Manual for the
Magnetohydrostatic Field Line Extrapolation Package
(MFLEX)

Lilli Nadol
lmn6@st-andrews.ac.uk

Contents

Introduction

The following manual describes the set-up of, and how to use, the Magnetohydrostatic Field Line Extrapolation (MFLEX) package that calculates a 3D magnetic field B from given photospheric light-of-sight magnetograms.

That MFLEX package consist of a data reading, a modelling and a visualisation module. Each of these modules will be described individually later in more detail. All these codes are run in cartesian coordinate system on even, non-staggered grids.

The main part of the package, the B field modelling module, generally follows the model of Neukirch and Wiegelmann (2019) with the change of using an asymptotic approximation of the defining ordinary differential equation instead of the originally presented solution. The main differences and more details on the derivation of the used set of equations can be found in Nadol and Neukirch (2024).

The MFLEX package also includes codes for extrapolation of magnetic field lines using fieldline3D.py from pyvis from the Magnetic Skeleton Analysis Tools Package (MSAT) written by Ben Williams [see Williams PhD Thesis 2018, https://github.com/benmatwil/msat].

These codes have been written in Python 3 2023 standard. The following Python modules are used: etc etc etc

Quick Start for Solar Orbiter Archive Data

1. First you need to set-up data on your computer. Solar Orbiter Archive data files can be downloaded at https://soar.esac.esa.int/soar/#search . It is necessary to use _blos.fits files for B field line of sight observations.
2. Compile Python MFLEX packages using make. (I guess)
3. Run the .fits file reading routine read_fits_soar(path_to_blos_fits_file)
4. Until now, specify MHS model parameters. Hopefully soon, use internal parameter optimisation.
5. Run the 3D magnetic field modelling routine magnetic_field(etc etc etc)
6. Visualising the magnetic field line skeleton plot_fieldlines_grid(etc etc etc)
7. Optional: Run the Bz partial derivatives routine bz_partial_derivatives()
8. Optional: Run the plasma parameter and visualisation routine (not yet existent)

If any problems arise or for customisation please refer to the main part of the manual.

Set-up and Compilation using the Makefile

Make makefile would probably be good.

Preparing the Data

Describe file reading routines and interactive input that is needed from user.

All the codes require a 2D vector field, from now on called the background or photospheric magnetic field. The data must be stored in a .fits file with a header structured in the SOAR standard. An example of a suitable data file can be found in /data as solo_L2_phi-hrt-blos_20220307T000609_V01.fits.

Describe header reading and writing routine here.

Describe how a data set can be created from an analytical expression as seen in Von Mises distribution dipole in Neukirch and Wiegelmann (2019). Write routine for that.

Main Codes

The 3D magnetic field modelling and visualisation modules are the core of the MFLEX package. The main codes are the field component computer magnetic_field and the field line extrapolator plot_fieldlines_grid. (Wow these are all shit names.) Each are run using the following command:

```
bfield: np.ndarray[np.float64, np.dtype[np.float64]] =
magnetic_field(data_bz,z0,deltaz,a,b,alpha,xmin,xmax,ymin,ymax,zmin,zmax,nresol_
x,nresol_y,nresol_z,pixelsize_x,pixelsize_y,nf_max)

plot_fieldlines_grid(bfield,h1,hmin,hmax,eps,nresol_x,nresol_y,nresol_z,-
xmax,xmax,-ymax,ymax,zmin,zmax,a,b,alpha,nf_max)
```

They must be run in the above order or they will not work as the next code depends on the output from the previous code. The codes work in grid coordinates (a coordinate is given between 0 and nresol_x in the x direction for example). The output from each is in length scale normed real coordinates (a coordinate is given between 0.0 and 1.0 for the shorter side of the magnetogram and between 0.0 and nresol_y/nresol_x if the x direction is the shorter side for example).

Describe how choice of pixelsize_z influences results and run time.
Make choice of upper boundary (currently set at 10Mm) and transitional region (centre currently set at 2Mm) possible.

Seehafer Mirroring

Fast Fourier Transformation

Parameters

Description of parameters for each routine here. Will be endless.

Visualisation Routines

Describe how foot points are chosen and how grid spacing can be influenced. For description of fieldline3D interpolation look at Ben Williams repository.

Examples