HELIX$^+$
He-Line Information Extractor$^+$

Flexible inversion code for the
Radiative Transfer Equation

Andreas Lagg
Max-Planck-Institut für Sonnensystemforschung
Justus-von-Liebig-Weg 3
37077 Göttingen

Göttingen, 14th September 2022

Contents

1 Usage Guideline

HELIX$^+$ can be used freely. Please add references to *Lagg et al.* [2004, 2009] if you publish results obtained with HELIX$^+$.

Additionally, add a reference to the HAZEL paper [*Asensio Ramos et al.*, 2008] if you use the constant property slab model (Hanle-slab mode) for the analyzing He I data.

2 Introduction

The HELIX$^+$ code is based on fitting the observed Stokes profiles with synthetic ones obtained from an analytic solution of the Unno-Rachkovsky equations [*Unno*, 1956; *Rachkowsky*, 1967] in a Milne-Eddington atmosphere (see e.g. *Landi Degl'Innocenti* [1992]). In its latest version, HELIX$^+$ also supports the constant property slab model and the full computation of atomic polarization by using the core of the HAZEL code *Asensio Ramos et al.* [2008]; *Trujillo Bueno et al.* [2005].

Dependent on the applied model, these synthetic profiles are functions of the magnetic field strength $|B|$, its inclination and azimuth, the line-of-sight velocity, the Doppler width, the damping constant, the ratio of the line center to the continuum opacity, and the slope of the source function and the source function at $\tau = 0$, the optical thickness of the slab and the height of the slab. The atmospheric parameters that ensure the minimum of the merit function are obtained using a very reliable genetic algorithm (PIKAIA, *Charbonneau* [1995]). This algorithm reaches the global minimum of the merit function with higher reliability than a classical Levenberg-Marquardt algorithm, which is also implemented in the code.

The code is able to handle an arbitrary number of atmospheric components. Atomic data files allow the flexible usage for various spectral lines.

Different versions of the code are available:

- plain IDL version: Only IDL routines are used. This version is very slow, but OS independent.
- plain FORTRAN 90 version: no IDL is required to run the code. Several architectures are supported (e.g. Linux, Solaris). This version is optimized for speed. An MPICH / OPEN MPI version is available to run the code on workstation clusters. The FORTRAN 90 version uses DISLIN to display results.
- IDL version calling FORTRAN 90 shared objects: this version allows fast inversions within the IDL environment. This is useful for analyzing the results of the inversion.

Additionally, there is an extensive IDL library available to analyze inversions of a whole map of Stokes profiles.

3 The Physics

For a description of the physics see *Lagg et al.* [2004]. The constant property slab model is described in *Asensio Ramos et al.* [2008] and *Trujillo Bueno et al.* [2005].

4 Obtaining / Installing the Software

There are three methods to install the software:

- installation using the `git` repository at the GWDG (see Sect. 4.1),
- network installation: use this option if you want to use the software without changing it on an MPS server. Only available inside MPS (see Sect. 4.2,
- download the tgz-file from the website of the author (for offline installation, `git` is the recommended option! See Sect. 4.3).

4.1 Installation from the `git` repository

This is the recommended way of installing the HELIX$^+$ software on computers outside MPS. It guarantees that you use always the latest version of HELIX$^+$. Also, the version control system will make it easy to reproduce results of a publication which were done with a certain release of HELIX$^+$.

HELIX$^+$ is on a public `git` repository at the GWDG: https://gitlab.gwdg.de/andreas.lagg/helix.

```
#checkout: create a new directory containing the latest version:
#e.g. in home-directory:
cd $HOME
git clone https://gitlab.gwdg.de/andreas.lagg/helix
```

To set the git-hooks you must execute the gitinit command once (this command is also run everytime when you run `configure`):

```
./scripts/gitinit
```

Now you should have the HELIX$^+$ source files in the directory `helix`. Before working with the code make sure that you use the latest version. This can be done using the `pull` command:

```
#update: go to the working directory:
cd $HOME/helix
git pull
```

Do not forget to compile the code if there were changes in the source files. The binaries are not part of the `git` distribution! See Sect. 5 for details.

4.2 Network Installation

The network installation will only create links to the program files located at the server at the MPS. Network-installation only works for PCs inside the MPS-network. You will find a shell-script called `net_install` in the official distribution directory. To perform the network installation please run this shell script:

```
cd /data/slam/software/helix ; ./net_install your_directory
```

The option `your_directory` specifies the location of the network installation. It must be a directory where you have write access. If you do not specify an installation directory your home directory will be used.

The script will install the code into the directory `your_directory/helix`. Example:

```
cd /data/slam/software/helix
./net_install $HOME/helix
```

The script will create the necessary symbolic links to the distribution directory. It will also copy and synchronize some directories which contain user-files (such as `input wgt atom`). It creates input and output directories (`ps sav atm_archive profile_archive`).

The user has to take care that

- enough disk space is available in the output directories
- the profile directory `profile_archive` contains valid observations.

4.3 Local Installation

The most recent version of the software is located on the MPS-servers in the directory `/data/slam/software/helix`. This directory contains all source files, documentation and sample input files. The software is also available on the password-protected website http://www.mps.mpg.de/homes/lagg/ under "SLAM download section". Ask the author to get access to this web-page.

To install the software locally you should get the zipped tar-file `helix_yymmdd.tar.gz` in the directory `/data/slam/software/helix/archive`. Here are the step-by-step instructions for the installation:

- Copy the tar-file to a location of your choice, eg. `$HOME/helix/`:
  `cp /data/slam/software/helix/helix_yymmdd.tar.gz $HOME`
- untar / unzip the file and enter the HELIX$^+$ directory:
  `gunzip helix_yymmdd.tar.gz ; tar xf helix_yymmdd.tar`
- create the link to the correct executable:
  `ln -sf bin/helix.intel helix`
  If the correct executable is not present, you have to recompile the code (see Sect. 5).
- Match the directory structure to your needs (the input file can be used to define the path to these directories):
  `$HOME/helix/ps` - directory for postscript output
  `$HOME/helix/sav` - directory for IDL-sav files
  `$HOME/helix/profile_archive` - directory or link to the spectra
  `$HOME/helix/atm_archive` - output directory for atmospheres / synthetic spectra
  `$HOME/helix/wgt` - directory for wavelength dependent weighting functions
  `$HOME/helix/atom` - directory for atomic data files

5 Compiling the Code

HELIX$^+$ must be compiled before it can be used on your system. The precompiled binaries do no longer exist since the variety of architectures is too large to offer a running binary for every system. Compilation should be automatic and hopefully without problems if you follow this instructions. The author is happy to provide help in case you have compiling problems.

5.1 Required libraries / tools

Use your package manager to install the following programs:
`autoconf make cfitsio` (for CFITSIO see Sect. C). Note: If you want to create files larger than 2^{31} bytes you may need to recompile CFITSIO (see Sect. C for details).

For X11 output in the FORTRAN 90 version you need to install DISLIN (installation see Sect. B), for the multi processor version you need MPI. Tested MPI distributions are OPEN MPI, MPICH-3 and MVAPICH-2. DISLIN requires the OpenMotif libraries.

HELIX$^+$ is tested with the FORTRAN 90 compilers `gfortran` and the Intel compiler `ifort` (installation seeSect. A). `ifort` performs significantly better on Intel machines and is the recommended option.

5.2 Using the `autoconf` tool

To enhance compatibility with various systems the compilation now uses the `autoconf` tool. For the compilation of the code please follow these steps:

1. set the environment variables required for your FORTRAN 90 compiler, for example at MPS you setup the FORTRAN 90 compiler like this:
   `/opt/intel/bin/ifortvars.sh intel64`
   or, e.g. for the MPI-version at MPS:
   `module load mvapich2_intel`.
2. run `autoconf` in the HELIX$^+$ root directory:
   `autoconf`
3. run the `configure` script with one of the following options:
   - create a single CPU FORTRAN 90 binary with DISLIN X11 support:
     `./configure --with-dislin[=PATH-TO-DISLIN-LIBRARIES]`
     Note 1: MAC users may need to specify the path to the OpenMotif libraries (i.e. the location of libXm.a) using the option `--with-openmotif=PATH-TO-libXm.a`.
     Note 2: MAC users might have to update their command line tools and xcode. Update your xcode installation and type in a terminal: `xcode-select --install ; sudo xcode-select -switch /Applications/Xcode.app/Contents/Developer ; sudo xcodebuild -license`.
   - create the MPI-version binary:
     `./configure --with-mpi[=PATH-TO-MPI-DISTRO]`
   - create the shared objects to be used in the IDL version:
     `./configure --with-idl`

   Options al parameters to `./configure`:

   - Add `FC=ifort` or `FC=gfortran` to manually select the compiler.
   - Add `--with-cfitsio=/opt/local/cfitsio` to tell the compiler the location of the CFITSIO libraries.
   - Add `--enable-debug` to set the compiler options for a debug build.

4. to compile and install the code into the `bin` directory, type:
   `make clean ; make ; make install`

This should create the HELIX+ binary and place it into `./bin` or the IDL shared-object files in `./idl.so`

Some example shell scripts to compile HELIX+ on different machines can be found in the directory `./scripts/compile_*`. These scripts show all available options and may help you to find the right options for your environment.

If the compilation fails try `make distclean` and repeat the above steps. If it still fails, contact the author and complain.

6 Running HELIX+

6.1 Setting up the environment

Since the compilation produces a shared binary you have to define the following environment variables before you run HELIX+:

```
export LD_LIBRARY_PATH=./idl.so:(your-DISLIN-dir):\
             (your-FORTRAN-compiler-lib-dir):(your-CFITSIO-lib-dir)
```

(Note: for MAC users this environment variable is called `DYLD_LIBRARY_PATH`).

A typical example for an Ubuntu Linux system is:

```
. /opt/intel/bin/ifortvars.sh intel64
LD_LIBRARY_PATH=./idl.so:/usr/local/dislin/:$LD_LIBRARY_PATH
```

6.2 Running FORTRAN 90 version

To start the FORTRAN 90 version (e.g. compiled with Intel-compiler) with the input file `default.ipt` type in the HELIX+ root directory:

```
./bin/helix.intel -i default.ipt [ -x xxx -y yyy ]
```

The optional keywords `-x` and `-y` can be used to select a pixel from a map of observed profiles.

6.3 Running OPEN MPI / MPICH version

To use the MPI-version you have to setup your MPI environment (see Sect. E.1, Sect. D). The MPI-version is useful for the simultaneous treatment of several pixels on different CPUs. There is no speed increase for a single pixel. To start the MPI-version and run helix on 4 processors use the command:

Command for MPICH-3 and the version compiled with Intel Fortran:

```
/data/slam/software/mpich-3/bin/mpiexec -np 4 ./bin/helix.intel.mpi -i mpitest.ipt
```

See Sect. D and Sect. E.1 for details on installation and usage of the MPICH version.

6.4 Running IDL version

The IDL-version is the original version of the software. Since it is slower and not MPI-capable the main distribution is the FORTRAN 90 version. The IDL version should not differ in results from the Fortran 90 version.

Run the startup-script:
`./run`

If error messages appear, you may have to adapt the startup script `run` (eg. `emacs run`). You will have to change the location of the IDL software (default: `/opt/rsi/idl`).

Type `helix` at your IDL-prompt and the software should start with a selection list of the 5 most recent input files. You can also specify the name of an input-file directly:

```
IDL> helix,ipt='default.ipt'
```

If an error message like this appears:
`% CALL_EXTERNAL: Error loading sharable executable.`
you have to set the `LD_LIBRARY_PATH` environment variable before you start `idl`.

7 Using the Software

7.1 Basics

The inversion is controlled using ASCII-input files. These files are located in the subdirectory `input`. An inversion using the information from a specific input-file (eg. `default.ipt`) is invoked by the command at the IDL-prompt:

```
helix,ipt='default.ipt'
```

or for the FORTRAN 90-version type at the system console:

```
./bin/helix.intel -i default.ipt
```

Besides the input file there are some other important control files (filename and location of these files are controlled in the input file):

- The atomic data file (example `atom/fe630.dat`): contains the definition of spectral lines. See Sect. 7.6.
- The weighting file (example: `wgt/fe_630.wgt`): this file defines the wavelength dependent weighing scheme used for calculation of the fitness function. See Sect. 7.8 and Sect. 7.9.

Input- and output data are stored in various directories. Their names can be controlled within the input file, the default locations are:

- `profile_archive`: the location for the observations (Stokes profiles).
- `atm_archive`: the location for the results (storage for atmospheres). Contains best-fit atmospheres and best-fit Stokes profiles.
- `ps`: standard directory for postscript output. Postscript files can be created only using the IDL version.
- `sav`: standard directory for IDL sav-files (can be created and used only in the IDL version.

7.2 Some Examples

The easiest way to start using the code is by running and editing the sample input files (a detailed description of the parameters in the input file is given in Sect. 7.3). Several examples are available:

1. `ex_synth.ipt` IDL-call: `helix,ipt='ex_synth.ipt'`
   FORTRAN 90-call: `./bin/helix.intel -i ex_synth.ipt`
   Creates a synthetic profile for the He 10830 triplet of a single atmospheric component. Adds random noise to the profiles and performs an inversion. Should result in an almost perfect fit to the synthetic profile.

2. `ex_synth_2c.ipt` IDL-call: `helix,ipt='ex_synth_2c.ipt'`
   FORTRAN 90-call: `./bin/helix.intel -i ex_synth_2c.ipt`
   Similar to (1) but with two atmospheric components, shifted 20 km/s. In addition, this example adds some noise to the synthetic profile, and then starts an inversion of this noisy profile.

3. `ex_hinode.ipt` IDL-call: `helix,ipt='ex_hinode.ipt',x=1,y=134`
   FORTRAN 90-call: `./bin/helix.intel -i ex_hinode.ipt -x 1 -y 134`
   Reads in a Hinode FITS file (`./sample_data/hinode4d.20070429_1520-1520.fits`) and performs a 2-component fit (one magnetic component and a straylight component).

7.3 The Input File

The input file consists of several parts (comments, directory structure, control parameters, data set, model atmospheres, analysis method and PIKAIA parameters). The order of the lines within the input file is not important, but should be kept like that to enhance readability. All entries beginning with a semicolon '`;`' are treated as comments.

Flags are set by using '1' (=TRUE) or '0' (=FALSE).

The following sections describe the keywords which are available in every part of the input file:

All lines beginning with a word which is not a keyword described in the sections below is regarded as comment. This comment is read in by the HELIX$^+$ program and is stored in the final result. The comments in the input file following a semicolon are not stored, they are for better readability of the input file only.

The number of comment lines is unlimited.

7.3.1 Directory Structure

Here you define the input and output paths:

`PS`	`./ps/`	path for postscript output
`SAV`	`./sav/`	path for output of IDL sav-files (these files contain input information as well as the result)

PROFILE_ARCHIVE	./profile_archive/	directory containing sav-file of observation, a dat-file with single spectrum (SPINOR-format), a directory containing Hinode FITS files, a single Hinode FITS File or a mask defining Hinode FITS files (e.g. "*SP4*.fits" or "*SP3*.fits"), or the FITS file (TIP-data: cc-file) of the observation.
ATM_ARCHIVE	./atm_archive/	directory for output atmospheres. The best-fit atmosphere is written out to this directory under the following conditions: (1) you use the FORTRAN 90 version, (2) you invert more than one pixel, (3) you call the IDL version of HELIX$^+$ with the keyword /force_write.
ATM_SUFFIX	v01	add a suffix to atm-directory to identify this run
WGT	./wgt/	directory for wgt-files
ATOM	./atom/	directory for atomic data files

7.3.2 Control Parameters

These parameters control the behavior during the minimization.

DISPLAY_PROFILE	1	The input and fit Stokes profiles are displayed for every inverted profile (X11 and IDL version).
DISPLAY_COMP	1	If 1 then the individual atmospheric components of the fitted profiles are displayed (only active when DISPLAY_PROFILE is set to 1).
SAVE_FITPROF	1	If 1 then the fitted profile is saved as an IDL-sav file (can be used as input profile).
FITSOUT	1	Map results (atmospheres and profiles) are written to FITS file. See Sect. 9.1.
OUTPUT	PS	controls output media: PS for postscript, all other settings for screen (IDL-version only).
VERBOSE	1	controls verbosity: 0=none, 1=normal, 2=female
OLD_NORM	0	Use code with old normalization of the various atmospheric components. See Sect. H.1.

7.3.3 Data Set

This section describes the observation to be used as well as the way it should be processed.

The code is optimized to be used with data from the TIP / TIP-2 instrument. If you wish to apply the code to other data, it is recommended that you store the data in the TIP FITS format. A description of this format can be found in Sect. 8.5.

`OBSERVATION`	`13may01.014.profiles.abs.sav`	filename of the sav-file containing the observation.

Several formats are accepted for the observation:

- 4D Fits files. This is the recommended format. The format is described in Sect. 8.3.
- IDL-sav file of single profile (must contain structure with the tags IC, WL, I, Q, U and V (IDL-version only!)
- IDL-sav file containing a 2-D map of profiles (output from `line_id` data reduction program, IDL-version only!)
- IDL-sav file in solar MHD-format (see `read_solmhd.pro`) (IDL-version only!)
- single profile in SPINOR-format (extension: `.dat`)
- reduced TIP FITS file (eg. `19oct05.012cc`). For split files (TIP2) specify the name of the first FITS file (eg. `19oct05.002-01cc`). If no auxiliary FITS file (eg. `19oct05.012ccx`) is present, then the the user has to specify the wavelength calibration in the input file using the keywords `WL_DISP` and `WL_OFF`. Furthermore, the continuum level is determined only by a simple routine based on the maximum value of the I-profile. It is therefore highly recommended that you create a ccx-file (auxiliary data reduction information) using the `line_id`.
- Hinode data files. `OBSERVATION` can be:
  - a directory containing Hinode FITS files,
  - a single Hinode FITS file or
  - a mask defining Hinode FITS files (SP4 or SP3), e.g. `07apr30/cal/SP4*_05*.fits`.

HELIX$^+$ should automatically detect the format of the observation file.

`WL_RANGE`	`10825 10835`	WL-range to be used for analysis. The maximum range will be defined by the observed profile. The use of this keyword allows to restrict the WL-range of the observed profile. In synth-mode (`SYNTH 1`) this keyword defines the WL-range.
`WL_NUM`	`256`	# of WL-points (for synthesis only)
`WL_BIN`	`1`	bin size for WL-binning

WL_OFF	0	wavelength calibration: offset. if WL_OFF is not equal to zero then the WL-calibration from the input file is used - **the WL-calibration contained in the data file is ignored!** The WL axis is computed as follows: WL = bin#(from 0 to # of WL-points -1)*WL_DISP + WL_OFF. Not used for synthesis-mode (i.e. SYNTH 1).
WL_DISP	0	wavelength calibration: dispersion per wavelength bin. See WL_OFF for details.
XPOS	000 150	two-elements vector containing xmin,xmax of the observation map to be analyzed
YPOS	000 100	two-elements vector containing ymin,ymax of the observation map to be analyzed
PROFILE_LIST	list.dat	filename of profile list. The profile list contains a two-column file with the x- and the y-values of profiles to be inverted (format: 2 integers separated by a blank). This list overwrites the XPOS and YPOS keywords. The profile list file must be located in the HELIX$^+$ root directory (./) or in the directory specified with the PROFILE_ARCHIVE keyword.
STEPX	1	step size for going from xmin to xmax. This is also the step size for the averaging.
STEPY	1	step size for going from ymin to ymax. This is also the step size for the averaging.
AVERAGE	1	if set to one then the profiles of an observation are averaged over an area of size stepx,stepy
SCANSIZE	0	step-size of multiple scans within one observation
SYNTH	0	if set to one, a synthetic profiles is minimized. The synthetic profile is calculated using the atmosphere defined in the the next section.
NOISE	0.0	Add artificial random noise of level x to IQUV.
INOISE	0.0	Add artificial random noise of level x to Stokes I only.
SMOOTH	0 0	smoothen profiles. First value defines smooth value, second value smooth method (0=IDL-smooth function, 1= FFT Low-Pass). Only available in IDL-version!

MEDIAN	0	apply median filter to input profiles. Helpful for removing spikes in observed profiles. Uses the IDL median function, a similar function is implemented in the FORTRAN 90 version.
SPIKE	0	if set to values $\geq$1 removes spikes (e.g. spikes caused by electronic noise). See routine `remove_spike.pro`.
MIN_QUV	0.	Define minimum magnetic signal. If the average of the 4 largest values in the magnetic signal $\sqrt{Q^2+U^2+V^2}$ over the wavelength range where the weighting functions are non-zero is smaller than `MIN_QUV` then the magnetic field is set to 0 and is not fitted.
SLIT_ORIENTATION	94.0	slit-orientation of observation (used for azimuth correction in the display routines).
SOLAR_POS	0.00 0.00	pos. of observation (x and y in arcseconds)
SOLAR_RADIUS	950.00	radius of sun in arcsec for time of obs.
HELIO_ANGLE	49.2	heliocentric angle of observation in degrees (for full Hanle). For TIP: $\theta = asin(\sqrt{x^2+y^2}/r$, (x,y) = position of observation in terrestrial coordinates. See Sect. 7.10.
+Q2LIMB	105.	pos. Q to limb direction in degrees (for full Hanle). For TIP: $Q2L = atan(x,y)$, emission vector angle $\gamma = 360. -$ `+Q2LIMB` $+90$.
HIN_SCANNR	0	Scan-number for Hinode data files (identifies the scan number for short, repetitive scans). See Sect. 8.6.1.
NORM_CONT	local	type of cont. normalization: local, slit or image. "local" means that every single Stokes vector is normalized to its continuum, "slit" normalizes to the maximum continuum level along a single slit and "image" normalizes to the maximum value of the continuum level in the whole observation. NOTE: in order to fit the continuum level the Voigt-S_0 parameter mode (see Sect. 7.3.5)! See also Sect. 7.3.9.
NORM_STOKES	IC	normalization of Stokes profiles: I or IC

IC_LEVEL	0.	set level for Ic. A positive number overwrites the Ic value contained in the observational data. To use the IC level from the observational data this value must be set to 0 or to a negative number.
LOCALSTRAY_RAD	40	add a local straylight component: take the average Stokes profile of the surrounding profiles within the radius specified here and mix it via alpha to the other components. The radius is in units of the pixel size defined with the STEPX / STEPY keywords.
LOCALSTRAY_FWHM	20	FWHM of the Gaussian weighting used for the local straylight profiles
LOCALSTRAY_CORE	5	exclude core profiles within the radius specified here
LOCALSTRAY_POL	1	flag to determine whether the local straylight correction should be done with the unpolarized (=0) or polarized (=1) profile. Default is polarized local straylight.

7.3.4 Post Processing Options

Convolution This section of the input file contains the parameters on how the profile is changed after the synthesis. A good example for such post processing parameters is the convolution with a telescope filter function. If a synthetic spectrum should be compared with an observed spectrum, it is important that the synthetic spectrum is "sent through the telescope". If this is not done, then you compare a clean synthetic spectrum with a spectrum which is affected by, e.g., the filter curve of a Fabry-Perot.

At the moment the convolution with an arbitrary filter function is possible. This convolution is described by three parameters:

CONV_FUNC	./convolve/fp_630.dat	ASCII-file containing the filter function. Two columns in the ASCII file represent wavelength and filter transmission.
CONV_NWL	256	# of wavelength points for the filter function
CONV_MODE	FFT	method for applying the convolution function. Possible methods are: FFT, MUL. Method MUL uses a simple multiplication of the convolution function with the synthetic spectrum. FFT performs a convolution using a fast Fourier trafo. FFT is faster for small number of convolution wavelengths (CONV_NWL). MUL is faster for small number observed wavelengths and large numbers for CONV_NWL.

CONV_OUTPUT	0	flag to control the display/output of the fitted profile. If 1 then the profile is displayed with CONV_NWL WL pixels, if 0 the WL-pixels of the observation are used. NOTE: for local straylight correction this flag is set to 0.
CONV_WLJITTER	5E-3	uncertainty in Å in knowledge of exact wavelength position for every filter position (e.g. caused by error in etalon voltage). Make sure that the wavelength resolution is high enough (i.e. increase CONV_NWL)! This value can be used to simulate instrumental effects (SO/PHI). Must be set to ZERO for real data!
PREFILTER		filename containing the prefilter curve
PREFILTER_WLERR		error in Å in knowledge of prefilter curve (e.g. caused by unknown temperature). A random WL-shift is applied to the prefilter curve before the inversion is done. Must be set to ZERO for real data!

The filtering is done by applying an FFT convolution. Some points are important in order to do a correct convolution with the filter curve:

- Since the filter function might have contributions at wavelengths outside the WL-range interesting for the analysis (defined with the WL_RANGE keyword, by the weighting function or by the observation itself), it is essential that the WL-range for the calculation of the profile is extended.
- This extension of the WL-range is done using the WL_RANGE keyword. Internally the code calculates a profile over this WL-range, whereas the comparison between observation and fit and the output is done using the original WL-range defined via the observation.
- The filter function is interpolated to match the spacing of the observation. It is highly recommended that you use the same WL-binning for the observation and the convolution by using the keyword CONV_NWL. A warning message will be issued if the WL-bins between observation and convolution function differ by more than 5 mÅ.
- For irregular gridded observation (e.g. filtergramms with 5 or 6 wavelength positions) try to find a WL-binning for the convolution which fulfills 2 criteria:
  - WL-bins of convolution function as close as possible to the observed WL-bins and
  - number of WL-bins for the convolution should be as small as possible (computing time).
- The keyword CONV_OUTPUT defines whether the fitted profile should be displayed with the wavelength binning used for the convolution (CONV_NWL wavelength pixels), or with the wavelength pixels of the observation data. Note that when using the local straylight correction you cannot obtain the fitted profile with the wavelength binning of the convolution, since the local straylight component is only available for the observed wavelength pixels.
- The keyword PREFILTER contains a file which describes a prefilter curve (format is similar to the file containing the convolution function). The prefilter curve is fixed in wavelength.

To find the optimum WL-binning for the convolution you can use the IDL-tool `find_opt_binning`. This tool displays the difference between the binning of the observation and the equally spaced binning used for the convolution as a function of the number of convolution wavelength bins. A good choice for the binning is where this difference is as small as possible. The IDL command is:

```
find_opt_binning,wlin=[Wl-vector of data],wlrg=[6171.0,6175.5]
```

Prefilter The effect of a prefilter of imaging polarimeters can be treated using the `PREFILTER` keyword. The syntax is:

`PREFILTER` `./convolve/prefilter.dat` filename of prefilter function.

The filename specified with this keyword should contain two columns: The first column specifies the wavelength, the second column the prefilter transmission. The example file `prefilter.dat` contains a Gaussian prefilter for the 617.3 nm Fe line.

7.3.5 Model Atmospheres

Here the model atmosphere is defined. An atmosphere describes the conditions on the Sun where the line is formed. One pixel can contain more than one atmosphere: straylight from the surroundings can affect the signal at the position of the pixel and / or unresolved structures can lead to a linear superposition of profiles.

HeLIx$^+$ can be used in six different modes. The modes are automatically set by the physical parameters of the model atmosphere:

- Gauss-mode:
  `BFIEL`, `AZIMU`, `GAMMA`, `VELOS`, `WIDTH`, `AMPLI`, `SGRAD`, `ALPHA`
- Voigt-mode:
  `BFIEL`, `AZIMU`, `GAMMA`, `VELOS`, `VDAMP`, `VDOPP`, `SGRAD`, `EZERO`, `ALPHA`
- Voigt-Gdamp mode: similar to voigt-mode, but instead of the damping constant `DAMP` (a) the damping factor `GDAMP` (Γ) in units of the velocity of light (as defined in *Balasubramaniam and West* [1991]) is used:
  `BFIEL`, `AZIMU`, `GAMMA`, `VELOS`, `GDAMP`, `VDOPP`, `SGRAD`, `EZERO`, `ALPHA`
  $a = \Gamma \frac{\lambda_0^2}{4\pi\Delta\lambda_D}$
- Voigt-S_0 mode: same as Voigt-Gdamp mode. The only difference is that the source function at $\tau = 0$ is a free parameter. This allows fitting of profiles which are not normalized to its local continuum level. The fit parameters are:
  `BFIEL`, `AZIMU`, `GAMMA`, `VELOS`, `GDAMP`, `VDOPP`, `SZERO`, `SGRAD`, `EZERO`, `ALPHA`
  $a = \Gamma \frac{\lambda_0^2}{4\pi\Delta\lambda_D}$
- Voigt-physical mode, see *Balasubramaniam and West* [1991]:
  `BFIEL`, `AZIMU`, `GAMMA`, `VELOS`, `GDAMP`, `VMICI`, `DENSP`, `TEMPE`, `SGRAD`, `ALPHA`
- Hanle-slab mode (constant property slab model): This mode computes the Stokes profiles for the He 10830 line including the full treatment of atomic polarization in the Hanle-slab model. See Sect. 7.10 for details.

BFIEL, AZIMU, GAMMA, VELOS, VDAMP, VDOPP, DSLAB, SLHGT, ALPHA
Note: You can omit the parameter VDAMP in the Hanle-slab mode. Then the damping constant a is calculated from VDOPP using the formula
$a = A\lambda \cdot 10^{-10}/(v_{Dopp} \cdot 10^3 \cdot 4\pi)$,
(A = Einstein coefficient, v_{Dopp} in km/s, λ in Å. This corresponds to take into account thermal broadening effects only.

Note: There is a mismatch between the units of VDOPP between the Hanle-slab mode and the other modes. In the Hanle-slab mode VDOPP is the thermal broadening of the spectral line in km s^{-1}, in all other model atmospheres VDOPP corresponds to the broadening in Å! This discrepancy is not nice, but it allows direct comparison between HAZEL [*Asensio Ramos et al.*, 2008] and HELIX$^+$ results. The user has o take care of this issue and set the value / range for these parameters accordingly. The values can be converted by using this formula:
$\mathtt{VDOPP}_{other} = \mathtt{VDOPP}_{Hanle-slab}\, 10^{11} \,/\, (\lambda\, c)$, with $\mathtt{VDOPP}_{Hanle-slab}$ in km s^{-1}, λ in Å, and c the speed of light in m s^{-1} (for He 10830 the factor is 0.0308).

7.3.6 Fitting atmospheric parameters

Every parameter of an atmosphere has the following form:

```
PAR_NAME          value  min    max    fit    ;comment
```

The name of the parameter is followed by it's initial value (important for synthesis), an allowed range (min to max) for this parameter and a fit-flag. The values for fit-flags are:

- 0: do not fit this parameter - use initial value
- not zero: treat this parameter as free parameter
- negative number: couple this parameter with other parameters of same negative number. This feature is useful to couple for example the magnetic field inclination angle for between two magnetic components.
  Note: The coupling only works between the same parameter of different atmospheric components. **The parameters are coupled according to the ratios defined by the initial values. The ratio of the initial values is preserved by the coupling!**

7.3.7 Defining a model atmosphere

The following keywords list the available parameters for model atmospheres. Only the combinations mentioned in Sect. 7.3.5 can be used to define valid atmospheres. It is not possible (yet) to mix different model atmospheres (e.g. for different spectral lines to be fitted) within one inversion. HELIX$^+$ automatically detects which type of model atmosphere is selected from the choice of parameters below:

```
NCOMP             1                          number of atmospheric components

; NAME            Value  MIN    MAX    FIT    Comment

BFIEL             200    0.0    2000   1      magnetic field value in Gauss
```

AZIMU	0.0	-90	90	-2	azimuthal angle of B-vector in degrees
GAMMA	0.0	0	180	-3	inclination angle of B-vector in degrees
VELOS	0.0	-7000	40000	1	line-of-sight velocity in m/s
WIDTH	0.20	0.10	0.70	1	line width (Gauss profile only!)
AMPLI	0.5	0.0	1.0	1	amplitude of line (Gauss profile only!)
VDAMP	0.1	0.05	10.0	1	damping constant (Voigt profile only!)
VDOPP	0.05	0.0	1.0	1	Doppler broadening (Voigt profile only!) in units of $\mathrm{km\,s^{-1}}$ for the Hanle-slab model and of Å for all other model atmospheres.
EZERO	1.0	0.0	20.0	1	amplitude of components of propagation matrix (Voigt profile only!)
GDAMP	0.1	0.05	10.0	1	damping constant (Voigt profile, physical units mode only!), see *Balasubramaniam and West* [1991]
VMICI	0.1	0.0	1000.0	1	micro turbulence in m/s (Voigt profile, physical units mode only!), see *Balasubramaniam and West* [1991]
DENSP	0.1	0.0	100.0	1	density parameter (Voigt profile, physical units mode only!), see *Balasubramaniam and West* [1991]
TEMPE	10000.	0.0	20000.	1	temperature (Voigt profile, physical units mode only!), see *Balasubramaniam and West* [1991]
SZERO	0.02	0.00	0.04	1	source function at $\tau = 0$
SGRAD	1.0	0.9	1.1	1	gradient of source (Planck) function
DSLAB	0.100	0.02	0.40	1	optical thickness of He slab (Hanle-slab model only!), dimensionless
SLHGT	3.000	1.00	6.00	0	He slab height in arcsec (Hanle-slab model only!)
ALPHA	0.5	0.0	1.0	1	Filling factor for this component (useful only with multiple components). If the filling factor is not fitted, then the value of SGRAD is used to calculate the filling factor. Note that the filling factor of all components is normalized to one before the inversion.

IMPORTANT: If the filling factor for one component is a free parameter, then the filling factor for at least another component must also be a free parameter. Otherwise the code has no possibility to adjust the filling factor!

USE_ATOM	he1083.0.new.dat	atomic data file(s) to be used for this component (more than 1 file possible). This keyword finalizes the definition of one component and must therefore be the last keyword for this component. See Sect. 7.6.
ATM_INPUT	filename.fits	filename for input atmosphere: defines initial value for atmospheric parameters or delivers the input to synthesize a map. See Sect. 7.7.
USE_LINE	He	OBSOLETE KEYWORD! USE USE_ATOM INSTEAD! Lines to be used for this component.

For a two-component atmosphere you have to set NCOMP=2 and duplicate the parameter block in the input file. It makes sense to have different ranges for the different components. You can for example use component #1 as a slow component (velocity range -7000 to +7000 m/s) and component #2 as a fast downflow component (velocity range +5000 to +40000 km/s).

7.3.8 Straylight Atmospheres

Straylight atmospheres can be treated in various ways. The usual way is to add an atmospheric component representing the straylight atmosphere. A typical straylight component has the following parameters:

- BFIEL is set to 0.
- VELOS is set to 0.
- VDOPP, EZERO, SGRAD, GDAMP are set to values determined from a quiet sun profile.
- ALPHA is a free parameter.

For Hinode and TIP-format fits files the LOCALSTRAY_RAD keyword is available: with this keyword an average local straylight profile is computed from the Stokes I profiles surrounding the pixel for which the inversion should be carried out. The parameter of the LOCALSTRAY_RAD keyword determines the radius to be used for the calculation of the average straylight profile. The radius is given in pixels, taking into account the values of the STEPX and STEPY keyword. The Stokes I profiles within this box are weighted with a Gaussian profile of full width half maximum defined with the LOCALSTRAY_FWHM keyword. The profiles within the radius defined with the keyword LOCALSTRAY_CORE are excluded. The so determined straylight profile is mixed with the profiles explicitly calculated from the atmospheric parameters in the input file via a filling factor. The keyword LOCALSTRAY_POL determines whether the local straylight is polarized (=1) or unpolarized (=0).

For Hinode data the usage of the local straylight mode (see keywords LOCALSTRAY_xxx) is highly recommended. This mode computes a straylight profile from the surrounding pixels. Additionally, NORM_CONT should be set to IMAGE or SLIT and the Voigt-S_0 parameter mode should be used. This should lead to results similar to the MILOS inversions described by *Orozco Suárez et al.* [2007]. See also Sect. 8.6.1.

7.3.9 Normalization of profiles

The code offers two different methods of normalizing the Stokes profiles:

1. Normalization to the continuum of every single profile (`NORM_CONT LOCAL`):
   Here you define the continuum level of every individual profile to be the value for that pixel, defined in the ccx file. If you used `make_ccx`[1] to create the ccx file this means, that the code analyzed the I profile of this pixel and tried to find the continuum level for this profile. Dividing the I profile by the continuum level of this pixel will set the I profile outside the line to 1.

2. Normalization to the continuum level of the image (`NORM_CONT IMAGE`):
   The `make_ccx` routine looks for the brightest I profiles in the map and calculates an average I profile out of these profiles. The global continuum level is then set to the intensity of this profile outside the line, dividing this average I profile by this global continuum level results in a Stokes I profile of unity outside the line. These brightest profiles should represent the average quiet Sun profile. For the inversions, every profile is then divided by the global continuum value. In darker regions, e.g. intergranular lanes, pores, spots, the I profile outside the line has a value of much less than unity.

When using method 2 the parameter `SZERO` must be a free parameter. This parameter takes into account the reduction of the continuum level, i.e. the lower temperature in darker regions. According to *Orozco Suárez et al.* [2007], this normalization in combination with local stray light correction is the only way to reliably determine the straylight contamination of every single pixel.

With method 1 you cannot fit `SZERO` (every individual profile has a continuum of unity). The straylight determination is less reliable there.

The third normalization method (`NORM_CONT SLIT`) is only available for Hinode data. Here there is an additional continuum level calculated for every slit index. This method has not practical use in the general case.

7.3.10 Line Dependent Parameters

Under certain circumstances it is necessary to fit the some parameters of the atomic line. This is the case for example

- if the wavelength of the line is not known accurately enough
- if the ratio of two lines is not equal to the ratio of the corresponding loggf values (i.e. the line is formed at different temperatures)
- if a velocity gradient with height leads to different Doppler shifts because of different formation heights of the lines.

Line parameters can be specified in the input file for every spectral line contained in the used atomic data files. Every line is identified by its wavelength (`LINE_ID` keyword). Available line parameters are `LINE_STRENGTH` and `LINE_WLSHIFT`. The syntax of the input file entries is similar to the atomic parameters. The first value denotes the initial value (also used for synthesis), the second and third value give the fit range, the fourth value decides whether the parameter is free or not. The same

[1] For the usage of the `make_ccx` routine see Sect. 8.6.1 and 8.8.

coupling mechanism as for atmospheric parameters is working (coupling using negative values for the fourth parameter).

LINE_ID	6301.5012				wavelength to identify the line
; NAME	Value	MIN	MAX	FIT	Comment
LINE_STRENGTH	1.00	0.80	1.20	1	factor to adjust line strength
LINE_WLSHIFT	0.00	-0.20	0.20	1	adjust central WL of line (in Å)

7.3.11 Telluric Blends

Telluric blends can be fitted to the spectrum with a Voigt profile. The parameters of the blends can be specified similar to the parameters of the atmospheric components. Multi-iteration and coupling for blend parameters is disabled, since it does not really make sense here.

To limit the number of fit parameters, the blend values can be determined once (e.g. in the quiet sun or in the flat field), and then be used as predefined values (set fit value to 0), or with only very small ranges.

The telluric blend is fitted to the I profile. Since the telluric blend reduces the light level at a specific wavelength, the Q, U, and V profiles are also affected. HELIX$^+$ takes this into account by also reducing Q, U, and V by the same amount as the I profile is reduced.

NBLEND	1				number of telluric blends
; NAME	Value	MIN	MAX	FIT	Comment
BLEND_WL	10832	10831	10833	1	wavelength and possible range for blend
BLEND_WIDTH	0.3	0.0	1.0	1	width of voigt-profile for blend
BLEND_DAMP	0.3	0.0	1.0	1	damping of voigt-profile for blend
BLEND_A0	0.5	0.0	1.0	1	amplitude of voigt-profile for blend (blend is not used if A0=0)

7.3.12 General Fit Parameters

This section of the input file contains general fit parameters, i.e. parameters which affect the general shape of the profile.

CCORR	1.0	0.95	1.05	1	Continuum Correction Fit. If this parameter is a fit-parameter (FIT=1), then the continuum level is adjusted within the specified range. This is useful for data where the continuum cannot be determined reliably.

STRAYLIGHT	0.0	0.0	1.0	1	Correction for **instrumental straylight**. If this parameter is a fit-parameter (FIT=1), then a non-polarized, non-dispersive straylight is added. It will add a constant background to the I profile, and it will not affect Q, U and V.
RADCORRSOLAR	1.0	0.95	1.05	0	Correction for the intensity of the solar radiation field (for Hanle mode only). The intensity is computed according to Allen [pie, 2000]. This parameter is a multiplicative correction for the Allen approximation computed in `hanle_module`.

Note that this straylight correction does not change the filling factor of the atmospheric components. It takes into account a constant background illumination resulting from straylight in your instrument, and must not be mixed up with straylight atmospheric components! For example, if the code determines a straylight correction of 0.2, then 20% of the light is unpolarized straylight (originating from instrumental effects), and only 80% of the light are of solar origin.

7.3.13 Analysis Method

Here you specify whether an approximation for the calculation of the magnetic field direction should be used or not. The approximation used is based on *Auer et al.* [1977].

APPROX_AZI	1	use approximation to calculate magnetic field azimuthal angle directly from Q and U profile [*Auer et al.*, 1977]
APPROX_DIR	1	use approximation for magnetic field direction. This keyword sets `APPROX_AZI` to 1. This approximation for the inclination is not very accurate for low B and high inclinations
APPROX_DEV_INC	0.	The approximation for the inclination angle of the magnetic field direction is used as an initial value for the minimization. The value is allowed to vary $\pm$X° around the approximation.
APPROX_DEV_AZI	0.	Same for azimuthal angle.
IQUV_WEIGHT	1. 1. 1. 1.	4-element vector defining relative weighting of IQUV (in that order). Additionally the code does a weighting according to the weighting file defined with the `WGT_FILE` keyword. This weighting scheme is used in the PIKAIA fit routine. See Sect. 7.8 and Sect. 7.9.
WGT_FILE	he_default.wgt	file with WL-dependent weighting function for IQUV, can be different for multiple iterations

PROFILE	voigt	functional form for π- and σ components of spectral line. Available: gauss or voigt
MAGOPT	1	include magneto-optical effects (dispersion coefficients, (Voigt profile only!))
USE_GEFF	1	use effective Landé factor (=1) or real Zeeman pattern (=0)
USE_PB	1	if set, the Zeeman-splitting and strength includes the Paschen-Back effect (from the table by Socas-Navarro, see section 7.14)
PB_METHOD	1	use polynomials (=poly) or table interpolations (=table) to calculate the PB-effect

7.3.14 Pikaia Parameters

Here you control the call of the Pikaia or other minimization routines.

CODE	FORTRAN	Language to use. Available: FORTRAN (=fast) or IDL (=platform independent). The IDL code is of course only usable within the IDL interface.
METHOD	PIKAIA	Minimization method. Available: PIKAIA (=slow, but better convergence), POWELL (fast, may be trapped in local minimum), LMDIFF or PIK_LM. POWELL is available in IDL only. LMDIFF involves a Levenberg-Marquardt algorithm, based on the ODRPACK95 routines [*Zwolak et al.*, 2007]. This algorithm should be very fast and reliable. However, PIKAIA will show the best convergence. See Sect. 7.4 for details on the combined convergence strategy PIK_LM.
NCALLS	200	number of iterations in PIKAIA/POWELL/LMDIFF routine.
CHI2MODE	0	Method to calculate χ^2/fitness (any value = default method, JM = Borrero method, PLAIN = weight only depends on IQUV_WEIGHT values). The default weighting is proportional to the inverse of the signal strength of the individual Stokes parameters. See Sect. 7.9.
PIXELREP	1	number of repetitions per pixel (to perform statistical analysis). See Sect. 7.18.
KEEPBEST	1	If set to one, then the result of an inversion is only stored to the FITS file if the fitness of the solution is better than the fitness already stored in the FITS file for this pixel (FITSOUT only).

PIKAIA_STAT	POP75	Switch on and control statistic module based on PIKAIA populations. Requires FITSOUT=1 and PIXELREP=1. Available modes are: NONE (=default), POPxx Example: POP75 - take the fittest 75% of the PIKAIA population.
PIKAIA_POP	0	number of PIKAIA populations (0=automatic). A larger number increases the probability of finding the fittest solution to the problem, but also increases the computing time.

7.4 Convergence Strategies

PIKAIA provides a very robust technique to find the best fit to the measured Stokes vector. However, it is also very slow. The Levenberg-Marquardt based algorithm LMDIFF is approximately a factor of 40 faster than PIKAIA. Since this algorithm more likely gets trapped in local minima, the correct choice of initial parameters is important.

HELIX$^+$ includes the possibility to combine the advantages of PIKAIA with the speed of LMDIFF when inverting a whole map of Stokes parameters. This is done by setting the minimization method (keyword METHOD) to PIK_LM. In a rectangular box of size BX×BY pixels the first run is done using PIKAIA for the lower left pixel of this box. All subsequent runs in this box are done using LMDIFF, with the initial parameters being the results of the PIKAIA run or any following LMDIFF run with higher fitness.

METHOD	PIK_LM	BX	BY	0.7	Use PIKAIA for the first pixel and LMDIFF for the other pixels in a box of size BX×BY. Use PIKAIA also for results with fitness lower than 0.7 × the best (so far) fitness in this box.

The fourth parameter of the keyword METHOD is optional and defines the fraction by how much the fitness of a LMDIFF inversion is allowed to be worse than the best fitness in the box. If the fitness is less than, e.g. 0.7, then the inversion for this pixel is repeated using PIKAIA. Setting this value to 0 (or not specifying it at all) will disable this fitness-check. Setting this value to 1 means that no LMDIFF fit is allowed to be worse than the PIKAIA fit. A higher value therefore increases computation time (since more pixels are calculated using the slow PIKAIA algorithm).

Speed Comparisons

Table 1 lists the performance of the various methods. PIKAIA clearly wins in the fitness of the derived atmospheric parameters, LMDIFF is by far the fastest method. Fig. 1 shows the resulting maps for magnetic field strength and the velocity contour lines.

Figure 1: Magnetic field maps for the observation of an emerging flux region recorded on May 13, 2001. The results are from a two-component inversion, the methods used are in the same order as in Table 1. The contour lines show the velocity. The input file for this example can be found in Sect. F.

Table 1: Comparison of various convergence strategies: The box size for the combined LM_DIFF strategy was chosen to be 1×20 pixels.

method	input file entry	average fitness	time	rel. time [%]
PIKAIA	`METHOD PIKAIA`	2.26	81 h	100.0
LM_PIK	`METHOD PIK_LM 1 20 0.9`	2.17	28 h	34.6
LM_PIK	`METHOD PIK_LM 1 20 0.7`	2.18	11.2 h	13.8
LM_PIK	`METHOD PIK_LM 1 20 0.0`	2.15	6.6 h	8.1
LMDIFF	`METHOD LMDIFF`	1.82	2.9 h	3.6

7.5 Re-Write Input Files

The IDL-routine `nice_input.pro` can be used to reformat the input file. It will add the default comments and will make the file nice-looking. The calling sequence is (applied to input file `default.ipt`:

```
nice_input,'input/default.ipt'
```

7.6 Atomic Data Files

The atomic data are defined in ascii files in the directory defined with the `ATOM` keyword. You must define at least one atomic data files for every atmospheric components by specifying the keyword `USE_ATOM` for each component. These atomic data files can be different for different components (e.g. one component uses the He-line, another component only affects the Si-line).

The format of the atomic data file is (see here the example for the He triplet at 1083 nm):

```
;atomic data file for he triplet at 1083 nm
;WL        Element Ion.   LOG_GF   ABUND  GEFF   SL   LL   JL   SU   LU   JU
10829.0911   He    1      -0.745    0.00   2.0  1.0  0.0  1.0  1.0  1.0  0.0
10830.2501   He    1      -0.268    0.00  1.75  1.0  0.0  1.0  1.0  1.0  1.0
10830.3397   He    1      -0.047    0.00  1.25  1.0  0.0  1.0  1.0  1.0  2.0
```

The column entries are: wavelength, element name, ionization state (1=neutral), log-gf value (f is the oscillator strength and g is the degeneracy of the lower level), solar abundance (not used), effective Landé factor and the quantum numbers s, l and j for the lower and the upper level. The effective Landé-factor is calculated from the quantum numbers. If it differs from the value given in the atomic data file, a warning message is issued and the value from the atomic data file is used.

7.6.1 Oscillator Strength

The value `LOG_GF` is used to adjust the strengths of the individual Zeeman components. Every component is multiplied by $10^{\mathrm{LOG\_GF}}$. It allows to define the line ratio of blended lines. The atomic data file for the He 10830 triplet shown above reflects the ratio of $0.11 : 0.33 : 0.55$.

Note that this method changes the standard definition of `EZERO` used in other Milne-Eddington codes (η_0). To achieve a standard `EZERO` definition you must set `LOG_GF` to zero. Then the factor for the individual Zeeman components becomes unity.

7.7 Atmospheric Parameter File

The keyword ATM_INPUT can be used to specify the file containing atmospheric parameters to be used as initial conditions for the inversion or as a 2D-map input for doing a line synthesis (in combination with NCALLS set to zero. The file can be specified for every atmospheric component separately. It must be a fits-file fulfilling the following requirements:

- The parameters must be stored as floats in a fits file of dimension $n_{\text{par}} \times n_x \times n_y$.
- Header parameters to describe the atmospheric parameters. For example:

```
PAR1    = 'BFIEL   '           / name of atmospheric 1st parameter
PAR2    = 'AZIMU   '           / name of atmospheric 2nd parameter
PAR3    = 'GAMMA   '           / name of atmospheric 3rd parameter
```

- The atmospheric parameters must match the name in the input file. Any other parameters will be ignored.
- Not all atmospheric parameters have to be specified in the ATM_INPUT fits file. It is, e.g. possible, to only provide the magnetic field strength. All the other parameters of the atmosphere will then be taken from the input file and are constant over the whole $n_x \times n_y$ map.

An example of an ATM_INPUT file is contained in the HELIX+ distribution:
./sample_data/atm_init.fits

7.8 Weighting

The weighting function defines the wavelength dependent weighting for the I, Q, U and V Stokes vector. It is defined in a file defined with the WGT_FILE keyword. The file must be located in the directory defined with the WGT keyword (default: ./wgt/. Multiple weighting files can be specified when running the code with multiple iterations.

The default weighting file is he_default.wgt:

```
;define weighting function for helix
;Syntax:
;StokesPar   wgt    WLmin    WLmax
;you can define an arbitrary number of WL-ranges for every Stokes vector.
;The succeeding line will overwrite the preceeding weighting funtion if the
;WL-ranges overlap.
I            1.0    10829.6  10831.5    ;good region for I, no telluric blend
I            0.5    10832.5  10835.0    ;include region right of telluric blend

V            0.2    10828.5  10830.0    ;V-weighting for region with Ca-Blend
V            1.0    10829.6  10832.0    ;V-weighting right of Ca-blend

Q            0.2    10828.5  10830.0    ;Q-weighting for region with Ca-Blend
Q            1.0    10829.6  10832.0    ;Q-weighting right of Ca-blend

U            0.2    10828.5  10830.0    ;U-weighting for region with Ca-Blend
U            1.0    10829.6  10832.0    ;U-weighting right of Ca-blend
```

The keyword IQUV_WEIGHT multiplies the weighting function of every Stokes parameter with the specified value.

7.9 Fitness Function

The fitness function f used for the Pikaia algorithm was defined as

$$\frac{1}{f} = \chi^2 = \frac{1}{N}\sum_{n=1}^{N}\sum_{S} w_S \left(S_{Obs} - S_{Fit}\right)^2, \tag{1}$$

where n is the index number of the wavelength bins, S is the index for the Stokes vector (I, Q, U and V), and $w_S(n)$ the wavelength-dependent weighting scheme defined using the weighting file described in Sect. 7.8.

Details can be found in `piktools.f90`, function `fitness`.

7.10 Constant property slab model for the correct treatment of the Hanle effect in the He 10830 line

Implemented!

Documentation will follow.

Refer to Lauras thesis and to *Asensio Ramos et al.* [2008]; *Trujillo Bueno et al.* [2005].

7.10.1 Correct determination of emission vector

Since the Hanle effect strongly depends on the direction of illumination a correct determination of the emission vector is essential. The emission vector is specified by 3 angles, described in Fig. 1 of *Asensio Ramos et al.* [2008]:

- Θ - heliocentric angle (keyword in input file: `HELIO_ANGLE`):
  $\Theta = asin(\sqrt{x^2+y^2}/r)$,
  with (x, y) = position of observation in arcseconds, r = solar radius. Θ is 90° for off-limb observations and $< 90°$ for observations on disk.

- χ - LOS orientation. This angle is usually set to 0 (i.e. the line-of-sight direction is contained in the $x - z$ plane of Fig. 1 of *Asensio Ramos et al.* [2008]). There is no input file keyword for this angle.

- γ - LOS orientation of positive Q direction. For TIP data this angle is calculated as follows: The positive Q direction for TIP is always terrestrial N-S. Therefore, the $+Q$ to limb direction (`+Q2LIMB`) is given by
  `+Q2LIMB` $= atan(x/y)$, therefore the angle γ is given by
  $\gamma = 360. -$ `+Q2LIMB` $+ 90. = 360. - atan(y/x) + 90.$
  In the input file you must directly specify the positive Q to limb angle using the keyword `+Q2LIMB`.

Important: In the Hanle diagnostics the magnetic field direction (parameters `AZI` and `GAMMA`) retrieved from an inversion is in solar coordinates! An inclination angle of `GAMMA`= 0° corresponds to a magnetic field vertical to the solar surface (90° is parallel to the solar surface). The 0°direction for the azimuth is defined by the plane between the solar vertical and the LOS direction.

To rotate the azimuth back to the reference frame of a typical TIP map (with $+x$ (=the scan direction perpendicular to the slit direction) defining azimuth 0°, angles increase counterclockwise) the following steps must be done:

- rotate AZI to VTT +Q direction: $\chi_1 = \texttt{AZI} + 90 + \texttt{+Q2LIMB}$ and
- rotate the map according to the scan direction. The scan direction should always be perpendicular to the slit orientation, the slit orientation is always terrestrial N-S: $\chi_{TIPmap} = \chi_1 + (\texttt{SLIT\_ORIENTATION} - 180) + 90$.

This rotation (χ_{TIPmap} =AZI++Q2LIMB+(SLIT_ORIENTATION) is implemented in the IDL routine `azi_qu2xy.pro`.

Note: this description is only valid for TIP data. Applying it to other data sets needs thorough analysis of the geometry.

7.11 Correction for Scattering-Polarization

Note: The Hanle-slab model described in Sect. 7.10 is the correct choice for treating the effects of scattering and atomic polarization correctly! This subsection is describing the old method, applied in *Solanki et al.* **[2003] and is only valid for a very special geometry.**

The analysis of the data set 13may01.014 showed, that the Q and U profiles can be affected by scattering-polarization. The unpolarized light beam from the solar surface hits the atoms in higher layers. The scattered light will show a linear polarization signal due to two different effects (see *Trujillo Bueno et al.* [2002]):

- the Hanle effect modifies the atomic polarization of degenerate atomic levels due to the magnetic field. The magnetic field vector has to be inclined significantly for this effect to operate
- anisotropic illumination of the atomic system (close to or above limb) will cause an imbalance in the population of the sublevel of degenerate atomic levels.

Both effects produce a linear polarization signal. The resulting Q and U signals can be used to calculate the magnetic field orientation and the magnetic field strength by applying complicated polarization diagrams (eg. *Nagendra et al.* [1998]; *Faurobert-Scholl* [1992]; *Bommier et al.* [1991]).

In the special case of a LOS-direction perpendicular to the solar surface (disk center) and a magnetic field direction parallel to the solar surface (horizontal field) the polarization signal due to the Hanle effect is oriented along the magnetic field direction:

$$\tan(2\phi) \;=\; (U/Q) \tag{2}$$

This relation gives reasonable results for horizontal fields down to a μ-angle of >0.6 (Javier Trujillo Bueno, personal communication).

To correct for this problem, we made the following approach:

- The unpolarized radiation is assumed to show a Gaussian line profile with a line width and a velocity shift calculated from the I profile.
- a Gaussian profile with a free amplitude A_0 but fixed values for width and line shift (from I) is fit to the Q and U signals simultaneously with the fit of all other free parameters of the inversion.
- the Gauss profile is distributed to Q and U according to the preferred direction (either B-field or limb direction)

- every line of a multiplet can have a separate amplitude.

To apply this correction in the inversion the following line in the 'DATA SET' part of the input file is needed:

STRAYPOL_CORR	50	B	iteration steps and orientation of scattering-polarization correction. Orientation: 'B' = along B-field, 'X' = a number defining an angle manually

The first number gives the number of iterations for the Gauss fit to the I profile (determination of with & velocity shift). The second entry can be either 'B' for a correction according to Hanle, or a number in degrees giving the angle ϕ for a manual correction.

STRAYPOL_AMP	0.0015	This defines the straypol-amplitude to be used for a synthesis. Not used for fitting!

7.12 Multi-Iteration Method

The multi-iteration method is a technique to change the parameters relevant for the minimization after a certain number of iteration steps. The idea is to start with a simple atmospheric model to fix for example line-of-sight velocity and line broadening. In a next step a detailed analysis of the magnetic field direction can be performed.

Example: The minimization should start with the approximation for directions. After 200 iterations we consider all parameters to be determined except for the magnetic field, which should now be treated as a free parameter (no approximation anymore!). The implementation in the input file is:

NCALLS	200	50	number of iterations in PIKAIA routine.

The first Pikaia run will have 200 iterations, the second run has 50 iterations.

Now we must change the lines defining the atmosphere:

BFIEL	200	0.0	2000	1	20	1	magnetic field strength
AZIMU	0.0	-90	90	-2	100	-2	azimuthal angle
GAMMA	0.0	0	180	-3	100	-3	inclination angle

These lines do the following: The first 200 iterations will be calculated in the usual manner. The result of this run is then used as the initial guess for the second run. The second run now only changes the parameters BFIEL, AZIMU and GAMMA. The value for BFIEL is allowed to vary 20% of the initial scaling range around the new initial value. Azimuthal and inclination angle are completely free (100% variation allowed). If a second magnetic component is present, then the '-' sign would indicate a coupling between the magnetic field direction of both components.

All other atmospheric parameters are left unchanged during the last 50 iteration steps.

In order to use the approximation for the first 200 iterations but not for the last 50, we have to set the line:

APPROX_DIR	1	0	use approx. for first run only!

The following keywords can be set differently for multiple iteration runs: BFIEL, AZIMU, GAMMA, VELOS, WIDTH, AMPLI, VDAMP, VDOPP, ZERO, SGRAD, EZERO, ALPHA, APPROX_AZI, APPROX_DIR, APPROX_DEV_INC, APPROX_DEV_AZI, IQUV_WEIGHT (for example, use 8 values for 2 multi-iterations), MAGOPT, NCALLS, METHOD, WGT_FILE.

7.13 Applying Different Atmospheres

With a small driver program it is possible to apply different model atmospheres for one data set depending on the results of a first run. This allows to invert the data set with a simple model and apply a more complex model only to the some pixels of the data set. This driver program is written in IDL, it therefore works only with the IDL version.

The following IDL-code is an example for such a driver program (see driver.pro in idlpro/ directory):

```
;program to perform more complex Helix depending on result of
;simple helix
pro driver


                                ;do simple helix
  helix,ipt='emf_1comp_abs.ipt',savall=savall
  restore,savall

                                ;check for parameters which make a
                                ;more detailed analysis necessary
  idx=where(pikaia_result.fit.atm.par.vlos gt 5000.)


  if idx(0) eq -1 then return


                                ;call more complex helix
  list=transpose([[pikaia_result.fit(idx).x],[pikaia_result.fit(idx).y]])
  atm_ini=pikaia_result.fit(idx).atm
  helix,ipt='emf_2comp_abs.ipt',list=list,atm_ini=atm_ini

end
```

First, an inversion according to an input-file emf_1comp_abs.ipt is performed. The results of this inversion are stored in a sav-file, the name of this sav-file is returned in the variable savall. This sav-file is restored.

Then we check for the parameter which decides whether we want a more complex inversion or not. In this case, all points with line-of-sight velocities greater than 5000 m/s should be picked out.

A list of the x and y-pixels for these points is created (list). The result of the simple 1-component inversion is used as an initial guess for the more complex, 2-component run defined in the input-file emf_2comp_abs.ipt.

7.14 Paschen-Back Effect

The parameter USE_PB forces HELIX$^+$ to include the tabulated version of the Paschen-Back effect for the He 10830 line [*Socas-Navarro et al.*, 2004]. Two methods are implemented:

- PB_METHOD poly uses the polynomials to fit the deviation between the Zeeman levels with and without Paschen-Back effect [*Socas-Navarro et al.*, 2005]

- `PB_METHOD table` uses a tabulated version and a quadratic interpolation to calculate splitting and strength of Zeeman sublevels [*Socas-Navarro*, 2005].

7.15 Running HELIX+ in Batch Mode

This chapter applies to the IDL version only. The FORTRAN 90 version can easily be run in batch mode using the standard UNIX `at` command.

A small shell-script allows to run HELIX+ within a batch-file. The shell script is in the root-directory of the HELIX+ program and is called `batch`.

You can run several input files subsequently (nice for long weekends). The file is based on the `run` script file. The command to call several input files is:

```
./batch ipt-file1.ipt ; ./batch ipt-file2.ipt ; ./batch ipt-file3.ipt
```

It is a good idea to set the parameters `DISPLAY_PROFILE` and `DISPLAY_MAP` to 0. X-display export might not work in batch mode. Also it is recommended to set the verbosity to 0, otherwise a huge log-file might be created.

7.16 Azimuth Correction

The azimuthal angle returned by the HELIX+ program lies between -90° and +90°. It is not corrected for the 180° ambiguity of the Zeeman-analysis. An azimuthal angle of 0° points to the $+Q$ direction. The $+Q$ direction for TIP is always oriented along terrestrial north-south direction.

To convert this angle to the x- and y-axis of the observation the procedure `azi_qu2xy.pro` can be used. The conversion in this routine is:

- rotate the angle from the $+Q$ direction to the $+y$-axis: $\phi_1 = \phi_Q + (\alpha_{Slit} - 180)$ where α_{Slit} is the slit orientation as given in the header of the observation files.
- rotate the angle to the $+x$-axis: $\phi_{xy} = \phi_1 - 90$.

In this system an angle of 0° points to the $+x$-axis, and $+90°$ point to the $+y$-axis.

7.17 Display results

7.17.1 Structure of output data

The results of an inversion are stored in the directory defined with the `ATM_ARCHIVE` variable (default: `./atm_archive/`. Profiles and atmospheres are stored for every single pixel of the inverted map. The filenames for the final atmospheres / fitted profiles are:

```
./atm_archive/observation_ATM_SUFFIX/x000/x000y000_atm.dat
./atm_archive/observation_ATM_SUFFIX/x000/x000y000_profile.dat
```

for pixel x=0 and y=0 of your map.

7.17.2 Prepare data for IDL display routine using `make_sav.pro`

The IDL routine `make_sav` is used to convert the directory structure returned by the FORTRAN 90 or IDL inversion (directory `./atm_archive/`) to an IDL sav-file. The usage of this routine is:

```
make_sav,data_dir='atm_archive/atm_13may01.014'
```

This will create a sav-file from the specified directory tree.

You can combine the results of different runs into one sav-file. This is recommended if you run for example a 1-component inversion for the whole data set and a 2-component inversion for a small part of this data set. The command to combine the results of two inversions is:

```
make_sav,data_dir='atm_archive/'+ \
    ['atm_13may01.014_comp1','atm_13may01.014_comp2']
```

This will first read the directory `atm_archive/atm_13may01.014_comp1`. Every pixel not containing a valid fit (i.e. with fitness=0) will then be replaced by the results stored in the directory `atm_archive/atm_13may01.014_comp2`. The number of components in the resulting sav-file is equal to the maximum number of components of the individual inversions.

Be aware that the order of the directories is important: The pixels from the first directory are not overwritten by the results contained in the second or subsequent directory. There are 4 different methods to combine the results from different directories:

[f] best fitness: store atmospheres with best fitness.

[r] fitness ratio: overwrite atmosphere of first directory only when fitness of 2nd (3rd...) directory is better by a specified factor.

[o] overwrite: overwrite atmospheres from first directory with results from 2nd (3rd, ...) directory.

[z] zero fitness: only overwrite the atmospheres where the fitness is zero (i.e. no atmosphere present).

You will be prompted to enter the corresponding method interactively when running `make_sav`.

The filename of the newly created sav-file will be constructed from the name of the observation and the atm-suffixes of the output directories for the atmospheres (`ATM_SUFFIX` keyword).

7.17.3 Plot Fitted Profile of a Map

To plot the fitted profile from a single pixel of an inverted map you can use the IDL-routine `get_stokes.pro`. The syntax of the routine is:

```
fitprof=get_stokes(result=pikaia_result,x=0,y=0,/show,verbose=0)
```

The structure `pikaia_result` is the output of the routine `make_sav` (see Sect. 7.17.2). The keywords `x=0, y=0` specify the pixel of the map and the keywords `/show, verbose=0` control if the profile is printed to the screen an the verbosity.

The atmosphere and the input file parameters controlling the calculation of the synthetic profile are taken from the result stored in the IDL-structure `pikaia_result`.

7.17.4 Print Profile

Every plot of profiles on the screen can easily be printed by just typing `p` on the IDL command line. `p` is a shortcut to the widget `profile_settings` and opens a widget where you can change some parameters concerning the layout of the plot.

7.17.5 Widget Application `xfits.pro`

`xfits` is an IDL based application to display the FITS files resulting from HELIX$^+$ (see Sect. 9.1) and SPINOR inversions.

```
IDL> xfits
xfits - data analysis tool to display inversion results
        stored in FITS files
Usage:
xfits,'atm_archive/atm_fitsout.fits'      - reads in FITS file
xfits,'atm_archive/atm_fitsout.xfits.sav' - reads in sav file created with xfits
```

Note: only the fits file for the atmospheric results needs to be specified. The `xfits` code automatically selects the corresponding fitted profiles or uses HELIX$^+$ to recompute the profiles from the atmospheric parameters.

The widget application is more or less self-explanatory. All parameters of the inversion can be displayed in a grid-like format. Ranges, contours can be defined by right-clicking on the selected plot in the `XFITS-Plot` widget. The layout of a plot can be stored into an IDL sav-file.

`xfits` also contains the tools to analyze the statistical errors computed using the methods described in Sect. 7.18.1 and Sect. 7.18.2. A corrsponding dropdown menu appears automatically if the fits file contains the statistical information.

7.18 Statistical analysis of Errors

There are two methods to estimate the statistical error of the fit parameters. Method one (Sect. 7.18.1) is based on the random number nature of the PIKAIA algorithm. Method 2 (Sect. 7.18.2) is based on the fitness of the individual populations of one PIKAIA iteration.

7.18.1 `PIXELREP` method

To the author's knowledge PIKAIA does not offer a way to perform an error propagation analysis for the retrieved parameters. However, since PIKAIA is based on random number generators, you can invert a pixel several times, the spread in the parameters gives you a kind of reliability. You can do this manually, simply by running an inversion several times for the same pixel of your map, and check the spread in the parameter values. Very quickly you will see which parameters are reliably retrieved (the values for every inversion do not change significantly; values are "robust"), and which parameters are fluctuating a lot.

HELIX$^+$ also offers a method to automate this process using the input file keyword `PIXELREP`. An example input file to perform a statistical analysis on a 4×2 sized region of a Hinode data set is shown in `stat_anal.ipt`. Running this example:

```
IDL> helix,ipt='stat_anal.ipt'
```

will invert every pixel of the region defined with XPOS and YPOS 30 times (PIXELREP is set to 30). The results are stored, as usual, in the directory specified with the ATM_ARCHIVE keyword. In this example, this is the directory ./atm_archive/atm_hinode_data_stat/. When looking into this directory you will find 30 files for every pixel of the map. The helper routine make_sav can be used to read in these files[2]:

```
IDL> make_sav,data_dir='atm_archive/atm_hinode_data_stat'
```

This command creates the IDL sav-file as usual (see Sect. 7.17.2). It also detects that the directory contains multiple results for one single pixel and stores this information into a separate IDL structure. You can access this information like this:

```
IDL> restore,/v,'./sav/hinode_data_stat.pikaia.sav'
% RESTORE: Portable (XDR) compressed SAVE/RESTORE file.
% RESTORE: Save file written by lagg@lxlagg, Thu Feb 14 09:54:25 2008.
% RESTORE: IDL version 7.0 (linux, x86_64).
% RESTORE: Restored variable: PIKAIA_RESULT.
% RESTORE: Restored variable: STATISTIC.
IDL> print,size(statistic)
           2           4           2           8           8
IDL>  help,/st,statistic
** Structure <a9e698>, 6 tags, length=4356, data length=4356, refs=1:
   MEAN            STRUCT    -> <Anonymous> Array[2]
   VAR             STRUCT    -> <Anonymous> Array[2]
   SKEW            STRUCT    -> <Anonymous> Array[2]
   KURT            STRUCT    -> <Anonymous> Array[2]
   DIFF            STRUCT    -> <Anonymous> Array[2, 30]
   N               LONG                30
```

The IDL structure statistic contains the mean, variance, skewness, kurtosis for every pixel of the map, as well as the difference of every single run to the mean value (for a definition of the moments see the IDL help to the IDL command moment). The size of the structure is 4×2 , according to the definition of the XPOS and YPOS keywords in the input file.

A summary report of the statistical data contained in the sav-file can be created using the IDL routine stat_summary.pro. The syntax is:

```
IDL> stat_summary,'./sav/hinode_data_stat.pikaia.sav'
```

There is no display routine for the statistical data yet. The following example illustrates how to use the statistical data:

```
IDL> sdevb=sqrt(statistic.var(0).b)
IDL> plot,statistic.mean(0).b
IDL> errplot,statistic.mean(0).b-0.5*sdevb,statistic.mean(0).b+0.5*sdevb
```

This example plots the standard deviation of the magnetic field of the first atmospheric component (index 0) of the 30 inversion results for the 8 pixels, specified with XPOS and YPOS in the input file, as error bars on top of the mean value.

Note: All pixels for which the statistical analysis should be performed, must be run with the same number of pixel repetitions (PIXELREP keyword). Otherwise the routine make_sav will not work properly. If the output directory of your inversions contains results with different values for PIXELREP you have to copy the results with the same number of pixels to a new output directory or delete the results with other PIXELREP values.

[2]It is recommended to use FITSOUT 1 to store the data into fits files. In this case the size of the fits file grows by the factor PIXELREP. The helper routine xfits should be used for the statistical analysis of the data.

7.18.2 PIKAIA_STAT method

This method is based on the fact that for every PIKAIA iteration a population of solutions is required (the so-called "genes", one complete set of fit parameters describing the atmosphere). The number of genes in a populations per PIKAIA iteration is defined automatically within HELIX+. It can be set manually using the input file keyword PIKAIA_POP. A typical value lies between 30 and 100.

This method only works for output in fits-files (FITSOUT 1) and does not work in combination with the PIXELREP method described above.

The input file parameter PIKAIA_STAT switches this statistical method on and allows to control its beahvior. Example:

PIKAIA_STAT	POP75	Switch on and control statistic module based on PIKAIA populations. Requires FITSOUT=1 and PIXELREP=1. Available modes are: NONE (=default), POPxx Example: POP75 - take the fittest 75% of the PIKAIA population.

With this keyword the statistical method based on the PIKAIA genes of the last population is switched on. The best (i.e., the fittest) 75% of the genes of the last PIKAIA iteration are then stored into the output files (atm_ and prof_ fits-files). If you want to store all the genes, then set this keyword to POP100.

The file size of the output fits files grows significantly: instead of storing the result of only the fittest gene, 75% of the whole population are stored. If the population is 100, then the increase in file size is a factor of 75.

It is recommended to analyze the statistical data for both methods (Sect. 7.18.1 and Sect. 7.18.2) with the xfits tool (Sect. 7.17.5).

7.19 De-speckling of Maps

When inverting a whole map of an observation it is unavoidable that some pixels are not converting to the optimal parameters. The routine despeckle.pro can be used to run the HELIX+-procedure on these speckles. The call of the despeckle routine is:

```
despeckle,sav='savname.sav',par='inc',dev_perc=10.
```

This example runs the speckle-filter for the results of a map (stored in the sav-file savname.sav). The area used for despeckling is defined by clicking on the map. It checks for speckles in the parameter INC (mag. field inclination) and allows a deviation of the surrounding pixels of 10%. If the deviation is larger, then the pixel is recalculated using the original input-file parameters. If the recalculation gives a better fitness, the original result will be replaced.

The parameter par='inc' can be replaced by a vector of parameters in order to take into account the smoothness in more than one parameter, eg: par=['inc','fitness','azi'].

The despeckle routine can also be used on pixels with a low fitness. The call

```
despeckle,sav='savname.sav',minfit=1.5,input='xdisplay.ipt'
```

will re-run all pixels with a fitness of less than 1.5 using the input file xdisplay.ipt.

Further parameters / keywords to despeckle are:

`/full`	Apply despeckling to whole map (no selection using mouse).
`position=[bx,by,ux,uy]`	Apply despeckling to selected region.
`/all`	Apply despeckling also to pixels surrounding the 'bad' pixel.
`maxval=0.55`	If value of the parameter is larger than `maxval` the pixel is recalculated. Must have same number of elements as `par`.
`minval=0.20`	Similar to `maxval`.
`comp=1`	Analyze only the map of the atmospheric component 1. Default is to use the first component, which is component 0.

The despeckle routine displays the pixels to be recalculated on a map. If you are happy with the selection of the pixels the code writes out a list of bad pixels to `despeckle.list` which can be used in an input file with the `PROFILE_LIST` keyword. If you answer the question 'Starting helix using ipt=...ipt for speckles [Y/N]?' with 'Y' HeLIx$^+$ will run in IDL mode and calculate marked pixels.

8 Input Data Formats

The input-file keyword `OBSERVATION` defines the file / directory from which the profiles for the inversion should be read. The path to the observation files can be specified using the keyword `PROFILE_ARCHIVE`. HeLIx$^+$ can handle the following data formats:

8.1 ASCII-profiles (file extension `*.dat`)

This format can be used to read in single profiles (1 full Stokes vector) of the following format (adopted from the SPINOR code):

```
  1  'WIVQU'    issi_synth_01 ./input/issi_synth_01.ipt 20110203
    1       6301.5000   201     1.00000E+00  |      1  201    1  201   ''
  -5.000000E-01   9.972506E-01   2.249229E-04  -1.371022E-05   8.473638E-06
  -4.900000E-01   9.971439E-01   2.382306E-04  -1.476878E-05   9.180436E-06
  -4.800000E-01   9.970307E-01   2.526207E-04  -1.593275E-05   9.962267E-06
.... (201 lines) ....
```

8.2 IDL-save files (file extension `*.sav`)

Available in the IDL version only! The IDLsave file must contain a structure like this:

```
IDL> help,/st,observation
** Structure <2737908>, 6 tags, length=4832, data length=4828, refs=1:
   IC              FLOAT           1.00000
   WL              DOUBLE    Array[201]
   I               FLOAT     Array[201]
   Q               FLOAT     Array[201]
   U               FLOAT     Array[201]
   V               FLOAT     Array[201]
```

8.3 4D FITS files

This is the recommended input format. The extension of the filename must be `.fits`. It is identical to the SPINOR profile format. The dimensions of the FITS file are `NWL×NSTOKES×NX×NY`. The default order for the Stokes parameters is I, Q, U, V

Optional keywords are:

```
STOKES  = 'IQUV    '           / order of stokes parameters
XOFFSET =                    0 / X-offset
YOFFSET =                    0 / Y-offset
```

The wavelength information can be provided either by FITS extensions (see below) or by the keywords (wavelength values in Å):

```
WLREF   =       6.30150120E+03 / reference wavelength
WLMIN   =      -1.00000000E+00 / min. wavelength
WLMAX   =       2.75000000E+00 / max. wavelength
NWL     =                  376 / WL-points
```

8.3.1 FITS extensions

Several FITS extensions are recognized:

Wavelength vector Must be a 1D vector of length `NWL` containing the wavelength positions. The header must contain the keyword `WLREF` defining the reference wavelength.

Wavelength array Must be a 3D array of size `NWL×NX×NY` containing the wavelength positions for all x and y positions in the map. The header must contain the keyword `WLREF` defining the reference wavelength.

Continuum Image Must be a 2D array of size `NX×NY` containing the continuum level for every pixel in the map.

8.4 Standard FITS files (extension `*.fits`)

HELIX$^+$ can directly read standard FITS files in the format described in Sect. 8.5.1. These FITS files must have 3 dimensions: `NAXIS1` = number of wavelength points, `NAXIS2` = y-dimension (for slit instruments the dimension along the slit), `NAXIS3` = Stokes Parameter and x-direction. This is the standard definition for FITS files from the TIP instrument, for details see Sect. 8.5.1.

8.5 TIP-files

TIP data files reduced with the standard TIP data reduction routines can be directly accessed from HELIX$^+$. The data file must follow the standard TIP filename convention (e.g. 07jun10.002cc). Wavelength calibration and continuum image can be contained in an auxiliary data file (extension ccx, see Sect. 8.5.2).

8.5.1 The TIP FITS file

HELIX+ was developed for the polarimetric data from the TIP instrument. In order to use HELIX+ for other data sets it is recommended to use the same FITS format. This section describes the TIP FITS format.

Important: The filename must end with the letters 'cc', indicating that the data file is calibrated and cross-talk removed. Example: `myfitsfile.fitcc`.

The header of the FITS file is as follows:

```
SIMPLE  =                    T / FITS standard
BITPIX  =                   32 / 4 byte twos-compl. signed integers
NAXIS   =                    3 / number of axes
NAXIS1  =                 1009 / lambda
NAXIS2  =                  454 / y
NAXIS3  =                   12 / number of pixels in x-direction
```

The example above contains the Stokes parameters in the following format:

- NAXIS1: this field contains the number of pixels in the wavelength direction
- NAXIS2: this field contains the number of pixels in the y-direction. For TIP data, this is the slit direction
- NAXIS3: this field contains the number of pixels in the x-direction ×**4!** For every pixel in x the four Stokes vectors must be present in the order: I, Q, U, V

Note: If no `ccx` file is present, the wavelength calibration must be specified in the input file using the parameters `WL_OFF` and `WL_BIN` (values given in Å)! Sect. 8.5.2 describes the contents of the `ccx` file.

The following IDL example describes how to create a correct FITS file. As input data it uses a TIP data file. As output, it creates the same data file. The user can apply corrections to the data (eg. offset-corrections) by changing the `data` variable.

This example also demonstrates how to create a data set in TIP format from other instruments. The user only has to change the values of `NAXIS1-3` and to define the variable `data`. The example is contained in the HELIX+ distribution, file: codeidlpro/create_fits.pro.

```
;example to create fits file in the format of TIP.
pro create_fits

                                ;get data: Read in a TIP-FITS file
  data=readfits_ssw('/data/slam/VTT-data/VTT-Apr07/02May07/02may07.004-04cc', $
                   header)

                                ;now data contains an array with 1009
                                ;WL-pixels, 454 pixels in y direction
                                ;and 12 pixels in x-direction (3x4
                                ;Stokes vectors)
  szd=size(data)
  print,'============ SIZE ======================'
  print,'Size: ',szd
  print,'WL-pixels: ',szd(1)
  print,'y-pixels: ',szd(2)
  print,'x-pixels: ',szd(3)


  fitsfile='~/tmp/test.fitscc'
  print,'======= WRITE NEW FITS ================='
  print,'File: ',fitsfile

  writefits,fitsfile,data,header,append=0
  print,'Done.'
end
```

8.5.2 The Auxiliary Data File (ccx file)

The purpose of the auxiliary data file is to define

- the wavelength calibration and
- the continuum level for every single pixel.

The ccx file is a standard FITS file with the following header:

```
SIMPLE  =                    T / Written by IDL:  Thu May  3 09:15:45 2007
BITPIX  =                  -32 / Number of bits per data pixel
NAXIS   =                    2 / Number of data axes
NAXIS1  =                    3 / x-size of cont. image
NAXIS2  =                  454 / y-size of cont-image
WL_NUM  =                 1009 / WL-bins
WL_OFF  =       15643.52386351 / WL-Offset
WL_DISP =           0.02317366 / WL-Dispersion
```

It contains one image of size NAXIS1×NAXIS2 containing the continuum level for every single profile. The header values WL_NUM, WL_OFF and WL_DISP contain the number of wavelength points, the starting value for the wavelength vector and the dispersion (increment for every wavelength pixel) respectively.

For an irregular wavelength grid the wavelength calibration can be defined in the first extension of the FITS file. This can be done by using the writefits.pro file from Solar Software:

```
IDL> wlvec=[15650.0d, 15650.2d, 15650.5d ,15650.6d, 15650.8d]
IDL> writefits,'name_of_file.001ccx',wlvec,/append
```

This will create an extension to the FITS file which will be used as the wavelength calibration. In this example the wavelength vector has 5 elements.

HELIX+ will always try to use the **first extension** in the FITS file as the wavelength calibration. If this information is not contained in the .ccx FITS file then the FITS header values WL_OFF and WL_DISP are used. **Note:** The values WL_OFF and WL_DISP in the input file have highest priority. These values will be used for the wavelength calibration if present in the input file!

8.6 Hinode fits files

If the input-file keyword OBSERVATION is a directory then the code searches for files *SP3*.fits or *SP4*.fits. Wavelength calibration and continuum image can be specified using auxiliary data files (see Sect. 8.6.1).

8.6.1 Using Hinode Data

The example ex_hinode.ipt (Sect. 7.2) shows how to work with Hinode data. The code is able to directly read in the calibrated Hinode FITS files. The code needs additional information in the format of the ccx files described above. These files are created using make_ccx.pro contained in the ./idlpro/ directory of your distribution.

You can test the usage of the make_ccx.pro routine by applying it to the Hinode sample data set contained in the distribution:

```
IDL> make_ccx,hinode_dir='./sample_data/hinode_data/',level_quiet=0.001
```

make_ccx performs a Voigt function fit to the iron lines to determine the wavelength calibration. For this calibration it uses an average over the 'most quiet' profiles along one slit position (all profiles with magnetic signals smaller then the value defined with the level_quiet keyword). It also determines the local continuum value, the average continuum value along the slit and the average continuum value for the whole image.

It is recommended that you place the Hinode FITS files (*.fits and *.fits.ccx) you want to invert in a separate directory. Then you can specify the pixel which should be inverted using the keywords XPOS, YPOS and HIN_SCANNR. YPOS defines the pixel along the Hinode SOT-SP slit, XPOS defines the value of the fits header variable SLITINDX. If there is more than one fits file with the same value of the header variable SLITINDX (e.g. for repetitive scans over the same region), you must use the keyword HIN_SCANNR to specify which scan you would like to use.

For Hinode data the usage of the local straylight (see keywords LOCALSTRAY_xxx) mode is highly recommended . This mode computes a straylight profile from the surrounding pixels. Additionally, NORM_CONT should be set to IMAGE (or SLIT) and the Voigt-S_0 parameter mode should be used. This should lead to results similar to the MILOS inversions described by *Orozco Suárez et al.* [2007]. A comparison of both inversion procedures was not done yet, the HeLIx$^+$ author would appreciate if someone could do this!

8.6.2 Hinode Fixed Slit data

If Hinode SOT/SP performs a "fixed slit scan" then the header variable SLITINDX is zero and the scanning steps are contained in the header variable HIN_SCANNR. HeLIx$^+$ automatically detects Hinode fixed slit scans if the Hinode data directory ONLY contains Hinode FITS files in fixed slit scan mode!

If a Hinode "fixed slit scan" is detected, the input file variable XPOS refers to the value of the header variable HIN_SCANNR.

Note: HeLIx$^+$ does not have information on the surrounding pixels in x-direction when a "fixed slit scan" was performed. Therefore the local straylight correction does not have the necessary information to work accurately. The calculation of the local straylight profile is done in the usual way by computing the weighted average over the surrounding profiles.

8.7 Sunrise IMaX data

If the filename contains the string imax the file is treated as a data file from the IMaX instrument on Sunrise.

8.8 Using CRISP data

The routine make_ccx also works for SST-CRISP data. Pleas proceed in a similar way as for Hinode SOT/SP data (see Sect. 8.6.1). Currently only data obtained in the V5-6 and the L12-2 mode are implemented. The file format is a 4-dimensional fits-file: NAXIS1 = x-direction, NAXIS2 = y-direction, NAXIS3 = Wavelength pixels, and NAXIS4 = Stokes Parameter (V5-6 mode I,Q,U,V, L12-2 mode I,V).

9 Output Data Formats

The standard output method is to write the best fit parameters of every inverted pixel into a single ascii file. This file will be located in a subdirectory of the `ATM_ARCHIVE` directory. The directory name for this subdirectory is constructed like this:
atmospheres: `$(ATM_ARCHIVE)/atm_$(OBSERVATION)_$(ATM_SUFFIX)`
profiles: `$(ATM_ARCHIVE)/prof_$(OBSERVATION)_$(ATM_SUFFIX)` The subdirectory will contain the input-file (renamed to `input.ipt`) and directories containing the atmospheric data files (=inversion results). Using the flag `SAVE_FITPROF` allows to additionally write out the fitted profiles.

9.1 FITS Output

Especially when inverting maps it is highly recommended to write out the data as FITS files. This is achieved by setting the keyword `FITSOUT` to `1`. The file name of the FITS file is:
`$(ATM_ARCHIVE)/atm_$(OBSERVATION)_$(ATM_SUFFIX).fits`

FITS file format (atmospheres):

Header: contains size info, offsets, the name of fitted parameters, and all ascii files required for the inversion (input file, atomic data file(s), weight file(s), convolution- and prefilter files as ascii table extensions).

Data: 3D array containing for every (x, y)-position a list of fitted parameters and the fitness as specified in the header (dimension `NPAR×NX×NY`). All free and coupled parameters are written out. Atmospheric and line parameters are ordered the same way as in the input file. This means, that e.g. the first occurrence of the parameter `BFIEL` correspond to the first atmospheric component defined in the input file. If the multi-iteration method was used (Sect. 7.12) then this parameter list is repeated for every multi-iteration step.

FITS file format (profiles):

Header: contains size info, offsets, number of wavelength points, order of Stokes vector (default: $IQUV$). The first image extension contains the continuum levels (one value for every profile, dimension: `NX×NY`), the second image extension contains the wavelength vector (`NWL×NX×NY`)

Data: 4D array containing for every (x, y)-position a list of fitted parameters and the fitness as specified in the header (`NWL×4×NX×NY`). All free and coupled parameters are written out. Atmospheric and line parameters are ordered the same way as in the input file. This means, that e.g. the first occurrence of the parameter `BFIEL` correspond to the first atmospheric component defined in the input file. If the multi-iteration method was used (Sect. 7.12) then this parameter list is repeated for every multi-iteration step.

Automatic Extension of Results

For a new run of HELIX$^+$, an existing FITS file is automatically extended (i.e. to a larger area) or updated (pixels which were already inverted) if

- the output file name is the same,

- the number of free parameters did not change,
- the list of free parameters did not change, and
- the number of wavelength points of the output did not change.

The parameter KEEPBEST can be used to control the behavior for overwriting already inverted pixels. If KEEPBEST is set to 1 then the atmosphere and the profile in the FITS file are only replaced if the fitness of the new inversion resulted in a higher fitness.

If the FITS file cannot be extended or updated, HELIX+ stops to prevent accidental overwriting of inverted maps.

The IDL widget application xfits (see Sect. 7.17.5) allows for convenient display of the inversion results.

10 Tips & Tricks

10.1 Writing out synthetic profiles

You can write out a synthetic profile by setting the SYNTH keyword to 1, the SAVE_FITPROF to 1 and the number of iterations NCALLS to 0. The synthetic profile is written to
./profile_archive/ipt-name.profile.sav (IDL-version) and
./atm_archive/ipt-name.profile.dat (FORTRAN 90-version).

10.2 Convergence

Several fitting parameters influence the shape of the spectrum in a similar way: the damping VDAMP, the amplitudes of components of propagation matrix EZERO and the source function gradient SGRAD are all changing the depth of the absorption signature. Test runs showed that the convergence significantly improves when fitting only one of these parameters and setting the other two parameters to a fixed, intermediate value. Good results were achieved with the following settings:

```
VDAMP       0.35      0.00      0.70  0 ;damping constant (Voigt only)
VDOPP       0.30      0.01      0.70  1 ;doppler broading (Voigt only)
EZERO       2.00      0.00     10.00  0 ;amplitude of components of
                                        ;  propagation matrix (Voigt only!)
SGRAD       1.00      0.00      8.00  1 ;gradient of source function
ALPHA       1.00      0.00      0.00  0 ;Filling factor for this component
```

In this example the depth of the absorption is controlled by SGRAD and the width of the line by VDOPP. EZERO and VDAMP are fixed to 2.0 and 0.3 respectively. Remember that this setting is a trick to increase the stability of the inversion. Parameters like magnetic field or velocity are retrieved more reliably. The physical meaning of VDAMP, EZERO, SGRAD and VDOPP is changed!

10.3 Filling Factor

In a 2-component model the filling factor ALPHA is changing the amplitude of the 2 components relative to each other. This is very similar (but mathematically not identical!) to a change in the source function gradient SGRAD.

Stability of the fit greatly increases when coupling `SGRAD` for both parameters and fitting only the filling factor `ALPHA`. This is the Recommended technique when fitting multiple components, especially in the He I 1083 nm line.

Another method is to set the filling factor to a fixed value and allow `SGRAD` to be free for all components. Then the ratio of `SGRAD` for the two components can be taken as a proxy for the filling factor. When the filling factor `ALPHA` is constant for both components, the code uses the ratio of `SGRAD` for the different components as the filling factor.

Defining both, `ALPHA` and `SGRAD` as free parameters will likely cause a non-stable determination of both parameters.

A Installing FORTRAN 90

Intel-FORTRAN 90 Download F90 compiler for Linux at http://www.intel.com → software → Compilers → Fortran Compilers for Linux → Free evaluation Download
(direct link: http://www.intel.com/software/products/compilers/flin/eval.htm). Follow installation instructions.

GNU-FORTRAN 90 GNU-FORTRAN 90 can be installed using the package manager of your Linux-distribution.

B Installing DISLIN

Download DISLIN from http://www.dislin.de/ and follow installation instructions.

Note: DISLIN requires OpenMotif which is available from http://www.ist-inc.com/motif.

C Installing CFITSIO

The FORTRAN 90 version uses the library CFITSIO to read and write FITS file. The CFITSIO routines are automatically compiled with the using the makefile described above.

The CFITSIO libraries used in HELIX$^+$ were downloaded from
http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html. It is recommended to use your Linux package manager to install CFITSIO. You will need the packages `libcfitsio3` and `libcfitsio3-dev`.

Note: If you need to create files larger than 2^{31} bytes you may need to recompile CFITSIO manually with the flags `-D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64`. Also, your OS must have support for files larger than 2 GB.

D Installing OPEN MPI / MPICH

D.1 OPEN MPI

Follow the instructions on the OPEN MPI website for the installation: http://www.open-mpi.de/.

D.2 MPICH-3

A good description on how to use and install MPICH can be found on the MPICH website:
http://www.mpich.org/documentation/guides/.

Download the latest version from the MPICH website: http://www.mpich.org/downloads/. The GNU C-compiler `g++` and the Intel Fortran Compiler need to be installed first. Before starting the configure script, make sure that `ifort` is set up:

```
/opt/intel/bin/ifortvars.sh intel64
./configure --prefix=/opt/mpich-3
make
```

```
sudo -i
. /opt/intel/bin/ifortvars.sh intel64
make install
```

D.3 MVAPICH-2

HELIX+ also works with the MPICH implementation MVAPICH-2.
See http://mvapich.cse.ohio-state.edu/ for download / installation.

E Running HELIX+ on Clusters

E.1 Running HELIX+ using MPICH/MPICH-3

When inverting several pixels the use of MPICH is highly recommended. The MPICH version will divide the list of pixels to several machines, and/or on multiple CPUs of one machine. MPICH has to be installed on the calling machine (from now on called the master). It is not necessary to have MPICH on the slaves (the machines which are used for the computation). See Section D.

E.1.1 Passwordless authentication

MPICH can use several protocols to communicate with other machines. This description is based on the communication using `ssh`. You have to configure `ssh` for password-less authentication using a public / private key pair. To test if password-less authentication is working, type the shell command in a terminal on the master:

```
ssh user@slave1 ls
```

This should list the directory content of the home directory of user on slave1. If you are asked for a password then read the ssh-documentation on how to switch to password-less authentication.

The latest version of MPICH-3 uses HYDRA as the default process management framework. It uses existing daemons on nodes (e.g., ssh, pbs, slurm, sge) to start MPI processes. More information on Hydra can be found at http://wiki.mcs.anl.gov/mpich2/index.php/Using_the_Hydra_Process_Manager.

E.1.2 Defining the MPICH hosts

The machines used for MPICH are defined in a machine-file. The machine file can be selected with the keyword `-machinefile` (see Sect. E.1.2 for the syntax). A typical machine-file looks like this:

```
# Change this file to contain the machines that you want to use
# to run MPI jobs on.  The format is one host name per line, with either
#    hostname
# or
#    hostname:n
# where n is the number of processors in an SMP.  The hostname should
# be the same as the result from the command "hostname"
pulpo:8
mojo:48
```

In this example the master is pulpo (1 CPU), the slaves are pulpo (7 CPUs), mojo (48 CPUs each).

Note: Every slave has to know the name of the master and all other slaves. Update your `/etc/hosts` file accordingly (or use the IP-numbers instead of the hostnames)!

Testing your MPICH-3 environment

The inversion must be run in a directory where the necessary files are accessible from the master and all slaves. You cannot use a local directory of one of the machines. The paths to the input-files (eg. profile archive) and the output files (eg. atmosphere archive) have to be the same for all machines (master and slaves). A good choice for a common directory is `/scratch/user` since this directory can be mounted from every machine at the MPS.

The role of the master is only to distribute the pixels to the different slaves. This uses only very little computing power, therefore it makes sense to use the master also as slave (i.e. use 9 CPUs on pulpo, instead of the physically present 8 CPUs). Use the `top` command on the different machines (master and slaves) to check the performance of MPICH.

A small routine called `./bin/mpitest.intel.mpi` is available to test the MPICH environment). If you want to test your setup on, e.g., lx40 use the command:

MPICH-3 command:

```
lx40:/scratch/lagg/helix> /data/slam/software/mpich-3/bin/mpiexec \
         -machinefile $HOME/mpd.hosts -np 9 ./bin/mpitest.intel.mpi
```

This will produce the following output:

```
 Hello, I am the master: pulpo
Hello, I am slave#   2 of   8 on pulpo
Hello, I am slave#   4 of   8 on pulpo
Hello, I am slave#   1 of   8 on pulpo
Hello, I am slave#   3 of   8 on pulpo
Hello, I am slave#   6 of   8 on pulpo
Hello, I am slave#   5 of   8 on pulpo
Hello, I am slave#   7 of   8 on pulpo
Hello, I am slave#   8 of   8 on pulpo
```

Example 1

The command to run HELIX+ on 4 slaves (1 master + 4 slaves = 5 machines) is:

```
nice -15 /data/slam/software/mpich2/bin/mpiexec \
       -machinefile $HOME/mpd.hosts -np 8 \
       ./bin/helix.intel.mpi -i mpitest.ipt
```

All slaves have to be accessible via ssh (without password) and all files have to be reachable from all machines under the same path. If this is the case the inversion should run and the progress will be displayed.

The `nice -15` preceding the `mpirun` command gives the processes a lower priority. You should decrease the priority (i.e. increase the 'nice' number) when other users are also using the same machine.

Example 2

A very useful property of MPICH is that you can run programs in parallel on machines with different architectures. It is therefore possible to run an inversion on, let's say, Linux and Solaris machines at the same time.

Passwordless authentication has to be enabled between the server and the slaves (see above). The MPICH-binaries for the different architectures must exist.

Example: To run an inversion on the Linux machines lx?? and on the Solaris machine helios at the MPS you submit the command on an lx-machine:

```
nice -15 /data/slam/software/mpich2/bin/mpiexec \
        -arch mpi -machinefile ./machines.lx -np 4 \
        -arch solmpi -machinefile ./machines.solaris -np 5 \
        ./bin/helix.\%a -i mpitest.ipt
```

This example runs the inversion defined in `mpitest.ipt` on the master (the machine where you typed in the command) plus 3 slave nodes defined in the machinefile `machines.lx` and on 5 nodes of the helios cluster. The binaries `bin/helix.intel.mpi` and `bin/helix.solmpi` must exist and be accessible from both architectures.

E.2 Running HELIX$^+$ on the Clusters gwdu102 and gwds1

The Linux Cluster in Göttingen (gwdu102) consists of 99 double processor Xeon 3.06GHz boards. It is ideally suited for inverting large maps. The MPI implementation in Göttingen is called SCALI.

Since January 2007 a new SGI Altix 4700 cluster exists: It has 512 CPU cores. Both clusters share the same home directory and temporary disk space.

Please read the homepages of the GWDG carefully before using the clusters: http://www.gwdg.de/service/index.html. A description of the queuing system can be found here: http://www.gwdg.de/service/rechenanlagen/parallelrechner/cluster_beschreibung/lsf.html.

F Input File Examples

2-component atmosphere, test for convergence strategies File: `input/lmpik_test.ipt`

```
;--------- COMMENTS ----------------------------------------------------
2-component fit
test for convergence strategy PIK_LM
;--------- DIRECTORIES -------------------------------------------------
PS            ./ps/        ;directory for postscript output
SAV           ./sav/       ;directory for storing results (sav-file)
PROFILE_ARCHIVE /data/slam/VTT-data/data_may01/13may01/ ;input directory for
                           ;  profiles / observations
ATM_ARCHIVE   ./atm_archive/ ;output directory for results (atmospheres)
ATM_SUFFIX    lmpik_test_minfit90 ;add a suffix to the atm-directory to
                           ;  identify this run
WGT           ./wgt/       ;directory for wgt-files
ATOM          ./atom/      ;directory for atomic data files
;--------- CONTROL -----------------------------------------------------
DISPLAY_MAP  0             ;if 1 then display maps of parameters after
                           ;  successful run (multiple pixels only!)
DISPLAY_PROFILE 0             ;if 1 then display fitted & observed profiles
DISPLAY_COMP 0             ;if 1 then display individual atm. components of
                           ;  fitted profiles
```

```
VERBOSE      0            ;verbosity of output: 0=quiet, 1=normal, 2=women
SAVE_FITPROF 0            ;if 1 then save fitted profile as sav-file (can be
                          ;  used as input profile)
FITSOUT      1            ;map results are written to FITS file
OUTPUT       X            ;set to 'PS' for postscript output
;--------- DATA SET -----------------------------------------------------
OBSERVATION  13may01.014cc ;observation in one of three formats: 1.) sav file
                          ;  of the format 'obsname'.profiles.sav, 2.
                          ;  spinor-profile, 3. sav-file created with helix
                          ;  'SAVE_FITPROF' keyword
WL_RANGE     10825.0000 10832.0000 ;WL-range to be used for analysis (may be
                          ;  only a part of the observation WL-range)
WL_NUM       256          ;# of WL-points (for synthesis only)
WL_DISP      0.0000       ;WL-calibration: dispersion per WL-bin number
WL_OFF       0.0000       ;WL-calibration: offset (used if != 0)
WL_BIN       1            ;wavelength binning
XPOS         0 147        ;two-elements vector containing xmin,xmax of the
                          ;  observation map to be analyzed
YPOS         0 96         ;two-elements vector containing ymin,ymax of the
                          ;  observation map to be analyzed
STEPX        1            ;step size for going from xmin to xmax
STEPY        1            ;step size for going from ymin to ymax
AVERAGE      1            ;if 1 then average observation over the stepx/stepy
                          ;  size
SCANSIZE     0            ;stepsize of multiple scans within one observation
SYNTH        0            ;if 1 then create synthetic profile
NOISE        0.0000       ;noise level for adding artificial random noise
SMOOTH       0 0          ;ssmooth-value for profiles and smooth-method:
                          ;  (0=IDL-smooth function,1=FFT Low-Pass)
MEDIAN       3            ;median filter for observed profiles (0=off, >=2
                          ;  median filter width)
SPIKE        0            ;if >=1 then remove electronic spikes. Higher values
                          ;  remove broader spikes.
STRAYPOL_AMP 0.0000       ;amplitude for stray-polarization (only used for
                          ;  synthesis)
STRAYPOL_CORR     0 B      ;iteration steps and orientation of
                          ;  scattering-polarization correction. Orientation:
                          ;  'B' = along B-field, 'X' = a number defining an
                          ;  angle manually
SLIT_ORIENTATION  172.90      ;slit-orientation (used for straypol-corr)
SOLAR_POS        0.00     0.00 ;pos. of observation (x and y in arcseconds)
SOLAR_RADIUS   950.00     ;;radius of sun in arcsec for time of obs.
;--------- POST PROCESSING ----------------------------------------------
CONV_FUNC                 ;convolution function of instrument
CONV_NWL     0            ;# of bins for convolution (used if number of
                          ;  WL_bins in data is small
;--------- ATMOSPHERES --------------------------------------------------
NCOMP        2            ;number of components
; --- atmospheric component 1 ---
; NAME     Value   SCL_MIN   SCL_MAX FIT %RG FIT
BFIEL     892.22     50.00   2000.00  4 ;magnetic field value in Gauss
AZIMU      37.05    -90.00     90.00  3 ;azimut of B-vector [deg]
GAMMA      43.50      0.00    180.00  2 ;inclination of B-vector [deg]
VELOS    1429.54  -7000.00   7000.00  1 ;line-of-sight velocity in m/s
VDAMP       0.01      0.01      0.70  0 ;damping constant (Voigt only)
VDOPP       0.42      0.10      1.00  1 ;doppler broading (Voigt only)
EZERO       1.00      0.00     10.00  0 ;amplitude of components of
                          ;  propagation matrix (Voigt profile only!)
SGRAD       2.60      1.00      8.00 -5 ;gradient of source function
ALPHA       0.59      0.01      0.99  1 ;Filling factor for this component
                          ;  (only with multiple components)
ATM_INPUT                 ;filename for input atmosphere: defines initial
                          ;  value for atmospheric parameters or delivers the
                          ;  input to synthesize a map.
USE_ATOM     he1083.0.new.dat ;atomic data file(s) to be used for this component
                          ;USE_ATOM must be the last keyword for this component.
; --- atmospheric component 2 ---
; NAME     Value   SCL_MIN   SCL_MAX FIT %RG FIT
BFIEL    1984.00     50.00   2000.00  4
AZIMU      30.61    -90.00     90.00  3
GAMMA      47.42      0.00    180.00  2
VELOS   39876.10   5000.00  40000.00  1
VDAMP       0.01      0.01      0.70  0
VDOPP       0.99      0.10      1.00  1
EZERO       1.00      0.00     10.00  0
SGRAD       2.60      1.00      8.00 -5
ALPHA       0.41      0.01      0.99  1
USE_ATOM     he1083.0.new.dat
ATM_INPUT                 ;filename for input atmosphere: defines initial
```

```
                          ;  value for atmospheric parameters or delivers the
                          ;  input to synthesize a map.
;--------- LINE PARAMETERS ---------------------------------------------
; --- fit parameters for line 1 ---
; NAME            Value   SCL_MIN   SCL_MAX  FIT
LINE_ID 10829.0911  ;wavelength to identify the line
LINE_STRENGTH       1.00      0.00      0.00  0 ;factor to adjust line strength
LINE_WLSHIFT       0.00      0.00      0.00  0 ;adjust central WL of line
; --- fit parameters for line 2 ---
; NAME            Value   SCL_MIN   SCL_MAX  FIT
LINE_ID 10830.2501  ;wavelength to identify the line
LINE_STRENGTH       1.00      0.00      0.00  0 ;factor to adjust line strength
LINE_WLSHIFT       0.00      0.00      0.00  0 ;adjust central WL of line
; --- fit parameters for line 3 ---
; NAME            Value   SCL_MIN   SCL_MAX  FIT
LINE_ID 10830.3397  ;wavelength to identify the line
LINE_STRENGTH       1.00      0.00      0.00  0 ;factor to adjust line strength
LINE_WLSHIFT       0.00      0.00      0.00  0 ;adjust central WL of line
;--------- BLENDS ------------------------------------------------------
NBLEND       1 ;number of telluric blends
; --- telluric blend 1 ---
; NAME            Value   SCL_MIN   SCL_MAX FIT
BLEND_WL           0.00      0.00      0.00  0 ;WL and WL-range for blend
BLEND_WIDTH        0.00      0.00      0.00  0 ;width of voigt-profile
BLEND_DAMP         0.00      0.00      0.00  0 ;damping of voigt-profile
BLEND_A0           0.00      0.00      0.00  0 ;amplitude of voigt-profile
                          ;  (blend is not used if BLEND_A0=0
;--------- GENERAL FIT PARAMETERS --------------------------------------
; NAME            Value   SCL_MIN   SCL_MAX FIT
CCORR              1.00      0.90      1.10  0 ;factor for continuum correction
STRAYLIGHT         0.00      0.00      1.00  0 ;straylight correction
                          ;  (non-dispersive, non-polarized)
;--------- ANALYSIS METHOD ---------------------------------------------
IQUV_WEIGHT  1.0000 1.0000 1.0000 1.0000 ;4-element vector defining relative
                          ;  weighting of IQUV (in that order). Additionally
                          ;  the code does an automatic weighting according to
                          ;  the strength of the I signal compared to the QUV
                          ;  signals. This weighting scheme is used in the
                          ;  PIKAIA fit routine.
WGT_FILE     he_default.wgt ;file with WL-dependent weighting function for IQUV
PROFILE      voigt        ;functional form for pi- and sigma components of
                          ;  spectral line. Available: gauss, voigt or
                          ;  voigt_phys
MAGOPT       1            ;include magneto-optical effects (dispersion
                          ;  coefficients, (Voigt profile only!))
USE_GEFF     0            ;use effective Lande factor (=1) or real Zeeman
                          ;  pattern (=0)
USE_PB       1            ;if set, the zeeman-splitting and strength includes
                          ;  the Paschen-Back effect (from the table by
                          ;  Socas-Navarro)
PB_METHOD    poly         ;use polynomials (=poly) or table interpolations
                          ;  (=table) to calculate the PB-effect
;--------- PIKAIA PARAMATERS -------------------------------------------
CODE         FORTRAN      ;PIKAIA code to use. Available: FORTRAN (=fast) or
                          ;  IDL (=platform independent).
METHOD       PIK_LM 1 20 0.90 ;minimization method: PIKAIA, POWELL (fast),
                          ;  LMDIF (fast) or PIK_LM (combined, for maps)
NCALLS       500          ;number of iterations in PIKAIA routine / max.
                          ;  number of calls for POWELL or LMDIF
;--------- END OF INPUT FILE -------------------------------------------
```

G Copyright

HELIX$^+$ can be used freely. Please add a reference to *Lagg et al.* [2004] if you publish results obtained with HELIX$^+$.

If you use the Hanle-slab model, add a reference to *Asensio Ramos et al.* [2008].

Please note that this code uses some libraries for which copyright restrictions might apply:

DISLIN: The DISLIN distributions for Linux, FreeBSD and for the PC compilers GCC, G77 and LCC can be used freely for non-commercial applications.

Intel Fortran Compiler: Please read the Ënd User License Agreementfor Fortran Compilers for Linux on the intel-web-site http://www.intel.com/cd/software/products/asmo-na/eng/compilers/219715.htm.

MPICH: Copyright Notice:
+ 1993 University of Chicago
+ 1993 Mississippi State University
Permission is granted to use, reproduce, prepare derivative works, and to redistribute to others.

ODRPACK95 Version 2.01 [*Zwolak et al.*, 2007]: Software for Weighted Orthogonal Distance Regression
Paul T. Boggs, Richard H. Byrd, Janet E. Rogers and Robert B. Schnabel Applied and Computational Mathematics Division, Center for Computing and Applied Mathematics, U.S. DEPARTMENT OF COMMERCE National Institute of Standards and Technology Gaithersburg, MD 20899

CFITSIO: Copyright U.S. Government.
Please note that some files contained in this library are distributed under the GNU General Public License. Read `License.txt` in the directory `cfitsio`.

H Known Bugs

H.1 Normalization Problem (Jan-19 2007)

Until January 19th 2007 the code used a slightly wrong method to sum up the individual profiles resulting from different atomic lines / transitions. The calculated profiles were scaled with the the reciprocal of the number of atomic lines / transitions (e.g. in the case of an inversion using the He 1083 nm line the absorption signature was weakened by a factor of 1/3).

This scaling was removed. Now every single line produces the same absorption signal, no matter on how many lines / transitions are calculated. The results from the old and the new runs are 1:1 exchangeable. All parameters calculated with the old method are still valid, only the parameter `SGRAD` will be affected: the value of `SGRAD` is decreased by a factor of one over the number of lines.

The effect of this error was that the `SGRAD` value changes when the number of atomic lines / transitions is changed. In the updated version `SGRAD` is independent on the number of lines (as it should be).

In order to maintain compatibility with old results the keyword `OLD_NORM` was introduced (for a description of the keyword see page 8). If this keyword is set to one, then the code uses the old normalization method, resulting in (number of lines)$\times$ higher values of `SGRAD`. Not setting this keyword means that the code uses the new (and hopefully correct) normalization.

H.2 $\log gf$ values for He 10830

Until December 14, 2009 the $\log gf$ values in the atomic data file for the He triplet at 10830 Å transition were erroneous. This error causes the strength of the He absorption to be scaled by the same factor for all three lines. It therefore only affects the value for `EZERO`. New inversions should use the atomic data file

```
he1083.0.new.dat
```

The file `he1083.0.dat` still contains the old values for compatibility reasons.

and many others bugs... If you find a bug or an undesired result / behavior please describe it briefly and send it with all files necessary to reproduce the error (data, weighting file, input file, atomic data file) to the author.

References

Allen's Astrophysical Quantities, chap. 14.7, p. 355, 4th ed., New York: Springer, 2000.

Asensio Ramos, A., J. Trujillo Bueno, and E. Landi Degl'Innocenti, Advanced Forward Modeling and Inversion of Stokes Profiles Resulting from the Joint Action of the Hanle and Zeeman Effects, *ApJ*, *683*, 542–565, 2008.

Auer, L. H., L. L. House, and J. N. Heasley, The determination of vector magnetic fields from Stokes profiles, *Sol. Phys.*, *55*, 47–61, 1977.

Balasubramaniam, K. S., and E. A. West, Vector magnetic fields in sunspots. I - Stokes profile analysis using the Marshall Space Flight Center magnetograph, *ApJ*, *382*, 699–705, 1991.

Bommier, V., S. Sahal-Brechot, and E. Landi Degl'Innocenti, Resonance line polarization and the Hanle effect in optically thick media. II - Case of a plane-parallel atmosphere, *A&A*, *244*, 383–390, 1991.

Charbonneau, P., Genetic algorithms in astronomy and astrophysics, *A&AS*, *101*, 309–334, 1995, (http://www.hao.ucar.edu/public/research/si/pikaia/pikaia.html).

Faurobert-Scholl, M., Hanle effect with partial frequency redistribution. II - Linear polarization of the solar CA I 4227 A line, *A&A*, *258*, 521–534, 1992.

Lagg, A., J. Woch, N. Krupp, and S. K. Solanki, Retrieval of the full magnetic vector with the He I multiplet at 1083 nm. Maps of an emerging flux region, *A&A*, *414*, 1109–1120, 2004.

Lagg, A., R. Ishikawa, L. Merenda, T. Wiegelmann, S. Tsuneta, and S. K. Solanki, Internetwork Horizontal Magnetic Fields in the Quiet Sun Chromosphere: Results from a Joint Hinode/VTT Study, in *The Second Hinode Science Meeting: Beyond Discovery-Toward Understanding*, edited by B. Lites, M. Cheung, T. Magara, J. Mariska, and K. Reeves, vol. 415 of *ASP Conf. Ser.*, p. 327, 2009.

Landi Degl'Innocenti, E., Magnetic field measurements, in *Solar Observations: Techniques and Interpretation*, edited by F. Sánchez, M. Collados, and M. Vázquez, p. 71, First Canary Islands Winter School, Cambridge Univ. Press, Cambridge UK, 1992.

Nagendra, K. N., H. Frisch, and M. Faurobert-Scholl, An operator perturbation method for polarized line transfer. III. Applications to the Hanle effect in 1D media, *A&A*, *332*, 610–628, 1998.

Orozco Suárez, D., et al., Strategy for the Inversion of Hinode Spectropolarimetric Measurements in the Quiet Sun, *Publications of the Astronomical Society of Japan*, *59*, 837–+, 2007.

Rachkowsky, D. N., The reduction for anomalous dispersion in the theory of the absorption line formation in a magnetic field (in Russian), *Izv. Krym. Astrofiz. Obs.*, *37*, 56–61, 1967.

Socas-Navarro, H., personal communication, 2005.

Socas-Navarro, H., J. Trujillo Bueno, and E. Landi Degl'Innocenti, Signatures of incomplete paschen-back splitting in the polarization profiles of the he i λ10830 multiplet, *ApJ*, *612*, 1175–1180, 2004.

Socas-Navarro, H., J. Trujillo Bueno, and E. Landi Degl'Innocenti, Polynomial Approximants for the Calculation of Polarization Profiles in the He I 10830 Å Multiplet, *ApJS*, *160*, 312–317, 2005.

Solanki, S. K., A. Lagg, J. Woch, N. Krupp, and M. Collados, Three-dimensional magnetic field topology in a region of solar coronal heating, *Nature*, *425*, 692–695, 2003.

Trujillo Bueno, J., E. Landi Degl'Innocenti, M. Collados, L. Merenda, and R. Manso Sainz, Selective absorption processes as the origin of puzzling spectral line polarization from the Sun, *Nature*, *415*, 403–406, 2002.

Trujillo Bueno, J., L. Merenda, R. Centeno, M. Collados, and E. Landi Degl'Innocenti, The Hanle and Zeeman Effects in Solar Spicules: A Novel Diagnostic Window on Chromospheric Magnetism, *ApJL*, *619*, L191–L194, 2005.

Unno, W., Line Formation of a Normal Zeeman Triplet, *Publications of the Astronomical Society of Japan*, *8*, 108–125, 1956.

Zwolak, J. W., P. T. Boggs, and L. T. Watson, Algorithm 869: ODRPACK95: A weighted orthogonal distance regression code with bound constraints, *ACM Transactions on Mathematical Software*, *33*, 27, 2007, article 27, 12 pages.

author: Andreas Lagg, e-mail: lagg@mps.mpg.de
Max-Planck-Institut für Sonnensystemforschung
Justus-von-Liebig-Weg 3
37077 Göttingen, Germany
Tel: +49-551-384979-465