The Spherical Laplacian

Skew-Symmetry of the Derivative Operator

Given two differentiable functions $f, g : [a, b] \to \mathbb{R}$, we can use the product rule to get:

$$
\int_a^b f'(t)g(t) \, dt = \left[ f(b)g(b) - f(a)g(a) \right] - \int_a^b f(t)g'(t) \, dt.
$$

Thus, if $f(a) = f(b) = 0$, if $g(a) = g(b) = 0$, or if $f(a) = f(b)$ and $g(a) = g(b)$ then the integral reduces to:

$$
\int_a^b f'(t)g(t) \, dt = - \int_a^b f(t)g'(t) \, dt.
$$

Parameterization

We assume a parameterization of the sphere, $S^2 = \{ p \in \mathbb{R}^3 \| p \| = 1 \}$, in terms of angles of zenith, $\theta \in [0, \pi]$, and azimuth $\phi \in [0, 2\pi)$:

$$
\Phi(\theta, \phi) = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta).
$$

Inner-Products

Given the parameterization $\Phi$, the inner-product of two functions $f, g : S^2 \to \mathbb{R}$ is defined as:

$$
\langle f, g \rangle = \int_0^\pi \int_0^{2\pi} f(\theta, \phi)g(\theta, \phi) \sin \theta \, d\phi \, d\theta.
$$

Laplacians

Given a function $f : S^2 \to \mathbb{R}$, the Laplacian of $f$ is defined as:

$$
\Delta f = \frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \left( \sin \theta \frac{\partial f}{\partial \theta} \right) + \frac{1}{\sin^2 \theta} \frac{\partial^2 f}{\partial \phi^2}.
$$

Laplacians and Inner-Products

Given two functions $f, g : S^2 \to \mathbb{R}$ we can express the inner-product of the Laplacian of $f$ with $g$ as:

$$
\langle \Delta f, g \rangle = \int_0^\pi \int_0^{2\pi} \left( g \cdot \frac{\partial}{\partial \theta} \left[ \sin \theta \frac{\partial f}{\partial \theta} \right] + \frac{g}{\sin \theta} \frac{\partial}{\partial \phi} \left[ \frac{\partial f}{\partial \phi} \right] \right) \, d\phi \, d\theta.
$$

Since, for fixed $\theta_0$, the functions $f(\theta_0, \phi)$ and $g(\theta_0, \phi)/\sin(\theta_0)$ are periodic functions in $\phi$, with period $2\pi$, we can bring over the partial-derivative with respect to $\phi$ in the last term to get:

$$
\langle \Delta f, g \rangle = \int_0^\pi \int_0^{2\pi} \left( g \cdot \frac{\partial}{\partial \theta} \left[ \sin \theta \frac{\partial f}{\partial \theta} \right] - \frac{\partial}{\partial \phi} \left[ g/\sin \theta \frac{\partial f}{\partial \phi} \right] \right) \, d\phi \, d\theta.
$$
Similarly, since for fixed $\phi_0$, the function $\sin \theta \frac{\partial f}{\partial \theta}(\theta, \phi_0)$ is zero-valued at the end-points $\theta = 0$ and $\theta = \pi$, we can bring over the partial-derivative with respect to $\theta$ in the middle term to get:

$$\langle \Delta f, g \rangle = \int_0^\pi \int_0^{2\pi} \left( -\frac{\partial g}{\partial \theta} \cdot \left[ \sin \theta \frac{\partial f}{\partial \theta} \right] - \frac{\partial}{\partial \phi} \left[ \frac{g}{\sin \theta} \right] \frac{\partial f}{\partial \phi} \right) d\phi d\theta. $$

Simplifying, the equation for the inner-product of the Laplacian of $F$ with $G$ reduces to:

$$\langle \Delta f, g \rangle = -\int_0^\pi \int_0^{2\pi} \left( \sin \theta \frac{\partial g}{\partial \theta} \frac{\partial f}{\partial \theta} + \frac{1}{\sin \theta} \frac{\partial g}{\partial \phi} \frac{\partial f}{\partial \phi} \right) d\phi d\theta. \quad (1)$$

**Discussion**

1. Since our work will be considering finite elements $f(\theta, \phi)$ and $g(\theta, \phi)$ that are separable, piecewise polynomial functions, computing the inner-product of the Laplacian of $f$ with $g$ reduces to the problem of computing definite integrals of the form $P(\theta) \cdot \sin \theta$ and $Q(\theta) / \sin \theta$ where $P$ and $Q$ are arbitrary polynomials. The first integral can be computed in closed form using the identities:

$$\int x^n \sin x \, dx = -x^n \cos x + n \int x^{n-1} \cos x \, dx$$

$$\int x^n \cos x \, dx = x^n \sin x - n \int x^{n-1} \sin x \, dx. $$

The second integral will likely require approximation via discrete summation.

2. The advantage of Equation 1 is that it expresses the inner-product of the Laplacian of $f$ with $g$ directly from the partials of $f$. Since the derivatives of finite elements can be expressed as the difference between elements of one degree less, the equation should give us the “hook” for defining the Poisson equation when the constraints are given in terms of finite differences.

3. Since the integral in Equation 1 may not be well-defined near the poles (where $\sin \theta$ goes to zero) care must be taken in defining the elements whose support overlaps the poles. In particular, this motivates a choice of elements at the poles that are strictly functions of $\theta$ so that $\partial f / \partial \phi$ will be zero at the poles, canceling out the singularity.
Spherical Finite Elements

Defining the Elements

**Notation:** We denote by $B^0(t)$ the rectangular function on the interval $[-0.5, 0.5]$:  

$$B^0(t) = \begin{cases} 1 & \text{if } |t| < 0.5 \\ 0 & \text{otherwise} \end{cases}$$

We denote by $B^d(t)$ the $d$-th order element, obtained by convolving the step function $B^0$ with itself $d$ times:  

$$B^d(t) = \left(B^0(t)\right)^d$$

which satisfies the derivative property:  

$$\frac{d}{dt}B^d(t) = B^{d-1}(t + 1/2) - B^{d-1}(t - 1/2).$$

Since we work strictly with second-order elements, we simplify the notation by setting $B(t) = B^2(t)$, which is a function compactly supported on the domain $[-1.5, 1.5]$. For the second-order elements, we also have the nesting property allowing us to express a coarser element as the linear combination of finer elements:

$$B(t) = \sum_{k=0}^{3} \alpha_k B(2(t + 3/4 - k/2)),$$

with weighting coefficients $(\alpha_0, \alpha_1, \alpha_2, \alpha_3) = \frac{1}{4}(1, 3, 3, 1)$.

**Notation:** Given $M, N \in \mathbb{Z}$, we define the space of spherical functions $B^{M,N}$ as the span of the $(M-2)\times N+2$ basis functions:

$$B^{M,N} = \text{Span}\{b_{0}^{M,N}(\theta, \phi), b_{1,0}^{M,N}(\theta, \phi), \ldots, b_{1,N-1}^{M,N}(\theta, \phi), \ldots, b_{M-2,0}^{M,N}(\theta, \phi), \ldots, b_{M-2,M-1}^{M,N}(\theta, \phi), b_{M-1}^{M,N}(\theta, \phi)\}.$$

For $i \notin \{0, M-1\}$, the elements $b_{i,j}^{M,N}$ are just the shifts and scales of the second-order element to the center of the $(i,j)$-th cell on the sphere:

$$b_{i,j}^{M,N}(\theta, \phi) = B\left(\frac{M}{\pi}\left(\theta - \frac{\pi}{M}(i + 1/2)\right)\right) \cdot B\left(\frac{N}{\pi}\left(\phi - \frac{2\pi}{N}(j + 1/2)\right)\right),$$

and for the two “cap functions” we get elements that are rotationally symmetric about the $z$-axis:

$$b_{0}^{M,N}(\theta, \phi) = B\left(\frac{M}{\pi}\left(\theta - \frac{\pi}{M}1/2\right)\right) + B\left(\frac{M}{\pi}\left(\theta + \frac{\pi}{M}1/2\right)\right)$$

$$b_{M-1}^{M,N}(\theta, \phi) = B\left(\frac{M}{\pi}\left((\theta - \pi) - \frac{\pi}{M}1/2\right)\right) + B\left(\frac{M}{\pi}\left((\theta - \pi) + \frac{\pi}{M}1/2\right)\right).$$

**Defining the Prolongation Operator**

Using the the nesting property of the 1D elements, we can express the elements at resolution $M \times N$ as linear combinations of elements at resolution $2M \times 2N$:

$$b_{i,j}^{M,N}(\theta, \phi) = \sum_{k,l=0}^{3} \alpha_k \alpha_l b_{i+k-1,2j+l-1}^{2M,2N}(\theta, \phi).$$
\begin{align*}
\quad b_{0}^{M,N}(\theta, \phi) &= b_{0}^{2M,2N}(\theta, \phi) + \frac{3}{4} \sum_{j=0}^{2N-1} b_{1,j}^{2M,2N}(\theta, \phi) + \frac{1}{4} \sum_{j=0}^{2N-1} b_{2,j}^{2M,2N}(\theta, \phi) \\
\quad b_{M-1}^{M,N}(\theta, \phi) &= b_{2M-1}^{2M-1,N}(\theta, \phi) + \frac{3}{4} \sum_{j=0}^{2N-1} b_{2M-2,j}^{2M,2N}(\theta, \phi) + \frac{1}{4} \sum_{j=0}^{2N-1} b_{2M-3,j}^{2M,2N}(\theta, \phi). 
\end{align*}

Note that since for “interior functions” we have \( i \not\in \{0, M-1\} \), this implies that \( 2i+k-1 \not\in \{0, 2M-1\} \) and hence the “interior” function \( b_{i,j}^{M,N} \) at resolution \( M \times N \) is expressed as the linear combination of “interior functions” at resolution \( 2M \times 2N \).
Finite Differences

Given a spherical function $g$, expressed as the linear combination of elements at resolution $M \times N$:

$$g(\theta, \phi) = g_0 b_0(\theta, \phi) + g_{M-1} b_{M-1}(\theta, \phi) + \sum_{i=1}^{M-2} \sum_{j=0}^{N-1} g_{i,j} b_{i,j}(\theta, \phi),$$

we can express the partial derivatives of $g$ as:

$$\frac{\partial}{\partial \theta} g(\theta, \phi) = \frac{M}{\pi} \sum_{i=1}^{M-1} (g_{i,j} - g_{i-1,j}) B^1 \left( \frac{M}{\pi} \left( \theta - \pi \frac{i}{M} \right) \right) \left[ \sum_{j=0}^{N-1} B^2 \left( \frac{N}{2\pi} \left( \phi - \frac{2\pi}{N} (j + 1/2) \right) \right) \right]$$

$$\frac{\partial}{\partial \phi} g(\theta, \phi) = \frac{N}{2\pi} \sum_{i=1}^{M-2} B^2 \left( \frac{M}{\pi} \left( \theta - \pi \frac{i+1/2}{M} \right) \right) \left[ \sum_{j=0}^{N-1} (g_{i,j} - g_{i,j-1}) B^1 \left( \frac{N}{2\pi} \left( \phi - \frac{2\pi}{N} j \right) \right) \right]$$

where for simplicity of notation, we set:

$$g_{0,j} \equiv g_0, \quad g_{M-1,j} \equiv g_{M-1}, \quad \text{and} \quad g_{i,j+N} \equiv g_{i,j}.$$  

Thus, if we are given the finite differences of the function $g$, we can treat these values as the coefficients for the mixed-degree elements whose linear combination gives the partial derivatives of $g$. 


Experimental Results

Figure 1: Parameterizations of the unit sphere with:

**Left:** The zenith-/azimuth-angle parameterization.

**Center:** The TOAST parameterization with no subdivision.

**Right:** The TOAST parameterization with five levels of subdivision.

To evaluate our method, we compared the solutions to the Poisson equation using the zenith-/azimuth-angle parameterization with two different solutions obtained using the TOAST parameterization. The first TOAST parameterization is the standard parameterization, defining a bijective mapping from the domain $[0, 4] \times [0, 1]$ to the unit-sphere (modulo boundary trickery). The second parameterization is obtained by using the periodicity properties of the TOAST parameterization to obtain a mapping of the domain $[0, 4] \times [0, 4]$ which is a four-fold covering of the unit sphere and has toroidal symmetry. (The triangulation resulting from the zenith/azimuth and TOAST parameterizations are shown in Figure .) To compare the parameterizations, we used the fact that spherical harmonics at frequency $l$ are the eigenvectors of the spherical Laplacian, with eigenvalues $-l \cdot (l + 1)$ so if our input constraints are the spherical harmonics, we know what the expected solution values should be.

More specifically, for the three parameterizations we compute:

1. $h$: The vector of sample values, with $h_i$ equal to value of the spherical harmonic at vertex $i$,

2. $D$: The dot-product matrix, with $D_{ij}$ equal to the dot product of the $i$-th basis element with the $j$-th basis element, and

3. $L$: The Laplacian matrix, with $L_{ij}$ equal to the dot product of the Laplacian of the $i$-th basis element with the $j$-th basis element.

Using these, we can compute the vector of solutions values, $g$ by computing the dot-product constraints, and solving the Laplacian:

$$g = L^{-1} Dh.$$  

(For full correctness, we should also incorporate the sampling matrix, to transition between sample values and coefficients, but in previous experiments we had found that it is reasonable safe to just interpret sample values as coefficients.)
First, since we are using a multigrid solver, we would like to measure the accuracy of the solver, given by constraints given by an error

The figure highlights several important results. First, we see that the Laplacian does not return the correct solution, with noticeable seams occurring at the “folds” of the octahedron where the orientation of the local frame changes discontinuously.

Second, we would like to measure how close the computed solution to the true solution:

\[ E_{\text{solver}}(f_l) = \frac{\|f_l - \Delta f_l\|}{\|f_l\|}. \]

Second, we would like to measure how close the computed solution to the true solution:

\[ E_{\text{harmonic}}(f_l) = \frac{\|f_l + l \cdot (l + 1) f_l\|}{\|f_l\|}. \]

Table 1 shows these errors for the different parameterizations, for multigrid solvers with \( k = 1 \) and \( k = 5 \) updates per level. The table further corroborates the visual results we had seen previously. For all the parameterizations, we find that the multigrid solver converges efficiently and hence the failure of the TOAST parameterizations to return the scaled harmonic is likely due to the incorrect formulation of the Poisson equation, and not the solver itself.

Note: The previous error problem has been resolved, so that now solutions error are not quite as small, and correspond more reasonably to harmonic errors.
Figure 2: These are the old images and should be replaced with images from the updated solver. 
**Top Row:** The Laplacian constraints, corresponding to harmonics at frequencies 1, 2, 3, and 4. 
**Second Row:** The solutions obtained using the zenith/azimuth parameterization. 
**Third Row:** The solutions obtained using the bijective TOAST parameterization. 
**Bottom Row:** The solutions obtained using the 4-fold covering TOAST parameterization.
finite elements interpretation

for both the TOAST and the zenith/azimuth parameterization, we can formulate what is going on in terms of the Galerkin method. In both situations we have the following set-up:

1. We have the space of real-valued functions on the sphere, which we denote \( \mathcal{F} \).
   (For all \( f \in \mathcal{F} \) and all \( p \in S^2 \) we have \( f(p) \in \mathbb{R} \).)

2. We have the space of vector-fields on the sphere, which we denote \( \mathcal{V} \).
   (For all \( V \in \mathcal{V} \) and all \( p \in S^2 \) we have \( V(p) \in T_p S^2 \).)

3. We have a linear operator \( G \) that takes a function on the sphere and returns a vector-field, \( G: \mathcal{F} \to \mathcal{V} \).

4. And finally, we have a finite set of functions \( f_1, \ldots, f_n \in \mathcal{F} \) spanning a sub-space of functions.

Given this set-up, the challenge addressed using both parameterizations can be stated as follows: Given a vector field \( \tilde{V} \in \mathcal{V} \), find the spherical function \( f(p) = \sum \alpha_j f_j(p) \), minimizing:

\[
\| G(f) - \tilde{V} \|^2.
\]

Using the Galerkin method, the coefficients \( \alpha^i = \{ \alpha_1, \ldots, \alpha_n \} \) minimizing the normed-difference are precisely those satisfying the system \( M \alpha = \nu \) where the matrix \( M \) and the constraint vector \( \nu \) are defined by:

\[
M_{ij} = \langle G(f_i), G(f_j) \rangle \quad \text{and} \quad \nu_j = \langle \tilde{V}, G(f_j) \rangle.
\]

In terms of the TOAST parameterization, there are two separate problems. First, the operator \( G \) need not the gradient operator (as it takes derivatives in the parameter domain, rather than over the sphere) and second, the inner-products are computed by integrating over the parameter domain, not over the sphere.

To get a better sense of how these two problems interact, and why this problem does not arise in the zenith/azimuth parameterization, we need to review a bit of differential geometry, focusing on how we can transition functions, gradient fields, and metrics from a manifold to a parameterization domain and back.
Differential Geometry

Pull-Backs and Push-Forwards

Given a manifold $M \subset \mathbb{R}^3$ and given a parametrization of the manifold $\Phi : D \to M$, where $D \subset \mathbb{R}^2$ is the parameterization domain, the pull back $\Phi^*$ is a map that takes functions defined over $M$ to functions defined over $D$. Specifically, given $F : M \to \mathbb{R}$, the pull-back of $F$ is defined through composition as:

$$\Phi^* F \equiv F \circ \Phi.$$

Using the parameterization $\Phi$, we can integrate functions over $M$ by integrating the pull-back over $D$ with the appropriate change of variables term:

$$\int_M F(p) dp = \int_D (\Phi^* F)(q)|d\Phi| dq,$$

where $|d\Phi|$ is the area term, computed as the length of the cross-product:

$$|d\Phi| = \left| \begin{array}{c} \frac{\partial \Phi}{\partial x} \\ \frac{\partial \Phi}{\partial y} \end{array} \right|.$$

We can also define the push-forward $\Phi_*$ that maps vectors in the tangent space $T_q D$ to vectors in the tangent space $T_{\Phi(q)} M$, where the push-forward of the tangent vector $\vec{v}$ acts on a function $F : M \to \mathbb{R}$ by differentiation of the pull-back:

$$(\Phi_* \vec{v}) F \equiv \vec{v}(\Phi^* F).$$

If we think of the tangent space of $T_{\Phi(q)} M$ as the sub-space of $\mathbb{R}^3$ spanned by the vectors tangent to $M$ at $\Phi(q)$, the mapping from $T_q D$ to $T_{\Phi(q)} M$ can be expressed as:

$$\Phi_* \left( \alpha_1 \frac{\partial}{\partial x} + \alpha_2 \frac{\partial}{\partial y} \right) = \alpha_1 \frac{\partial \Phi}{\partial x} + \alpha_2 \frac{\partial \Phi}{\partial y}.$$

Finally, we can pull-back the metric on $M$, defined by its embedding in $\mathbb{R}^3$, to a metric on $D$. Specifically, given vectors $\vec{v}, \vec{w} \in T_q D$, we define the pull-back of the inner-product on $T_q D$ to be:

$$\langle \vec{v}, \vec{w} \rangle^* \equiv \langle \Phi_* \vec{v}, \Phi_* \vec{w} \rangle.$$

If we choose a basis for $T_q D$, defining the pull-back metric is equivalent to defining a symmetric, positive definite, matrix $Q$ such that $\vec{v}^T Q \vec{w} = \langle \Phi_* \vec{v}, \Phi_* \vec{w} \rangle$. And, for the canonical basis $(\partial/\partial x, \partial/\partial y)$, the metric tensor $Q$ takes the form:

$$Q = \begin{pmatrix} \left( \begin{array}{c} \frac{\partial \Phi}{\partial x} \\ \frac{\partial \Phi}{\partial x} \end{array} \right) \\ \left( \begin{array}{c} \frac{\partial \Phi}{\partial y} \\ \frac{\partial \Phi}{\partial y} \end{array} \right) \end{pmatrix}.$$

If we choose vector fields $\vec{V}_1$ and $\vec{V}_2$ defining an orthonormal frame on $M$ then relative to this frame, the push-forward of the vector field can be represented by a $2 \times 2$ matrix $P$ whose columns are the coefficients of $\partial \Phi/\partial x$ and $\partial \Phi/\partial y$ relative to $\vec{V}_1$ and $\vec{V}_2$. Note that by construction, this implies that:

$$Q = P^T P.$$
Gradients

Given a function $F : M \rightarrow \mathbb{R}$, we can define the gradient of $F$ by differentiating in the $\vec{V}_1$ and $\vec{V}_2$ directions:

$$\nabla_M F = (\nabla_{\vec{V}_1} F) \vec{V}_1 + (\nabla_{\vec{V}_2} F) \vec{V}_2.$$  

With a little bit of work, we can show that the coefficients of the tangent vector in $T_qD$ whose push-forward is the evaluation of the gradient field $\nabla_M F$ at $q$ can be computed by applying the inverse of the symmetric, positive-definite matrix $Q$ to the partial derivatives of the pull-back of $F$ to $D$. Specifically, setting $f = F \circ \Phi$ we get:

$$\Phi_* \left( \alpha_1 \frac{\partial}{\partial x} + \alpha_2 \frac{\partial}{\partial y} \right) = \nabla_M F(q) \quad \text{where} \quad \alpha = Q^{-1} \nabla f.$$  

Thus, given two functions $F, G : M \rightarrow \mathbb{R}$, with pull-backs $f = F \circ \Phi$ and $g = G \circ \Phi$ on $D$, we can express the gradient fields of the two functions, relative to the basis $\{ \Phi_* \partial/\partial x, \Phi_* \partial/\partial y \}$ as

$$\nabla_M F = Q^{-1} \nabla f \quad \text{and} \quad \nabla_M G = Q^{-1} \nabla g.$$  

Thus, the dot-product of the gradients of the two functions is:

$$\langle \nabla_M F, \nabla_M G \rangle = (\nabla f)^t (Q^{-1})^t QQ^{-1} \nabla g = (\nabla f)^t Q^{-1} \nabla g$$  

and hence the integral of the dot-products of the two vector fields over all of $M$ can be computed as:

$$\int_M \langle \nabla_M F, \nabla_M G \rangle dp = \int_D (\nabla f)^t Q^{-1} \nabla g |d\Phi| dq.$$  

Zenith/Azimuth Parameterization

In this parameterization, the mapping $\Phi$ from the domain $D = [0, \pi] \times [0, 2\pi)$ into $S^2$ is given by:

$$\Phi(\theta, \phi) = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta).$$  

Computing the derivatives, we get:

$$\frac{\partial \Phi}{\partial \theta} = (\cos \theta \cos \phi, \cos \theta \sin \phi, -\sin \theta)$$  

$$\frac{\partial \Phi}{\partial \phi} = (-\sin \theta \sin \phi, \sin \theta \cos \phi, 0)$$  

$$|d\Phi| = \sin \theta$$  

$$Q = \begin{pmatrix} 1 & 0 \\ 0 & \sin^2 \theta \end{pmatrix}$$

Thus, given two functions $f(\theta, \phi)$ and $g(\theta, \phi)$ the inner-product of the gradients, integrated over the entire sphere, can be expressed as:

$$\int_{S^2} \langle \nabla_{S^2} f(p), \nabla_{S^2} g(p) \rangle dp = \int_0^\pi \int_0^{2\pi} \left( \frac{\partial f}{\partial \theta} \frac{\partial f}{\partial \phi} \right) \left( \frac{\sin \theta}{0} \frac{0}{1\sin \theta} \sin \theta \right) \left( \frac{\partial g}{\partial \theta} \frac{\partial g}{\partial \phi} \right)^t \sin \theta \sin \theta d\phi d\theta$$

$$= \int_0^\pi \int_0^{2\pi} \left( \sin \theta \frac{\partial f}{\partial \theta} + \frac{1}{\sin \theta} \frac{\partial f}{\partial \phi} \right) \left( \frac{\sin \theta}{0} \frac{0}{1\sin \theta} \sin \theta \right) \frac{\partial g}{\partial \theta} \frac{\partial g}{\partial \phi} \sin \theta \sin \theta d\phi d\theta$$
Which is the expression that we came up with in Equation 1. (The sign change is the typical one arising when transitioning from the dot-product of the Laplacian of $f$ with $g$ to the dot product of the gradient of $f$ with the gradient of $g$.)

**TOAST Parameterization**

In the TOAST parameterization we are less careful, ignoring both the area term and the push-forward operator, and evaluate the integrals simply by integrating the gradients, evaluated in the parameterization domain:

$$\int_D \left( \frac{\partial f}{\partial x} \frac{\partial g}{\partial x} + \frac{\partial f}{\partial y} \frac{\partial g}{\partial y} \right) dxdy.$$

The error of this estimation can be quantified at each point by measuring the extent to which the matrix $Q^{-1}|d\Phi|$ deviates from the identity matrix. One way to do this is by computing the eigenvalues of the matrix at each point in the parameterization domain and measuring their geometric mean and deviation. That is, if $\lambda_1(q)$ and $\lambda_2(q)$ are the two eigenvalues of $Q^{-1}|d\Phi|$ at $q$, we define the error functions $e_m$ and $e_d$ as:

$$e_m(q) = \sqrt{\lambda_1(q) \cdot \lambda_2(q)} \quad \text{and} \quad e_d(q) = \sqrt{\lambda_1(q)/\lambda_2(q)}.$$

Regardless of the parameterization domain $D$ and the mapping $\Phi$, the geometric mean will always be equal to 1. (This is because $Q = P^t P$ and so the determinant of $Q$ will be equal to the square of the determinant of $P$. Since the determinant of $P$ is the area of the parallelogram defined by $\partial\Phi/\partial\theta$ and $\partial\Phi/\partial\phi$, and since this also the value of $|d\Phi|$ everything comes out in the wash.) In contrast, Figure indicates that using the TOAST parameterization will result in different types of distortions as we move closer to and further from the four singular vertices.

Figure 3: These images should be updated with the new, subdivision, results. **Left:** A visualization of the area term $|d\Phi|$. **Right:** A visualization of the square-root of the ratios of the eigenvalues of $Q$. 

12
To compare the performance of the two parameterizations in stitching applications, we took several texture maps and generated two stitching tiles by offsetting, scaling, and exponentiating pixels on one half of the sphere. We then stitched together the gradient fields, computed their divergence, and used the Poisson solver defined by the equirectangular and TOAST parameterizations to generate a seamless image.

The results of these stitching applications can be seen in Figure 1.

1. **Earth**: One tile.
2. **Moon**: Two tiles, offset = (0.15, 0.15, 0.15).
3. **Jupiter**: Two tiles, scale = 0.8.
4. **Venus**: Two tiles, exponent = 1.6.
5. **Mars**: Eight tiles, offset = (0.15, 0.15, 0.15), scale = 0.8, and exponent = 1.6.

### Table 2: Error for the zenith/azimuth and TOAST solutions in the RGB channels for the different stitching examples shown in Figure 1.

<table>
<thead>
<tr>
<th></th>
<th>Zenith/Azimuth</th>
<th>TOAST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>G</td>
</tr>
<tr>
<td>Earth</td>
<td>$3 \times 10^{-3}$</td>
<td>$3 \times 10^{-3}$</td>
</tr>
<tr>
<td>Moon</td>
<td>$1 \times 10^{-2}$</td>
<td>$1 \times 10^{-2}$</td>
</tr>
<tr>
<td>Jupiter</td>
<td>$3 \times 10^{-3}$</td>
<td>$3 \times 10^{-3}$</td>
</tr>
<tr>
<td>Venus</td>
<td>$5 \times 10^{-3}$</td>
<td>$5 \times 10^{-3}$</td>
</tr>
<tr>
<td>Mars</td>
<td>$2 \times 10^{-2}$</td>
<td>$2 \times 10^{-2}$</td>
</tr>
</tbody>
</table>
Figure 4: Results of running gradient domain stitching showing the stitched colors (left), the stitching results obtained using the equirectangular parameterization (center), and the stitching results obtained using the TOAST parameterization (right).
Problems

One of the problems that becomes apparent when we consider the results of Table 1 and Table 2 is that the convergence of the equi-rectangular parameterization is slower than that of the TOAST parameterization. It appears that the error creeps in primarily near the polar caps. (This could be due to fact that at the caps very small elements abut a very large element. Or, it could be due to the fact that numerical approximation of the integral with the cosecant becomes unstable near the caps where the cosecant function has a pole.) A visualization of this can be seen when we solve the Poisson equation given constraints defined by a first-order harmonic, shown in Figure.

Figure 5: The solutions at resolutions 32 × 32, 64 × 64, 128 × 128, and 256 × 256 for a cascadic solver using \(k = 5\) Gauss-Seidel iterations to find the function whose Laplacian is a first-order harmonic.

The figure shows the results of a cascadic solver at resolutions 32 × 32, 64 × 64, 128 × 128, and 256 × 256. Although the solver converges to the correct solution with sufficient iterations (an indication that the Laplacian matrix is probably set up correctly) with only \(k = 5\) Gauss-Seidel iterations at each resolution, the solution exhibits distinctive dimples near the poles.

It’s possible that part of the problem is due to the fact that at the poles, any spherical function expressed
as the sum of the basis elements must have a zero gradient (since the partial derivatives of all basis functions
at the poles are zero). This would explain what we are seeing in the visualization where the spherical function
wants to have constant value near the poles (corresponding to constant radius in the visualization) resulting
in the dimpling artifact.

What makes this particularly troublesome is that the problem seems to occur directly at the prolongation
phase. This can be seen for the previous example where we prolong the correct solution from a resolution
of 32 × 32 to a solution at resolution 64 × 64 as shown in Figure . Despite the fact that the solution (left)
is correct at the lower resolution, the act of prolongation alone seems to introduce the dimpling artifacts
the higher resolution (center). With sufficiently many iterations (k = 64) this dimpling artifact can be
removed (right), indicating that the basis has sufficient resolution to represent the correct solution, but the
convergence is slow. (So perhaps using an approach such as WEB-splines, where we change the function
basis but not the space of functions they span, could work.)

Figure 6: Dimpling artifacts caused by the prolongation operation can be removed with sufficiently many
relaxation iterations.

This slow convergence can also be seen if we plot the coefficient of the north-pole basis function, the
coefficient of one of its 1-neighbors, and the coefficient of one of its 2-neighbors as a function of the number of
Gauss-Seidel iterations. As Figure (left) indicates, though the coefficient of the north-pole’s basis function
remains nearly constant throughout the relaxations, the coefficient of the adjacent basis function converges
rather slowly. Since for these examples it appears that the convergence is monotonic, it is natural to consider
an SOR scheme in which we “overshoot” in the relaxation phase. Although this appears to work in the above
example, as demonstrated by the tighter convergence plots in Figure (right), the performance with successive
over-relaxation deteriorates for the more complicated constraints in Figure .

So the question remains: Do we try to address the dimpling problem by refining the space of functions,
so that the prolongation gives a better solution than the one shown in Figure , or do we modify the basis
elements but keep the same space? The motivation for the former would be that the prolongation of
the correct solution is more reasonable. The motivation for the latter is that, as Figure (right) indicates, the
space has sufficient resolution to get us to the correct answer, it’s just that we need to find a way to get to
it more efficiently.

Thoughts: Is it possible that near the poles, the high-frequency of the elements could result in the fact
that elements with large differences in φ-index can still effect each other, and so it requires more relaxation
iterations to get the effects to propagate? Could this be solved by explicitly running multiple passes on these
Figure 7: The coefficient values for the north-pole basis function, the coefficient values for one of its 1-neighbors, and the coefficient values for one of its 2-neighbors as a function of the number of Gauss-Seidel relaxations. The plots on the left show convergence without successive over-relaxation, and the plots on the right show the results with SOR, where the over-relaxation value was in the range [1.0,1.75] and was determined by the area of under the basis function.

rows before proceeding on to the next rows? Alternatively, near the poles, can we use \( \phi \) functions with larger supports to reduce the frequency and allow the system to diffuse further in a single iteration?
Table 3: New solver and harmonic errors for the different parameterizations, at different frequencies and with different numbers of Gauss-Seidel updates per level.

**New Experimental Results**

To address the problem discussed in the previous section, we modified the solver used in the zenith/azimuth parameterization so that near the poles it would modify the relaxation scheme to allow the solution to diffuse across a greater range of $\phi$ indices. Specifically, in performing an iteration of the Gauss-Seidel solver, on a sphere sampled at a resolution of $M \times N$, with $N = 2^n$, we update as follows:

1. We update the coefficient corresponding to the basis function $b_{0, 0}$ once.
2. We update the coefficients corresponding to the basis functions $b_{1, j}$, $N/2$ times.
3. We update the coefficients corresponding to the basis functions $b_{2, j}$, $N/4$ times.
4. ... 
5. We update the coefficients corresponding to the basis functions $b_{n, 1}$, twice.
6. We update the coefficients corresponding to the basis functions $b_{n, m}$, with $m \in [n, M - n)$, ones.
7. We update the coefficients corresponding to the basis functions $b_{n, 1}$, twice.
8. ... 
9. We update the coefficients corresponding to the basis functions $b_{n, M}$, $N/2$ times.
10. We update the coefficient corresponding to the basis function $b_{n, M}$, one.

This results in a modified values for the Zenith-Azimuth errors previously shown in Table 1. The new results are shown in Table 3.

Similarly, using the updated solver, we get modified values for the Zenith-Azimuth errors for the stitching errors previously shown in Table 2. The new results are shown in Table 4.
Table 4: New error for the zenith/azimuth and TOAST solutions in the RGB channels for the different stitching examples shown in Figure.

<table>
<thead>
<tr>
<th></th>
<th>Zenith/Azimuth</th>
<th>TOAST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>G</td>
</tr>
<tr>
<td>Earth</td>
<td>1×10⁻³</td>
<td>1×10⁻³</td>
</tr>
<tr>
<td>Moon</td>
<td>9×10⁻⁴</td>
<td>9×10⁻⁴</td>
</tr>
<tr>
<td>Jupiter</td>
<td>4×10⁻⁴</td>
<td>3×10⁻⁴</td>
</tr>
<tr>
<td>Venus</td>
<td>7×10⁻⁴</td>
<td>7×10⁻⁴</td>
</tr>
<tr>
<td>Mars</td>
<td>2×10⁻³</td>
<td>2×10⁻³</td>
</tr>
</tbody>
</table>
An Adaptive Approach

An alternate approach that we consider is to modulate the number of elements in each row of elevation so that the elements are associated more-or-less equal areas. Figure compares the traditional zenith-azimuth parameterization of the sphere with the adaptive parameterization.

Specifically, to simulate the traditional zenith/azimuth parameterization sampled on an $M \times N$ grid, we set the number of finite elements in the $m$-th row to be equal to the smallest integer $c$ such that:

$$2^c \geq \sin \left( \frac{m + 0.5}{M} \pi \right) \cdot N.$$ 

Thus, around the equator we still have the same $N$ elements per row, but as we get closer to the poles, the number of elements diminishes.

This type of parameterization could be promising for two reasons. First, it should allow the solution to diffuse in a more well-behaved fashion near the poles. Second, it leverages the fact that near the poles we are over-sampling in the $\phi$ direction and uses that to reduce the dimensionality of the system.

Using this new parameterization, we re-ran both the spherical harmonics experiments and the stitching experiments. The results of these can be seen in Tables 5 and 7 respectively. Examining these tables, several patterns appear to emerge. First, the adaptive Zenith-Azimuth appears to consistently give slightly better harmonic errors. Second, the adaptive parameterization appears to do noticeably worse with just one relaxation step, but tightens the gap as more relaxation steps are used. Finally, for the stitching experiments, it seems that the adaptive approach seems to slightly outperform the regular one.
Table 5: Solver and harmonic errors for the different parameterizations, at different frequencies and with different numbers of Gauss-Seidel updates per level.

<table>
<thead>
<tr>
<th>Freq.</th>
<th>Zenith/Azimuth</th>
<th>TOAST</th>
<th>Adaptive Zenith/Azimuth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Solver Error</td>
<td>Harmonic Error</td>
<td>Solver Error</td>
</tr>
<tr>
<td></td>
<td>$k = 1$</td>
<td>$k = 5$</td>
<td>$k = 1$</td>
</tr>
<tr>
<td>1</td>
<td>$1 \times 10^{-4}$</td>
<td>$2 \times 10^{-4}$</td>
<td>$5 \times 10^{-4}$</td>
</tr>
<tr>
<td>2</td>
<td>$6 \times 10^{-4}$</td>
<td>$6 \times 10^{-5}$</td>
<td>$5 \times 10^{-4}$</td>
</tr>
<tr>
<td>3</td>
<td>$5 \times 10^{-4}$</td>
<td>$4 \times 10^{-5}$</td>
<td>$4 \times 10^{-4}$</td>
</tr>
<tr>
<td>4</td>
<td>$5 \times 10^{-4}$</td>
<td>$3 \times 10^{-5}$</td>
<td>$5 \times 10^{-4}$</td>
</tr>
<tr>
<td>5</td>
<td>$5 \times 10^{-4}$</td>
<td>$2 \times 10^{-5}$</td>
<td>$4 \times 10^{-4}$</td>
</tr>
<tr>
<td>6</td>
<td>$5 \times 10^{-4}$</td>
<td>$2 \times 10^{-5}$</td>
<td>$4 \times 10^{-4}$</td>
</tr>
<tr>
<td>7</td>
<td>$4 \times 10^{-4}$</td>
<td>$2 \times 10^{-5}$</td>
<td>$4 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

Table 6: Errors for the zenith/azimuth and TOAST solutions in the RGB channels for the different stitching examples shown in Figure . In this experiment, the number of iterations to $k = 5$.

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>G</th>
<th>B</th>
<th>R</th>
<th>G</th>
<th>B</th>
<th>R</th>
<th>G</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earth</td>
<td>$1 \times 10^{-4}$</td>
<td>$1 \times 10^{-4}$</td>
<td>$2 \times 10^{-4}$</td>
<td>$3 \times 10^{-3}$</td>
<td>$4 \times 10^{-3}$</td>
<td>$8 \times 10^{-3}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moon</td>
<td>$2 \times 10^{-4}$</td>
<td>$2 \times 10^{-4}$</td>
<td>$2 \times 10^{-4}$</td>
<td>$4 \times 10^{-5}$</td>
<td>$4 \times 10^{-5}$</td>
<td>$6 \times 10^{-5}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jupiter</td>
<td>$8 \times 10^{-5}$</td>
<td>$7 \times 10^{-5}$</td>
<td>$7 \times 10^{-5}$</td>
<td>$3 \times 10^{-5}$</td>
<td>$3 \times 10^{-5}$</td>
<td>$1 \times 10^{-4}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Venus</td>
<td>$1 \times 10^{-4}$</td>
<td>$1 \times 10^{-4}$</td>
<td>$2 \times 10^{-4}$</td>
<td>$3 \times 10^{-5}$</td>
<td>$3 \times 10^{-5}$</td>
<td>$9 \times 10^{-5}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mars</td>
<td>$4 \times 10^{-4}$</td>
<td>$5 \times 10^{-4}$</td>
<td>$5 \times 10^{-4}$</td>
<td>$3 \times 10^{-5}$</td>
<td>$4 \times 10^{-5}$</td>
<td>$8 \times 10^{-5}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Finally, to compare the two different spherical parameterizations, we can also run the solvers for a large number of iterations and then measure the difference (RMS and max) between the computed solution and the “ground-truth”. One thing to keep in mind when comparing these results is that at least part of the reason that the adaptive approach may have a lower maximum error is due to the fact that it effectively performs smoothing of the gradient field’s divergence near the poles, precisely the area where the regular parameterization seems to have the most trouble.
Table 7: Errors for the zenith/azimuth and TOAST solutions in the RGB channels for the different stitching examples shown in Figure . In this experiment, the number of iterations to $k = 5$. 

<table>
<thead>
<tr>
<th></th>
<th>Zenith/Azimuth</th>
<th></th>
<th>Adaptive Zenith/Azimuth</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>G</td>
<td>B</td>
<td>R</td>
</tr>
<tr>
<td>Earth</td>
<td>$\frac{3 \cdot 10^{-4}}{3 \cdot 10^{-3}}$</td>
<td>$\frac{2 \cdot 10^{-1}}{1 \cdot 10^{-3}}$</td>
<td>$\frac{2 \cdot 10^{-4}}{1 \cdot 10^{-3}}$</td>
<td>$\frac{3 \cdot 10^{-4}}{1 \cdot 10^{-3}}$</td>
</tr>
<tr>
<td>Moon</td>
<td>$\frac{4 \cdot 10^{-4}}{2 \cdot 10^{-2}}$</td>
<td>$\frac{4 \cdot 10^{-4}}{2 \cdot 10^{-2}}$</td>
<td>$\frac{4 \cdot 10^{-4}}{2 \cdot 10^{-2}}$</td>
<td>$\frac{5 \cdot 10^{-9}}{4 \cdot 10^{-4}}$</td>
</tr>
<tr>
<td>Jupiter</td>
<td>$\frac{4 \cdot 10^{-5}}{1 \cdot 10^{-3}}$</td>
<td>$\frac{6 \cdot 10^{-5}}{1 \cdot 10^{-3}}$</td>
<td>$\frac{6 \cdot 10^{-5}}{1 \cdot 10^{-3}}$</td>
<td>$\frac{4 \cdot 10^{-5}}{4 \cdot 10^{-4}}$</td>
</tr>
<tr>
<td>Venus</td>
<td>$\frac{9 \cdot 10^{-5}}{1 \cdot 10^{-3}}$</td>
<td>$\frac{8 \cdot 10^{-5}}{1 \cdot 10^{-3}}$</td>
<td>$\frac{8 \cdot 10^{-5}}{1 \cdot 10^{-3}}$</td>
<td>$\frac{3 \cdot 10^{-5}}{4 \cdot 10^{-4}}$</td>
</tr>
<tr>
<td>Mars</td>
<td>$\frac{5 \cdot 10^{-4}}{1 \cdot 10^{-2}}$</td>
<td>$\frac{5 \cdot 10^{-4}}{2 \cdot 10^{-2}}$</td>
<td>$\frac{5 \cdot 10^{-4}}{1 \cdot 10^{-2}}$</td>
<td>$\frac{4 \cdot 10^{-4}}{7 \cdot 10^{-4}}$</td>
</tr>
</tbody>
</table>