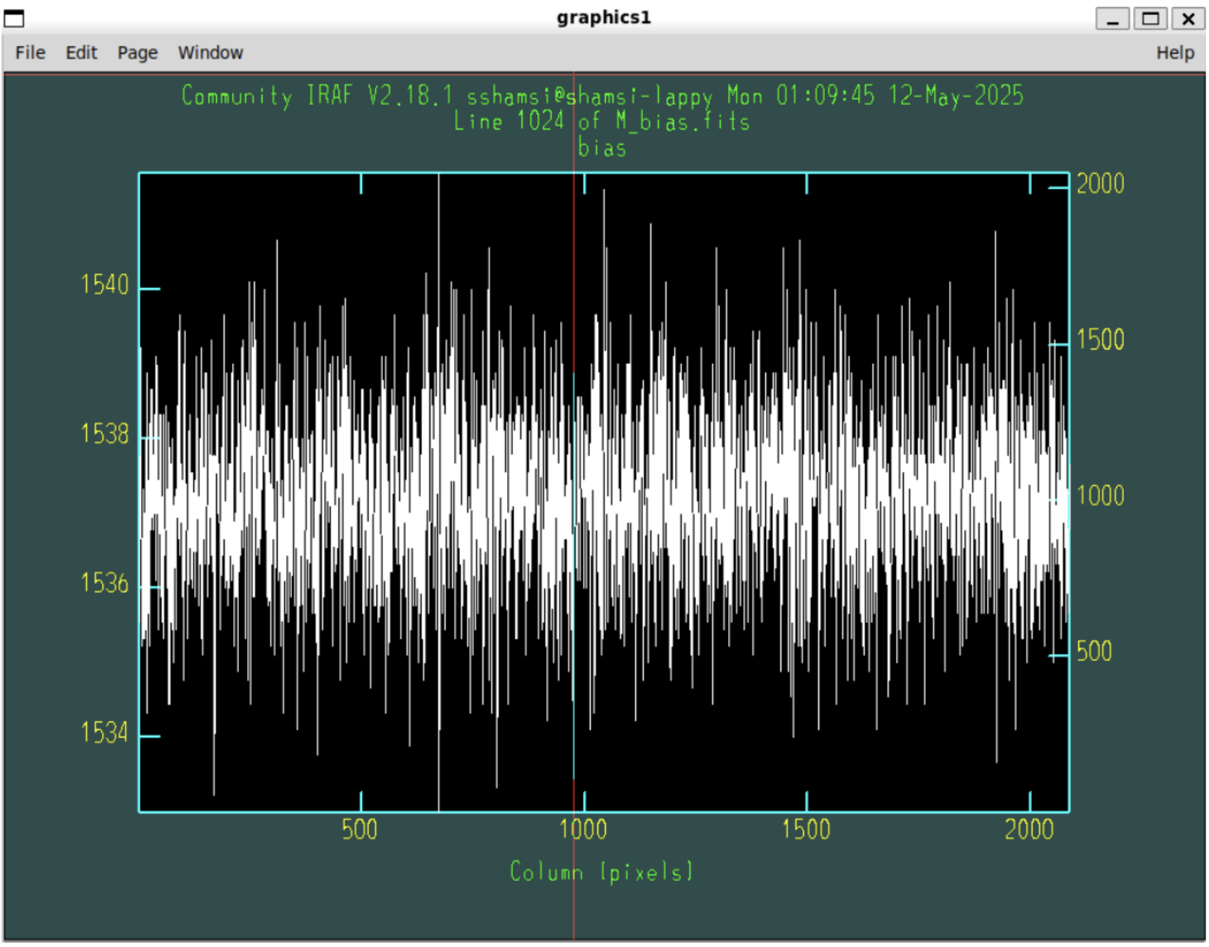
Fill in all parameters as in the tutorial:

```
input = @bias.txt
output = M_bias
ccdtype = EMPTY (remove default value 'zero')
```

```
rdnoise = rdnoise
gain = gain
```



Take a look at `M_bias` using `implot M_bias`:



We apply the bias correction to all images and trim the original images to get rid of the under- and/or overscan areas of the detector using this task:

```
noao
imred
ccdred
epar ccdproc
```

and fill in the parameters:

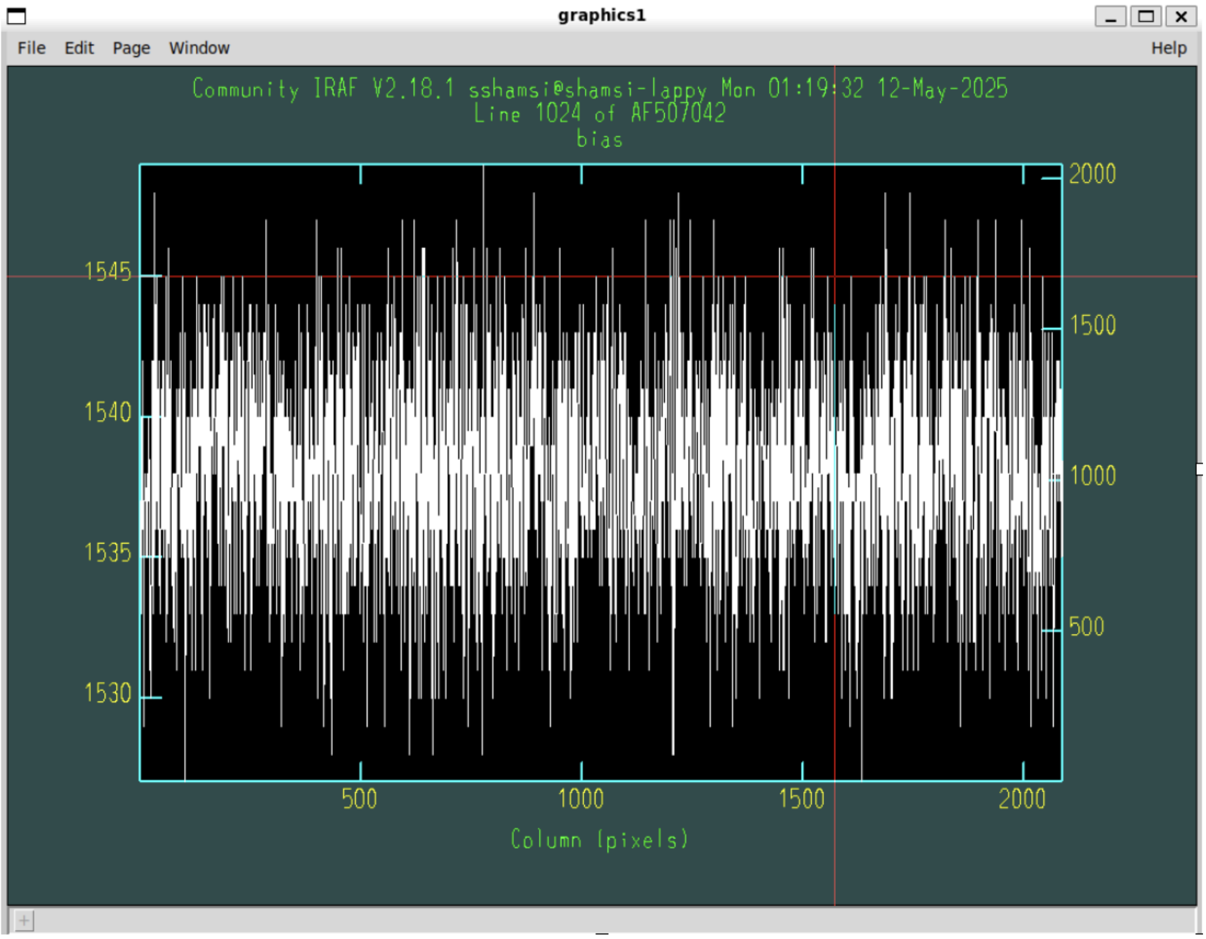
```
input = @all.txt
output = @all_bias.txt
ccdtype = EMPTY
fixpix = no
oversca = no
trim = yes
zerocor = yes
```

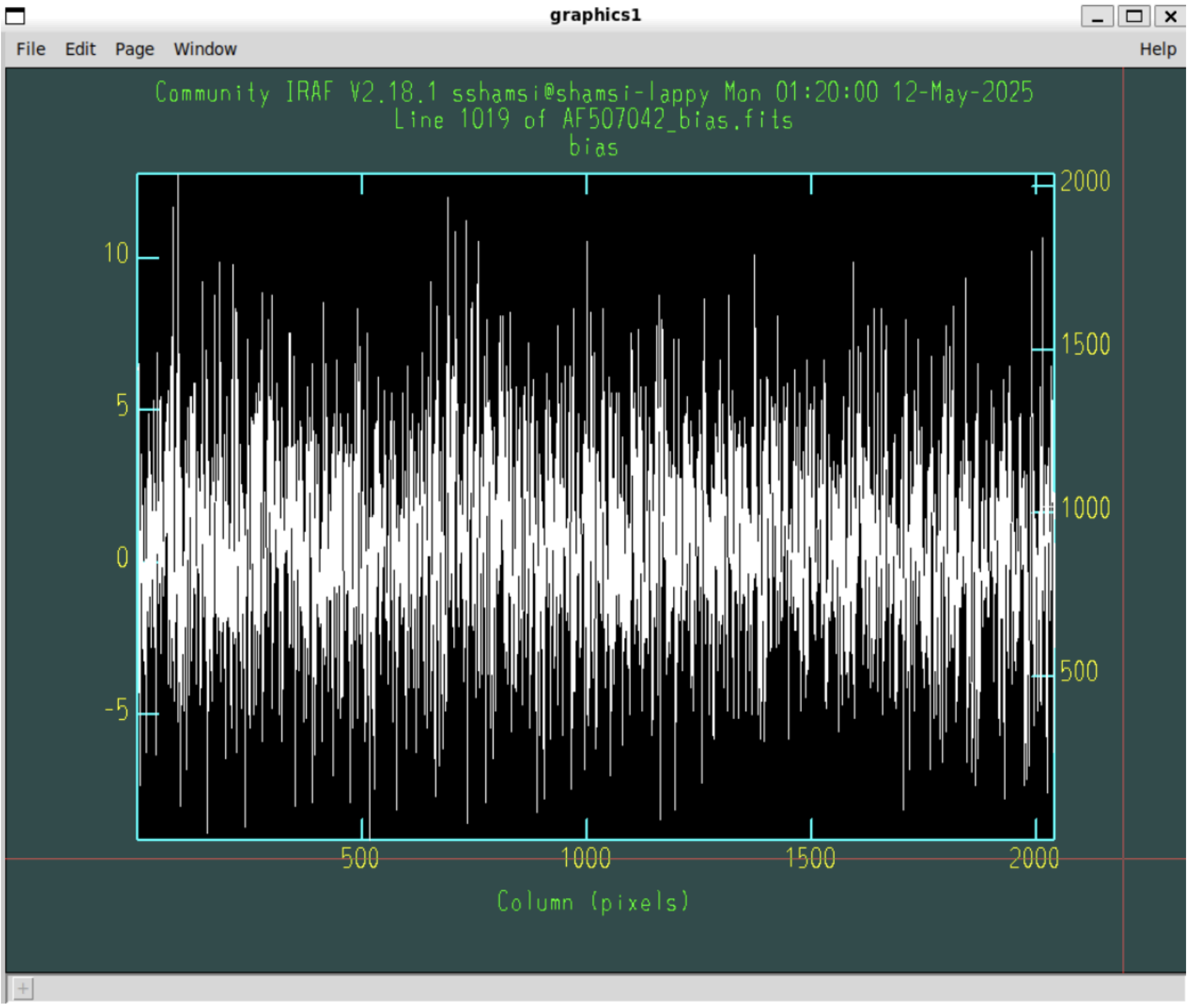
```
darkcor = no
flatcor = no
trimsec = [26:2065,7:2044] (values for Copernico data)
zero = M_bias
```

where `@all_bias.txt` is a list of the file names to assign to the bias-corrected output files, which we have to create manually.

Running the task creates the new bias-corrected *.fits* files named like `AF507055_bias.fits`

Let us plot an image before and after bias correction to visualise the effects:





Flat normalization

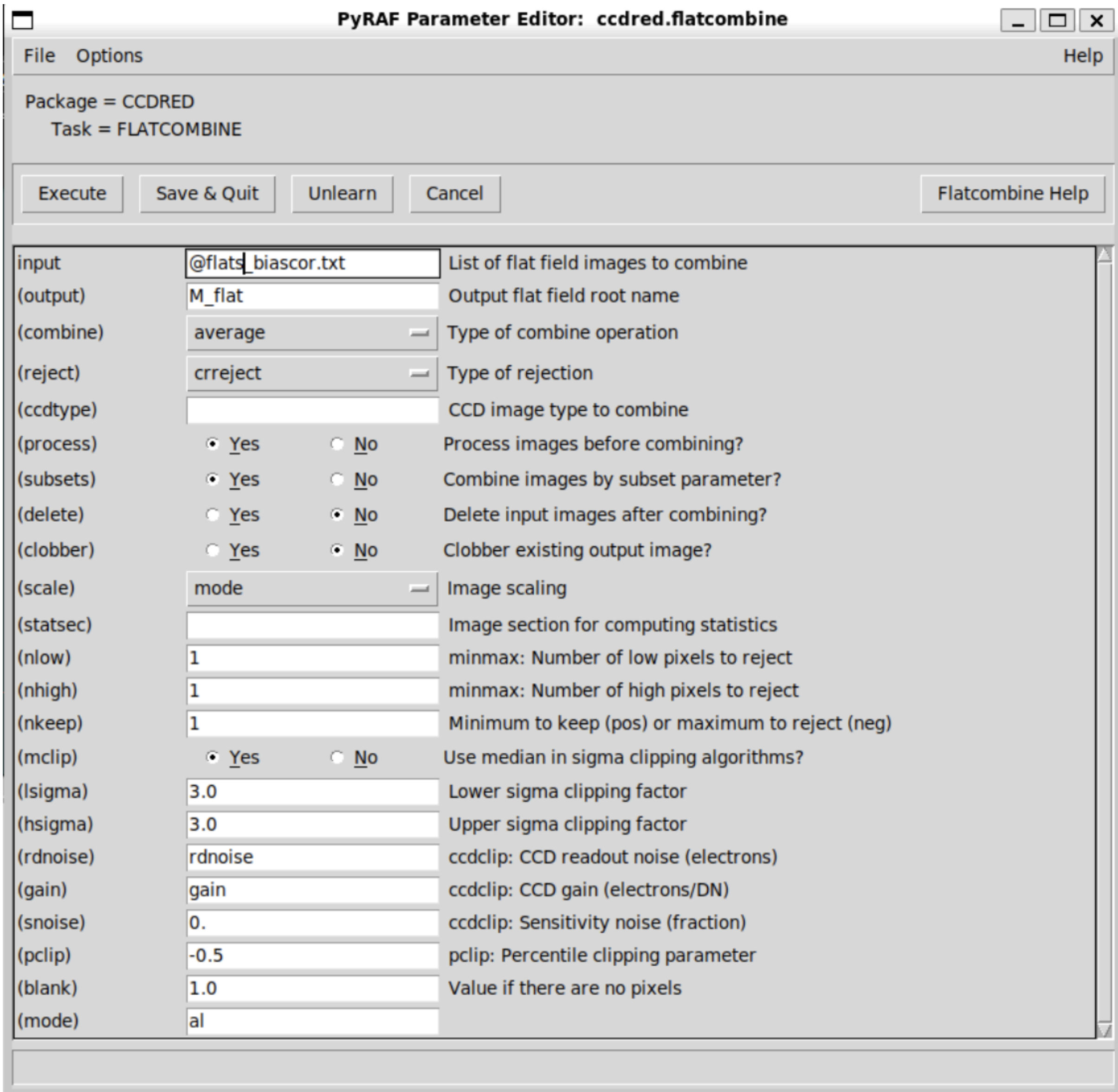
We will combine all flat fields taken into a master flat, and then use this to correct our images. We now manually create a text file `flats_biascor.txt` with the list of bias-corrected flats to combine. With this list, we can now run the task:

```
noao
imred
ccdred
epar flatcombine
```

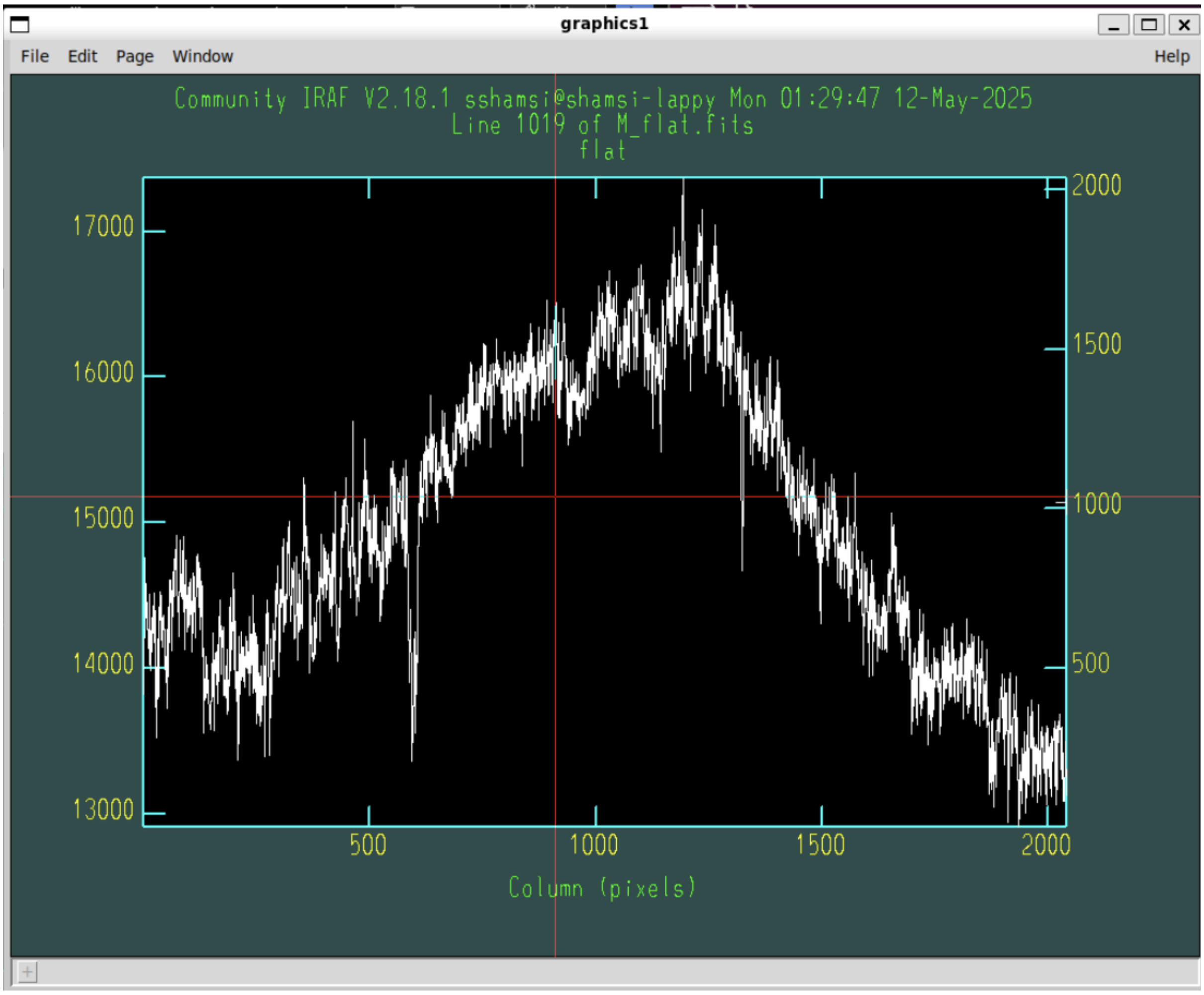
with the following parameters:

```
input = @flats_biascor.txt
output = M_flat
reject = crreject
ccdtype = EMPTY
```

```
rdnoise = rdnoise
gain = gain
```



Which will create the master flat `M_flat` file, which we can take a look at using `implot`:



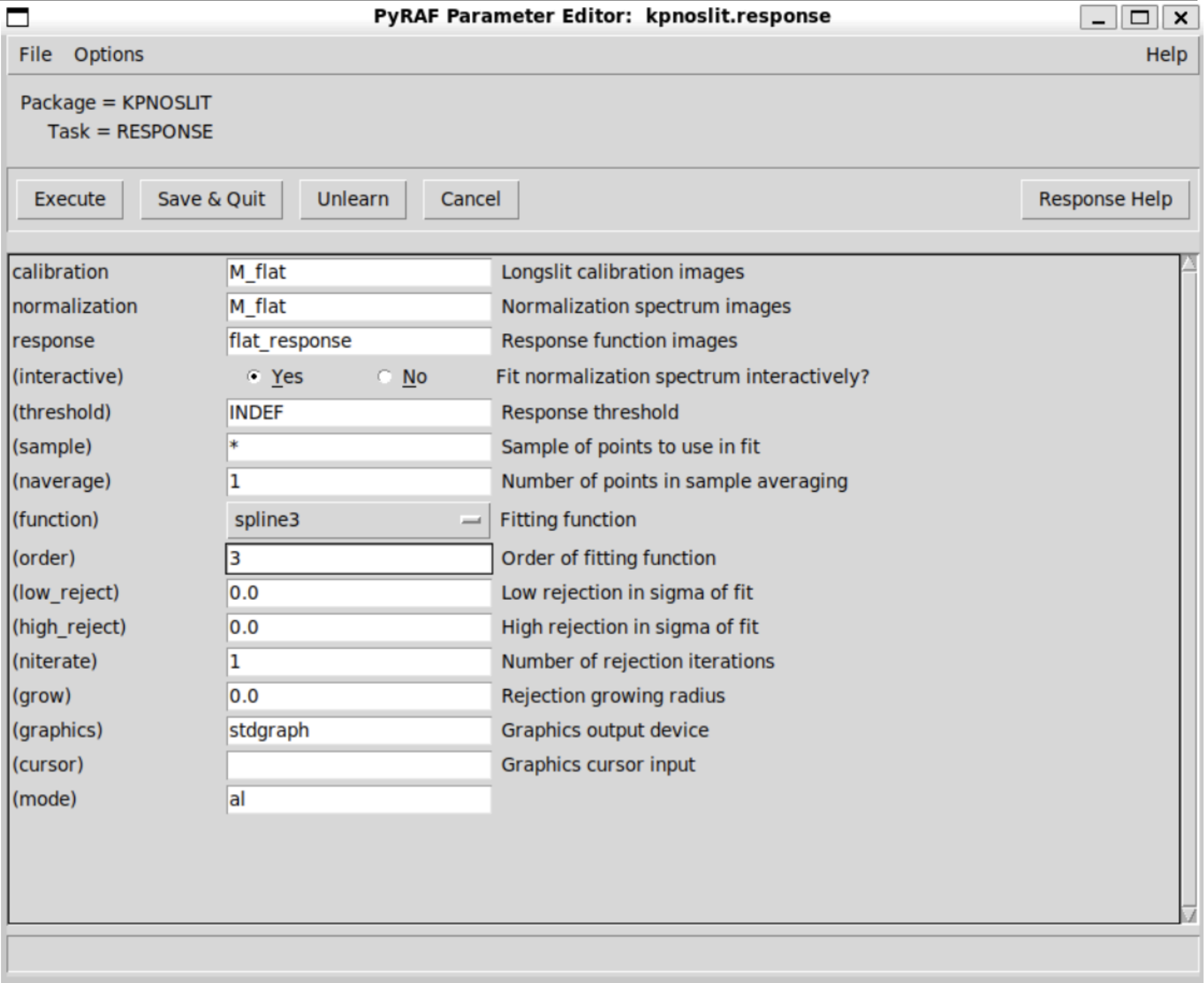
The created `M_flat` will be used to determine and isolate the pixel-to-pixel variations to be corrected for. For this, the flat is corrected across the spatial axis, so that a 1D regression can be performed to fit the flat lamp's spectral distribution, and only the high frequency variations are left over. We use the task:

```
noao
imred
kpnoslit
epar response
```

Then fill in the parameters as:

```
calibrat = M_flat
normaliz = M_flat
response = flat_response
interac = yes
functio = spline3
```

```
order = 3
```

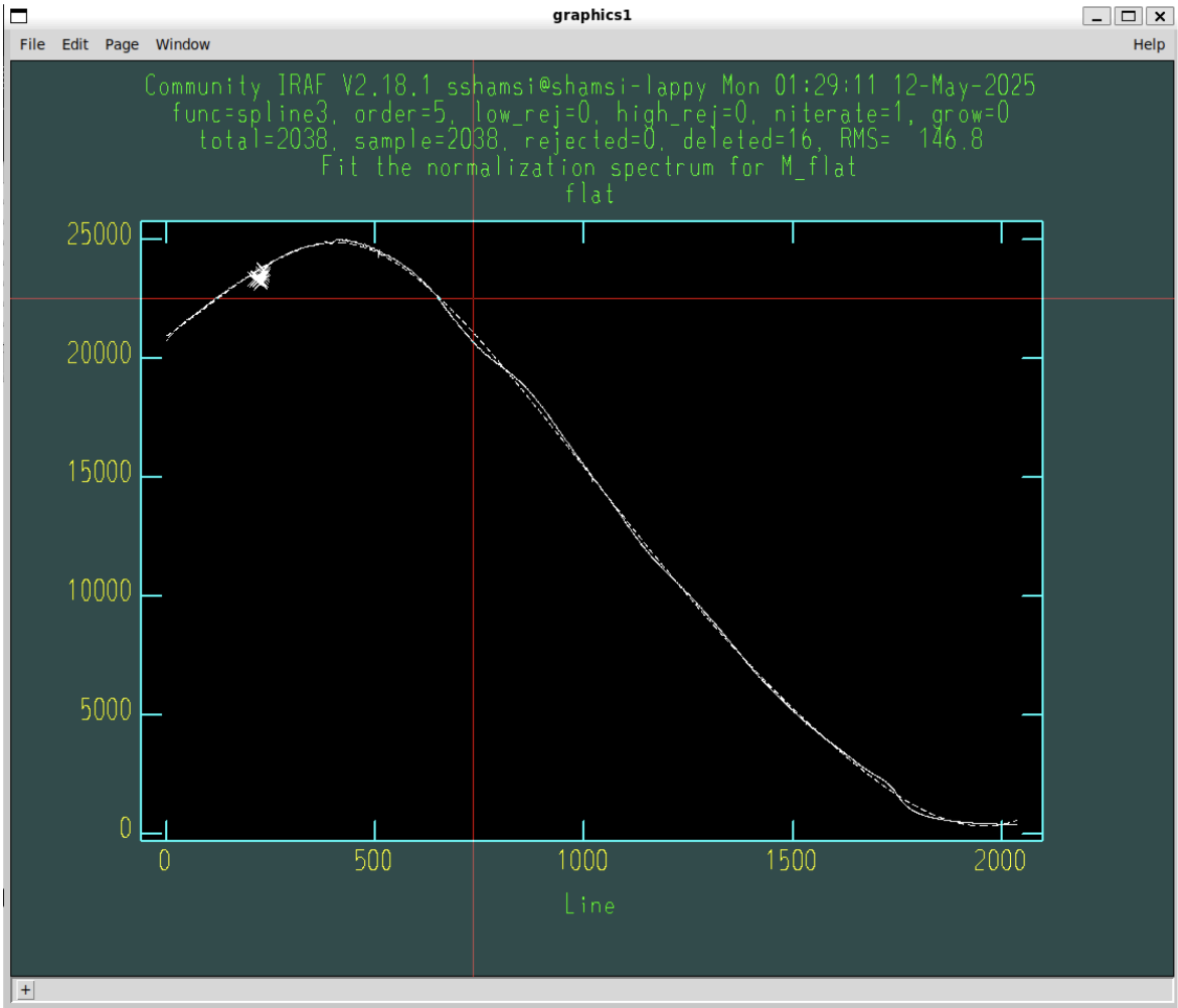


When interactive is yes, you will be asked to verify - type yes.

You will see a plot with collapsed flux on the y-axis vs. pixel dimension (position) on the x-axis. Dashed overplot depicts current fit, which you will fit to resemble the flat lamp spectrum. To modify fitting order, keep plotting window open, and type:

- :o 5 to change to order = 5 and press ENTER.
- press f to fit it again.
- **Outliers** = single data points with significant deviations, can be eliminated by hovering the cursor on top and pressing d for delete. (Deleted 1 points are marked with an X).
- Once satisfied with results, press q to quit task. If the fitting is sufficient, when the lower frequency variations are well resembled by the fitting curve.

The results from our fitting are:



The flat normalized variation map is now used to correct the science data for unwanted higher order variations. All features apparent in the flat field exposures, on top of the fitted lamp spectrum, will be corrected for.

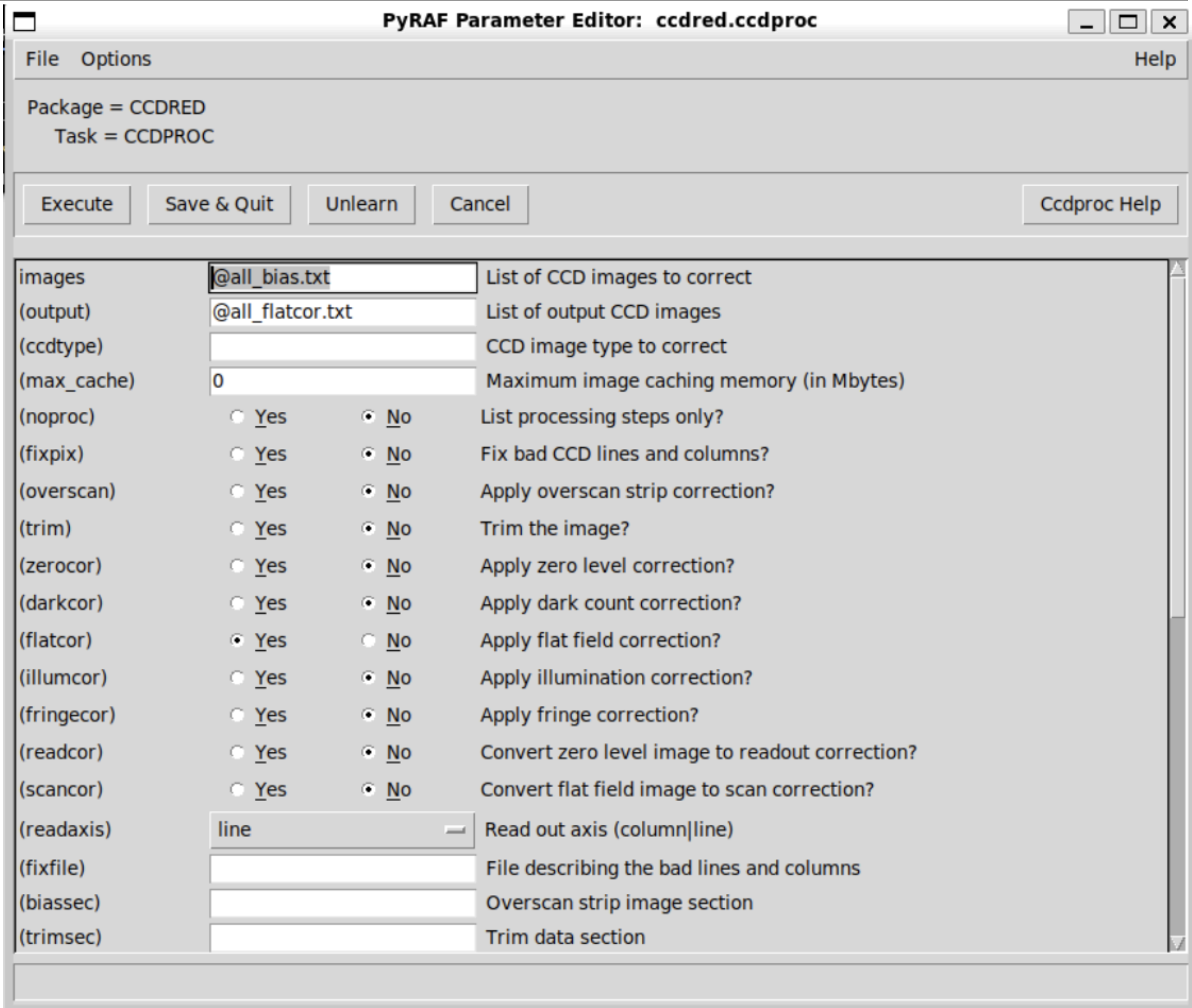
Use the task:

```
noao
imred
ccdred
epar ccdproc
```

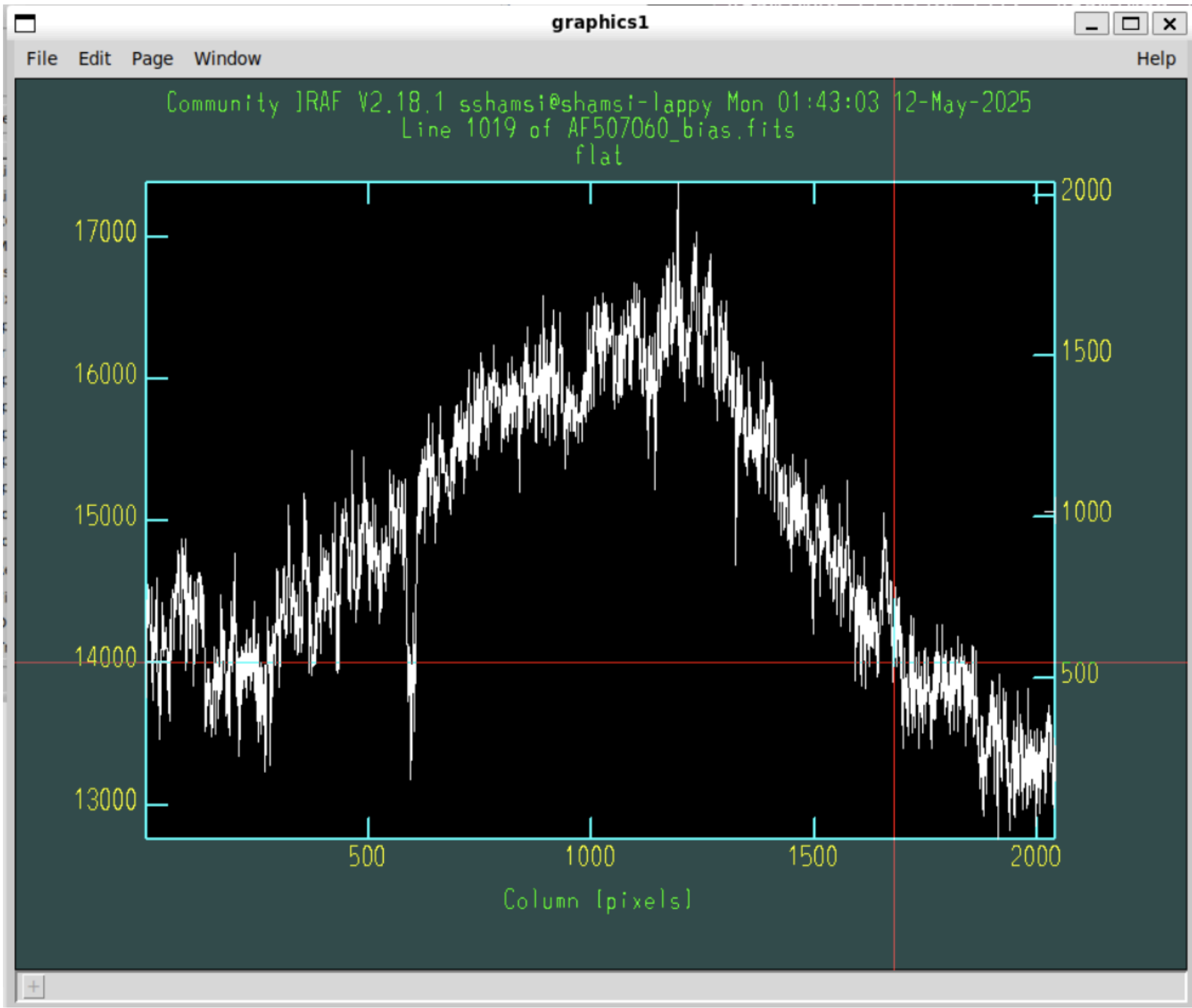
We manually create a list of output file names `all_flatcor.txt` containing all the filenames modified.

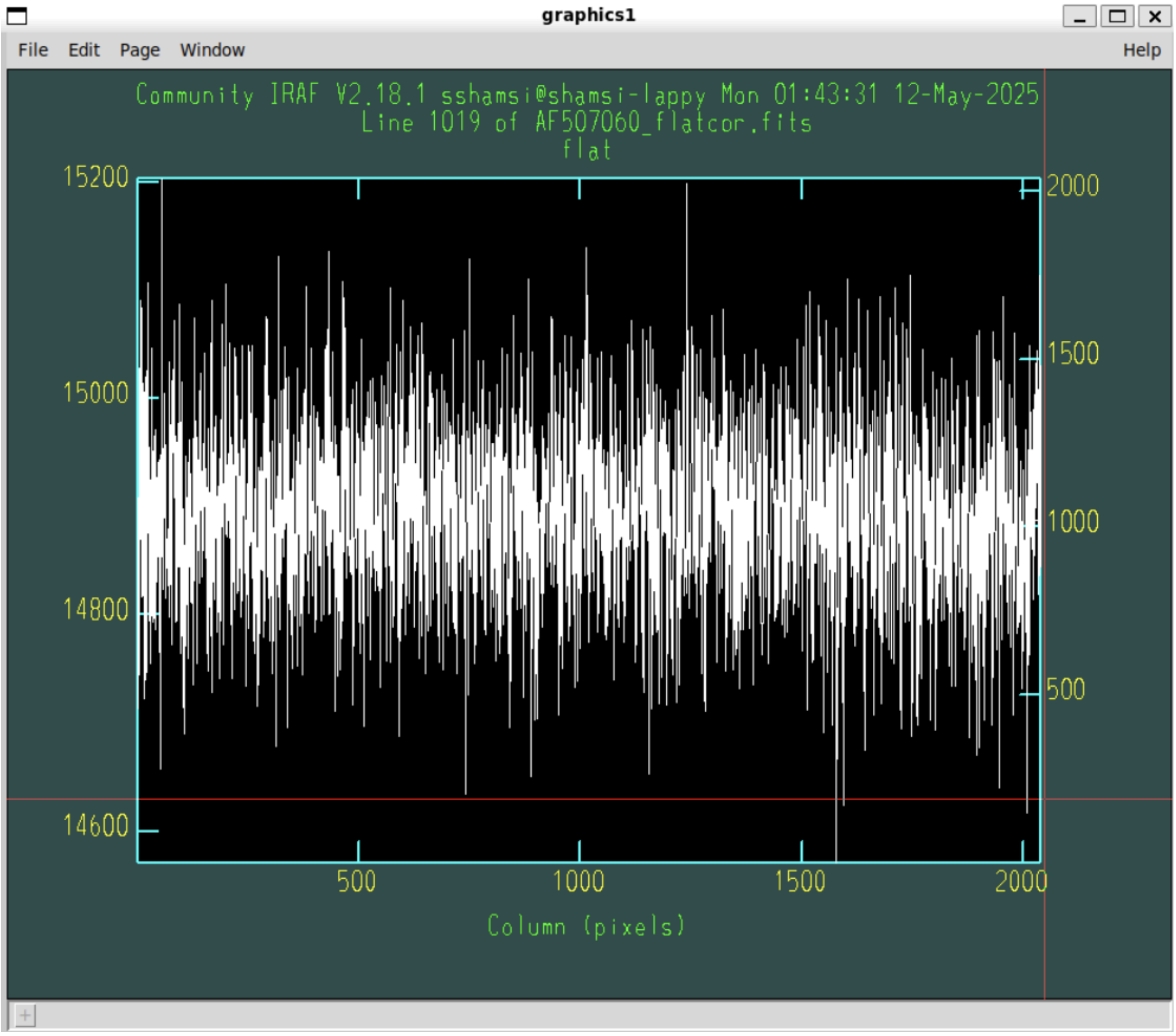
Now we can fill in and run the task with the following parameters:

```
input = @all.txt
output = @all_flatcor.txt
ccdtype = EMPTY
fixpix = no
oversca = no
trim = no
zerocor = no
darkcor = no
flatcor = yes
flat = flat_response (created in previous task)
```



Let us take a look at what this did to our images by plotting an image with only bias-correction, and another with bias and flat corrections.


graphics1
File Edit Page Window
Help
Community IRAF V2.18.1 sshamsi@shamsi-lappy Mon 01:43:03 12-May-2025
Line 1019 of AF507060_bias.fits
flat
17000
16000
15000
14000
13000
2000
1500
1000
500
500
1000
1500
2000
Column [pixels]



Spectrum extraction

In the following step the one dimensional spectra will be extracted from the two dimensional images. As most of the sources were observed more than once, the individual images will be combined (stacked) in the first step.

The flat normalized target exposures will be combined to one image. Weighting will be applied according to the exposure times inside the file headers.

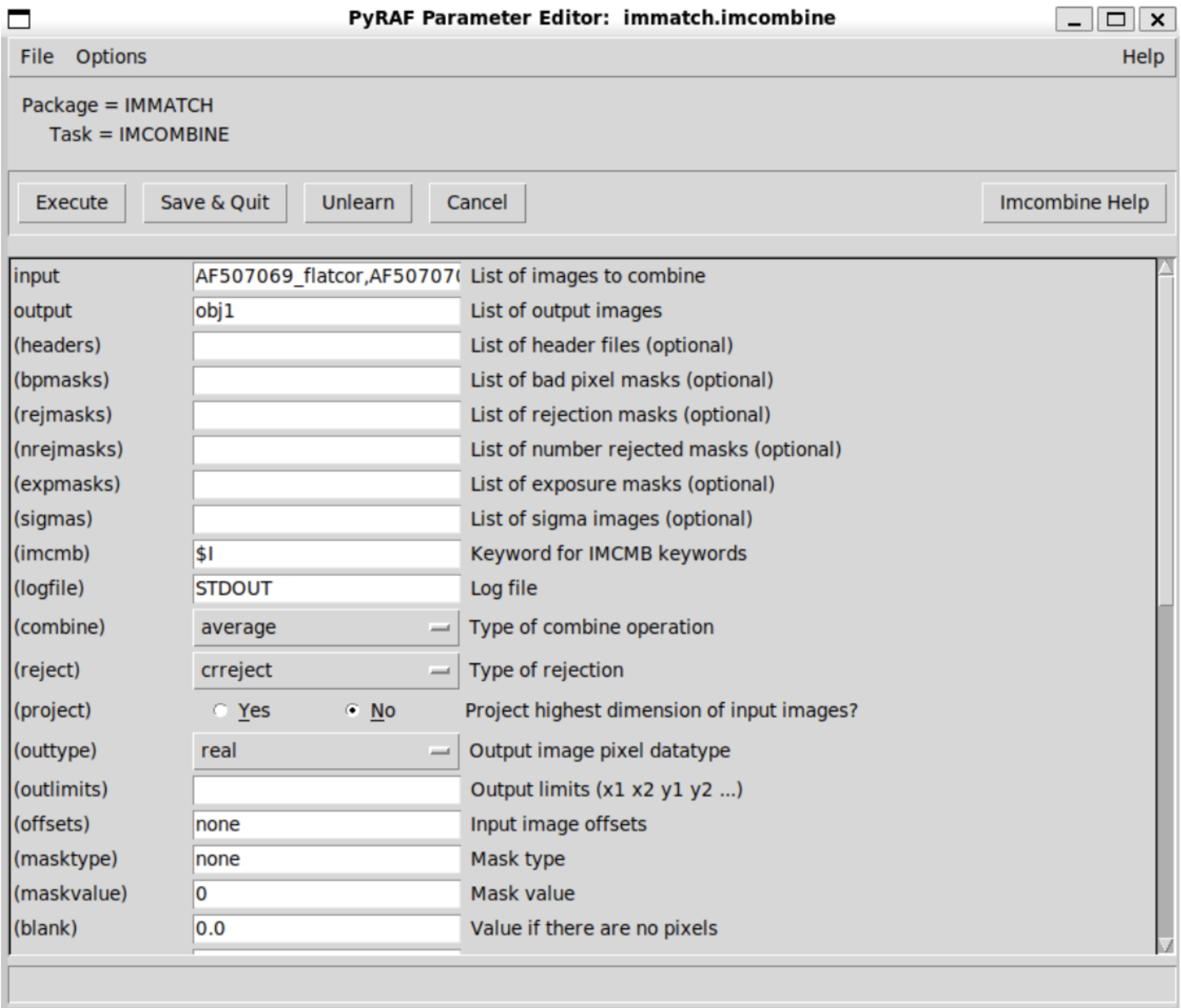
Use the task:

```
immatch
epar imcombine
```

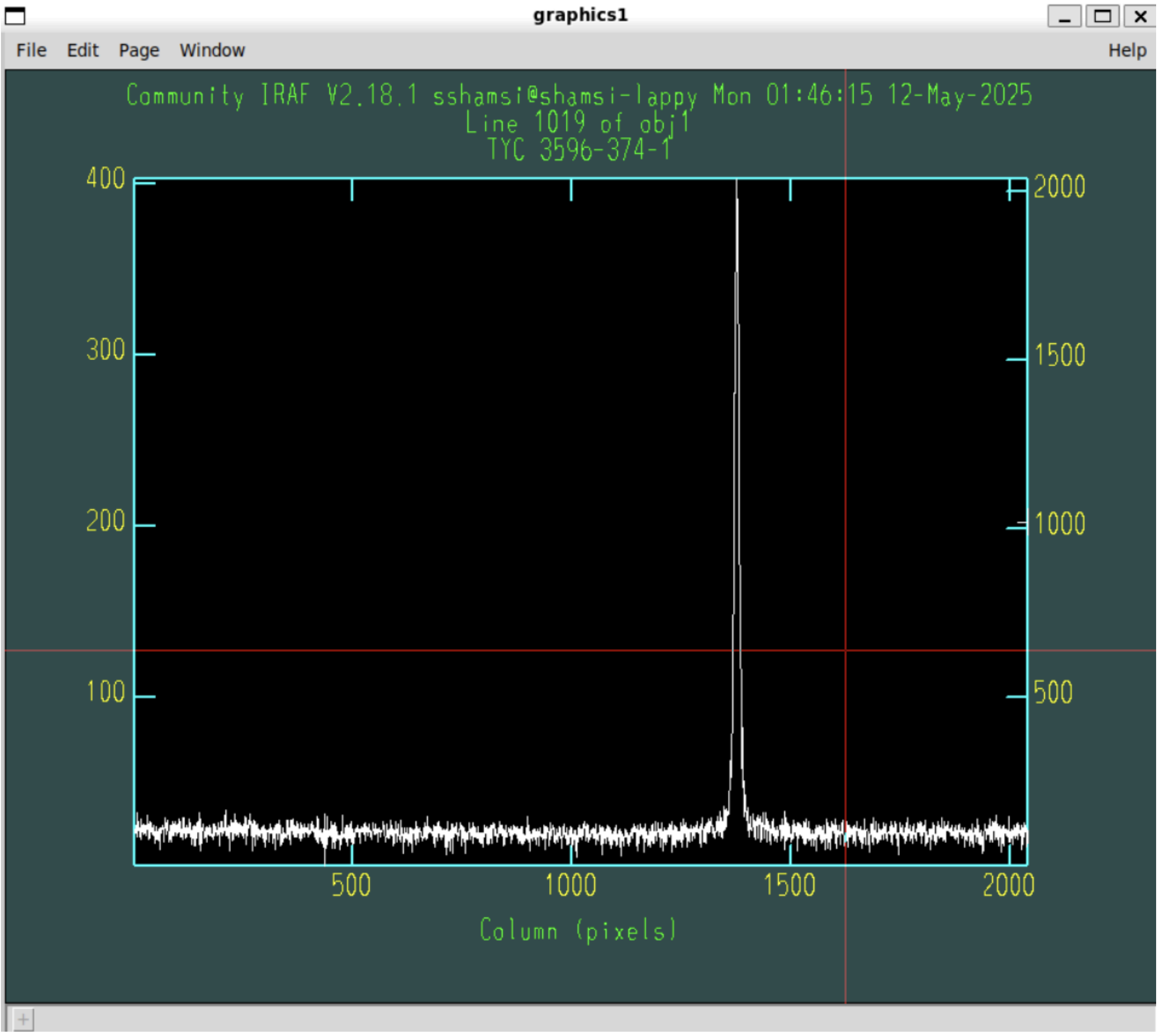
and set the input parameters:

```
input = @obj1_flatcor.txt
output = obj1
combine = average
reject = crreject
weight = exposure
rdnoise = 0
gain = 1
```

where `@obj1_flatcor.txt` is a list of file names of the observations relating to Object1, after bias and flat corrections.



Let us plot the output file `obj1.fits` to observe the effects:



In the raw images, the spectra follow a distinct trace along the dispersion direction. In this task, we search for this trace in the target image. First, we sum the pixel intensities along the dispersion axis to find the rough position of the spectrum. Then, we place the aperture on top of the spectrum and trace the spectrum, fitting by a smooth function which is used for the final extraction. During the extraction, the background is also modeled and subtracted from the flux. Therefore, the background level is sampled in a specific region around the aperture, defined by the parameter `b_sampl`. The background might be overestimated if the regions are chosen too close to the target, and would need to be modified.

We use the task:

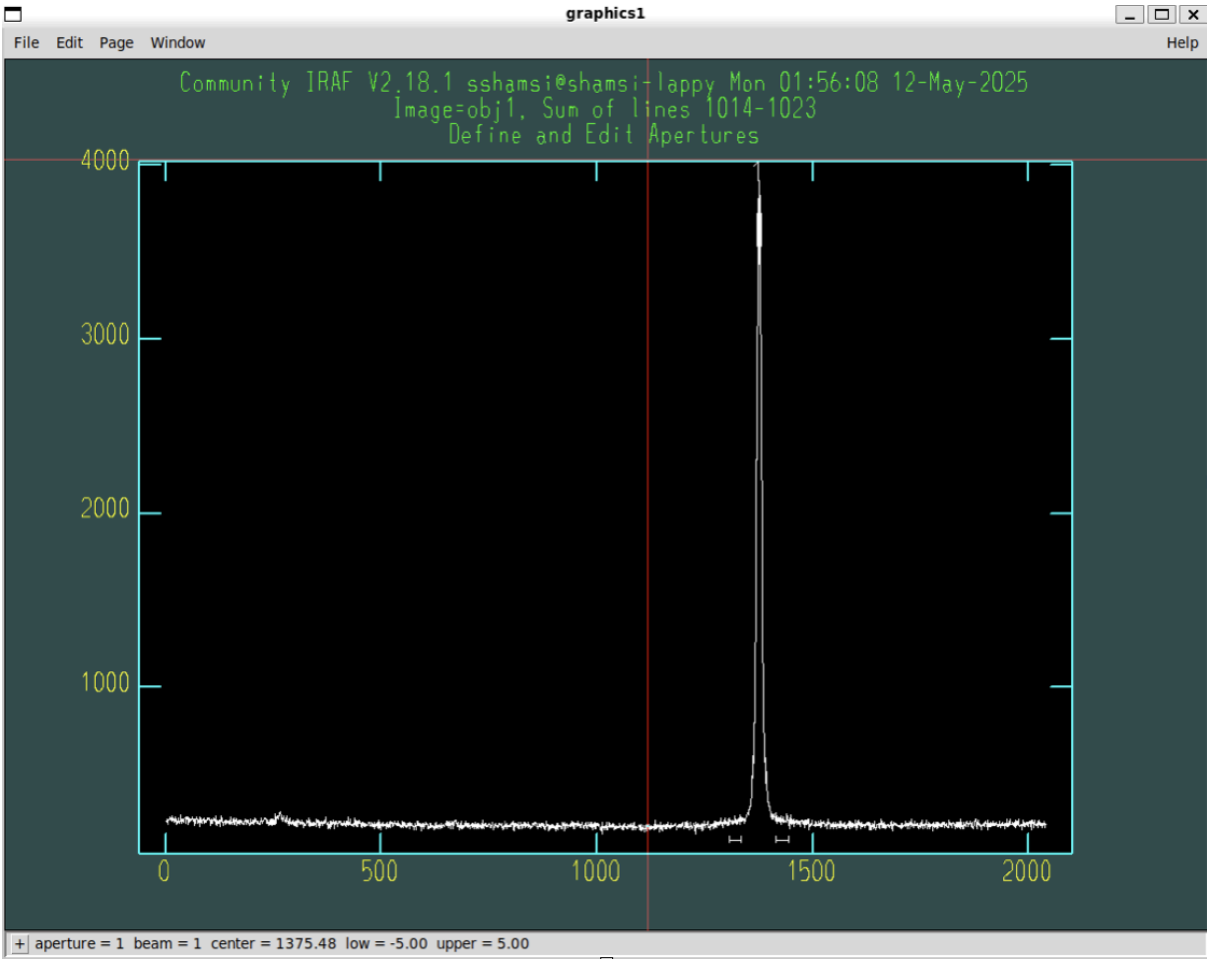
```
noao
imred
```

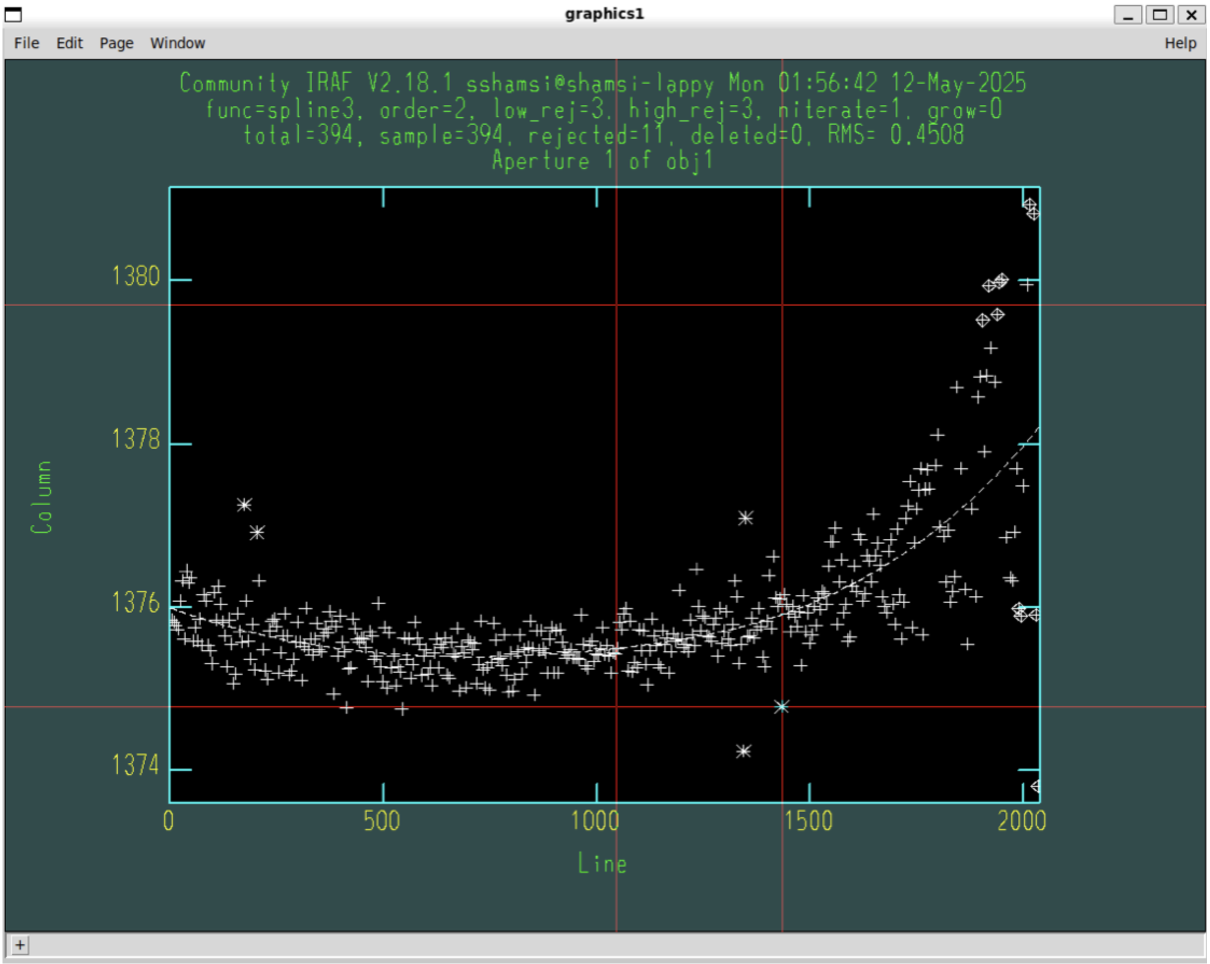
```
kpnoslit
epar apall
```

and submit the parameters:

```
input = obj1
output = obj1_spec
recente = yes
resize = no
edit = yes
trace = yes
b_funct = chebyshev
b_order = 2
b_sample = -70:-40,40:70
b_naver = -15
b_niter = 2
width = 5
radius = 5
nfind = 1
bkg = yes
t_nsum = 5
t_step = 5
t_funct = spline3
t_niter = 1
backgro = fit
weights = variance
clean = yes
```

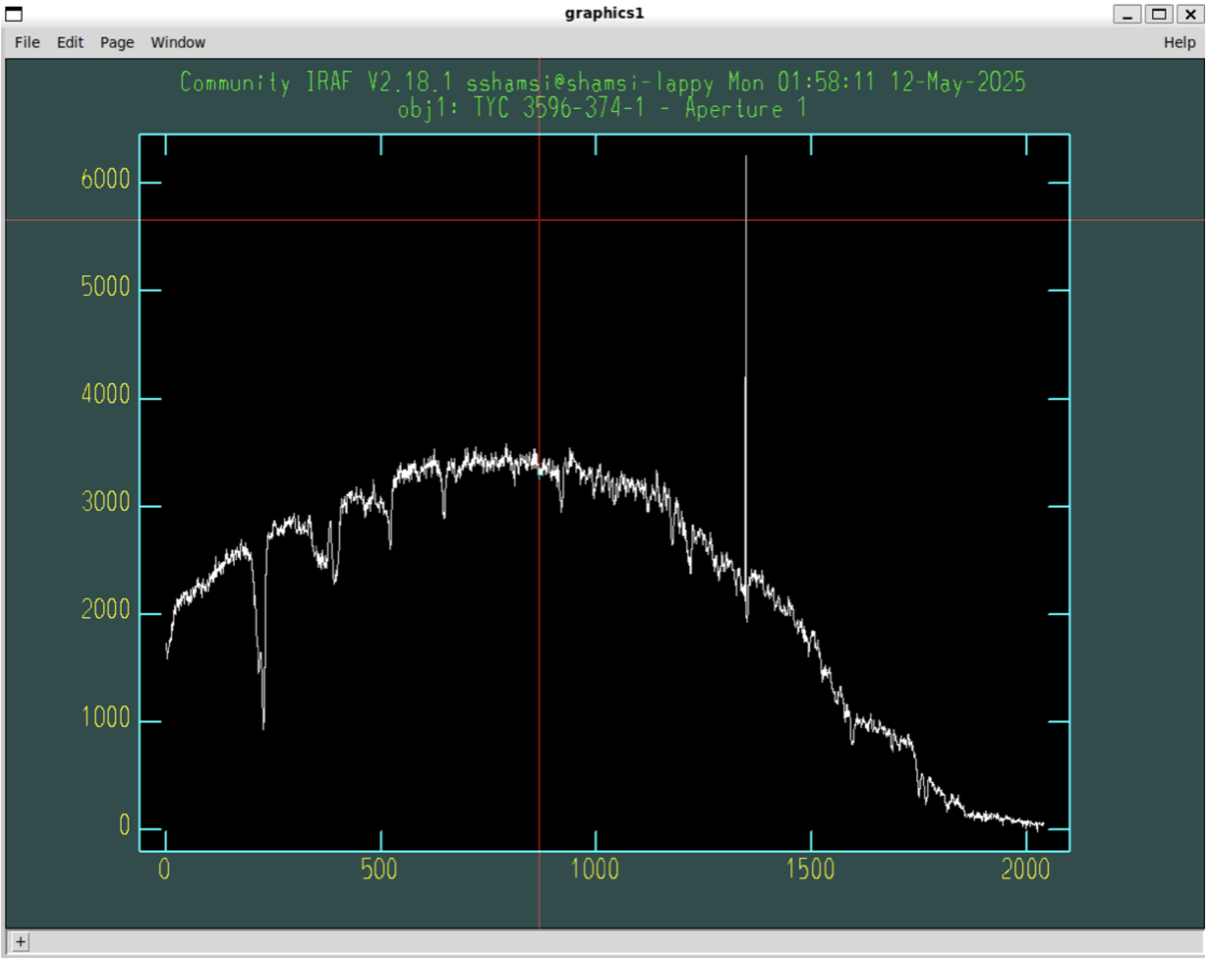
The fitting procedure looks like the following:


graphics1
File Edit Page Window
Help
Community IRAF V2.18.1 sshamsi@shamsi-lappy Mon 01:56:08 12-May-2025
Image=obj1, Sum of lines 1014-1023
Define and Edit Apertures
4000
3000
2000
1000
0
500
1000
1500
2000
aperture = 1 beam = 1 center = 1375.48 low = -5.00 upper = 5.00



We interactively fit by deleting outliers (d, now marked with an X) and refitting the function (f) as before.

and our final extracted spectrum looks like this:



Wavelength Solution

While the target spectra are already extracted, if you plot them, you will see that the flux is plotted against pixels along the dispersion direction. This is the case, as there is no wavelength information available in the data. In the process of the following wavelength calibration, a mapping from pixel to wavelength space is created, which is often also called wavelength solution. To do this, one generally needs some spectra with lines at well known positions. For this reason, emission line spectra of arc lamps have been taken during the observations that in the following will be used.

Note that for the Copernico telescope, the relevant lamps are Ne and HgCd.

We use the command:

```
noao
```

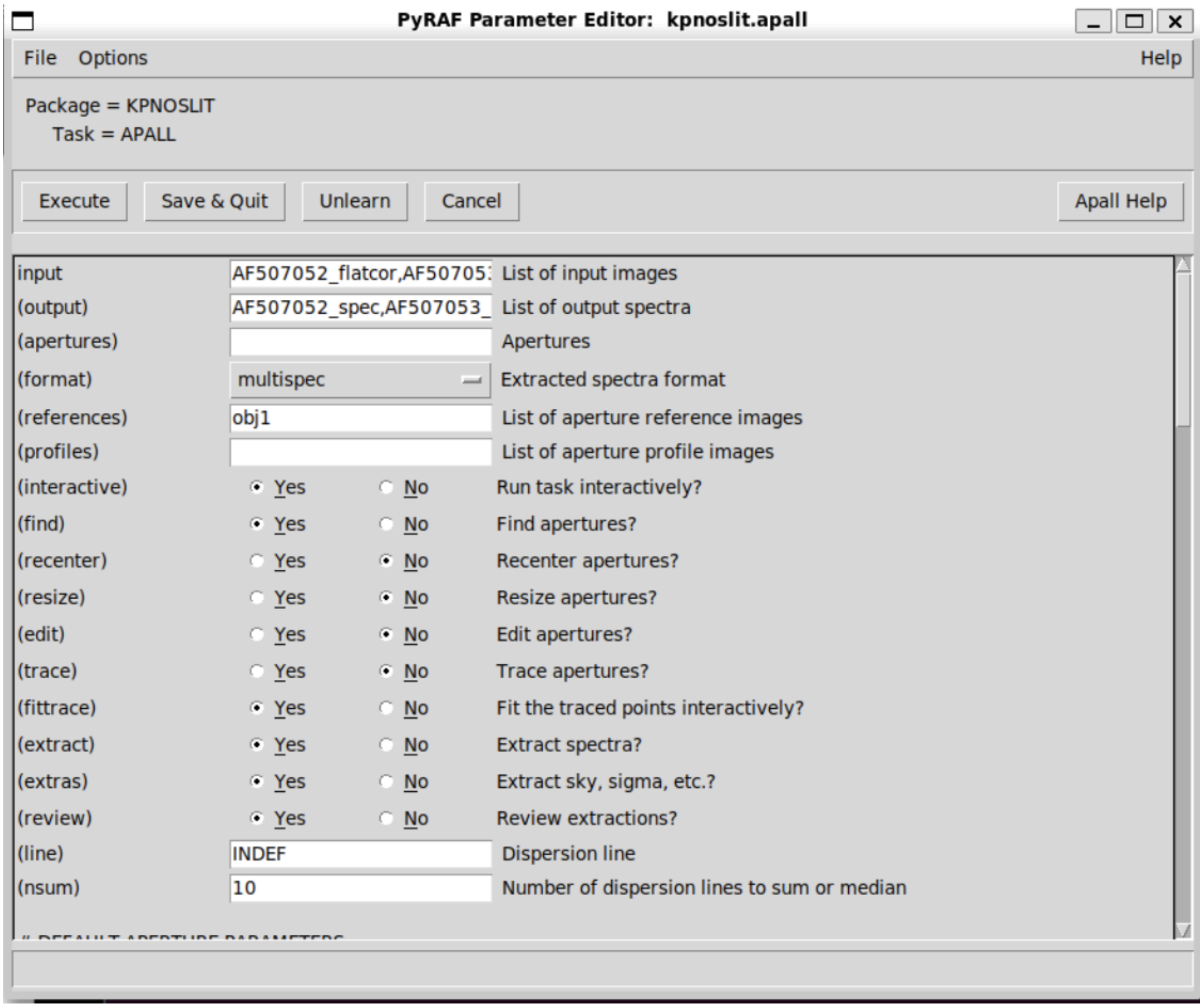
```
imred
kpnoslit
epar apall
```

First, let's perform the task for the Ne lamp.

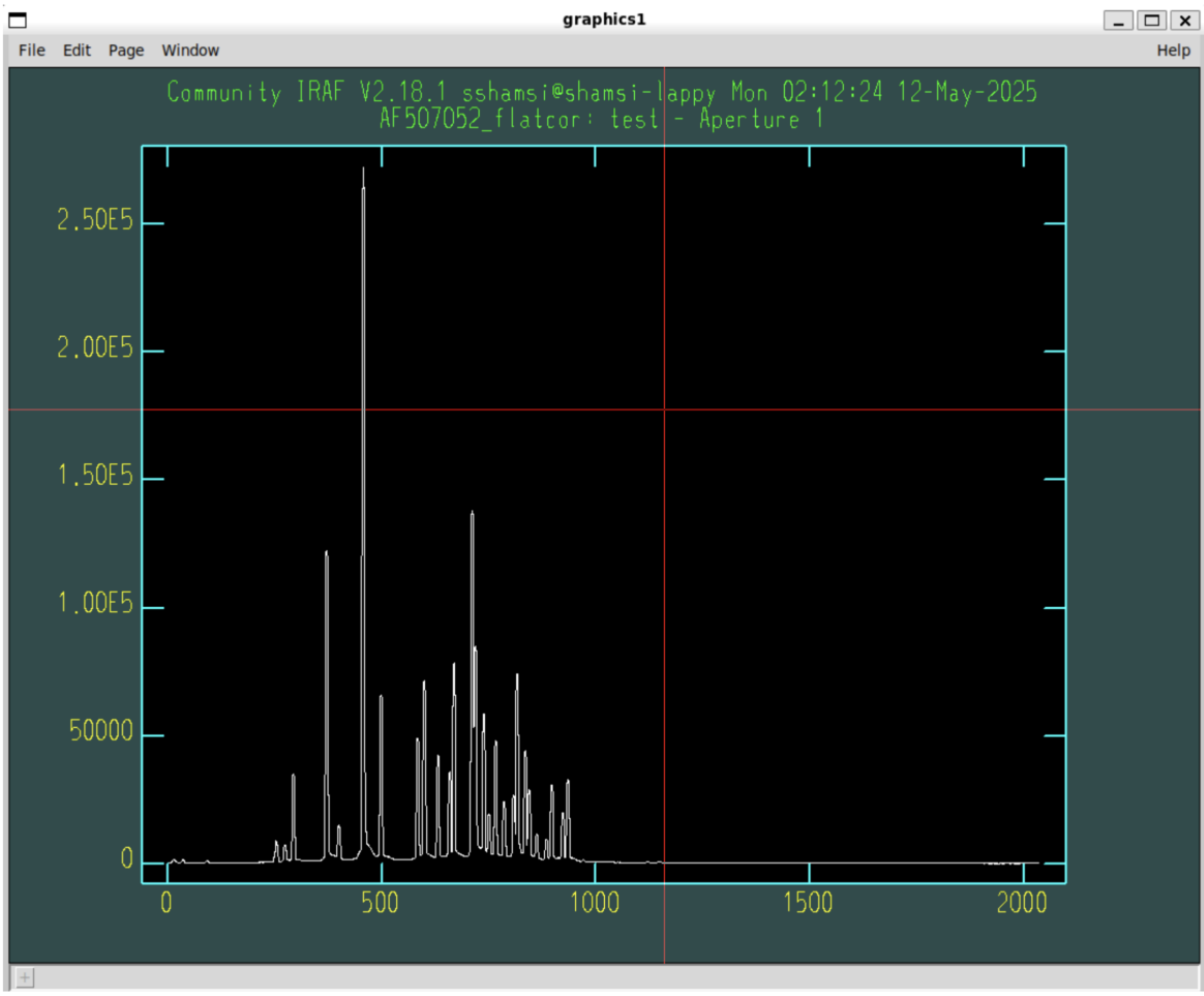
We need to create a file `ne_flatcor.txt` which contains the list of filenames containing images of the Ne lamp which are further bias and flat corrected. And also a file `ne_spec.txt`, which is a list of filenames to assign to the outputted extracted spectra.

Now we can run the task with the following parameters to extract the Ne lamp spectrum:

```
input = @ne_flatcor.txt
output = @ne_spec.txt
referen = obj1
recente no
resize = no
edit = no
trace = no
bkg = no
backgro = none
```

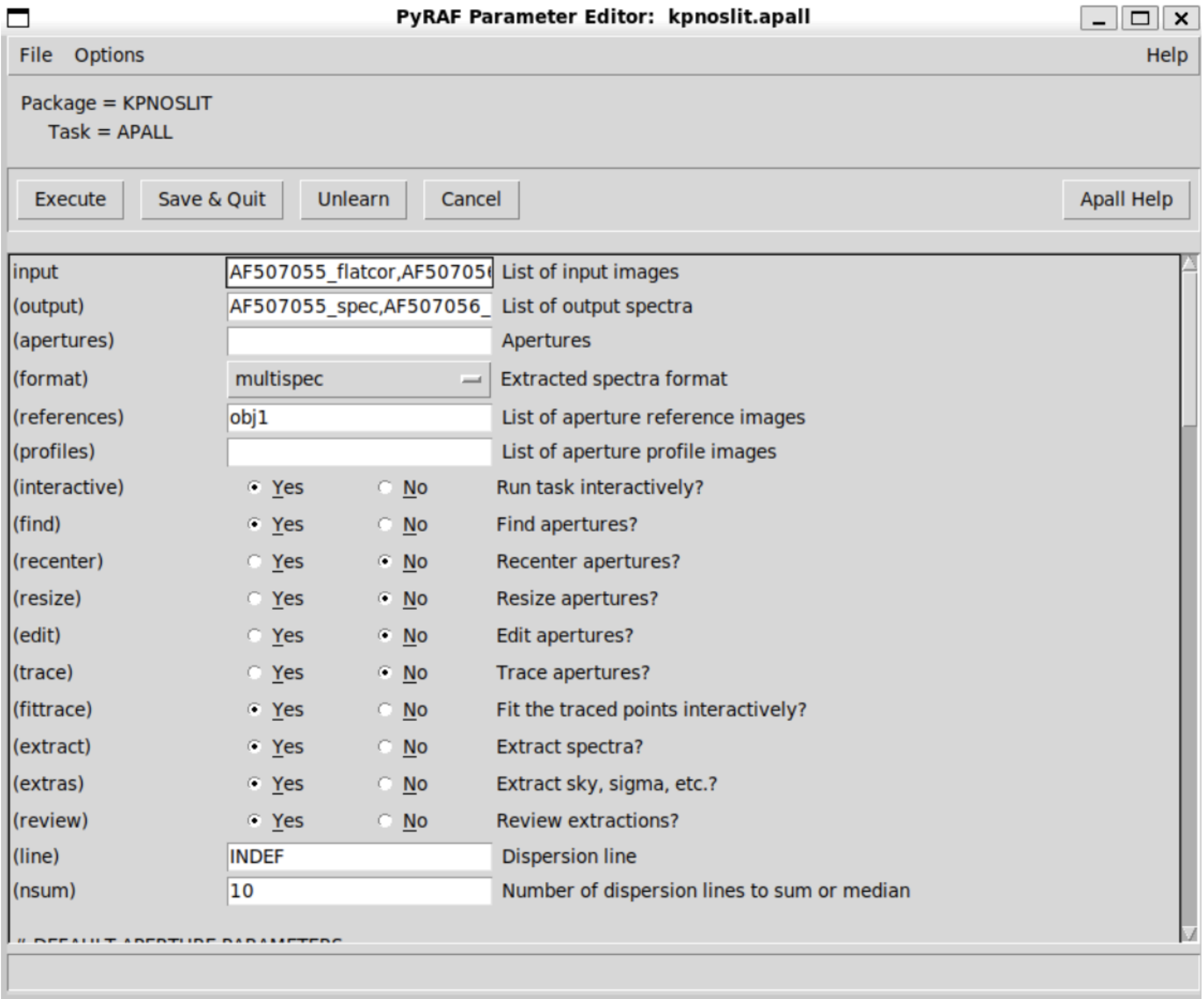


We obtain spectra like the following:

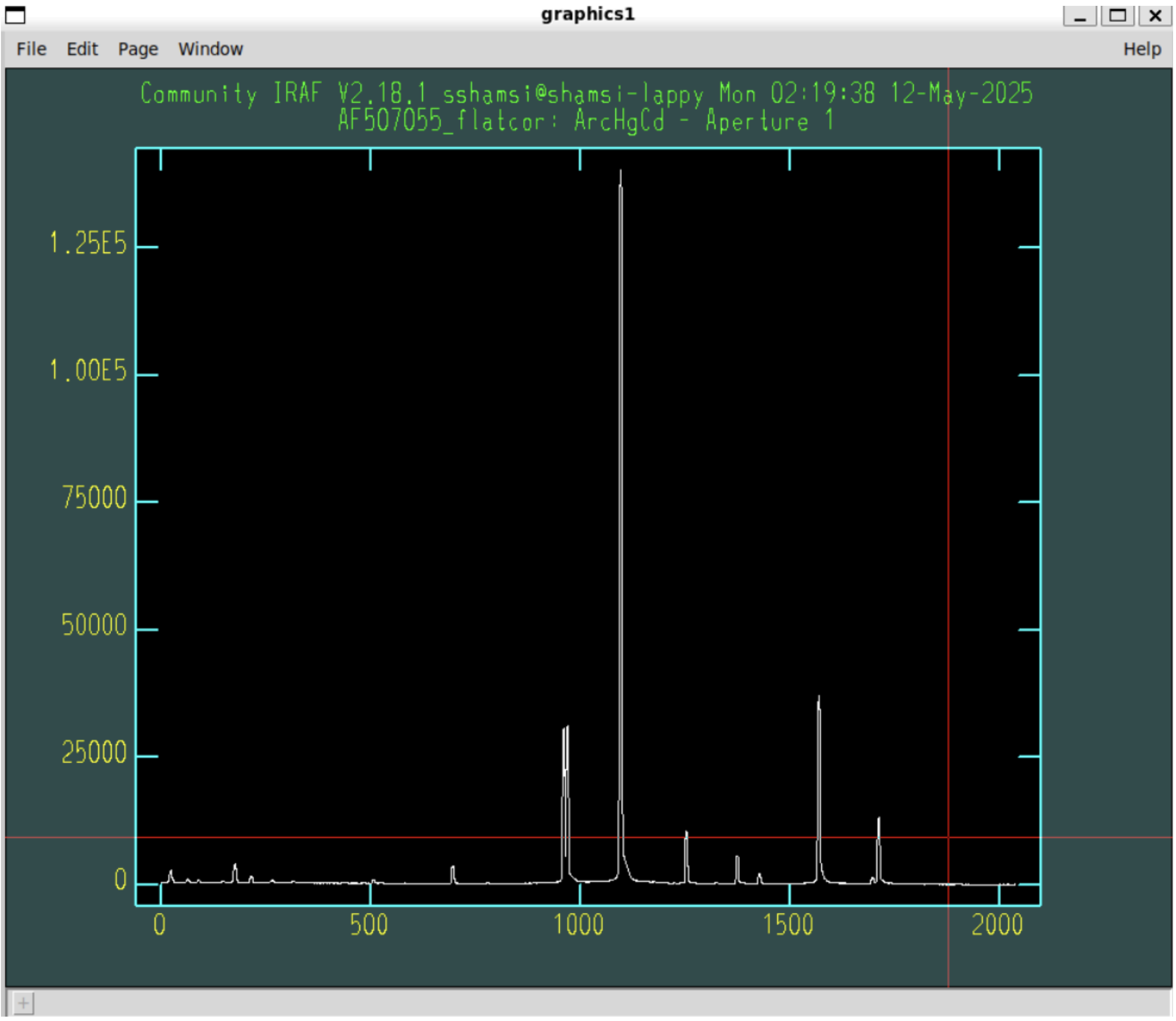


We now repeat this process with the HgCd lamp. We create the corresponding text files for input/output filenames and run the task with the following parameters for the HgCd lamp:

```
input = @hgcd_flatcor.txt
output = @hgcd_spec.txt
referen = obj1
recente no
resize = no
edit = no
trace = no
bkg = no
backgro = none
```



We obtain spectra like the following:



For the 1.82m Copernico data set two different arc lamps were used. As the Ne lamp exhibits emission lamps mostly at one side of the spectrum, while the HgCd lamp on the other, in this case it is in fact necessary to combine them to one final spectrum.

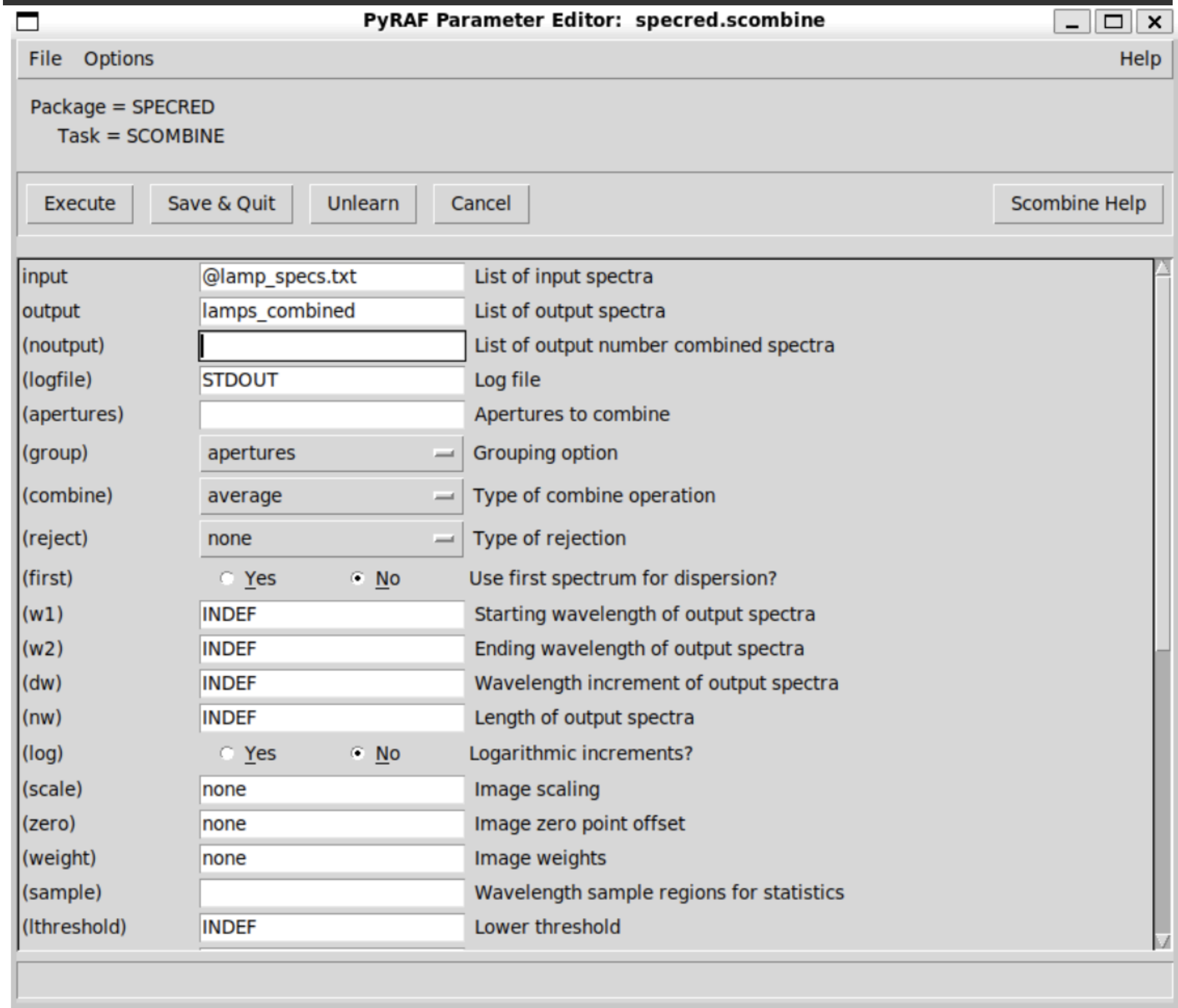
To combine this we use the task:

```
noao
imred
specred
epar scombine
```

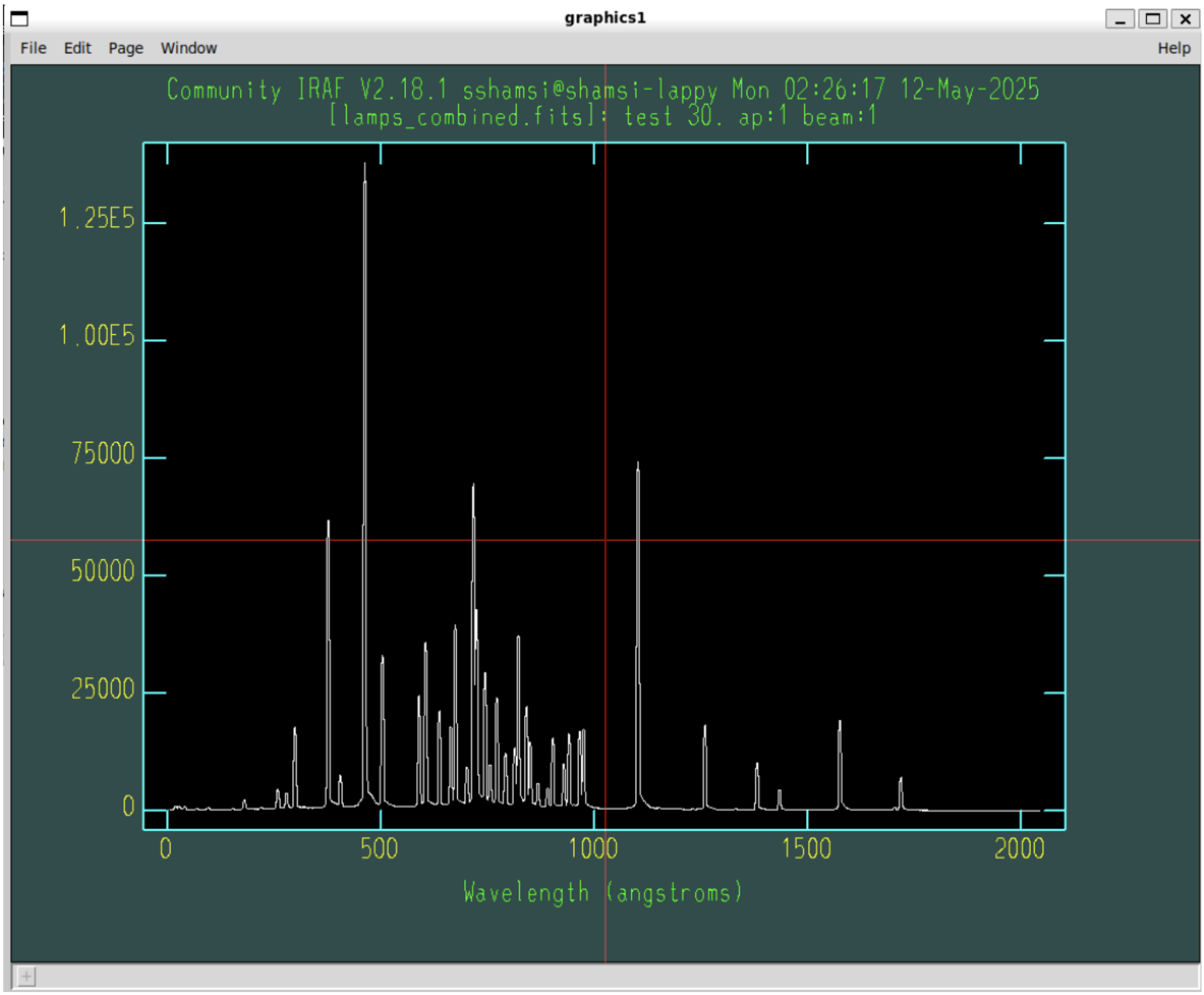
To combine all the spectra from all images of both the Ne and HgCd lamps, we have to create a complete list of filenames `lamps_specs.txt`

We run the task with the following parameters set:

```
input = @lamps_specs.txt
output = lamps_combined
scale = none
zero = none
weight = none
```



Let's take a look at this combined spectrum using splot:



To find the correct wavelength fit, we compare the combined lamp spectrum to a given reference to identify the emission lines.

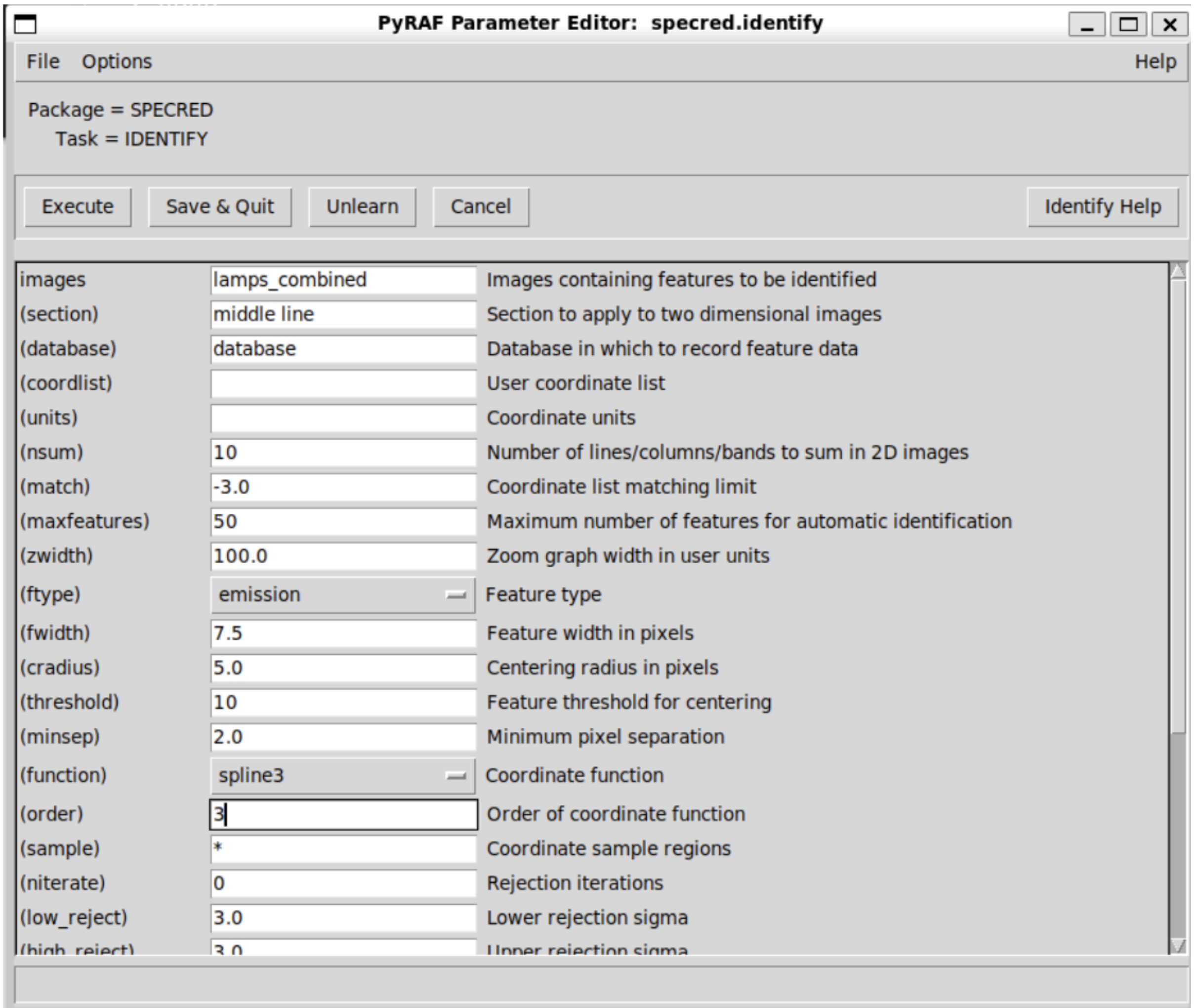
Use the task:

```
noao
imred
kpnoslit
epar identify
```

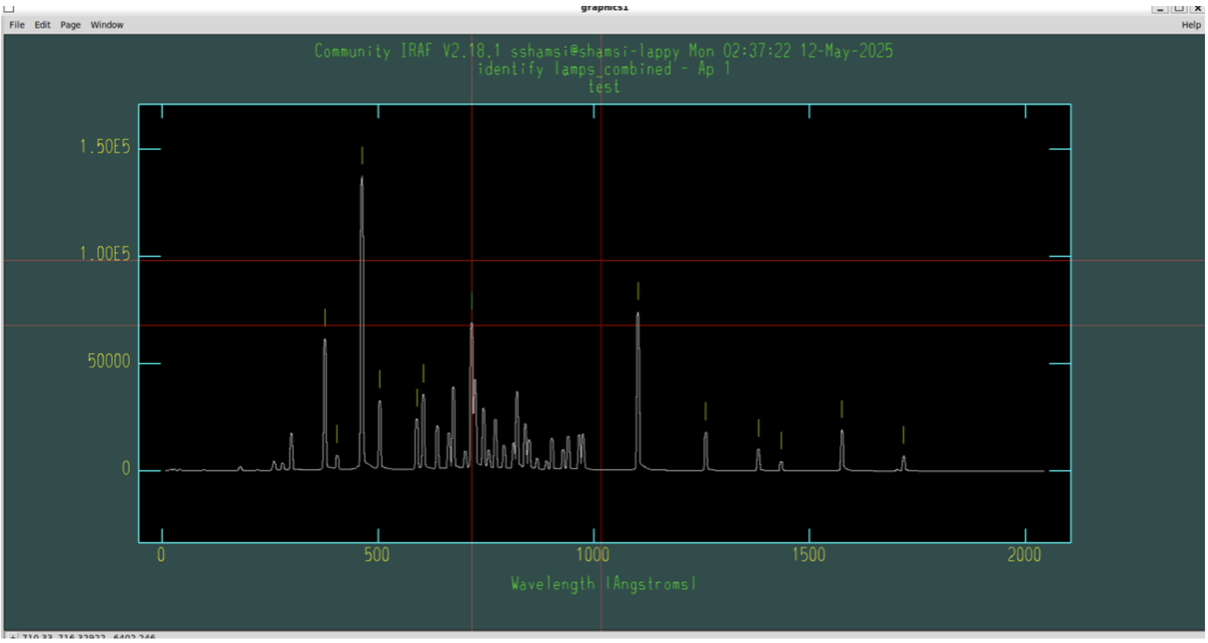
and the parameters:

```
images = lamps_combined
coordli = EMPTY
fwidth = 7.5
```

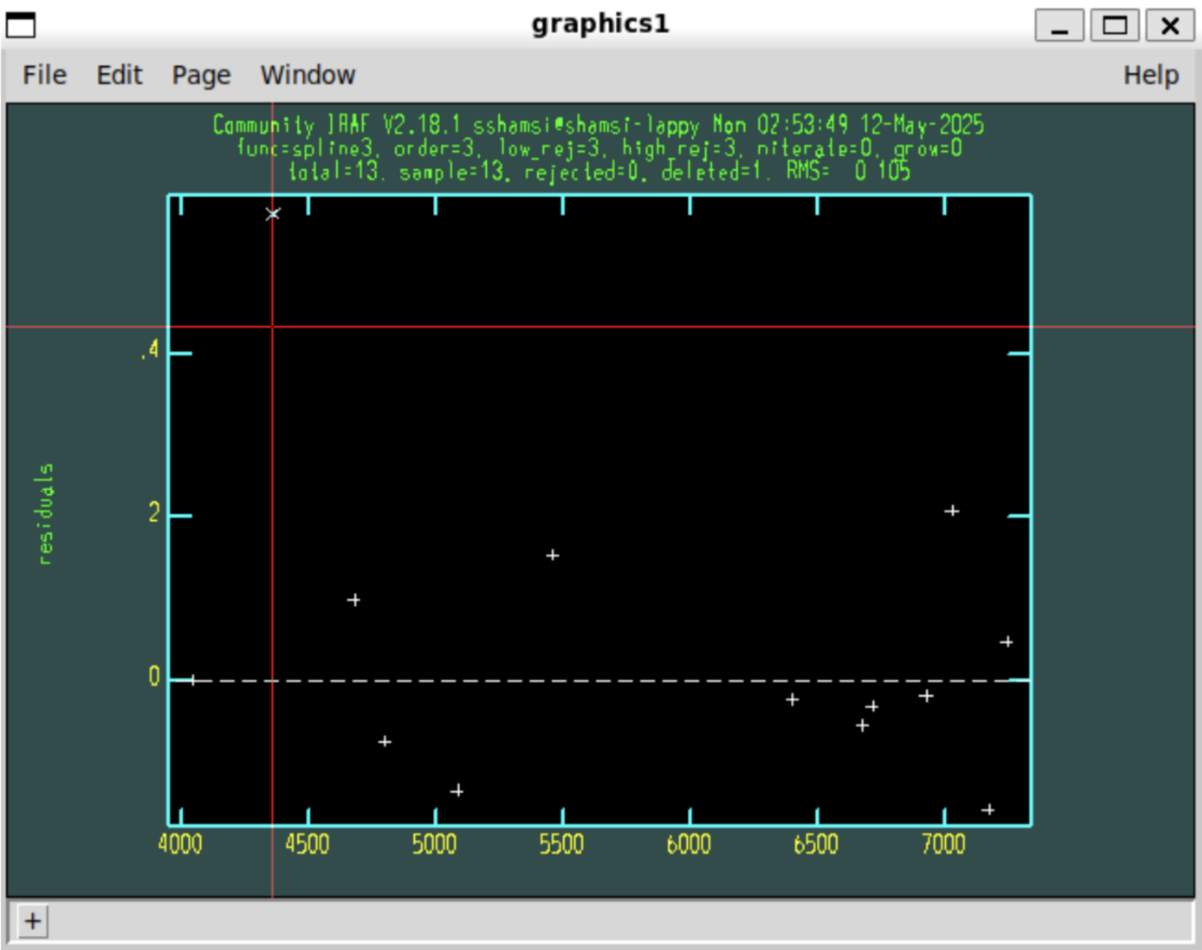
```
cradius = 5.
thesho = 10.
functio = spline3
order = 3
```



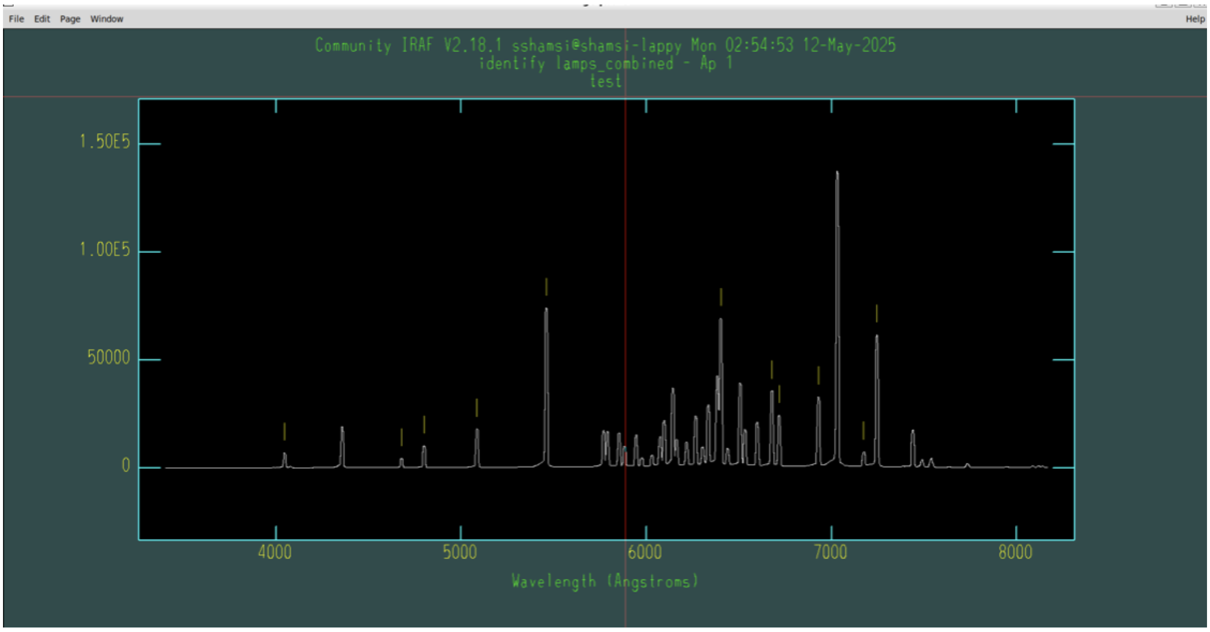
The lamp spectrum is plotted, we use references for line identification (e.g. pg 25 of the tutorial). In order to enter a particular line's wavelength, we move the crosshair to center it on the line. Press m to mark it, then enter the wavelength of the line in units of Angstrom and press ENTER.



Then, press f to start the fitting process, at which point the corresponding residual plot will be shown:



Here, we are again able to fit the line interactively by deleting outliers (d) and updating the fit (f). When finished, press q to quit and end the process.



Wavelength mapping in science spectra

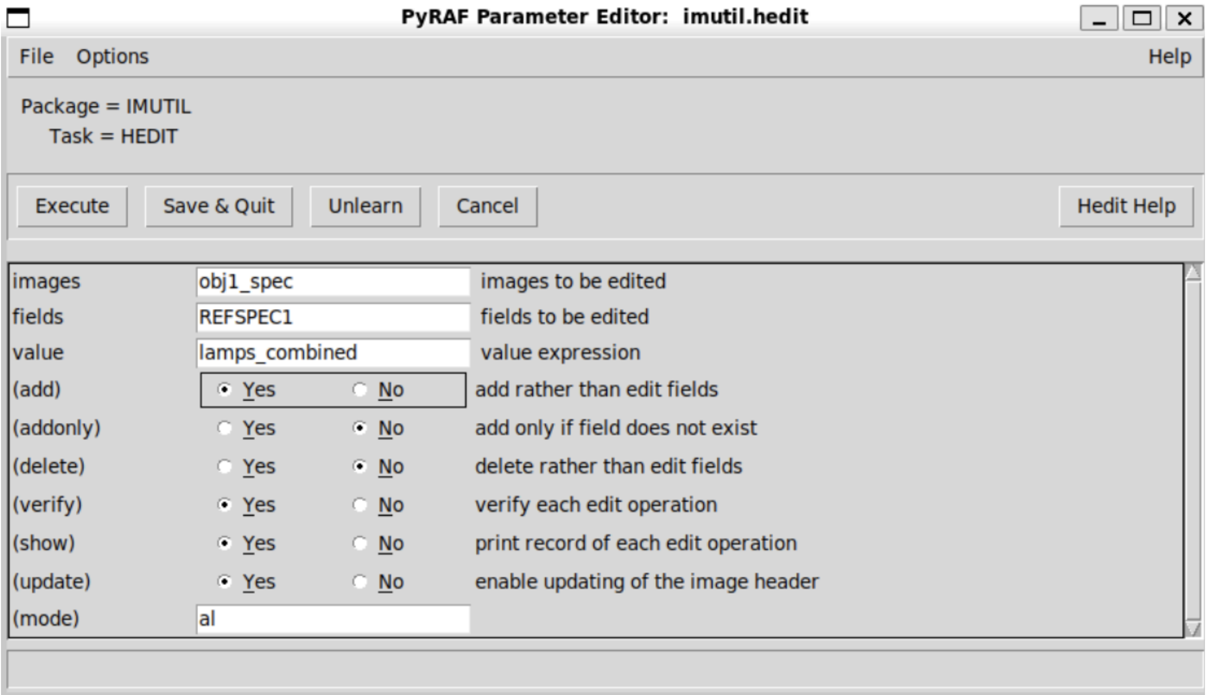
Now, we have a proper wavelength solution, which now needs to be applied to the extracted science target spectra.

Use the command:

```
imutil
epar hedit
```

And use the parameters:

```
images = obj1_spec
fields = REFSPEC1
value = lamps_combined
add = yes
```

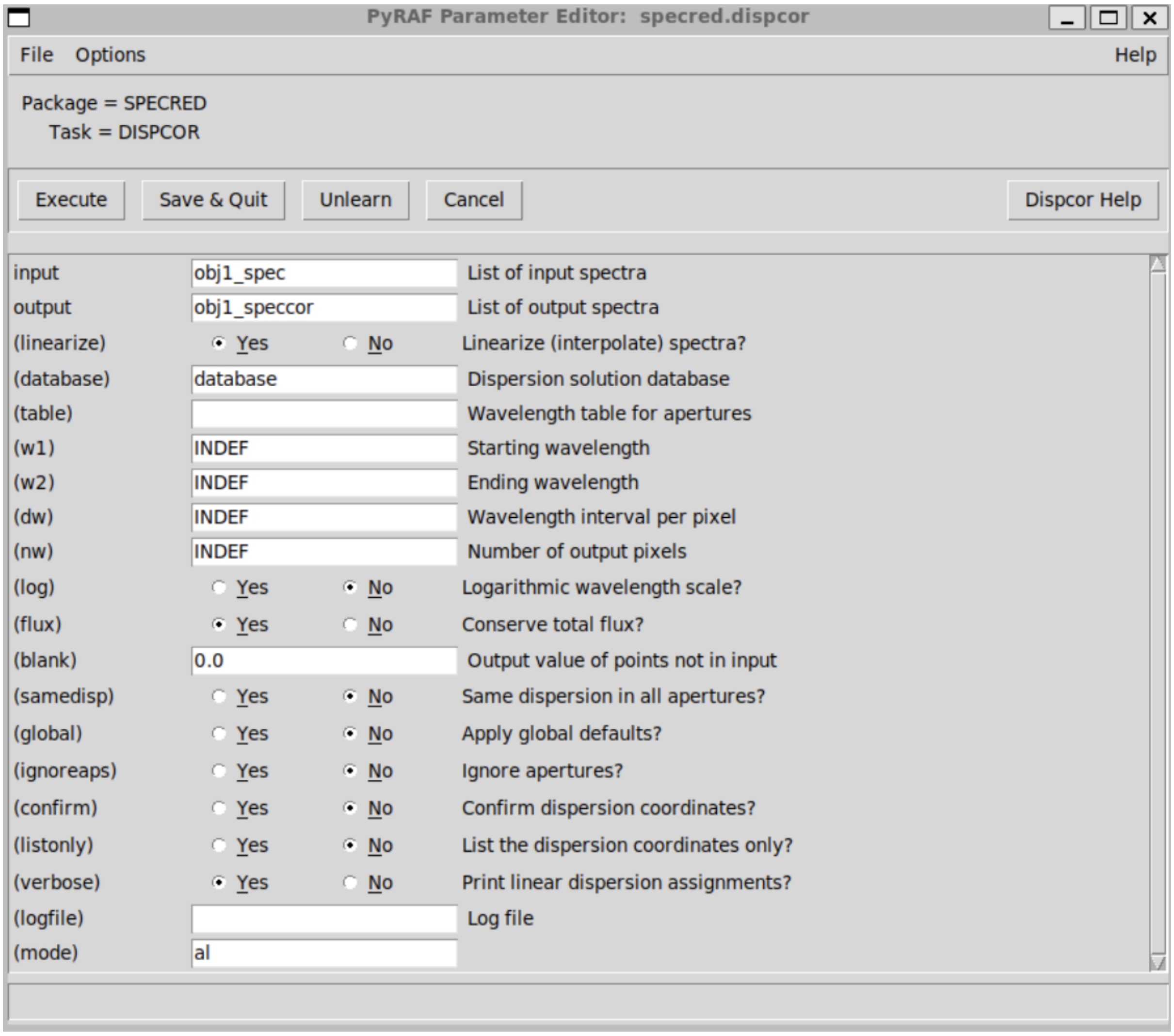


Using package:

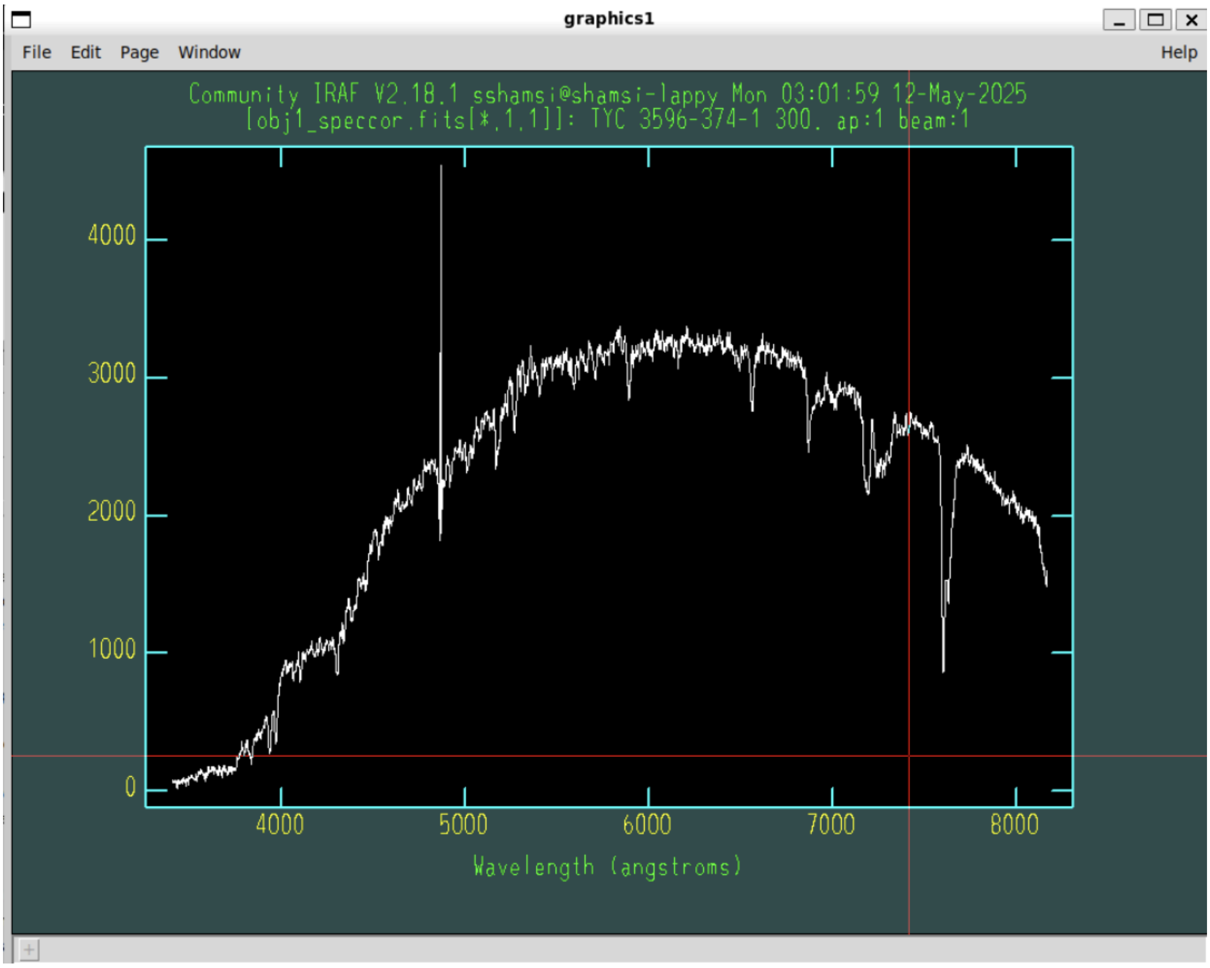
```
noao
imred
kpnoslit
epar dispcor
```

and the parameters:

```
input = obj1_spec
output = obj1_speccor
```



Now, our spectrum when running the splot shows the spectrum plotted vs. wavelength instead of pixels:



Flux Calibrations

The flux distribution in the spectra is currently still affected by the instrument, as well as by further effects, such as the interstellar extinction. In the following, we want to correct both.

Instrumental Response

We can see that the star BD+28 4211 is the standard star to be used for calibration, which from `Copernico_coded.txt`, we know is stored in the file *AF507068.fits*

However, we must first extract the spectrum using the previous method:

```
noao
imred
kpnoslit
```

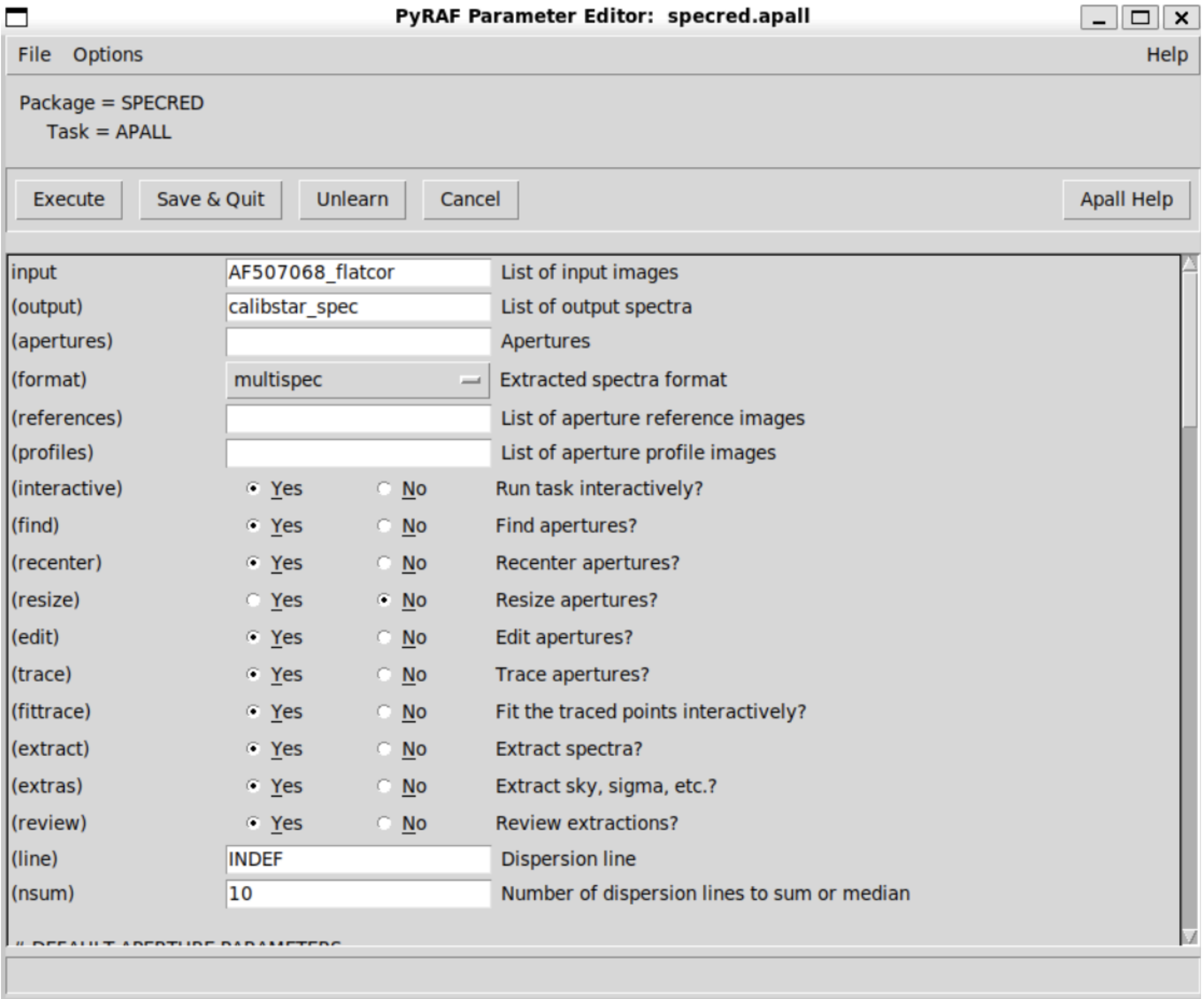
First, reset all parameters to default by using `unlearn apall`, and then run:

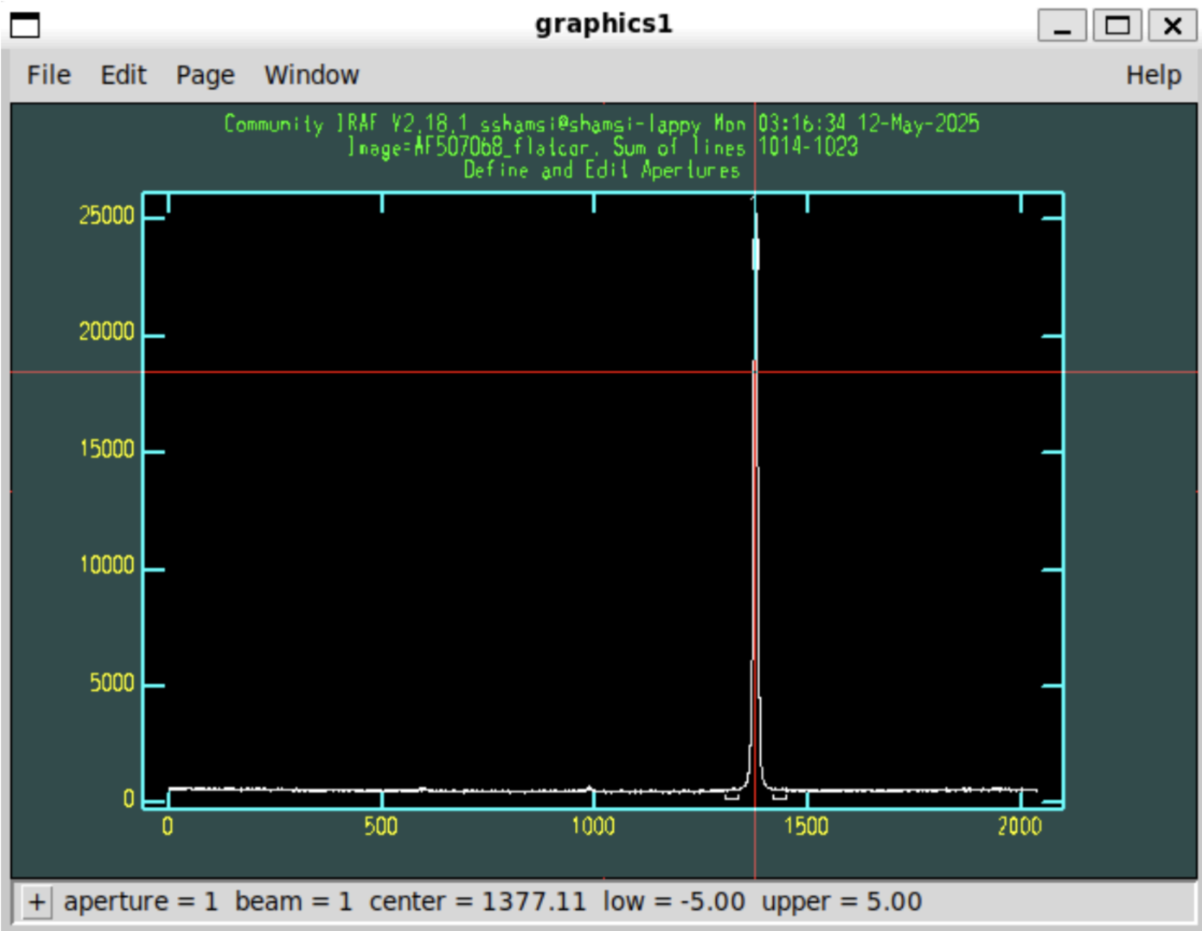
`epar apall`

and use the parameters:

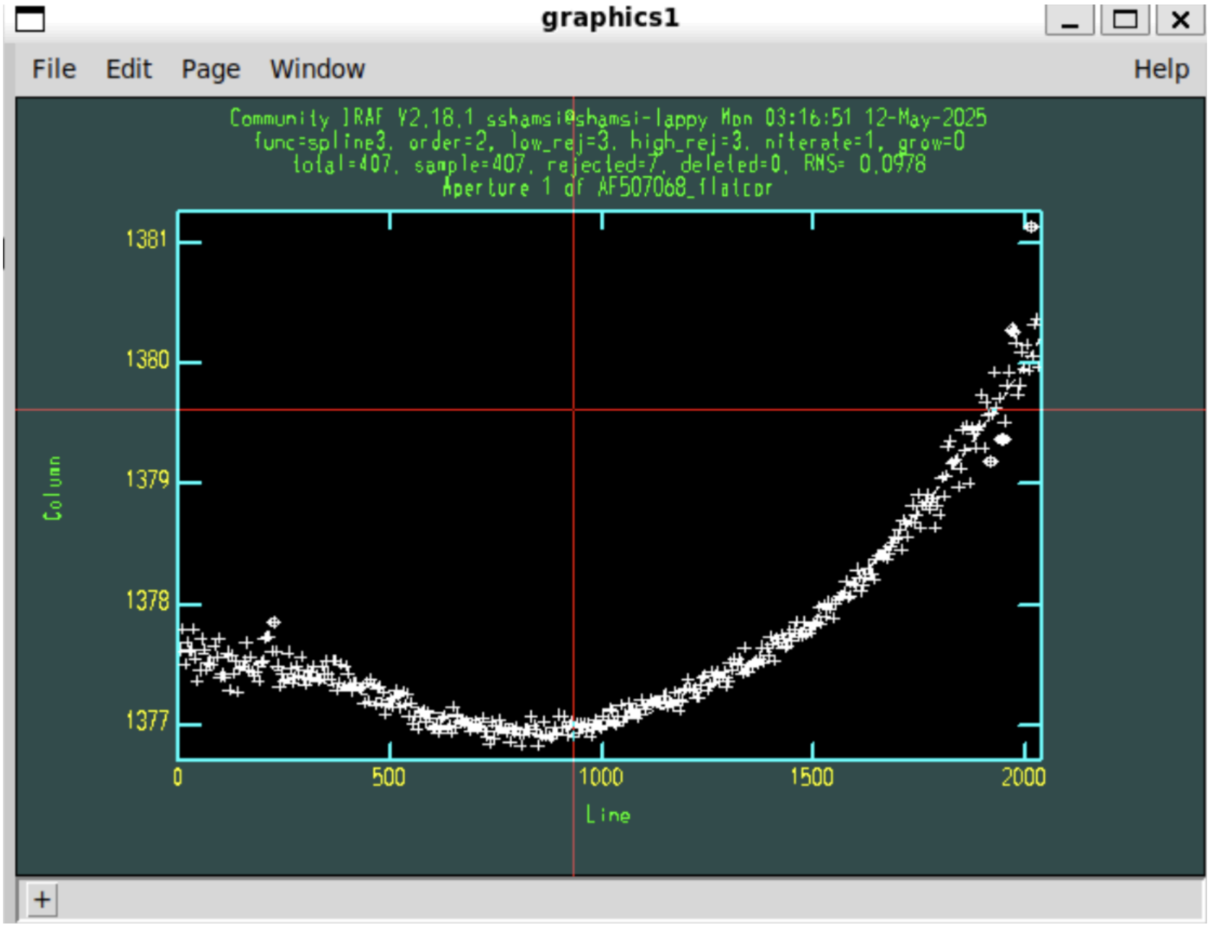
```
input = AF507068.fits
output = calibstar_spec
recente = yes
resize = no
edit = yes
trace = yes
b_funct = chebyshev
b_order = true
b_sample = -70:-40,40:70
b_naver = -15
```

..etc.

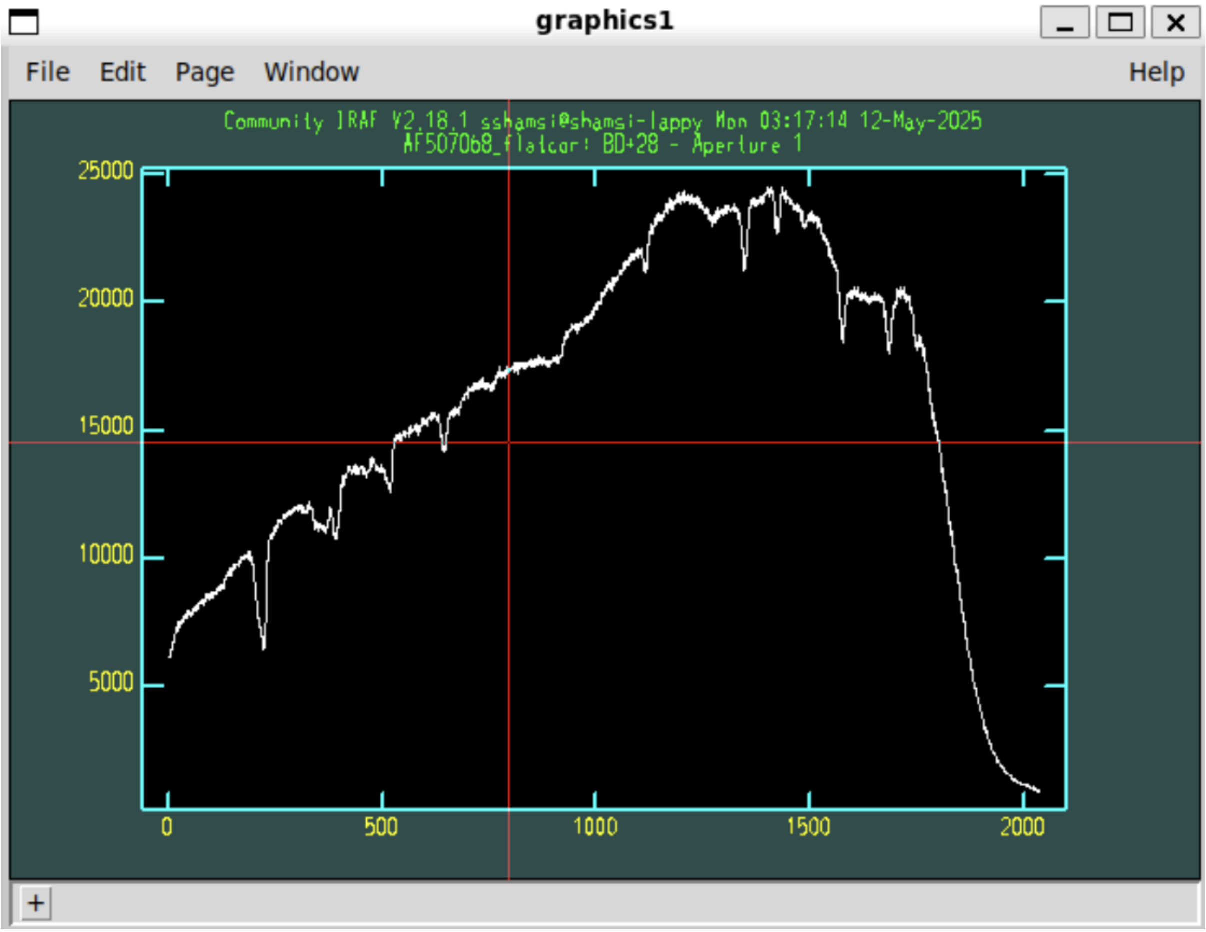

PyRAF Parameter Editor: specred.apall
File Options
Help
Package = SPECRED
Task = APALL
Execute
Save & Quit
Unlearn
Cancel
Apall Help
input AF507068_flatcor List of input images
(output) calibstar_spec List of output spectra
(apertures) Apertures
(format) multispec Extracted spectra format
(references) List of aperture reference images
(profiles) List of aperture profile images
(interactive) Yes No Run task interactively?
(find) Yes No Find apertures?
(recenter) Yes No Recenter apertures?
(resize) Yes No Resize apertures?
(edit) Yes No Edit apertures?
(trace) Yes No Trace apertures?
(fittrace) Yes No Fit the traced points interactively?
(extract) Yes No Extract spectra?
(extras) Yes No Extract sky, sigma, etc.?
(review) Yes No Review extractions?
(line) INDEF Dispersion line
(nsum) 10 Number of dispersion lines to sum or median



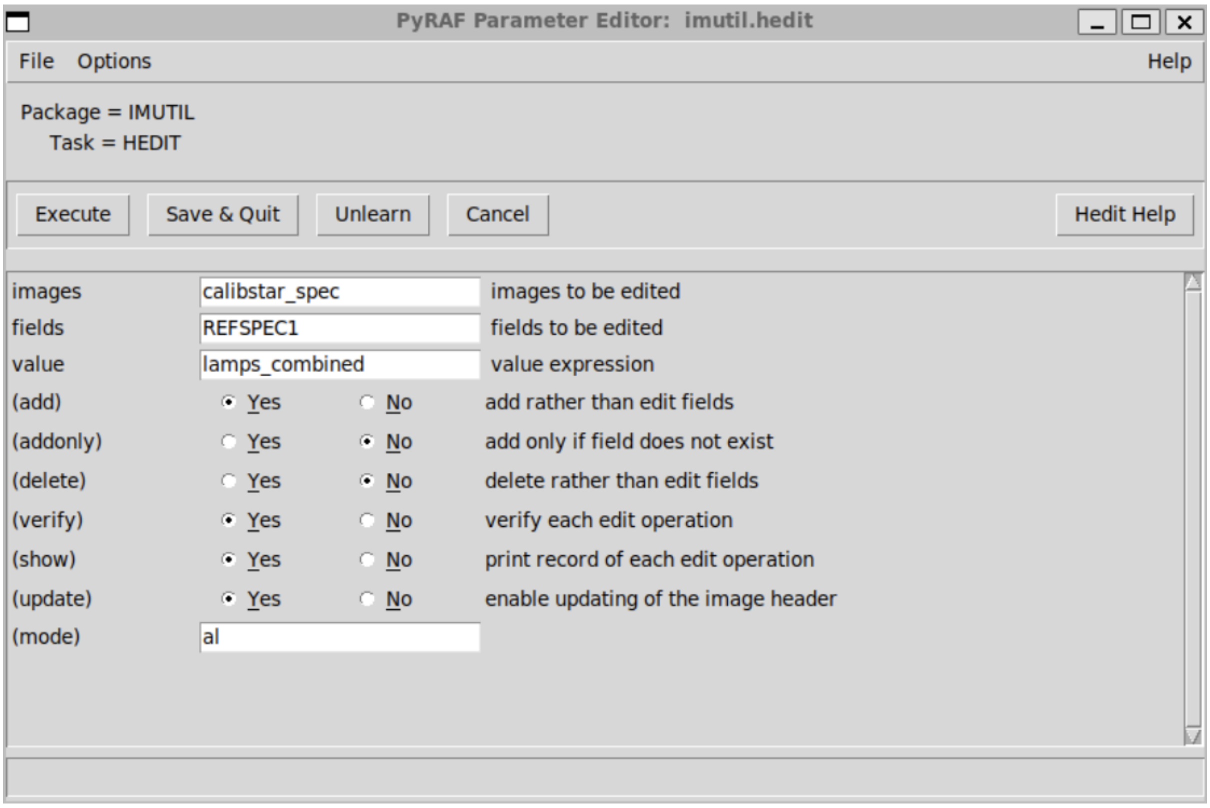
And we get the following:

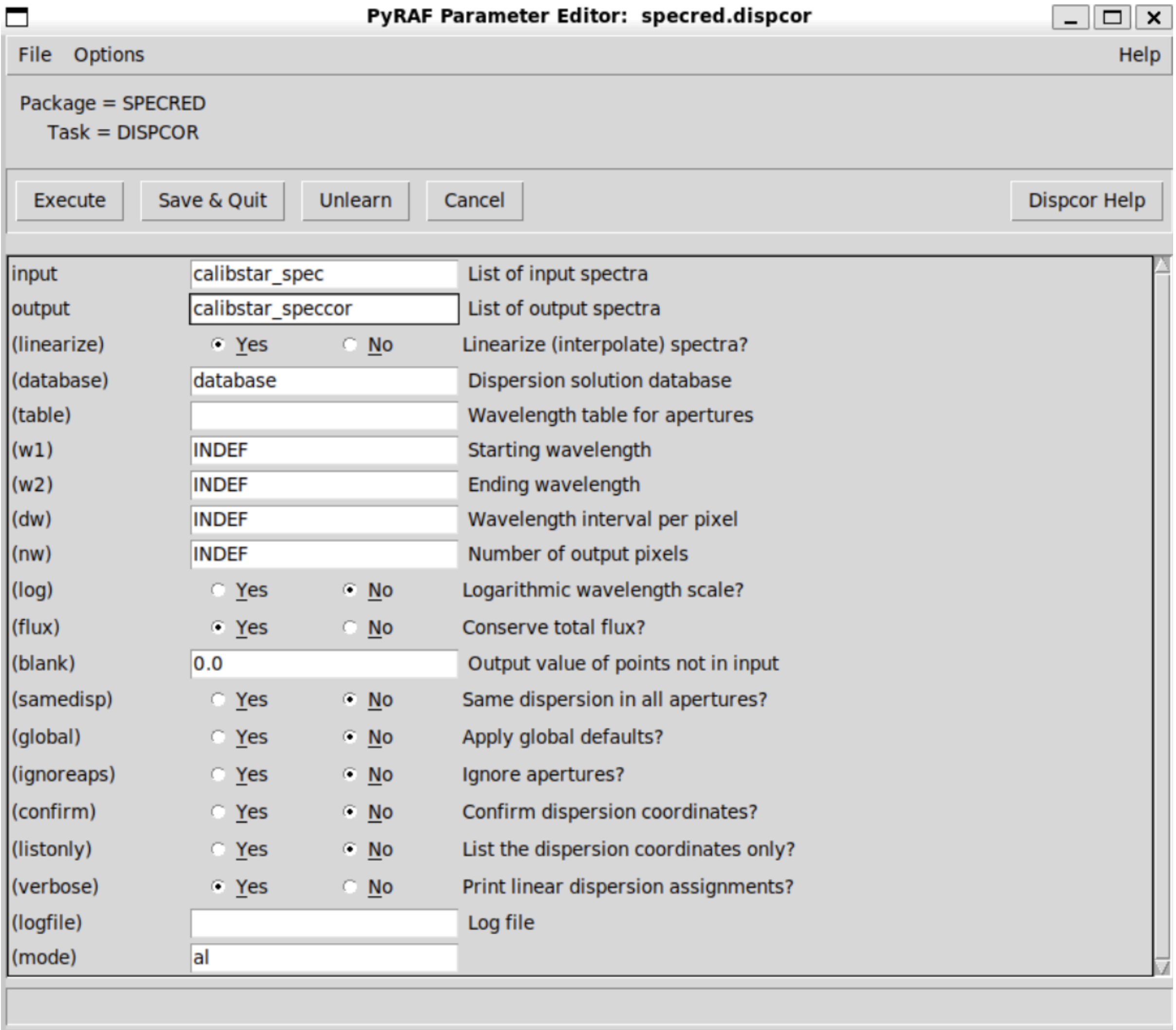


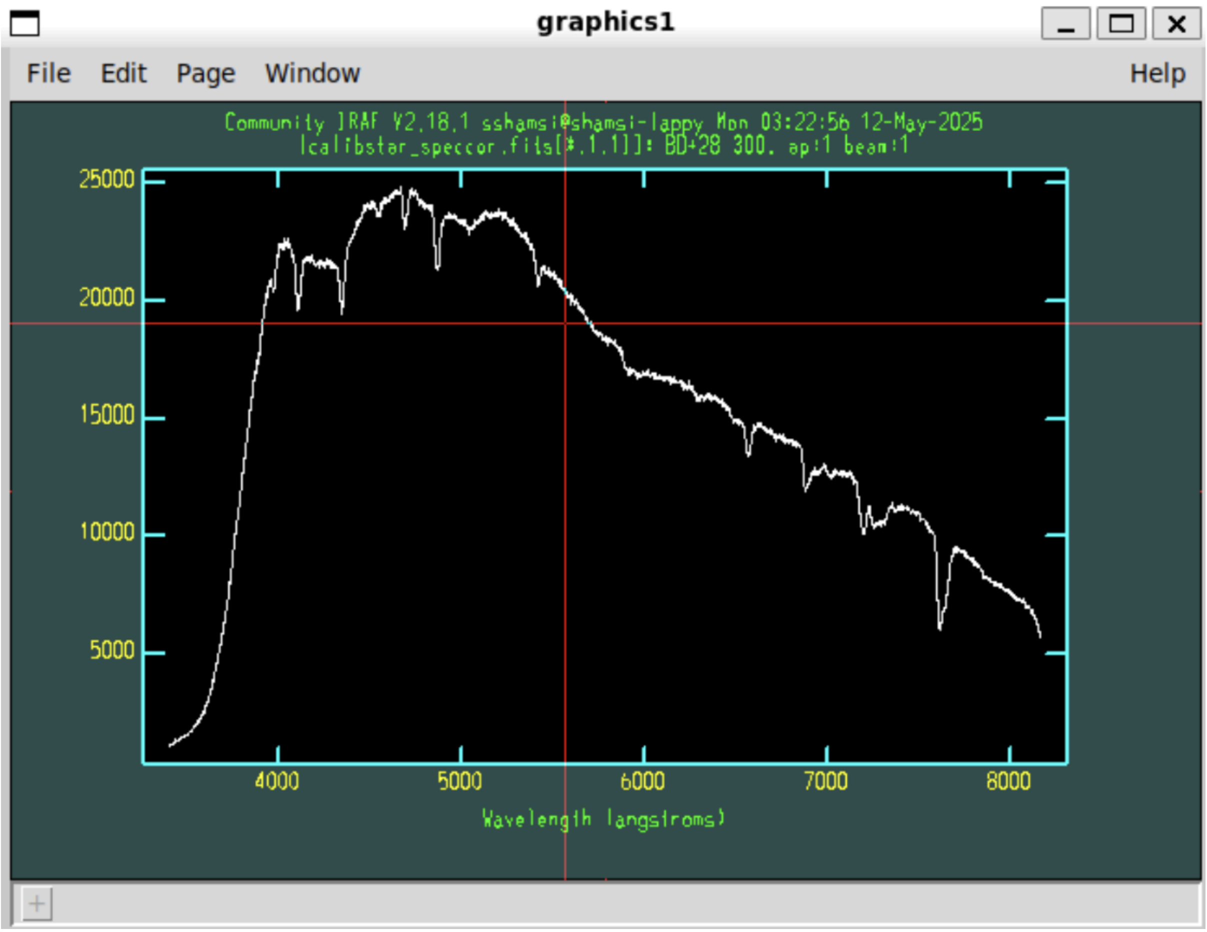
And the spectrum is:



We also apply the wavelength solution found previously to obtain the spectrum `calibstar_speccor` across wavelength in Angstrom (illustrated below using `splot`):


PyRAF Parameter Editor: imutil.hedit
File
Options
Help
Package = IMUTIL
Task = HEDIT
Execute
Save & Quit
Unlearn
Cancel
Hedit Help
images
calibstar_spec
images to be edited
fields
REFSPEC1
fields to be edited
value
lamps_combined
value expression
(add)
Yes
No
add rather than edit fields
(addonly)
Yes
No
add only if field does not exist
(delete)
Yes
No
delete rather than edit fields
(verify)
Yes
No
verify each edit operation
(show)
Yes
No
print record of each edit operation
(update)
Yes
No
enable updating of the image header
(mode)
al


PyRAF Parameter Editor: specred.dispcor
File Options
Help
Package = SPECRED
Task = DISPCOR
Execute
Save & Quit
Unlearn
Cancel
Dispcor Help
input calibstar_spec List of input spectra
output calibstar_speccor List of output spectra
(linearize) Yes No Linearize (interpolate) spectra?
(database) database Dispersion solution database
(table) Wavelength table for apertures
(w1) INDEF Starting wavelength
(w2) INDEF Ending wavelength
(dw) INDEF Wavelength interval per pixel
(nw) INDEF Number of output pixels
(log) Yes No Logarithmic wavelength scale?
(flux) Yes No Conserve total flux?
(blank) 0.0 Output value of points not in input
(samedisp) Yes No Same dispersion in all apertures?
(global) Yes No Apply global defaults?
(ignoreaps) Yes No Ignore apertures?
(confirm) Yes No Confirm dispersion coordinates?
(listonly) Yes No List the dispersion coordinates only?
(verbose) Yes No Print linear dispersion assignments?
(logfile) Log file
(mode) al

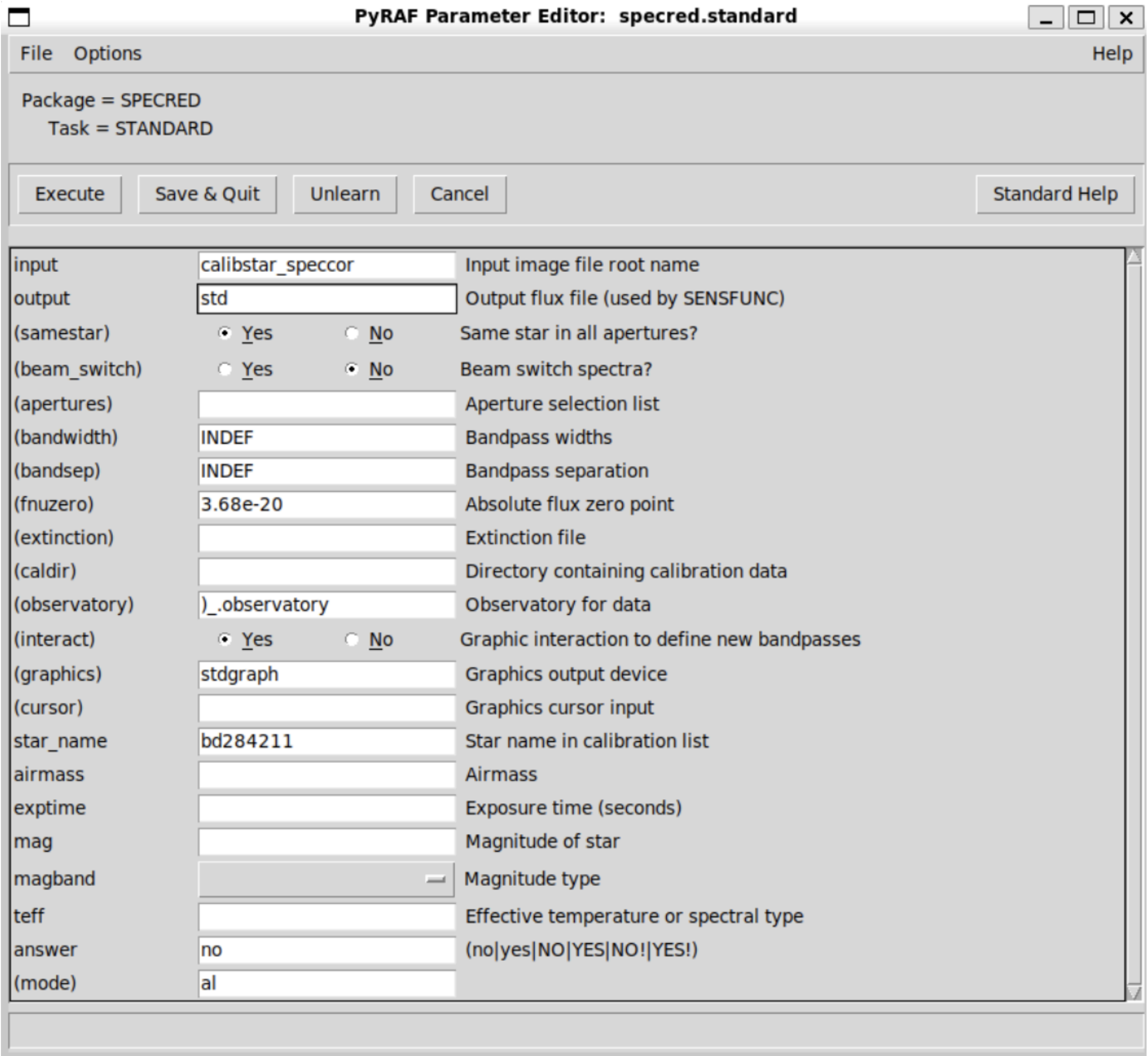


Now, for the flux calibration, we use the following command:

```
noao
imred
kpnoslit
epar standard
```

We use the following parameters:

```
images = calibstar_speccor
output = std
caldir = onedstds$spec50cal/
star_nam = bd284211
airmass = EMPTY (will be taken from file header)
exptime = EMPTY (will be taken from file header)
```



PyRAF Parameter Editor: specred.standard

File Options Help

Package = SPECRED
Task = STANDARD

Execute | Save & Quit | Unlearn | Cancel | Standard Help

input	calibstar_speccor	Input image file root name
output	std	Output flux file (used by SENSFUNC)
(samestar)	(•) Yes ( ) No	Same star in all apertures?
(beam_switch)	( ) Yes (•) No	Beam switch spectra?
(apertures)		Aperture selection list
(bandwidth)	INDEF	Bandpass widths
(bandsep)	INDEF	Bandpass separation
(fnuzero)	3.68e-20	Absolute flux zero point
(extinction)		Extinction file
(caldir)		Directory containing calibration data
(observatory)	)_.observatory	Observatory for data
(interact)	(•) Yes ( ) No	Graphic interaction to define new bandpasses
(graphics)	stdgraph	Graphics output device
(cursor)		Graphics cursor input
star_name	bd284211	Star name in calibration list
airmass		Airmass
exptime		Exposure time (seconds)
mag		Magnitude of star
magband		Magnitude type
teff		Effective temperature or spectral type
answer	no	(no|yes|NO|YES|NO!|YES!)
(mode)	al	

To fit the sensitivity function:

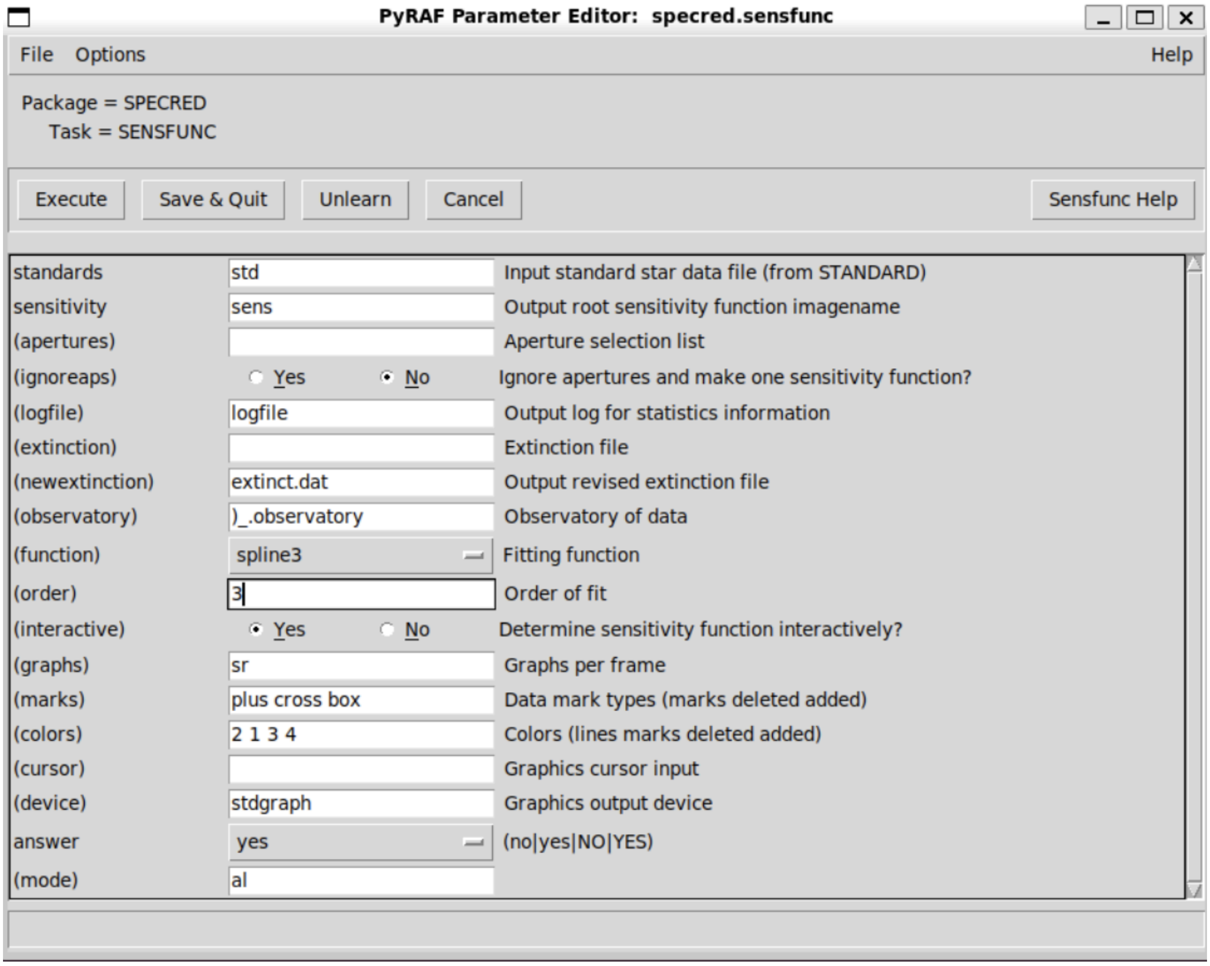
Command:

```
noao
imred
kpnoslit
epar sensfunc
```

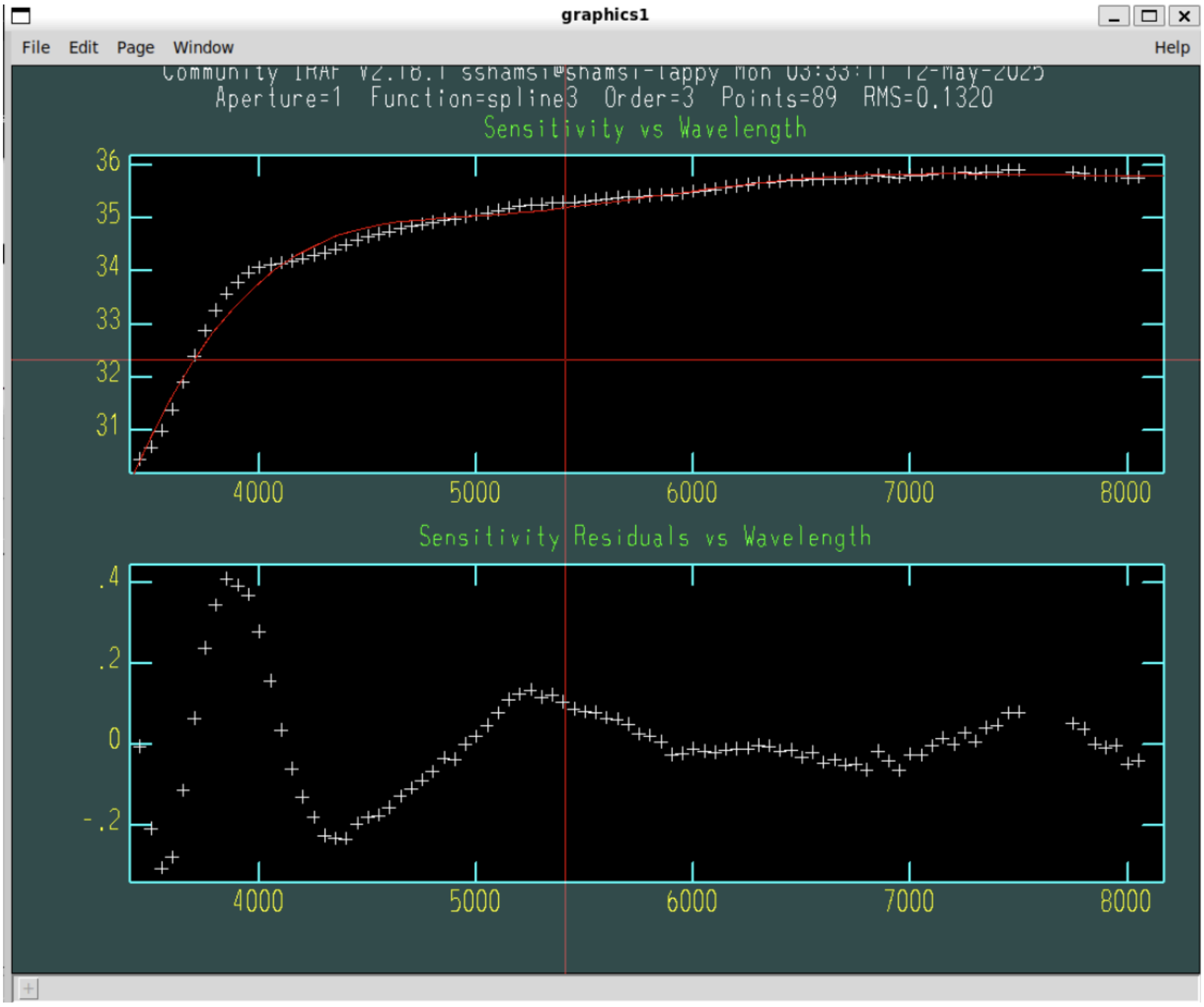
Parameters:

```
standard = std
sensitiv = sens
```

```
functio = spline3
order = 3
```



And can fit the sensitivity function using the following:



The fitted sensitivity function is used to correct the flux within the science targets' spectra.

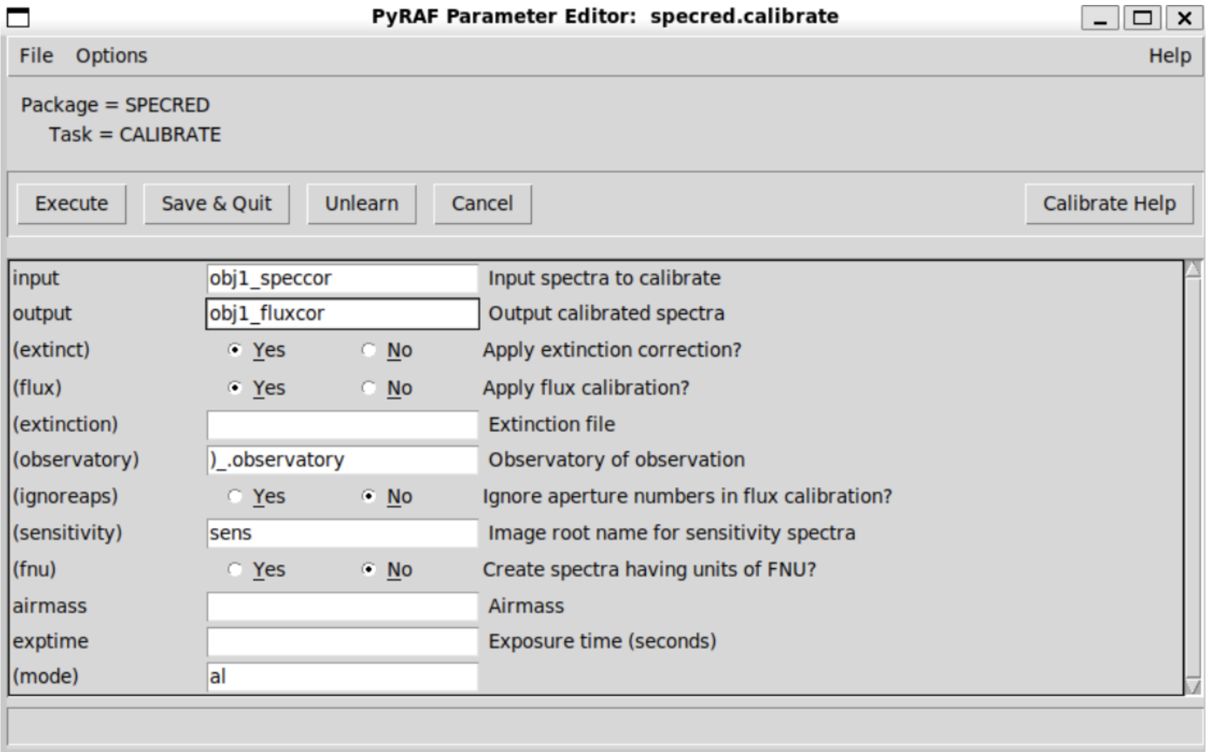
Use the task:

```
noao
imred
kpnoslit
epar calibrate
```

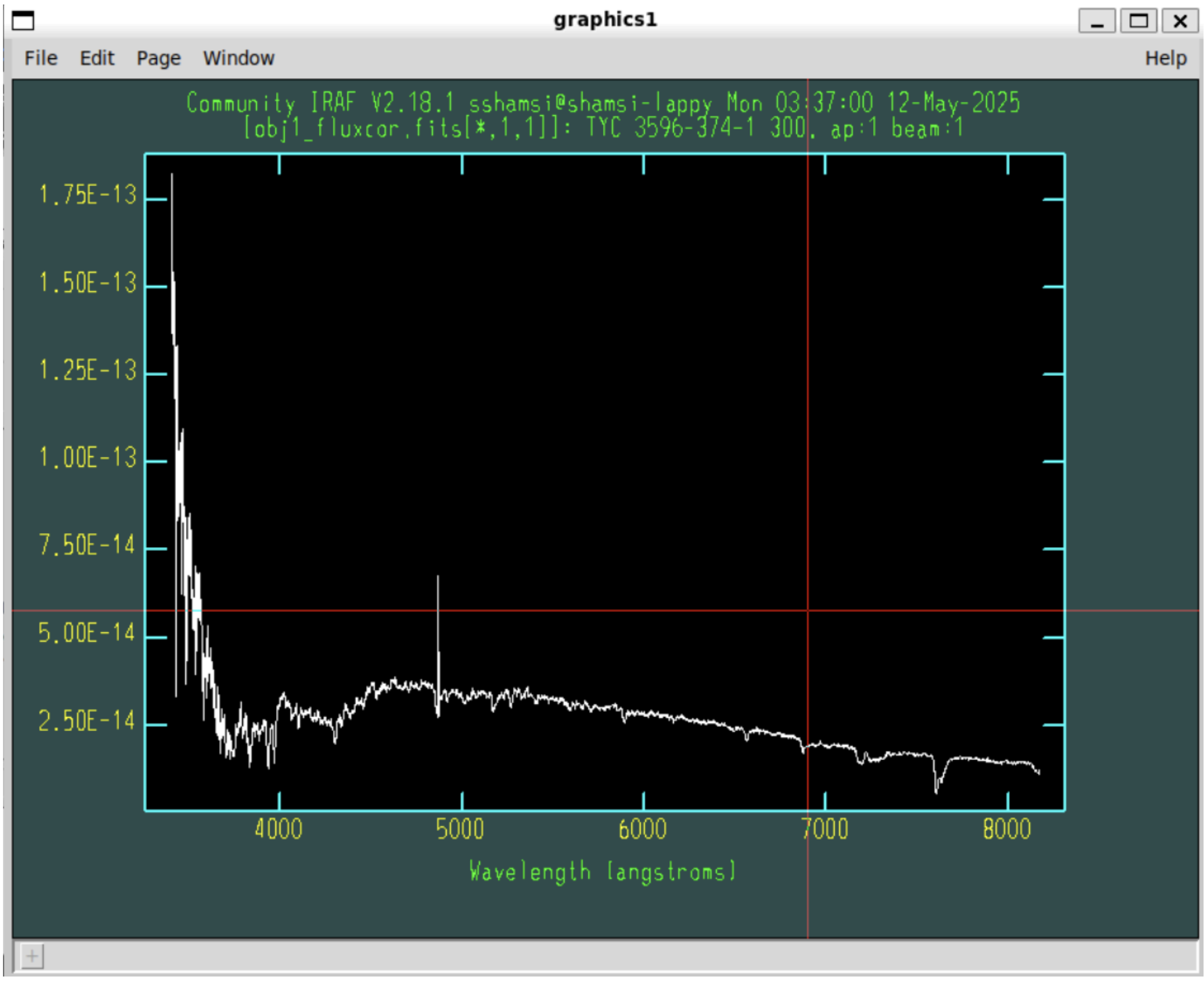
and the input parameters:

```
input = obj1_speccor
output = obj1_fluxcor
```

```
extinct = yes
flux = yes
airmass = EMPTY
exptime = EMPTY
```



and we get the flux-corrected spectrum for object 1:



Correction for interstellar extinction

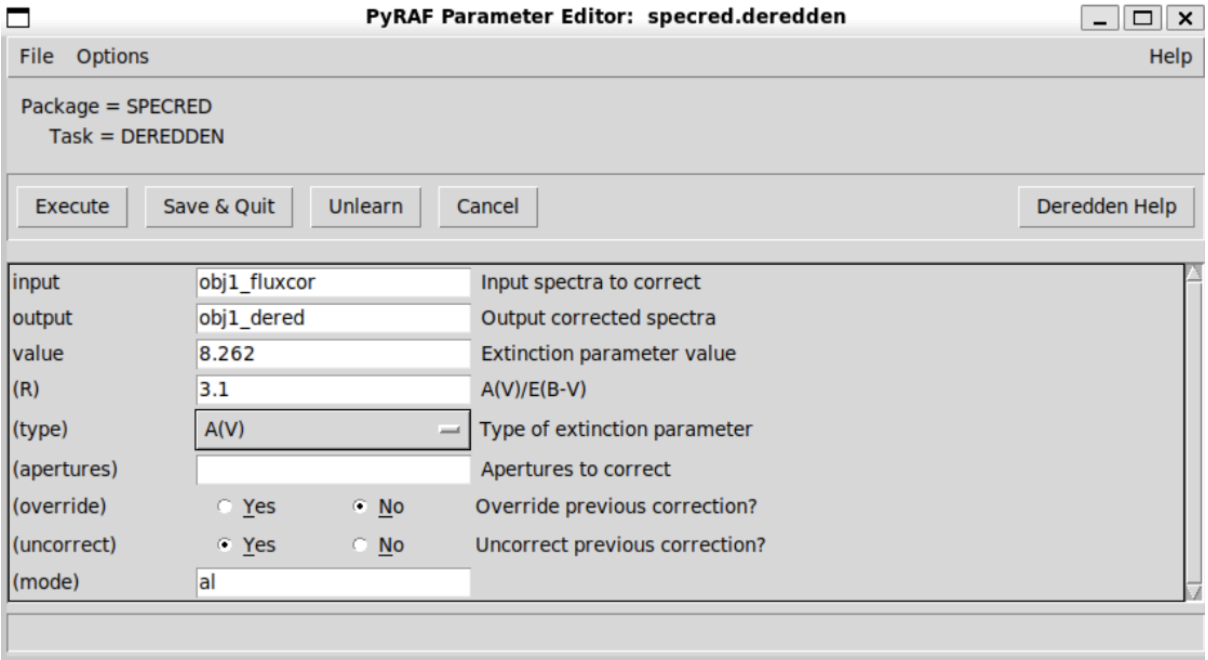
We need to de-redden the wavelengths to remove the effect of extinction due to interstellar matter in the Milky Way.

Use the task:

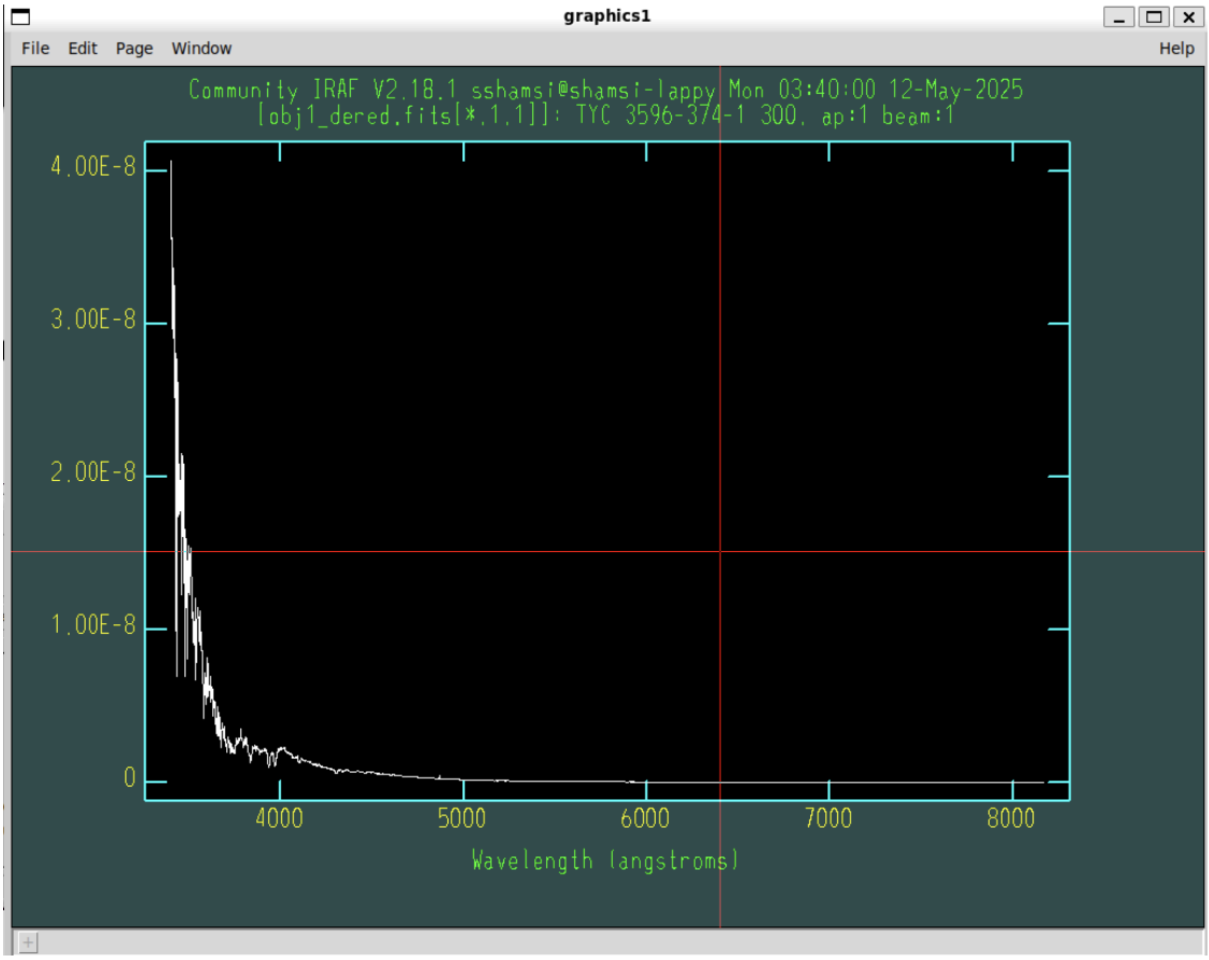
```
noao
imred
kpnoslit
epar deredden
```

And obtaining the value of the extinction experienced by observations of object 1 from Table 5.

```
input = obj1_fluxcor
output = obj1_dered
value = 8.262
R = 3.1
type = A(V)
```



We obtain now the dereddened flux-corrected spectrum:


graphics1
File Edit Page Window
Help
Community IRAF V2.18.1 sshamsi@shamsi-lappy Mon 03:40:00 12-May-2025
[obj1_dered.fits[*,1,1]]: TYC 3596-374-1 300. ap:1 beam:1
4.00E-8
3.00E-8
2.00E-8
1.00E-8
0
4000
5000
6000
7000
8000
Wavelength (angstroms)