# VFISV inversion pipeline for GRIS

G. Vigeesh, J. M. Borrero, N. Bello González, C. Schaffer

Draft
November 2021

## Contents

**1 Introduction** **2**

**2 Stokes inversion** **2**

**3 The Pipeline** **3**

**4 Standard data product** **7**

**5 Testing and Analysis** **10**

**6 Conclusions** **14**

# 1 Introduction

The Science Data Center (SDC) provides users with the vector magnetic field and line-of-sight velocity data of the GRIS (GREGOR Infrared Spectrograph; Collados et al. 2012) spectropolarimeteric observations. This standard Level 2 (L2) data product is computed using Very Fast Inversion of the Stokes Vector (VFISV; Borrero et al. 2011), a Milne-Eddington inversion code. This technical report describes the inversion pipeline code implemented for GRIS data archive.

# 2 Stokes inversion

A brief summary of the inversion procedure is given here. Please refer to Borrero et al. (2011) for a more detailed information about VFISV inversions.

Given a set of observations, an inversion algorithm aims to find the best physical model that reproduces the observations. In the case of Stokes inversions, the set of observations are the Stokes vector $(I, Q, U, V)$ at discrete wavelength position $\lambda_i$. The model parameters ($\mathcal{M}$) are physical quantities along the line of sight in an atmosphere that may have an influence on the emergent Stokes signal. A simple case of an atmosphere involves assuming that the physical parameters do not vary along the line-of-sight, called a Milne-Eddington (M-E) atmosphere. In the case of M-E, the inversion proceeds first by computing a synthetic profile from a guess set of model parameters, and iteratively improves the synthetic profile to match the observed profile to finally give a set of best fit model parameters. The iterative step involves in the minimization of a cost function or $\chi^2$. In the case of VFISV, the cost function is defined as a weighted ($w_j$) sum of the mean squared error (MSE) between the observed and the synthetic Stokes vector (see Eq. 1 of Borrero *et al.* 2011), as

$$\chi^2 = \frac{1}{N}\sum_{j=1}^{4} w_j \cdot \mathrm{MSE}(I_j) = \frac{1}{N}\sum_{j=1}^{4} w_j \sum_{i=1}^{L} \left[I_j^{\mathrm{obs}}(\lambda_i) - I_j^{\mathrm{syn}}(\lambda_i, \mathcal{M}_h)\right]^2, \tag{1}$$

where $\mathcal{M}_h$ refers to the model parameters used to construct the synthetic profile at $h$th iteration and $N = 4L - F$ refers to the number of degrees of freedom in the inversion. Here $L$ is the number of observed wavelength points and $F$ refers to the list of free parameters that describe the M-E atmosphere. They are:

- $S_0$ - the source function at the observer
- $S_1$ - the gradient of the source function
- $\eta_0$ - center to continuum absorption coefficient
- $a$ - damping
- $\Delta\lambda_D$ - Doppler width of the spectral line
- $B$ - the magnetic field strength
- $\gamma$ - the inclination of magnetic field with respect to the observer
- $\Psi$ - the azimuth of the magnetic field vector in the plane perpendicular to the observer
- $v_{\mathrm{los}}$ - line-of-sight velocity of the plasma
- $v_{\mathrm{mac}}$ - the macroturbulent velocity
- $\alpha_{\mathrm{mag}}$ - filling factor

The minimization is carried out using the Levenberg–Marquardt minimization algorithm (Press *et al.* 1986). A detailed description of the minimization procedure can be found in Borrero et al. (2011). A discussion about the weights used in the pipeline and test based on different weights is presented later in Section 5.

# 3 The Pipeline

The current implementation of the pipeline is a modified version of VFISV code to work with the GRIS data. The GRIS Level 1 (L1) header and Stokes data is extracted using a Python core module and sent to a bare VFISV via an MPI intercommunicator. The inversion is performed using VFISV and the buffer with the inversion results is communicated back to the Python module. The Python module propagates the header keywords from L1 and packages the inversion results and outputs the inversion results as FITS file (when used as a command-line interface) or returns NDarrays (when called within a script).

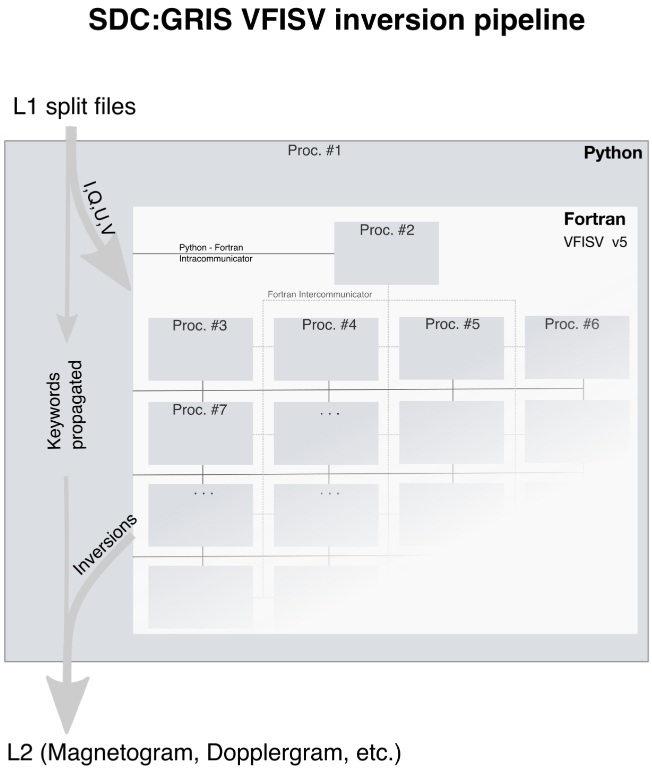


**Figure 1:** Workflow showing the pipeline running on $N$ processors, with the main python processor spawning the VFISV inversion on $N-1$ processors.

### Install and usage

The pipeline code is provided as a Python package. It can be installed natively using the python `pip` command. But, the most easiest and quickest way to start using the pipeline is with help of Docker.

### Using Docker

A container image is now available with the VFISV pipeline code pre-installed along with all required dependencies. To start using with Docker, first get the Level 1 split files which is downloadable from the GRIS archive. In the location where the data folder (e.g. `level1split/`) is copied, issue the Docker pull/run command,

```
docker run -it --rm -v $PWD:/home/grisuser ghcr.io/vigeesh/sdc-grisinv
```

This will pull and run the latest image from the container repository and mount the present working directory under `/home/grisuser` inside the container. Once within the new container shell, you can run the inversion code as follows,

```
mpiexec -n 1 vfisv \
  --path='level1split/' \
  --id=3 \
  --line=15648.514 \
  --numproc=20 \
  --width=1.8 \
  --out='output.fits'
```

More information about the command line arguments follows below.

### Native installation

If you want to install the pipeline locally, please follow the instructions below.

### Requirements

It is highly recommended to install the pipeline in a new conda environment. If you don't have conda installed, check Miniconda for a minimal installer for conda. To install Miniconda:

```
wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh
```

when prompted, install it in your home path (e.g. `~/conda`). Conda is cross-language and can install non-Python libraries and tools (compilers, OpenMPI) in the user space. We recommend, you use the `conda-forge` channel for installing the required packages.

> Note: In the conda-forge channel, NumPy is built against a dummy "BLAS" package. When a user installs NumPy from conda-forge, that BLAS package then gets installed together with the actual library - this defaults to OpenBLAS ... (see Numpy Documentation for more info.)

First, create a conda environment (e.g named `gris_env`) and install the required packages from `conda-forge`. Next, activate the newly created conda environment.

```
conda create -n gris_env -c conda-forge \
      mpi4py numpy scipy gfortran_linux-64 \
      lapack matplotlib
conda activate gris_env
```

### Installation

You can install the pipeline code from the GitLab repo directly using pip

```
pip install git+https://gitlab.leibniz-kis.de/gris/sdc/grisinv.git
```

or clone the repo and run setup

```
git clone https://gitlab.leibniz-kis.de/gris/sdc/grisinv.git
cd grisinv
python setup.py install
```

This builds the code and installs the command-line tool (vfisv) and in your conda environment. Set the `LD_LIBRARY_PATH` so that the MPI libraries provided by conda takes precedence.

```
export LD_LIBRARY_PATH=$CONDA_PREFIX/lib:$LD_LIBRARY_PATH
```

### Help

To find the available command line args for the VFISV code,

```
Usage: vfisv [OPTIONS]

  SDC:GRIS Inversion Pipeline code

Options:
  -p, --path TEXT         Path to the fits files
  -d, --id INTEGER        Observation ID
  -o, --out TEXT          Output fits file
  -n, --numproc INTEGER   Number of processors to run on
  -l, --line FLOAT        Wavelength of the line Angstrom
  -w, --width FLOAT       Wavelength range in Angstrom
  --preview TEXT          Filename to save the plot
  --errors TEXT           Filename to save the uncertainties
  --diagnose TEXT         Filename to write the inversion fits
  --log TEXT              Filename to write the log file
  -h, --help              Show this message and exit.
```

### Example usage

To run the VFISIV inversion on multi-processor, you can either call mpiexec with one processor and set the –numproc to the required number of processors to run the inversion on.

```
#activate your conda enivorment where grisinv is installed.
conda activate gris_env
#only once after activate
export LD_LIBRARY_PATH=$CONDA_PREFIX/lib:$LD_LIBRARY_PATH

mpiexec -n 1 vfisv \
  --path='/dat/sdc/gris/20150919/level1_split/' \
  --id=3 \
  --line=15648.514 \
  --numproc=20 \
  --width=1.8 \
  --out='output.fits'
```

or, call mpiexec directly with the required number of processors and the python core will distribute the inversion.

```
mpiexec -n 20 vfisv \
  --path='/dat/sdc/gris/20150919/level1_split/' \
  --id=3 \
  --line=15648.514 \
  --width=1.8 \
  --out='output.fits'
```

To output uncertainties, use the --errors='<filename_errors.fits>' option. If you want to check the line fits, use --diagnose='<filename_diagnose.fits>'. To account for observations with multiple maps, filename_diagnose is appended by the map-number, e.g. filename_diagnose_001.fits. A run with all the options will look like,

```
mpiexec -n 1 vfisv \
  --path='/dat/sdc/gris/20150919/level1_split/' \
  --id=3 --line=15648.514 \
  --numproc=20 \
  --width=1.8 \
  --out='test_inversion.fits' \
  --preview='test_preview.png'  \
  --errors='test_errors.fits' \
  --diagnose='test_diagnose.fits' \
  --log='test_log.txt'
```

### Python interface

The pipeline can be also be run within a python script. To do so, you need to call the python using mpiexec with one processor. With example.py:

```
from grisinv import vfisv
inversions, fits, header = vfisv('/dat/sdc/gris/20150920/level1_split/',5,15648.514,1.8,20)
```

giving the path to the L1 split files, the observation id (5), the wavelength of the line (15648.514), the ±width of the spectral region (1.8), and the number of processors to use, you can run,

```
mpiexec -n 1 python example.py
```

### Displaying results

For a quick-look and preliminary analysis, the fits file can be easily viewed using SAOImage DS9. Using the multiframe and with match and lock, you can view all the HDU in the fits file.

```
ds9 -multiframe \
    -match frame image \
    -lock slice image \
    -lock frame image \
    -single -zoom to fit output.fits
```

or, with python (using sunpy)

```
from sunpy.map import Map
m = Map('output.fits')
#Fitted continuum
m[0].peek()
#Line-of-sight velocity
m[1].peek()
#Inclination
m[2].peek()
#Azimuth
m[3].peek()
```

## 4 Standard data product

The SDC provides inversion results for the entire archive. This section explains the standard data product available via SDC webpage.

### L2 files

For every GRIS slit spectropolarimetric observation, the inversion results are available as L2 downloadable tar file. The tar contains several files. Here, for example, the observation: `gris_20170902_006`, the L2 files are:

A readme file describing the details of the run, including the path, the spectral line, the width, the total number of processors used, the time it took for performing the inversions, the chi-square value.

```
1. gris_20170902_006_inversion_readme.txt
```

The inversion result with the line-of-sight velocity ($v_{\rm los}$), the magnetic field strength ($B$), the inclination angle ($\gamma$), and the azimuth ($\Psi$) are provided as a SOLARNET compliant FITS files.

```
2. gris_20170902_006_inversion.fits
```

The uncertainties of the above quantities along with the $\chi^2$ for each pixel is provided as,

```
3. gris_20170902_006_inversion_uncertainty.fits
```

The observed Stokes profiles and the final fits to the profile by VFISV is also given for individual maps in the observation.

```
4. gris_20170902_006_inversion_diagnose_000.fits
5. gris_20170902_006_inversion_diagnose_001.fits
6. ...
```

Finally, quicklook images of the $B_{\rm los}$, $B_{\rm tra}$, and $v_{\rm los}$ are also provided. These preview images are available in the info page of the individual observations.

```
1. gris_20170902_006_inversion_bpara_preview.png
2. gris_20170902_006_inversion_bperp_preview.png
3. gris_20170902_006_inversion_vlos_preview.png
```

### Spectral line

GRIS observes in two wavelength region, one in the 10830 Å region and the other one in the 15650 Å range as shown in Figure 2. Many important lines are covered in the 10830 range, among them the Si I 10827.091 Å and the Ca I 10838.970 Å lines are quite relevant for Solar physics. However, we have selected Ca I 10838.970 Å as the main spectral line as part of the standard data product. In the case of the 15650 Å range, we chose the Fe I 15662.017 Å line for the standard data product. In both cases, we set a spectral width of $\pm 1.8$ Å to carry out the inversions.

### Header definition

Keywords are propagated from the GRIS Level 1 to Level 2 along with new keywords that are created by the pipeline. Many keywords are retained from the L1 header. Some are changed to account for the new dataset. VFISV pipeline adds its own list of keywords under "Inversion Parameters". For the temporal information, we modify the keywords adding a start_time, end_time, and the avg_time based on Section 4. of the SOLARNET standards. The pipeline also adds the provenance in the end of the fits header list.

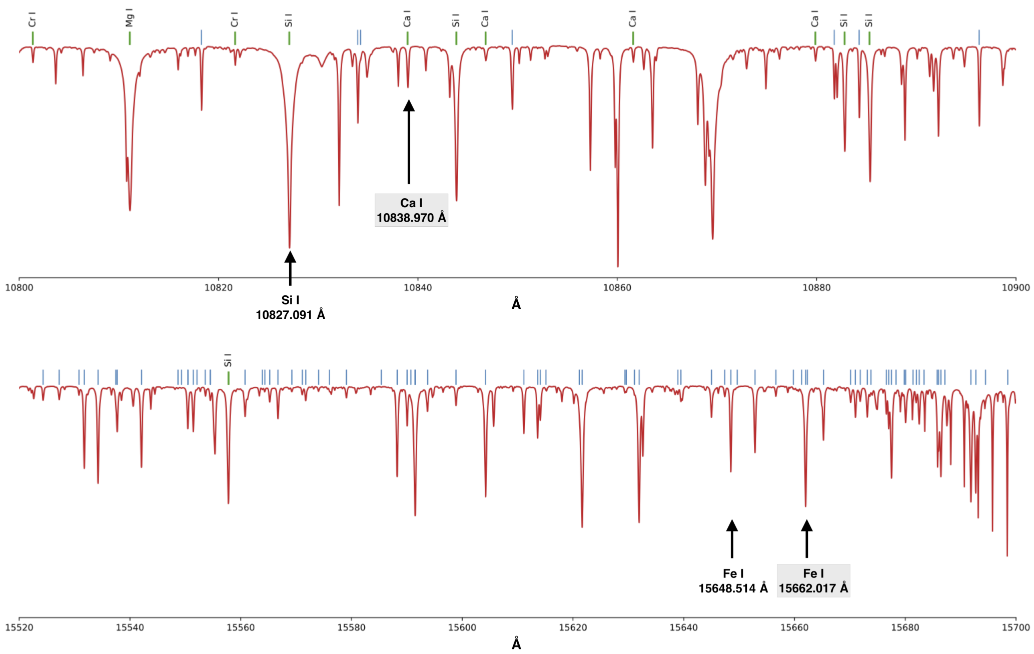


**Figure 2:** The two spectral ranges covered by GRIS showing the prominent lines. The standard inversion product delivered by SDC are for the Ca I 10838.970 Å line for the 10830 Å range and the Fe I 15662.017 Å line in the 15650 Å range.

## Archival Coverage

Figure 3 shows the archival coverage from 2014-2020 for which the inversions are available for download from the SDC webpage.

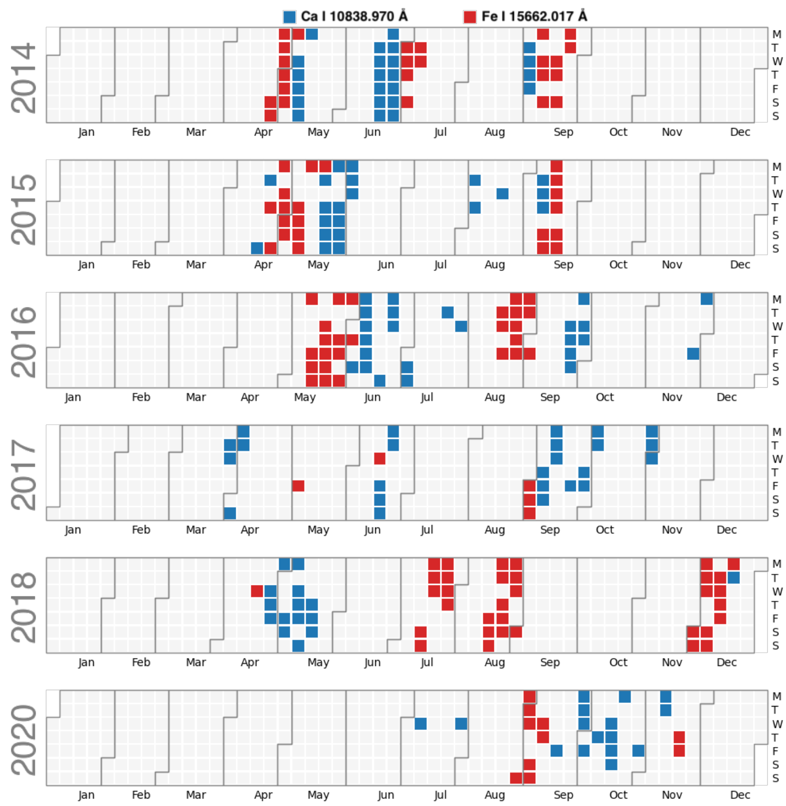


**Figure 3:** Archival coverage from 2014-2020 for which the inversions are available. The blue square show the dates for which the inversion for the Ca I 10838.970 Å are available and the red square show the dates for which the inversion for the Fe I 15662.017 Å are available.

Figure 4 shows the histogram of the $v_{\rm los}$, $B_{\rm los}$, and $B_{\rm tra}$ over the entire archive from 2014-2020, obtained using inversions with the Ca I 10838.970 Å line.

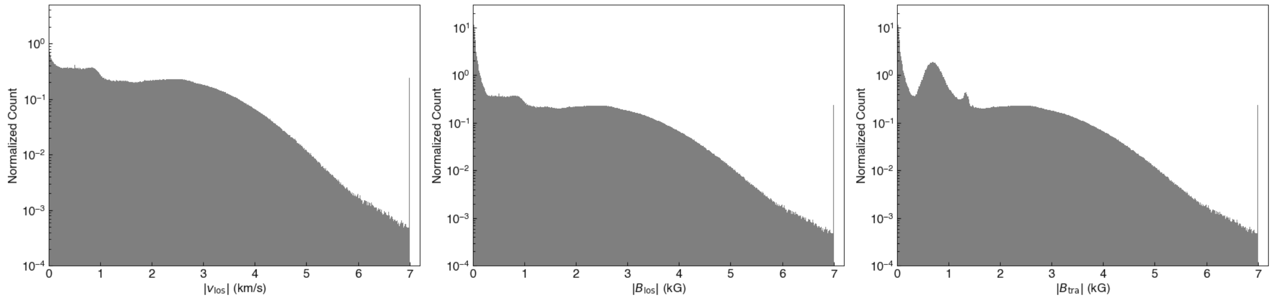


**Figure 4:** Histogram of the $v_{\rm los}$ (left), $B_{\rm los}$ (center), and $B_{\rm tra}$ (right) over the entire archive from 2014-2020, obtained using inversions with the Ca I 10838.970 Å line.

Figure 5 shows the histogram of the $v_{\rm los}$, $B_{\rm los}$, and $B_{\rm tra}$ over the entire archive from 2014-2020, obtained using inversions with the Fe I 15662.017 Å line.

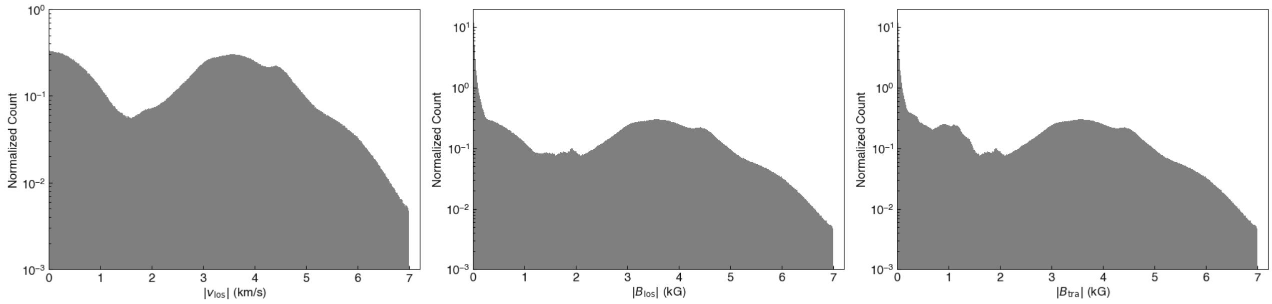


**Figure 5:** Histogram of the $v_{\rm los}$ (left), $B_{\rm los}$ (center), and $B_{\rm tra}$ (right) over the entire archive from 2014-2020, obtained using inversions with the Fe I 15662.017 Å line.

### Continuous integration and continuous delivery (CI/CD)

The pipeline is planned to be under continuous development with contributions from existing and new users. The repository containing the source code is publicly available and is thought to the main contact point for issue tracking. To ensure proper automation and continuous development, we have implemented a CI/CD for automated build, test, and merge with the public repository.

### Performance

We performed a single node scaling test of the pipeline code. Figure 6 shows the speedup as a function of processors on three different machines. The dots represent the full pipeline performance, including the I/O. The plus markers represent only the inversion part of the pipeline, which shows a close to ideal scaling, except for one machine. The archive data have been inverted using 20 processors.

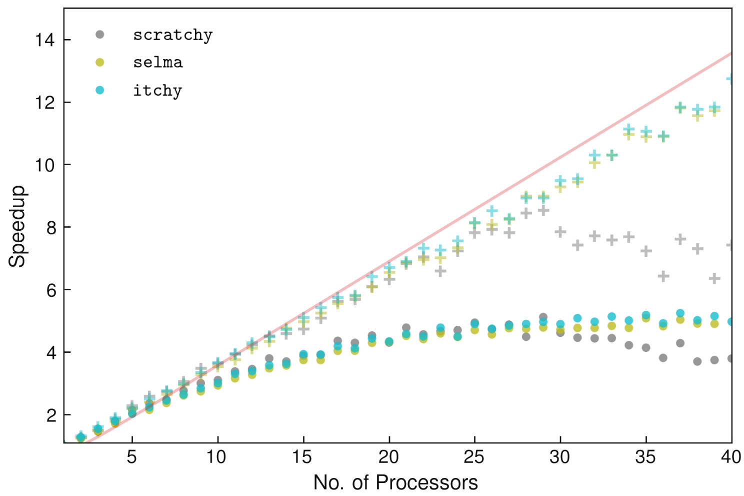


**Figure 6:** Speedup as a function of processors on three different machines. The dots represent the full pipeline performance, including the I/O. The plus markers represent only the inversion part of the pipeline.

## 5 Testing and Analysis

The inversions using VFISV was carried out with a defined set of weights (see Section 2 for the definition of weights). The weights used in the pipeline code is not fixed for the entire archive, but are computed for each observations based on the noise level. Here we have performed few tests with fixed weights and also compared it with the inversions obtained using the SIR code (Ruiz Cobo & del Toro Iniesta 1992). We present here two sunspots that were analysed using both inversion codes and present some comparisons.

### Case 1

The sunspot observations was carried out on April 23, 2015, starting with the first slit position at 09:46:36 UTC. The heliocentric angle at the center of the field of view (FOV) was 5°. Figure 7 shows the location of the sunspot and the intensity on the helioprojective cartesian grid.

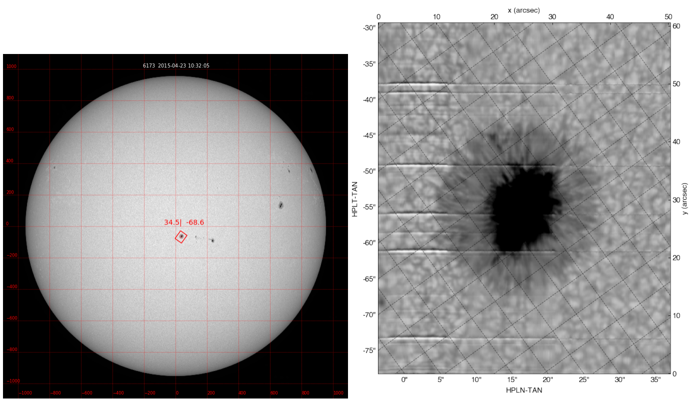


**Figure 7:** Left: The location of the FOV on the HMI image taken at the same time of the observation. Right: the intensity image of the sunspot.

GRIS observed the sunspot in the 15650 Å spectral window. We performed inversions of the Fe I 15662 Å and the Fe I 15648 Å lines using different weights. With VFISV, we performed inversions with the default weights calculated based on the noise and with fixed weights: $w_j = \{1, 2, 2, 2\}$ and $w_j = \{2, 1, 1, 1\}$. The SIR inversion were performed with the same input dataset with single-node for two sets of fixed weights: $w_j = \{1, 2, 2, 2\}$ and $w_j = \{2, 1, 1, 1\}$. Figure 8 compares the residuals (difference between SIR and VFISV values) of the $v_{\rm los}$ (top panels), $B_{\rm los}$ (middle panels), and $B_{\rm tra}$ (bottom panels) as a function of $B$ for the Fe I 15648 Å line. Here we compare the inversion between different weights. Similarly, the results from the Fe I 15662 Å line are shown in Figure 9. We see that the inversions performed using VFISV are consistent with

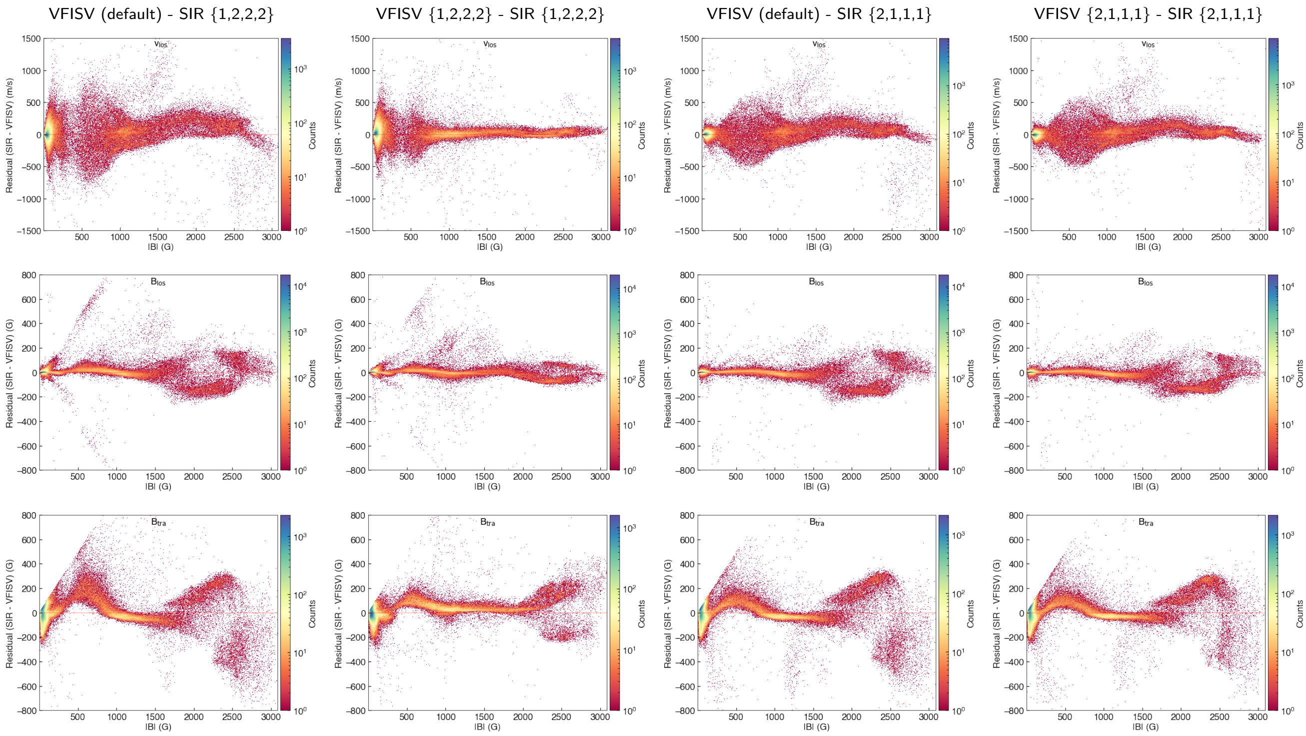


**Figure 8:** Density plot showing the residuals (VFISV - SIR) of the $v_{\rm los}$ (top panels), $B_{\rm los}$ (middle panels), and $B_{\rm tra}$ (bottom panels) as a function of $B$ for the Fe I 15648 Å line. Inversions with different weights are compared.

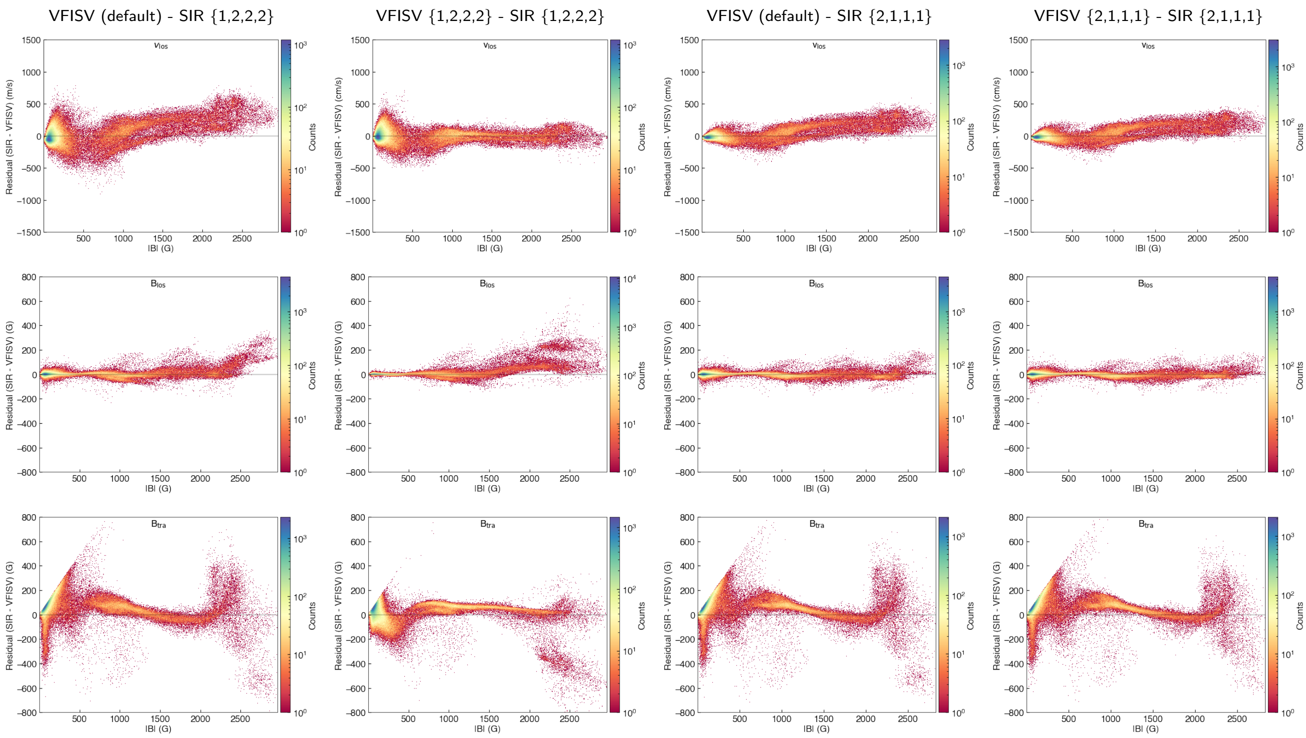


**Figure 9:** Density plot showing the residuals (VFISV - SIR) of the $v_{\rm los}$ (top panels), $B_{\rm los}$ (middle panels), and $B_{\rm tra}$ (bottom panels) as a function of $B$ (SIR) for the Fe I 15662 line. Inversions with different weights are compared.

that obtained using SIR for all combinations of weights. It can be see that, when the same set of fixed weights are used for VFISV and SIR we get a more tight correspondence between the two. Using $w_j = \{2, 1, 1, 1\}$ for

VFISV shows the best match with SIR.

## Case 2

A second sunspot with observations carried out on September 2, 2017, starting with the first slit position at 09:07:23 UTC is also considered. The heliocentric angle at the center of the field of view (FOV) was 25°. Figure 10 shows the location of the sunspot and the intensity.

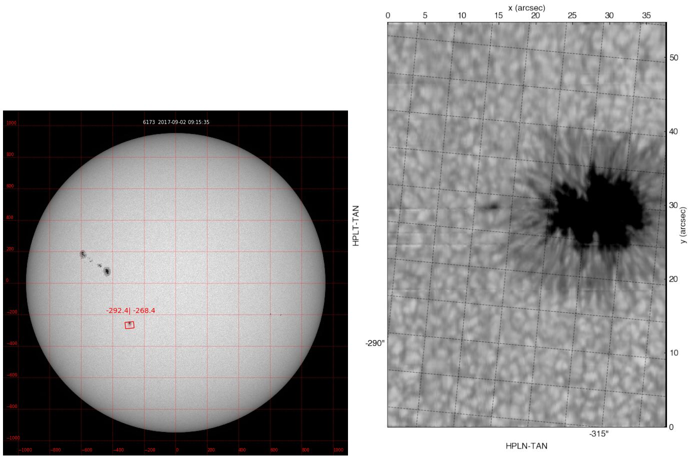


**Figure 10:** Left: The location of the FOV on the HMI image taken at the same time of the observation. Rigth: the intensity image of the sunspot.

GRIS observed the sunspot in the 15650 Å spectral window. Similar to Case 1 presented above, we performed inversions of the Fe I 15662 Å and the Fe I 15648 Å lines using different weights. Figure 11 compares the residuals (difference between SIR and VFISV values) of the $v_{\rm los}$ (top panels), $B_{\rm los}$ (middle panels), and $B_{\rm tra}$ (bottom panels) as a function of $B$ for the Fe I 15648 Å line. Here we compare the inversion between different weights. Similarly, the results from the Fe I 15662 Å line are shown in Figure 12.

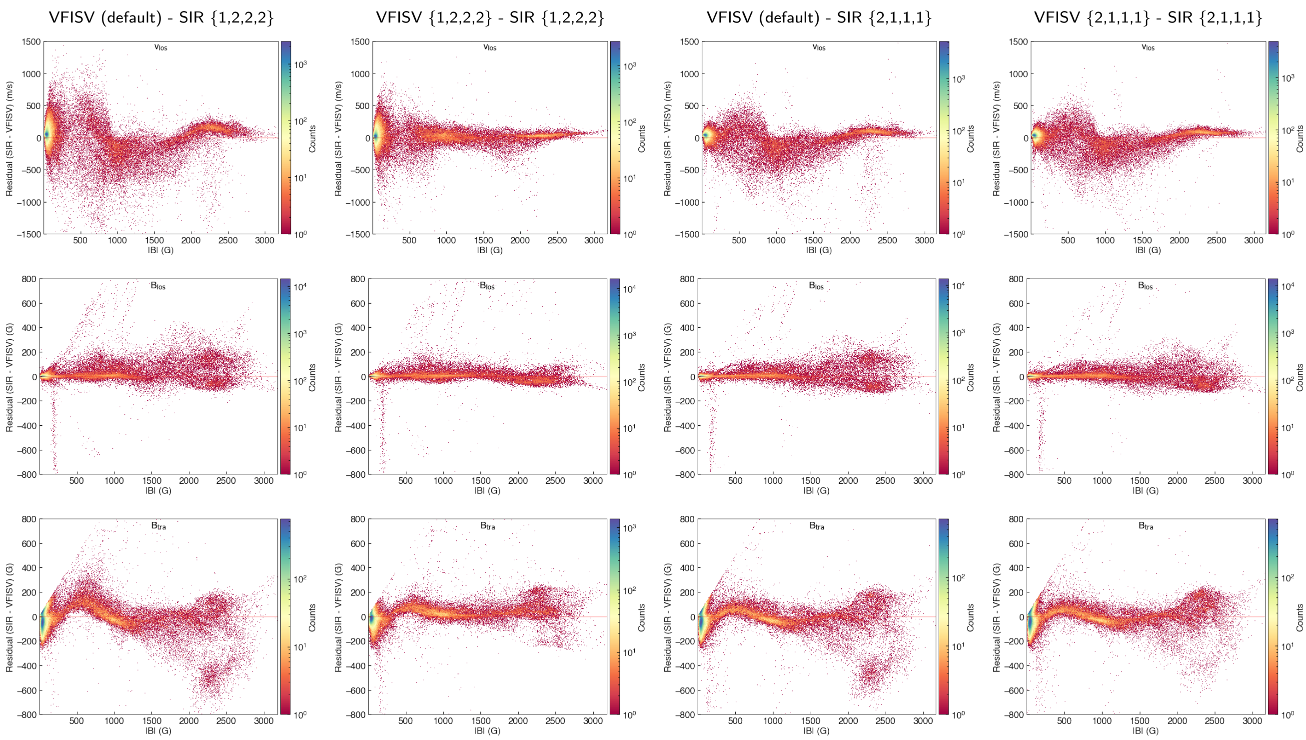


**Figure 11:** Density plot showing the residuals (VFISV - SIR) of the $v_{\rm los}$ (top panels), $B_{\rm los}$ (middle panels), and $B_{\rm tra}$ (bottom panels) as a function of SIR $B$ for the Fe I 15648 line. Inversions with different weights are compared.

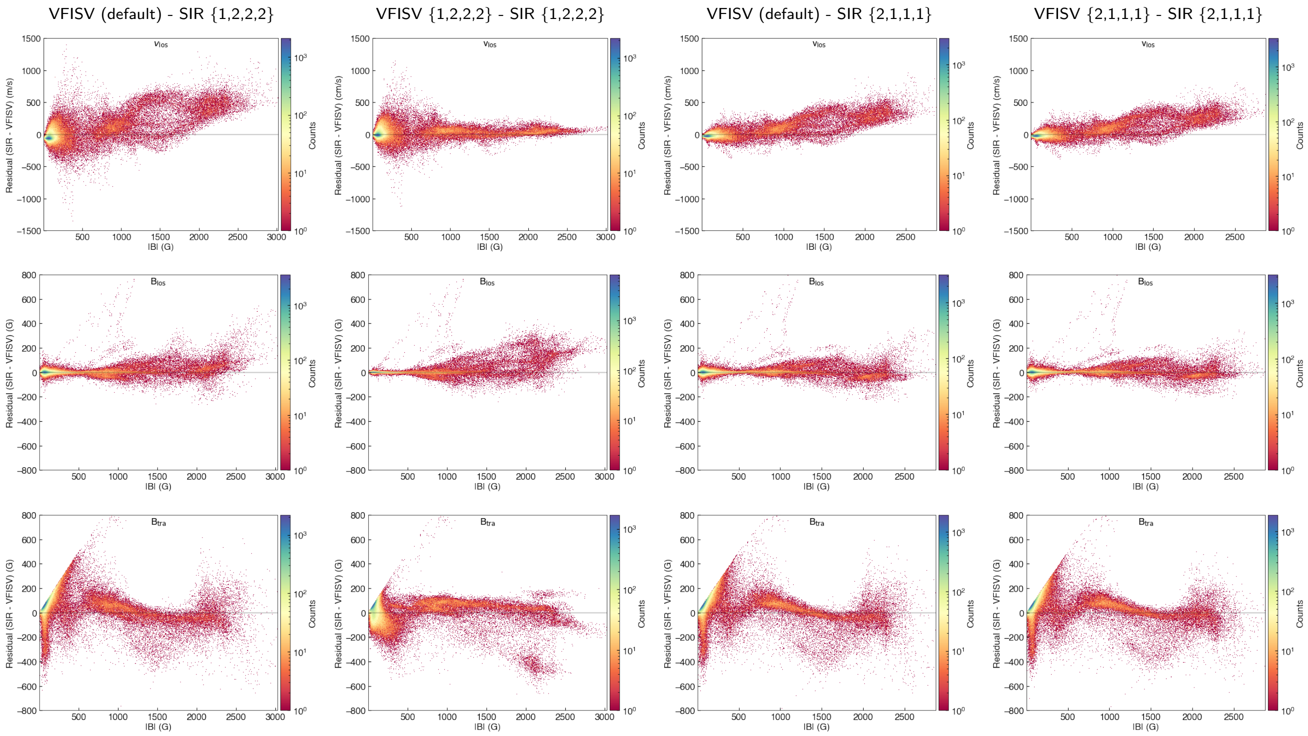


**Figure 12:** Density plot showing the residuals (VFISV - SIR) of the $v_{\rm los}$ (top panels), $B_{\rm los}$ (middle panels), and $B_{\rm tra}$ (bottom panels) as a function of SIR $B$ for the Fe I 15662 line. Inversions with different weights are compared.

Here again, we see that the inversions performed using VFISV are consistent with that obtained using SIR for all combinations of weights. Like in the case of the previous dataset (Case 1), using $w_j = \{2, 1, 1, 1\}$ for VFISV shows the best match with SIR.

For this particular sunspot case, we also present in Figure 13 the location and fits for Stokes spectra at three pixels representing a quiet Sun (blue point in Fig. 13(leftmost); left panels in Fig. 13), penumbra (orange point; middle panels), and umbra (green point; right panels). The observed spectra is marked in black, and the fits performed by SIR and VFISV with different set of weights are also shown. The red and orange curves represent the fits from SIR with weights $w_j = \{1, 2, 2, 2\}$ and $w_j = \{2, 1, 1, 1\}$, respectively. The fits obtained with VFISV are shown in blue for the default weights, in cyan for $w_j = \{1, 2, 2, 2\}$ and in purple for $w_j = \{2, 1, 1, 1\}$. The residuals are plotted under each panel. Although the weights $w_j = \{2, 1, 1, 1\}$ show a good correspondence with the SIR, from the residual plots, we see that VFISV with $w_j = \{1, 2, 2, 2\}$ fits best to the observed spectra.

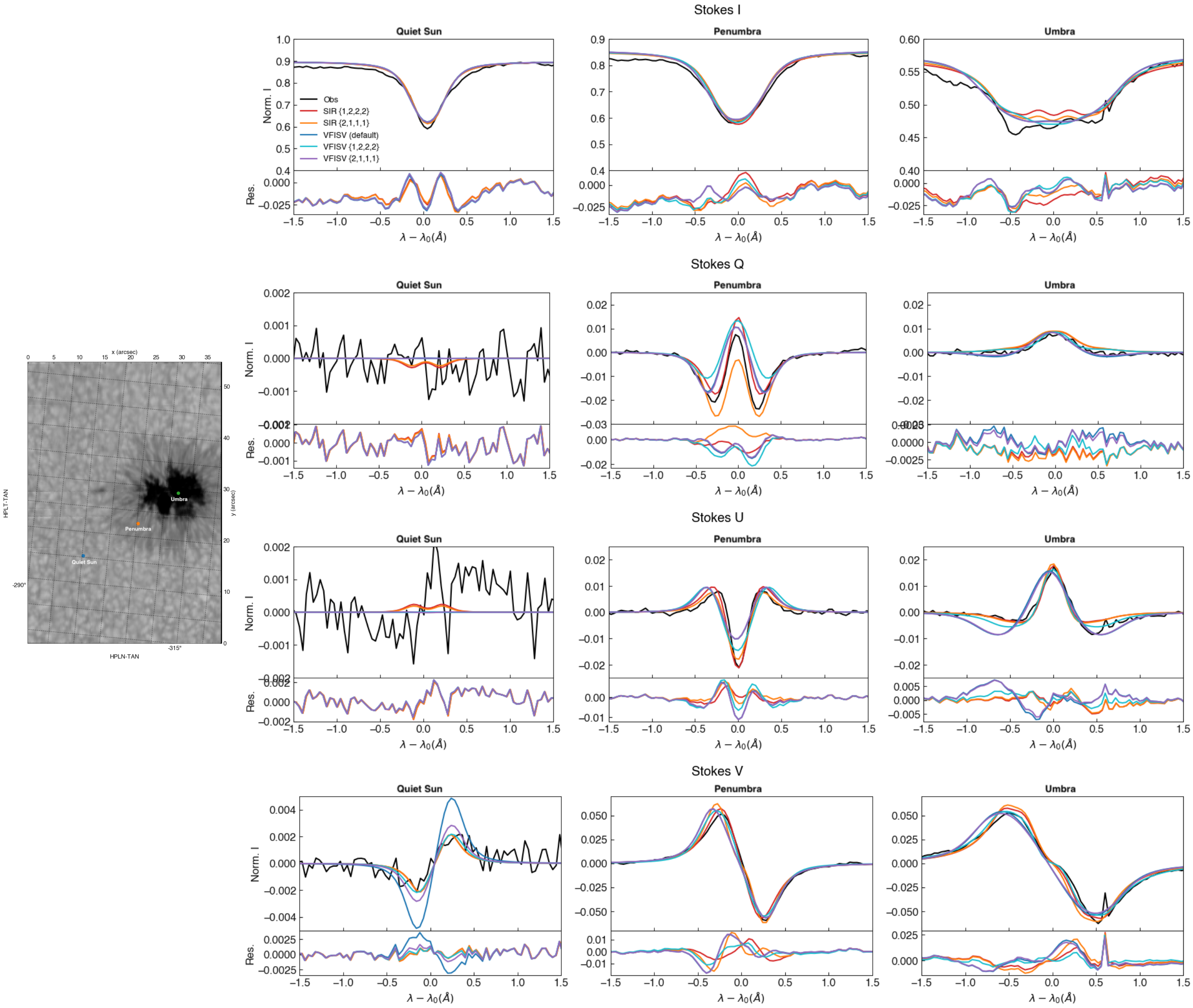


**Figure 13:** Leftmost panel: Intensity map of the Sunpsot observed on September 2, 2017, starting with the first slit position at 09:07:23 UTC. Three point are marked at locations in quiet Sun (blue point), penumbra (orange), and umbra (green). The panels on the right show the observed Stokes spectra in black and the different fits performed by SIR and VFISV with different set of weights as explained in the text. The residuals are shown in the bottom plots of each panel.

# 6 Conclusions

We present here the inversion pipeline code that is implemented for the GRIS spectropolarimeteric observations. The pipeline provides users with vector magnetic field and line-of-sight velocity data. The Stokes inversions are carried using the Very Fast Inversion of the Stokes Vector (VFISV), a Milne-Eddington inversion code. In this report, we present the standard data product available as part of the SDC. The weights used in the pipeline code is computed for each observations based on the noise level. A comparison of different weights were presented and also a comparison with SIR was discussed. We find that fixed weights of $w_j = \{1, 2, 2, 2\}$ give a better fit and is also consistent with SIR inversions.

# References

Borrero, J. M., Tomczyk, S., Kubo, M., et al. 2011, , 273, 267

Collados, M., López, R., Páez, E., et al. 2012, Astronomische Nachrichten, 333, 872

Ruiz Cobo, B. & del Toro Iniesta, J. C. 1992, , 398, 375