

# Field line helicity tools

A. R. Yeates

Department of Mathematical Sciences, Durham University, UK

July 20, 2018

## Abstract

This is a set of Python tools for computing (relative) field-line helicity in Cartesian domains. Full background and motivation for the code is given in the accompanying paper [Yeates and Page, 2018].

Any questions: `anthony.yeates@durham.ac.uk`.

## 1 Overview

The main file `flhcart.py` contains the definition of a `BField` class describing a magnetic field snapshot on a regular Cartesian grid. This class has methods for interpolating, computing vector potentials, tracing magnetic field lines, computing the reference potential field, computing field-line helicity, etc. Supporting Fortran code given in the source files `fastfl.f90` and `fastflh.f90` provides the necessary fast tracing of magnetic field lines and computing of line integrals along them.

To run the code, the user must supply a netcdf file containing a datacube of three-dimensional arrays  $B_x$ ,  $B_y$ ,  $B_z$ , defined at grid points of a regular grid (not on a staggered grid). Alternatively, it would be straightforward to alter the file input to read a different format. The script `demo_b2nc.py` illustrates how to create such a netcdf file for an analytical magnetic field. Usually, however, it would be derived from a numerical experiment.

Several scripts are provided to illustrate different possible uses of the code:

- `demo_plot3d.py` – plot the 3D magnetic field lines, and compute and plot those for the corresponding potential reference field.
- `demo_flh_devore.py` – two-dimensional plots of field-line helicity on the lower boundary, using DeVore gauge.
- `demo_flh_minimal.py` – two-dimensional plots field-line helicity on the lower boundary, using minimal gauge  $\mathbf{A}^*$ .
- `demo_compare_hr.py` – compare the total relative helicity computed by volume integration (Finn-Antonsen formula) versus integration of the relative field-line helicity.

Field lines are traced by calling one of two Fortran 90 codes. The first is `fastfl.f90`, which is used when you want to return the entire field line path. The second is `fastflh.f90`, which still integrates the field lines but only returns the field-line helicity. These codes are compiled when you create a `BField` object, by default using `gfortran`. They should have no further dependencies. If OpenMP is available on your system, the code should take advantage of it automatically to trace field lines in parallel. You may need to change the Fortran compiler or compiler options on lines 30-31 of `flhcart.py`.

The Python codes should run in either Python 2.7 or 3.x.

## 2 Example output

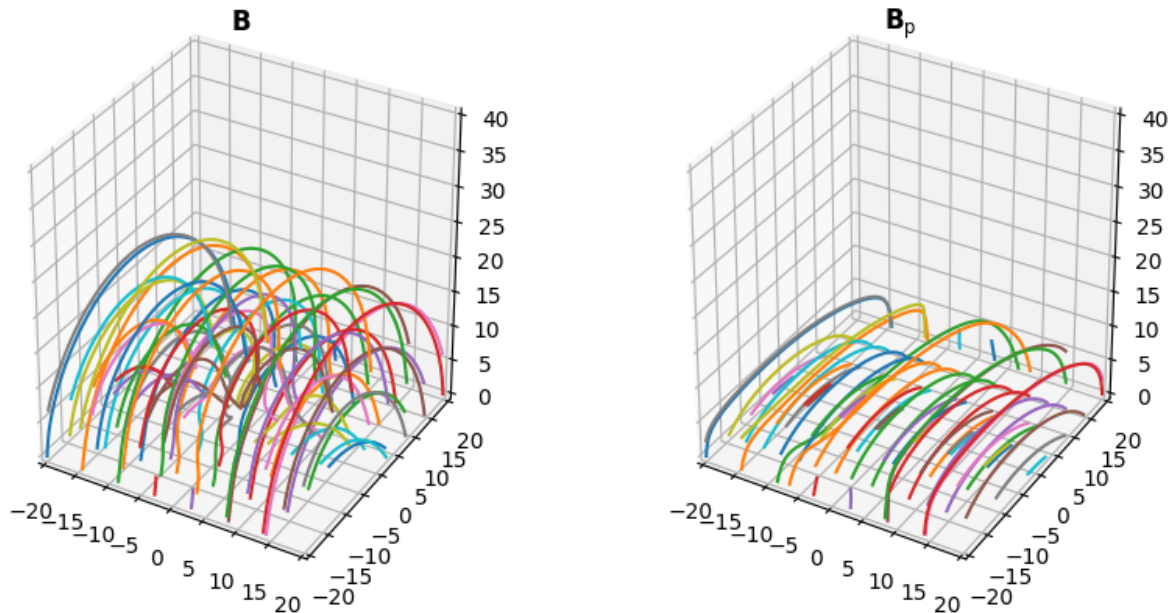
Here I show the demo output that you should obtain by running the scripts as supplied. Here `demo.b2nc.py` creates a data cube for the magnetic field

$$B_x = -2, \quad B_y = -z - \frac{t(1 - z^2)}{(1 + z^2/25)^2(1 + x^2/25)}, \quad B_z = y. \quad (1)$$

This is a simple toy model for a magnetic flux rope in the solar corona [Hesse et al., 2005].

### 2.1 Magnetic field lines

The script `demo_plot3d.py` should produce the following (rotatable) output:

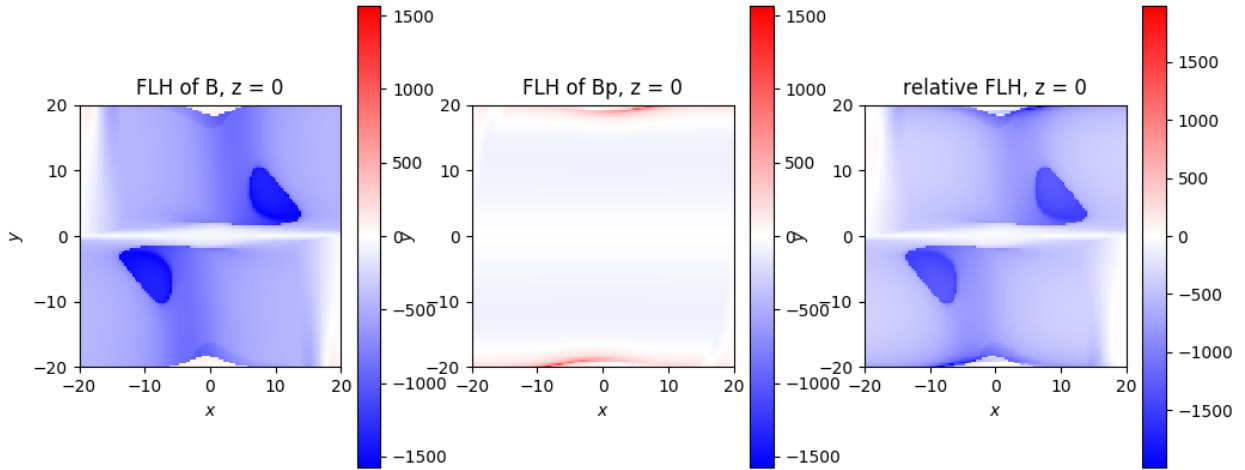


The left-hand plot shows the field lines of **B** (for  $t = 2$ ), while the right plot shows the corresponding potential reference field **B<sub>p</sub>** whose normal component  $B_{pn}$  matches  $B_n$  on all six boundaries.

Notice that, despite the fact that  $B_z$  on the lower boundary is independent of  $x$ , the field lines of  $\mathbf{B}_p$  are not orthogonal to the polarity-inversion line  $y = 0$  because the distribution of  $B_y$  on the  $y = \pm 20$  boundaries is a function of  $x$ . This illustrates the importance of matching  $B_n$  on all boundaries of the domain when computing the reference field.

## 2.2 DeVore gauge field-line helicity

The script `demo_flh_devore.py` should produce the following plots (where field lines were traced on a grid of resolution  $128 \times 128$ ):



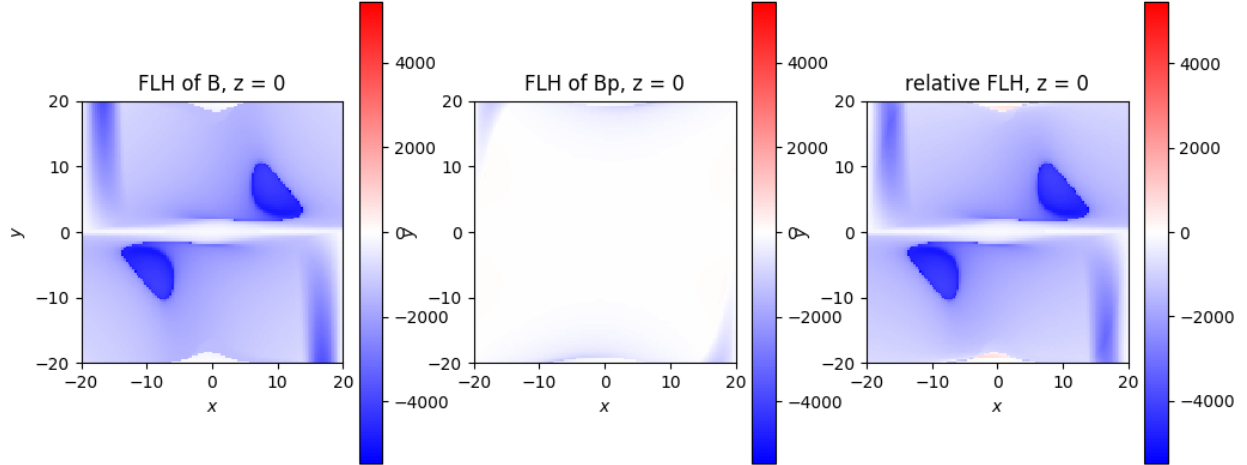
The left plot shows the field-line helicity  $\mathcal{A}(\mathbf{x}) := \int_{L(\mathbf{x})} \mathbf{A} \cdot d\mathbf{l}$  when  $\mathbf{n} \times \mathbf{A}$  is matched to  $\mathbf{n} \times \mathbf{A}_p$  on all six boundary faces, where the reference vector potential  $\mathbf{A}_p$  is in the upward-integrated DeVore-Coulomb gauge. The middle plot shows  $\mathcal{A}_p(\mathbf{x}) := \int_{L_p(\mathbf{x})} \mathbf{A}_p \cdot d\mathbf{l}$ , i.e. the field-line helicity of the reference field itself. The right plot shows the relative field line helicity  $\mathcal{A}^R(\mathbf{x}) := \mathcal{A}(\mathbf{x}) - \mathcal{A}_p(\mathbf{x})$ . The upward DeVore-Coulomb gauge is computed as

$$\mathbf{A}_p(x, y, z) = \mathbf{A}_0(x, y) + \int_{z_0}^z B_{py}(x, y, z') dz' \mathbf{e}_x - \int_{z_0}^z B_{px}(x, y, z') dz' \mathbf{e}_y, \quad (2)$$

where  $\mathbf{A}_0$  is the two-dimensional Coulomb gauge on the  $z = z_0$  (lower) boundary of the domain. All of these can easily be plotted at a different height by changing the variable  $z_0$  in the script. In this case note that  $\mathbf{x} \in \{z = z_0\}$  so  $\mathcal{A}^R$  on a given field line  $L(\mathbf{x})$  will change owing to the change of reference field line  $L(\mathbf{x}_p)$ .

## 2.3 Minimal gauge field-line helicity

The script `demo_flh_minimal.py` shows how to do the same computation but with  $\mathbf{A}_p$  and  $\mathbf{A}$  in a gauge satisfying the minimal gauge condition that  $\nabla_h \cdot \mathbf{A} = 0$  on all six boundary faces. This should produce the following plots (where field lines were traced on a grid of resolution  $128 \times 128$ ):



## 2.4 Total relative helicity

The script `demo_compare_hr.py` compares different numerical estimates of the relative helicity

$$H^R = \int_V (\mathbf{A} + \mathbf{A}_p) \cdot (\mathbf{B} - \mathbf{B}_p) d^3x, \quad (3)$$

for this datacube. Firstly, it evaluates this Finn-Antonsen formula directly using the three-dimensional composite trapezium rule with the integrand  $(\mathbf{A} + \mathbf{A}_p) \cdot (\mathbf{B} - \mathbf{B}_p)$ , for (a) the original DeVore-Coulomb gauges of  $\mathbf{A}$  and  $\mathbf{A}_p$  (with no matching), and (b) both  $\mathbf{A}$  and  $\mathbf{A}_p$  in the minimal gauge.

Then, the script evaluates  $H^R$  by first computing  $\mathcal{A}^R$  on all six boundary faces (by tracing field lines on a grid), then using the formula

$$H^R = \frac{1}{2} \oint_{\partial V} \mathcal{A}^R |B_n| d^2x. \quad (4)$$

## References

- M. Hesse, T. G. Forbes, and J. Birn. On the Relation between Reconnected Magnetic Flux and Parallel Electric Fields in the Solar Corona. *Astrophys. J.*, 631:1227–1238, Oct. 2005. doi: 10.1086/432677.
- A. R. Yeates and M. H. Page. Relative field-line helicity in bounded domains. (*in preparation*), 2018.