Pracownia astrofizyczna IV rok –
opracowanie obserwacji spektroskpowych *echelle* za pomoca irafa

1. Przekonwertuj obserwacje (pliki fits) na pliki o formacie całkowitym, które będą opracowywane przez irafa:

                    Image Reduction and Analysis Facility
   PACKAGE = dataio
   TASK = **rfits**

   fits_fil= @inp_rfits FITS data source
   file_lis= File/extensions list
   iraf_fil= @out_rfits IRAF filename
   (make_im= yes) Create an IRAF image?
   (long_he= no) Print FITS header cards?
   (short_h= yes) Print short header?
   (datatyp= **u**) IRAF data type
   (blank = 0.) Blank value
   (scale = yes) Scale the data?
   (oldiraf= no) Use old IRAF name in place of iraf_file?
   (offset = 0) Tape file offset
   (mode   = ql)

2. Popraw wszystkie *bias-y* na overscan i odetnij go od obrazków:

   Wejdź do *imred* > *ccdred* i popraw biasy na *overscan*, a następnie go obetnij:
                    Image Reduction and Analysis Facility
   PACKAGE = ccdred
     TASK = **ccdproc**

   images = @inp_ccdproc List of CCD images to correct
   (output = @out_ccdproc) List of output CCD images
   (ccdtype=) CCD image type to correct
   (max_cac= 0) Maximum image caching memory (in Mbytes)
   (noproc = no) List processing steps only?

   (fixpix = no) Fix bad CCD lines and columns?
   (oversca= **yes**) Apply overscan strip correction?
   (trim   = **yes**) Trim the image?
   (zerocor= **no**) Apply zero level correction?
   (darkcor= no) Apply dark count correction?
   (flatcor= no) Apply flat field correction?
   (illumco= no) Apply illumination correction?
   (fringec= no) Apply fringe correction?
   (readcor= no) Convert zero level image to readout correction?

(scancor= no) Convert flat field image to scan correction?

(readaxi= line) Read out axis (column|line)
(fixfile=) File describing the bad lines and columns
(biassec= **[1045:1095]**) Overscan strip image section
(trimsec= **[16:1039,1:1024]**) Trim data section
(zero = bias_median) Zero level calibration image
(dark =) Dark count calibration image
(flat =) Flat field images
(illum =) Illumination correction images
(fringe =) Fringe correction images
(minrepl= 1.) Minimum flat field value
(scantyp= shortscan) Scan type (shortscan|longscan)
(nscan = 1) Number of short scan lines

(interac= no) Fit overscan interactively?
(functio= legendre) Fitting function
(order = 1) Number of polynomial terms or spline pieces
(sample = *) Sample points to fit
(naverag= 1) Number of sample points to combine
(niterat= 1) Number of rejection iterations
(low_rej= 3.) Low sigma rejection factor
(high_re= 3.) High sigma rejection factor
(grow = 0.) Rejection growing radius
(mode = ql)

3. Obejrzyj każdy z biasów używając *ds9* oraz *imexamine* (w irafie). Upewnij się, że fluktuacje w biasach są małe, czyli mniejsze niż 20 ADU ("l" wyświetla linie, "c", kolumny). Jeśli znajdziesz *biasy* z dużymi fluktuacjami, odrzuć je.

4. Policz średni *bias*:

Image Reduction and Analysis Facility
PACKAGE = ccdred
TASK = **zerocombine**

input = @list_bias_out List of zero level images to combine
(output = bias_average) Output zero level name
(combine= average) Type of combine operation
(reject = minmax) Type of rejection
(ccdtype=) CCD image type to combine
(process= no) Process images before combining?
(delete = no) Delete input images after combining?
(clobber= no) Clobber existing output image?
(scale = none) Image scaling
(statsec=) Image section for computing statistics
(nlow = 0) 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= 10) ccdclip: CCD readout noise (electrons)
(gain = 2.5) ccdclip: CCD gain (electrons/DN)
(snoise = 12.) ccdclip: Sensitivity noise (fraction)
(pclip = -0.5) pclip: Percentile clipping parameter
(blank = 0.) Value if there are no pixels
(mode = ql)

5. Popraw wszystkie obserwacje (gwiazdy i lampy) na średni bias:

Image Reduction and Analysis Facility
PACKAGE = ccdred
TASK = **ccdproc**

images = @inp_ccdproc List of CCD images to correct
(output = @out_ccdproc) List of output CCD images
(ccdtype=) CCD image type to correct
(max_cac= 0) Maximum image caching memory (in Mbytes)
(noproc = no) List processing steps only?

(fixpix = no) Fix bad CCD lines and columns?
(oversca= yes) Apply overscan strip correction?
(trim = yes) Trim the image?
(zerocor= **yes**) Apply zero level correction?
(darkcor= no) Apply dark count correction?
(flatcor= no) Apply flat field correction?
(illumco= no) Apply illumination correction?
(fringec= no) Apply fringe correction?
(readcor= no) Convert zero level image to readout correction?
(scancor= no) Convert flat field image to scan correction?

(readaxi= line) Read out axis (column|line)
(fixfile=) File describing the bad lines and columns
(biassec= [1045:1095]) Overscan strip image section
(trimsec= [16:1039,1:1024]) Trim data section
(zero = **bias_median**) Zero level calibration image
(dark =) Dark count calibration image
(flat =) Flat field images
(illum =) Illumination correction images
(fringe =) Fringe correction images
(minrepl= 1.) Minimum flat field value
(scantyp= shortscan) Scan type (shortscan|longscan)
(nscan = 1) Number of short scan lines

(interac= no) Fit overscan interactively?

(functio= legendre) Fitting function
(order = 1) Number of polynomial terms or spline pieces
(sample = *) Sample points to fit
(naverag= 1) Number of sample points to combine
(niterat= 1) Number of rejection iterations
(low_rej= 3.) Low sigma rejection factor
(high_re= 3.) High sigma rejection factor
(grow = 0.) Rejection growing radius
(mode = ql)

6. Obejrzyj wszystkie flatfieldy za pomocą programu skycat i upewnij się, że żaden z nich nie jest prześwietlony.

7. Utwórz średni *flatfield*:

Image Reduction and Analysis Facility
PACKAGE = ccdred
TASK = **flatcombine**

input = @list_flat List of flat field images to combine
(output = flat_median) Output flat field root name
(combine= median) Type of combine operation
(reject = avsigclip) Type of rejection
(ccdtype=) CCD image type to combine
(process= yes) Process images before combining?
(subsets= yes) Combine images by subset parameter?
(delete = no) Delete input images after combining?
(clobber= no) Clobber existing output image?
(scale = mode) Image scaling
(statsec=) Image section for computing statistics
(nlow = 1) minmax: Number of low pixels to reject
(nhigh = 1) minmax: Number of high pixels to reject
(nkeep = 1) Minimum to keep (pos) or maximum to reject (neg)
(mclip = yes) Use median in sigma clipping algorithms?
(lsigma = 50.) Lower sigma clipping factor
(hsigma = 7.) Upper sigma clipping factor
(rdnoise= 10) ccdclip: CCD readout noise (electrons)
(gain = 2.5) ccdclip: CCD gain (electrons/DN)
(snoise = 12) ccdclip: Sensitivity noise (fraction)
(pclip = -0.5) pclip: Percentile clipping parameter
(blank = 1.) Value if there are no pixels
(mode = ql)

8. Zbuduj bazę danych dla szerokich apertur używając widma β Oph:

Wejdź do *imred* > *echelle*:

Image Reduction and Analysis Facility

PACKAGE = echelle
TASK = **apall**

input = beta_Oph.fits List of input images
(output =) List of output spectra
(apertur=) Apertures
(format = echelle) 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 = **no**) Resize apertures?
(edit = yes) Edit apertures?
(trace = yes) Trace apertures?
(fittrac= yes) Fit the traced points interactively?
(extract= **no**) Extract spectra?
(extras = no) Extract sky, sigma, etc.?
(review = no) Review extractions?

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

DEFAULT APERTURE PARAMETERS

(lower = -6.) Lower aperture limit relative to center
(upper = 6.) Upper aperture limit relative to center
(apidtab=) Aperture ID table (optional)

DEFAULT BACKGROUND PARAMETERS

(b_funct= chebyshev) Background function
(b_order= 1) Background function order
(b_sampl= -10:-6,6:10) Background sample regions
(b_naver= -3) Background average or median
(b_niter= 0) Background rejection iterations
(b_low_r= 3.) Background lower rejection sigma
(b_high_= 3.) Background upper rejection sigma
(b_grow = 0.) 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  =                 19  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 wi
(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 be
(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=              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  =                no) Detect and replace bad pixels?
(saturat=           INDEF) Saturation level
(readnoi=              10.) Read out noise sigma (photons)
(gain   =              2.5) Photon gain (photons/data number)
(lsigma =               10.) Lower rejection threshold
```

(usigma = **10**.) Upper rejection threshold
(nsubaps= 1) Number of subapertures per aperture
(mode = ql)

9. Teraz popraw redni *flatfield*, obserwacje gwiazd i lamp kalibracyjnych na światło rozproszone:

Image Reduction and Analysis Facility
PACKAGE = echelle
TASK = **apscatter**
input = @inp_apscat List of input images to subtract scattered light
output = @out_apscat List of output corrected images
(apertur=) Apertures
(scatter=) List of scattered light images (optional)
(referen= O_20060621_194009_b.fits) List of aperture reference images

(interac= yes) Run task interactively?
(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?
(subtrac= **yes**) Subtract scattered light?
(smooth = **yes**) Smooth scattered light along the dispersion?
(fitscat= **yes**) Fit scattered light interactively?
(fitsmoo= **yes**) Smooth the scattered light interactively?

(line = INDEF) Dispersion line
(nsum = 10) Number of dispersion lines to sum or median
(buffer = 1.) Buffer distance from apertures
(apscat1=) Fitting parameters across the dispersion
(apscat2=) Fitting parameters along the dispersion
(mode = ql)

10. Skoryguj szerokość rzędów używając widma β Oph poprawionego na światło rozproszone. Użyj ponownie apall ale daj **resize: yes**. Upewnij się, że szerokości rzędów są dobrze dobrane.

11. Wydobądź widma β Oph, lamp kalibracyjnych i średniego *flatfieldu,* **nie czyszcząc** widm z promieni kosmicznych w sposób automatyczny. Dla każdego obiektu widmo zostanie zapisane w pliku z rozszerzeniem ec.fits. Jako odniesienia użyj widma β Oph poprawionego na światło rozproszone.

Image Reduction and Analysis Facility
PACKAGE = echelle
TASK = **apall**
input = @inp_apall_th List of input images

(output =) List of output spectra
(apertur=) Apertures
(format = echelle) Extracted spectra format
(referen= O_20050613_204207**bs**.fits) List of aperture reference images
(profile= O_20050613_204207**bs**.fits) List of aperture profile images

(interac= **yes**) Run task interactively?
(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 = no) Extract sky, sigma, etc.?
(review = **yes**) Review extractions?
(Pozostałe podpunkty tak samo jak poprzednio.)
(lsigma = **50.**) Lower rejection threshold
(usigma = **12**.) Upper rejection threshold

12. Wydobądź widma pozostałych, słabych gwiazd czyszcząc je automatycznie z promieni kosmicznych. (clean = **yes**) Detect and replace bad pixels?

13. Zidentyfikuj linie widmowe w widmach lamp kalibracyjnych dla każdej nocy osobno. Zacznij od widma pierwszej lampy kalibracyjnej, czyli pierwszego pliku C*.ec.fits Pliki ze zidentyfikowanymi liniami lampy Th-Ar w danych aperturach pobierz ze strony ćwiczeń.

PACKAGE = echelle
TASK = ecidentify

images = C_20060621_193244_bs.ec.fits Images containing features to be identified
(databas= database) Database in which to record feature data
(coordli= linelists$thar.dat) User coordinate list
(units =) Coordinate units
(match = 1.) Coordinate list matching limit in user units
(maxfeat= 100) Maximum number of features for automatic identification
(zwidth = 10.) Zoom graph width in user units
(ftype = emission) Feature type
(fwidth = 4.) Feature width in pixels
(cradius= 5.) Centering radius in pixels
(thresho= 20.) Feature threshold for centering
(minsep = 2.) Minimum pixel separation
(functio= chebyshev) Coordinate function
(xorder = 4) Order of coordinate function along dispersion
(yorder = 4) Order of coordinate function across dispersion
(niterat= 0) Rejection iterations
(lowreje= 3.) Lower rejection sigma

(highrej= 3.) Upper rejection sigma
(autowri= no) Automatically write to database?
(graphic= stdgraph) Graphics output device
(cursor =) Graphics cursor input
(mode = ql)

Zwróć uwagę na to, że skala na obrazku jest odwrócona. Należy odbić oś X względem Y poleceniem *w e-e* (pierwsze *e* naciskasz, gdy kursor jest w miejscu, które ma się stać lewym dolnym rogiem obrazka, drugie *e*, w miejscu które ma się stać prawym górnym rogiem obrazka). Aby zaznaczyć linię użyj *m*. Następnie podaj jej długość w skali angstremów. *k* przechodzi do następnej apertury. Gdy skończysz ze wszystkimi 19 rzędami, **zostań w 19-tym rzędzie** i użyj *f* aby znaleźć dopasowanie. Zwróć uwagę, że miara dopasowania musi być rzędu 0.01. Absolutnie nie może przekraczać kilku setnych! Jeśli jest większa, wróć do apertur przez *q* i znajdź źródło problemu. Gdy dopasowanie jest satysfakcjonujące, naciśnij *q*, a następnie wpisz *:threshold 500* aby zmniejszyć próg identyfikacji, oraz *l* aby pobrać więcej linii. Sprawdź jakość dopasowania przez *f* i gdy skończysz, wyjdź przez *q, q.*

14. Dla drugiej (i kolejnych) lamp z tej nocy użyj ecreidentify aby zidentyfikować linie:

PACKAGE = echelle
TASK = ecreidentify

images = C_20060621_193358_bs.ec.fits Spectra to be reidentified
referenc= C_20060615_192334_bs.ec Reference spectrum
(shift = 0.) Shift to add to reference features
(cradius= 5.) Centering radius
(thresho= 500.) Feature threshold for centering
(refit = yes) Refit coordinate function?
(databas= database) Database
(logfile= STDOUT,logfile) List of log files
(mode = ql)

15. Znajdź dopasowanie do **widma** uśrednionego *flatfieldu*. Do dopasowania użyj wielomianu **wysokiego** rzędu. Dopasowanie powinno być jak najlepsze. Zwróć uwagę co się dzieje na końcach przedziału.

PACKAGE = echelle
TASK = continuum

input = flat_aver Input images
output = flat_aver_fit Output images
(lines = *) Image lines to be fit
(bands = *) Image bands to be fit
(type = **fit**) Type of output
(replace= no) Replace rejected points by fit?
(wavesca= yes) Scale the X axis with wavelength?
(logscal= no) Take the log (base 10) of both axes?

(overrid=	no)	Override previously fit lines?
(listonl=	no)	List fit but don't modify any images?
(logfile=	logfile)	List of log files
(interac=	yes)	Set fitting parameters interactively?
(sample =	*)	Sample points to use in fit
(naverag=	1)	Number of points in sample averaging
(functio=	legendre)	Fitting function
(order =	**23**)	Order of fitting function
(low_rej=	2)	Low rejection in sigma of fit
(high_re=	2)	High rejection in sigma of fit
(niterat=	10)	Number of rejection iterations
(grow =	1.)	Rejection growing radius in pixels
(markrej=	no)	Mark rejected points?
(graphic=	stdgraph)	Graphics output device
(cursor =	)	Graphics cursor input
ask =	yes	
(mode =	ql)	

16. Znormalizuj to dopasowanie do maksimum przy linii Hα (apertura 2), czyli podziel otrzymane dopasowanie przez wartość, jaką odczytasz jako maksimum w drugiej aperturze.

PACKAGE = echelle
TASK = sarith

input1 =	**flat_aver_fit**	List of input spectra
op =	/	Operation
input2 =	*wartosc_max_Ap2*	List of input spectra or constants
output =	**flat_norm**	List of output spectra
(w1 =	INDEF)	Starting wavelength
(w2 =	INDEF)	Ending wavelength
(apertur=	)	List of input apertures or columns/lines
(bands =	)	List of input bands or lines/bands
(beams =	)	List of input beams or echelle orders
(apmodul=	0)	Input aperture modulus (0=none)
(reverse=	no)	Reverse order of operands in binary operation?
(ignorea=	no)	Ignore second operand aperture numbers?
(format =	multispec)	Output spectral format
(renumbe=	no)	Renumber output apertures?
(offset =	0)	Output aperture number offset
(clobber=	no)	Modify existing output images?
(merge =	no)	Merge with existing output images?
(rebin =	yes)	Rebin to exact wavelength region?
(errval =	0.)	Arithmetic error replacement value
(verbose=	no)	Print operations?
(mode =	ql)	

17. Podziel widma gwiazdowe przez znormalizowane dopasowanie do średniego *flatfieldu*. Lista wynikowa powinna zawierać nazwy gwiazd.

PACKAGE = echelle
TASK = sarith

input1 =	@lista_gwiazd	List of input spectra
op =	/	Operation
input2 =	**flat_norm**	List of input spectra or constants
output =	@lista_star_names	List of output spectra
(w1 =	INDEF)	Starting wavelength
(w2 =	INDEF)	Ending wavelength
(apertur=	)	List of input apertures or columns/lines
(bands =	)	List of input bands or lines/bands
(beams =	)	List of input beams or echelle orders
(apmodul=	0)	Input aperture modulus (0=none)
(reverse=	no)	Reverse order of operands in binary operation?
(ignorea=	no)	Ignore second operand aperture numbers?
(format =	multispec)	Output spectral format
(renumbe=	no)	Renumber output apertures?
(offset =	0)	Output aperture number offset
(clobber=	no)	Modify existing output images?
(merge =	no)	Merge with existing output images?
(rebin =	yes)	Rebin to exact wavelength region?
(errval =	0.)	Arithmetic error replacement value
(verbose=	no)	Print operations?
(mode =	ql)	

18. Zdefiniuj widma referncyjne, które będą użyte w rozwiązaniu dyspersyjnym. List of input spectra to lista z nazwami plików z widmami gwiazd, List of reference spectra to lista widm lamp kalibracyjnych z danej nocy.

PACKAGE = echelle
TASK = refspectra

input =	@lista_star_names	List of input spectra
(referen=	@list_ThAr)	List of reference spectra
(apertur=	)	Input aperture selection list
(refaps =	)	Reference aperture selection list
(ignorea=	yes)	Ignore input and reference apertures?
(select =	interp)	Selection method for reference spectra
(sort =	UTMIDDLE)	Sort key
(group =	)	Group key
(time =	yes)	Is sort key a time?
(timewra=	17.)	Time wrap point for time sorting
(overrid=	yes)	Override previous assignments?
(confirm=	yes)	Confirm reference spectrum assignments?

(assign = yes) Assign the reference spectra to the input spectrum?
(logfile= STDOUT,logfile) List of logfiles
(verbose= yes) Verbose log output?
answer = yes Accept assignment?
(mode = ql)

19. Przenieś widma gwiazdowe na skalę angstremów, poprawiając na dyspersję. Wynikowe **widma gwiazdowe powinny znaleźć się na skali angstremów.** Sprawdź to używając splot.

PACKAGE = echelle
TASK = dispcor
input = @lista_star_names List of input spectra
output = @lista_star_names 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 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)

20. Znormalizuj widma gwiazdowe. Użyj continuum ale tym razem wielomianu niskiego rzędu!
PACKAGE = echelle
TASK = continuum

input = @lista_stars_names Input images
output = @lista_stars_norm Output images
(lines = *) Image lines to be fit
(bands = *) Image bands to be fit
(type = **ratio**) Type of output
(replace= no) Replace rejected points by fit?
(wavesca= yes) Scale the X axis with wavelength?
(logscal= no) Take the log (base 10) of both axes?
(overrid= no) Override previously fit lines?
(listonl= no) List fit but don't modify any images?
(logfile= logfile) List of log files
(interac= yes) Set fitting parameters interactively?

(sample = *) Sample points to use in fit
(naverag= 1) Number of points in sample averaging
(functio= legendre) Fitting function
(order = **4**) Order of fitting function
(low_rej= **1.4**) Low rejection in sigma of fit
(high_re= **2.5**) High rejection in sigma of fit
(niterat= 20) Number of rejection iterations
(grow = 1.) Rejection growing radius in pixels
(markrej= no) Mark rejected points?
(graphic= stdgraph) Graphics output device
(cursor =) Graphics cursor input
ask = yes
(mode = ql)

21. Usuń pozostałe promienie kosmiczne. Użyj *splot*, funkcji *x* do wycinania kawałków widma, oraz *i* do zapisywania zmian w pliku. Pamiętaj, że każdą aperturę należy zachowywać osobno, inaczej zmiany nie zostaną zapisane.

22. Ustaw w irafie parametry obserwatorium Serra la Nave:

PACKAGE = noao
TASK = observatory

command = set Command (set|list|images)
obsid = sln Observatory to set, list, or image default
images = List of images
(verbose= no) Verbose output?

(observa= sln) Observatory identification
(name = SLN - Catania Astrophysical Observatory) Observatory name
(longitu= 345.02666666667) Observatory longitude (degrees)
(latitud= 37.691666666667) Observatory latitude (degrees)
(altitud= 1725.) Observatory altitude (meters)
(timezon= -1.) Observatory time zone
override= sln Observatory identification
(mode = ql)

23. Sprawdź, czy skróty nazw stosowane opracowywanych fitsach odpowiadają skrótom stosowanym przez irafa (keywpars), oraz popraw, co trzeba.

PACKAGE = rv
TASK = keywpars

(ra = RA) Right Ascension keyword
(dec = DEC) Declination keyword
(ut = ASTERUT) UT of observation keyword
(utmiddl= UTMIDDLE) UT of mid-point of observation keyword
(exptime= EXPTIME) Exposure time keyword

(epoch = EPOCH) Epoch of observation keyword
(date_ob= DATE-OBS) Date of observation keyword

(hjd = HJD) Heliocentric Julian date keyword
(mjd_obs= MJD-OBS) Modified Julian Date of observation keyword
(vobs = VOBS) Observed velocity keyword
(vrel = VREL) Relative velocity keyword
(vhelio = VHELIO) Heliocentric velocity keyword
(vlsr = VLSR) LSR velocity keyword
(vsun = VSUN) Epoch of solar motion keyword
(mode = ql)

24. Sprawdź jaka jest wartość prędkości radialnej β Oph i wpisz tę informację do pliku fits:
hedit beta_gem.0427_190327n VHELIO 3.3 add+ ver- (przykład dla β Gem)

25. Wyznacz prędkości radialne gwiazd programowych używając β Oph jako standardu:
PACKAGE = rv
TASK = fxcor

objects = HIP94898 List of object spectra
template= beta_Oph List of template spectra
(apertur= *) Apertures to be used
(cursor =) Graphics input cursor

(continu= both) Continuum subtract spectra?
(filter = none) Fourier filter the spectra?
(rebin = smallest) Rebin to which dispersion?
(pixcorr= no) Do a pixel-only correlation?
(osample= *) Object regions to be correlated ('*' => all)
(rsample= *) Template regions to be correlated
(apodize= 0.2) Apodize end percentage

(functio= gaussian) Function to fit correlation
(width = 150.) Width of fitting region in pixels
(height = 0.) Starting height of fit
(peak = no) Is height relative to ccf peak?
(minwidt= 3.) Minimum width for fit
(maxwidt= 21.) Maximum width for fit
(weights= 1.) Power defining fitting weights
(backgro= 0.) Background level for fit
(window = 300.) Size of window in the correlation plot
(wincent= INDEF) Center of peak search window

(output = rv) Root spool filename for output
(verbose= long) Verbose output to spool file?
(imupdat= no) Update the image header?
(graphic= stdgraph) Graphics output device

```
(interac=                yes) Interactive graphics?
(autowri=                 yes) Automatically record results?
(autodra=                 yes) Automatically redraw fit results?
(ccftype=               image) Output type of ccf

(observa=             obspars) Observation location database
(continp=                   ) Continuum processing parameters
(filtpar=                   ) Filter parameters pset
(keywpar=                   ) Header keyword translation pset

(mode   =                  ql
```

Zapisywanie widm (pliki typu fits) do pliku tekstowego:

1)

```
PACKAGE = ttools
   TASK = imtab

input   =        Altair_20050630_001224_normc  name of input image
outtable=        Altair_20050630_001224_normc.tab  name of output table
colname =            lam  name of column for image data
(pname  =            1) root for pixel position column name, or null
(wcs    =           world) coordinate system for pixels
(formats=                ) list of pixel coordinate formats
(tbltype=           default) row or column ordered table
(mode   =                al)
```

2)

```
PACKAGE = ttools
   TASK = tdump

table   =                Altair_20050630_001224_normc.tab  name of table to dump
(cdfile =           STDOUT) output file for column definitions
(pfile  =                STDOUT) output file for header parameters
(datafil=           output) output file for table data
(columns=                  ) list of columns to be dumped
(rows   =                  -) range of rows to print
(pwidth =                 -1) output page width
(mode   =                 al)
```

Wprowadzanie poprawki heliocentrycznej (pliki typu fits) :

1) Upewniamy się, czy w pliku keywpars kluczem do czasu uniwersalnego jest ASTERUT:

```
PACKAGE = rv
  TASK = keywpars

(ra    =              RA) Right Ascension keyword
(dec   =             DEC) Declination keyword
(ut    =        ASTERUT) UT of observation keyword
(utmiddl=      UTMIDDLE) UT of mid-point of observation keyword
(exptime=       EXPTIME) Exposure time keyword
(epoch =         EPOCH) Epoch of observation keyword
(date_ob=      DATE-OBS) Date of observation keyword
```

2) Obliczamy wartość poprawki heliocentrycznej, VHELIO:

```
PACKAGE = rv
  TASK = rvcorrect

(files  =                ) List of files containing observation data
(images = HD157881_20090712_225501.fits) List of images containing observation data
(header =              yes) Print header?
(input  =              yes) Print input data?
(imupdat=               no) Update image header with corrections?

(epoch  =           INDEF) Epoch of observation coordinates (years)
(observa=             sln) Observatory  <-- tu można wpisać też: observatory
(vsun   =             20.) Solar velocity (km/s)
(ra_vsun=             18.) Right ascension of solar velocity (hours)
(dec_vsu=             30.) Declination of solar velocity (degrees)
(epoch_v=           1900.) Epoch of solar coordinates (years)

(year   =                ) Year of observation
(month  =                ) Month of observation (1-12)
(day    =                ) Day of observation
(ut     =                ) UT of observation (hours)
(ra     =                ) Right ascension of observation (hours)
(dec    =                ) Declination of observation (degrees)
(vobs   =              0.) Observed radial velocity
(hjd    =                ) Helocentric Julian Day (output)
(vhelio =                ) Helocentric radial velocity (km/s) (output)
(vlsr   =                ) Local standard or rest radial velocity (km/s) (output)
(mode   =              ql)
```

Wynik działania programu:

RVCORRECT: Observatory parameters for SLN - Catania Astrophysical Observatory
latitude = 37:41.5
longitude = 345:01:36
altitude = 1725.

##YR MO DY UT RA DEC VOBS
##HJD VOBS **VHELIO** VLSR VDIURNAL VLUNAR VANNUAL VSOLAR

2009 7 12 22:53:12 17:26:14 2:06:12 0.0
2455025.45820 0.00 **-13.47** 3.97 -0.171 0.011 -13.313 17.438

3) Poprawiamy widmo na wyliczoną poprawkę:

PACKAGE = echelle
  TASK = dopcor

input = HD157881_20090712_225501.fits List of input spectra
output = HD157881_20090712_225501_rvcor List of output spectra
redshift= **-13.47** Redshift or velocity (Km/s)
(isveloc= yes) Is the redshift parameter a velocity?
(add = no) Add to previous dispersion correction?
(dispers= yes) Apply dispersion correction?
(flux = no) Apply flux correction?
(factor = 3.) Flux correction factor (power of 1+z)
(apertur=) List of apertures to correct
(verbose= no) Print corrections performed?
(mode = ql)

Wyliczone widmo, HD157881_20090712_225501_rvcor .fits, jest przesunięte na skali angstremów tak, by kompensować ruch Ziemi wokół Słońca. Nagłówek fits jednak nie jest zmieniony, co znaczy, że dla tego widma program rvcorrect policzyłby VHELIO=-13.47, a nie 0.