ASTR 4880/PHYS 5880 — Fall 2008 — Dr. Morrison

Lab 4, Due: December 15, 2008

1 Introduction

At the end of Lab 3, you produced two apertures of an extracted, flat fielded stellar spectrum, ready for wavelength calibration—the last step of data *reduction*—and continuum normalization, the first step of data *analysis*. Correction of the spectrum for telluric contamination is an intermediate step that will not be included here. After continuum normalization, you will be asked to proceed to measure equivalent widths and radial velocities for selected spectral lines.

2 Procedure for Wavelength Calibration

Before and after each stellar spectrum, Th-Ar comparison spectra were taken. Copy to your working directory the appropriate comparison spectra for your star, and process them as you did in Lab 3 for the stellar spectrum. Extract them in the same way as you did your stellar spectrum, using the flat lamp as a reference and doing no interactive treatment of the apertures. You should also insert the universal time in the image header as a `'UT'` keyword.

Open the first Th-Ar spectrum with *onedspec:identify*. With reference to the attached line identification charts, mark 4 lines spanning the spectrum. A suggested parameter list is given in Section 6. With the `section` parameter left blank, you'll be working on both apertures. In the graphics screen, keystrokes `'j'` and `'k'` will move you to the previous and the next aperture, respectively. Some tips on this task not covered in class:

- With the `'match'` parameter set to 0.1 or 0.2 (in Å), you don't need to type the complete wavelength; just round it to the nearest tenth. The task will find it in the line list. See the help for *identify* for explanation.
- Make an initial fit, then type `'y'`. With `maxfeatures` set to 10 or 15, you'll get only the 10 or 15 strongest lines. You'll still have to delete some and add a few by hand, since the strongest lines are not necessarily the best.
- The feature width is the width of the features across the base, not the full width at half maximum as erroneously stated in class. A wildly wrong feature width will lead to inconsistent results.

Open the second Th-Ar spectrum. In the graphics window, type `:read firstfile` where `firstfile` is the name of the image you just finished identifying. The task will use the line identifications you just made; you just need to make a new fit. Document your work with plots of the fitting function and the residuals. Don't forget to write your results to the database for both spectra.

To the image header of your stellar spectrum, add the keywords `refspec1` and `refspec2`. They are the names of your first and second Th-Ar spectrum, respectively. Now edit the parameters for the task *dispcor*, following the model in Section 6, and execute it. This task will apply the wavelength calibration derived from the Th-Ar spectra to your stellar spectrum, interpolating linearly in time between them. You now have a wavelength-calibrated stellar spectrum.

You still need to compute and apply the heliocentric correction. Open the package *astutil* and edit the parameters for the task *observatory* (see Section 6). Now edit the parameters for *rvcorrect* and run it to insert the keyword `VHELIO` into the image header of your stellar spectrum. Make a copy of the stellar spectrum and Doppler correct the copy in place with:

```
dopcor stard.hhh "" -vhelio isvel+
```

3 Procedure for Continuum Normalization

Edit the parameters for the task *continuum* following the example in Section 6. The '`*`' in the `lines` parameter will cause the task to operate on both apertures in succession. You can adjust the nature and order of the fitting function interactively.

Examine your graph carefully to make sure that the function that is fitted to the continuum follows the middle of the noise fluctuations in line-free regions of the spectrum. If necessary, restrict the fit to your designated samples with two '`s`' keystrokes. Document your work with screen shots of the fitting function with the `:.write file` command or the `:.snap epsl` command or another method. The graphics file produced by `:.write` can be displayed with the task *gkimosaic*.

4 Measurements

In your finished spectrum, measure the wavelengths, heliocentric radial velocities, and equivalent widths of the following lines, if they are present. Ignore any narrow water lines. Document your work with screen shots of the result of the '`e`' or the '`k`' key, as appropriate.

Atomic species	Wavelength (Å)
Si II	6347.091
Si II	6371.359
H I	6562.817
C II	6582.88
C II	6578.05

Type your star's name in the SIMBAD web site at:
`http://simbad.harvard.edu/simbad/sim-fid` and look up its radial velocity. Comment on the agreement or lack thereof between your value and the published value. In some supergiant stars, mass loss will cause the radial velocity derived from Hα to be different from that of the star as a whole. Looking up published equivalent widths for comparison is a more advanced project, but you may want to attempt it for extra credit.

5 Your Report

Everything in this section of the instructions for Lab 1 also applies to this lab. Always be sure to include enough figures and tables to enable the reader to replicate your work and diagnose errors, if any.

6 Sample Parameter Lists

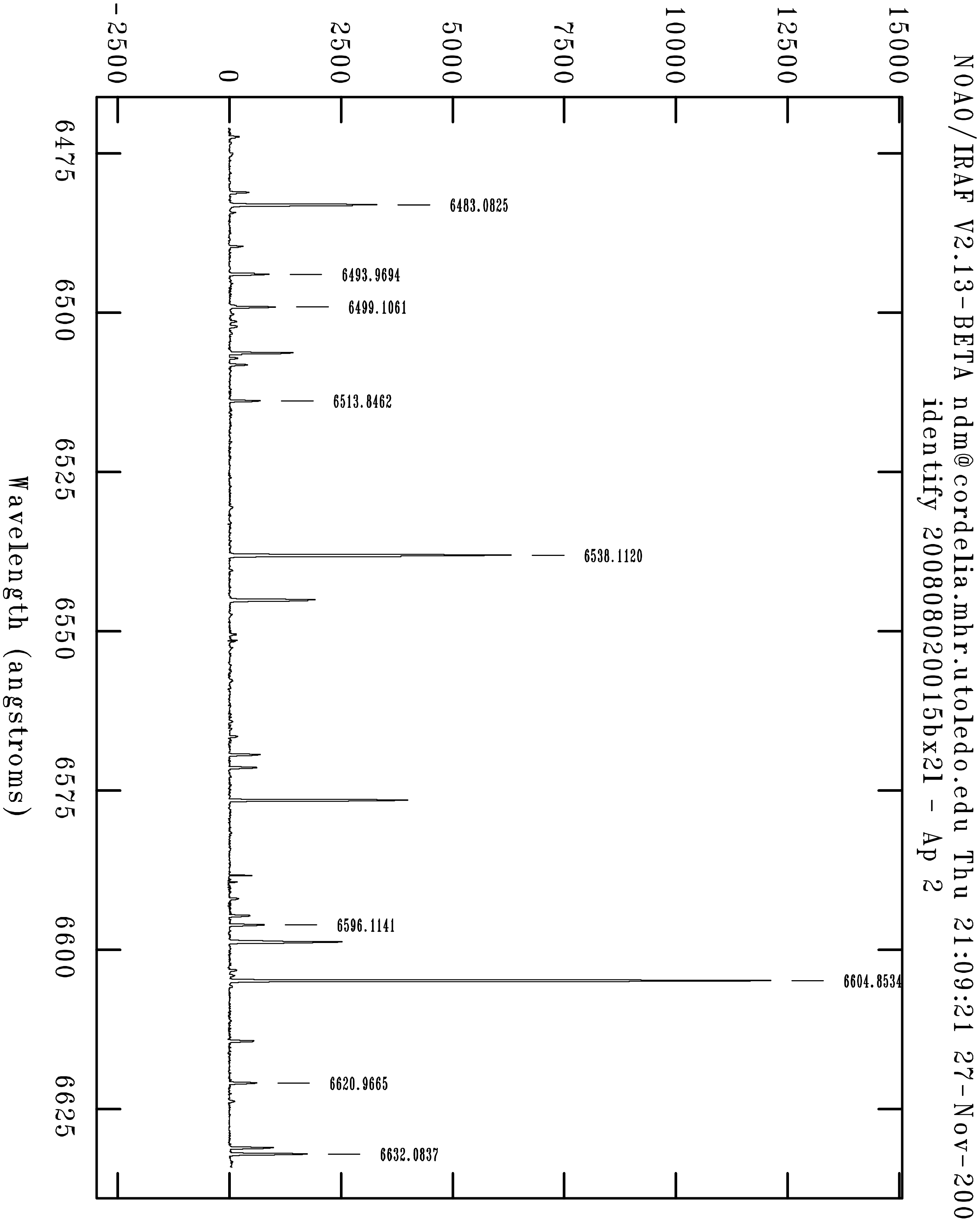
Some of these listings continue across page breaks.

```
PACKAGE = onedspec
   TASK = identify

images  =    200710220018x.hhh  Images containing features to be identified
(section=                     ) 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   =                    1) Number of lines/columns/bands to sum in 2D images
(match  =                  0.2) Coordinate list matching limit
(maxfeat=                   15) Maximum number of features for automatic identification
(zwidth =                 100.) Zoom graph width in user units
(ftype  =             emission) Feature type
(fwidth =                   7.) Feature width in pixels
(cradius=                  12.) Centering radius in pixels
(thresho=                   0.) Feature threshold for centering
(minsep =                   2.) Minimum pixel separation
(functio=            chebyshev) Coordinate function
(order  =                    4) Order of coordinate function
(sample =                    *) Coordinate sample regions
```

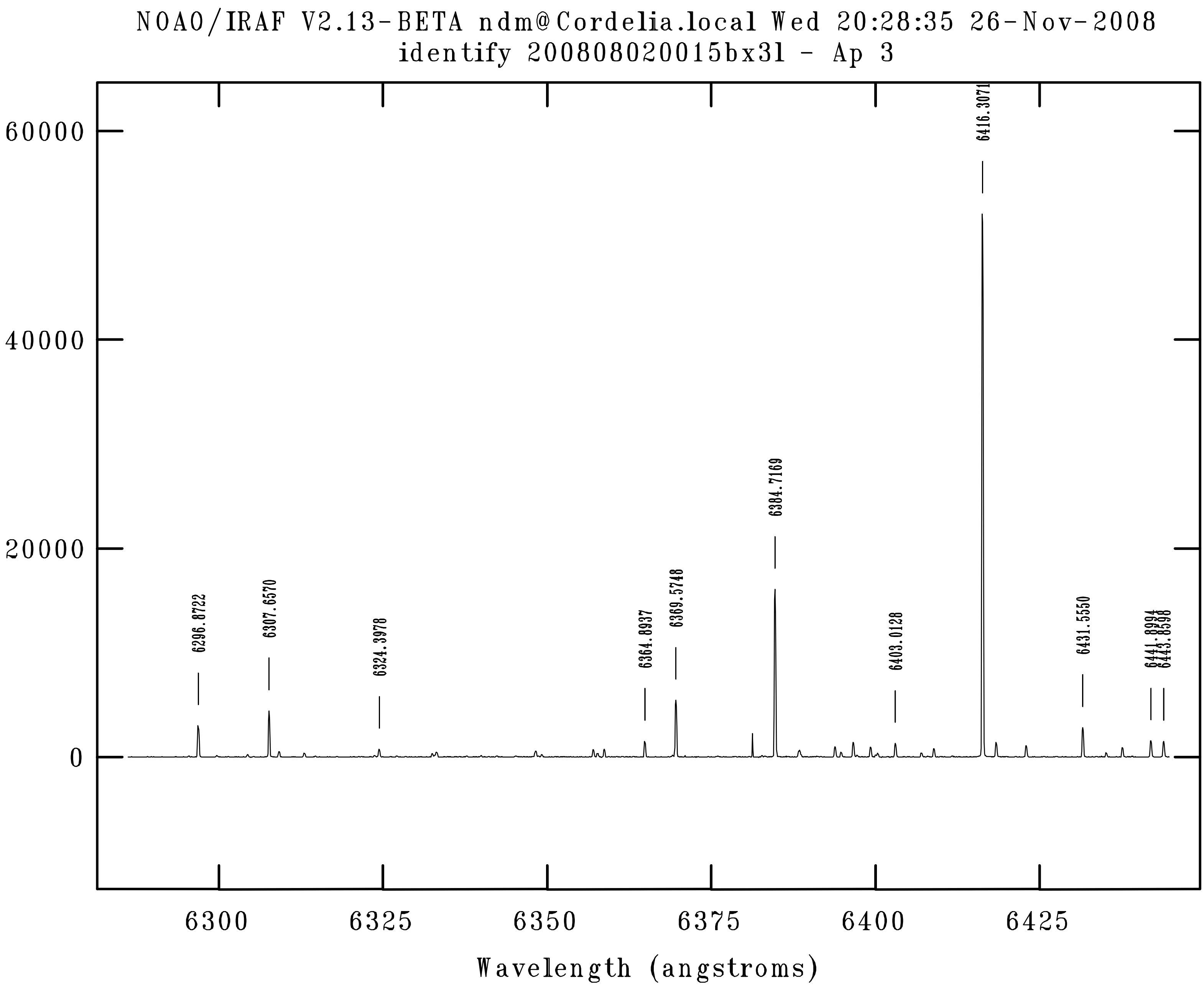
```
PACKAGE = onedspec
   TASK = dispcor

input   = 200710220019x2f2.hhh  List of input spectra
output  = 200710220019x2f2l2.hhh  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?
(samedis=                  no) Same dispersion in all apertures?
(global =                  no) Apply global defaults?
(ignorea=                 yes) Ignore apertures?
(confirm=                 yes) Confirm dispersion coordinates?
(listonl=                  no) List the dispersion coordinates only?
(verbose=                 yes) Print linear dispersion assignments?

PACKAGE = noao
   TASK = observatory

command =                 set  Command (set|list|images)
obsid   =             obspars  Observatory to set, list, or image default
images  =                      List of images
(verbose=                 yes) Verbose output?

(observa=             obspars) Observatory identification
(name   =              Ritter) Observatory name
(longitu=               83.61) Observatory longitude (degrees)
(latitud=               41.65) Observatory latitude (degrees)
(altitud=                100.) Observatory altitude (meters)
(timezon=                  5.) Observatory time zone
override=             obspars  Observatory identification
(mode   =                  ql)

PACKAGE = astutil
   TASK = rvcorrect

(files  =                    ) List of files containing observation data
(images =    200710220019x2f2) List of images containing observation data
(header =                 yes) Print header?
(input  =                 yes) Print input data?
```

```
(imupdat=                  yes) Update image header with corrections?

(epoch  =                INDEF) Epoch of observation coordinates (years)
(observa=              obspars) Observatory

[defaults follow]

PACKAGE = onedspec
   TASK = continuum

input   = 200710220019x2f2l.hhh  Input images
output  =   200710220019x2f2ln  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=                  yes) 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=              spline3) Fitting function
(order  =                    1) Order of fitting function
(low_rej=                   2.) Low rejection in sigma of fit
```

[etc.]



Aperture 1 for the purposes of this lab



Aperture 2 for the purposes of this lab