AST3002F: Reduction Guide (2018)

N. Titus, P.A. Woudt

Stellar Spectroscopy

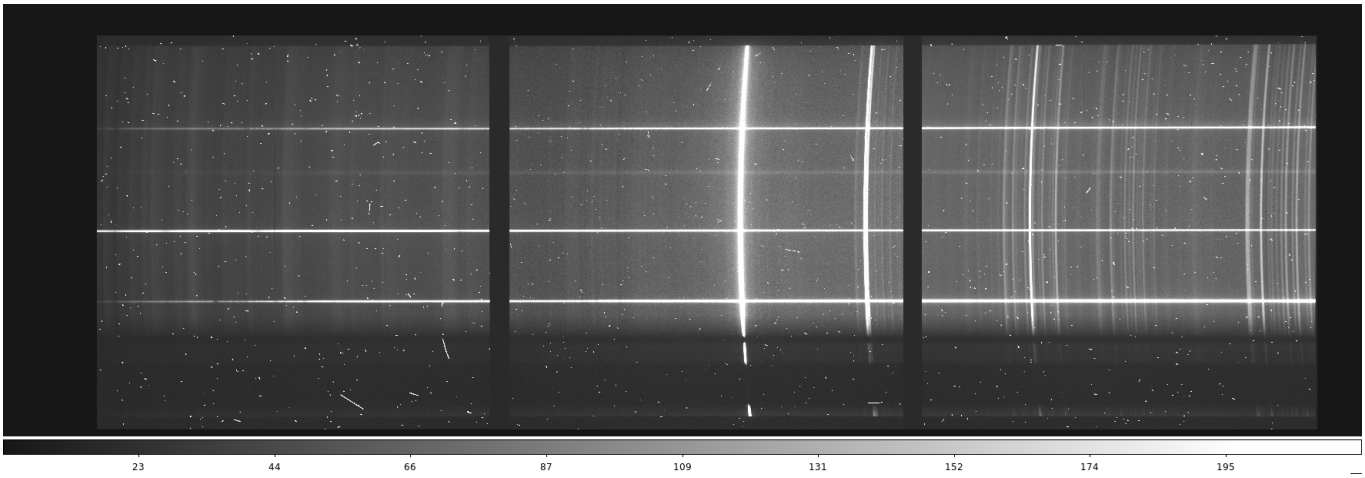
The 2018 AST3002F spectroscopy practical involves the reduction and analysis of SALT spectroscopy of candidate white dwarfs in the Kepler field. The data are associated with a SALT Science project (2016-1-SCI-038).

The reduction guide describes, step-by-step, the reduction process for the spectroscopic practical which includes a number of steps, namely:

- a visual inspection of the spectra
- removal of cosmic rays
- wavelength calibration
- sky subtraction
- extraction of the spectrum of the primary target
- extraction of the spectrum of other stars near the target

[**possible addition**]
- calibrate the intensity scale of the spectrum by using spectra-photometric standards

Following the extraction of the spectra, you will then:
- perform a spectral classification of the extracted spectra,
- quantify the quality of the spectra.
- describe the reduction and classification process in a report



Linux Cheat Sheet

There is no undoing anything that has been done in the terminal. If you overwrite a file using command line, there is no retrieving the old document. So be super careful before you hit enter. When using the ***rm*** command it removes the specified file from your computer, it does not send it to trash - there is no going back.

Command line logic:

command/application input output

Note:
- directory is synonymous to folder
- terminal is synonymous to command line

Command	Action
cd	change directory
cd ../	go back a directory
ls	list contents of directory
man *command*	help on specified command
cp one.txt two.txt	makes a copy of one.txt and names it two.txt
mv one.txt two.txt	renames one.txt, new name two.txt
mv one.txt test/	move one.txt into test (test is a directory)
rm one.txt	deletes one.txt
more one.txt	displays one.txt in terminal
pwd	displays location of current directory
ds9	displays fits files

1. Step 1 - getting to know IRAF

1.0. A short introduction

In this first section of the practical, we will perform a first visual inspection of the data and start to characterize the data. It contains a first introduction to IRAF (Image Reduction Analysis Facility).

All the groups have 4 SALT spectra, which includes one star in common: EPIC229228129. In this practical guide, we will use this particular system as an example. The class is split up in 5 groups:

Group 1 : Jade, Sarayn, Kelebogile
Group 2 : Chloe, Petro
Group 3 : Callan, Geoff
Group 4 : Simon, Anke
Group 5 : Jordan, Shilpa (= use Group 2 data for now) - new set of data will be prepared

Throughout this practical, it is recommended to make a log of the observations (in tabular form) for inclusion in the report at a later stage, so make plenty of notes as you go along.

1.1. Log in to IRAF

The linux machines in the computer room (RW James 5.17) - and your laptops with the 'virtual machine' environment - should all have the necessary software installed to reduce and analyse the SALT spectra.

1.1.1 Linux-based computers

The Linux machines should be up and running and there is no need to log in to this machine. A standard session should be open for you to use.

Alternatively, on your laptop, start the virtual machine.

Open a 'terminal' (left bar) and at the prompt type:

```
xgterm &            [enter]      this starts a graphics terminal
```

In the new xgterm (a graphics terminal), type:

```
cd                  [enter]      this changes to the default home directory
cd Group1           [enter]      this changes the directory to where the data are for Group 1
ds9 &               [enter]      this will open a graphics display unit
```

1.1.2 IRAF

mkiraf description

Start IRAF by typing in the xgterm:

```
cl                  [enter]      this start the IRAF command language (cl)
```

That will show up something like the following information (but note that the version of IRAF on the desktop is slightly older than the one I am showing here (2.14.1 versus 2.15.1a).

```
   NOAO/IRAF PC-IRAF Revision 2.15.1a EXPORT Mon Feb 21 18:54:16 MST 2011
     This is the EXPORT version of IRAF V2.15.1a supporting PC systems.


  Welcome to IRAF.  To list the available commands, type ? or ??.  To get
  detailed information about a command, type `help <command>'.  To run  a
  command  or  load  a  package,  type  its name.   Type  `bye' to exit a
  package, or `logout' to get out  of the CL.    Type `news' to find  out
  what is new in the version of the system you are using.

  Visit http://iraf.net if you have questions or to report problems.

  The following commands or packages are currently defined:

      apropos     guiapps.    mscred.     plot.       sirius.     stsdas.
      color.      images.     noao.       proto.      softools.   system.
      dataio.     language.   obsolete.   pysalt.     stecf.      tables.
      dbms.       lists.      phist.      rvsao.      stlocal.    utilities.

ecl>
```

check help data base on Ubuntu install

The observations are separated by the type of observations, and contain separate folders for observing logs and finding charts (listed in the directories '**Logs**' and '**Finders**', respectively).

To see the content of the directory type: 'ls' or 'dir' [ls and dir give the same output]

```
ecl> ls
Finders Flats   Images  Logs    Spectra
```

To go to the directory contanining the stellar spectra, type: 'cd Spectra'

```
ecl> cd Spectra
```

The content of this directory can be listed by typing: 'ls'

```
ecl> ls
mbxgpP201604250012.fits mbxgpP201605050044.fits mbxgpP201605290054.fits
```

```
mbxgpP201604250013.fits mbxgpP201605150053.fits mbxgpP201605290055.fits
mbxgpP201605050043.fits mbxgpP201605150054.fits
```

This shows the complete list of spectra to be analyses by Group 1; this is a set of eight fits images. The format of the images show the data on which they were taken. If you look at the first image, the naming convention of the data is:

```
mbxgpP201604250012.fits

m: merged
b: bias subtracted
x: cross-talk corrected
g: gain corrected
p: prepare (for reductions)
P: indicates spectrum taken with the Robert-Stobie Spectrograph (RSS)
xxxx: year
xx: month
xx: day
xxxx: sequence of observation on a given night
```

The data are stored in FITS[1] format (flexible image transport system). It stores the intensity information (number of recorded electrons per pixel) as well as information describing the observations (exposure time, object, position on the sky, etc.). The latter is stored in the header.

The format of the SALT fits images is such that images are stored in a multi-dimensional data cube. Each fits image contains two 'extensions' - the first extension contains the header information, and the second extension contains the data.

The example below shows the header of EPIC229228129; a number of relevant keywords are highlighted in **bold**.

A standard IRAF **task** to view the image header is **`imhead`** (Image Header). You can view the header by typing:
`'imhead mbxgpP201605050043.fits'` on the command line.

```
ecl> imhead mbxgpP201605050043.fits
mbxgpP201605050043.fits: FXF: must specify which FITS extension (mbxgpP201605050043.fits)
```

This indicates that the data are multi-dimensional, and you have to specify the FITS extension

```
ecl> imhead mbxgpP201605050043.fits[0]
mbxgpP201605050043.fits[0][0][short]: EPIC229228129
```

This represents the part of the FITS image that contains useful information about the observation.

```
ecl> imhead mbxgpP201605050043.fits[1]
mbxgpP201605050043.fits[1][3171,1026][real]: EPIC229228129
```

This represents the part of the FITS image that contains the data, and shows the number of columns (**3171**) and rows (**1026**), as well as the data format (**real**), and the name of the object (**EPIC229228129**).

To view the full content of the first extension of FITS header, type: `'imhead mbxgpP201605050043.fits 1+'` on the command line.

---

[1] http://en.wikipedia.org/wiki/FITS

```
ecl> imhead mbxgpP201605050043.fits[0] l+

mbxgpP201605050043.fits[0][0][short]: EPIC229228129
No bad pixels, min=0., max=0. (old)
Line storage mode, physdim [0], length of user area 6845 s.u.
Created Sat 06:45:56 25-Mar-2017, Last modified Fri 18:27:51 06-May-2016
Pixel file "mbxgpP201605050043.fits" [NO PIXEL FILE]
EXTEND  =                    T / FITS dataset may contain extensions
COMMENT   FITS (Flexible Image Transport System) format is defined in 'Astronomy
COMMENT   and Astrophysics', volume 376, page 359; bibcode: 2001A&A...376..359H
VERFITS = '2.2     '           / Current version of FITS header
NWINDOW =                    0 / Number of windows
NEXTEND =                    2 / Number of data extensions
PROPID  = '2016-1-SCI-038    ' / Proposal ID
PROPOSER= 'Buckley           ' / Proposer
OBSERVER= '        '           / Name of observer
OBSERVAT= 'SALT    '           / Southern African Large Telescope
SITELAT =             -32.3795 / Site latitude (degrees)
SITELONG=               20.812 / Site longitude (degrees)
INSTRUME= 'RSS     '           / Instrument name
DETSWV  = 'pdet-4.86.13'       / Detector software version
OBJECT  = 'EPIC229228129     ' / Target Object
RA      = '13:45:06.75       ' / Target RA
PM-RA   =                   0. / Proper motion in RA (mas/yr)
DEC     = '-06:32:54.30      ' / Target declination
PM-DEC  =                   0. / Proper motion in Dec (mas/yr)
EQUINOX =                 2000 / Equinox of target coordinates
EPOCH   =                2000. / Epoch of object RA, Dec
DATE-OBS= '2016-05-05'         / Date of observation
TIME-OBS= '23:33:52.812'       / UTC start of observation
UTC-OBS = '23:33:52.812'       / UTC start of observation
TIMESYS = 'UTC     '           / Time system for the TIME-OBS keyword
LST-OBS = '16:09:21'           / Local SIDT of start of observation
JD      =     2457514.48185185 / Julian day
EXPTIME =              900.192 / Exposure time
AIRMASS =     1.35127992049436 / Air mass
MOONANG =                180. / Angle between Moon and pointing (degrees)
PUPSTA  =    0.865671820862723 / Pupil size at start of observation
PUPEND  =    0.865671820862723 / Pupil size at end of observation
OBSMODE = 'SPECTROSCOPY      ' / Config State Machine State
DETMODE = 'Normal            ' / Detector mode
OBSTYPE = 'OBJECT            ' / Observation Type
PIXSCALE=               0.1267 / Image scale (arcseconds/unbinned pixel)
NAMPS   =                    6 / Number of amplifiers
NCCDS   =                    3 / Number of CCDs in detector
CCDSUM  = '2 4     '           / On chip summation
GAINSET = 'FAINT   '           / CCD gain mode
ROSPEED = 'SLOW    '           / CCD readout speed
CCDTEM  =            157.88445 / CCD temperature (K)
DEWTEM  =             18.03832 / Cooler box temperature (C)
AMPTEM  =                 28.8 / SDSU controller temperature (C)
CENTEM  =            129.86481 / Cold end temperature (K)
… (much more)
```

There is so much information in this header, that you cannot view it on a single screen. In order to save the content of the header into an ascii file to look at it later, you can perform the following command:

```
ecl> imhead mbxgpP201605050043.fits[0] l+ > output
```

```
ecl> prows mbxgpP201605050043.fits[1] 516 521
ecl> prows mbxgpP201605050043.fits[1] 516 521 wx1=0 wx2=1050  (arrow up gets previous command)
ecl> prows mbxgpP201605050043.fits[1] 516 521 wx1=0 wx2=1050 wy1=0 wy2=2000
```

Task: From the information in the header, extract useful information for your observing log [to be included in your write-up of the project]. What is the exposure time of this spectrum, what is the pixel scale?

Now let's have a look at the spectrum.

1.2 House keeping

As mentioned above, each fits file has a image header containing useful information about the instrumental setup, as well as the data (the spectrum). To make it easier while we process the data we can strip the header so that our fits file only contains the data. To do this we will use **imcopy** (image copy), which can be loaded by typing **images** followed by **imutil**

```
vocl> images
       imcoords.     imfit.        immatch.      tv.
       imfilter.     imgeom.       imutil.

images> imutil
       chpixtype     imdelete      imheader      imslice       listpixels
       hedit         imdivide      imhistogram   imstack       minmax
       hselect       imexpr        imjoin        imstatistics  nhedit
       imarith       imfunction    imrename      imsum         sections
       imcopy        imgets        imreplace     imtile
```

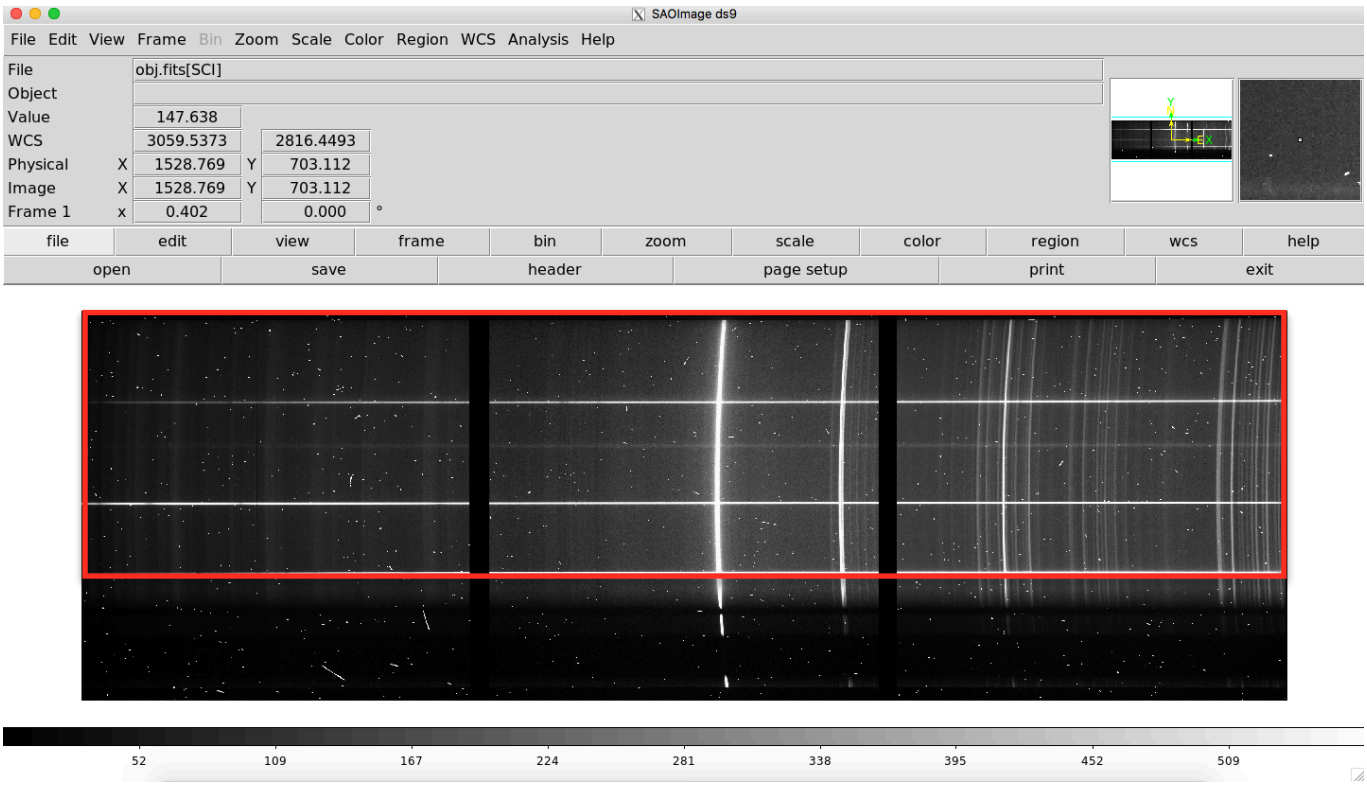
To strip the header type:

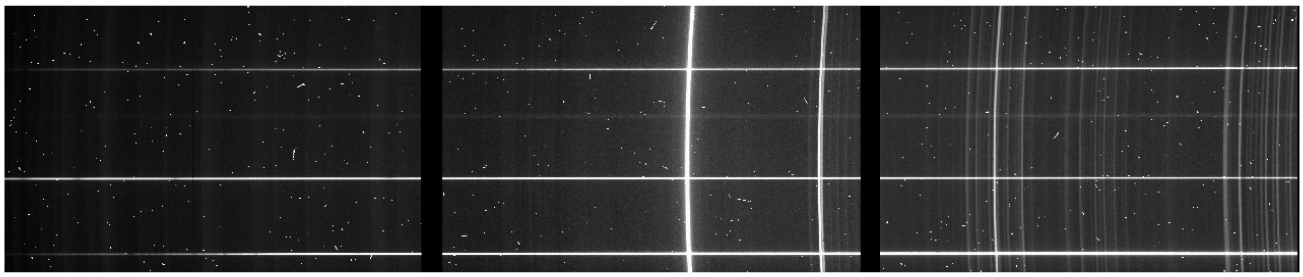
```
imutil> imcopy mbxgpP201605050043.fits[SCI,inherit] smbxgpP201605050043.fits
imutil> imcopy mbxgpP201605050044.fits[SCI,inherit] smbxgpP201605050044.fits
```

If you typed the command correctly, you will now have new files with a 's' prefix, as you specified in the command, to indicated that the header was stripped. When typing commands into the command line, you first specify the package you would like to use (**imcopy**), followed by the input file (mbxgpP201605050043.fits), and in this case you add '[SCI,inherit]' which specifies what you would like to apply to the input file, and finally followed by the output file (smbxgpP201605050043.fits).

The last thing we need to do is trim the images to get rid of the unusable bits. To do this we will use **imcopy** and ds9 to determine which part of the spectra are usable. Display the spectrum in ds9 and note the x and y ranges. In our case we will use the entire x range, and specify the y range [290,940] Type in the xgterm:

```
imutil> imcopy smbxgpP201605050043.fits[,290:940] tsmbxgpP201605050043.fits
imutil> imcopy smbxgpP201605050044.fits[,290:940] tsmbxgpP201605050044.fits
```





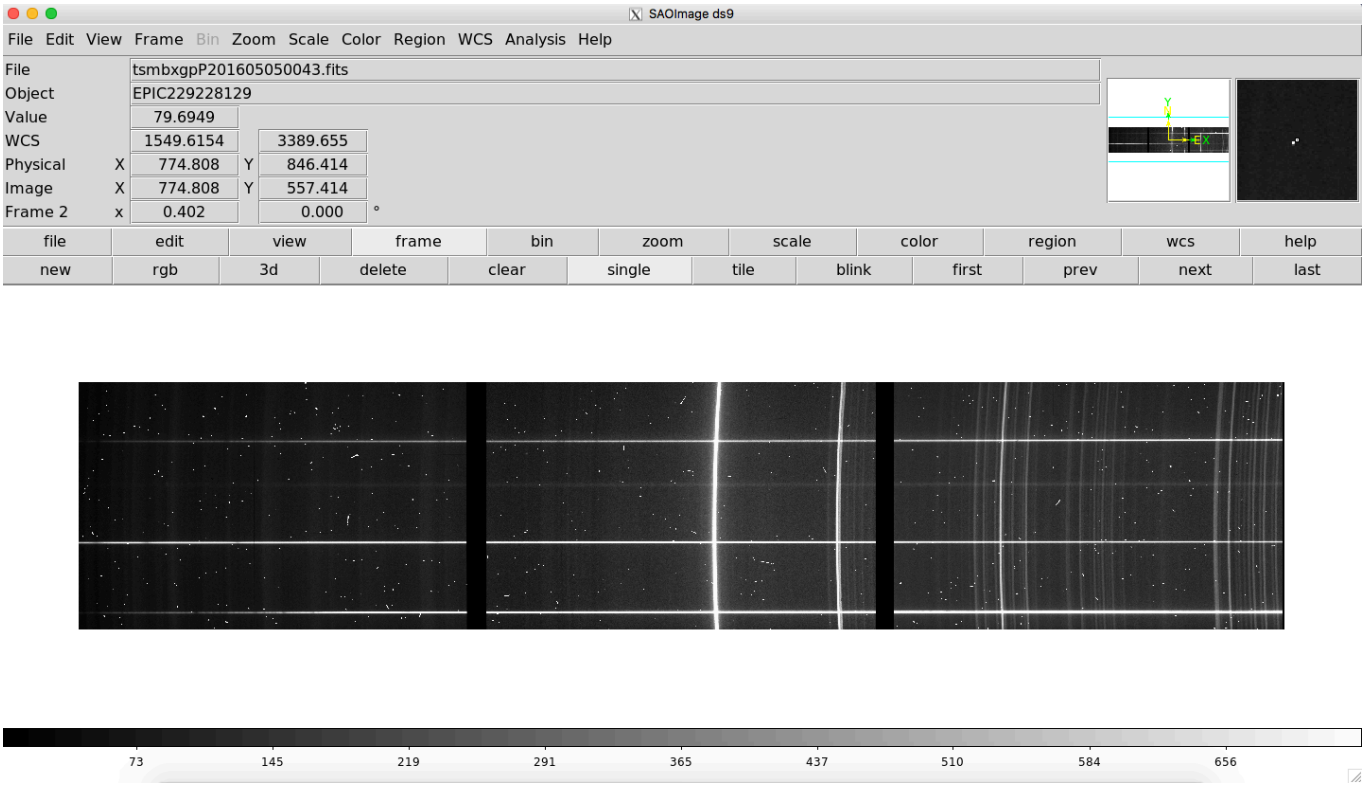
After you trimmed the image it will look something like this, you will notice that the unusable section at the bottom of the spectrum has been excluded.

1.3 Wavelength calibration

The spectra we have are 2-dimensional spectra mapping the intensities of each CCD pixel, but for us to extract meaningful information we need to map the spectra from pixel space to wavelength space along the dispersion axis (in this case the x-axis). Below is a raw spectrum before the wavelength correction was applied.

What do you see when you look at this window?

- The horizontal lines are the spectra that fell on the slit, i.e these are the stellar spectra.
- The vertical curved lines are sky lines. The lines are curved due to the hexagonal primary mirror of SALT.
- Cosmic rays
- Note that the X values are still in pixel space



1.3.1 IRAF tasks for wavelength calibration

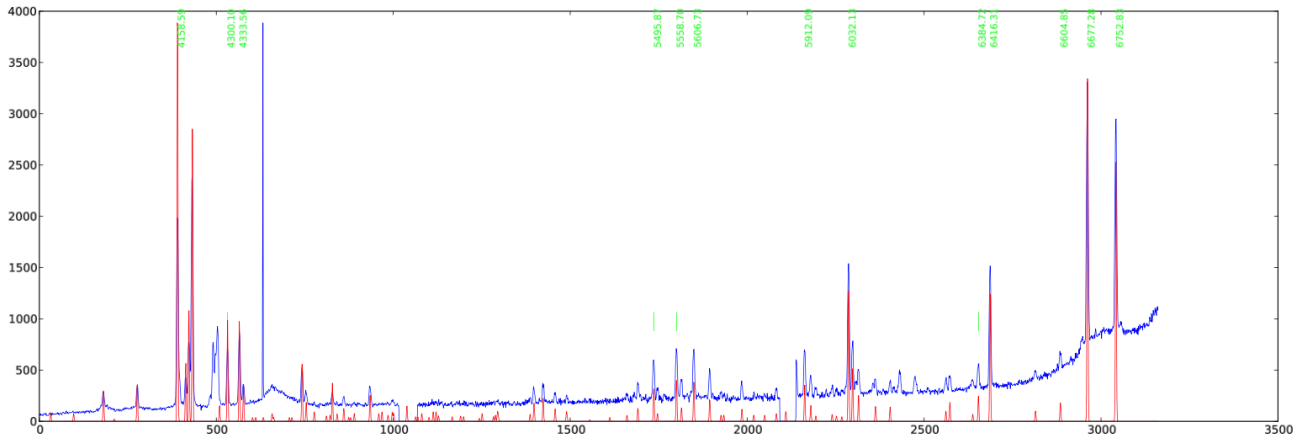
To find a map from pixel space to wavelength space we determine a wavelength solution using an **ARC** lamp with distinct and know emission lines to determine which pixel corresponds to which wavelength. In our case the low resolution ThAr arc lamp was used, thus before we start you will need the line list (**Ar.txt**) and the corresponding atlas (**plot_line_argon_lores.pdf**) in the same directory as your fits files.

When you open **plot_line_argon_lores.pdf** you will notice that there are various arc spectra depending on your grating settings, most notably the grating and the grating angle. For the same grating you can adjust the wavelength range that was covered by changing the grating angle.

Task: By looking at the image header note the quantities of the following keywords: OBJECT, EXPTIME, CCDTYPE, LAMPID, GR-ANGLE, GRATING

To identify the ARC lookout for **CCDTYPE = ARC**. Once you have identified the ARC, and determined what grating was used and the grating angle you should select the corresponding arc spectrum **plot_line_argon_lores.pdf**.

Figure 10: 1225 — 004: Ar PG0900 GR = 14.000 AR = 27.998 $\lambda\lambda$ = 3771 – 6863



The IRAF package we will use for the wavelength calibration is **longslit** (longslit spectroscopy) which is located in **twodspec** (two dimensional spectra). To load **twodspec** type 'twodspec' in the xgterm, and you will see a list of packages contained in twodspec. To load **longslit** type 'longslit'.

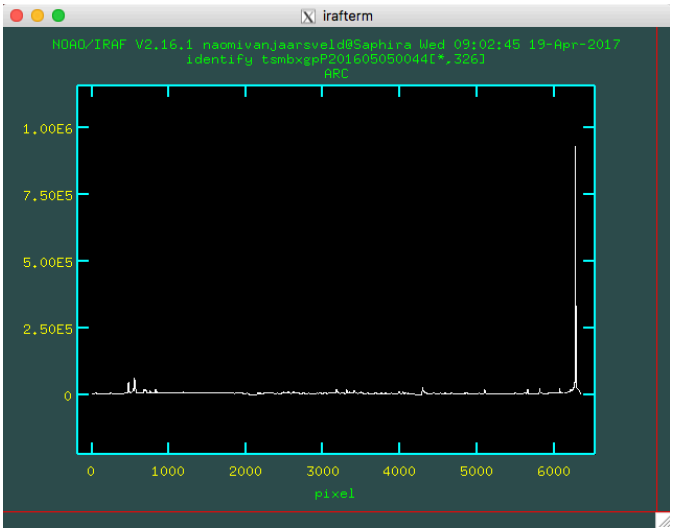
```
vocl> twodspec
      apextract.       longslit.

twodspec> longslit
       aidpars@          deredden      identify      sarith        specplot
       autoidentify      dopcor        illumination  scopy         specshift
       background        extinction    lcalib        sensfunc      splot
       bplot             fceval        lscombine     setairmass    standard
       calibrate         fitcoords     reidentify    setjd         transform
       demos             fluxcalib     response      sflip
```

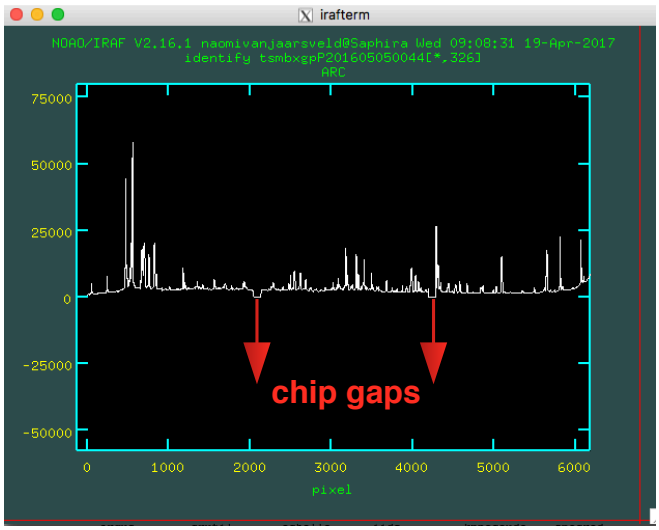
The tasks we will use to determine a wavelength solution:

- **identify** - determine wavelength solution for middle line
- **reidentify** - determine wavelength solution for remainder of spectrum
- **fitcoord** - determine curvature correction

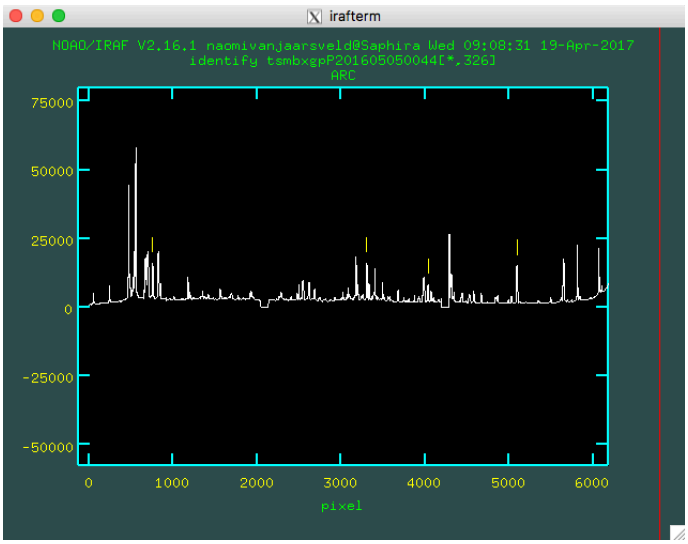
```
                                  I R A F
                       Image Reduction and Analysis Facility
PACKAGE = longslit
   TASK = identify

images  = tsmbxgpP201605050044  Images containing features to be identified
(section=          middle line) Section to apply to two dimensional images
(databas=             database) Database in which to record feature data
(coordli=               Ar.txt) User coordinate list
(units  =                     ) Coordinate units
(nsum   =                   10) Number of lines/columns/bands to sum in 2D images
(match  =                 -3.) Coordinate list matching limit
(maxfeat=                   50) Maximum number of features for automatic identifi
(zwidth =                100.) Zoom graph width in user units
(ftype  =             emission) Feature type
(fwidth =                   4.) Feature width in pixels
(cradius=                   5.) Centering radius in pixels
(thresho=                   0.) Feature threshold for centering
(minsep =                   2.) Minimum pixel separation
(functio=              spline3) Coordinate function
(order  =                    1) Order of coordinate function
(sample =                    *) Coordinate sample regions
(niterat=                    0) Rejection iterations
(low_rej=                   3.) Lower rejection sigma
(high_re=                   3.) Upper rejection sigma
(grow   =                   0.) Rejection growing radius
(autowri=                   no) Automatically write to database
(graphic=             stdgraph) Graphics output device
(cursor =                     ) Graphics cursor input
crval   =                       Approximate coordinate (at reference pixel)
cdelt   =                       Approximate dispersion
(aidpars=                     ) Automatic identification algorithm parameters
(mode   =                   ql)
```

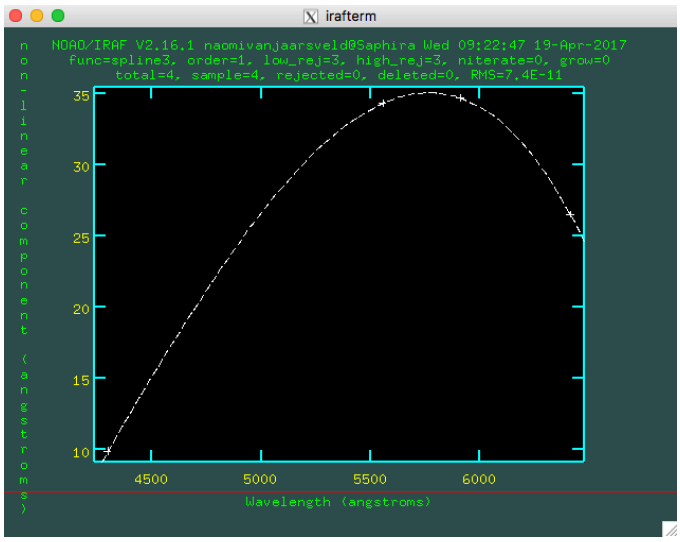
To set the parameters for the tasks we type ‘epar identify’, which will bring up the following window:





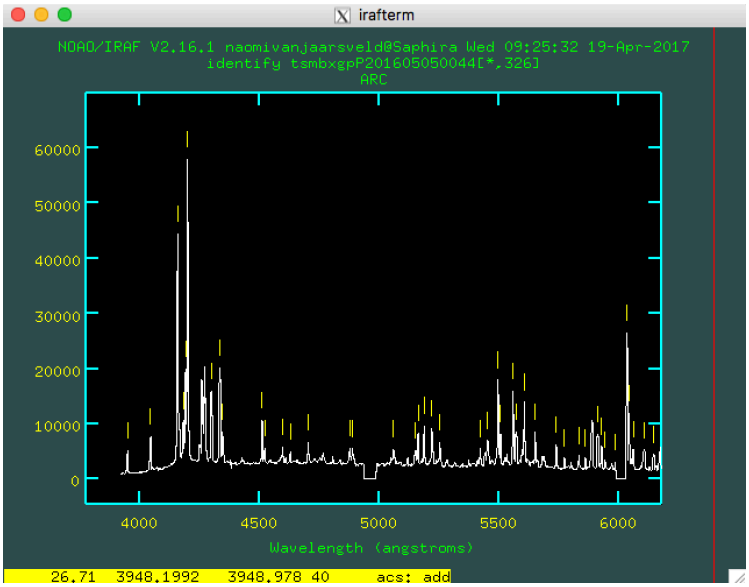
When you’ve set all the parameters you type ‘:go’ and hit enter. This will bring up the following window

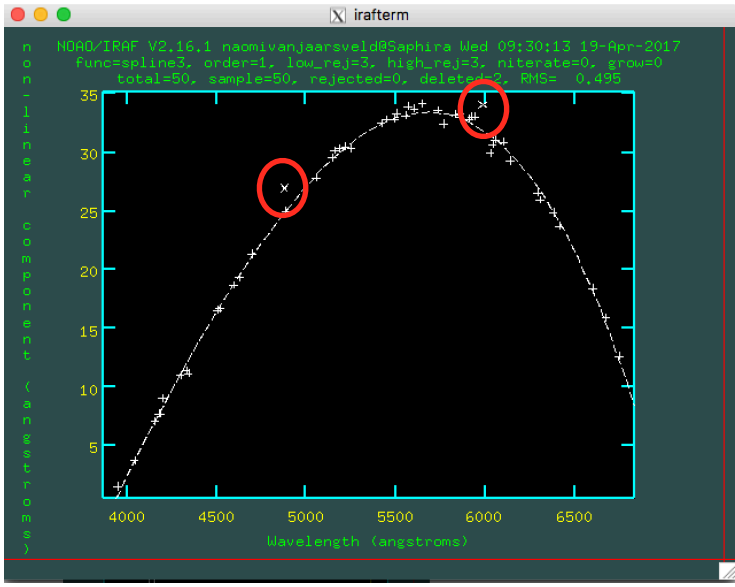




To interact with the window you have to click on it. Now to zoom in on the arc spectrum hit the **w** key and then **e** in the lower left corner and **e** again in the upper right corner. Now you should be able to see the arc lines clearly. Using **plot_line_argon_lores.pdf** we have to identify at least 4 lines roughly evenly spaced throughout the spectrum. Once you've identified a line, hover over the centroid with you cursor and hit **m,** you can then type in the wavelength of the line and hit enter. Do this for at least 4 lines.

When you are confident that you identified the correct lines hit **f** to determine a fit. The window will now change (right panel above), and you can flip through various views to investigate the fit by pressing **j, k or l**. To determine you identified the correct lines hit the **l** key to start off with and you should see a upside down parabola with a RMS~E-11. Hit **q** to get back to the previous window, and hit **l** to automatically identify the remainder of the lines in the spectrum (left panel below). Hit **f** again to investigate the identified lines. Ideally you would like an RMS~0.1, but it depends on the quality of your spectra. If



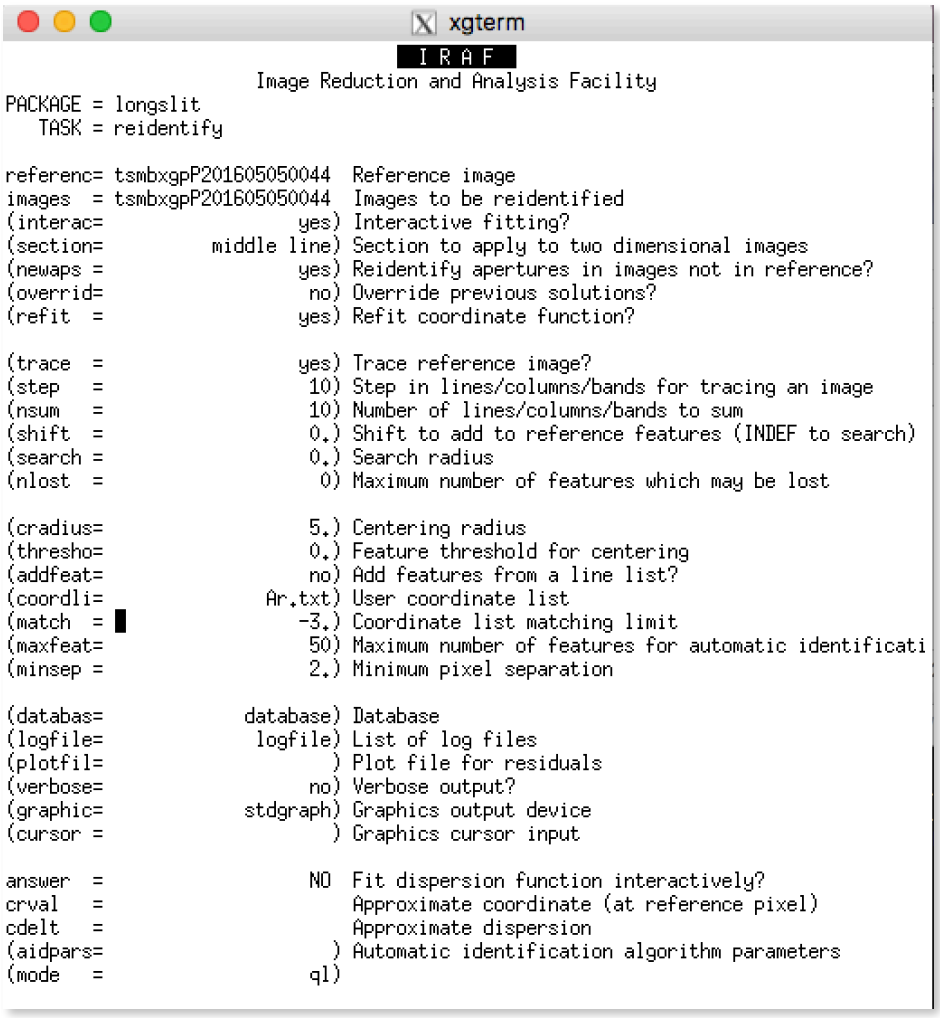


there are any points that deviate significantly from the parabola you can delete them by hitting **d**, followed by **f** for a new fit (right panel below). When you do this you will notice that the RMS decreases. You can use the other views to identify outliers as well, but don't delete too many points, because that will also decrease the accuracy of your wavelength solution. Once you are happy with the solution hit **q** twice, this will take you back to the xgterm where it will ask

```
Write feature data to the database (yes)?
```

Hit enter if you are happy or if you made a mistake type 'no' and hit enter, you can then repeat the process if you'd like.

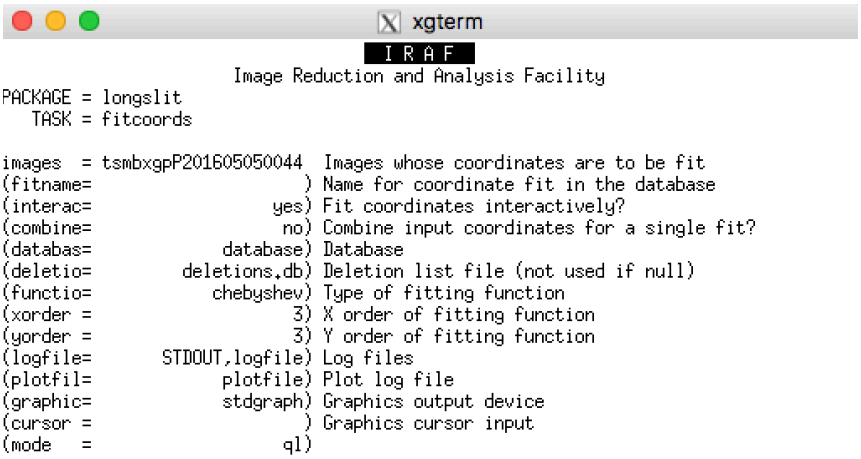
This would have created a new folder called 'database' in the directory, this is the folder that contains the wavelength solution information. To determine the wavelength solution for the whole 2-D spectrum we will use **`reidentify`**, so type 'epar reidentify' and the following window will open



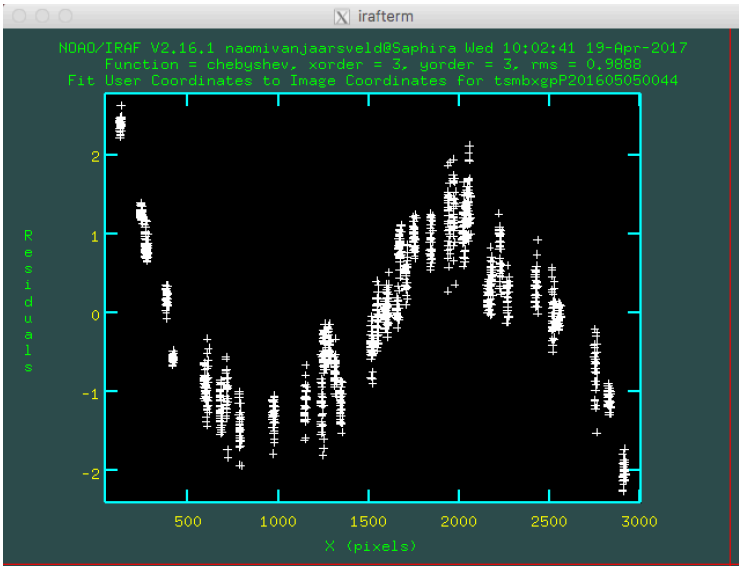
When you've updated the fields accordingly type ':go' and hit enter. In the xgterm it will ask you if you'd like to look at solution interactively. If you type 'yes' it will bring up a similar window to the **`identify`**

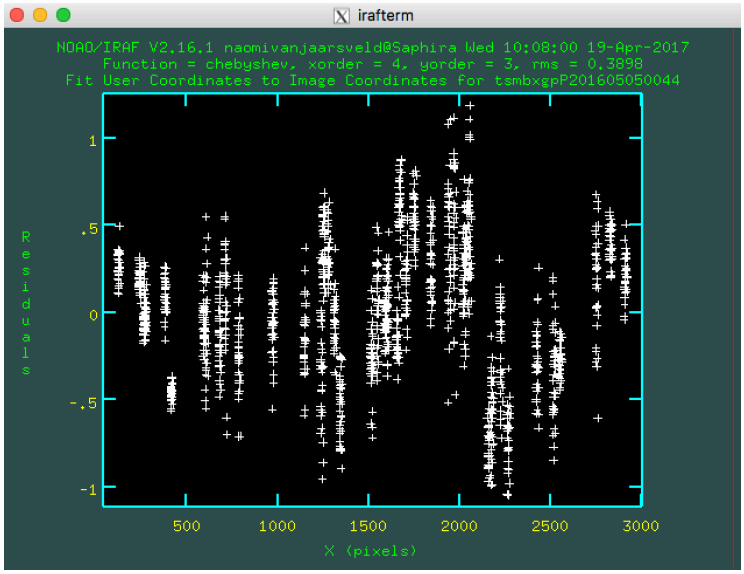
task, but we don't have to do this interactively so type 'NO', if you type 'no' instead it will ask you for every line it steps through.

Remember when we looked at the raw spectrum in ds9, you saw that the vertical sky lines were curved. We must correct for this, because this will also decrease the accuracy of our wavelength solution. For this we will use the **`fitcoord`** task. Again we type 'epar fitcoord':



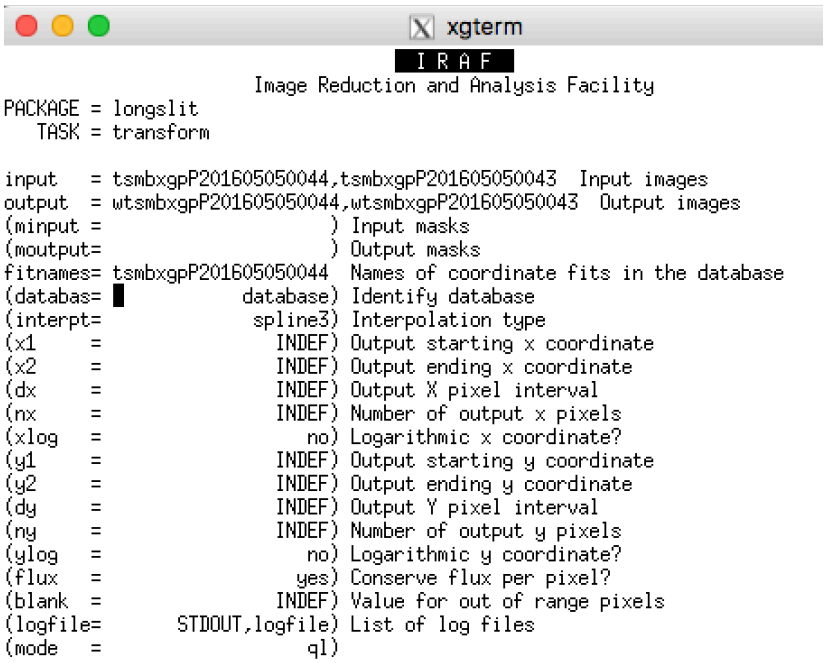
Once you made the changes, type ':go'. In the xgterm IRAF will ask you if you would like to complete the process interactively, type 'yes'. A window will pop up, plotting the X pixels vs Residuals (left panel below), this sinusoid shape is the curvature we saw in the ds9 image. You would like the lines to be alined, thus increase the xorder to 4 by typing: ':xorder 4', hit enter and press **f** to calculate a new fit (right panel below). You will also notice that the RMS decreased from ~1 to ~0.4. To exit hit **q**, IRAF will ask you again if you would like to save the result in the xgterm**,** if you are happy type 'yes' and press enter. Now that we have a complete wavelength solution we can apply it to the 2-D spectra.

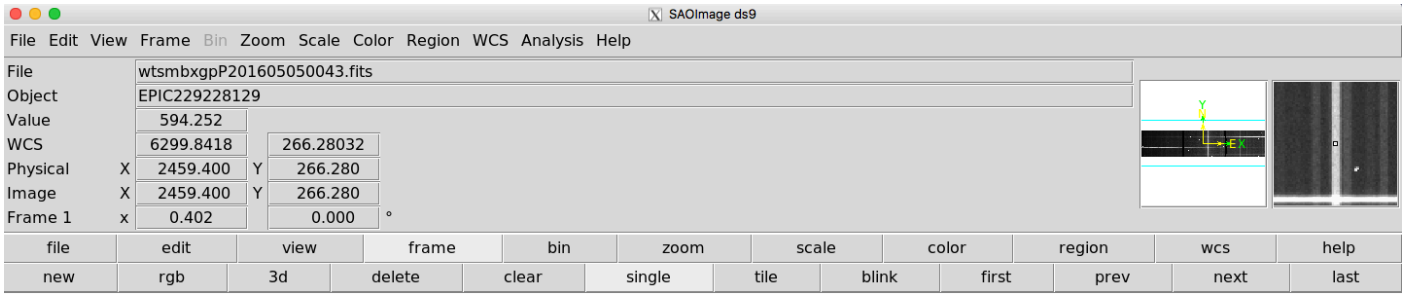


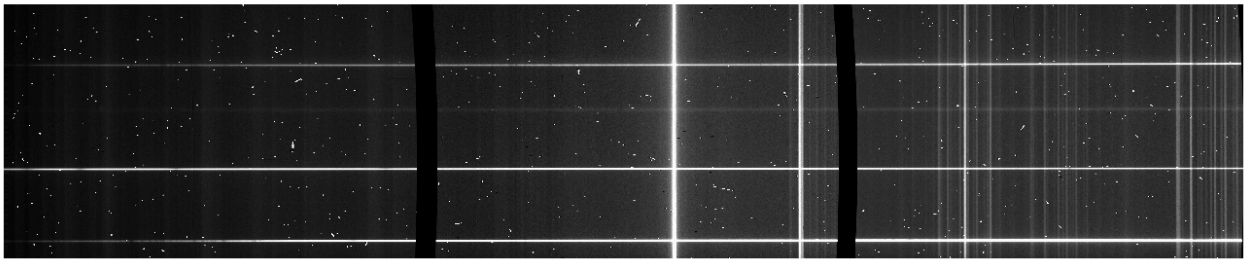


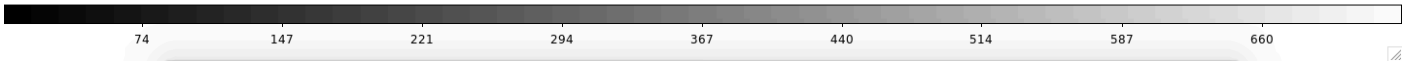
1.3.2 Applying the wavelength solution

Now that we've determined a complete wavelength solution we can apply the wavelength solution to the raw trimmed spectra and extract the 1-D spectrum. To do this we will use the **`transform`** task, when typing 'epar transform' the following window will appear. Make all the necessary changes and type ':go' as always.







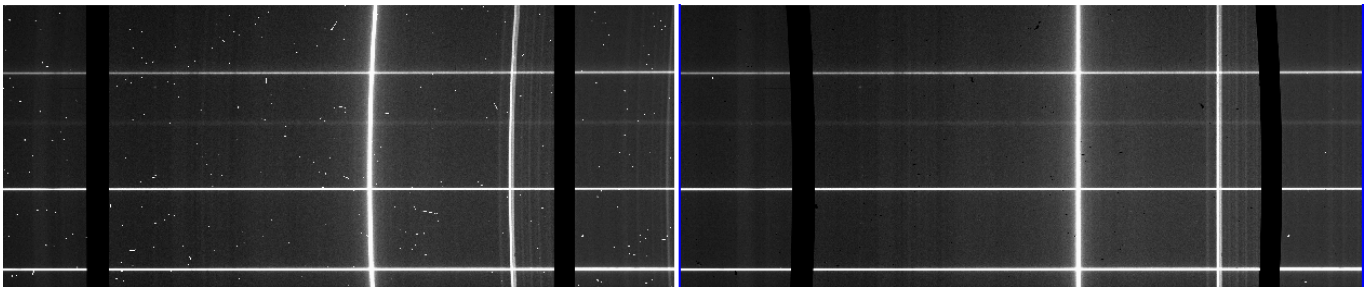


We applied the wavelength solution to both the arc (tsmbxgpP201605050044) and the science spectrum (tsmbxgpP201605050043). To check the wavelength solution open the transformed images (with a 'w' prefix) in ds9. You will notice that the x axis is now in angstroms and the previously curved sky lines are now straight.

1.4 The 1-dimensional spectrum

1.4.1 Cosmic ray removal

Before we can extract we have to try and remove most of the cosmic rays. To do this we will use a python script which is based on the L.A Cosmic routine. In the normal terminal cd to the directory where you've been working in the xgterm. Place the two python scripts (**cosmic.py** and **remove_cosmic_ray.py**) in the same directory as well. Open remove_cosmic_ray.py and enter the input file in the 'name' field, as well as a suitable prefix in the 'prefix' field - remember to save the changes. In the terminal type 'python remove_cosmic_ray.py'. Below you can see the before and after of cosmic ray removal.



1.4.1 Spectra extraction

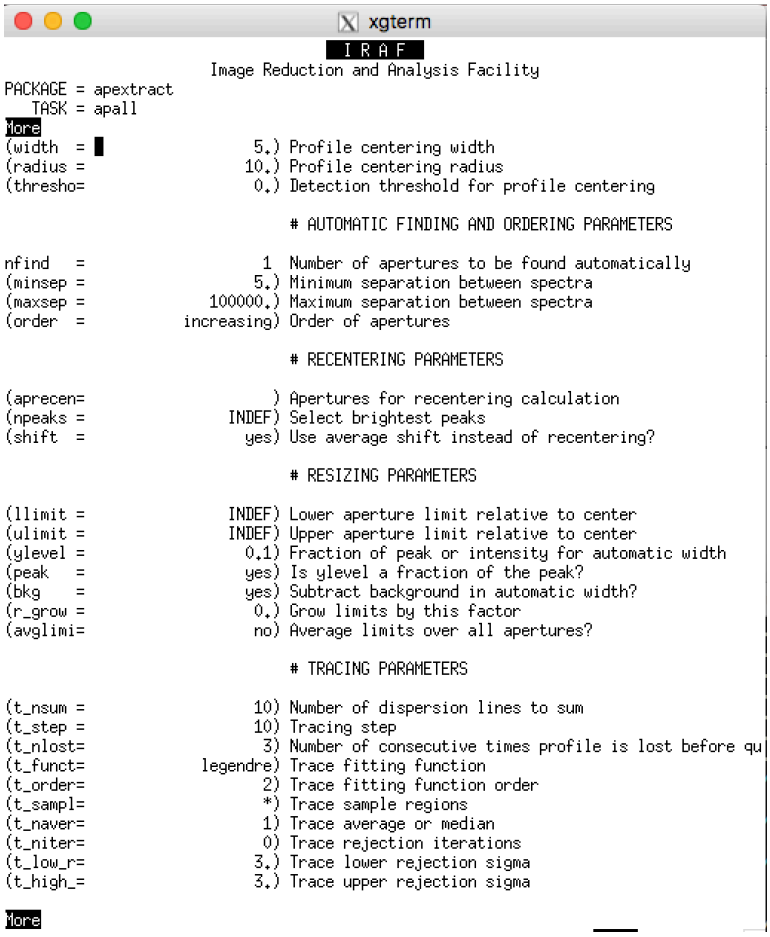
Now that we've applied the wavelength solution and removed the cosmic rays we can extract the 1-D spectrum. For this we will use the **apall** task in the **ctioslit** package. By running **apall** we can subtract the skylines, trace the spectrum and extract the spectrum.

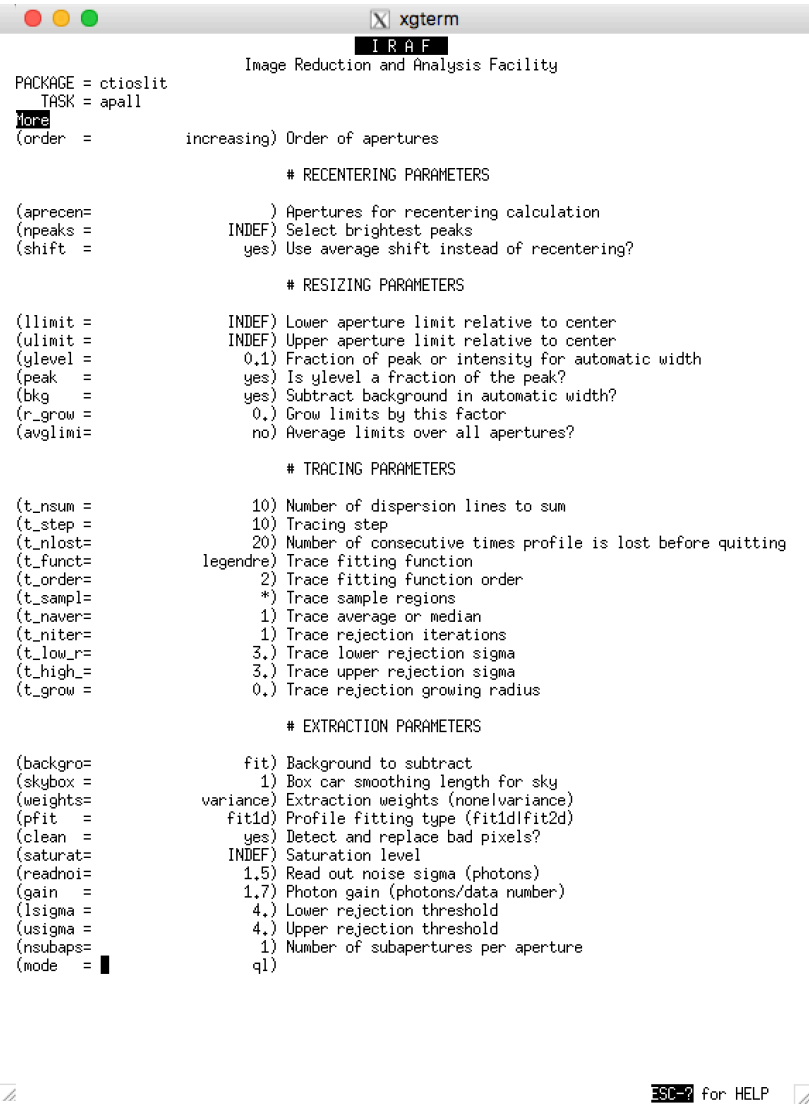
To load **ctioslit** type:

```
vocl> imred
       argus.       crutil.      echelle.     iids.        kpnocoude.   specred.
       bias.        ctioslit.    generic.     irred.       kpnoslit.    vtel.
       ccdred.      dtoi.        hydra.       irs.         quadred.

vocl> ctioslit
       aidpars@     apresize     demos        reidentify   sflip
       apall        apsum        deredden     response     slist
       apdefault@   aptrace      discor       sarith       specplot
       apedit       autoidentify dopcor       scombine     specshift
       apfind       background   doslit       scopy        splot
```

```
apflatten    bplot        identify     sensfunc     splot
apnormalize  calibrate    illumination setairmass
aprecenter   continuum    refspectra   setjd
```

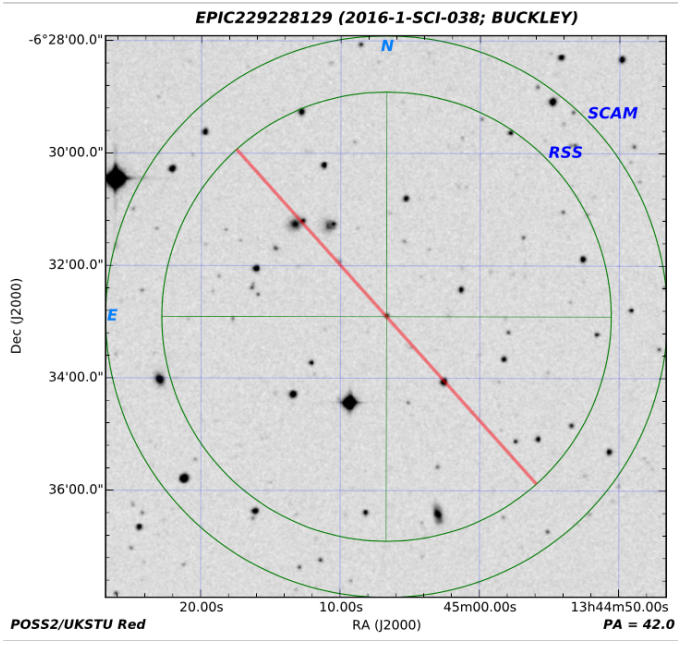
First we have to set various parameters in **`apall`** so we type ‘epar apall’, make all the necessary changes (image below) and type ‘:go’.

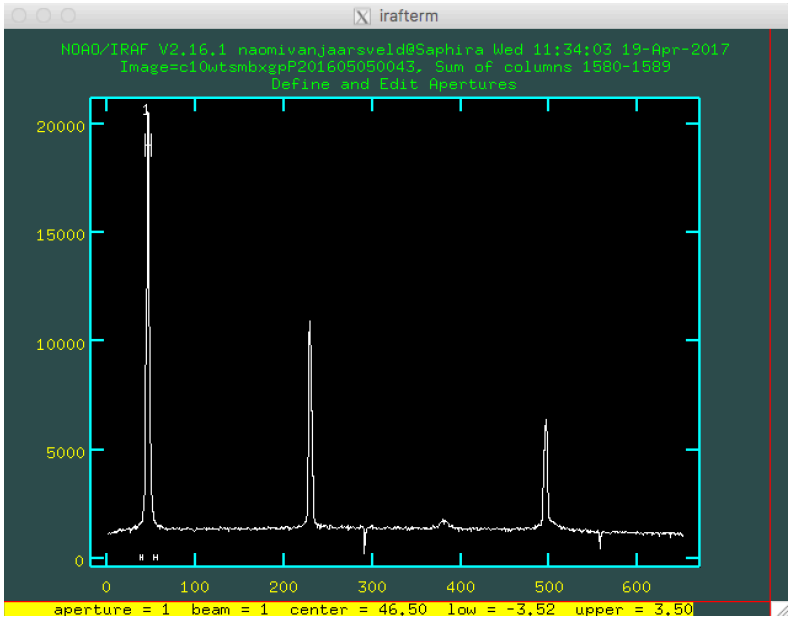




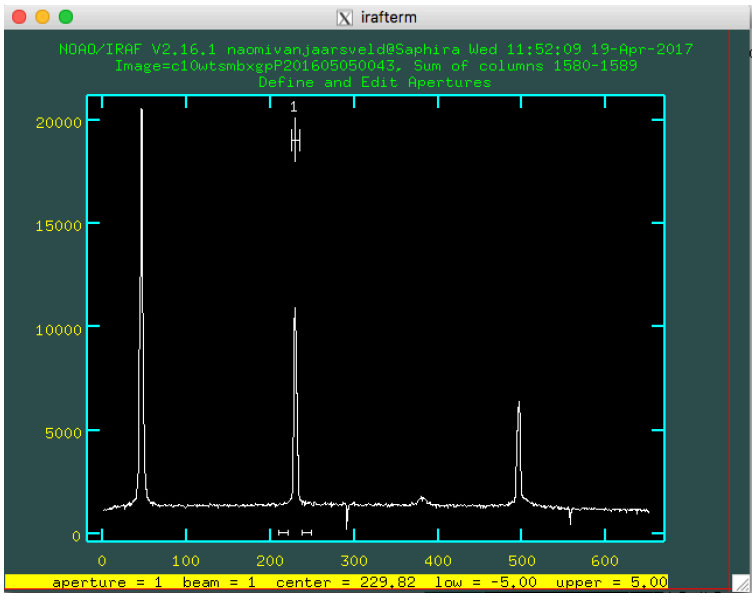
This will open a window displaying a vertical cut across the spectrum, meaning that there will be a spike for every start that fell on the slit. To determine which is the spectrum of EPIC229228129 we have to look at the finder as well. From the finder we can see that the middle aperture is the spectrum of EPIC229228129.

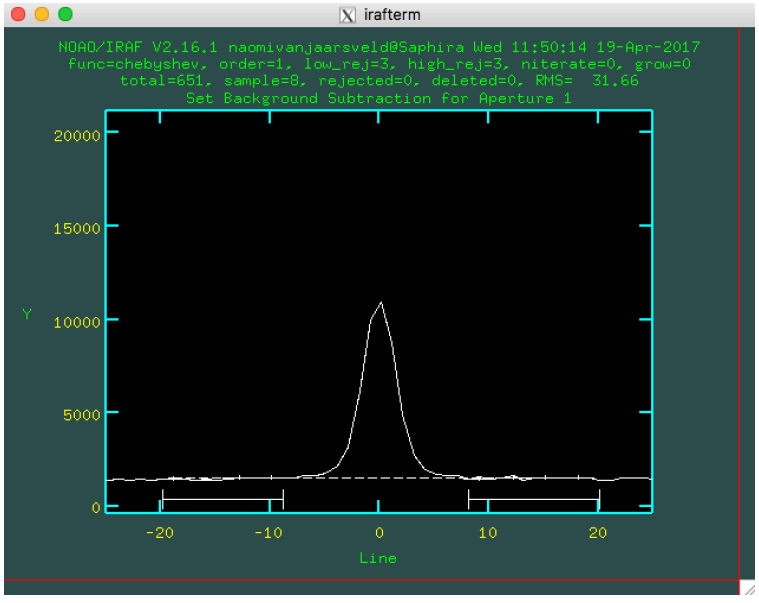
To mark the central peak we hover over the first peak and hit **d**, because IRAF chose the wrong aperture. Then we hover over the central peak and hit **m** to mark the correct aperture (left panel below). Secondly we have to define the sky or otherwise know as the background region by hitting **b** (righ panel



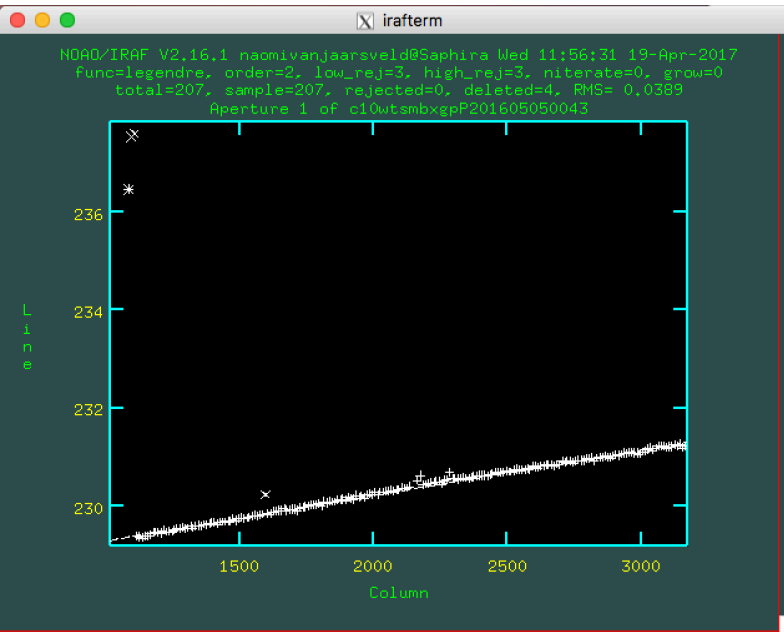


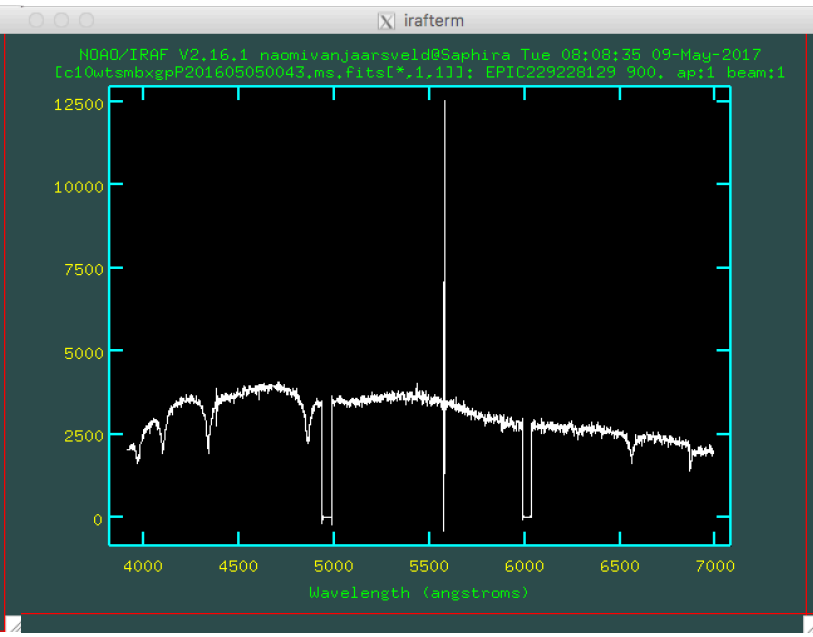
below), you can delete the previously defined background regions by hovering over them and hitting **z**. You can redefine the background regions by hitting **s** to set a lower limit and **s** again to set a upper limit. It is ideal to do this on both sides of the aperture, finally hit **f** to fit the new background. Press **q** exit.



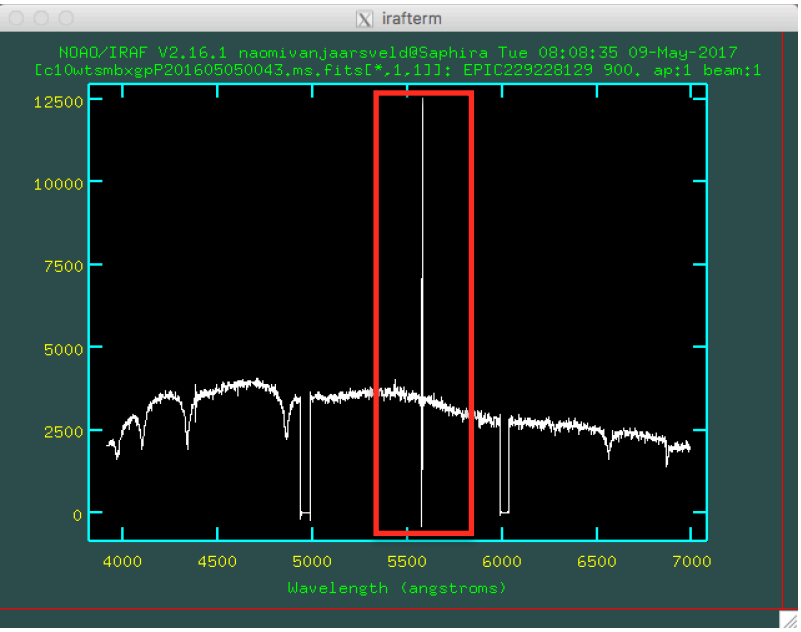


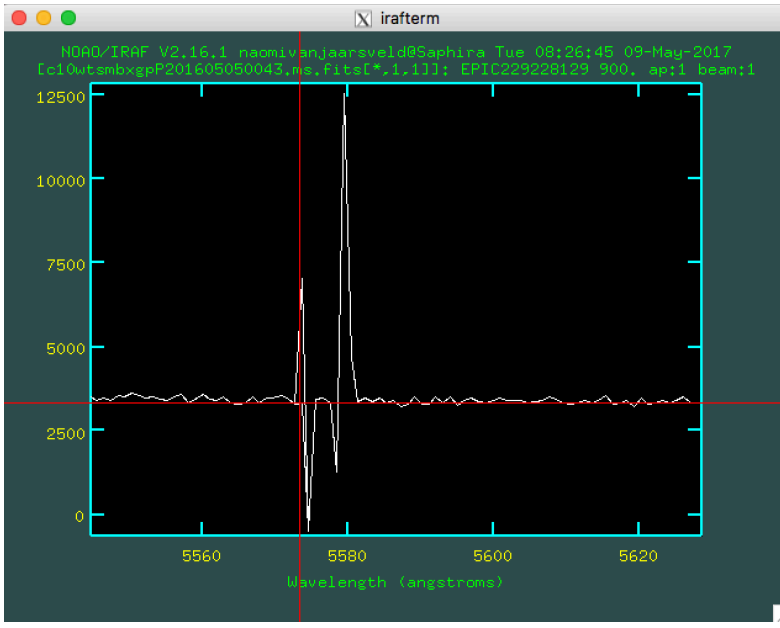
Hit **q** again and IRAF will ask if you would like to trace the apertures. If you are happy type 'yes' to all the questions in the window. In the trace window delete all the outliers by hovering over them and hitting **d** followed by **f** to refine the fit (left panel below). When you've deleted all the outliers hit **q**, IRAF will ask you if you would like to save the changes, if you are happy type 'yes' in the window to all the questions. When the spectrum is extracted it will display the final spectrum. To exit hit **q**. The extracted spectrum will have a '.ms.fits' extension.





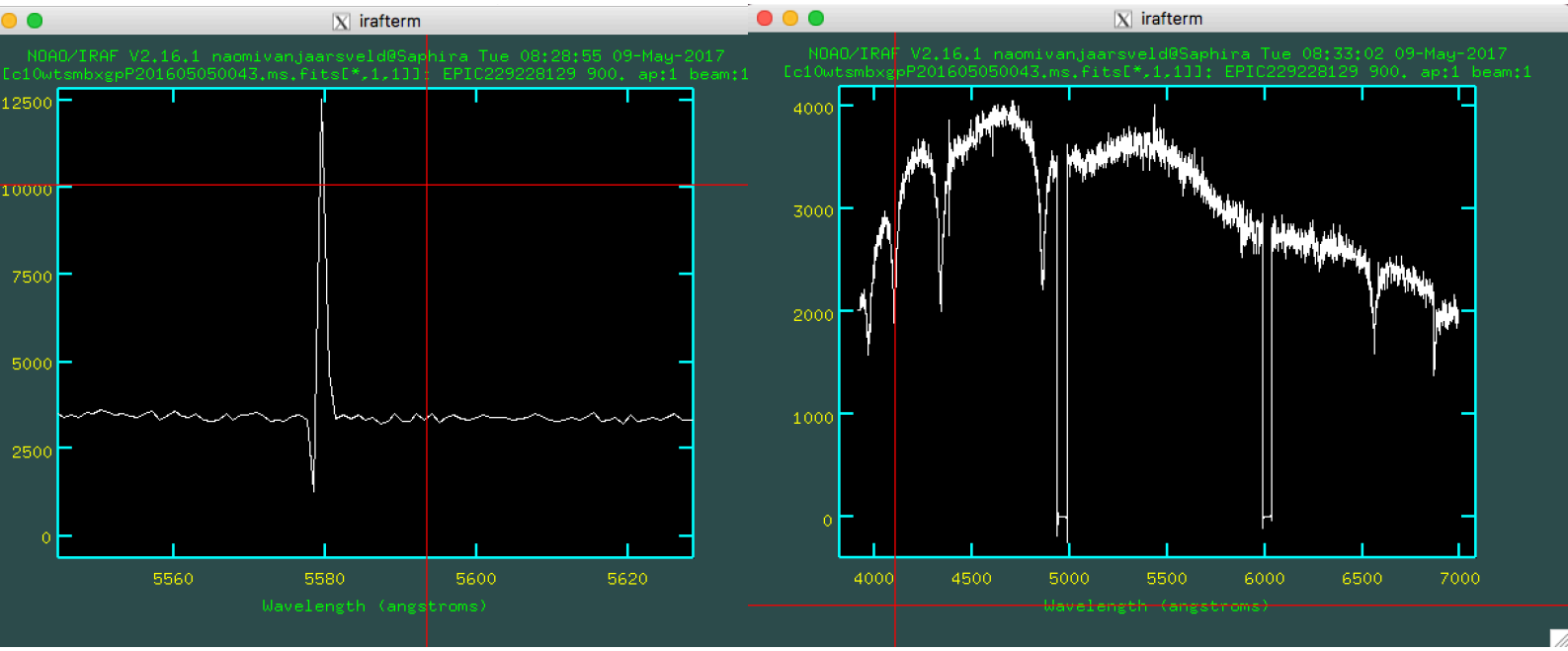
You can now view the extracted spectrum interactively using **splot** by typing 'splot c10wtsmbxgpP201605050043.ms.fits'. You will notice that around 5500 A there is a massive spike, this is a residual from the sky subtraction. In other cases you might see a spike (or sharp dip) due a residual cosmic ray. The easiest way to get rid of it, is to use **splot** to fudge out the point. To do this we will zoom in around the feature (left panel below) by hitting **w** and then **e** when the cursor is hovering at the lower left corner and then hit **e** again with the cursor hovering at the upper right corner of the area that you want to zoom in on. Repeat the process until you are happy with the view.





Now you can hover over with the cursor at the continuum lever and hit **j**, this will fudge out that one point, to view the change hit **r** to redraw the spectrum. You will now see that the peak would be removed (left

panel below). Repeat it for all the spikes and dips in the figure above (right panel). Once you are done you can zoom out again by hitting **w** followed by **a** - the whole feature would be gone now (right panel below). To save the new spectrum hit **i** and type in a sensible file name for the final spectrum and hit enter. This would have saved the updated spectrum.



1.5 A shortcut…

Once you've managed to wavelength calibrate and extract the spectrum, and have a rough idea what spectra extraction entails, you can use a shortcut for the remaining spectra. I will now rehash the process leading up to the wavelength calibration and spectra extraction, followed by the commands.

1. Make two separate lists, one for the arcs and another for the science images.
2. Run **`reidentify`** to automatically identify the features in all the remaining arcs with tsmbxgpP201605050044 as the reference image.
3. Run **`fitcoord`** for all the arcs.
4. Transform all the science images.
5. Open the transformed 2-D spectra to see that the wavelength and curvature solutions were applied properly.
6. Look at the finders and 2-D spectra to identify the correct spectrum
7. Extract the science spectra

1.5.1 Constructing lists

Before you do this, you need to look in the fits headers and identify which images are the arcs and which are the science frames. Use **`files`** command in IRAF to make a list of all the fits images - this assumes that you are in the directory containing only the remaining arcs and science images.

```
vocl> files m*.fits > all_list
vocl> files m*.fits > out
```

This will create two identical lists (called *all_list* and *out*), containing all the names of all the fits files in your current directory. As before, we need to strip the headers. To do this open *all_list* using gedit and add ***[SCI,inherit]*** to the back of each file name (e.g. mbxgpP201604250012.fits[SCI,inherit]) . Also open *out* with gedit and remove the ***.fits*** extension from each line. Remember to save the changes you made. Before we typed in the name of the respective files, but now you will give lists as inputs and outputs. You identify a list input or output by adding an *@* sign before the list name.

```
imutil> imcopy @all_list s@out
```

If it all went well you'll see the following printed in the IRAF terminal:

```
mbxgpP201604250012.fits[SCI,inherit] -> smbxgpP201604250012
mbxgpP201604250013.fits[SCI,inherit] -> smbxgpP201604250013
mbxgpP201605150053.fits[SCI,inherit] -> smbxgpP201605150053
mbxgpP201605150054.fits[SCI,inherit] -> smbxgpP201605150054
mbxgpP201605290054.fits[SCI,inherit] -> smbxgpP201605290054
mbxgpP201605290055.fits[SCI,inherit] -> smbxgpP201605290055
```

To strip the images and replace the ***[SCI,inherit]*** at the end of each file name in *all_list* with the specified trimming section (e.g. smbxgpP201604250012.fits[,290:940]) . It will be ideal if there is one specification that suits all the science frames, but you must check this. Have a look at the finding charts and the corresponding science frames (spectra) and see if you can identify a trimming section that will suit all the science images. If you can't, you'll have to specify a unique section for each science frame. Also open *out* with gedit and add a "s" in front of each file name.

```
imutil> imcopy s@all_list t@out
```

If it all went well you'll see the following printed in the IRAF terminal:

```
smbxgpP201604250012.fits[,290:940] -> tsmbxgpP201604250012
smbxgpP201604250013.fits[,290:940] -> tsmbxgpP201604250013
smbxgpP201605150053.fits[,290:940] -> tsmbxgpP201605150053
smbxgpP201605150054.fits[,290:940] -> tsmbxgpP201605150054
smbxgpP201605290054.fits[,290:940] -> tsmbxgpP201605290054
smbxgpP201605290055.fits[,290:940] -> tsmbxgpP201605290055
```

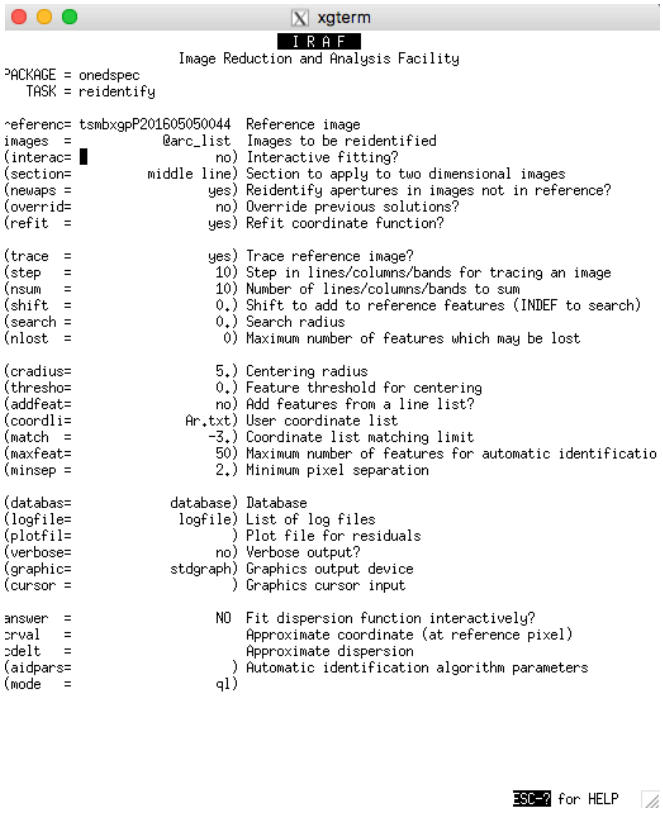
Now we can use the **`files`** command to make a list for the arc and science images respectively.

```
vocl> files t*.fits > science_list
vocl> files t*.fits > arc_list
```

Use *gedit* to open *science_list* and remove all the file names corresponding to the arc images. Do the same for *arc_list,* but this time remove all the files corresponding to the science images. You can also remove the ***.fits*** extensions, IRAF is clever enough to recognize the format of the images.

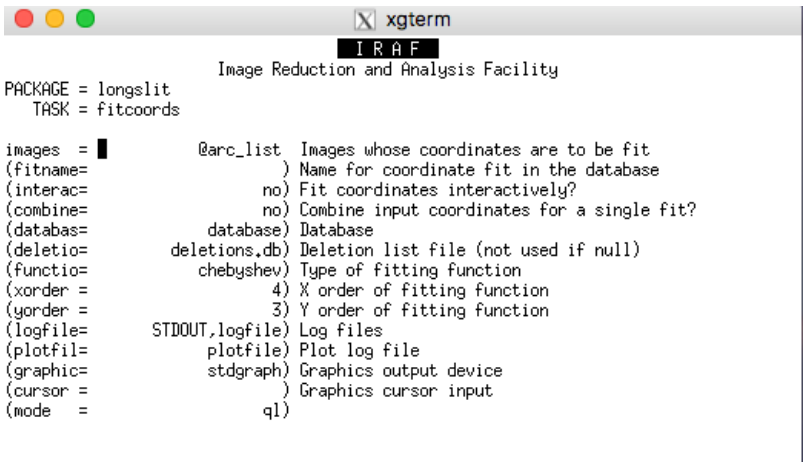
1.5.2 Automatic line identification

Before we run **reidentify** to automatically determine the wavelength solution for the remaining arcs you need to copy the directory called ***database*** to the directory containing all your arcs and science images. The ***database*** directory contains the wavelength solution for the first arc that we identified. Since the same arc was used for all the observations, we can use this solution as reference to identify the line features for the remaining arcs. Make the following changes to **reidentify**.



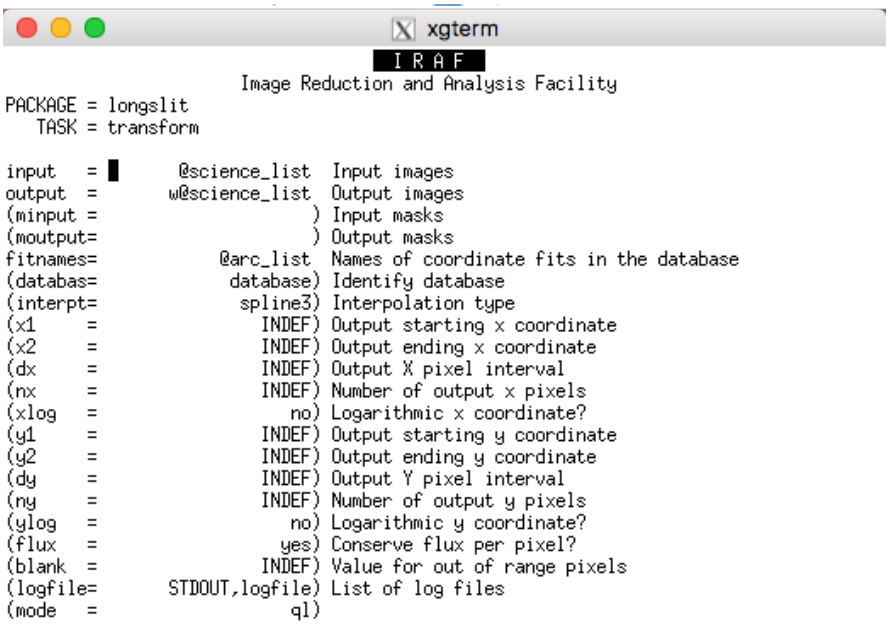
1.5.3 Automatic curvature solution

Following this we will run **fitcoord** to determine a curvature solution. Make the following changes to **fitcoord**:



1.5.3 Applying the curvature and wavelength solution

Following this we will run **`transform`** to apply the wavelength and curvature solution. Make the following changes to **`transform`**:



This will leave you with wavelength and curvature corrected science images - have a look at the output files starting with **"w"** in ds9. Following the same procedure, you can also feed IRAF a list of files to extract the spectra. I will leave this for you to do.

1. Measurements in IRAF

Once you've extracted the spectra you should have ~4 files with a **".ms.fits"** extension. We can now manipulate and make various measurements of the spectra. We will use **`splot`** to manipulate the spectra. First use **`files`** to make a list of the extracted spectra.

```
vocl> files *.ms.fits > extracted_list
```

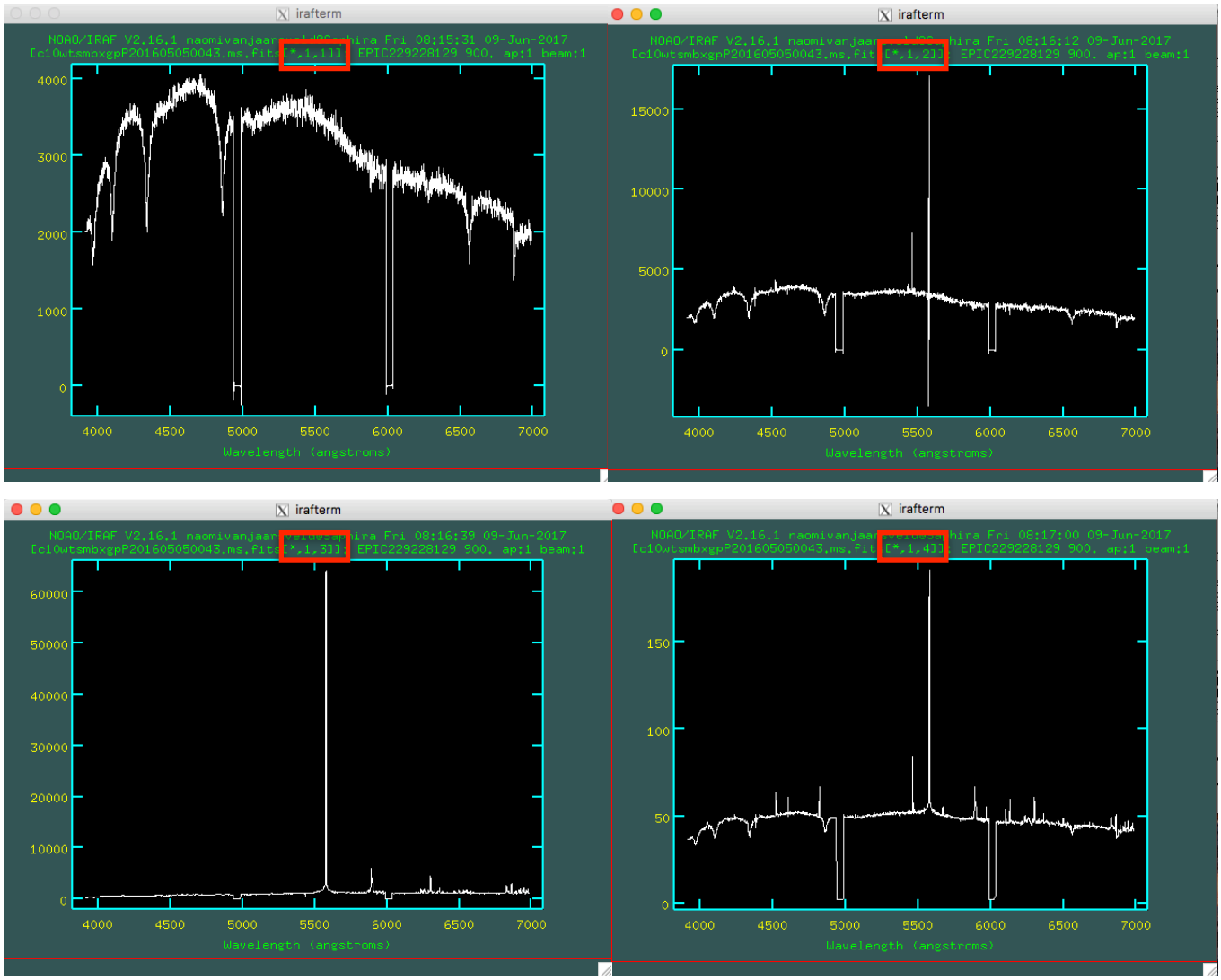
Now we can have a look at the spectra in **`splot`**.

```
vocl> splot @extracted_list
```

The **".ms"** extension is short for multi-spec. There are 4 different plots for each spectrum:

1. spectrum - background fit, weights variance, clean yes - **this is spectrum we want**
2. spectrum - background fit, weights none, clean no
3. Background (i.e. sky)
4. sigma - background fit, weights variance, clean yes

You can navigate between the different views by hitting **")"** to go forward and **"("** to go backwards. The last number highlighted in the red boxes will tell you in which view you are.



The first thing we will do is construct a SNR plot of the spectrum using **`sarith`** to divide the spectrum with the sigma "spectrum". The first entry is the spectrum, the **"/"** implies division, the second entry is the sigma spectrum, and finally the output file name. In the command line type:

```
vocl> sartih final.fits[*,1,1] / final.fits[*,1,4] EPIC229228129_SNR.fits
```

When I saved my spectrum after smudging out the outliers, I called it final. A more sensible name is advisable, e.g. the target name for instance. You can have a look at the SNR plot using **`splot`**.

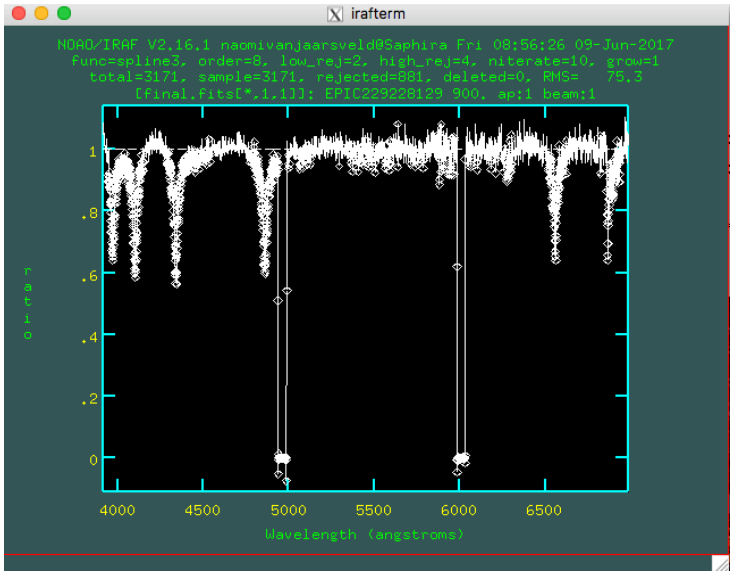
We will now manipulate the spectrum (first plot - **final.fits[*,1,1]**) and make various measurements using **`splot`**:

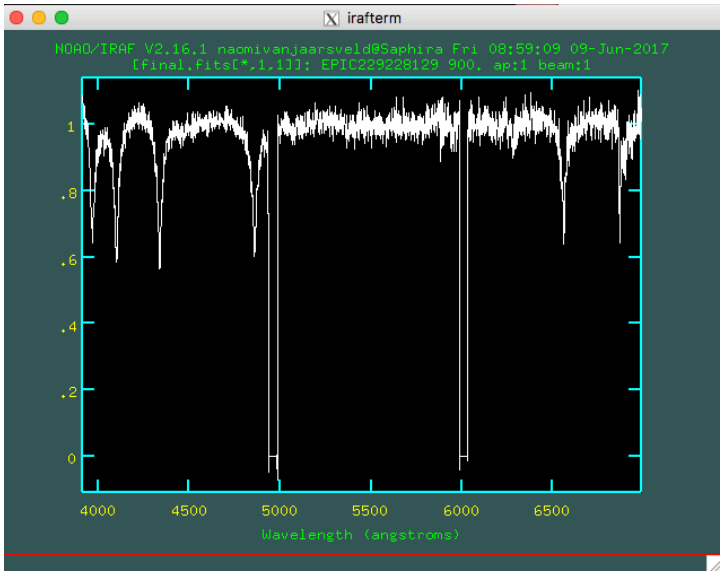
Flatten and Normalise

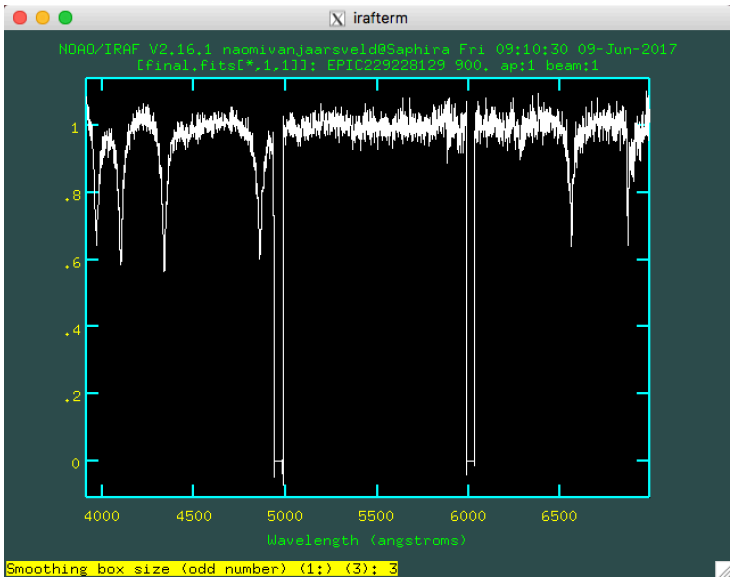
To view the spectrum type:

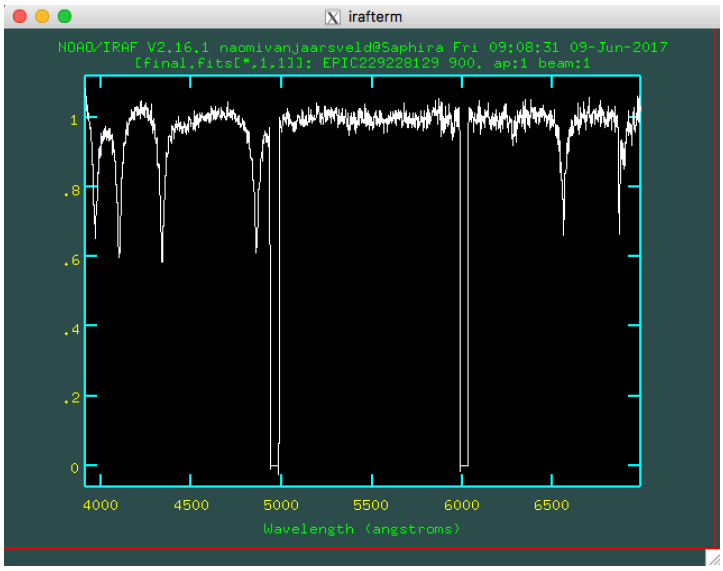
```
vocl> splot final.fits[*,1,1]
```

To normalise (the spectrum hit **"t"** followed by **"/"**, this will bring up a fitting window. You can increase the order to 8 (type **:order 8**) followed by **f** to fit the updated function. When you are happy with the fit hit **q**. Your spectrum will now be flattened and normalised. Finally we can smooth (lower panels) the spectrum by hitting **s**, and then setting the smoothing box size to **3**, and hitting enter. Save the updated spectrum by hitting **i** and giving the updated spectrum a sensible name (e.g. add **sn** as a prefix), and hit enter.



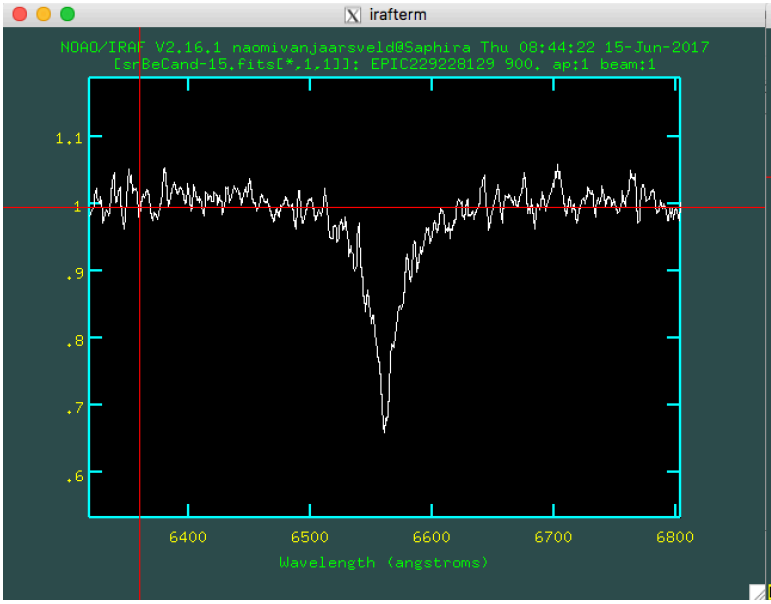


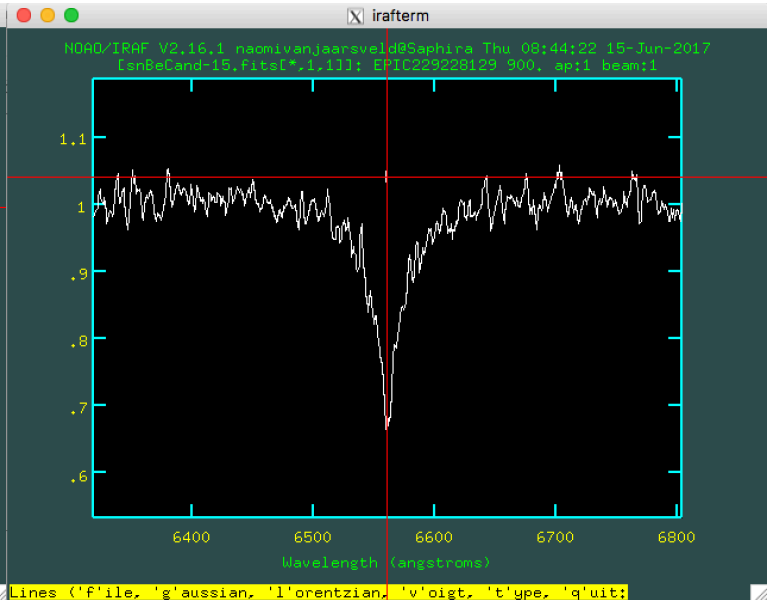


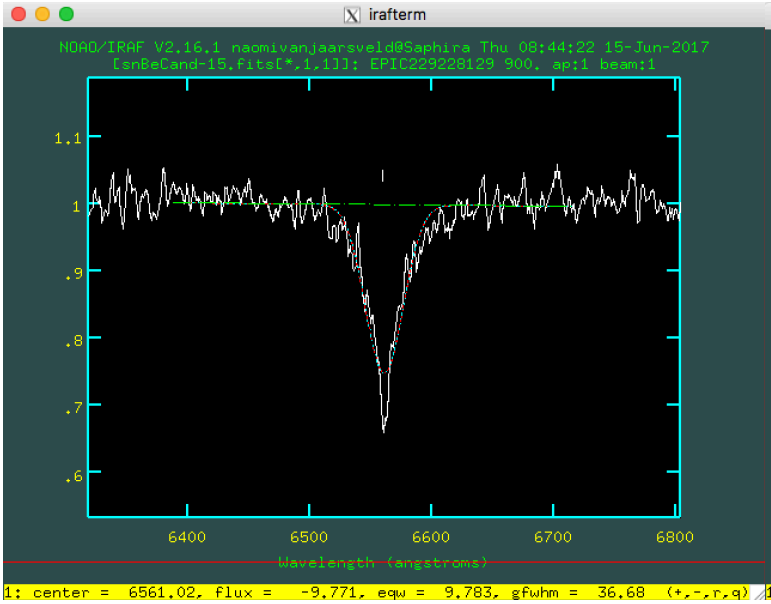


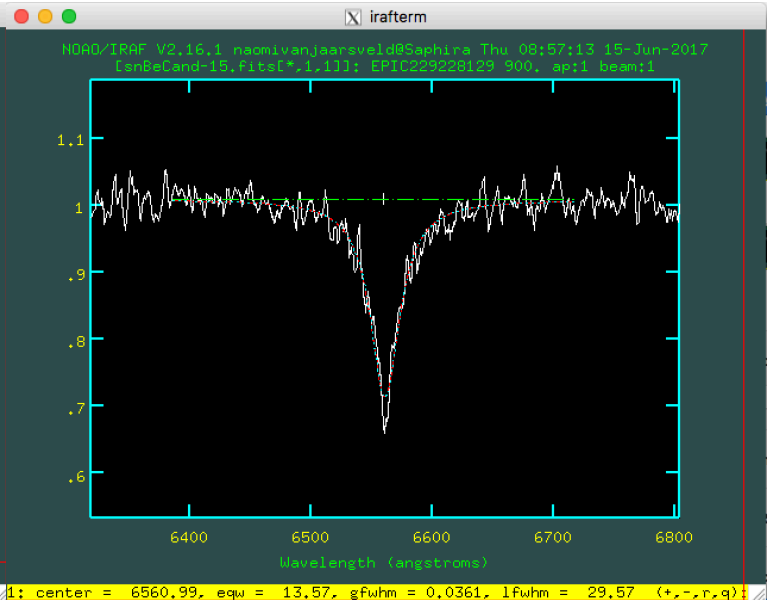
Measurements:

We can now take various measurements of the absorption/emission lines in spectra. We will continue to use **splot** for this. You can fit a gaussian to each line, but I will just do one example. By fitting a gaussian/lorentzian/voight to any given line, we can measure the equivalent width, centroid, and FWHM of the line. Typically try the gaussian first, and if that is a bad fit you can try the other profiles Plot the flattened and normalized spectra in **splot**. Now window (zoom) in to a particular line using **"w"** and **"e"** - the same as before. Hit **"d"** on either side of the absorption line (being careful to include enough of the continuum - top panel on the left), hover around the centroid of the line and hit either **"g", "l", "v"** for the 3 different profiles - top panel on the right. Let's try a gaussian fit first (hit **'g'**). Once the centroid is marked (you'll see a **dash** where you marked the centroid - top panel on the right) hit **"q"**, followed by **"a"** and **"a"** again - to fit all positions. Then hit **"y"** to fit the background, and **"b"** to display the fit and the background. At this point you will see that a gaussian profile did not fit the profile well - bottom panel on the left. You will also see the center, flux, eqw, and gwhm displayed at the bottom. Since the gaussian did not fit well you can redo it and try the lorentzian profile - bottom panel on the right. Just hit **"q"** twice, followed by **"r"** to redraw the spectrum.









Convert fits files to text files:

This is the final step :) Now that you have one dimensional spectra that are flattened, normalised and smoothed you can convert them to ascii files so that you can plot them in Python. We will use **wspectext** which is found in **onedspec** to convert the fits files to txt files. Make a list of all final spectra, also make a list with the names of the final spectra, but with a **.txt** extension instead of a **.fits** extension. Before running the command type **epar wspec** and set **header** to **no**, and quit by typing **:q** and hitting enter. Then simply type:

```
vocl> wspec @input_list @output_list
```

This will generate the **.txt** files that you specified in the ouput_list.