# Potential Field Source Surface Solver

A. R. Yeates

Department of Mathematical Sciences, Durham University, UK

October 2, 2025

## 1  Basic Equations

This code solves the for a magnetic field satisfying

$$\nabla \times \boldsymbol{B} = 0, \qquad \nabla \cdot \boldsymbol{B} = 0 \tag{1}$$

in a spherical shell $1 \leq r \leq R_{ss}$, given boundary conditions

$$B_r(\theta, \phi) = g(\theta, \phi) \quad \text{on} \quad r = 1, \tag{2}$$

$$B_\theta = B_\phi = 0 \quad \text{on} \quad r = R_{ss}. \tag{3}$$

A finite-difference method is used.

In addition, there is now the option to replace boundary condition (3) by the alternative condition

$$B_r(\theta, \phi) = \hat{g}(\theta, \phi) \quad \text{on} \quad r = R_{ss}, \tag{4}$$

although this is not usually recommended for most modelling purposes.

## 2  Code Setup

### 2.1  Files

The user needs to provide a boundary data file for $g(\theta, \phi)$.

The main script to run is then `compute_pfss.py`. This reads in the boundary data, interpolates it to the computational grid (see below), and calls the potential field solver.

The solver itself is in `pfss.py`. This outputs the resulting magnetic field $(B_r, B_\theta, B_\phi)$ in spherical polar coordinates in a netcdf file, using routines in `output_netcdf.py`. (It is also capable of outputting the vector potential, for legacy reasons.)
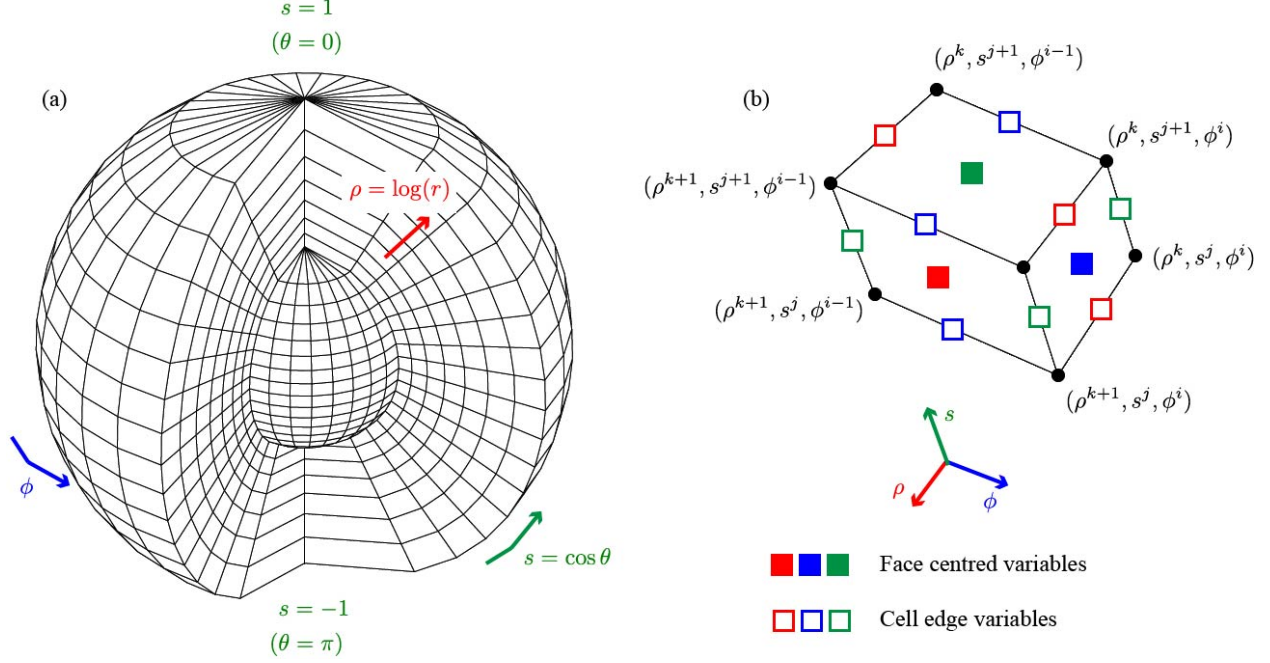
Figure 1: The numerical grid used in the solver.

## 2.2 Grid

The solver uses a rectilinear grid that is equally spaced in $\rho$, $s$, $\phi$, where

$$\rho = \ln(r), \quad s = \cos\theta$$

in terms of spherical coordinates $(r, \theta\, \phi)$. The coordinate scale factors $|d\boldsymbol{r}/d\rho|$, $|d\boldsymbol{r}/ds|$, $|d\boldsymbol{r}/d\phi|$ are

$$h_\rho = r = e^\rho, \quad h_s = \frac{r}{\sin\theta} = \frac{e^\rho}{\sqrt{1-s^2}}, \quad h_\phi = r\sin\theta = e^\rho\sqrt{1-s^2}.$$

The grid is illustrated in Figure 1. Note that the longitudinal cell size goes to zero at the poles; these points are treated specially in the calculation of $\boldsymbol{B}$. Note also that, since $s$ is a decreasing function of $\theta$, the components of a vector $\boldsymbol{v}$ in $(\rho, s, \phi)$ are $v_\rho = v_r$ but $v_s = -v_\theta$.

We define the number of grid cells $n_\rho$, $n_s$, $n_\phi$, with corresponding uniform spacings

$$\Delta_\rho = \frac{\ln(R_{ss})}{n_\rho}, \quad \Delta_s = \frac{2}{n_s}, \quad \Delta_\phi = \frac{2\pi}{n_\phi}.$$

Note that the boundaries in $s$ are at the poles $s = \pm 1$, at which points $h_\phi$ is not defined. The solution is periodic in the longitudinal ($\phi$) direction.

## 2.3 Visualization

No code is included for visualizing the output, but many options are available since the netcdf file can be read in Python, Matlab, IDL, Fortran, C etc.

Alternatively, the free software VisIt[1] can read the netcdf file directly (provided you choose `output='bg'` when writing the file). In VisIt, you need to apply a Transform from `Spherical(r, theta, phi)` to `Cartesian(x, y, z)` coordinates. Streamlines may be plotted by creating a vector mesh variable with the definition `{br, bth, bph}`.

# 3 Numerical method

## 3.1 Overall strategy

Rather than writing $\boldsymbol{B} = \nabla\chi$ and solving $\nabla^2\chi = 0$, we write instead $\boldsymbol{A} = \nabla \times (\psi\,\mathrm{e}_\rho)$. Then accounting for the unusual coordinates we get

$$\boldsymbol{B} = \frac{1}{h_\rho h_s h_\phi} \begin{vmatrix} h_\rho\,\mathrm{e}_\rho & h_\phi\,\mathrm{e}_\phi & h_s\,\mathrm{e}_s \\ \partial_\rho & \partial_\phi & \partial_s \\ 0 & \frac{h_\phi}{h_s}\partial_s\psi & -\frac{h_s}{h_\phi}\partial_\phi\psi \end{vmatrix} \tag{5}$$

$$= -\frac{1}{h_s h_\phi}\left[\partial_s\left(\frac{h_\phi}{h_s}\partial_s\psi\right) + \partial_\phi\left(\frac{h_s}{h_\phi}\partial_\phi\psi\right)\right]\mathrm{e}_\rho + \frac{1}{h_\rho h_\phi}\partial_\phi\partial_\rho\psi\,\mathrm{e}_\phi + \frac{1}{h_\rho h_s}\partial_s\partial_\rho\psi\,\mathrm{e}_s \tag{6}$$

$$= -\Delta_\perp\psi\,\mathrm{e}_\rho + \frac{1}{h_\phi}\partial_\phi\left(\frac{1}{h_\rho}\partial_\rho\psi\right)\mathrm{e}_\phi + \frac{1}{h_s}\partial_s\left(\frac{1}{h_\rho}\partial_\rho\psi\right)\mathrm{e}_s. \tag{7}$$

This will take the curl-free form $\boldsymbol{B} = \nabla\left(\frac{1}{h_\rho}\partial_\rho\psi\right)$ provided that

$$\nabla_\perp^2\psi = -\frac{1}{h_\rho}\partial_\rho\left(\frac{1}{h_\rho}\partial_\rho\psi\right), \tag{8}$$

so our strategy is to solve Equation (8) for $\psi$, then reconstruct $\boldsymbol{A}$ and $\boldsymbol{B}$. The reason for doing it this way is that it allows us to compute $\boldsymbol{A}$ as well as $\boldsymbol{B}$ (again, for legacy reasons).

## 3.2 Numerical solution

We follow the method described in van Ballegooijen et al. [2000], except that we modify the finite-difference discretisation to suit our particular coordinates.

The discretisation is chosen so that we will have $\nabla \times \boldsymbol{B} = 0$ to machine precision on a staggered grid, when the curl is taken using central differences. This property of essentially zero current density is required when using the PFSS solution to, e.g., initialize non-potential simulations. It would not typically be achieved by interpolating a spherical harmonic solution onto the numerical grid. However, we will see that the discrete solution effectively computes discrete approximations of the spherical harmonics, tailored to our particular difference formula.

In the following subsections, we describe the numerical solution in more detail.

---

[1] `https://wci.llnl.gov/simulation/computer-codes/visit`

### 3.2.1  Variables

Let the coordinate grid points be defined by

$$\rho^k = k\Delta_\rho, \qquad k = 0, \ldots, n_\rho,$$
$$s^j = j\Delta_s - 1, \qquad j = 0, \ldots, n_s,$$
$$\phi^i = i\Delta_\phi, \qquad i = 0, \ldots, n_\phi.$$

In the code the first two arrays are called `rg` and `sg` (that for `pg` is not required). There are also arrays `rc`, `sc` and `pc` corresponding to the cell centres, i.e. $\rho^{k+1/2}$, $s^{j+1/2}$ and $\phi^{i+1/2}$.

To deal with the curvilinear coordinates, we define the edge lengths

$$L_\rho^{k+1/2,j,i} = \int_{\rho^k}^{\rho^{k+1}} h_\rho \, d\rho = e^{\rho^{k+1}} - e^{\rho^k},$$

$$L_s^{k,j+1/2,i} = \int_{s^j}^{s^{j+1}} h_s \, ds = e^{\rho^k}\left( \arcsin(s^{j+1}) - \arcsin(s^j) \right),$$

$$L_\phi^{k,j,i+1/2} = \int_{\phi^i}^{\phi^{i+1}} h_\phi \, d\phi = e^{\rho^k} \sigma^j \Delta_\phi.$$

Here we used the fact that $\Delta_\rho$, $\Delta_s$ and $\Delta_\phi$ are constant, and used the shorthand

$$\sigma^j := \sqrt{1 - (s^j)^2}.$$

Similarly we define the areas of the cell faces

$$S_\rho^{k,j+1/2,i+1/2} = \int_{\phi^i}^{\phi^{i+1}} \int_{s^j}^{s^{j+1}} h_s h_\phi \, ds d\phi = e^{2\rho^k} \Delta_s \Delta_\phi,$$

$$S_s^{k+1/2,j,i+1/2} = \int_{\rho^k}^{\rho^{k+1}} \int_{\phi^i}^{\phi^{i+1}} h_\rho h_\phi \, d\phi d\rho = \tfrac{1}{2}\left( e^{2\rho^{k+1}} - e^{2\rho^k} \right) \sigma^j \Delta_\phi,$$

$$S_\phi^{k+1/2,j+1/2,i} = \int_{\rho^k}^{\rho^{k+1}} \int_{s^j}^{s^{j+1}} h_\rho h_s \, ds d\rho = \tfrac{1}{2}\left( e^{2\rho^{k+1}} - e^{2\rho^k} \right)\left( \arcsin(s^{j+1}) - \arcsin(s^j) \right).$$

In the code the face areas are stored in arrays `Sbr`, `Sbs` and `Sbp` (with only the dimensions required).

In the code the magnetic field $\boldsymbol{B}$ is defined staggered on the face centres, so $B_\rho^{k,j+1/2,i+1/2}$, $B_s^{k+1/2,j,i+1/2}$, $B_\phi^{k+1/2,j+1/2,i}$. These variables are called `br`, `bs` and `bp`.

The vector potential is located on the corresponding cell edges, so $A_\rho^{k+1/2,j,i}$, $A_s^{k,j+1/2,i+1/2}$, $A_\phi^{k,j,i+1/2}$. In fact, these values are never required on their own, only multiplied by the corresponding edge lengths. So the variables `alr`, `als` and `alp` correspond to the products $L_\rho A_\rho$, $L_s A_s$ and $L_\phi A_\phi$, respectively.

Finally, the potential $\psi$ is located on the $\rho$-faces (like $B_\rho$), so $\psi^{k,j+1/2,i+1/2}$. It is stored in the variable `psi`.

### 3.2.2 Derivatives

Firstly, we have $A = \nabla \times (\psi\,e_\rho)$. Numerically, this is approximated by

$$A_s^{k,j+1/2,i} = -\frac{\psi^{k,j+1/2,i+1/2} - \psi^{k,j+1/2,i-1/2}}{L_\phi^{k,j+1/2,i}}, \qquad A_\phi^{k,j,i+1/2} = \frac{\psi^{k,j+1/2,i+1/2} - \psi^{k,j-1/2,i+1/2}}{L_s^{k,j,i+1/2}}. \qquad (9)$$

The magnetic field $B = \nabla \times A$ is then approximated by

$$(S_\rho B_\rho)^{k,j+1/2,i+1/2} = (L_s A_s)^{k,j+1/2,i+1} - (L_s A_s)^{k,j+1/2,i} - (L_\phi A_\phi)^{k,j+1,i+1/2} + (L_\phi A_\phi)^{k,j,i+1/2}, \qquad (10)$$

$$(S_s B_s)^{k+1/2,j,i+1/2} = (L_\phi A_\phi)^{k+1,j,i+1/2} - (L_\phi A_\phi)^{k,j,i+1/2}, \qquad (11)$$

$$(S_\phi B_\phi)^{k+1/2,j+1/2,i} = -(L_s A_s)^{k+1,j+1/2,i} + (L_s A_s)^{k,j+1/2,i}. \qquad (12)$$

These formulae correspond to Stokes' Theorem applied to the cell face. The condition $\nabla \times B = 0$ may be expressed similarly as

$$0 = (L_s B_s)^{k+1/2,j,i-1/2} - (L_s B_s)^{k+1/2,j,i+1/2} - (L_\phi B_\phi)^{k+1/2,j+1/2,i} + (L_\phi B_\phi)^{k+1/2,j-1/2,i}, \qquad (13)$$

$$0 = (L_\phi B_\phi)^{k+1/2,j+1/2,i} - (L_\phi B_\phi)^{k-1/2,j+1/2,i} - (L_\rho B_\rho)^{k,j+1/2,i+1/2} + (L_\rho B_\rho)^{k,j+1/2,i-1/2}, \qquad (14)$$

$$0 = (L_\rho B_\rho)^{k,j+1/2,i+1/2} - (L_\rho B_\rho)^{k,j-1/2,i+1/2} - (L_s B_s)^{k+1/2,j,i+1/2} + (L_s B_s)^{k-1/2,j,i+1/2}. \qquad (15)$$

Note that the factors $L_\rho$, $L_s$, $L_\phi$ here are defined normal to the cell faces, not on the edges. But they have the same formulae.

In fact, condition (13) is automatically satisfied. This may be shown using equations (9) to (12), together with our formulae for $L_s$, $L_\phi$, $S_s$ and $S_\phi$.

Below, we will discretise (8) in such a way that conditions (14) and (15) are also satisfied exactly (up to rounding error).

### 3.2.3 Boundary conditions for $B$

Boundary conditions are needed when br, bs, bp are averaged to the grid points for output. We use a layer of "ghost cells", whose values are set by the following boundary conditions:

1. In $\phi$, br and bs are simply periodic.

2. At the outer boundary $\rho = \log(R_{ss})$, ghost values of bs and bp are set assuming constant gradient in $\rho$.

3. At the inner boundary, $\rho = 0$, ghost values of bs and bp are set using equations (14) and (15) (effectively assuming zero horizontal current density).

4. At the polar boundaries, the ghost value of br is set to the polemost interior value from the opposite side of the grid. Similarly, bp is set to minus the polemost interior value from the opposite side of the grid. The values of bs at the poles are not meaningful as the cell faces have zero area. However, they are set to the average of the two neighbouring interior values at that longitude (with the opposite one being reversed in sign).

Some of these conditions are chosen for compatibility with other codes, and are not necessarily the most straightforward option for a pure PFSS solver.

### 3.2.4  Discretization of Equation (8)

First, we approximate the two-dimensional Laplacian $\nabla_\perp^2 \psi$ by

$$\nabla_\perp^2 \psi^{k,j+1/2,i+1/2} = \frac{1}{S_\rho^{k,j+1/2,i+1/2}} \left[ \frac{L_s^{k,j+1/2,i+1}}{L_\phi^{k,j+1/2,i+1}} \left( \psi^{k,j+1/2,i+3/2} - \psi^{k,j+1/2,i+1/2} \right) - \frac{L_s^{k,j+1/2,i}}{L_\phi^{k,j+1/2,i}} \left( \psi^{k,j+1/2,i+1/2} - \psi^{k,j+1/2,i-1/2} \right) \right.$$

$$\left. + \frac{L_\phi^{k,j+1,i+1/2}}{L_s^{k,j+1,i+1/2}} \left( \psi^{k,j+3/2,i+1/2} - \psi^{k,j+1/2,i+1/2} \right) - \frac{L_\phi^{k,j,i+1/2}}{L_s^{k,j,i+1/2}} \left( \psi^{k,j+1/2,i+1/2} - \psi^{k,j-1/2,i+1/2} \right) \right]$$

As shorthand we define the quantities

$$U^{j+1/2} = \left( \frac{L_s}{\Delta_s \Delta_\phi L_\phi} \right)^{j+1/2}, \qquad V^j = \left( \frac{L_\phi}{\Delta_s \Delta_\phi L_s} \right)^j,$$

noting that these both depend on $j$ only. In the code these are called Uc and Vg. Then we can write our discretization as

$$\nabla_\perp^2 \psi^{k,j+1/2,i+1/2} = \frac{1}{e^{2\rho^k}} \left[ U^{j+1/2} \left( \psi^{k,j+1/2,i+3/2} + \psi^{k,j+1/2,i-1/2} \right) + V^{j+1} \psi^{k,j+3/2,i+1/2} + V^j \psi^{k,j-1/2,i+1/2} \right.$$

$$\left. - \left( 2U^{j+1/2} + V^{j+1} + V^j \right) \psi^{k,j+1/2,i+1/2} \right]. \tag{16}$$

This is the left-hand side of (8).

To discretize the right-hand side of (8), we use the approximation

$$-\frac{1}{h_\rho} \partial_\rho \left( \frac{1}{h_\rho} \partial_\rho \psi \right)^{k,j+1/2,i+1/2} = -\frac{c(\Delta_\rho)}{L_\rho^{k,j+1/2,i+1/2}} \left( \frac{\psi^{k+1,j+1/2,i+1/2} - \psi^{k,j+1/2,i+1/2}}{L_\rho^{k+1/2,j+1/2,i+1/2}} - \frac{\psi^{k,j+1/2,i+1/2} - \psi^{k-1,j+1/2,i+1/2}}{L_\rho^{k-1/2,j+1/2,i+1/2}} \right), \tag{17}$$

where

$$c(\Delta_\rho) = \frac{2 e^{\Delta_\rho/2}}{e^{\Delta_\rho} + 1} = \operatorname{sech}\left( \frac{\Delta_\rho}{2} \right).$$

Combining this with (16), we arrive at

$$U^{j+1/2} \left( \psi^{k,j+1/2,i+3/2} + \psi^{k,j+1/2,i-1/2} \right) + V^{j+1} \psi^{k,j+3/2,i+1/2} + V^j \psi^{k,j-1/2,i+1/2} - \left( 2U^{j+1/2} + V^{j+1} + V^j \right) \psi^{k,j+1/2,i+1/2}$$

$$= -\frac{c(\Delta_\rho) e^{2\rho^k}}{L_\rho^{k,j+1/2,i+1/2}} \left( \frac{\psi^{k+1,j+1/2,i+1/2} - \psi^{k,j+1/2,i+1/2}}{L_\rho^{k+1/2,j+1/2,i+1/2}} - \frac{\psi^{k,j+1/2,i+1/2} - \psi^{k-1,j+1/2,i+1/2}}{L_\rho^{k-1/2,j+1/2,i+1/2}} \right). \tag{18}$$

The reader may verify algebraically that conditions (14) and (15) follow if this finite-difference equation is satisfied.

### 3.2.5 Method of solution

Equation (18), together with the appropriate boundary conditions, yields a large (but sparse) system of $n_\rho n_s n_\phi \times n_\rho n_s n_\phi$ linear equations to solve. Fortunately, following van Ballegooijen et al. [2000], we can reduce this to a series of symmetric tridiagonal eigenvalue problems, if we look for eigenfunctions of the form

$$\psi^{k,j+1/2,i+1/2} = f^k Q_{lm}^{j+1/2}\, e^{2\pi Imi/n_\phi}. \tag{19}$$

Here the $k$ in $f^k$ is a power, not an index, and $I$ is the square root of $-1$ (since we already used $i$ and $j$ for indices). This reduction will enable very efficient solution of the linear system.

Substituting (19) in Equation (18) gives

$$-V^j Q_{lm}^{j-1/2} + \left( V^j + V^{j+1} + 4U^{j+1/2}\sin^2\left(\tfrac{\pi m}{n_\phi}\right) \right) Q_{lm}^{j+1/2} - V^{j+1} Q_{lm}^{j+3/2}$$
$$= \frac{c(\Delta_\rho)e^{2\rho^k}}{L_\rho^{k,j+1/2,i+1/2}} \left( \frac{f-1}{L_\rho^{k+1/2,j+1/2,i+1/2}} - \frac{1-f^{-1}}{L_\rho^{k-1/2,j+1/2,i+1/2}} \right) Q_{lm}^{j+1/2}. \tag{20}$$

The right-hand side can be simplified since the dependence on $k$ cancels out. This leaves the tridiagonal eigenvalue problem

$$-V^j Q_{lm}^{j-1/2} + \left( V^j + V^{j+1} + 4U^{j+1/2}\sin^2\left(\tfrac{\pi m}{n_\phi}\right) \right) Q_{lm}^{j+1/2} - V^{j+1} Q_{lm}^{j+3/2} = \lambda_{lm} Q_{lm}^{j+1/2}, \tag{21}$$

where $f$ is determined from the eigenvalues $\lambda_{lm}$ by solving the quadratic relation

$$\lambda_{lm} = \frac{c(\Delta_\rho)}{e^{\Delta_\rho/2} - e^{-\Delta_\rho/2}} \left( \frac{1-f^{-1}}{1-e^{-\Delta_\rho}} - \frac{f-1}{e^{\Delta_\rho}-1} \right). \tag{22}$$

This may be rearranged into the form

$$f^2 - \left[ 1 + e^{\Delta_\rho} + \operatorname{sech}\left(\frac{\Delta_\rho}{2}\right) \lambda_{lm}(e^{\Delta_\rho}-1)(e^{\Delta_\rho/2}-e^{-\Delta_\rho/2}) \right] f + e^{\Delta_\rho} = 0, \tag{23}$$

with two solutions for each $l, m$ given by

$$f_{lm}^+, f_{lm}^- = F_{lm} \pm \sqrt{F_{lm}^2 - e^{\Delta_\rho}}, \quad \text{where} \quad F_{lm} = \tfrac{1}{2}\left[ 1 + e^{\Delta_\rho} + \lambda_{lm}(e^{\Delta_\rho}-1)\sinh(\Delta_\rho) \right]. \tag{24}$$

In the code, the eigenvalues are called `lam` and the corresponding matrix of eigenvectors is `Q`. The solutions $f_{lm}^+$ and $f_{lm}^-$ are called `ffp` and `ffm` respectively.

The solution may then be written as a sum of these two sets of radial eigenfunctions:

$$\psi^{k,j+1/2,i+1/2} = \sum_{l=0}^{n_s-1} \sum_{m=0}^{n_\phi-1} \left[ c_{lm}(f_{lm}^+)^k + d_{lm}(f_{lm}^-)^k \right] Q_{lm}^{j+1/2}\, e^{2\pi Imi/n_\phi}. \tag{25}$$

The coefficients $c_{lm}$ and $d_{lm}$ are then determined by the radial boundary conditions:

1. At the inner boundary $\rho = 0$, where $k = 0$, we want $B_\rho = -\nabla_\perp^2 \psi$ to match our given distribution $g^{j+1/2,i+1/2}$. We have

$$B_\rho^{k,j+1/2,i+1/2} = \sum_{l=0}^{n_s-1} \sum_{m=0}^{n_\phi-1} \frac{\lambda_{lm}}{e^{2\rho^k}} \left[ c_{lm}(f_{lm}^+)^k + d_{lm}(f_{lm}^-)^k) \right] Q_{lm}^{j+1/2} e^{2\pi Imi/n_\phi},$$

so

$$g^{j+1/2,i+1/2} = \sum_{l=0}^{n_s-1} \sum_{m=0}^{n_\phi-1} \frac{\lambda_{lm}}{e^{2\rho^0}} \left[ c_{lm} + d_{lm} \right] Q_{lm}^{j+1/2} e^{2\pi Imi/n_\phi}.$$

We take the discrete Fourier transform of $g^{j+1/2,i+1/2}$ in $i$, so that (noting $e^{\rho^0} = 1$),

$$\sum_{l=0}^{n_s-1} \sum_{m=0}^{n_\phi-1} \lambda_{lm} \left[ c_{lm} + d_{lm} \right] Q_{lm}^{j+1/2} e^{2\pi Imi/n_\phi} = \sum_{m=0}^{n_\phi-1} b_m^{j+1/2} e^{2\pi Imi/n_\phi}.$$

Then the orthonormality of $Q_{lm}^{j+1/2}$ for different $l$ allows us to determine

$$c_{lm} + d_{lm} = \frac{1}{\lambda_{lm}} \sum_{j=0}^{n_s-1} b_m^{j+1/2} Q_{lm}^{j+1/2}. \tag{26}$$

2. At the source (outer) surface $\rho = \ln(R_{ss})$, where $k = n_\rho$, there are two options.

   (a) *Radial field.* We impose $\partial_\rho \psi = 0$, in the form $\psi^{n_\rho,j+1/2,i+1/2} = \psi^{n_\rho-1,j+1/2,i+1/2}$, which gives

   $$\frac{d_{lm}}{c_{lm}} = \frac{(f_{lm}^+)^{n_\rho} - (f_{lm}^+)^{n_\rho-1}}{(f_{lm}^-)^{n_\rho-1} - (f_{lm}^-)^{n_\rho}}. \tag{27}$$

   (Numerically it is better to compute this ratio the other way up, to prevent overflow.)

   (b) *Imposed $B_r$.* In this case the boundary condition is treated similarly to the inner boundary. We require

   $$\hat{g}^{j+1/2,i+1/2} = \sum_{l=0}^{n_s-1} \sum_{m=0}^{n_\phi-1} \frac{\lambda_{lm}}{e^{2\rho^{n_\rho}}} \left[ c_{lm}(f_{lm}^+)^{n_\rho} + d_{lm}(f_{lm}^-)^{n_\rho} \right] Q_{lm}^{j+1/2} e^{2\pi Imi/n_\phi},$$

   so we may again take the discrete Fourier transform to end up with

   $$c_{lm}(f_{lm}^+)^{n_\rho} + d_{lm}(f_{lm}^-)^{n_\rho} = \frac{e^{2\rho^{n_\rho}}}{\lambda_{lm}} \sum_{j=0}^{n_s-1} \hat{b}_m^{j+1/2} Q_{lm}^{j+1/2}, \tag{28}$$

Solving (27) simultaneously with either (26) or (28) gives $c_{lm}$ and $d_{lm}$. These are called `clm` and `dlm` in the code.

Remark: as we increase the grid resolution, the eigenfunctions $Q_{lm}^{j+1/2}$, which are functions of $\theta$, should converge to the corresponding associated Legendre polynomials $P_l^m(\cos\theta)$, up to normalization. This is illustrated in Figure 2.

# References

A. A. van Ballegooijen, E. R. Priest, and D. H. Mackay. Mean Field Model for the Formation of Filament Channels on the Sun. *ApJ*, 539:983–994, Aug. 2000. doi: 10.1086/309265.
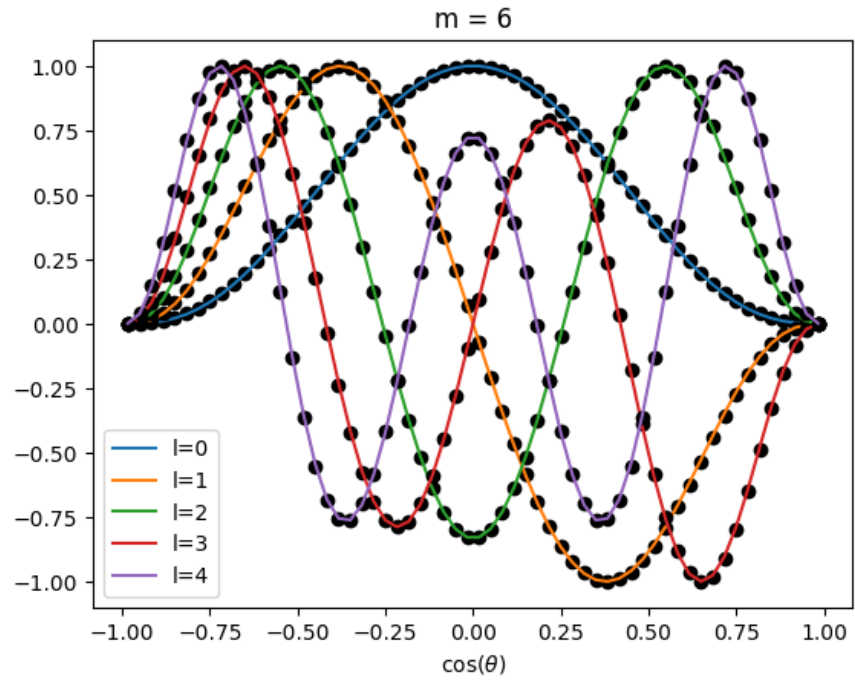
Figure 2: Comparison of $P_l^m(\cos\theta)$ (coloured lines) with the discrete eigenfunctions $Q_{lm}^{j+1/2}$ (black dots), for $m = 6$ and $l = 0, \ldots, 4$, at resolution $n_s = 60$ and $n_\phi = 120$.