Notes on work

Daniel Thaagaard Andreasen

30. januar 2014

Indhold

1 To-do

Here comes my big to-do list:

1. Use IRAF to determine the EW of all the selected lines
2. Sort some of the lines out from the VALD list
   - Remove lines with EW below 1mÅ and above 250mÅ
3. Use the EW in MOOG to determine solar parameters
4. Determine stellar parameters for the Sun with line list in NIR and compare to literature
5. Do the same for FGK-dwarfs from the CRIRES-pub
6. Use the successful line list on FGK dwarfs and compare with literature
7. Start using the line list on M dwarfs!

2 Obtaining the solar data

The solar data was obtained from the homepages:

- ftp://vso.nso.edu/pub/atlas/photatl/
- ftp://vso.nso.edu/pub/atlas/niratl/

or via the terminal:

Listing 1: ftp solar atlas

```
wget -r -l 1 ftp://vso.nso.edu/pub/atlas/photatl/
wget -r -l 1 ftp://vso.nso.edu/pub/atlas/niratl/
```

to download the entire folder.

All the files are combined in the SolarSpectrumLL.py script, where a ASCII version and a FITS version were saved.

3 Obtaining the atomic line data

The atomic lines where obtained from the VALD database, http://vald.astro.univie.ac.at/~vald/php/vald.php. Here the mode *Extract element* were used.

Fe I:

- Wavelength: 7 300 − 54 200 Å.
- Element: Fe 1
- Extraction format: Short
- Retrieve data via: FTP
- Configuration: Default

Fe II: Same as above, except:

- Element: Fe 2

The downloaded files (.gz) where prepared for Python using the bash script VALDprepare.sh. For information how to use the script, see the source code.

4 Compiling the line list

Compiling a line list is a long and tedious process. The process is only described for Fe I since it is the same for Fe II.

1. Plot the solar spectrum in a small range (a small wavelength window makes it more faster to zoom and gives a better overview).
2. Overplot all the atomic lines in the same wavelength range.
3. Mark the wavelength in a separate file, where a spectral line appears with a atomic line for the whole wavelength range.
4. Repeat previous steps for the next wavelength window. I used a range of 0.05 Å.
5. Create a mask of the marked wavelength to all the atomic lines.
6. Use the mask to greatly reduce the numbers of atomic lines.
7. Save the masked atomic lines in a separate file with $\log gf$ and χ_{ext} – this is the line list.
8. Use IRAF to measure EW of all the saved lines from above. Some is blended lines, so here I use the *d* function in IRAF to fit several lines. For every line I add a Greatness Of Fit (GOF) ranging from 0-5, where 0 means it is not a line/not a Fe line, and 5 means it is a great line.
9. Use /programs/scripts/splotCommenter.sh to make the output from splot in IRAF ready for Python and np.loadtxt. This script is also called by functions inside LLpicker.py.

4.1 Recalculate $\log gf$ with MOOG

1. Pick one line at the time
2. Change the $\log gf$ untill the abundanc is the given value (7.47 for the Sun)

5 IRAF

5.1 Install IRAF

5.2 Use IRAF to measure EW

Here is a short guide how to use IRAF to measure EW's:

1. Start xgterm.fedora (Alt+F2) or from terminal
2. Type cl in home directory to start IRAF
3. In IRAF type:
   (a) noao
   (b) imred
   (c) echelle
   (d) cd /path/to/fitsfile/

(e) splot file.fits

4. **Useful commands:**

   ?: To get a help text

   a: Put cursor on left side, press a and then do the same on the right side to zoom in. To zoom out (a little) put the cursor outside the plot and press a. To zoom all the way, press a two times at the same place. Only the place on x-axis matters when zooming

   k: Fit one gaussian. Place cursor on continuum on left side and press k, and do the same on the right side

   d: For deblending and fit with multiple gaussians, voigt's profiles etc. See the text in the bottom for this.

   p: From pixel values to wavelength values on the x-axis. This values will be saved so only have to look up the beginning and end of the wavelength range the first time.

6 MOOG

> MOOG is a code that performs a variety of LTE line analysis and spectrum synthesis tasks. The typical use of MOOG is to assist in the determination of the chemical composition of a star.

More info can be found at: http://www.as.utexas.edu/~chris/moog.html.

6.1 Install MOOG 2010

Basically follow the installation guide on the webpage. Here it is:

1. Edit Begin.f: 'pcl' for linux system and right path for the MOOG installation.
2. Edit Makefile.rh64: Write the right PATH's. For SuperMongo (sm) use /opt/sm/lib
3. Compile with: make -f Makefile.rh64. Output is MOOG

6.2 Install MOOG 2013

Same steps as above, but for some reason it currently doesn't work in linux mint 14.

6.3 Install SuperMongo (sm)

To install sm simple follow the README file. It is suggested to install it in /opt/sm/

1. ./set_opts (nothing special here other than remember to type X11 when asked for).
2. Delete the first lines in the file copyright.h in other to compile
3. make; make install

7 ARES

I wonder if ARES works for this?

7.1 Install ARES

Simple follow the README file.

8 Developed software

All the glorious software developed for the above purposes.

8.1 Python

All the tasks are done in Python with call to small scripts outside

8.1.1 SolarSpectrumLL.py

This is the main file. It performs several task well explained by comments.

1. All the ASCII data for the solar spectrum is combined to a single ASCII and one fits file. Only need to do this one as it is saved locally on the harddisk.
2. Read the VALD data and store it in e.g. FeI.
3. Make a line list by using a mask on the above VALD data and save the result.
4. Create a MOOG file.
5. The last part of the file is for plotting purposes.

8.1.2 LLpicker.py

This does all the task of saving picking lines and creating e.g. the MOOG file.

picker: Program to pick lines from VALD data base compared to wavelength manually choosen using a minimazation method.

saver: Small function that saves the VALD data to a final line list (without any EW).

sorter: Sort the linelist for double lines and "Not a line"-lines.

IRAFreader: Program that reads the splot log-file from IRAF and returns the center of wavelength (from fit) and the EW. This function also comments the splot log-file proberly.

MOOGcreator: Program that 'merge' the splot file and own LL from VALD to the MOOG readable format (where spaces are important). This function is called from the main file and then calls the IRAFreader, why this function also comment the splot log-file.

8.2 bash scripts

The small bash scripts to make life easier.

8.2.1 splotCommenter.sh

This function comments the splot log-file with '#' everytime there is a 'center'. Just call it with the file to comment as the first parameter. No commenting is done if there already is a '#'.

8.2.2 VALDprepare.sh

This small script unzip the VALD data downloaded via FTP from their website and create two files. A data file (.dat) ready to use for Python and a file with references (.ref). Call the function with the file to be unzipped and the output-name without any extension.