Surface Flux Transport Model User Manual

Code Version 1.0

Institute for Astronomy

University of Hawaii

October 16, 2023

This code is a copyright protected software.

Contents

Chapter 1

Introduction

This document describes a working prototype of the Surface Flux Transport model (SFT). The SFT was developed to provide a computationaly inexpensive solver to explore the parameter space to better understand the evolution of solar surface magnetic field evolution in response to the changes in source functions and transport parameters. In its current form the SFT is numerically stable and provides a user-friendly test-bed to understand how the properties of sunspots, advection profiles and diffusivity impact the global solar surface magnetic field evolution.

The core of the SFT are written in Fortran 90. As the equations are simplified to evolve in one dimensions, presently the code does not implement parallel communications libraries. The SFT creates a single executable which can be compiled with netCDF libraries and run on any computer as long as the GNU compiler and netCDF libraries are properly installed.

Acknowledgments

This numerical implementation of the model follows Yeates [2020]. The first version of the SWMF was developed at the Institute for Astronomy (IFA) of the University of Hawaii.

1.1 System Requirements

In order to install and run the SFT 1D the following minimum system requirements apply.

- The SFT runs only under the UNIX/Linux operating systems. This now includes Macintosh system 10.x because it is based on BSD UNIX. It does not run under any

Microsoft Windows operating system.

- A GNU FORTRAN compiler must be installed.
- The file writing subroutine use netCDF output. For these the serial/parallel version of the netCDF library has to be installed.
- In order to generate the documentation, LaTex has to be installed.

As the code solves the magnetic field evolution in only one dimension, it does not need huge computing power or higher memory load. It can run on a single processor with a nominal RAM memory.

In addition to the above requirements, the SFT output is typically visualized using Python. Other visualization packages may also be used, but the output file formats should be suported by those visualization softwares.

Chapter 2

Quick Start

2.1 A Brief Description of the SFT 1D code

The distribution in the form of the compressed tar image includes the SFT source code. The top level directory contains the following subdirectories:

- `doc` - the documentation directory
- `bipole_file` - the bipole properties to generate the source terms for the model
- `input_files` - initial magnetic field configuration for the SFT code, typically a synoptic magnetogram interpolated into the targeted sine-latitude – longitude grid
- `src` - all the fortran routines for building the executable
- `plots` - some example plots of a standard run

and the following files

- `initial_Parameters.nml` - user input of the initial parameter values for the variables
- `Makefile` - the main makefile
- `makefile.git` - makefile to build and test the code on github with continous integration (CI)
- `hmi_polar_field.p` - pickle file contaiing HMI observed polar field obtained from JSOC webpage
- `install_netcdf.sh` - shell script to download and install necessary netCDF libraries on a linux system

There are eight fortran files in the `src` directory which cotains the source code.

- `evolSFT.f90` - a short instruction on installation and usage
- `flows.f90` - description of CON parameters
- `gridSFT.f90` - to construct the grid of the solver
- `init_condition.f90` - code to set-up the initial conditions
- `output.f90` - subroutine to write the files in ASCII format
- `main.f90` - the main code
- `variables.f90` - definition of the variables to be used in the code
- `write_data.f90` - code to write the output data in netCDF format.

2.2 General Hints

Getting help with the Makefile

You can find all the possible targets that can be built by typing

```
make help
```

Compiling the code

Before going to compile the code it is essential to modify a few things in the `Makefile`. Currently, the fortran compiler and the paths of the netCDF libraries are set to this path.

```
# Set FORTRAN90 compiler
FC = gfortran

# Location of files for netcdf library
NETCDF = -I/usr/local/netcdf/include/
NETCDFLIB = -L/usr/local/netcdf/lib/ -lnetcdff
```

You need to change the paths and the compiler options according to your system. Once this is correctly specified, compile the code by typing `make`. This will create two additional sub-directories; `bin, obj`. The executable `SFT_1D` will be created in the bin directory by compiling codes from `src` director, if there are no errors in the compilation process.

Running the code

Surface flux transport model requires two major input from the user. First being the meridional flow profile and magnetic diffusivity (η). And the second being the source functions i.e., bipole properties which are added to the evolution at the time when they appear on the solar photosphere.

- Bipolar Magnetic Region (BMRs) properties are pre-written to an ASCII file and saved in `bipole_file/all_bmrs.txt`.
- Meridional flow profile is defined in the subroutine `MC_flow` contained in `flows.f90` file.
- Magnetic diffusivity (η) is defined in units of km^2/s in the `initial_Parameters.nml` file.

These parameters are passed to the code using a `namelist` file `initial_Parameters.nml`. This will create the necessary subdirectories for saving the output and execute the code based on the selected parameters. After compiling the code, type `>./bin/SFT_1D initial_Parameters.nml` to run the simulation.

There is a sample plotting file (`plot_results.py`) is also distributed with this version of the code along with some supporting files. If all the necessary python packages are available, this script will generate the butterfly diagram, comparative plot between the HMI polar field and the SFT polar field and the dipole moment. With the current set of initial conditions, the example of these plots are provided with this version of the code.

Chapter 3

The Basics

3.1 Magnetic field evolution on the solar surface

The surface flux transport (SFT) model, which solves the radial component of the magnetic field on the solar surface, has demonstrated remarkable effectiveness in simulating the dynamics of the large-scale magnetic field on the photosphere. The governing equation can be written as,

$$\frac{\partial B}{\partial t} = \frac{D}{R_\odot^2}\left[\frac{\partial}{\partial s}\left((1-s^2)\frac{\partial B}{\partial s}\right) + \frac{1}{1-s^2}\frac{\partial^2 B}{\partial \phi^2}\right] - \frac{\partial}{\partial s}\left[\frac{v_s(s)}{R_\odot}\sqrt{1-s^2}B\right] - \Omega(s)\frac{\partial B}{\partial \phi}, \tag{3.1}$$

where $s = \sin\theta$, D is the magnetic diffusivity, $\Omega(s)$ is the angular velocity in east-west direction on a sine-latitude grid, $v_s(s)$ is the flow profile along north-south direction on a sine-latitude grid and ϕ is the longitude. As the velocity profiles involved in transporting the magnetic flux on the photosphere is a function of latitude only, we can simplify this equation by taking average in the longitudinal direction which will improve the computational efficiency and provide us a lesser parameter space to comprehensively explore the dynamics. After averaging the B_r,

$$\overline{B}(s,t) = (2\pi)^{-1}\int_0^{2\pi} B(s,\phi,t)\,\mathrm{d}\phi. \tag{3.2}$$

Using this reduced form of B_r, we can re-write equation-3.1 as,

$$\frac{\partial \overline{B}}{\partial t} = \frac{\partial}{\partial s}\left[\frac{D}{R_\odot^2}(1-s^2)\frac{\partial \overline{B}}{\partial s} - \frac{v_s(s)}{R_\odot}\sqrt{1-s^2}\,\overline{B}\right], \tag{3.3}$$

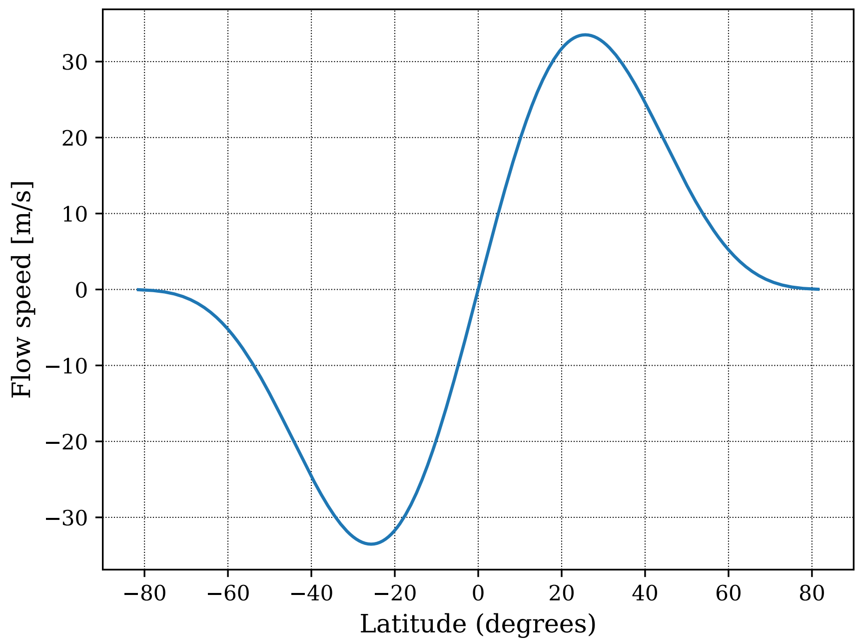

Figure 3.1: Example flow profile in the north-south direction using equation-3.4.

where the north-south velocity is,

$$v_s(s) = D_u s(1 - s^2)^{p/2}. \tag{3.4}$$

In this form the velocity, $s = \sin\theta$ and D_u controls the amplitude of the function.

Presently the code solves equation-3.3 on a sine latitude grid, figure-3.2 shows an example of the grid point distribution on a (s,ϕ) grid containing (10,20) points.

3.1.1 Bipolar approximation

We introduce a source function to model the approximating bipolar magnetic region (BMR) for an observed SHARP. The location of the center of the BMR we use the locations of the positive and negative polarity positions (s_+, ϕ_+) and (s_-, ϕ_-) on the computational grid. Here s denotes sine-latitude and ϕ denotes (Carrington) longitude. We compute,

- centroid of the BMR,

$$s_0 = \frac{1}{2}(s_+ + s_-), \qquad \phi_0 = \frac{1}{2}(\phi_+ + \phi_-) \tag{3.5}$$

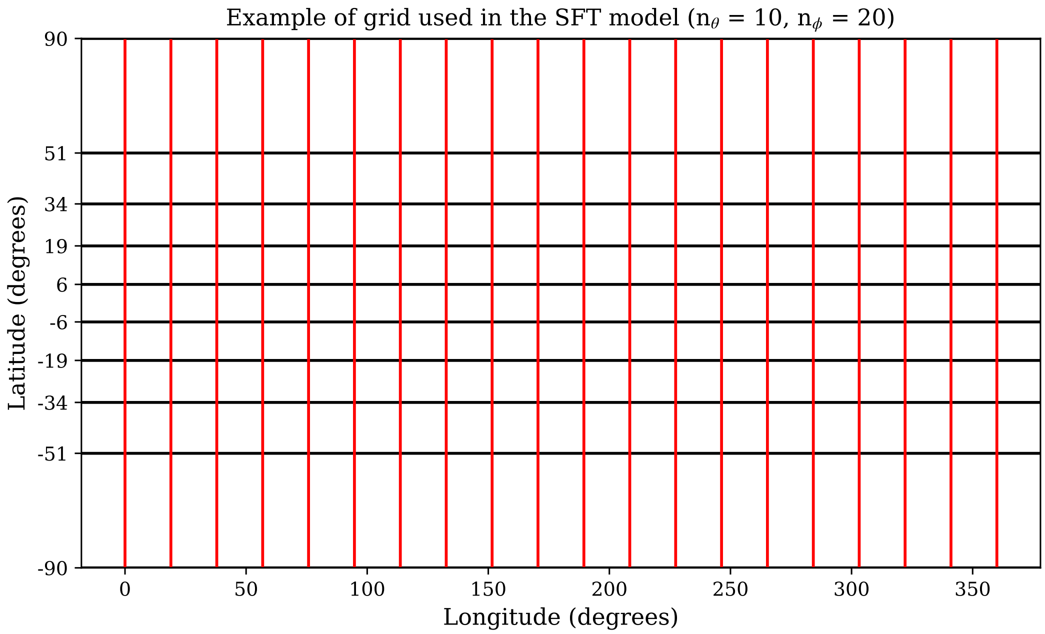

Figure 3.2: Example grid visualization with 10, 20 points in sine latitude (s) and longitude (ϕ direction respectively.

- polarity separation, which is the heliographic angle,

$$\rho = \arccos\left[s_+ s_- + \sqrt{1-s_+^2}\sqrt{1-s_-^2}\cos(\phi_+ - \phi_-)\right] \tag{3.6}$$

- the tilt angle with respect to the equator, given by,

$$\gamma = \arctan\left[\frac{\arcsin(s_+) - \arcsin(s_-)}{\sqrt{1-s_0^2}(\phi_- - \phi_+)}\right] \tag{3.7}$$

Together with the unsigned flux, $|\Phi|$, these parameters define the BMR for our chosen functional form. For an untilted BMR centered at $s = \phi = 0$, this functional form is defined as

$$B(s,\phi) = F(s,\phi) = -B_0\frac{\phi}{\rho}\exp\left[-\frac{\phi^2 + 2\arcsin^2(s)}{(a\rho)^2}\right], \tag{3.8}$$

where the amplitude B_0 is scaled to match the corrected flux of the observed region on the computational grid. To account for the location (s_0, ϕ_0) and tilt γ of a general region, we set $B(s,\phi) = F(s',\phi')$, where (s',ϕ') are spherical coordinates in a frame where the region

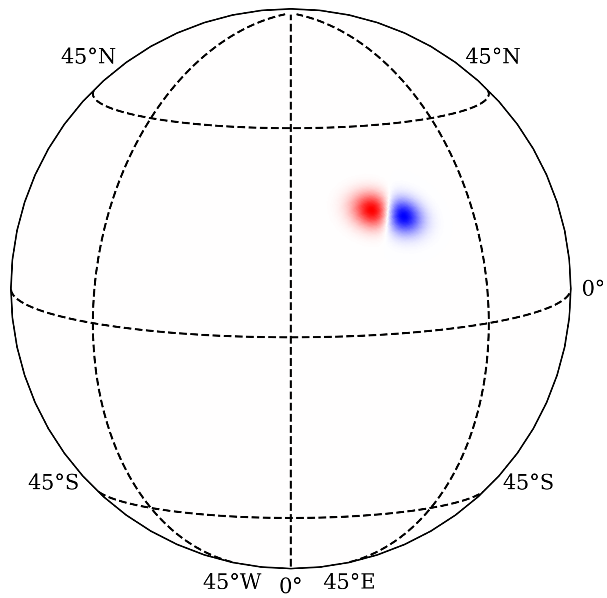

Figure 3.3: A BMR centered at 25° latitude and 22.5° longitude with a tilt of 30° with respect to the equator calculated using equation-3.8

is centered at $s' = \phi' = 0$ and untilted. Figure 3.3 shows an example BMR.

3.1.2 Coordinate transformation

The coordinate transformation from the frame (s, ϕ) where the BMR is centred at $s = \phi = 0$ to the frame (s', ϕ') where it is centred at (s_0, ϕ_0) with tilt γ. This amounts to a rotation, which is easiest to express in Cartesian coordinates

$$x = \cos\phi\sqrt{1 - s^2}, \quad y = \sin\phi\sqrt{1 - s^2}, \quad z = s. \tag{3.9}$$

Multiplying by the rotation matrices for the sequence of rotations indicated in Figure 3.4 shows that Cartesian coordinates in the rotated frame are

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} \cos\lambda_0 & 0 & \sin\lambda_0 \\ 0 & 1 & 0 \\ -\sin\lambda_0 & 0 & \cos\lambda_0 \end{bmatrix} \cdot$$
$$\cdot \begin{bmatrix} \cos\phi_0 & \sin\phi_0 & 0 \\ -\sin\phi_0 & \cos\phi_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{3.10}$$

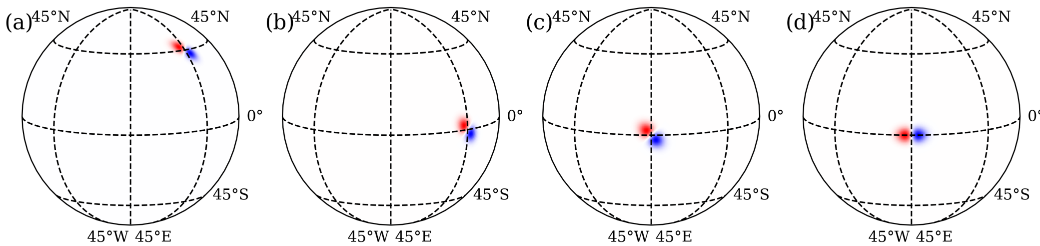

Figure 3.4: A BMR centered at 45° latitude and 45° longitude with a tilt of 45° with respect to the equator is shown in (a). Coordinate transformation operations along ϕ,λ and γ are shown in (b), (c) and (d). The corresponding transformation matris is given by equation-3.10

where $s_0 = \sin \lambda_0$. From these we determine $\phi' = \arctan(y'/x')$ and $s' = z'$.

The parameter a in equation-3.8 controls the size of the BMR relative to the separation, ρ, of the original polarity centroids. For given values of λ_0, γ, and ρ, and B_0 chosen to give the required magnetic flux, the parameter a may be chosen to control the axial dipole moment of the BMR. A good match to the axial dipole moment of the original SHARP is obtained with $a = 0.56$, and the same value works for every region.

Bibliography

Yeates, A. R. 2020, Sol. Phys., 295, 119, doi: 10.1007/s11207-020-01688-y