

Primer redukcije posmatranja u IRAF¹-u

Zadatak ove vežbe je upoznavanje studenata sa elementarnom redukcijom posmatranja korišćenjem IRAF-a. Takođe, ovo je prilika i za upoznavanje sa radom u Linux operativnom sistemu.

U ovoj vežbi koristimo posmatranje spektra zvezde 7 Vul² (HD 183537, HR 7409, HIP 95818) od 09. 10. 2009. godine realizovano na Ondřejov opservatoriji, Češka republika (CCD SITe 800×2030 pix kamera na coudé spektrografu 2m-teleskopa). Opseg talasnih dužina je od 6255 do 6767 Å.

Jasno je da na fotone, detektovane CCD kamerom utiče veliki broj faktora: prostiranje kroz međuzvezdanu sredinu i Zemljinu atmosferu, optika instrumenata kao i karakteristike samog CCD-a. U tom smislu, pre naučne analize neophodno je izvršiti redukciju posmatranja.

Bitno je istaći da je dole navedena jedna varijanta redukcije posmatranja u IRAF-u. Svakako, do istih rezultata se može doći i neznatno drugačijim nizom naredbi. Takođe, u zavisnosti od tipa posmatranja zavisice i način redukcije ali su sve ključne stavke, opisane u ovoj vežbi, nužno prisutne.

1 Priprema za redukciju posmatranja

Pokrenimo IRAF. Kada ga startujemo bitno je da proverimo u kom direktorijumu se nalazimo. To radimo uz pomoć komande: `pwd`.

Sada, kada znamo u kom direktorijumu se nalazimo, prelazimo u direktorijum gde se nalaze naši podaci: `cd raw_data`. Možemo se koristiti prečicom pri kucanju adrese našeg direktorijuma: otkucamo par slova naziva i onda **Tab** - u komandnoj liniji će biti dopisan potpun naziv foldera. Komandom `ls` možemo izlistati sadržaj direktorijuma (imena fajlova u direktorijumu).

Slike (posmatranja) su u formi FIT(S) formata. Pored same slike, FIT(S) format sadrži zaglavlje (header³) u kome mogu biti smešteni podaci o posmatranju (datum, vreme, ekspoziција), objektu koji se posmatra (koordinate,...) kao i instrumentu i posmatračima,... Reč je o standardnom formatu u astrofizici.

Komandom `imhead *` (zvezdica podrazumeva da se radi o svim podacima u direktorijumu) biće ispisana kratka zaglavlja svih (.fit) fajlova u direktorijumu u kojem se nalazimo. Na osnovu podataka iz zaglavlja može da se zaključi o kom tipu podataka se radi.

U našem slučaju dobijamo:

```
imhead *
```

¹Image Reduction and Analysis Facility

² α (J2000) = 19h 29m 20.90s, δ (J2000) = +20° 16' 47".05, V = 6.33. Detalji se mogu naći na npr. SIMBAD-u.

³U ovoj skripti se koristi veliki broj Anglicizama bez posebnog truda da se prevodi na naš jezik.

```

sj090001.fit[2030,800][ushort]: zero
sj090002.fit[2030,800][ushort]: zero
sj090003.fit[2030,800][ushort]: zero
sj090004.fit[2030,800][ushort]: zero
sj090005.fit[2030,800][ushort]: zero
sj090006.fit[2030,800][ushort]: zero
sj090007.fit[2030,800][ushort]: zero
sj090008.fit[2030,800][ushort]: zero
sj090009.fit[2030,800][ushort]: zero
sj090010.fit[2030,800][ushort]: zero
sj090012.fit[2030,800][ushort]: 27 Vul
sj090013.fit[2030,800][ushort]: comp-BS
sj090015.fit[2030,800][ushort]: 7 Vul
sj090016.fit[2030,800][ushort]: comp-BS
sj090019.fit[2030,800][ushort]: flat-BS
sj090020.fit[2030,800][ushort]: flat-BS
sj090021.fit[2030,800][ushort]: flat-BS
sj090022.fit[2030,800][ushort]: flat-BS
sj090023.fit[2030,800][ushort]: flat-BS
sj090024.fit[2030,800][ushort]: flat-BS
sj090025.fit[2030,800][ushort]: flat-BS
sj090030.fit[2030,800][ushort]: 7 Vul
sj090031.fit[2030,800][ushort]: comp-BS

```

Kao što je rečeno, nas zanimaju posmatranja zvezde 7 Vul. Naravno, pored *sirovih* posmatranja spektra zvezde 7 Vul neophodne su nam i bias (zero) i flat-field (flat-BS) slike kao i posmatranja spektra komparacione lampe (comp-BS) koja nam koriste za kalibraciju talasnih dužina (vidi kasnije). Takođe, sniman je i spektar zvezde 27 Vul (i njoj asociranog spektra lampe) koji služi za telluric korekciju (vidi kasnije).

Primećujemo da postoje dva posmatranja zvezde 7 Vul (date posmatračke večeri). Posle svakog posmatranja stoje snimci komparacione lampe (koji su nam potrebni).

Ukoliko želimo da vidimo ceo header, pojedinačnih fajlova, onda koristimo sledeću komandu: `imhead ime_fajla l+`. Znači, npr. komandom `imhead sj*15.fit l+` možemo videti detalje (datum, vreme, ekspozicija,...) prvog posmatranja (*sirovog*) spektra zvezde 7 Vul (ovde prikazujemo samo početak hedera):

```

sj090015.fit[2030,800][ushort]: 7 Vul
No bad pixels, min=610., max=652.
Line storage mode, physdim [2030,800], length of user area 4131 s.u.
Created Wed 10:08:32 24-Aug-2011, Last modified Wed 10:08:32 24-Aug-2011
Pixel file "sj090015.fit" [ok]
BSCALE = 1 / REAL=TAPE*BSCALE+BZERO
BZERO = 32768 /
EXTEND = F / File may contain extensions
OBJECT = '7 Vul ' / Title of observation
ORIGIN = 'BIAS-ONDREJOV' / FITS file originator
DATE = '2009-10-09' / Date FITS file was generated
SYSVER = 'BIAS-headers' / Wed Feb 27 13:21:35 CET 2008

```

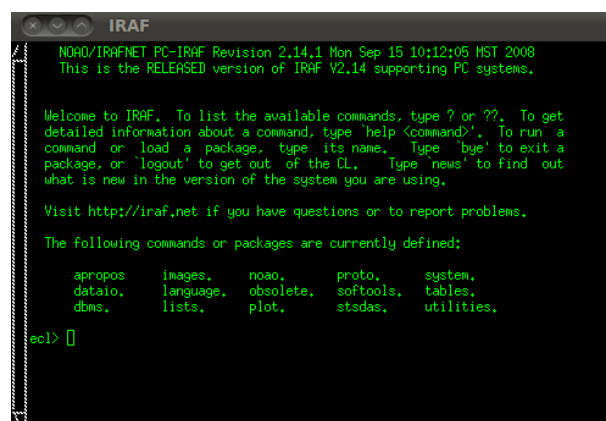
```

DATAMIN =                610 / DATA MIN
DATAMAX =                652 / DATA MAX
OBSERVAT= 'ONDREJOV'      / Name of observatory (IRAF style)
LATITUDE=         49.910555 / Telescope latitude (degrees), +49:54:38.0
LONGITUD=         14.783611 / Telescope longitud (degrees), +14:47:01.0
HEIGHT  =                0528 / Height above sea level [m].
OBSERVER= 'KubatRezbaSurlan' / Observers
IMAGETYP= 'object'        / Type of observation, eg. FLAT
FILENAME= 'sj090015.fit'   / BIAS filename
TELESCOP= 'ZEISS-2m'      / 2m Ondrejov observatory telescope
TELSYST = 'COUDE'         / Telescope setup - COUDE or CASSEgrain
EXPTIME =                1800.000 / Length of observation excluding pauses
DARKTIME=                1800.000 / Length of observation including pauses
EXPVAL  =                3.132 / Exposure value in photon counts [Mcounts]
DATE-OBS= '2009-10-09'    / UTC date start of observation
TM_START=                65604 / 18:13:24
TM_END  =                67433 / 18:43:53
COMMENT =                /
COMMENT1=                /
UT      = '18:13:24'      / UTC of start of observation

...

```

Uz pomoć **Shift + PgUp** (ili **PgDn**) možemo listati (skrolovati) tekst unutar konzole IRAF-a (vidi sliku 1). Za kopiranje i paste-ovanje potrebno je selektovati (mišem ili sa **Shift**) i zatim kliknuti srednjim klikom na mišu ili pritisnuti **Shift+Insert** na tastaturi. U IRAF konzoli kada se uđe u neku funkciju ne radi **Backspace** odnosno za brisanje slova se koristi **Del**. Ukoliko želimo da izbrišemo neki fajl to radimo uz pomoć **del ime_fajla**. IRAF poseduje odličan help - dovoljno je ukucati **help ime funkcije**.



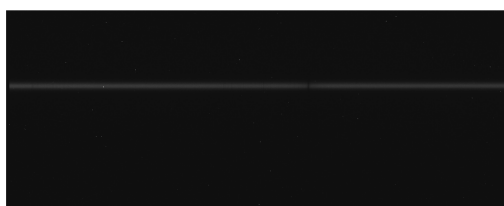
Slika 1: Primer IRAF konzole.

Sirovi (2D) spektar zvezde možemo videti uz pomoć ds9-a, npr.: **!ds9 s*15.fit**. Header-i se, takođe, mogu videti upotrebom ds9-a.

Sve standardne linux komande (npr.: **mkdir**, **rm**, ...) mogu se koristiti u IRAF-u ukoliko

se doda ! ispred (npr. !mkdir). To važi i za pozivanje programa koji su izvan IRAF-a (kao što je ds9).

Na slikama 2 i 3 prikazan je sirov 2D spektar prvog posmatranja (spektra) zvezde 7 Vul (*15.fit) i sirov spektar komparacione lampe koja odgovara prvom posmatranju zvezde 7 Vul (*16.fit). Na slikama 4 i 5 su, takođe dati 2D spektri zvezde i lampe ali podešeni (u ds9, scale > histogram) tako da se uočava kontaminacija različitim faktorima zbog kojih je redukcija neophodna. Na slikama 6 i 7 predstavljeni su jedan bias i jedan flat-field snimak.



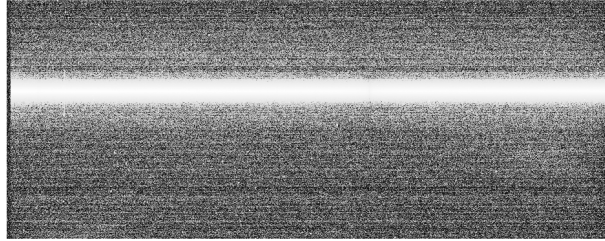
Slika 2: 2D sirov spektar zvezde (linearna raspodela boja).



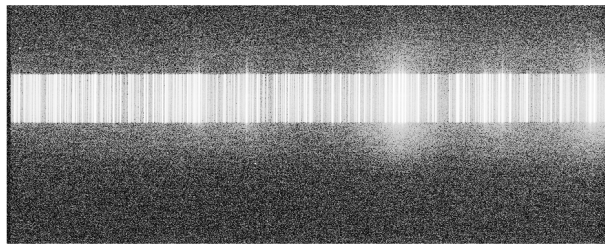
Slika 3: 2D spektar lampe (linearna raspodela boja).

Na ovom mestu možemo pogledati naše podatke uz pomoć komande `implot *`. Kada se koristi funkcija `implot`, bitno je zapamtiti da se iz grafičkog okruženja izlazi **ISKLUČIVO** pritiskom na **q** (u protivnom mora da se restartuje IRAF...) ! Isto važi za sve funkcije koje koriste grafičko okruženje.

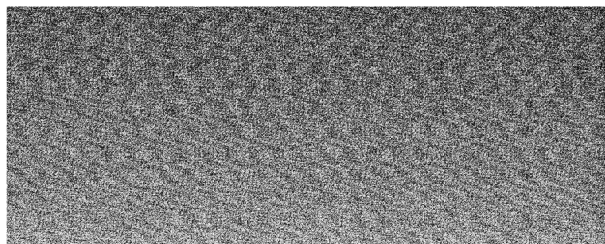
Svakom pikselu (čip posmatramo kao 2D matricu - vidi sliku 8) je asociran odgovarajući fluks, tako da u `implot`-u možemo prelaziti iz raspodele fluksa po kolonama (columns) u raspodelu fluksa po vrstama (linijama - lines) pritiskanjem **c** i **l**. Označavanjem proizvoljnog



Slika 4: 2D sirov spektar zvezde (histogram)

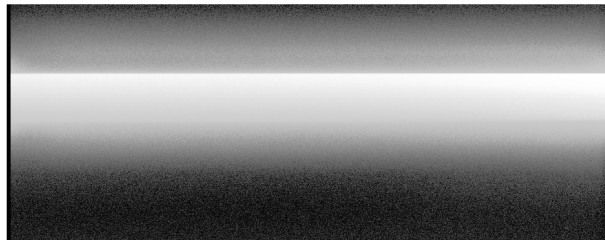


Slika 5: 2D spektar lampe (histogram)



Slika 6: Jedan bias snimak.

regiona na slici kursorom i pritiskanjem **e** pri svakom označavanju, može da se zumira slika. Komandom **r** (redraw) se posle zumiranja vraća na staro. Sa **n** se prelazi na sledeći fajl



Slika 7: Jedan flat-field snimak.

a sa **m** se vraća na stari. Bitno je naglasiti da se pomenute operacije razlikuju od jedne do druge IRAF funkcije (vidi kasnije).

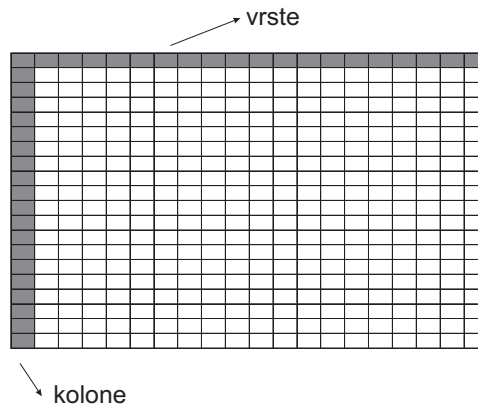
Korisno je primetiti da kada se nalazimo u raspodeli fluksa po kolonama sa leve strane su date vrednosti fluksa dok su sa desne strane oznacene vrste (linije) - tekuća linija je označena crticom (vidi sliku 9, crtica koja označava tekuću liniju je odmah ispod broja 400 sa desne strane). Kursorom možemo prelaziti sa jedne na drugu liniju i posmatrati raspodelu fluksa po kolonama za različite linije. Slično važi i za raspodelu po vrstama (vidi sliku 10). Sada su sa desne strane date kolone.

Da bi se primetilo da li ima nekih nepravilnosti na čipu, korisno je otvoriti sliku u ds9-u i izvršiti vizuelnu inspekciju (da li se vide *bele tačkice* odnosno *mrlje* ili čak linije (bad columns) - vidi sliku 2). Za lakše definisanje tačnih granica loših piksela može se predstaviti raspodela fluksa usrednjena po kolonama ili vrstama (linijama) - obično su loše kolone kako se po njima vrši čitanje CCD čipa (usrednjavamo po linijama da ne bi morala da se gleda svaka linija posebno - ako postoji neki pik on će biti jasno uočen na usrednjenoj slici). Dakle, pritisnemo **c** i dobijemo raspodelu fluksa po linijama (na desnoj strani je skala kolona). Zatim označimo početnu i krajnju liniju tako što postavimo kursor na željeno mesto i pritisnemo **a**. Sada smo dobili usrednjenu raspodelu fluksa po kolonama. Ukoliko se jave veliki pikovi znači da nešto nije u redu sa tim pikselom. Ako se isti artefakt na istom pikselu javlja na svim fajlovima onda je problem do čipa (taj piksel mora biti odstranjen iz analize - vidi kasnije) a ako se javlja samo u jednom fajlu onda je najverovatnije reč o kosmičkom zraku (vidi kasnije).

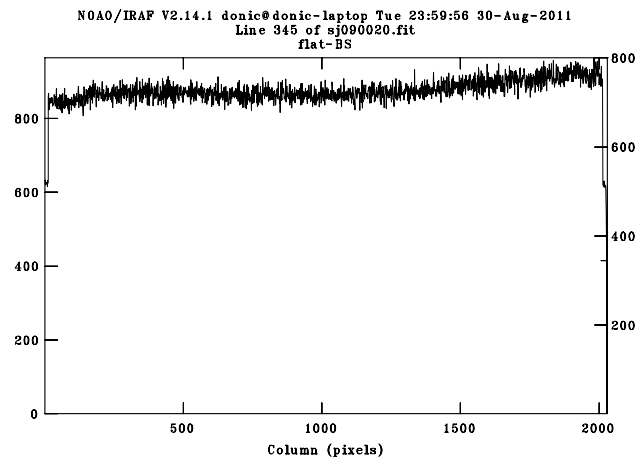
Ako želimo da sačuvamo sliku u .eps formatu to možete učiniti komandama `:.snap epsf1` ili `:.snap epsf` dok se nalazite npr. u implot-u ili u nekom drugom grafičkom okruženju (epsf1 za landscape, epsf za portrate).

2 Oduzimanje over-scan-a

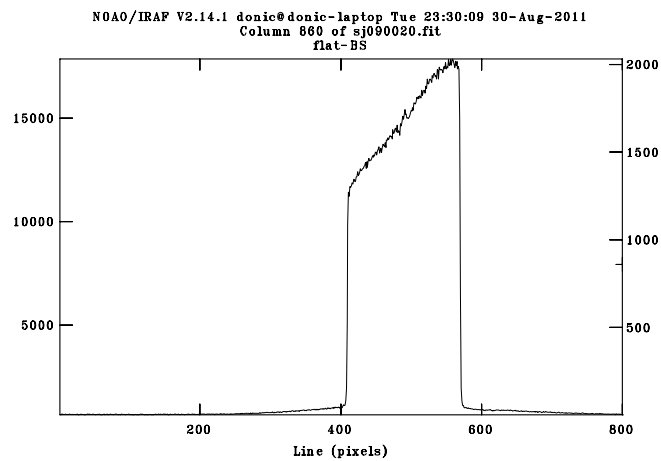
Već pri prvoj ispekciji slika (vidi sliku 9 i 10) uočavamo da postoji *lažna nula* (over-scan) u signalu, reda nekoliko stotina odbroja (u našim posmatranjima oko 600). Over-scan se najbolje uočava na flat-field slikama (flat-field je ravan i za razliku od biasa nije oko nule).



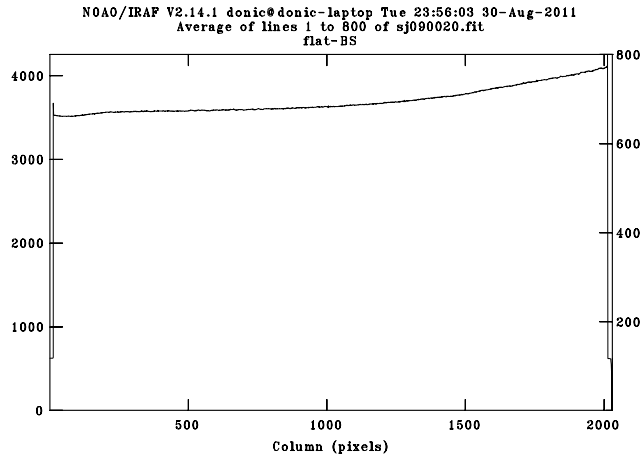
Slika 8: Skica CCD čipa - vrste i kolone.



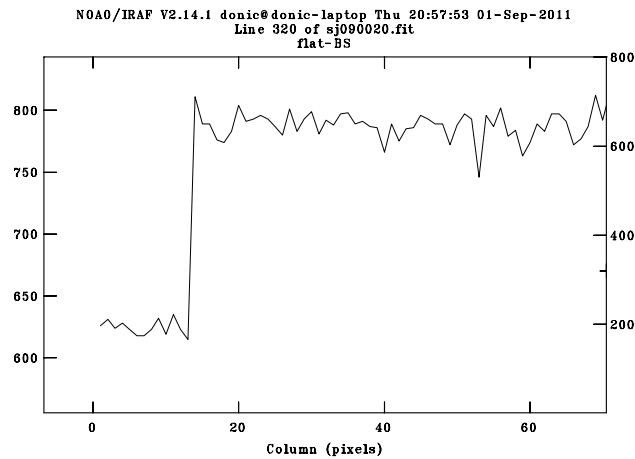
Slika 9: Raspodela fluksa po kolonama za jednu od flat-field slika.



Slika 10: Raspodela fluksa po vrstama (linijama) za jednu od flat-field slika.



Slika 11: Usrednjena raspodela fluksa po kolonama za jednu od flat-field slika (ne primećujemo izrazite pikove).

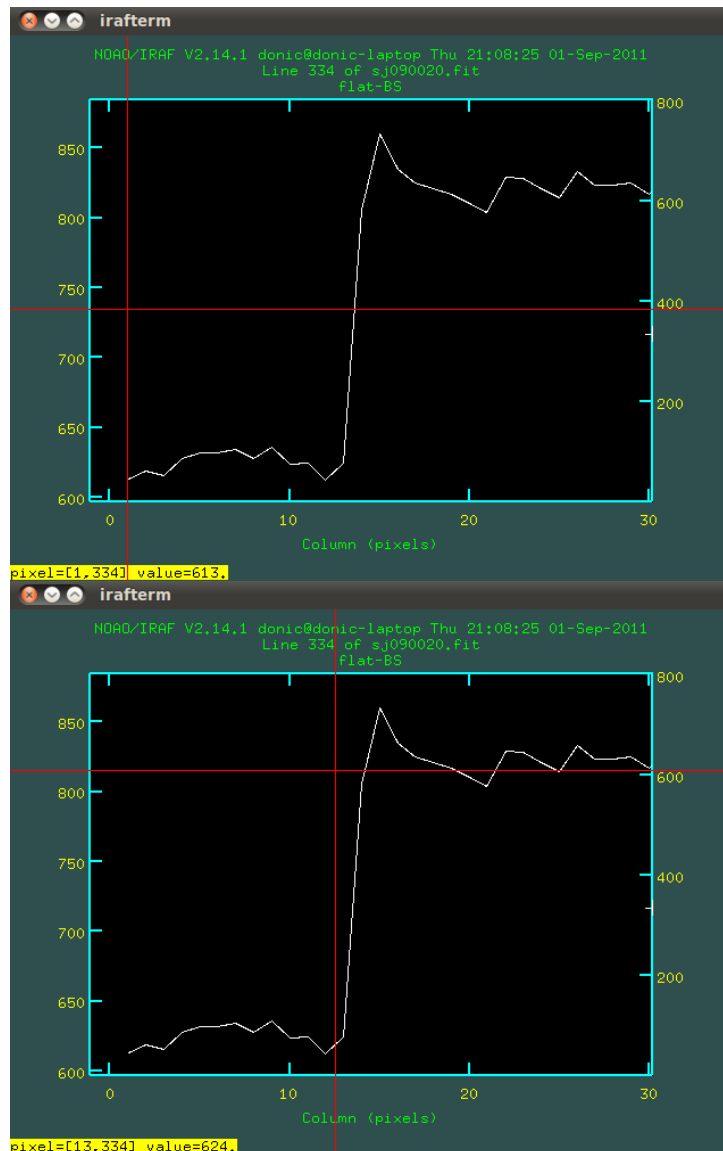


Slika 12: Zumiran region over-scena (jedna od flat-field slika).

Ukoliko implot-ujemo jedan flat-field fajl možemo utvrditi gde je over-scan region. Koristimo raspodelu fluksa po kolonama - kucamo **l**. Kursorom stajemo na granice over-scan-a (vidi slike 11-13) i pritiskamo **Space**. Pamtime vrednosti 1 i 13 (pixel = [1,...], pixel = [13,...]) što predstavlja granice over-scan-a u kolonama (jedna linija je fiksirana - u našem primeru, slika 13, linija 334).

Dovoljno je da se analizira jedan over-scan region ako ih ima više. Over-scan regioni se javljaju na svim snimcima (dovoljno je da se analizira jedna flat-field slika i da se zaključí o svim fajlovima). Ne zaboraviti, iz implot-a se izlazi sa **q** !

Potrebno je da se oduzme ovaj *lažni* nivo od svih snimaka. Radi lakše obrade koristićemo liste (da ne bisno radili fajl po fajl). Prvo napravimo listu imena svih fajlova (datoteka koja sadrži imena svih fajlova): **ls s* > list_all** i nazovimo je **list_all**. **s*** znači da nas zanimaju sva imena koja počinju sa **s** (to smo zaključili iz početne inspekcije). Ukoliko ne bismo stavili **s** onda bi nam se u listi nalazilo i ime novo stvorene



Slika 13: Zumiran region over-scena (jedna od flat-field slika).

liste (`list_all`).

Napravimo sada listu imena fajlova koji su korigovani za ovaj lažni nivo (over-scan). Zamenjujemo nazive u listi da ne bi došlo do prepisivanja starih fajlova koje želimo da sačuvamo: `cp list_all list_osc`. Editovanje lista se može raditi uz pomoć komande `vi`: `vi list_osc`. Važno je biti veoma oprezan pri radu sa listama ! Pritiskom na **Shift** + **i** pozicioniramo se na početak linije gde se kursor nalazi i zatim ukucamo npr. `osc_` ispred ranijeg imena da bismo sugerisali da je reč o fajlu korigovanom za over-scan. Zatim pritisnemo **Esc** i onda se spustimo na sledeće ime (strelica dole) i pritisnemo `.`, itd. `.` je komanda za ponavljanje prethodne komande. Ukoliko želimo da sačuvamo listu pritisnemo `:wq` a ako ne želimo da je sačuvamo pritisnemo `:q!`. Inače, iz liste se izlazi sa `:q`. Pored `vi`-a može se koristiti i `!gedit list_osc`.

Sada prelazimo na postupak oduzimanja lažne nule. Over-scan region fitujemo kon-

stantnom funkcijom uz pomoć procedure `fi1d`. Možemo ukucati `fit1d` i onda unositi podatke interaktivno ili ukucati: `epar fit1d`. Bitno je da se unesu korektni podaci (ne zaboraviti da se ispred liste mora pisati `@`) ! Ranije smo zapamtili over-scan region [1:13] koji prepisujemo. Treći parametar (`axis=1`) znači da se vrste fituju za razliku od kolona. Izlazna slika je razlika (`type=difference`) svake, individualne slike i linearnog fita. Biramo Ležandrove polinome reda 1 (u IRAF-u je tako numerisano - zapravo je reč o ravnoj liniji - linearni fit).

```
fit1d
```

```

                                I R A F
                        Image Reduction and Analysis Facility

PACKAGE = imfit
      TASK = fit1d

input    =          @list_all  Images to be fit
output   =          @list_osc  Output images
(axis    =              1) Axis to be fit
type     =          difference Type of output (fit, difference, ratio)
(interac=              no) Set fitting parameters interactively?
(sample  =          1:13) Sample points to use in fit
(naverag=              1) Number of points in sample averaging
(funcutio=          legendre) Fitting function
(order   =              1) Order of fitting function
(low_rej=              3.) Low rejection in sigma of fit
(high_re=              3.) High rejection in sigma of fit
(niterat=              1) Number of rejection iterations
(grow    =              1.) Rejection growing radius in pixels
(graphic=          stdgraph) Graphics output device
(cursor  =              ) Graphics cursor input
(mode    =              ql)
```

Kada sve ukucamo napišemo: `:go` i zatim pritisnemo **Enter**.

Proveravamo dobijene slike (korigovane za over-scan region): `implot osc*`. Sa slike 14 jasno uočavamo da je signal sada sveden na pravu nulu.

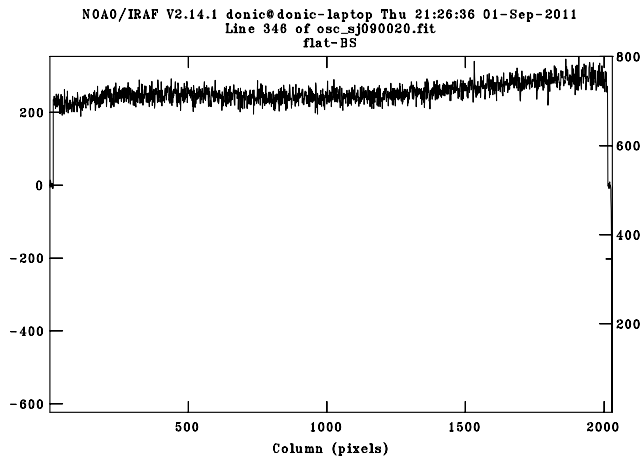
3 Kombinovanje i oduzimanje biasa

Bias slike se prave radi otklananja bilo kakve unutrašnje nepravilnosti na čipu. Reč je o snimcima nulte ekspozicije. Obično se snima desetak biasa koji se zatim kombinuju u jedan.

Napravimo sada dve liste: listu imena svih bias fajlova i listu imena svih onih fajlova koji nisu biasi:

```

cp list_osc list_bi
vi list_bi
cp list_osc list_others
```



Slika 14: Raspodela fluksa po kolonama za jednu over-scan korigovanu flat-field sliku.

```
vi list_others
```

Sa **dd** se briše red u listi a sa **:wq** se čuva lista.

Naravno, od sada koristimo samo fajlove koji su overscan korigovani. Da bismo znali koji su fajlovi bias (prvih deset) a koji nisu koristimo: **imhead osc***.

Sada kombinujemo sve biase u jedan master bias uz pomoć funkcije **imcombine**. Kucamo: **epar imcombine**. Nalazimo srednju vrednost (**combine=average**) bias slika. Takođe, odbacujemo sve van 3 sigma (standardne devijacije) od srednje vrednosti što omogućava da uklonimo kosmičke zrake (**lsigma=3, hsigma=3**). Pored toga, dobro je staviti **reject=avsigclip**, jer se u tom slučaju najbolje čisti slika. Izlazni fajl je master bias koji nazivamo **bias.fit**.

```
imcombine
```

I R A F

Image Reduction and Analysis Facility

```
PACKAGE = immatch
```

```
TASK = imcombine
```

```
input      =      @list_bi  List of images to combine
output     =      bias.fit  List of output images
(headers=  )      ) List of header files (optional)
(bpmasks= )      ) List of bad pixel masks (optional)
(rejmask=  )      ) List of rejection masks (optional)
(nrejmas= )      ) List of number rejected masks (optional)
(expmask=  )      ) List of exposure masks (optional)
(sigmas =  )      ) List of sigma images (optional)
(imcmb =   $I)      Keyword for IMCMB keywords
(logfile=  STDOUT) Log file

(combine=   average) Type of combine operation
(reject =   avsigclip) Type of rejection
```

```

(project=          no) Project highest dimension of input images?
(outtype=         real) Output image pixel datatype
(outlimi=         ) Output limits (x1 x2 y1 y2 ...)
(offsets=         none) Input image offsets
(masktyp=         none) Mask type
(maskval=         0) Mask value
(blank  =         0.) Value if there are no pixels

(scale  =         none) Image scaling
(zero   =         none) Image zero point offset
(weight =         none) Image weights
(statsec=         ) Image section for computing statistics
(expname=        ) Image header exposure time keyword

(lthresh=        INDEF) Lower threshold
(hthresh=        INDEF) Upper threshold
(nlow   =         1) minmax: Number of low pixels to reject
(nhigh  =         1) minmax: Number of high pixels to reject
(nkeep  =         1) Minimum to keep (pos) or maximum to reject (neg)
(mclip  =         yes) Use median in sigma clipping algorithms?
(lsigma =         3.) Lower sigma clipping factor
(hsigma =         3.) Upper sigma clipping factor
(rdnoise=         0.) ccdclip: CCD readout noise (electrons)
(gain   =         1.) ccdclip: CCD gain (electrons/DN)
(snoise =         0.) ccdclip: Sensitivity noise (fraction)
(sigscal=        0.1) Tolerance for sigma clipping scaling corrections
(pclip  =        -0.5) pclip: Percentile clipping parameter
(grow   =         0.) Radius (pixels) for neighbor rejection
(mode   =         ql)

```

Zatim kucamo: :go i pritiskamo **Enter**.

Dobijamo sledeći output:

Aug 24 11:51: IMCOMBINE

```

combine = average, scale = none, zero = none, weight = none
reject = avsigclip, mclip = yes, nkeep = 1
lsigma = 3., hsigma = 3.
blank = 0.

```

Images

```

osc_sj090001.fit
osc_sj090002.fit
osc_sj090003.fit
osc_sj090004.fit
osc_sj090005.fit
osc_sj090006.fit
osc_sj090007.fit
osc_sj090008.fit

```

```
osc_sj090009.fit
osc_sj090010.fit
```

```
Output image = bias.fit, ncombine = 10
```

Sada pravimo novu listu u kojoj će biti smeštena imena novih, bias korigovanih fajlova (dakle, svi osim biasa): `cp list_others list_others_bicorr, vi list_others_bicorr`. Slično kao i ranije: sa **Shift + i** se pozicionira na početak reda. Menjamo imena dodajući `bicorr_`. Pritiskamo **Esc** i pomeramo se strelicom dole i pritiskamo `.` i na kraju čuvamo listu sa `:wq`.

Sada je potrebno od svih fajlova oduzeti master bias `bias.fit`. Koristimo funkciju `imarith` odnosno kucamo: `epar imarith`. Komanda `epar` (edit parameter) znači da želimo da menjamo parametre funkcije. Ulazni podaci su lista over-scan korigovanih fajlova koji nisu biasi, znak `-` koji govori o tome da oduzimamo a treći parametar je ime master bias-a. Četvrti parametar je lista izlaznih fajlova.

```
epar imarith
```

```

                                I R A F
                        Image Reduction and Analysis Facility

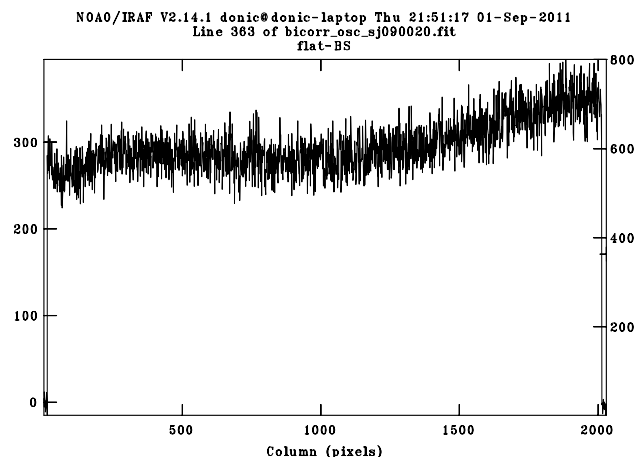
PACKAGE = imutil
TASK = imarith

operand1=      @list_others  Operand image or numerical constant
op           =                      - Operator
operand2=      bias.fit      Operand image or numerical constant
result  = @list_others_bicorr Resultant image
(title  =                      ) Title for resultant image
(divzero=      0.) Replacement value for division by zero
(hparams=      ) List of header parameters
(pixtype=      ) Pixel type for resultant image
(calctype=     ) Calculation data type
(verbose=      no) Print operations?
(noact  =      no) Print operations without performing them?
(mode   =      ql)
```

Sada su napravljeni over-scan i bias korigovani fajlovi čija imena su data u `list_others_bicorr`. Na slici 15 prikazana je jedna flat-field slika koja je bias korigovana.

4 Trimovanje

Sada ćemo odseći loše delove (gde nisu detektovani fotoni) odnosno sačuvati regione koje želimo da analiziramo. Te loše delove smo primetili i ranije (vidi npr. slike 9 i 10 - jasno se uočavaju (ne)korisne oblasti) a sada ih uklanjamo. Plot-ujemo bias korigovane slike sa `implot bicorr*`. Dovoljno je uočiti za jedan fajl (flat-field najlakše) - kod ostalih je sve isto.



Slika 15: Raspodela fluksa po kolonama za jednu bias i over-scan korigovanu flat-field sliku.

Zumiramo željene oblasti sa **e** (sa **r** se vraća na staro). Čitamo granice *dobrih* regiona (vidi slike 16 i 17) i sa kolona i sa vrsta - npr. prvo **l** pa **c**. Pritiska se **Space** i očitavaju vrednosti. Dobijamo da je koristan region približno: 15:2012 za kolone i 411:567 za vrste (linije).

Sada pravimo listu novih imena fajlova koji su (će biti) trimovani (menjamo nazive - dodajemo **trim_**):

```
cp list_others_bicorr list_trim
vi list_trim
```

Takođe, dodajemo u listu imena svih bias korigovanih slika podatke o regionu koji želimo da sačuvamo: **vi list_others_bicorr**. **Shift + a** nas pozicionira na kraj reda gde se nalazi kursor u listi. Dodajemo: [15:2012,411:567] pa pritiskamo **Esc** pa, kao i ranije, strelica dole i **.** itd., i na kraju **:wq**. Slova u listi se brišu sa **x**.

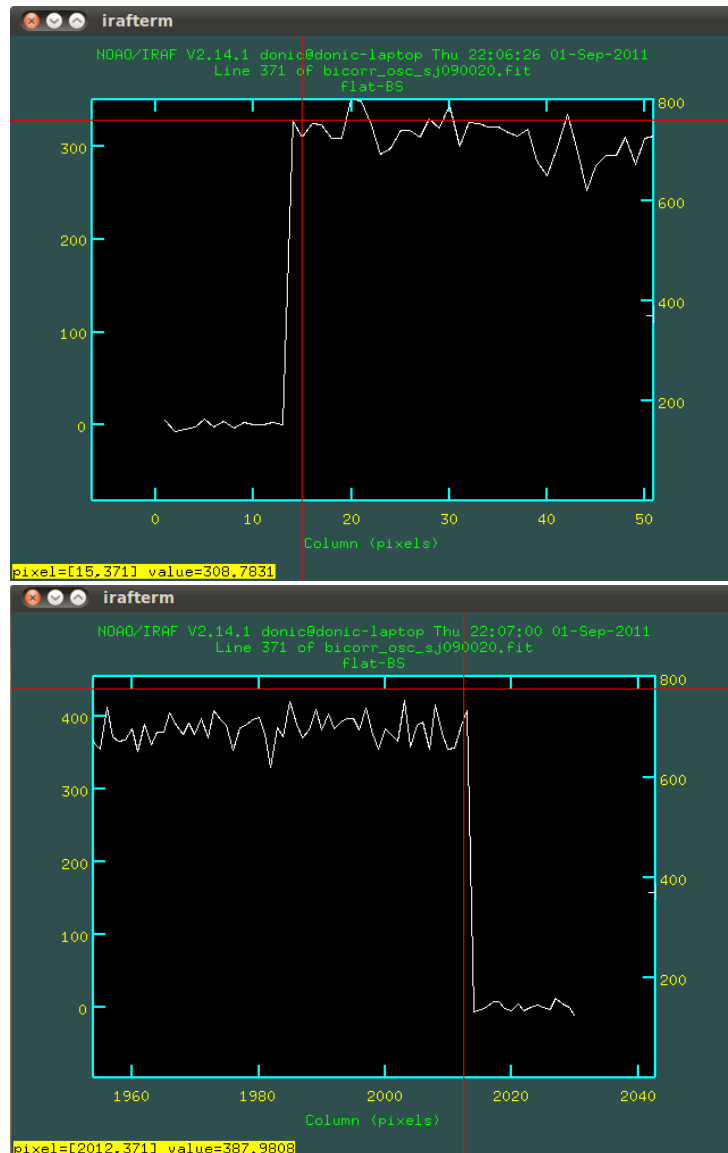
Sada trimujemo sa: **imcopy** (praktično kopiramo stare u nove ali kod starih je definisana oblast koja se kopira!).

```
imcopy
```

```
Input images: @list_others_bicorr
```

```
Output images or directory: @list_trim
```

```
bicorr_osc_sj090012.fit[15:2012,412:567] -> trim_bicorr_osc_sj090012.fit
bicorr_osc_sj090013.fit[15:2012,412:567] -> trim_bicorr_osc_sj090013.fit
bicorr_osc_sj090015.fit[15:2012,412:567] -> trim_bicorr_osc_sj090015.fit
bicorr_osc_sj090016.fit[15:2012,412:567] -> trim_bicorr_osc_sj090016.fit
bicorr_osc_sj090019.fit[15:2012,412:567] -> trim_bicorr_osc_sj090019.fit
bicorr_osc_sj090020.fit[15:2012,412:567] -> trim_bicorr_osc_sj090020.fit
bicorr_osc_sj090021.fit[15:2012,412:567] -> trim_bicorr_osc_sj090021.fit
bicorr_osc_sj090022.fit[15:2012,412:567] -> trim_bicorr_osc_sj090022.fit
bicorr_osc_sj090023.fit[15:2012,412:567] -> trim_bicorr_osc_sj090023.fit
bicorr_osc_sj090024.fit[15:2012,412:567] -> trim_bicorr_osc_sj090024.fit
```



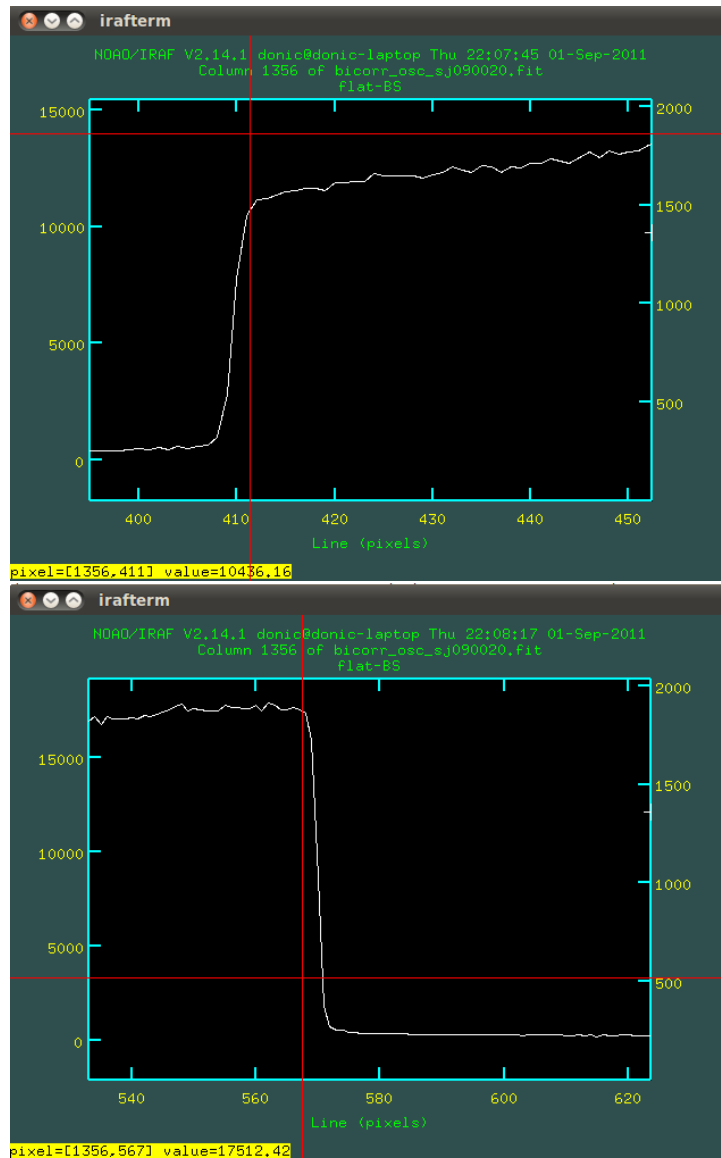
Slika 16: Trimovanje (primer jedne flat-field slike - kolone).

```
bicorr_osc_sj090025.fit[15:2012,412:567] -> trim_bicorr_osc_sj090025.fit
bicorr_osc_sj090030.fit[15:2012,412:567] -> trim_bicorr_osc_sj090030.fit
bicorr_osc_sj090031.fit[15:2012,412:567] -> trim_bicorr_osc_sj090031.fit
```

Plotujemo dobijene slike: `implot trim*`. Na slici 18 je predstavljen jedan trimovani flat-field (uporedi sa slikom 10).

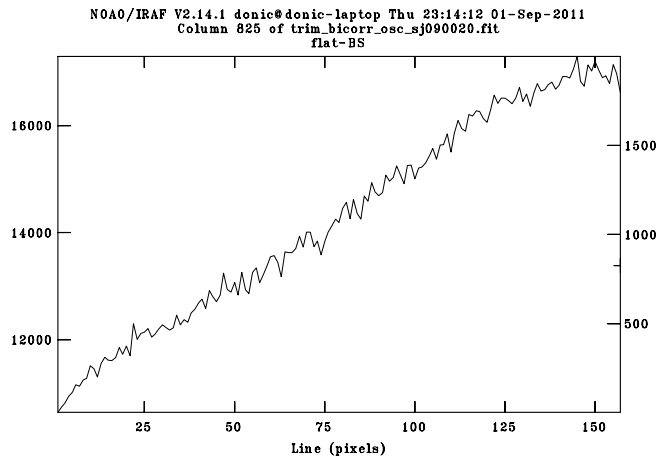
Dodatno odstranjivanje kosmičkih zraka može da se uradi i ulazeći u pakete `imred` / `crutil` i zatim upotrebom funkcije `cosmicrays`. Druga opcija je da se napravi fajl (`vi <ime>`) i unesu odgovarajuće vrednosti loših piksela, npr.:

```
235 237 135 1080
456 567 789 2048
```



Slika 17: Trimovanje (primer jedne flat-field slike - linije).

odnosno, za svaki pik na slici gleda se od koje do koje kolone se prostire i to se zapisuje prvo (u prvom redu od 235 do 237), onda se stane na taj pik i klikne **c** da bi se sada videlo od koje do koje linije se prostire (u ovom primeru od 135 do 1080). Ako ima više pikova, za svaki se pišu odgovarajuće vrednosti u novom redu. Kada napravimo taj fajl pozivamo funkciju **fixpix** koja se nalazi u paketu **oned** (dobro je kopirati ulazne fajlove kako nema opcije za nove izlazne fajlove već se izmene čuvaju na postojećim). U principu, kao treća opcija, moguće je ukloniti lažne pikove i nakon *otpakivanja* spektra pri plotovanju funkcijom **splot** (vidi kasnije). Ipak, uglavnom, pri otpakivanju spektra, funkcija **apa11** odradi veliki deo posla (vidi kasnije). U principu, imcombine za bias i flat-field može da uradi dosta dobar posao. Sve zavisi od konkretnog posmatranja.



Slika 18: Jedan trimovani flat-field (raspodela fluksa po linijama).

5 Flat-field korekcija

Flat-field slike se prave kako bi se utvrdila (uklonila) varijacija u osetljivosti na CCD čipu. Idealni flat-field je slika osvetljena konstantnim izvorom svetlosti koji se ne menja sa talasnom dužinom i koji ne manifestuje spektralne linije (kvarcne lampe, nebo u sumrak).

Kao i u slučaju biasa uzima se oko 5-10 (obično 5) flat-field snimaka koji se kombinuju u master flat-field.

Procedura je slična kao kod bias-a. Potrebno je da se napravi lista imena flat-field fajlova kao i lista imena svih ostalih fajlova (trimvanih i bias korigovanih).

```
cp list_trim flats
cp list_trim others
vi flats
vi others
```

Ovde opet može da posluži `imhead trim*` da bismo znali šta je flat-field a šta nije.

Sada se pravi master flat-field pomoću komande `imcombine`. Input je isti kao i kod bias-a osim što definišemo master flat kao `flat.fit` i unosimo listu flat-field-ova.

```
epar imcombine
...
```

```
Sep  2 12:25: IMCOMBINE
combine = average, scale = none, zero = none, weight = none
reject = avsigclip, mclip = yes, nkeep = 1
lsigma = 3., hsigma = 3.
blank = 0.
```

```
Images
trim_bicorr_osc_sj090019.fit
trim_bicorr_osc_sj090020.fit
trim_bicorr_osc_sj090021.fit
trim_bicorr_osc_sj090022.fit
```

```
trim_bicorr_osc_sj090023.fit
trim_bicorr_osc_sj090024.fit
trim_bicorr_osc_sj090025.fit
```

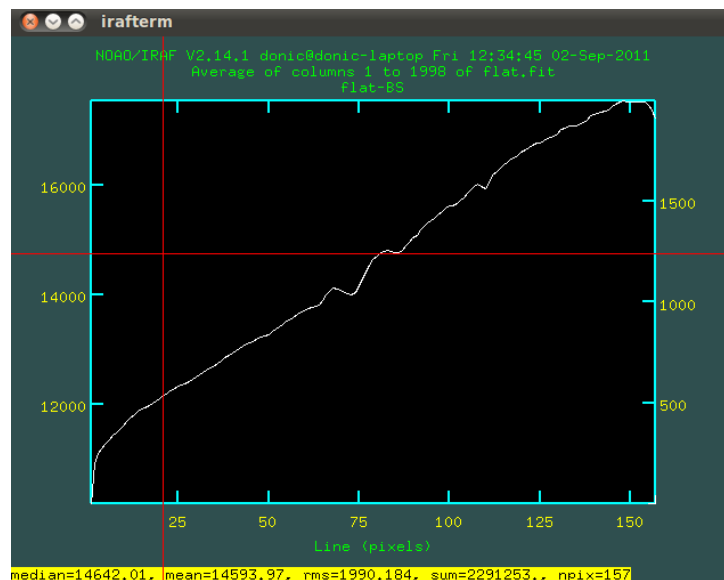
```
Output image = flat.fit, ncombine = 7
```

U spektroskopiji flat-field-ovi nisu potpuno ravni (različita je osetljivost u različitim regionima talasnih dužina kao i osetljivost različitih delova čipa). Pre same flat-field korekcije potrebno je normalizovati master flat-field (on samo predstavlja funkciju osetljivosti samog čipa - karakteristika čipa).

Prvi način (koji koristimo u vežbi) jeste deljenje master flat-field-a sa konstantnom vrednošću. Otvorimo master flat-field i uđemo u raspodelu po kolonama (1). Sa leve i desne strane kucamo po **a** (usrednjimo po linijama) i zatim isto tako kucamo **s** (sa leve i desne strane - **s** znači da tražimo statistiku). Očitamo vrednost mean=14593.97 (vidi sliku 19). Master flat-field, sada, normalizujemo deljenjem sa 14594 uz pomoć komande **imarith**. Statistiku slike možemo dobiti i korišćenjem funkcije **imstat**. Normalizovani master flat-field nazivamo **flat_norm.fit** (zapravo, nije toliko bitno da se nađe mean, može i da se otprilike vidi sa slike, bitno je da flat-field bude oko 1).

```
imarith
```

```
Operand image or numerical constant (): flat.fit
Operator (+|-|*|/|min|max) (): /
Operand image or numerical constant (): 14594.0
Resultant image (flat_norm.fit): flat_norm.fit
```



Slika 19: Nalaženje konstante za normalizaciju master flat-field-a.

Druga opcija je da se master flat-field fituje polinomom niskog reda u smislu normalizacije. U tu svrhu može se koristiti funkcija **fit1d**. Input je master flat-field i izlazna, normalizovana,

slika (type=ratio, graphic=stdgraph). Proces je u principu iterativan. Red polinoma se može menjati kucanjem npr. `:order 3` (u okviru plotu). Tip polinoma se može menjati kucanjem npr. `:f spline1` (legendre, spline1, spline3, chebyshev). Promene u fitu se prepoznaju nakon kucanja `f`.

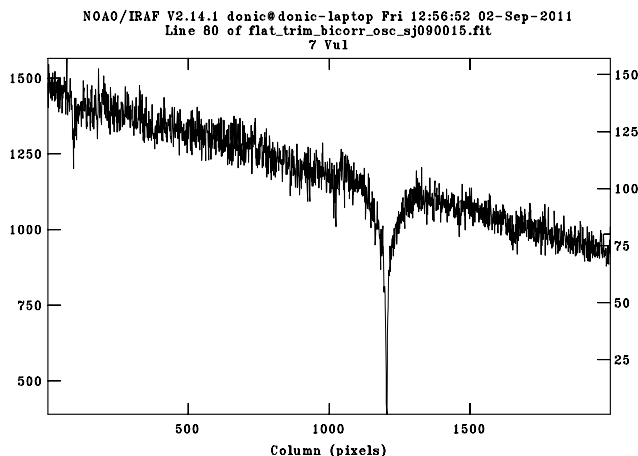
Ukoliko postoje varijacije koje su rezultat različite osetljivosti CCD čipa a ne usled toga da je flat-field lampa manje sjajna u toj oblasti, može doći do grešaka.

Sada možemo podeliti preostale fajlove sa normalizovanim master flat-fieldom, opet, uz pomoć funkcije `imarith`. Pre toga je neophodno napraviti listu imena fajlova u kojima se nalaze spektri koji su over-scan, bias, trim i flat-field korigovani: `flatcorr_spectra` (cp others flatcorr_spectra, vi flatcorr_spectra, dodajemo flat_).

```
imarith
```

```
Operand image or numerical constant (): @others
Operator (+|-|*|/|min|max) (): /
Operand image or numerical constant (): flat_norm.fit
Resultant image (): @flatcorr_spectra
```

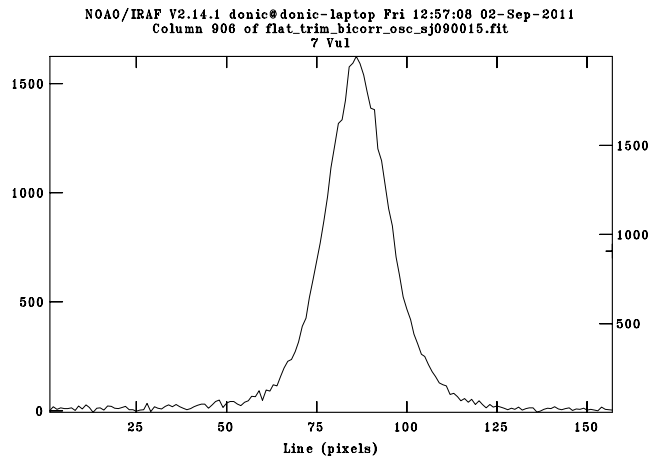
Dobijene slike plotujemo sa `implot flat_trim_*` (vidi slike 20 i 21). Iako još uvek nije izvršeno otpakivanje spektra i kalibracija talasnih dužina, kako znamo u kom opsegu talasnih dužina smo snimali spektar zvezde, zaključujemo, posmatrajući sliku 16 da je jasno uočljiva $H\alpha$ linija.



Slika 20: Neoptpakovani, flat-field korigovani spektar zvezde 7Vul (raspodela fluksa po kolonama).

6 Otpakivanje spektra

Nas zanima da dobijemo raspodelu fluksa po talasnim dužinama. Sada želimo da otpakujemo jednodimenzionalni spektar. U tom smislu moramo definisati lokaciju spektra na 2D slici (*otisak spektra* na 2D matrici).



Slika 21: Neoptakovani, flat-field korigovani spektar zvezde 7Vul (raspodela fluksa po linijama).

U svrhu otpakivanja spektra prvo ćemo napraviti četiri liste: listu imena flat-field korigovanih spektara, listu imena (budućih) otpakovanih spektara (`ex_`), listu imena flat-field korigovanih spektara komparacione lampe i listu imena (budućih) otpakovanih spektara komparacione lampe (`ex_`).

```
cp flatcorr_spectra star
cp flatcorr_spectra extracted_star
cp flatcorr_spectra lamp
cp flatcorr_spectra extracted_lamp
vi star
vi extracted_star
vi lamp
vi extracted_lamp
```

Sada ulazimo u paket `twod` pa unutar njega u paket `apextract` (iz header-a `.fit` fajla nalazimo da je disperziona osa duž vrsta (linija) - što se može uočiti i vizuelnom inspekcijom!). U našem slučaju setujemo `dispaxi=1`.

```
epar apextract
```

I R A F

Image Reduction and Analysis Facility

PACKAGE = twodspec

TASK = apextract

```
(dispaxi=          1) Dispersion axis (1=along lines, 2=along columns)
(databas=        database) Database
(verbose=        no) Verbose output?
(logfile=         ) Text log file
(plotfil=         ) Plot file
```

```
(version= APEXTRACT V3.0: August 1990)
(mode    =                               ql)
($nargs  =                               0)
```

:go i **Enter**

Kao i ranije kucanjem **epar** ispred naziva komande mi zapravo setujemo parametre te funkcije. Nije neophodno u startu poznavati sve parametre kako je reč o interaktivnoj funkciji!

Ne zaboravimo, takođe, da je spektar ranije trim-ovan (kolone [15:2012], linije [412:567]). Broj kolona i vrsta je smanjen (broj kolona je sada 1997).

Dakle, na redu je fitovanje *otiska* spektra (spectral trace) - odnosno tražimo gde se nalazi spektar na 2D matrici (slici).

Sada izvršavamo komandu **apall** (ne zaboravi da pišeš @ pre imena liste) koja *obavlja* otpakivanje spektra. Po default-u **apall** uzima centralnu kolonu i ± 5 oko centra (kasnije sami definišemo aperturu ako nismo zadovoljni kako je IRAF uradio posao). Ulazimo u interaktivnu proceduru:

```
epar apall
```

I R A F

Image Reduction and Analysis Facility

```
PACKAGE = apextract
```

```
TASK = apall
```

```
input      =                @star List of input images
(output    =                @extracted_star) List of output spectra
(apertur=   1) Apertures
(format    =                multispec) Extracted spectra format
(referen=   ) List of aperture reference images
(profile=   ) List of aperture profile images

(interac=   yes) Run task interactively?
(find      =   yes) Find apertures?
(recente=   yes) Recenter apertures?
(resize    =   yes) Resize apertures?
(edit      =   yes) Edit apertures?
(trace     =   yes) Trace apertures?
(fittrac=   yes) Fit the traced points interactively?
(extract=   yes) Extract spectra?
(extras    =   yes) Extract sky, sigma, etc.?
(review    =   yes) Review extractions?

(line      =                INDEF) Dispersion line
(nsum      =                10) Number of dispersion lines to sum or median
```

```
# DEFAULT APERTURE PARAMETERS
```

```

(lower =          -5.) Lower aperture limit relative to center
(upper =          5.) Upper aperture limit relative to center
(apidtab=        ) Aperture ID table (optional)

```

DEFAULT BACKGROUND PARAMETERS

```

(b_funct=        legendre) Background function
(b_order=         3) Background function order
(b_sampl=       -10:-6,6:10) Background sample regions
(b_naver=         1) Background average or median
(b_niter=         3) Background rejection iterations
(b_low_r=        3.) Background lower rejection sigma
(b_high_r=       3.) Background upper rejection sigma
(b_grow =        1.) Background rejection growing radius

```

APERTURE CENTERING PARAMETERS

```

(width  =         5.) Profile centering width
(radius =        10.) Profile centering radius
(thresho=        0.) Detection threshold for profile centering

```

AUTOMATIC FINDING AND ORDERING PARAMETERS

```

nfind  =          Number of apertures to be found automatically
(minsep =          5.) Minimum separation between spectra
(maxsep =       1000.) Maximum separation between spectra
(order  =    increasing) Order of apertures

```

RECENTERING PARAMETERS

```

(aprecen=        ) Apertures for recentering calculation
(npeaks =    INDEF) Select brightest peaks
(shift  =        yes) Use average shift instead of recentering?

```

RESIZING PARAMETERS

```

(llimit =    INDEF) Lower aperture limit relative to center
(ulimit =    INDEF) Upper aperture limit relative to center
(ylevel =    0.1) Fraction of peak or intensity for automatic width
(peak   =    yes) Is ylevel a fraction of the peak?
(bkg    =    yes) Subtract background in automatic width?
(r_grow =    0.) Grow limits by this factor
(avglimi=    no) Average limits over all apertures?

```

TRACING PARAMETERS

```

(t_nsum =    10) Number of dispersion lines to sum
(t_step =    10) Tracing step

```

```

(t_nlost=          3) Number of consecutive times profile is lost befo
(t_funct=          legendre) Trace fitting function
(t_order=          2) Trace fitting function order
(t_sampl=          *) Trace sample regions
(t_naver=          1) Trace average or median
(t_niter=          0) Trace rejection iterations
(t_low_r=          3.) Trace lower rejection sigma
(t_high_=          3.) Trace upper rejection sigma
(t_grow =          0.) Trace rejection growing radius

```

EXTRACTION PARAMETERS

```

(backgro=          fit) Background to subtract
(skybox =          1) Box car smoothing length for sky
(weights=          none) Extraction weights (none|variance)
(pfit   =          fit1d) Profile fitting type (fit1d|fit2d)
(clean  =          yes) Detect and replace bad pixels?
(saturat=          INDEF) Saturation level
(readnoi=          0.) Read out noise sigma (photons)
(gain   =          1.) Photon gain (photons/data number)
(lsigma =          4.) Lower rejection threshold
(usigma =          4.) Upper rejection threshold
(nsubaps=          1) Number of subapertures per aperture
(mode   =          ql)

```

:go i **Enter**

Na postavljena pitanja koja daje **apall** odgovaramo potvrdno i unosimo 1 za broj apertura. Otvara se grafik koji omogućava da sami definišemo aperturu (slika 23). Pozicioniranjem na aperturu i pritiskom na **d** briše se apertura (koja je data po default-u, nije uvek najbolja). Pozicioniranje na željeni centar aperture i pritiskom **n** definiše se nova apertura pri čemu se granice definišu pozicioniranjem kursora i pritiskom **l** (lower) za levu i **u** (upper) za desnu granicu, respektivno. Ako smo zadovoljni sa definicijom aperture (ovde imamo samo jednu aperturu, vidi primer rada sa echelle-om za komplikovaniji slučaj) prelazimo na definiciju pozadine (koju želimo oduzeti) pritiskom **b** (slika 24). Sada, sa **z** brišemo regione po default-u definisane od strane IRAF-a a sa pozicioniranjem kursora na levu i desnu granicu i svaki put pritiskanjem **s** definiše se nova pozadina (možemo menjati red i tip funkcije). Ako se desi problem sa grafikom može se pritisnuti **r** (redraw). Sada pritiskamo **f** odnosno fitujemo pozadinu. Zatim pritisnemo **q** dva puta i potvrdno odgovorimo na sva pitanja (za izlaz iz background prozora i izlaz iz definisanja aperture).

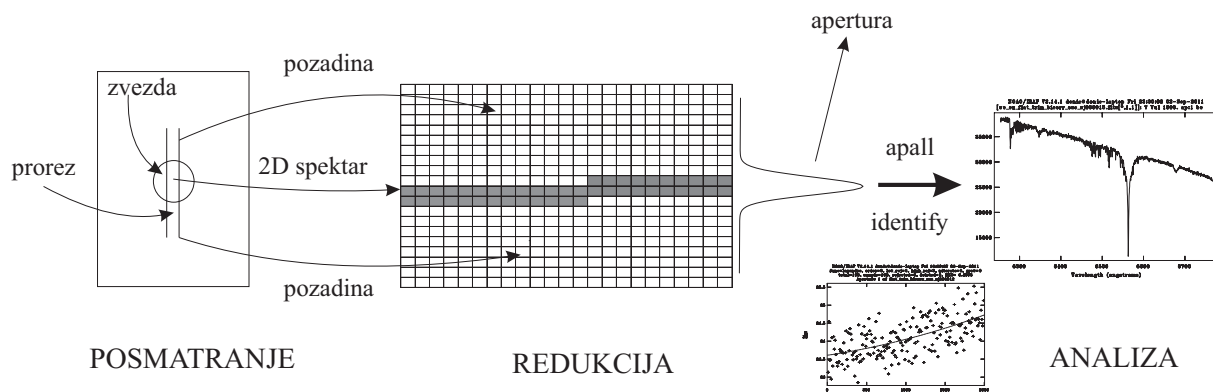
Na redu je fitovanje tačaka koje predstavljaju *otisak* spektra npr. Ležandrovim polinomima. Otvara se novi grafik linije vs. kolone (vidi slike 25 i 26). Ako želimo da promenimo red polinoma kojim se fituje ukucamo npr. :o 3 i pritisnemo **Enter** i zatim **f**. Naravno, možemo promeniti i samu funkciju sa :f ime funkcije. Korisno je koristiti manji red polnoma ali naravno paziti da li dobro fituje.

Ukoliko smo zadovoljni fitom pritisnemo **q**, odgovorimo pozitivno na pitanja i zatim

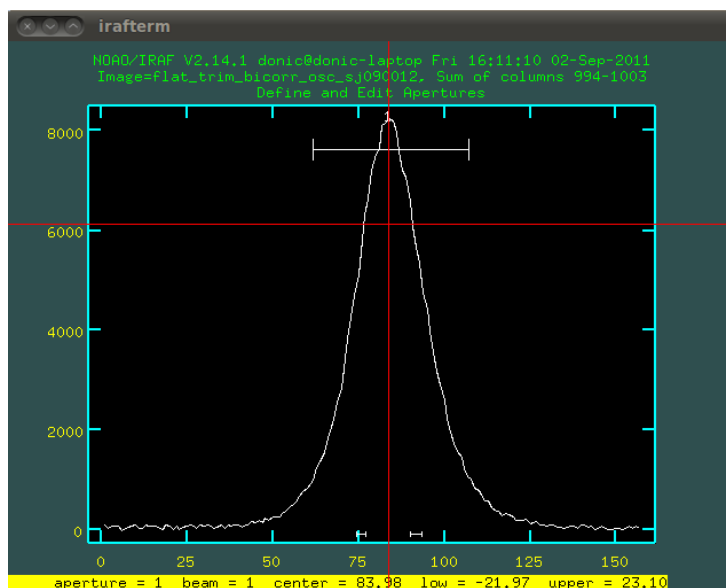
ponovo pritisnemo **q**. Sada ulazimo u sledeći fajl u ulaznoj listi i ponavljamo istu proceduru (u našem slučaju imamo tri spektra - dva 7 Vul i jedan 27 Vul)!

IRAF kreira folder po imenu **database** u kome se čuvaju relevantni podaci (koji će biti potrebni kasnije pri kalibraciji talasnih dužina).

Da bismo izašli iz funkcije bez njenog izvršavanja (ako se nešto pogreši) ukucajte **:q**.



Slika 22: Skica: Od posmatranja do analize.

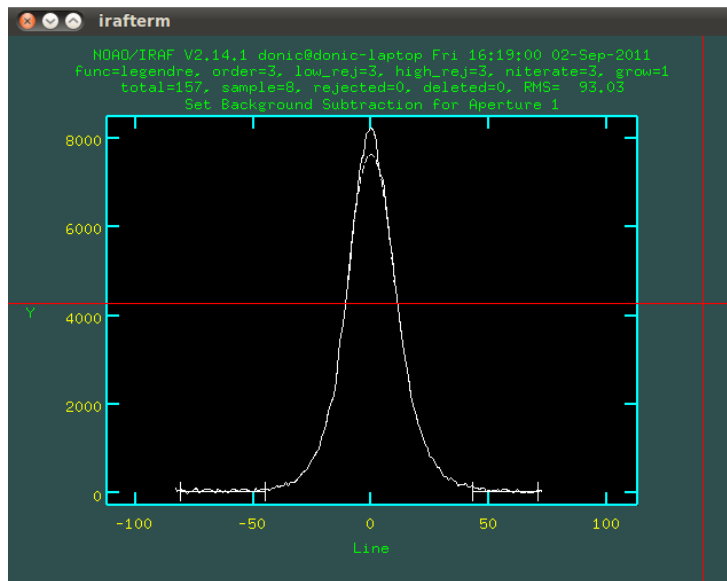


Slika 23: Definisanje aperture.

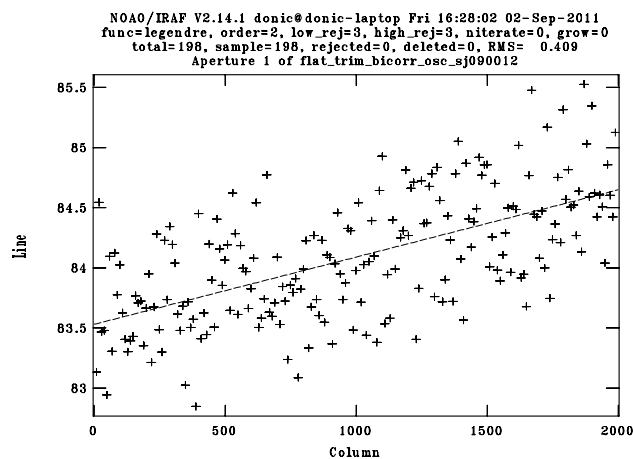
Sličan postupak je i za komparacionu lampu (arc spectrum) samo sa par bitnih izmena (unos se referentna lista - lista sa imenima fajlova za zvezde, takođe, parametri od find do fittrac se setuju na **no** kako koristimo podešavanja koja smo uneli pri otpakivanju spektara iz referentne liste, i na kraju **backgro=none**, opet, kako koristimo ranija podešavanja).

epar apall

I R A F



Slika 24: Definisanje pozadine.



Slika 25: Fit sa order=2.

Image Reduction and Analysis Facility

PACKAGE = apextract

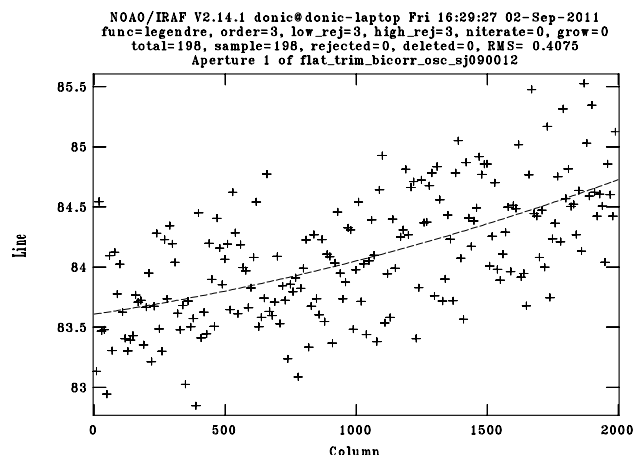
TASK = apall

```

input      =          @lamp List of input images
(output =   @extracted_lamp) List of output spectra
(apertur=   1) Apertures
(format =   multispec) Extracted spectra format
(referen=   @star) List of aperture reference images
(profile=   ) List of aperture profile images

(interac=   yes) Run task interactively?

```



Slika 26: Fit sa order=3.

```
(find      =      no) Find apertures?
(recente=   no) Recenter apertures?
(resize =   no) Resize apertures?
(edit      =   no) Edit apertures?
(trace     =   no) Trace apertures?
(fittrac=   no) Fit the traced points interactively?
(extract=   yes) Extract spectra?
(extras =   yes) Extract sky, sigma, etc.?
(review =   yes) Review extractions?

(line      =      INDEF) Dispersion line
(nsum      =      10) Number of dispersion lines to sum or median

# DEFAULT APERTURE PARAMETERS

(lower     =      -5.) Lower aperture limit relative to center
(upper     =      5.) Upper aperture limit relative to center
(apidtab=   ) Aperture ID table (optional)

# DEFAULT BACKGROUND PARAMETERS

(b_funct=   legendre) Background function
(b_order=   3) Background function order
(b_sampl=   -10:-6,6:10) Background sample regions
(b_naver=   1) Background average or median
(b_niter=   3) Background rejection iterations
(b_low_r=   3.) Background lower rejection sigma
(b_high_=   3.) Background upper rejection sigma
(b_grow =   1.) Background rejection growing radius

# APERTURE CENTERING PARAMETERS
```

```

(width =          5.) Profile centering width
(radius =        10.) Profile centering radius
(thresho=        0.) Detection threshold for profile centering

# AUTOMATIC FINDING AND ORDERING PARAMETERS

nfind =          Number of apertures to be found automatically
(minsep =         5.) Minimum separation between spectra
(maxsep =       1000.) Maximum separation between spectra
(order =          increasing) Order of apertures

# RECENTERING PARAMETERS

(aprecen=         ) Apertures for recentering calculation
(npeaks =        INDEF) Select brightest peaks
(shift =         yes) Use average shift instead of recentering?

# RESIZING PARAMETERS

(llimit =        INDEF) Lower aperture limit relative to center
(ulimit =        INDEF) Upper aperture limit relative to center
(ylevel =         0.1) Fraction of peak or intensity for automatic width
(peak =         yes) Is ylevel a fraction of the peak?
(bkg =          yes) Subtract background in automatic width?
(r_grow =         0.) Grow limits by this factor
(avglimi=        no) Average limits over all apertures?

# TRACING PARAMETERS

(t_nsum =        10) Number of dispersion lines to sum
(t_step =        10) Tracing step
(t_nlost=         3) Number of consecutive times profile is lost befo
(t_func=        legendre) Trace fitting function
(t_order=         2) Trace fitting function order
(t_sampl=         *) Trace sample regions
(t_naver=         1) Trace average or median
(t_niter=         0) Trace rejection iterations
(t_low_r=         3.) Trace lower rejection sigma
(t_high_=         3.) Trace upper rejection sigma
(t_grow =         0.) Trace rejection growing radius

# EXTRACTION PARAMETERS

(backgro=        none) Background to subtract
(skybox =         1) Box car smoothing length for sky
(weights=        none) Extraction weights (none|variance)
(pfit =         fit1d) Profile fitting type (fit1d|fit2d)

```

```

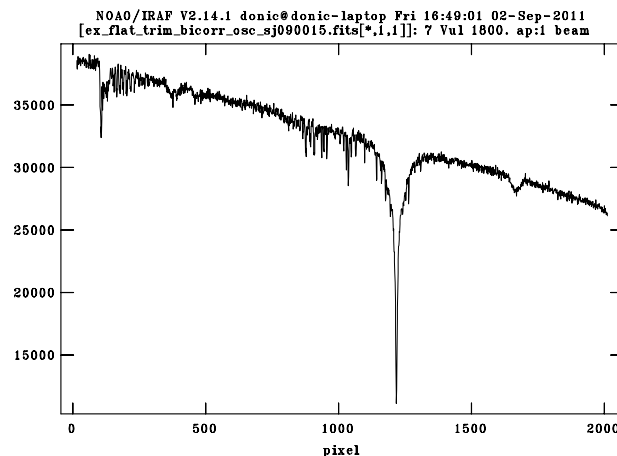
(clean =          yes) Detect and replace bad pixels?
(saturat=        INDEF) Saturation level
(readnoi=        0.) Read out noise sigma (photons)
(gain =          1.) Photon gain (photons/data number)
(lsigma =        4.) Lower rejection threshold
(usigma =        4.) Upper rejection threshold
(nsubaps=        1) Number of subapertures per aperture
(mode =          ql)

```

:go i **Enter**

Na sva pitanja odgovaramo pozitivno (ne zaboravi da imaš tri lampe).

Da bismo pogledali otpakovane spektre izlazimo iz `twod` sa `bye` (dok ne dodemo do `ec1>`) i ulazimo u `oned` (mogli smo i odmah da ukucamo `oned`) i kucamo `splot ex*` (vidi sliku 27). Prepoznamo nešto što liči na uobičajeni izgled spektra jedino što na x -osi još uvek nisu talasne dužine. Iz `splot`-a (i svih grafičkih okruženja) se izlazi ISKLJUČIVO pritiskom na **q** !



Slika 27: Otpakovani spektar zvezde 7 Vul (prvo posmatranje).

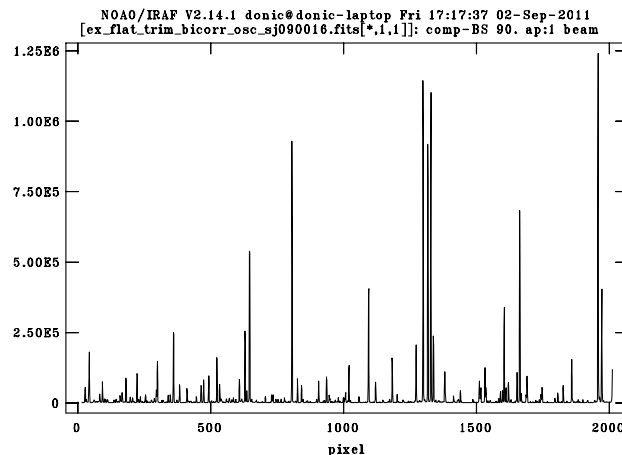
7 Kalibracija talasnih dužina

Sada je na redu kalibracija talasnih dužina (zanimaju nas raspodela fluksa po talasnim dužinama). Ulazimo u paket `oned` i pokrećemo funkciju `identify: epar identify`. U našem slučaju koristi se Torijum-Argon lampa. Zato je neophodno izabrati u IRAF-u baš tu opciju (u folderu `\iraf\noao\lib\line lists` postoji veliki broj datoteka koji odgovaraju različitim komparacionim lampama - mi biramo Torijum-Argon lampu: `thar.dat`).

U tu svrhu nam je potreban atlas spektralnih linija za Torijum-Argon lampu. Npr. na stranici: <http://www.noao.edu/kpno/specatlas/thar/thar.html> možemo naći željeni atlas. Ukucamo željeni opseg talasnih dužina (od 6255 do 6767 Å) i izaberemo da prikaže

npr. 25 linija (previše je pregusto a i nije potrebno). Spektar možemo sačuvati u .ps formatu (vidi sliku 31).

Spektar komparacione lampe koji odgovara prvom posmatranju zvezde 7 Vul je dat na slici 28.



Slika 28: Spektar komparacione lampe (asocirana prvom posmatranju zvezde 7 Vul).

Pre nego što krenemo sa identifikacijom linija bitno je da primetimo da je IRAF u jednom momentu kreirao slike formata .fits ! Do sada smo imali .fit fajlove i oni su pod tim imenom u našoj listi `extracted_lamp` (isto i za `extracted_star`). To moramo prepraviti!

Potrebno je odrediti funkciju koja povezuje raspodelu po pikselima sa raspodelom po talasnim dužinama.

`epar identify`

I R A F

Image Reduction and Analysis Facility

PACKAGE = onedspec

TASK = identify

```

images =      @extracted_lamp Images containing features to be identified
(section=      middle line) Section to apply to two dimensional images
(databas=      database) Database in which to record feature data
(coordli=      linelists$thar.dat) User coordinate list
(units =      ) Coordinate units
(nsum =      10) Number of lines/columns/bands to sum in 2D image
(match =      -3.) Coordinate list matching limit
(maxfeat=      50) Maximum number of features for automatic identif
(zwidth =      100.) Zoom graph width in user units
(ftype =      emission) Feature type
(fwidth =      4.) Feature width in pixels
(cradius=      5.) Centering radius in pixels

```

```

(thresho=          0.) Feature threshold for centering
(minsep =          2.) Minimum pixel separation
(funcio=          legendre) Coordinate function
(order =          2) Order of coordinate function
(sample =          *) Coordinate sample regions
(niterat=         2) Rejection iterations
(low_rej=         3.) Lower rejection sigma
(high_re=         3.) Upper rejection sigma
(grow =           0.) Rejection growing radius
(autowri=         no) Automatically write to database
(graphic=        stdgraph) Graphics output device
(cursor =         ) Graphics cursor input
crval =           Approximate coordinate (at reference pixel)
cdelt =           Approximate dispersion
(aidpars=         ) Automatic identification algorithm parameters
(mode =           ql)

```

: go i **Enter**

Sada je bitno biti veoma oprezan - identifikovati linije na spektru komparacione lampe. Sa kursorom i pritiskom na **m** označavamo liniju i unosimo talasnu dužinu i pritiskamo **Enter**. Bitno je da se dobro pokriju svi delovi spektra kako bi fit bio bolji. Koristimo **w** pa **e** za zumiranje a za vraćanje je **w** pa **a**. Kada identifikujemo par linija (bar 3-4, pri čemu vodimo računa da spektar bude dobro pokriven) možemo fitovati spektar kucanjem **f**. Mi zapravo tražimo funkciju koja povezuje raspodelu po pikselima sa raspodelom po talasnim dužinama! Ako grafik koji dobijemo (grafik razlika! - vidi slike 29 i 30) pokazuje izvesnu pravilnost (proverimo vizuelnom inspekcijom) (želimo stohastičnost razlika - NE ŽELIMO PRAVILNOST U RAZLIKAMA!) onda menjamo red polinoma kojim fitujemo uz pomoć npr. :o 3 i ponavljamo fitovanje sa **f**. Koristimo Ležandrovu funkciju :f **legendre** (mogli smo i odmah da setujemo kroz **epar identify**). Kada smo zadovoljni sa fitom završavamo sa kucanjem **q**. Na sva pitanja odgovaramo pozitivno. Možemo postupiti na sledeći način: prvo identifikujemo npr. deset linija pa fitujemo a zatim se vratimo i dodajemo ostale (ako je fit bio dobar onda će IRAF sam prepoznati talasne dužine selektovanih linija a mi ćemo samo kucati **Enter** što olakšava posao). Ako hoćemo da proverimo vrednosti talasnih dužina asociranih linijama možemo ukucati: :labels coord (vidi sliku 29). Ako želimo da uklonimo pokazivanje vrednosti talasnih dužina linija na grafiku kucamo: :labels none.

Kada se završi sa identifikacijom linija neophodno je *dodeliti zvezdi lampu*. Koristimo funkciju **refspec** i pozivamo: **epar refspec**.

epar refspec

I R A F

Image Reduction and Analysis Facility

PACKAGE = onedspec

TASK = refspectra


```

answer =          yes  Accept assignment?
(mode   =          ql)

```

:go i Enter

Odgovorimo pozitivno na sva pitanja. Na kraju pozivamo funkciju `dispcor` sa `epar dispcor` da bismo kreirali novi fajl u kojem je smešten spektar zvezde kalibrisan za talasne dužine odnosno da bi se umesto piksela na x -osi dobile vrednosti talasnih dužina. Pre toga je neophodno napraviti listu imena fajlova u kojima se nalaze spektri koji su kalibrisani za talasne dužine: `cp extracted_star wc_star, vi wc_star` (dodajemo `wc_`).

```
epar dispcor
```

I R A F

Image Reduction and Analysis Facility

```
PACKAGE = onedspec
```

```
TASK = dispcor
```

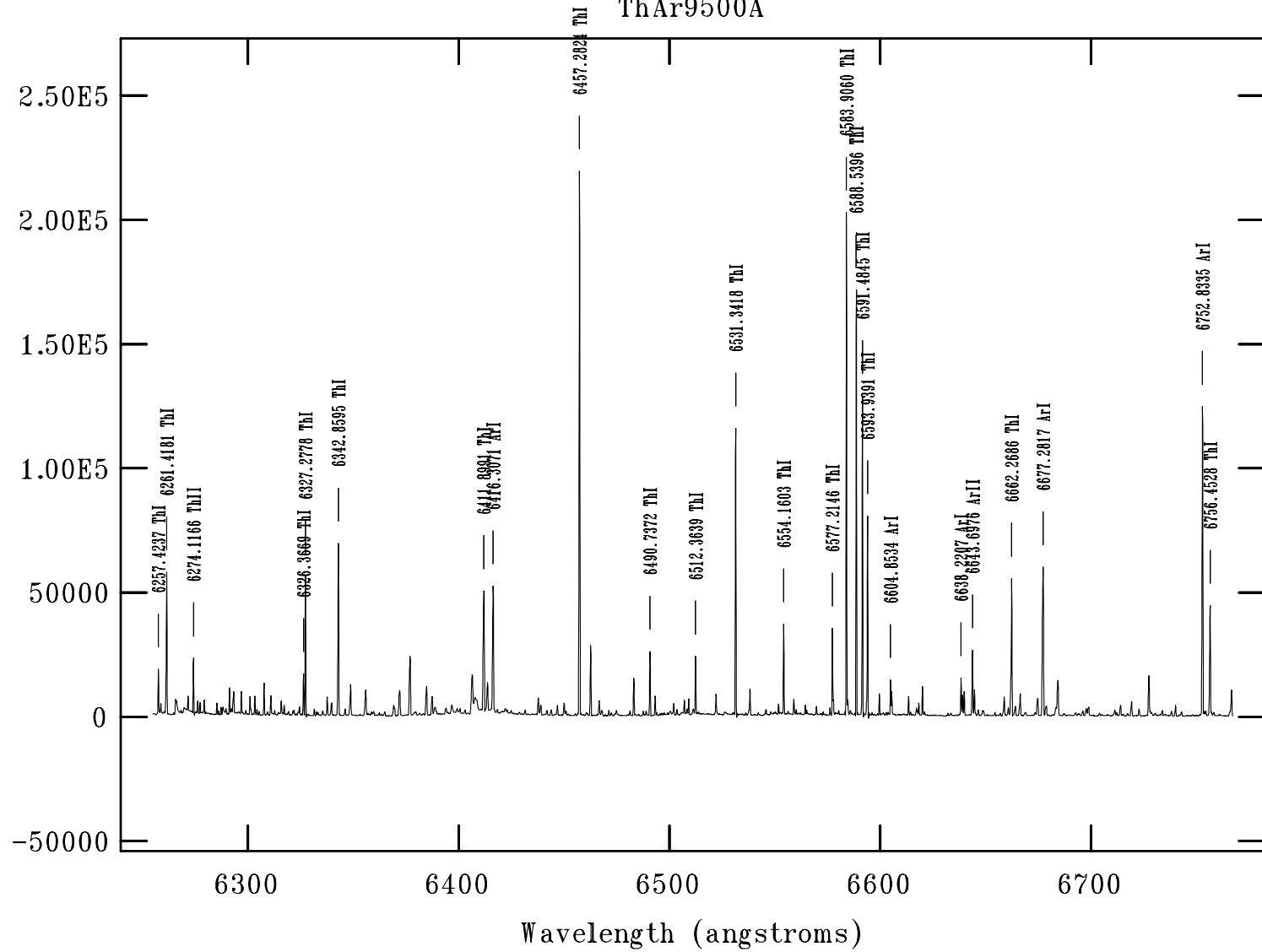
```

input   =      @extracted_star  List of input spectra
output  =      @wc_star        List of output spectra
(lineari=      yes) Linearize (interpolate) spectra?
(databas=      database) Dispersion solution database
(table   =      ) Wavelength table for apertures
(w1      =      INDEF) Starting wavelength
(w2      =      INDEF) Ending wavelength
(dw      =      INDEF) Wavelength interval per pixel
(nw      =      INDEF) Number of output pixels
(log      =      no) Logarithmic wavelength scale?
(flux     =      yes) Conserve total flux?
(blank    =      0.) Output value of points not in input
(samedis=      no) Same dispersion in all apertures?
(global   =      no) Apply global defaults?
(ignorea=      no) Ignore apertures?
(confirm=      no) Confirm dispersion coordinates?
(listonl=      no) List the dispersion coordinates only?
(verbose=      yes) Print linear dispersion assignments?
(logfile=      ) Log file
(mode    =      ql)

```

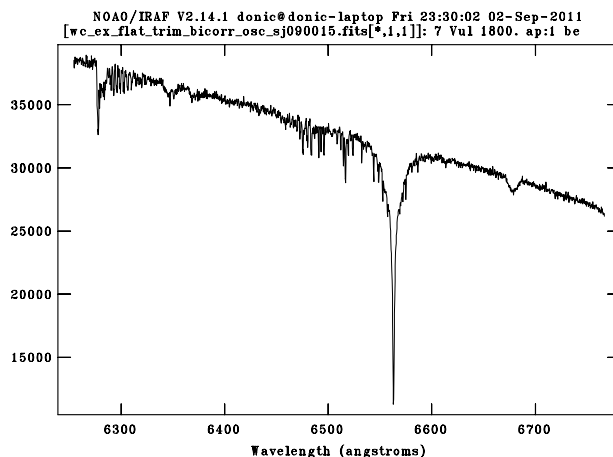
:go i Enter

NOAO/IRAF V2.14.1 @www.tuc.noao.edu Fri 12:26:16 02-Sep-2011
 identify output88373 - Ap 1
 ThAr9500A



Slika 31: Torijum - Argon atlas u željenom opsegu talasnih dužina.

Sada smo dobili spektar koji je kalibrisan za talasne dužine (vidi sliku 32). Sada je jasno da je linija koju smo ranije uočili zaista $H\alpha$ linija.



Slika 32: Spektar zvezde 7Vul kalibrisan za talasne dužine.

Kalibracija fluksa

Ukoliko nas zanima npr. određivanje radijalnih brzina (proučavanje dvojnosti, pulsacija,...) onda nije neophodno fluks kalibrisati posmatranja. U ovoj vežbi neće biti reči o fluks kalibraciji (vidi vežbu 2) kako je i između ostalog reč o malom opsegu talasnih dužina tako da kalibracija fluksa ne daje više informacija.

Na ovom mestu smo završili sa osnovnom redukcijom posmatranja! Sada nastavljamo sa daljom obradom posmatranja koja će nas dovesti do podataka spremnih za naučnu analizu.

8 Telluric korekcija

Ukoliko splot-ujemo spektar zvezde (slika 32) jasno se uočavaju linije koje potiču iz naše atmosfere (npr. linija vodene pare). Ukoliko, pre naučne analize, želimo da uklonimo linije koje potiču iz naše atmosfere onda možemo postupiti na sledeći način.

Za telluric korekciju obično se koriste zvezde u okolini posmatrane zvezde (i u datom trenutku vremena). Bilo bi idealno kada bi ta referentna zvezda imala spektar bez izraženih linija. U našem slučaju, iskoristili smo posmatranje zvezde 27 Vul.

Prvo je neophodno normalizovati spektar referentne zvezde (27 Vul, HR7880). To radimo uz pomoć funkcije `splot`. Unutar `splot`-a pritiskamo `t` i zatim `/`. Biramo npr. `legendre` dok je red u našem slučaju oko 4-6. Druga varijanta je da koristimo funkciju `conti` za normalizaciju spektra.

Sada je neophodno da napravimo model spektra. Ono što mi, zapravo, želimo jeste da spektar referentne zvezde (27 Vul) podelimo sa modelom te zvezde (koji naravno nema atmosferske linije!). Na taj način dobijamo template atmosferskih linija koji dalje koristimo pri korekciji našeg spektra (7 Vul).

Model možemo napraviti na sledeći način. Zapisujemo parametre linija za koje smo sigurni da potiču sa zvezde - fitujemo linije Lorencovim profilom (kursorom se pozicioniramo na levu i desnu granicu linije pri čemu sa leve strane pritisnemo **k**, a sa desne **l**), Gausovim profilom (sa leve i sa desne **k**) ili Fojtovim profilom (sa leve **k**, a sa desne **v**). Sve podatke zapisujemo u jednu datoteku npr.:

```
6562.45 -0.39 v 12.0 11.7
6346.76 -0.020 g 8.8
6370.79 -0.012 g 7.3
```

Prva vrednost je izmerena centralna talasna dužina, druga vrednost je jačina linije (fluks - negativna znači da je reč o apsorpciji), zatim slede oznake g, v, ili l da sugerišu o kom profilu je reč i konačno FWHM linija. Ako je izabran Fojtov profil onda je neophodno da se zapišu i Gausovska i Lorencova FWHM.

Sada ulazimo u paket `artdata` i koristimo funkciju `mk1dspec` koja generiše model 1D spektra zvezde na osnovu podataka iz datoteke koju smo kreirali. Generalno, ovaj deo ume da bude veoma naporan i zahteva iskustvo.

`mk1dspec`

I R A F

Image Reduction and Analysis Facility

PACKAGE = `artdata`

TASK = `mk1dspec`

```
input      =      model.fits  List of input spectra
(output    =      ) List of output spectra
(ap        =      1) Image line
(rv        =      0.) Radial velocity (km/s) or redshift
(z         =      no) Is velocity a redshift?
```

IF NEW SPECTRUM

```
(title     =      model) Title of spectrum
(ncols     =      1998) Number of columns
(naps      =      1) Number of lines (apertures)
(header    =      artdata$stdheader.dat) Image or header keyword file
(wstart    =      6255.) Starting wavelength (Angstroms)
(wend      =      6767.) Ending wavelength (Angstroms)
```

CONTINUUM

```
(continuum =      1.) Continuum at first pixel
(slope     =      0.) Continuum slope per pixel
```

```

(tempera=          0.) Blackbody temperture (Kelvin)
(fnu    =          no) F-nu or F-lamda?

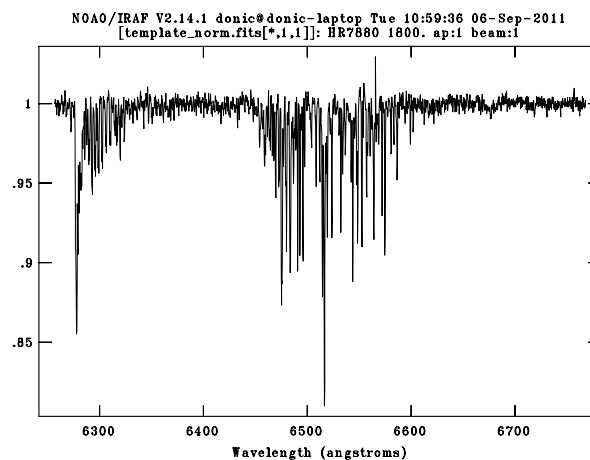
                                LINES
(lines   =          lines_proba) List of files containing lines
(nlines =          0) Number of random lines (if new line list)
(profile=          lorentzian) Default profile type
(peak   =          0.) Default peak strength (relative to continuum)
(gfwhm  =          0.) Default Gaussian FWHM (Angstroms)
(lfwhm  =          0.) Default Lorentzian FWHM (Angstroms)
(seed   =          1) Random number seed

(comment=          yes) Add comments to image?
(mode   =          ql)

```

Bitno je da proverimo kako smo trimovali naše spektre na početku kako bi se dimenzije modela i našeg spektra poklapale!

Sada delimo posmatrani spektar referentne zvezde (27 Vul) sa modelom uz pomoć funkcije `sarith` (slično kao `imarith` kod 2D spektra) i dobijamo template za telluric korekciju. Na slici 33 dat je primer template-a za telluric korekciju.



Slika 33: Primer template-a za telluric korekciju.

Konačno, izvršavamo funkciju `telluric`, koja uklanja linije koje potiču iz Zemljine atmosfere (ulazni podaci: ime fajla koji korigujemo ili lista sa imenima fajlova, ime, korigovanog, izlaznog fajla ili lista imena i ime telluric tamplat-a - podaci o airmass za trenutak posmatranja se mogu naći preko JSkyCalc-a:

<http://www.dartmouth.edu/~physics/faculty/skycalc/flyer.html>.

Funkcija `telluric` translira i skalira template po spektru naše zvezde ne bi li uklonila linije koje potiču iz Zemljine atmosfere sa što manjom greškom (vidi IRAF help).

```

cp wc_star star_list
vi star_list

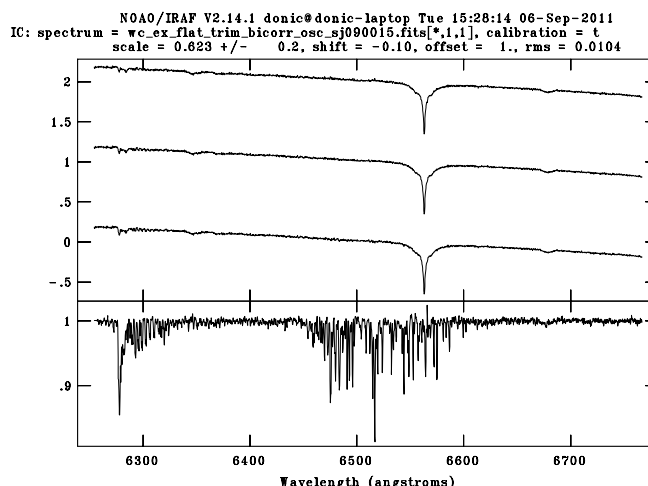
```

```
cp star_list tell_corr_list
vi tell_corr_list
```

```
telluric @star_list @tell_corr_list template_27Vul.fits
```

```
wc_ex_flat_trim_bicorr_osc_sj090015.fits: Airmass (1.): (2.647):
template_27Vul.fits: Airmass (1.): (2.647):
wc_ex_flat_trim_bicorr_osc_sj090015.fits[* ,1,1]: Search interactively?
(no|yes|NO|YES) (no): y
```

Na slici 34 predstavljen je deo procesa telluric korekcije upotrebom IRAF funkcije `telluric`. Centralni spektar je glavni ponuđeni (najmanja rms). Ukoliko nismo zadovoljni tim spektrom, (onim što nam je IRAF *ponudio*) IRAF nudi listu spektara (za različite translacije, skaliranja,...). Pozicioniranjem kursora na gornji spektar i pritiskom na **x** (za različito translirane) ili **y** (za različito skalirane spektre) ili pozicioniranjem kursora na donji spektar i opet pritiskom na **x** (ili **y**) možemo menjati spektre i na kraju izabrati onaj koji deluje da je najbolji. Pritiskom na **q** završavamo sa telluric korekcijom i izabrani spektar je onaj koji je u tom trenutku bio u sredini.

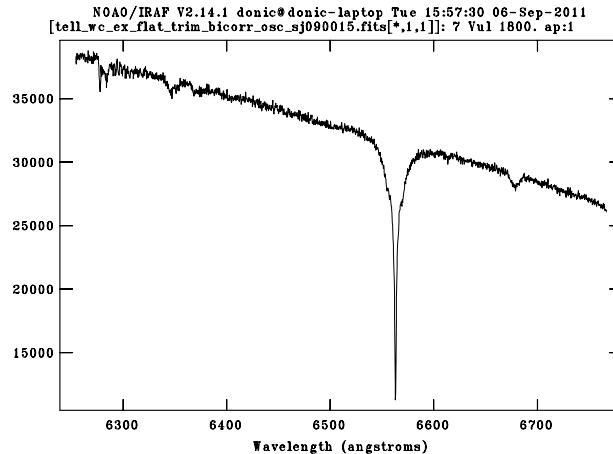


Slika 34: Proces telluric korekcije.

Na slici 35 dat je spektar zvezde 7 Vul nakon telluric korekcije. Vidi se da su linije vodene pare i dalje prisutne (ali umanjene). One se nikada ne mogu potpuno ukloniti kako je atmosfera nelinearni dinamički sistem (javlja se kaos) odnosno promene u atmosferi su brze i slučajne.

Ukoliko nas zanima koliki je odnos signala prema šumu (dok se nalazimo u `splot-u`) potrebno je da prvo normalizujemo spektar i zatim izaberemo najpravniji deo bez izraženih linija (u našem slučaju između $H\alpha$ i HeI linije) i pritisnemo **m** sa obe strane regiona.

Ukoliko želimo da uđemo u `hepl` dok smo u grafiku (`splot`, `implot`) dovoljno je da ukucamo **?**. Iz `help-a` se vraćamo u grafik kucanjem **q**.



Slika 35: Spektar zvezde 7 Vul (prvo posmatranje) nakon telluric korekcije.

9 Heliocentrična korekcija

Kako se posmatranje vrši na površini Zemlje neophodno je izvršiti tzv. heliocentričnu korekciju kako bismo uklonili uticaj Zemljinog kretanja. U tu svrhu koristimo program JSkyCalc (ili bilo koji drugi) koji možemo naći na:

<http://www.dartmouth.edu/~physics/faculty/skycalc/flyer.html>.

za određivanje popravki u brzini (barricentric correction).

U zavisnosti od naučnog problema koji analiziramo, heliocentričnu korekciju možemo izvršiti na dva načina.

Prvi način, koji ćemo mi koristiti, jeste da iz dobijenog spektra odredimo radijalne brzine (konkretno, mi radimo za $H\alpha$ - vidi kasnije) i da zatim izvršimo korekciju za Zemljino kretanje. Sa JSkyCalc (ili koristeći `rvcorrect` u `astutill` paketu) očitavamo vrednosti korekcija brzina za trenutak posmatranja i koordinate naše zvezde (sve informacije se nalaze u hederu slike):

$$V_c = V_{obs} + V_{BC}$$

Pored tako određenih brzina, koje zatim koristimo u naučnoj analizi, zapisujemo i heliocentrične (baricentrične) Julijanske dane HJD.

Drugi način bi bio da prvo izvršimo korekciju talasnih dužina (Doplerov pomak usled Zemljinog kretanja):

$$\lambda_c = \lambda_{obs} \left(1 + \frac{V_{BC}}{c} \right)$$

U tu svrhu je potrebno napisati program koji vrši datu korekciju. Naravno, opet koristimo JSkyCalc.

Da bismo dobili datoteke (.dat fajlove) iz fits fajlova koristimo IRAF funkciju `wspectext`. Kako nakon heliocentrične popravke želimo spektre zvezda u fits formatu koristimo IRAF funkciju `rspectext`. Bitno je VODITI RAČUNA da funkcija `rspectext` pravi novi heder sa informacijom o kreiranju novog fajla - **STARI HEDER JE IZGUBLJEN**. Sada

možemo odrediti radijalne brzine koje su već korigovane za Zemljino kretanje (vidi kasnije).

Ovde je dat primer programskog koda (u cpp-u, Tomić, 2011) koji može da se koristi za Heliocentričnu korekciju (ulazni podaci: datoteka koja sadrži listu imena ulaznih fajlova (ulazni fajlovi su spektri zvezde u .dat formatu), datoteka koja sadrži heliocentrične brzine u istom poretku kao i lista imena fajlova i datoteka koja sadrži imena izlaznih fajlova; izlazni podaci: fajlovi sa korigovanim spektrima u .dat formatu):

```
#include <iostream>
#include <string>
#include <fstream>
#include <ostream>
#include <locale>
#include <iomanip>
using namespace std;
int main()
{
    char s1[17];
    char s2[18];
    long double x,y,z,a;
    char c;
    // Lista sa imenima ulaznih fajlova
    ifstream ul("list_dat");
    // Lista sa imenima izlaznih fajlova
    ifstream ul2("list_dat2");
    // Lista sa heliocentričnim brzinama u istom poretku kao i fajlovi
    ifstream ulhel("helio.dat");
    // Ovde vrši račun i sve ostalo
    for(int i=0; i<2; i++)
    {
        ul.getline(s1,17);
        ul2.getline(s2,18);
        ifstream uldat(s1);
        ofstream izldat(s2);
        ulhel >> x;
        while (uldat >> y)
        {
            a=y*(1+x/299792.458);
            izldat.setf(ios::fixed, ios::floatfield);
            izldat.precision(4);
            izldat << a << " " << '\t';
            c=uldat.get();
            while (c!='\n')
            {
                izldat << c;
                c=uldat.get();
            }
        }
    }
}
```

```

        izldat << '\n';
    }
    izldat.close();
    uldat.close();
}
ul.close();
ul2.close();
ulhel.close();
return 0;
}

```

Program se kompajlira iz komandne linije kucanjem npr.: `g++ popravka.cpp -o executable` a izvršava naredbom `./executable` (proverite da li imate instaliran⁴ g++).

10 Primer naučne analize: Da li je data zvezda spektroskopski dvojna ?

Naš zadatak je da proverimo da li je zvezda spektroskopski dvojna. U tu svrhu ćemo meriti radijalnu brzinu na osnovu $H\alpha$ linije (6562.8 Å).

U svrhu merenja radijalnih brzina prikazaćemo spektar pomoću funkcije `splot`. Radi jednostavnosti i kako nas zanimaju isključivo centri linije (najizraženija), fitovaćemo $H\alpha$ liniju Gausovim profilom - kursorom se pozicioniramo na levu i desnu granicu linije pri čemu svaki put pritisnemo **k** (za Lorencov profil bi bilo sa leve **k**, a sa desne **l** dok je za Fojtov profil sa leve **k**, a sa desne **v**). Zapisujemo vrednosti centra linije i računamo radijalnu brzinu (za oba posmatranja date večeri). Iz `splot`-a se izlazi ISKLJUČIVO pritiskom na **q** !

Radijalnu brzinu računamo po poznatoj formuli:

$$V_r = \frac{\lambda - \lambda_0}{\lambda_0} c$$

pri čemu je λ_0 laboratorijska talasna dužina. Inače, za identifikaciju linija i za utvrđivanje laboratorijskih talasnih dužina može se npr. koristiti NIST-ova baza:

<http://www.nist.gov/pml/data/asd.cfm>

ili Kurucz:

<http://www.cfa.harvard.edu/amp/ampdata/kurucz23/sekur.html>

10.1 Kombinovanje rezultata

U gornjem primeru, redukovali smo dva posmatranja zvezde 7 Vul tokom jedne posmatračke večeri 09. 10. 2009. godine. Data zvezda je, na istoj opservatoriji, posmatrana od 23. 05. 2007. do 29. 06. 2010. godine. Kombinovaćemo naše rezultate (dve vrednosti

⁴Ubuntu pruža mogućnost jednostavne instalacije za veliki broj programa upotrebom (System>Administration) Synaptic Package Manager-a. Druga opcija je iz komandne linije komandom `sudo apt-get install ime_programa`.

radijalnih brzina) sa rezultatima dobijenim iz preostalih posmatranja kako bismo utvrdili da li je reč o spektrskopski dvojnjoj zvezdi.

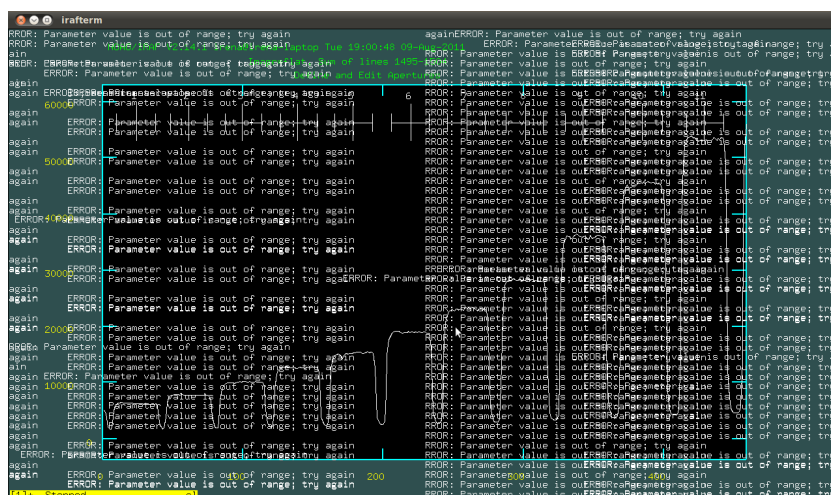
U datoteci `comb.txt` date su vrednosti HJD, baricentrične korekcije za radijalne brzine, merene talasne dužine, njene greške kao i korigovane radijalne brzine i njoj asocirane greške za $H\alpha$ liniju koje kombinujemo sa našim rezultatima. Na osnovu analize tako dobijene vremenske serije možemo utvrditi da li je reč o spektroskopski dvojnjoj zvezdi (tražimo da li postoji periodičnost).

Analizom vremenske serije možemo utvrditi da li postoji periodičnost ili ne pa tako i vrednost perioda. Klasična Furijeova transformacija je u ovom slučaju nepodesna kako je reč o vremenskoj seriji koja nije ekvidistantna (ravnomerno uzorkovana). Jedna varijanta bi bila da se iskoristi npr. splajn interpolacija i da se zatim primeni FFT (u ovom slučaju postoji opasnost od generisanja *lažnih* pikva koji odgovaraju *rupama* u vremenskoj seriji) dok je druga opcija npr. Lomb-Scargle periodogram. Period inosi oko 69 dana.

Ilustrativno je uraditi isti postupak i za liniju helijuma He I (6678.15 Å). Rezultati su veoma zanimljivi ($H\alpha$ i He I linija potiču sa različitih zvezda) !

U radu Vennes et al.: *On the nature of the Be star HR 7409 (γ Vul)* (2011, MNRAS, 413, 2760V) su dati detalji vezani za prezentovanu analizu zvezde 7Vul.

Na kraju, treba napomenuti da, ukoliko se tokom rada IRAF *zakuca* ne treba očajavati (vidi sliku 36). Potrebno je odmoriti se i krenuti iz početka.



Slika 36: Ponekad je potreban predah!

Veliku zahvalnost dugujem A. Kawki, S. Jonić, I. Pirković, S. Tomić, M. Smole, V. Randelovu i S. Miloševiću na pomoći pri pisanju ove skripte. Ova skripta je nastala kao rezultat rada tokom letnje prakse na Ondřejov observatoriji, Češka republika (leto 2011.).

(Pripremio: Dušan Onić)

Sadržaj

1	Priprema za redukciju posmatranja	1
2	Oduzimanje over-scan-a	6
3	Kombinovanje i oduzimanje biasa	10
4	Trimovanje	13
5	Flat-field korekcija	17
6	Otpakivanje spektra	19
7	Kalibracija talasnih dužina	28
8	Telluric korekcija	34
9	Heliocentrična korekcija	38
10	Primer naučne analize: Da li je data zvezda spektroskopski dvojna ?	40
10.1	Kombinovanje rezultata	40