

BACCHUS

Brussels Automated Code
for Characterizing High accuracy Spectra

Manual

Thomas Masseron, Thibault Merle and Keith Hawkins

November 14, 2018

1 Introduction

The BACCHUS code [1] is intended to automatically perform 1D LTE stellar spectroscopy for one to thousands late-type stars. Three modes are possible:

- simple elemental abundance analysis (*bacchus.abund* module)
- precise equivalent width measurements (*bacchus.eqw* module)
- determination of stellar parameters (*bacchus.param* module)

The BACCHUS code is based on:

- the 1D LTE radiative transfer fortran code Turbospectrum-v15.1 [2]
- the MARCS model atmosphere grid [3]
- the model atmosphere interpolation fortran code [4]
- cshell scripts
- the SuperMongo plotting software
(<http://www.astro.princeton.edu/~rhl/sm>)

2 Installation

The package contains binaries compiled for Linux with gfortran and should work straight away after unpacking from the unpacked directory. As a test for successful installation, go into the BACCHUS directory, and run the command:

```
bacchus.param Sun
```

If all goes fine, all the output in the *Sun/* directory should be identical to the one in the example included in the *examples* directory.

However, if you need to compile again, here are the few files to check:

- Install Turbospectrum: go to *Turbo-v15.1/EXPORT-15.1/exec-gf-v15.1*, remove all files except the Makefile, edit it, and run the `make` command.
- In the *INTERPOL* directory compile *interpol_modeles.f*.
- In the *bin* directory compile *showline.f* and *faltbo3.f*.

Some bug may occur when merging ps files and screw the pdf files (either they are blank pages, figures are shrunk or overlap, ...), depending on your system. In that case you need to uncomment another option for merging ps in the routine *bacchus.abund* and *bacchus.param* (at two locations in this latter script).

There may also be some incompatibility for installation on Mac, especially concerning the Supermongo routines. Although there may be some minor changes to fix it (you can contact Keith Hawkins), it would be more advantageous to forget about Mac.

3 How to...

The code consists in 3 main modules (*bacchus.abund*, *bacchus.eqw* and *bacchus.param*) that can be run independently. However, BACCHUS is built with a bottom-up scheme: the *bacchus.eqw* module runs the *bacchus.abund* one, and the *bacchus.param* module runs the *bacchus.eqw* one.

- **Prepare a new star**

Fill up the *stellar_parameters.tab* with basic parameters and the location of the observation file. The observation file should contain at least 2 columns in ascii, $\lambda(\text{air})$ in Å flux in whatever unit, and optionnally the variance of the spectrum in the same unit as the flux. The spectrum should be radial velocity corrected or the radial velocity value must be provided. An initial *<star>.par* file will be created by running:

```
load_parameters.com Sun
```

- **Do abundance analysis for an element**

knowing the model atmosphere (the *<star>.par* file must exist):

```
bacchus.abund Arcturus Mg
```

```
bacchus.abund Arcturus Mg 6.606 6.906 7.206 7.506 7.806
```

The results of these examples are included in the *examples/Arcturus* directory.

- **Measure the equivalent widths of an element**

knowing the model atmosphere (the *<star>.par* file must exist):

```
bacchus.eqw Arcturus Si
```

The results of this example are included in the *examples/Arcturus* directory.

- **Determine microturbulence, convolution and metallicity**

with fixed T_{eff} and $\log g$: set **TURBVELunknown**, **CONVOLunknown** and **METALLICunknown** variables to 1 in the *init.com* file and then run:

```
bacchus.eqw Sun Fe
```

The results of this example are included in the *examples/Sun* directory.

- **Do the determination of the atmospheric parameters**

- (T_{eff} , $\log g$, $[\text{Fe}/\text{H}]$ and ξ_t) from a spectrum: fill up the *stellar_parameters.tab* with some guess of the initial parameters and then run: `bacchus.param Sun`

The results of this example are included in the *examples/Sun* directory.

- (T_{eff} , $[\text{Fe}/\text{H}]$ and ξ_t) but with fixed $\log g$:

fill up the *stellar_parameters.tab* with some guess of the initial parameters, set **LOGGunknown** variable to 0 in the *init.com* file and then run `bacchus.param Sun`

- **Switch to semi-manual mode**

The basic idea is that you can edit and modify all the text files in the star directory and the code will take the edited values. But do not forget to switch the appropriate variables so that the edited files are not removed. Examples: To adjust the continuum placement, edit the *<element>-<star>.plt* file. To use your own model atmosphere and/or abundances, edit the *<star>.par* file.

- **Change the line selection**

Edit the *elements.wln* file (or the *elements_MP.wln* file for metal-poor stars, i.e. $[\text{Fe}/\text{H}] < -1.5$). Make sure that the wavelength coverage of your selected lines are encompassed by the linelists in the *linelists* directory and used in the *bsyn.com* file.

- **Change/update the atomic and/or molecular linelists**

Edit the *bsyn.com* script. This package contains a limited number of atomic ([4200–9200] Å from VALD [6]) and molecular (CH, C₂, CN [7, 8, 9, 10, 11]) linelists for data volume reason. More linelists are available upon request but would potentially need citation.

- **Diagnose a bug**

Set the **debug** variable to 1 and run the code again.

4 Structure

4.1 Directory tree

The structure is very similar to the structure of Turbospectrum (called here TBS).

```
BACCHUS _
|_ DATA (data files for TBS)
|_ linelists (atoms and molecules)
|_ plots _
|       |_ sm _ (.sm default rainbow.dat rainbow.lut)
|       |_ ...
|_ eqwidt (symlink to /tmp/eqwidt: equivalent widths files from TBS)
|_ syntspec (symlink to /tmp/syntspec: synthetic spectra from TBS)
|_ contopac (symlink to /tmp/contopac: continuum opacity files from TBS)
|_ models _ (contains MARCS model atmosphere grid)
|       |_ MARCS_st_ppl_t01_mod
|       |_ MARCS_st_sph_t02_mod
|       |_c+0.00_
|               |_a-1.00
|               |_a-0.75
|               |_a-0.50
|               |_a-0.25
|               |_a+0.00
|               |_a+0.25
|               |_a+0.50
|               |_a+0.75
|               |_a+1.00
|       |_c-1.00_
|               |_a-1.00
|               |_a-0.75
|               |_a-0.50
|               |_a...
|       |_c...
|_ INTERPOL (interpol_planparallel.com interpol_spherical.com interpol_modeles)
|_ bin (showline and faltbo3)
|_ Turbo-v15.1 _ (TBS directory)
|       |_ EXPORT-15.1 _
|       |_ ...
|_ examples _
|       |_ Arcturus _
|               |_ models
|       |_ Sun _
|               |_ models
|_ <star> _ (all output for star <star>)
|       |_ models
```

4.2 Inputs/outputs

Input	module	arguments <i>[optional]</i>	Output
· stellar_parameters.tab	load_parameters	<star>	-<star>.par
· <star>.par · init.com · elements.wln or · <element>-<star>.plt	bacchus.abund	<star> <element> [<log(A1) log(A2) log(A3)...]	·<element>-<star>.abu ·<element>-<star>.plt ·<elet>-<star>_<model>.pdf (Fig. 1)
·<star>.par or · stellar_parameters.tab · init.com ·<element>-<star>.abu ·<element>alllines.list or linelist_ref	bacchus.eqw	<star> <element> [<log(A1) log(A2) log(A3)...]	·<element>-<star>.eqw ·<element>-<star>.list ·<star>_<model>_<element>eqw.pdf (Fig. 2) ·<element>guess.tab ·<element>alllines.list
·<star>.par or · stellar_parameters.tab · init.com ·<element>-<star>.list	bacchus.param	<star> [$T_{\text{eff} \textit{init}}$ $\log g_{\textit{init}}$ $[M/H]_{\textit{init}}$ $\xi_{\textit{init}}$]	·<model>.eqwidth · modelfit<i>_<star>.tab · fixedmicroturb_parameters.tab · best_parameters.tab ·<star>_<element>param.pdf (Fig. 3)

4.3 Program files

- *bacchus.abund*: module to derive abundances for a given model atmosphere.
- *bacchus.eqw*: module to derive equivalent widths, but also possibly microturbulence, convolution parameter and metallicity. This script needs the <element>-<star>.abu file and call *bacchus.abund* if necessary
- *bacchus.param*: module to derive stellar parameters (T_{eff} , $\log g$, $[\text{Fe}/\text{H}]$ and ξ_t). This script needs the <element>-<star>.list file and will call *bacchus.eqw* if necessary.
- *load_parameters.com*: routine to generate the basic stellar parameters file.
- *bacchus_fast.abund* (not currently running)
- *bacchus_fast.param* (not currently running)
- *babsma.com*: subscript called by *bacchus.abund* for generating continuum opacity file.
- *bsyn.com*: subscript called by *bacchus.abund* to generate synthetic spectrum; it notably contains the linelists.
- *faltbo3.f*: convolve the output spectrum.
- *eqwidt.com*: subscript called by *bacchus.eqw* to generate the synthetic equivalent widths.
- *interpol_spherical.com*: subscript called by *load_parameters.com*, *bacchus.eqw* and *bacchus.param* to generate model atmosphere in spherical geometry.

- *interpol_planparallel.com*: subscript called by *load_parameters.com*, *bacchus.eqw* and *bacchus.param* to generate model atmosphere in plan parallel geometry.
- *interpol_modeles.f*: Atmosphere interpolation code.
- *showline.f*: subroutine to show on the plots lines identification from either Sun or Arcturus.

SuperMongo graphic and data process files:

- *.sm*: SuperMongo options.
- *synspec.sm*: make the individual lines figures and determine abundances.
- *equidit.sm*: make excitation potential and equivalent widths trends figures and compute microturbulence, metallicity and macroturbulence.
- *best_model.sm*: compute stellar parameters with fixed microturbulence.
- *rvcorrect.sm* (*not working*): compute radial velocity by CCF.

Data files:

- *solabu.dat*: contains the solar abundances assumed.
- *init.com*: contains default variables and their values.
- *elements.wln*: each line contain the selected wavelengths of the element to be measured.
- *elements_MP.wln*: each line contain the selected wavelengths of the element to be measured for metal-poor stars ($[Fe/H] < -1.5$).
- *Moore*: Sun line identification.
- *Arcturus_lines.list*: Arcturus line identification.
- *stellar_parameters.tab*: table containing name, observed spectrum and initial stellar parameters.

4.4 Output files

- *<star>/*: directory where the output files corresponding to the star are stored.
- *<star>.par*: parameter file containing all specific variables of the *<star>*. All variable values in this file supersede the ones in the *init.com* file.
- *<element>_alllines.list*: This file contains the line physical information for all the lines in the *elements.wln* and *element_MP.wln* files. It is extracted from the original linelist as pointed by the **linelist_ref** variable. It is meant for saving some computational time by preventing for searching for the line information all throughout the entire linelist file at each iteration and for each star. However, if the variable **alllines_list** is set to 1, the file is removed and reconstructed each time the code is run. We advise to leave it to 1 to prevent inconsistency with the linalists.

4.4.1 Tables

1. *<element>-<star>.plt*: contains the values used for the spectrum fit. The first line contains the list of lines to be fitted. If the variable **reset** is set to 0, then those wavelengths are taken instead of the ones listed in the *element.wln* and *element_MP.wln* files. The columns are: input line wavelength, slope value for the linear regression of the continuum points, scale factor of the normalisation of the observation to the continuum points, zooming value for the y axis of the bottom panel, lower and upper wavelength of the selected points for the abundance determination, wavelength range of the bottom panel, and flag for the continuum normalisation (0 for no continuum points, hence adjust to max flux points, 1 for linear regression to continuum points, 2 for computing only the mean of continuum point). If the variable **reset=0**, then the values for continuum placement in this file are adopted.
2. *<element>-<star>.abu*: contains the derived line-by-line abundances. The first line shows the parameters used for the synthesis (model, metallicity, microturbulence, convolution and list of log(A)). Columns are: input line wavelength, observed equivalent width, log of abundance derived by the synthesis method (“syn”) and corresponding flag, log of abundance derived by the equivalent width method (“eqw”) and corresponding flag, log of abundance derived by the intensity method (“int”) and corresponding flag, log of abundance derived by the χ^2 method and corresponding flag, value of the χ^2 , estimated SNR, and estimated value of the upper limits for the “syn”, “eqw” and “int” methods. If a *<element>-<star>.abu* file already exists, but with different synthesis parameters, it will be moved to a *<element>-<star>_<formermodel>.abu* file.
Meaning of the flags for the 4 abundance determination techniques:
 - syn: flags:0=upper limit; 1=ok; 2=extrapolation; 3= suspicious large offset between the observed min and observed synthesis (likely caused by a bad fitted strong blend) the condition for upper limit is: $A_{\text{syn}} \leq 3/\text{snr} \cdot \sqrt{n_{\text{syn_points}}}$.
 - eqw: 0=upper limit ; 1=ok; 2=extrapolation; 3= suspicious large offset between the observed min and observed synthesis (likely caused by a bad fitted strong blend) the condition for upper limit is $\text{eqw}_{\text{obs}} \leq 3/\text{snr} \cdot \sqrt{n_{\text{obs_points}}}$
 - int: 0=upper limit; 1=ok; 2= suspicious large offset between line cores OR strong line (i. e. line intensity is below 0.4 of the continuum) the condition for upper limit is: $\text{int}_{\text{obs}} \leq 3/\text{snr}$
 - χ^2 : 0=beyond range of abundances; 1=ok ; 2=method crashed
3. *<element>-<star>.eqw*: contains line-by-line abundances and line information after line selection. The first line shows the parameters used for the synthetic equivalent widths computation (model, metallicity, microturbulence, and convolution). The columns are: element, ionisation stage, wavelength as provided in the linelist, excitation potential, log(gf), computed equivalent widths, observed pseudo-equivalent width, error in observed equivalent width

and log abundance of the line. If a *<element>-<star>.eqw file* already exists, but with different synthesis parameters, it will be moved to a *<element>-<star>_<formermodel>.eqw file*. Otherwise, if a file exists and match the parameters, the equivalent widths computation will be skipped.

4. *<element>-<star>.list*: similar to *<element>-<star>.eqw*, but in Turbospectrum linelist format ready for stellar parameter determination for *bacchus.param*. If this file exists, *bacchus.eqw* is skipped.
5. *<element>guess.tab*: contains the successive values for metallicity, microturbulence and convolution estimates at each iteration when their determination is requested with the **METALLICunknown**, **TURBVELunknown**, and **CONVOLunknown** variables. Columns are: metallicity convergence flag, output metallicity from individual lines mean and its standard deviation, input metallicity, microturbulence convergence flag, output microturbulence (in km/s) and its standard deviation, input microturbulence, convolution convergence flag, output convolution (in km/s) and its standard deviation, and input convolution. The number of iterations is limited to 10.
6. *<model>.eqwidth*: output abundances for each *<model>* as derived with Turbospectrum from the equivalent widths provided in the *<element>-<star>.list* file. Columns are: line wavelength (Å), excitation potential (eV), log(gf), equivalent width (mÅ)x2, error on equivalent width, delta abundance, derived abundance (x3). If this file exists, the code will skip the computation.
7. *modelfit<i>_<star>.tab*: for a set of six models with various combination of T_{eff} s and loggs, it contains the values of the abundances against excitation potentials and equivalent widths trends as well as the ratio of the ionised over neutral lines abundance values. Columns are: model number, T_{eff} , logg and microturbulence as input values, output metallicity from neutral lines, error, output metallicity from ionised lines, error, value of the slope from the abundances against excitation potentials, error, value of the slope from the abundances against equivalent widths, error, intercept value of the abundances against excitation potentials, intercept value of the abundances against equivalent widths, rms of linear regression of the abundances against excitation potentials, rms of the linear regression of the abundances against equivalent widths, and the mean metallicity of neutral lines with similar equivalent widths as the ionised lines.
8. *fixedmicroturb_parameters.tab*: contains the successive values of the stellar parameter determination iteration with fixed microturbulence resulting from the *modelfit<i>_Sun.tab* table analysis. Columns are: convergence flag, T_{eff} solution and error from method 1, T_{eff} solution and error for method 2, logg solution and error, metallicity solution and error, microturbulence solution from method 1 and error, microturbulence solution from method 2 and error, and previously determined T_{eff} , logg, metallicity and microturbulence. The number of iterations is limited to 10.
9. *best_parameters.tab*: contains the successive values of the stellar parameter determination iteration with microturbulence as a free parameter. Columns are:

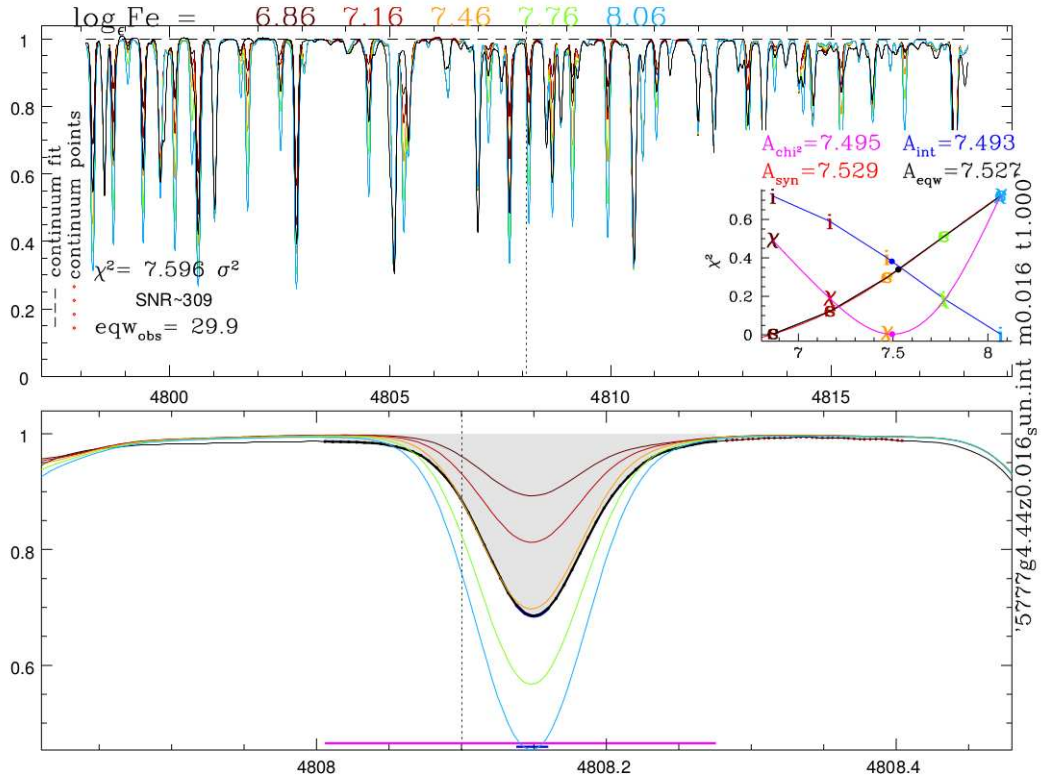


Figure 1: Example of *bacchus.abund* module output figure. In this example, the Fe line is best fitted by a synthesis lying between the orange synthesis (corresponding to $\log(\text{Fe})=7.46$) and the green synthesis (corresponding to $\log(\text{Fe})=7.76$). Indeed the 4 automatic method of abundance determination indicate values ranging between 7.495 and 7.527.

convergence flag, T_{eff} solution and error, input T_{eff} , convergence flag for T_{eff} , $\log g$ solution and error, input $\log g$, convergence flag for $\log g$, output metallicity and error, input metallicity, microturbulence solution and error, input microturbulence and convergence flag for microturbulence. The number of iterations is limited to 10.

4.4.2 Figures

1. `<elet>-<star>_<model>.pdf`: **output from *bacchus.abund* module.**

For each page, it shows the result of individual line fitting and abundance determination (see Fig. 1). The figure is divided in 3 panels: the upper panel shows the observed normalised spectrum (black) and the synthesis obtained with various abundances (colour) as detailed on the top. The long dashed line trace the continuum fit over the continuum points (red dots). The short-dashed line(s) indicates the input wavelength to be studied. Also indicated the estimated SNR, the reduced χ^2 of the line best fit, and the pseudo-equivalent width value.

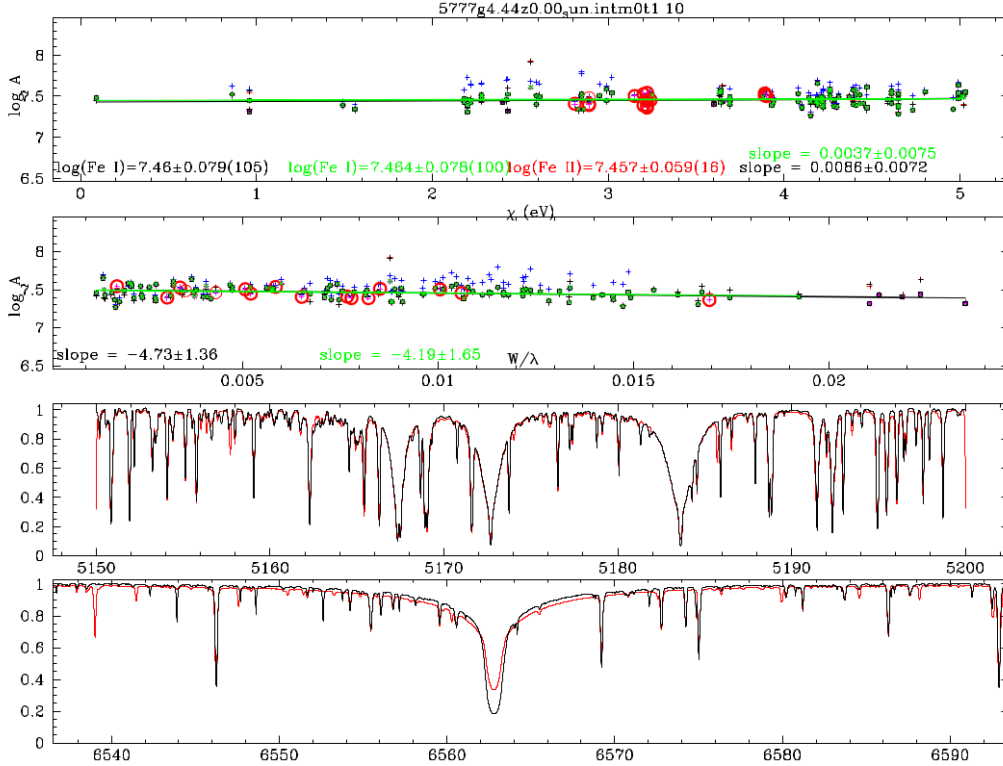


Figure 2: Example of *bacchus.eqw* module output figure. While the abundance against the excitation potential show no trend (upper panel) testifying for a correct input T_{eff} , the trend for abundance against equivalent width is significant (second panel), indicating that the input microturbulence is too large. Moreover, the blue “+” symbols are offset compared to the “+” black symbols indicating that the convolution parameter is too large.

The bottom panel shows the same normalised observed spectrum and synthesis as in the upper panels but zoomed in the line of interest. The short-dashed line(s) indicate the input wavelength. The horizontal magenta show the selected region for the “ χ^2 ” and the “syn” abundance determination methods while the blue line indicate the region for the “int” method. The points for the abundance determination are shown by a thickened line on the observed spectrum. The shaded area represents the pseudo equivalent width. The blue parabolic line indicate the points used for the fit of the “intensity” method. The short-dashed line indicates the input wavelength. Continuum points are indicated by red dots.

The third panel -inserted in the upper panel- shows the diagram for the 4 methods for the abundance determination. Each of the method is represented by a different colour (χ^2 : magenta, synthesis: red, equivalent width: black and intensity: blue) and by a letter for each input abundance (χ^2 : “ χ ”, synthesis: “s”, equivalent width: “e” and intensity: “i”). The final interpolated

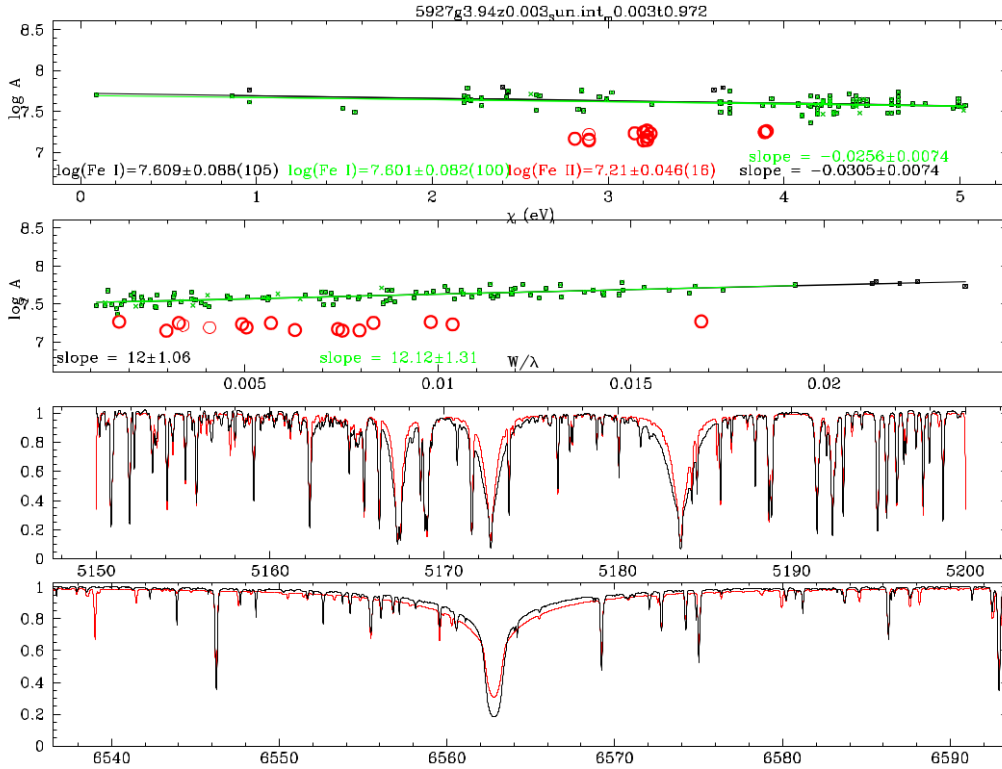


Figure 3: Example of *bacchus.param* module output figure. The abundances against the excitation potentials diagram (upper panel) show a negative trend testifying for a too high input T_{eff} , while the ionised lines (red circles) are underabundant implying that gravity is too low. Those two statements are confirmed by the visual inspection of the two bottom panels where Mg-triplet lines wings are too weak and the H α line wings are too strong. Moreover, the abundances against equivalent widths show a positive trend (second panel) implying that microturbulence is not strong enough.

abundance value for each method is indicated by a dot. Finally, on the right hand side is mentioned the model used.

2. `<star>_<model>_<element>eqw.pdf`: **output of the *bacchus.eqw* module.**

It shows the individual line abundance points against their excitation potential (upper panel of Fig. 2) and against their reduced equivalent widths (second upper panel). "+" signs stand for the abundances obtained with the different methods from the *bacchus.abund* module (χ^2 : magenta, synthesis: red, equivalent width: black and intensity: blue). The green crosses represent the selected abundance points for neutral lines, while the red circles are for ionised lines. Thick symbols highlight unblended lines. The green line is the linear regression over the points with reduced equivalent width is less than `eqw_rej` variable, while the black line is the fit over all selected points. The respective

mean and slope values are indicated as well as their respective errors.

The two bottom panels show an illustration of the spectrum (black) against the synthesis (red). These windows are just indicative and are not used by the code to constrain parameters. The wavelength range of these panels can be set by the **lam_min1**, **lam_max1**, and **lam_min2**, **lam_max2**. On top of the figure is mentioned the stellar parameter used.

3. `<star>_<element>param.pdf`: **output of the *bacchus.param* module.**
It is essentially the same as the `<star>_<model>_<element>eqw.pdf` figure, except that each page is for a different model atmosphere (see Fig. 3). Note that the "+" signs are not shown as in the `<star>_<model>_<element>eqw.pdf` figure because the displayed abundance values are computed from the synthetic equivalent width, and not from the line fitting.

5 Variables

Here are the minimum variable names to be set in the `<star>.par` file:

- **star**: name of the star. This name must be consistent with the one reported in the `stellar_parameters.tab` file. The subdirectory all files will show that name. Warning: For the star name the script does not handle properly some characters like "+".
- **MODEL**: model atmosphere filename.
- **MARCSFILE**: `.true.` for MARCS web or binary format of the atmosphere model and `.false.` for interpolated format.
- **SPH**: T or F for reciprocally spherical or plan parallel treatment of the radiative transfer. The model atmosphere must correspond.
- **mpath**: directory where the model atmosphere is stored. By default it is `<star>/models/`.
- **METALLIC**: metallicity of the star.
- **TURBVEL**: microturbulence value in km/s.
- **obs_spectrum**: filename and location of the observed spectrum.

Below the list of variables set by default in the `init.com` file. This file may be edited/adapted depending on the spectrograph specifications (i.e. instrument resolution, wavelength coverage) and the analysis type (i.e. differential analysis, non-solar scaled abundances, line strength cut-off).

All variables default values are superseded if set in the `<star>.par` file.

- **turbo_path**: Turbospectrum executables directory
- **util_path**: utility programs directory (showline interpol_modeles)
- **interpol_path**: interpolation script directory

- **dotsm**: location of the SuperMono special macros file
- **solabuf**: filename of the solar abundances
- **wavereff**: filename containing the wavelengths of the element to be plotted and measured
- **linelist_ref**: file name containing the linelists
- **alllines_list**: 1 to systematically remove the <element>_alllines.list. 0 to keep the file. It is advised to keep it to 1, but can be set to 0 to speed up processing of large number of stars.
- **profile_conv**: convolution profile value (1=EXP, 2=GAUSS, 3=RAD-TAN, 4=ROT)
- **convol_inst**: value of the instrumental resolution in km/s. It MUST be always negative. This value is used as a lower limit for the convolution parameter determination.
- **rv**: radial velocity in km/s
- **SNR**: mean signal-to-noise value. This value is used to determine the continuum points and the upper/lower limits of the abundances. If this variable is not declared or is negative, the code will evaluate it for each plotted window. This value is superseded if the variance of the spectrum is provided in the third column of the observed spectrum.
- **diff_star**: reference star name for line-by-line abundance differential analysis; the program will look for the corresponding <star>/<elet>-<star>.eqw file the reference star. Set it to nothing or to a dummy name for not running it.
- **diff_star_offset**: contains the element reference abundance value ($\log(A)$) of the reference star (e.g. 7.45 for the Sun's Fe abundance)
- **alpha**: alpha element enhancement to be considered in the synthesis; by default the Galactic one is assumed (i.e. 0.0 for $[\text{Fe}/\text{H}] \geq 0.0$; -0.4 for $[\text{Fe}/\text{H}] \leq -1.0$; linearly increasing for $0.0 \geq [\text{Fe}/\text{H}] \geq -1.0$;
- **s_process**: enhancement of the s-process element to be considered in the synthesis. The s-process fraction of each element considered is written in the solar abundance file.
- **r_process**: enhancement of the r-process element to be considered in the synthesis. The r-process fraction of each element considered is written in the solar abundance file.
- **ncpu**: max number of processors to be used in parallel (correspond to one per abundance). WARNING: it is neither openMP nor MPI parallelisation. It just runs the different synthesis in the background. Hence, do not use values > 1 while running on shared computer like supercomputer or Condor. Alternatively, for supercomputer use, set to "ncup" to 1 and submit 1 job per star.

- **compute:** 0 if you want to just redo the plot. 1 will force the synthetic spectrum computation.
- **reset:** 0 will keep *<element>-<star>.plt* file. 1 will delete it.
- **nonorm:** 1 if you do not want the observed spectrum normalised; 0 otherwise
- **manual:** for on screen continuum point selection (not working; instead modify the continuum values in the *<element>-<star>.plt* files and run again with **reset=0**).
- **rvcorrect:** compute the radial velocity from CCF of the local region (disabled=0 because very unstable)
- **cleanup:** 1 to remove the synthesis files in the *syntspec*, *contopac* and *eqwid* directories. 0 if you want to keep the synthetic spectrum and the observed spectrum. Warning: if set to 0, your disk can be quickly filled up.
- **debug:** 1 to diagnose crash or bug. All the output will then be displayed on the screen. 0 for a minimum screen output.
- **eqw_rej:** limit of the reduced equivalent width value for computing the microturbulence determination trend
- **TEFFunknown:** 0 to fix the T_{eff} value when executing *bacchus.param*.
- **LOGGunknown:** 0 to fix the *logg* value when executing *bacchus.param*.
- **METALLICunknown:** 1 to compute the metallicity. The *<star>.par* file is updated accordingly.
- **TURBVELunknown:** 1 to compute microturbulence velocity. The *<star>.par* file is updated accordingly.
- **CONVOLunknown:** 1 to compute the convolution parameter. The *<star>.par* file is updated accordingly.
- **updateabu:** 1 will write in the *<star>.par* file the mean abundance of the non-flagged lines. However the 0 option and manual edition is generally more suitable for double check.
- **norm_Mg:** scaling factor for the observed spectrum in the upper window of *<star>_<model>_<element>eqw.pdf* and *<star>_<element>param.pdf* figures
- **norm_Ha:** scaling factor for the observed spectrum in the lower window *<star>_<model>_<element>eqw.pdf* and *<star>_<element>param.pdf* figures
- **lam_min1:** lower bound wavelength in the upper window of *<star>_<model>_<element>eqw.pdf* and *<star>_<element>param.pdf* figures

- **lam_max1:** upper bound wavelength in the upper window of `<star>_<model>_<element>eqw.pdf` and `<star>_<element>param.pdf` figures
- **lam_min2:** lower bound wavelength in the lower window of `<star>_<model>_<element>eqw.pdf` and `<star>_<element>param.pdf` figures
- **lam_max2:** upper bound wavelength in the lower window of `<star>_<model>_<element>eqw.pdf` and `<star>_<element>param.pdf` figures foreach element in `<Li C N O F Na Mg Al Si P S Cl K Ca Sc Ti V Cr Mn Co Ni Fe Cu Zn Ga Ge As Rb Sr Y Zr Nb Mo Ru Rh Pd Ag Cd In Sn Sb Ba La Ce Pr Nd Sm Eu Gd Tb Dy Ho Er Tm Yb Lu Hf Ta W Re Os Ir Pt Au Tl Pb Bi Th U>`, a corresponding variable is assigned containing its abundance such that $\text{element} = \log(A)$ (e.g. $\$Fe$ is 7.45 for the Sun) Note that if no explicitly set in the `<star>.par` file, the assumed value is the solar scaled abundance (metallicity being provided by the METALLIC variable) + alpha possible enhancement for alpha elements and/or s-process/r-process enhancement as provided by respectively the **alpha**, **s_process** and **r_process** variables.
- **C12C13:** carbon isotopic ratio (default: solar=90)
- **N14N15:** nitrogen isotopic ratio (default: solar=330)

6 Some technical details about the procedures

- The code take into account the impact of carbon on the atmosphere structure. Indeed the tarball contains several atmosphere subgrids with different carbon enhancement ($[C/Fe]=-1.0,-0.75,-0.5,-0.25,0.0,0.25,0.5,0.75$ and 1.0) and α element enhancements (α includes O, Ne, Mg, Si, S, Ar, Ca, and Ti) ($[\alpha/Fe]=-1.0,-0.75,-0.5,-0.25,0.0,0.25,0.5,0.75$ and 1.0). The code will select the appropriate grid according to the value of C and α as set in the `<star>.par` file. Note that for very metal-poor stars ($[Fe/H] < -2.5$) the code will only select $[C/Fe]=0.0$. Note that while the code can update the C value `<star>.par` file and thus can eventually adopt another model atmosphere grid, it does not compute the $[\alpha/Fe]$ beyond the value established in the `init.com` script, so that this must be manually updated by the user.
- The code interpolate the model atmosphere structure among 8 models (see the manual for more details in the *INTERPOL* directory). It first selects spherical or plan parrallel atmosphere structure based on the `logg` value (`<3.0` for spherical). In case the subroutine `interpol_spherical_int.com/interpol_planparallel_int.com` fails, it will try alternate selections of models. If still fails, an extrapolated model is computed. The model name will have a “.ext” extension and the results derived from it must be taken with extreme cautiousness.
- Automatic normalisation of the observed spectrum is obtained by selecting continuum point from the synthetic spectrum over 20-30 Å around the

targeted line. After a sigma-clipping algorithm is applied on the selected points, a linear fit is computed over the remaining points. The observed spectrum is then divided by this linear fit. If no continuum points are found (e.g. in crowded spectra), then the observation spectrum is scaled to the the max value of the synthetic spectrum.

SNR as displayed in the figure `<elet>-<star>_<model>.pdf` and as written in `<element>-<star>.abu` table is estimated from the variance of the spectrum (if provided in the third column), from the value provided in the **SNR** variable (if set), or at last by the code. In this latter case, the SNR is computed by the code by evaluating the rms of the continuum point (if any). If there are not enough continuum point the SNR is derived from the ratio of the synthesis and the observed spectrum.

- Boundaries of the line mask range is determined by the overlap of i) cancellation of the first derivative of the observed spectrum ii) cancellation of the second derivative of the observed spectrum iii) flux variation of the synthetic spectrum with abundance change. Also, observed points of the normalised spectrum below 0.4 are disregarded. The final selected points are represented as a thick black line in the plot.
- *bacchus.abund* abundance determination methods (reported in the `<element>-<star>.abu` file):

1. "eqw": the code compute for each input abundance A the following:

$$\Delta eqw(A) = eqw_{syn}(A) - eqw_{obs} \quad (1)$$

where eqw_{obs} and $eqw_{syn}(A)$ are respectively defined by the n observed points defined by the line mask as described above. Then the code interpolate the $\Delta eqw(A)$ against $\log(A)$ and numerically search for the null solution. While this method is known to be independent of stellar rotation, macroturbulence, or instrument resolution -hence convolution-, it can be greatly affected by badly-fitted blending lines.

2. "syn": the code compute for each input abundance A the following:

$$\Delta syn(A) = \sum_n (x_o - x_i(A)) \quad (2)$$

where x_o and x_i are respectively the n observed points defined by the line mask as described above. Then the code interpolate the $\Delta syn(A)$ against $\log(A)$ and numerically search for the null solution. While this method is mathematically similar to the "eqw" method and offers similar advantages, but can numerically provide slight different results. Moreover, this method allows some extrapolation as well as estimates for upper limits. But the accuracy of this method depends on the quality of fit, hence the quality of line physical parameters (hfs, NLTE, collisionnal broadening, ...)

3. "int": the code first select the five points around the line center. A polynomial fit is computed over those points (and displayed in the bottom panel of the figure `<elet>-<star>_<model>.pdf` as a blue line). The line intensities (or depths) are then computed by averaging the value of the five selected points for both the syntheses $I(A)$ and the obs I_{obs} . Then the code interpolate the $I(A) - I_{obs}$ against $\log(A)$ and numerically search for the null solution. This method is not expected to be quite insensitive to blends as long as the selected line is not in the wing of another stronger line. However, it is extremely sensitive to the convolution parameter.
4. " χ^2 ": the code compute for each input abundance A the following:

$$\chi^2(A) = \sum_n (x_o - x_i(A))^2 \quad (3)$$

where x_o and x_i are respectively the n observed points defined by the line mask as described above. Then the code fit a polynome to the $\chi^2(A)$ against $\log(A)$ and numerically find the minimum. This method is the default method, because it shown present a good compromised among the other methods strengths and weaknesses (mildly sensitive to convolution and mildly sensitive to the line physical parameters).

By default, selected lines by *bacchus.eqw* are required to be all flags to 1 or extrapolation (chi2=1 int=1 syn=2 eqw=2). However, you may be a bit less conservative for abundance derivation from just weak blends (e.g. Cu in APOGEE). In that case you will have either to select yourself from the `<elet>-<star>.abu` file or to change it in the *bacchus.eqw* script. To find lines with upper limits, flags should be syn=eqw=int=2 and chi2=1

Notes:

1. The code always resample the synthetic wavelengths to the observed wavelengths.
 2. The method-to-method dispersion may be used to evaluate the quality of the line fit: the less agreement between the method, the worst is the fit.
 3. The preferred and default method for abundance selection in the *bacchus.eqw* module is the χ^2 method.
- Equivalent widths as derived in the `<elet>-<star>.eqw` and `<elet>-<star>.list` files – and as used in the parameter determination module *bacchus.param* – are synthetic equivalent widths and not the usual integrated flux over the observed spectrum. To derive the equivalent widths, the code use the same stellar parameters and the best abundance of the line from the `<elet>-<star>.abu` file, and compute the equivalent width by integrating the flux over the synthetic spectrum but with only this line. This method is expected to i) provide a more accurate equivalent width than fitting a simple gaussian or voigt profile to the observed line and ii) derive equivalent widths even for blended lines. However, in the case of heavily blended lines by another strong line, this method does not account properly for the additional source of opacity in the equivalent width determination.

- T_{eff} is determined by obtaining null trend for the abundance ($\log[A]$) of a given element against excitation potential (χ). To optimize the convergence, we established a relation between this trend (s) and the effective temperature of the star. Starting from the scaling relation in the case of weak line at LTE [5], Eq. (16.4):
 $\log(eqw/\lambda) = \log[\text{constant}] + \log(A) + \log gf\lambda - \frac{5040}{T}\chi - \log\kappa$ (where eqw is the equivalent width of a line at wavelength λ , κ is the continuum opacity, gf is the oscillator strength, and T is the temperature), we can obtain s such as $s = \frac{d\log(A)}{d\chi} = -\frac{5040}{T}$. Hence, the dependence of s with the effective temperature (T_{eff}) is expressed by $\frac{ds}{dT} \approx \frac{ds}{dT_{\text{eff}}} = \frac{5040}{T_{\text{eff}}^2}$. Because BACCHUS is mainly meant for F, G, K type star, we make the simplification such that in this relation $T_{\text{eff}} = 5040$, thus $\frac{ds}{dT_{\text{eff}}} = 0.0002$.
- $\log g$ is obtained by ionisation equilibrium. To optimize the convergence, we used the approximation established by [5], Eq. (13.25), such that “the deduced abundances [of ionised lines] vary as $g^{+1/3}$ ”, hence $\frac{d\log(A_{\text{ion}}) - \log(A_{\text{neutral}})}{d\log g} \approx 0.3$.
- metallicity is obtained by the mean of individual lines abundances relative to the solar value.
- microturbulence velocity ξ_t is determined by obtaining null trend for the abundance of a given element against equivalent widths. To optimize the convergence, we established a relation between this trend and microturbulence. Unlike T_{eff} and $\log g$, it is more difficult to establish analytically such a relation. Therefore we empirically derived it: $\frac{ds}{d\xi_t} \approx 42$, with $s = \frac{d\log(A)}{deqw}$.

However, during the first step of stellar parameter convergence process with a empirically fixed microturbulence in the *bacchus.param* module, the microturbulence adopted is the following:

– for dwarfs ($\log g > 3.5$):

if $T_{\text{eff}} > 5250$ K:

$$\xi_t = 1.15 + 2 \times 10^{-4}(T_{\text{eff}} - 5500) + 3.95 \times 10^{-7}(T_{\text{eff}} - 5500)^2 - 0.13(\log g - 4.0) + 0.13(\log g - 4.0)^2$$

if $T_{\text{eff}} < 5250$ K:

$$\xi_t = 1.15 + 2 \times 10^{-4}(5250 - 5500) + 3.95 \times 10^{-7}(5250 - 5500)^2 - 0.13(\log g - 4.0) + 0.13(\log g - 4.0)^2$$

– for giants ($\log g < 3.5$):

$$\xi_t = 0.94 + 2.2 \times 10^{-5}(T_{\text{eff}} - 5500) - 0.5 \times 10^{-7}(T_{\text{eff}} - 5500)^2 - 0.1(\log g - 4.0) + 0.04(\log g - 4.0)^2 - 0.37[\text{Fe}/\text{H}] - 0.07[\text{Fe}/\text{H}]^2$$

this relation is also used as a initial guess when no value is provided.

- convolution in BACCHUS means convolution of the synthetic spectrum by a given profile (gaussian by default). Because for a large majority of stellar spectra observations do not allow a clear distinction, the convolution value (in

km/s) encompasses all-in-one of the instrument resolution, macrobroturbulence broadening and projected rotational velocity $v \sin i$. Convolution parameter is derived by matching abundance from the intensity method and equivalent width method (the second method being insensitive to the convolution parameter, while the first one is very sensitive).

- Convergence criteria for metallicity is met when initial values equal the final value within one sigma.
- Convergence criteria for T_{eff} and microturbulence are met when abundance trend against excitation potential or respectively the equivalent widths including the error on the linear fit encompass the null value.
- Convergence criteria for $\log g$ is met when the mean abundace of neutral lines is in agreement with the abundance of ionised lines within the line-by-line rms.
- Convergence criteria for convolution is when the mean abundance obtained by the “int” abundance method agree with the mean abundace of the “eqw” abundance method.
- The line-by-line upper limits such as computed in the last columns of the `<element>-<star>.abu` are obtained by comparing the variance or the SNR of the observed spectrum to the behaviour of the line growth of the synthetic spectrum. The upper limit corresponds to the abundance where the line intensity is comparable to the variance and is consequently sensitive to stellar parameters as well as spectral resolution or signal-to-noise ratio. Warning, belended lines may not have adequate upper limits estimations.
- The execution time primarily depends on: a) the resolution of the spectrum b) the wavelength coverage of the selected lines. Indeed, most of the time is spent on computing synthetic spectrum on the fly, which is directly proportional to the number of synthetic points to compute. However, this can be optimised by allocating as many cpus as input abundances of the `bacchus.abund` module (with the `ncpu` variable).

7 Limitations and caveats

- BACCHUS is primarily a line-by-line base code. Hence, it is intended to work for high enough resolution spectra so that individual lines are resolved.
- The parameter range is limited to the model atmosphere grid in the model path, hence currently F,G,K, and M type star.
- T_{eff} and $\log g$ determination with `bacchus.param` module works from line excitation trend and ionisation equilibrium. Hence, `bacchus.param` will fix the $\log g$ if no ionisation or neutral lines are found. Furthermore, we advise to use that module in automatic mode when at least ~ 30 lines of the same element are found in the spectrum with a good enough excitation potential baseline.

- Equivalent widths (hence *bacchus.eww*) for molecular lines is not (yet) working. But abundance determination (hence *bacchus.abund*) does work for molecular lines.
- Although BACCHUS can work in principle over any wavelength range, linelists and line selection are in this package only for optical. You need to modify *bsyn.com* and *elements.wln* (+ *elements_MP.wln*) file if you want to study other regions. More linelists are available upon request but would potentially need citation.
- Most IMPORTANTLY, although the BACCHUS code includes line flags, the pre-selected lines in *elements.wln* (+ *elements_MP.wln*) are crucial as the code results and their quality strongly dependent on them.

8 Version

Code version: November 14, 2018

9 References

References to be all cited:

References

- [1] Masseron, T., Merle, T., & Hawkins, K. 2016, Astrophysics Source Code Library, ascl:1605.004
- [2] Plez, B. 2012, Astrophysics Source Code Library, ascl:1205.004
- [3] Gustafsson, B., Edvardsson, B., Eriksson, K., et al. 2008, A&A, 486, 951

Other references:

- [4] Masseron, T., PhD thesis, 2006, Observatoire de Paris
- [5] Gray, D. F., 2008, Cambridge University Press "The Observation and Analysis of Stellar Photospheres"

Linelists included:

- [6] Ryabchikova, T., Piskunov, N., Kurucz, R. L., et al. 2015, Physica Scripta, 90, 054005 "This work has made use of the VALD database, operated at Uppsala University, the Institute of Astronomy RAS in Moscow, and the University of Vienna."
- [7] Masseron, T., Plez, B., Van Eck, S., et al. 2014, A&A, 571, A47
- [8] Brooke, J. S. A., Ram, R. S., Western, C. M., et al. 2014, ApJS, 210, 23

- [9] Brooke, J. S. A., Bernath, P. F., Schmidt, T. W., & Bacskay, G. B. 2013, JQSRT, 124, 11
- [10] Ram, R. S., Brooke, J. S. A., Bernath, P. F., Sneden, C., & Lucatello, S. 2014, ApJS, 211, 5
- [11] Hinkle, K. H., Wallace, L., Ram, R. S., et al. 2013, ApJS, 207, 26
observed spectra:
- [Hinkle et al.(2000)] Hinkle, K., Wallace, L., Valenti, J., & Harmer, D. 2000, Visible and Near Infrared Atlas of the Arcturus Spectrum 3727-9300 A ed. Kenneth Hinkle, Lloyd Wallace, Jeff Valenti, and Dianne Harmer. (San Francisco: ASP) ISBN: 1-58381-037-4, 2000.,
- [Neckel(1999)] Neckel, H. 1999, Solar Physics, 184, 421