P-MILOS Documentation

Manuel Cabrera Morales and Luis Bellot Rubio

Instituto de Astrofísica de Andalucía-CSIC
lbellot@iaa.es

December 17, 2020

Contents

1 Overview

This document is a user's guide to P-MILOS, a highly-efficient, massively-parallel Milne-Eddington Stokes inversion code capable of inverting large data sets in real time. The code is written in C and consists of a parallel and a sequential version.

P-MILOS was developed within the SOLARNET H2020 project for the inversion of data from the SPRING network, but it is a general-purpose code that will suit the needs of the solar community at large. It is based on an earlier C implementation by Cobos Carrascosa (2015) of the IDL MILOS code (Orozco Suárez & del Toro Iniesta 2007).

P-MILOS has been thoroughly designed and optimized to provide:

- Input/output based on the FITS[1] file format
- SIR-like control files, for compatibility with the SIR, SIR-parallel, and DESIRE suite
- Very efficient memory management
- Highly optimized numerical calculations
- Efficient implementation of convolutions with direct methods and FFT algorithms
- Inversion of individual maps using all available processor cores
- Support for time series inversion

The sequential code can be used to synthesize/invert Stokes profiles from individual pixels or to invert a full Stokes data cube in a single processor core. The parallel code is used to invert a full Stokes data cube or time sequences of data cubes in multiple processor cores using MPI. This documentation describes the parallel code mainly.

2 Installation

2.1 Downloading the code

The latest stable version of P-MILOS can be downloaded from the IAA-CSIC Solar Physics Group GitHub repository at https://github.com/IAA-InvCodes/P-MILOS. To copy the source code to your computer, open a terminal and do

```
git clone https://github.com/IAA-InvCodes/P-MILOS
```

To update an existing installation, cd to the parent directory P-MILOS and do

```
git pull
```

[1] Flexible Image Transport System, see http://fits.gsfc.nasa.gov/

2.2 Installation requirements

P-MILOS must be compiled on the target machine. The compilation makefile and executables have been tested only on machines running Ubuntu 18.04 and Red Hat Enterprise Linux Server release 6, but the code is portable and should work on any 64-bit Linux system.

The following programs and libraries should be installed on your system to compile the code:

- Intel C compiler (icc) version 11.1 or GNU gcc compiler version 4.1.2 (or higher)
- OpenMPI version 1.4-4 or higher, compiled with icc or gcc
- CFITSIO version 3.3.4.0 or higher
- FFTW version 3.3.3 or higher
- GSL version 1.13-3 or higher

W**e strongly recommend** you to use the latest version of the Intel C compiler to ensure the best performance, as there are significant differences in speed with respect to older versions. This is particularly important when the objective is to invert data in real time. The code has been tested up to icc version 19.1.1.

P-MILOS requires at least OpenMPI version 1.4-4, but it has been tested up to version 3.1.3. In the same way, the code runs with version 1.13-3 of GSL, but we recommend version 2.6 because of its much better performance. The code was developed using the FFTW and CFITSIO libraries specified above, which are the latest ones at the time of writing.

2.3 Make options

To compile the code, use the command 'make' in the directory where the distribution is located. On Intel machines, the makefile included in the distribution activates the highest instruction set offered by the processor (normally AVX-2 or AVX-512) via the -xHost option. On AMD machines, please edit the makefile and change -xHost to -march=core-avx2 to activate this option.

Using the AVX-2 or AVX-512 instruction set increases the code speed at the expense of less consistent floating-point results. For some pixels, this may lead to small differences in the retrieved atmospheric parameters. If you want to have exact floating-point precision, remove the options -Xhost -mp1 -no-fma from the makefile and add the -fma option, but be warned that this will reduce the speed by about 15% without producing significant differences in the results (e.g., less than 0.1 G in field strength for << 1% of the pixels). If, on the other hand, you want a a 5% speed increase through more aggressive optimizations, use the options -Xhost -fma instead of -Xhost -mp1 -no-fma.

- Compile and create executables **pmilos.x** and **milos.x**

```
make
```

- Compile parallel version and create executable **pmilos.x**

```
make pmilos.x
```

- Compile sequential version and create executable **milos.x**

```
make milos.x
```

- Clean objects and executables:

```
make clean
```

The executables **pmilos.x** and **milos.x** will be located in the directory where the make command is run. In the github distribution, this is the main P-MILOS directory.

3 Control files

P-MILOS uses two files with extensions ".mtrol" and ".minit" to control the inversion conditions and the execution of the parallel code, respectively.

The format of the .mtrol file is very similar to that of the .trol file used by the SIR inversion code (Ruiz Cobo & del Toro Iniesta 1992), while .minit files are the same as the .init files of SIR-parallel (Thornhofer & Bellot Rubio 2015). Thus, SIR and SIR-parallel users will be familiar with these files. The extensions are changed to reflect the fact that they apply to a Milne-Eddington inversion code, however.

3.1 Mtrol file

Both the sequential and parallel versions of P-MILOS require a mtrol file.

1. **In the sequential version**, the mtrol file is used to
    a. Synthesize Stokes profiles from a given model atmosphere.
    b. Invert a set of Stokes profiles provided through a ".per" file.
    c. Invert a full map of Stokes profiles provided through a FITS cube.

2. **In the parallel version**, the mtrol file is used to invert a single FITS Stokes cube or a full time series of FITS cubes.

Mtrol files have 42 lines that must be ordered as in the example of Table 1. Each line is divided in three parts. The first one extends until the colon mark (:). The second extends from the colon mark to the exclamation point (!). The third one goes from the exclamation point to the end of the line. These three parts contain the name of the field, the parameters to be used by P-MILOS, and comments or additional explanations to the user, respectively. P-MILOS ignores all what is written in the line after the exclamation point. **The characters : and ! are mandatory.** Entries marked with (*) are also mandatory.

```
Number of cycles          (*):50                ! 0 = synthesis, n= max iter, -1 = classical
Observed profiles         (*):data/obs.fits           !
Stray light file             :                  ! (none=no stray light)
PSF file                     :49.2              ! file or Gaussian FWHM in mA
Wavelength grid file      (*):wavelength.grid   !
Atomic parameters file    (*):LINES             !
Abundance file               :                  ! not used
Initial guess model 1     (*):init.mod          !
Initial guess model 2        :                  ! not implemented
Weight for Stokes I          :1                 ! (DEFAULT=1; 0=not inverted)
Weight for Stokes Q          :1                 ! (DEFAULT=1; 0=not inverted)
Weight for Stokes U          :1                 ! (DEFAULT=1; 0=not inverted)
Weight for Stokes V          :1                 ! (DEFAULT=1; 0=not inverted)
AUTOMATIC SELECT. OF NODES? :                   ! not implemented
Nodes for S_0 1              :1                 ! (0 or blank=no, 1=yes)
Nodes for S_1 1              :1                 ! (0 or blank=no, 1=yes)
Nodes for eta0 1             :1                 ! (0 or blank=no, 1=yes)
Nodes for magnetic field 1 :1                   ! (0 or blank=no, 1=yes)
Nodes for LOS velocity 1     :1                 ! (0 or blank=no, 1=yes)
Nodes for gamma 1            :1                 ! (0 o r blank=no, 1=yes)
Nodes for phi 1              :1                 ! (0 or blank=no, 1=yes)
Nodes for lambda_dopp 1 :1                      ! (0 or blank=no, 1=yes)
Nodes for damping 1          :1                 ! (0 or blank=no, 1=yes)
Invert macroturbulence 1  :0                    ! (0 or blank=no, 1=yes)
Nodes for S_0 2              :                  ! not used
Nodes for S_1 2              :                  ! not used
Nodes for eta0 2             :                  ! not used
Nodes for magnetic field 2 :                    ! not used
Nodes for LOS velocity 2     :                  ! not used
Nodes for gamma 2            :                  ! not used
Nodes for phi 2              :                  ! not used
Nodes for lambda_dopp 2 :                       ! not used
Nodes for damping 2          :                  ! not used
Invert macroturbulence 2?:                      ! not used
Invert filling factor?       :                  ! not used
Invert stray light factor?   :                  ! (0 or blank=no, 1=yes)
mu=cos (theta)               :                  ! (DEFAULT mu=1, mu<0 => West)
Estimated S/N for I          :1000              !
Continuum contrast           :                  ! not used
Initial diagonal element     :0.1               ! (DEFAULT value: 1.e-1)
Use FFT for convolutions     :0                 ! (0 or blank= direct convol; 1 = FFT)
Diagonal element acceler     :1                 ! (0 or blank=no, 1=yes)
```

Table 1. Example of a control file

The meaning of the various entries in the control file is the following:

- **Number of cycles**
  - If set to -1, the code uses **classical methods** to determine the LOS velocity, the field strengh, the inclination and the azimuth from the observed profile(s). The result are stored in a file with the same name as the observed profile plus the suffix "_ce".
  - If set to 0, the code performs a spectral **synthesis** using the model atmosphere given in the field "Initial guess model 1". This option works only with the sequential code.
  - When the number of cycles is larger than 0, the code performs the **inversion** of a single data cube or several data cubes from a time series. The value indicates the maximum number of iterations allowed to find a solution.
- **Observed profiles**. Name of the file containing the Stokes profiles. The parallel code only accepts FITS files. To invert a single cube, the full filename should be provided, including the .fits extension. To invert a time sequence, only the base name should be given (without the time step number or the .fits extension). More details will be provided later.
- **Stray light file**. Name of the file containing the stray light profile. It can be of type FITS or .per. If the filling factor indicated in the model atmosphere is different from 1 and no stray light is supplied, the code will abort with an error message. FITS stray-light files can have 2 or 4 dimensions. In the first case, the stray light is stored in an 2-dimension array with (number of wavelengths, 2) elements. The first column contains the wavelengths, the second the stray light intensities (no polarization). In the second case, the four dimensions are (number of rows, number of columns, number of wavelengths, Stokes), just like regular FITS profile files. A four-dimension array can be used to specify different stray light profiles in different pixels.
- **PSF file**. Name of the file contaning the spectral transmission profile (PSF) of the instrument. If a numerical value is given instead of a file, it is taken to be the FWHM of the transmission profile, in mÅ, assumed to be Gaussian.
- **Wavelength grid file**. Name of the file specifying the wavelength points. It can be of type .grid or FITS (see later). This file is mandatory.
- **Atomic parameters file**. Name of the file containing the atomic parameters of the observed spectral line. This file is mandatory.
- **Abundances file**. Not used by P-MILOS. Kept here to maintain compatibility with SIR and SIR-parallel control files.
- **Initial guess model 1**. Name of the file containing the model atmosphere to be used for spectral synthesis or the initial guess model to start an inversion.
- **Initial guess model 2**. Not implemented. For the moment, P-MILOS uses one-component model atmospheres only.
- **Weight for Stokes I**. Weight of Stokes I in the chi2 merit function.
- **Weight for Stokes Q.** Weight of Stokes Q in the chi2 merit function.
- **Weight for Stokes U**. Weight of Stokes U in the chi2 merit function.
- **Weight for Stokes V**. Weight of Stokes V in the chi2 merit function.

- **AUTOMATIC SELECT. OF NODES**. Not implemented in P-MILOS.
- **Nodes for S_0 1**. Use 1 to invert S0, 0 or blank to keep it fixed.
- **Nodes for S_1 1**. Use 1 to invert S1, 0 or blank to keep it fixed.
- **Nodes for eta0 1**. Use 1 to invert eta0, 0 or blank for keeping it fixed
- **Nodes for magnetic field 1**. Use 1 to invert the field strength B, 0 or blank to keep it fixed
- **Nodes for LOS velocity 1**: Use 1 to invert the LOS velocity, 0 or blank to keep it fixed
- **Nodes for gamma 1**. Use 1 to invert the field inclination, 0 or blank to keep it fixed
- **Nodes for azimuth 1**. Use 1 to invert the field azimuth, 0 or blank to keep it fixed
- **Nodes for lambda_doppler 1**. Use 1 to invert Doppler width, 0 or blank to keep it fixed
- **Nodes for damping 1**. Use 1 to invert the damping parameter, 0 or blank to keep it fixed
- **Nodes for macroturbulence 1**. Use 1 to invert macroturbulence, 0 or blank to keep it fixed
- **Nodes for S_0 2**. Not used at this moment.
- **Nodes for S_1 2**. Not used at this moment.
- **Nodes for eta0 2**. Not used at this moment.
- **Nodes for magnetic field 2**. Not used at this moment.
- **Nodes for LOS velocity 2**. Not used at this moment.
- **Nodes for gamma 2**. Not used at this moment.
- **Nodes for phi 2**. Not used at this moment.
- **Nodes for lambda_doppler 2**. Not used at this moment.
- **Nodes for damping 2**. Not used at this moment.
- **Nodes for macroturbulence 2**. Not used at this moment.
- **Invert filling factor**. Not used at this moment.
- **Invert stray light factor.** Use 1 to invert the stray light factor, 0 or blank to keep it fixed
- **mu=cos (theta)**. Cosine of the heliocentric angle. Default values is 1.
- **Estimated S/N for I**. Estimated signal-to-noise ratio. The default value is 1000.
- **Continuum contrast**. Not used at this moment.
- **Initial diagonal element**. Initial Levenberg-Marquardt fudge parameter. Default is 1e-1.
- **Use FFT for convolutions**. By default, all convolutions are performed using direct multiplication of the two functions to be convolved. To employ Fourier transforms, set this field to 1.
- **Diagonal element acceleration**. If this option is activated with 1, the convergence of the Levenberg-Marquardt algorithm is accelerated by multiplying the diagonal element with a factor that changes depending on the degree of improvement of the χ^2 merit function.

3.2 Minit file

P-MILOS needs a compulsory init-file to specify the parameters for the inversion of a time series. An example is given in Table 2. The layout is similar to that of the control file (3 columns, with the data column between the symbols ':' and '!').

The minit file contains the following entries:

- **Name of control file**: the path and name of the control file.
- **Type of input stokes**. Currently, only FITS files are allowed.
- **Type of input straylight**. Currently, only FITS files are allowed. If no stray light is needed, this field can be left blank.
- **nx, ny**. Not necessary, as the spatial dimensions are read from the FITS file headers. These parameters are kept for compatibility with SIR-parallel init files.
- **subx1, subx2, suby1, suby2**: Spatial coordinates of the subfield to be inverted
- **outfile**: common prefix for all the output files containing the best-fit profiles and model atmospheres. This can be used to set the path, too.
- **mask file**. A FITS file containing the pixel mask. Pixels having 0 in the mask are not inverted. If no mask file is provided, all pixels are inverted.
- **t1:** first index of the time series. The path and name of the files to be inverted are given in the control file and must be of the form "*basenameNNN.fits*", where *basename* is the base name and *NNN* is a three-digit number specifying the timestep, left-padded with zeros. The first time step to be inverted will be *basenamet1.fits*.
- **t2:** last index of the time series. The last time step to be inverted will be *basenamet2.fits*.
- **save best-fit profiles.** Select 0 to stop writing the best-fit profiles to disk. The default is to store them.

Name of control file (*)	:pmilos.mtrol	!
Type of input stokes (*)	:fits	! (fits)
Type of input straylight	:fits	! (fits)
nx	:	! (number of pixels in x-direction)
ny	:	! (number of pixels in y-direction)
subx1	:	! (0 or blank = invert all pixels)
subx2	:	! (0 or blank = invert all pixels)
suby1	:	! (0 or blank = invert all pixels)
suby2	:	! (0 or blank = invert all pixels)
outfile	:output/inv_	! (path+prefix for output files)
mask file	:	! (mask for excluding pixels)
t1	:20	! (blank= do not invert timeseries)
t2	:21	! (blank= do not invert timeseries)
Save best-fit profiles	:0	! (0 = no; 1 or blank = yes)

Table 2: Example of a minit-file

If t1 and t2 are empty, only the FITS file specified in the field “Observed profiles” of the control file will be inverted.

If an asterisk (*) is used in t2, the code will invert all the data cubes present in the directory and will wait for more cubes to arrive at the same directory. If no new files are detected after 5 minutes, the execution will be terminated. This is useful when inverting data streams in real time. In this situation, new Stokes cubes are expected to be received at regular intervals.

4 Input files

4.1 Observed profiles (.fits, .per)

P-MILOS must be fed with cubes containing the Stokes profiles observed in the entire field of view. This is the usual way of inverting measurements from narrow-band filter imagers such as SST/CRISP.

The Stokes profiles recorded at time t must be stored as a FITS file containing an array of four dimensions. The default arrangement of the data cube is (x,y,wavelength,stokes). To identify the actual order of the parameters, P-MILOS uses the FITS header and looks for the keywords CTYPE1, CTYPE2, CTYPE3 and CTYPE4. The values of these keywords define the parameter stored in each dimension, following the SOLARNET WP20.3 standard:

- '**HPLN-TAN’** indicates spatial coordinate x
- '**HPLT-TAN**' indicates spatial coordinate y
- '**WAVE-GRI**' indicates wavelength axis
- '**STOKES** ' indicates Stokes parameters

The default order of the FITS file is

```
CTYPE1  = 'HPLN-TAN'
CTYPE2  = 'HPLT-TAN'
CTYPE3  = 'WAVE-GRI'
CTYPE4  = 'STOKES  '
```

An alternative order would be

```
CTYPE1  = 'STOKES  '
CTYPE2  = 'WAVE-GRI'
CTYPE3  = 'HPLN-TAN'
CTYPE4  = 'HPLT-TAN'
```

In this example, the data are stored in an 4-dimension array (stokes, wavelength, x, y). Using the default order is not mandatory, but highly recommended because it is 50% more efficient (the array will not be rearranged internally, speeding up the reading process).

The number of elements in each dimension is given by the header keywords NAXIS1, NAXIS2, NAXIS3 and NAXIS4.

P-MILOS expects the data to be of FLOAT type and will use 32 bits to represent them. The code will exit if any other data type is detected. To exclude specific Stokes I, Q, U or V points from the fit, the user should set them to a negative value below -1. Those points will not be considered in the calculation of the χ^2 function. This can be used to add "ghost" wavelengths to deal with unevenly spaced wavelength grids or to extend the wavelength range on either side of the line for more accurate convolutions. It can also be used to supress telluric blends from the intensity profile (the Stokes I values would be set to, e.g., -2 in the corresponding wavelength positions, keeping the measured Stokes Q, U and V values, which are usually not affected by telluric blends). This is the same strategy used by SIR and SIR-parallel.

Different time steps should be stored in separate FITS files. To invert a time series, the file names must be of the form *basenameNNN.fits*, where *basename* is provided in the control file under the field "Observed profiles" and *NNN* is a 3-digit number, left-padded with zeros, specifiying the timestep (e.g., 001 for the first cube, 002 for the second, etc). No gaps are allowed in the sequence.

Observed profiles (.per)

The Stokes profiles of individual pixels can also be stored in ASCII files with extension .per. These files have the same format as SIR profile files. The sequential code uses .per files as input when inverting one pixel and as output when synthesizing the profiles from a given model atmosphere. Also, they are used to store the stray light profile (both sequential/parallel code).

These files have one row per wavelength sample and 6 columns containing:

- The index of the spectral line in the atomic parameter file
- The wavelength offset with respect to the central wavelength (in mÅ)
- The value of Stokes I
- The value of Stokes Q
- The value of Stokes U
- The value of Stokes V

This is an example of a file containing the Stokes parameters of spectral line number 1 in 30 wavelength positions, from -350 to + 665 mÅ:

```
1    -350.0    9.836711e-01    6.600326e-04    4.649822e-04    -3.694108e-03
1    -315.0    9.762496e-01    1.186279e-03    8.329745e-04    -6.497371e-03
1    -280.0    9.651449e-01    2.305113e-03    1.581940e-03    -1.135160e-02
1    -245.0    9.443904e-01    5.032997e-03    3.333831e-03    -2.191048e-02
1    -210.0    9.018359e-01    1.146227e-02    7.265856e-03    -4.544966e-02
1    -175.0    8.222064e-01    2.265146e-02    1.368713e-02    -8.623441e-02
1    -140.0    7.066048e-01    3.263217e-02    1.847884e-02    -1.242511e-01
1    -105.0    5.799722e-01    3.157282e-02    1.560218e-02    -1.238010e-01
1    -70.0     4.711627e-01    2.068015e-02    6.887295e-03    -8.728459e-02
1    -35.0     4.014441e-01    9.837587e-03   -1.054865e-03    -4.476189e-02
```

```
1     -0.00     3.727264e-01    4.631597e-03   -4.830483e-03   -9.482273e-03
1     35.00     3.799767e-01    5.985593e-03   -3.846331e-03    2.321622e-02
1     70.00     4.249082e-01    1.378049e-02    1.806246e-03    6.157872e-02
1     105.0     5.119950e-01    2.571598e-02    1.073034e-02    1.046772e-01
1     140.0     6.316050e-01    3.361904e-02    1.782490e-02    1.297000e-01
1     175.0     7.571660e-01    2.941514e-02    1.718121e-02    1.115998e-01
1     210.0     8.600360e-01    1.762856e-02    1.087526e-02    6.786425e-02
1     245.0     9.230015e-01    8.213106e-03    5.305210e-03    3.366419e-02
1     280.0     9.545796e-01    3.605521e-03    2.427676e-03    1.654710e-02
1     315.0     9.701367e-01    1.734934e-03    1.205792e-03    9.068003e-03
1     350.0     9.786569e-01    9.418463e-04    6.697733e-04    5.552034e-03
1     385.0     9.838719e-01    5.600237e-04    4.053978e-04    3.671347e-03
1     420.0     9.873039e-01    3.608273e-04    2.646724e-04    2.540035e-03
1     455.0     9.896545e-01    2.543154e-04    1.872806e-04    1.792397e-03
1     490.0     9.912774e-01    1.968866e-04    1.437179e-04    1.263287e-03
1     525.0     9.923597e-01    1.690016e-04    1.205914e-04    8.552026e-04
1     560.0     9.929766e-01    1.638180e-04    1.128983e-04    4.949613e-04
1     595.0     9.930763e-01    1.826333e-04    1.215294e-04    1.047099e-04
1     630.0     9.923094e-01    2.385952e-04    1.569032e-04   -4.666936e-04
1     665.0     9.895550e-01    3.734447e-04    2.531845e-04   -1.597078e-03
```

4.2 PSF file

The synthetic Stokes profiles and response functions must be convolved with the transmission profile of the instrument. The transmission profile (or point spread function, PSF) is provided in a file with extension .psf. Alternatively, the user can specify the FWHM of the transmission profile in the control file, in mÅ, and P-MILOS will generate a Gaussian profile with that FWHM. This Gaussian profile will be saved to disk under the name *raizperfiles.psf*.

PSF files have two columns. The first one contains $\Delta\lambda$ (in mÅ) and the second the value of the transmission profile at that $\Delta\lambda$. The PSF does not need to be sampled at the same wavelengths as the observed profiles. When the sampling is different, the code linearly interpolates the PSF to the observed wavelength grid before carrying out any convolution. If the PSF does not cover the entire wavelength range, the missing values are set to zero.

4.3 Atomic parameter file

This file contains the atomic parameters of the observed spectral lines. It is the same file used by SIR and SIR-parallel. There is a default atomic parameter file called LINES in the *run* directory.

Valid atomic parameter files have eigth columns as follows:

```
1=FE 1   6173.3356    1.0   2.223   -2.879   5P 1.0- 5D 0.0  0.266     7.8635e-15
2=FE 1   6301.5012    1.0   3.64    -0.59    5P 2.0- 5D 2.0  0.242846  2.3437e-14
3=FE 1   6302.4936    1.0   3.69    -0.97    5P 1.0- 5D 0.0  0.240599  2.9727e-14
6=FE 1   5250.2080    1.0   0.121   -4.938   5D 0.0- 7D 1.0  0.252     5.79467e-15
7=FE 1   5250.6450    1.0   2.198   -2.047   5P 2.0- 5P 3.0  0.268     9.61869e-15
```

The first column gives the index with which the line is identified in the profile and the wavelength grid files. The index is separated by a "=" sign from the atomic symbol of the element. To specify the atomic element, capital or lower case letters may be used (but do not mix them!). The ionization stage is specified by a number: 1 means neutral atom, and 2 singly ionized atom.

The second column specifies the (laboratory) central wavelength of the transition (in Å). The third column gives the enhancement factor to the van der Waals coefficient Γ_6. The fourth and fifth columns give the excitation potential of the lower level (in eV) and the logarithm of the multiplicity of the level times the oscillator strength (log gf), respectively. The sixth column specifies the atomic transition. Finally, the last two columns provide the collisional broadening parameters α and σ resulting from the quantum mechanical theory of Anstee, Barklem, and O'Mara. σ is expressed in cm^2. These parameters can be set to zero.

P-MILOS only uses this file to extract the central wavelength and the atomic transition of the line, which are needed to compute the number, position and strength of the various Zeeman components. The other atomic parameters are kept in the file to maintain compatibility with SIR and SIR-parallel.

4.4 Wavelength grid file (.grid, .fits)

The observed wavelengths must be specified to invert a set of Stokes profiles (parallel/sequential code) or to compute synthetic spectra from a given model atmosphere (sequential code only). The wavelengths can be given as .grid files or FITS files.

Wavelength grid files are ASCII files with four columns recognized by the extension .grid. They have the same format as the their SIR cousins and can be used interchangeably. The first column indicates the observed spectral line by means of an index which must be present in the atomic parameter file. The wavelength range is specified with three numbers: the initial wavelength, the wavelength step, and the final wavelength (all in mÅ). The initial and final wavelengths are relative to the central wavelength. Here is an example:

```
Line indices          :   Initial lambda     Step     Final lambda
(in this order)                (mA)          (mA)          (mA)
-------------------------------------------------------------------
1                     :       -350,           35,           665
```

In this case, the Stokes profiles of the spectral line that has index 1 in the atomic parameter file was observed from -350 to +665 mÅ in steps of 35 mÅ.

The wavelengths can also be specified by means of a FITS file containing an array of 2 dimensions with (n_wavelengths, 2) elements. The first column stores the line index in the atomic parameter file and the second the observed *absolute* wavelengths (in Å).

Examples of .grid and FITS wavelength files can be found in the *run* directory (*wavelength.grid*, *wavelengths.fits*).

4.5 Model atmospheres (.mod)

The parameters of the model atmosphere are stored in ASCII files with extension .mod. These files are used in three situations:

1. To specify the initial atmosphere for an inversion with the parallel or sequential code
2. To specify the model atmosphere for a spectral synthesis using the sequential code
3. To store the best-fit model atmosphere resulting from the inversion of a .per file using the sequential code

Milne-Eddington atmospheres are fully described with 11 parameters: the line-to-continuum absorption coefficient ratio (eta0), the magnetic field strength (B), inclination (gamma) and azimuth (phi), the LOS velocity, the Doppler width, the damping parameter, the source function constant (S_0), the source function gradient (S_1), the macroturbulent velocity (v_mac) and the filling factor (the area of the resolution element occupied by the magnetic atmosphere). Internally, the filling factor f is converted into a stray light factor $\alpha = 1 - f$ which is used for all subsequent calculations together with the user-provided stray light file. The units of the parameters are: Gauss (magnetic field strength), km/s (LOS velocity and macroturbulent velocity), Å (Doppler width), and degrees (inclination and azimuth). The rest of parameters do not have units.

The following is an example of a model atmosphere that can be used for the Fe I 6173 Å line in the quiet Sun (file *FeI6173_quietsun.mod* in the *run* directory). The parameters must be given in the same order and must be preceded by a colon:

```
eta_0                :13.0
magnetic field [G]   :500.
LOS velocity [km/s]  :0.2
Doppler width [A]    :0.035
damping              :0.19
gamma [deg]          :30.
phi   [deg]          :30.
S_0                  :0.26
S_1                  :0.74
v_mac [km/s]         :0.
filling factor       :1.
```

P-MILOS and SIR .mod files are not interchangeable because the parameters needed to describe the atmospheres are different in the two cases.

4.6 Mask file

To exclude pixels from the inversion, a FITS file containing a pixel mask can be specified in the init file. The mask is an array of 2 dimensions with (n_x, n_y) elements. A value of 0 indicates that the pixel should not be inverted. A value of 1 means the code will invert the pixel. If no mask is given, all pixels are inverted.

5 Output files

The model atmospheres resulting from the inversion of a FITS data cube are stored in another FITS cube containing an array of 3 dimensions with (number of rows, number of columns, 13) elements. The 13 elements of the third dimension contain the eleven parameters of the model, the number of iterations used by the algorithm to find the solution, and the χ^2 value of the fit, respectively:

1. eta0 = line-to-continuum absorption coefficient ratio
2. B = magnetic field strength [Gauss]
3. vlos = line-of-sight velocity [km/s]
4. dopp = Doppler width [Angstroms]
5. aa = damping parameter
6. gm = magnetic field inclination [deg]
7. az = magnetic field azimuth [deg]
8. S0 = source function constant
9. S1 = source function gradient
10. mac = macroturbulent velocity [km/s]
11. filling factor of the magnetic component [0-1]
12. Number of iterations performed
13. Chisqr value

The best-fit profiles are stored in a FITS Stokes file containing an array with the same dimensions as the observed profiles.

The names of the output model atmospheres and best-fit profiles resulting from the inversion are constructed using the base name of the observed profiles specified in the control file (with no path or extension), adding the prefix indicated in the minit file under the field “outfile” and the suffixes “_mod.fits” and “_stokes.fits”, respectively. So, for example, if the output prefix “results/inv_” is used, the inversion of the cube “data/2014.09.28_09:18:00_xtalk_t020.fits” will be stored in

results/inv_2014.09.28_09:18:00_xtalk_t020_mod.fits
results/inv_2014.09.28_09:18:00_xtalk_t020_stokes.fits

6 Execution

6.1 Sequential execution

The sequential code must be executed by passing the control file as a parameter. This is an example, assuming you are in the *run* directory and the executable is the parent directory:

```
../milos.x pmilos.mtrol
```

6.2 Parallel execution

The parallel code must be executed using the command mpirun or mpiexec. In the local host, one can specify the number of processors to be used with the -np option, as in the following example (with N=16 processors):

```
mpiexec -np 16 ../pmilos.x pmilos.minit
```

It is also possible to run the code simultaneously on several machines via ethernet. In that case, the names of the machines or their IP addresses must be specified in a file *hostnames* using the option -f. The code should be run as follows:

```
mpiexec -f hostnames -np 600 ../pmilos.x pmilos.minit
```

Use the option -iface eth0 in the call to mpiexec if the machines are not reachable from the host computer. Note that SSH keys must be properly installed on every machine, so that connections can be established between them without prompting the user for a password.

The strategy adopted to divide the inversions between the N processors is as follows:

1. If the number of data cubes is equal to or larger than N, then the first N cubes are sent to one processor each. The remaining cubes are processed when the first batch is finished. If there is another set of N cubes, step 1 is repeated. If not, the processing continues with steps 2 or 3.
2. If the number of cubes is larger than or equal to N/2 but smaller than N, then each of the first N/2 cubes is split in two halves and sent to two processors. Once these inversions are finished, the remaining cubes are processed as indicated in step 3.
3. If the number of cubes is smaller than N/2 or new data cubes are arriving at the directory, then each of them is divided into N parts and sent to all N processors one after the other, until all the cubes have been inverted.

The number of available processors (or rather processor cores) limits N. For the best performance, keep in mind the cache size of your processors: it may be better to use a limited number of cores than to split the cubes into all the available cores to avoid replacing information in the cache too often.

In general, the fastest execution is achieved when each processor inverts a single cube. The second fastest strategy is to split each cube in two parts and send them to two processors. The slowest performance occurs when a cube is split into N parts and sent to N processors.

So, for example, if you have a N=64 core machine and you need to invert 60 cubes, you may find that restricting the number of cores in the call to mpiexec with -np 60 will produce a faster inversion than -np 64, because each cube is sent to exactly one core instead of being split into 64 parts and sent to all the available cores.

6.3 Execution on clusters

On HPC clusters, you are usually not allowed to run your code directly and have to submit a job instead.

qsub is a common system for submitting parallel jobs to a cluster. It takes care of distributing the workload among the available computing resources. An example of a qsub script called run.job that can be used to run P-MILOS is given in Table 3. A more detailed tutorial can be found at

https://devwikis.nyu.edu/display/NYUHPC/Copy+of+Tutorial+-+Submitting+a+job+using+qsub

The job is submitted via

```
qsub -V -l nodes=N run.job
```

where N is the number of processors. Be aware that it is usually not necessary to specify N in the call to mpiexec, but in the call to qsub itself. Depending on the system configuration, the option -l nodes=N refers to processor cores or compute nodes. In the latter case it is possible to gain more control using -l nodes=N:ppn=n, where n is the number of processor cores per compute node, so the total number of launched processes is actually N x n.

```
#!/bin/sh
#PBS-N invert
#PBS-e P-MILOS/run/log/errors
#PBS-o P-MILOS/run/log/output

date

cd P-MILOS/run/
mk
mpiexec ../pmilos.x pmilos.minit

date
```

Table 3: Script run.job. The executable pmilos.x is assumed to be in the master directory P-MILOS.

In this example, the submitted job is called 'invert', the error output is stored in the file *P-MILOS/run/log/errors* and the standard output in the file *P-MILOS/run/log/output*.

Some basic information about the state of all running jobs is provided by the command

```
qstat -a
```

6.4 Inverting a time series

P-MILOS can invert all Stokes cubes of a time sequence automatically. This feature is enabled by specifying the t1 and t2 values in the init-file. t1 and t2 are the indices of the first and last time steps to be inverted, respectively. All cubes will be inverted under the conditions indicated in the control file (same number of free parameters, initial model, PSF, stray light, etc).
When a range of time steps is to be inverted, the FITS cubes containing the observed profiles must be named as *basenameNNN.fits*. *basename* is the name specified under the entry "Observed profiles" in the control file (no extension .fits should be present in this case). *NNN* is the time step number, a three-digit integer padded with zeros on the left.

For example, if the name in "Observed profiles" is *stokes* and the user sets t1:9 and t2:11, P-MILOS will invert the cubes *stokes009.fits, stokes010.fits* and *stokes011.fits*. If t1:200 and t2:202, P-MILOS will invert the cubes *stokes200.fits, stokes201.fits* and *stokes202.fits.*

The models and best-fit profiles resulting from the inversion of each cube will be stored in FITS files called *outfilebasenameNNN_mod.fits* and *oufilebbasenameNNN_stokes.fits*, respectively. Here, *basename* is the name of the profiles in the control file, *NNN* the time step number, and *outfile* is a prefix that can be specified in the init file (often used to set the path for the inversion results). Thus, with *outfile=results/inv_*, the output files in our last example would be:

results/inv_stokes200_mod.fits	*results/inv_stokes200_stokes.fits*
results/inv_stokes201_mod.fits	*results/inv_stokes201_stokes.fits*
results/inv_stokes202_mod.fits	*results/inv_stokes202_stokes.fits*

7 An inversion example

The *pmilos.mtrol* and *pmilos.minit* files in the *run* directory can be used to invert two Stokes cubes with names

2014.09.28_09:18:00_xtalk_t020.fits
2014.09.28_09:18:00_xtalk_t021.fits

stored in the *data* directory. The output models are saved in the *results* directory, with prefix *inv_*.

These cubes contain two spectral scans of a sunspot taken by the CRISP instrument at the Swedish 1-m Solar Telescope (La Palma, Spain). The measurements consist of the four Stokes parameters of the Fe I 6173 line sampled at 30 wavelength positions as indicated in the *wavelength.grid* file (or *wavelengths.fits*). The monochromatic images have 894 x 883 pixels. The total scan time was 32 s. More information about the observations can be found in Esteban Pozuelo, Bellot Rubio & de la Cruz Rodríguez (2015).

The data cubes needed to run this example (723 MB) can be downloaded from

https://www.dropbox.com/sh/4cfx5hnaaq90bbl/AAB6QanzdZjQQNegqRMp-aoZa?dl=0