Data reduction of the echelle spectra using IRAF

Mauricio Cabezas
08.09.2021
Workshop on observational techniques
6-17 September 2021 at Ondřejov observatory

Contents

- What is IRAF?
- **O**ndřejov **E**chelle **S**pectrograph
- Diagram – reduction

– HANDS ON!

- Run – install IRAF
- oesred.cl, the ~pipeline
- Step by Step

IRAF

Image **R**eduction and **A**nalysis **F**acility

Is distributed by the National Optical Astronomy Observatories, which is operated by the Association of the Universities for Research in Astronomy, inc. (AURA) under cooperative agreement with the National Science Foundation

Highly portable, many platform installation is not an easy task!

Designed to be user-extensible.
>100 oficial external packages

#truestory: Everyone hates IRAF... but, why?

Still used for scientists

At the end of the day, you can do almost everything with IRAF!

Around 30 000 mentions of IRAF in ADS refereed papers

“The spectra were reduced, combined, and extracted using standard procedures in IRAF, ...”

“Standard data reduction procedures were performed by using the available IRAF tasks....”

Other community resources

Astroconda

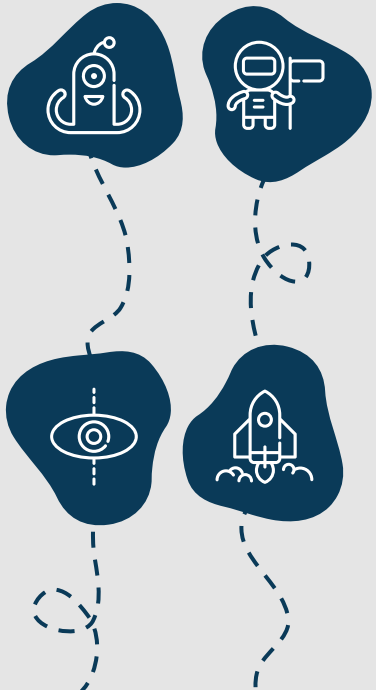
Actual STScI astronomy software distribution (Miniconda)

CFITSIO

FITS library and compression tools

WCSTOOLS

Image header utilities



DS9

Image display tool

STSDAS/TABLES

IRAF packages for statistics and use of tables

PyRAF

Python-based alternative to the IRAF

STScI | SPACE TELESCOPE SCIENCE INSTITUTE

2018 Volume 35 Issue 03

Back to All Articles

Removing the Institute's Dependence on IRAF (You can do it too!)

S. Ogaz (ogaz[at]stsci.edu) and E. Tollerud (etollerud[at]stsci.edu)

https://iraf-community.github.io/

Abstract

There's no way around it, the community-workhorse Image Reduction and Analysis Facility (IRAF) is getting on in years. It has served astronomy for three productive and fruitful decades and is appreciated by many. But as with many things in the software realm, the landscape has changed significantly since the inception of IRAF. Most modern astronomy analysis tools are built in languages like Python, IDL, and C/C++. As the tide has turned towards these newer languages, IRAF has become more and more difficult to build and maintain on current 64-bit architectures. A large portion of the IRAF tasks cannot be compiled as a 64-bit executable, and must be built as a 32-bit program. For these reasons the Space Telescope Science Institute (STScI) has been working towards IRAF independence for all our instrumentation and calibration work. This effort has included the development of transition resources, re-writes of older IRAF scripts, and some additions to Astropy (the current community-supported Python Astronomy package) when needed. If you are interested in transitioning from IRAF, this article is for you.

De-IRAFing the institute was made possible by both cross-divisional communications and effort within the institute, and making extensive use of GitHub. To ensure a smooth transition, the Data Analysis Tools Branch worked closely with the Instruments Division, the HST Mission Office and the Data Science Mission Office to gather the needs and requirements of STScI staff, as well as feedback and testing through the development of new tools. GitHub was an effective forum to track the work being done, as well as feedback and reviews from internal users. In particular, GitHub became indispensible for the project when communication was needed with Astropy on existing community tools, and for having Instruments Division staff review new transition content.

Ondřejov Echelle Spectrograph

Ondřejov Echelle Spectrograph (OES) is a fibre-fed high-resolution spectrograph. The spectrograph is installed in a temperature-stabilised room. The detector is a nitrogen-cooled EEV 2048×2048 pixel CCD, with a pixel size of 13.5 μm and a dynamical range of 65535 ADUs. The read-out noise is of about 3.5 e− rms, and a dark current of 1 e−/p/hr. In addition, the spectrograph is fed by a calibration lamp.

The wavelength coverage of OES is from near UV (**3753 Å**) up to near IR (**9195 Å**). The resolving power is R=51600 at 5000 Å (R~40000 in Hα) and spectral sampling is 2.4 Å/mm. The spectral range is covered by 56 usable orders. The number of spectral orders range from 92 to 36. The individual spectral order covers ~70 Å in the near UV region and ~145 Å in near IR regions. In blue, spectral orders overlap, thus, it is possible to merge them. The overlapping interval is ~20 Å in near UV.

A more detailed technical description with mechanical setup and all optical elements can be found in the report from the installation phase of the OES (Koubský at al., Ondřejov Echelle Spectrograph - OES, Publ. Astron. Inst. ASCR, 92:37–43, 2004)*.
**(https://stelweb.asu.cas.cz/web/index.php?section=telescope_instrumentation)*

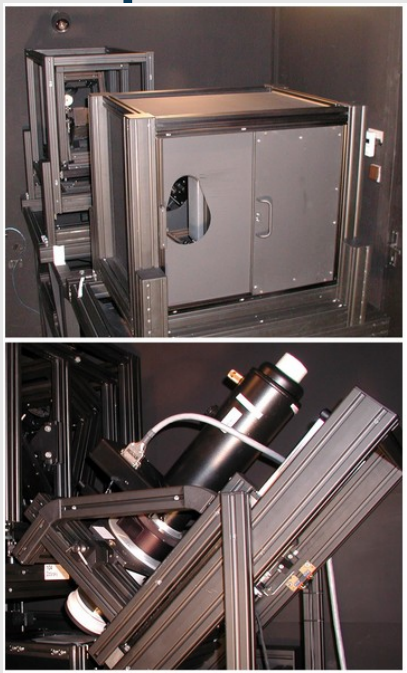
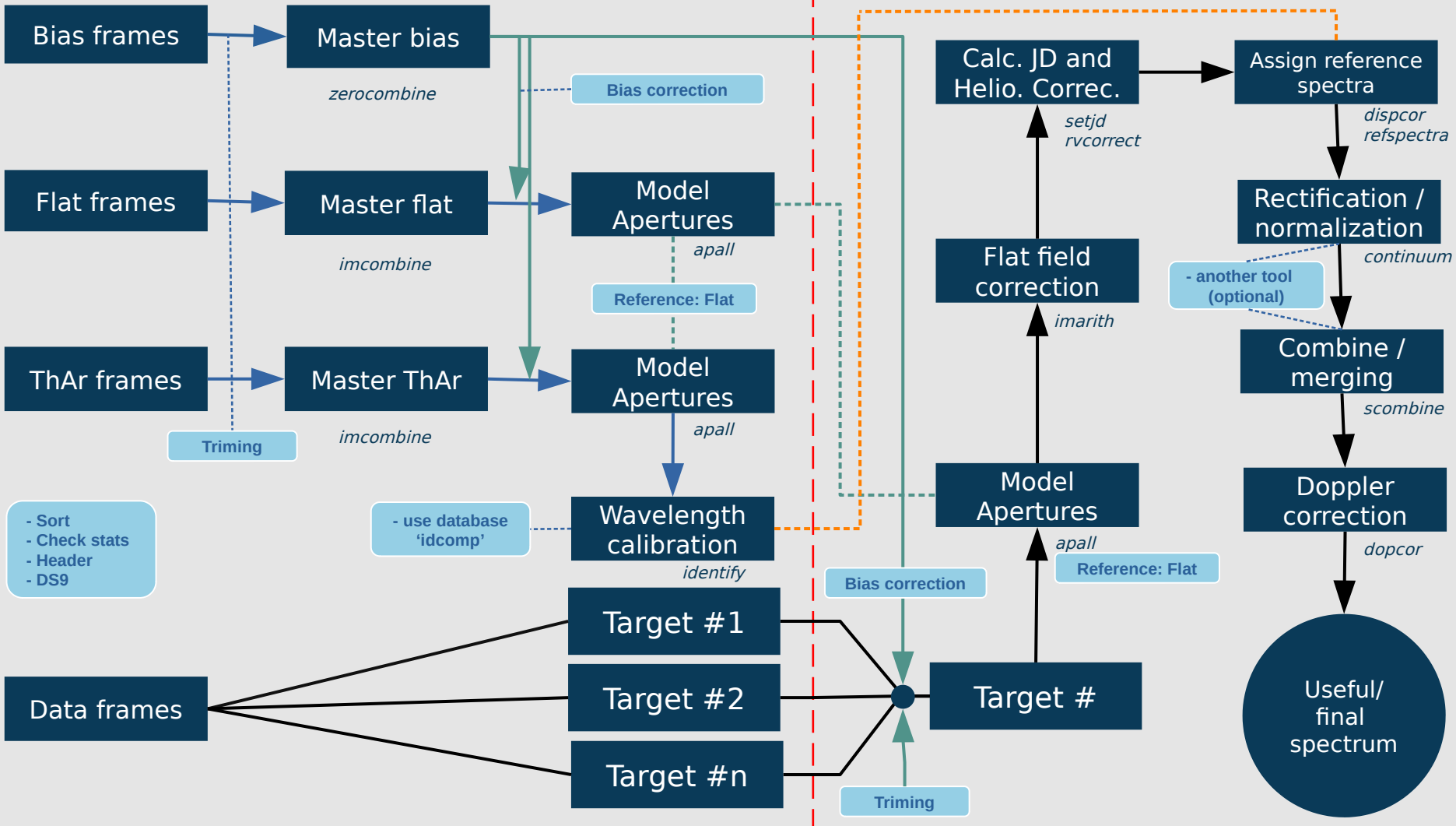


Fig. 2: Upper image - the mechanical parts holding the optical elements of the Ondrejov Echelle Spectrograph. Lower image - the detector with the dewar wessel. Credit: Miroslav Šlechta.


Bias frames
Master bias
zerocombine
Bias correction
Flat frames
Master flat
imcombine
Model Apertures
apall
Reference: Flat
ThAr frames
Master ThAr
imcombine
Model Apertures
apall
Triming
- Sort
- Check stats
- Header
- DS9
- use database 'idcomp'
Wavelength calibration
identify
Data frames
Target #1
Target #2
Target #n
Bias correction
Triming
Target #
Model Apertures
apall
Reference: Flat
Flat field correction
imarith
Calc. JD and Helio. Correc.
setjd
rvcorrect
Assign reference spectra
dispcor
refspectra
Rectification / normalization
continuum
- another tool (optional)
Combine / merging
scombine
Doppler correction
dopcor
Useful/ final spectrum

HANDS ON.!

- run IRAF
- http://www.asu.cas.cz/~cabezas/workshop/

ssh -X student1@merak.asu.cas.cz

IRAF installation

(tested in Linux/Ubuntu)

ttps://faculty1.coloradocollege.edu/~sburns/courses/18-19/pc362/Anaconda_IRAF_install.html

1) install CONDA

wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh

2) install IRAF 64bit

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

3) instalar Astroconda (will take time)

conda config --add channels http://ssb.stsci.edu/astroconda
conda create -n iraf27 python=2.7 iraf-all pyraf-all stsci

4) xgterm in 64bit

sudo apt-get install libxmu6:i386

5) launch astrocondA

conda activate iraf27
mkiraf
xgterm
cl

 # in order to open IRAF easily, add some alias in .bashrc

alias iraf="conda activate PATH/miniconda3/envs/iraf27; xgterm; cl"

(check the path where you have xgterm)

alias xgterm="exec PATH/miniconda3/envs/iraf27/bin/xgterm -font *-18-* -sb -fg "grey" -bg "black" -e cl"

IRAF - first steps

https://iraf.net

help task

Plot spectrum

splot spec.fit

Plot set of spectra

specplot @spec.list

specplot e*.fit,01.fit,02.fit...

Check header

imhead spec.fit/@spec.list l+ | page

Select some field from header

hselect spec.fit/@spec.list $I,obj-name,exptime yes

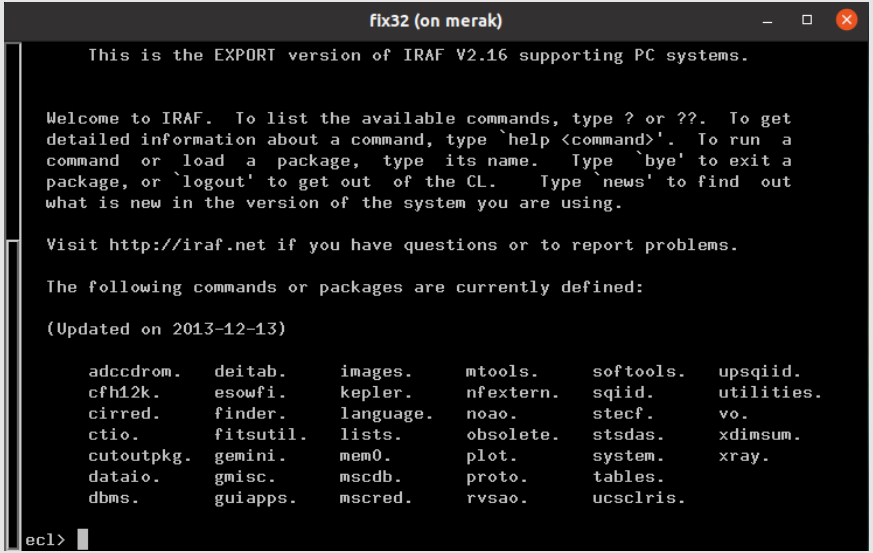
Check stats of spectrum

imstat spec.fit/@spec.list

See image with ds9

!ds9 &

display spec.fit Nframe (nframe=1,..,12)



EDIT:

1) login.cl line ~34

set stdimage = **imt4096**
set imextn = "oif:imh fxf:**fit,fits** ..."

2) inlcude data of Ondrejov observatory in the database, edit file **obsdb.dat** (path: ~noao/lib/obsdb.dat)

observatory = "ondrejov"
name = "Ondrejov observatory"
longitude = 345:12:59
latitude = 49:54:38
altitude = 528
timezone = -1

OESRED.CL

task oesred=path/oesred.cl

e 2021 09 08 0001.fit

year **month** **day** **id-numer**

- SEMI automatic

- always check your data!
  - imstat
  - ds9
  - splot

- Divided in two parts: Calibration and Science.

- Parameters where tested and works exclusively for OES.

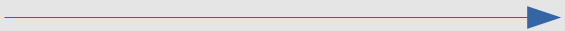
- for now, recommend stepbystep

- Still fix some problems / logical errors

- Future work -> **Python**! (in progress)

epar oesred *(or any task- **e**dit **par**ameter)*

Quit ***:q***
Go! ***:g***



fix32 (on merak)

```
                                  I R A F
                       Image Reduction and Analysis Facility
PACKAGE = clpackage
   TASK = oesred

input   = e202102040008.fit  Spectrum target to reduce(.fit)
(output =             hd54482) Output filename
(idtarge=           HD 54482) Target name on header
(napertu=                 49) Number of apertures to be found
(id     =               0008) Observation id number

                              # CALIBRATION PARAMETERS

(orgfile=                 no) do you want organize files?
(zerocom=                 no) Combine zero level images?
(trimcal=                 no) Trim flat and comp?
(iftrimc=                 no) Use trim flat & comp?
(zerocor=                 no) Apply zero level correction to flat & comp?
(compcom=                 no) Combine comparison lamp images?
(flatcom=                 no) Combine flat field images?
(flatapa=                 no) Extract flat apertures?
(compapa=                 no) Extract comparison apertures?
(iddatab=                 no) Use database folder for identification?
(idfolde=              idcomp) folder name with identification database
(idencom=                 no) Identify features in spectrum for dispersion sol

                              # OBJECT PARAMETERS

(trimob =                 no) Trim object?
(iftrimo=                 no) Use trim object?
(zerocor=                 no) Apply zero level correction to object?
(crays  =                 no) Remove cosmic rays?
(ifcrays=                 no) Use object with cosmic rays extraction?
(objecta=                 no) Extract object apertures?
(flatcor=                 no) Apply flat correction to object?
(helioco=                 no) calculate JD + heliocentric correction?
(idref  =                 no) refer database identification to images?
(norm   =                 no) normalize spectra?
(ncombin=                 no) combine normalized spcectra?
(mode   =                 ql)

                                                    ESC-? for HELP
```

1) Initial parameters!

Check type of “image”, can be:

Flat: flat fields

Zero: bias

Comp: ThAr comparison spectra

Object: science

- check header!

imhead e202109060001.fit l+ | page

- print list filename and object type

hselect e*.fit $I,object yes

- for example, we wanna start with alp Lyr.

- take note about the filename and the target name in the header!

```
                Image Reduction and Analysis Facility
PACKAGE = clpackage
   TASK = oesred

input   =    e202109060016.fit  Spectrum target to reduce(.fit)
(output =              alpLyr) Output filename
(idtarge=             alp Lyr) Target name on header
(napertu=                  49) Number of apertures to be found
(id     =                0016) Observation id number
```

ALWAYS keep raw data as backup!!

fix32 (on merak)

```
SGH-OIC =                   2 / OES Iodine cell
TM-DIFF =                  -1 / T1630950141 - P1630950142
OBJECT  = 'zero    '         / Title of observation
IMAGETYP= 'zero    '         / Type of observation, eg. FLAT
OBSERVER= 'Geier Rezba'      / Observers
SYSVER  = 'PESO exported.exp'
READSPD = '100kHz  '
FILENAME= 'e202109060001.fit'
CAMFOCUS=                3080. / Camera focus position
SPECTEMP=                 21.8 / 17912
SPECFILT=                    0 / Spectral filter
SLITHEIG=                 1.07 / Slit hight in mm
TM_START=                63741 / 17:42:21, 1630950141
UT      = '17:42:21'         / UTC of  start of observation
EPOCH   =                2000. / Same as EQUINOX - for back compat
EQUINOX =                2000. / Equinox of RA and DEC
DATE-OBS= '2021-09-06'       / UTC date start of observation
TM_END  =                63742 / 17:42:22, 1630950142
EXPTIME =                    1 / Length of observation excluding pauses
DARKTIME=                    1 / Length of observation including pauses
CCDTEMP =                 -110 / Detector temperature
STDIN-line 63-file 1 of 1
```

fix32 (on merak)

```
e202109060008.fit       flat
e202109060009.fit       flat
e202109060010.fit       flat
e202109060011.fit       comp
e202109060012.fit       comp
e202109060013.fit       comp
e202109060014.fit       comp
e202109060015.fit       comp
e202109060016.fit       "alp Lyr"
e202109060017.fit       "alp Lyr"
e202109060018.fit       "HD 10780"
e202109060019.fit       "BD+44 2417"
e202109060020.fit       "HD 153911"
e202109060021.fit       "KIC 11134982"
e202109060022.fit       "HD 209027"
e202109060023.fit       "HD 194905"
e202109060024.fit       comp
e202109060025.fit       comp
e202109060026.fit       comp
e202109060027.fit       comp
e202109060028.fit       comp
e202109060029.fit       comp
```

2) sort files!

- complete first set of parameteres, in the part of calibration set:

orgfile = yes

:go

After this step files are organized, you can check each folder and files.

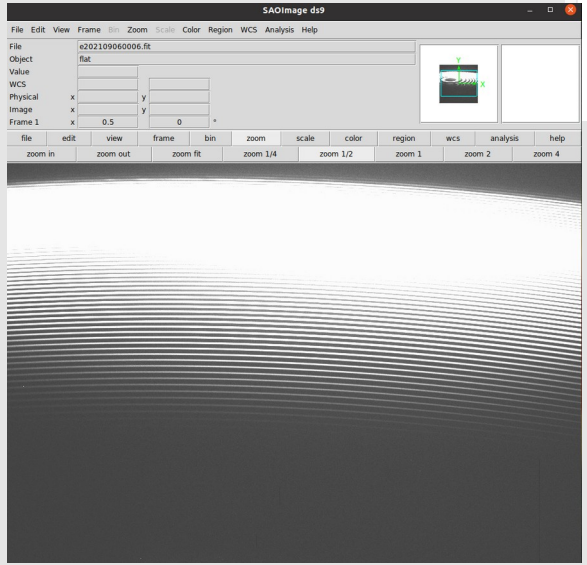
Example:

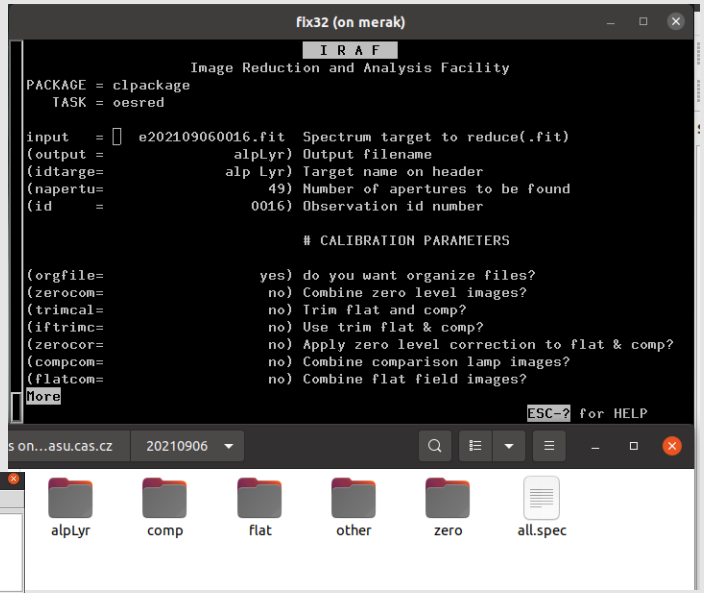
imstat @flat.dat

```
#              IMAGE     NPIX      MEAN     STDDEV      MIN      MAX
  e202109060006.fit  4194304     931.1     2621.       0.    25203.
  e202109060007.fit  4194304     949.1     2679.       0.    25797.
  e202109060008.fit  4194304     944.7     2664.       0.    25560.
  e202109060009.fit  4194304     941.4     2654.       0.    25521.
  e202109060010.fit  4194304     940.2     2649.       0.    25405.
```

!ds9 &

display e202109060006.fit 1





Before each run be sure we are in the main folder!

pwd

3) Combine zeros!

- before, is possible edit the list **zero.dat**
- change the previous task to “no”

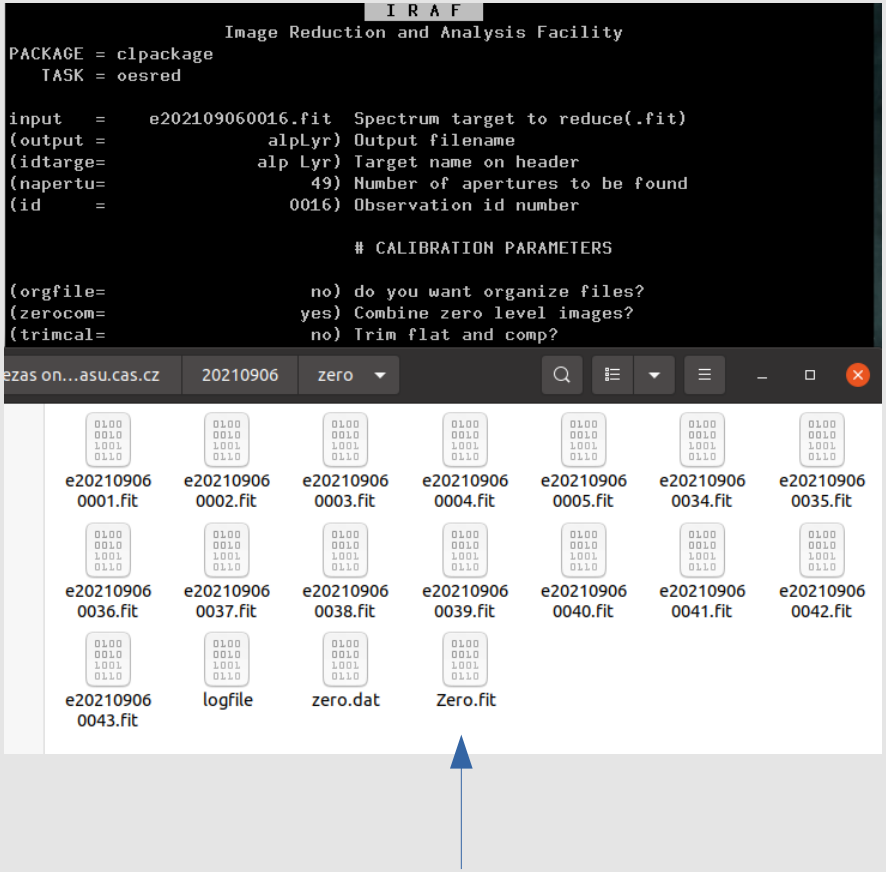
  orgfile = no

  zerocom = yes

  :go

- new file: **Zero.fit**

```
######  ZEROCOMBINE
if (access("zero/Zero.fit")){
     zerocomb=no
}
if (zerocomb==yes){
     cd "zero/"
     unlearn zerocombine
     zerocombine.reject="minmax"
     zerocombine.rdnoise= "READNOIS"
     zerocombine.gain    = "GAIN"
     zerocombine (input="@zero.dat",output="Zero.fit")
     cd "../"
}
```



4) trim flat and comp (optional)

- completely optional, but sometimes the first and last pixel of each aperture is saturated and can be annoying during normalization.

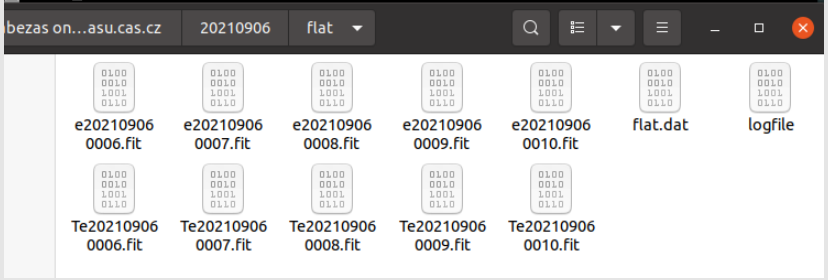
- lines 120 in oesred.cl

zerocom = no
trimcal = yes
iftrimc = yes
:go

- new files, prefix **T**
Te*.fit

```
117
118   if (trimcal==yes){
119        unlearn ccdproc
120        ccdproc.trimsec = "[2:2035,*]"
121   #    ccdproc.trimsec = "[5:2025,800:1500]"
122        ccdproc.trim = yes
123        ccdproc.fixpix = no
124        ccdproc.overscan = no
125        ccdproc.darkcor= no
126        ccdproc.zerocor=no
127        ccdproc.flatcor=no
128        #
129        cd "flat/"
130        ccdproc.ccdtype = "flat"
131        ccdproc (images="@flat.dat",output="T@flat.dat")
132        cd "../"
133        #
134        cd "comp/"
135        ccdproc.ccdtype = "comp"
136        ccdproc (images="@comp.dat",output="T@comp.dat")
137        cd "../"
138
139   }
140
```

```
                                  I R A F
                  Image Reduction and Analysis Facility
PACKAGE = clpackage
   TASK = oesred

input    =    e202109060016.fit  Spectrum target to reduce(.fit)
(output =                alpLyr) Output filename
(idtarge=               alp Lyr) Target name on header
(napertu=                    49) Number of apertures to be found
(id     =                  0016) Observation id number

                                 # CALIBRATION PARAMETERS

(orgfile=                    no) do you want organize files?
(zerocom=                    no) Combine zero level images?
(trimcal=                   yes) Trim flat and comp?
(iftrimc=                   yes) Use trim flat & comp?
(zerocor=                    no) Apply zero level correction to flat & comp?
(compcom=                    no) Combine comparison lamp images?
(flatcom=                    no) Combine flat field images?
(flatapa=                    no) Extract flat apertures?
(compapa=                    no) Extract comparison apertures?
More
ecl>
```



If you apply it and you wanna use the trimmed images, you need to set iftrimc=yes during the whole reduction!

5) Bias correction flat and comp

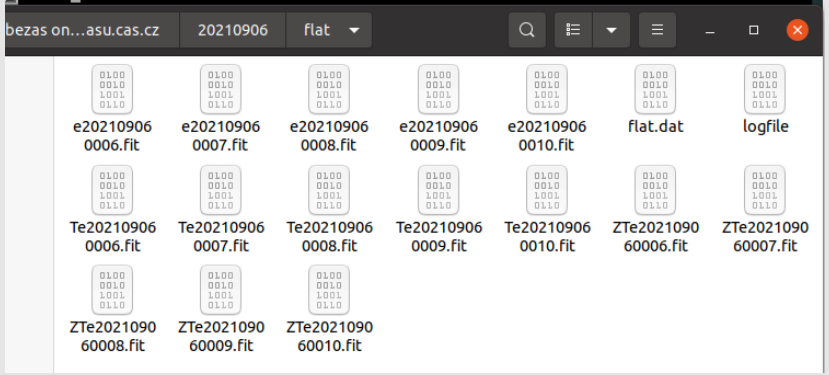
trimcal = no
iftrimc = yes
zerocor = yes
:go

- new files, prefix **Z**

ZTe*.fit

```
143  ###### SUBTRACT zero
144  if (zerocorcal==yes){
145  #    imarith (operand1="flat/flat.fits", op="/", operand2="zero/Zero.fits", result="Zflat.fits")
146  #    imarith (operand1="comp/comp.fits", op="/", operand2="zero/Zero.fits", result="Zcomp.fits")
147       unlearn ccdproc
148       ccdproc.ccdtype="zero"
149  #    ccdproc.trimsec = "[5:2025,800:1500]"
150       ccdproc.fixpix = no
151       ccdproc.overscan = no
152       ccdproc.darkcor= no
153       ccdproc.zerocor=no
154       ccdproc.flatcor=no
155       #
156       cd "flat/"
157       ccdproc.ccdtype="flat"
158       ccdproc.zerocor=yes
159       ccdproc.zero="../zero/Zero.fit"
160            if (iftrimc==yes){
161            ccdproc (images="T@flat.dat",output="ZT@flat.dat")
162            } else {
163            ccdproc (images="@flat.dat",output="Z@flat.dat")
164            }
165       cd "../"
166       #
167       cd "comp/"
168       ccdproc.ccdtype="comp"
169       ccdproc.zerocor=yes
170       ccdproc.zero="../zero/Zero.fit"
171                 if (iftrimc==yes){
172            ccdproc (images="T@comp.dat",output="ZT@comp.dat")
173            } else {
174            ccdproc (images="@comp.dat",output="Z@comp.dat")
175            }
176       cd "../"
177  }
```

```
                    I R A F
          Image Reduction and Analysis Facility
PACKAGE = clpackage
   TASK = oesred

input   =    e202109060016.fit  Spectrum target to reduce(.fit)
(output =              alpLyr) Output filename
(idtarge=             alp Lyr) Target name on header
(napertu=                   49) Number of apertures to be found
(id     =                 0016) Observation id number

                                # CALIBRATION PARAMETERS

(orgfile=                   no) do you want organize files?
(zerocom=                   no) Combine zero level images?
(trimcal=                   no) Trim flat and comp?
(iftrimc=                  yes) Use trim flat & comp?
(zerocor=                  yes) Apply zero level correction to flat & comp?
(compcom=                   no) Combine comparison lamp images?
(flatcom=                   no) Combine flat field images?
(flatapa=                   no) Extract flat apertures?
(compapa=                   no) Extract comparison apertures?
More
ecl>
```



6) Combine comp (ThAr)

iftrimc = yes ← (now always “yes”)
zerocor = no
compcom = yes
:go

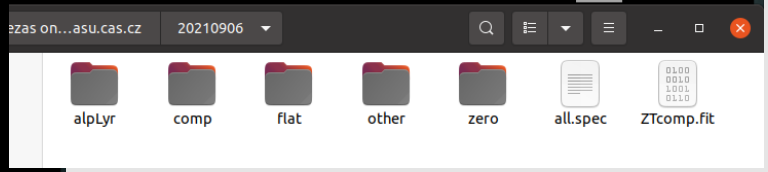
- new file
ZTcomp.fit

7) Combine flat field

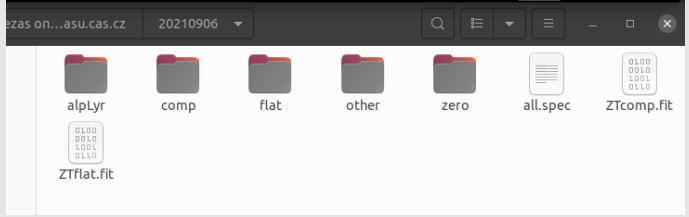
iftrimc = yes
compcom = no
flatcom = yes
:go

- new file
ZTflat.fit

```
                    I R A F
          Image Reduction and Analysis Facility
PACKAGE = clpackage
   TASK = oesred

input   = e202109060016.fit  Spectrum target to reduce(.fit)
(output =              alpLyr) Output filename
(idtarge=             alp Lyr) Target name on header
(napertu=                   49) Number of apertures to be found
(id     =                 0016) Observation id number

                              # CALIBRATION PARAMETERS

(orgfile=                   no) do you want organize files?
(zerocom=                   no) Combine zero level images?
(trimcal=                   no) Trim flat and comp?
(iftrimc=                  yes) Use trim flat & comp?
(zerocor=                   no) Apply zero level correction to flat & comp?
(compcom=                  yes) Combine comparison lamp images?
(flatcom=                   no) Combine flat field images?
(flatapa=                   no) Extract flat apertures?
(compapa=                   no) Extract comparison apertures?
More
                                                    ESC-? for HELP
```




```
                    I R A F
          Image Reduction and Analysis Facility
PACKAGE = clpackage
   TASK = oesred

input   = e202109060016.fit  Spectrum target to reduce(.fit)
(output =              alpLyr) Output filename
(idtarge=             alp Lyr) Target name on header
(napertu=                   49) Number of apertures to be found
(id     =                 0016) Observation id number

                              # CALIBRATION PARAMETERS

(orgfile=                   no) do you want organize files?
(zerocom=                   no) Combine zero level images?
(trimcal=                   no) Trim flat and comp?
(iftrimc=                  yes) Use trim flat & comp?
(zerocor=                   no) Apply zero level correction to flat & comp?
(compcom=                   no) Combine comparison lamp images?
(flatcom=                  yes) Combine flat field images?
(flatapa=                   no) Extract flat apertures?
(compapa=                   no) Extract comparison apertures?
More
                                                    ESC-? for HELP
```



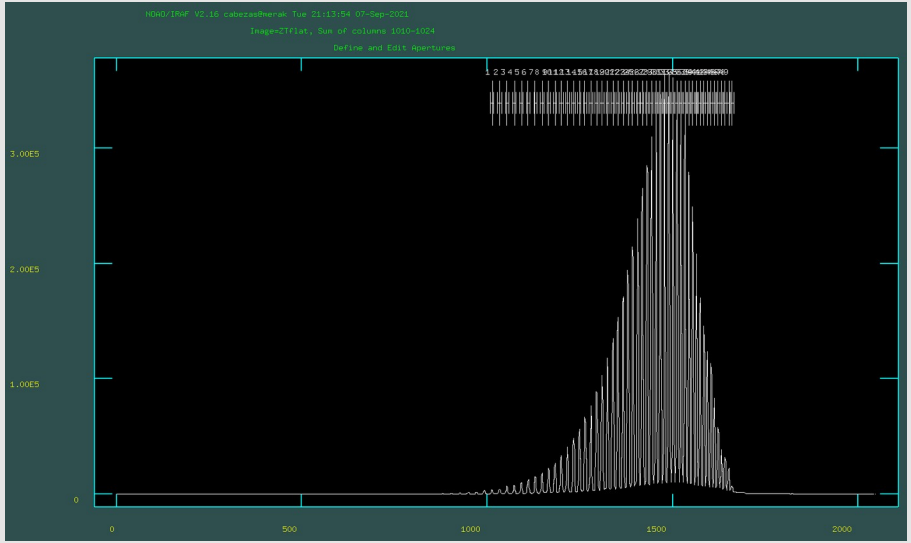

```
178   ############
179
180   ###### COMBINE - comp/lamp
181   if (compcomb==yes){
182        cd "comp/"
183        unlearn imcombine
184        # imcombine.reject = "none"
185        imcombine.reject = "none"
186        imcombine.lsigma = 3
187        imcombine.hsigma = 3
188        imcombine.rdnoise= "READNOIS"
189        imcombine.gain   = "GAIN"
190        imcombine.scale = "exposure"
191        imcombine.expname="EXPTIME"
192   #    imcombine (input="@comp.dat",output = "comp.fits")
193             if (iftrimc==yes){
194             imcombine (input="ZT@comp.dat",output = "../ZTcomp.fit")
195             } else {
196             imcombine (input="Z@comp.dat",output = "../Zcomp.fit")
197             }
198        cd "../"
199   }
200
```

8) Model apertures - Flat

iftrimc = yes
flatcom = no
flatapa = yes
:go

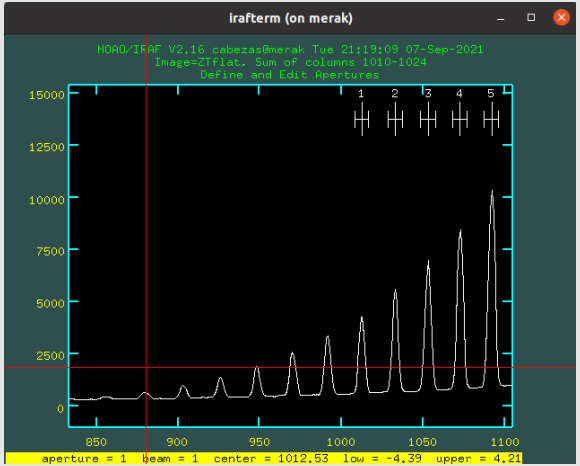
Find apertures for Ztflat? (yes):
Number of apertures to be found automatically (49):
Resize apertures for ZTflat? (yes):
Edit apertures for Ztflat? (yes):

- accept everything with “enter” or typing “yes”. Default answer in parenthesis.



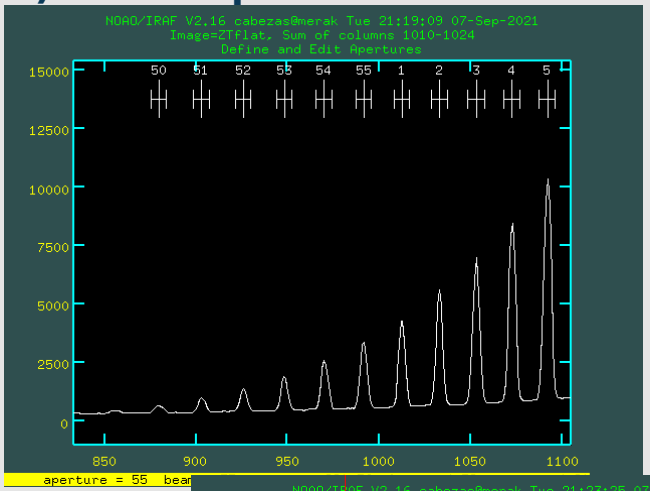
- in order to do everything a bit easy/fast. We will use a database for wavelength calibration, for that we need to choose 49 apertures.
  - first aperture near the pixel 890 mark: **m**

Background fitting is not neccesary because overlapping.

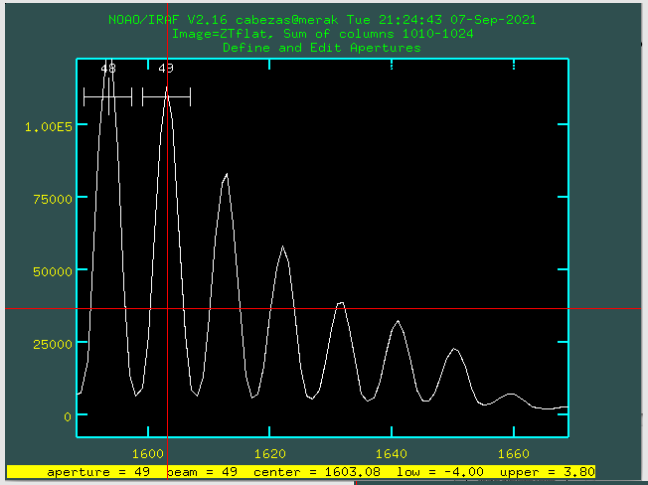



```
   TASK = oesred

input   =    e202109060016.fit  Spectrum target to reduce(.fit)
(output =              alpLyr) Output filename
(idtarge=             alp Lyr) Target name on header
(napertu=                  49) Number of apertures to be found
(id     =                0016) Observation id number

                                 # CALIBRATION PARAMETERS

(orgfile=                  no) do you want organize files?
(zerocom=                  no) Combine zero level images?
(trimcal=                  no) Trim flat and comp?
(iftrimc=                 yes) Use trim flat & comp?
(zerocor=                  no) Apply zero level correction to flat & comp?
(compcom=                  no) Combine comparison lamp images?
(flatcom=                  no) Combine flat field images?
(flatapa=                 yes) Extract flat apertures?
(compapa=                  no) Extract comparison apertures?
More
Find apertures for ZTflat?  (yes):
Number of apertures to be found automatically (49):
Resize apertures for ZTflat?  (yes):
Edit apertures for ZTflat?  (yes):
```

9) Model apertures - Flat

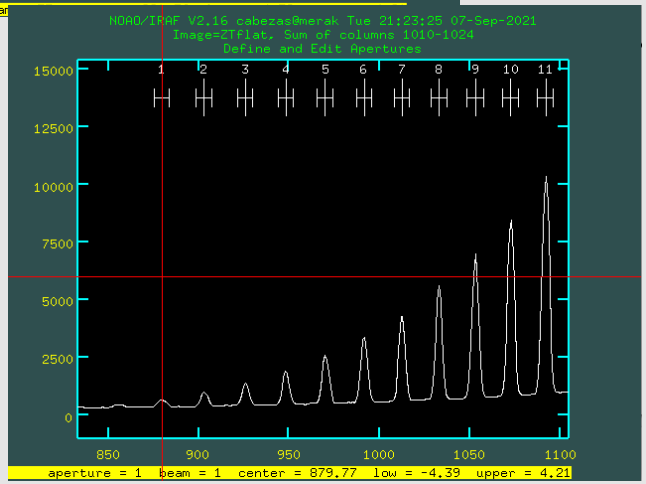


- delete last aperture: **d**

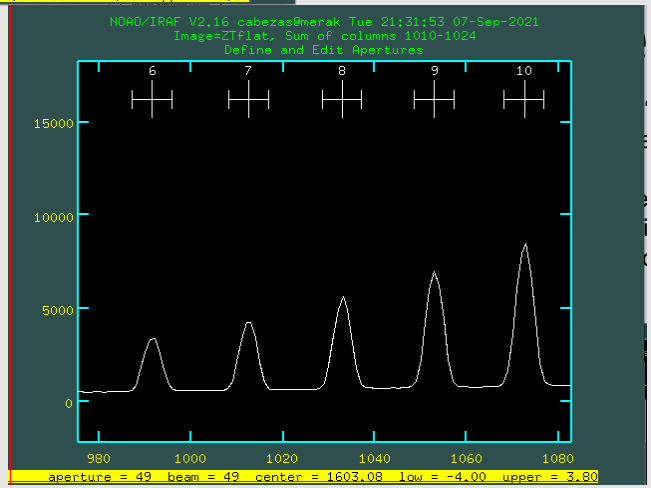



```
226  #####APERTURES - APALL FLAT
227  if (flatapall==yes){
228        echelle
229        unlearn apall
230        apall.format = "echelle"
231
232        apall.extras=no
233        apall.extract=yes
234
235        apall.nsum=15
236
237        apall.lower=-5
238        apall.upper=5
239        apall.b_order=3
240        apall.b_sample="-10:-6,6:10"
241
242        apall.nfind=nap
243        #apall.minsep=10
244        apall.minsep=5
245        apall.maxsep=1000
246
247
248        apall.ylevel = 0.04
249        apall.bkg=yes
250        #apall.bkg=no
251
```

Sort: **o**
”1”



- delete last aperture: **d**



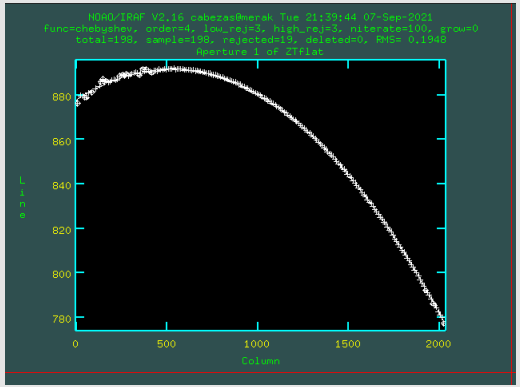
9) Model apertures - Flat

Trace apertures for Ztflat? (yes):

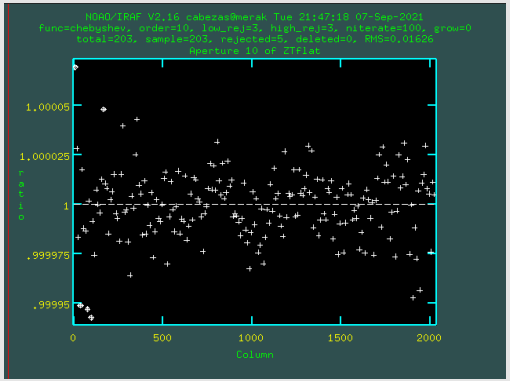
Fit traced positions for Ztflat interactively? (yes):

Fit curve to aperture 1 of Ztflat interactively (yes):

- accept everything with “enter” or typing “yes”. Default answer in parenthesis.



Polinomial fitting of the echelle orders



- change order **:o N**
- relative residuals **k**
- residuals (px) **j**
- aperture **h**

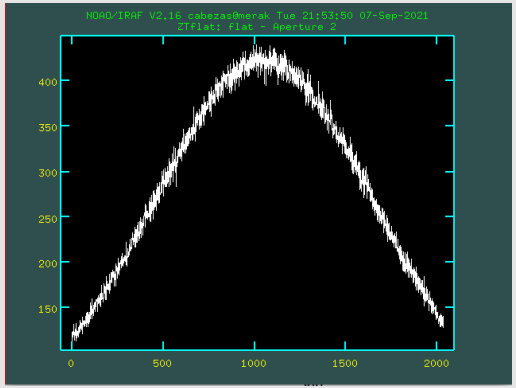
Fit curve to aperture N of Ztflat interactively (yes): *YES*

Write apertures for ZTflat to database (yes):

Extract aperture for ZTflat? (yes):

Review extracted spectra from ZTflat? (yes):

Review extracted spectrum for aperture 1 from ZTflat? (yes)
YES



- new file: **AZTflat.fit**

10) Model apertures – Comparison lamp

- Template: AZTflat.fit

iftrimc = yes

flatapa = no

compapa = yes

:go

Edit apertures for Ztflat? (yes):

Can be “no” but always is better to check it!

q

Write apertures for Ztcomp to database (yes):

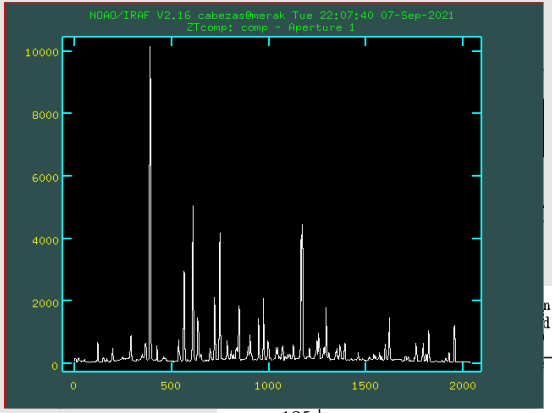
Extract aperture spectra for Ztcomp? (yes):

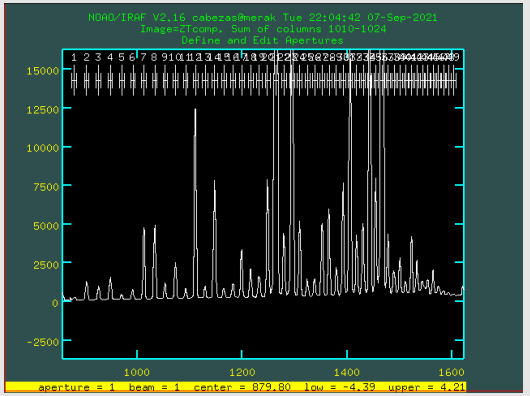
Review extracted spectra from Ztcomp? (yes):

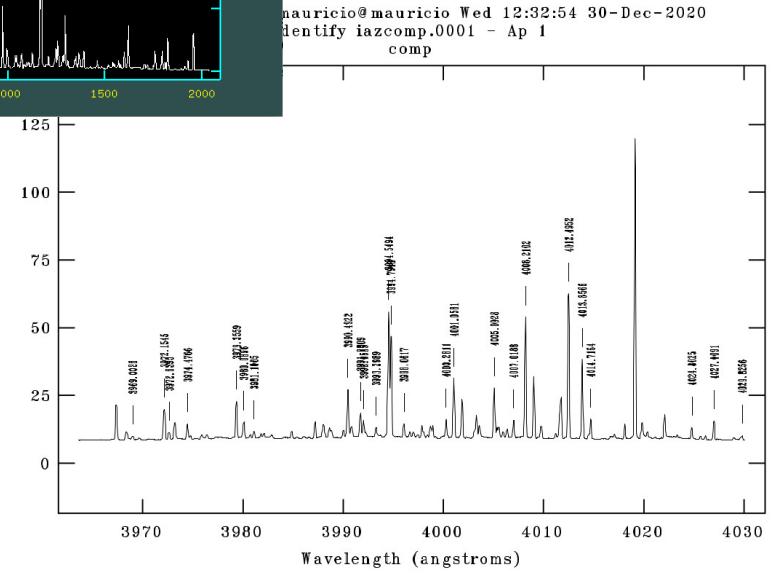
Review extracted spectrum for aperture 1 from ZTcomp? (yes)

- if you are using database, check if the first aperture corresponds to the first aperture in the atlas. (inside folder ***idcomp***)

- new file: **AZTcomp.fit**








```
271   #####APERTURES - COMP
272   if (compapall==yes){
273         apall.referen=inflat
274         apall.format = "echelle"
275         apall.find=no
276         apall.recente=no
277         apall.resize=no
278         apall.trace=no
279         apall.fittrace=no
280         apall.extras=no
281         apall.ylevel = 0.04
282         apall.extract=yes
283         apall (input=incomp, output="A"//incomp//".fit")
284   }
```

12) Wavelength calibration

- using database, folder “idcomp” must be in the main path (20210906/)

iftrimc = yes
compapa = no
iddatab = yes
idfolder = idcomp
idencom = yes
:go

- database can be shifted few pixels, in order to fix we need to “shift” the points of our database.

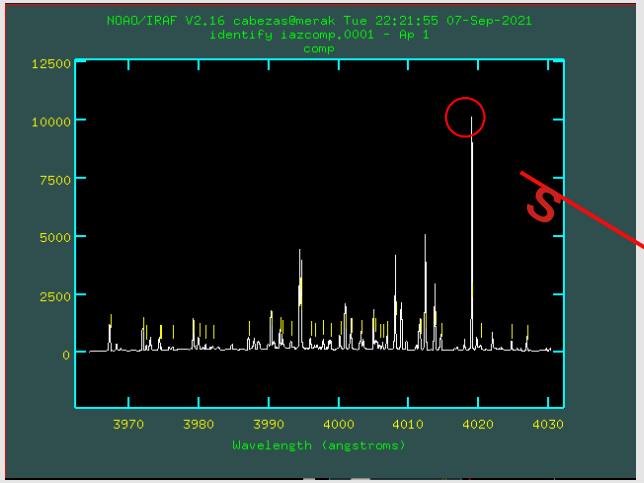
s

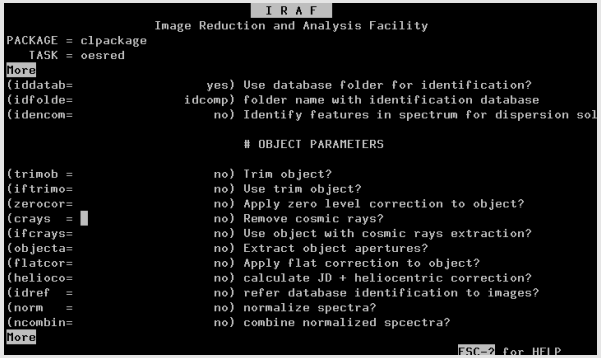
- always fit!

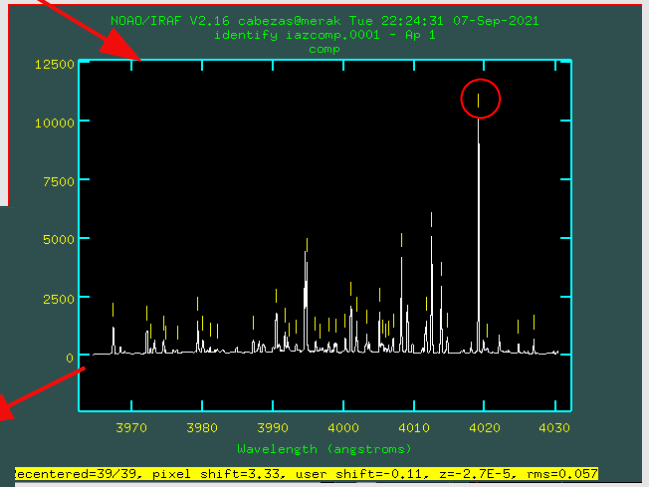
f (delete point d), **q, q**

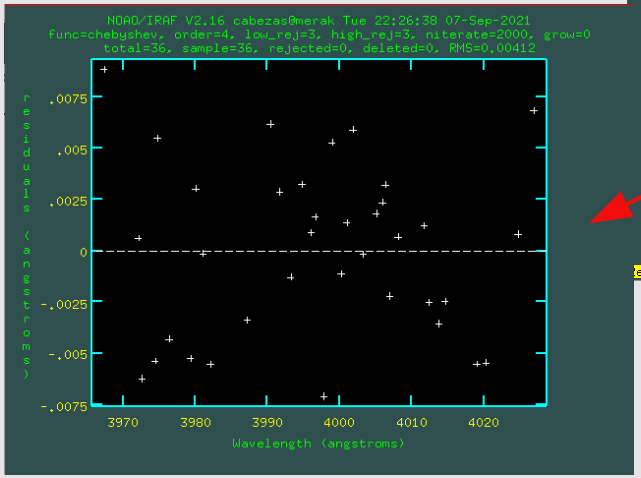
- rms ~ 0.003

- Write feature data to the database (yes)?









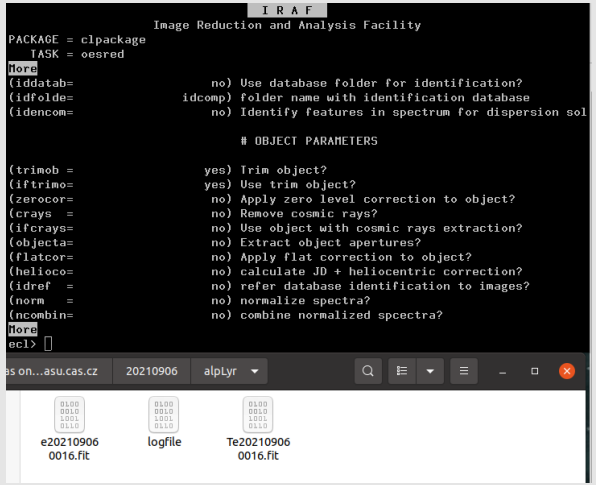
- mark lines **m**
- centering **c**
- add automatic lines **l**

13) Trim Object

iftrimc = yes
iddatab = no
idfolder = idcomp *(does not matter)*
idencom = no

trimob = yes
iftrimo = yes
:go

New file: **Te202109060016.fit**



14) Bias correction object

iftrimc = yes
trimob = no
iftrimo = yes
zerocor = yes
:go

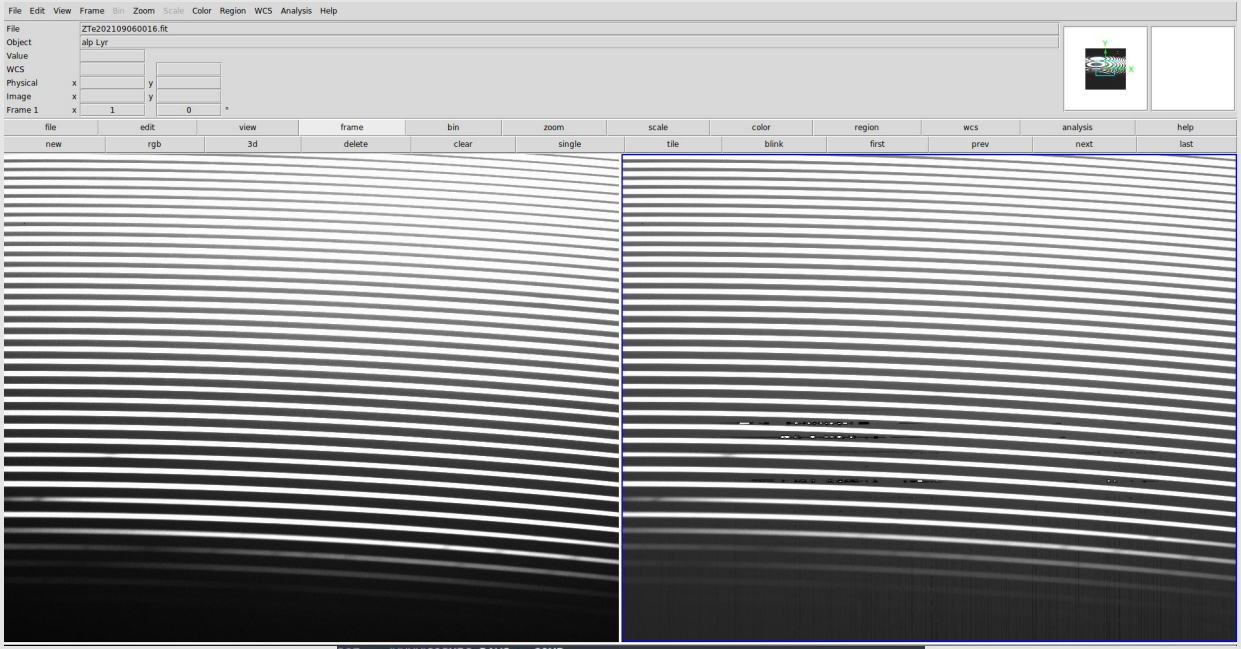
New file: **ZTe202109060016.fit**

15) Cosmic Rays – LACOS (2001PASP..113.1420V)

iftrimc = yes
iftrimo = yes
zerocor = no
crays = yes
ifcrays = yes
:go

New file: **CrZTe202109060016.fit**

For now in STAND-BY



Some issues with parameters
Best setup?
Different for each target/night/exposure
http://www.astro.yale.edu/dokkum/lacosmic/

```
397  #####COSMIC RAYS - COMP
398  if (crays==yes){
399       stsdas
400  #######
401  #read gain
402  cd (oname)
403  hselect (images="Z"//inobject,fields="GAIN", exp=yes) | scan (gainh)
404  print (gainh)
405  inputCR="Z"//inobject
406  outputCR="CrZ"//inobject
407  outmaskCR="MCrZ"//inobject
408  gainCR = gainh # 2 #3
409  readnCR = 2 #2
410  xorderCR = 3 #0 #3
411  yorderCR = 3 #0 #3
412  sigclipCR = 4.5 #0.1 #4.5
413  sigfracCR = 0.3 #0.5 #0.01
414  objlimCR = 1 #1. #1
415  niterCR = 5 #3
416  verboseCR = no
```

16) Model apertures - Object

iftrimc = yes
iftrimo = yes
crays = no
ifcrays = no
objecta = yes
:go

- Template: AZTflat.fit

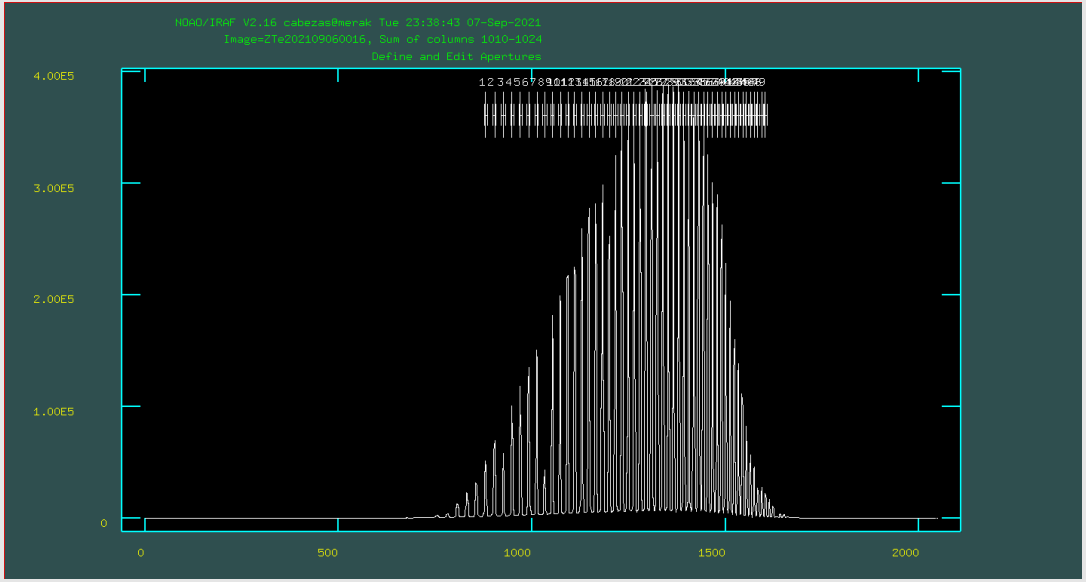
Edit apertures for ZTe202109060016? (yes):
Can be “no” but always is better to check it!

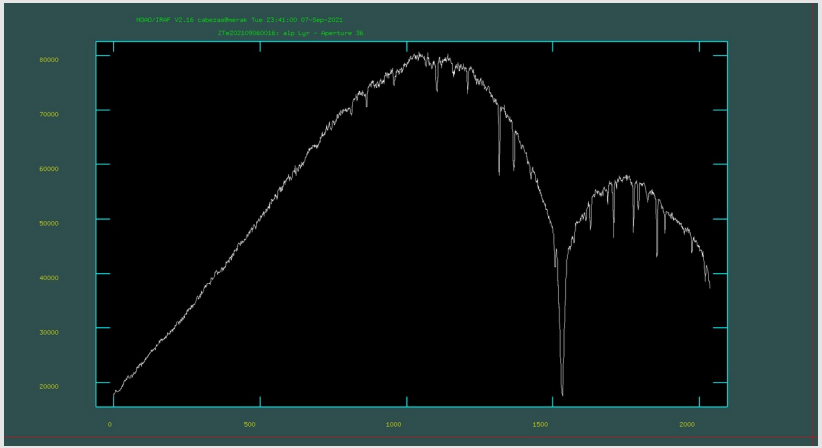
q

Review extracted spectrum for aperture 1 from ZTe202109060016? (yes) *YES*

- still each spectrum in pixels

new file: ***AZTe202109060016.fit***

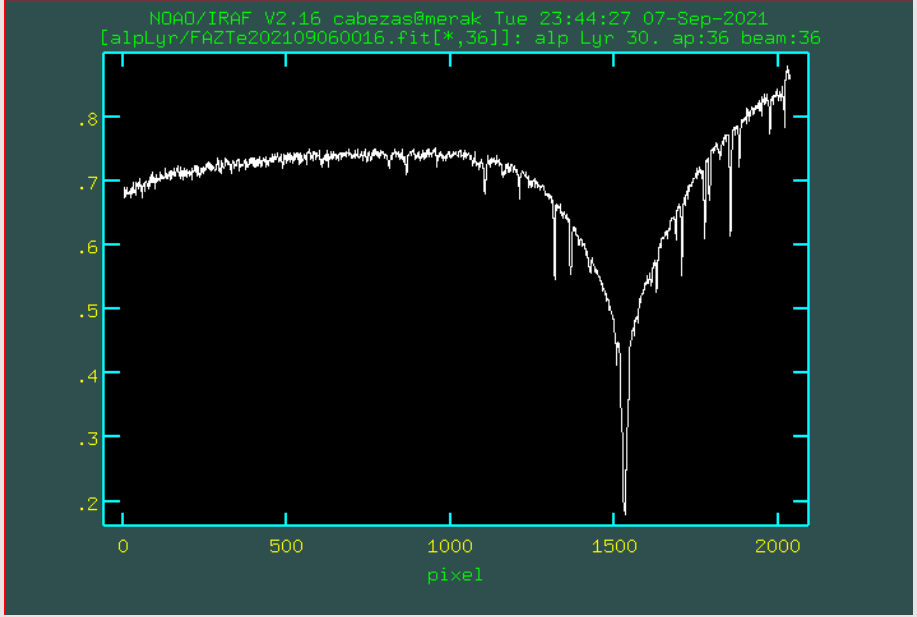




17) Flat correction

iftrimc = yes
iftrimo = yes
ifcrays = no
objecta = no
flatcor = yes
:go

new file: ***FAZTe202109060016*.fit**



18) JD + heliocentric correction

iftrimc = yes
iftrimo = yes
ifcrays = no
flatcor = no
helioco = yes
:go

New UTMIDDLE

```
334  ################################################ OBJECT
335  ###### create obeject folder
336  ###### always
337  unlear directory
338  directory.sort=no
339  directory oname//"/" | scan (iddir)
340  if (iddir=="no"){
341       mkdir (newdir=oname)
342       copy (input="other/"//spec,output=oname//"/"//spec)
343  }
344  #CALC UTMIDDLE
345  hselect (images="other/"//spec,fields="TM_START", exp=yes) | scan (utstart)
346  hselect (images="other/"//spec,fields="EXPTIME", exp=yes) | scan (expt)
347  utmidhr=int((utstart + expt/2)/3600)
348  utmidmin=int((((utstart + expt/2)/3600)-utmidhr)*60)
349  utmidsec=int(((((utstart + expt/2)/3600)-utmidhr)*60 - utmidmin)*60)
350  utmid = (utmidhr//":"//utmidmin//":"//utmidsec)
351  printf ("%d:%d:%d\n",utmidhr,utmidmin,utmidsec) | scan (utmid)
```

imhead FAZTe202109060016.fit l+

```
HJD     =     2459464.25767855
VHELIO  =     -12.0510150041803
VLSR    =     7.57409507948893
VSUN    = '   20.    18.    30.  1900.'
ecl>
```

```
(crays  =                    no) Remove cosmic rays?
(ifcrays=                    no) Use object with cosmic rays extraction?
(objecta=                    no) Extract object apertures?
(flatcor=                    no) Apply flat correction to object?
(helioco=                   yes) calculate JD + heliocentric correction?
(idref  =                    no) refer database identification to images?
(norm   =                    no) normalize spectra?
(ncombin=                    no) combine normalized spcectra?
More
18:9:0ZTe202109060016.fit
#             Image            jd          hjd       ljd
# SETJD: Observatory parameters for Ondrejov observatory
#       timezone = -1
Warning: Image header parameter not found (UTMIDDLE)
# RVCORRECT: Observatory parameters for Ondrejov observatory
#       latitude = 49:54:38
#       longitude = 345:12:59
#       altitude = 528
##YR MO DY    UT       RA       DEC       VOBS
##   HJD          VOBS    VHELIO     VLSR    VDIURNAL    VLUNAR   VANNUAL    VSOLAR
2021  9  6 18:09:00 18:36:56  38:47:01      0.0
2459464.25768     0.00    -12.05      7.57      0.026    -0.005   -12.072    19.625
FAZTe202109060016.fit
ecl>
```

19) Ref spectrum

iftrimc = yes
iftrimo = yes
ifcrays = no
helioco = no
idref = yes
:go

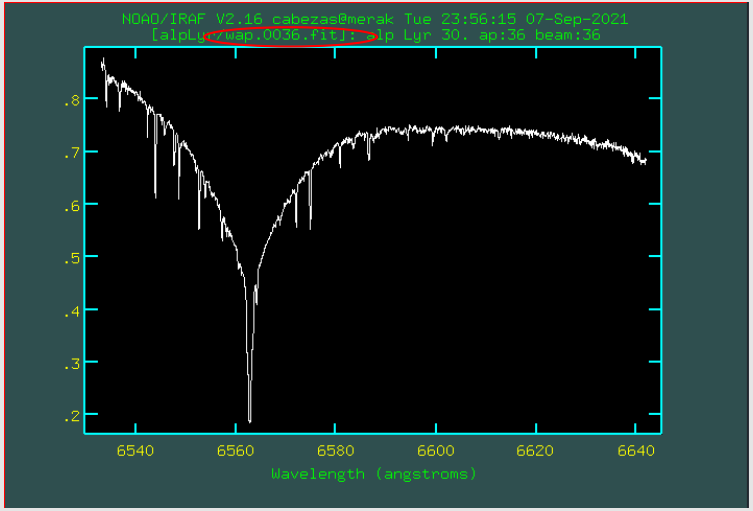
Read database of wavelength calibration and apply!

New files:

ap.00XX.fit – 1D spectra in pixels

wap.00XX.fit – 1D spectra in Angstrom

```
(ncombin=                    no) combine normalized spcectra?
More
[ap.0001] refspec1='iazcomp.0001'
ap.0001.fit: REFSPEC1 = 'iazcomp.0001 1.'
wap.0001.fit: ap = 1, w1 = 3964.476, w2 = 4030.398, dw = 0.032426, nw = 2034
[ap.0002] refspec1='iazcomp.0002'
ap.0002.fit: REFSPEC1 = 'iazcomp.0002 1.'
wap.0002.fit: ap = 2, w1 = 4009.502, w2 = 4076.196, dw = 0.032806, nw = 2034
[ap.0003] refspec1='iazcomp.0003'
ap.0003.fit: REFSPEC1 = 'iazcomp.0003 1.'
wap.0003.fit: ap = 3, w1 = 4055.563, w2 = 4123.023, dw = 0.033183, nw = 2034
[ap.0004] refspec1='iazcomp.0004'
ap.0004.fit: REFSPEC1 = 'iazcomp.0004 1.'
wap.0004.fit: ap = 4, w1 = 4102.691, w2 = 4170.963, dw = 0.033582, nw = 2034
[ap.0005] refspec1='iazcomp.0005'
ap.0005.fit: REFSPEC1 = 'iazcomp.0005 1.'
wap.0005.fit: ap = 5, w1 = 4150.934, w2 = 4220.016, dw =  0.03398, nw = 2034
[ap.0006] refspec1='iazcomp.0006'
ap.0006.fit: REFSPEC1 = 'iazcomp.0006 1.'
wap.0006.fit: ap = 6, w1 =  4200.32, w2 = 4270.232, dw = 0.034389, nw = 2034
[ap.0007] refspec1='iazcomp.0007'
ap.0007.fit: REFSPEC1 = 'iazcomp.0007 1.'
wap.0007.fit: ap = 7, w1 = 4250.877, w2 = 4321.653, dw = 0.034813, nw = 2034
[ap.0008] refspec1='iazcomp.0008'
```

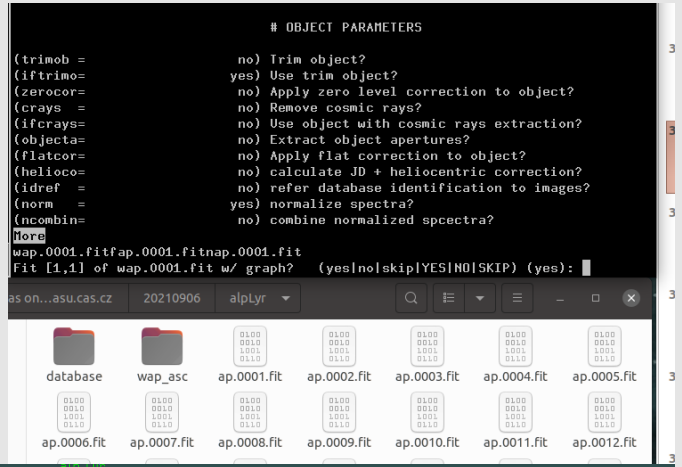


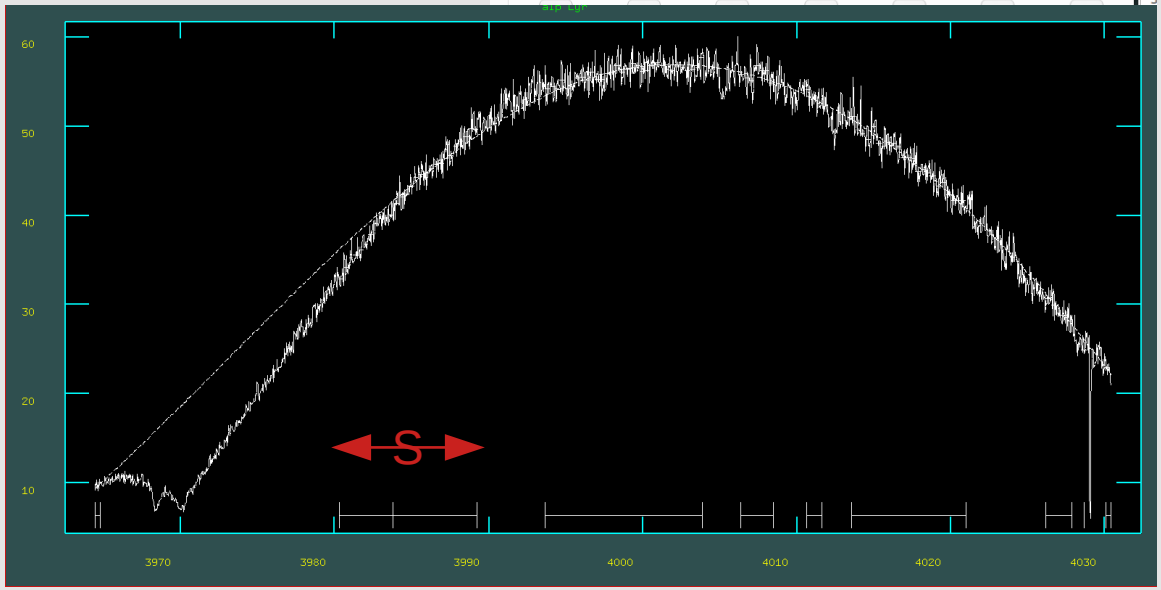
20) Normalization

iftrimc = yes
iftrimo = yes
ifcrays = no
idref = no
norm = yes
:go

Fit [1,1] of wap.0001.fit w/ graph? (yes|no|skip|YES|NO|SKIP) (yes):

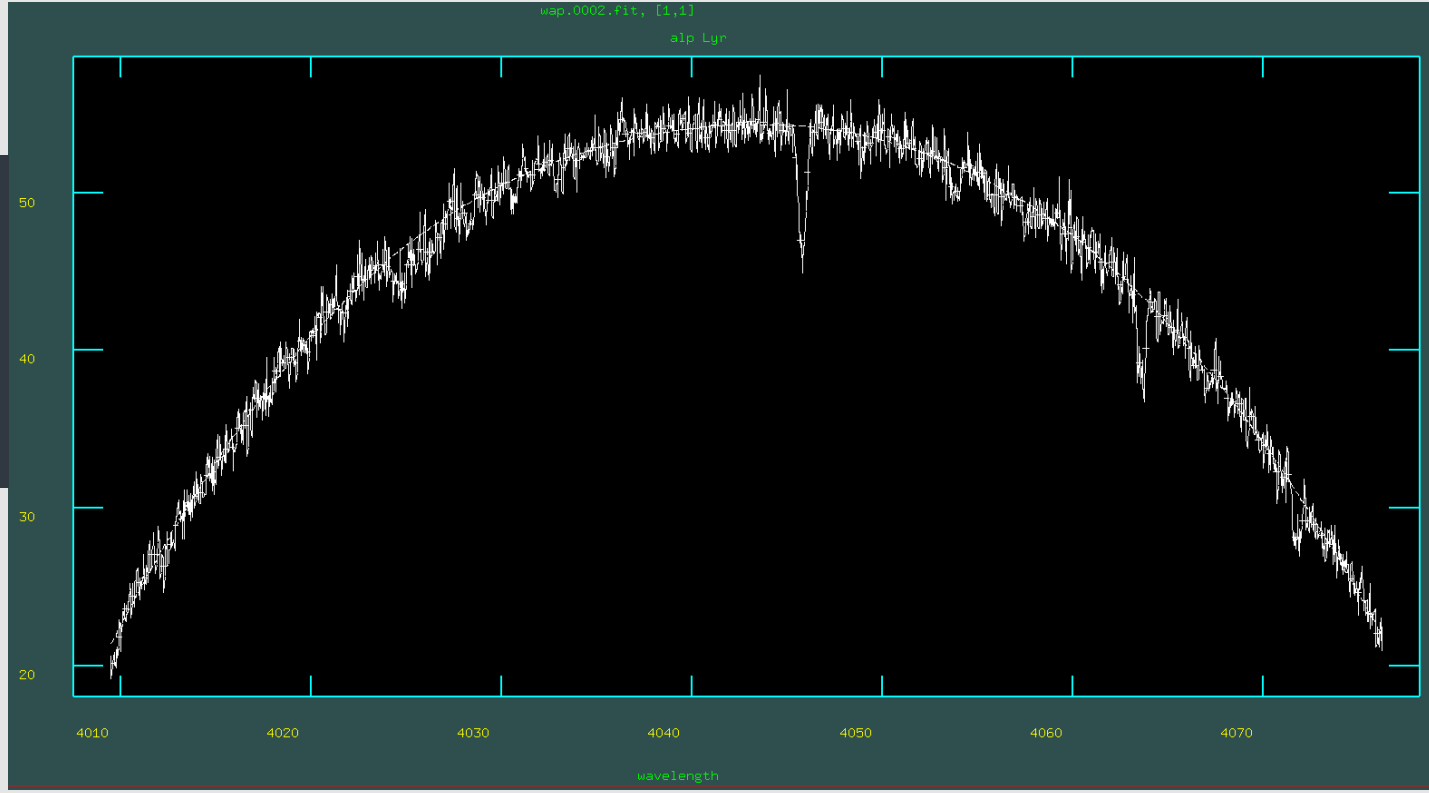
- change order **:o N**
- residuals **j**
- high rejection **:hi N**
- low rejection **:lo N**
- function **:f legendre/spline3/chebyshev**
- select region **s..s**
- delete region **z**
- delete all regions **t**
- zoom/window **w, e..e**
  - resize **w,a**
  - move right **w,r**
  - move left **w,l**
  - move up **w,u**
  - move down **w,d**





20) Normalization

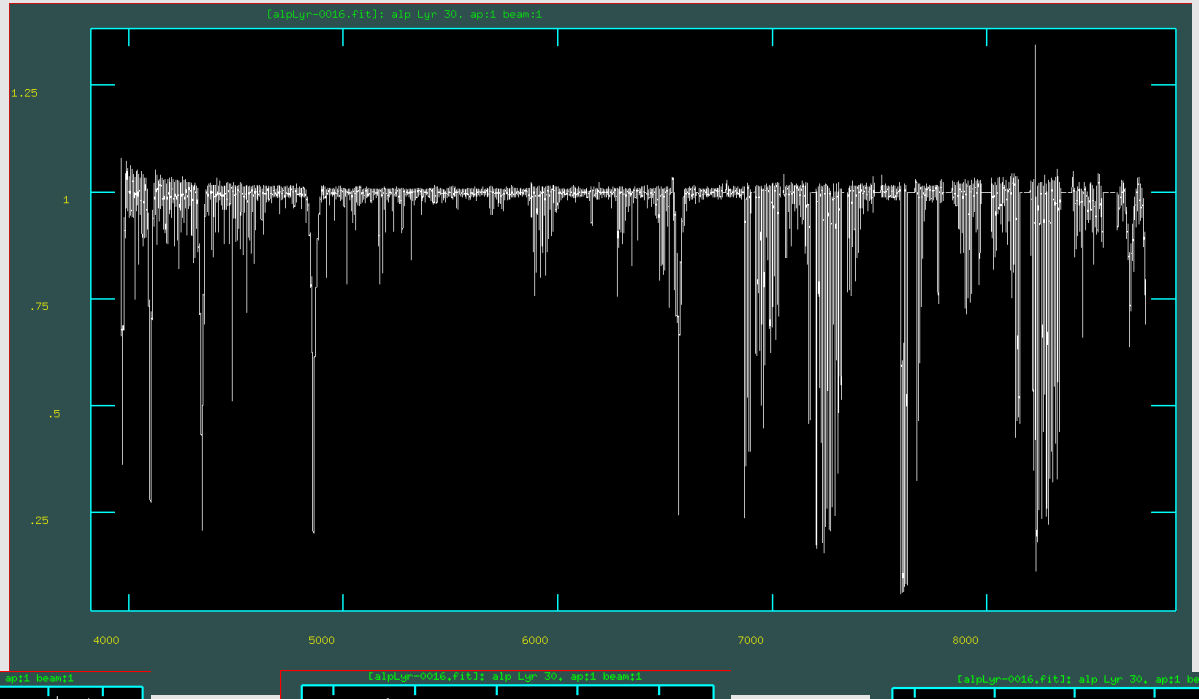
```
836  ##### normalization
837  if (norm==yes){
838       cd (oname)
839       unlearn continuum
840       unlearn scombine
841       continuum.type="fit"
842       continuum.function="spline3"
843       continuum.order=5
844       continuum.naverage=10
845       continuum.markrej=no
846       continuum.niterat=2000
847       continuum.high_re=2
848       continuum.grow=0
849       for (i=1; i <=nap; i+=1) {
850            printf ("wap.00%02d.fit\n",(i)) | scan(wap)
851            printf ("fap.00%02d.fit\n",(i)) | scan(fap)
852            printf ("nap.00%02d.fit\n",(i)) | scan(nnap)
853            print (wap, fap, nnap)
854       continuum (input=wap, output=fap)
855       }
856       cd "../"
```

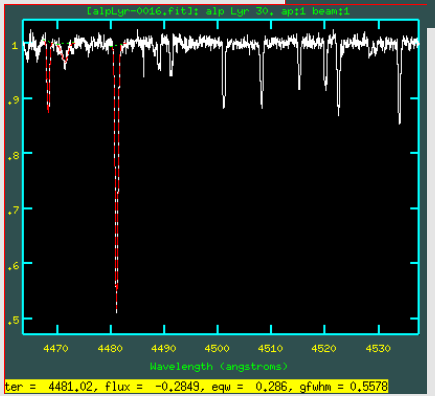


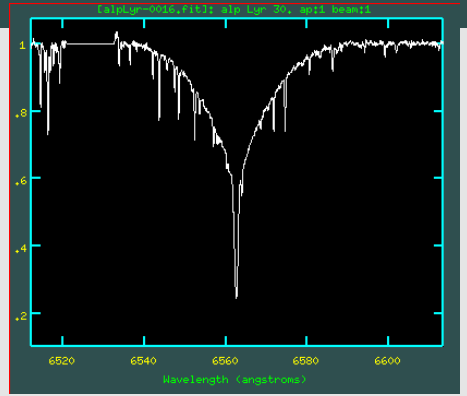
21) merging

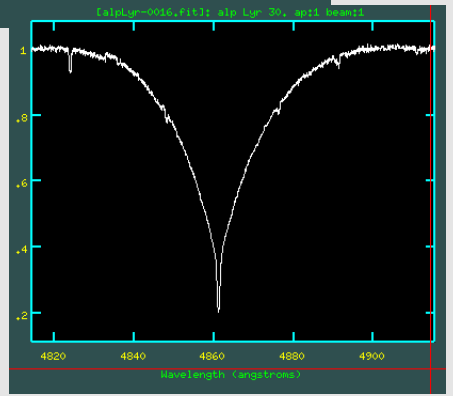
iftrimc = yes
iftrimo = yes
ifcrays = no
norm = no
ncombine = yes
:go

new file: **alpLyr-0016.fit**









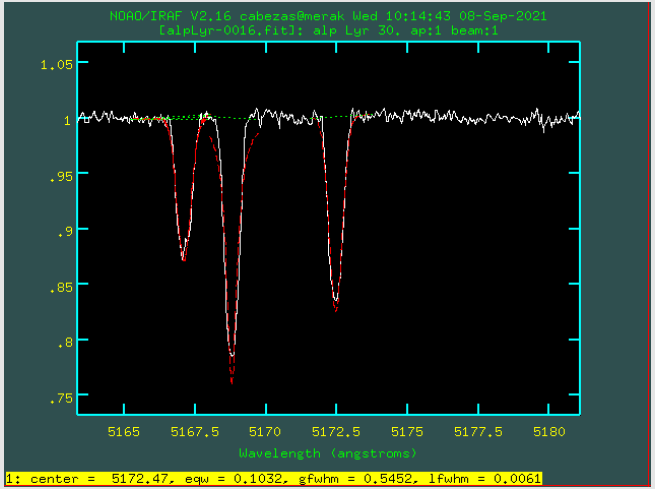
SPLOT

splot alpLyr-0016.fit

Fit: gaussian: **k..k(**or **g)**
lorentzian: **k..l**
voigt: **k..v**
centroid **e..e**

snr: **m..m**

Change unit (angstrom to km/s)
:u km/s 6562.8 an
:u an



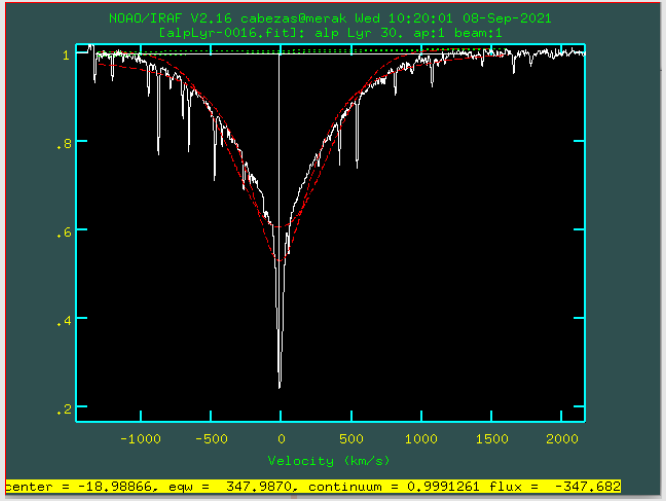
SPECPLOT

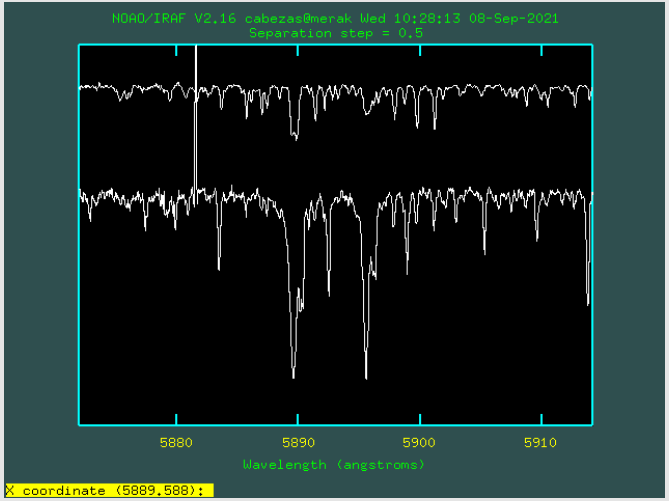
specplot spec1.fit,spec2.fits

Change step: **step 1** (or any number)
See wavelength: **u**
snr: **m..m**

Change unit (angstrom to km/s)
:u km/s 6562.8 an
:u an

Replot: **r**





General Remarks

Acknowledgments

-Astronomical Institute of the Czech Academy of Sciences (ASU)
Stellar Physics Department
Galaxies and Planetary Systems Department
-Dr. Brankica Kubátová
-Dr. Mirek Šlechta
-Dr. Petr Škoda
-Dr. Marek Skarka

Thank you!!

- Don't get frustrated if you don't understand it the first time, or the second (or n-time) ... there will always be other options.
- The optimal reduction process always will be different for each instrument.
- IRAF sometimes is a bit tricky, but really useful.
- Quick check or inspection of spectra.
- Versatil program because many parameters (sometimes too much).
- Pre-defined task.
- “opensource” you can write your own task/package.
- Xgterm – nice interactive tool.
- Slow with computation, python/idl/fortran would be good option.