

Documentation for the Method used in `amisr_fit.py`

Leslie J Lamarche

October 21, 2020

1 Objective

To create a 3D analytical model of any parameter within the FoV of an AMISR through harmonic interpolation.

2 Motivation

The Advanced Modular Incoherent Scatter Radars (AMISR) utilize electronic beam steering to image volumes of the ionosphere. However, the availability of measurements within the radar FoV is still constrained by the location of the beams for any given mode, e.g., the radar beams are still discrete and you can't measure between beams. This is an attempt to get around this problem by creating an analytic function that describes a given radar measurement anywhere within the radar field of view (FoV). In addition to creating an easy way to interpolate between beam measurements (because measurements are not on a grid, traditional interpolation is non-trivial), the function is analytic and therefore differentiable everywhere, allowing 3D gradient vectors to be calculated.

3 Approach

A truncated expansion of 3D linear basis functions are used to model AMISR parameters at all points within the FoV. The coefficients to this expansion are found with a regularized least-squares fit to the data. Interpolated densities or gradients can be found by reloading the fitted coefficients and plugging coordinates into the analytic expression. Details of the fitting process are discussed in section ...

4 Implementation

This code was written in Python 2.7.13. It requires the following packages installed:

- numpy 1.12.1
- scipy 0.19.0
- datetime 4.3
- tables 3.3.0

- matplotlib 2.0.2

Note: Version numbers given refer to the version of a package code was developed using. Program SHOULD run with other versions, but this has not been tested.

5 Model

The mathematical model used is a 3D function of z , θ , and ϕ . It is composed of a truncated expansion of linear 3D basis functions, $B_n(z, \theta, \phi)$. The coordinates z , θ , and ϕ will be expressed as a single vector, $\vec{r} = (z, \theta, \phi)$ for brevity. The spatial resolution of the model depends on the total number of basis function, N . As N increases, smaller features will be resolved, but the computation time also increases.

$$f(z, \theta, \phi) = \sum_{n=0}^N c_n B_n(\vec{r})$$

This can be expressed in matrix notation to represent an ensemble of input points and output coordinates. Assume a total of P points, $\vec{r}_0, \vec{r}_1, \dots, \vec{r}_P$.

$$\vec{c} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_N \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} B_0(\vec{r}_1) & B_1(\vec{r}_1) & \dots & B_N(\vec{r}_1) \\ B_0(\vec{r}_2) & B_1(\vec{r}_2) & \dots & B_N(\vec{r}_2) \\ \vdots & \vdots & \ddots & \vdots \\ B_0(\vec{r}_P) & B_1(\vec{r}_P) & \dots & B_N(\vec{r}_P) \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} f(\vec{r}_1) \\ f(\vec{r}_2) \\ \vdots \\ f(\vec{r}_P) \end{bmatrix} = \begin{bmatrix} B_0(\vec{r}_1) & B_1(\vec{r}_1) & \dots & B_N(\vec{r}_1) \\ B_0(\vec{r}_2) & B_1(\vec{r}_2) & \dots & B_N(\vec{r}_2) \\ \vdots & \vdots & \ddots & \vdots \\ B_0(\vec{r}_P) & B_1(\vec{r}_P) & \dots & B_N(\vec{r}_P) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_N \end{bmatrix} \quad (2)$$

$$\vec{f} = \mathbf{A} \vec{c} \quad (3)$$

This form also makes it easy to calculate gradients, ∇f . Because the coefficients c_n are constant, you only need to take the gradient of the 3D basis functions, $B_n(\vec{r})$ in the summation.

$$\nabla f(\vec{r}) = \sum_{n=0}^N c_n \nabla B_n(\vec{r})$$

Following the conventions from above, define a new matrix of the gradients of basis function.

$$\mathbf{A}_g = \begin{bmatrix} \nabla B_0(\vec{r}_1) & \nabla B_1(\vec{r}_1) & \dots & \nabla B_N(\vec{r}_1) \\ \nabla B_0(\vec{r}_2) & \nabla B_1(\vec{r}_2) & \dots & \nabla B_N(\vec{r}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \nabla B_0(\vec{r}_P) & \nabla B_1(\vec{r}_P) & \dots & \nabla B_N(\vec{r}_P) \end{bmatrix}$$

$$\vec{\nabla} f = \mathbf{A}_g \vec{c}$$

The 3D basis functions used are composed of an exponentially decaying Laguerre polynomial vertically ($e^{-z} L_k(z)$) multiplied by real-valued spherical cap harmonic functions ($Y_{\nu m}(\theta, \phi)$).

$$B_n(\vec{r}) = e^{-z} L_k(z) Y_{\nu m}(\theta, \phi) \quad (4)$$

5.1 Spherical Cap Harmonics

Spherical cap harmonics are spherical harmonics functions that have been modified to have a range of a small area (or "cap") on a sphere instead of over the entire globe. This allows higher resolution over a small area using much fewer degrees than a full spherical harmonics expansion would require. The initial use of spherical cap harmonics in geophysics is detailed in Haines (1985). Other useful sources include De Santis (1991); Thébault et al. (2006).

Spherical cap harmonics are just spherical harmonic functions where the degree (typically denoted as l) is allowed to be any positive real value. To make this distinction clear, the non-integer degree in spherical cap harmonics is denoted with ν , but l is still used to index the functions. The value of ν can be found by enforcing certain boundary conditions at the edge of the cap.

$$P_\nu^m(\cos(\theta_{cl})) = 0 \quad l - m \text{ odd} \quad (5)$$

$$\frac{dP_\nu^m(\cos(\theta))}{d\theta} \Big|_{\theta_{cl}} = 0 \quad l - m \text{ even} \quad (6)$$

These can be solved using a root-finding algorithm, but it is much more convenient to use the following approximation (Thébault et al., 2006).

$$\nu(l) \approx \left(2l + \frac{1}{2}\right) \frac{\pi}{2\theta_{cl}} - \frac{1}{2} \quad (7)$$

In total, real-valued spherical cap harmonics can be expressed as follows.

$$Y_{\nu m}(\theta, \phi) = \begin{cases} K_{\nu|m|} P_\nu^{|m|}(\cos \theta) \sin(|m|\phi) & m < 0 \\ K_{\nu m} P_\nu^m(\cos \theta) & m = 0 \\ K_{\nu m} P_\nu^m(\cos \theta) \cos(m\phi) & m > 0 \end{cases} \quad (8)$$

$$K_{\nu m} = \sqrt{\frac{2\nu + 1}{4\pi} \frac{\Gamma(\nu - m + 1)}{\Gamma(\nu + m + 1)}} \quad (9)$$

It is convenient to define a new azimuthal function, $A_{\nu m}(\phi)$, which contains both the discontinuous ϕ dependence and the constant $K_{\nu m}$. This both makes the expression look cleaner and puts it in a clearly separable form.

$$A_{\nu m}(\phi) = \begin{cases} K_{\nu|m|} \sin(|m|\phi) & m < 0 \\ K_{\nu m} & m = 0 \\ K_{\nu m} \cos(m\phi) & m > 0 \end{cases} \quad (10)$$

Real-valued spherical harmonics are used here instead of their more traditional complex counterparts to avoid the computational complications of complex numbers. Although modern computer languages like python have built-in functionality for complex numbers, maintaining a base method that uses only real numbers makes it easier to translate this code into lower-level languages, such as C, if speed ever becomes an important concern.

5.2 Laguerre Polynomials

Laguerre polynomials, $L_k(z)$, are sensitive to variations approximately in the range (0, 10). The native coordinates of the data are in meters from the center of the Earth, which corresponds

approximately to a range of $6.3 \times 10^6, 7 \times 10^6$. Essentially, the ionosphere is a thin shell over the surface of the Earth in a range that Laguerre polynomials are not sensitive to variations at all, so the data must be transformed into coordinates where the polynomials are "well behaved". If the altitude (or radial distance) of a point is originally expressed in the geocentric coordinate r , the following transformation is used.

$$z = 100 \left(\frac{r}{R_E} - 1 \right) \quad (11)$$

This essentially stretches and shifts the input geocentric coordinates to be in a range of approximately (0,10). This is a useful trick, but does introduce complications when calculating the gradients (see Section ...).

Laguerre polynomials go to infinity as z approaches infinity, so instead, weighted Laguerre polynomials, $e^{-z/2}L_k(z)$ are used. Weighted Laguerre polynomials go to zero as $z \rightarrow \infty$, which works well for electron density which is expected to physically decrease at high altitudes, but not for temperature, which actually increases. The consequence of this is that for temperature particularly, the model CANNOT be trusted at altitudes higher than the range of the data, which constrains the fit. There are some additional complications regarding the regularization as well, but these will be discussed in Section ...

5.3 3D Basis Functions

The full 3D basis functions have the following form.

$$B_n = e^{-\frac{z}{2}} L_k(z) P_\nu^m(\cos \theta) A_{\nu m}(\phi) \quad (12)$$

The basis functions are indexed by k (the radial index number), l (the longitudinal index number), and m (the azimuthal index number), where ν is a function of l . Each 3D index number, n , corresponds to one specific set of (k, l, m) . The total number of 3D basis functions is defined by the maximum number of radial indices (k_{max}) and the maximum number of longitudinal indices (l_{max}), both selected by the user.

$$N = k_{max} \times l_{max}^2 \quad (13)$$

Note that because indexing starts at 0, the actual maximum values of k and l will be $k_{max} - 1$ and $l_{max} - 1$, respectively. The value of n doesn't end up factoring into the calculated basis functions at all, so the exact method in which n maps to (k, l, m) is not important, but it MUST be consistent throughout all processes and functions. This is handled in `Model.eval_basis()` with the following convection.

$$k = \text{floor} \left(\frac{n}{l_{max}^2} \right) \quad (14)$$

$$l = \text{floor} \left(\sqrt{n \% l_{max}^2} \right) \quad (15)$$

$$m = n \% l_{max}^2 - l(l + 1) \quad (16)$$

As an example, if $k_{max} = 2$ and $l_{max} = 3$, there are a total of $N = 18$ 3D basis functions used with the index mapping displayed in Table ...

n	k	l	m
0	0	0	0
1	0	1	-1
2	0	1	0
3	0	1	1
4	0	2	-2
5	0	2	-1
6	0	2	0
7	0	2	1
8	0	2	2
9	1	0	0
10	1	1	-1
11	1	1	0
12	1	1	1
13	1	2	-2
14	1	2	-1
15	1	2	0
16	1	2	1
17	1	2	2

5.4 Coordinates

The input data should be in geocentric coordinates (r in meters, θ and ϕ in radians), but due to limitations in the modeling functions used, calculations within the model must be done in special “model” coordinates. As detailed in Section 5.2, the data must be “stretched” radially according to Equation 11. Additionally, because the Spherical Cap Harmonics (Equation 8) are centered around the north pole, the data must be rotated on the sphere to be similarly centered. This is done through the Rodriguez’ rotation formula.

$$\vec{v}_{rot} = \vec{v} \cos \theta + (\vec{k} \times \vec{v}) \sin \theta + \vec{k}(\vec{k} \cdot \vec{v})(1 - \cos \theta) \quad (17)$$

Here, \vec{v} is the original vector, \vec{k} is a unit vector describing the axis of rotation, and θ is the angle by which \vec{v} is rotated. For this application, the center of an AMISR FoV in a particular mode (the center point) is found by calculating the average colatitude (θ_{cp}) and the average azimuth (ϕ_{cp}) of the data points. Then the angle that the points must be rotated through is given by θ_{cp} and the rotation axis is in the plane of the equator at an angle of $\phi_{cp} + \pi/2$.

5.5 Gradients

The 3D basis functions are given in Equation 12. They are used to calculate the scalar value of parameters given the fitted coefficient vector \vec{c} . The gradients of the 3D basis functions are needed to calculate gradients of the parameters. The expressions are shown below, but details of their derivation are reserved for Appendix A.

$$B_n = e^{-\frac{z}{2}} L_k(z) P_\nu^m(\cos \theta) A_{\nu m}(\phi) \quad (18)$$

$$(\nabla B_n)_r = -e^{-\frac{z}{2}} \left[\frac{1}{2} L_k(z) + L_{k-1}^1(z) \right] P_\nu^m(\cos \theta) A_{\nu m}(\phi) \frac{100}{R_E} \quad (19)$$

$$(\nabla B_n)_\theta = \frac{1}{R_E(z/100 + 1) \sin \theta} e^{-\frac{z}{2}} L_k(z) [\nu \cos \theta P_\nu^m(\cos \theta) - (m + \nu) P_{\nu-1}^m(\cos \theta)] A_{\nu m}(\phi) \quad (20)$$

$$(\nabla B_n)_\phi = -\frac{m}{R_E(z/100 + 1) \sin \theta} e^{-\frac{z}{2}} L_k(z) P_\nu^m(\cos \theta) dA_{\nu m}(\phi) \quad (21)$$

The derivative azimuthal function, $dA_{\nu m}(\phi)$, is defined as follows.

$$dA_{\nu m}(\phi) = \begin{cases} K_{\nu|m|} \cos(|m|\phi) & m < 0 \\ 0 & m = 0 \\ K_{\nu m} \sin(m\phi) & m > 0 \end{cases} \quad (22)$$

6 Fit

The model is fit to a particular data set by finding the coefficient vector \vec{c} that minimizes the square of the residuals. This is a standard least-squares optimization problem.

$$\min \left\{ \sum_i \left(\frac{d_i - f(\vec{r}_i)}{\sigma_i} \right)^2 \right\} \quad (23)$$

Here, \vec{r}_i , d_i , and σ_i are the position, value, and error of a given point in the data and $f(\vec{r}_i)$ is the model evaluated at that point. It can be shown that equation 23 is equivalent to finding the least-squares solution to the following matrix equation.

$$\mathbf{A}^\dagger \mathbf{W} \mathbf{A} \vec{c} = \mathbf{A}^\dagger \mathbf{W} \vec{d} \quad (24)$$

Here, \vec{d} is the vector of data points and \mathbf{W} is the matrix of the inverse of their errors squared.

$$\vec{d} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_P \end{bmatrix} \quad \mathbf{W} = \begin{bmatrix} 1/\sigma_1^2 & 0 & \dots & 0 \\ 0 & 1/\sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/\sigma_P^2 \end{bmatrix} \quad (25)$$

\mathbf{A} and \vec{c} have been defined previously in Equation 1.

The solution to Equation 24 can be expressed in closed form by multiplying both sides by the inverse of $\mathbf{A}^\dagger \mathbf{W} \mathbf{A}$.

$$\vec{c} = (\mathbf{A}^\dagger \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^\dagger \mathbf{W} \vec{d} \quad (26)$$

However, numerically inverting matrices is usually a bad idea (<https://www.johndcook.com/blog/2010/01/19/dont-invert-that-matrix/>), so instead, solve Equation 24 directly with a standard linear algebra equation solver (e.g. `scipy.linalg.lstsq`).

6.1 Regularization

Regularization refers to a variety of techniques that are used to try to ensure that the model gives a reasonable estimation for reality even when the data coverage is sparse. Typically, it involves making some kind of assumption about the system and adding a term to the residual equation so

that solutions that are not very close to that assumption are penalized. This modifies Equation 23.

$$\min \left\{ \sum_i \left(\frac{d_i - f(\vec{r}_i)}{\sigma_i} \right)^2 + \alpha \Gamma(\vec{r}_i) \right\} \quad (27)$$

Here, Γ is some function that defines the regularization and α is a constant, the regularization parameter, that determines how heavily the model is regularized. This is useful for preventing things like over-fitting, where the model hits almost all the data exactly by oscillating wildly between points. Two types of regularization are utilized here: curvature and zeroth-order.

6.2 Curvature

Curvature regularization is a relatively common technique, which attempts to minimize the curvature of the output model to reduce the number of sharp direction changes, which is often a symptom of over-fitting. Instead of computing the actual curvature, usually just the second derivative squared (or in the case of a 3D function, the Laplacian squared) is used.

$$\Gamma = \int \nabla^2 f(\vec{r}) d\vec{r} \quad (28)$$

In matrix notation, this is equivalent to the following term, which is added to the left hand side of Equation 24.

$$\Gamma = \mathbf{\Omega} \vec{c} \quad (29)$$

$\mathbf{\Omega}$ is a matrix of the integral of the product of the Laplacian of two basis functions.

$$\mathbf{\Omega} = \begin{bmatrix} \int (\nabla^2 B_0(\vec{r}))^2 d\vec{r} & \int \nabla^2 B_0(\vec{r}) \nabla^2 B_1(\vec{r}) d\vec{r} & \dots & \int \nabla^2 B_0(\vec{r}) \nabla^2 B_N(\vec{r}) d\vec{r} \\ \int \nabla^2 B_1(\vec{r}) \nabla^2 B_0(\vec{r}) d\vec{r} & \int (\nabla^2 B_1(\vec{r}))^2 d\vec{r} & \dots & \int \nabla^2 B_1(\vec{r}) \nabla^2 B_N(\vec{r}) d\vec{r} \\ \vdots & \vdots & \ddots & \vdots \\ \int \nabla^2 B_N(\vec{r}) \nabla^2 B_0(\vec{r}) d\vec{r} & \int \nabla^2 B_N(\vec{r}) \nabla^2 B_1(\vec{r}) d\vec{r} & \dots & \int (\nabla^2 B_N(\vec{r}))^2 d\vec{r} \end{bmatrix} \quad (30)$$

The Laplacian of the 3D basis function is computed in Appendix B.

$$\nabla^2 B_n(\vec{r}) = \left(\frac{100}{R_E} \right)^2 \frac{1}{r} \left[2 \left(\frac{R_E}{100} \right) \frac{\partial Z}{\partial z} + r \frac{\partial^2 Z}{\partial z^2} \right] P_\nu^m(\cos \theta) A_{\nu m}(\phi) \quad (31)$$

$$+ e^{-\frac{z}{2}} L_k(z) \frac{1}{r^2} \Theta(\theta) A_{\nu m}(\phi) \quad (32)$$

$$+ e^{-\frac{z}{2}} L_k(z) \frac{-m^2}{r^2 \sin^2 \theta} P_\nu^m(\cos \theta) A_{\nu m}(\phi) \quad (33)$$

$$\frac{\partial Z}{\partial z} = -e^{-\frac{z}{2}} L_k(z) \left[\frac{1}{2} L_k(z) + L_{k-1}^1(z) \right] \quad (34)$$

$$\frac{\partial^2 Z}{\partial z^2} = e^{-\frac{z}{2}} L_k(z) \left[\frac{1}{4} L_k(z) + L_{k-1}^1(z) + L_{k-2}^2(z) \right] \quad (35)$$

$$\Theta(\theta) = \frac{1}{\sin^2 \theta} \left[(m^2 - \nu^2 (\cos^2 \theta + 1) - \nu) P_\nu^m(\cos \theta) + \nu(\nu + m) \cos \theta P_{\nu-1}^m(\cos \theta) + \nu(\nu - m + 1) \cos \theta P_{\nu+1}^m(\cos \theta) \right] \quad (36)$$

If Equation 31 isn't unwieldy enough, each element of $\mathbf{\Omega}$ is the volumetric integral of the product of two 3D basis function Laplacians. The resulting integrand is extremely messy and can only be solved numerically. However, due to the number of coupled terms including special functions, the required triple integral is actually quite computationally expensive.

Instead of regularizing with the computationally expensive full Laplacian, use a modified “perpendicular” Laplacian. This is very similar to the standard Laplacian in spherical coordinates, but set all radial derivatives to zero. The assumption that $\partial/\partial r = 0$ is meant less to indicate that no gradients are expected in the radial direction (quite the contrary, this is the ONLY direction in which variation is expected at all times) but more that we are not concerned about over-fitting in this direction. Furthermore, the zeroth-order regularization should take care of smoothing the model radially.

$$\nabla_{\perp}^2 B_n(\vec{r}) = \frac{e^{-\frac{z}{2}}}{z^2} L_k(z) \Theta'_{\nu m}(\theta) A_{\nu m}(\phi) \quad (37)$$

$$\Theta'_{\nu m}(\theta) = \frac{1}{\sin^2 \theta} \left[-\nu(\nu \cos^2 \theta + \nu + 1) P_{\nu}^m(\cos \theta) + \nu(\nu + m) \cos \theta P_{\nu-1}^m(\cos \theta) + \nu(\nu - m + 1) \cos \theta P_{\nu+1}^m(\cos \theta) \right] \quad (38)$$

The details of this derivation are also included in Appendix B.

Each element in the curvature regularization matrix $\mathbf{\Omega}$ (Equation 30) consists of the integral of the product of the Laplacians of two independent basis functions.

$$\Omega_{ij} = \int \nabla_{\perp}^2 B_i(\vec{r}) \nabla_{\perp}^2 B_j(\vec{r}) d\vec{r} \quad (39)$$

$$= \int \left[\frac{e^{-\frac{z}{2}}}{z^2} L_{k_i}(z) \Theta'_{\nu_i m_i}(\theta) A_{\nu_i m_i}(\phi) \right] \left[\frac{e^{-\frac{z}{2}}}{z^2} L_{k_j}(z) \Theta'_{\nu_j m_j}(\theta) A_{\nu_j m_j}(\phi) \right] z^2 \sin \theta dz d\theta d\phi \quad (40)$$

$$= \int \frac{e^{-z}}{z^2} L_{k_i}(z) L_{k_j}(z) dz \int \Theta'_{\nu_i m_i}(\theta) \Theta'_{\nu_j m_j}(\theta) \sin \theta d\theta \int A_{\nu_i m_i}(\phi) A_{\nu_j m_j}(\phi) d\phi \quad (41)$$

The fact that this can separate into three single integrals instead of one triple integral is actually a huge computational advantage.

6.3 Zeroth Order

In zeroth order regularization, minimize the difference between the output model and a set function that represents a very basic, “zeroth order” approximation of what the output model is expected to be. This is essentially making an au priori assumption about what the output model SHOULD look like, and then allowing perturbations to that model to fit the data. In the ionosphere, where there are often established, first-principle based approximate models of various parameters, this kind of regularization is particularly appropriate.

$$\Gamma = \int (f(\vec{r}) - \delta(\vec{r}))^2 d\vec{r} \quad (42)$$

The function $\delta(\vec{r})$ represents whatever zeroth order model is appropriate for a given parameter. Although these can be 3D functions, horizontal variation is difficult to guess a priori (hence the need for a data based fit), so simple models of the profile in altitude are perfectly appropriate. For electron density, a standard Chapman function is chosen (Equation 43) while an inverse hyperbolic sine function represents the general shape of the temperature profile (Equation 44).

$$\delta(\vec{r}) = A e^{1 - \frac{z-B}{H} - e^{-\frac{z-B}{H}}} \quad (43)$$

$$\delta(\vec{r}) = A \sinh^{-1} \left(\frac{x - B}{H} \right) + C \quad (44)$$

The constants in Equations 43 and 44 are found by doing a non-linear least-squares fit to the raw data.

In matrix form, Equation 42 is equivalent to the following, which is added to the left-hand side of Equation 24.

$$\Gamma = \Psi \vec{c} - \vec{\Upsilon} \quad (45)$$

Ψ is a matrix composed of the integral of the product of two basis functions.

$$\Psi = \begin{bmatrix} \int (B_0(\vec{r}))^2 d\vec{r} & \int B_0(\vec{r}) B_1(\vec{r}) d\vec{r} & \dots & \int B_0(\vec{r}) B_N(\vec{r}) d\vec{r} \\ \int B_1(\vec{r}) B_0(\vec{r}) d\vec{r} & \int (B_1(\vec{r}))^2 d\vec{r} & \dots & \int B_1(\vec{r}) B_N(\vec{r}) d\vec{r} \\ \vdots & \vdots & \ddots & \vdots \\ \int B_N(\vec{r}) B_0(\vec{r}) d\vec{r} & \int B_N(\vec{r}) B_1(\vec{r}) d\vec{r} & \dots & \int (B_N(\vec{r}))^2 d\vec{r} \end{bmatrix} \quad (46)$$

$\vec{\Upsilon}$ is a vector composed of the integral of the product of a basis function and the zeroth order function, $\delta(\vec{r})$.

$$\vec{\Upsilon} = \begin{bmatrix} \int B_0(\vec{r}) \delta(\vec{r}) d\vec{r} \\ \int B_1(\vec{r}) \delta(\vec{r}) d\vec{r} \\ \vdots \\ \int B_N(\vec{r}) \delta(\vec{r}) d\vec{r} \end{bmatrix} \quad (47)$$

Each element in the matrix Ψ (Equation 46) consists of the integral of two independent basis functions.

$$\Psi_{ij} = \int B_i(\vec{r}) B_j(\vec{r}) d\vec{r} \quad (48)$$

$$= \int \left[e^{-\frac{z}{2}} L_{k_i}(z) P_{\nu_i}^{m_i}(\cos \theta) A_{\nu_i m_i}(\phi) \right] \left[e^{-\frac{z}{2}} L_{k_j}(z) P_{\nu_j}^{m_j}(\cos \theta) A_{\nu_j m_j}(\phi) \right] z^2 \sin \theta dz d\theta d\phi \quad (49)$$

$$= \int e^{-z} L_{k_i}(z) L_{k_j}(z) z^2 dz \int P_{\nu_i}^{m_i}(\cos \theta) P_{\nu_j}^{m_j}(\cos \theta) \sin \theta d\theta \int A_{\nu_i m_i}(\phi) A_{\nu_j m_j}(\phi) d\phi \quad (50)$$

Similarly, if $\delta(\vec{r})$ is separable or a 1D f each element in $\vec{\Upsilon}$ can also be expressed as the product of 1D integrals.

$$\Upsilon_n = \int B_n(\vec{r}) \delta(\vec{r}) d\vec{r} \quad (51)$$

$$= \int \left[e^{-\frac{z}{2}} L_k(z) P_{\nu}^m(\cos \theta) A_{\nu m}(\phi) \right] \delta(z) z^2 \sin \theta dz d\theta d\phi \quad (52)$$

$$= \int e^{-\frac{z}{2}} L_k(z) \delta(z) z^2 dz \int P_{\nu}^m(\cos \theta) \sin \theta d\theta \int A_{\nu m}(\phi) d\phi \quad (53)$$

Similar to the curvature regularization case, the fact that both Ψ and $\vec{\Upsilon}$ can be calculated as a product of 1D integrals is computationally advantageous.

A Derivation of Gradients of 3D Basis Function

The 3D basis function has the following form (Equation 12)

$$B_n = e^{-\frac{z}{2}} L_k(z) P_{\nu}^m(\cos \theta) A_{\nu m}(\phi) \quad (54)$$

where

$$z = 100 \left(\frac{r}{R_E} - 1 \right) \quad (55)$$

. Using the standard expression for the gradient in a spherical coordinate system,

$$(\nabla B_n)_r = \frac{\partial B_n}{\partial r} = \frac{\partial B_n}{\partial z} \frac{\partial z}{\partial r} \quad (56)$$

$$(\nabla B_n)_\theta = \frac{1}{r} \frac{\partial B_n}{\partial \theta} \quad (57)$$

$$(\nabla B_n)_\phi = \frac{1}{r \sin \theta} \frac{\partial B_n}{\partial \phi} \quad (58)$$

The scaling factor and r can be expressed in terms of the model coordinate z .

$$r = R_E \left(\frac{z}{100} + 1 \right) \quad \frac{\partial z}{\partial r} = \frac{100}{R_E} \quad (59)$$

The radial component requires calculation of the derivative $\partial B_n / \partial z$.

$$\frac{\partial B_n}{\partial z} = \frac{\partial}{\partial z} \left[e^{-\frac{z}{2}} L_k(z) \right] P_\nu^m(\cos \theta) A_{\nu m}(\phi) \quad (60)$$

$$= \left[\left(-\frac{1}{2} \right) e^{-\frac{z}{2}} L_k(z) + e^{-\frac{z}{2}} \left(-L_{k-1}^1(z) \right) \right] P_\nu^m(\cos \theta) A_{\nu m}(\phi) \quad (61)$$

$$= -e^{-\frac{z}{2}} \left[\frac{1}{2} L_k(z) + L_{k-1}^1(z) \right] P_\nu^m(\cos \theta) A_{\nu m}(\phi) \quad (62)$$

$$(\nabla B_n)_r = -e^{-\frac{z}{2}} \left[\frac{1}{2} L_k(z) + L_{k-1}^1(z) \right] P_\nu^m(\cos \theta) A_{\nu m}(\phi) \frac{100}{R_E} \quad (63)$$

The latitudinal component requires the calculation of $\partial B_n / \partial \theta$.

$$\frac{\partial B_n}{\partial \theta} = e^{-\frac{z}{2}} L_k(z) \frac{\partial}{\partial \theta} \left[P_\nu^m(\cos \theta) \right] A_{\nu m}(\phi) \quad (64)$$

$$= e^{-\frac{z}{2}} L_k(z) \frac{1}{\sin \theta} \left[\nu \cos \theta P_\nu^m(\cos \theta) - (\nu + m) P_{\nu-1}^m(\cos \theta) \right] A_{\nu m}(\phi) \quad (65)$$

$$(\nabla B_n)_\theta = \frac{1}{R_E(z/100 + 1) \sin \theta} e^{-\frac{z}{2}} L_k(z) \left[\nu \cos \theta P_\nu^m(\cos \theta) - (m + \nu) P_{\nu-1}^m(\cos \theta) \right] A_{\nu m}(\phi) \quad (66)$$

The azimuthal component requires the calculation of $\partial B_n / \partial \phi$.

$$\frac{\partial B_n}{\partial \phi} = e^{-\frac{z}{2}} L_k(z) P_\nu^m(\cos \theta) \frac{\partial}{\partial \phi} \left[A_{\nu m}(\phi) \right] \quad (67)$$

$A_{\nu m}(\phi)$ is a piece-wise function defined in Equation 10

$$\frac{\partial}{\partial \phi} [A_{\nu m}(\phi)] = \begin{cases} \frac{\partial}{\partial \phi} [K_{\nu|m|} \sin(|m|\phi)] & m < 0 \\ \frac{\partial}{\partial \phi} [K_{\nu m}] & m = 0 \\ \frac{\partial}{\partial \phi} [K_{\nu m} \cos(m\phi)] & m > 0 \end{cases} \quad (68)$$

$$= \begin{cases} |m| K_{\nu|m|} \cos(|m|\phi) & m < 0 \\ 0 & m = 0 \\ -m K_{\nu m} \sin(m\phi) & m > 0 \end{cases} \quad (69)$$

$$= -m \begin{cases} K_{\nu|m|} \cos(|m|\phi) & m < 0 \\ 0 & m = 0 \\ K_{\nu m} \sin(m\phi) & m > 0 \end{cases} \quad (70)$$

Using the definition for $dA_{\nu m}(\phi)$ given in Equation 22, $\partial B_n / \partial \phi$ can be expressed in a much cleaner form.

$$\frac{\partial B_n}{\partial \phi} = -m e^{-\frac{z}{2}} L_k(z) P_\nu^m(\cos \theta) dA_{\nu m}(\phi) \quad (71)$$

$$(\nabla B_n)_\phi = -\frac{m}{R_E(z/100 + 1) \sin \theta} e^{-\frac{z}{2}} L_k(z) P_\nu^m(\cos \theta) dA_{\nu m}(\phi) \quad (72)$$

B Derivation of the Laplacian of 3D Basis Function

B.1 Full Laplacian

The Laplacian in spherical coordinates has the following general form.

$$\nabla^2 f = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial f}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial f}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 f}{\partial \phi^2} \quad (73)$$

The 3D basis function has the following form (Equation 12)

$$B_n = e^{-\frac{z}{2}} L_k(z) P_\nu^m(\cos \theta) A_{\nu m}(\phi) \quad (74)$$

where

$$z = 100 \left(\frac{r}{R_E} - 1 \right) \quad (75)$$

. In taking the Laplacian of a single basis function, the chain rule must be employed due to the z transformation.

$$\frac{\partial B_n}{\partial r} = \frac{\partial B_n}{\partial z} \frac{\partial z}{\partial r} \quad (76)$$

Apply this to the first term in the general Laplacian equation.

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial B_n}{\partial r} \right) = \frac{\partial}{\partial z} \left(r^2 \frac{\partial B_n}{\partial z} \frac{\partial z}{\partial r} \right) \frac{\partial z}{\partial r} \frac{1}{r} \quad (77)$$

$$= \left[2r \frac{\partial r}{\partial z} \left(\frac{\partial B_n}{\partial z} \frac{\partial z}{\partial r} \right) + r^2 \frac{\partial^2 B_n}{\partial z^2} \frac{\partial z}{\partial r} \right] \frac{\partial z}{\partial r} \frac{1}{r} \quad (78)$$

$$= \left(\frac{\partial z}{\partial r} \right)^2 \frac{1}{r} \left[2 \frac{\partial r}{\partial z} \frac{\partial B_n}{\partial z} + r \frac{\partial^2 B_n}{\partial z^2} \right] \quad (79)$$

Calculate the first and second derivatives of the basis function.

$$\frac{\partial B_n}{\partial z} = -e^{-\frac{z}{2}} \left[\frac{1}{2} L_k(z) + L_{k-1}^1(z) \right] P_\nu^m(\cos \theta) A_{\nu m}(\phi) \quad (80)$$

$$\frac{\partial^2 B_n}{\partial z^2} = e^{-\frac{z}{2}} \left[\frac{1}{4} L_k(z) + L_{k-1}^1(z) + L_{k-2}^2(z) \right] P_\nu^m(\cos \theta) A_{\nu m}(\phi) \quad (81)$$

For simplicity, define the function $Z(z)$ such that

$$\frac{\partial Z}{\partial z} = -e^{-\frac{z}{2}} \left[\frac{1}{2} L_k(z) + L_{k-1}^1(z) \right] \quad (82)$$

$$\frac{\partial^2 Z}{\partial z^2} = e^{-\frac{z}{2}} \left[\frac{1}{4} L_k(z) + L_{k-1}^1(z) + L_{k-2}^2(z) \right] \quad (83)$$

Substitute the expressions for the basis function derivatives back into the first term of the Laplacian.

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial B_n}{\partial r} \right) = \left(\frac{100}{R_E} \right)^2 \frac{1}{r} \left[2 \left(\frac{R_E}{100} \right) \frac{\partial Z}{\partial z} + r \frac{\partial^2 Z}{\partial z^2} \right] P_\nu^m(\cos \theta) A_{\nu m}(\phi) \quad (84)$$

Consider the second term in the Laplacian

$$\frac{\partial B_n}{\partial \theta} = e^{-\frac{z}{2}} L_k(z) \frac{\partial P_\nu^m(\cos \theta)}{\partial \theta} A_{\nu m}(\phi) \quad (85)$$

To evaluate the derivative of the Legendre polynomial, first make the substitution $x = \cos \theta$ such that $\frac{\partial x}{\partial \theta} = -\sin \theta$ and $\frac{\partial P_\nu^m}{\partial \theta} = \frac{\partial P_\nu^m}{\partial x} \frac{\partial x}{\partial \theta} = -\sin \theta \frac{\partial P_\nu^m}{\partial x}$ and then evaluate $\frac{\partial P_\nu^m}{\partial x}$ using the standard properties of Legendre polynomials.

$$\frac{\partial P_\nu^m}{\partial x} = \frac{1}{x^2 - 1} (x \nu P_\nu^m(x) - (\nu + m) P_{\nu-1}^m(x)) \quad (86)$$

$$\frac{\partial P_\nu^m}{\partial \theta} = \frac{1}{\sin \theta} [\nu \cos \theta P_\nu^m(\cos \theta) - (\nu + m) P_{\nu-1}^m(\cos \theta)] \quad (87)$$

Folding this result into the entire second term in the Laplacian yields the following.

$$\frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial B_n}{\partial \theta} \right) = e^{-\frac{z}{2}} L_k(z) \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial P_\nu^m}{\partial \theta} \right) A_{\nu m}(\phi) \quad (88)$$

Consider only the derivatives in θ .

$$\frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial P_\nu^m}{\partial \theta} \right) = \frac{\partial}{\partial \theta} [\nu \cos \theta P_\nu^m(\cos \theta) - (\nu + m) P_{\nu-1}^m(\cos \theta)] \quad (89)$$

$$= \frac{\partial}{\partial x} [\nu x P_\nu^m(x) - (\nu + m) P_{\nu-1}^m(x)] \frac{\partial x}{\partial \theta} \quad (90)$$

$$= \frac{1}{x^2 - 1} [- (\nu^2(x^2 + 1) + \nu - m^2) P_\nu^m(x) + \nu x(\nu + m) P_{\nu-1}^m(x) + \nu x(\nu - m + 1) P_{\nu+1}^m(x)] \frac{\partial x}{\partial \theta} \quad (91)$$

$$= \frac{1}{\sin \theta} [(m^2 - \nu^2(\cos^2 \theta + 1) - \nu) P_\nu^m(\cos \theta) + \nu(\nu + m) \cos \theta P_{\nu-1}^m(\cos \theta) + \nu(\nu - m + 1) \cos \theta P_{\nu+1}^m(\cos \theta)] \quad (92)$$

For simplicity, define the function $\Theta(\theta)$.

$$\Theta(\theta) = \frac{1}{\sin^2 \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial P_\nu^m}{\partial \theta} \right) \quad (93)$$

$$= \frac{1}{\sin \theta} [(m^2 - \nu^2(\cos^2 \theta + 1) - \nu)P_\nu^m(\cos \theta) + \nu(\nu + m) \cos \theta P_{\nu-1}^m(\cos \theta) + \nu(\nu - m + 1) \cos \theta P_{\nu+1}^m(\cos \theta)] \quad (94)$$

In this way, the second term of the Laplacian becomes

$$\frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial B_n}{\partial \theta} \right) = e^{-z/2} L_k(z) \frac{\Theta(\theta)}{r^2} A_{\nu m}(\phi) \quad (95)$$

The ϕ term of the Laplacian is significantly simpler and can be calculated as follows.

$$\frac{\partial B_n}{\partial \phi} = e^{-\frac{z}{2}} L_k(z) P_\nu^m(\cos \theta) \frac{\partial A_{\nu m}}{\partial \phi} \quad (96)$$

$$\frac{\partial^2 B_n}{\partial \phi^2} = e^{-\frac{z}{2}} L_k(z) P_\nu^m(\cos \theta) \frac{\partial^2 A_{\nu m}}{\partial \phi^2} \quad (97)$$

Take the first and second derivatives of $A_{\nu m}$.

$$\frac{\partial A_{\nu m}}{\partial \phi} = m \begin{cases} K_{\nu|m|} \cos(|m|\phi) & m < 0 \\ 0 & m = 0 \\ -K_{\nu m} \sin(m\phi) & m > 0 \end{cases} \quad (98)$$

$$\frac{\partial^2 A_{\nu m}}{\partial \phi^2} = -m^2 \begin{cases} K_{\nu|m|} \sin(|m|\phi) & m < 0 \\ 0 & m = 0 \\ K_{\nu m} \cos(m\phi) & m > 0 \end{cases} \quad (99)$$

$$(100)$$

This results in the convenient property

$$\frac{\partial^2 A_{\nu m}}{\partial \phi^2} = -m^2 A_{\nu m} \quad (101)$$

such that the last term in the Laplacian can be expressed as

$$\frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 B_n}{\partial \phi^2} = -e^{-\frac{z}{2}} L_k(z) \frac{m^2}{r^2 \sin^2 \theta} P_\nu^m(\cos \theta) A_{\nu m} \quad (102)$$

In total, the full Laplacian of the 3D basis function can be expressed as follows.

$$\begin{aligned} \nabla^2 B_n = & \left(\frac{100}{R_E} \right)^2 \frac{1}{r} \left[2 \left(\frac{R_E}{100} \right) \frac{\partial Z}{\partial z} + r \frac{\partial^2 Z}{\partial z^2} \right] P_\nu^m(\cos \theta) A_{\nu m}(\phi) \\ & + e^{-z/2} L_k(z) \frac{\Theta(\theta)}{r^2} A_{\nu m}(\phi) + -e^{-\frac{z}{2}} L_k(z) \frac{m^2}{r^2 \sin^2 \theta} P_\nu^m(\cos \theta) A_{\nu m} \end{aligned} \quad (103)$$

B.2 Perpendicular Laplacian

The perpendicular Laplacian is defined by setting all radial derivatives in the full Laplacian to zero.

$$\nabla_{\perp}^2 f = \frac{1}{z^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial f}{\partial \theta} \right) + \frac{1}{z^2 \sin^2 \theta} \frac{\partial^2 f}{\partial \phi^2} \quad (104)$$

Both terms have actually been calculated previously for the full Laplacian.

$$\frac{1}{z^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial B_n}{\partial \theta} \right) = \frac{e^{-\frac{z}{2}}}{z^2} L_k(z) \Theta(\theta) A_{\nu m}(\phi) \quad (105)$$

$$\frac{1}{z^2 \sin^2 \theta} \frac{\partial^2 B_n}{\partial \phi^2} = -\frac{m^2 e^{-\frac{z}{2}}}{z^2 \sin^2 \theta} L_k(z) P_{\nu}^m(\cos \theta) A_{\nu m}(\phi) \quad (106)$$

Combining these terms yields the following perpendicular Laplacian.

$$\nabla_{\perp}^2 B_n = \frac{e^{-\frac{z}{2}}}{z^2} L_k(z) \Theta(\theta) A_{\nu m}(\phi) - \frac{m^2 e^{-\frac{z}{2}}}{z^2 \sin^2 \theta} L_k(z) P_{\nu}^m(\cos \theta) A_{\nu m}(\phi) \quad (107)$$

$$= \frac{e^{-\frac{z}{2}}}{z^2} L_k(z) \left[\Theta(\theta) - \frac{m^2}{\sin^2 \theta} P_{\nu}^m(\cos \theta) \right] A_{\nu m}(\phi) \quad (108)$$

Create a new function $\Theta'_{\nu m}(\theta)$ to represent all θ contributions to the perpendicular Laplacian.

$$\Theta'_{\nu m}(\theta) = \Theta - \frac{m^2}{\sin^2 \theta} P_{\nu}^m(\cos \theta) \quad (109)$$

$$= \frac{1}{\sin^2 \theta} \left[-\nu (\nu \cos^2 \theta + \nu + 1) P_{\nu}^m(\cos \theta) + \nu(\nu + m) \cos \theta P_{\nu-1}^m(\cos \theta) + \nu(\nu - m + 1) \cos \theta P_{\nu+1}^m(\cos \theta) \right] \quad (110)$$

References

- De Santis, A. (1991). Translated origin spherical cap harmonic analysis. *Geophys. J. Int.*, 106:253–263.
- Haines, G. V. (1985). Spherical cap harmonic analysis. *J. Geophys. Res.*, 90:2583–2591.
- Thébault, E., Schott, J. J., and Manda, M. (2006). Revised spherical cap harmonic analysis (R-SCHA): Validation and properties. *J. Geophys. Res.*, 111.