Documentation of P-MILOS

Manuel Cabrera Morales & Luis Bellot Rubio

Contents

1 Overview

This document is a user guide and short documentation of P-MILOS, including a sequential version and a parallel version.

The main contributions of the P-MILOS code compared to IDL-MILOS are:

- Efficient memory management, using single precision by-default.
- Avoid many repeated calculations per iteration when making an investment.
- Calculation of convolutions are more efficient, using direct convolutions by default, instead of FFT.
- Inversion of more than one pixel (typically the whole FOV) using more processor cores.
- Reading the input spectra from standardized FITS[1] files.
- Writing all output data in the standardized FITS format.
- Support for timeseries inversion.
- Efficient use of processor microarchitecture

Please note that this documentation is written for a Linux runtime environment. The C code is portable between platforms, but the Makefile to compile the code, as well as the tests done, have been done only under 64 bits Linux platform.

2 Compilation

The code must be compiled on the destination system. It was tested in two different environments: Ubuntu 18.04 64-bit and Red Hat Enterprise Linux Server release 6 (Santiago) 64-bit, although it may run on any 64-bit Linux system. The following libraries and programs should be installed on your system as a pre-compilation requirement:

- Intel C (icc) Version 11.1 (or higher) 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)

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

- FFTW Version 3.3.3 (or higher)
- GSL Version 1.13-3 (or higher)

In the case of compilers **we strongly recommend** to use the latest version available of the one you are going to use, both icc and gcc, because, we have noticed very important improvements in the performance of the application.

IMPORTANT NOTE:

These improvements are related to taking advantage of modern compilers of all the capabilities that the instruction sets of the machine on which you want to use **P-MILOS** can provide. For example, modern INTEL instruction sets that include the **AVX-512, AVX2, or AVX** vectorised instructions provide a significant increase in program performance. The compilation file is designed to use the highest micro-architecture provided by your microprocessor, so if you are thinking of getting the most out of this program, use processors with the latest generation of instruction sets from INTEL or AMD.

For OpenMPI, the above version is the least tested, but has been used up to version 3.1.3.

For GSL, we are in the same situation. The version of the library specified is the minimum with which the code works correctly, but we recommend the use of version 2.6, which is the one we have found the better performance.

In addition, for FFTW and CFITSIO we recommend the use of the specified versions before. Because, they are the ones we have used in our tests and are the last ones published.

2.1 Make options

In order to deploy the application, this must first be compiled on the target machine. To do this, you must use the command line option 'make' from same directory where the source code is located. Therefore, the first thing is to position ourselves in the P-MILOS directory. The code is compiled by default using SINGLE PRECISION for reduces the quantity of memory allocated during the executions, but if you want DOUBLE PRECISION, you can compile the code specifying the following variable 'use_double=yes' added to make command.

The other options are to choose which version of the code to compile (sequential or parallel), or whether to clean up the generated object code and executables:

- Compile and create executable **milos**

```
make milos
```

- Compile and create executable **pmilos**

```
make milosMPI
```

- Compile and create both: **milos** and **pmilos**

```
make
```

- Compile and create both using double precision:

```
make use_double=yes
```

- Clean objects files and executable files.

```
make clean
```

The executables **milos** and **pmilos** will be located in the same directory where you run the make command.

2.2 Location of the code for download

The latest stable version of P-MILOS is located in the GitHub repository: https://github.com/IAA-InvCodes/P-MILOS

3 Control files

There are two types of control files. The first one are files with extension ".mtrol", they are to control common parameters between sequential and parallel version. The seconds are

files with extension ".minit", they control the parameters to launch parallel version for make time series inversion.

3.1 Mtrol file

Both the sequential and the parallel version need to have an .mtrol file. However, the meaning of the options may be different in each case:

1. Sequential: In the sequential version we can indicate through this file that the program performs the following actions:
   a. Synthesis of a given model.
   b. Inversion of a profile provided through a ".per" file.
   c. Inversion of a full image provided by in a ".fits" file

2. Parallel: Specify location of image sequence to be read.

The layout to the control file is 3 columns, data column between ':' and '!'. **Both characters are mandatory to delimit the data field**. Here the detailed description of the entries:

```
Number of cycles                :0                          ! 0 = synthesis, n= max num of
iterations, -1 = use classical estimates
Observed profiles               :run/init.per               !
Stray light file                :                           ! (none=no stray light contam)
PSF file                        :                           ! .psf file -> read numeric psf; is a
number -> Use a gaussian filter with FWHM (mA); empty -> NOT FILTER
Wavelength grid file            :run/malla.grid  ! (automatic selection, grid, or wavelength file)
Atomic parameters file          :run/LINES                  !
Abundances file                 :                           ! not used
Initial guess model 1           :run/initModel.mod          !
Initial guess model 2           :                           ! not used for the moment
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 used for the moment
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 or blank=no, 1=yes)
Nodes for phi 1	:1	! (0 or blank=no, 1=yes)
Nodes for lambda_doppler 1	:1	! (0 or blank=no, 1=yes)
Nodes for damping 1	:1	! (0 or blank=no, 1=yes)
Invert macroturbulence 1	:1	! (0 or blank=no, 1=yes)
Nodes for S_0 2	:0	! not used for the moment
Nodes for S_1 2	:0	! not used for the moment
Nodes for eta0 2	:0	! not used for the moment
Nodes for magnetic field 2	:0	! not used for the moment
Nodes for LOS velocity 2	:0	! not used for the moment
Nodes for gamma 2	:0	! not used for the moment
Nodes for phi 2	:0	! not used for the moment
Nodes for lambda_doppler 2	:0	! not used for the moment
Nodes for damping 2	:0	! not used for the moment
Invert macroturbulence 2?	:0	! not used for the moment
Invert filling factor?	:0	! not used for the moment
Invert stray light factor?	:0	! (0 or blank=no, 1=yes)
mu=cos (theta)	:1	! (DEFAULT: mu=1. mu<0 => West)
Estimated S/N for I	:1000	!
Continuum contrast	:1e-12	! Not used for the moment
Initial diagonal element	:0.1	! (DEFAULT value: 1.e-1)
Use FFT for convolutions	:0	! (0 or blank = use direct convolution, 1 = use FFT)
Diagonal element acceleration	:0	! (0 or blank=no, 1=yes)

Table 1: Example of an mtrol-file

- **Number of cycles**:
  - Using value -1 the program apply the classical estimates to the profile or profiles, specified in the field *Observed Profiles*. The result is stored in a file with the same root of input file and suffix: "***_ce"***.
  - Using value 0 the program creates a synthesis of Initial guess model 1. The output file has the root name of observed profiles if it is present, in other case the program takes the root of Initial guess model 1.
  - With value greater than 0 the program does the inversion or pixel in the case of .per file or image in the case of .fits file. In this case, the number will be the number of iterations to find a solution. The output models file will have the suffix "**_model**", in both cases, using the root of the input observed profiles file.

- **Observed profiles**: specify the profiles to invest in case of .per file given and the image to invest in case of .fits file given.
- **Stray light file**: It can be a file of type .PER or .FITS (This will be verified by the program, if a file with one of these two extensions is not provided an error will be presented indicating that StrayLight will not be used). If a fits file is specified, it can be of dimensions 2 or 4. If it is of 2 dimensions these will be (number of wavelengths, 4). If it is a 4-dimensional file it will be (number of wavelengths, 4, number of rows, number of columns). The order of the dimensions may be different from that indicated above, except that in the case of 4 dimensions, the dimension of the number of rows must always precede the dimension of the number of columns. For the synthesis and inversion of a profile, only 2-dimensional files will be accepted, with the program displaying an error message if this is not the case.
- **PSF file**: used to specify the name of .psf file with the filter to apply. If you want a Gaussian filter then specify the number of FWHM in mA. If you do not want to apply filter, just put in blank the field.
- **Wavelength grid file**: this field is mandatory. It is for specify the wavelengths to use. You can use a grid file or a fits file. Both types of files admitted will be explained later.
- **Atomic parameters file**: this field is mandatory. File with the spectral lines.
- **Abundances file**: not used at this moment.
- **Initial guess model 1**: File with the initial model for the synthesis and invest cases.
- **Initial guess model 2**: not used at this moment.
- **Weight for Stokes I**: weight of parameter I in the inversion.
- **Weight for Stokes Q**: weight of parameter Q in the inversion.
- **Weight for Stokes U**: weight of parameter U in the inversion.
- **Weight for Stokes V**: weight of parameter V in the inversion.
- **AUTOMATIC SELECT. OF NODES**: not used at this moment.
- **Nodes for S_0 1**: Use 1 to indicate if invert "S0", 0 or blank for not invert.
- **Nodes for S_1 1**: Use 1 to indicate if invert "S1", 0 or blank for not invert.
- **Nodes for eta0 1**: Use 1 to indicate if invert "eta0", 0 or blank for not invert.
- **Nodes for magnetic field 1**: Use 1 to indicate if invert "magnetic field", 0 or blank for not invert.
- **Nodes for LOS velocity 1**: Use 1 to indicate if invert "LOS velocity", 0 or blank for not invert.
- **Nodes for gamma 1**: Use 1 to indicate if invert "gamma", 0 or blank for not invert.
- **Nodes for phi 1**: Use 1 to indicate if invert "phi", 0 or blank for not invert.
- **Nodes for lambda_doppler 1**: Use 1 to indicate if invert "lambda_doppler", 0 or blank for not invert.
- **Nodes for damping 1**: Use 1 to indicate if invert "damping", 0 or blank for not invert.
- **Nodes for macroturbulence 1**: Use 1 to indicate if invert "macroturbulence", 0 or blank for not invert.
- **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 indicate if invert "stray light factor", 0 or blank for not invert.
- **mu=cos (theta)**: Scalar containing the cosine of the heliocentric angle. Default values is 1.
- **Estimated S/N for I**: estimated signal-to-noise ratio. Default value is 1000.
- **Continuum contrast**: not used at this moment.
- **Initial diagonal element**: Initial value for the Levenberg-Marquardt's fudge parameter. Default is 1e-3
- **Use FFT for convolutions**: By default, all the necessary convolutions are performed using direct convolution implementations. If you want to use FFT for calculate convolutions, just mark with 1 this field.
- **Diagonal element acceleration**: By default, this parameter has value 0. If this option is activated, an acceleration process is performed in the convergence of the Levenberg Marquardt algorithm.

3.2 Minit file

Minit file: The parallel version of P-MILOS needs the compulsory init-file (which extends the control file). An example is given in table 2. The layout is similar to the control file (3 columns, data column between ':' and '!'). Here the detailed description of the entries:

- **Name of the .trol file**: the path and name to the control file.
- **Type of input stokes**: for now all images must be specified as fits file.
- **Type of input straylight**: Like the stokes parameters, only fits files are valid. If no straylight file is provided, this field can be left blank.

```
Name of .trol-file (*)          : run/cmilos.mtrol   !
Type of input stokes (*)        :fits                ! (fits)
Type of input straylight        :fits                ! (fits,per)
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	! (prefix for output files)
mask file	:	!(not used at this moment)
t1	:10	! (blank=do not invert timeseries, invert only one image)
t2	:12	! (blank=do not invert timeseries, invert only one image)
Save Best-fit profiles	:1	! (0 or blank = don't save file with adjusted stokes during inversion, 1 save it. Default value is 1)

Table 2: Example of a minit-file

- **subx1, subx2, suby1, suby2**: specify the subregion of the whole input data that should be inverted. If provided, only the rectangular region specified by these pixel indices is inverted. If not set, the whole input data are inverted. The sub-index should range from 1 to the number of rows or the number of columns, as appropriate.
- **outfile**: the common suffix for all the created output files.
- **mask file**: a FITS file containing the pixel mask. If no file is provided, all the pixels are inverted.
- **t1**: first index of a timeseries. The input files of the first image to be inverted have the names $*t_1$.fits.
- **t2**: last index of a timeseries. The input files of the last image to be inverted have the names $*t_2$.fits.
- **Save Best-Fit Profiles**: Indicate if save file with adjusted stokes during inversion.

If t1 and t2 are blank then the fits file specified in the field “Observed profiles” will be inverted.

4 Input files

4.1 Observed profiles

Following the standard *SOLARNET WP 20.3*, the fits files used for pass to the program the spectro image must contain four dimensions: *number_rowsXnumber_cols Xnumber_of_wavelengthsXnumber_stokes*. The order or these parameters can change and for identify each one the header of fits file must contain the type of each dimension with this correspondence:

- Number of Rows: include CTYPE with the value **'HPLN-TAN'**
- Number of Cols: include CTYPE with the value '**HPLT-TAN**'
- Number of Wavelengths: include CTYPE with the value '**WAVE-GRI**'
- Number of Stokes: include CTYPE with the value '**STOKES** '

An example can be this:

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

In the case that these headers cannot be added to the FITS files, files that include only the 4 dimensions will also be accepted: number_rowsXnumber_colsXnumber_of_wavelengthsXnumber_stokes , without headers that indicate what each one corresponds to, as long as they comply with the restriction that the number of rows and columns are specified in the order number_rowsXnumber_cols, never as number_colsXnumber_rows.

As mentioned above the order may not be as specified, but this has an impact on the read speed of the file. If the order of the dimensions of the file is the same as that specified, the reading will be 50% more efficient, since these dimensions will not have to be reordered to adapt them to the form used internally by the application.

The datatype of the images must be in FLOAT_IMG type, since the program will treat them with this type of data and the accuracy will be adjusted to the 32 bits used for the C float representation. The program will exit if detect one image with other datatype.

4.2 Wavelength file

The wavelengths can be provided with a grid file or using a FITS file containing the wavelengths and indices of the observed profiles, (the same information as in the first two columns of .per files). In both cases, the observed spectra of all pixels use the same wavelength grid. These are the two ways explained:

- For Grid format is a file with four columns. The first indicates the number of spectral line to read from the file with spectral lines. The second, third and fourth columns are for specify the range of wavelengths; the values are relative from the central wavelength. The second column is the initial lambda from range, the third the step between wavelengths and the fourth is the end of range.
- For FITS format, the FITS file must contain a single 2D array with dimension number of wavelength-points×2. The first column must contain the index with which the spectral line is identified according to the atomic parameter file. The second column must contain the wavelengths (in mÅ). Examples of these files can be found in the repository, path *run/lambda.fits*.

An error will occur if the data in the FITS file does not fit either of these two possibilities. We recommend the usage of a wavelength grid file. If the number of wavelength points, spectral lines of the observed spectra and the wavelength grid differ, an error will occur. Either a wavelength file, or a wavelength grid file, or both of them must be used. If only a grid file is used, the wavelengths of the observed profiles are assumed exactly the ones of the grid file. If a wavelength file or a grid file is not specified, the code cannot determine the observed wavelengths and will abort with an error.

4.3 Mask file

A FITS file containing the pixel mask. If no file is provided, all the pixels are inverted.

4.4 Init models file

The values for different initial model atmospheres have to be specified in the init models file. It is possible to use only one value for the parameters: `eta_0` (line-to-continuum absorption coefficient ratio), `magnetic field` (magnetic field strength), `LOS velocity` (line-of-sight velocity), `doppler width` (Doppler width), damping (damping parameter), gamma (magnetic field inclination), phi (magnetic field azimuth), S_0 (source function constant), S_1 (source function gradient), `v_mac` (macroturbulent velocity), filling factor (filling factor of magnetic component). The values of the variables must always be as specified below and followed by ":"

`eta_0:`	20

magnetic field:	1100
LOS velocity:	0,2
Doppler width:	0,03
damping:	0,05
gamma:	120
phi:	150
S_0:	0,35
S_1:	0,5
v_mac:	1
filling factor:	1

4.5 Output files

There are four types of outputs files depending of the option and type of files specify in the ".mtrol" file:

Synthesis case: The output in case we are making a synthesis will be the root of the observed profiles name, in case this field is not empty. If it is empty, then the program use the root of initial model atmosphere file provided. In both cases, the extension used for the output file is ".per".

Inversion of a given profile by .per file: the name of output file is compound by the root of observed profiles name (if this field is empty, the root of initial atmosphere file is used) and the suffix"_model.mod".

Inversion of a given image FITS: in this case, the output will be another FITS image. The data is saved in FLOAT precision and the dimensions of image will be: numberOfRows X numberOfCols X 13. The number 13 comes from the eleven parameters of the model, the number of interations used by the algorithm to found the solution in that pixel and the value of Chisqr calculated for the result model of that pixel respect the input profile. Therefore, the order of the third dimension of the file will be:

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. alpha = filling factor of the magnetic component [0->1]
12. Number of iterations needed.
13. Value of Chisqr.

In the case of sequential execution, the name policy is identical to the one followed by the inversion of a ".per" file.

If a timeseries is inverted, the output files are numbered according to the index that already the input files contained. This means that in this case, the number of the time step extends the common prefix for the output files.

Output profiles: The files for each inversion cycle contain an array with the same dimension as the input profiles.

5 Execution

We will differentiate between parallel execution and sequential execution:

5.1 Sequential execution

For the case of sequential execution, you only need to specify as parameter the name of the configured .mtrol file. This is an example:

```
./milos run/pmilos.trol
```

5.2 Parallel execution

The exact command depends on the system. Be aware that on cluster systems, you are usually not allowed to directly run your command but you have to submit a job instead (e.g. using qsub). Usually the command for calling P-MILOS is something like `mpiexec -np N ./pmilos run/pmilos.minit`. Here N is the number of MPI processes that are launched. The strategy followed to divide the work between the N processors is as follows:

1. If the number of images to invert is greater or equal to the number of processes selected, then the number of processes divides the number of images. Each

processor receives the integer part of the division and the rest is of division is scatter as follow.

2. If the rest of division is greater than cero and half the number of processors, then the images are split in half and distributed to two processors.
3. If the rest is larger and not more than half the number of processors, or in the latter case we still have images pending. Then we take those images and divide them among all the processors proportionally.

The number of physically available processors (or rather processor cores) limits N. If you are not working on a cluster system but working on different machines, which are connected via ethernet, the option -iface eth0 might be necessary for the call to mpiexec.

For the best performance, take in mind the size of cache of your physical processors. It may be better to distribute the tasks between different processor cores, to avoid replacing information in the cache.

Example: usage of qsub: qsub is a common system for submitting jobs on a cluster system. It takes care of distributing the workload among the available computing resources. An example of a qsub script for launching P-MILOS is given in listing 1. A more detailed tutorial can be found on https://wikis.nyu.edu/display/NYUHPC/Tutorial+-+Submitting+a+job+using+qsub. qsub is launched via qsub -V -l nodes=N run.job, where N is again the number of processes. Be aware that usually it is not necessary to specify N in the call to mpiexec, but in the call to qsub itself. Depending on the settings of the system the option -l nodes=N refers to processor cores or computing nodes. In the latter case it is possible to gain more control by using -l nodes=N:ppn=n, where n is the number of processor cores per computing node, so the total number of processes launched is then $N \cdot n$.

Listing 1: qsub script run.job

```
#!/bin/sh

#PBS -N invert
#PBS -e p-milos/run/log/errors
#PBS -o p-milos/run/log/output

date

cd p-milos/ mpiexec ./pmilos
run/pmilos.minit date
```

In the example, the 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 states of all running qsub-jobs lists the program qstat -a.

5.3 Inverting a time series

The parallel P-MILOS code optionally can invert all images of a time series automatically. This feature has to be enabled by specifying the t1 and t2 values in the init-file. t1 is the index of the first timestep to be inverted, t2 the index of the last. If a range of timeseries shall be inverted, the input stokes spectra files have to be named prefixtNNN.fits. Prefix is in this case the filename specified in the controlfile (Observed profiles) minus the extension .fits. NNN is the index of the timestep. It has to be a 3-digit integer Therefore, if for instance, the filename for the input Stokes spectra was “stokes0” in the controlfile, with e.g. t1:9 and t2:11, then the program will invert the files “stokes009.fits, stokes010.fits and stokes11.fits”. Another example, with the same input Stokes spectra with t1:200 and t2:202, then the program will invert the files "stokes200.fits, stokes201.fits and stokes202.fits". The output files will be named e.g. prefixtNNN_mod_outputsuffix.fits for models and prefixtNNN_stokes_outputsuffix.fits for synthesis adjusted. The timesteps will be inverted using the same settings.