High-Order Spectral Element Transport Scheme on the Cubed Sphere using SAT

Implementation and Validation

Wang, Chi-Wei

December 21, 2025

Department of Applied Mathematics
National Chung Hsing University (NCHU)

Outline

Introduction

Motivation & Concept

Limitation of Traditional Grids:

- Pole Singularities in Lat-Lon grids force extremely small time steps (CFL constraint), strictly limiting computational efficiency.

Our Solution: Cubed Sphere

- The Cubed Sphere Approach: Eliminates singular points by decomposing the sphere into 6 identical local coordinate systems, enabling larger time steps and better scalability.

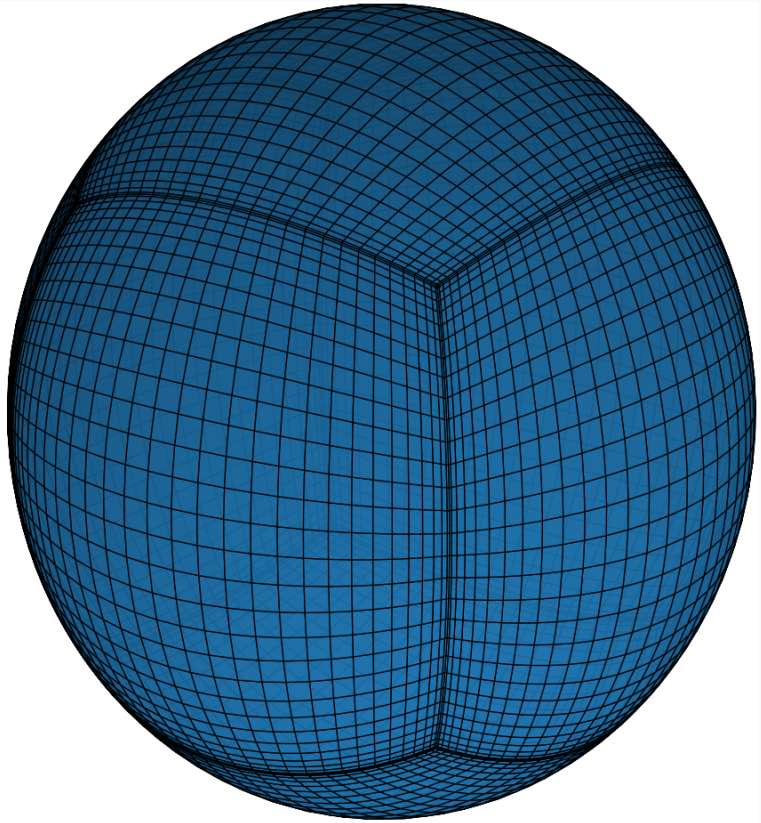


Figure: Projecting a cube onto a sphere

Objective & Methodology

Our primary goal is to implement a mass conservative transport solver for the advection equation on cubed sphere:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0 \tag{1}$$

Key Implementation Techniques:

1. **Geometry:** Equiangular Gnomonic Projection to optimize grid uniformity.
2. **Discretization:** Spectral Element Method (SEM) on Legendre-Gauss-Lobatto (LGL) nodes for high-order accuracy.
3. **Boundary Treatment:** Simultaneous Approximation Term (SAT) with **upwind numerical flux** for stable interface coupling.
4. **Time Integration:** Low-Storage Runge-Kutta (LSRK) scheme for memory-efficient explicit time stepping.

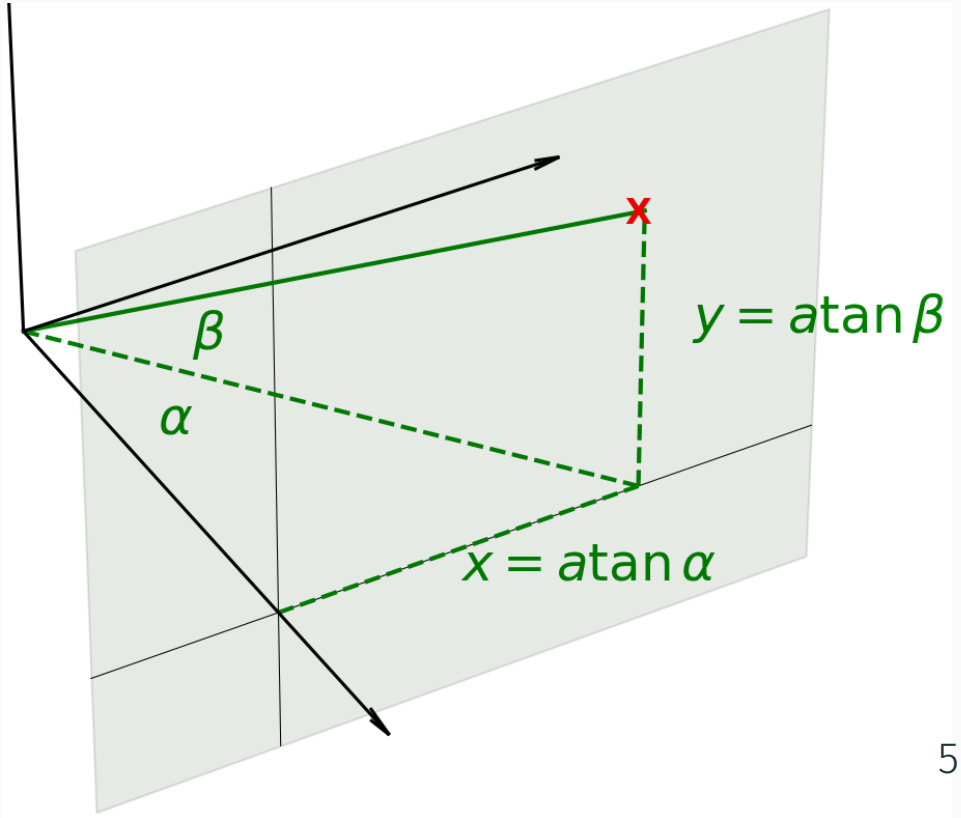
Geometric Framework

Geometry: Equiangular Projection

1. **Mapping Definitions**
   Points on the cube faces (x, y) are mapped to the sphere using central angles (α, β):

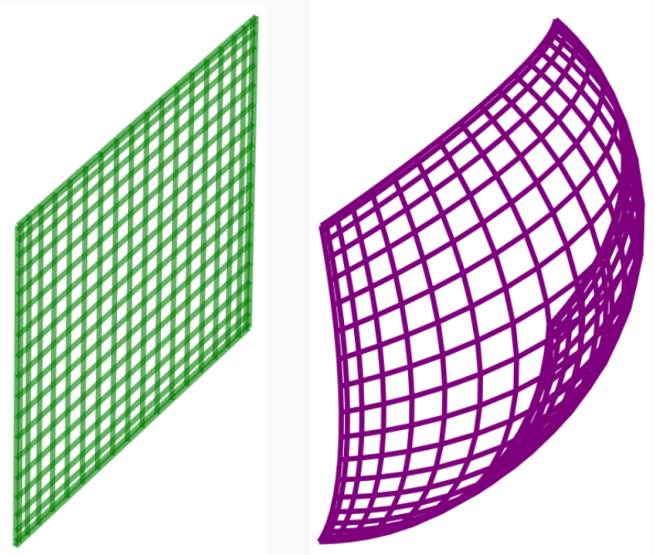
$$x = a \tan \alpha, \quad y = a \tan \beta$$

2. **Discretization**
   - The angles α and β are discretized using **Legendre-Gauss-Lobatto (LGL)** nodes.
   - **Domain:** $\alpha, \beta \in [-\pi/4, \pi/4]$.



Coordinate Transformation (2D to 3D)

Visual Representation



Left: Local face (xy)
Right: Global sphere (XYZ)

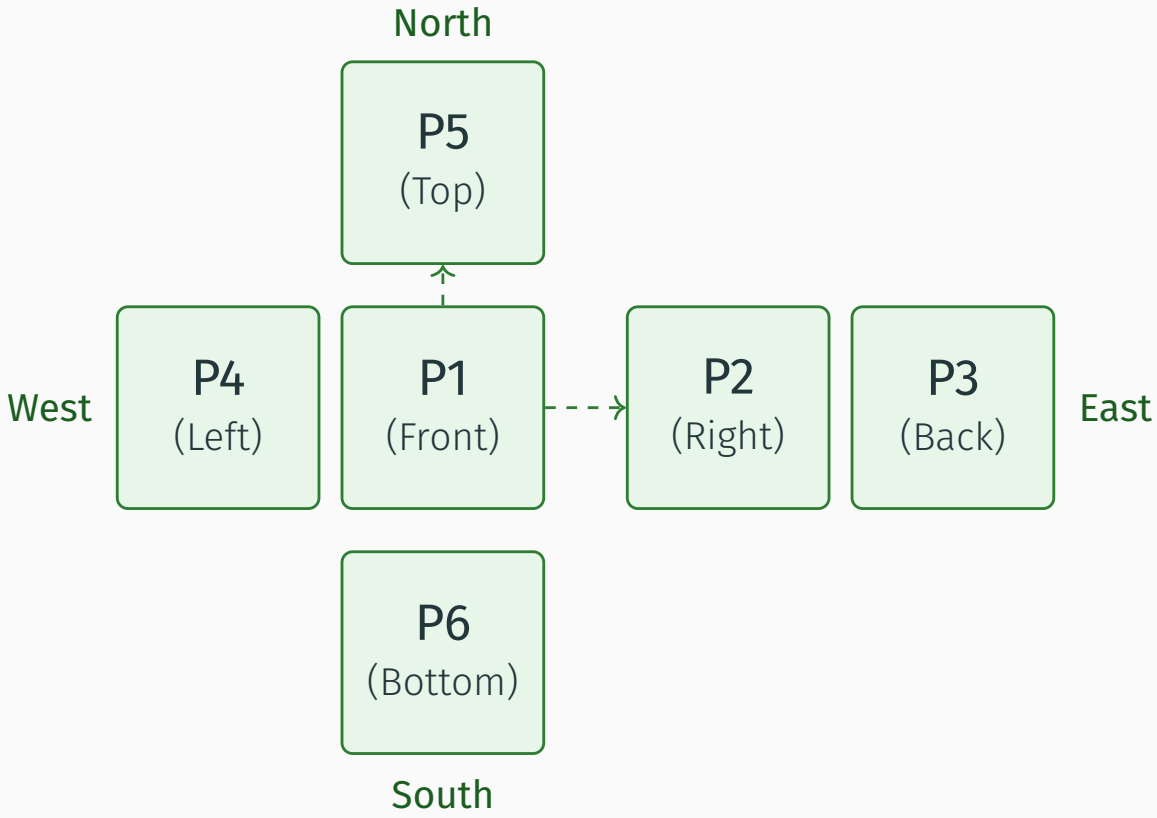
Face-to-Sphere Logic

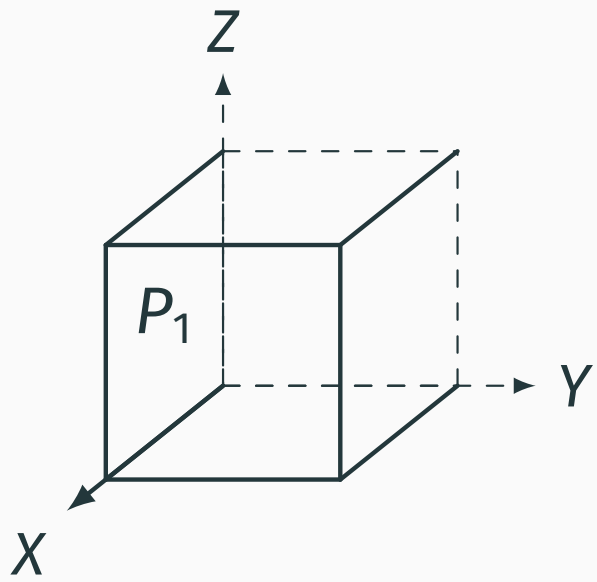
Face	X	Y	Z
P_1 (Front)	$s \cdot a$	$s \cdot x$	$s \cdot y$
P_2 (Right)	$-s \cdot x$	$s \cdot a$	$s \cdot y$
P_3 (Back)	$-s \cdot a$	$-s \cdot x$	$s \cdot y$
P_4 (Left)	$s \cdot x$	$-s \cdot a$	$s \cdot y$
P_5 (Top)	$-s \cdot y$	$s \cdot x$	$s \cdot a$
P_6 (Bottom)	$s \cdot y$	$s \cdot x$	$-s \cdot a$

Source: This table is from Appendix A of Nair et al. (2005).

Grid Topology & Connectivity

Since each face is a local coordinate system, passing information across boundaries requires coordinate rotation logic defined by a connectivity map.





3D View: Local Face P_1 definition

Metric Terms & Velocity Transformation

To solve the transport equation on the cubed sphere, we need two key geometric quantities derived from the transformation matrix A:

1. Area Element (Jacobian)

- The Jacobian $\sqrt{g}$ acts as the **area scaling factor** of the mapping.
- It is defined by the relation:

$$dS = \sqrt{g}\, d\alpha\, d\beta$$

- This term ensures mass conservation on the distorted grid (derived from $g_{ij} = A^T A$).

2. Contravariant Velocity

- Physical winds (u, v) must be projected onto the curvilinear coordinates (α, β).
- Using the inverse transform A^{-1}:

$$\begin{bmatrix} u^1 \\ u^2 \end{bmatrix} = A^{-1} \begin{bmatrix} u \\ v \end{bmatrix}$$

- u^1, u^2: Effective transport velocity components.

Geometry: Transformation Matrix A

Construction of Matrix A (Equiangular Projection)

- The transformation matrix A is constructed using the chain rule, combining spherical metric factors, coordinate derivatives, and the equiangular mapping derivatives:

$$A = \underbrace{R \begin{pmatrix} \cos\theta & 0 \\ 0 & 1 \end{pmatrix}}_{\text{Spherical Scale}} \underbrace{\begin{pmatrix} \frac{\partial\lambda}{\partial x} & \frac{\partial\lambda}{\partial y} \\ \frac{\partial\theta}{\partial x} & \frac{\partial\theta}{\partial y} \end{pmatrix}}_{\text{Coord. Transform}} \underbrace{\begin{pmatrix} a\sec^2\alpha & 0 \\ 0 & a\sec^2\beta \end{pmatrix}}_{\text{Equiangular Map}}$$

Note: The detailed derivation of matrix elements involves spherical trigonometry and chain rules. Please refer to **Appendix A & B of Nair et al. (2005)** for the complete analytical forms.

Geometry: Jacobian & Divergence

1. Jacobian for Equiangular Projection

- Using the metric tensor relation from the previous step, the Jacobian $\sqrt{g}$ for the equiangular projection is derived analytically:

$$\sqrt{g} = \frac{R^2}{\rho^3 \cos^2\alpha \cos^2\beta}, \quad \text{where } \rho^2 = 1 + \tan^2\alpha + \tan^2\beta$$

2. Application: Flux-Form Divergence

- To conserve mass on the sphere, physical fluxes must be scaled by $\sqrt{g}$ before applying the differentiation operator on the reference element:

$$\nabla \cdot (\mathbf{u}\phi) = \frac{1}{\sqrt{g}} \left[\frac{\partial}{\partial\alpha}(\sqrt{g}u^1\phi) + \frac{\partial}{\partial\beta}(\sqrt{g}u^2\phi) \right]$$

Numerical Formulation

Numerical Implementation: Skew-Symmetric Form

1. Vector Formulation

To ensure numerical stability and minimize aliasing, we use the Skew-Symmetric form instead of the standard flux divergence:

$$\mathbf{u} \cdot \nabla \phi = \frac{1}{2} \underbrace{\nabla \cdot (\phi \mathbf{u})}_{\text{Flux}} + \frac{1}{2} \underbrace{\mathbf{u} \cdot \nabla \phi}_{\text{Advective}} - \frac{1}{2} \underbrace{\phi (\nabla \cdot \mathbf{u})}_{\text{Correction}}$$

2. Component Expansion (Computational Space)

Expanding terms using curvilinear coordinates (α, β) and Jacobian $\sqrt{g}$:

$$\text{Flux Term} = \frac{1}{\sqrt{g}} \left[\frac{\partial}{\partial \alpha} (\sqrt{g} u^1 \phi) + \frac{\partial}{\partial \beta} (\sqrt{g} u^2 \phi) \right]$$

$$\text{Advective Term} = u^1 \frac{\partial \phi}{\partial \alpha} + u^2 \frac{\partial \phi}{\partial \beta}$$

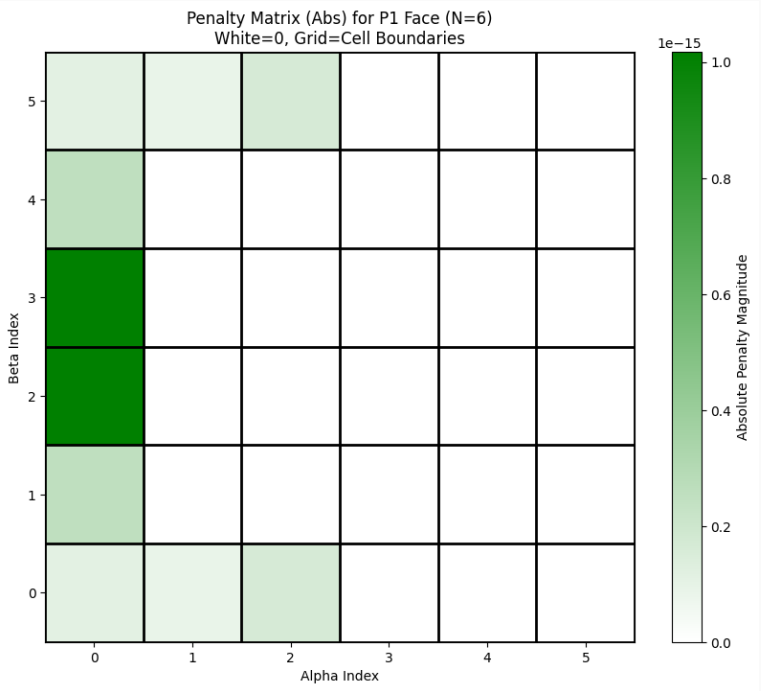
$$\text{Correction Term} = \phi \left\{ \frac{1}{\sqrt{g}} \left[\frac{\partial}{\partial \alpha} (\sqrt{g} u^1) + \frac{\partial}{\partial \beta} (\sqrt{g} u^2) \right] \right\}$$

u^1, u^2: Contravariant velocity components defined previously.

The Penalty Term (SAT)

Concept:
Instead of imposing hard boundaries, we add a **SAT penalty term** to the RHS. **Formula (inflow only, $V_n < 0$):**

$$\text{SAT} = \frac{1}{w}\frac{V_n - |V_n|}{2}(\phi_{\text{self}} - \phi_{\text{neighbor}})$$



where:

- ϕ_{self} is the interior solution.
- ϕ_{neighbor} is the exterior / boundary data.
- w is the boundary metric weight.
- V_n is the outward normal velocity.

Semi-Discrete Formulation (The Full System)

To solve for the time evolution, we combine the **skew-symmetric spatial operator** with the **SAT boundary/interface penalty**.

Semi-discrete ODE at node (i, j):

$$\frac{d\phi_{ij}}{dt} = -\left[\mathbf{u} \cdot \nabla\phi\right]_{ij} + \mathrm{SAT}_{ij}$$

SAT term (sum over faces):

$$\mathrm{SAT}_{ij} = \frac{1}{w_\xi J} \sum_{f \in \text{faces}} \frac{1}{2}(\mathbf{V}_n - |\mathbf{V}_n|)(\phi_{\text{self}} - \phi_{\text{neighbor}})$$

Next Step: This RHS is explicitly integrated in time using the **LSRK54** scheme.

Time Integration: LSRK Scheme

To advance the semi-discrete system $\frac{d\phi}{dt} = \mathbf{RHS}$, we employ a **Low-Storage Runge-Kutta (LSRK)** scheme.

Scheme Details (Carpenter & Kennedy, 1994):

- **Order:** 4th-order accurate (matches high-order spatial accuracy).
- **Stages:** 5 stages per time step (LSRK54).

Time Step Selection:

$$\Delta t = \frac{\text{CFL}}{2\nu_{\max}} \frac{2}{N^2}$$

Note that the time step scales with $O(N^{-2})$ due to the clustering of LGL nodes near the element boundaries.

Validation & Results

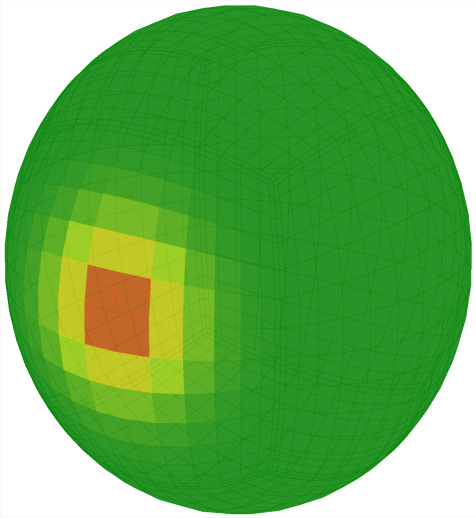
Test Case: Solid Body Rotation

Setup:

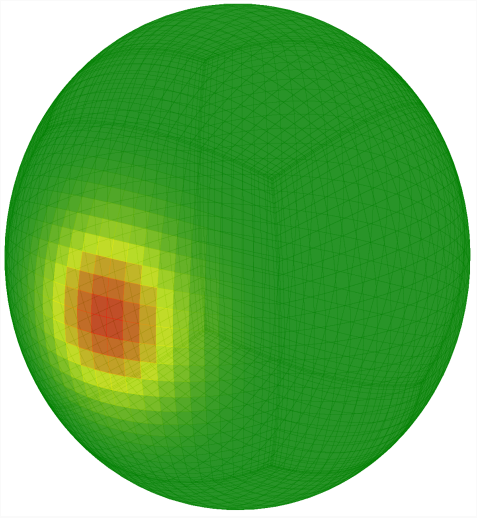
- Advection of a Gaussian bell. $\phi = h_0 \exp\left(-\left(\frac{r_d}{r_0}\right)^2\right)$
- Velocity $u_0 = 2\pi$ (one full revolution in $T = 1.0$).
- The angle between the axis of solid-body rotation and the polar axis $\alpha_0 = -\pi/4$ (flows over corners/edges).

Visual Results for varying N:

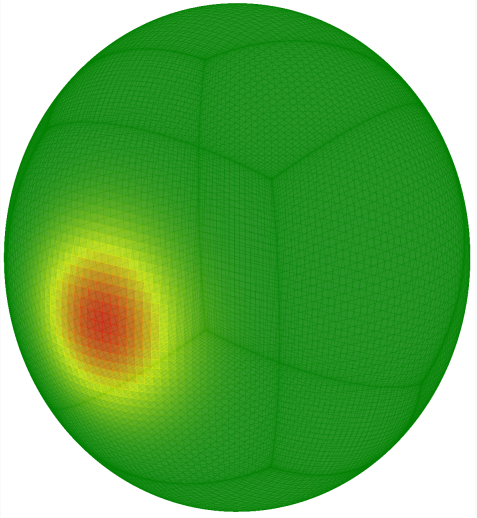
$N = 16$



$N = 32$



$N = 64$



Convergence Table & Analysis

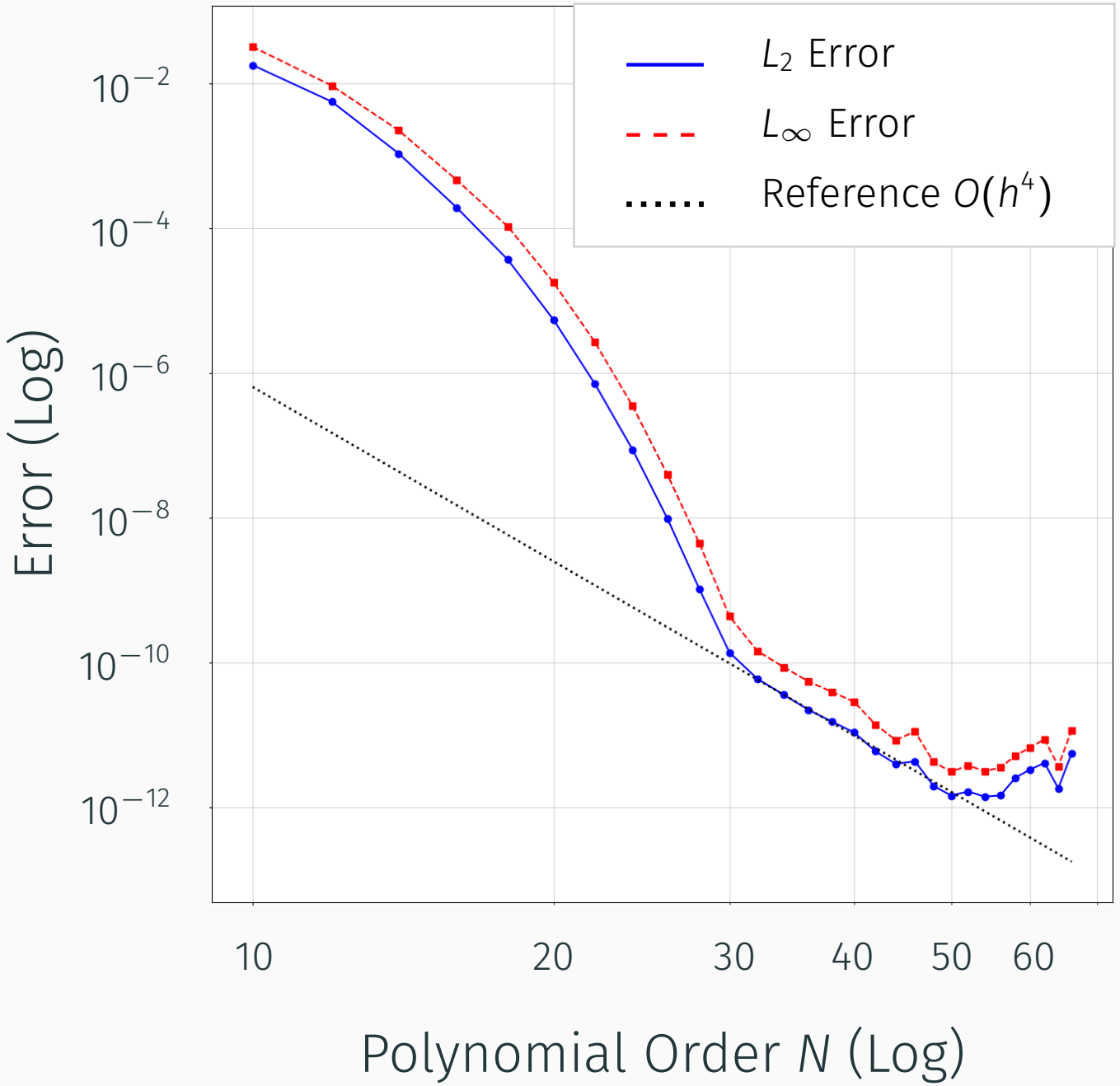
Regime Analysis:

- **Spectral Regime ($10 \leq N \leq 30$):** Spatial discretization error dominates. The steep slope indicates exponential convergence typical of spectral methods.
- **Temporal Saturation ($N > 32$):** Time-stepping error dominates. The spatial error is so small that the total error is now limited by the 4th-order accuracy of the LSRK scheme ($O(h^4)$).

Error and Convergence Rate:

N	L2 Err.	L2 C.R.	Linf Err.	Linf C.R.
10	1.77e-02	–	3.18e-02	–
12	5.55e-03	3.19	9.24e-03	3.39
14	1.07e-03	5.34	2.23e-03	4.61
16	1.91e-04	6.44	4.60e-04	5.91
18	3.66e-05	7.02	1.05e-04	6.27
20	5.41e-06	9.07	1.77e-05	8.45
22	7.04e-07	10.70	2.65e-06	9.95
24	8.67e-08	12.03	3.47e-07	11.69
26	9.71e-09	13.67	3.93e-08	13.61
28	1.02e-09	15.21	4.38e-09	14.80
30	1.37e-10	14.55	4.32e-10	16.79
32	5.92e-11	6.50	1.43e-10	8.56
34	3.57e-11	4.17	8.59e-11	4.20
36	2.21e-11	4.18	5.44e-11	4.00
38	1.52e-11	3.49	3.92e-11	3.04
40	1.07e-11	3.36	2.84e-11	3.14
42	5.90e-12	6.13	1.38e-11	7.35
44	3.96e-12	4.27	8.47e-12	5.27

Convergence Rate Plot (Log-Log)



Conclusion

Summary & Conclusion

1. **Framework:** Successfully implemented a High-Order Spectral Element Solver on the Cubed Sphere.
2. **Geometry:** Equiangular projection avoids pole singularities.
3. **Stability:** The **SAT Penalty Term** effectively handles the discontinuous grid interfaces.
4. **Accuracy:** Numerical experiments confirm **spectral convergence**, validating the implementation against theoretical expectations.

References i

- R. D. Nair, S. J. Thomas, and R. D. Loft.
  A Discontinuous Galerkin Transport Scheme on the Cubed Sphere.
  Monthly Weather Review, 133:814–828, 2005.
- Penalty Term Code Implementation Tutorial.
  ***Internal Course Material*, 2025.**