

IRAF Task

Astrophysical Techniques and Data Reduction

Shoaib Jamal Shamsi

We built **iraf** and **x11iraf**, and installed **pyraf** on our Ubuntu 24.04 system on WSL. We will now follow the tutorial highlighted in the **IRAF_task.pdf** document.

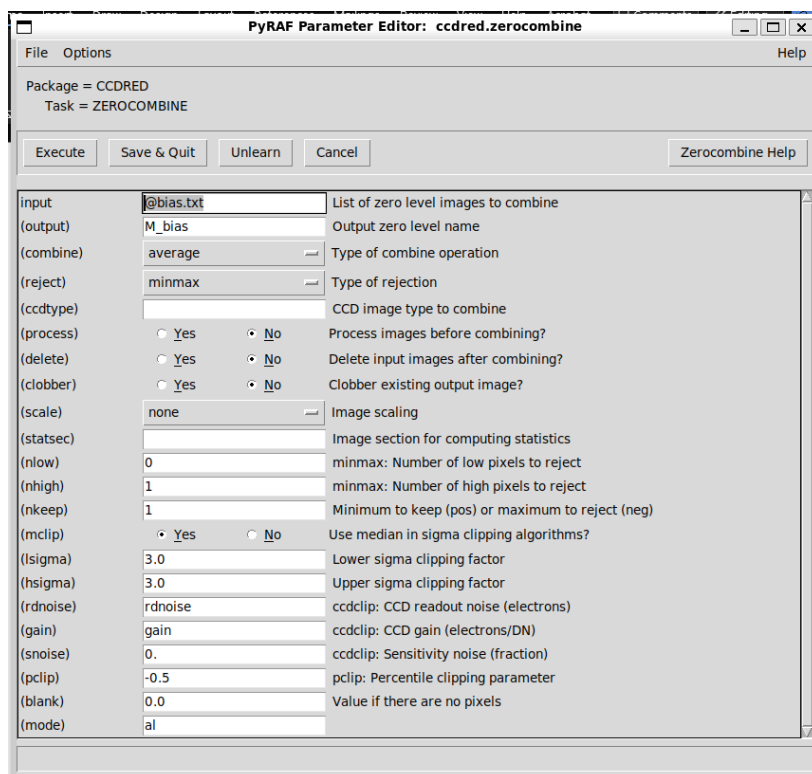
5) Data Reduction

5.1) Calibration

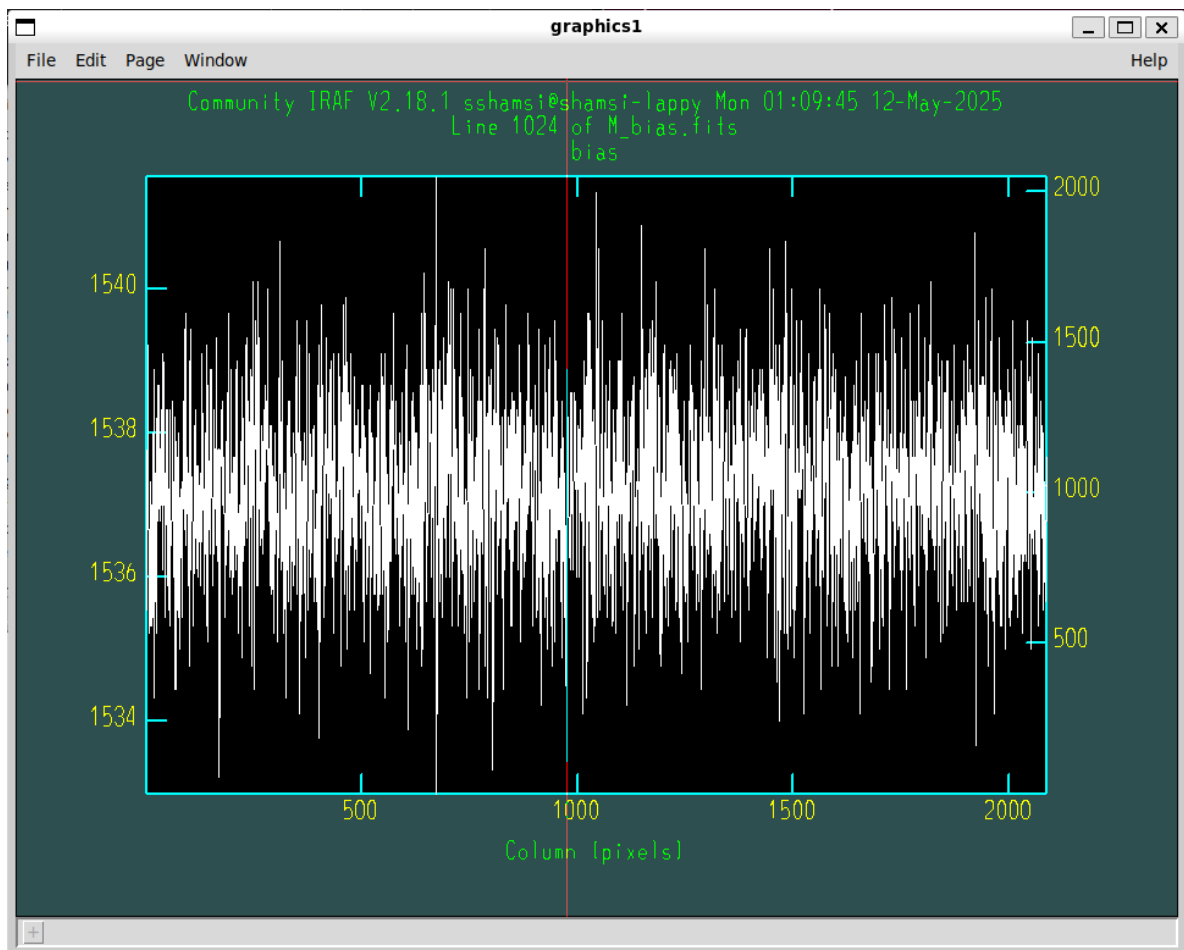
Baiais Correction

i) Combination of the Bias Files

We have some bias FITS files, listed both in the **Copernico_coded.txt** and **bias.txt** files. We will run the **zerocorrection** task:



This outputs the **M_bias** file, which can be seen below:



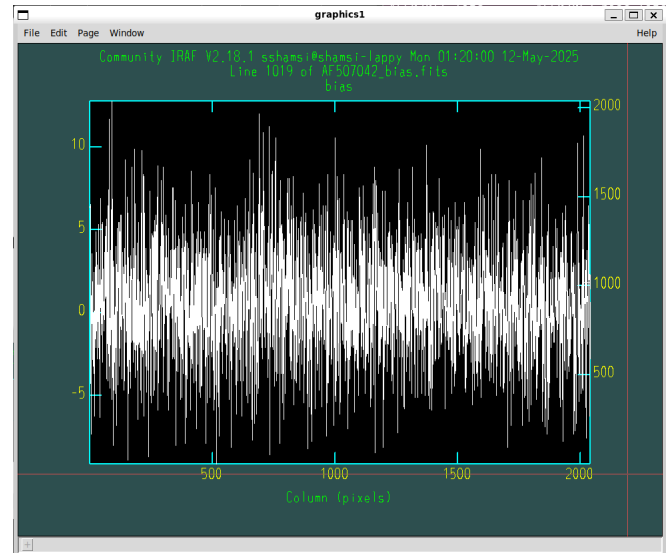
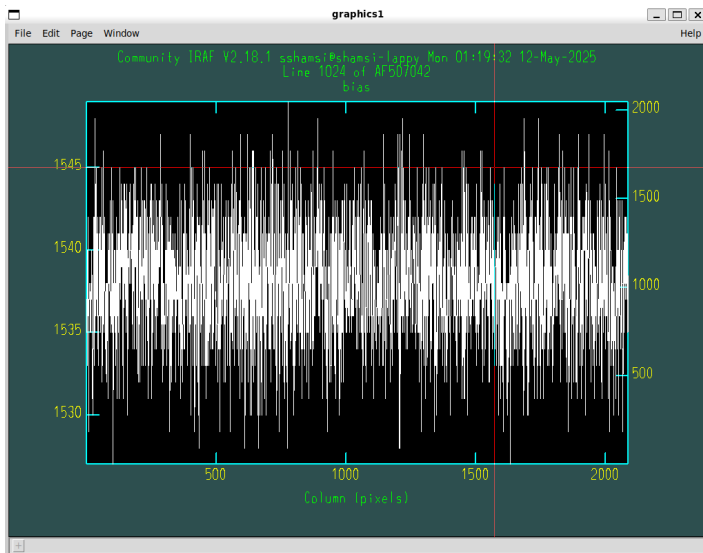
ii) Applying the bias correction and trimming

We can now use the task **ccdproc** to apply bias correction from **M_bias** to all our images, saving the results with '**_bias**' after the filename. We define the new filenames in a separate file called **all_bias.txt**.

```
TASK : ccdproc
PACKAGE : noao -> imred -> ccdred
```

```
input = @all.txt Text
output = @all_bias.txt
ccdtype = BLANK
fixpix = no
oversca = no
trim = yes
zerocor = yes
darkcor = no
flatcor = no
trimsec = [26:2065,7:2044]
zero = M_bias
```

The change in an example file and its bias-corrected counterpart can be seen below:



Flat Normalisation

- i) Combination of the flat images

We take our bias-corrected flat files, and use the task **flatcombine** to combine them:

PyRAF Parameter Editor: ccdred.flatcombine

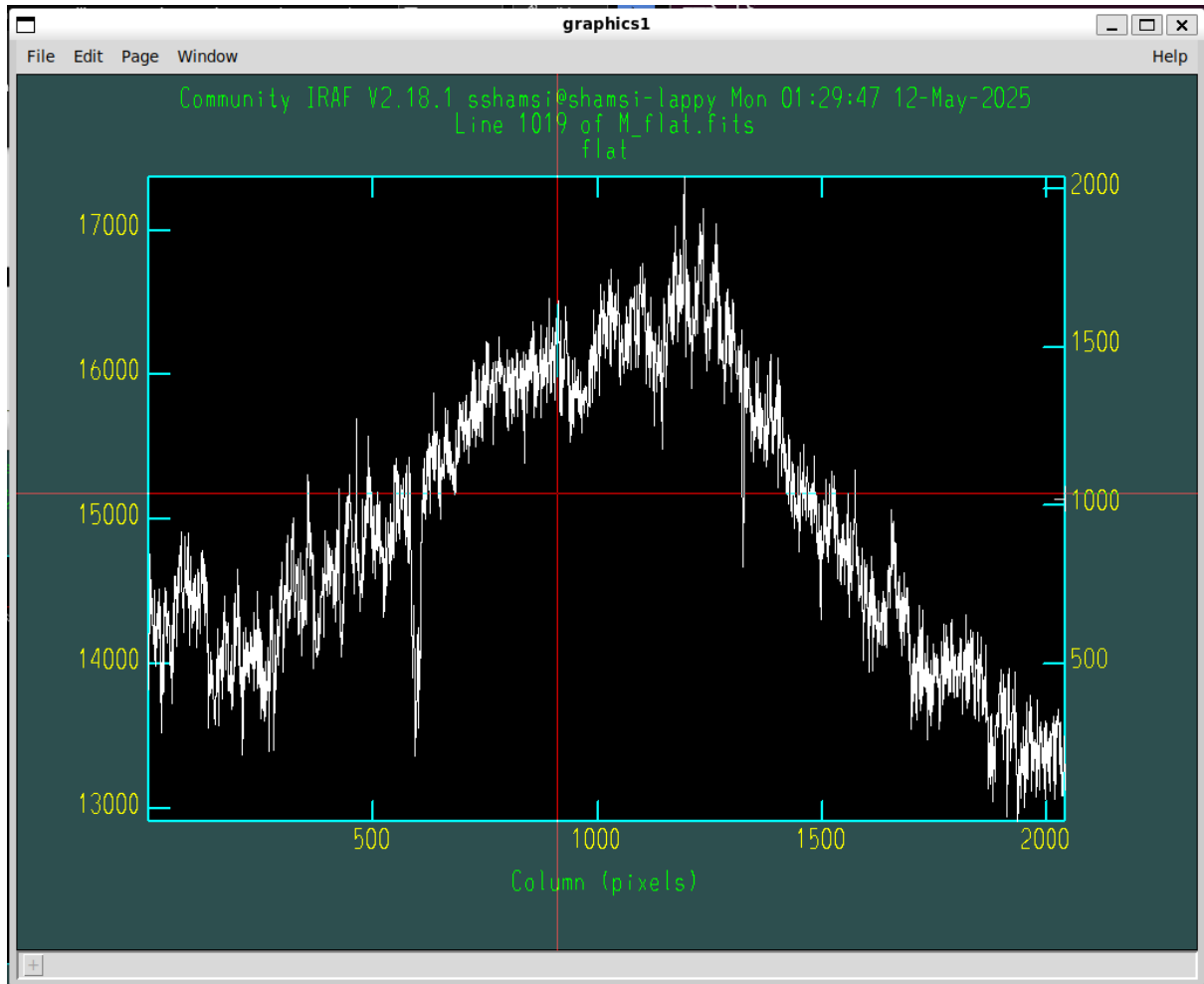
File Options Help

Package = CCDRED
Task = FLATCOMBINE

Execute Save & Quit Unlearn Cancel Flatcombine Help

input	@flats_biascor.txt	List of flat field images to combine
(output)	M_flat	Output flat field root name
(combine)	average	Type of combine operation
(reject)	crreject	Type of rejection
(ccdtype)		CCD image type to combine
(process)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Process images before combining?
(subsets)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Combine images by subset parameter?
(delete)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Delete input images after combining?
(clobber)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Clobber existing output image?
(scale)	mode	Image scaling
(statsec)		Image section for computing statistics
(nlow)	1	minmax: Number of low pixels to reject
(nhigh)	1	minmax: Number of high pixels to reject
(nkeep)	1	Minimum to keep (pos) or maximum to reject (neg)
(mclip)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Use median in sigma clipping algorithms?
(lsigma)	3.0	Lower sigma clipping factor
(hsigma)	3.0	Upper sigma clipping factor
(rdnoise)	rdnoise	ccdclip: CCD readout noise (electrons)
(gain)	gain	ccdclip: CCD gain (electrons/DN)
(snoise)	0.	ccdclip: Sensitivity noise (fraction)
(pclip)	-0.5	pclip: Percentile clipping parameter
(blank)	1.0	Value if there are no pixels
(mode)	al	

This produces a **M_flat** file for us to do further corrections. It looks like the following:



ii) Fitting the flat lamp spectrum

We use the response task to model the flat lamp's spectral response at high frequencies, so that only low frequency variations remain.

PyRAF Parameter Editor: kpnoslit.response

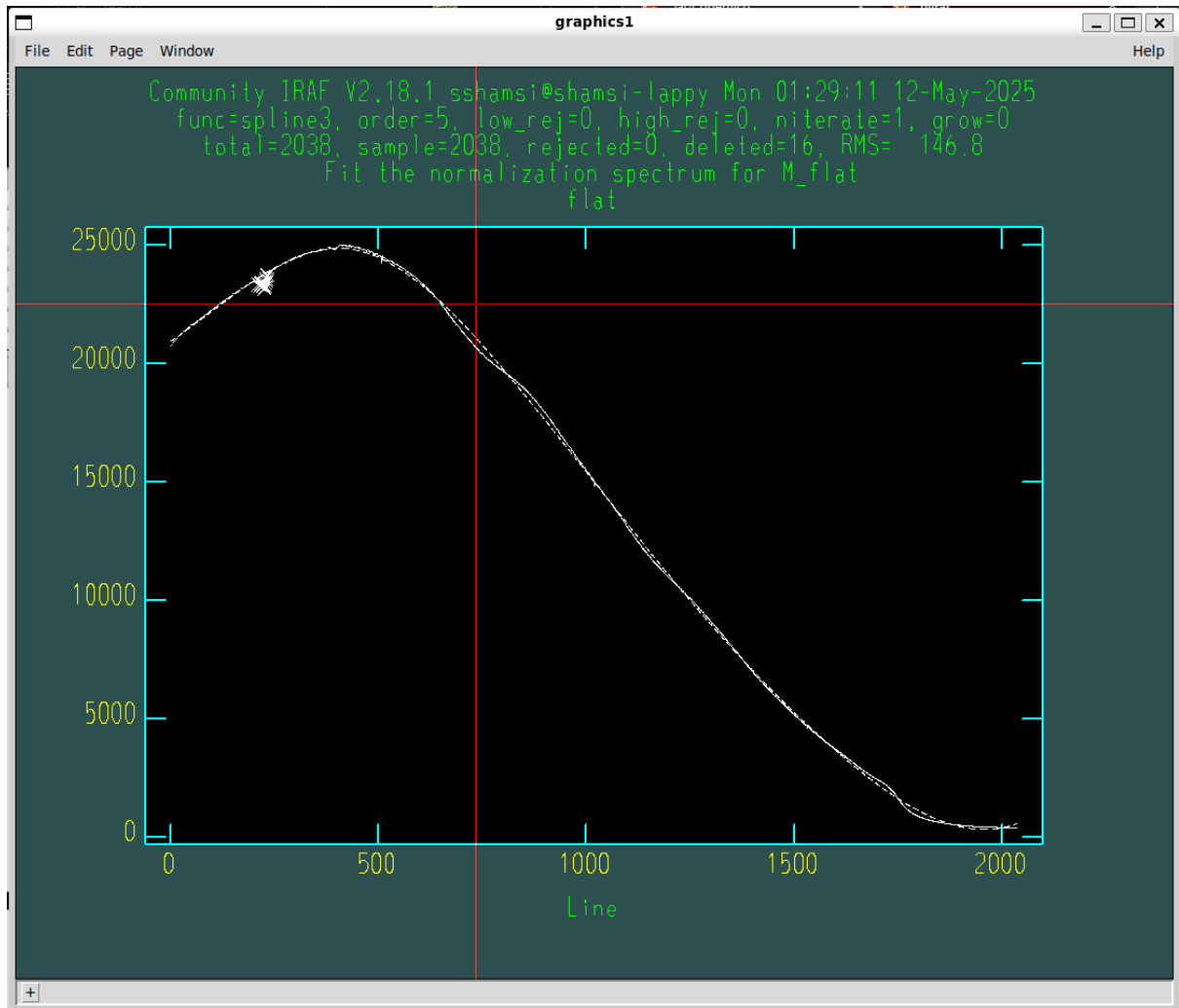
File Options Help

Package = KPNOSLIT
Task = RESPONSE

Execute Save & Quit Unlearn Cancel Response Help

calibration	M_flat	Longslit calibration images
normalization	M_flat	Normalization spectrum images
response	flat_response	Response function images
(interactive)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Fit normalization spectrum interactively?
(threshold)	INDEF	Response threshold
(sample)	*	Sample of points to use in fit
(naverage)	1	Number of points in sample averaging
(function)	spline3	Fitting function
(order)	3	Order of fitting function
(low_reject)	0.0	Low rejection in sigma of fit
(high_reject)	0.0	High rejection in sigma of fit
(niterate)	1	Number of rejection iterations
(grow)	0.0	Rejection growing radius
(graphics)	stdgraph	Graphics output device
(cursor)		Graphics cursor input
(mode)	al	

We interactively edit the **spline3** function, removing some datapoint deviations and editing the order of the fit. We find that the order 3 best represents the response function without resorting to frequencies which were too high.



iii) Applying the flat correction

We once again use the **ccdproc** task, this time applying the flat correction to all of our bias corrected files using the **flat_response** file.

PyRAF Parameter Editor: ccdred.ccdproc

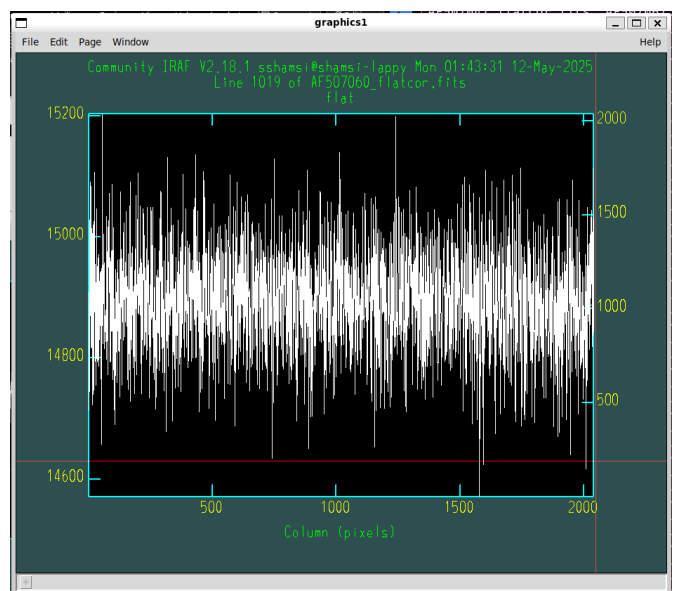
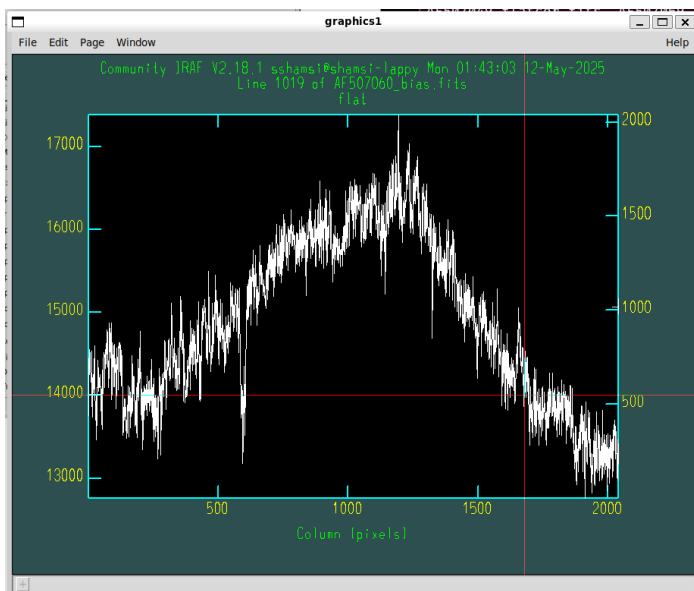
File Options Help

Package = CCDRED
Task = CCDPROC

Execute Save & Quit Unlearn Cancel Ccdproc Help

images	@all_bias.txt	List of CCD images to correct
(output)	@all_flatcor.txt	List of output CCD images
(ccdtype)		CCD image type to correct
(max_cache)	0	Maximum image caching memory (in Mbytes)
(noprocs)	<input type="radio"/> Yes <input checked="" type="radio"/> No	List processing steps only?
(fixpix)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Fix bad CCD lines and columns?
(overscan)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Apply overscan strip correction?
(trim)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Trim the image?
(zerocor)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Apply zero level correction?
(darkcor)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Apply dark count correction?
(flatcor)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Apply flat field correction?
(illumcor)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Apply illumination correction?
(fringeor)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Apply fringe correction?
(readcor)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Convert zero level image to readout correction?
(scancor)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Convert flat field image to scan correction?
(readaxis)	line	Read out axis (column line)
(fixfile)		File describing the bad lines and columns
(biassec)		Overscan strip image section
(trimsec)		Trim data section

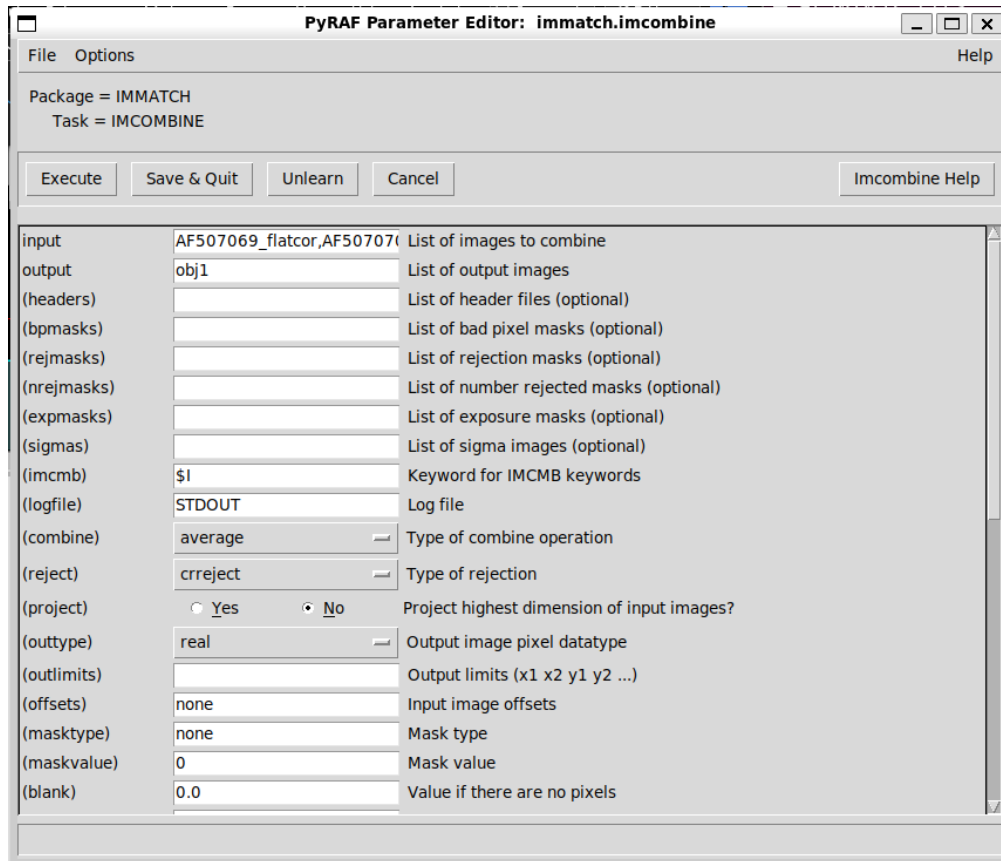
The effect of the flat correction can be seen in the following images, before and after the correction:



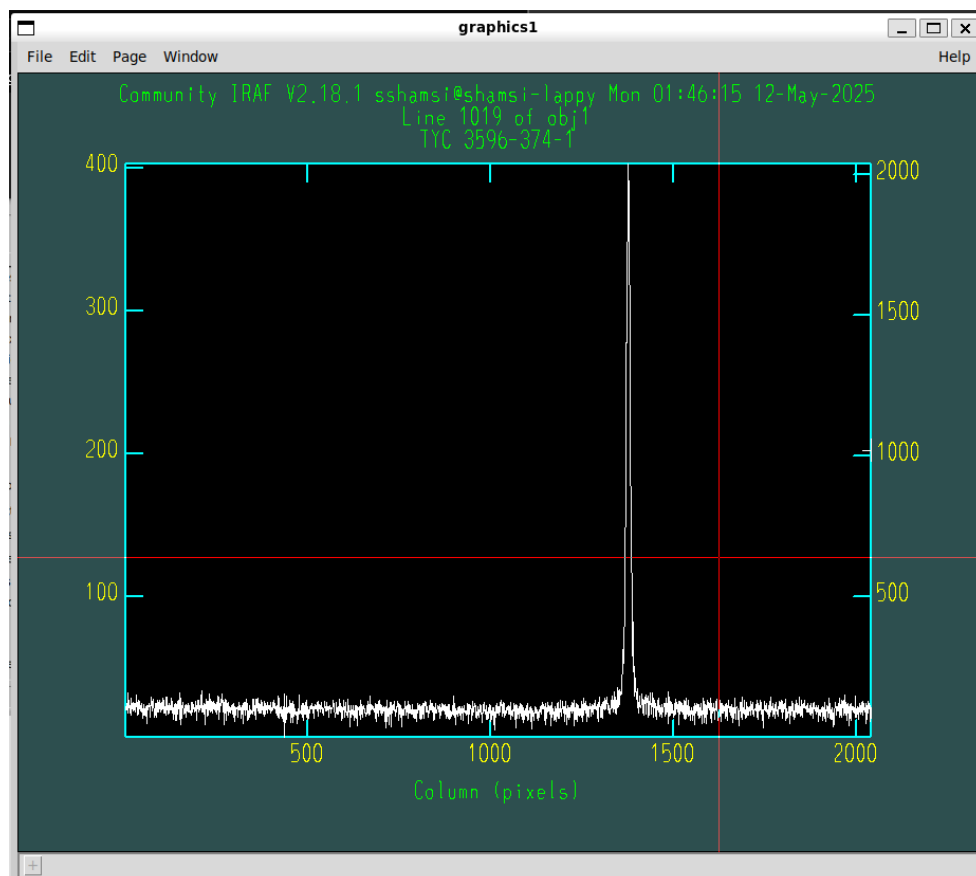
5.2) Spectrum Extraction

i) Image Stacking

We use the **imcombine** task to combine the bias and flat corrected images for each of our objects. **Copernico_coded.txt** tells us which images are associated with which target files.



The stacked **obj1** image plotted with **implot** looks like:

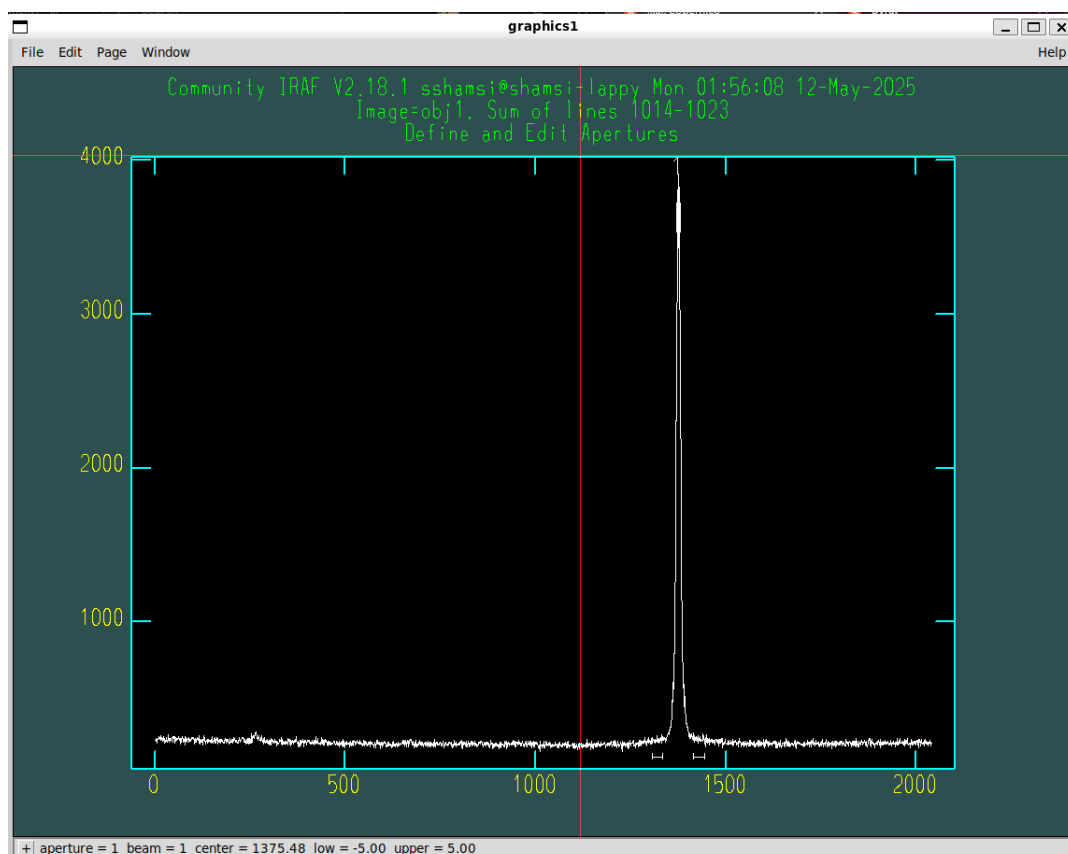


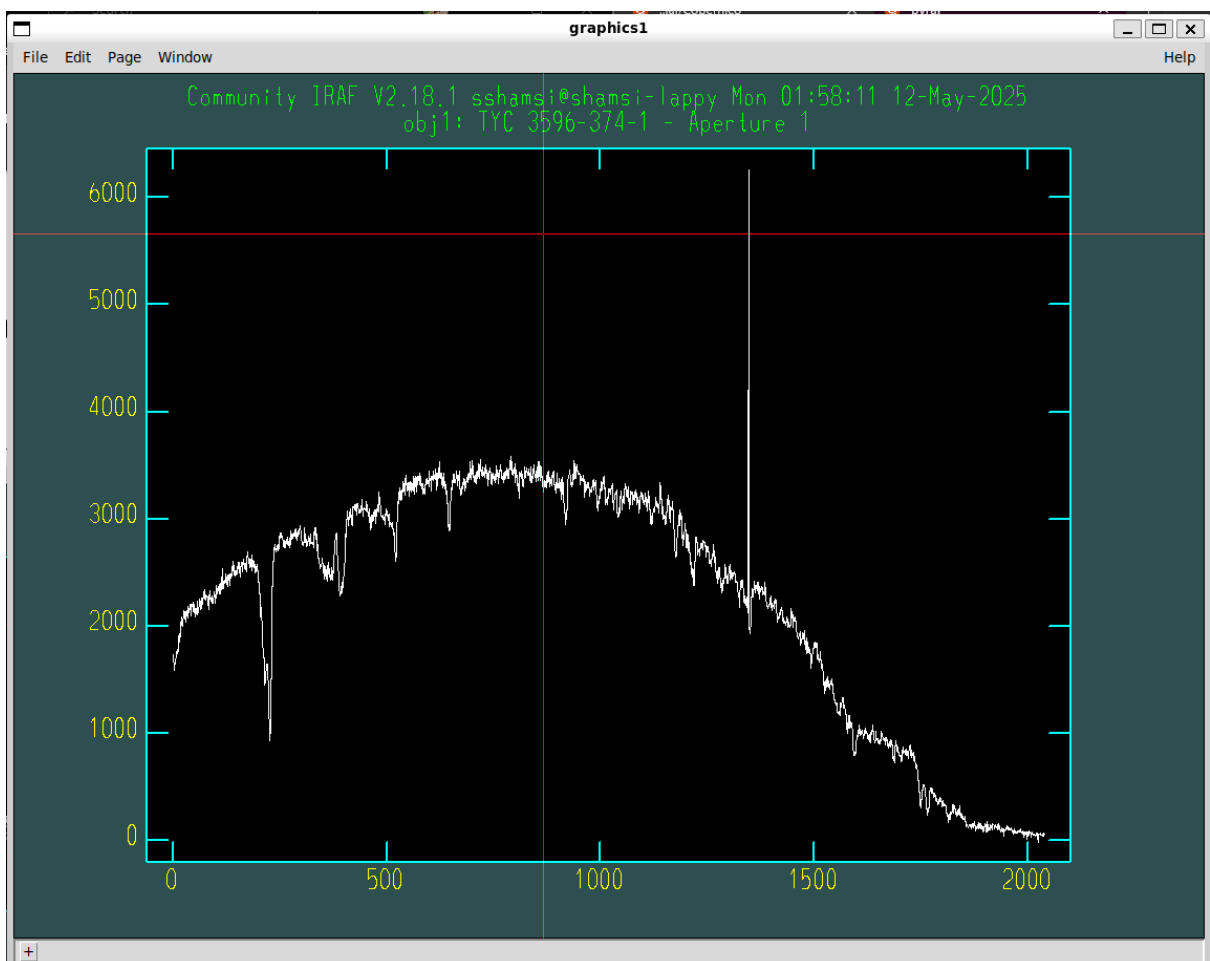
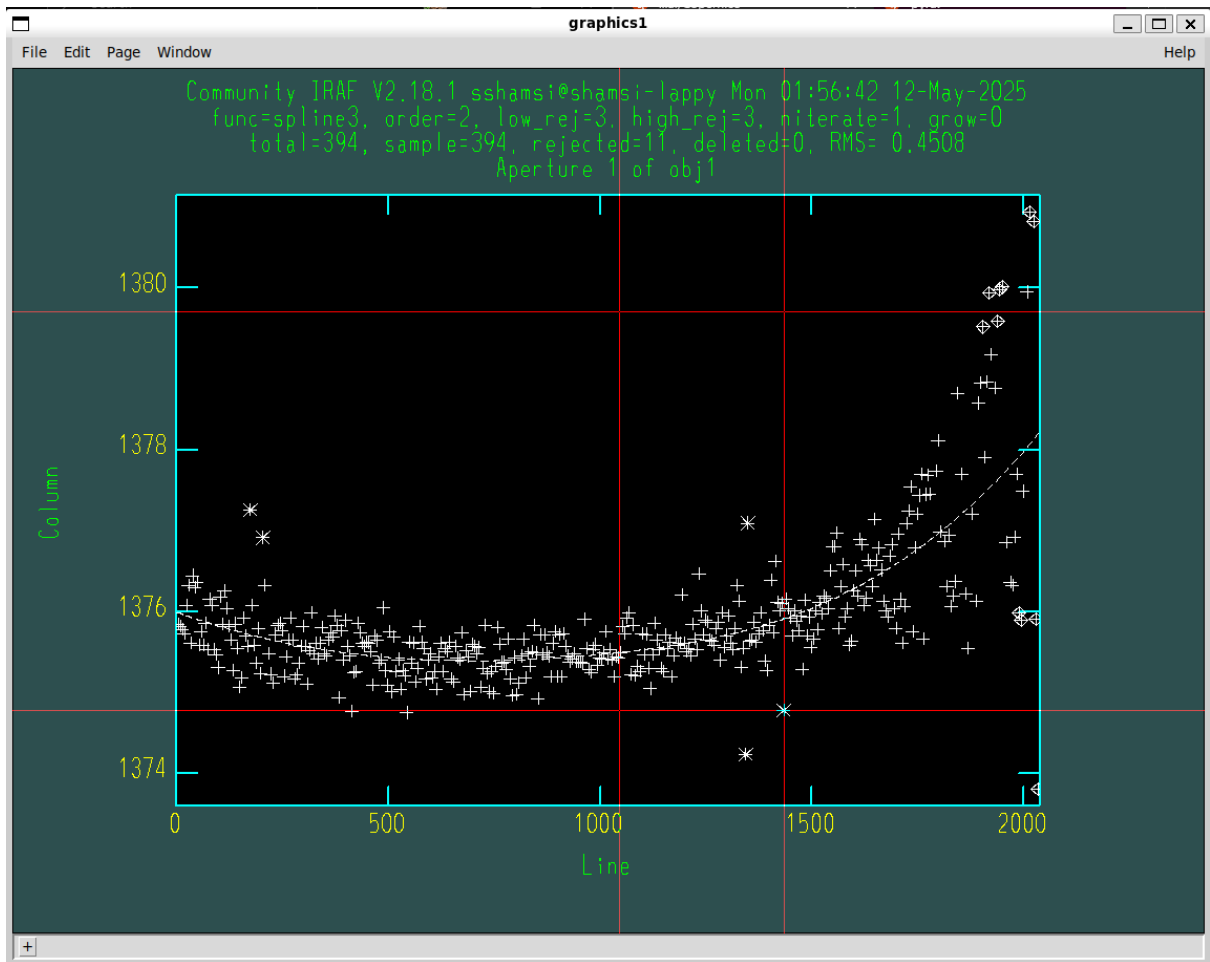
ii) Spectrum extraction

We then use the **apall** task to extract the spectrum, obtaining the following aperture, trace, and extracted spectrum files:

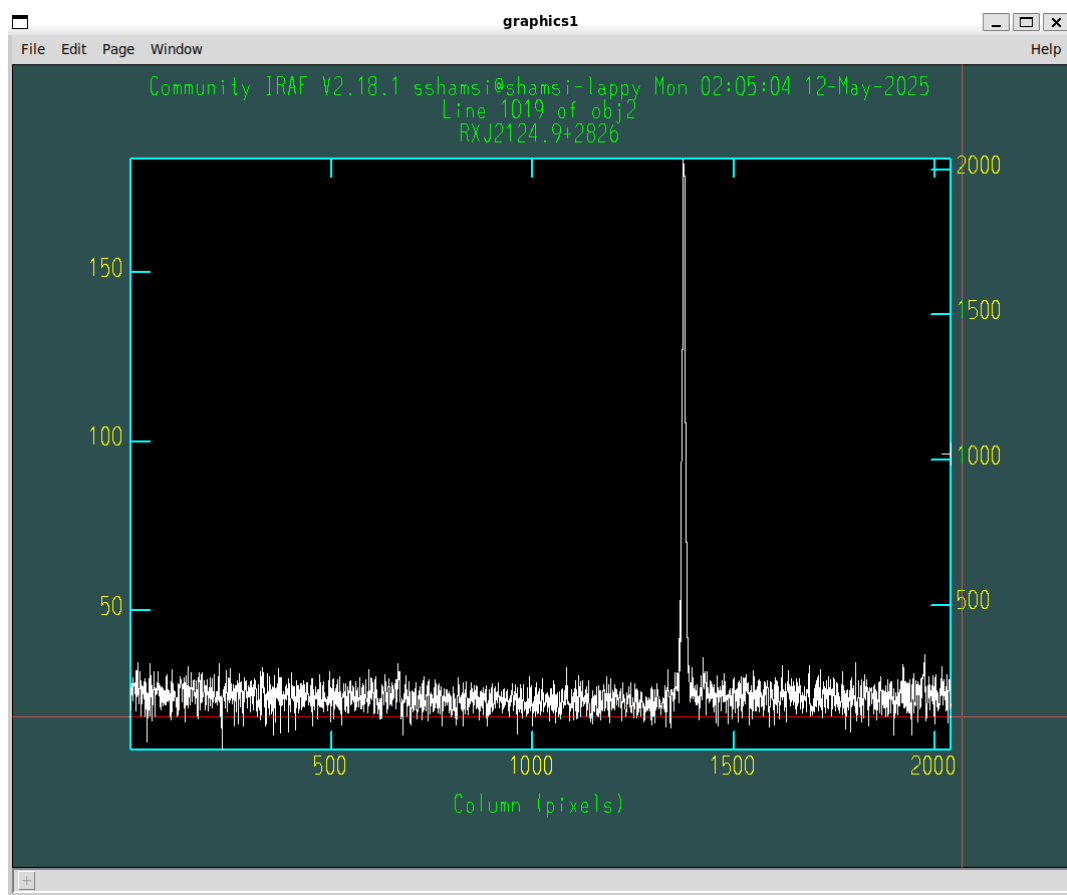
```
TASK : apall
PACKAGE : noao -> imred -> kpnoslit

input = obj1
output = obj1
recente = yes
resize = no
edit = yes
trace = yes
b_funct = chebyshev
b_order = 2
b_sampl = -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
```

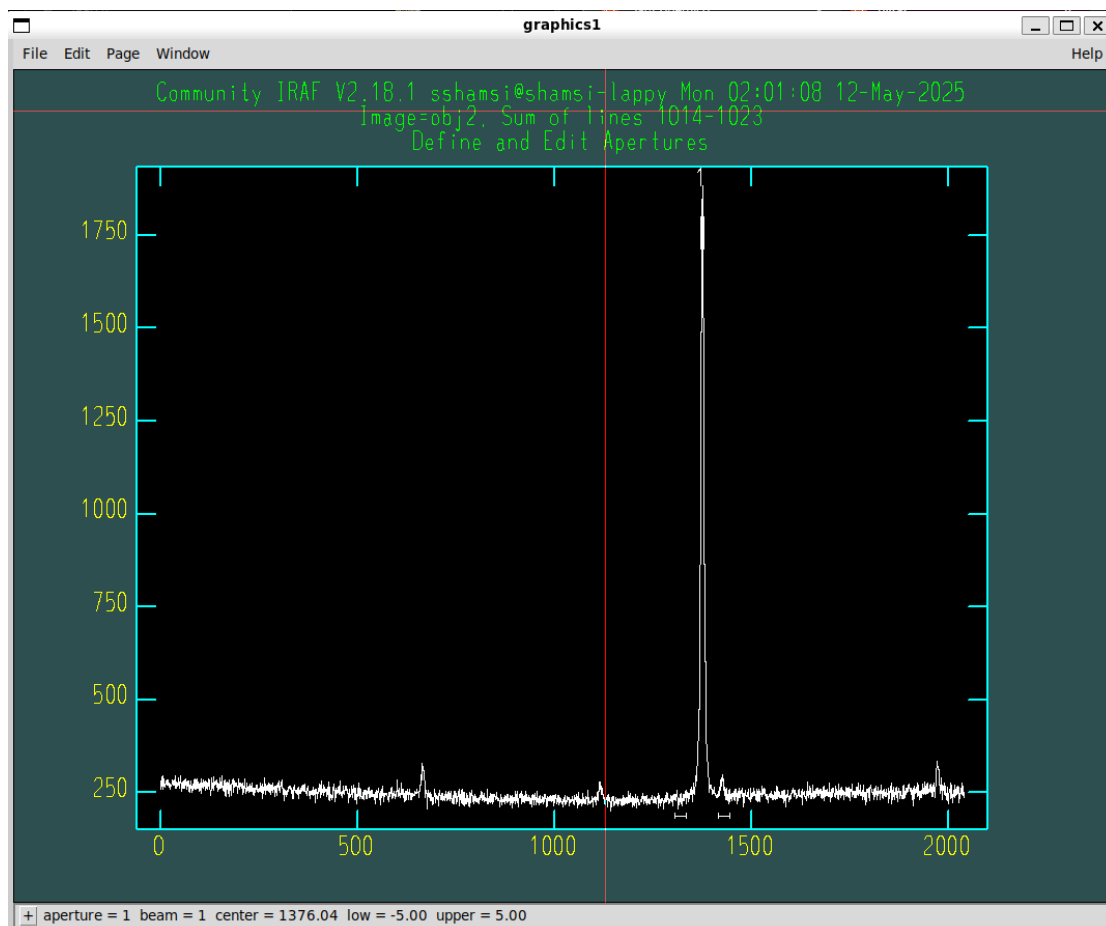


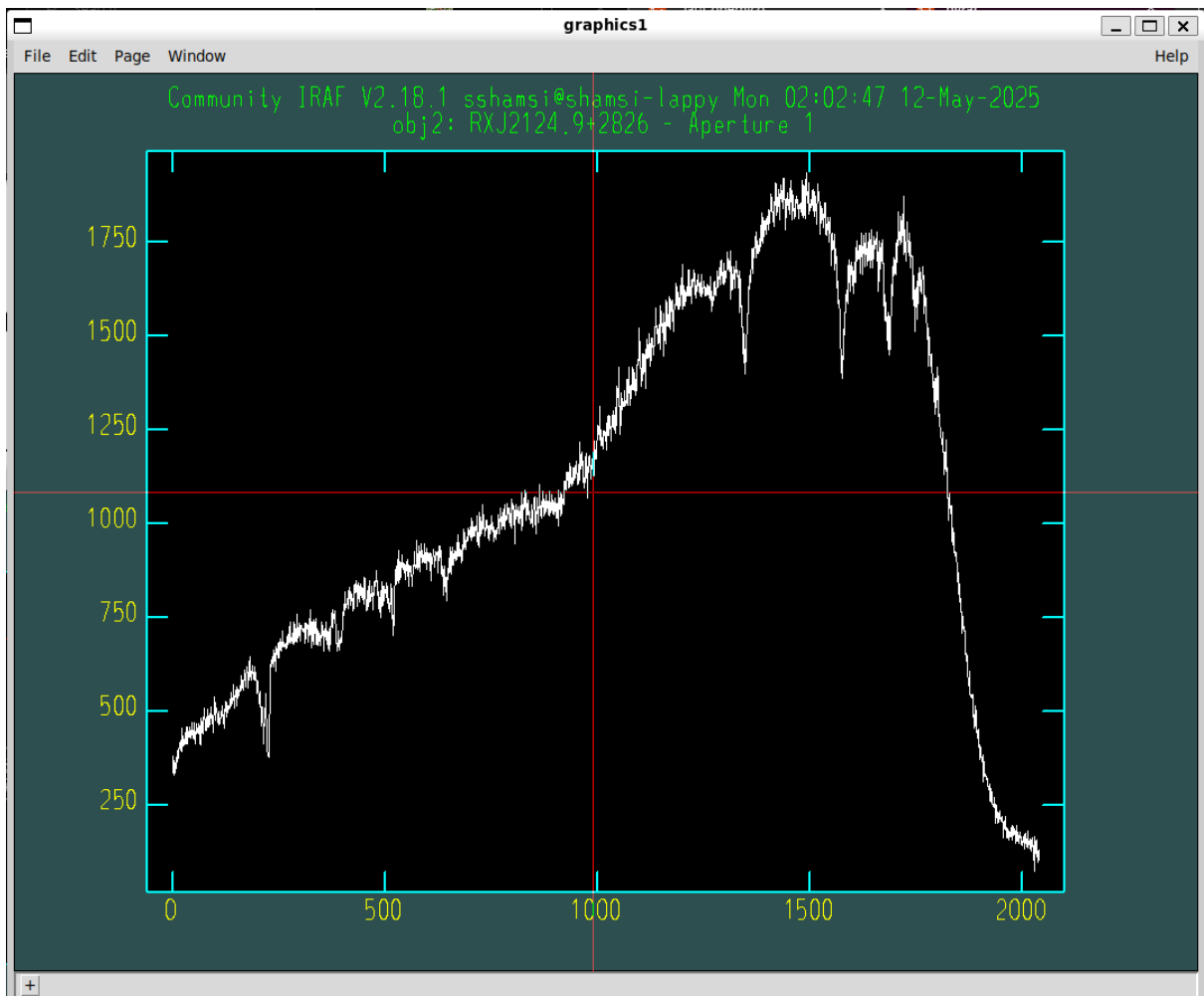
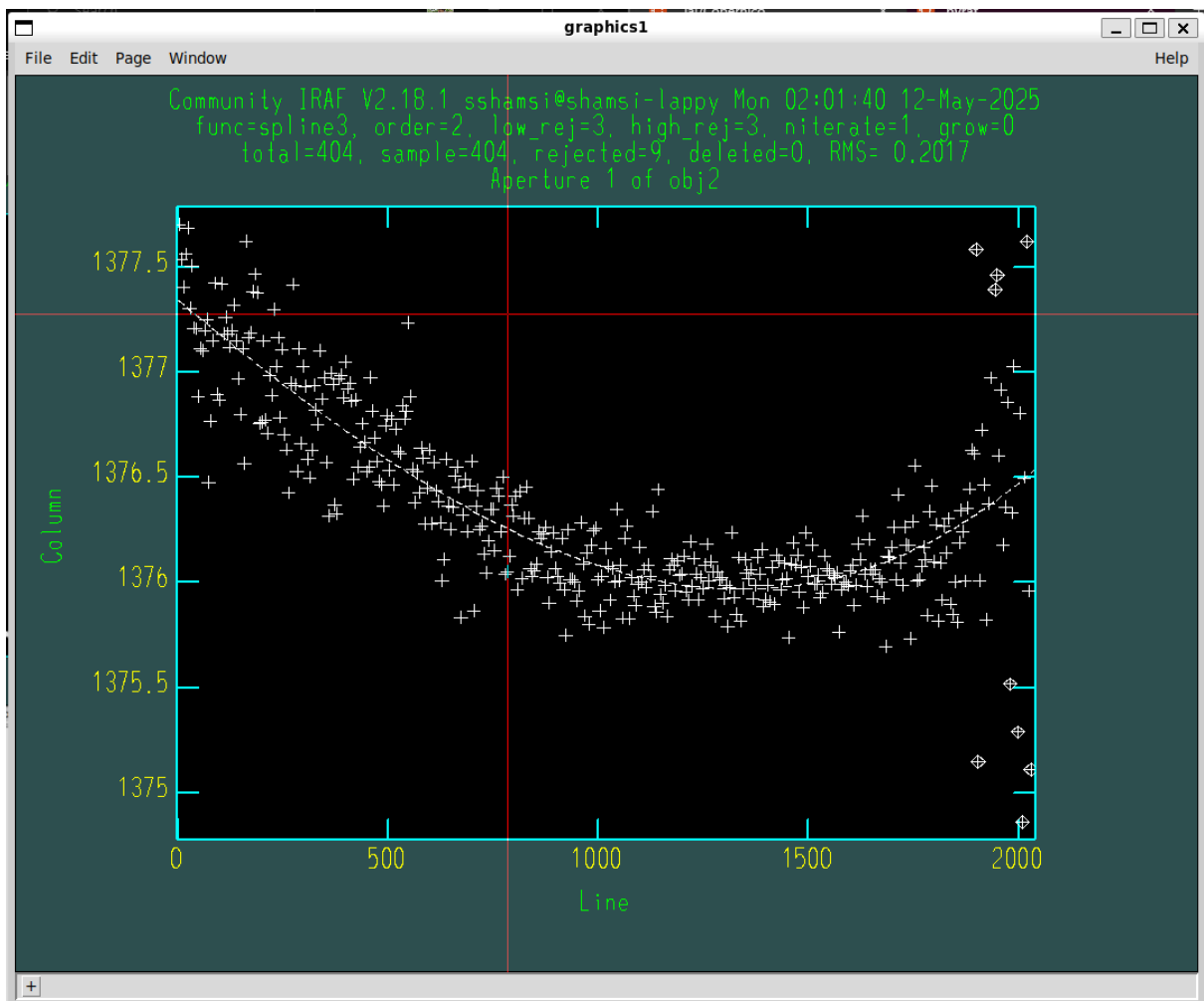


And we'll do the same for **obj2**. The stacked **obj2** image looks like:



The aperture, trace, and extracted spectrum looks like:





5.3 Wavelength Solution

Processing emission lamp spectra

i) Extraction of lamp spectra

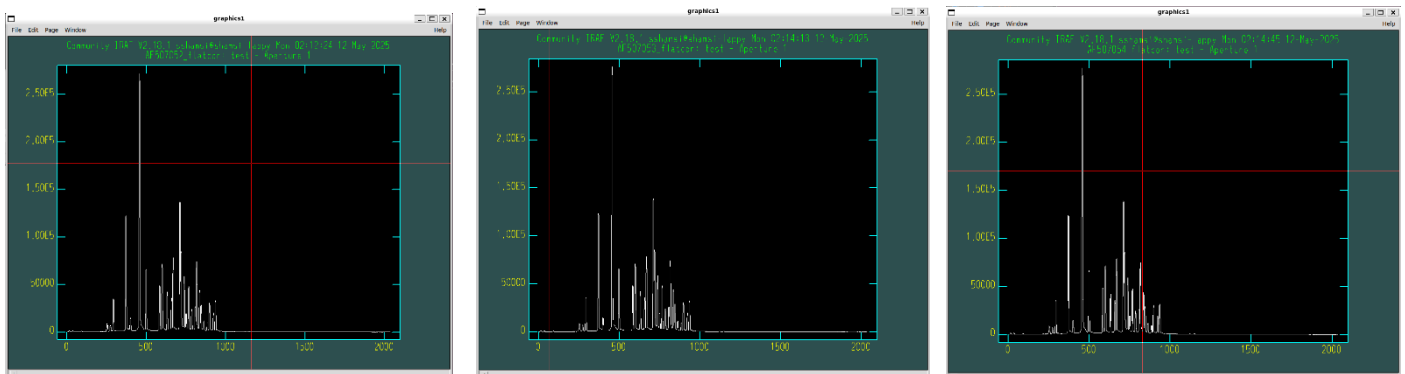
We use the **apall** task to extract the spectra from each of the Ne lamps.

The image shows the PyRAF Parameter Editor window for the task 'kpnoslit.apall'. The window has a menu bar with 'File', 'Options', and 'Help'. Below the menu bar, it displays 'Package = KPNOSLIT' and 'Task = APALL'. There are four buttons: 'Execute', 'Save & Quit', 'Unlearn', and 'Cancel', and an 'Apall Help' button. The main area contains a list of parameters and their values:

Parameter	Value	Description
input	AF507052_flatcor,AF507053	List of input images
(output)	AF507052_spec,AF507053	List of output spectra
(apertures)		Apertures
(format)	multispec	Extracted spectra format
(references)	obj1	List of aperture reference images
(profiles)		List of aperture profile images
(interactive)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Run task interactively?
(find)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Find apertures?
(recenter)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Recenter apertures?
(resize)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Resize apertures?
(edit)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Edit apertures?
(trace)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Trace apertures?
(fittrace)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Fit the traced points interactively?
(extract)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Extract spectra?
(extras)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Extract sky, sigma, etc.?
(review)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Review extractions?
(line)	INDEF	Dispersion line
(nsum)	10	Number of dispersion lines to sum or median

Below the list, there is a section for 'DEFAULT APERTURE PARAMETERS'.

This yields the following spectra files:



And we repeat the same procedure for the HgCd files:

PyRAF Parameter Editor: kpnoslit.apall

File Options Help

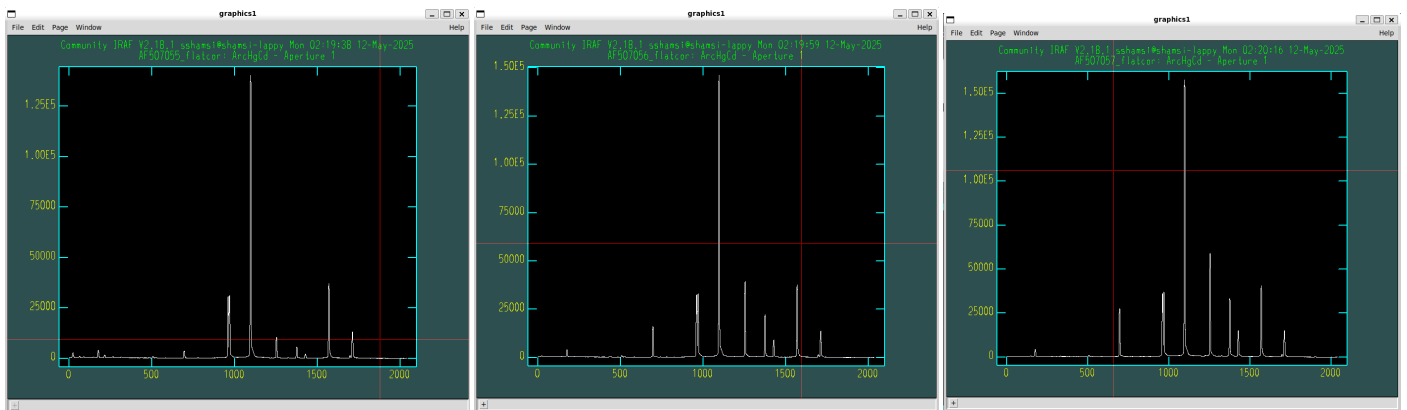
Package = KPNOSLIT
Task = APALL

Execute Save & Quit Unlearn Cancel Apall Help

input	AF507055_flatcor,AF507055_	List of input images
(output)	AF507055_spec,AF507056_	List of output spectra
(apertures)		Apertures
(format)	multispec	Extracted spectra format
(references)	obj1	List of aperture reference images
(profiles)		List of aperture profile images
(interactive)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Run task interactively?
(find)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Find apertures?
(recenter)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Recenter apertures?
(resize)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Resize apertures?
(edit)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Edit apertures?
(trace)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Trace apertures?
(fittrace)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Fit the traced points interactively?
(extract)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Extract spectra?
(extras)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Extract sky, sigma, etc.?
(review)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Review extractions?
(line)	INDEF	Dispersion line
(nsum)	10	Number of dispersion lines to sum or median

DEFAULT APERTURE PARAMETERS

Which yields the following spectra files:



ii) Combining lamp spectra

We then combine all these spectra into a single **lamps_combined** file with the **scombine** task.

PyRAF Parameter Editor: specred.scombine

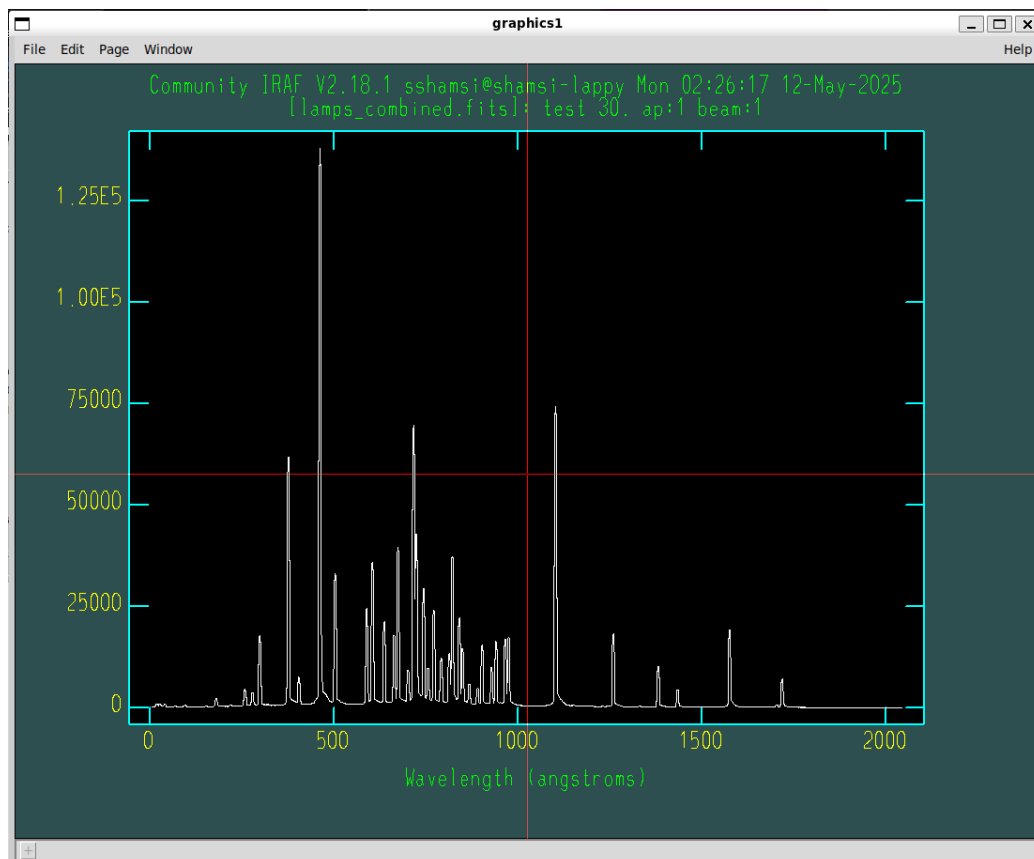
File Options Help

Package = SPECRED
Task = SCOMBINE

Execute Save & Quit Unlearn Cancel Scombine Help

input	@lamp_specs.txt	List of input spectra
output	lamps_combined	List of output spectra
(noutput)		List of output number combined spectra
(logfile)	STDOUT	Log file
(apertures)		Apertures to combine
(group)	apertures	Grouping option
(combine)	average	Type of combine operation
(reject)	none	Type of rejection
(first)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Use first spectrum for dispersion?
(w1)	INDEF	Starting wavelength of output spectra
(w2)	INDEF	Ending wavelength of output spectra
(dw)	INDEF	Wavelength increment of output spectra
(nw)	INDEF	Length of output spectra
(log)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Logarithmic increments?
(scale)	none	Image scaling
(zero)	none	Image zero point offset
(weight)	none	Image weights
(sample)		Wavelength sample regions for statistics
(lthreshold)	INDEF	Lower threshold

The combined file then looks like:



iii) Line identification and wavelength fitting:

Now we can use the identify task to identify and mark lines. We make sure to identify a healthy number of lines (~13) across the wavelength range, using the Ne and MgCd emission lines presented in the task.

PyRAF Parameter Editor: specred.identify

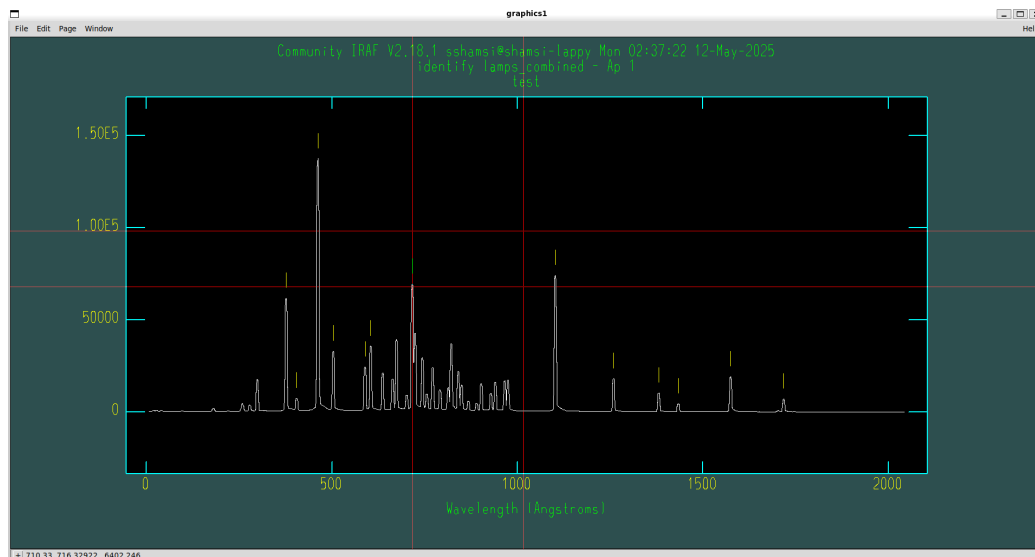
File Options Help

Package = SPECRED
Task = IDENTIFY

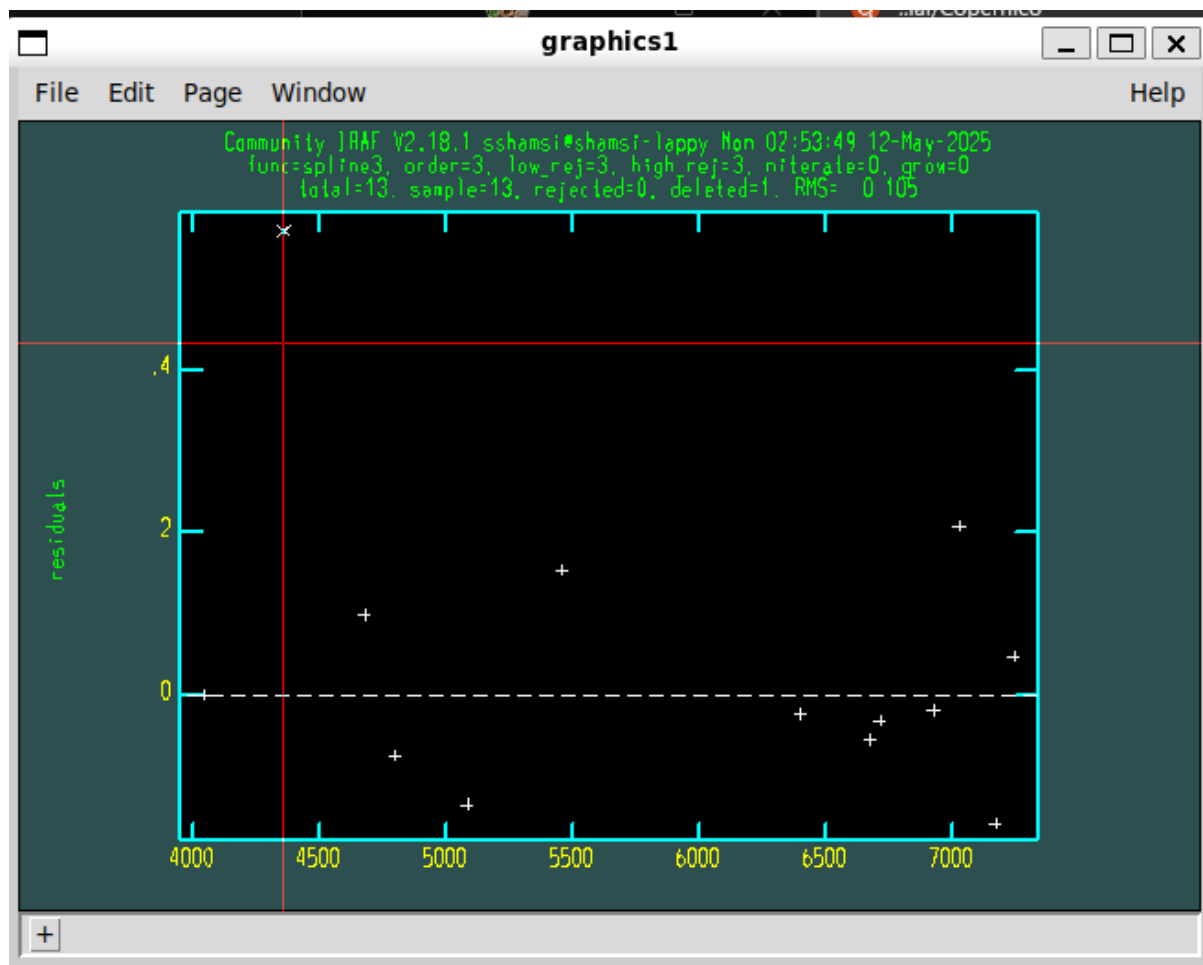
Execute Save & Quit Unlearn Cancel Identify Help

images	lamps_combined	Images containing features to be identified
(section)	middle line	Section to apply to two dimensional images
(database)	database	Database in which to record feature data
(coordlist)		User coordinate list
(units)		Coordinate units
(nsum)	10	Number of lines/columns/bands to sum in 2D images
(match)	-3.0	Coordinate list matching limit
(maxfeatures)	50	Maximum number of features for automatic identification
(zwidth)	100.0	Zoom graph width in user units
(ftype)	emission	Feature type
(fwidth)	7.5	Feature width in pixels
(cradius)	5.0	Centering radius in pixels
(threshold)	10	Feature threshold for centering
(minsep)	2.0	Minimum pixel separation
(function)	spline3	Coordinate function
(order)	3	Order of coordinate function
(sample)	*	Coordinate sample regions
(niterate)	0	Rejection iterations
(low_reject)	3.0	Lower rejection sigma
(high_reject)	3.0	Upper rejection sigma

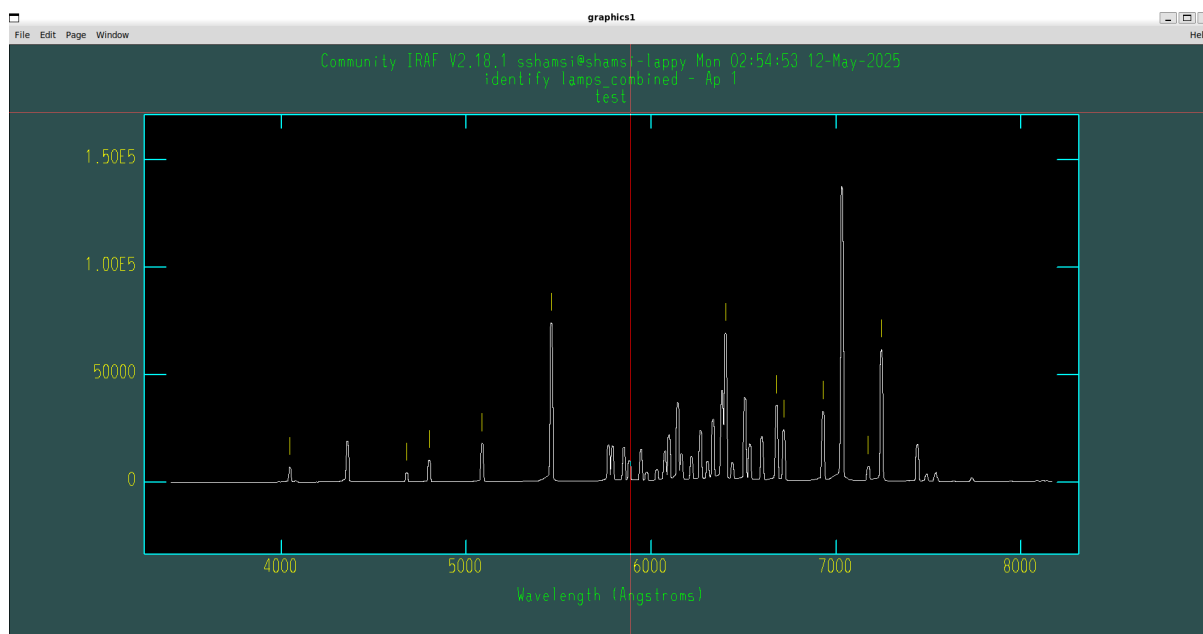
The identified lines are shown below. We note that the **lamps_combined** file is initially plotted with the wavelength on the x-axis in a descending order. We take care to flip our reference images before identifying the lines.



The following file shows the residuals after fitting. We remove one point which had a higher than usual residual.



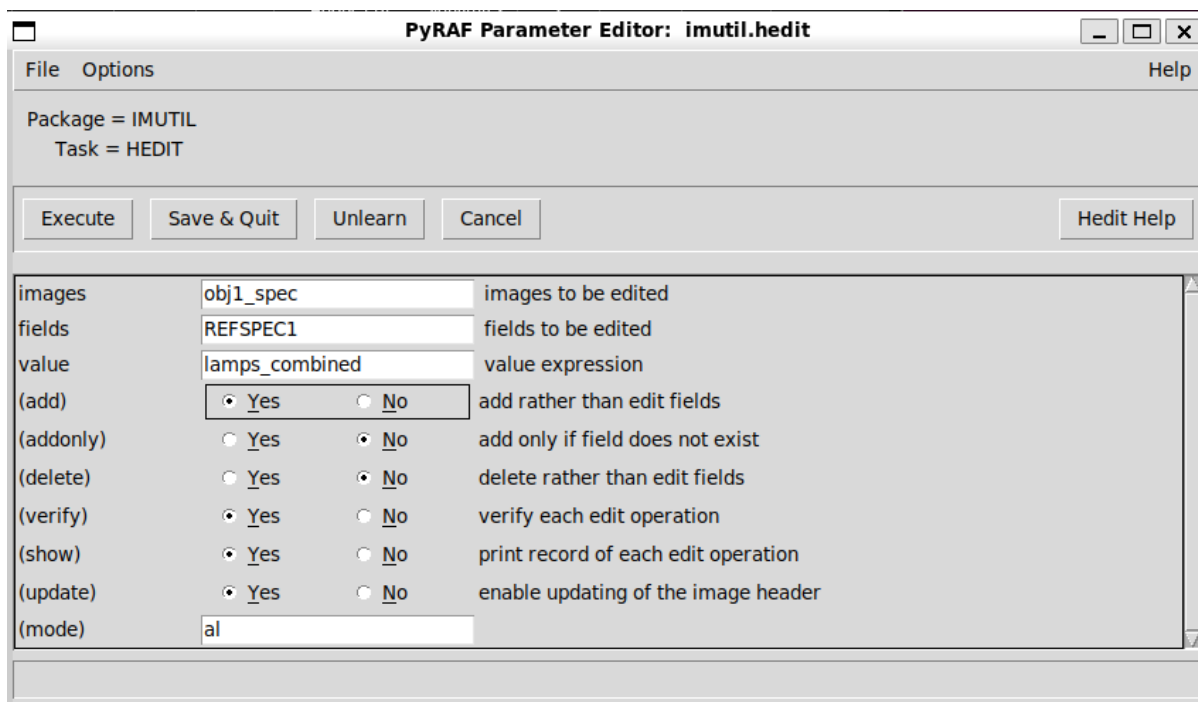
After identification, the **lamps_combined** file is flipped the right way around. We save this.



Wavelength mapping in science spectra

i) Editing file header

It is now time to apply this spectrum reference to our **obj1** spectrum file.

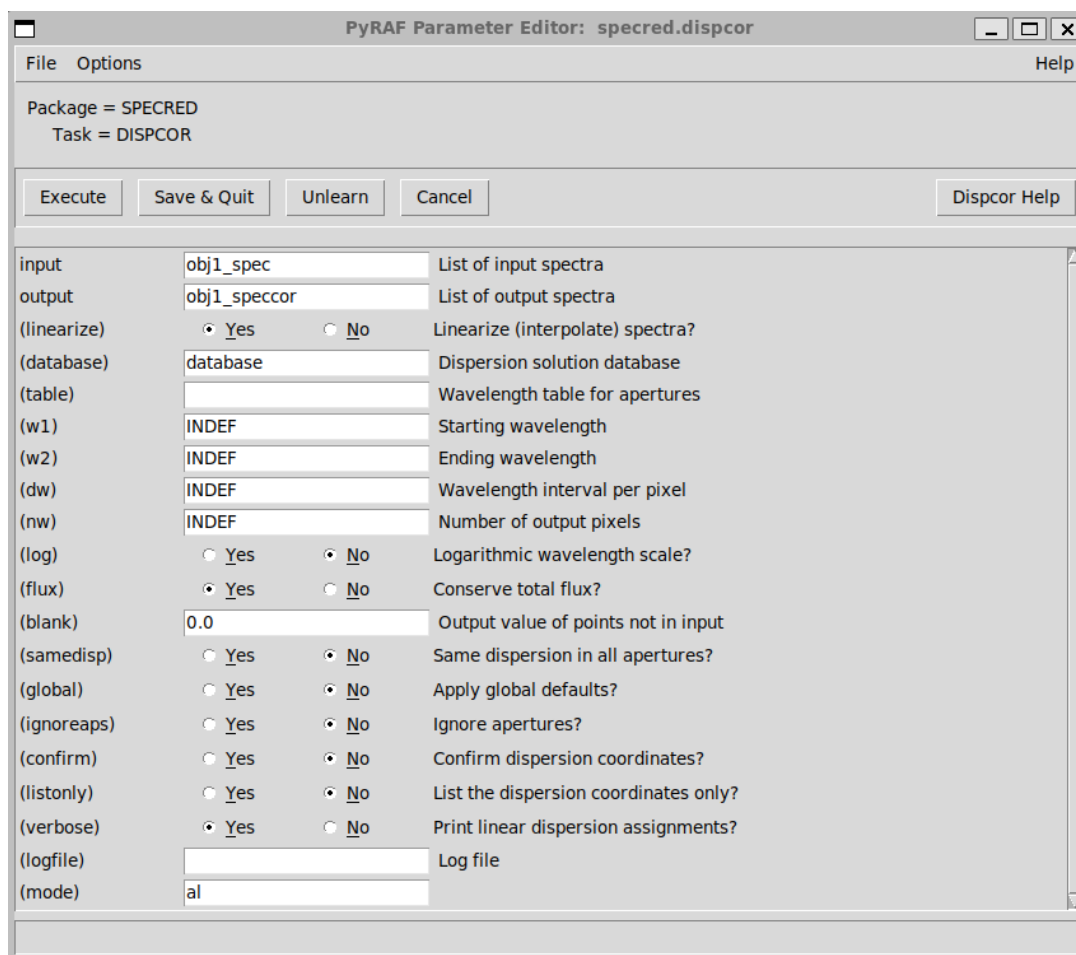


The image shows the 'PyRAF Parameter Editor: imutil.hedit' window. It has a menu bar with 'File', 'Options', and 'Help'. Below the menu bar, it displays 'Package = IMUTIL' and 'Task = HEDIT'. There are four buttons: 'Execute', 'Save & Quit', 'Unlearn', and 'Cancel', and a 'Hedit Help' button. The main area contains a list of parameters with their values and descriptions:

Parameter	Value	Description
images	obj1_spec	images to be edited
fields	REFSPEC1	fields to be edited
value	lamps_combined	value expression
(add)	<input checked="" type="radio"/> Yes <input type="radio"/> No	add rather than edit fields
(addonly)	<input type="radio"/> Yes <input checked="" type="radio"/> No	add only if field does not exist
(delete)	<input type="radio"/> Yes <input checked="" type="radio"/> No	delete rather than edit fields
(verify)	<input checked="" type="radio"/> Yes <input type="radio"/> No	verify each edit operation
(show)	<input checked="" type="radio"/> Yes <input type="radio"/> No	print record of each edit operation
(update)	<input checked="" type="radio"/> Yes <input type="radio"/> No	enable updating of the image header
(mode)	al	

ii) Apply the wavelength solution

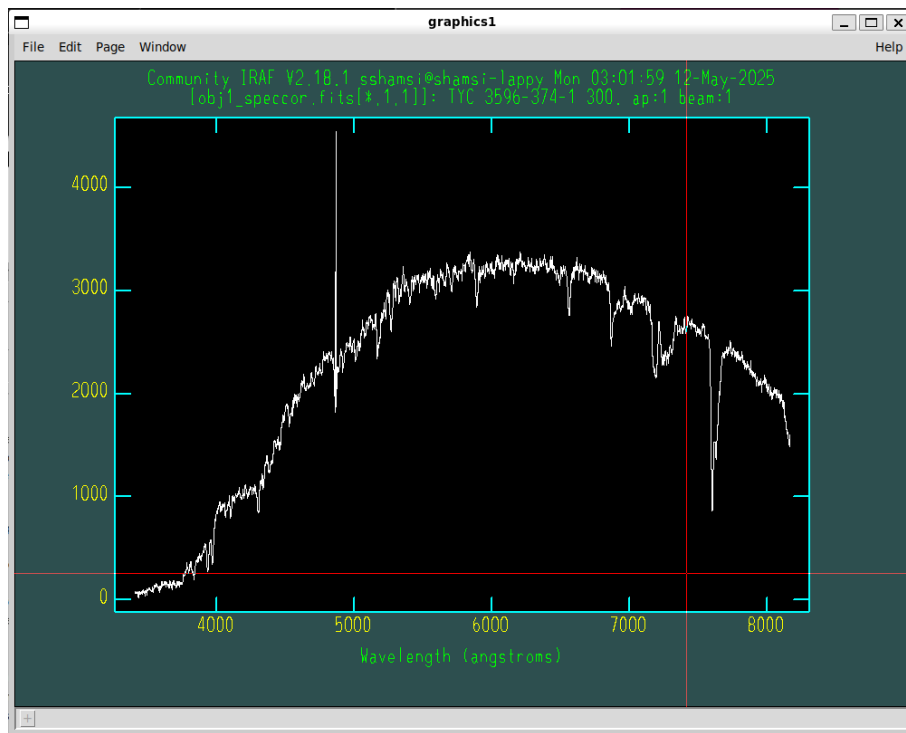
With the information in the header, we can now apply the wavelength solution using the **dispcor** task.



The image shows the 'PyRAF Parameter Editor: specred.dispcor' window. It has a menu bar with 'File', 'Options', and 'Help'. Below the menu bar, it displays 'Package = SPECRED' and 'Task = DISPCOR'. There are four buttons: 'Execute', 'Save & Quit', 'Unlearn', and 'Cancel', and a 'Dispcor Help' button. The main area contains a list of parameters with their values and descriptions:

Parameter	Value	Description
input	obj1_spec	List of input spectra
output	obj1_speccor	List of output spectra
(linearize)	<input checked="" type="radio"/> Yes <input type="radio"/> 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)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Logarithmic wavelength scale?
(flux)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Conserve total flux?
(blank)	0.0	Output value of points not in input
(samedisp)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Same dispersion in all apertures?
(global)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Apply global defaults?
(ignoreaps)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Ignore apertures?
(confirm)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Confirm dispersion coordinates?
(listonly)	<input type="radio"/> Yes <input checked="" type="radio"/> No	List the dispersion coordinates only?
(verbose)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Print linear dispersion assignments?
(logfile)		Log file
(mode)	al	

Finally, plotting **obj1_speccor** using **splot** gives us the following graph:



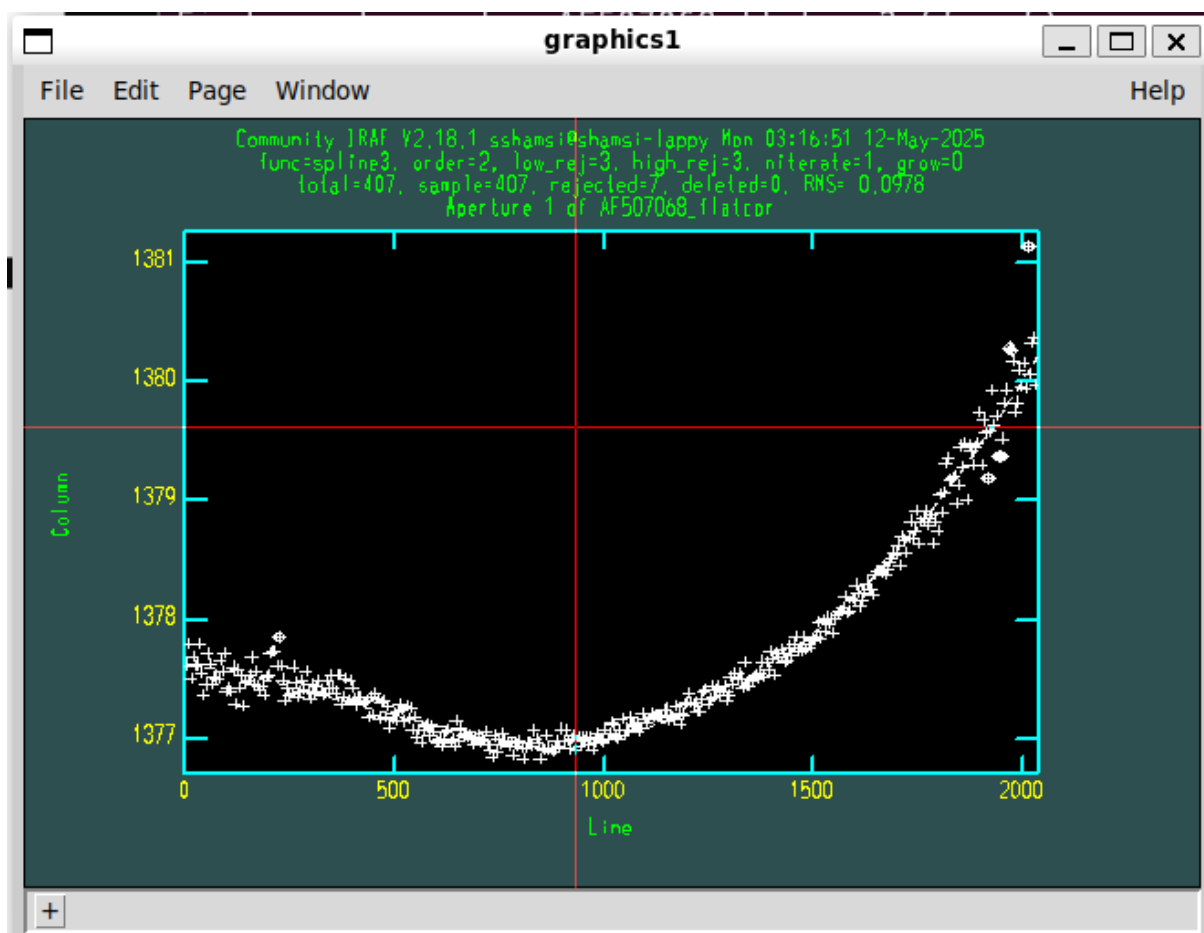
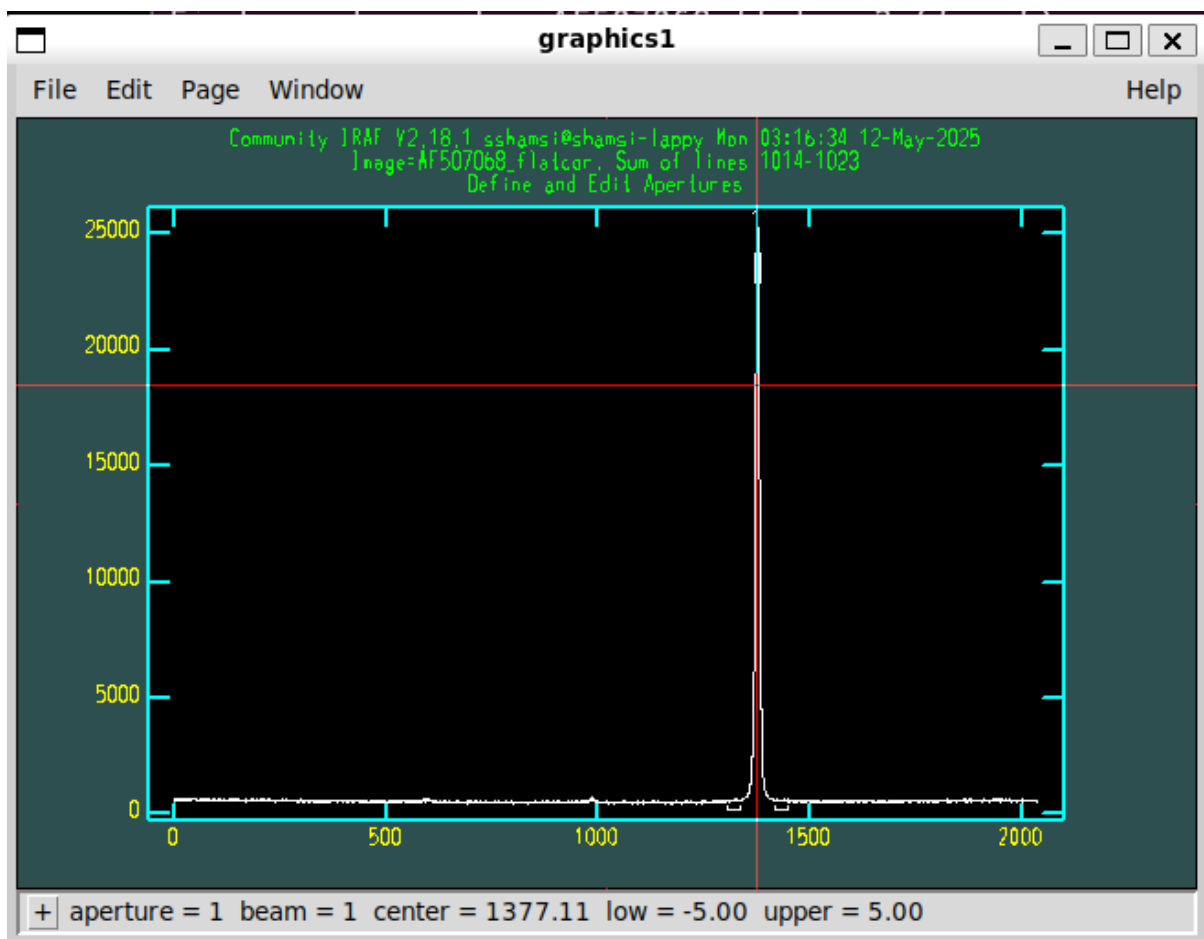
5.4) Flux Calibrations

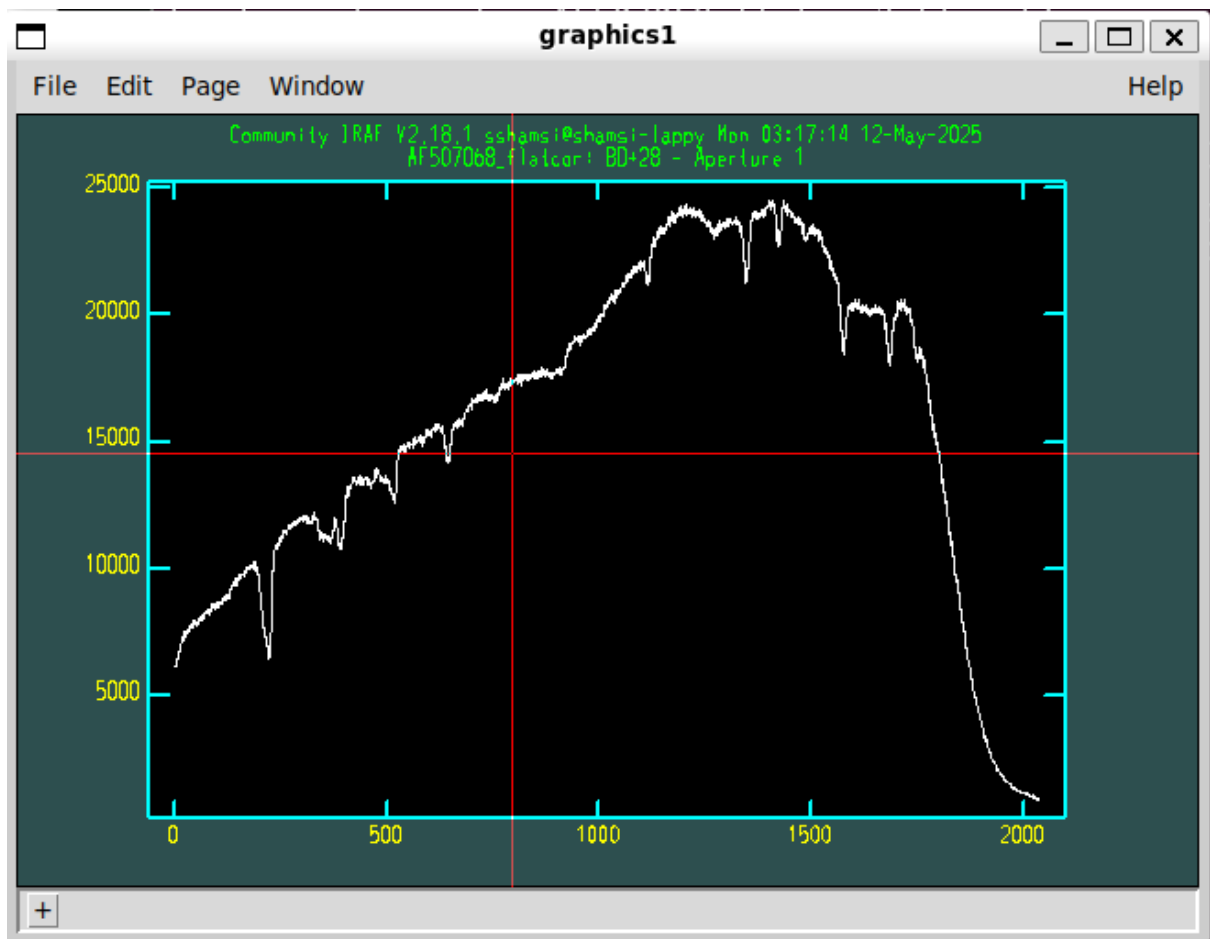
Instrumental response

Let's extract the spectrum for the calibration star BD284211 with **apall** as before.

Parameter	Value	Description
Package	SPECRED	
Task	APALL	
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)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Run task interactively?
(find)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Find apertures?
(recenter)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Recenter apertures?
(resize)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Resize apertures?
(edit)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Edit apertures?
(trace)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Trace apertures?
(fittrace)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Fit the traced points interactively?
(extract)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Extract spectra?
(extras)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Extract sky, sigma, etc.?
(review)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Review extractions?
(line)	INDEF	Dispersion line
(nsum)	10	Number of dispersion lines to sum or median

The aperture, trace and spectrum for the calibration star are as follows:





We'll add the reference spectrum information to the calibration star header:

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)	<input checked="" type="radio"/> Yes <input type="radio"/> No	add rather than edit fields
(addonly)	<input type="radio"/> Yes <input checked="" type="radio"/> No	add only if field does not exist
(delete)	<input type="radio"/> Yes <input checked="" type="radio"/> No	delete rather than edit fields
(verify)	<input checked="" type="radio"/> Yes <input type="radio"/> No	verify each edit operation
(show)	<input checked="" type="radio"/> Yes <input type="radio"/> No	print record of each edit operation
(update)	<input checked="" type="radio"/> Yes <input type="radio"/> No	enable updating of the image header
(mode)	al	

And then apply the wavelength solution to the calibration star from its header:

PyRAF Parameter Editor: specred.dispcor

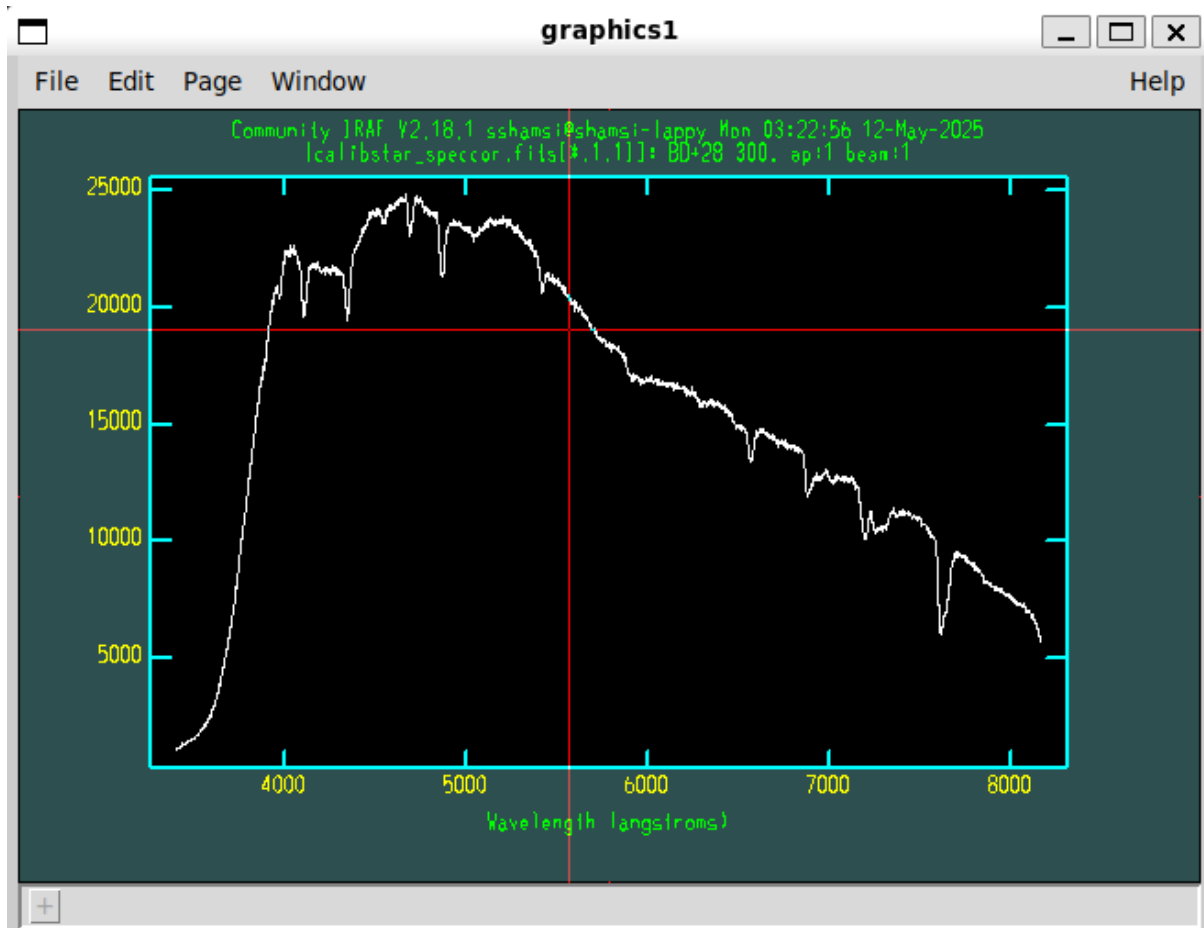
File Options Help

Package = SPECRED
Task = DISPCOR

Execute Save & Quit Unlearn Cancel Discpor Help

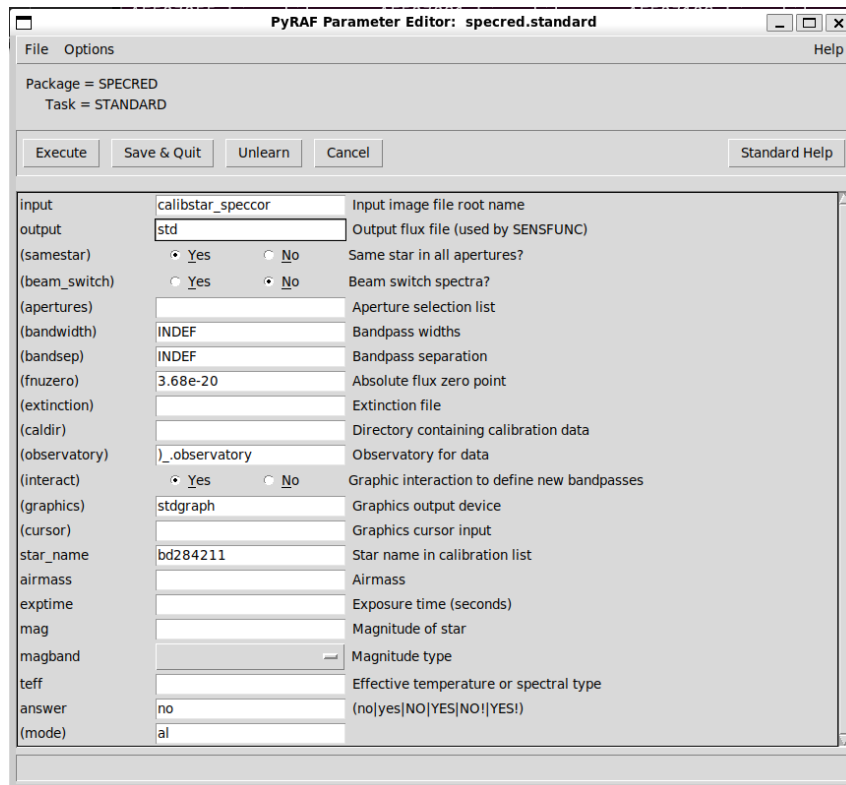
input	calibstar_spec	List of input spectra
output	calibstar_speccor	List of output spectra
(linearize)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Linearize (interpolate) spectra?
(database)	database	Dispersion solution database
(table)		Wavelength table for apertures
(w1)	INDEF	Starting wavelength
(w2)	INDEF	Ending wavelength
(dw)	INDEF	Wavelength interval per pixel
(nw)	INDEF	Number of output pixels
(log)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Logarithmic wavelength scale?
(flux)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Conserve total flux?
(blank)	0.0	Output value of points not in input
(samedisp)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Same dispersion in all apertures?
(global)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Apply global defaults?
(ignoreaps)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Ignore apertures?
(confirm)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Confirm dispersion coordinates?
(listonly)	<input type="radio"/> Yes <input checked="" type="radio"/> No	List the dispersion coordinates only?
(verbose)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Print linear dispersion assignments?
(logfile)		Log file
(mode)	al	

The resulting calibration star spectrum looks like:



i) Setting up flux calibration

Iraf has this information in its database and will find the flux calibration using the **standard** task.



PyRAF Parameter Editor: specred.standard

File Options Help

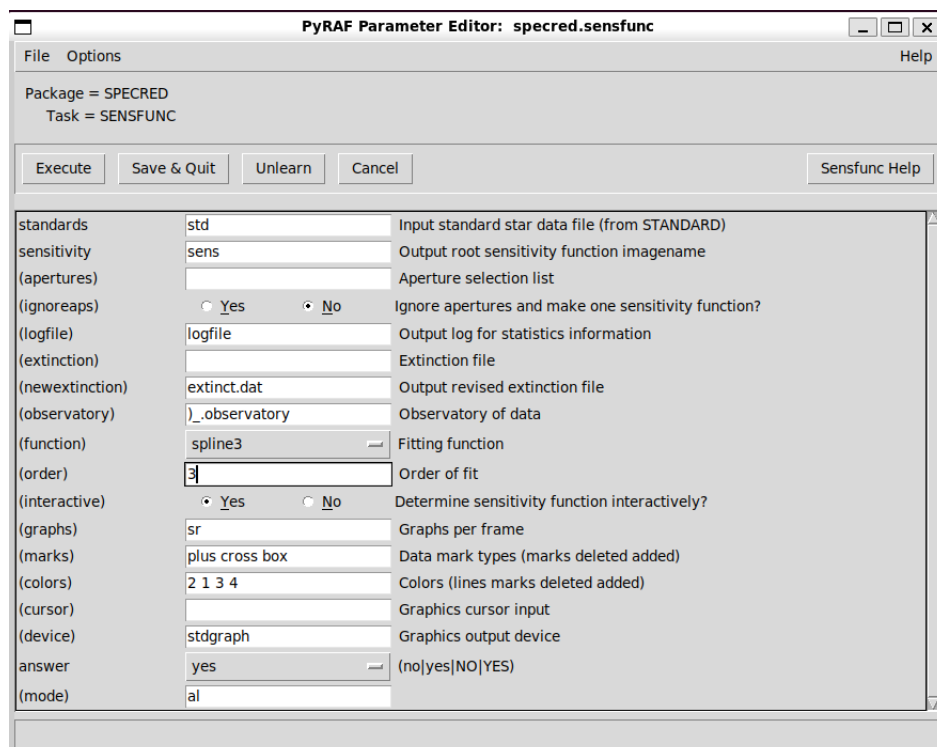
Package = SPECRED
Task = STANDARD

Execute Save & Quit Unlearn Cancel Standard Help

input	calibstar_speccor	Input image file root name
output	std	Output flux file (used by SENSFUNC)
(samestar)	<input type="radio"/> Yes <input type="radio"/> No	Same star in all apertures?
(beam_switch)	<input type="radio"/> Yes <input type="radio"/> 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)	<input type="radio"/> Yes <input type="radio"/> 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	

ii) Fitting the sensitivity function

Using the **sensfunc** task, we'll fit the sensitivity function:



PyRAF Parameter Editor: specred.sensfunc

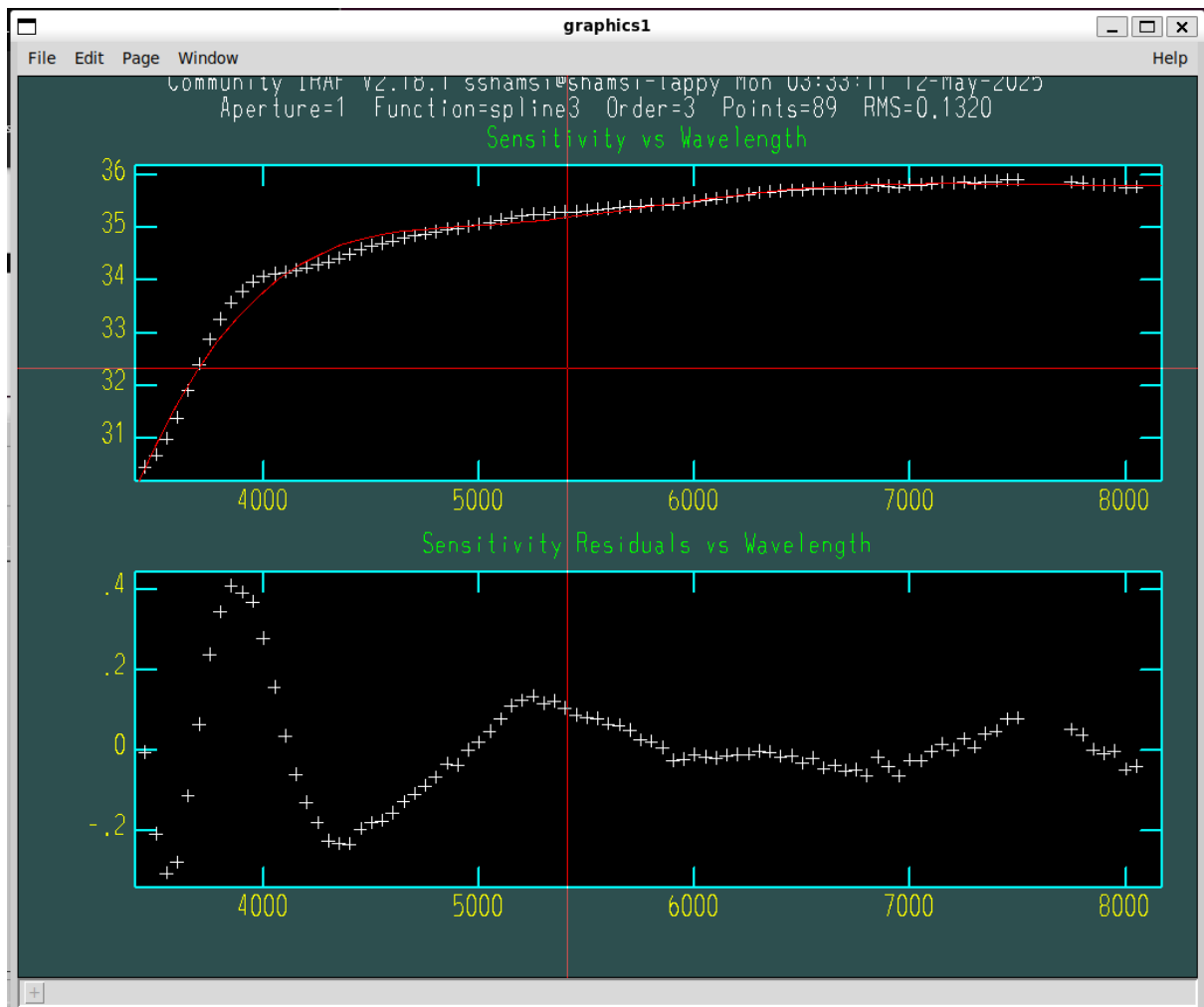
File Options Help

Package = SPECRED
Task = SENSFUNC

Execute Save & Quit Unlearn Cancel Sensfunc Help

standards	std	Input standard star data file (from STANDARD)
sensitivity	sens	Output root sensitivity function imagename
(apertures)		Aperture selection list
(ignoreaps)	<input type="radio"/> Yes <input type="radio"/> No	Ignore apertures and make one sensitivity function?
(logfile)	logfile	Output log for statistics information
(extinction)		Extinction file
(newextinction)	extinct.dat	Output revised extinction file
(observatory)	}_observatory	Observatory of data
(function)	spline3	Fitting function
(order)	3	Order of fit
(interactive)	<input type="radio"/> Yes <input type="radio"/> No	Determine sensitivity function interactively?
(graphs)	sr	Graphs per frame
(marks)	plus cross box	Data mark types (marks deleted added)
(colors)	2 1 3 4	Colors (lines marks deleted added)
(cursor)		Graphics cursor input
(device)	stdgraph	Graphics output device
answer	yes	(no yes NO YES)
(mode)	al	

This yields the following plot, which we accept:



iii) Applying the flux correction using the **calibrate** task:

PyRAF Parameter Editor: specred.calibrate

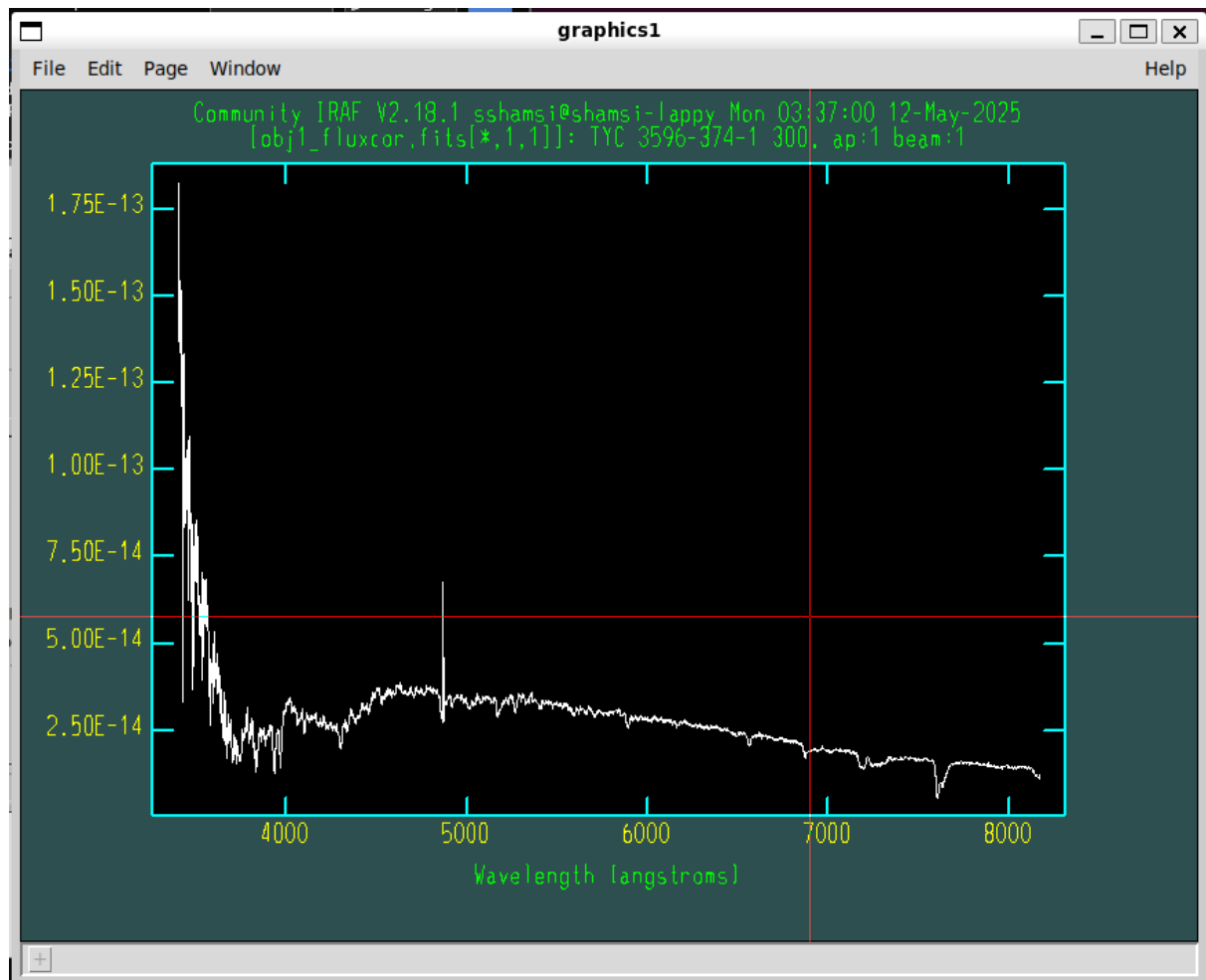
File Options Help

Package = SPECRED
Task = CALIBRATE

Execute Save & Quit Unlearn Cancel Calibrate Help

input	obj1_speccor	Input spectra to calibrate
output	obj1_fluxcor	Output calibrated spectra
(extinct)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Apply extinction correction?
(flux)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Apply flux calibration?
(extinction)		Extinction file
(observatory)	_observatory	Observatory of observation
(ignoreaps)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Ignore aperture numbers in flux calibration?
(sensitivity)	sens	Image root name for sensitivity spectra
(fnu)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Create spectra having units of FNU?
airmass		Airmass
exptime		Exposure time (seconds)
(mode)	al	

This gives us the following flux corrected spectrum for **obj1**:



Correction for interstellar extinction

We use the **deredden** task as follows:

PyRAF Parameter Editor: specred.deredden

File Options Help

Package = SPECRED
Task = DEREDDEN

Execute Save & Quit Unlearn Cancel Deredden Help

input	obj1_fluxcor	Input spectra to correct
output	obj1_dered	Output corrected spectra
value	8.262	Extinction parameter value
(R)	3.1	A(V)/E(B-V)
(type)	A(V)	Type of extinction parameter
(apertures)		Apertures to correct
(override)	<input type="radio"/> Yes <input checked="" type="radio"/> No	Override previous correction?
(uncorrect)	<input checked="" type="radio"/> Yes <input type="radio"/> No	Uncorrect previous correction?
(mode)	al	

Finally, we can plot the dereddened spectrum:

