VFISV: Very Fast Inversion of the Stokes Vector
&
FISV: Fast Inversion of the Stokes Vector

Version 1.01 for Hinode/SP: December 2007
Version 1.02 for Hinode/SP: April 2008
Version 1.03 for Hinode/SP: May 2008
Version 1.05 for Hinode/SP: November 2009

Juan Manuel Borrero
borrero@ucar.edu
borrero@mps.mpg.de

Introduction:

I started creating VFISV in late 2006 as a Post Doc at HAO in charge of developing an inversion code for the Helioseismic and Magnetic Imager (HMI/SDO). HMI will scan the Fe I line at 6173.3 A across 6 wavelength points in all four polarization states. One of my main objectives was to demonstrate that 6 wavelength points are enough to obtain an accurate measure of the magnetic field vector. Meanwhile the Japanese Solar-B/Hinode spacecraft had been launched and the data started to flow. It soon became evident that the code I was developing for HMI could very easily be modified to work with Hinode data.

After a few weeks preparing VFISV to be used with Fe I 6302.5 it is now ready to be released to the scientific community. My main purpose at this point is to make the code available for beta-testing. Consequently I am awaiting for your feedback, reports on bugs and errors that you might find in VFISV.

A few words about the usefulness of VFSIV are in order. This code was written for HMI. HMI needs to invert about 1 million pixels per minute. So you can expect the code to be much faster that other Milne-Eddington codes. As such, it is particularly suitable to be used with large datasets. However, as any Milne-Eddington code, the amount of information one can extract from the profiles is limited. If you are more interested in studying in detail a small amount of profiles I would personally recommend to use inversion codes such as SIR (Ruiz Cobo & Del Toro Iniesta), SPINIOR (Frutiger et al.) or LILIA (Socas-Navarro).

Finally, I would like to give my thanks to all those colleagues who have helped (directly or indirectly) to develop VFISV, in particular to Hector Socas-Navarro for his invaluable help with the Neural Networks.

VFISV and FISV for Hinode/SP data:

Hinode spectropolarimeter (SP) records the full Stokes vector of Fe I 6301.5 and 6302.5. Version 1.01 of VFISV/FISV is only designed to invert Fe I 6302.5. The reason is purely historical: VFISV was conceived to invert only Fe I 6173.3 for HMI. However, it turns out that this line and Fe I 6302.5 are both a normal Zeeman triplets (J=0,1 transition). Future releases of VFISV will be adapted to work with other normal Zeeman triplets : Fe I 15648.5, Ni 6768.7, Fe I 5250.2. These versions will be named after famous instruments used to obseve them TIP, MDI and Imax.

If you are interested in inverting simultaneously both Fe I 6301.5 and 6302.5, VFISV is not your code (right now at least). I would recommend to use MELANIE (Socas-Navarro) or MERLIN (Garcia & Lites). However, I have been closely involve in the development of MERLIN and my personal feeling is that, in the Milne-Eddington case, it does not make much difference to invert 1 or 2 spectral lines (unless you are in regions of very low magnetic fluxes, where having a second spectral line can be very useful).

Downloading and installing CFITSIO:

Hinode data can be found at any of the different mirrors available. It is normally found in FITS format. For this reason, VFISV uses the CFITSIO routines. Source code and installation guide can be found at the CFISTIO Home page:

http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html

I order to compile and install CFITSIO you will need either a Fortran or C compiler. If you use Linux distributions like Suse or Ubuntu you might want to install CFITSIO from the update manager or synoptic package manager. This is specially convenient since they resolve for you all dependence problems.

VFISV will not work unless you have installed the appropriate library: libcfitsio.a. The makefiles for VFISV might need to be modified to account for the location of this library. The default location is /opt/local/lib

I have tested VFISV/FISV with CFITSIO version 3.1 and earlier. I cannot guarantee that using a newer version will work.

Downloading and installing MPICH2:

VFISV uses the MPICH2 implementation of the Message Passage Interface for Parallel processing. Nowadays many Desktops and Laptops are equipped with Dual Core or Quad Core chips whose computing power can be successfully exploited with MPI. Without MPICH2 you will not be able to run VFISV (Very Fast Inversion of the Stokes Vector), but you still can run FISV (Fast Inversion of the Stokes Vector).

To download and install MPICH2 please visit the project's Home page:

http://www.mcs.anl.gov/research/projects/mpich2/

I order to compile and install MPICH2 you will need either a Fortran or C compiler. As with CFITSIO, it might be better to install MPICH2 using the package manager of your Linux distribution to avoid dependence problems.

The newest distribution of MPICH2 I have used so far is 1.0.8. Earlier versions should work as well. However, I cannot guarantee that different versions from 1.0.5p4 will work.

Downloading and installing LAPACK/BLAS:

VFISV uses LAPACK and BLAS (Linear-Algebra Package) to solve linear system of equations with the Singular Value Decomposition method. You can find these in:

http://www.netlib.org/lapack/

To install LAPACK and BLAS you will need a Fortran compiler. I have found that the installation of LAPACK becomes quite difficult (lots of flags to take care at compilation time, installation of BLAS has to be done before LAPACK, etc). Therefore, I highly recommend to use your package manager in your Linux distribution to avoid all these problems.

So far, I have tested VFISV only with version 3.2 of LAPACK. This is the newest one as of November 2009. I have not checked with earlier versions, and therefore I cannot guarantee that VFISV will run properly with them.

Downloading and compiling VFISV and FISV:

Once CFITSIO, MPICH2 and LAPACK/BLAS are installed, download the source code (currently vfisv_v1.05.tar.gz) from:

http://www.iaa.es/hinode_europe/index.php/gb/inversion_codes

or,

http://www.hao.ucar.edu/projects/csac/nextgen.php#hmi

Untar and ungzip the file. You will see several directories created:

/vfisv_v1.05/src/
This directory contains the original Fortran 77 and Fortran 90 source code as well as the makefiles.
/vfisv_v1.05/documents/
This directory contains this PDF file describing VFISV and FISV.
/vfisv_v1.05/fits_example/
This directory contains a sample of 20 fits files with Hinode data. They are sample fits from observations of AR 10953 in May 3, 2007.

/vfisv_v1.05/invert_results/
This directory is originally empty.
/vfisv_v1.05/ann_dat/
This directory contains a number of files with the ".ann" extensions. They are needed for the Neural Network initialization used by VFISV.

Now go to /src/ and choose among the following makefiles:

makefile_gen
This makefile is suitable for compiling FISV with the Intel Fortran 90 compiler "ifort". I have successfully used ifort 9.1 and ifort 10.3 both in 32 and 64 mode. If you are using this compiler rename this file as "makefile" and then type "make fisv". An executable fisv.x of approximately 4.07 MB should appear after compilation.
makefile_mpi
This makefile is suitable for compiling VFSIV using whatever Fortran 90 compiler was set up by default during the installation of MPICH2. Rename this file as "makefile" and type "make vfisv" to obtain the executable vfisv.x (about 4.65 MB) that runs in parallel in any number of CPUs.

Note that in the makefiles you will have to change the variable LIBS to point to the correct libraries (.a) for CFITSIO, BLAS, LAPACK and MPICH2.

Running VFISV and FISV:

Now you are ready to invert you test example of Hinode data. Either type:

./fisv.x

or,

mpd &
mpirun -n # ./vsifv.x (# number of cpus to run in parallel)

you will see an output looking like:

```
[borrero@localhost src]$ ./fisv.x
 SP4D20070503_105617.8C.fits
 SP4D20070503_105622.9C.fits
 SP4D20070503_105628.0C.fits
 SP4D20070503_105633.1C.fits
 SP4D20070503_105638.2C.fits
 SP4D20070503_105643.3C.fits
 SP4D20070503_105648.4C.fits
 SP4D20070503_105653.5C.fits
 SP4D20070503_105658.6C.fits
 SP4D20070503_105703.7C.fits
```

```
 SP4D20070503_105708.7C.fits
 SP4D20070503_105713.8C.fits
 SP4D20070503_105718.9C.fits
 SP4D20070503_105724.0C.fits
 SP4D20070503_105729.1C.fits
 SP4D20070503_105734.2C.fits
 SP4D20070503_105739.3C.fits
 SP4D20070503_105744.4C.fits
 SP4D20070503_105749.5C.fits
 SP4D20070503_105754.6C.fits
    52.08 sec for    20480 profiles
  393.13 profiles per second
  429 is the number of points with Chi2 > 10
    19 is the number of points with Chi2 > 25
     3.48 is the mean Chi2 achieved
     0.25 is the Best Chi2 achieved
  46.38 is the Worst Chi2 achieved
```

Each fits file corresponds to one of the 20 files in /fits_example/. Each of them corresponds to one slit position of the Hinode spectropolarimeter and contains 1024 Stokes I, Q, U and V profiles. The inversion took 52 seconds to be completed.

What has just been done can be easily understood if you edit the file map_param.f90 contained in /src/. The important parts of that module are:

WORKDIR='../fits_example/'
Directory where the fits file are located.
OUTDIR='../invert_results/'
Directory where the output of the inversion will be stored. Must exist prior to the inversion.
OUTFIL='test.dat'
File name containing the results from the inversion.
LIST='list.dat'
File containing the list of fits files to be inverted. This file must be contained in WORKDIR. Look for it there.
NFIT = 20
Number of fits files inside LIST=”list.dat” that will be inverted. In our previous example all list.dat contains 20 fits files and we inverted all of them. If changed to < 20 only the first NFIT files in “list.dat” will be inverted.
NPIX=1024
Numer of pixels along the slit. For Hinode's normal operation model NPIX is 1024. Hinode has also a fast scan mode, where NPIX = 512. Change if your data was taken in fast mode.
GLOBAL_STRAYLIGHT, LOCAL_STRAYLIGHT and STRFIL:
All these parameters will be discussed in detail in the section dedicated to straylight.

QUICKLOOK = .TRUE./.FALSE.
If .TRUE. all pixels in the Hinode observations will be subject to a full ME inversion. If .FALSE. only those pixels above a certain polarization threshold will be fully inverted. In the rest of the pixels (with low polarization levels) the magnetic field vector and LOS velocity will be obtained using a combination of Neural Networks and Weak-Field approximation. The polarization threshold can be changed in inv_param.f90 (TREPOL variable).
ERRORS = .TRUE. / .FALSE.
Set this logical to .TRUE. or .FALSE. if you want to obtain errors in the determination of the three components of the magnetic field vector and the LOS velocity.
XYFIL=''
You can set-up the name of a file that contains the X & Y coordinates of the pixels whose observed profiles and best-fit profiles will be written as an output. This will allow the user to check the fit versus observations at any location and study whether the fits are reasonable or not. The file specified in XYFIL must be located inside the WORKDIR directory described above. The format of the XYFIL must be:

x1 y1
x2 y2
x3 y3

So, for example, if x1=50 and y1=75, the code will output the observed and fitted Stokes profiles in pixel 50,75 of the image. Same for x2,y2, etc. The output will be written in the OUTDIR directory described above.

FREE PARAMETERS:

Finally, in map_param.f90 you can also select which parameters of the Milne-Eddington atmosphere are to be inverted by changing the to .TRUE. (will be inverted) to .FALSE. (will not be inverted). In our previous example we had:

```
FREE(1) = .TRUE.          ! ETA0
FREE(2) = .TRUE.          ! FIELD INCLINATION
FREE(3) = .TRUE.          ! FIELD AZIMUTH
FREE(4) = .FALSE.         ! DAMPING
FREE(5) = .TRUE.          ! DOPPLER WIDTH
FREE(6) = .TRUE.          ! FIELD STREGNTH
FREE(7) = .TRUE.          ! MAGNETIC LOS VELOCITY
FREE(8) = .TRUE.          ! SOURCE FUNCTION CONTINUUM
FREE(9) = .TRUE.          ! SOURCE FUNCTION GRADIENT
FREE(10) = .FALSE.        ! MAGNETIC FILLING FACTOR
FREE(11) = .FALSE.        ! MACROTURBULENCE
FREE(12) = .FALSE.        ! NON MAGNETIC LOS VELOCITY (Not Supported Yet)
```

Remember that any time you change something in map_param.f90 you need to recompile “make ...” for the changes to take effect.

Output data and format:

It is now time to have a look to the results. VFISV and FISV wrote them in the file “test.dat” inside the directory “invert_results/”. They are “UNFORMATTED”. The array has the following dimensions: [NFIT,NPIX,18]. In our example this is [20,1024,18]. NFIT refers to the direction perpendicular to the slit. NPIX refers to the slit's direction. The third dimension refers to:

1-to-12: Best Fit Milne-Eddington parameters in the same order than the array FREE in map_params.f90
13: chi2
14: Magnetic field strength according to the Neural Networks initialization.
15: Magnetic field inclination according to the Neural Networks initialization.
16: Magnetic field azimuth according to the Neural Networks initialization.
17: LOS velocity according to the Neural Networks initialization.
18: Filling factor according to the Neural Networks initialization.

If you selected ERRORS=.TRUE. in map_param.f90 then the elements 14-17 will include the errors in the determination of the field strength, inclination, azimuth, LOS velocity and filling factor instead of the Neural Networks initialization.

Note that the real results from the inversion are 1-to-12. 14 to 18 only stores the initial guess for the inversion process. Do as follows In order to read this data using IDL:

```
IDL>inv=dblarr(20,1024,18)
IDL>openr,1,'test.dat',/f77
IDL>readu,1,inv
IDL>close,1
IDL>tvscl,reform(inv(*,*,5)) ; for the field strength
```

INV_PARAM:

You may want to edit this file in order to change the parameters ITER, SVDTOL or GOODFIT. They control the maximum number of iterations, SVD tolerance and the Chi2 below which the inversion considers the fits is sufficiently good and stops. You may want to experiment with them a little bit. ITER is by default set to 30. This seems to be a good number of iterations. Decreasing it to 25 or 20 affects the results only slightly, but it increases the speed to the inversion of 17 and 33 % respectively.

In inv_param.f90 you can also change the value of TREPOL. This is the polarization threshold used to make a quick or full inversion whenever QUICKLOOK=.TRUE. in map_param.f90. Smaller values will imply that VFISV will invert pixels will low polarization signals.

Straylight:

VFISV and FISV include the capability to consider the effects of straylight. This is done by adding an unpolarized spectra to the total synthetic profiles. The area covered by the magnetic component is considered to be a free parameter. Therefore you need to set FREE(10) = .TRUE. in map_param.f90

Also, we need to consider STRFIL, LOCAL_STRAYLIGHT and GLOBAL_STRAYLIGHT. These variables allow you to use the following methods to account for the straylight:

Global straylight obtained from the observed FITS automatically:

In this case you need to set GLOBAL_STRAYLIGHT= .TRUE. & LOCAL_STRAYLIGHT = .FALSE. (also FREE(10) = .TRUE.). The code will go through all the available Stokes profiles in all your fits files in LIST. It will locate those points where the polarization signal is very small and it will then use the average Stokes I as the straylight profile. The same straylight is used in the inversion of all pixels.

In the example of the inversion with NFIT = 20 that we did before, we have only profiles next to a sunspot. If you use this method here you risk that the straylight profile calculated in this fashion is not really representative of the quiet Sun.

Local straylight obtained from the observed FITS automatically:

In this case you need to set LOCAL_STRAYLIGHT= .TRUE. & GLOBAL_STRAYLIGHT = .FALSE. (also FREE(10) = .TRUE.). The code will go through all the available Stokes profiles in all your fits files in LIST. It will determine a different straylight profile for each pixel by averaging the Stokes I profile of the neighboring pixels only (cube of 3x3 pixels).

Straylight given to the code in a ascii file:

To use this method you need to set: FREE(10) = .TRUE., GLOBAL_STRAYLIGHT= .FALSE. & LOCAL_STRAYLIGHT= .FALSE.. In addition, you need to set STRFIL to the name of an ascii file that contains Stokes I. This ascii file must be located in WORKDIR.

The regular distribution of VFISV and FISV includes a file called: “straylight.dat” in fits_example/. In contains the intensity values at 31 wavelength positions. This number “31” might appear random but it actually corresponds to wavelength points 62 thorugh 92 out of the 112 wavelength points in Hinode data. According to the code's internal wavelength calibration this translates into a wavelength grid that goes from -280.02 mA to +366.18 mA in steps of 21.53 mA. To test this method just use the file “straylight.dat' as STRFIL.

Straylight calculated from Look-up tables:

To use this method you need to set: FREE(10) = .TRUE., GLOBAL_STAYLIGHT= .FALSE. & LOCALSTRAYLIGHT= .FALSE.. In addition, STRFIL must be empty (='').

In this case VFISV and FISV will use the heliocentric angle of the observations (obtained from the fits files) to obtain a predetermined straylight profile from internal look-up tables. These straylight profiles where obtained by making synthesis with SIR (Ruiz Cobo & Del Toro Iniesta) using different the HSRA atmospheric model at different heliocentric angles.