Documentation of GMILOS

Manuel Cabrera Morales & Luis Bellot Rubio

Contents

1 Overview 2

2 Compilation 2

2.1 Make options 3
2.2 Location of the code for download 4
3 Control files 5

3.1 Mtrol file 5
4 Input files 10

4.1 Observed profiles 10
4.2 Grid file 11
4.3 Init models file 11
4.4 Output files 12

1 Overview

This document is a user guide and short documentation of GMILOS.

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

- Efficient memory management, using single precision by-default.
- Leveraging the parallel process of the GPU for better performance.
- Avoid many repeated calculations per iteration when making an investment.
- Calculation of convolutions are more efficient, using direct convolutions 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.

Please note that this documentation is written for a Linux runtime environment. The CUDA 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 6.8 (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:

- CUDA runtime environment and toolkit.
- CFITSIO Version 3.3.4.0 (or higher)
- GSL Version 1.13-3 (or higher)

IMPORTANT NOTE:

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

For GSL, 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 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.

CUDA Requirements

Since the CUDA code has been developed without using any library, the only restriction is the one imposed by the use of dynamic parallelism. Dynamic parallelism is available in CUDA 5.0 and later on devices of Compute Capability 3.5 or higher (sm_35). (See NVIDIA GPU Compute Capabilities.) Therefore, a version later than 5.0 of CUDA must be installed on the system and the GPU must have a Compute Capability greater than or equal to 3.5.

2.1 Make options

In order to deploy the application, it 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. So, the first thing is to position ourselves in the GPU-MILOS.

There are two environment variables that you must define in your console, they are CUDA_PATH and SMS. The first one expresses the path where CUDA Toolkit is installed in your machine and the second one expresses the CUDA code compatibility with which it will be compiled (this value must be known from your graphic card specifications). Here is an example of how to create the two environment variables using a bash command console. If you want to make these two variables permanent, you can add them to your ~/.bashrc file.

```
export CUDA_PATH="/usr/local/cuda-10.1"
export SMS="35"
```

If these two environmental variables do not exist, by default the makefile compile the code with compatibility for this generations of CUDA: 35 37 50 52 60 70 , and using the CUDA_PATH "/usr/local/cuda-10.1" .

- Compile and create executable **gmilos**

```
make
```

- Clean objects files and executable files.

```
make clean
```

The execution of the code for the inversion of a cube is very simple, just pass as a parameter to the executable the name of the control file. Obviously, this control file must have the appropriate configuration. In it you can choose the GPU in which you want to make the inversion (in case there are more than one installed in the computer) and the number of streams to use for the inversion.

If you do not know how many GPUs you have on your system, you can use the command line for the "nvidia-smi" command, no need arguments. This will tell you how many GPUs you have on your system, and the number assigned to each device.

CUDA does not restrict the maximum number of streams that can be used on a GPU, but from the tests we have done this should be no more than 16.

An example positioning us in the same directory where the executable has been created, being the control file "run/invert.mtrol":

```
./gmilos run/invert.mtrol
```

2.2 Location of the code for download

The code can be downloaded from the repository: https://github.com/IAA-InvCodes/GPU-MILOS

If you are not familiar with the use of GIT we recommend that you follow this link, which explains the basic steps for performing repository cloning on your local machine.

We give an example of the two most common uses, cloning a repository and updating it.

To clone a repository, we open a command console, navigate to the directory where we want to clone the repository and execute this command:

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

To update it, if necessary, once it has been cloned. Always from a command console, you navigate to the cloned directory and execute the next command (Important, if you make changes to one of the files in the repository, git will mark it as a conflict and you will not be able to update it).

```
git fetch && git pull
```

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. They control the parameters to launch parallel version for make time series inversion.

3.1 Mtrol file

The program needs to have an .mtrol file in the input to control the options to use in the execution.

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)
Diagonal element acceleration	:0	! (0 or blank=no, 1=yes)
Number of Streams	:2	! (number of streams to overlap
GPU process)		
Device number	:0	! (0 or blank=0) number of device
to use in local machine for inversion		

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
- **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.
- **Number of Streams:** Indicates the number of streams to be used in the execution of the code on the GPU. A stream is a way of distributing work on the GPU in which the pixels to be inverted are divided by the number of streams and in this way when there is a wait on one stream another can make use of the GPU's resources. The number to use should be approximated by the user depending on their GPU. We recommend starting with a minimum value of 4 streams to create a sufficient minimum overhead. With that value for an old card will be a good choice. In cards

that are more modern, you can increase this value up to 16, which is what we have achieved as an optimal empirical limit.

- **Device number**: Indicates the number of the GPU in the system. This field is useful when we have more than one GPU in our system and we want to choose in which GPU the program will run. Its default value is 0, in case there is only one GPU installed in the system.

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* X*number_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 Grid file

The wavelengths can be provided in a grid file containing four parameters. The first indicates the index of the spectral line in the atomic parameter file. The second, third and fourth parameters specify the wavelength range. All values are relative from the central wavelength. The second column is the initial lambda, the third the wavelength step, and the fourth the final lambda of the wavelength grid.

This is an example:

Line indices (in this order)	:	Initial lambda (mA)	Step (mA)	Final lambda (mA)

1	:	-350,	35,	665

4.3 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
```

4.4 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.

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