FDIPS User Manual

Code Version 1.20

Gábor Tóth, Bart van der Holst, Zhenguang Huang

April 19, 2012

1 Introduction

The Finite Difference Iterative Potential field Solver (FDIPS) can be used to obtain a potential field solution from a magnetogram.

1.1 Quick Start

```
This document outlines how to install the FDIPS code

#################
# Opening FDIPS #
#################

# Open FDIPS tar ball
cd {where_you_want_to_have_the_fdips_code}
tar -xzf {path_to_file}/FDIPS_v{version_number}.tgz

#################
# Install FDIPS #
#################

# Many machines used by UofM are already recognized by the
# share/Scripts/Config.pl script which is called by the top
# level Config.pl script.
# For these platform/compiler combinations installation is very simple:
```

```
Config.pl -install

# On other platforms the Fortan compiler should be explicitly given. Type

Config.pl -compiler

# for a list of options. Note that "f95" refers to the NAG Fortran compiler,
# and "gfortran" to the GNU Fortran compiler.
# Select the appropriate compiler and install the code, for example:

Config.pl -install -compiler=ifort

# If you do not have an MPI library installed, use the -nompi flag
# for installation, for example

Config.pl -install -nompi -compiler=gfortran

# Note that only the serial FDIPS1 code can work without MPI.

####################
# Read the manuals #
####################

# The manual can be found in

doc/FDIPS.pdf

#################
# Running tests #
#################

# Try running the standard test suite by typing

make test

# For serial test only type
```

```
make test_fdips1

# Successful passing of the test is indicated by empty *.diff files.

####################
# Editing FDIPS.in #
####################

# The run directory is created by the test, but it can also be created with
make run

# The FDIPS input parameter file run/FDIPS.in can either be edited with
# a standard text editor (like emacs) or with the parameter editor GUI:
share/Scripts/ParamEditor.pl run/FDIPS.in

# This GUI is based on the PARAM.XML file that can also be read directly.
```

1.2 Preparing Input File

The magnetogram files are usually provided in FITS format. We provide IDL scripts that can read these files and convert them into a plain text files. You need IDL installed.

First rename the FITSFILE as

```
mv original_fits_file fitsfile.fits
```

Then start IDL in the same directory and run the `fits_to_ascii.pro` script in the `input` directory:

```
IDL> input/fits_to_ascii.pro
```

The script will prompt you to tell if the magnetogram grid is uniform in latitude or the cosine of the latitude (this only matters for the `fitsfile_idl.out` file). The script will generate 3 files:

```
fitsfile.H        -- header information
fitsfile_idl.out  -- ASCII file containing longitude, latitude, Br
fitsfile.dat      -- ASCII magnetogram file containing array size and Br
```

Only the last file is needed for FDIPS. It can be renamed to any name.

Note that you can create input files by other means too. The format of the ASCII magnetogram file is described at the #MAGNETOGRAM command.

1.3 Compiling FDIPS

To create a serial executable type

```
make FDIPS1
```

For larger problems, the parallel code may be preferred. This requires a working MPI library and a parallel (or at least multi-core) machine. Type

```
make FDIPS
```

in the main directory. The executables will be in the `bin/` directory.

1.4 Running FDIPS

Create a run directory with

```
make run
```

and move the ASCII magnetogram file into it. Then edit the input parameter file `run/FDIPS.in` either with a standard editor, or by running the parameter editor Graphical User Interface (GUI) with

```
share/Scripts/ParamEditor.pl run/FDIPS.in
```

You can also consult the next section or the PARAM.XML file to see how the input parameters should be set.

Once the `FDIPS.in` file is correctly setup, the serial version of FDIPS can be run as

```
./FDIPS1.exe
```

while the parallel version as

```
mpirun -np 8 FDIPS.exe
```

Note that the number of processors should match the data decomposition defined in FDIPS.in (see the description of the #PARALLEL command)

The serial code will create 1 or 2 complete output files that can be used by another code or visualized. The parallel code will produce separate files per processor. These need to be combined with the `redistribute.pl` script. For example if the domain is decomposed into 4×2 subdomains in the θ, ϕ coordinates, then the field and potential data can be collected into complete files using

```
./redistribute.pl fdips_field_np010402_000.out fdips_field.out
./redistribute.pl fdips_pot_np010204_000.out fdips_pot.out
```

The ASCII output files have the following format:

```
Radius [Rs] Longitude [Rad] Latitude [Rad] B [G]
      0  0.00000E+00  3  2  3
      21      21      20
  1.00000E+00  2.50000E+00
Radius Longitude Latitude Br Bphi Btheta Rmin Rmax
  1.0000000000E+00  0.0000000000E+00 ...
  1.0750000000E+00  0.0000000000E+00 ...
  1.1500000000E+00  0.0000000000E+00 ...
  ...
```

There are five header lines followed by the data. The first header line describes the units of the data. The second line contains timestep, time, number of dimensions, number of scalar parameters, and number of variables saved. The third line contains the grid size. The fourth line contains the scalar parameters. The fifth header line contains the names of the coordinates, variables, and scalar parameters. Each data line corresponds to one grid point. The data line contains the coordinates followed by the variables. The data lines are ordered such that the first coordinate changes fastest.

The binary (real4 and real8) output files contain the same information as the ASCII output file but in Fortran binary format. The output files are saved by the `save_plot_file` subroutine in `share/Library/src/ModPlotFile.f90`. The `read_plot_file` subroutine of the same module can be used to read the FDIPS output files from a Fortran code.

2 Input Commands for the FDIPS

List of commands used in the FDIPS.in file

2.1 All commands

2.1.1 #MAGNETOGRAM command

```
#MAGNETOGRAM
fitsfile.dat            NameFileIn
T                       UseCosTheta
3500.0                  BrMax
```

The NameFileIn parameter defines the name of the file containing the magnetogram data as a plain text file. This file can be produced from a standard FITS magnetogram file with the fits_to_ascii.pro IDL script found in the input directory.

The UseCosTheta parameter should be true if the magnetogram is on a grid with equal spacing in cos(theta). If the parameter is false, the grid spacing is uniform in the colatitude theta.

The BrMax parameter defines the largest allowed radial field magnitude in the magnetogram. If any Br is above BrMax, it will be replaced with BrMax using the sign of Br.

The magnetogram data should start with a header like this:

```
#ARRAYSIZE
 360            nLongitude
 180            nLatitude

#START
   0.526314E+00
   0.526234E+00
   ...
```

The nLongitude and nLatitude define the size of the magnetogram. The #START signals the beginning of the magnetogram data itself, i.e. the radial magnetic field in arbitrary units. The data is ordered by longitude and latitude. For a uniform-theta grid the first point is near the south pole at latitude -90 + 90/nLatitude degrees, 0 longitude, the second one is at longitude 360/nLongitude, ... and the final point is at latitude 90 - 90/nLatitude

and longitude 360 - 360/nLongitude. For the uniform in cos(theta) grid the ordering is the same, but the latitudes values are different.

Default parameter values are shown by the example.

2.1.2 #DOMAIN command

```
#DOMAIN
1.0                     rMin
2.5                     rMax
```

The parameters rMin and rMax define the radial distances of the inner and outer domain boundaries, respectively. rMax is sometimes referred to as the radius of the source surface.

Default values are shown by the example.

2.1.3 #GRID command

```
#GRID
150                     nR
180                     nThetaAll
360                     nPhiAll
```

The three parameters define the number of grid points across the whole domain in the R, Theta and Phi directions, respectively. If nThetaAll is smaller than nLatitude in the magnetogram (see the #MAGNETOGRAM command), then it is rounded so that nThetaAll' is an integer fraction of nLatitude. This means that the cells of the magnetogram can be merged together without losing magnetic flux. The same applies to nPhi and nLongitude. It is best to set these values to be either the same as nLatitude and nLongitude or an integer factor smaller.

Default values are shown by the example.

2.1.4 #PARALLEL command

```
#PARALLEL
2                       nProcTheta
3                       nProcPhi
```

This command only works in parallel FDIPS. The serial version ignores this command. The nProcTheta and nProcPhi parameters define the number of subdomains in the Theta and Phi directions, respectively. The total number of subdomains must be equal to the number of processors nProc, so the above example can only run on 6 processors.

The default domain decomposition is in the theta direction only, i.e., nProcTheta = nProc and nProcPhi = 1.

2.1.5 #TIMING command

```
#TIMING
T               UseTiming
```

Switches on or off measuring the execution time. The timing is only implemented in the parallel version. The serial version ignores this command.

In the parallel version the default is to do timing.

2.1.6 #SOLVER command

```
#SOLVER
T               UsePreconditioner
1e-10           Tolerance
```

Parameters for the iterative solver. The UsePreconditioner determines if a preconditioner is used or not. The Tolerance parameter gives the tolerance for the final residual relative to the initial residual in the L2 norm.

Default values are shown.

2.1.7 #OUTPUT command

```
#OUTPUT
field                   TypeOutput  (field or potential)
fdips_field.out         NameFileOut
ascii                   TypeFormat  (ascii, real4 or real8)

#OUTPUT
potential               TypeOutput  (field or potential)
fdips_pot.out           NameFileOut
ascii                   TypeFormat  (ascii, real4 or real8)
```

This command can occur once for each output file (at most twice). The TypeOutput parameter determines the content of the file:

- field: all magnetic field components centered on the radial faces. The coordinates and variables in this output file are Radius, Longitude, Latitude, Br, Bphi, and Btheta.

- potential: the cell centered potential and the face centered magnetic field, div B and the right-hand-side of the linear problem. The coordinates and variables in this output file are r, theta, phi, pot, br, btheta, bphi, divb, and rhs.

The TypeFormat parameter determines if the output file is saved as a text file (ascii), single precision binary file (real4) or double precisiion binary file (real8).

Default is not to save any output files, so it is a good idea to include at least one #OUTPUT command into FDIPS.in.