A High-Order Galerkin Solver for the Poisson Problem on the Surface of the Cubed Sphere

Michael Levy

University of Colorado at Boulder
Department of Applied Mathematics

August 10, 2007

Outline

Basic Premise

Goal: solve $-\nabla^2 u = f$ on the surface of a sphere.

Why?

- Eventually will solve global Shallow Water equations in vorticity-divergence form.
- Vorticity and divergence related to stream functions / velocity potentials via Laplace operator.

How?

- Spectral Element Method on the cubed sphere.

Spectral Element Method

1. Partition spatial domain into elements Ω^e.
2. Solve in weak form: multiply by a test function and integrate over each element.

$$-\int_{\Omega^e} (\nabla^2 u)\phi d\Omega = \int_{\Omega^e} f\phi d\Omega$$

2D Basics

- Looking for an approximate solution $u_h(x, y) \approx u(x, y)$ in a finite-dimensional function space.
- Let $\mathcal{V}_h = \{v(x, y) : v(x, y) = p_n(x)q_n(y)\}$, where p_n and q_n are polynomials of degree $\leq n$.
  - Functions in $\mathcal{V}_h$ must be continuous over element boundaries.
  - Both u_h and ϕ are in $\mathcal{V}_h$.
- Also need a quadrature rule for evaluating integrals: the Gauss-Lobatto-Legendre rule is used in both x and y

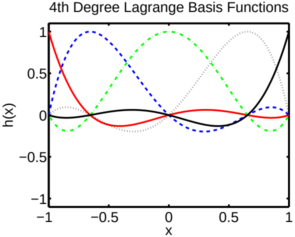
$\Longrightarrow$ GLL nodes affinely mapped to elements define grid

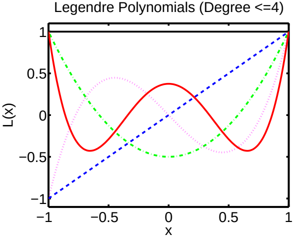
More SEM Set-up (Spatial Discretization)

- Gaussian Quadrature: $\int_{-1}^{1} \omega(x)p(x)dx = \sum_{i=0}^{n} w_i p(x_i)$

  GLL: $\omega(x) \equiv 1,\ x_0 = -1,\ x_n = 1$

- Interpolation: two options for basis functions





Nodal expansion (Lagrange basis)

$$h_i(x_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

$f(x) \approx \sum f(x_i) h_i(x)$

Modal expansion (Legendre basis)

$$f(x) \approx \sum_i f_i L_i(x)$$

$f_i = \int_{-1}^{1} f(x) L_i(x) dx$

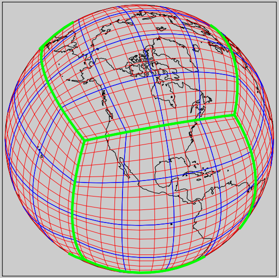
Cubed Sphere Basics

How can this methodology be extended to a spherical domain?

To use rectangular elements, turn to cubed sphere.

Cube Sphere Set-up

1. A cube is inscribed in a sphere
2. Points on the surface of the cube are projected onto the sphere
   - Gnomic / central projection (ray from center to surface of sphere)
3. Each face is tiled with elements as in the 2D case



In the figure above, each face of the cube has a 4×4 element grid with a 6×6 GLL grid.

Mapping Vectors Between the Sphere and the Cube

For a sphere of radius a:

- On each face, the metric tensor g_{ij} is given by

$$g_{ij} = \frac{a^2}{\rho^4 \cos^2 x^1 \cos^2 x^2} \begin{bmatrix} 1 + \tan^2 x^1 & -\tan x^1 \tan x^2 \\ -\tan x^1 \tan x^2 & 1 + \tan^2 x^2 \end{bmatrix},$$

where $\rho = (1 + \tan^2 x^1 + \tan^2 x^2)^{1/2}$.

- Defining the matrix A by

$$A = a \begin{bmatrix} \cos\theta\, \partial\lambda/\partial x^1 & \cos\theta\, \partial\lambda/\partial x^2 \\ \partial\theta/\partial x^1 & \partial\theta/\partial x^2 \end{bmatrix},$$

where (x^1, x^2) are the cartesian coordinates on the face of the cube, it follows that $A^T A = g_{ij}$.

Laplacian Operator on the Cubed Sphere

In spherical coordinates, the Laplacian is given on the surface of a sphere by

$$\nabla^2 u = \nabla \cdot \nabla u = \frac{1}{a^2 \cos\theta} \frac{\partial}{\partial\theta} \left[\cos\theta \frac{\partial u}{\partial\theta} \right] + \frac{1}{a^2 \cos^2\theta} \frac{\partial^2 u}{\partial\lambda^2}$$

On the surface of the cubed sphere, with $\nabla_g = (\partial/\partial x^1, \partial/\partial x^2)^T$,

$$\nabla^2 u = \frac{1}{\sqrt{g}} \nabla_g \cdot \left[\sqrt{g} A^{-1} A^{-T} \nabla_g u \right],$$

where $g = \det(g_{ij}) \implies \sqrt{g} = a^2 / (\rho^3 \cos^2 x^1 \cos^2 x^2)$.

Weak Form

(1) So the problem to solve is

$$-\frac{1}{\sqrt{g}}\nabla_g \cdot \left[\sqrt{g}A^{-1}A^{-T}\nabla_g u\right] = f.$$

(2) Or, slightly re-arranging terms,

$$-\nabla_g \cdot \left[\sqrt{g}A^{-1}A^{-T}\nabla_g u\right] = f\sqrt{g}.$$

(3) The first step is to cast in weak form:

$$-\int_{\Omega^e} \nabla_g \cdot \left[\sqrt{g}A^{-1}A^{-T}\nabla_g u\right]\phi d\Omega = \int_{\Omega^e} f\phi\sqrt{g}d\Omega.$$

(4) Integrating by parts simplifies the calculations:

$$\int_{\Omega^e} (A^{-T}\nabla_g u) \cdot (A^{-T}\nabla_g \phi)\sqrt{g}d\Omega = \int_{\Omega^e} f\phi\sqrt{g}d\Omega.$$

Quadrature

Letting $\phi(x^1, x^2) = h_p(x^1)h_q(x^2)$ for $p, q \in \{1, \ldots, N\}$, and applying the GLL quadrature to the weak form, results in the linear system

$$K^e\mathbf{u}^e = M^e\mathbf{f}^e,$$

where $\mathbf{u}^e$ and $\mathbf{f}^e$ are vectors containing the nodal coefficients of u and f on Ω^e, respectively.

The solution must be continuous across element boundaries, and this is enforced by using global assembly to construct a global system: $K = \bigwedge_e K^e$, $M = \bigwedge_e M^e$ and the system

$$K\mathbf{u} = M\mathbf{f}$$

is solved using the conjugate gradient method.

Test Problem

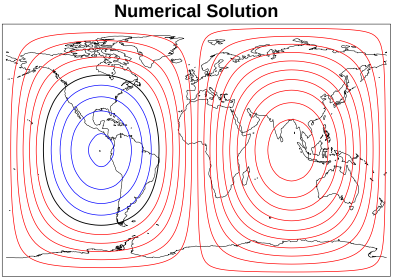
If $u = \sin(\lambda)\cos(\theta) + C$ then $-\nabla^2 u = 2\sin(\lambda)\cos(\theta)/a^2$. Working backwards, the test problem solved is

$$-\nabla^2 u = \frac{2\sin(\lambda)\cos(\theta)}{a^2}$$

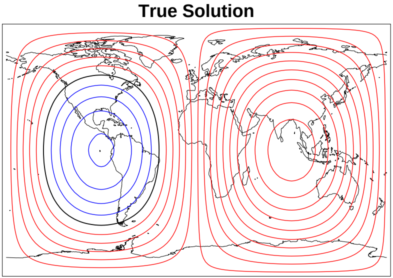
The numerical solution u_h is compared to the true solution $u = \sin(\lambda)\cos(\theta) + C$. The GLL quadrature rule is used to calculate the relative $L2$ error:

$$\epsilon = \left(\frac{\int_\Omega (u - u_h)^2 \sqrt{g} d\Omega}{\int_\Omega u^2 \sqrt{g} d\Omega} \right)^{1/2}$$

Contour Plots



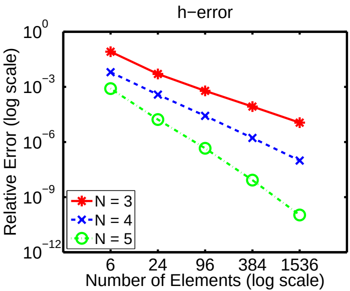
Contour plot of u_h.



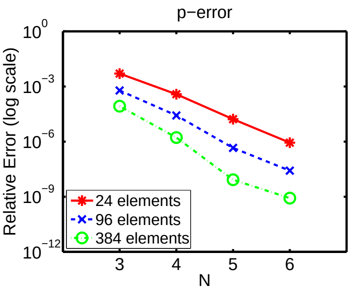
Contour plot of u.

For both plots, each face of the cube sphere had a 6×6 grid of elements and each element had a 4×4 GLL grid.

Error Plots



The *h*-error is measured by leaving the number of nodes per element constant but increasing the number of elements.



The *p*-error is measured by leaving the number of elements constant but increasing the number of nodes per element.

Future Work

1. Parallelization: this method is expensive, but fairly local so it should scale well.
2. Preconditioning: The conjugate gradient method is converging slowly for bigger grids / more elements; a diagonal preconditioner has been implemented but a better option may be needed.
3. Shallow Water Model: the work presented here, combined with an advection solver, will provide a high-order method for solving the shallow water equations (more at *PDEs on a Sphere '07*).