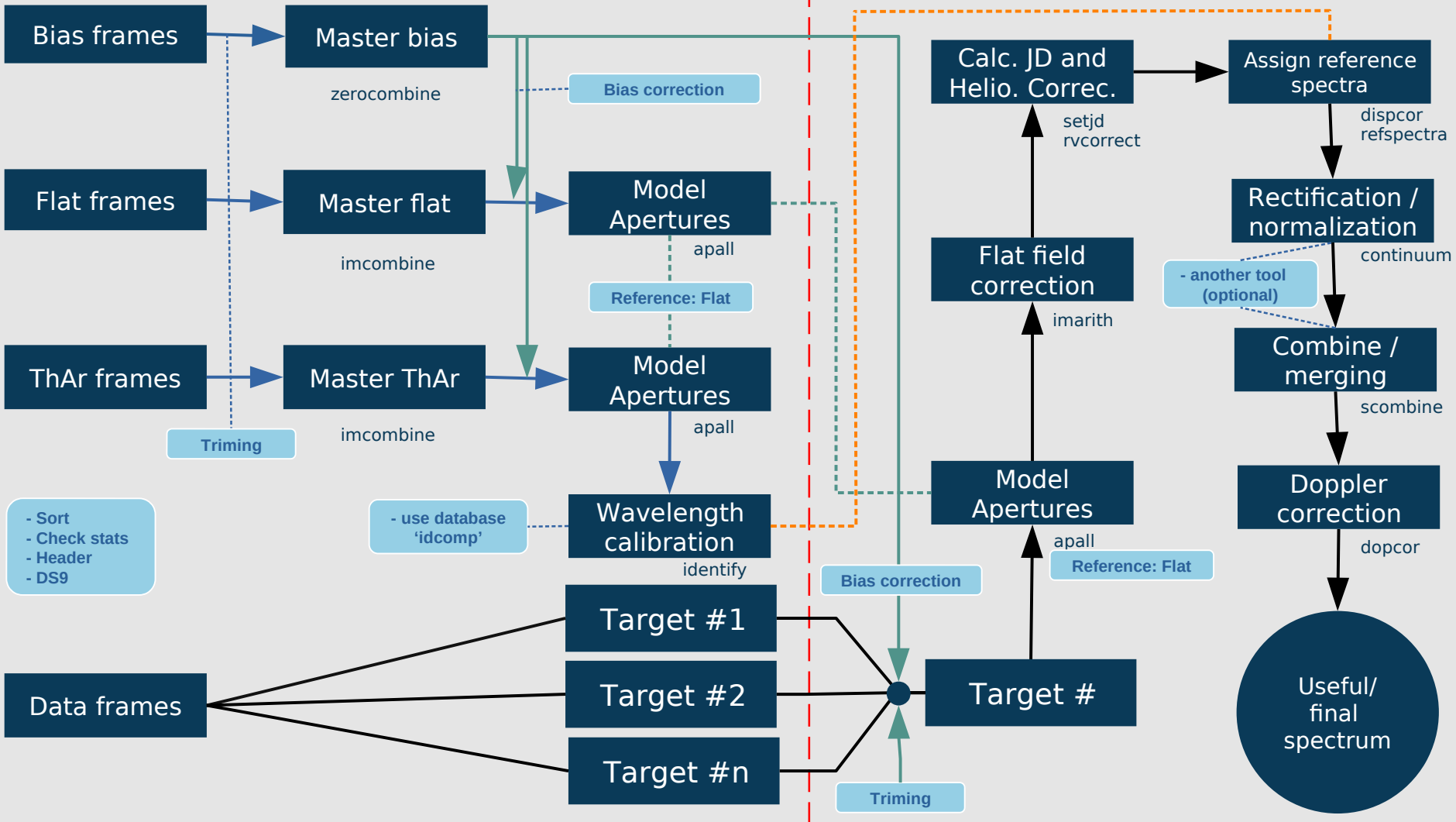
Data reduction of the Echelle spectra using IRAF

Mauricio Cabezas
23.08.2023
Workshop on observational techniques
21 August – 1 September 2023 at Ondřejov observatory


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
Bias correction
Target #1
Target #2
Target #n
Data frames
Target #
Triming
Calc. JD and Helio. Correc.
setjd
rvcorrect
Assign reference spectra
dispcor
refspectra
Rectification / normalization
continuum
- another tool (optional)
Flat field correction
imarith
Combine / merging
scombine
Model Apertures
apall
Reference: Flat
Doppler correction
dopcor
Useful/ final spectrum

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

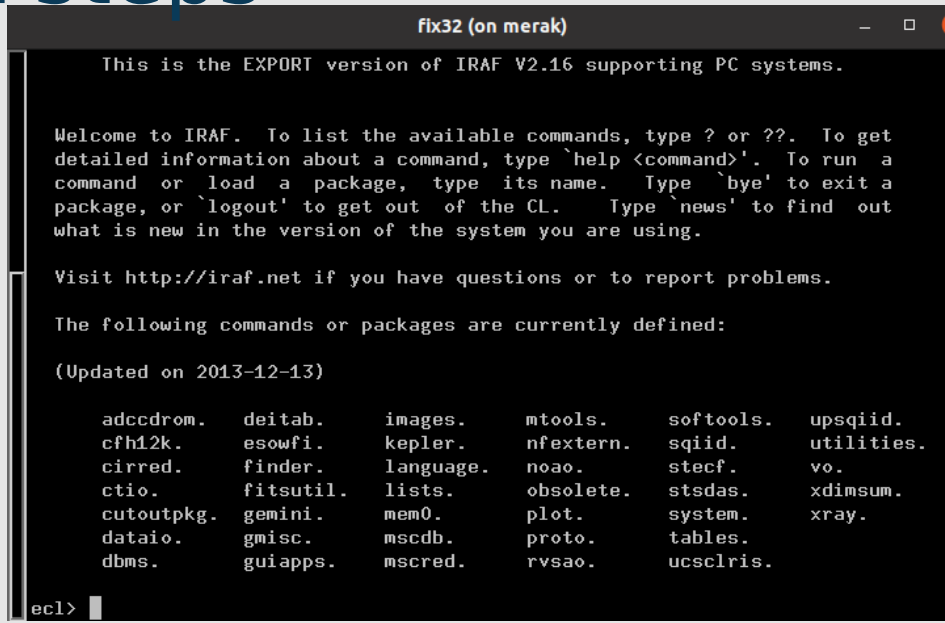
Check stats of spectrum

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) include data of Ondrejov observatory in the database, edit file **obsdb.dat** (path: ~miniconda3/envs/iraf38/iraf/noao/lib/obsdb.dat)

observatory = "ondrejov"

name = "Ondrejov observatory"

longitude = 345:12:59

latitude = 49:54:38

altitude = 528

timezone = -1

OESRED.CL

- SEMI automatic
- always check your data!

  imstat
  ds9
  splot

- Divided in two parts: Calibration and Science.

  Useful for many exposures same night

- Parameters were tested and works exclusively for OES.
- for now, recommend stepbystep

- WORKSHOP/
- **oesred.cl**
- **20230629**
  - **e2023*.fit**
- **20232308**
  - **e2023*.fit**

e 2023 06 29 0034.fit

year **month** **day** **Id-number**

lmhead *.fit *(quick inspection)*

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

Quit ***:q***

Go! ***:g***

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

input   =    e202306290034.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     =                 0022) Observation id number

                                # CALIBRATION PARAMETERS

(orgfile=                  yes) 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_2307) folder name with identification database
(idencom=                   no) Identify features in spectrum for dispersion solution?

                                # 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?

                                # TASK PARAMETERS

(nfuncti=             legendre) Continuum fitting function
(norder =                    5) Order of continuum fitting function
(t_funct=              spline3) Trace apertures fitting fucntion
(t_order=                    5) Order of apertures fitting function
(edit_o =                   no) Edit object apertures?
(review_=                   no) Review object apertures?
(mode   =                   ql)
```

1) Initial parameters!

Check type of “image”:

flat: flat fields

zero: bias

comp: ThAr comparison spectra

Object: science

imhead e*.fit(short way)

- print list filename and object type

hselect e*.fit $I,object yes

- check header!

imhead e202109060001.fit l+ | page

- for example, alp Lyr spectrum.

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

```
cl
DISPAXIS=                    1 / Dispersion axis along lines
GRATNAME= '3       '           / Grating name - ID
SLITTYPE= 'BLADE   '           / Type of slit - blade or image slicers
AUTOGUID= 'NO      '           / Status of autoguider system
SLITWID =                  0.6 / Slit width in mm
COLIMAT = 'open    '           / Collimator mask status
TLE-TRCS= '0       '           / Correction Set
TLE-TRGV= '-22.3 -15.3'        / Guiding Value
TLE-TRHD= '-46.9083 38.8601'   / Hour and Declination Axis
TLE-TRRD= '183656.340 +38470'  / Right ascension and Declination
TLE-TRUS= '0.0000 0.0000'      / User Speed
SGH-MCO = 'coude   '           / Mirror Coude Oes
SGH-MSC = 'star    '           / Mirror Star Calibration
SGH-OIC =                    2 / OES Iodine cell
TM-DIFF =                   -2 / T1688068811 - P1688068813
OBJECT  = 'alp Lyr '           / Title of observation
IMAGETYP= 'object  '           / Type of observation, eg. FLAT
OBSERVER= 'Kubatova, Novotny'  / Observers
SYSVER  = 'PESO exported.exp'
READSPD = '100kHz  '
FILENAME= 'e202306290034.fit'
CAMFOCUS=                 3080. / Camera focus position
SPECTEMP=                  22.9 / 18288
SPECFILT=                     0 / Spectral filter
SLITHEIG=                  1.07 / Slit hight in mm
```

ALWAYS keep raw data as backup!!

```
cl
e202306290029.fit[2048,2048][ushort]: comp
e202306290030.fit[2048,2048][ushort]: comp
e202306290031.fit[2048,2048][ushort]: zero
e202306290032.fit[2048,2048][ushort]: zero
e202306290033.fit[2048,2048][ushort]: zero
e202306290034.fit[2048,2048][ushort]: alp Lyr
e202306290035.fit[2048,2048][ushort]: Cyg X-1
e202306290036.fit[2048,2048][ushort]: HD 340883
e202306290037.fit[2048,2048][ushort]: HD 339368
e202306290038.fit[2048,2048][ushort]: comp
e202306290039.fit[2048,2048][ushort]: comp
e202306290040.fit[2048,2048][ushort]: comp
```

```
64  ####### define variables
65  name="Mauricio Cabezas "
66  email='mauricio.cabezas@asu.cas.cz '
```

Good to know and have contact who performed the reduction

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

input   =   e202306290034.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     =                  0022) Observation id number
```

input: The complete fit file name of our target, science

output: Filename of the final reduced spectrum > “output-id.fit”, why ‘id’?
→ same target, same night

Idtarget: EXACT name of the target which is in the header!!

naperture: Number of apertures, important if we wanna use the wavelenght calibration database. Keep in 49.

id: recommended the last four numbers of the fit file

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:

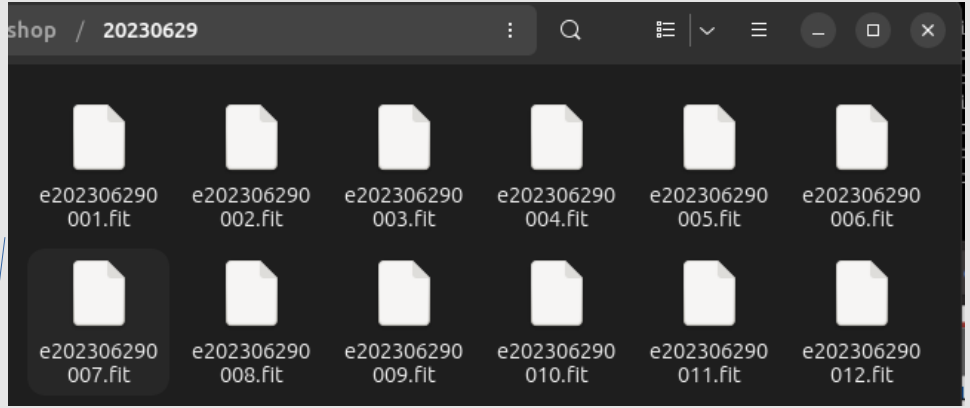
imstat @flat.dat

#	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.

- visual inspection:

!ds9 &

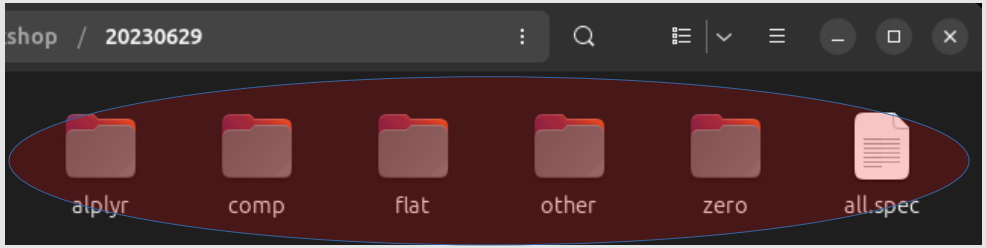
display e202306290011.fit 1




```
input    =   e202306290034.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      =                 0022) Observation id number

                                 # CALIBRATION PARAMETERS

(orgfile=                   yes) do you want organize files?
(zerocom=                    no) Combine zero level images?
(trimcal=                    no) Trim flat and comp?
```

Before each run be sure we are in the MAIN folder (20230629/)!

pwd



3) Combine zeros!

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

  orgfile = no

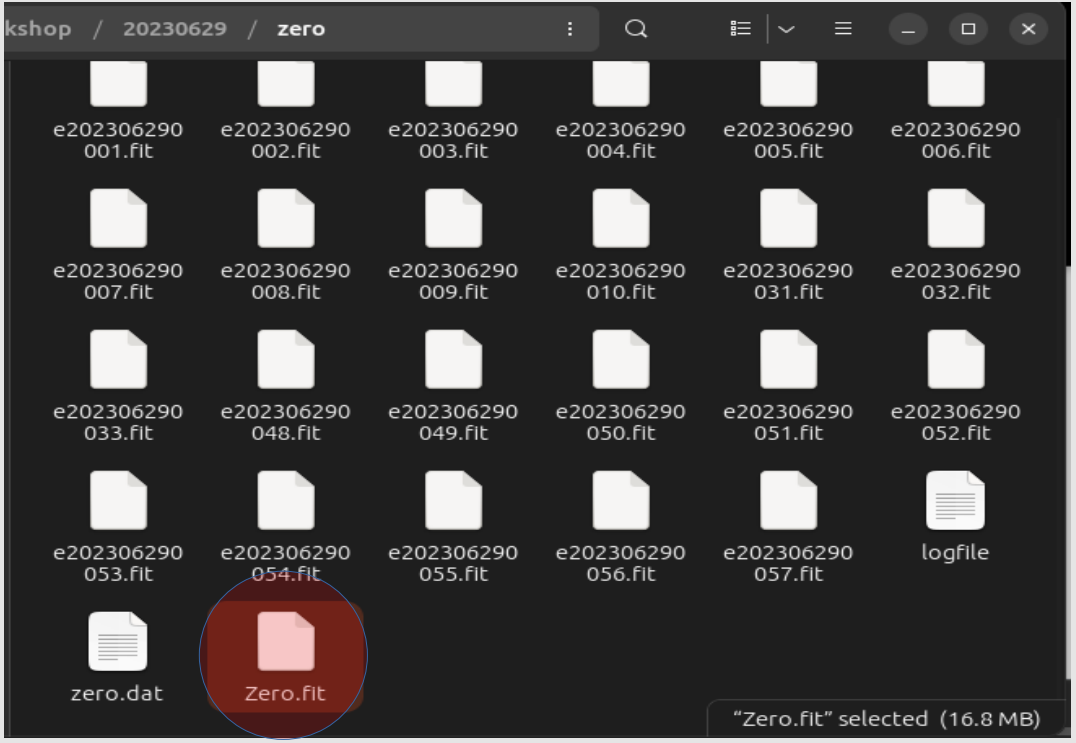
  zerocom = yes

  :go

- new file: **Zero.fit**

```
# CALIBRATION PARAMETERS

(orgfile=                    no) do you want organize files?
(zerocom=                   yes) Combine zero level images?
(trimcal=                    no) Trim flat and comp?
(iftrimc=                    no) Use trim flat & comp?
(zerocor=                    no) Apply zero level correction to flat & c
(compcom=                    no) Combine comparison lamp images?
```

```
112  ###### ZEROCOMBINE
113  if (access("zero/Zero.fit")){
114       zerocomb=no
115  }
116  if (zerocomb==yes){
117       cd "zero/"
118       unlearn zerocombine
119       zerocombine.reject="minmax"
120       zerocombine.rdnoise= "READNOIS"
121       zerocombine.gain    = "GAIN"
122       zerocombine (input="@zero.dat",output="Zero.fit")
123       cd "../"
124  }
```



4) trim flat and comp (recommended)

- completely optional, but usually the first and last pixel of each aperture is saturated and can be annoying during normalization.
- related with lines database ‘idcomp’

zerocom = no

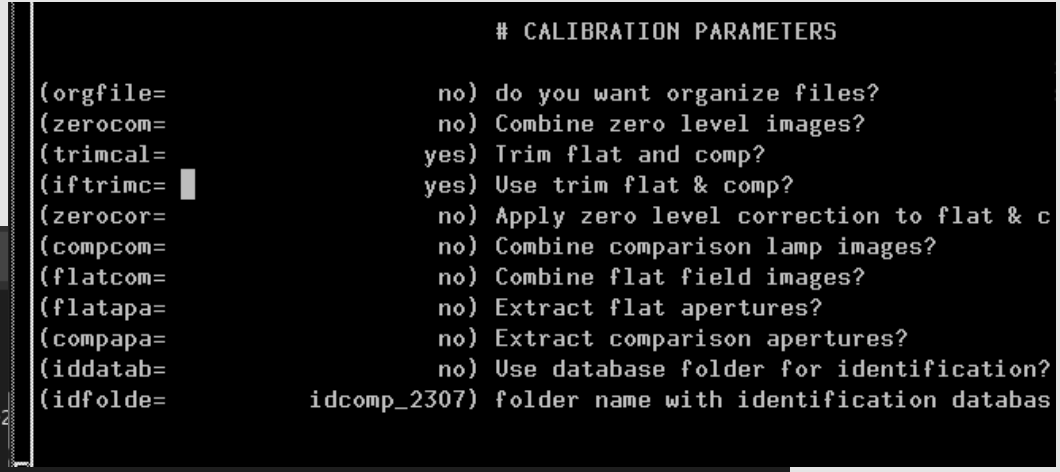
trimcal = yes

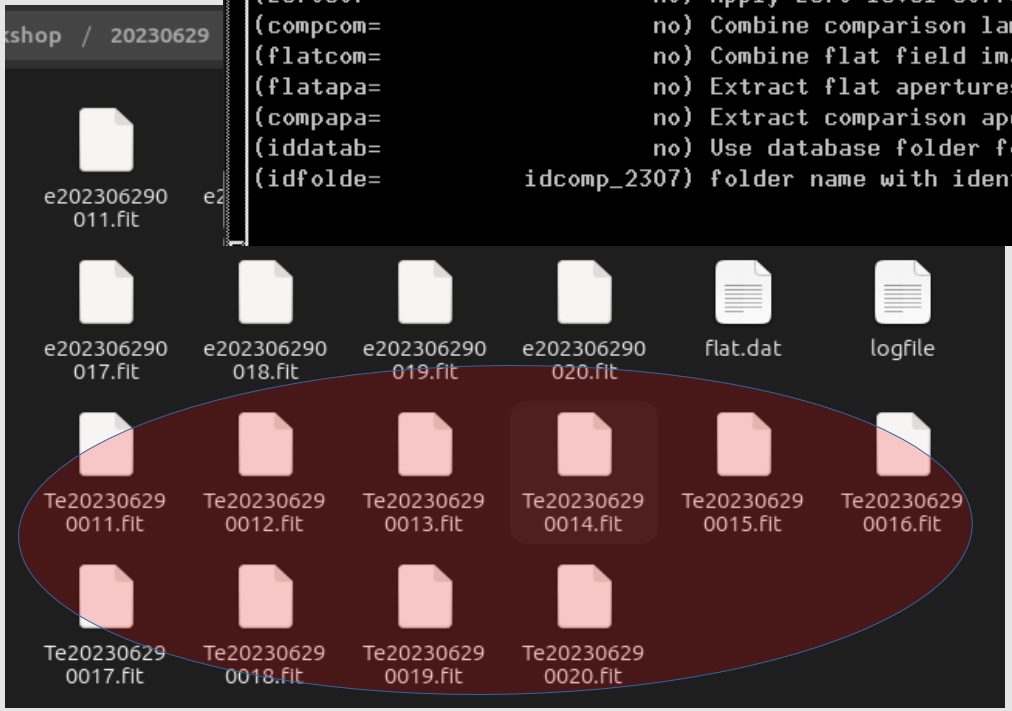
iftrimc = yes

:go

- new files, prefix **T**

Te*.fit





4) trim flat and comp (recommended)

If we apply trim and you want to use the trimmed images, you need to set iftrimc=yes during the whole reduction!

```
if (trimcal==yes){
     unlearn ccdproc
     ccdproc.trimsec = "[2:2035,*]"
#    ccdproc.trimsec = "[5:2025,800:1500]"
     ccdproc.trim = yes
     ccdproc.fixpix = no
     ccdproc.overscan = no
     ccdproc.darkcor= no
     ccdproc.zerocor=no
     ccdproc.flatcor=no
     #
     cd "flat/"
     ccdproc.ccdtype = "flat"
     ccdproc (images="@flat.dat",output="T@flat.dat")
     cd "../"
     #
     cd "comp/"
     ccdproc.ccdtype = "comp"
     ccdproc (images="@comp.dat",output="T@comp.dat")
     cd "../"
```

5) Bias correction → flat and comp

trimcal = no
iftrimc = yes
zerocor = yes
:go

- new files, prefix **Z**
  ZTe*.fit

```
                                    # 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>
```

me / workshop / 20230629 / flat

Te20230629 0017.fit
Te20230629 0018.fit
Te20230629 0019.fit
Te20230629 0020.fit
ZTe202306290011.fit
ZTe202306290012.fit
ZTe202306290013.fit
ZTe202306290014.fit
ZTe202306290015.fit
ZTe202306290016.fit
ZTe202306290017.fit
ZTe202306290018.fit
ZTe202306290019.fit
ZTe202306290020.fit

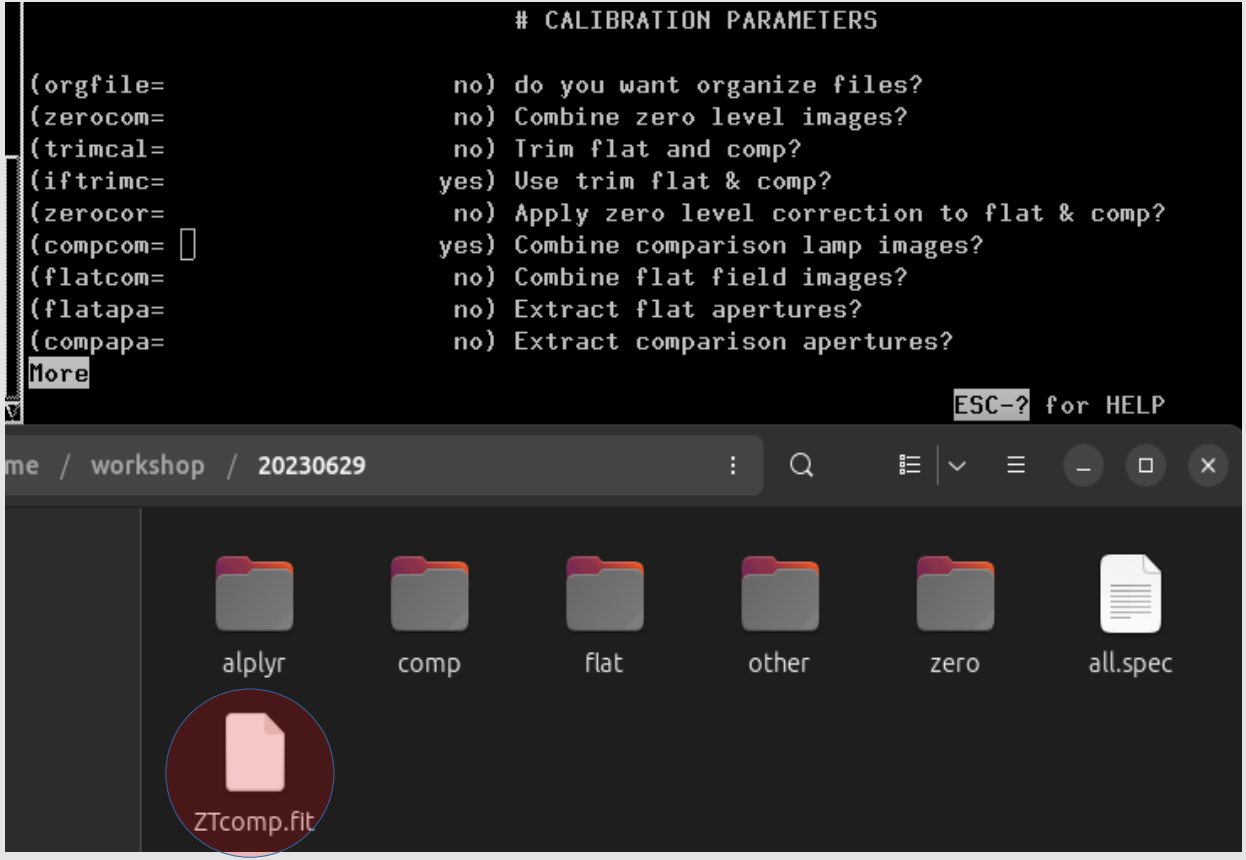
5) Bias correction flat and comp

```
###### SUBTRACT zero
if (zerocorcal==yes){
#    imarith (operand1="flat/flat.fits", op="/", operand2="zero/Zero.fits", result="Zflat.fits")
#    imarith (operand1="comp/comp.fits", op="/", operand2="zero/Zero.fits", result="Zcomp.fits")
     unlearn ccdproc
     ccdproc.ccdtype="zero"
#    ccdproc.trimsec = "[5:2025,800:1500]"
     ccdproc.fixpix = no
     ccdproc.overscan = no
     ccdproc.darkcor= no
     ccdproc.zerocor=no
     ccdproc.flatcor=no
     #
     cd "flat/"
     ccdproc.ccdtype="flat"
     ccdproc.zerocor=yes
     ccdproc.zero="../zero/Zero.fit"
          if (iftrimc==yes){
          ccdproc (images="T@flat.dat",output="ZT@flat.dat")
          } else {
          ccdproc (images="@flat.dat",output="Z@flat.dat")
          }
     cd "../"
     #
     cd "comp/"
     ccdproc.ccdtype="comp"
     ccdproc.zerocor=yes
     ccdproc.zero="../zero/Zero.fit"
               if (iftrimc==yes){
          ccdproc (images="T@comp.dat",output="ZT@comp.dat")
          } else {
          ccdproc (images="@comp.dat",output="Z@comp.dat")
          }
     cd "../"
}
```

6) Combine comp (ThAr)

iftrimc = yes (keep “yes”)
zerocor = no
compcom = yes
:go

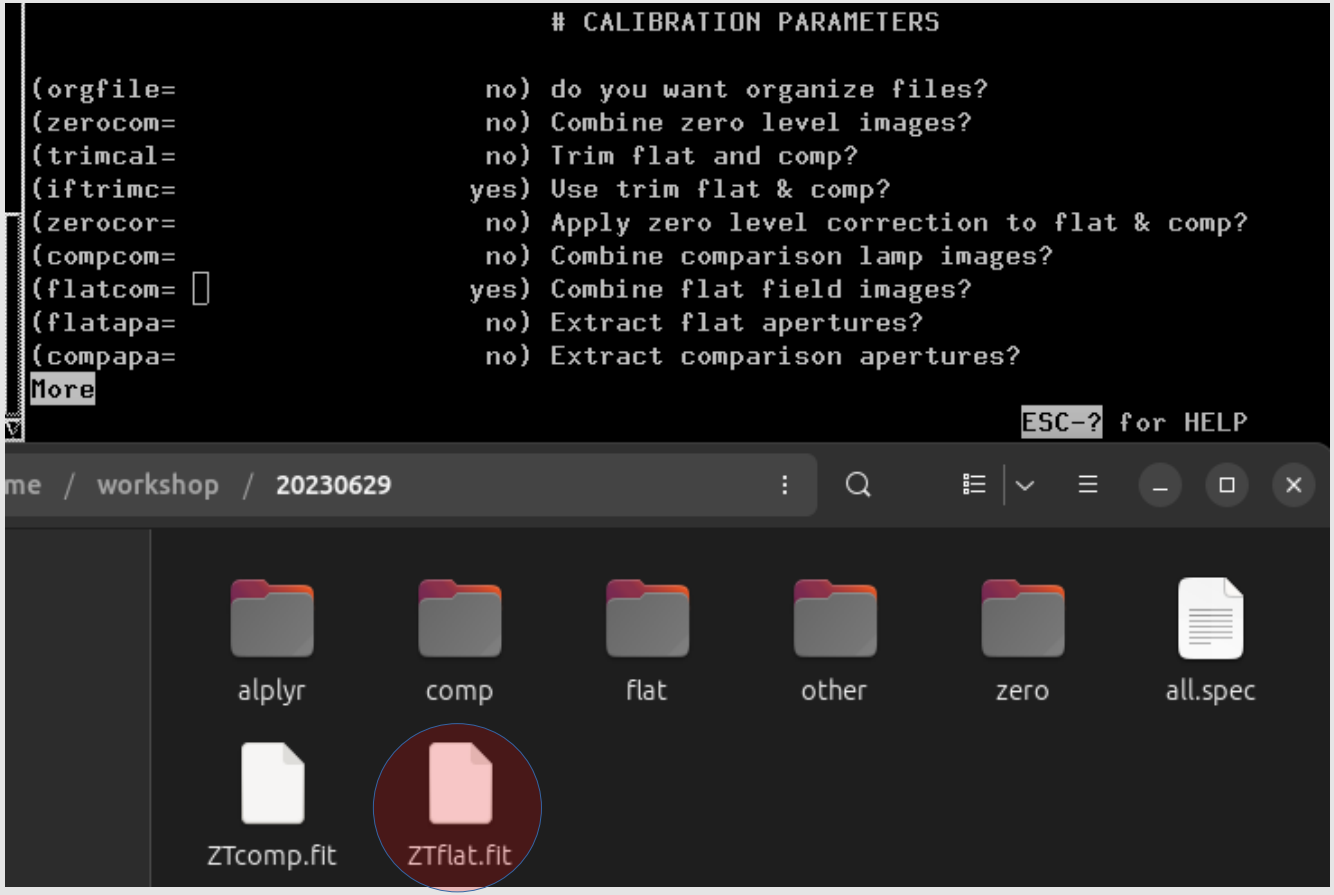
- new file
ZTcomp.fit



7) Combine flat field

iftrimc = yes
compcom = no
flatcom = yes
:go

- new file
  ZTflat.fit



6) and 7) Combine comp/flat field

```
###### COMBINE - comp/lamp
if (compcomb==yes){
     cd "comp/"
     unlearn imcombine
     # imcombine.reject = "none"
     imcombine.reject = "none"
     imcombine.lsigma = 3
     imcombine.hsigma = 3
     imcombine.rdnoise= "READNOIS"
     imcombine.gain   = "GAIN"
     imcombine.scale = "exposure"
     imcombine.expname="EXPTIME"
#    imcombine (input="@comp.dat",output = "comp.fits")
          if (iftrimc==yes){
          imcombine (input="ZT@comp.dat",output = "../ZTcomp.fit")
          } else {
          imcombine (input="Z@comp.dat",output = "../Zcomp.fit")
          }
     cd "../"
}

###### COMBINE flat
if (flatcomb==yes){
     cd "flat/"
          if (iftrimc==yes){
          imcombine (input="ZT@flat.dat",output = "../ZTflat.fit")
          } else {
          imcombine (input="Z@flat.dat",output = "../Zflat.fit")
          }
     cd "../"
}
```

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):

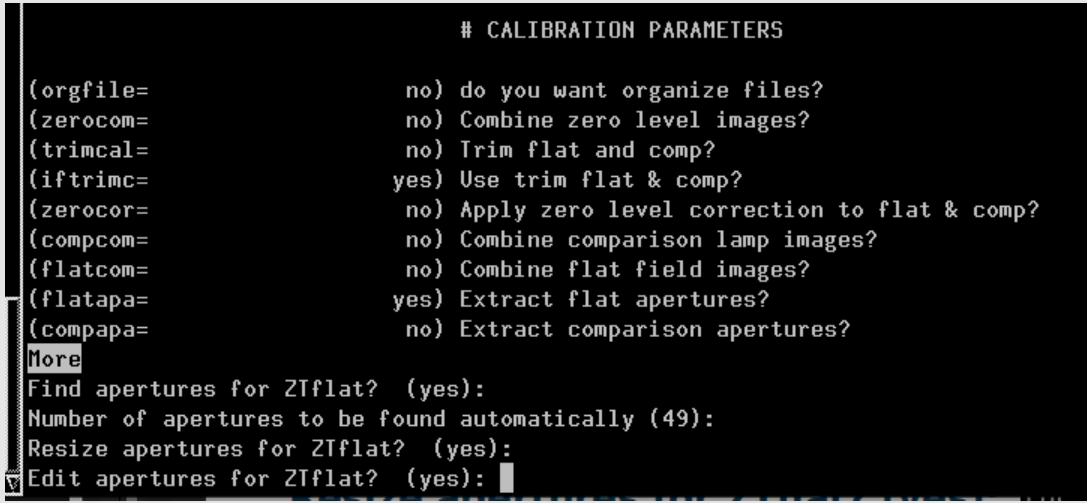
Edit apertures for Ztflat? (yes):

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

- value in parenthesis (yes/no) is Predefined. press ENTER

- 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.

Background fitting is not necessarily because overlapping.



CURSOR KEY - interactive

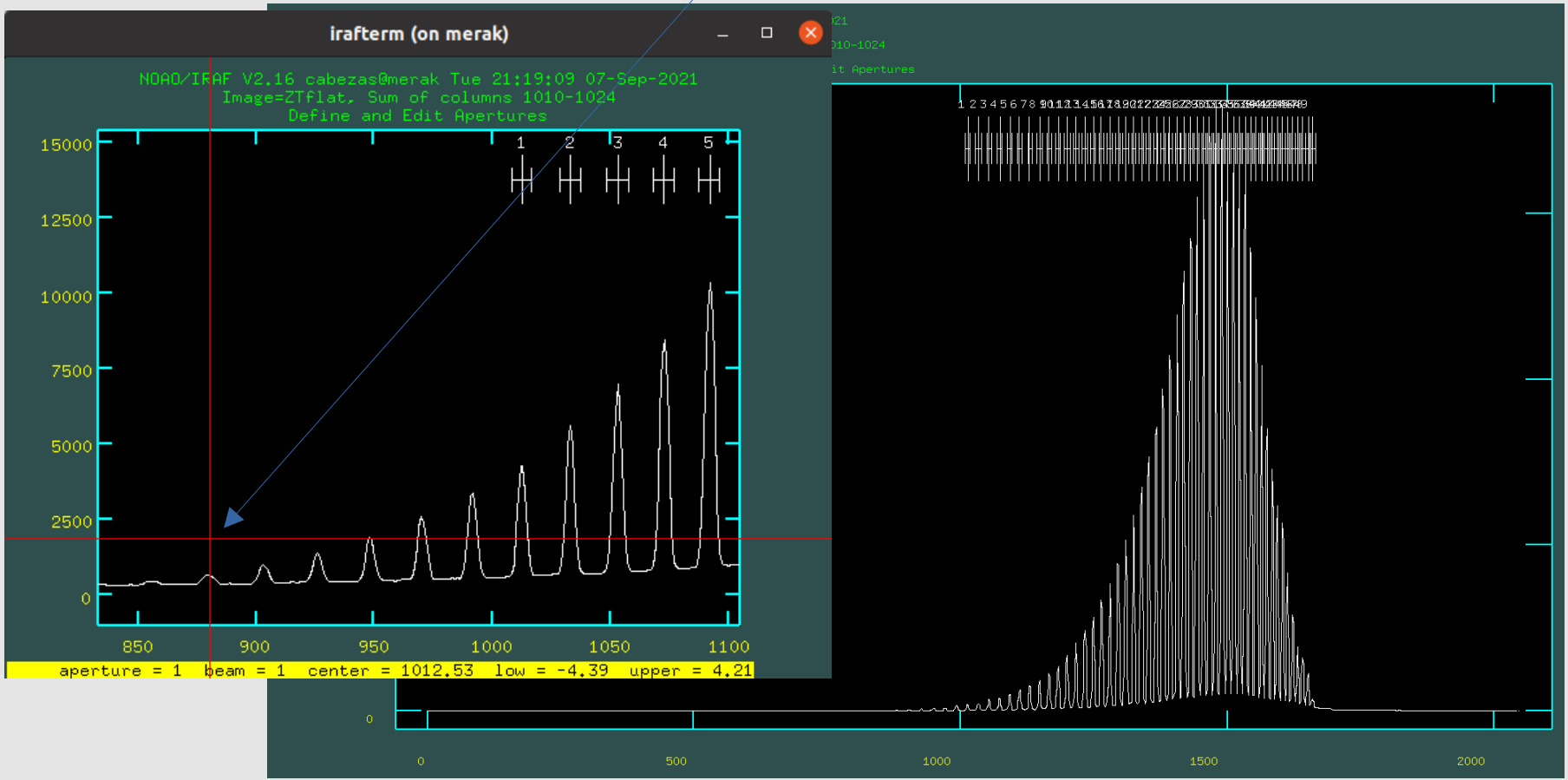
Good practice:
- **NEVER** resize the window directly when interactive is activated!
  if yes→ ':' resize → enter
- **NEVER** do click on the interactive window!
  if yes→ click on green square→ Delete

Help window: '?'
q+q → exit help
Window: w+e+e → Zoom bottom left/top right corner
w+a → redraw/ show all
w+u/d/l/r → move up/down/left/right

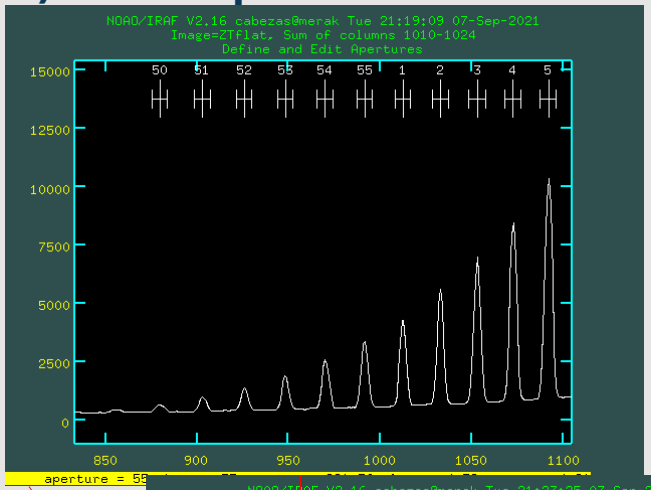
```
                    APEXTRACT CURSOR KEY SUMMARY

?  Print help              j  Set beam number         u  Set upper limit(s)
a  Toggle all flag         l  Set lower limit(s)      w  Window graph
b  Set background(s)       m  Mark aperture           y  Y level limit(s)
c  Center aperture(s)      n  New uncentered ap.      z  Resize aperture(s)
d  Delete aperture(s)      o  Order ap. numbers       I  Interrupt
e  Extract spectra         q  Quit                    +  Next aperture
f  Find apertures          r  Redraw graph            -  Previous aperture
g  Recenter aperture(s)    s  Shift aperture(s)       .  Nearest aperture
i  Set aperture ID         t  Trace aperture(s)

                    APEXTRACT COLON COMMAND SUMMARY

:apertures      :center         :npeaks         :show           :t_width
:apidtable      :clean          :nsubaps        :skybox         :threshold
:avglimits      :database       :nsum           :t_function     :title
:b_function     :extras         :order          :t_grow         :ulimit
:b_grow         :gain           :parameters     :t_high_reject  :upper
:b_high_reject  :image          :peak           :t_low_reject   :usigma
:b_low_reject   :line           :plotfile       :t_naverage     :weights
:b_naverage     :llimit         :r_grow         :t_niterate     :width
:b_niterate     :logfile        :radius         :t_nlost        :write
:b_order        :lower          :read           :t_nsum         :ylevel
:b_sample       :lsigma         :readnoise      :t_order
:background     :maxsep         :saturation     :t_sample
:bkg            :minsep         :shift          :t_step

                    APEXTRACT CURSOR KEYS

?    Print help
a    Toggle the ALL flag
b an Set background fitting parameters
c an Center aperture(s)
apextract options-(46%)-line 34-file 1 of 1
```

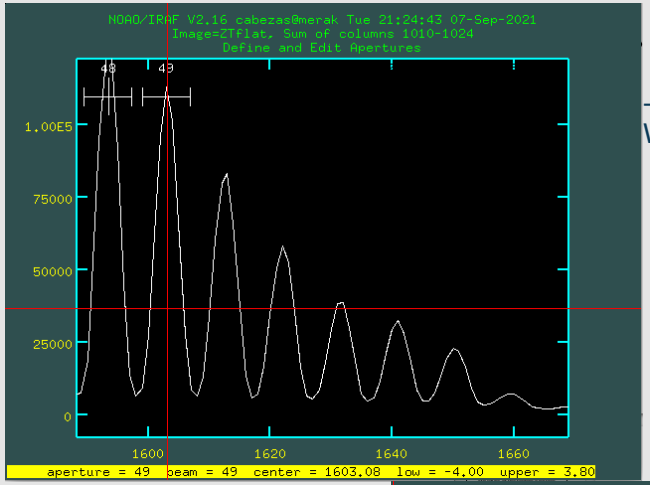
8) Model apertures - Flat

- first aperture near the pixel ~860-890
  mark: **m**

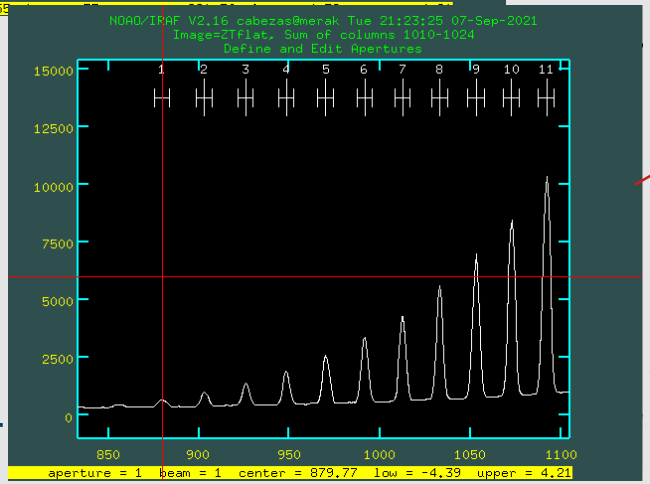


8) Model apertures - Flat





- delete extra apertures
We need only 49
→ over the aperture
Then : **d**

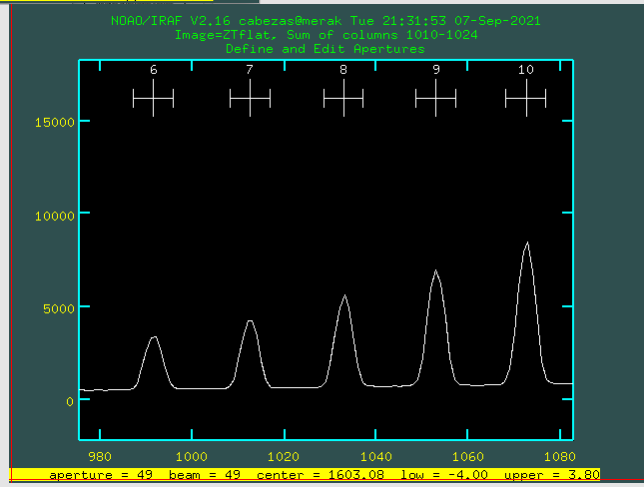


First aperture
Number 1,
no 50

Sort: **o**
'1'
enter

All right? → **'q'**
'Trace apertures for Ztflat?
(yes)

- check apertures,
Zoom in/out and
Aperture numbers



8) Model apertures - Flat

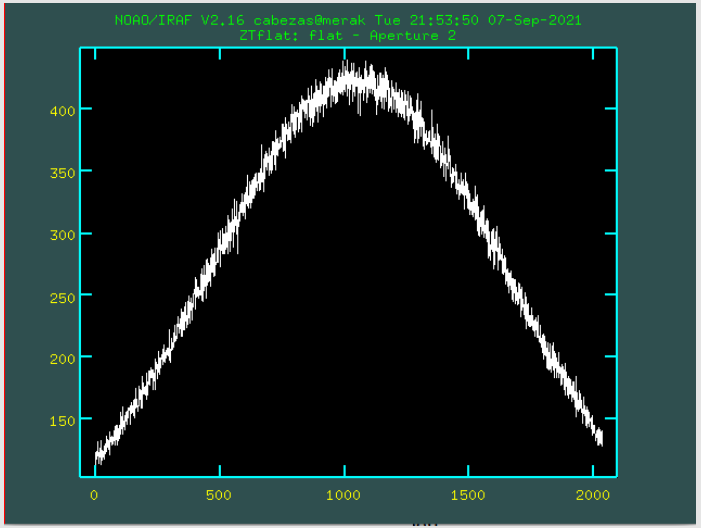
Fit curve to aperture N of Ztflat interactively (yes): YES

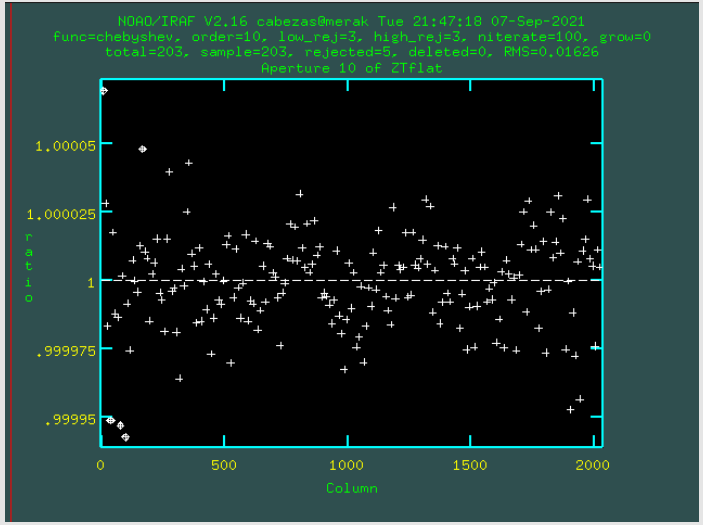
Write apertures for ZTflat to database (yes):

Extract aperture for ZTflat? (yes):

Review extracted spectra from ZTflat? (yes): YES

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

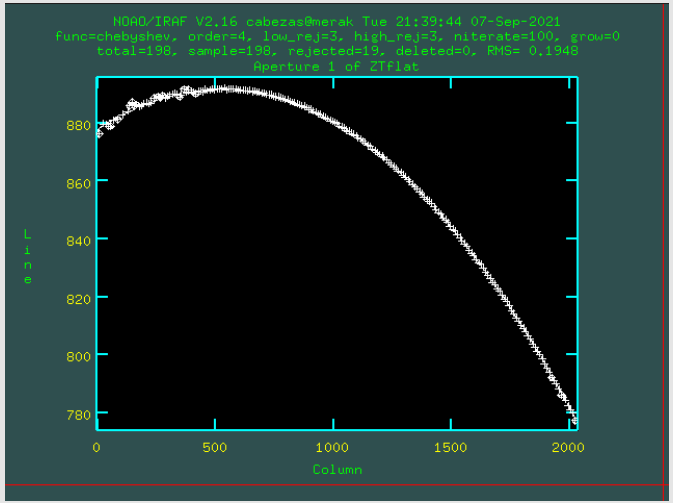




Polinomial fitting of the echelle orders

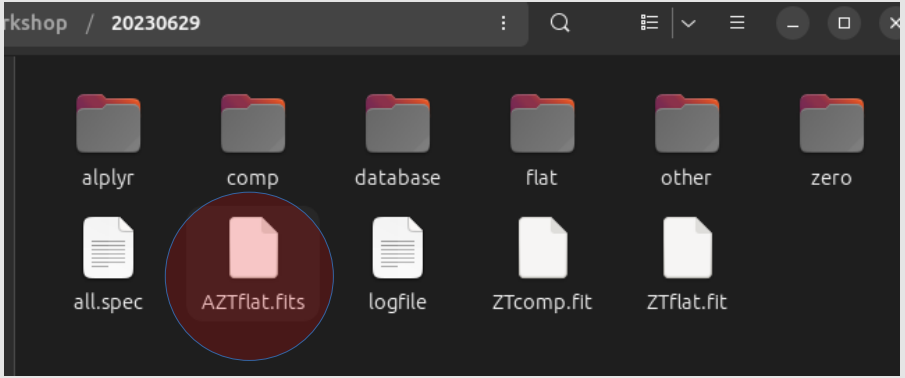
- change order **:o N**
- change function: **:f spline3/legendre/ chebyshev**

- relative residuals **k**
- residuals (px) **j**
- aperture **h**



8) Model apertures - Flat

- new file: **AZTflat.fit**




```
#####APERTURES - APALL FLAT
if (flatapall==yes){
     echelle
     unlearn apall
     apall.format = "echelle"

     apall.extras=no
     apall.extract=yes

     apall.nsum=15

     apall.lower=-5
     apall.upper=5
    apall.b_order=3
     apall.b_sample="-10:-6,6:10"

     apall.nfind=nap
     #apall.minsep=10
     apall.minsep=5
     apall.maxsep=1000


     apall.ylevel = 0.04
     apall.bkg=yes
     #apall.bkg=no

     apall.t_nsum = 10
     apall.t_function = tfunct
     apall.t_niter=100
     apall.t_order=torder

     apall.clean=no
     apall.readnoi= 0
     apall.gain   = 1



     #apall.width=9
     #apall.width=5
#    apall.weights = "none"
     apall (input=inflat, output = "A"//inflat)
}
```

9) Model apertures – Comparison lamp

- Template: AZTflat.fit

iftrimc = yes

flatapa = no

compapa = yes

:go

Edit apertures for ZTcomp? (yes):

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

q

Write apertures for ZTcomp to database (yes):

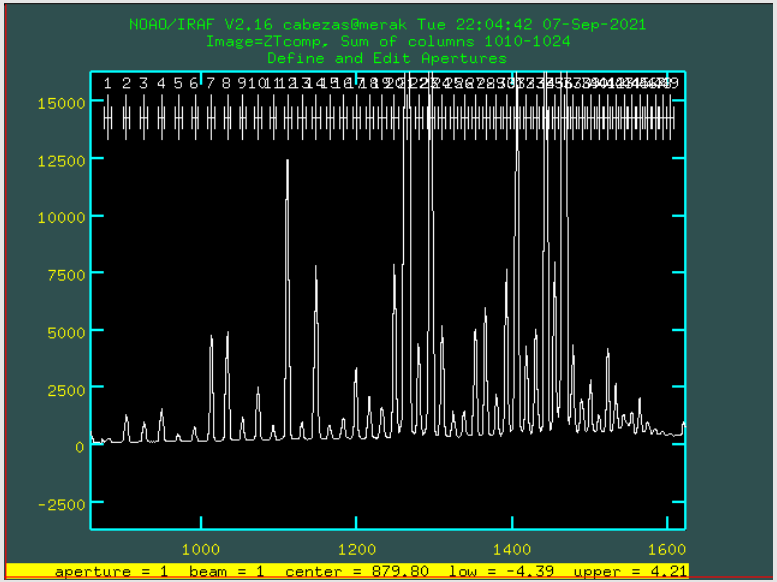
Extract aperture spectra for ZTcomp? (yes):

Review extracted spectra from ZTcomp? (yes):

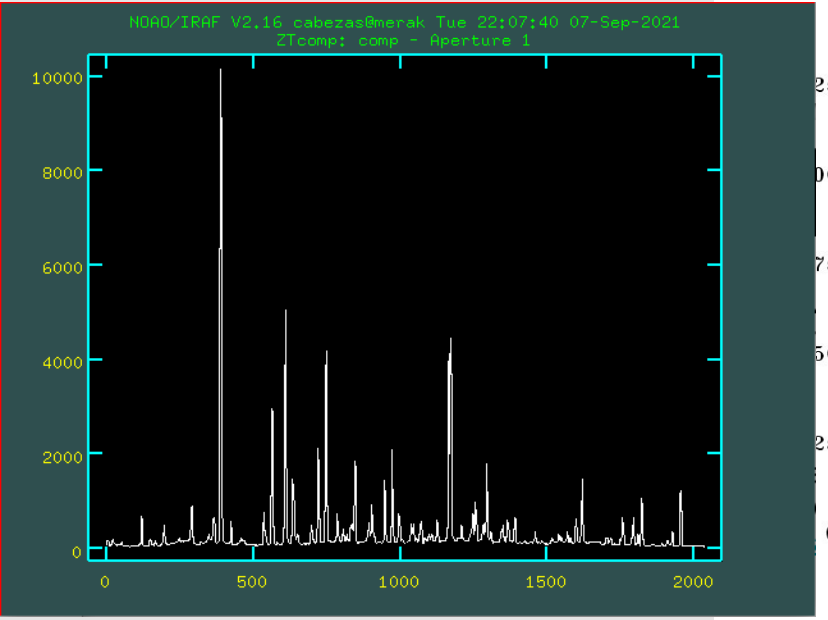
Review extracted spectrum for aperture 1 from ZTcomp? (yes) YES

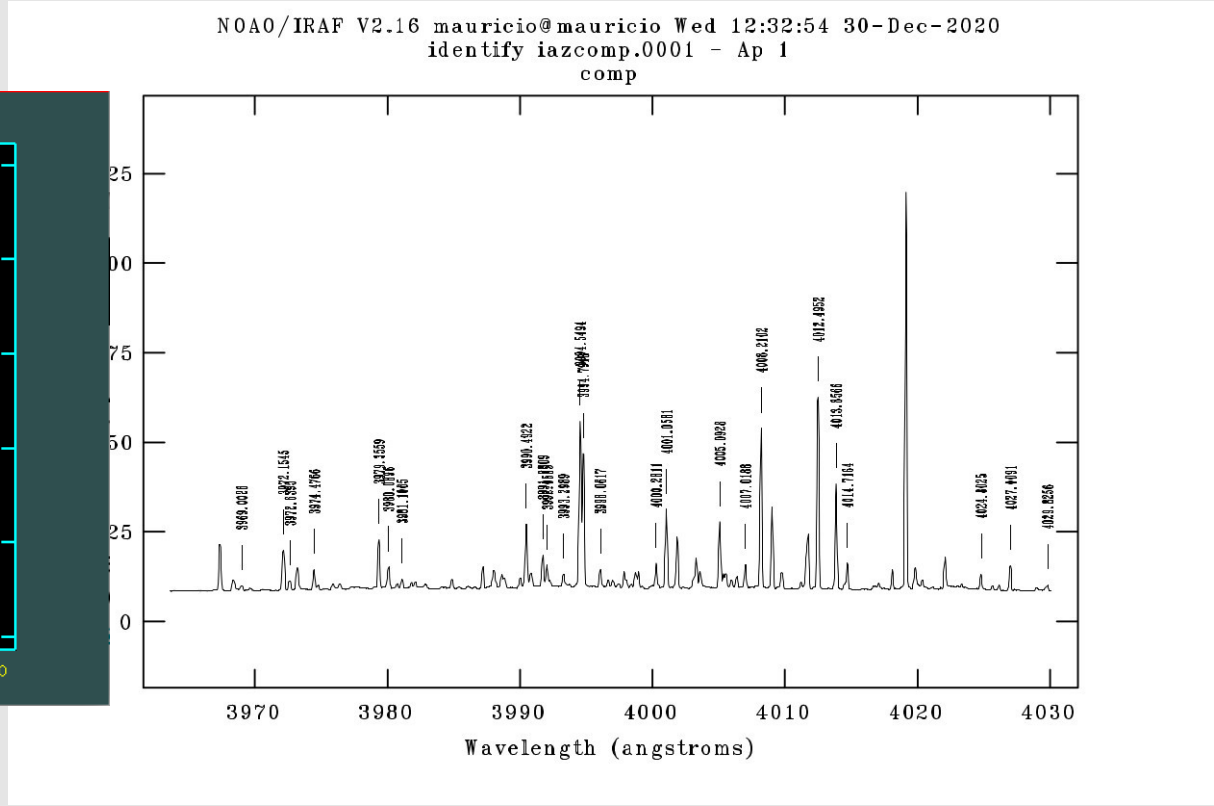
- if you are using database, check if the first aperture corresponds to the first aperture in the atlas. (inside folder **idcomp_***

```
                                # 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=                   no) Extract flat apertures?
(compapa=                  yes) Extract comparison apertures?
(iddatab=                   no) Use database folder for identification?
(idfolde=           idcomp_2307) folder name with identification database
(idencom=                   no) Identify features in spectrum for dispersion sol
```



9) Model apertures – Comparison lamp



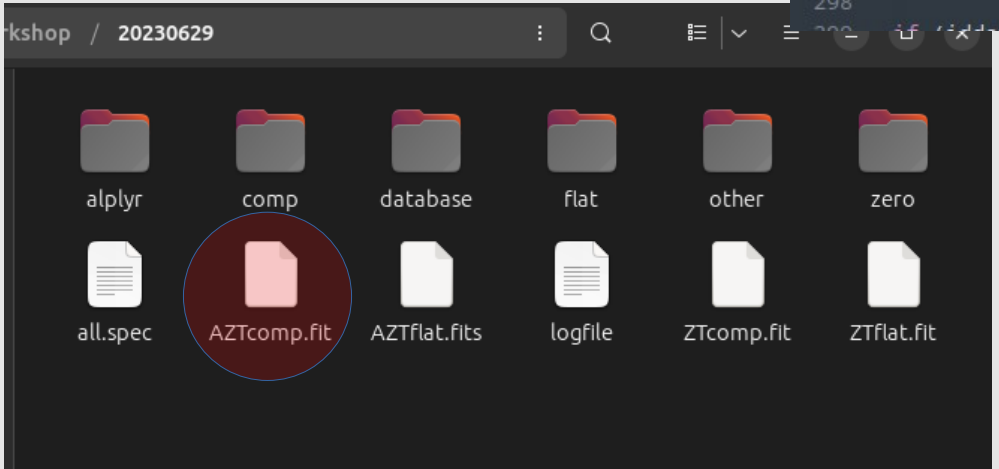


9) Model apertures – Comparison lamp

- reference → **AZTflat**

- new file: **AZTcomp.fit**

```
#####APERTURES - COMP
if (compapall==yes){
    apall.referen=inflat
    apall.format = "echelle"
    apall.find=no
    apall.recente=no
    apall.resize=no
    apall.trace=no
    apall.fittrace=no
    apall.extras=no
    apall.ylevel = 0.04
    apall.extract=yes
    apall (input=incomp, output="A"//incomp//".fit")
}
```



10) Wavelength calibration

- using database, folder “idcomp” must be in the main path

iftrimc = yes
compapa = no
iddatab = yes
idfolder = idcomp_2307
idencom = yes
:go

```
                                # 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=                   no) Extract flat apertures?
(compapa=                   no) Extract comparison apertures?
(iddatab=                  yes) Use database folder for identification?
(idfolde=          idcomp_2307) folder name with identification database
(idencom=                  yes) Identify features in spectrum for dispersion so
```

- in database, lines can be shifted few pixels,

in order to fix it we need to “shift” the lines of our database. If doesn’t work, recommend delete every line with **d**, and mark new lines **(m)** comparing with the atlas.

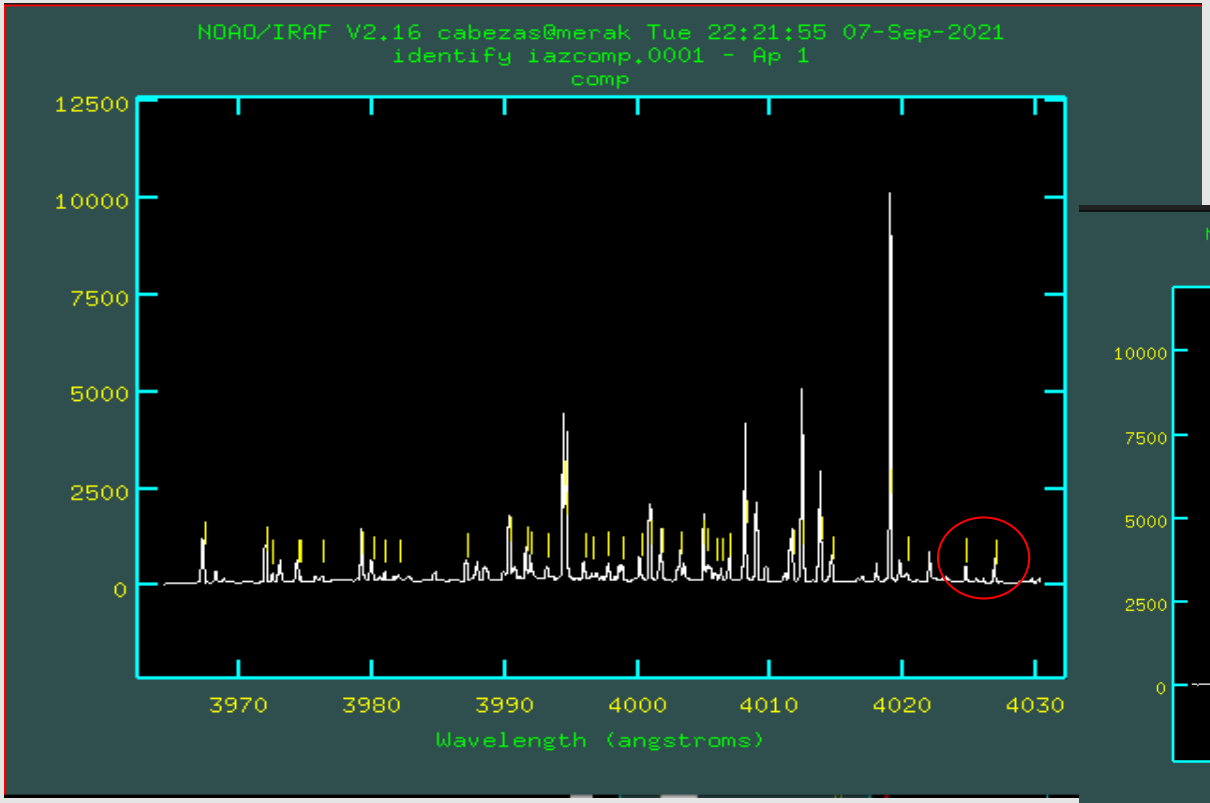
Shift → **s**

- always fit!

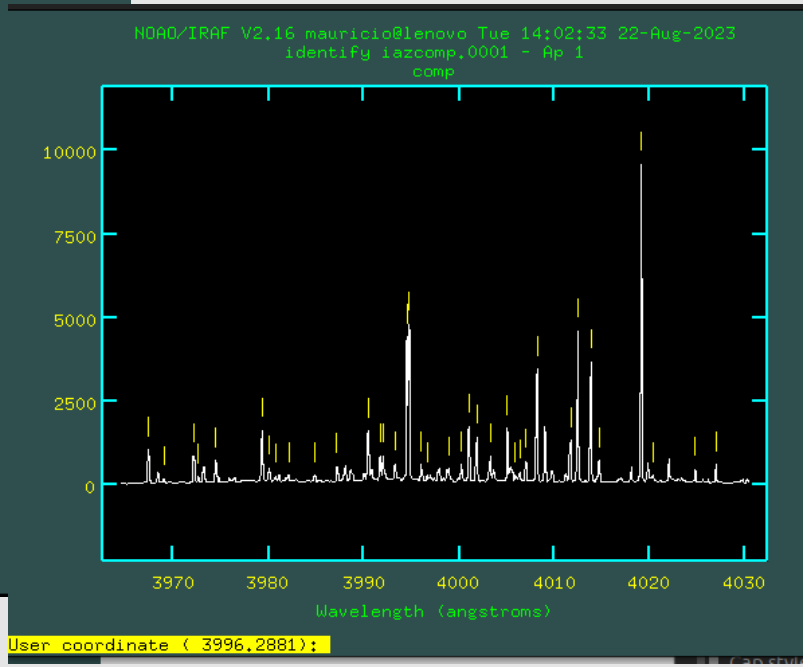
f (delete point **d**), **q, q**

- rms ~ 0.007 acceptable

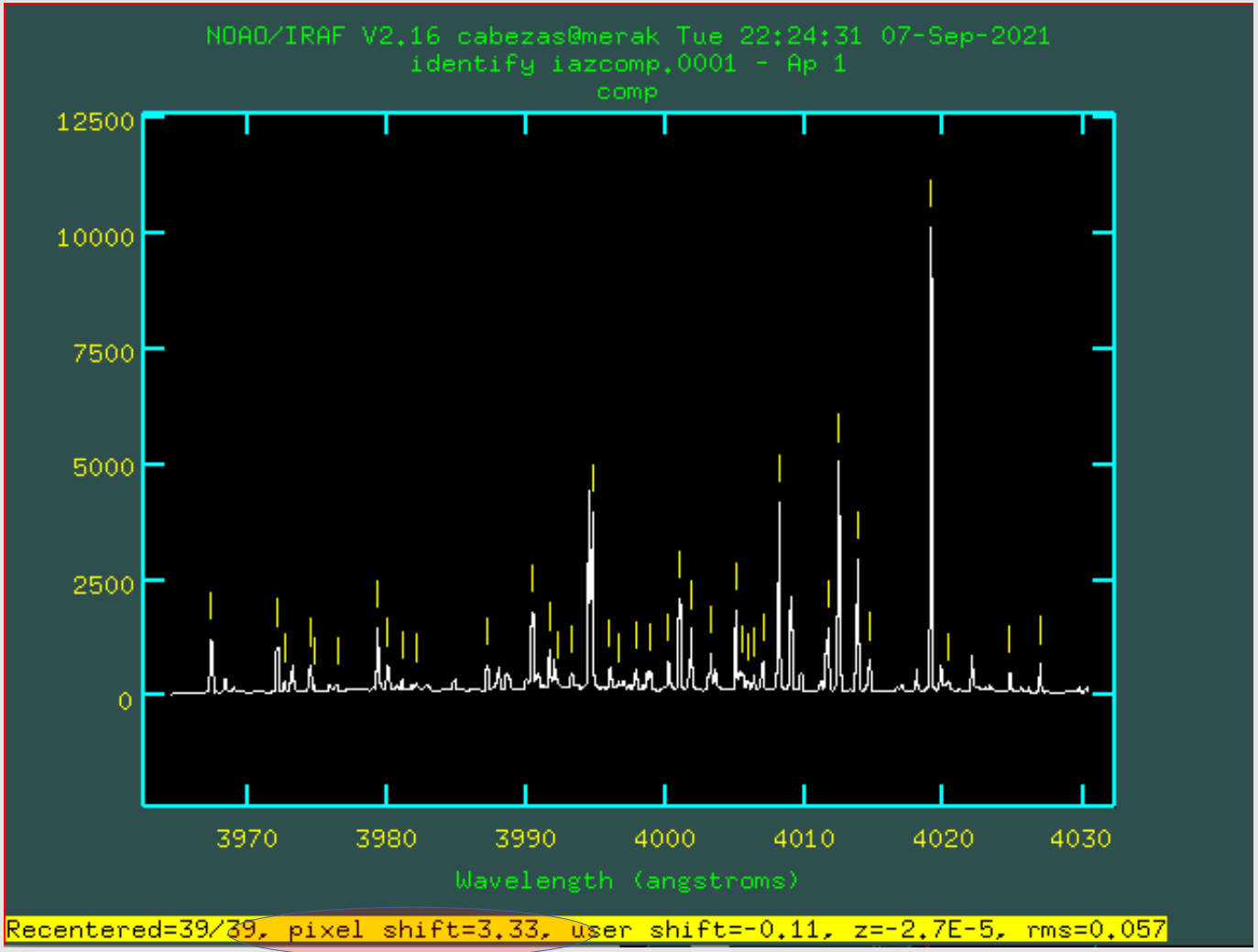
- Write feature data to the database (yes)?



No match!
Press **“s”**

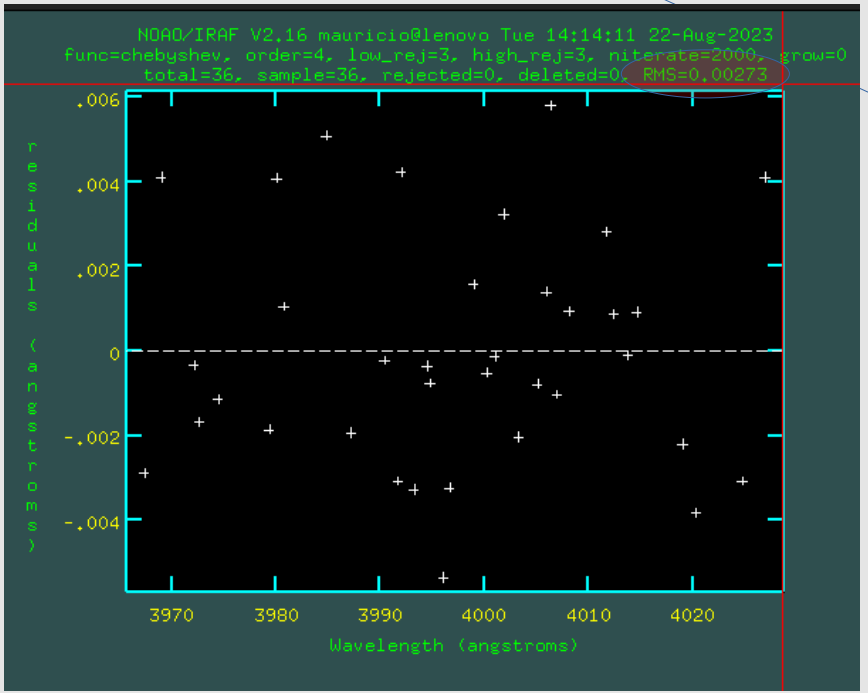


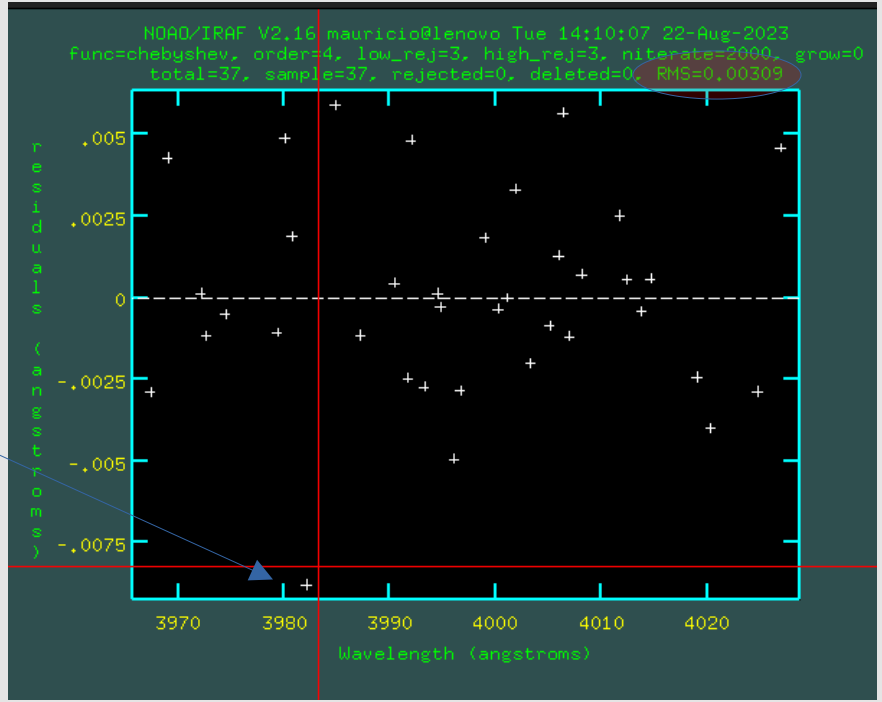
: enter


NOAO/IRAF V2.16 cabezas@merak Tue 22:24:31 07-Sep-2021
identify iazcomp.0001 - Ap 1
comp
12500
10000
7500
5000
2500
0
3970
3980
3990
4000
4010
4020
4030
Wavelength (angstroms)
Recentered=39/39, pixel shift=3.33, user shift=-0.11, z=-2.7E-5, rms=0.057

10) Wavelength calibration - fit

- NO pattern in residuals
- no high order is recommended
- **d** to eliminate point
- **q, f** to refit and update plot





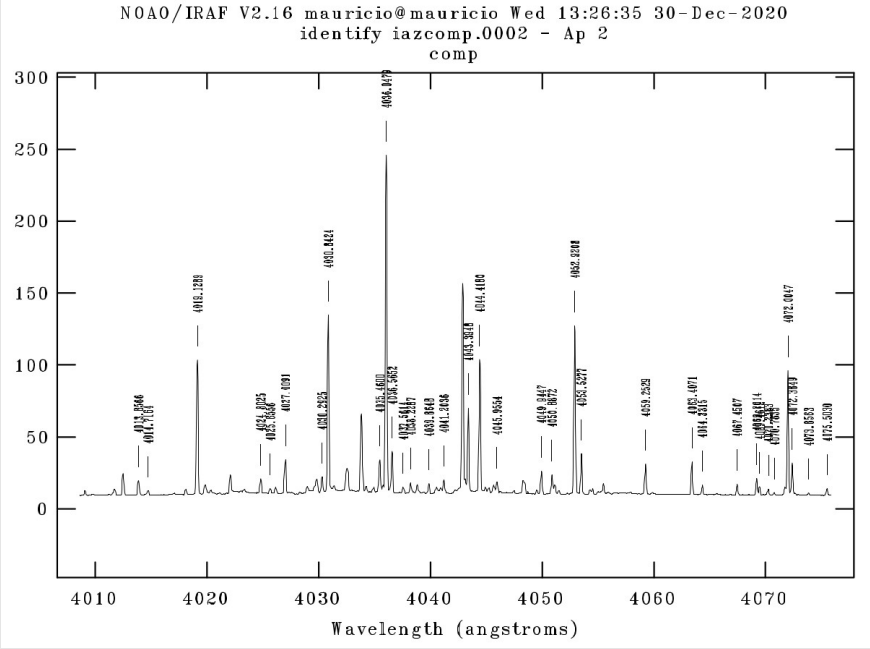
- better RMS, no significant
→ **q,q** to the next aperture

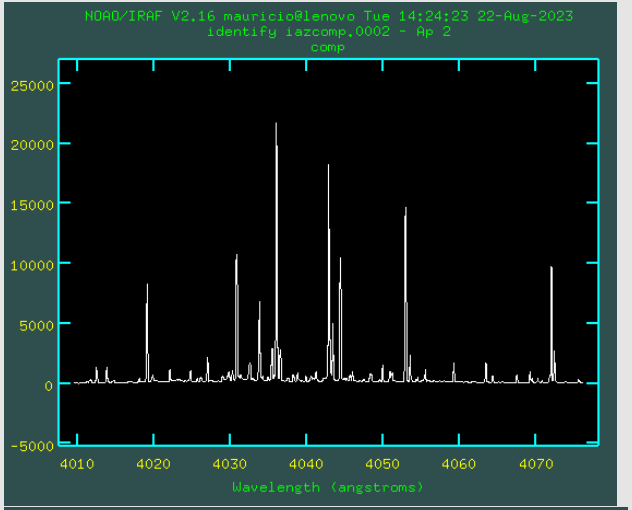
```
identify iazcomp.0001 - Ap 1
Write feature data to the database (yes)?
```

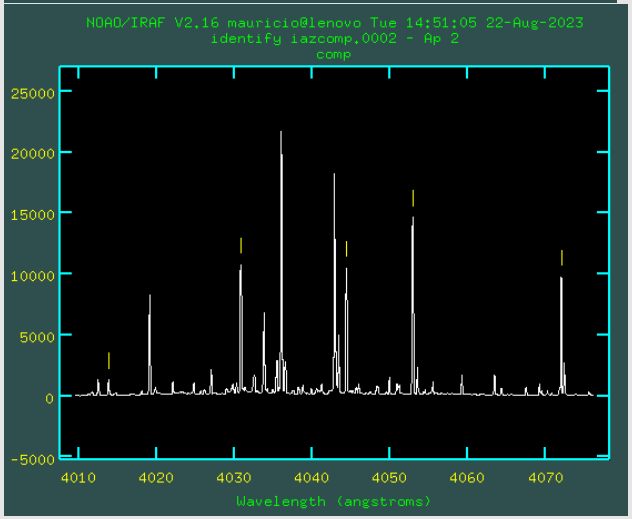
- yes, 49 times :)

10) Wavelength calibration – but..if not match?? :(

- delete all lines
- **NO press F!**
- compare the Atlas and mark manually > 5 lines
  - mark: **m,** wavelength, **enter**
  - delete: **d**
  - zoom in/out **(w,e,e)**
- now **f** to fit, check residuals**, q**



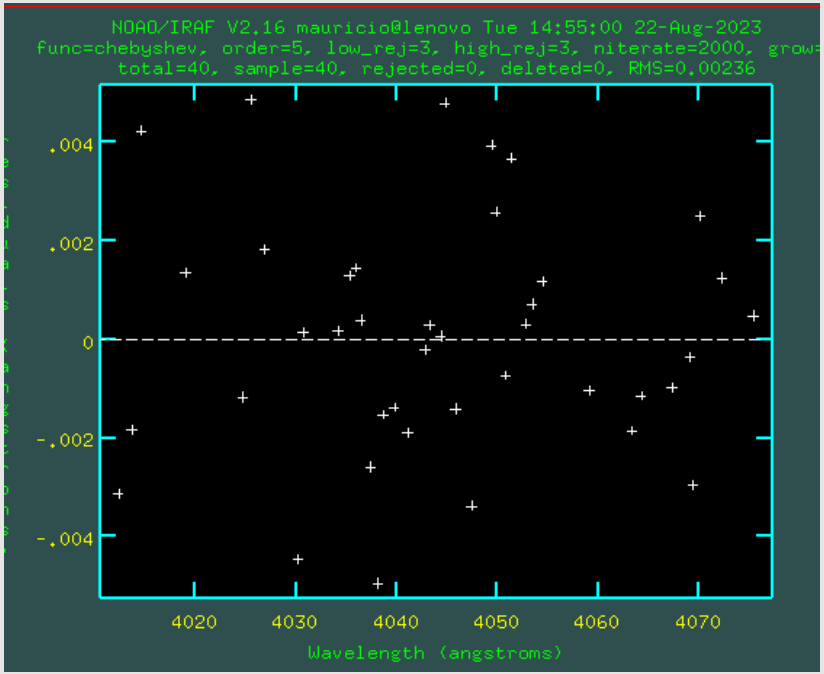


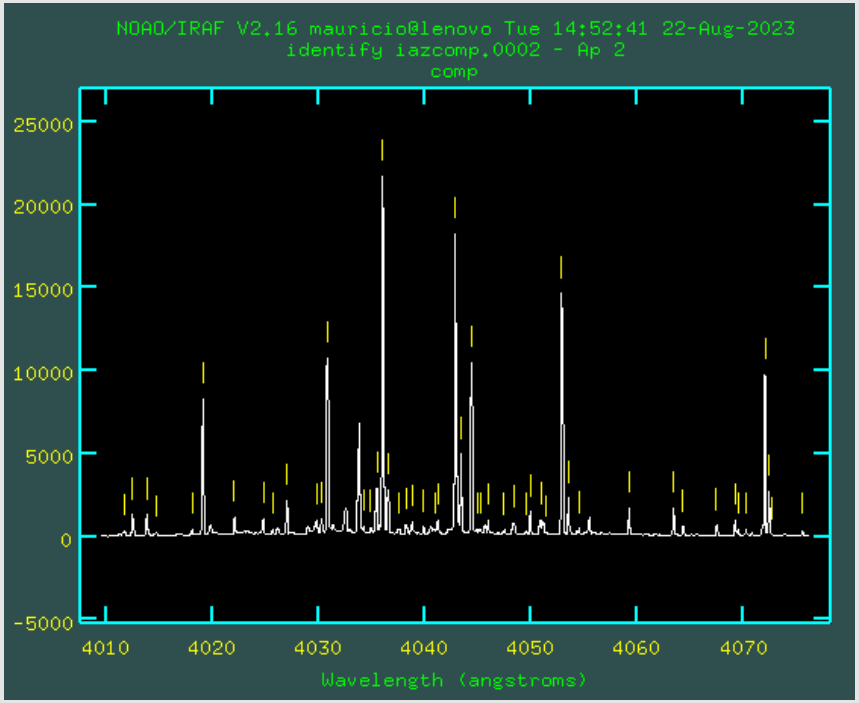


10) Wavelength calibration – but..if not match?? :(

- **L**, automatic mark
- **f,** refit
- ‘clean’

→ happy, **q,q**

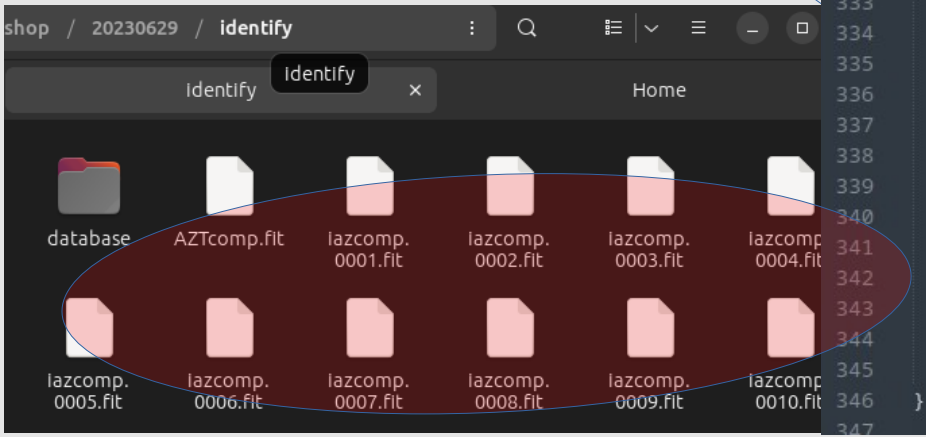




10) Wavelength calibration

- using the database
- for each aperture a file is created with the calibration information, which later will use as reference to the object's apertures.

- new files: **iazcomp.00**.fit**




```
##### IDENTIFY 1D
if (idencomp==yes){
    unlear directory
    unlearn scopy
    directory.sort=yes
    directory "identify/" | scan (iddir)
    if (iddir=="no"){
        mkdir (newdir="identify")
    }
    copy (input="A"//incomp,output="identify/")
    cd "identify/"
    lpar scopy
    scopy.format="onedspec"
    scopy (input="A"//incomp, output="iazcomp")
    print "second"
    unlearn refspectra
    unlearn hedit
    hedit.addonly=yes
    hedit.verify=no
    hedit.show=no
    for (i=1; i <=nap; i+=1) {
        printf ("iazcomp.00%02d.fit\n",(i)) | scan(ecname)
        if (iddatabase==yes){
            printf ("iazcomp.00%02d\n",(i)) | scan(refname)
            hedit (images=ecname, fields="REFSPEC1", value=refname)
        }
        identify.coordli="linelists$thar.dat"
        #lpar identify
        identify (images=ecname)
    }
    cd "../"
}
```

10) Wavelength calibration – idcomp_* database

- pixel

- Last one is the valid

- real/fitted wavelength

- marked wavelength

```
# Mon 16:09:08 14-Aug-2023
begin   identify iazcomp.0020 - Ap 20
    id  iazcomp.0020
    task    identify
    image   iazcomp.0020 - Ap 20
    aperture    20
    aplow   1245.04
    aphigh  1254.39
    units   Angstroms
    features    39
             44.33 5122.49923  5122.4995   4.0 1 1
            248.49 5115.04518  5115.0448   4.0 1 1
            349.35  5111.2775  5111.2781   4.0 1 1
            390.23 5109.73442  5109.7331   4.0 1 1
            613.85 5101.12945  5101.1299   4.0 1 1
            692.37 5098.04247  5098.0432   4.0 1 1
            731.70 5096.48345  5096.4848   4.0 1 1
            767.33 5095.06367  5095.0639   4.0 1 1
            891.65 5090.05504  5090.0513   4.0 1 1
            912.21 5089.21879  5089.2192   6.0 1 1
            986.44 5086.17967  5086.1774   6.0 1 1
           1015.29  5084.9903  5084.9935   6.0 1 1
           1100.62 5081.44574  5081.4462   6.0 1 1
           1242.46 5075.46606  5075.4659   4.0 1 1
           1261.72 5074.64575  5074.6465   6.0 1 1
           1385.29 5069.33375  5069.3384   6.0 1 1
           1416.62 5067.97393  5067.9737   4.0 1 1
           1435.82 5067.13772  5067.1379   6.0 1 1
           1444.03 5066.77966  5066.7773   4.0 1 1
           1458.76 5066.13623  5066.1355   4.0 1 1
           1493.74 5064.60369   5064.602   6.0 1 1
           1518.46 5063.51693  5063.5157   4.0 1 1
           1531.75 5062.93097  5062.9325   4.0 1 1
           1551.96 5062.03813  5062.0371   4.0 1 1
           1560.53 5061.65909  5061.6562   4.0 1 1
           1601.01 5059.86252  5059.8611   6.0 1 1
```

11) Trim Object

IMPORTANT, first check these parameters are ok!

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

input   = e202306290034.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     =                 0022) Observation id number
```

iftrimc = yes
iddatab = no
idfolder = idcomp
idencom = no

trimob = yes
iftrimo = yes
:go

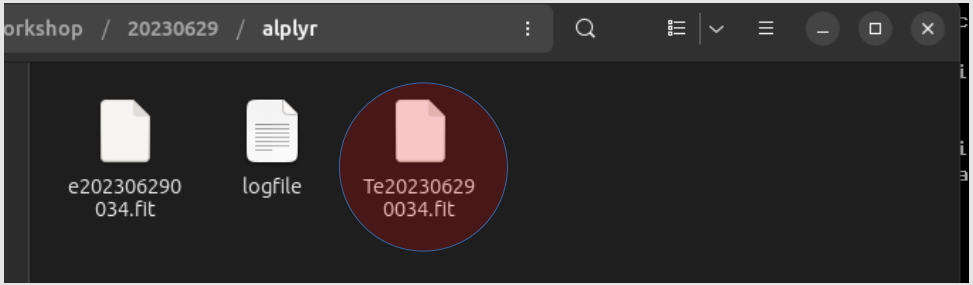
```
                              # OBJECT PARAMETERS

(trimob =                  yes) Trim object?
(iftrimo=                  yes) 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?

More
```

11) Trim Object

- trimsection has to be the same as calibration DON’T change!

```
if (trimob==yes){
    unlearn ccdproc
    ccdproc.trimsec = "[2:2035,*]"
    ccdproc.trim = yes
    ccdproc.fixpix = no
    ccdproc.overscan = no
    ccdproc.darkcor= no
    ccdproc.zerocor=no
    ccdproc.flatcor=no
    #
    cd (oname)
    ccdproc.ccdtype = "object"
    ccdproc (images=spec, output="T"//spec)
    cd "../"
}
```

New file: **alplyr/Te2023*.fit**

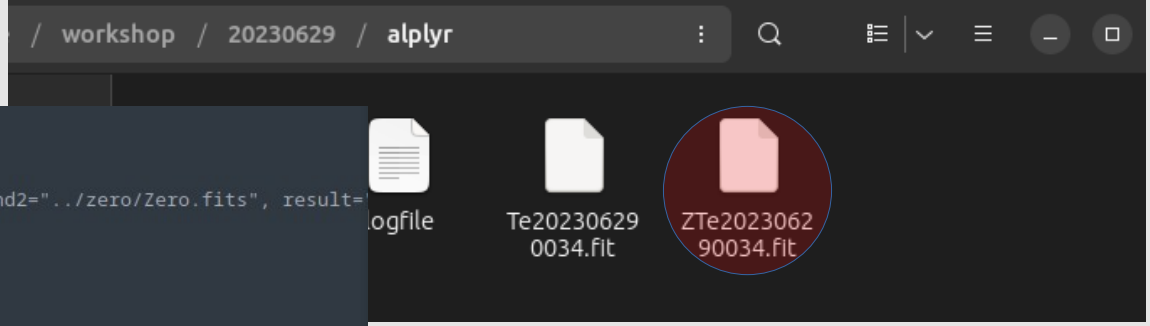


12) Bias correction object

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

New file: **alplyr/ZTe2023*.fit**

```
                                    # OBJECT PARAMETERS

(trimob =                     no) Trim object?
(iftrimo=                    yes) Use trim object?
(zerocor=                    yes) 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?
```




```
if (zerocorob==yes){
      cd (oname)
      #imarith (operand1=spec, op="/", operand2="../zero/Zero.fits", result=

      unlearn ccdproc
      ccdproc.ccdtype="zero"
      ccdproc.fixpix = no
      ccdproc.overscan = no
      ccdproc.darkcor= no
      ccdproc.zerocor=no
      ccdproc.flatcor=no
      #
      ccdproc.ccdtype="object"
      ccdproc.zerocor=yes
      ccdproc.zero="../zero/Zero.fit"
      ccdproc (images=inobject,output="Z"//inobject)
      cd "../"
}
```

13) Cosmic Rays – LACOS (*2001PASP..113.1420V*)

http://www.astro.yale.edu/dokkum/lacosmic/

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

```
                                    # OBJECT PARAMETERS

(trimob =                     no) Trim object?
(iftrimo=                    yes) Use trim object?
(zerocor=                     no) Apply zero level correction to object?
(crays  =                    yes) Remove cosmic rays?
(ifcrays=                    yes) 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?
```

- read GAIN and READtoNOISE from header

```
#####COSMIC RAYS - COMP
if (crays==yes){
     stsdas
#######
#read gain
cd (oname)
hselect (images="Z"//inobject,fields="GAIN", exp=yes) | scan (gainh)
hselect (images="Z"//inobject,fields="READNOIS", exp=yes) | scan (readnh)
#print (gainh)
inputCR="Z"//inobject
outputCR="CrZ"//inobject
outmaskCR="MCrZ"//inobject
gainCR = gainh # 2 #3
readnCR = readnh #2
xorderCR = 3
yorderCR = 3
sigclipCR = 4.5
sigfracCR = 0.3
objlimCR = 0.75 #0 to 5, 5 more conservative discrimination
niterCR = 5
verboseCR = no
```

New files: **alplyr/CrZTe2023*.fit** ← cleaned
alplyr/MCrZTe2023*.fit ← mask

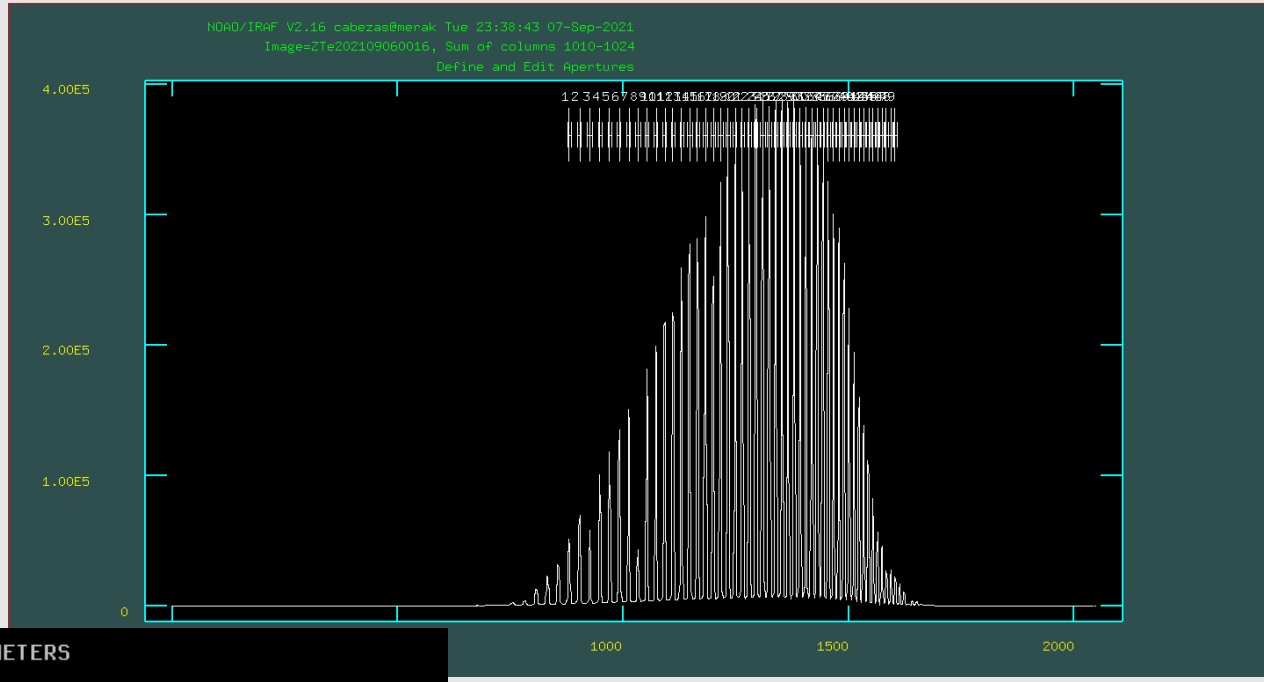
workshop / 20230629 / alplyr

CrZTe20230 6290034.fit | e202306290 034.fit | logfile | MCrZTe2023 06290034.fit | Te20230629 0034.fit | ZTe202306 90034.fit

14) Extract apertures - Object

- Template: AZTflat.fit

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




```
                                # OBJECT PARAMETERS

(trimob =                    no) Trim object?
(iftrimo=                   yes) Use trim object?
(zerocor=                    no) Apply zero level correction to object?
(crays  =                    no) Remove cosmic rays?
(ifcrays=                   yes) Use object with cosmic rays extraction?
(objecta=                   yes) Extract object apertures?
(flatcor=                    no) Apply flat correction to object?
(helioco=                    no) calculate JD + heliocentric correction?
(idref  =                    no) refer database identification to images?
```

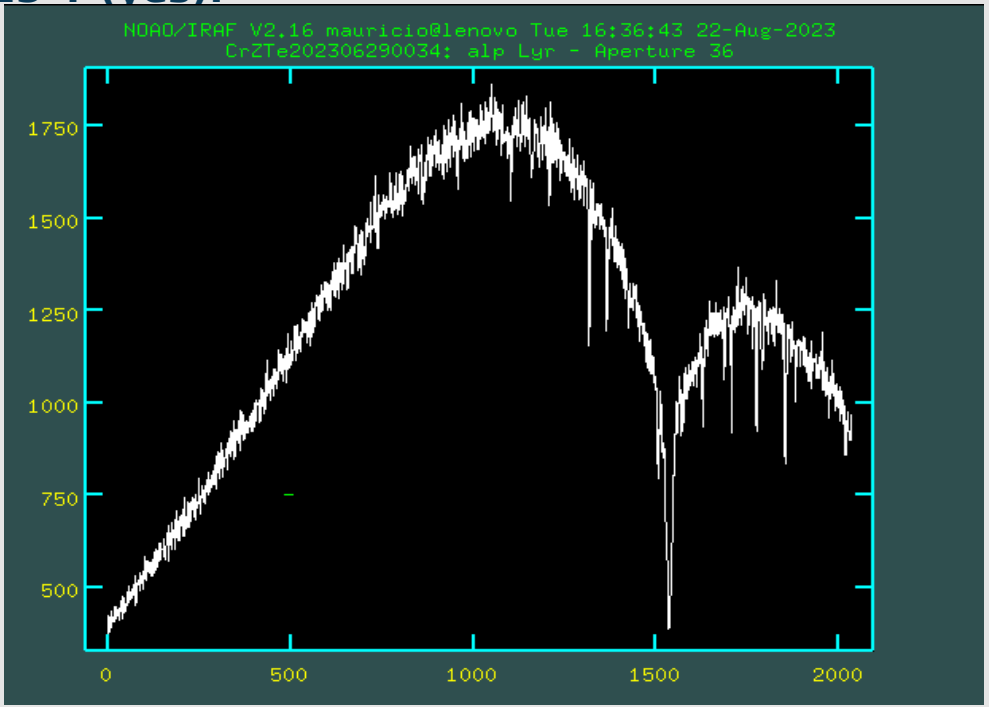
14) Extract apertures - Object

Edit apertures for CrZTe2023*? (yes):
Review extracted spectrum for aperture 1 from CrZTe2023*? (yes)
Write apertures for CrZTe2023* to database? (yes):
q → if yes, with q you move to the next aperture
Extract aperture spectra for CrZTe2023*? (yes):

First view of our target!!!
BUT!! :)

Still in pixel :(

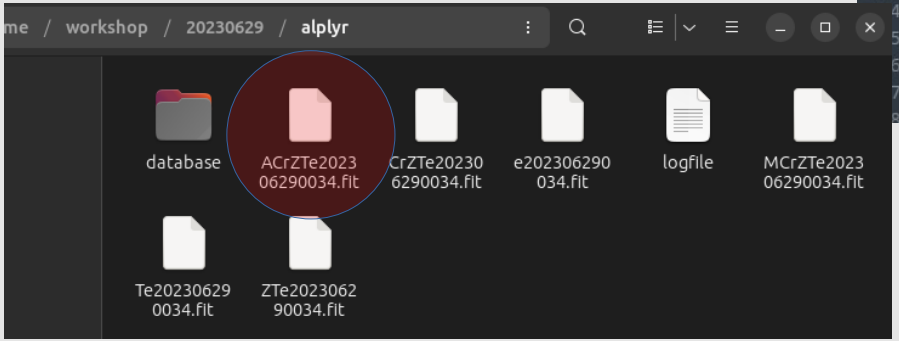
For example aperture 36 is
the spectral region of H-alpha



14) Extract apertures - Object

```
#####APERTURES - OBJECT
if (objectapall==yes){
     apall.referen=inflat
     apall.format = "echelle"
     apall.find=no
     apall.recente=no
     apall.resize=no
     apall.trace=no
     apall.fittrace=no
     apall.extras=no
     apall.extract=yes
     apall.edit=edit_o
     apall.review=review_o
     ## check database
     unlear directory
     directory.sort=yes
     directory oname//"/database/" | scan (iddir)
     if (iddir=="no"){
          mkdir (newdir=oname//"/database")
          copy (input="database/*",output=oname//"/database/")
     }
     ##
     cd (oname)
     apall (input=inobject, output="A"//inobject)
     cd "../"
}
```

new file: **alplyr/ACrZTe2023*.fit**



15) Flat correction

iftrimc = yes

iftrimo = yes

ifcrays = yes/no

objecta = no

flatcor = yes

:go

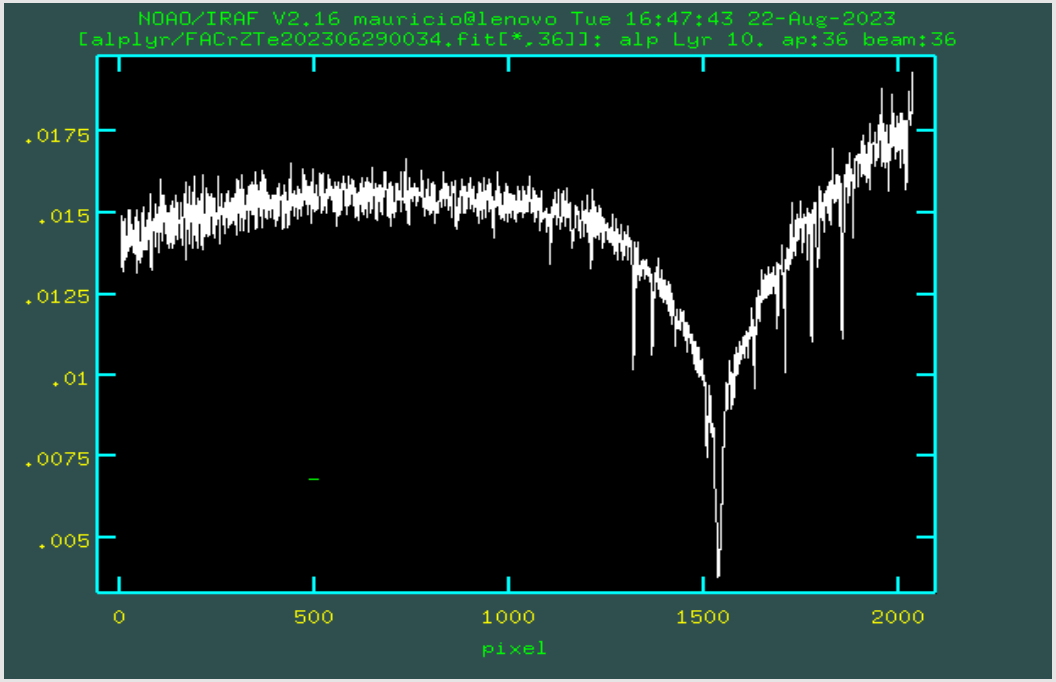
To see or plot use the task

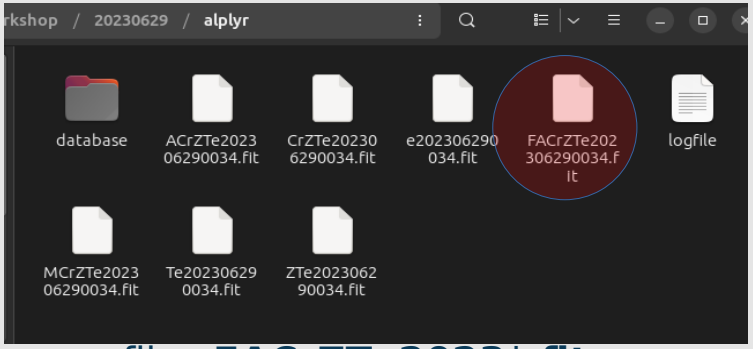
splot spec.fit

To move between orders

(→ to the right/ higher aperture

) → to the left/ lower aperture





new file: **FACrZTe2023*.fit**

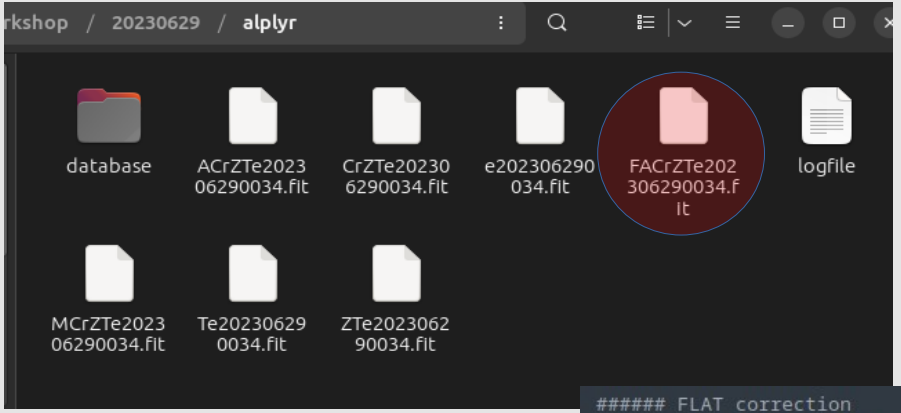
```
                    # OBJECT PARAMETERS

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

15) Flat correction

In Flat correction we have to divide. By dividing the science spectrum by the flat-field image, we are effectively correcting for the variations in

- sensitivity
- illumination
- detector response.

new file: **FACrZTe2023*.fit**




```
###### FLAT correction
#print ("FA"//inobject)
if (flatcor==yes){
     imarith (operand1=oname//"/A"//inobject, op="/", operand2="A"//inflat, result=oname//"/FA"//inobject)

     #imcopy("FA"//inobject,oldoutput,verb-)
     #move (files="FA"//inobject,newdir=oname//"/")

}
```

16) JD + heliocentric correction

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

```
(ifcrays=                    yes) 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?

More
20:0:17CrZTe202306290034.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
2023  6 29 20:00:17 18:36:56  38:47:01      0.0
2460125.33628      0.00     2.07    21.70      0.171   -0.004    1.907   19.625
FACrZTe202306290034.fit
ecl>
```

Check long header:
imhead FCrAZTe2023*.fit l+

```
CD2_2   =                    1.
HJD     =     2460125.33628259
VHELIO  =     2.07395459376786
VLSR    =     21.6990646774371
VSUN    = '   20.    18.    30.  1900.'
ecl>
```

16) JD + heliocentric correction

```
#CALC UTMIDDLE
hselect (images="other/"//spec,fields="TM_START", exp=yes) | scan (utstart)
hselect (images="other/"//spec,fields="EXPTIME", exp=yes) | scan (expt)
utmidhr=int((utstart + expt/2)/3600)
utmidmin=int((((utstart + expt/2)/3600)-utmidhr)*60)
utmidsec=int(((((utstart + expt/2)/3600)-utmidhr)*60 - utmidmin)*60)
utmid = (utmidhr//":"//utmidmin//":"//utmidsec)
printf ("%d:%d:%d\n",utmidhr,utmidmin,utmidsec) | scan (utmid)
```

UTMIDDLE

```
# ##### SETJD + HELIOCOR
if (heliocor==yes){
    cd (oname)
    #
    print (utmid,inobject)
    unlearn hedit
    hedit.addonly=no
    hedit.verify=no
    hedit.show=no
    hedit (images="FA"//inobject, fields="UT", value=utmid)
    ##hedit (images="N"//oname//".fit", fields="UT", value=utmid)
    # hedit (images="N"//oname//".fits", fields="UTMIDDLE", value=utmid)
    unlearn setjd
    setjd.observatory="ondrejov"
    setjd.epoch="epoch"
    setjd.time="UTMIDDLE"
    setjd (images="FA"//inobject)
    rv
    unlearn rvcorrect
    rv.rvcorrect.input=yes
    rv.rvcorrect.imupdate=yes
    rv.rvcorrect.epoch=2000.
    rv.rvcorrect.observatory="ondrejov"
    rv.rvcorrect (images="FA"//inobject)
    ##save header
    print ("FA"//inobject)
    #imheader (images="FA"//inobject)
    imhead ("FA"//inobject, l+ ,> oname//".hd")
    #dopcor.isvelocity=yes
    #dopcor (input="N"//oname//id//".fit", output=oname//"-"//id//".fit", re
    #system.move (files=oname//"-"//id//".fit",newdir="../")
    cd "../"
}
```

17) Ref spectrum

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

```
                                # OBJECT PARAMETERS

(trimob =                   no) Trim object?
(iftrimo=                  yes) Use trim object?
(zerocor=                   no) Apply zero level correction to object?
(crays  =                   no) Remove cosmic rays?
(ifcrays=                  yes) 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  =                  yes) refer database identification to images?
(norm   =                   no) normalize spectra?

More
```

```
wap.0045.fit: ap = 45, w1 =   7839.5, w2 = 7970.266, dw = 0.064321, nw = 2034
[ap.0046] refspec1='iazcomp.0046'
ap.0046.fit: REFSPEC1 = 'iazcomp.0046 1.'
wap.0046.fit: ap = 46, w1 = 8017.647, w2 = 8151.399, dw =  0.06579, nw = 2034
[ap.0047] refspec1='iazcomp.0047'
ap.0047.fit: REFSPEC1 = 'iazcomp.0047 1.'
wap.0047.fit: ap = 47, w1 = 8204.084, w2 = 8340.946, dw =  0.06732, nw = 2034
[ap.0048] refspec1='iazcomp.0048'
ap.0048.fit: REFSPEC1 = 'iazcomp.0048 1.'
wap.0048.fit: ap = 48, w1 = 8399.397, w2 = 8539.591, dw = 0.068959, nw = 2034
[ap.0049] refspec1='iazcomp.0049'
ap.0049.fit: REFSPEC1 = 'iazcomp.0049 1.'
wap.0049.fit: ap = 49, w1 = 8605.969, w2 = 8788.151, dw = 0.089612, nw = 2034
ecl>
```

17) Ref spectrum

Read database of wavelength calibration and apply!

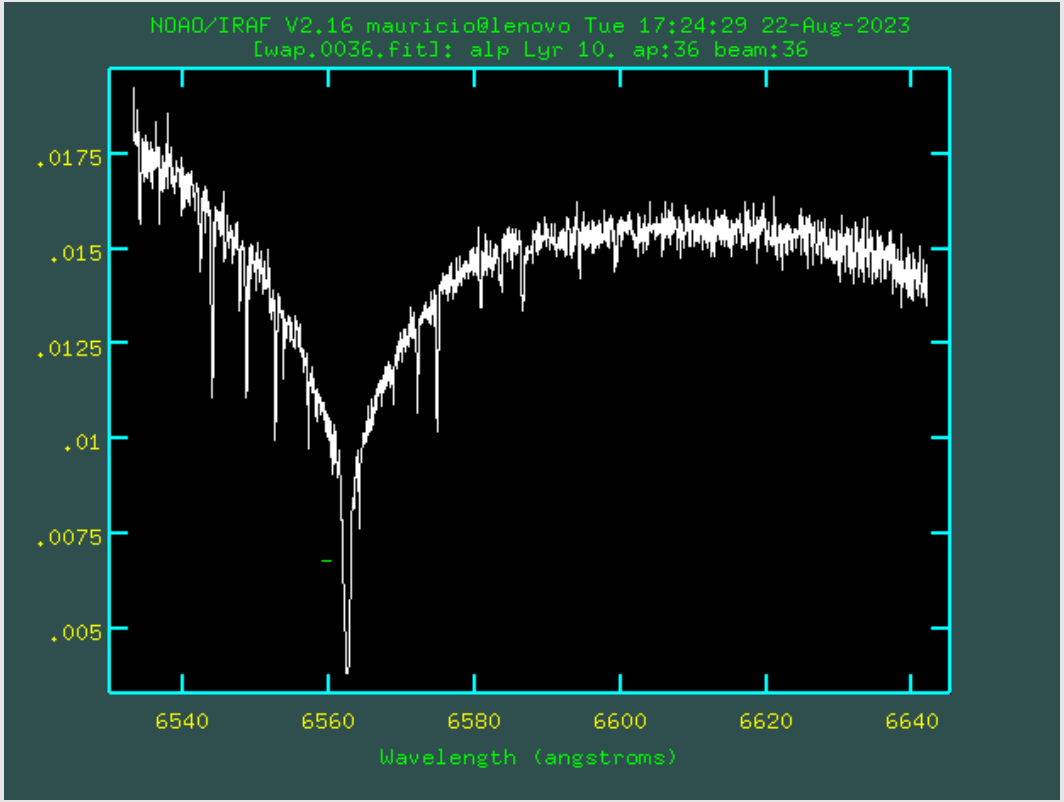
splot wap.0036.fit

NO more pixels!!! :)

→ aperture 36 → H-alpha

Check if ok

Good Wavelength calibration !!



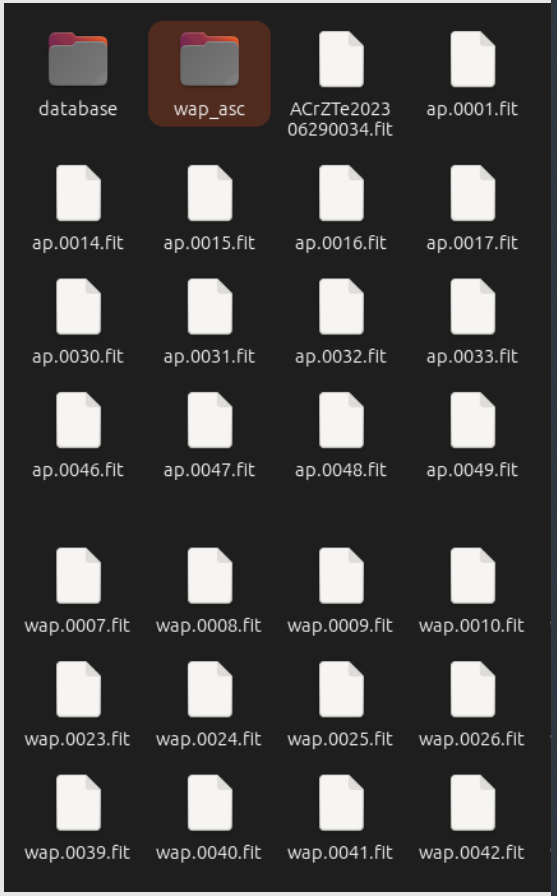
17) Ref spectrum

New files:

ap.00XX.fit –
1D spectra in pixels

wap.00XX.fit –
1D spectra in Angstrom

wap_asc/wap.00XX.asc –
ascii file for each aperture (useful)




```
if (idref==yes){
    unlear directory
    directory.sort=no
    directory oname//"/" | scan (iddir)
    if (iddir=="no"){
        mkdir (newdir=oname)
    }
    cd oname//"/"
    unlear onedspec
    scopy.format="onedspec"
    scopy (input="FA"//inobject, output="ap")
    unlearn refspectra
    dispcor.w1=INDEF
    dispcor.w2=INDEF
    dispcor.nw=INDEF
    dispcor.flux=no
    cd "../identify/"
    system
     for (i=1; i <=nap; i+=1) {
        printf ("ap.00%02d.fit\n",(i)) | scan(oap)
        printf ("iazcomp.00%02d.fit\n",(i)) | scan(ecname)
        refspectra.sort="epoch"
        refspectra.group="epoch"
        refspectra.answer=yes
        refspectra.confirm=no
        system.move (files="../"//oname//"/"//oap, newdir="../identify/")
        refspectra (input=oap, referen=ecname)
        dispcor (input=oap, output="w"//oap)
        system.move (files=oap,newdir="../"//oname)
        system.move (files="w"//oap,newdir="../"//oname)
    }
    cd "../"
    cd oname//"/"
    wspectext.header=no
    mkdir (newdir="wap_asc")
    for (i=1; i <=nap; i+=1) {
        printf ("ap.00%02d\n",(i)) | scan(oap)
        wspectext (input="w"//oap//".fit",output="w"//oap//".asc")
        system.move (files="w"//oap//".asc",newdir="wap_asc")

    }
    system.move (files=oname//".hd",newdir="wap_asc")
    cd "wap_asc"
    !ls wap*.asc > norm.list
    #### create python script
    cd "../../"
```

18) Normalization

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

```
(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   =                   yes) normalize spectra?

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

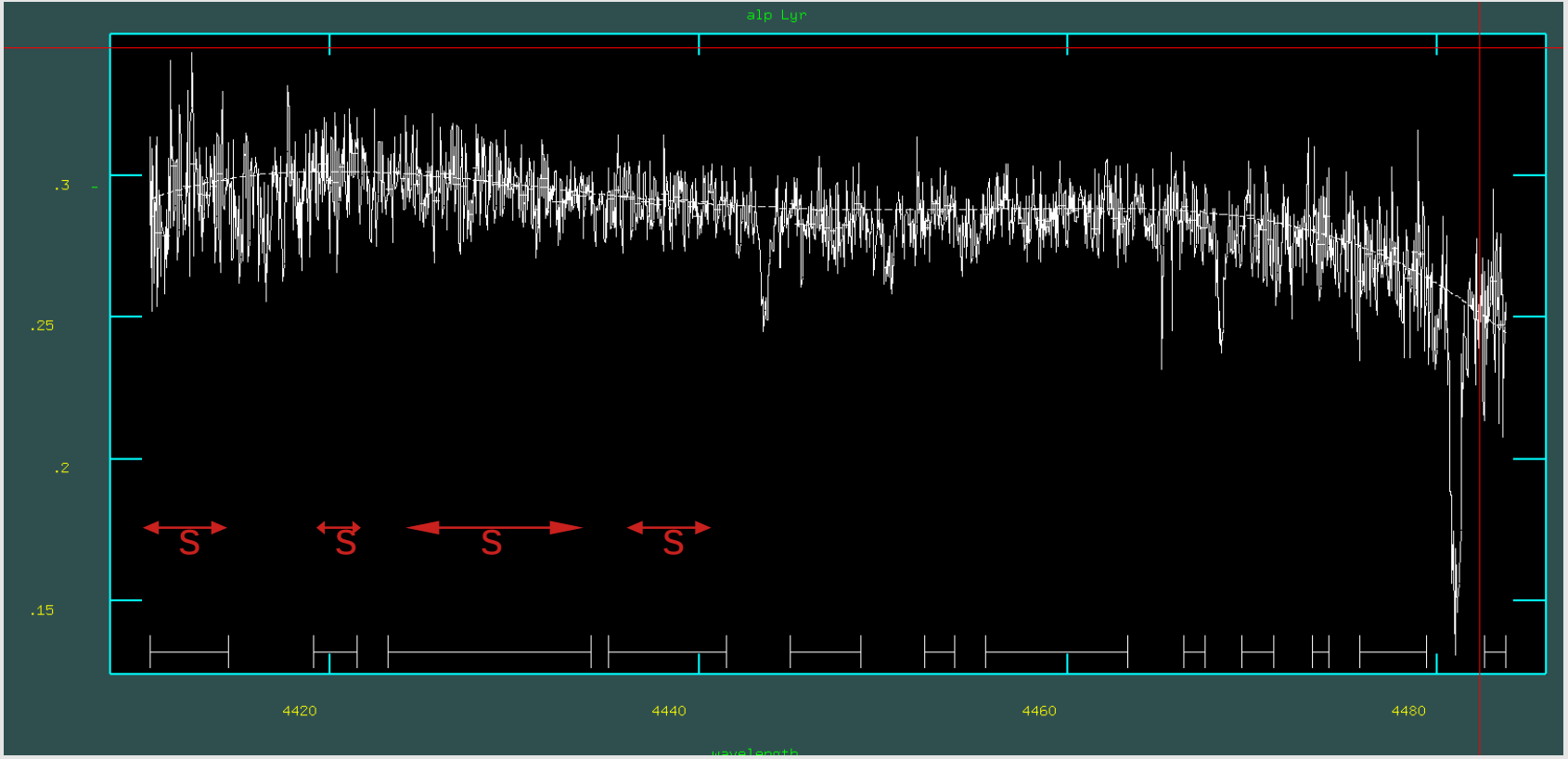
Fit [1,1] of wap.0001.fit w/ graph?
(yes|no|skip|YES|NO|SKIP) (yes):

18) Normalization

- 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**

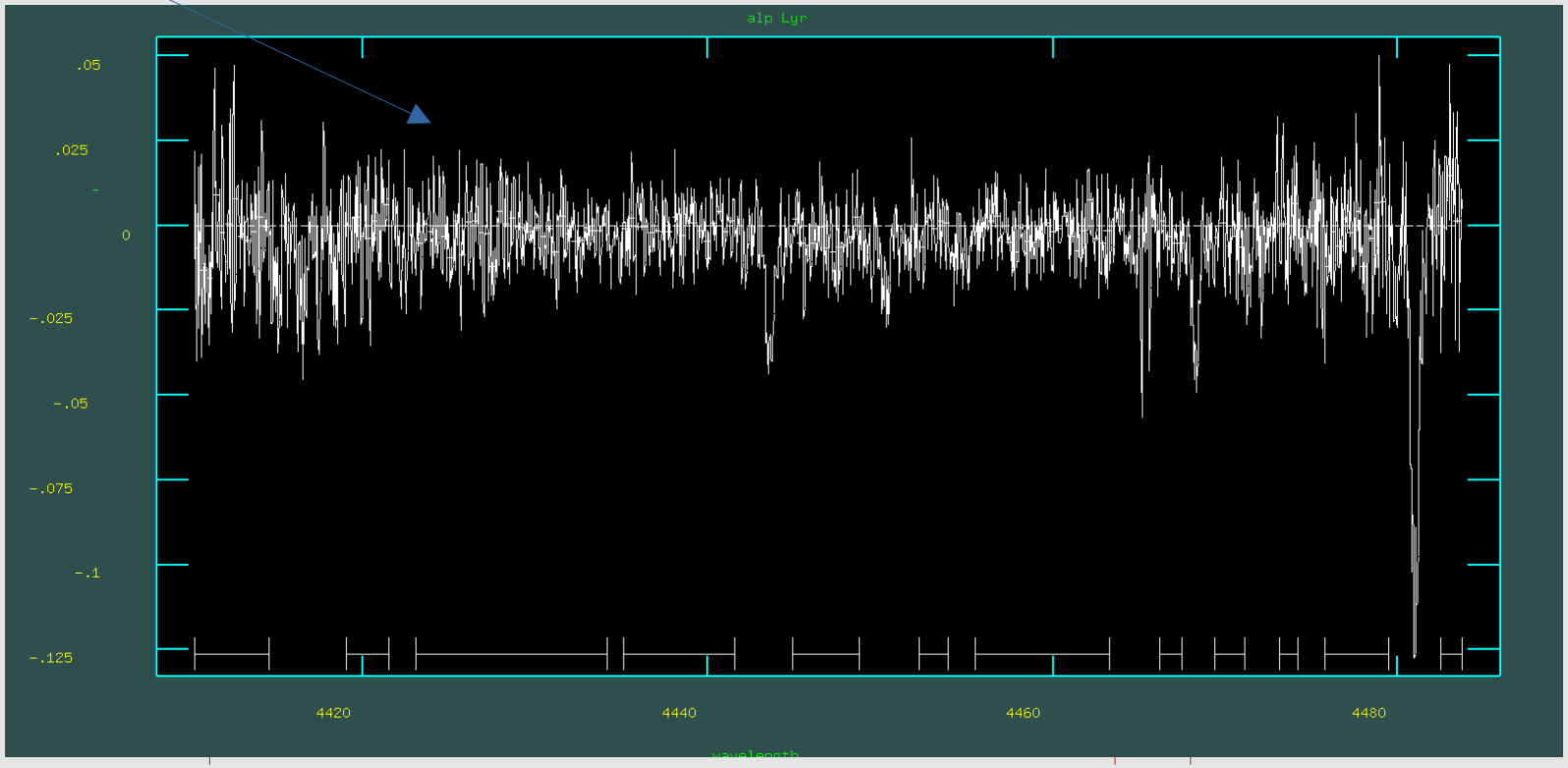


18) Normalization

- 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**



18) Normalization

Advises / Good practices

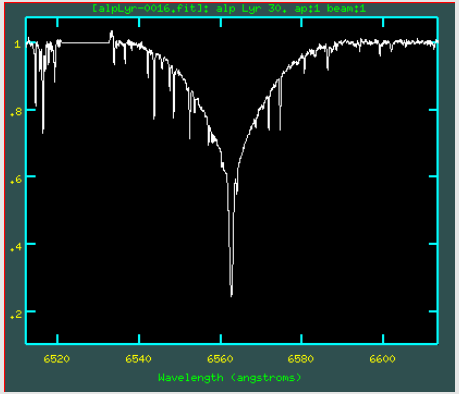
- Start with low order
- Hot stars or wide lines → Legendre or chebyshev
- sometimes faster just change high/low rejection
- Dont frustrate if poor normalization is Just experience, there is not ABSOLUTE way

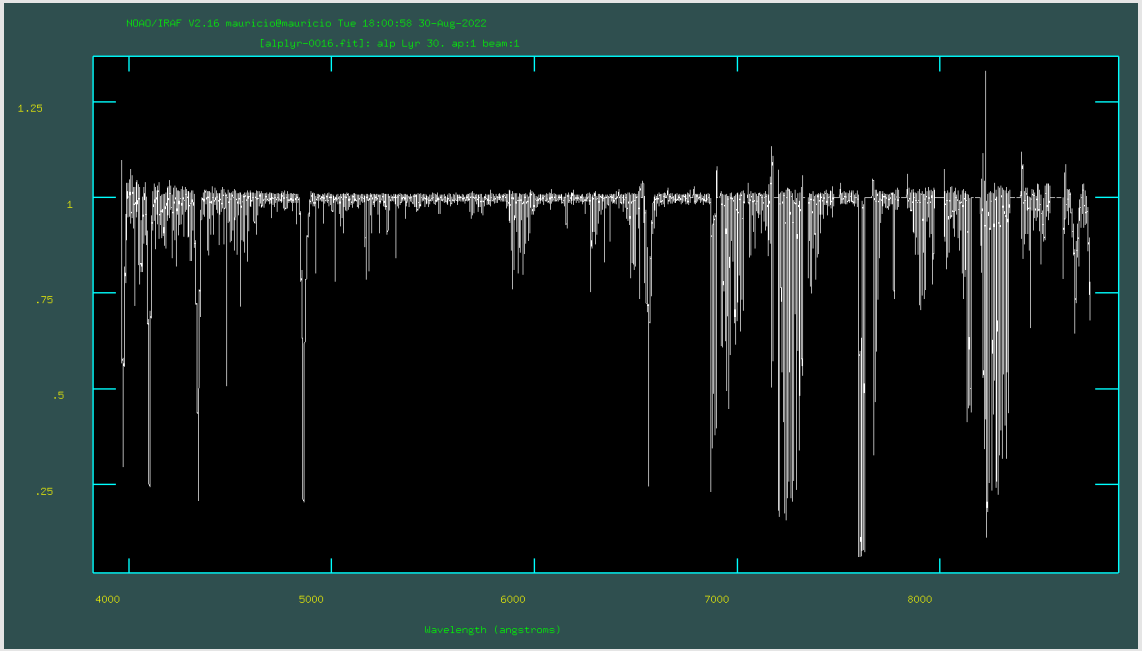
```
##### normalization
if (norm==yes){
    cd (oname)
    unlearn continuum
    unlearn scombine
    continuum.type="fit"
    continuum.function=cfunction
    continuum.order=corder
    continuum.naverage=10
    continuum.markrej=no
    continuum.niterat=2000
    continuum.high_re=2
    continuum.low_re=1.5
    continuum.grow=0
    for (i=1; i <=nap; i+=1) {
        printf ("wap.00%02d.fit\n",(i)) | scan(wap)
        printf ("fap.00%02d.fit\n",(i)) | scan(fap)
        printf ("nap.00%02d.fit\n",(i)) | scan(nnap)
        print (wap, fap, nnap)
    continuum (input=wap, output=fap)
    }
    cd "../"
}
```

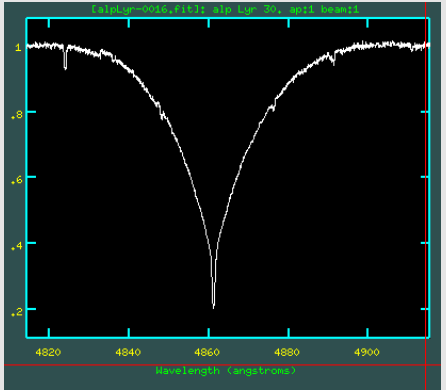
21) merging

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

new file: **alpLyr-0022.fit**







SPLOT

splot alpLyr-0016.fit

Fit: gaussian: **k..k(**or **g)**

lorentzian: **k..l**

voigt: **k..v**

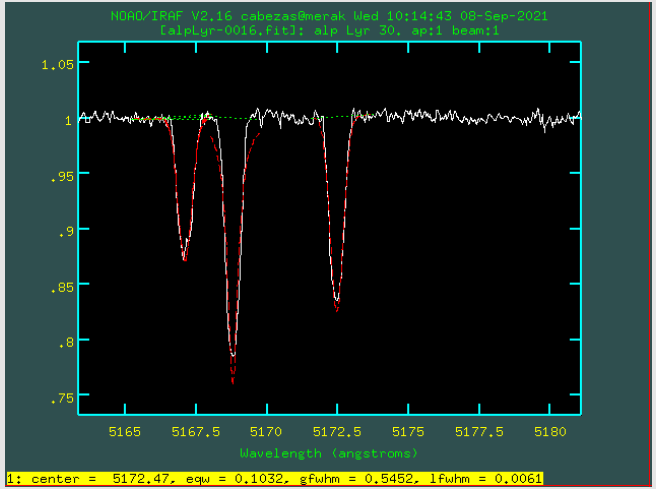
centroid **e..e**

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**

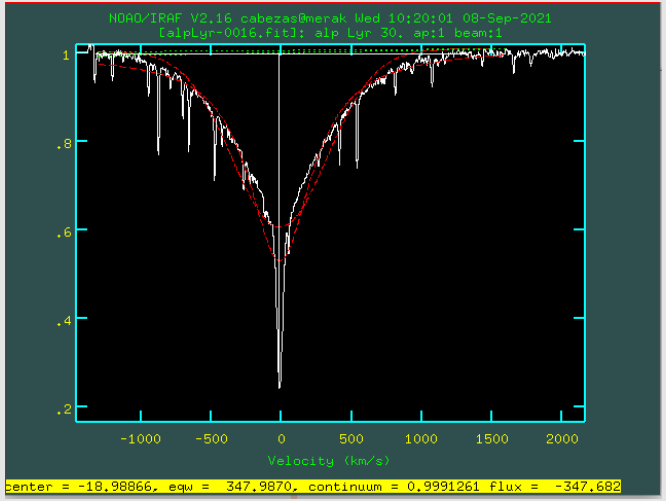
snr: **m..m**

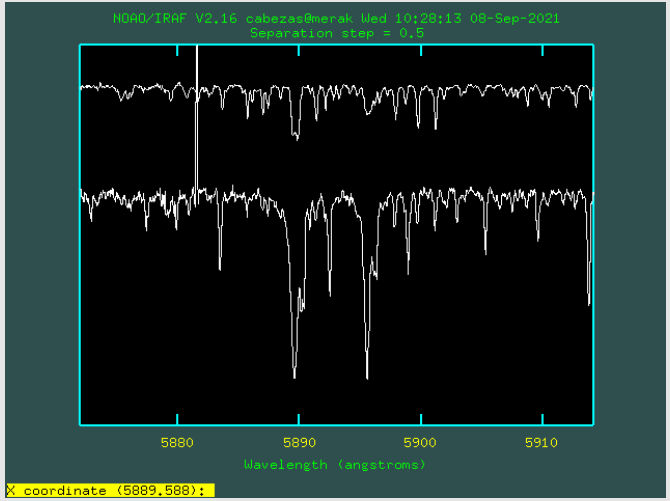
Change unit (angstrom to km/s)

:u km/s 6562.8 an

:u an

Replot: **r**





General Remarks

- The optimal reduction process always will be different for each instrument.
- IRAF “sometimes” is a bit tricky, but really useful.
- Quick check/inspection of spectra!!
- Versatil program because many parameters (sometimes too many).
- 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.

Thank you!!