The LMARS based shallow-water dynamical core on the cubed-sphere geometry

Xi Chen[1,2*]

[1]Program in Atmospheric and Oceanic Sciences, Princeton University, Princeton, NJ, USA
[2]Geophysical Fluid Dynamics Laboratory, NOAA, Princeton, NJ, USA

Key Points:

- We present an LMARS-based unstaggered shallow-water model on arbitrary gnomonic cubed-sphere grids
- The solver demonstrates a broad range of diffusion control without any explicit filters
- A newly introduced splash on the sphere test verifies the solver's desirable dispersion properties

---

*300 Forrestal Road, Sayre Hall, Princeton, NJ 08540-6654

Corresponding author: Xi Chen, `xic@princeton.edu`

Abstract

The rapidly increasing computing powers allow global atmospheric simulations with aggressively high resolutions, which challenges traditional model design principles. This paper presents a Low Mach number Approximate Riemann Solver (LMARS) based unstaggered finite-volume model for solving the shallow-water equations on arbitrary gnomonic cubed-sphere grids. Using a novel reference line-based grid-generation process, it unifies the representation of arbitrary gnomonic cubed-sphere grid projections and permits high-efficiency 1D reconstruction in the halo regions. The numerical discretization also extends a widely used pressure gradient algorithm with the LMARS viscous term, thus improves the model's stability for various numerical applications. The solver demonstrates a broad range of organic diffusion control without any explicit filters, validated by a comprehensive set of test cases. Lastly, a newly introduced splash on the sphere test verifies the solver's desirable dispersion properties and consistent performance among different grid types. This study paves a solid foundation for a new generation of global circulation models with kilometer horizontal scales.

Plain Language Summary

Computing powers and architectures historically influence the numerical algorithm designs of global atmospheric simulations at the fundamental levels. The next generation of the global circulation models can push its resolution to kilometer horizontal scales, which requires vital capabilities in a balanced representation of all motion modes and handling sharp discontinuities such as topography. Here we demonstrate a new framework of a dynamical core that inherits advantages in both traditional geophysical fluid (GFD) modelings and versatile general computational fluid dynamics (CFD) techniques. This new development introduces several innovations, including the unified grid description, numerical optimization, stability enhancement, and a newly designed test to illustrate specific numerical properties cleanly. The desirable results of this work increase our confidence in creating a unique global circulation model to leverage next-generation high-performance computing and to improve our fundamental understanding of the atmospheric processes.

1 Introduction

Building a dynamical core for the atmospheric model is an art of balancing the requirements between accuracy and efficiency. Although a model's accuracy can be more definitive to measure in benchmark experiments - at least for smooth solutions, computational efficiency is a relative concept that strongly associates with the computing platform characteristics. In a way, the available computational power and machine architectures have been dictating the scope of the research topics and the design of the numerical simulation. For example, coming out of the wide adoption of the massively parallel computing, many modern models prefer quasi-uniform computational grids, such as the icosahedral (Ringler et al., 2000; Tomita et al., 2001; Du et al., 2003; Rípodas et al., 2009) and cubed-sphere grids (Adcroft et al., 2004; Putman & Lin, 2007), over the traditional latitude-longitude grids to evenly distribute the computation tasks over a large number of processors. However, the concepts of both icosahedral and cubed-sphere grids were explored very early in Sadourny et al. (1968); Sadourny (1972); Williamson (1968), but gained very little attention. During this time, single-processor-based latitude-longitude or Gaussian spectral methods demonstrated excellent efficiency in delivering accurate results (Bourke, 1972, 1974) - a perfect example of the evolving standard of the numerical algorithm efficiency.

The essential motivation of implementing the quasi-uniform grids is to transform the severe two parallel-unfriendly singular points from the latitude-longitude grid (the north and south poles) to less severe eight singularities in the cubed-sphere grid (corners of the six cubed-sphere tiles) or twelve singularities in the icosahedral grid (pentagon cells in a Voronoi perspective). Although the quasi-uniform grids avoid polar filters and other numerical damping schemes due to the polar singularities, they have both pros and cons. In some cases, the numerical artifacts and errors caused by the twelve icosahedral singularities are less severe than the ones caused by cubed-sphere corners. The cubed-sphere grid, however, has several desirable properties. Firstly, many attractive high-order multi-dimensional algorithms achieved optimal computational efficiency by taking advantage of logically Cartesian grids (i.e., rectangles) on each cubed-sphere tile (Rossmanith, 2006; Putman & Lin, 2007; C. Chen & Xiao, 2008; Ullrich et al., 2010; Taylor & Fournier, 2010). Secondly, the logical 3D cubic geometry on each cube-tile creates one additional dimension to the computational data structure, which offers more flexibility in the parallel computational optimization designs in both horizontal and vertical directions. Lastly, each tile of the cubed-sphere geometry has a significant similarity with computational grids in regional models. In fact, with separately defined metric terms, the global cubed-sphere dynamical core requires almost no code change to convert into a regional solver in orthogonal grids. This property provides exceptional convenience and flexibility in unifying real-world applications and experimenting new algorithms with idealized validation tests.

The global circulation models (GCMs) cover a vast range of scales and operate on the most powerful high-performance computers available. Therefore, geophysical fluid models usually discretize the governing equations with various grid staggering choices (Arakawa & Lamb, 1977, namely A-, B-, C-, D-Grid) to gain numerical advantages in resolving the smallest waves. In particular, the C-Grid results in natural pressure gradient in the momentum equations and straightforward divergence representation (Adcroft et al., 2019); the D-Grid forms a perfect vorticity mode (Lin, 2004; Harris & Lin, 2013), which dominants large-scale to mesoscale atmospheric motions. With the recent rapid increase of computational power, the GCM community is pushing the model resolution aggressively to about a globally-kilometer horizontal scale (Stevens et al., 2019; Satoh et al., 2019). At this scale, both divergence and vorticity play equally important roles. Therefore, the unstaggered discretization could be a balanced choice in global kilometer-scale model development.

Although most GCMs use staggered discretization, a few influential models utilize the A-Grid, including NICAM (Tomita et al., 2001) with a stencil-based scheme, and the E3SM (Taylor et al., 2020) with a compact Spectral-Element-based algorithm. Colocating the velocity components and scalars with a stencil-based scheme has several unique and attractive advantages. It allows direct coupling between the explicitly simulated dynamical process and parameterized physics processes. No interpolation of the velocity components is required, therefore, eliminating errors associated with such practices. Furthermore, it can yield energy conservation in non-hydrostatic models. With colocated prognostic variables, it is possible to formulate a flux-form prognostic total-energy governing equation, thus automatically provide energy conservation to the models. Energy conservation is beyond the scope of the shallow-water implementation and will be extensively discussed in future work. Preliminary work has demonstrated promising results in Li and Chen (2019). Last but not least, unstaggered schemes are widely implemented in traditional computational fluid dynamics applications. Many well-tested techniques can inspire the creation of algorithms in the geophysical fluid simulations.

Despite many efforts, historically, the unstaggered discretization was overwhelmed by staggered methods due to, at least, two primary considerations. One challenge is that compared with the staggered schemes, the unstaggered discretization tends to produce severe errors in resolving phase speed for waves with very short wavelengths. Additionally, it is more challenging to design numerically robust algorithms with an unstaggered grid. For example, an unstaggered central-differencing of gradient term could produce a so-called "grid decoupling" problem and lead to checkerboard-pattern noise. Indeed, the long-term success of an unstaggered model depends on the proper treatment to address the above difficulties.

To address the first concern, the previous installment of this work thoroughly investigated the dispersive and dissipative relations between different grid staggering choices and orders of accuracy (X. Chen et al., 2018). One of the principal findings is that with high-order algorithms, e.g., using 3-points or 5-points stencil polynomial schemes, waves with problematic phase speeds will be largely pushed to four-grid-spacing and below. On the other hand, in practice, the smallest resolvable wavelengths are contaminated by many sources of errors, such as numerical diffusion, strong gradients in the solutions, and, accordingly, heavily damped or removed by numerical techniques. Therefore, using high-order numerical schemes can prevent the unstaggered model from problematic phase speeds. X. Chen et al. (2018) also introduced simple-to-setup tests to validate the dispersion and dissipation properties of any sophisticated solvers.

Unstaggered algorithms for geophysical flows can inherit various robust approaches from other fields. Unlike the geophysical fluid modeling field, the general computational fluid dynamics (CFD) community widely adopts unstaggered algorithms via the implementation of the approximate Riemann Solvers. There is a small but growing literature that takes advantage of the Riemann Solvers in the A-Grid GCMs and achieved stability, high order accuracy, and are free of explicit diffusion (Giraldo et al., 2002; C. Chen & Xiao, 2008; Ullrich et al., 2010; Yang et al., 2010). However, compared with state-of-science GCMs, the traditional Riemann solvers can be less efficient or more diffusive. Moreover, the Riemann solvers with carefully designed mathematical expressions are most effective in simulating generic flow types, including the vacuum, shock, sharp discontinuities. Therefore, it is not easy to understand the Riemann solvers' inherent numerical properties and make the comparison to traditional GCM implementations. Lastly, traditional Riemann solvers require careful adjustments to make them accurate on the unique geometry on the Earth's surface with the gravity and the Coriolis forces.

The Low Mach number Approximate Riemann Solver (LMARS) introduced in X. Chen et al. (2013) is a highly efficient tool in the finite-volume method based GCMs. The design of LMARS bases on the fact that the geophysical flows do not create any vacuum or sharp discontinuities and the subsonic flows Mach number is always less than 1. The resulting simple mathematical expression of LMARS requires only one approximation, which assumes the sound wave speed (or gravity wave speed in a shallow-water model) is continuous at the finite volume interfaces. The LMARS discretization in a full 3D atmosphere takes the gravity into account and results in better accuracy than traditional Riemann solvers (Li & Chen, 2019). Previous work has implemented LMARS in many atmospheric applications, including hydrostatic and non-hydrostatic flows with either vertical Lagrangian coordinates or Eulerian coordinates (X. Chen et al., 2013). It is also implemented in the multi-gas planet atmosphere environment with an intrinsic energy-conserving framework (Li & Chen, 2019). As a promising candidate for a new unstaggered GCM, LMARS is not yet tested with a cubed-sphere geometry.

This study aims to develop an LMARS-based unstaggered finite-volume shallow-water solver on the gnomonic cubed-sphere grids. The shallow-water equations offer a standard testbed to validate the horizontal advection algorithms. Whereas previous studies implemented various cubed-sphere grids in different solvers, this work is the first to unify arbitrary gnomonic cubed-sphere grid generation processes. Although the numerical discretization of LMARS on the cubed-sphere grid does not contain any explicit filter, this solver can still exhibit a broad range of diffusion properties by controlling the polynomial reconstructing methods and the strength of the forward-backward techniques (Mesinger, 1977). Lastly, various model numerical properties are illustrated and gauged by a comprehensive set of idealized tests, including the traditional famous Williamson et al. (1992) shallow water test suite, and a recently introduced colliding modons test (Lin et al., 2017). Given the traditional discussions about unstaggered grid dispersion uncertainties, this study designs a new "splash on the sphere" test to illustrate and warrant satisfactory dispersion and dissipation properties on specific wavelengths.

The remainder of the paper is organized as follows. Section 2 provides a brief primer of the gnomonic cubed-sphere grids, in which a unified grid generation process is introduced. Section 3 describes both spatial and temporal numeric discretization. The model is validated in section 4. In this section, a novel splash on the sphere test is introduced to gauge the dispersion and dissipation properties. Finally, the main findings and discussions are concluded in Section 5. Appendix A collects all major symbols to make the notations consistent and clear. Appendix B provides a mathematical description of the grid generating process. The governing equations in the cubed-sphere grids are provided in Appendix C. In this section, some optimizations are discussed to yield more efficient mathematical expressions.

2 A brief primer on gnomonic cubed-sphere grids and the duo-grid system

The FV3 (Finite-Volume Cubed-Sphere Dynamical Core) (Lin, 2004; Putman & Lin, 2007; Harris & Lin, 2013) has strongly influenced the development of this work. Although being different in staggering choices, the new unstaggered dynamical core is designed to be a seamless evolution within the existing FV3 framework. Therefore, this work follows a significant amount of naming conventions from the existing FV3 code base, including grid type names and dynamical parameter definitions and symbol names.

The cubed-sphere grid is obtained from projecting a gridded cube onto the surface of the sphere, which avoids the polar singularities due to the convergence of the meridians from the traditional latitude-longitude coordinate system. The cubed-sphere grid shifts the severe polar convergence of meridians to eight weaker singularities at the corners where cube tiles intersect. Figure 1 illustrates the mapping from the cube to the sphere in a C8 resolution. The naming convention C[N] denotes that each tile of the cube-sphere has N by N cell distribution, which results in $N \times N \times 6$ total cells on the sphere.

2.1 Interlock patterns

There are two conventional logical arrangements to interlock the six cubed-sphere tiles. Figure 2a illustrates the first pattern (Ronchi et al., 1996; Rossmanith, 2006; Yang et al., 2010) with four tropical tiles and two polar tiles, denoted by the “tropic-belt” logical arrangement. Unfortunately, the interlock between the tiles is not symmetric and

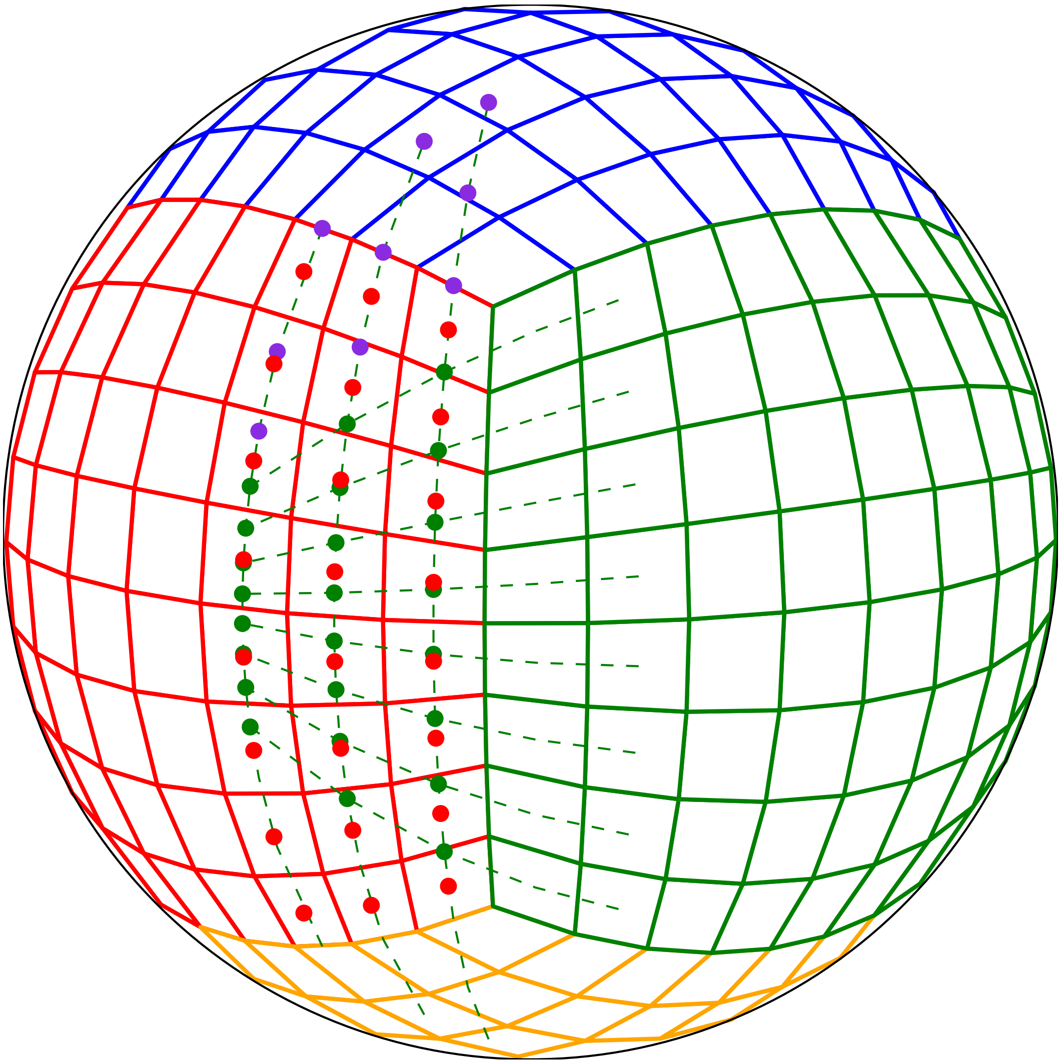


Figure 1. C8 (8 × 8 × 6) cubed-sphere grid with three layers of ghost cells. When aligned on the same great circle, a simple 1-D polynomial interpolation from the red dots can provide the green dots' values with optimum accuracy and speed.

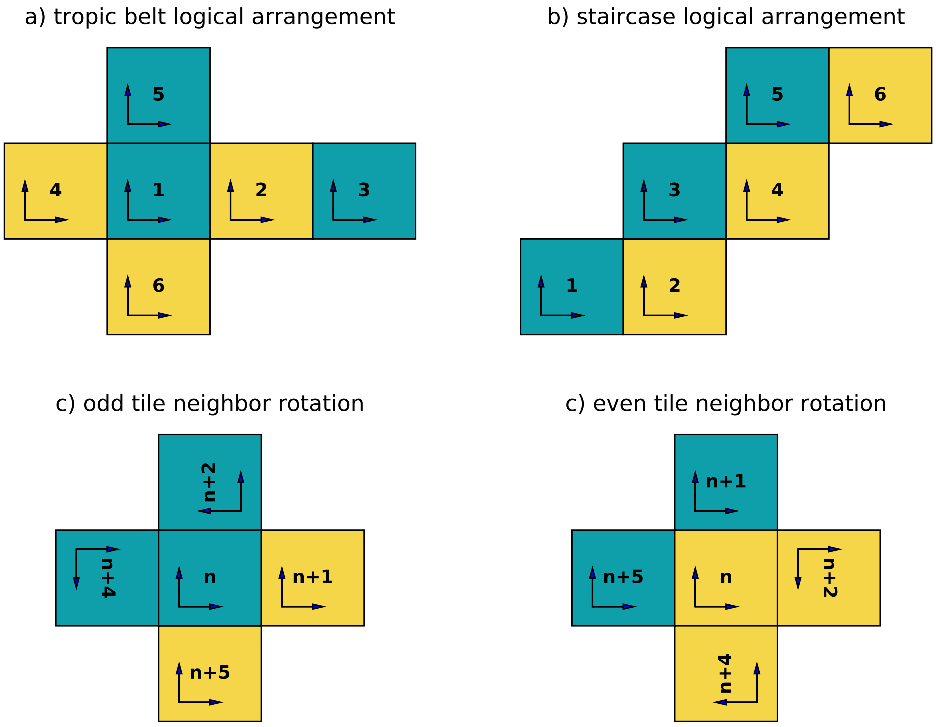

a) tropic belt logical arrangement
5
4
1
2
3
6
b) staircase logical arrangement
5
6
3
4
1
2
c) odd tile neighbor rotation
n+2
n+4
n
n+1
n+5
c) even tile neighbor rotation
n+1
n+5
n
n+2
n+4

Figure 2. The logic arrangement of the six cubed-sphere tiles. a) the "tropic belt" arrangement has four tropic tiles and two polar tiles, the local coordinate directions are as illustrated at the lower-left corners. b) the "staircase" arrangement and the local coordinates illustration. The "staircase" arrangement simplifies the determination of the neighbor tile indices and orientations into c) odd-index-tile and d) even-index-tile cases. Note if the number $n + l > 6$, the corresponding tile number is $n + l - 6$.

require separate distinction for each of the tile. The second pattern (Adcroft et al., 2004) is illustrated in Figure 2b, and denoted by the "staircase" logical arrangement. The "staircase" logical arrangement has better symmetry when exchanging information between tiles. For example, the calculation of fluxes between two adjacent tiles needs to determine the tile numbers and the tile-local coordinates rotations. The "staircase" arrangement simplifies the neighbor tile-number and rotation patterns into two odd-index-tile (tiles 1, 3, 5) and even-index-tile (tiles 2, 4, 6) scenarios, illustrated by Figures 2c, and d. Both logical arrangements are valid for cubed-sphere applications. This work implements the "staircase" arrangement to gain some programming simplicities.

2.2 Unified gnomonic projections

The grid lines connecting the grid points are continuous great circles on the sphere. Therefore, the locations of vertices can uniquely determine the full grid system on each cube-sphere tile. On each tile, e.g., the tile centered at $(\lambda, \phi) = (0, 0)$, the vertices can be generated by either the gnomonic (Sadourny, 1972; Ronchi et al., 1996) or the conformal (McGregor, 1996; Rančić et al., 1996) projections. The gnomonic projection projects a Cartesian grid from six straight-lines-meshed cube surfaces to the sphere surface. The conformal projection maximizes the orthogonality of the coordinates. Putman and Lin (2007) examined the most popular approaches: the equidistant projection (Sadourny, 1972), the equiangular projection (Ronchi et al., 1996), the more orthogonal conformal mapping (Rančić et al., 1996), the numerical modification to analytical mappings by an elliptic solver (Khamayseh & Mastin, 1996) or the spring dynamics generator (Tomita et al., 2001). Considering the eight singularities at cube corners remain nonorthogonal in the conformal grid, and the cell size distribution in conformal grids is usually less uniform than the gnomonic choices, this work implements the gnomonic grid.

Figure 3 illustrates a typical gnomonic projection between a grid point on the cube-tile and the sphere-tile. Let the grid points on a cube-tile be indexed by $[i, j]$, with local coordinate $[X, Y]$. In a gnomonic projection, each row of grid points $[i, :]$ shares the same Y value, and each column of grid points $[:, j]$ shares the same X value. The projection of the rows and columns on each cube face forms great circles on the sphere. Furthermore, the coordinates on the sphere converge to a pair of local north-/south-poles, and "west-/east-poles" for each sphere tile (Rossmanith, 2006, Figure 1).

This work offers a novel approach to unite generic gnomonic grid descriptions. Although previous work demonstrated different ways to create various gnomonic grids, the gnomonic grids share distinct properties. Given the orthogonality and symmetry of the gnomonic projected grid points on each cube-tile, the locations of grid points in a single row or column can fully determine the entire gnomonic cubed sphere. Therefore, Figure 3 can determine three widely used gnomonic projections in the literature by three pairs of reference lines with grid points equally distributed. The red reference lines stand for the most traditional equidistance projection (Sadourny, 1972). The equiangular grid (Ronchi et al., 1996) can be obtained by projecting the green lines with grid points equally distributed back to the cube tile and populating them to the entire $[X, Y]$ space. Lastly,

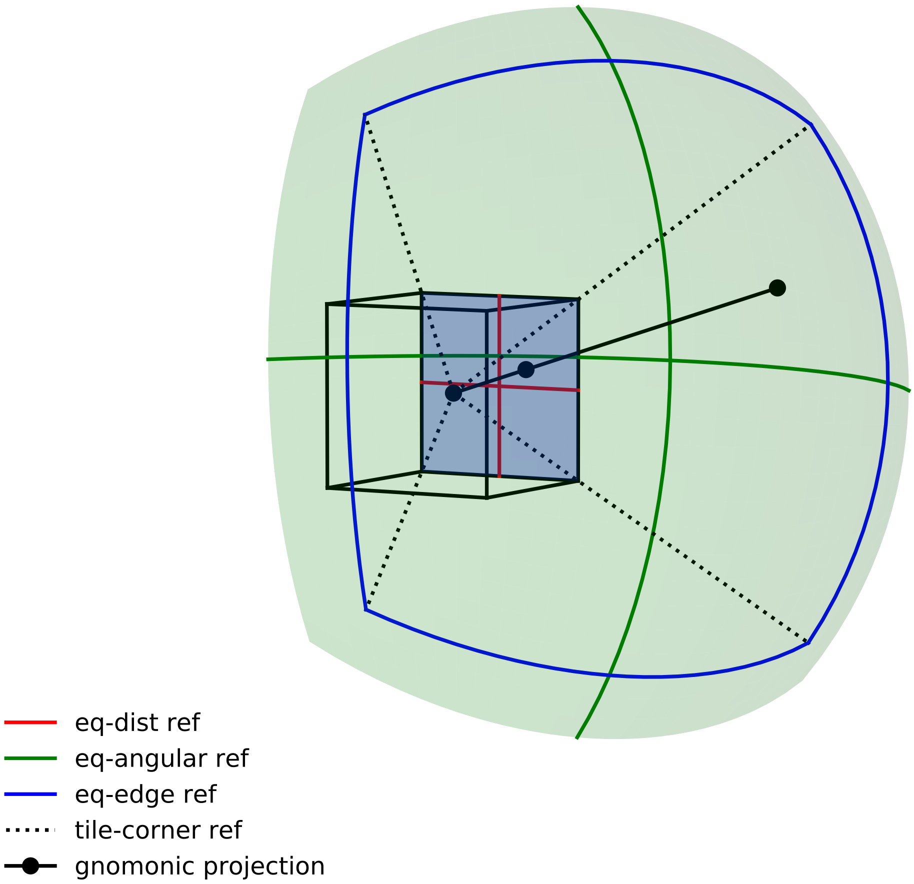


Figure 3. The gnomonic projection. The blue shaded surface is a ***cube-tile***, and the green shaded surface is a ***sphere-tile***. The solid black line from the sphere center to the sphere surface illustrates a ***gnomonic projection*** between a grid point on the cube-tile and a grid point on the sphere-tile. The dotted lines define the corners of the bounds to one of the six surfaces of the cubed-sphere grid. The red lines are the ***gnomonic projection reference lines*** of the equidistance projection, and the green and blue lines are of the equiangular and ***equi-edge*** projections. The extension of the green surface indicate the ghost cell region for a finite-volume application.

although not formally documented, FV3 (Putman & Lin, 2007; Harris & Lin, 2013; Harris et al., 2016) introduced a gnomonic projection with more uniformly distributed cells on the tile interfaces, and is now adopted in the Next Generation Global Prediction System (NGGPS) project (Zhou et al., 2019). The FV3 grid is denoted by the "equi-edge" grid and can be obtained similarly by populating the grid-points-equally-distanced blue lines on the sphere. The following procedures describe the process to populate the reference line grid points to the full gnomonic projections:

1. Project the equally distributed grid points from the reference lines onto the gray-color cube-tile (this procedure is redundant for the red line).
2. Mesh the gray cube-tile with the projected 1D distribution of the grid points from the cube surface.
3. Project the fully meshed grid points from the gray cube tile to the green sphere tile.

The equidistance grid produces less uniformly distributed volumes on the sphere. Therefore, most modern cubed-sphere models do not implement the equidistance grid. Although the equiangular projection yield slightly more uniform grid point distribution on the sphere, the equi-edge grid distributes grid points evenly near the sphere tile connections, which are the primary sources that cause grid imprinting with a cubed sphere. Additionally, in a stretched grid (Harris et al., 2016), the equi-edge grid produces the focused tile with more evenly distributed cells. This work only discusses the equi-edge and the equiangular grids, namely grid_type 0 and 2, following FV3 naming convention.

2.3 The duo-grid system to handle tile-edge connections

This work implements a maximum 5-point-stencil polynomial reconstruction for the finite volume scheme. Therefore each tile requires three more layers of ghost cells by the wave propagation method (X. Chen et al., 2013, 2018; Li & Chen, 2019). Unfortunately, a main numerical challenge with the gnomonic cubed-sphere grid is that the grid coordinates are not continuous across tile interfaces. Figure 1 illustrates the ghost cells on the local west-side of the green tile, which creates a west halo region. The native neighbor cell centers in the halo region are the red dots, which forms "kinks" in coordinates between the green tile and its west halo. Therefore, the name "kinked grid" denotes this halo type constructed by directly copying values from the neighboring tiles. A natural extension of the green tile coordinates (green dashed lines) into the halo region results in green dots, denoted by the “extended grid.”

There are two reasonable choices for the ghost cell numerical algorithms. FV3 powered models, for example, directly employ the red dots in the kinked-grid to form one-sided flux calculations to the tile boundaries. This method has several advantages. The operations in the halo region (e.g., west halo of the green tile) can be identical to the native calculations on the neighboring tile (red tile) from different calculating processors (Putman & Lin, 2007; Harris & Lin, 2013). Additionally, calculations associated with halo create almost no computational overhead, a valuable property to massively parallel computing. This work implements the other approach, which is to remap the red dots from the kinked-grid to the green dots on the extended grid (Rossmanith, 2006; Yang et al., 2010; Ullrich et al., 2010; Katta et al., 2015). Therefore, the halo operations are a natural extension from the interior calculations, and no extra adjustment is required for the ghost cells. The second approach can effectively reduce the cubed-sphere grid imprinting at the tile edges, but creates undesired overhead in the parallel calculations. The extra calculations are mainly: 1. tile-interface flux synchronization between different tiles; 2. remapping of the prognostic variable values from the kinked grid to the extended grid.

The first synchronization overhead is due to inconsistent coordinate directions between tiles. The fluxes at the same interfaces by different tiles need to be sent to the adjacent tile and get averaged for conservation properties. This operation is only performed at the end of a full cycle of integration to minimize the message passing.

The interpolating algorithm must be at minimum complexity to alleviate the duo-grid remapping overhead. Most equiangular projection applications take advantage of that the green dots and red dots are aligned on the same great circle. Thus 1-D interpolation is sufficient for the remapping procedure. This property is denoted by duo-grid 1D alignment. Unfortunately, this 1D alignment only applies to equiangular projection. A direct halo generation with equidistance or equi-edge projections (i.e., by extending three more layers into the halo during the interior grid creation) does not produce the duo-grid 1D alignment. A few extra steps by the following procedures can solve this problem and warrant the duo-grid 1D alignment for arbitrary gnomonic projections:

1. Populate the non-halo gnomonic projection grid points based on the reference lines (red for equidistance, blue for equi-edge, and green for equiangular).

2. Identify the grid points on the resulting sphere tile center lines (i.e., the green lines in Figure 3).
3. Mirror the outermost three points from the green lines to the ghost region.
4. Populate ghost cell grid points in the halo region base on the mirrored points on the green lines.

To minimize the remapping overhead, in this work, the minimum 1D piecewise linear interpolation is used to remap values from the red to the green points. No discernible degradation is observed compared with high-order 1D polynomial remapping algorithms. Detailed mathematic procedures in the gnomonic cubed-sphere generation are described in Appendix B.

3 The numerical discretization

3.1 The governing equations discretization

A standard mathematical derivation of the governing equations on the cubed-sphere is included in Appendix C. Before rearranging and discretizing the governing equations, denote η an arbitrary variable, the following finite-volume (FV) operators are defined:

$$\delta_x \left[\eta\right]_{i,j} = \eta_{i+0.5,j} - \eta_{i-0.5,j}, \tag{1}$$

$$\delta_y \left[\eta\right]_{i,j} = \eta_{i,j+0.5} - \eta_{i,j-0.5}, \tag{2}$$

where $\eta_{i\pm0.5,j}$ are the values of η evaluated at cell interfaces in the x-direction, and $\eta_{i,j\pm0.5}$ are the values of η evaluated at cell interfaces in the y-direction. Unless specifically specified, the subscripts (i,j) are omitted for cleaner mathematical expressions, and the variables without subscripts are evaluated at the cell centers at location (i,j).

The continuity equation can be rearranged in the flux form:

$$\frac{\partial h}{\partial t} = -\frac{1}{\Delta A}\left(\delta_x\left(hu_\perp^x \Delta y\right) + \delta_y\left(hu_\perp^y \Delta x\right)\right), \tag{3}$$

Define the flux coefficient:

$$f^x_{i+0.5,j}\left(u_\perp^x, \Delta t\right) = u_\perp^x \Delta t \Delta y_{i+0.5,j}, \tag{4}$$

$$f^y_{i,j+0.5}\left(u_\perp^y, \Delta t\right) = u_\perp^y \Delta t \Delta x_{i,j+0.5}. \tag{5}$$

Therefore, the final discretization of the continuity equation is:

$$h^{n+1} = h^n - \frac{1}{\Delta A}\left(\delta_x\left(f^x h\right) + \delta_y\left(f^y h\right)\right), \tag{6}$$

and a straightforward discretization to update the cell center velocity components is:

$$u^{n+1} = u^n + \left(\sqrt{g}\hat{v}\left(\zeta + f\right) - \frac{1}{\Delta x}\delta_x\left(\Pi + \Phi_s + K\right)\right)\Delta t, \tag{7}$$

$$v^{n+1} = v^n + \left(-\sqrt{g}\hat{u}\left(\zeta + f\right) - \frac{1}{\Delta y}\delta_y\left(\Pi + \Phi_s + K\right)\right)\Delta t, \tag{8}$$

The relative vorticity ζ at cell centers can be numerically evaluated by:

$$\zeta = \frac{1}{\Delta A}\left(\delta_x\left(u_{\parallel}^{x}\Delta y\right) - \delta_y\left(u_{\parallel}^{y}\Delta x\right)\right), \tag{9}$$

and the kinetic energy K can be numerically evaluated at cell interfaces by:

$$K = \frac{1}{2}\left(u_{\perp}u_{\perp} + u_{\parallel}u_{\parallel}\right). \tag{10}$$

Note that K or $u_{\perp}$, $u_{\parallel}$ are all defined at the cell interfaces, and the labels in x and y directions are omitted.

All cell-center or cell-averaged variable values are straightforward to calculate. The next step is to determine the values at the cell interfaces using an efficient approximate Riemann solver and evaluate the terms in the δ_x and δ_y operators.

3.2 Calculation of cell-interface values by LMARS

The remaining variables to be determined at the cell interfaces are: $u_{\perp}$, h, Π, Φ_s, and K (i.e., $u_{\perp}$, $u_{\parallel}$).

Following the naming conventions in X. Chen et al. (2018), $\eta_{i,j}^{W}$ and $\eta_{i.j}^{E}$ denote the "west" and "east" volume boundary values of arbitrary variable η calculated by polynomial reconstruction within volume (i,j). Therefore, the mismatching $\eta_{i,j}^{E}$ and $\eta_{i+1,j}^{W}$ are evaluated at two sides of the interface between cell (i,j) and $(i+1,j)$. The first step of LMARS is to evaluate cell interface values of Π and $u_{\perp}$. In the x-direction:

$$\left(u_{\perp}^{x}\right)_{i+0.5,j} = \frac{1}{2}\left(\left(u_{\perp}^{x}\right)_{i,j}^{E} + \left(u_{\perp}^{x}\right)_{i+1,j}^{W}\right) + \frac{1}{2a_{i+0.5,j}}\left(\Pi_{i,j}^{E} - \Pi_{i+1,j}^{W}\right), \tag{11}$$

$$\Pi_{i+0.5,j} = \frac{1}{2}\left(\Pi_{i,j}^{E} + \Pi_{i+1,j}^{W}\right) + \frac{a_{i+0.5,j}}{2}\left(\left(u_{\perp}^{x}\right)_{i,j}^{E} - \left(u_{\perp}^{x}\right)_{i+1,j}^{W}\right), \tag{12}$$

where a is the gravity wave speed (group velocity) estimated at cell interface:

$$a_{i+0.5,j} = \sqrt{\frac{1}{2}\left(\Pi_{i,j}^{E} + \Pi_{i+1,j}^{W}\right)}. \tag{13}$$

Once the velocity normal to the cell interface determined, the values of the variables to be transported are chosen using upwind values:

$$h_{i+0.5,j} = \begin{cases} h^E_{i,j} & \text{if } (u^x_\perp)_{i+0.5,j} > 0 \\ h^W_{i+1,j} & \text{else} \end{cases}, \tag{14}$$

$$\left(u^x_\parallel\right)_{i+0.5,j} = \begin{cases} \left(u^x_\parallel\right)^E_{i,j} & \text{if } (u^x_\perp)_{i+0.5,j} > 0 \\ \left(u^x_\parallel\right)^W_{i+1,j} & \text{else} \end{cases}, \tag{15}$$

and the operations on the y-direction is symmetric and analogous.

Although algebraically h and Π are interchangeable with the relation $\Pi = Gh$, they play different roles in the governing equation. Π stands for pressure forcing term, and h is associated with material transportation. Therefore, their treatments are different in the LMARS solver. In a fully compressible model, their corresponding terms are the density and pressure (X. Chen et al., 2013; Li & Chen, 2019).

The last piece for a single sub-cycle update is to reconstruct the prognostic variables' values at four horizontal cell edges. Following X. Chen et al. (2018) conventions, assuming an N_c-point stencil polynomial reconstruction for an arbitrary variable η:

$$\eta^W_{i,j} = \sum_{l=1-N_g}^{N_g-1} W^{(l)} \eta_{i+l,j}, \tag{16}$$

$$\eta^E_{i,j} = \sum_{l=1-N_g}^{N_g-1} E^{(l)} \eta_{i+l,j}, \tag{17}$$

$$\eta^S_{i,j} = \sum_{l=1-N_g}^{N_g-1} W^{(l)} \eta_{i,j+l}, \tag{18}$$

$$\eta^N_{i,j} = \sum_{l=1-N_g}^{N_g-1} E^{(l)} \eta_{i,j+l}, \tag{19}$$

where $W^{(l)} = E^{(-l)}$, $N_c = 2N_g - 1$, and N_g is the layers of ghost cells required at the tile edges. The lookup coefficient tables for point-value reconstruction and finite-volume reconstruction from X. Chen et al. (2018) is repeated in Tables 1 and 2 for reference. Although this work does not implement any monotonic filters, most filtering schemes can be considered equivalent to altering the reconstruction coefficient values locally according to the shape of the solutions.

Table 1. The coefficient to calculate "east" side midpoint value on a stencil number of $N_c = 2N_g - 1$ from Point-Value variables

N_g	$E_{pt}^{(-2)}$	$E_{pt}^{(-1)}$	$E_{pt}^{(0)}$	$E_{pt}^{(1)}$	$E_{pt}^{(2)}$
1			1		
2		-1/8	3/4	3/8	
3	3/128	-5/32	45/64	15/32	-5/128

Table 2. the coefficient to calculate "east" side midpoint value on a stencil number of $N_c = 2N_g - 1$ from Volume-Mean variables

N_g	$E_{vm}^{(-2)}$	$E_{vm}^{(-1)}$	$E_{vm}^{(0)}$	$E_{vm}^{(1)}$	$E_{vm}^{(2)}$
1			1		
2		-1/6	5/6	1/3	
3	1/30	-13/60	47/60	9/20	-1/20

3.3 Extension of the pressure gradient algorithm in L97 with LMARS viscous terms and the forward-backward algorithm

Up to this part, the numerical discretization closely follows the original LMARS approach described by X. Chen et al. (2013, 2018), which is sufficient for this shallow water solver. Alternatively, Lin (1997) (L97) introduces an innovative spatial discretization to the pressure gradient term. Although mathematically equivalent to this work, the L97 pressure gradient algorithm considerably simplifies the discretization in a full 3D compressible model and makes the numerical expression backward compatible with a shallow-water solver. One drawback of L97 is that the expression is a geometric realization of the volume interface pressure force integration. Therefore, it lacks the implicit diffusion calculated by a Riemann solver, and extra filters are required to stabilize the model. Fortunately, LMARS simple expression allows the separation of the geometric derivative terms and viscous terms. It is straightforward to extend the L97 pressure gradient algorithm with the LMARS viscous contribution and result in a fast and stable pressure gradient scheme.

As discussed previously, the volume interface pressure force obtained by LMARS described in Eq. 12 can be represented by an averaging term and a viscous term:

$$\Pi_{i+0.5,j} = \bar{\Pi}_{i+0.5,j} + \Pi^{\mathrm{vis}}_{i+0.5,j}, \tag{20}$$

where

$$\bar{\Pi}_{i+0.5,j} = \frac{1}{2}\left(\Pi^{E}_{i,j} + \Pi^{W}_{i+1,j}\right), \tag{21}$$

$$\Pi^{\mathrm{vis}}_{i+0.5,j} = \frac{a_{i+0.5,j}}{2}\left(\left(u^{x}_{\perp}\right)^{E}_{i,j} - \left(u^{x}_{\perp}\right)^{W}_{i+1,j}\right). \tag{22}$$

Therefore, the pressure gradient with the topographic contribution, for example, in the x-direction is discretized:

$$\frac{\partial}{\partial x}\left(\Pi + \Phi_s\right) = \frac{1}{\Delta x}\delta_x\left(\bar{\Pi} + \bar{\Phi}_s\right) + \frac{1}{\Delta x}\delta_x \Pi^{\mathrm{vis}}. \tag{23}$$

Note that X. Chen et al. (2018) explained that the viscous term Π^{vis} in 1D is equivalent to a $2N_g$-order diffusion term, which automatically matches the stencil size of the reconstruction schemes. The same function evaluates the surface geopotential at cell interface to warrant numerical consistency:

$$\left(\bar{\Phi}_s\right)_{i+0.5,j} = \frac{1}{2}\left(\left(\bar{\Phi}_s\right)^{E}_{i,j} + \left(\bar{\Phi}_s\right)^{W}_{i+1,j}\right). \tag{24}$$

The following equivalent expressions are defined to simplify the expression:

$$\eta^{x}_{(\pm,\pm)} \equiv \bar{\eta}_{i\pm0.5,j,k\pm0.5}, \tag{25}$$

and the first term in pressure gradient can be rearranged in L97 format:

$$\delta_x\left(\bar{\Pi} + \bar{\Phi}_s\right) = - \frac{\left(\Phi^{x}_{(-,+)} - \Phi^{x}_{(+,-)}\right)\left(p^{x}_{(+,+)} - p^{x}_{(-,-)}\right) + \left(\Phi^{x}_{(-,-)} - \Phi^{x}_{(+,+)}\right)\left(p^{x}_{(-,+)} - p^{x}_{(+,-)}\right)}{p^{x}_{(-,+)} + p^{x}_{(+,+)} - p^{x}_{(-,-)} - p^{x}_{(+,-)}}. \tag{26}$$

The expression in the y-direction is analogous. With this expression, the SWE and 3D compressible pressure gradient numerical discretization can share the identical code.

Another expansion of the pressure gradient calculation is to implement the forward-backward (FB) technique (Mesinger, 1977) to enhance the stability of the model. X. Chen et al. (2018) conducted several numerical analysis on the FB technique. One of the key findings in the 1D linearized environment is that the FB operation is equivalent to a second-order diffusion term on the velocity components, which is vital for the stability of a single-step scheme. In this model with multi-step schemes, the FB scheme is not a necessary component but can be useful in the future fully coupled 3D model as a diffusion moderator. Considering the FB is almost computationally equivalent to an explicit scheme, a relaxed-FB scheme is implemented by setting a parameter $\beta \in [0,1]$ to the geometric pressure gradient term following Harris and Lin (2013):

$$\delta_x \left(\bar{\Pi} + \bar{\Phi}_s\right) = (1-\beta)\, \delta_x \left(\bar{\Pi} + \bar{\Phi}_s\right)^{n+1} + \beta \delta_x \left(\bar{\Pi} + \bar{\Phi}_s\right)^{n}. \tag{27}$$

Therefore, the parameter β in the FB scheme is an implicit control coefficient of a 2nd-order diffusion term. The response to various beta values will be validated in the test section.

3.4 Temporal discretization

The linearized two-step and three-step Runge-Kutta schemes (rk2 and rk3) are implemented for time integration. In each substep, denote prognostic variable array U and right-hand-side term $\mathrm{RHS}(U, N_g, \beta)$, where the dynamical parameters N_g controls the stencil size of the reconstruction scheme and β controls the strength of the FB algorithm in the pressure gradient evaluation. Therefore, the numerical properties of each subcycle update can be tailored to various purposes. Unless explicitly declared, the dynamical parameters in rk2 scheme is:

$$U^{*} = U^{n} + \frac{1}{2}\Delta t \mathrm{RHS}\left(U^{n}, N_g = 1, \beta = 0\right), \tag{28}$$

$$U^{n+1} = U^{n} + \Delta t \mathrm{RHS}\left(U^{*}, N_g = 3, \beta = 0\right). \tag{29}$$

The default rk2 configuration represents a dynamical core with the least operation counts in each time step and most implicit diffusion.

The default rk3 scheme is:

$$U^{(1)} = U^{n} + \frac{1}{3}\Delta t \mathrm{RHS}\left(U^{n}, N_g = 3, \beta = 1\right), \tag{30}$$

$$U^{(2)} = U^{n} + \frac{1}{2}\Delta t \mathrm{RHS}\left(U^{(1)}, N_g = 3, \beta = 1\right), \tag{31}$$

$$U^{n+1} = U^{n} + \Delta t \mathrm{RHS}\left(U^{(2)}, N_g = 3, \beta = 1\right). \tag{32}$$

The default rk3 configuration represents dynamical core with minimized implicit diffusion.

Although the rk2 scheme cost fewer operations in each full cycle, the rk3 scheme allows larger time steps. In practice, a fully optimized rk2 should still maintain better computational efficiency over rk3, majorly due to the first-order substep is significantly cheaper than the standard high-order substep. The linearized Runge-Kutta schemes can highly optimize the computational memory storage cost since the values of RHS terms in each substep are not retained, and the storage of updated prognostic variables can be reused. Lastly, linearized rk3 only improves the CFL conditions and is 2rd-order accurate in time. Standard high-order accurate Runge-Kutta schemes can be found in X. Chen et al. (2013); Ullrich et al. (2010); C. Chen and Xiao (2008).

4 Tests and results

4.1 Testing Plan

Table 3 introduces the naming convention to label the model configurations for a shallow water test simulation. For example, a run label C48.g2.rk3.b1.00 stands for a simulation of a $48 \times 48 \times 6$ cubed-sphere grid with equiangular grid point distribution, and the discretization configuration is the rk3 scheme described in the previous subsection. The value of the forward-backward parameter is 1, which is fully explicit. As described above, the default rk2 configuration not only uses fully implicit forward-backward pressure gradient evaluation but also minimizes the first substep complexity by using a 1st-order accurate reconstruction scheme. The rk3 configuration implements a fully explicit pressure gradient algorithm and uses 5-point-stencil reconstruction for all three substeps. Therefore, the rk2 and rk3 configurations represent the two extremes of the numerical diffusion and performance properties spectrums, with rk2 at the more computationally efficient, more diffusive end, and vice versa. The diffusivities of other configurations lie between the two ends.

Most conventional tests require the simulation data on a traditional latitude-longitude grid for analysis. Therefore, all cubed-sphere simulation results are remapped by a 1st-order conservative algorithm onto a 144x72 spherical grid, which is about 2.5-degree grid spacing to minimize regridding introduced artifacts.

The validation of this model contains three stages. The first stage employs the widely adopted Williamson et al. (1992) shallow-water test suite (W92) to match the convergence of the results to the literature. The smooth solutions in W92 also allow the validations of the order of accuracy and model's overall response to diffusion. The sensitivities of the model-specific parameters, such as grid choices, discretization configura-

Table 3. Naming convention to label the model configurations of a shallow water test simulation. For example, a run label C48.g2.rk3.b1.00 stands for a simulation with a 48x48x6 cubed-sphere grid with equal-angular tile grid point distribution, and the time advancing is a 3-step Runge-Kutta scheme, the value of the forward-backward parameter is 1, which is fully explicit.

Notation	Description
C[N]	Cubed-sphere grid with N cells along each side of the cubed sphere tile. $N \times N \times 6$ total grid cells. $N \in [48, 96, 192, 384, 768]$.
g0	Equi-edge grid
g2	Equiangular grid
rk2	2-substep time scheme with 1-point- and 5-point-stencil polynomial reconstruction in two substeps.
rk3	3-substep time scheme with 5-point-stencil polynomial reconstruction in all substeps
b[β]	The forward-backward pressure gradient algorithm control parameter $\beta \in [0, 1]$. (only appear when β value is non-default)

tions, beta values, are tested. In GCM development, the rotational mode plays a dominant role in large to mesoscale motions. Therefore in stage 2, the recently proposed modon test (Lin et al., 2017) is implemented to check the model's quality in representing the vorticity. Last but not least, a new "splash test" inspired by X. Chen et al. (2018) is proposed in stage 3 to validate the model's dispersion and dissipation properties. Minimizing the phase speed errors at short wavelengths and maintaining scale-selective diffusion properties are vital for unstaggered algorithms.

The time step in each run configuration is set at the maximum allowed values that the simulations are stable in all resolutions. The courant number for LMARS based solver is estimated by:

$$c_{\mathrm{CFL}} = \frac{(a_{\max} + a_{\min})\,\Delta t}{(\Delta A/\Delta x)_{\min}}, \tag{33}$$

where $a_{\max}$ and $a_{\min}$ stand for the highest and lowest gravity wave speed evaluated at the initial condition, and $(\Delta A/\Delta x)_{\min}$ is the shortest distance across a grid cell. Granting that the maximum flow speed in LMARS should be less than the lowest gravity wave speed, the estimated courant number is an upper bound in the simulations. This def-

Table 4. Time step parameter n_split and the corresponding courant number in each test case. The courant numbers are associated with equi-edge grid configuration.

Test label	Test description	n_split (rk2)	cfl (g0.rk2)	n_split (rk3)	cfl (g0.rk3)
case1	Advection of a cosine bell	8		8	
case2	Steady-state geostrophic balanced flow	10	0.7	7	1
case5	Zonal flow over an isolated conical mountain	13	0.85	9	1.22
case6	Rossby-Haurwitz wave	19	0.81	13	1.18
modon	Colliding modons	11	1.02	7	1.61
splash	Splash on the sphere	2		2	

inition can be extended to the compressible 3D model by replacing the gravity wave speed with acoustic wave speed. In all simulations, the dynamic time step Δt is calculated via: $\Delta t = \text{dt\_atmos}/\left(\text{k\_split} \times \text{n\_split}\right)$, where dt_atmos=3600 s is a base time step, the cycling parameter n_split is unchanged in all resolutions, and k_split scales with resolutions with values [1, 2, 4, 8, 16] corresponding to [C48, C96, C192, C384, C768]. Among all parameters, the choice of the time-marching scheme is the most dominant factor in determining the maximum allowed time steps. The equiangular grid allows slightly larger time steps than the equi-edge grid. Table 4 lists the time step parameter n_spit and the corresponding courant numbers for all test cases.

Error norm measures follow the W92 in the height field h:

$$l_1\left(h\right) = \frac{I\left(\left|h - h_T\right|\right)}{I\left(\left|h_T\right|\right)}, \tag{34}$$

$$l_2\left(h\right) = \sqrt{\frac{I\left(\left(h - h_T\right)^2\right)}{I\left(h_T^2\right)}}, \tag{35}$$

$$l_\infty\left(h\right) = \frac{\max\left|h - h_T\right|}{\max\left|h_T\right|}, \tag{36}$$

with global mean operator:

$$I\left(\eta\right) = \frac{\sum \eta \Delta A}{\sum \Delta A}, \tag{37}$$

where ΔA is the discretized area of the cell that variable η occupies, h_T is the true solution of fluid depth, which is estimated by most accurate C768.g2.rk3.b1.00 results if no analytical solution is available.

4.2 W92 shallow-water tests

In W92, test cases 1, 2, 5, 6 (See test names in Table 4) are widely used in global shallow-water solver development. These four tests are conducted with various configurations and resolutions, as described in the testing plan. In particular, case 1 and 2 usually tests different rotating angles. Since the solutions do not show significant dependency on the rotation directions, only 45-degree rotation results are presented for brevity. The 45-degree advection of the cosine bell and the solid body rotation tests are labels as case1a45 and case2a45.

Figure 4 is the l_1, l_2, and l_{inf} error norms plotted against the simulation time at C48 resolution. A general finding is that the errors of most non-stationary runs depend mostly on rk2 and rk3 differences rather than the grid type choices, indicating diffusion properties are the dominant factor in error growth. Case 1 uses constant flows, which do not evaluate the pressure gradient. Therefore, the results represent the errors solely due to advection schemes. Additionally, no sudden error spike is observed when the cosine bell travels across cube-sphere tile boundaries in case 1. Thus, the singularities in a cubed-sphere geometry are sufficiently handled, and no grid imprinting is observed. In the solid-body rotation tests with stationary solutions, which are different from other tests, all the runs produce similar magnitudes of errors. In particular, the equi-edge grid (g0) produces slightly larger errors than the equiangular grid (g2). Lastly, in case 6, the error growth curves do not saturate to a steady number, indicating that the solutions remain in-phase (X. Chen et al., 2018) even after a significant period of simulation time.

To determine the general order of accuracy in different configurations, Figure 5 plots the l_2 error against different resolutions at specific dates. The results indicate that rk3 runs consistently maintain second-order overall accuracy, which is expected since no high-order multi-dimension scheme is employed in this work. The previous section explains that the forward-backward scheme (FB) is numerically equivalent to adding a 2nd-order stabilizing term to the advection equation, which will degrade the overall 2nd-order ac-

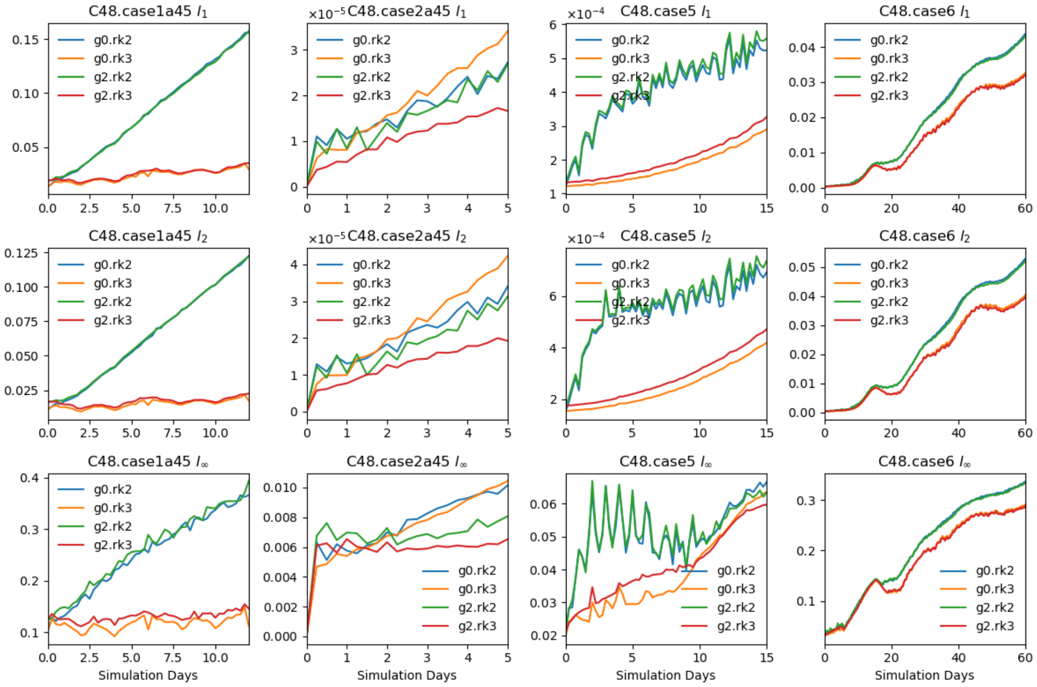


Figure 4. l_1, l_2 and l_{inf} error norms growth with the simulation time of W92 test cases 1, 2, 5 and 6 at the C48 resolution.

curacy. Therefore, in cases 5 and 6, rk2 runs, which employ a fully forward-backward pressure gradient algorithm, demonstrate lower than 2nd-order overall accuracy. Case 1 does not involve the pressure gradient. Thus, it is not affected by FB settings. It appears that the balanced stationary case 2 is also free of FB accuracy degradation.

A more in-depth analysis of FB impact is demonstrated in Figures 6 and 7. Interestingly, the overall order of accuracy gradually increases with higher beta values in Figure 6. In figure 7, although all results have good convergence towards C768 high-resolution solutions, FB parameter β has a significant influence on the solver's diffusion properties. The fully explicit C48.rk3.b1.00 run produces even higher maximum values (the small circle at the eastern equator) than the C768.rk3.b0.00 run with fully implicit pressure gradient evaluation.

The Rossby-Haurwitz test (case6) with a wavenumber four is adopted in various works. This test can validate the solvers' robustness in preventing the instability due to truncation error in the initial conditions. Figure 8 shows the height field at day 14, 40, 80 at resolutions C48 and C768 with rk2 and rk3 configurations. Only the equal-angular

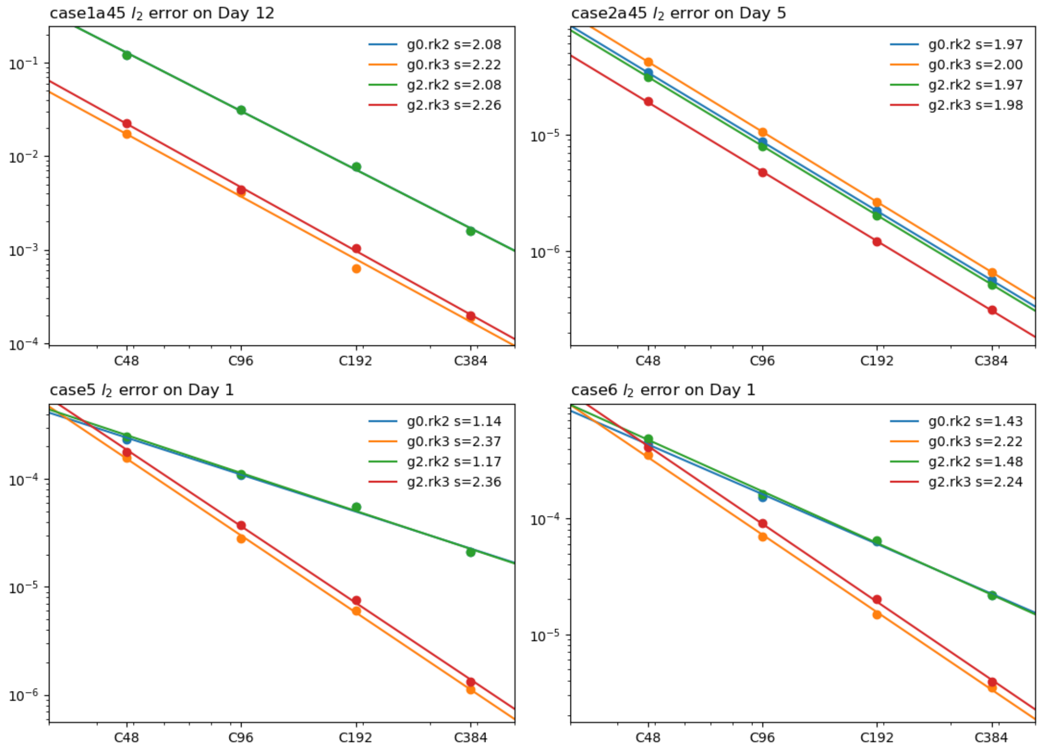


Figure 5. l_2 error against different resolutions at specific dates with different resolutions. The the value of s in the legends are slopes of each line that indicates the overall order of accuracy.

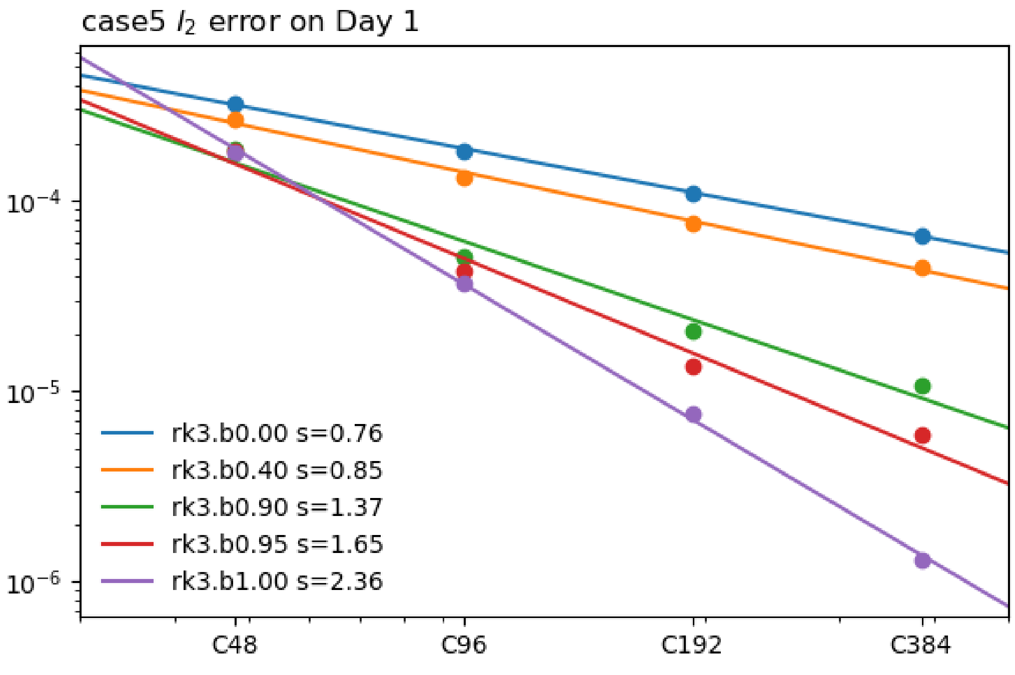


Figure 6. Same as Figure 5, but different beta values in case 5 are compared.

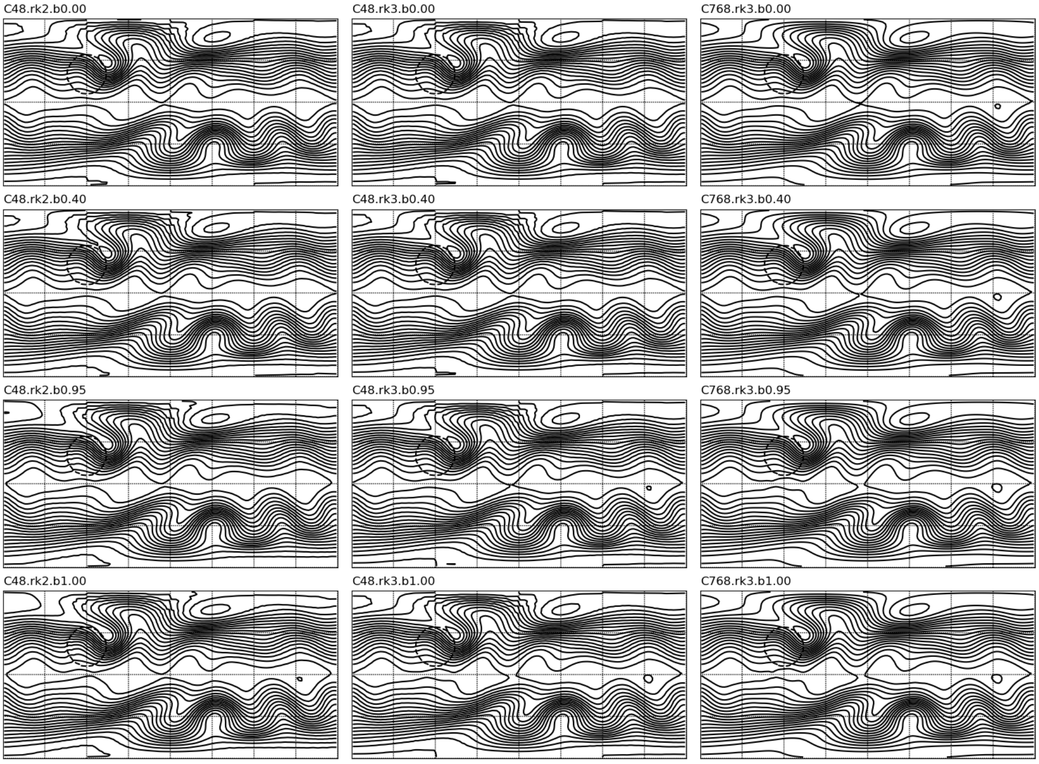


Figure 7. The final day flow height field of case 5 with C48 and C768 resolutions. Different configurations of rk2, rk3 and different strength of FB parameter β values are also compared. The dashed circle is the location of the conical mountain. The contour interval is 50 m.

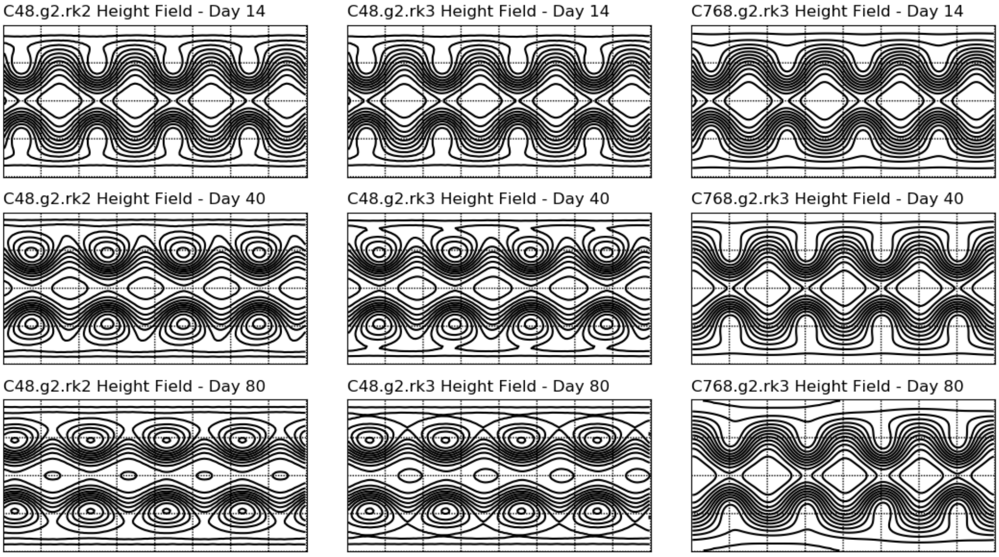


Figure 8. Height field at day 14, 40, 80 at resolutions C48 and C768 with rk2 and rk3 configurations. Only the equal-angular grid (g2) is displayed because no strong dependency from the grid type choices is found. The contour interval is 200 m.

grid is displayed because no strong dependency from the grid type choices is found. The solver achieves optimum symmetry for an extended simulation time of over 80 days. Although both being symmetric, the high-resolution C768 reference solutions maintain much better initial modal structures and intensity than the low-resolution solutions.

4.3 Colliding modons

The recently proposed colliding modons test (Lin et al., 2017; Zhang et al., 2019; Wang et al., 2019) evaluates the solver's ability to simulate vorticity dynamics, which plays a dominant role in large to mesoscale motions. The colliding modon pairs can fully reach steady shapes after one day and travel back to approximately their original locations at around day 100.A successful cycle can produce four symmetrical tracks in four quadrants, namely Q_{sw}, Q_{se}, Q_{nw}, Q_{ne}. The quality of the results can be evaluated by comparing the travel distances and amplitudes of the vortex.

Figure 9 demonstrates the colliding modons simulations with g2.rk3 configuration at various resolutions. The results show that the symmetry is well maintained in all runs. The simulations with C192 and better resolutions show good convergence. Figure 10 com-

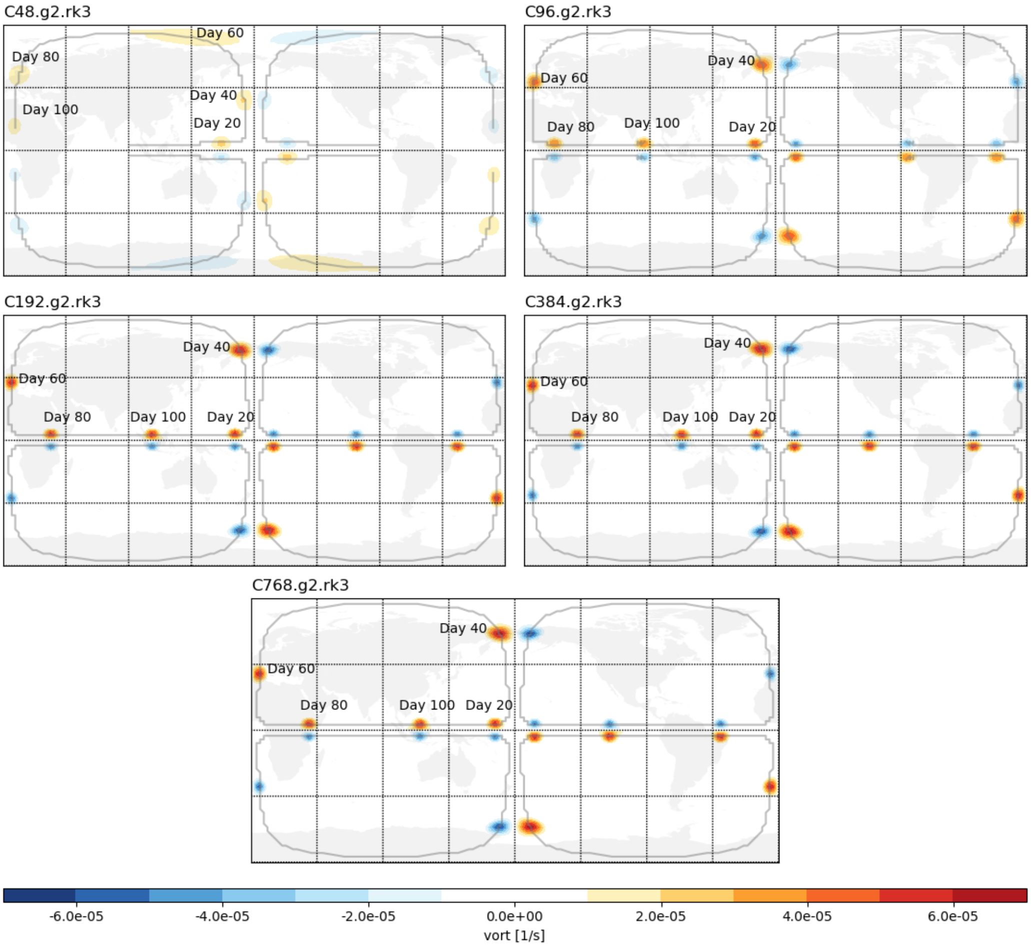


Figure 9. Colliding modons simulations with g2.rk3 configuration at various resolutions.

pares the travel distance and amplitude of the modons in the northwest quadrant at day 100. Note that the travel distance is measured starting at the end of day 1 to ignore initial adjustments. Although the grid choices, discretization configurations have impacts on both values, the resolution is the deterministic factor to resolve the modons movement correctly, and C192 and better resolutions result in converged values. The bar plots in Figure 10 provide a quantitative baseline for future development.

The colliding modons test provides an interesting perspective in atmospheric modeling applications. The spatial scale of each modon is at around 1000 km. The experiments show that although a 200 km resolution (C48) model can capture the motion of the modon, the speed is not correctly simulated, and the strength of the vortex is rapidly dissipated by model internal diffusion. Considering that many rotational processes that are of interest in global circulation models are of horizontal scales less than 1000 km, in-

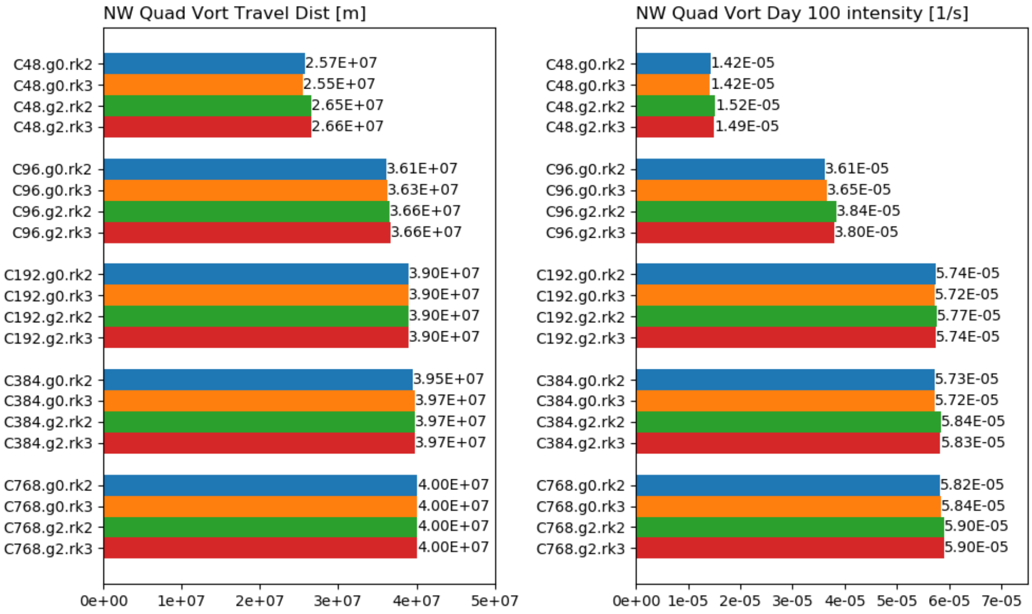


Figure 10. The travel distance and amplitude of the modons in the northwest quadrant at day 100. Note that the travel distance is measured starting at the end of day 1 to ignore initial adjustments.

creasing the model resolution may be the most effective way to improve the accuracy in simulating many rotational events such as cyclones and storms. Recent studies also suggest significant improvements in simulating atmospheric vortices by enhancing the model resolution. For example, Murakami et al. (2015) found that by improving resolution from 50 to 25 km, the model can better capture categories 4 and 5 hurricanes. Gao et al. (2019) showed better hurricane structures by locally increasing the model resolution from 25km to 8km.

4.4 Splash on the sphere

Previous subsections cover a comprehensive range of numerical model properties such as the correctness, orders of accuracy, faithful vortex representation. This section validates the dispersion and dissipation properties at various wavelengths by introducing a “splash on the sphere” test. Historically, unstaggered solvers are considered inferior in representing the short wave propagation due to poor dispersive properties. However, such conclusions usually are based on the low-order linearized analysis, which does not reflect high-order complex real-world applications. By discretizing the high-order lin-

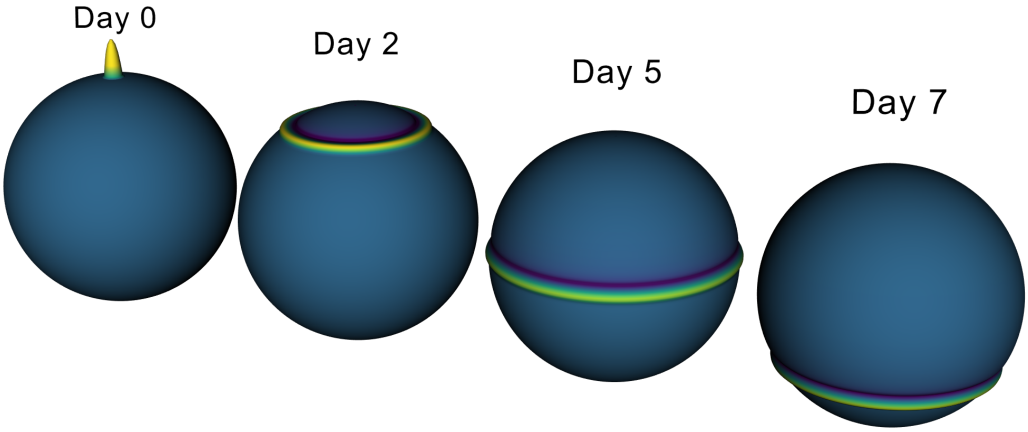


Figure 11. Exaggerated height field of the splash on the sphere test at Days 0, 2, 5, and 7.

earized analysis in Courant numbers and numerical phase, X. Chen et al. (2018) found the grid staggering choices have negligible phase speed influence in high-order algorithms. Furthermore, X. Chen et al. (2018) also introduced a simple 1D test to visualize the dispersion behavior at short waves and the diffusion control of grid-scale noise. Inspired by this work, this section presents an extended 2D version of a "splash on the sphere" test to evaluate solver capabilities in faithfully represent short wave propagations.

The test has a simple physical process to splash a sinusoidal droplet at the North Pole and freely propagate on the non-rotational sphere. The choice of the sinusoidal signal is to limit the waves in a single modal mode and test the sharp gradient between the perturbation and the background when a monotonic algorithm is implemented in the future. The Coriolis coefficient f and the topography are both zero, and the initial velocity field is stationary. The height field with the initial splashing perturbation is defined:

$$\Phi = \begin{cases} \Phi_0 + \Phi' \cos\left(\frac{\pi}{2}\frac{r}{R}\right) & \text{if}\, r < R, \\ \Phi_0 & \text{else,} \end{cases} \tag{38}$$

where r is the distance from the north pole, and $\Phi_0 = 50 \text{ m} \times G \approx (\pi R_e / 10\,\text{Day})^2$, $\Phi' = 1\,\text{m} \times G$, and $R = 500\,\text{km}$. The design of a 50 m background flow height allows the propagating waves to reach the South Pole at approximately ten days, which is an arbitrary choice. Figure 11 illustrates the exaggerated height field at Days 0, 2, 5, and 7.

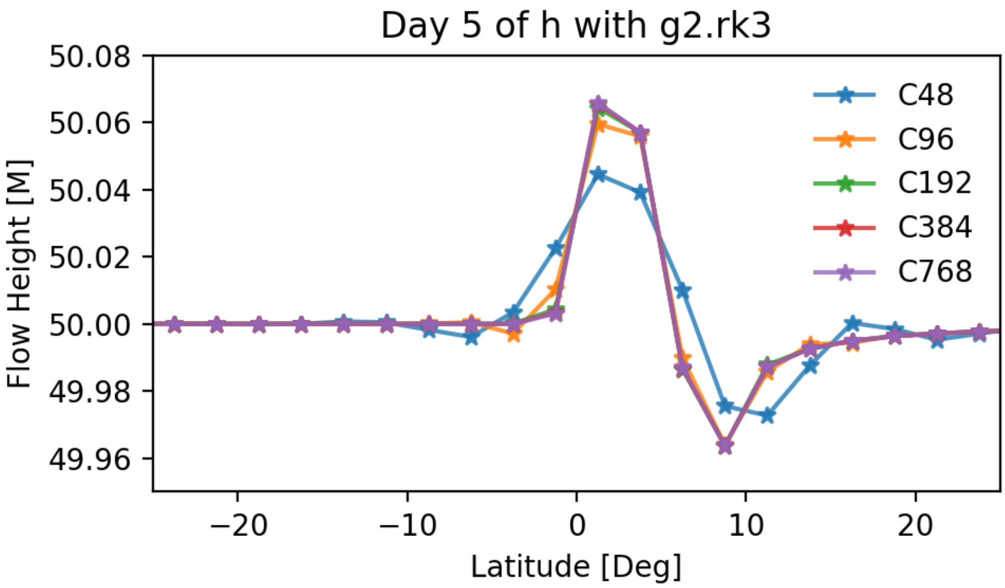


Figure 12. Day-5 snapshots in the meridional direction of g2.rk3 runs at various resolutions.

The analysis of this work focuses on day-5 snapshots when the splash waves propagate to the equatorial region. The results in all resolutions are regridded to 144 × 72 latitude-longitude resolution. Figure 12 shows the day-5 snapshots in the meridional direction of g2.rk3 runs at various resolutions. All simulations propagate the splash at the same location, indicating no phase error is observed. The C48 run shows an obvious spreading of the wave due to the excessive diffusion at this wavelength. Since this work does not implement any monotonic algorithm, overshoots and undershoots are also observed in the results.

Figure 13 quantitatively compares day-5 snapshots of distances to the north pole (NP), peak flow height, and maximum zonal wind absolute values with different discretization settings, gird choices, and resolutions. The identical distances to NP indicate the solver's excellent dispersive properties, which is highly desired with the unstaggered algorithm. In contrast to the modon tests, the peak flow height comparison shows an intriguing observation that the model's intrinsic diffusion properties have an even more significant influence on the results than the resolution differences. In the last column, a perfect numerical solver should not produce any zonal wind in this test. The non-zero values are due to the imperfect representation of the curvilinear cubed-sphere geometry. Therefore, it is an excellent test to check the grid discretization performance. It is

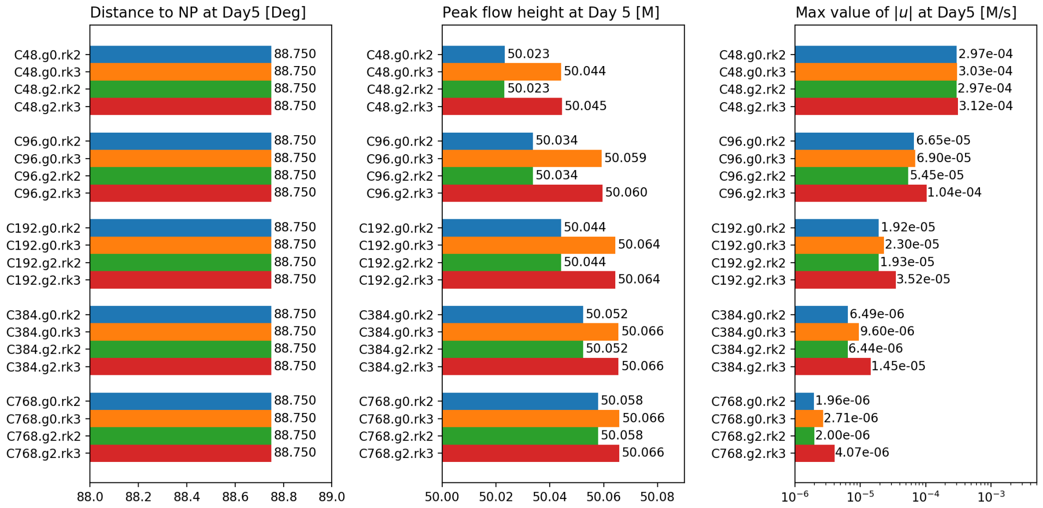


Figure 13. Day-5 snapshots of distances to the north pole (NP), peak flow height, and maximum zonal wind absolute values with different discretization settings, different gird choices, and resolutions.

reassuring to find that the different grid choices do not produce excessive discrepancies in geometric errors. The resolutions and discretization play more critical roles than grid choices.

This work only tests one splash configuration for brevity. In X. Chen et al. (2018), a square wave is also an informative setup to investigate sharp gradient handlings. In future work, a “square splash” can be an appealing option once more sophisticated local monotonic algorithms are included. The tracking of maximum zonal wind absolute values can be a useful gauge of high-order multi-dimension algorithms.

4.5 Discussion

The model demonstrates competitive performance in all numerical tests. Even without any explicit filter, the solver exhibits a broad range of organic implicit diffusion properties, mainly controlled by reconstruction stencil size and strength of forward-backward pressure gradient settings. Considering a full-fledged GCM is an interplay between numerical methods and highly uncertain physics parameterization, flexibility in diffusion control is helpful.

The results provide some perspectives in faithfully simulating various processes. The rotational motions in the colliding modon tests and the propagation of characteristics in the splash tests are both about a 1000-km spatial scale. Although the C48 simulations can capture both events, the qualities are significantly less reliable than those of higher resolution counterparts. C96 is the recommended minimum resolution to accurately represent these processes, which is about ten times grid-spacing. This guideline also indicates the unstaggered solver has balanced capabilities in resolving the rotational mode, and the divergence mode.

Lastly, with the duo-grid system and the tile-edge remapping, the solver is free of any grid imprinting issues. The grid-type choices show negligible differences in the solutions compared to discretization differences and resolutions, which indicates the desired consistency in grid choices for various purposes.

5 Conclusions and future work

In this work, we have successfully demonstrated the LMARS-based shallow-water solver on two gnomonic cubed-sphere grids with controllable variations of computational efficiency and diffusion properties. We use the W92 test suite to warrant the model to produce results that converge to the literature. The model illustrates a broad range of diffusion and controls the delicate balance between the order of accuracy and the robustness by various dynamical core parameters. In particular cases, numerical tests demonstrate that many resolvable processes are not necessarily reliable. We also designed a test to isolate numerical properties by visualizing the dissipation and dispersion. Lastly, Griffies (2004, Chap 17) discussed that explicit frictional operators might alter pressure gradient and angular momentum due to imbalanced design. With no explicit damping, the control of implicit diffusion in this model is organic and does not cause artifacts in pressure gradient calculations.

There are several new ideas in this work. We introduced the duo-grid system that unifies arbitrary gnomonic cubed-sphere grid representations. It also provides the 1D alignment for arbitrary gnomonic grids at halo regions. This work also extends the L97 pressure-gradient integration technique with LMARS diffusion contribution, resulting in a fast and stable discretization. In the numerical test section, the extension of more quantitive metrics to the modon tests allows straightforward comparison between different models and configurations. The splash on the sphere test offers an economical way to get 1D dispersion and dissipation properties of the model without extracting the schemes into a 1D testbed.

Balancing the computational performance and numerical accuracy is one of the top motivations in this development. Therefore, this work implements several optimizations. Some improvement does not sacrifice numerical accuracy. For example, the choice of unit-length basis vectors significantly simplifies the mathematical expressions of the curvilinear system, thus avoiding the computational storage of redundant vector transformation matrices. On the other hand, although halo region 1D remap can be made more sophisticated with higher-order schemes, we choose the simple piecewise linear reconstruction to maintain a minimum unbalanced workload in parallel computing. Lastly, a shallow-water testbed cannot determine the most optimal solver diffusion properties in a fully coupled GCM. Therefore, this work provides a guideline for a controllable range of organic numerical properties once integrated into a full model.

The next step is the extension of this work to a 3D compressible atmosphere with the cubed-sphere grid. Previous work has implemented LMARS in a vertical 2D compressible atmosphere with both Eulerian and Lagrangian vertical coordinates (X. Chen et al., 2013). Li and Chen (2019) demonstrated an energy-conserving model with simple microphysics under the LMARS Eulerian framework. Therefore, the LMARS 3D extension on the cubed-sphere grid has several core components validated in various environments. Another future topic is to include more sophisticated advection components, such as monotonic constraints (van Leer, 1979), high-order multi-dimension transport schemes (Lin & Rood, 1996) in the horizontal direction. With the vertical Lagrangian coordinates, this development shares substantial similarities with the FV3 framework, which can guide future research.

Appendix A List of constants, symbols and their values and units

To make the notations consistent and clear, Table A1 summarizes all important constants, symbols, their values and units in this work for reference.

Symbol	Description	Values	Units
G	Gravitational constant	9.80665	ms^{-2}
Ω	Rotational speed of the Earth	7.292e-5	s^{-1}

R	Radius of the Earth	6.3712e6	m
$\hat{e}_\lambda$	Unit vector in zonal direction	1	NA
$\hat{e}_\phi$	Unit vector in meridional direction	1	NA
$\hat{e}_1$	Unit vector in x-direction (local to cubed-sphere tile)	1	NA
$\hat{e}_2$	Unit vector in y-direction (local to cubed-sphere tile)	1	NA
$\hat{k}$	Unit vector in sphere radius direction	1	NA
λ	Longitude		rad
ϕ	Latitude		rad
i, j	Indices to label grid points in x- and y-directions		NA
α	angle between two curvilinear coordinates		rad
g_{ij}	Covariant 2D metric tensor of the curvilinear grid system, note the ij in this context is the dimension iteration in Einstein Notation		NA
g^{ij}	Contra-variant 2D metric tensor of the curvilinear grid system		NA
h	Thickness of the fluid		m
Π	Thickness of the fluid in term of geopotential		$\mathrm{m}^2\mathrm{s}^{-2}$
Φ	Geopotential		$\mathrm{m}^2\mathrm{s}^{-2}$
Φ_s	Surface geopotential		$\mathrm{m}^2\mathrm{s}^{-2}$
$\vec{u}$	2D velocity vector		ms^{-1}
f	Coriolis parameter		s^{-1}
ζ	Vertical component of the relative vorticity		s^{-1}
u	Covariant wind component in x-direction		ms^{-1}
v	Covariant wind component in y-direction		ms^{-1}
$\tilde{u}$	Contra-variant wind component in x-direction		ms^{-1}
$\tilde{v}$	Contra-variant wind component in x-direction		ms^{-1}
g	Metric coefficient of the curvilinear geometry		NA
u_λ	Wind component in zonal direction		ms^{-1}
u_ϕ	Wind component in meridional direction		ms^{-1}
J^{c2l}	Matrix to convert (u, v) to (u_λ, u_ϕ)		NA
J^{l2c}	Matrix to convert (u_λ, u_ϕ) to (u, v)		NA
$u^x_\perp$	Wind projection perpendicular to cell interface in x-direction		ms^{-1}

$u_{\parallel}^{x}$	Wind projection parallel to cell interface in x-direction	ms^{-1}
$u_{\perp}^{y}$	Wind projection perpendicular to cell interface in y-direction	ms^{-1}
$u_{\parallel}^{y}$	Wind projection parallel to cell interface in y-direction	ms^{-1}
a	Gravity wave speed (group speed of the fluid)	ms^{-1}

Table A1: List of parameters and symbols used throughout this document

Appendix B Gnomonic projection initialization algorithms

This section provides a detailed mathematical reference-line based gnomonic cubed-sphere generation process for reference.

B1 Basic geometric algorithms

Each gnomonic projection can be described by a 3D Cartesian vector $\vec{p}\,(x, y, z)$, which is uniquely determined by a point on the sphere or a 3D vector on the spherical coordinate with unit length $\vec{p}\,(\lambda, \phi)$. The valid information is the direction of the vector, and the length of the vector is trivial. The conversion can be written:

$$x = \cos\phi\cos\lambda, \tag{B1}$$

$$y = \cos\phi\sin\lambda, \tag{B2}$$

$$z = \sin\phi, \tag{B3}$$

and

$$\lambda = \arctan\frac{y}{x}, \tag{B4}$$

$$\phi = \arcsin\frac{z}{\sqrt{x^2+y^2+z^2}}, \tag{B5}$$

where the Fortran intrinsic functions atan2() is used to place λ in the range of $-\pi < \lambda \leq \pi$.

To calculate the great circle distance between two points:

$$\mathrm{dist}\left(\vec{p}_1\left(\lambda_1, \phi_1\right), \vec{p}_2\left(\lambda_2, \phi_2\right)\right) = R\arccos\left(\frac{\vec{p}_1 \cdot \vec{p}_2}{\left|\vec{p}_1\right|\left|\vec{p}_2\right|}\right), \tag{B6}$$

or

$$\mathrm{dist}\left(\vec{p}_1\left(\lambda_1,\phi_1\right),\vec{p}_2\left(\lambda_2,\phi_2\right)\right)=R\arccos\left(\cos\phi_1\cos\phi_2\cos\left(\lambda_1-\lambda_2\right)+\sin\phi_1\sin\phi_2\right). \tag{B7}$$

To calculate the area of a grid cell:

$$\mathrm{area}\left(\vec{p}_1,\vec{p}_2,\vec{p}_3,\vec{p}_4\right)=R^2\left(\alpha_{412}+\alpha_{123}+\alpha_{234}+\alpha_{341}-2\pi\right), \tag{B8}$$

where α_{ijk} is the angle between three points $(\vec{p}_i,\vec{p}_j,\vec{p}_k)$ with “right-hand rule” from $\vec{p}_j$, and can be calculated by:

$$\alpha_{ijk}=\arccos\left(\frac{\left(\vec{p}_i\times\vec{p}_j\right)\cdot\left(\vec{p}_j\times\vec{p}_k\right)}{\left|\vec{p}_i\times\vec{p}_j\right|\cdot\left|\vec{p}_j\times\vec{p}_k\right|}\right). \tag{B9}$$

The mid-point of two vectors $\vec{p}_1$ and $\vec{p}_2$ is:

$$\mathrm{mid}\left(\vec{p}_1,\vec{p}_2\right)=\frac{\vec{p}_1+\vec{p}_2}{2}. \tag{B10}$$

B2 Initializing gnomonic projection with reference lines on tile one

As illustrated in Fig. 3, the eight corners of the tiles in 3D Cartesian coordinates are $(x,y,z)=(\pm1,\pm1,\pm1)$. The resolution of the cubed-sphere grid is denoted by $C\left[N\right]$, which indicates $N\times N\times 6$ total grid cells. Let the first tile be centered at $(\lambda,\phi)=(0,0)$ or the cube-tile with vertices $(x,y,z)=(1,\pm1,\pm1)$ and denote (X,Y) the local coordinates on this cube-tile, the 3D Cartesian coordinates can be written by:

$$(x,y,z)=\left(1,X,Y\right), \tag{B11}$$

and the projection vector can be also written $\vec{p}\,(X,Y)$. Additionally, let (i,j) be the indices of the grid points on tile one, and the indices start counting from 1. Therefore, the coordinate sets (x,y,z), (λ,ϕ), (X,Y) and (i,j) are interchangeable and representing the same gnomonic projection vector. The vertices on the tile can be uniquely determined by a column of Y values on the cube tile surface.

B21 Equidistance projection

To get the Y values of the red reference column:

$$Y\left(i_{\mathrm{ref}},j\right)=-1+\left(j-1\right)\delta h, \tag{B12}$$

where $\delta h=2/N$, $i_{\mathrm{ref}}=N/2+1$. Then the red reference row is:

$$X\left(i,j_{\mathrm{ref}}\right)=Y\left(i_{\mathrm{ref}},i\right), \tag{B13}$$

where $i_{\mathrm{ref}} = N/2 + 1$, and the grid mesh on the cube tile can be populated from the two reference lines accordingly:

$$X\left(i,j\right) = X\left(i, j_{ref}\right) = Y\left(i_{\mathrm{ref}}, i\right), \tag{B14}$$

$$Y\left(i,j\right) = Y\left(i_{\mathrm{ref}}, j\right). \tag{B15}$$

Therefore, $\vec{p}\,(X,Y)$ can be easily converted to $\vec{p}\,(x,y,z)$ and $\vec{p}\,(\lambda,\phi)$, and the grid points on the cubed-sphere tile is generated. One may observe that the values of i_{ref} and j_{ref} are trivial

B22 Equiangular projection

Comparing to equidistance projection, the equiangular grid generation requires an extra first step to project grid points from the green reference column to the cube tile and get the Y values. The angular step of the grid points on the green row is:

$$\delta\zeta = \frac{2\alpha_{\mathrm{ref}}}{N}, \tag{B16}$$

where $\alpha_{\mathrm{ref}} = \pi/4$. Then the Y values are:

$$Y\left(i_{\mathrm{ref}}, j\right) = R_{\mathrm{ref}} \tan\left(\left(j-1\right)\delta\zeta - \alpha_{\mathrm{ref}}\right), \tag{B17}$$

where $R_{\mathrm{ref}} = 1$. Then the entire set of grid points on the tile can be populated following the same procedure as the equidistance projection.

B23 Equi-edge projection

The procedures to generate the equi-edge projection are identical to the equiangular projection except different parameters: $\alpha_{\mathrm{ref}} = \arcsin\left(\sqrt{3}\right)$, $R_{\mathrm{ref}} = \sqrt{2}$.

B3 Populating grid points to six tiles

Once the grid points on tile one are generated, the points on the rest of the tiles can be populated using the “staircase” tile-interlock pattern. The basic rotations to any 3D vector can be decomposed by the combination of the rotations about each axes of the Cartesian coordinate system. The rotation about each axes by an angle β following right-hand rule:

$$R_x\left(\beta\right) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{pmatrix}, \tag{B18}$$

$$R_y\left(\beta\right)=\begin{pmatrix}\cos\beta & 0 & \sin\beta\\ 0 & 1 & 0\\ -\sin\beta & 0 & \cos\beta\end{pmatrix},\tag{B19}$$

$$R_z\left(\beta\right)=\begin{pmatrix}\cos\beta & -\sin\beta & 0\\ \sin\beta & \cos\beta & 0\\ 0 & 0 & 1\end{pmatrix}.\tag{B20}$$

In particular,

$$R_x\left(90^\circ\right)=\begin{pmatrix}1 & 0 & 0\\ 0 & 0 & -1\\ 0 & 1 & 0\end{pmatrix},\tag{B21}$$

$$R_y\left(90^\circ\right)=\begin{pmatrix}0 & 0 & 1\\ 0 & 1 & 0\\ -1 & 0 & 0\end{pmatrix},\tag{B22}$$

$$R_z\left(90^\circ\right)=\begin{pmatrix}0 & -1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 1\end{pmatrix}.\tag{B23}$$

Therefore, the rotations for each tile:

$$\vec{p}\left(i,j;\text{tile}=2\right)=R_z\left(90^\circ\right)\vec{p}\left(i,j;\text{tile}=1\right),\tag{B24}$$

$$\vec{p}\left(i,j;\text{tile}=3\right)=R_x\left(90^\circ\right)R_z\left(90^\circ\right)\vec{p}\left(i,j;\text{tile}=1\right),\tag{B25}$$

$$\vec{p}\left(i,j;\text{tile}=4\right)=R_x\left(90^\circ\right)R_z\left(180\right)\vec{p}\left(i,j;\text{tile}=1\right),\tag{B26}$$

$$\vec{p}\left(i,j;\text{tile}=5\right)=R_y\left(90^\circ\right)R_z\left(270\right)\vec{p}\left(i,j;\text{tile}=1\right),\tag{B27}$$

$$\vec{p}\left(i,j;\text{tile}=6\right)=R_y\left(90^\circ\right)\vec{p}\left(i,j;\text{tile}=1\right),\tag{B28}$$

or:

$$\vec{p}\left(i,j;\text{tile}=1\right)=\left[1,X,Y\right],\tag{B29}$$

$$\vec{p}\left(i,j;\text{tile}=2\right)=\left[-X,1,Y\right],\tag{B30}$$

$$\vec{p}\left(i,j;\text{tile}=3\right)=\left[-X,-Y,1\right],\tag{B31}$$

$$\vec{p}\left(i,j;\text{tile}=4\right)=\left[-1,-Y,-X\right],\tag{B32}$$

$$\vec{p}\left(i,j;\text{tile}=5\right)=\left[Y,-1,-X\right],\tag{B33}$$

$$\vec{p}\left(i,j;\text{tile}=6\right)=\left[Y,X,-1\right].\tag{B34}$$

Appendix C The governing equations

This shallow water model employs the vector-invariant form governing equations:

$$\frac{\partial h}{\partial t} = -\nabla \cdot (h\vec{u}), \tag{C1}$$

$$\frac{\partial \vec{u}}{\partial t} = -(\zeta + f)\,\vec{k} \times \vec{u} - \nabla\left((\Pi + \Phi_s) + \frac{1}{2}\vec{u} \cdot \vec{u}\right), \tag{C2}$$

where

$$\Pi = Gh, \tag{C3}$$

$$\zeta = \hat{k} \cdot (\nabla \times \vec{u}), \tag{C4}$$

$$f = 2\Omega \sin\phi. \tag{C5}$$

The definition of the variables, constants, and their values are summarized in Table A1.

C1 Generic form in a curvilinear grid system

This solver uses covariant wind components as the prognostic variable. The covariant wind components are the projections of the wind vector on the curvilinear coordinates:

$$u = \vec{u} \cdot \hat{e}_1, \tag{C6}$$

$$v = \vec{u} \cdot \hat{e}_2, \tag{C7}$$

and the corresponding contra-variant wind components satisfy:

$$\vec{u} = \tilde{u}\hat{e}_1 + \tilde{v}\hat{e}_2. \tag{C8}$$

A covariant 2D metric tensor g_{ij} of the curvilinear grid system on a tile of the cubed-sphere is defined by Einstein Notation:

$$g_{ij} = \hat{e}_i \cdot \hat{e}_j, \tag{C9}$$

with the metric coefficient:

$$g = \det(g_{ij}). \tag{C10}$$

Note $\hat{e}_i$ and $\hat{e}_j$ in the Einstein Notation are iterations along dimensions $\hat{e}_1$ and $\hat{e}_2$. The contra-variant 2D metric tensor g^{ij} is the inverse of g_{ij}, and the conversions between covariant and contra-variant vector component are:

$$\begin{bmatrix} u \\ v \end{bmatrix} = g_{ij} \begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix}, \tag{C11}$$

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = g^{ij} \begin{bmatrix} u \\ v \end{bmatrix}. \tag{C12}$$

C2 The choice of the basis vectors and the optimizations

The choice of the definition of $\hat{e}_1$ and $\hat{e}_2$ are flexible. Modelers made various definitions of these terms in the literature. To yield concise mathematical expressions in numerical discretization, the covariant unit vectors are set to unit length in space, or $|\hat{e}_1| = |\hat{e}_2| = 1$. Thus, the values at the each grid point (i, j) on a cubed-sphere tile can be calculated in a discretized form:

$$(\hat{e}_1)_{i,j} = \frac{\left(\hat{k}_{i,j} \times \hat{k}_{i+1,j}\right) \times \hat{k}_{i,j}}{\left|\left(\hat{k}_{i,j} \times \hat{k}_{i+1,j}\right) \times \hat{k}_{i,j}\right|}, \tag{C13}$$

$$(\hat{e}_2)_{i,j} = \frac{\left(\hat{k}_{i,j} \times \hat{k}_{i,j+1}\right) \times \hat{k}_{i,j}}{\left|\left(\hat{k}_{i,j} \times \hat{k}_{i,j+1}\right) \times \hat{k}_{i,j}\right|}, \tag{C14}$$

with:

$$\hat{k}\left(\lambda, \phi\right) = \begin{bmatrix} \cos\phi \cos\lambda \\ \cos\phi \sin\lambda \\ \sin\phi \end{bmatrix}. \tag{C15}$$

This algorithm to determine $\hat{e}_1$ and $\hat{e}_2$ offers better flexibility when constructing various implementations of the cubed-sphere grids.

Define α the angle between two local curvilinear coordinates, or the unit vectors $\hat{e}_1$ and $\hat{e}_2$ on a cubed-sphere tile, it yields:

$$\sin\alpha = \hat{k} \cdot \left(\hat{e}_1 \times \hat{e}_2\right), \tag{C16}$$

$$\cos\alpha = \hat{e}_1 \cdot \hat{e}_2. \tag{C17}$$

Note that discrete grid subscripts i, j are omitted for cleaner expression, since all variables are co-located.

This set of basis produces concise mathematical forms of the metric tensors:

$$g_{ij} = \begin{bmatrix} 1 & \cos\alpha \\ \cos\alpha & 1 \end{bmatrix}, \tag{C18}$$

$$g^{ij} = \frac{1}{\sin^2\alpha} \begin{bmatrix} 1 & -\cos\alpha \\ -\cos\alpha & 1 \end{bmatrix}, \tag{C19}$$

$$\sqrt{g} = \sin\alpha. \tag{C20}$$

C3 Conversions of vector components to the regular latitude-longitude coordinates

The latitude-longitude basis vectors are:

$$\hat{e}_\lambda\left(\lambda,\phi\right)=\frac{1}{\cos\phi}\frac{\partial\hat{k}}{\partial\lambda}=\begin{bmatrix}-\sin\lambda\\ \cos\lambda\\ 0\end{bmatrix},\tag{C21}$$

$$\hat{e}_\phi\left(\lambda,\phi\right)=\frac{\partial\hat{k}}{\partial\phi}=\begin{bmatrix}-\sin\phi\cos\lambda\\ -\sin\phi\sin\lambda\\ \cos\phi\end{bmatrix},\tag{C22}$$

and define the regular zonal and meridional wind components u_λ and u_θ. The conversion matrices are:

$$J^{c2l}=\begin{bmatrix}\hat{e}_1\cdot\hat{e}_\lambda & \hat{e}_2\cdot\hat{e}_\lambda\\ \hat{e}_1\cdot\hat{e}_\phi & \hat{e}_2\cdot\hat{e}_\phi\end{bmatrix}g^{ij},\tag{C23}$$

$$J^{l2c}=\frac{g_{ij}}{\det\left(J^{c2l}\right)}\begin{bmatrix}J^{c2l}_{11} & -J^{c2l}_{12}\\ -J^{c2l}_{21} & J^{c2l}_{22}\end{bmatrix},\tag{C24}$$

with the conversion relations:

$$\begin{bmatrix}u_\lambda\\ v_\phi\end{bmatrix}=J^{c2l}\begin{bmatrix}u\\ v\end{bmatrix},\tag{C25}$$

$$\begin{bmatrix}u\\ v\end{bmatrix}=J^{l2c}\begin{bmatrix}u_\lambda\\ v_\phi\end{bmatrix}.\tag{C26}$$

C4 Conversions of vector components to the local orthogonal coordinates at cell interfaces

The preparation of the Riemann solver requires the wind vectors to be projected to the local orthogonal coordinates at the cell interfaces. Therefore, at cell interfaces in x-direction:

$$u^x_\perp=\tilde{u}\sin\alpha,\tag{C27}$$

$$u^x_\parallel=\tilde{u}\cos\alpha+\tilde{v}=v,\tag{C28}$$

or:

$$\begin{bmatrix}u^x_\perp\\ u^x_\parallel\end{bmatrix}=\begin{bmatrix}\sin\alpha & 0\\ \cos\alpha & 1\end{bmatrix}\begin{bmatrix}\tilde{u}\\ \tilde{v}\end{bmatrix},\tag{C29}$$

with inversion:

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sin\alpha} & 0 \\ -\frac{\cos\alpha}{\sin\alpha} & 1 \end{bmatrix} \begin{bmatrix} u_\perp^x \\ u_\parallel^x \end{bmatrix} \tag{C30}$$

At cell interfaces in y-direction:

$$u_\perp^y = \tilde{v}\sin\alpha, \tag{C31}$$

$$u_\parallel^x = \tilde{v}\cos\alpha + \tilde{u} = u, \tag{C32}$$

or:

$$\begin{bmatrix} u_\perp^y \\ u_\parallel^y \end{bmatrix} = \begin{bmatrix} 0 & \sin\alpha \\ 1 & \cos\alpha \end{bmatrix} \begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix}, \tag{C33}$$

with inversion:

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = \begin{bmatrix} -\frac{\cos\alpha}{\sin\alpha} & 1 \\ \frac{1}{\sin\alpha} & 0 \end{bmatrix} \begin{bmatrix} u_\perp^y \\ u_\parallel^y \end{bmatrix} \tag{C34}$$

A small optimization can be observed from the above subsections. With basis vectors $\hat{e}_1$ and $\hat{e}_2$ being restricted to unit length, many vector conversions can be carried out with simple operations using pre-calculated values of $\sin\alpha$ and $\cos\alpha$. Therefore, the model saves considerable memory storage and computational cost by dropping many metric matrices and reducing operation counts. Moreover, the metric terms generation are more flexible in arbitrary gnomonic projections and simpler than most of the literature.

C5 Projecting the governing equations to the curvilinear coordinates

Projecting the continuity equation and the vector-invariance form momentum equation to the curvilinear coordinates yields:

$$\frac{\partial h}{\partial t} = -\frac{1}{\sqrt{g}}\left(\frac{\partial}{\partial x}\left(\sqrt{g}h\tilde{u}\right) + \frac{\partial}{\partial y}\left(\sqrt{g}h\tilde{v}\right)\right), \tag{C35}$$

$$\frac{\partial u}{\partial t} = \sqrt{g}\hat{v}\left(\zeta + f\right) - \frac{\partial}{\partial x}\left(\Pi + \Phi_s + K\right), \tag{C36}$$

$$\frac{\partial v}{\partial t} = -\sqrt{g}\hat{u}\left(\zeta + f\right) - \frac{\partial}{\partial y}\left(\Pi + \Phi_s + K\right), \tag{C37}$$

with the relative vorticity and the kinetic energy defined by:

$$\zeta = \frac{1}{\sqrt{g}}\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right), \tag{C38}$$

$$K = \frac{1}{2}\left(u\tilde{u} + v\tilde{v}\right). \tag{C39}$$

These two terms can be further rearranged and optimized with better finite-volume representations in the numerical discretization process.

Acknowledgments

The author wants to thank Lucas Harris and Stephen Griffies for the constructive GFDL internal reviews. All materials used in this work are stored through GFDL's Archiving System at the location "/archive/Xi.Chen/PAPERS/2020.JAMES.SW/data" and can be obtained by contacting GFDL or the author. This report was prepared by Xi Chen under award NA18OAR4320123 from the National Oceanic and Atmospheric Administration, U.S. Department of Commerce. The statements, findings, conclusions, and recommendations are those of the author(s) and do not necessarily reflect the views of the National Oceanic and Atmospheric Administration, or the U.S. Department of Commerce.

References

Adcroft, A., Anderson, W., Balaji, V., Blanton, C., Bushuk, M., Dufour, C. O., … Zhang, R. (2019, 10). The GFDL Global Ocean and Sea Ice Model OM4.0: Model Description and Simulation Features. *Journal of Advances in Modeling Earth Systems*, *11*(10), 3167–3211. Retrieved from `https://onlinelibrary.wiley.com/doi/abs/10.1029/2019MS001726` doi: 10.1029/2019MS001726

Adcroft, A., Campin, J.-M., Hill, C., & Marshall, J. (2004, 12). Implementation of an Atmosphere-Ocean General Circulation Model on the Expanded Spherical Cube. *Monthly Weather Review*, *132*(12), 2845–2863. Retrieved from `http://dx.doi.org/10.1175/MWR2823.1` doi: 10.1175/MWR2823.1

Arakawa, A., & Lamb, V. R. (1977, 1). Computational design of the basic dynamical processes of the UCLA general circulation model. *Methods in computational physics*, *17*, 173–265.

Bourke, W. (1972, 9). An Efficient, One-Level, Primitive-Equation Spectral Model. *Monthly Weather Review*, *100*(9), 683–689. Retrieved from `http://journals.ametsoc.org/doi/abs/10.1175/1520-0493%281972%29100%3C0683%3AAEOPSM%3E2.3.CO%3B2` doi: 10.1175/1520-0493(1972)100⟨0683:AEOPSM⟩2.3.CO;2

Bourke, W. (1974, 10). A Multi-Level Spectral Model. I. Formulation and

Hemispheric Integrations. *Monthly Weather Review*, *102*(10), 687–701. Retrieved from `http://journals.ametsoc.org/doi/abs/10.1175/1520-0493%281974%29102%3C0687%3AAMLSMI%3E2.0.CO%3B2` doi: 10.1175/1520-0493(1974)102⟨0687:AMLSMI⟩2.0.CO;2

Chen, C., & Xiao, F. (2008, 5). Shallow water model on cubed-sphere by multi-moment finite volume method. *Journal of Computational Physics*, *227*(10), 5019–5044. Retrieved from `https://linkinghub.elsevier.com/retrieve/pii/S0021999108000570` doi: 10.1016/j.jcp.2008.01.033

Chen, X., Andronova, N., Van Leer, B., Penner, J. E., Boyd, J. P., Jablonowski, C., & Lin, S.-J. (2013, 7). A Control-Volume Model of the Compressible Euler Equations with a Vertical Lagrangian Coordinate. *Monthly Weather Review*, *141*(7), 2526–2544. Retrieved from `http://journals.ametsoc.org/doi/abs/10.1175/MWR-D-12-00129.1` doi: 10.1175/MWR-D-12-00129.1

Chen, X., Lin, S.-J., & Harris, L. M. (2018, 9). Towards an Unstaggered Finite-Volume Dynamical Core With a Fast Riemann Solver: 1-D Linearized Analysis of Dissipation, Dispersion, and Noise Control. *Journal of Advances in Modeling Earth Systems*, *10*(9), 2333–2356. Retrieved from `http://doi.wiley.com/10.1029/2018MS001361` doi: 10.1029/2018MS001361

Du, Q., Gunzburger, M. D., & Ju, L. (2003, 8). Voronoi-based finite volume methods, optimal Voronoi meshes, and PDEs on the sphere. *Computer Methods in Applied Mechanics and Engineering*, *192*(35-36), 3933–3957. Retrieved from `https://linkinghub.elsevier.com/retrieve/pii/S0045782503003943` doi: 10.1016/S0045-7825(03)00394-3

Gao, K., Harris, L., Chen, J.-H., Lin, S.-J., & Hazelton, A. (2019, 1). Improving AGCM Hurricane Structure With Two-Way Nesting. *Journal of Advances in Modeling Earth Systems*, *11*(1), 278–292. Retrieved from `http://doi.wiley.com/10.1029/2018MS001359` doi: 10.1029/2018MS001359

Giraldo, F., Hesthaven, J., & Warburton, T. (2002, 9). Nodal High-Order Discontinuous Galerkin Methods for the Spherical Shallow Water Equations. *Journal of Computational Physics*, *181*(2), 499–525. Retrieved from `https://linkinghub.elsevier.com/retrieve/pii/S0021999102971391` doi: 10.1006/jcph.2002.7139

Griffies, S. (2004). *Fundamentals of Ocean Climate Models.* Princeton, N.J:

Princeton University Press. Retrieved from `https://www.amazon.com/Fundamentals-Climate-Models-Stephen-Griffies-dp-0691118922/dp/0691118922/ref=mt_hardcover?_encoding=UTF8&me=&qid=1591938681`

Harris, L. M., & Lin, S.-J. (2013, 1). A Two-Way Nested Global-Regional Dynamical Core on the Cubed-Sphere Grid. *Monthly Weather Review*, *141*(1), 283–306. Retrieved from `http://journals.ametsoc.org/doi/abs/10.1175/MWR-D-11-00201.1` doi: 10.1175/MWR-D-11-00201.1

Harris, L. M., Lin, S.-J., & Tu, C. (2016, 3). High-Resolution Climate Simulations Using GFDL HiRAM with a Stretched Global Grid. *Journal of Climate*, *29*(11), 4293–4314. Retrieved from `http://dx.doi.org/10.1175/JCLI-D-15-0389.1` doi: 10.1175/JCLI-D-15-0389.1

Katta, K. K., Nair, R. D., & Kumar, V. (2015, 7). High-Order Finite-Volume Transport on the Cubed Sphere: Comparison between 1D and 2D Reconstruction Schemes. *Monthly Weather Review*, *143*(7), 2937–2954. Retrieved from `http://dx.doi.org/10.1175/MWR-D-13-00176.1https://journals.ametsoc.org/mwr/article/143/7/2937/72256/HighOrder-FiniteVolume-Transport-on-the-Cubed` doi: 10.1175/MWR-D-13-00176.1

Khamayseh, A., & Mastin, C. (1996). Computational Conformal Mapping for Surface Grid Generation. *Journal of Computational Physics*, *123*(2), 394–401. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0021999196900327` doi: http://dx.doi.org/10.1006/jcph.1996.0032

Li, C., & Chen, X. (2019, 2). Simulating Nonhydrostatic Atmospheres on Planets (SNAP): Formulation, Validation, and Application to the Jovian Atmosphere. *The Astrophysical Journal Supplement Series*, *240*(2), 37. Retrieved from `https://iopscience.iop.org/article/10.3847/1538-4365/aafdaa` doi: 10.3847/1538-4365/aafdaa

Lin, S.-J. (1997, 7). A finite-volume integration method for computing pressure gradient force in general vertical coordinates. *Quarterly Journal of the Royal Meteorological Society*, *123*(542), 1749–1762. Retrieved from `http://doi.wiley.com/10.1002/qj.49712354214` doi: 10.1002/qj.49712354214

Lin, S.-J. (2004, 10). A “Vertically Lagrangian” Finite-Volume Dynamical Core for Global Models. *Monthly Weather Review*, *132*(10), 2293–2307. Retrieved from `http://journals.ametsoc.org/doi/abs/10.1175/1520-0493(2004)`

`132<2293:AVLFDC>2.0.CO;2` doi: 10.1175/1520-0493(2004)132⟨2293:AVLFDC⟩2.0.CO;2

Lin, S.-J., Harris, L., Chen, X., Yao, W., & Chai, J. (2017, 11). Colliding Modons: A Nonlinear Test for the Evaluation of Global Dynamical Cores. *Journal of Advances in Modeling Earth Systems*, *9*(7), 2483–2492. Retrieved from `http://dx.doi.org/10.1002/2017MS000965http://doi.wiley.com/10.1002/2017MS000965` doi: 10.1002/2017MS000965

Lin, S.-J., & Rood, R. B. (1996, 9). Multidimensional Flux-Form Semi-Lagrangian Transport Schemes. *Monthly Weather Review*, *124*(9), 2046–2070. Retrieved from `https://doi.org/10.1175/1520-0493(1996)124%3C2046:MFFSLT%3E2.0.COhttp://0.0.0.2http://journals.ametsoc.org/doi/abs/10.1175/1520-0493(1996)124%3C2046%3AMFFSLT%3E2.0.CO%3B2` doi: 10.1175/1520-0493(1996)124⟨2046:MFFSLT⟩2.0.CO;2

McGregor, J. L. (1996, 6). Semi-Lagrangian Advection on Conformal-Cubic Grids. *Monthly Weather Review*, *124*(6), 1311–1322. Retrieved from `http://journals.ametsoc.org/doi/abs/10.1175/1520-0493%281996%29124%3C1311%3ASLAOCC%3E2.0.CO%3B2` doi: 10.1175/1520-0493(1996)124⟨1311:SLAOCC⟩2.0.CO;2

Mesinger, F. (1977). Forward-backward scheme, and its use in a limited area model. *Contrilb. Atmos. Phys.*, *50*, 200–210.

Murakami, H., Vecchi, G. A., Underwood, S., Delworth, T. L., Wittenberg, A. T., Anderson, W. G., … Zeng, F. (2015, 12). Simulation and Prediction of Category 4 and 5 Hurricanes in the High-Resolution GFDL HiFLOR Coupled Climate Model. *Journal of Climate*, *28*(23), 9058–9079. Retrieved from `http://journals.ametsoc.org/doi/10.1175/JCLI-D-15-0216.1` doi: 10.1175/JCLI-D-15-0216.1

Putman, W. M., & Lin, S.-J. (2007, 11). Finite-volume transport on various cubed-sphere grids. *Journal of Computational Physics*, *227*(1), 55–78. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0021999107003105` doi: 10.1016/j.jcp.2007.07.022

Rančić, M., Purser, R. J., & Mesinger, F. (1996, 4). A global shallow-water model using an expanded spherical cube: Gnomonic versus conformal coordinates. *Quarterly Journal of the Royal Meteorological Society*, *122*(532), 959–982. Retrieved from `http://dx.doi.org/10.1002/qj.49712253209` doi: 10.1002/qj.49712253209

Ringler, T. D., Heikes, R. P., & Randall, D. A. (2000, 7). Modeling the Atmospheric General Circulation Using a Spherical Geodesic Grid: A New Class of Dynamical Cores. *Monthly Weather Review*, *128*(7), 2471–2490. Retrieved from `http://journals.ametsoc.org/doi/full/10.1175/1520-0493%282000%29128%3C2471%3AMTAGCU%3E2.0.CO%3B2` doi: 10.1175/1520-0493(2000)128⟨2471:MTAGCU⟩2.0.CO;2

Rípodas, P., Gassmann, A., Förstner, J., Majewski, D., Giorgetta, M., Korn, P., … Heinze, T. (2009, 12). Icosahedral Shallow Water Model (ICOSWM): results of shallow water test cases and sensitivity to model parameters. *Geoscientific Model Development*, *2*(2), 231–251. Retrieved from `https://www.geosci-model-dev.net/2/231/2009/` doi: 10.5194/gmd-2-231-2009

Ronchi, C., Iacono, R., & Paolucci, P. (1996, 3). The "Cubed Sphere": A New Method for the Solution of Partial Differential Equations in Spherical Geometry. *Journal of Computational Physics*, *124*(1), 93–114. Retrieved from `https://linkinghub.elsevier.com/retrieve/pii/S0021999196900479` doi: 10.1006/jcph.1996.0047

Rossmanith, J. A. (2006, 4). A wave propagation method for hyperbolic systems on the sphere. *Journal of Computational Physics*, *213*(2), 629–658. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0021999105003967` doi: http://dx.doi.org/10.1016/j.jcp.2005.08.027

Sadourny, R. (1972, 2). Conservative Finite-Difference Approximations of the Primitive Equations on Quasi-Uniform Spherical Grids. *Monthly Weather Review*, *100*(2), 136–144. Retrieved from `http://journals.ametsoc.org/doi/abs/10.1175/1520-0493%281972%29100%3C0136%3ACFAOTP%3E2.3.CO%3B2` doi: 10.1175/1520-0493(1972)100⟨0136:CFAOTP⟩2.3.CO;2

Sadourny, R., Arakawa, A., & Mintz, Y. (1968, 6). Integration of the nondivergent barotropic vorticity equation with an icosahedral-hexagonal grid for the sphere. *Monthly Weather Review*, *96*(6), 351–356. Retrieved from `http://journals.ametsoc.org/doi/abs/10.1175/1520-0493(1968)096%3C0351:IOTNBV%3E2.0.CO;2` doi: 10.1175/1520-0493(1968)096⟨0351:IOTNBV⟩2.0.CO;2

Satoh, M., Stevens, B., Judt, F., Khairoutdinov, M., Lin, S.-J., Putman, W. M., & Düben, P. (2019, 5). Global Cloud-Resolving Models. *Current Climate Change Reports*. Retrieved from `http://link.springer.com/10.1007/s40641-019-00131-0` doi: 10.1007/s40641-019-00131-0

Stevens, B., Satoh, M., Auger, L., Biercamp, J., Bretherton, C. S., Chen, X., ... Zhou, L. (2019, 9). DYAMOND: the DYnamics of the Atmospheric general circulation Modeled On Non-hydrostatic Domains. *Progress in Earth and Planetary Science*, *6*(1), 61. Retrieved from `https://progearthplanetsci.springeropen.com/articles/10.1186/s40645-019-0304-z` doi: 10.1186/s40645-019-0304-z

Taylor, M. A., & Fournier, A. (2010, 8). A compatible and conservative spectral element method on unstructured grids. *Journal of Computational Physics*, *229*(17), 5879–5895. Retrieved from `https://linkinghub.elsevier.com/retrieve/pii/S0021999110001841` doi: 10.1016/j.jcp.2010.04.008

Taylor, M. A., Guba, O., Steyer, A., Ullrich, P. A., Hall, D. M., & Eldrid, C. (2020, 1). An Energy Consistent Discretization of the Nonhydrostatic Equations in Primitive Variables. *Journal of Advances in Modeling Earth Systems*, *12*(1). Retrieved from `https://onlinelibrary.wiley.com/doi/abs/10.1029/2019MS001783` doi: 10.1029/2019MS001783

Tomita, H., Tsugawa, M., Satoh, M., & Goto, K. (2001, 12). Shallow Water Model on a Modified Icosahedral Geodesic Grid by Using Spring Dynamics. *Journal of Computational Physics*, *174*(2), 579–613. Retrieved from `https://linkinghub.elsevier.com/retrieve/pii/S0021999101968974` doi: 10.1006/jcph.2001.6897

Ullrich, P. A., Jablonowski, C., & van Leer, B. (2010, 8). High-order finite-volume methods for the shallow-water equations on the sphere. *Journal of Computational Physics*, *229*(17), 6104–6134. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0021999110002317` doi: 10.1016/j.jcp.2010.04.044

van Leer, B. (1979, 7). Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics*, *32*(1), 101–136. Retrieved from `http://linkinghub.elsevier.com/retrieve/pii/0021999179901451` doi: 10.1016/0021-9991(79)90145-1

Wang, L., Zhang, Y., Li, J., Liu, Z., & Zhou, Y. (2019, 12). Understanding the Performance of an Unstructured-Mesh Global Shallow Water Model on Kinetic Energy Spectra and Nonlinear Vorticity Dynamics. *Journal of Meteorological Research*, *33*(6), 1075–1097. Retrieved from `http://link.springer.com/10.1007/s13351-019-9004-2` doi: 10.1007/s13351-019-9004-2

Williamson, D. L. (1968, 11). Integration of the barotropic vorticity equation on a spherical geodesic grid. *Tellus*, *20*(4), 642–653. Retrieved from `http://tellusa.net/index.php/tellusa/article/view/10044` doi: 10.1111/j.2153-3490.1968.tb00406.x

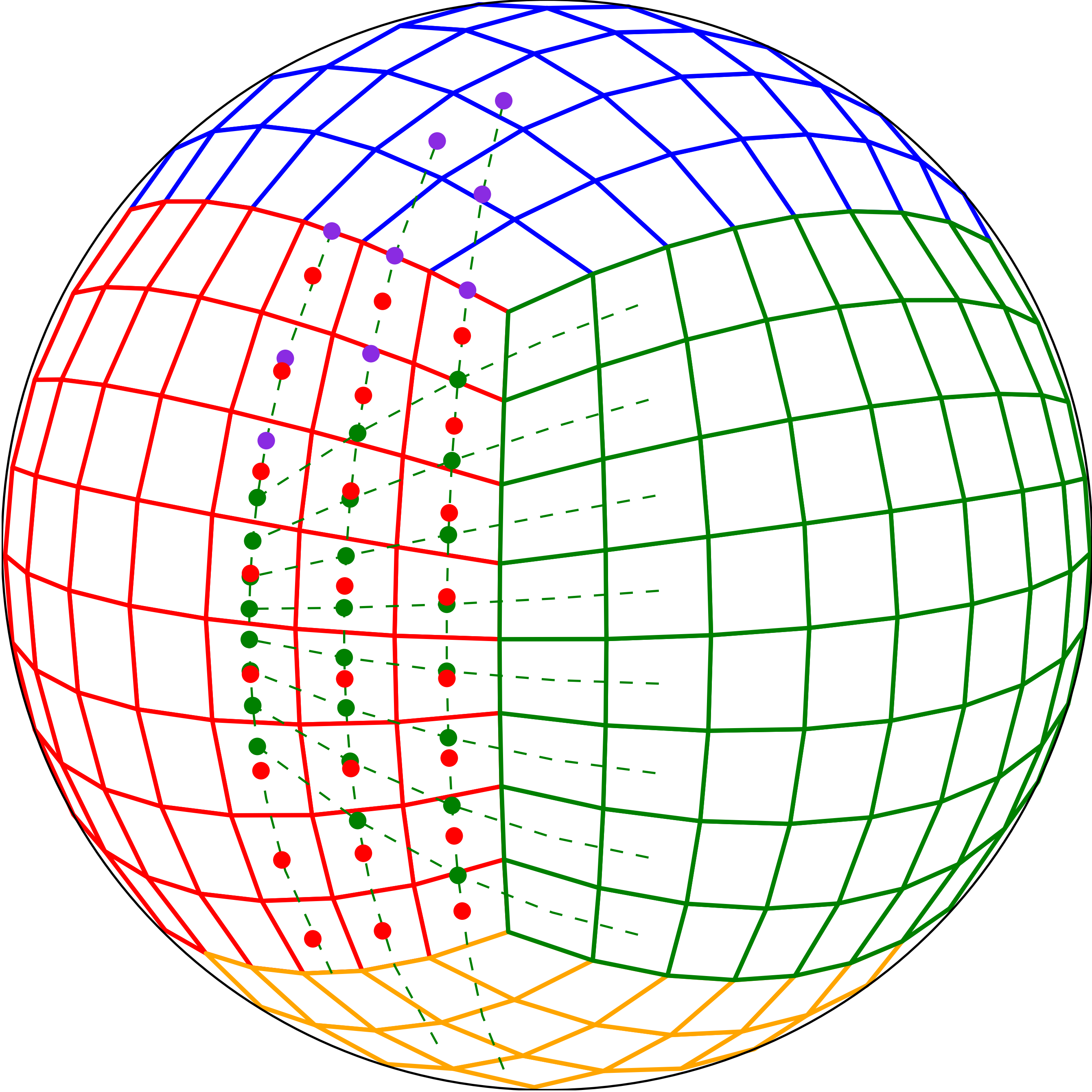
Williamson, D. L., Drake, J. B., Hack, J. J., Jakob, R., & Swarztrauber, P. N. (1992, 9). A standard test set for numerical approximations to the shallow water equations in spherical geometry. *Journal of Computational Physics*, *102*(1), 211–224. Retrieved from `https://linkinghub.elsevier.com/retrieve/pii/S0021999105800166` doi: 10.1016/S0021-9991(05)80016-6

Yang, C., Cao, J., & Cai, X.-C. (2010, 1). A Fully Implicit Domain Decomposition Algorithm for Shallow Water Equations on the Cubed-Sphere. *SIAM Journal on Scientific Computing*, *32*(1), 418–438. Retrieved from `http://epubs.siam.org/doi/abs/10.1137/080727348` doi: 10.1137/080727348

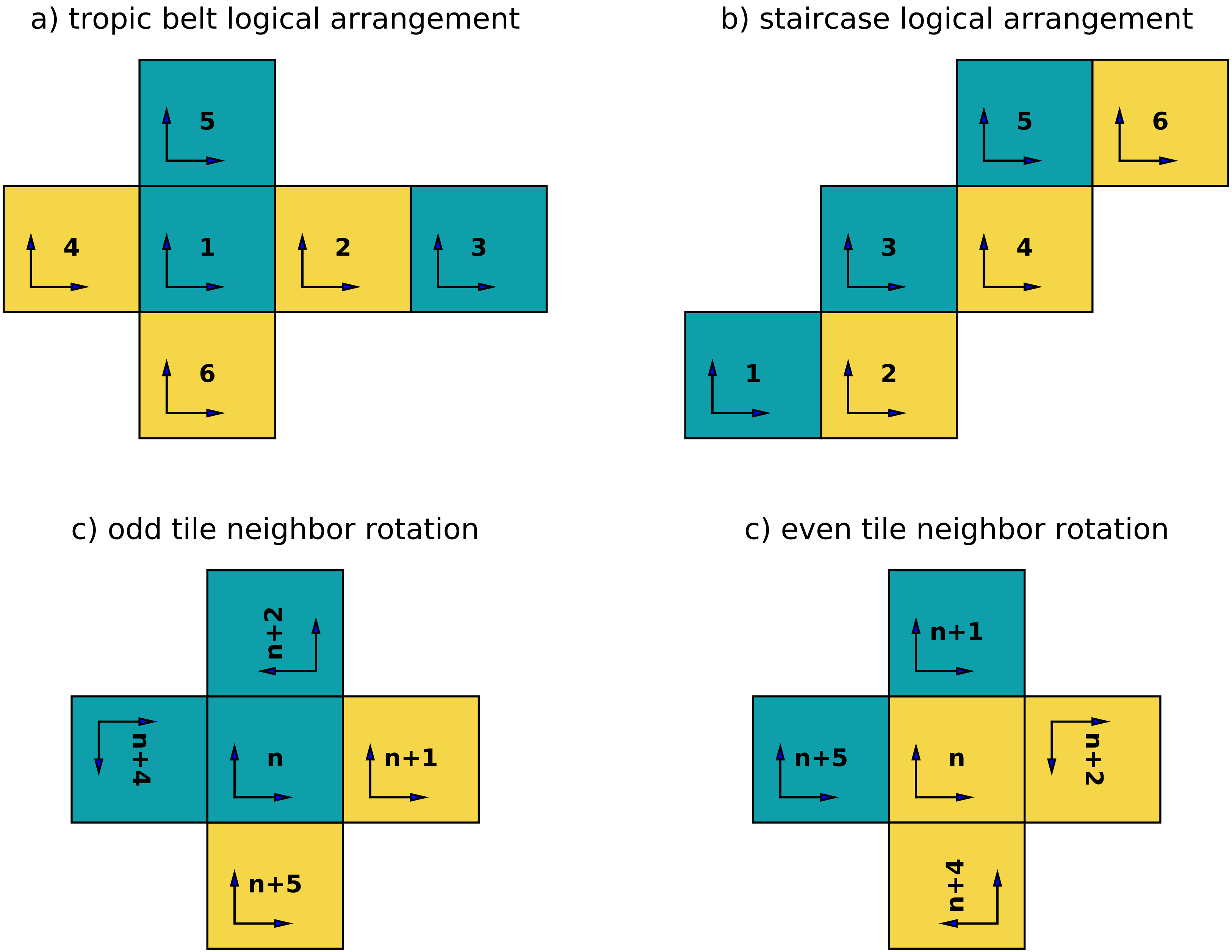
Zhang, Y., Li, J., Yu, R., Zhang, S., Liu, Z., Huang, J., & Zhou, Y. (2019, 6). A Layer-Averaged Nonhydrostatic Dynamical Framework on an Unstructured Mesh for Global and Regional Atmospheric Modeling: Model Description, Baseline Evaluation, and Sensitivity Exploration. *Journal of Advances in Modeling Earth Systems*, *11*(6), 1685–1714. Retrieved from `https://onlinelibrary.wiley.com/doi/abs/10.1029/2018MS001539` doi: 10.1029/2018MS001539

Zhou, L., Lin, S.-J., Chen, J.-H., Harris, L. M., Chen, X., & Rees, S. L. (2019, 7). Toward Convective-Scale Prediction within the Next Generation Global Prediction System. *Bulletin of the American Meteorological Society*, *100*(7), 1225–1243. Retrieved from `http://journals.ametsoc.org/doi/10.1175/BAMS-D-17-0246.1` doi: 10.1175/BAMS-D-17-0246.1

1.



2.



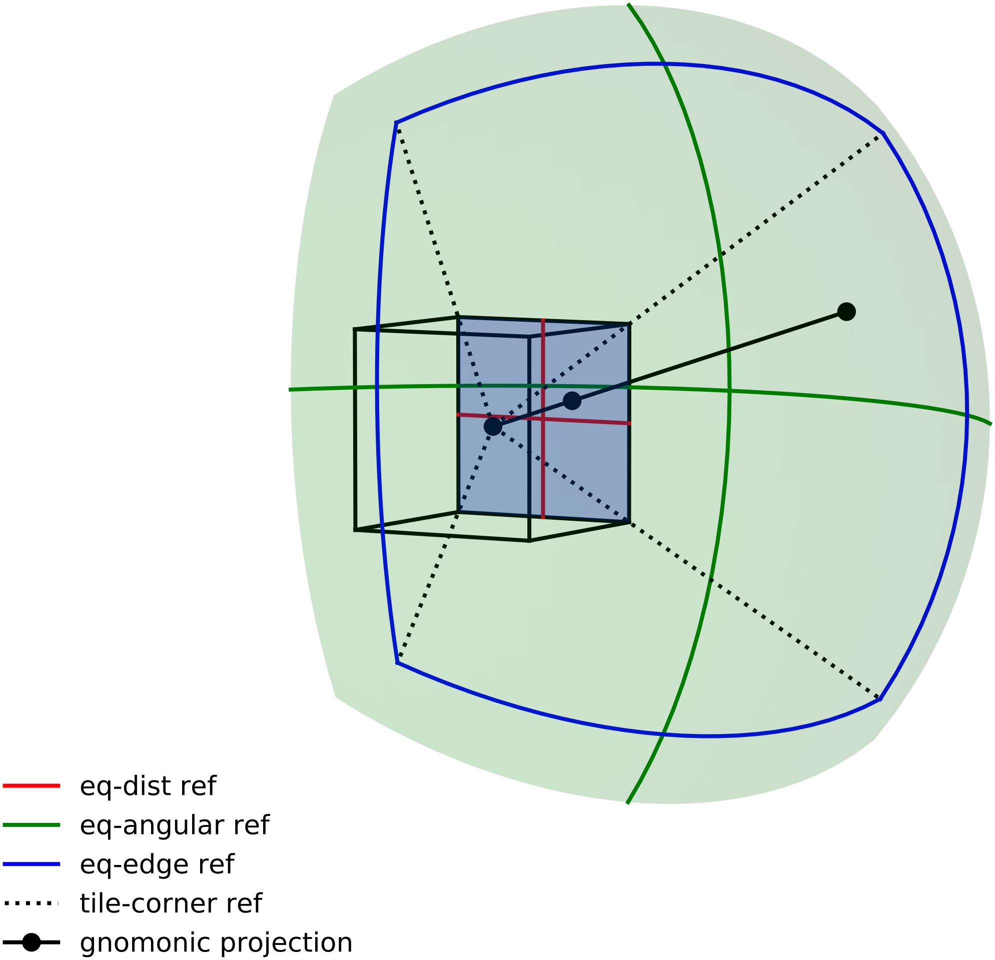
a) tropic belt logical arrangement

b) staircase logical arrangement

c) odd tile neighbor rotation

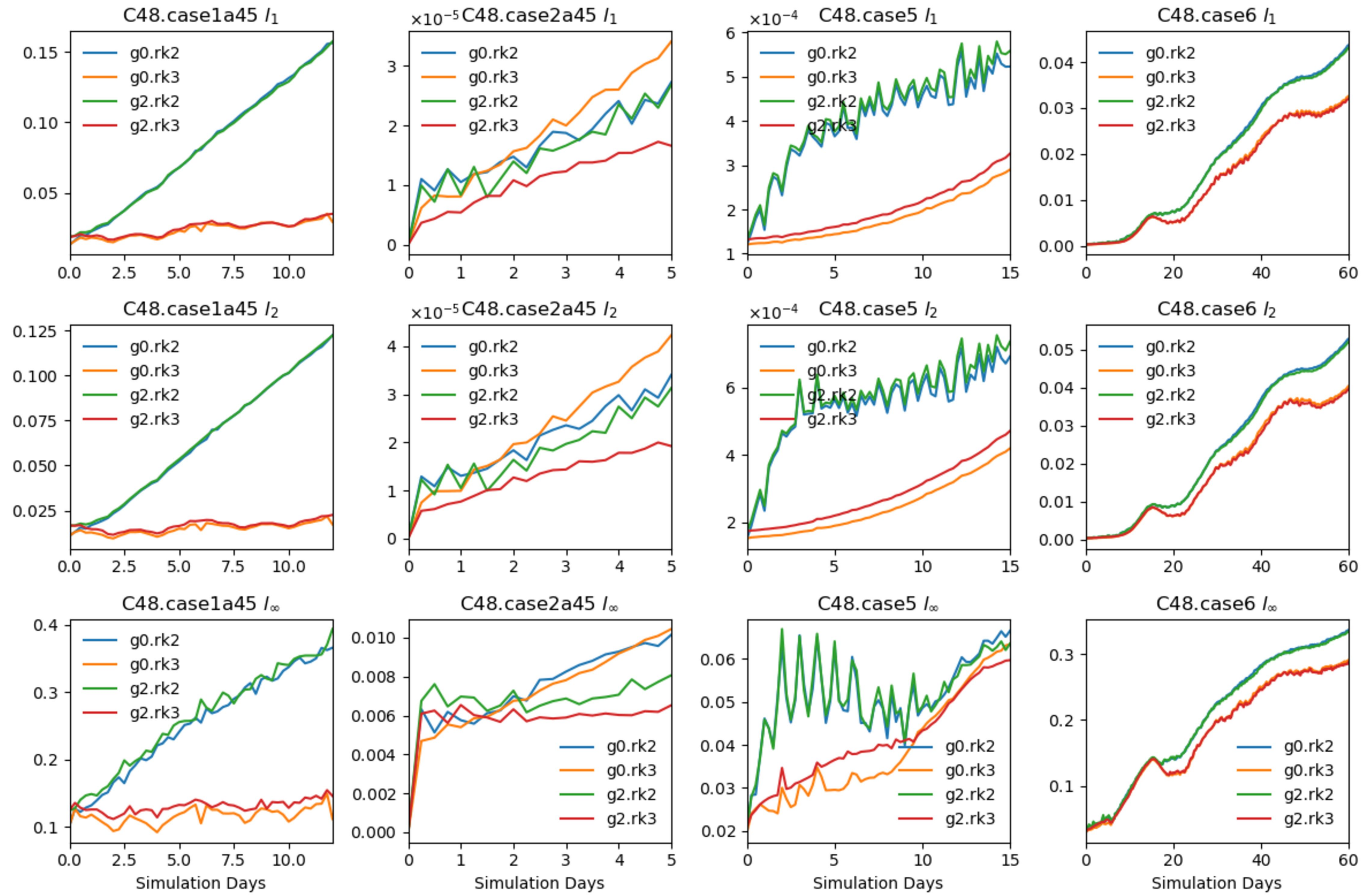
c) even tile neighbor rotation

3.

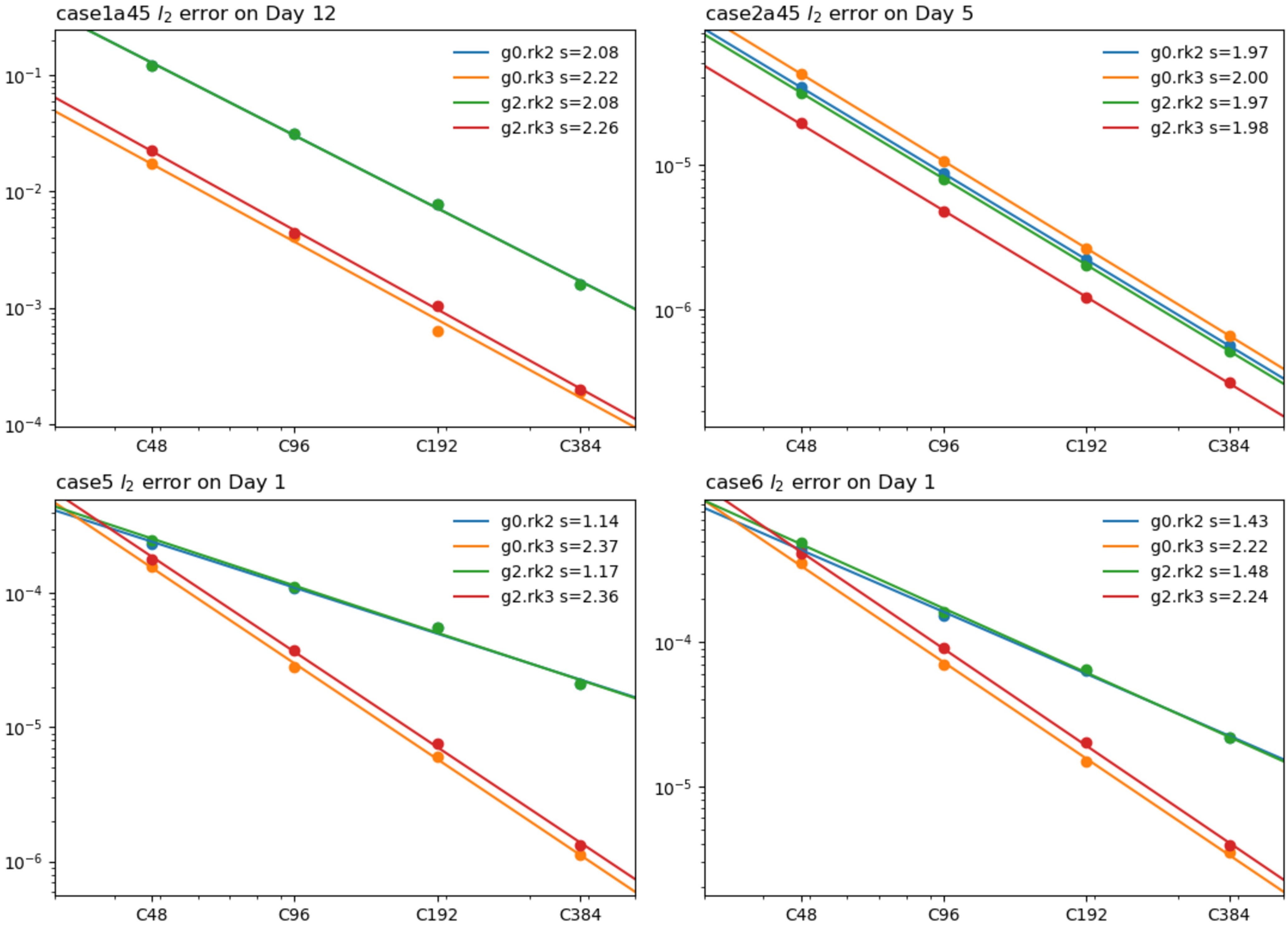


eq-dist ref
eq-angular ref
eq-edge ref
tile-corner ref
gnomonic projection

4.


C48.case1a45 l_1
0.15
0.10
0.05
0.0
2.5
5.0
7.5
10.0
g0.rk2
g0.rk3
g2.rk2
g2.rk3
$\times10^{-5}$ C48.case2a45 l_1
3
2
1
0
0
1
2
3
4
5
$\times10^{-4}$ C48.case5 l_1
6
5
4
3
2
1
0
5
10
15
C48.case6 l_1
0.04
0.03
0.02
0.01
0.00
0
20
40
60
C48.case1a45 l_2
0.125
0.100
0.075
0.050
0.025
$\times10^{-5}$ C48.case2a45 l_2
4
3
2
1
0
$\times10^{-4}$ C48.case5 l_2
6
4
2
C48.case6 l_2
0.05
0.04
0.03
0.02
0.01
0.00
C48.case1a45 l_∞
0.4
0.3
0.2
0.1
C48.case2a45 l_∞
0.010
0.008
0.006
0.004
0.002
0.000
C48.case5 l_∞
0.06
0.05
0.04
0.03
0.02
C48.case6 l_∞
0.3
0.2
0.1
Simulation Days
Simulation Days
Simulation Days
Simulation Days

5.

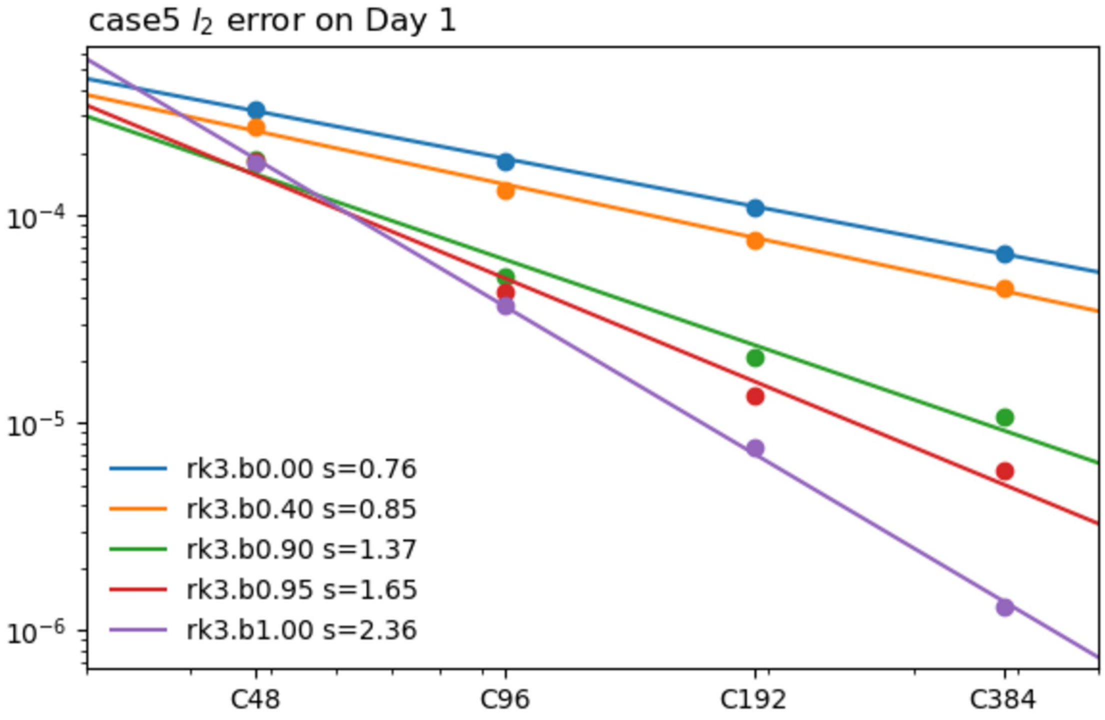

case1a45 l_2 error on Day 12

case2a45 l_2 error on Day 5

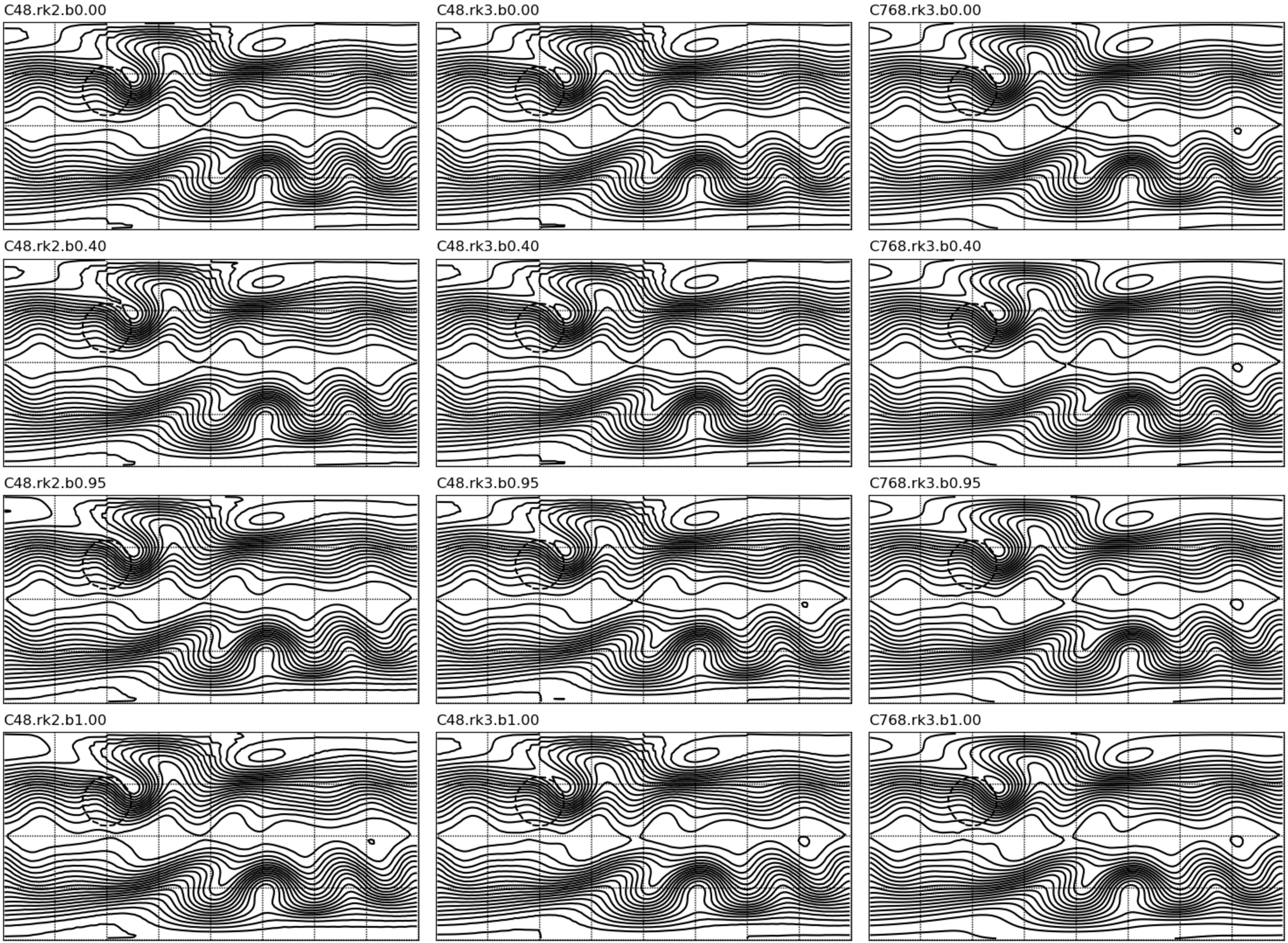
case5 l_2 error on Day 1

case6 l_2 error on Day 1

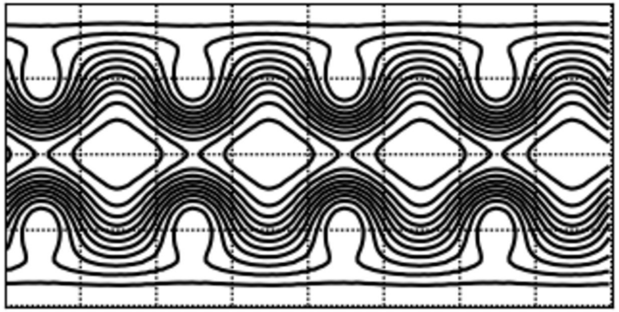
6.

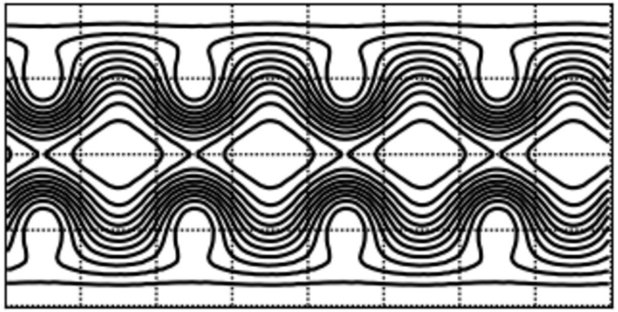

case5 l_2 error on Day 1
10^{-4}
10^{-5}
10^{-6}
C48
C96
C192
C384
rk3.b0.00 s=0.76
rk3.b0.40 s=0.85
rk3.b0.90 s=1.37
rk3.b0.95 s=1.65
rk3.b1.00 s=2.36

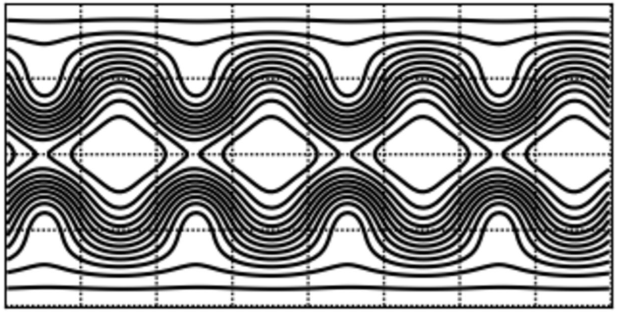
7.

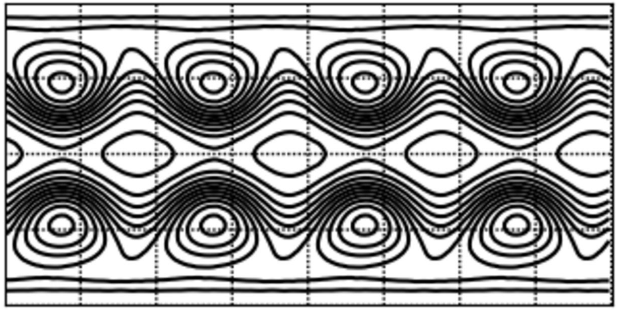

C48.rk2.b0.00
C48.rk3.b0.00
C768.rk3.b0.00
C48.rk2.b0.40
C48.rk3.b0.40
C768.rk3.b0.40
C48.rk2.b0.95
C48.rk3.b0.95
C768.rk3.b0.95
C48.rk2.b1.00
C48.rk3.b1.00
C768.rk3.b1.00

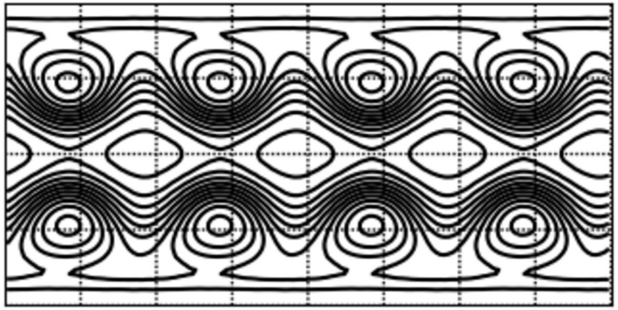
8.

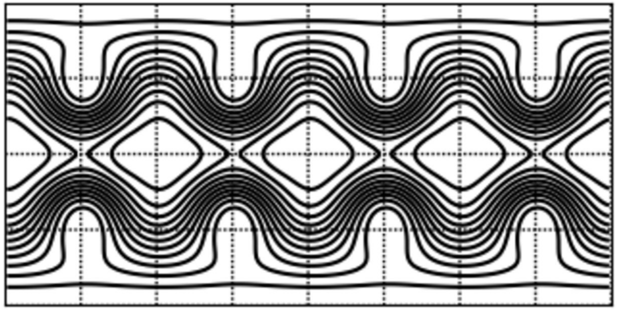

C48.g2.rk2 Height Field - Day 14

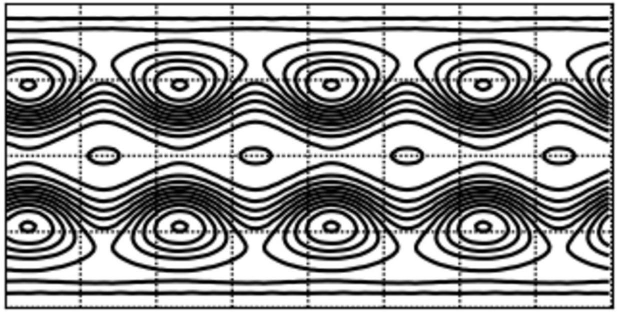

C48.g2.rk3 Height Field - Day 14

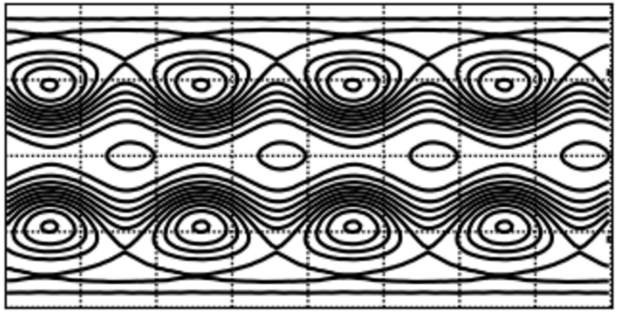

C768.g2.rk3 Height Field - Day 14

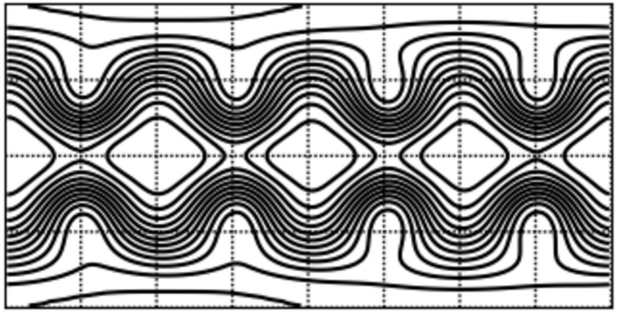

C48.g2.rk2 Height Field - Day 40


C48.g2.rk3 Height Field - Day 40

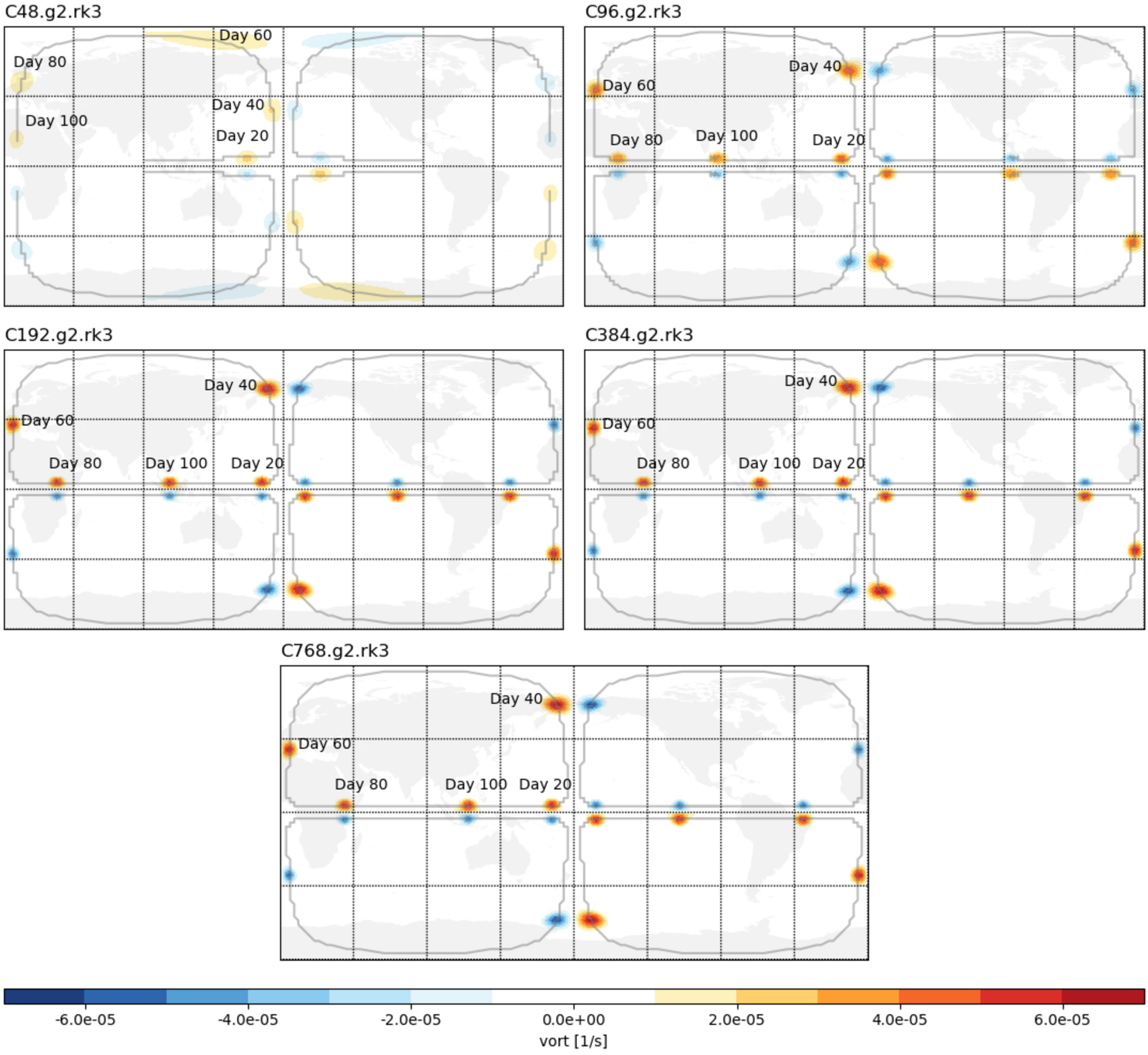

C768.g2.rk3 Height Field - Day 40


C48.g2.rk2 Height Field - Day 80

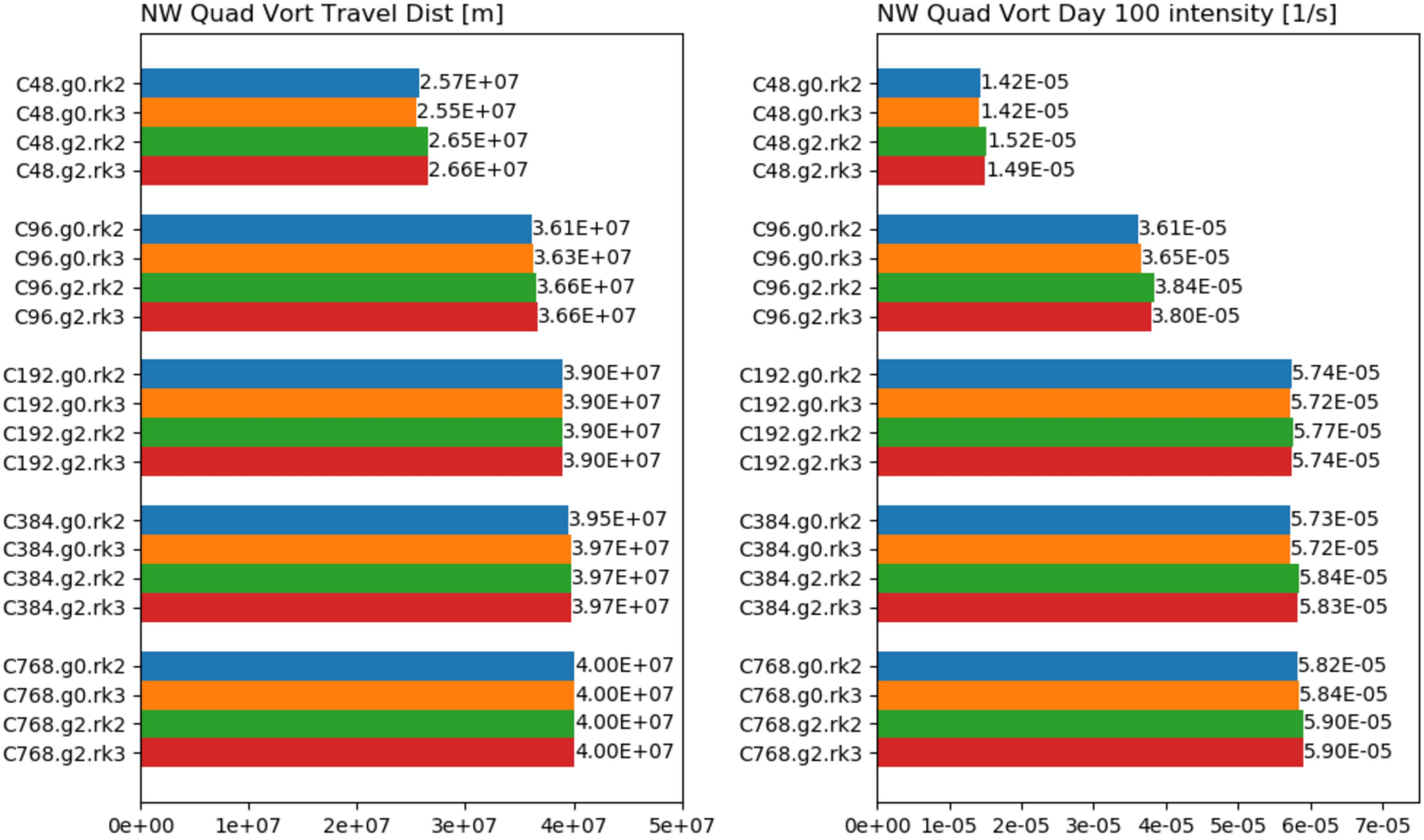

C48.g2.rk3 Height Field - Day 80


C768.g2.rk3 Height Field - Day 80

9.


C48.g2.rk3
Day 60
Day 80
Day 40
Day 100
Day 20
C96.g2.rk3
Day 40
Day 60
Day 80
Day 100
Day 20
C192.g2.rk3
Day 40
Day 60
Day 80
Day 100
Day 20
C384.g2.rk3
Day 40
Day 60
Day 80
Day 100
Day 20
C768.g2.rk3
Day 40
Day 60
Day 80
Day 100
Day 20
-6.0e-05
-4.0e-05
-2.0e-05
0.0e+00
2.0e-05
4.0e-05
6.0e-05
vort [1/s]

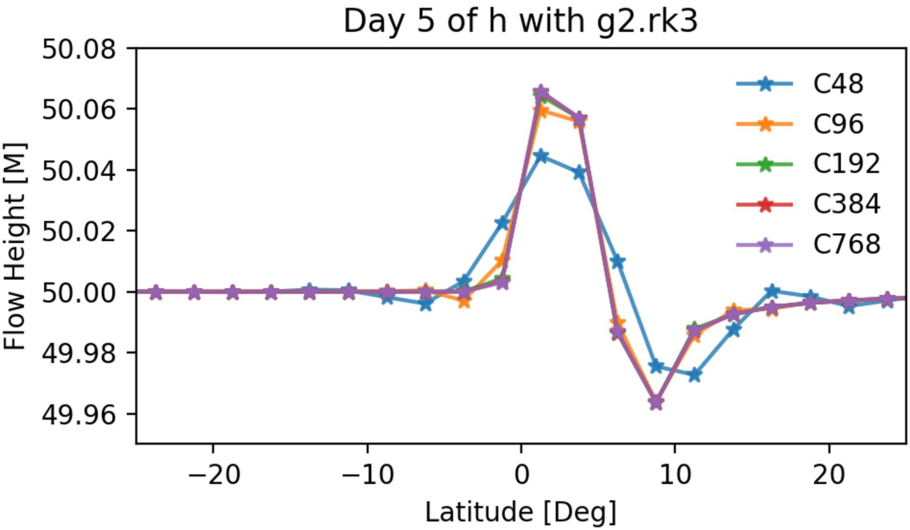
10.


NW Quad Vort Travel Dist [m]
C48.g0.rk2
C48.g0.rk3
C48.g2.rk2
C48.g2.rk3
C96.g0.rk2
C96.g0.rk3
C96.g2.rk2
C96.g2.rk3
C192.g0.rk2
C192.g0.rk3
C192.g2.rk2
C192.g2.rk3
C384.g0.rk2
C384.g0.rk3
C384.g2.rk2
C384.g2.rk3
C768.g0.rk2
C768.g0.rk3
C768.g2.rk2
C768.g2.rk3
2.57E+07
2.55E+07
2.65E+07
2.66E+07
3.61E+07
3.63E+07
3.66E+07
3.66E+07
3.90E+07
3.90E+07
3.90E+07
3.90E+07
3.95E+07
3.97E+07
3.97E+07
3.97E+07
4.00E+07
4.00E+07
4.00E+07
4.00E+07
0e+00
1e+07
2e+07
3e+07
4e+07
5e+07
NW Quad Vort Day 100 intensity [1/s]
C48.g0.rk2
C48.g0.rk3
C48.g2.rk2
C48.g2.rk3
C96.g0.rk2
C96.g0.rk3
C96.g2.rk2
C96.g2.rk3
C192.g0.rk2
C192.g0.rk3
C192.g2.rk2
C192.g2.rk3
C384.g0.rk2
C384.g0.rk3
C384.g2.rk2
C384.g2.rk3
C768.g0.rk2
C768.g0.rk3
C768.g2.rk2
C768.g2.rk3
1.42E-05
1.42E-05
1.52E-05
1.49E-05
3.61E-05
3.65E-05
3.84E-05
3.80E-05
5.74E-05
5.72E-05
5.77E-05
5.74E-05
5.73E-05
5.72E-05
5.84E-05
5.83E-05
5.82E-05
5.84E-05
5.90E-05
5.90E-05
0e+00
1e-05
2e-05
3e-05
4e-05
5e-05
6e-05
7e-05

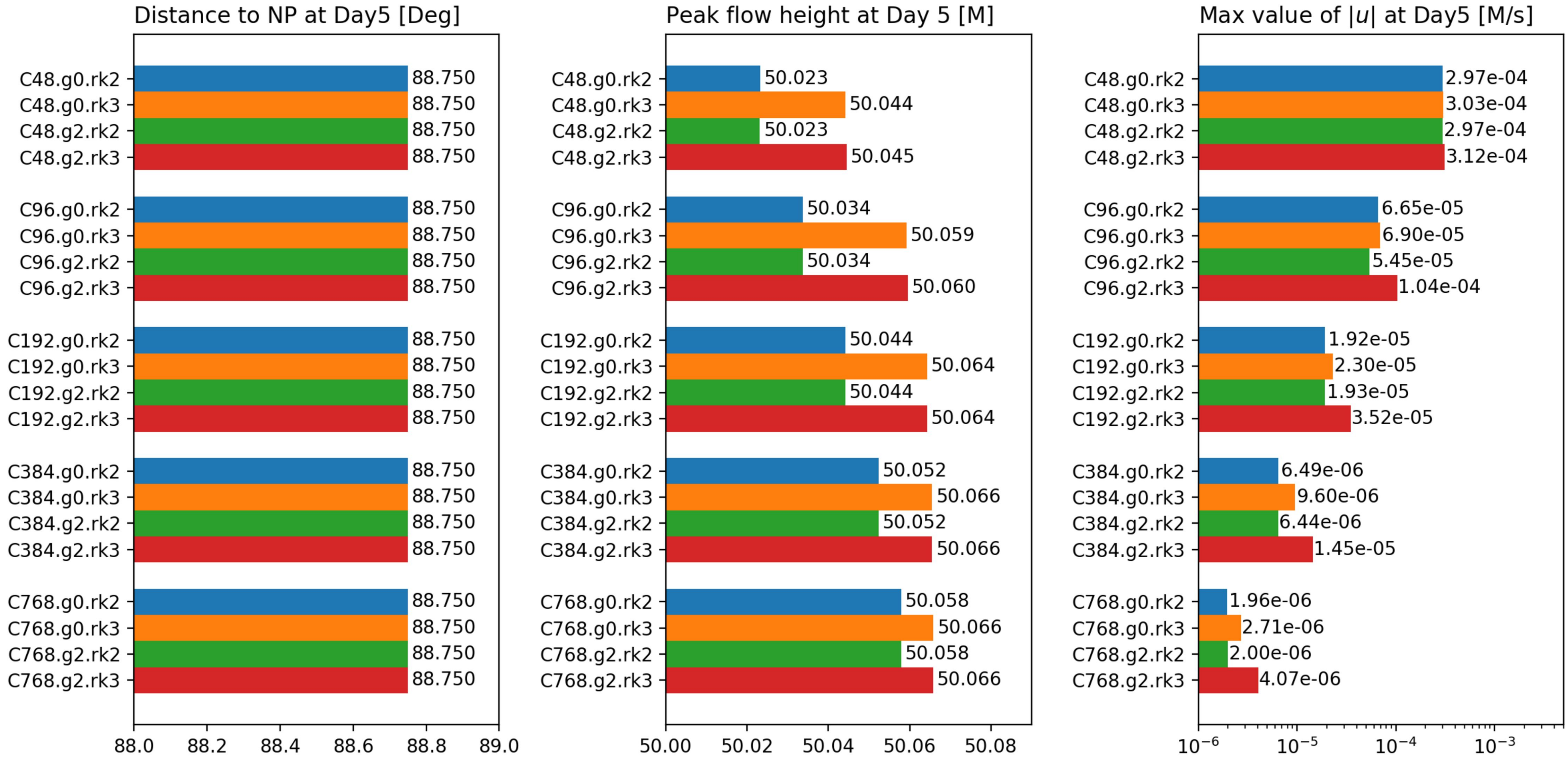
11.

Day 0
Day 2
Day 5
Day 7

12.


Day 5 of h with g2.rk3
Flow Height [M]
50.08
50.06
50.04
50.02
50.00
49.98
49.96
−20
−10
0
10
20
Latitude [Deg]
C48
C96
C192
C384
C768

13.


Distance to NP at Day5 [Deg]
C48.g0.rk2 88.750
C48.g0.rk3 88.750
C48.g2.rk2 88.750
C48.g2.rk3 88.750
C96.g0.rk2 88.750
C96.g0.rk3 88.750
C96.g2.rk2 88.750
C96.g2.rk3 88.750
C192.g0.rk2 88.750
C192.g0.rk3 88.750
C192.g2.rk2 88.750
C192.g2.rk3 88.750
C384.g0.rk2 88.750
C384.g0.rk3 88.750
C384.g2.rk2 88.750
C384.g2.rk3 88.750
C768.g0.rk2 88.750
C768.g0.rk3 88.750
C768.g2.rk2 88.750
C768.g2.rk3 88.750
88.0
88.2
88.4
88.6
88.8
89.0
Peak flow height at Day 5 [M]
C48.g0.rk2 50.023
C48.g0.rk3 50.044
C48.g2.rk2 50.023
C48.g2.rk3 50.045
C96.g0.rk2 50.034
C96.g0.rk3 50.059
C96.g2.rk2 50.034
C96.g2.rk3 50.060
C192.g0.rk2 50.044
C192.g0.rk3 50.064
C192.g2.rk2 50.044
C192.g2.rk3 50.064
C384.g0.rk2 50.052
C384.g0.rk3 50.066
C384.g2.rk2 50.052
C384.g2.rk3 50.066
C768.g0.rk2 50.058
C768.g0.rk3 50.066
C768.g2.rk2 50.058
C768.g2.rk3 50.066
50.00
50.02
50.04
50.06
50.08
Max value of |u| at Day5 [M/s]
C48.g0.rk2 2.97e-04
C48.g0.rk3 3.03e-04
C48.g2.rk2 2.97e-04
C48.g2.rk3 3.12e-04
C96.g0.rk2 6.65e-05
C96.g0.rk3 6.90e-05
C96.g2.rk2 5.45e-05
C96.g2.rk3 1.04e-04
C192.g0.rk2 1.92e-05
C192.g0.rk3 2.30e-05
C192.g2.rk2 1.93e-05
C192.g2.rk3 3.52e-05
C384.g0.rk2 6.49e-06
C384.g0.rk3 9.60e-06
C384.g2.rk2 6.44e-06
C384.g2.rk3 1.45e-05
C768.g0.rk2 1.96e-06
C768.g0.rk3 2.71e-06
C768.g2.rk2 2.00e-06
C768.g2.rk3 4.07e-06
10^{-6}
10^{-5}
10^{-4}
10^{-3}