Overview of the VoigtFit program

The dataset is first initiated with data defined through a “data” statement. This sets the spectral resolution. If the spectral resolution subsequently needs to be changed, this should be done using a “resolution” statement.

The transitions that should be fitted are then defined through “lines” statements and the component structure is defined through “component” statements (see also “copy component” and “delete component”). Note that components must be defined for all the ions that are defined in the “lines” statements.

If the data are not already normalized, a window for each line fitting region will pop up allowing the user to normalize the data by selecting a left and right continuum region. This will fit a straight line to the continuum and normalize the region.

After the continuum normalization is done, the user can define spectral masks for each line fitting region. For each region, the user must select left and right boundaries by clicking with the mouse in the plotting window. This will mask out the region in between the two boundaries, and the data in the masked region will therefore not be used in the fit. The user can define as many masked ranges as desired. When the masking for a given line is done, the user must confirm the mask by clicking enter in the terminal. Note that if an uneven number of boundaries are given, the ranges are invalid and the user must select the ranges again.
The masking step can be skipped by including the ‘nomask’ statement in the parameter file.

Lastly, the fit will be performed and the resulting best-fit parameters will be printed to the terminal. If the “save” statement is given in the parameter file, the parameters will also be saved to a file and the resulting best-fit profiles will be saved to a pdf file. Otherwise, the best-fit solution is plotted in an interactive window.

If the “abundance“ or “metallicity” statements are given, the total abundance for each element or its metallicity relative to Solar is printed to the terminal.

Future implementations:

- *If HI is included in the lines to be fitted, the metallicity will automatically be calculated.*
- *more detailed options available for the minimization and output.*
- *more molecular data to be included.*
- *molecular functions like isothermal column density distribution.*

VoigtFit parameter mini-language

The parameter mini-language allows the user to define parameters without using the python scripting which has a slightly more complex syntax. There are a few general rules for the mini-language:

Everything that comes after a '#' sign is regarded as a comment and is not included when parsing the parameters.

The number of spaces in a given line between parameters and values is not important.

All optional information below is stated in brackets.

data *filename* *resolution* [norm air]

filename specifies the path to the spectrum (should be ascii table with three columns: wavelength, flux, error).

resolution is the spectral resolution of the given spectrum in units of km/s.

Optinal arguments:

norm : if present in the line, this indicates that the spectrum in *filename* are normalized.
air : if present, the wavelengths in the spectrum will be converted from air to vacuum.

Ex:

```
data  'J2350-0052_uvb.tab'  40.  air
        This will load the data from the file named 'J2350-0052_uvb.tab',
        convert the wavelength column from air to vacuum, and assign
        a spectral resolution of 40 km/s.

data  "norm_data/norm_2350-0052_vis.tab"  32.7  norm
        This will load the data from the file named 'norm_2350-0052_vis.tab'
        in the directory 'norm_data' and assign a spectral resolution of 32.7 km/s.
        The keyword 'norm' is present, so the data will be marked as normalized,
        and no interactive normalization will therefore pop up during data preparation.
```

lines *line_tags* [span=_]

line_tags can be a single line or multiple lines separated by blank spaces. The line tag should match a line in the line-list, e.g., FeII_2374, SiII_1526, or HI_1215. For the Lyman series of hydrogen and deuterium, the following notation is also accepted: HI_1 for the Ly-alpha, HI_3 for Ly-gamma, and so on.

Optinal arguments:

span : if present, the value after the equal-sign is taken as the velocity span in km/s around each line to be defined. If not, the default span of 300 km/s will be used. Can also be called as *velspan*=_

Ex:
lines FeII_2260 FeII_2374 SiII_1808 HI_1215
This will define the two singly ionized iron transitions at 2260 and 2374Å together with the singly ionized silicon transition at 1808Å and the Ly-alpha line.

lines FeII_2374 SiII_1808
lines HI_1 HI_2 span=5000
This will define the iron and silicon lines with default velocity spans and the Ly-alpha and Ly-beta lines with a larger 5000 km/s velocity span.

molecules

for CO: add molecule CO AX(1-0), AX(0-0) [J=0 velspan=150]
molecule *molecule bands* [J=_ velspan=_]
So far only CO is defined; data for H2 and HD are not defined.

component *ion z b logN*

alt.: component *ion z=_ b=_ logN=_* [var_z=True/False var_b=True/False var_N=True/False
tie_z=_ tie_b=_ tie_N=_]

ion	specifies which ion the component should be defined for, e.g., FeII, SiII.
z	gives the redshift of the component.
b	gives the broadening parameter of the Voigt profile.
logN	gives the 10-base logarithm of the column density for the given ion in cm^{-2}.

Optional arguments:

Fixed parameters can be set by the optinal arguments fix_z for redshift, fix_b for broadening parameter, and fix_N for column density. These are passed as keyword values which are either True or False, the default is False.

Parameters for different components can be tied to each other uding the tie_z, tie_b, tie_N options. Mostly used to tie redshifts or broadening parameters for different species. The parameters are tied using the following naming rules: the name of a given parameter is made up by the base (which is either ‘z’, ‘b’, or ‘logN’), the component number (starting from 0), and the ion. Base and number are joined together with no spaces in between and the ion is added with an underscore (‘_’) as spacing.
Ex:	z0_FeII for the redshfit of the first iron component
b1_SiII for the broadening parameter of the second silicon component
logN2_ZnII for the column density of the third zinc component

copy components from *ion1* to *ion2* [scale *logn ref_comp* tie_z=True/False tie_b=True/False]

The components from one ion, which have already been defined, can be copied to another ion using the ‘copy components’ statement. The ion from which the components are copied are denoted as *ion1* and must follow the word ‘from’, and the ion to which the components are copied is denoted as *ion2* and must follow the word ‘to’. The positional order is not important.

Optinal arguments:

scale : this keyword scales the pattern of column densities from the input ion to the destination ion. The keyword takes two arguments:
logn gives the desired column density for the reference component
ref_comp gives the component number to match (starting from 0).

tie_z : will tie all redshifts for *ion2* to those of *ion1*. Default is True.
tie_b : will tie all broadening parameters for *ion2* to those of *ion1*. Default is True.

Ex:

```
copy components from FeII to SiII  scale 15.3  1
```

This will copy the component structure defined for FeII to SiII
and the logarithm of the column density of the 2nd component will be set to 15.3
while keeping the relative abundance pattern as defined for FeII.

```
copy components to CII from FeII  tie_b=False
```

This will copy components already defined for FeII to CII,
however, the broadening parameters are not fixed to those of FeII.

delete component *number* [from] *ion*

number gives the number of the component to delete (starting from 0).
ion gives the ion from which to delete the given component.
Note: the keyword ‘from’ before the ion is optional.

This function is useful for removing components that were defined using a “copy component” statement, if not all components should be fitted. For regular components, the component can simply be commented out (using ‘#’).

Ex:

Suppose that FeII has 5 components defined and the same component structure has been copied to ZnII, which is much weaker. Therefore, only 4 components can be constrained for ZnII. This would be defined as follows:

```
component FeII  2.0456  15.5  14.6
component FeII  2.0469  11.5  14.8
component FeII  2.0482  17.5  13.3
component FeII  2.0489  14.0  14.3
component FeII  2.0495  13.5  14.7

copy components from FeII to ZnII  scale 13.2  0
delete component 2 from ZnII
```

resolution *res [line_tag]*

res gives the spectral resolution in km/s
line_tag specifies for which line_tag the resolution should be changed. Default is all.

This function allows the user to update the spectral resolution. If some lines are defined in different spectra (loaded by different the “data” statements”) their spectral resolution will be different. Therefore, the spectral resolution should be updated for the given lines independently.

Warning: changing the spectral resolution in the “data” statement will not update the spectral resolution, unless the dataset is overwritten.

metallicity *logNHI err_logNHI*

logNHI the logarithm of the column density of neutral hydrogen in units of cm^{-2}.
err_logNHI the uncertainty on the logarithm of the column density of neutral hydrogen.

When this keyword is present, the best-fit total abundances for the defined ions in the dataset will be converted to metallicities for each ion, that is, the abundance ratio of the given ion to neutral hydrogen relative to Solar abundances from Asplund et al. (2009) is calculated.

save [*filename*]

filename the filename used for the graphic output and for parameter output
If no filename is given, the dataset ‘name’ will be used.

The graphic output will be saved in pdf format and the parameter file will be saved as an ascii file.

name : *dataset_name*

dataset_name gives the name of the dataset.

The dataset is automatically saved (as dataset_name.dataset), and if a dataset of the given name is present, it will be loaded automatically.

z_sys : *z_sys*

z_sys gives the systemic redshift of the absorption system.

Relative velocities are calculated with respect to this redshift.

nomask

When this keyword is present in the parameter file (except in the dataset_name), no interactive spectral masking will be performed.

abundance

When this keword is present in the parameter file (except in the dataset_name), the total abundances for each ion will be printed to the terminal output.