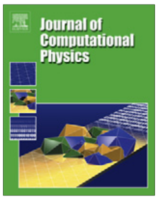
Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp



# MCore: A non-hydrostatic atmospheric dynamical core utilizing high-order finite-volume methods

Paul A. Ullrich *, Christiane Jablonowski

*Department of Atmospheric, Oceanic and Space Sciences, Space Research Building, University of Michigan, 2455 Hayward St., Ann Arbor, MI 48109, United States*

ARTICLE INFO

*Article history:*
Received 16 July 2011
Received in revised form 28 January 2012
Accepted 10 April 2012
Available online 28 April 2012

*Keywords:*
High-order
Finite-volume methods
Cubed-sphere
General circulation models
Dynamical cores
Atmospheric models

ABSTRACT

This paper presents a new atmospheric dynamical core which uses a high-order upwind finite-volume scheme of Godunov type for discretizing the non-hydrostatic equations of motion on the sphere under the shallow-atmosphere approximation. The model is formulated on the cubed-sphere in order to avoid polar singularities. An operator-split Runge–Kutta–Rosenbrock scheme is used to couple the horizontally explicit and vertically implicit discretizations so as to maintain accuracy in time and space and enforce a global CFL condition which is only restricted by the horizontal grid spacing and wave speed. The Rosenbrock approach is linearly implicit and so requires only one matrix solve per column per time step. Using a modified version of the low-speed AUSM$^{+}$-up Riemann solver allows us to construct the vertical Jacobian matrix analytically, and so significantly improve the model efficiency. This model is tested against a series of typical atmospheric flow problems to verify accuracy and consistency. The test results reveal that this approach is stable, accurate and effective at maintaining sharp gradients in the flow.

## 1. Introduction

In recent years, the exponential growth of computing power and trend towards massive parallelization of computing systems has had a profound influence on the atmospheric modeling community. Atmospheric cloud-resolving models are now pushing towards scales of only a few kilometers, meanwhile utilizing thousands to hundreds of thousands of processors. At these small scales many of the approximations that have been previously used in developing dynamical cores, such as the hydrostatic approximation, are no longer valid. As a consequence, there has been a trend towards developing atmospheric models which incorporate the full unapproximated hydrodynamic equations of motion. These developments have required a substantial paradigm shift in the way developers think about the algorithms and software behind geophysical models. Many design decisions that worked well in the past, including the use of the regular latitude-longitude grid and polar Fourier filtering, are no longer acceptable on large parallel systems since they either limit the choices for the parallel domain decompositions or necessitate additional parallel communication and thereby increase the computational overhead. Therefore, modifications must be made to accommodate this new generation of massively parallel hardware. As a consequence, the past ten years have seen substantial innovation in the modeling community as they push forward with efforts to determine the best candidates for the next-generation of atmospheric models.

Our focus in this paper is on non-hydrostatic modeling: That is, we are interested in models which treat the vertical velocity as a prognostic variable. In this case the vertical velocity has its own evolution equation and is not diagnosed from the other flow variables. Non-hydrostatic models are valid on essentially any horizontal scale and so can be used in

* Corresponding author.
*E-mail addresses:* paullric@umich.edu (P.A. Ullrich), cjablono@umich.edu (C. Jablonowski).

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.04.024

cloud-resolving simulations. Several non-hydrostatic models are now in use, having been largely developed in the past ten years in response to growing availability of computing power. These include the UK Met Office's Unified Model [6,41], the Non-hydrostatic ICosahedral Atmospheric Model (NICAM), which was developed in Tomita and Satoh [44] and Satoh et al. [38] in cooperation with the Center for Climate System Research (CCSR, Japan), the NOAA Non-hydrostatic Icosahedral Model (NIM) [15] and the Ocean–Land–Atmosphere Model (OLAM) [48]. Recently, the Geophysical Fluid Dynamics Laboratory (GFDL) and the NASA Goddard Space Flight Center (GSFC) have also developed a non-hydrostatic dynamical core on the cubed-sphere [8,30] based on the work of Putman and Lin [29]. These models all make use of some sort of conservative finite-difference or finite-volume formulation to ensure conservation of mass and most adopt the Arakawa C-grid staggering [2]. Other non-hydrostatic models include the Integrated Forecast System (IFS) [4,52] which is a semi-Lagrangian spectral transform model used at the European Centre for Medium-Range Weather Forecasts (ECMWF), the Canadian semi-Lagrangian Global Environmental Multiscale (GEM) model [57], the Model for Prediction Across Scales (MPAS) [39] under development at the National Center for Atmospheric Research (NCAR) and Los Alamos National Laboratory, and the 'ICOsahedral Non-hydrostatic' (ICON) model [11,49] which is a joint model by the Max-Planck Institute for Meteorology (MPI-M) and the German Weather Service.

In developing models for large-scale parallel computers, the choice of grid is of particular importance. Although non-hydrostatic dynamical cores have been developed on the regular latitude-latitude (RLL) grid, including the UK Met Office model, it is well known that the RLL grid suffers from the convergence of grid lines at the north and south poles. As a consequence, models using the RLL grid require the use of polar filters to remove instabilities associated with small grid elements, which can in turn severely degrade performance on parallel systems. Many recently developed hydrostatic and non-hydrostatic models have tended away from this grid, instead using quasi-uniform grids such as the icosahedral or cubed-sphere grids. Several hydrostatic models are now built on geodesic grids, including the icosahedral German Weather Service model GME [24,25] and the icosahedral-hexagonal model of Ringler et al. [32]. Non-hydrostatic models that use the icosahedral grid include NICAM, NIM, OLAM and ICON. The icosahedral grid has been shown to perform well on large parallel systems and is among the most uniform options for spherical grids. Another choice of quasi-uniform grid is the cubed-sphere grid, which was originally developed by Sadourny [37] and revived by Ronchi et al. [34]. In fact, the work of Ronchi et al. [34] introduced a precursor to some of the techniques described in this paper, including the fourth-order collocated stencils and treatment of the cubed-sphere edges. The cubed-sphere was later used as the basis for a shallow-water model by Rančić et al. [31]. Since then, shallow-water models have been developed using the cubed-sphere grid that utilize finite-volume methods [35,46], multi-moment finite-volume [5], the discontinuous Galerkin method [26] and the spectral element method [42]. The spectral element method was successfully extended to a full hydrostatic atmospheric model (the Spectral Element Atmosphere Model, SEAM) [9,43], which is part of the High-Order Method Modeling Environment (HOMME). HOMME incorporates both the spectral element method and an experimental implementation of the discontinuous Galerkin method, and has proven to scale efficiently to hundreds of thousands of processors [7]. Recently HOMME has become an optional dynamical core in the Community Atmosphere Model version 5 (CAM5) [27] which is under development at the NCAR and Sandia National Laboratories. The GFDL/NASA finite-volume dynamical core has been modified to use a cubed-sphere grid [29], and has been demonstrated to also be very effective at high resolutions.

This paper continues a series that describes the development of an atmospheric model based on unstaggered high-order finite-volume methods. In Ullrich et al. [46] a shallow-water model utilizing cell-centered third-and fourth-order finite-volume methods was described. This approach was demonstrated to be robust and highly competitive with existing methods when tested against the shallow-water test cases of [55]. The high-order finite-volume method was later extended to non-hydrostatic simulations in 3D Cartesian geometry in Ullrich and Jablonowski [45]. Therein the authors demonstrated an accuracy-preserving technique for splitting horizontal and vertical motions using interleaved explicit and implicit time steps. The work of this paper is a combination of Ullrich et al. [46] and Ullrich and Jablonowski [45], and describes the high-order finite-volume formulation in spherical geometry. The fluid model that arises from this work has been named MCore. MCore is an atmospheric dynamical core that provides support for both the shallow-water equations and the full non-hydrostatic fluid equations. However, our emphasis in this paper will be on the non-hydrostatic dynamical core under the shallow-atmosphere approximation.

MCore makes use of a fully Eulerian cell-centered finite-volume formulation that has been proven to be robust for problems from other fields, including aerospace, computational biology and high-energy physics. It uses a fourth-order-accurate numerical method in the horizontal and a second-order accurate method in the vertical. However, it should be noted that although MCore is designed to be fourth-order-accurate in the horizontal, this design decision has been made largely to ensure that the model accurately captures the correct dispersive behavior of low-to mid-frequency horizontally propagating waves. We do not generally expect to obtain fourth-order convergence except in very idealized circumstances.

The outline of this paper is as follows. In Section 2 we introduce the cubed-sphere grid, which underlies the MCore model. The non-hydrostatic fluid equations under the shallow-atmosphere approximation are introduced in Section 3. The numerical approach underlying the MCore model is presented in Section 4. Numerical results and test cases are described in Section 5. Finally, our conclusions and future work are discussed in Section 6. A list of variables used in this paper can be found in Table 1. A list of the constants used in this paper and their corresponding values can be found in Table 2. The appendices describe the mathematical formulation of the primitive equations in cubed-sphere coordinates, under both the deep-and shallow-atmosphere approximation. Throughout this paper we will make use of Einstein summation notation, especially when describing geometric relations, under which summation is implied over repeated indices.

**Table 1**
List of variables used in this paper.

| Constant | Description | Units |
|---|---|---|
| $\rho$ | Density | kg/m$^3$ |
| $\mathbf{u}$ | Velocity vector in cubed-sphere coordinates | m/s |
| $\theta$ | Potential temperature | K |
| $p$ | Pressure | kg/m/s$^2$ |
| $N_c$ | Indicator for horizontal cubed-sphere resolution | – |
| $N_r$ | Radial (vertical) resolution (number of vertical levels) | – |
| $n_p$ | Panel indicator ($n_p \in \{1,2,3,4,5,6\}$) | – |
| $\mathbf{g}_\alpha$ | Basis vector in the $\alpha$ direction | – |
| $\mathbf{g}_\beta$ | Basis vector in the $\beta$ direction | – |
| $\mathbf{g}_r$ | Basis vector in the $r$ direction | – |
| $G^{ij}$ | Contravariant metric tensor | – |
| $G_{ij}$ | Covariant metric tensor | – |
| $J$ | Metric Jacobian (square root of metric determinant) | m$^2$ |
| $r_s(\alpha,\beta,n_p)$ | Radius of the surface | m |
| $r_T$ | Radius of the model top | m |

**Table 2**
List of physical constants used in this paper.

| Constant | Description | Control value |
|---|---|---|
| $a$ | Radius of the Earth | $6.37122 \times 10^6$ m |
| $\omega$ | Rotational speed of the Earth | $7.292 \times 10^{-5}$ s$^{-1}$ |
| $g$ | Gravity | 9.80616 m s$^{-2}$ |
| $p_0$ | Reference pressure | 1000 hPa |
| $c_p$ | Specific heat capacity of dry air at constant pressure | 1004.5 J kg$^{-1}$ K$^{-1}$ |
| $c_v$ | Specific heat capacity of dry air at constant volume | 717.5 J kg$^{-1}$ K$^{-1}$ |
| $R_d$ | Ideal gas constant of dry air | 287.0 J kg$^{-1}$ K$^{-1}$ |

## 2. The cubed-sphere

The MCore model is implemented on a cubed-sphere grid, which can be imagined as the product of projecting a cube with regularly gridded faces onto the surface of a sphere. The cubed-sphere grid was originally suggested by Sadourny [37], but was not used in developing full geophysical codes until the work of Ronchi et al. [34]. There are several advantages to the cubed-sphere grid, such as grid regularity on each panel. Further, the cubed-sphere grid avoids the so-called "pole-problem," which refers to the issues associated with convergence of grid lines at the poles on a latitude-longitude projection. The delay between introduction and implementation of this grid was largely due to the fact that the overhead required in storing and computationally maintaining the grid did not outweigh its potential advantages. However, in an era where computational power is increasing exponentially and more computations must utilize large parallel systems, the inherent regularity of the cubed-sphere grid makes it an attractive option for next-generation models.

Mathematically, the cubed-sphere grid consists of six panels that form the face of a cube projected onto the surface of a concentric sphere. Several options are available for tiling each cube panel (see, for instance, Putman and Lin [29] for a review of the types of cubed-sphere grids). MCore makes use of the gnomonic (equiangular) cubed-sphere grid, which consists of grid lines that have equal central angles relative to the center of the sphere (this property is also held by lines of constant longitude on a latitude-longitude projection). This choice does not lead to a perfectly uniform tiling of the sphere; instead, as resolution increases the ratio of the area of the smallest grid element to largest grid element approaches $1/\sqrt{2} \approx 0.707$. Nonetheless, one powerful advantage of this choice is that grid lines parallel to panel edges are coincident across panels. A depiction of the cubed-sphere grid and panel edges is given in Fig. 1.

Equiangular cubed-sphere coordinates are generally given in terms of the component vector $(r,\alpha,\beta,n_p)$, with $\alpha,\beta \in \left[-\frac{\pi}{4},\frac{\pi}{4}\right]$ denoting the horizontal coordinate on each panel, $r$ denoting the radial distance to the center of the sphere and $n_p \in \{1,2,3,4,5,6\}$ denoting the panel index. By convention, we choose panels 1–4 to be along the equator, with panels 5 and 6 centered on the northern and southern pole, respectively. A closely related set of coordinates are gnomonic cubed-sphere coordinates $(r,X,Y,n_p)$, which are related to equiangular coordinates via the transformation

$$X = \tan\alpha, \quad Y = \tan\beta. \tag{1}$$

Gnomonic coordinates have the property that any straight line in a gnomonic projection forms the arc of a great circle. Geometric quantities are often more simply written in terms of gnomonic variables, and so we will use these two sets of coordinates interchangeably. In this paper we will also make use of the definition

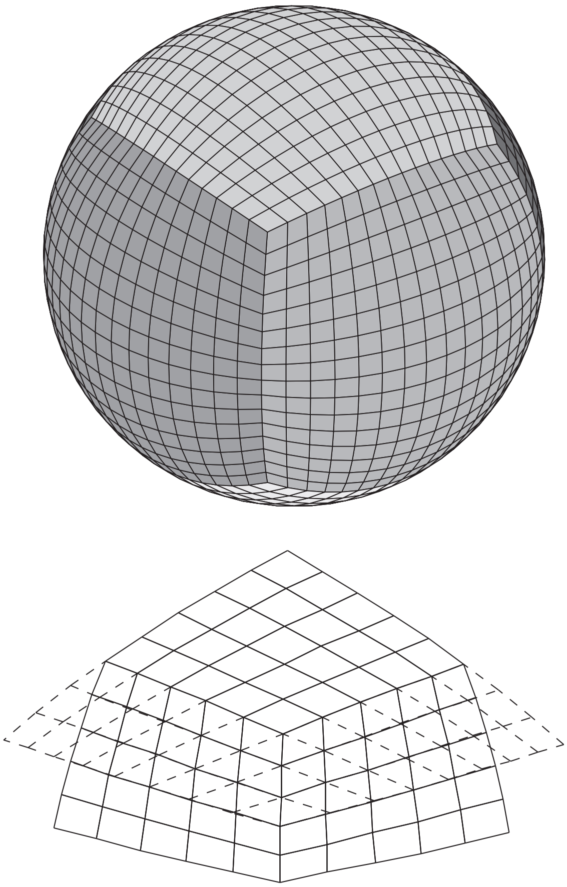


**Fig. 1.** Left: a 3D view of the tiling of the cubed-sphere along surfaces of constant radius, shown here with a 16 × 16 tiling on each panel. Right: a close-up view of one of the cubed-sphere corners, also showing the "halo region" of the upper panel, which consists of elements which have been extended into neighboring panels (dashed line).

$$\delta = \sqrt{1 + X^2 + Y^2}, \tag{2}$$

which appears frequently in the calculation of metric quantities associated with the cubed-sphere.

Uniform grid elements on the equiangular cubed-sphere form squares in the $(\alpha, \beta)$ plane, consisting of arcs of uniform angle $\Delta\alpha$. The discrete integer resolution on a uniform mesh is denoted by $N_c$, and related to $\Delta\alpha$ via

$$\Delta\alpha = \frac{\pi}{2N_c}. \tag{3}$$

raditionally, the resolution of the cubed-sphere grid is compactly denoted by $c\langle N_c \rangle$, where $N_c$ describes the number of grid cells in each horizontal direction on each panel. Hence, the total number of grid cells is equal to $6 \times N_c^2$ at each model level. Grid spacing in the radial direction can be chosen arbitrarily, independent of the horizontal grid. A list of some properties of the cubed-sphere grid is given in Table 3. The table lists the approximate equatorial spacing of grid elements, average area per element on the sphere, maximum area ratio and equivalent model resolutions of the

**Table 3**
Properties of the cubed-sphere grid for different resolutions. Here $\Delta x$ is the grid spacing at the equator, $A_{avg}$ is the average area of all cubed-sphere grid elements, $A_{min}$ is the minimum element area and $A_{max}$ is the maximum element area. $RLL_{equiv}$ denotes the equivalent grid spacing (in degrees) on the regular latitude-longitude grid with the same number of elements and $T_{equiv}$ denotes the approximate triangular truncation of a spectral transform method.

| Resolution | $\Delta x$ | $A_{avg}$ | $A_{min}/A_{max}$ | $RLL_{equiv}$ | $T_{equiv}$ |
|---|---|---|---|---|---|
| $c48$ | 208 km | $3.689 \times 10^4$ km$^2$ | 0.7189 | 2.16° | $T57$ |
| $c60$ | 167 km | $2.362 \times 10^4$ km$^2$ | 0.7165 | 1.73° | $T71$ |
| $c90$ | 111 km | $1.050 \times 10^4$ km$^2$ | 0.7133 | 1.15° | $T106$ |
| $c180$ | 55.6 km | $2.624 \times 10^3$ km$^2$ | 0.7102 | 0.58° | $T212$ |
| $c360$ | 27.8 km | $6.560 \times 10^2$ km$^2$ | 0.7087 | 0.29° | $T424$ |

regular latitude-longitude finite-volume (assuming an equal number of grid elements) and spectral transform models (under triangular truncation, as argued by Williamson [56]). Details of the geometric terms used in this reconstruction are given in Appendix A.

## 3. The non-hydrostatic fluid equations in cubed-sphere coordinates

MCore utilizes the full non-hydrostatic fluid equations in terms of conserved variables density $\rho$, momentum $\rho\mathbf{u}$ (with 3D velocity vector $\mathbf{u} = (u^\alpha, u^\beta, u^r)$) and potential temperature density $\rho\theta$ (with potential temperature $\theta$). Although this paper focuses on the shallow-atmosphere approximation to the equations of motion, MCore also implements the full equations of motion in a deep atmosphere. In the deep atmosphere, the equations of motion require a different treatment of the gravity and Coriolis terms. The formulations of these terms for both the deep-and shallow-atmosphere equations sets are described in the appendices. The differential form of the equations of motion under the shallow-atmosphere approximation can be written as follows:

$$\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho \mathbf{u}) = 0, \tag{4}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \mathrm{div}_j(\rho u^i u^j + G^{ij} p) = -\rho g \mathbf{g}_r - f \mathbf{g}_r \times (\rho \mathbf{u}), \tag{5}$$

$$\frac{\partial \rho\theta}{\partial t} + \mathrm{div}(\rho\theta \mathbf{u}) = 0. \tag{6}$$

Here the indices $i$ and $j$ span $\{\alpha, \beta, r\}$, $G^{ij}$ denotes the contravariant metric, $\mathbf{g}_r$ is the basis vector in the radial direction, $g$ is gravity and $f$ is the Coriolis parameter. The divergence operator is denoted by $\mathrm{div}(\cdot)$ (see also Appendix A). The pressure $p$ in the momentum equation is related to the potential temperature density via the equation of state

$$p = p_0 \left( \frac{R_d(\rho\theta)}{p_0} \right)^{c_p/c_v}, \tag{7}$$

where $p_0 = 1000$ hPa is the reference pressure, $R_d$ is the ideal gas constant for dry air and $c_p$ and $c_v$ denote the specific heat capacity of dry air at constant pressure and constant volume. The second terms on the left-hand-side of (4)–(6) are referred to as *flux terms*, since they determine the flow rate of the conservative state variables through the edges of a spatial region. The terms on the right-hand-side of these equations are *source terms*. Non-hydrostatic mesoscale models that use a closely related equation set include the Weather Research and Forecasting (WRF) model (see, for example, Skamarock and Klemp [40]) and the model by Ahmad and Lindeman [1]. Throughout this paper we will refer to the vector of prognostic quantities as the state vector and denote it by $\mathbf{q} = (\rho, \rho\mathbf{u}, \rho\theta)$.

A splitting is performed on the Eqs. (4)–(7) of the form

$$\rho(\mathbf{x}, t) = \rho^h(\mathbf{x}) + \rho'(\mathbf{x}, t), \tag{8}$$

$$p(\mathbf{x}, t) = p^h(\mathbf{x}) + p'(\mathbf{x}, t), \tag{9}$$

$$(\rho\theta)(\mathbf{x}, t) = (\rho\theta)^h(\mathbf{x}) + (\rho\theta)'(\mathbf{x}, t), \tag{10}$$

where $\mathbf{x} = (r, \alpha, \beta, n_p)$ and the superscript $h$ denotes fields which satisfy (7) and are in local hydrostatic balance, i.e.

$$\frac{\partial p^h}{\partial r} = -\rho^h g. \tag{11}$$

This choice is required to remove errors in approximating the hydrostatic state of the atmosphere that could be responsible for significant generation of spurious vertical momentum. The background state must be chosen from the space of hydrostatically balanced solutions, but can otherwise be any continuously differentiable function of space. For many idealized dynamical core test cases (e.g. see Section 5) the choice of the hydrostatic background is implicit in the test case formulation. When modeling real atmospheric motions, however, the background state can be chosen to be some hydrostatically balanced mean state of the atmosphere, potentially obtained by numerically projecting the non-hydrostatic initial data into the subspace of hydrostatic solutions. The hydrostatic base state is allowed to vary in the horizontal direction.

Using the curvilinear operators defined by cubed-sphere coordinates (see Appendix A) along with the shallow-atmosphere approximation (see Appendix B) the non-hydrostatic Eqs. (4)–(6) can be rewritten as

$$\frac{\partial \rho}{\partial t} + \frac{1}{J}\frac{\partial}{\partial x^k}(J\rho u^k) = 0, \tag{12}$$

$$\frac{\partial \rho u^\alpha}{\partial t} + \frac{1}{J}\frac{\partial}{\partial x^k}(J(\rho u^\alpha u^k + G^{\alpha k}p')) = \psi^\alpha_H + \psi^\alpha_M + \psi^\alpha_C, \tag{13}$$

$$\frac{\partial \rho u^\beta}{\partial t} + \frac{1}{J}\frac{\partial}{\partial x^k}(J(\rho u^\beta u^k + G^{\beta k}p')) = \psi^\beta_H + \psi^\beta_M + \psi^\beta_C, \tag{14}$$

$$\frac{\partial \rho u^r}{\partial t} + \frac{1}{J}\frac{\partial}{\partial x^k}(J(\rho u^r u^k + G^{rk}p')) = \psi^r_G, \tag{15}$$

$$\frac{\partial \rho\theta}{\partial t} + \frac{1}{J}\frac{\partial}{\partial x^k}(J\rho\theta u^k) = 0. \tag{16}$$

Here $J$ is the metric Jacobian, which is defined by (B.1). The source terms due to the horizontal variation of the hydrostatic background pressure are denoted by $\boldsymbol{\psi}_H = (\psi^\alpha_H, \psi^\beta_H)$ and given by applying the curvilinear gradient operator (A.11) to the hydrostatic pressure $p^h$,

$$\begin{pmatrix} \psi^\alpha_H \\ \psi^\beta_H \end{pmatrix} = -\frac{\delta^2}{(1+X^2)(1+Y^2)} \begin{pmatrix} (1+Y^2)\frac{\partial p^h}{\partial \alpha} + XY\frac{\partial p^h}{\partial \beta} \\ XY\frac{\partial p^h}{\partial \alpha} + (1+X^2)\frac{\partial p^h}{\partial \beta} \end{pmatrix}. \tag{17}$$

The source terms due to the cubed-sphere geometry are denoted by $\boldsymbol{\psi}_M = (\psi^\alpha_M, \psi^\beta_M)$, which under the shallow-atmosphere approximation take the form

$$\begin{pmatrix} \psi^\alpha_M \\ \psi^\beta_M \end{pmatrix} = \frac{2}{\delta^2} \begin{pmatrix} -XY^2\rho u^\alpha u^\alpha + Y(1+Y^2)\rho u^\alpha u^\beta \\ X(1+X^2)\rho u^\alpha u^\beta - X^2Y\rho u^\beta u^\beta \end{pmatrix}. \tag{18}$$

The third source term $\boldsymbol{\psi}_C = (\psi^\alpha_C, \psi^\beta_C)$ describes forcing due to the Coriolis effect. On equatorial panels under the shallow-atmosphere approximation it is given by

$$\begin{pmatrix} \psi^\alpha_C \\ \psi^\beta_C \end{pmatrix} = \frac{2\omega}{\delta^2} \begin{pmatrix} -XY^2 & Y(1+Y^2) \\ -Y(1+X^2) & XY^2 \end{pmatrix} \begin{pmatrix} \rho u^\alpha \\ \rho u^\beta \end{pmatrix}, \tag{19}$$

with the rotational speed of the Earth denoted by $\omega$. On polar panels it is given by

$$\begin{pmatrix} \psi^\alpha_C \\ \psi^\beta_C \end{pmatrix} = \frac{2s\omega}{\delta^2} \begin{pmatrix} -XY & (1+Y^2) \\ -(1+X^2) & XY \end{pmatrix} \begin{pmatrix} \rho u^\alpha \\ \rho u^\beta \end{pmatrix}, \tag{20}$$

where $s$ is a panel indicator defined by

$$s = \begin{cases} 1 & \text{on the northern panel} \quad (n_p = 5), \\ -1 & \text{on the southern panel} \quad (n_p = 6). \end{cases} \tag{21}$$

Finally, the gravitational source in the vertical momentum equation is denoted by $\psi^r_G$, and defined by

$$\psi^r_G = -\rho' g. \tag{22}$$

When topography is present, the terrain-following coordinates of Gal-Chen and Somerville [10] (hereafter referred to as GS coordinates) are used to deform the computational domain to match the physical space. The non-hydrostatic governing Eqs. (12)–(16) are not modified by this procedure, but instead we make use of orthonormalization and deorthonormalization operators to accurately compute fluxes in the presence of topography (see Section 4.5). In some sense, this approach is analogous to the treatment of edge fluxes that arise on unstructured grids. As we shall see, the spatial discretization of the horizontal will then be equivalent to the case of no topography, except with modified face areas and element volumes.

## 4. Numerical method

In this section we present the numerical methodology used to solve the non-hydrostatic equations of motion in a discrete context. MCore uses the method-of-lines to split the spatial and temporal components of the equations. Further, it splits the horizontal and vertical component of the fluid motion, solving for the former using a temporally explicit approach and the later using an implicit scheme. Coupling of these terms is managed via a Strang-carryover strategy, which ensures second-order-accuracy in time. In general, MCore is spatially fourth-order-accurate for horizontal motions and second-order-accurate for vertical motions.

### 4.1. Finite-volume discretization

In this section we present the discretization of the non-hydrostatic equations using a high-order finite-volume scheme. Under the finite-volume approach, we first integrate the Euler equations in the form (12)–(16) over an element $\mathcal{Z}$ (with volume $|\mathcal{Z}|$) and make use of Gauss' divergence theorem to write the flux term as an integral around the boundary $\partial\mathcal{Z}$, giving

$$\frac{\partial}{\partial t}\bar{\bar{\mathbf{q}}} + \frac{1}{|\mathcal{Z}|}\int\int_{\partial\mathcal{Z}} \boldsymbol{\mathcal{F}}\cdot\mathbf{n}dS = \bar{\bar{\boldsymbol{\psi}}}_H + \bar{\bar{\boldsymbol{\psi}}}_M + \bar{\bar{\boldsymbol{\psi}}}_C + \bar{\bar{\boldsymbol{\psi}}}_G. \tag{23}$$

Here the flux integral is taken over the surface with normal vector $\mathbf{n}$ and infinitesimal area element $dS$. The term $\boldsymbol{\mathcal{F}}\cdot\mathbf{n}$ is a vector quantity that denotes the outward flux of each of the state variables perpendicular to the boundary. The double overline denotes a 3D average of the form

$$\bar{\bar{\phi}} = \frac{1}{|\mathcal{Z}|}\int_{\mathcal{Z}} \phi dV, \tag{24}$$

where $dV = J\, d\alpha d\beta dr$ denotes the infinitesimal volume element. Here $\bar{\bar{\mathbf{q}}}$ denotes the averaged state vector in cell $\mathcal{Z}$. Likewise, $\bar{\bar{\boldsymbol{\psi}}}_H, \bar{\bar{\boldsymbol{\psi}}}_M,\ \bar{\bar{\boldsymbol{\psi}}}_C$ and $\bar{\bar{\boldsymbol{\psi}}}_G$ denote the element-averaged source terms due to the horizontal hydrostatic pressure gradient, the cubed-sphere geometry, Coriolis force and gravity, respectively. Note that the volume-averaged formulation (23) is exact, and so it remains to define an appropriate discretization over each of the terms in this expression.

A non-uniform grid spacing in the vertical is often needed to better resolve effects in the planetary boundary layer or near the model top. The non-uniform vertical coordinate is specified via an invertible conformal mapping that takes a point from the uniformly spaced vertical coordinate $\xi \in [0,1]$ (with grid spacing $\Delta\xi$) to the physical domain $r \in [r_s(\alpha,\beta), r_T]$ via

$$\Phi : (\xi,\alpha,\beta,n_p) \rightarrow (\mathcal{R}(\xi;\alpha,\beta,n_p),\alpha,\beta,n_p), \tag{25}$$

where $r_s(\alpha,\beta)$ is the surface elevation and $r_T$ is the radius of the model cap. Note that $\mathcal{R}(\xi) : [0,1] \rightarrow [r_s(\alpha,\beta,n_p), r_T]$ must be bijective and monotonically increasing in $\xi$. We depict such a coordinate transform in Fig. 2. In Section 5 we give one possible choice of $\mathcal{R}$, which will be used in our simulations to refine the grid spacing near the planetary boundary layer.

In the horizontal, element centroids are denoted by integer sub-indices, and over a single cubed-sphere panel are given by

$$\alpha_i = \left(i-\frac{1}{2}\right)\Delta\alpha - \frac{\pi}{4}, \quad \beta_j = \left(j-\frac{1}{2}\right)\Delta\alpha - \frac{\pi}{4}, \quad \xi_k = \left(k-\frac{1}{2}\right)\Delta\xi. \tag{26}$$

The horizontal indices span the range $(i,j) \in [1,N_c]^2$ and the vertical index ranges from $k \in [1,N_r]$, where $N_r$ denotes the number of vertical levels. The radial coordinate of element centroids is defined by mapping $\xi_k$ to physical space,

$$r_k = \mathcal{R}(\xi_k;\alpha_i,\beta_i,n_p). \tag{27}$$

Note that $k = 1$ corresponds to the lowermost model level whereas $k = N_r$ corresponds to the uppermost model level, which is the opposite of most pressure-based hydrostatic model formulations. Faces are midway between neighboring element centroids, and so are defined by half-indices and denoted by the symbol $\partial\mathcal{Z}$. For example, the edge $\partial\mathcal{Z}_{i+1/2,j,k}$ is at the interface between element $(i,j,k)$ and $(i + 1,j,k)$ and defines a plane that is constant in both $\alpha$ and $\beta$. Radial faces, denoted by $\partial\mathcal{Z}_{i,j,k+1/2}$ are constant in $\xi$, but are not necessarily constant in $r$ due to the horizontal dependence of the mapping $\mathcal{R}$. In fact, they are defined by the two-dimensional surface $r - \mathcal{R}(\xi;\alpha,\beta,n_p) = 0$ for $\xi$ fixed.

In the finite-volume discretization we must make use of both element volumes, denoted by $|\mathcal{Z}|$, and face areas, denoted by $|\partial\mathcal{Z}|$. In terms of the metric Jacobian (B.1), the element volume takes the form

$$|\mathcal{Z}|_{i,j,k} = \int_{\mathcal{Z}} dV = \int_{\alpha_{i-1/2}}^{\alpha_{i+1/2}}\int_{\beta_{j-1/2}}^{\beta_{j+1/2}}\int_{r_{k-1/2}}^{r_{k+1/2}} J dr d\beta d\alpha. \tag{28}$$

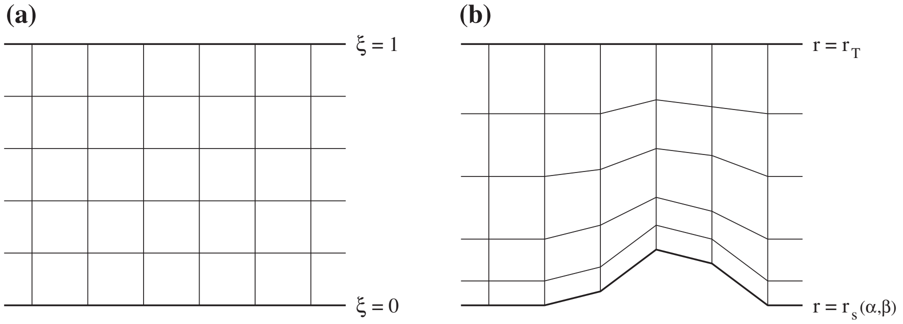


**Fig. 2.** (a) A 2D cross-section of a section of the computational grid, showing uniform spacing of the computational space. (b) The corresponding physical grid with radial coordinate $r \in [r_s(\alpha,\beta), r_T]$. The radial coordinate is related to the computational coordinate via the mapping $r = \mathcal{R}(\xi;\alpha,\beta,n_p)$.

The area of faces is likewise calculated by integrating the infinitesimal face areas (A.6)–(A.8). In practice these integrals are pre-computed using high-order Gaussian quadrature and stored for later use.

Quantities which are defined as edge averages will be denoted by a single overline. Average fluxes are defined at element faces and denoted here by $\overline{\mathbf{F}}_{i+1/2,j,k}$ for a flux across face $(i+1/2,j,k)$. They are defined by

$$\overline{\mathbf{F}}_{i+1/2,j,k} = \frac{1}{|\partial\mathcal{Z}|_{i+1/2,j,k}} \int\int_{\partial\mathcal{Z}_{i+1/2,j,k}} \mathcal{F}\cdot\mathbf{n}dS, \tag{29}$$

where $|\partial\mathcal{Z}|_{i+1/2,j,k}$ denotes the area of the face. Hence, the total flux across an interface, here denoted by $\mathbf{F}_{i+1/2,j,k}$, is simply the product of the average flux and the face area

$$\mathbf{F}_{i+1/2,j,k} = \overline{\mathbf{F}}_{i+1/2,j,k}|\partial\mathcal{Z}|_{i+1/2,j,k}. \tag{30}$$

Using (30) the volume averaged formulation (23) can be rewritten as

$$\frac{\partial}{\partial t}\bar{\bar{\mathbf{q}}}_{i,j,k} = \mathbf{H}(\mathbf{q}) + \mathbf{V}(\mathbf{q}), \tag{31}$$

where

$$\mathbf{H}(\mathbf{q}) = \frac{1}{|\mathcal{Z}|_{i,j,k}}[\mathbf{F}_{i-1/2,j,k} - \mathbf{F}_{i+1/2,j,k} + \mathbf{F}_{i,j-1/2,k} - \mathbf{F}_{i,j+1/2,k}] + \bar{\bar{\boldsymbol{\psi}}}_H + \bar{\bar{\boldsymbol{\psi}}}_M + \bar{\bar{\boldsymbol{\psi}}}_C, \tag{32}$$

and

$$\mathbf{V}(\mathbf{q}) = \frac{1}{|\mathcal{Z}|_{i,j,k}}\left[\mathbf{F}_{i,j,k-1/2} - \mathbf{F}_{i,j,k+1/2}\right] + \bar{\bar{\boldsymbol{\psi}}}_G. \tag{33}$$

Here we utilize a splitting based on separating horizontal motions, denoted by $\mathbf{H}(\mathbf{q})$, and vertical motions, denoted by $\mathbf{V}(\mathbf{q})$. Numerically this splitting is desirable since the relatively small vertical spacing of elements (on the scale of tens to hundreds of meters) imposes a severe time step restriction via the CFL condition. Hence, if we make use of an explicit treatment of the horizontal motions and treat the vertical motions implicitly, the model time step will not be affected by the grid spacing in the vertical.

### 4.2. Horizontal reconstruction

Under the finite-volume formulation only element-averaged information is known within each element. Hence, any non-constant sub-grid-scale reconstruction must use information from neighboring elements. MCore uses a novel reconstruction strategy which attains fourth-order accuracy using a minimal set of neighboring elements. In general, attaining greater than second-order accuracy is difficult since an $O(\Delta\alpha^n)$ reconstruction requires that the $p^{\text{th}}$ derivatives be approximated to $O(\Delta\alpha^{n-p})$. Standard finite-difference reconstruction formulae only guarantee $O(\Delta\alpha^2)$ accuracy, since such an approach approximates cell-centerpoint values using cell-averaged values. In a non-Cartesian domain, reconstructions must also incorporate information about the underlying geometry to recover high-order accuracy. For simplicity, we choose to drop the vertical index $k$ in this section since the horizontal reconstruction process is applied on each vertical level separately.

The strategy we have introduced relies on the convolution and deconvolution procedure of Barad and Colella [3]. This approach provides a mechanism for transforming cell-averages into cell-centered point values (denoted by the subscript (0)) over a sufficiently smooth data field $\phi$, according to

$$\phi_{(0)} = \bar{\bar{\phi}} - \frac{\Delta\alpha^4}{12|\mathcal{Z}|}\left(\frac{\partial\phi}{\partial\alpha}\frac{\partial\widetilde{J}}{\partial\alpha} + \frac{\partial\phi}{\partial\beta}\frac{\partial\widetilde{J}}{\partial\beta}\right) - \frac{\Delta\alpha^2}{24}\left(\frac{\partial^2\phi}{\partial\alpha^2} + \frac{\partial^2\phi}{\partial\beta^2}\right), \tag{34}$$

where $\widetilde{J}$ is the radially-integrated Jacobian,

$$\widetilde{J}(\alpha,\beta,n_p) = \int_{\mathcal{R}(\xi_{k-1/2};\alpha,\beta,n_p)}^{\mathcal{R}(\xi_{k+1/2};\alpha,\beta,n_p)} J(r,\alpha,\beta,n_p)dr. \tag{35}$$

If $\bar{\bar{\phi}}_{i,j}$ is known to at least fourth-order accuracy and the remaining derivative terms are known to at least $O(\Delta\alpha^2)$, this formula leads to a fourth-order-accurate approximation of $\phi_{(0)i,j}$, the cell-centered value of $\phi$ in element $(i,j)$. In this formulation, we directly approximate the derivatives of the radially-integrated Jacobian by

$$\frac{\partial\widetilde{J}}{\partial\alpha} \approx \frac{|\mathcal{Z}|_{i+1,j} - |\mathcal{Z}|_{i-1,j}}{2\Delta\alpha}, \tag{36}$$

$$\frac{\partial\widetilde{J}}{\partial\beta} \approx \frac{|\mathcal{Z}|_{i,j+1} - |\mathcal{Z}|_{i,j-1}}{2\Delta\alpha}. \tag{37}$$

For reasons of efficiency, these derivative are pre-computed prior to the simulation and stored for later use.

The reconstruction used by MCore proceeds as follows. The stencil we will use in the reconstruction step is depicted in Fig. 3. Second and third derivatives are first calculated using standard finite-difference formulae, which leads to approximations which are $O(\Delta\alpha^2)$ accurate. Here we use $q$ to denote one element of the state vector $\mathbf{q}$.

$$D_{\alpha\alpha\alpha}q_{i,j} = \frac{\bar{\bar{q}}_{i+2,j} - 2\bar{\bar{q}}_{i+1,j} + 2\bar{\bar{q}}_{i-1,j} - \bar{\bar{q}}_{i-2,j}}{12\Delta\alpha^3}, \tag{38}$$

$$D_{\alpha\alpha\beta}q_{i,j} = \frac{\bar{\bar{q}}_{i+1,j+1} - \bar{\bar{q}}_{i+1,j-1} - 2\bar{\bar{q}}_{i,j+1} + 2\bar{\bar{q}}_{i,j-1} + \bar{\bar{q}}_{i-1,j+1} - \bar{\bar{q}}_{i-1,j-1}}{4\Delta\alpha^3}, \tag{39}$$

$$D_{\alpha\beta\beta}q_{i,j} = \frac{\bar{\bar{q}}_{i+1,j+1} - 2\bar{\bar{q}}_{i+1,j} + \bar{\bar{q}}_{i+1,j-1} - \bar{\bar{q}}_{i-1,j+1} + 2\bar{\bar{q}}_{i-1,j} - \bar{\bar{q}}_{i-1,j-1}}{4\Delta\alpha^3}, \tag{40}$$

$$D_{\beta\beta\beta}q_{i,j} = \frac{\bar{\bar{q}}_{i,j+2} - 2\bar{\bar{q}}_{i,j+1} + 2\bar{\bar{q}}_{i,j-1} - \bar{\bar{q}}_{i,j-2}}{12\Delta\alpha^3}. \tag{41}$$

$$D_{\alpha\alpha}q_{i,j} = \frac{-\bar{\bar{q}}_{i+2,j} + 16\bar{\bar{q}}_{i+1,j} - 30\bar{\bar{q}}_{i,j} + 16\bar{\bar{q}}_{i-1,j} - \bar{\bar{q}}_{i-2,j}}{24\Delta\alpha^2}, \tag{42}$$

$$D_{\alpha\beta}q_{i,j} = \frac{\bar{\bar{q}}_{i+1,j+1} - \bar{\bar{q}}_{i-1,j+1} - \bar{\bar{q}}_{i+1,j-1} + \bar{\bar{q}}_{i-1,j-1}}{4\Delta\alpha^2}, \tag{43}$$

$$D_{\beta\beta}q_{i,j} = \frac{-\bar{\bar{q}}_{i,j+2} + 16\bar{\bar{q}}_{i,j+1} - 30\bar{\bar{q}}_{i,j} + 16\bar{\bar{q}}_{i,j-1} - \bar{\bar{q}}_{i,j-2}}{24\Delta\alpha^2}. \tag{44}$$

To attain fourth-order accuracy in space, we require that our approximation to the first derivative terms be at least $O(\Delta\alpha^3)$. To construct such a formula, we utilize a finite-difference relation over element point-values which have been reconstructed to $O(\Delta\alpha^4)$. First, define $O(\Delta\alpha^2)$ approximations to the first derivatives by

$$D^*_{\alpha}q_{i,j} = \frac{-\bar{\bar{q}}_{i+2,j} + 8\bar{\bar{q}}_{i+1,j} - 8\bar{\bar{q}}_{i-1,j} + \bar{\bar{q}}_{i-2,j}}{12\Delta\alpha}, \tag{45}$$

$$D^*_{\beta}q_{i,j} = \frac{-\bar{\bar{q}}_{i,j+2} + 8\bar{\bar{q}}_{i,j+1} - 8\bar{\bar{q}}_{i,j-1} + \bar{\bar{q}}_{i,j-2}}{12\Delta\alpha}. \tag{46}$$

High-order element averages of the state vector which do not incorporate geometric terms can then be computed via

$$q^*_{i+m,j} = \bar{\bar{q}}_{i+m,j} - \frac{\Delta\alpha^4}{12|\mathcal{Z}|_{i+m,j}}\left[\left(D^*_{\alpha}q_{i,j} + m\Delta\alpha D_{\alpha\alpha}q_{i,j}\right)\left(\frac{\partial\widetilde{J}}{\partial\alpha}\right)_{i+m,j} + \left(D^*_{\beta}q_{i,j} + m\Delta\alpha D_{\alpha\beta}q_{i,j}\right)\left(\frac{\partial\widetilde{J}}{\partial\beta}\right)_{i+m,j}\right], \tag{47}$$

and

$$q^*_{i,j+m} = \bar{\bar{q}}_{i,j+m} - \frac{\Delta\alpha^4}{12|\mathcal{Z}|_{i,j+m}}\left[\left(D^*_{\alpha}q_{i,j} + m\Delta\alpha D_{\alpha\beta}q_{i,j}\right)\left(\frac{\partial\widetilde{J}}{\partial\alpha}\right)_{i,j+m} + \left(D^*_{\beta}q_{i,j} + m\Delta\alpha D_{\beta\beta}q_{i,j}\right)\left(\frac{\partial\widetilde{J}}{\partial\beta}\right)_{i,j+m}\right]. \tag{48}$$

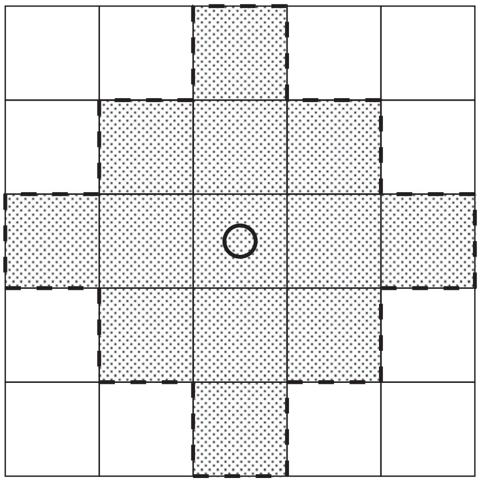


**Fig. 3.** A depiction of the stencil used for computing the fourth-order sub-grid-scale reconstruction on the cubed-sphere.

A third-order approximation to the first derivative terms is then given by

$$D_\alpha q_{i,j} = \frac{-q^*_{i+2,j} + 8q^*_{i+1,j} - 8q^*_{i-1,j} + q^*_{i-2,j}}{12\Delta\alpha} - \frac{\Delta\alpha^2}{24}\left(D_{\alpha\alpha\alpha}q_{i,j} + D_{\alpha\beta\beta}q_{i,j}\right), \tag{49}$$

$$D_\beta q_{i,j} = \frac{-q^*_{i,j+2} + 8q^*_{i,j+1} - 8q^*_{i,j-1} + q^*_{i,j-2}}{12\Delta\alpha} - \frac{\Delta\alpha^2}{24}\left(D_{\alpha\alpha\beta}q_{i,j} + D_{\beta\beta\beta}q_{i,j}\right) \tag{50}$$

A fourth-order approximation to the element centerpoint value is similarly obtained from the relation

$$q_{(0)i,j} = \bar{\bar{q}}_{i,j} - \frac{\Delta\alpha^2}{12|\mathcal{Z}|_{i,j}}\left[D_\alpha q_{i,j}\left(\frac{\partial\tilde{J}}{\partial\alpha}\right)_{i,j} + D_\beta q_{i,j}\left(\frac{\partial\tilde{J}}{\partial\beta}\right)_{i,j}\right] - \frac{\Delta\alpha^2}{24}\left[D_{\alpha\alpha}q_{i,j} + D_{\beta\beta}q_{i,j}\right]. \tag{51}$$

We make no attempt at introducing a limiter for enforcing monotonicity or positivity of thermodynamic quantities. Limiting is generally of greater importance for tracer transport problems where monotonicity and positivity must be guaranteed, and so it is not pursued here.

With the approximated centerpoint value and corresponding derivatives in hand, we can now write expressions for the reconstructed state vector at face centerpoints. These face centerpoint values are denoted with either a $L$ or $R$, depending on whether they are left or right of the interface:

$$q_{L,i+1/2,j} = q_{(0)i,j} + D_\alpha q_{i,j}\left(\frac{\Delta\alpha}{2}\right) + D_{\alpha\alpha}q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^2 + D_{\alpha\alpha\alpha}q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^3, \tag{52}$$

$$q_{R,i-1/2,j} = q_{(0)i,j} - D_\alpha q_{i,j}\left(\frac{\Delta\alpha}{2}\right) + D_{\alpha\alpha}q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^2 - D_{\alpha\alpha\alpha}q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^3, \tag{53}$$

$$q_{L,i,j+1/2} = q_{(0)i,j} + D_\beta q_{i,j}\left(\frac{\Delta\alpha}{2}\right) + D_{\beta\beta}q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^2 + D_{\beta\beta\beta}q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^3, \tag{54}$$

$$q_{R,i,j-1/2} = q_{(0)i,j} - D_\beta q_{i,j}\left(\frac{\Delta\alpha}{2}\right) + D_{\beta\beta}q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^2 - D_{\beta\beta\beta}q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^3. \tag{55}$$

The centerpoint values, as calculated from the above formulae are then used for computing edge fluxes across the interface via a Riemann solver. This calculation will be described in more detail later.

### *4.3. Vertical reconstruction*

In the vertical we can no longer rely on uniformity of the grid spacing, so the reconstruction must take into account the variance in the height of grid elements. Nonetheless, we can still construct a second-order sub-grid-scale reconstruction within each element by fitting a parabola through the centroid of each grid element and its immediate neighbors. As a result, we obtain the following formulae for the reconstructed field at each edge:

$$\mathbf{q}_{L,k+1/2} = -\frac{\Delta r_p^2}{4\Delta r_n(\Delta r_n + \Delta r_p)}\mathbf{q}_{k-1} + \frac{(2\Delta r_n + \Delta r_p)}{4\Delta r_n}\mathbf{q}_k + \frac{(2\Delta r_n + \Delta r_p)}{4(\Delta r_n + \Delta r_p)}\mathbf{q}_{k+1}, \tag{56}$$

$$\mathbf{q}_{R,k-1/2} = \frac{(\Delta r_n + 2\Delta r_p)}{4(\Delta r_n + \Delta r_p)}\mathbf{q}_{k-1} + \frac{(\Delta r_n + 2\Delta r_p)}{4\Delta r_p}\mathbf{q}_k - \frac{\Delta r_n^2}{4\Delta r_p(\Delta r_p + \Delta r_n)}\mathbf{q}_{k+1}, \tag{57}$$

where

$$\Delta r_p = r_{k+1} - r_k, \quad \Delta r_n = r_k - r_{k-1}. \tag{58}$$

Unlike in the horizontal, the subscripts $L$ and $R$ actually correspond to the elements immediately below and immediately above each edge, respectively.

Near the top and bottom boundaries, the stencil width is reduced by one element and so the sub-grid-scale reconstruction becomes linear. At the bottom boundary the reconstruction then reads

$$\mathbf{q}_{R,1/2} = \frac{(2\Delta r_n + \Delta r_p)}{\Delta r_n + \Delta r_p}\mathbf{q}_1 - \frac{\Delta r_n}{\Delta r_n + \Delta r_p}\mathbf{q}_2, \tag{59}$$

$$\mathbf{q}_{L,3/2} = \frac{\Delta r_p}{\Delta r_p + \Delta r_n}\mathbf{q}_1 + \frac{\Delta r_n}{\Delta r_p + \Delta r_n}\mathbf{q}_2, \tag{60}$$

where

$$\Delta r_p = r_3 - r_2, \quad \Delta r_n = r_2 - r_1. \tag{61}$$

At the top boundary, the reconstruction reads

$$\mathbf{q}_{L,N_r+1/2} = -\frac{\Delta r_p}{\Delta r_n + \Delta r_p}\mathbf{q}_{N_r-1} + \frac{2\Delta r_p + \Delta r_n}{\Delta r_n + \Delta r_p}\mathbf{q}_{N_r}, \tag{62}$$

$$\mathbf{q}_{R,N_r-1/2} = \frac{\Delta r_p}{\Delta r_p + \Delta r_n}\mathbf{q}_{N_r-1} + \frac{\Delta r_n}{\Delta r_p + \Delta r_n}\mathbf{q}_{N_r}, \tag{63}$$

where

$$\Delta r_p = r_{N_r} - r_{N_r-1}, \quad \Delta r_n = r_{N_r-1} - r_{N_r-2}. \tag{64}$$

### 4.4. Horizontal-vertical splitting and time-stepping scheme

To guarantee sufficient accuracy in time, MCore uses the Strang-carryover approach presented in Ullrich and Jablonowski [45] to couple explicit integration in the horizontal with implicit integration in the vertical. In the horizontal, this scheme uses the fourth-order Runge–Kutta scheme by default, although options are available for the strong-stability preserving Runge–Kutta integrators of Gottlieb et al. [14]. Either choice of time stepping algorithm leads to a scheme with implicit-explicit coupling terms of overall second-order temporal accuracy.

At the initial step, we solve for $\mathbf{q}^{(1)}$ via an implicit step of duration $\Delta t/2$:

$$\frac{\mathbf{q}^{(1)} - \mathbf{q}^n}{(\Delta t/2)} - \mathbf{V}(\mathbf{q}^{(1)}) = 0. \tag{65}$$

MCore allows for compile-time switching between linearly implicit integration and fully implicit integration in the vertical. The implicit solve is performed using Newton–Krylov iteration with either a numerically or analytically computed Jacobian matrix. Since our choice of Riemann solver in the vertical is nearly linear, the current implementation of MCore makes use of an analytically computed matrix. Except when stated otherwise, we apply the linearly implicit Rosenbrock approach, which only makes use of the first iteration of the Newton–Krylov method. This approach is roughly twice as fast as the fully implicit solve with almost negligible difference in the flow field.

After the implicit solve, the state $\mathbf{q}^{(1)}$ is used as input for an explicit scheme. The fourth-order Runge–Kutta (RK4) scheme leads to the sequence

$$\mathbf{q}^{(2)} = \mathbf{q}^{(1)} + \frac{\Delta t}{2}\mathbf{H}(\mathbf{q}^{(1)}), \tag{66}$$

$$\mathbf{q}^{(3)} = \mathbf{q}^{(1)} + \frac{\Delta t}{2}\mathbf{H}(\mathbf{q}^{(2)}), \tag{67}$$

$$\mathbf{q}^{(4)} = \mathbf{q}^{(1)} + \Delta t\mathbf{H}(\mathbf{q}^{(3)}), \tag{68}$$

$$\mathbf{q}^{*} = -\frac{1}{3}\mathbf{q}^{(1)} + \frac{1}{3}\mathbf{q}^{(2)} + \frac{2}{3}\mathbf{q}^{(3)} + \frac{1}{3}\mathbf{q}^{(4)} + \frac{\Delta t}{6}\mathbf{H}(\mathbf{q}^{(4)}). \tag{69}$$

Finally, $\mathbf{q}^*$ becomes the input for a final implicit solve of size $\Delta t/2$:

$$\frac{\mathbf{q}^{n+1} - \mathbf{q}^*}{(\Delta t/2)} - \mathbf{V}(\mathbf{q}^{n+1}) = 0. \tag{70}$$

The forcing due to the implicit step is then stored

$$\mathcal{G} = \mathbf{q}^{n+1} - \mathbf{q}^*. \tag{71}$$

At the following time step, the implicit forcing from the previous time step is used to predict the next input for the explicit scheme,

$$\mathbf{q}^{(1)} = \mathbf{q}^n + \mathcal{G}. \tag{72}$$

Observe that after the initial step, the resulting scheme only uses one implicit solve per time step.

### 4.5. Orthonormalization

MCore makes use of approximate Riemann solvers for computing the flux $\mathbf{F}(\mathbf{q})$ across each face. However, generic Riemann solvers are purely one-dimensional operators. To apply this class of solvers to multidimensional problems, we must first transform vector quantities into an orthonormal frame. In 3D, the orthonormal frame consists of one basis vector which is orthogonal to the active edge (denoted $\mathbf{e}_\perp$) and two components which are parallel to the edge (denoted $\mathbf{e}_1$ and $\mathbf{e}_2$). Hence, the non-orthogonal components define a tangent plane to the active edge. At the point where the orthogonal basis is defined, an arbitrary vector can be written in either the natural basis,

$$\mathbf{v} = v^\alpha\mathbf{g}_\alpha + v^\beta\mathbf{g}_\beta + v^r\mathbf{g}_r, \tag{73}$$

or in the orthonormal basis

$$\mathbf{v} = v^{\perp}\mathbf{e}_{\perp} + v^{1}\mathbf{e}_{1} + v^{2}\mathbf{e}_{2}. \tag{74}$$

The transformation between these two systems is accomplished via the orthonormalization matrix $\mathcal{O}_d(\alpha,\beta,r)$,

$$\begin{pmatrix} v^{\perp} \\ v^{1} \\ v^{2} \end{pmatrix} = \mathcal{O}_d(\alpha,\beta,r)\begin{pmatrix} v^{\alpha} \\ v^{\beta} \\ v^{r} \end{pmatrix}. \tag{75}$$

Here $d$ denotes the coordinate being held constant, $d \in \{\alpha,\beta,\xi\}$. Analogously, transforming from the orthonormal basis to the natural basis simply requires applying the inverse operation,

$$\begin{pmatrix} v^{\alpha} \\ v^{\beta} \\ v^{r} \end{pmatrix} = \mathcal{O}_d^{-1}(\alpha,\beta,r)\begin{pmatrix} v^{\perp} \\ v^{1} \\ v^{2} \end{pmatrix}. \tag{76}$$

Since the natural basis for the cubed-sphere consists of a radial basis vector which is already normalized and orthogonal to the basis vectors in the horizontal, at edges of constant $\alpha$ and constant $\beta$, we simply set $\mathbf{e}_2 = \mathbf{g}_r$. The orthonormalization matrix at edges of constant $\alpha$ can then be written as a $2 \times 2$ matrix,

$$\mathcal{O}_{\alpha} = \begin{pmatrix} \frac{\sqrt{1+X^2}}{\delta} & 0 \\ \frac{-XY\sqrt{1+X^2}}{\delta^2} & \frac{(1+Y^2)\sqrt{1+X^2}}{\delta^2} \end{pmatrix}. \tag{77}$$

Similarly, at edges of constant $\beta$ we have

$$\mathcal{O}_{\beta} = \begin{pmatrix} 0 & \frac{\sqrt{1+Y^2}}{\delta} \\ \frac{(1+X^2)\sqrt{1+Y^2}}{\delta^2} & \frac{-XY\sqrt{1+Y^2}}{\delta^2} \end{pmatrix}. \tag{78}$$

In each case the matrices defined above only work on the horizontal components of the vector $\mathbf{v}$. The corresponding deorthonormalization matrices are

$$\mathcal{O}_{\alpha}^{-1} = \begin{pmatrix} \frac{\delta}{\sqrt{1+X^2}} & 0 \\ \frac{XY\delta}{(1+Y^2)\sqrt{1+X^2}} & \frac{\delta^2}{(1+Y^2)\sqrt{1+X^2}} \end{pmatrix}, \tag{79}$$

$$\mathcal{O}_{\beta}^{-1} = \begin{pmatrix} \frac{XY\delta}{(1+X^2)\sqrt{1+Y^2}} & \frac{\delta^2}{(1+X^2)\sqrt{1+Y^2}} \\ \frac{\delta}{\sqrt{1+Y^2}} & 0 \end{pmatrix}. \tag{80}$$

The problem of deriving an orthonormalization matrix in the vertical is more difficult since a cell's vertical bounding surface (a surface of constant $\xi$) is only a surface of constant $r$ in the absence of terrain. When terrain is present, we utilize the fact that $\nabla^i f$ defines a vector perpendicular to the surface $f(\alpha,\beta,r) = r - \mathcal{R}(\xi;\alpha,\beta,n_p)$ (with $\xi$ = const.). Hence, the vector

$$\mathbf{e}_{\perp}^{*} = \left[\left[G^{11}\left(-\frac{\partial\mathcal{R}}{\partial\alpha}\right) + G^{12}\left(-\frac{\partial\mathcal{R}}{\partial\beta}\right)\right]\mathbf{g}_{\alpha} + \left[G^{21}\left(-\frac{\partial\mathcal{R}}{\partial\alpha}\right) + G^{22}\left(-\frac{\partial\mathcal{R}}{\partial\beta}\right)\right]\mathbf{g}_{\beta} + \mathbf{g}_{r}\right], \tag{81}$$

is orthogonal to the given terrain-following radial face (but is not of unit length). A vector of unit length can be obtained by simply scaling $\mathbf{e}_{\perp}^{*}$ according to

$$\mathbf{e}_{\perp} = \frac{1}{|\mathbf{e}_{\perp}^{*}|}\mathbf{e}_{\perp}^{*}, \tag{82}$$

where the notation $|\mathbf{v}| = \sqrt{G_{ij}v^i v^j}$ denotes the magnitude of a 3D vector $\mathbf{v}$ in cubed-sphere coordinates. Hereafter we use $e_{\perp}^{\alpha}, e_{\perp}^{\beta}$ and $e_{\perp}^{r}$ to denote the components of $\mathbf{e}_{\perp}$ along the $\alpha$, $\beta$ and $r$ directions. To obtain vectors orthogonal to $\mathbf{e}_{\perp}$, we simply apply the Gram–Schmidt orthonormalization procedure to the $\alpha$ and $\beta$ components of the natural basis. For example, for the $\alpha$ component, we have

$$\mathbf{e}_{1}^{*} = \mathbf{g}_{\alpha} - \langle \mathbf{g}_{\alpha}, \mathbf{e}_{\perp}\rangle \mathbf{e}_{\perp}, \tag{83}$$

with curvilinear inner product $\langle \mathbf{v},\mathbf{w}\rangle = G_{ij}v^j w^j$ of the two vectors $\mathbf{v}$ and $\mathbf{w}$. Hence, on simplifying,

$$\mathbf{e}_{1}^{*} = (1 - Q^{\alpha}e_{\perp}^{\alpha})\mathbf{g}_{\alpha} - Q^{\alpha}e_{\perp}^{\beta}\mathbf{g}_{\beta} - Q^{\alpha}e_{\perp}^{r}\mathbf{g}_{r}, \tag{84}$$

where $Q^{\alpha} = G_{11}e_{\perp}^{\alpha} + G_{12}e_{\perp}^{\beta}$. Similarly,

$$\mathbf{e}_{2}^{*} = -Q^{\beta}e_{\perp}^{\alpha}\mathbf{g}_{\alpha} + (1 - Q^{\beta}e_{\perp}^{\beta})\mathbf{g}_{\beta} - Q^{\beta}e_{\perp}^{r}\mathbf{g}_{r}, \tag{85}$$

where $Q^\beta = G_{21}e_\perp^\alpha + G_{22}e_\perp^\beta$. As before, we normalize these vectors to obtain $\mathbf{e}_1$ and $\mathbf{e}_2$. The deorthonormalization matrix $\mathcal{O}_\xi^{-1}$ is then defined as the matrix whose columns are given by $\mathbf{e}_\perp$, $\mathbf{e}_1$ and $\mathbf{e}_2$. To obtain $\mathcal{O}_\xi$ at each edge, we invert the corresponding deorthonormalization matrix numerically.

In the formulation used by MCore, the orthonormalization and deorthonormalization matrices are only required at the centerpoint of each face. Hence, these matrices can be pre-computed, for reasons of efficiency, and stored for later use. After certain optimizations to the algorithmic implementation of the Riemann solver, it turns out that only three components of each of the matrices are required during runtime.

### 4.6. Riemann solvers

Under any upwind finite-volume formulation, reconstructed edge values are inherently discontinuous – that is, the reconstructed left edge value is almost never identically equal to the right edge value. The discrepancy between left and right edge values is a measure of the roughness of the underlying fields. Godunov-type finite-volume methods [13], such as the one in this paper, solve the Riemann problem at interfaces so as to obtain a single-valued flux at each edge. Since computing the exact solution of the Riemann problem is generally expensive, we instead rely on so-called approximate Riemann solvers. However, dozens of such approximate solvers are now available. Ullrich et al. [46] analyzed three such solvers, including the solver of Rusanov [36], the solver of Roe [33] and the recently introduced AUSM$^+$-up solver of Liou [23]. They concluded that for horizontal atmospheric motions the Rusanov solver was far too diffusive, whereas both Roe and AUSM$^+$-up yielded comparable results, with AUSM$^+$-up slightly outperforming Roe.

#### 4.6.1. The AUSM$^+$-up solver in the horizontal direction

Based on the results of Ullrich et al. [46] we have adopted the full AUSM$^+$-up solver for computing horizontal fluxes. The details of this Riemann solver are given in Liou [23], and so the algorithm is not repeated here. The only notable change in our implementation is the choice of $K_u = 0$, instead of Liou's suggested choice $K_u = 1/4$. This change has been observed to improve the stability of the algorithm and reduce numerical diffusion for atmospheric flows.

#### 4.6.2. The quasi-linear AUSM$^+$-up solver in the vertical direction

For very low-speed flows, such as vertical atmospheric motions, Liou's solver can be simplified further by linearizing the diffusive terms. This process leads to the quasi-linear AUSM$^+$-up solver, which was first derived in Ullrich and Jablonowski [45]. This solver was demonstrated to correctly model flows in both the 2D and 3D Cartesian domain on a variety of scales, and its quasi-linearity allows simple formulation of the analytic Jacobian when performing implicit integration. For completeness, we provide a short description of this solver.

The quasi-linear AUSM$^+$-up solver makes use of the hydrostatic background state to define a background sound speed $a_{1/2}^h = \sqrt{\gamma p^h/\rho^h}$ at each vertical interface (with $\gamma = c_p/c_v$). Given left and right state vectors $\mathbf{q}_L = (\rho_L, (\rho\mathbf{u})_L, (\rho\theta)_L)$ and $\mathbf{q}_R = (\rho_R, (\rho\mathbf{u})_R, (\rho\theta)_R)$, we define a modified velocity at the interface via

$$w_{1/2} = \frac{(\rho v^\perp)_L + (\rho v^\perp)_R}{\rho_L + \rho_R} - K_p a_{1/2}^h \frac{(\rho\theta)_R - (\rho\theta)_L}{(\rho\theta)^h}, \tag{86}$$

where $K_p$ is the dimensionless pressure diffusion coefficient and $(\rho\theta)^h$ is the potential temperature density of the hydrostatic background at the interface. Similarly, we define an interface pressure using (7) that is given by

$$p_{1/2} = p_0 \left( \frac{R_d}{p_0} \left( \frac{(\rho\theta)_L + (\rho\theta)_R}{2} \right) \right)^{c_p/c_v} - \frac{K_u}{2} c_{1/2} \left( (\rho v^\perp)_R - (\rho v^\perp)_L \right), \tag{87}$$

where $p^h$ is the hydrostatic pressure at the interface and $K_u$ is the dimensionless momentum diffusion coefficient. The pointwise flux across the interface is then defined by

$$\mathbf{F}_{(0)}^{AUSM} = \mathbf{p}'_{1/2} + \begin{cases} w_{1/2}\mathbf{q}_L & \text{if } w_{1/2} > 0, \\ w_{1/2}\mathbf{q}_R & \text{otherwise}, \end{cases} \tag{88}$$

where

$$\mathbf{p}'_{1/2} = \left( 0, p_{1/2} - p_{1/2}^h, 0, 0, 0 \right), \tag{89}$$

i.e. $\mathbf{p}_{1/2}$ only has one non-zero value, which is a contribution associated with the perpendicular component of the momentum. This solver uses the tuning parameters $K_u$ and $K_p$, which for vertical motions have been chosen as $K_u = 2$ and $K_p = 1/4$. In Ullrich and Jablonowski [45] we have observed that this larger value of $K_u$ is required to damp spurious vertical oscillations when the aspect ratio is large (as opposed to the recommended value of 3/4 given in [23]). Finally, since the vertical flux is only second-order accurate, we make the approximation

$$\overline{\mathbf{F}}_{i,j,k+1/2} = \mathbf{F}_{(0)i,j,k+1/2}, \tag{90}$$

i.e. we take the average flux across the interface to be equal to the pointwise flux evaluated from (88).

### 4.7. Fourth-order horizontal accuracy

To solve for the flux across horizontal faces we make use of the reconstructed pointwise values of the state variables on either side of the interface (52)–(55). Fluxes are evaluated pointwise at element face centerpoints, and are obtained by solving a Riemann problem using left and right state vectors. The resulting pointwise flux vector is a second-order approximation to the average flux across the interface. To obtain fourth-order accuracy in computing the flux across the interface, some form of high-order quadrature is needed. In this section we again drop the vertical index $k$, which is assumed constant.

MCore makes use of a convolution formula for computing average face fluxes $\overline{\mathbf{F}}$ from pointwise fluxes $\mathbf{F}_{(0)}$. On faces of constant $\alpha$ and $\beta$, this formula takes the form

$$\overline{\mathbf{F}}_{i+1/2,j} = \mathbf{F}_{(0)i+1/2,j} + \frac{\Delta\alpha^2}{24}\left(\frac{\partial^2\mathbf{F}}{\partial\beta^2}\right)_{i+1/2,j} + \frac{\Delta\alpha^2}{12|\partial\mathcal{Z}|_{i+1/2,j}}\left(\frac{\partial\mathbf{F}}{\partial\beta}\right)_{i+1/2,j}\left(\frac{\partial\widetilde{J}_\alpha}{\partial\beta}\right)_{i+1/2,j}, \tag{91}$$

$$\overline{\mathbf{F}}_{i,j+1/2} = \mathbf{F}_{(0)i,j+1/2} + \frac{\Delta\alpha^2}{24}\left(\frac{\partial^2\mathbf{F}}{\partial\alpha^2}\right)_{i,j+1/2} + \frac{\Delta\alpha^2}{12|\partial\mathcal{Z}|_{i,j+1/2}}\left(\frac{\partial\mathbf{F}}{\partial\alpha}\right)_{i,j+1/2}\left(\frac{\partial\widetilde{J}_\beta}{\partial\alpha}\right)_{i,j+1/2}. \tag{92}$$

Here $\widetilde{J}_\alpha$ and $\widetilde{J}_\beta$ are the radially-integrated metric co-factors (B.2) and (B.3). Under our formulation, their derivatives are approximated in terms of neighboring face areas as

$$\left(\frac{\partial\widetilde{J}_\alpha}{\partial\beta}\right)_{i+1/2,j} \approx \frac{|\partial\mathcal{Z}|_{i+1/2,j+1} - |\partial\mathcal{Z}|_{i+1/2,j-1}}{2\Delta\alpha}, \tag{93}$$

and

$$\left(\frac{\partial\widetilde{J}_\beta}{\partial\alpha}\right)_{i,j+1/2} \approx \frac{|\partial\mathcal{Z}|_{i+1,j+1/2} - |\partial\mathcal{Z}|_{i-1,j+1/2}}{2\Delta\alpha}. \tag{94}$$

To maintain fourth-order-accuracy, the pointwise fluxes in (91) and (92) must be evaluated to at least $O(\Delta\alpha^4)$ and flux derivative terms must be evaluated at $O(\Delta\alpha^2)$. Hence, it is sufficient to make use of the compact second-order derivative formulae

$$\left(\frac{\partial\mathbf{F}}{\partial\beta}\right)_{i+1/2,j} \approx \frac{\mathbf{F}_{(0)i+1/2,j+1} - \mathbf{F}_{(0)i+1/2,j-1}}{2\Delta\alpha}, \tag{95}$$

$$\left(\frac{\partial^2\mathbf{F}}{\partial\beta^2}\right)_{i+1/2,j} \approx \frac{\mathbf{F}_{(0)i+1/2,j+1} - 2\mathbf{F}_{(0)i+1/2,j} + \mathbf{F}_{(0)i+1/2,j-1}}{\Delta\alpha^2}, \tag{96}$$

$$\left(\frac{\partial\mathbf{F}}{\partial\alpha}\right)_{i,j+1/2} \approx \frac{\mathbf{F}_{(0)i+1,j+1/2} - \mathbf{F}_{(0)i-1,j+1/2}}{2\Delta\alpha}, \tag{97}$$

$$\left(\frac{\partial^2\mathbf{F}}{\partial\alpha^2}\right)_{i,j+1/2} \approx \frac{\mathbf{F}_{(0)i+1,j+1/2} - 2\mathbf{F}_{(0)i,j+1/2} + \mathbf{F}_{(0)i-1,j+1/2}}{\Delta\alpha^2}. \tag{98}$$

To achieve fourth-order accuracy overall, source terms of the horizontal momentum equations must be evaluated to at least third-order accuracy. As with edge fluxes, simply evaluating the source terms using cell-averaged values $\bar{\bar{\mathbf{q}}}_{i,j,k}$ only leads to a second-order discretization. Hence, MCore follows a convolution and deconvolution strategy that leads to a fourth-order-accurate approximation of the source terms. To begin, source terms are first evaluated at element centerpoints ($\boldsymbol{\psi}_{(0)i,j} = \boldsymbol{\psi}(\mathbf{q}_{(0)i,j})$). Notably, the state vector $\mathbf{q}_{(0)i,j}$ has already been calculated at this point during reconstruction using (51). Now, to obtain a fourth-order accurate approximation to the forcing term within this element we apply a convolution operator of the form

$$\bar{\bar{\boldsymbol{\psi}}}_{i,j} = \boldsymbol{\psi}_{(0)i,j} + \frac{\Delta\alpha^2}{24}\left[\left(\frac{\partial^2\boldsymbol{\psi}}{\partial\alpha^2}\right)_{i,j} + \left(\frac{\partial^2\boldsymbol{\psi}}{\partial\beta^2}\right)_{i,j}\right] + \frac{\Delta\alpha^4}{12|\mathcal{Z}|_{i,j}}\left[\left(\frac{\partial\boldsymbol{\psi}}{\partial\alpha}\right)_{i,j}\left(\frac{\partial\widetilde{J}}{\partial\alpha}\right)_{i,j} + \left(\frac{\partial\boldsymbol{\psi}}{\partial\beta}\right)_{i,j}\left(\frac{\partial\widetilde{J}}{\partial\beta}\right)_{i,j}\right]. \tag{99}$$

As with the flux operators, all first and second derivatives are approximated to second-order accuracy using standard centered finite-difference operators. The derivatives of the vertically-integrated metric Jacobian are approximated by (36) and (37).

### 4.8. Inclusion of topography

Topography does not explicitly enter the formulation of MCore, but instead enters implicitly in two ways. First, element volumes and face areas are modified by the presence of topography. This has a direct effect on the calculation of total edge fluxes. Notably, under the shallow-atmosphere approximation the areas of vertical bounding faces are actually unmodified by the presence of topography. The second method topography enters the formulation is via pressure terms that are accounted for by the deorthonormalization matrix. Namely, when fluxes are computed across vertical surfaces the deorthonormalization procedure is responsible for applying pressure forcing appropriately to all components of the momentum. This has the effect of applying pressure forcing to the horizontal momentum components whenever the vertical bounding face is tilted.

### *4.9. Treatment of panel boundaries*

One must be careful in the treatment of panel boundaries on the cubed-sphere grid. First, the underlying coordinate system is disjoint at panel boundaries so one must be certain that vector quantities are in the correct reference frame. Second, since the grid is not smooth in the vicinity of the panel boundary a direct application of the reconstruction stencil near grid edges can potentially generate significant numerical noise and reduce the accuracy of the simulation. As a consequence, we advocate the use of halo elements around panels (see Fig. 1) in combination with an appropriate remapping algorithm.

The remapping process we apply in MCore is identical to the one described in Ullrich et al. [46]. Under this approach, halo elements are extended outward from each panel. Sub-grid-scale fourth-order reconstructions are built along each panel edge using one-sided reconstruction stencils and the reconstructions are sampled at Gaussian quadrature points. As a result, we obtain fourth-order-accurate approximations to element averages in halo elements. The usual horizontal reconstruction stencil can then be applied, minimizing the generation of spurious boundary noise.

### *4.10. Rayleigh friction*

It is well-known that atmospheric models that have a rigid-lid upper boundary condition typically suffer significant wave reflection at the model top. The traditional approach to deal with this problem has been to add a sponge layer that is responsible for damping out oscillations in the velocity fields high in the atmosphere. Wave reflection is particularly problematic near steep topography or in the presence of strong vertical motions.

MCore implements an optional Rayleigh friction layer. Since Rayleigh friction is a potentially stiff effect, it is included in the implicit stage of the solver as a source term of the full 3D momentum equation that takes the form

$$\boldsymbol{\psi}_R = -R_c(\alpha,\beta,\xi)(\rho\mathbf{u} - \rho\mathbf{u}_0), \tag{100}$$

where $R_c$ denotes the strength of the friction term, $\mathbf{u}$ is the 3D velocity vector and $\mathbf{u}_0$ denotes some appropriate reference state for the velocity. The strength of the friction term can be chosen arbitrarily, but should transition smoothly from zero forcing at lower levels to some maximum at the model top. For simplicity, we choose

$$R_c(\alpha,\beta,\xi) = \begin{cases} 0 & \text{if } \xi < \xi_R, \\ \frac{1}{\tau_R}\left(\frac{\xi-\xi_R}{1-\xi_R}\right)^2 & \text{otherwise}. \end{cases} \tag{101}$$

Here $\tau_R$ is the timescale of the damping and $\xi_R$ is the height of the damping layer in $\xi$ coordinates. By default, we define $\tau_R = 1$ day and $\xi_R = 0.7$, which roughly places the Rayleigh damping layer at an altitude of 20 km for model tops around 30 km.

### *4.11. Design features*

The high-order finite-volume methods of MCore are designed to allow run-time switching between 2D shallow-water, non-hydrostatic 3D shallow-atmosphere and 3D deep-atmosphere configurations. Further, compile-time switching is available for the choice of time-discretization, the form of the non-hydrostatic evolution equations, the choice of Riemann solver and the form of the horizontal reconstruction. The model attempts to maximize efficiency by pre-computing geometric quantities and storing the corresponding values for the duration of the simulation. To enhance modularity and readability the model is implemented in C++ using classes to isolate functionality.

MCore has been parallelized and currently utilizes the message passing interface (MPI) approach for parallel communication. Future implementations will likely also harness OpenMP for shared memory computing, which may allow some degree of parallelization in the vertical direction. Since the method is discretized on the quasi-uniform cubed-sphere grid, inter-processor communication is only required among local processors which should work to enhance parallel scalability. However, in its current form processors which administrate elements adjacent to panel edges must also manage the coordinate transform at panel boundaries. Consequently, we do not expect perfect scalability under the current approach, but additional efforts to better balance processor load may be helpful in alleviating this problem. Work is ongoing to test the parallel scalability of this method on large-scale supercomputing clusters, although tests on our local cluster are promising.

MCore makes use of the GECoRe remapping package of Ullrich et al. [47] for performing conservative remapping between cubed-sphere and latitude-longitude grids. The data output format can be chosen to be MATLAB-readable.mat files or Network Common Data Form (NetCDF).nc data.

## 5. Numerical results

Several test cases have been chosen to demonstrate the robustness and accuracy of MCore, including a baroclinic instabilty, 3D Rossby–Haurwitz wave, mountain-induced Rossby wave train, gravity wave test and Held–Suarez climatology. Rayleigh friction, as described in Section 4.10, is only used for the mountain-induced Rossby wave train test case. Most test runs make use of a $r_T$ = 30 km model top unless noted otherwise (such as in Section 5.4) and a vertical grid spacing which we have chosen to be

$$\mathcal{R}(\xi;\alpha,\beta,n_p) = r_s(\alpha,\beta,n_p) + (r_T - r_s(\alpha,\beta,n_p))\tilde{\phi}(\xi), \tag{102}$$

where

$$\tilde{\phi}(\xi) = \frac{1}{\sqrt{b+1}-1}\left[\sqrt{b\xi^2+1}-1\right]. \tag{103}$$

Recall that $\xi$ specifies the auxiliary vertical coordinate, which is defined on the interval [0,1] at each point $(\alpha,\beta,n_p)$. This choice of vertical grid spacing is chosen to match operational conditions, where additional resolution is generally desired near the surface. Here $b$ is a flattening parameter which we choose to be $b$ = 10. It determines how quickly model levels tend towards equal spacing at higher altitudes.

We make use of the RK4 time stepping strategy (66)–(69) for the explicit integration with a fixed time step. All of the test cases examined in this document are written in terms of spherical coordinates with longitude $\lambda$ and latitude $\varphi$. Coordinate transforms are provided in Appendix A for translating these quantities to the cubed-sphere.

### *5.1. Baroclinic instability*

The baroclinic instability test of Jablonowski and Williamson [20] has become an important test case for hydrostatic dynamical cores. Although this test is idealized, the background field is chosen to closely resemble the known background state of the real atmosphere. As such, this test may shed light into the treatment of realistic atmospheric motions by the atmospheric model. This test case was originally formulated in pressure-based coordinates, which are defined by $\eta = p/p_s$, and so had to be translated to height coordinates as outlined by the procedure in the appendix of [20].

The initial state without perturbations is in both hydrostatic and gradient-wind balance, and hence in a perfect model it should be maintained indefinitely. However, numerical errors due to the discretization will nonetheless creep into the solution and cause the steady nature of this profile to be lost. These errors eventually accumulate and result in the collapse of the flow into turbulent motions which are determined by the discrete properties of the numerical method.

In order to trigger an instability which is more deterministic in its evolution, Jablonowski and Williamson [20] introduce an overlaid perturbation in the zonal velocity field. They select a Gaussian profile centered at $(\lambda_c,\varphi_c) = (\pi/9, 2\pi/9)$, which leads to an instability in the northern jet. The perturbation is Gaussian, given by

$$u'(\lambda,\varphi,\eta) = u_p \exp\left(-\left(\frac{r_c}{R}\right)^2\right), \tag{104}$$

with radius $R = a/10$ and maximum amplitude $u_p$ = 1 m s$^{-1}$. Here $r_c$ is the great circle distance

$$r_c = a\arccos[\sin\varphi_c \sin\varphi + \cos\varphi_c \cos\varphi \cos(\lambda-\lambda_c)], \tag{105}$$

where $a$ denotes the Earth's radius.

The hydrostatic background state for these simulations is given by the initial conditions (without the perturbation). The test case is run for 10 days at c90 resolution, which roughly corresponds to 111 km equatorial resolution or 1° grid spacing, and a time step of 250 s. In Fig. 4 we plot snapshots of the baroclinic wave test case at day 7 and 9 showing surface pressure, 850 hPa temperature and 850 hPa relative vorticity. The latter two are interpolated to pressure levels from height levels using a linear interpolant. An intercomparison of the performance of various dynamical cores on this test case can be found in Jablonowski and Williamson [20] and Lauritzen et al. [21]. The results from MCore are very competitive with other dynamical cores, correctly capturing the location and strength of pressure minima and maintaining sharp gradients in the relative vorticity field. Of particular note is the fact that our results do not show any visually apparent grid imprinting. That is, there are no clear signs of wave number four forcing which would be triggered by anisotropy in the cubed-sphere grid.

In order to understand model convergence under mesh refinement for this test, we have run this test at c60, c90 and c180 resolution and compared the surface pressure against a c360 simulation which is used as reference. Following Jablonowski and Williamson [20] we compute the root mean square (RMS) norms from

$$l_2(p_s(t)) \approx \left[\frac{\sum_k((p_s)_k - (p_s^R)_k)^2|\mathcal{Z}|_k}{\sum_k|\mathcal{Z}|_k}\right]^{1/2}, \tag{106}$$

where $(p_s)_k$ represents the extrapolated surface pressure in element $k$ and $(p_s^R)_k$ is the extrapolated surface pressure in the c360 reference solution which has been averaged onto the lower resolution grid. The sums in (106) span all elements in the

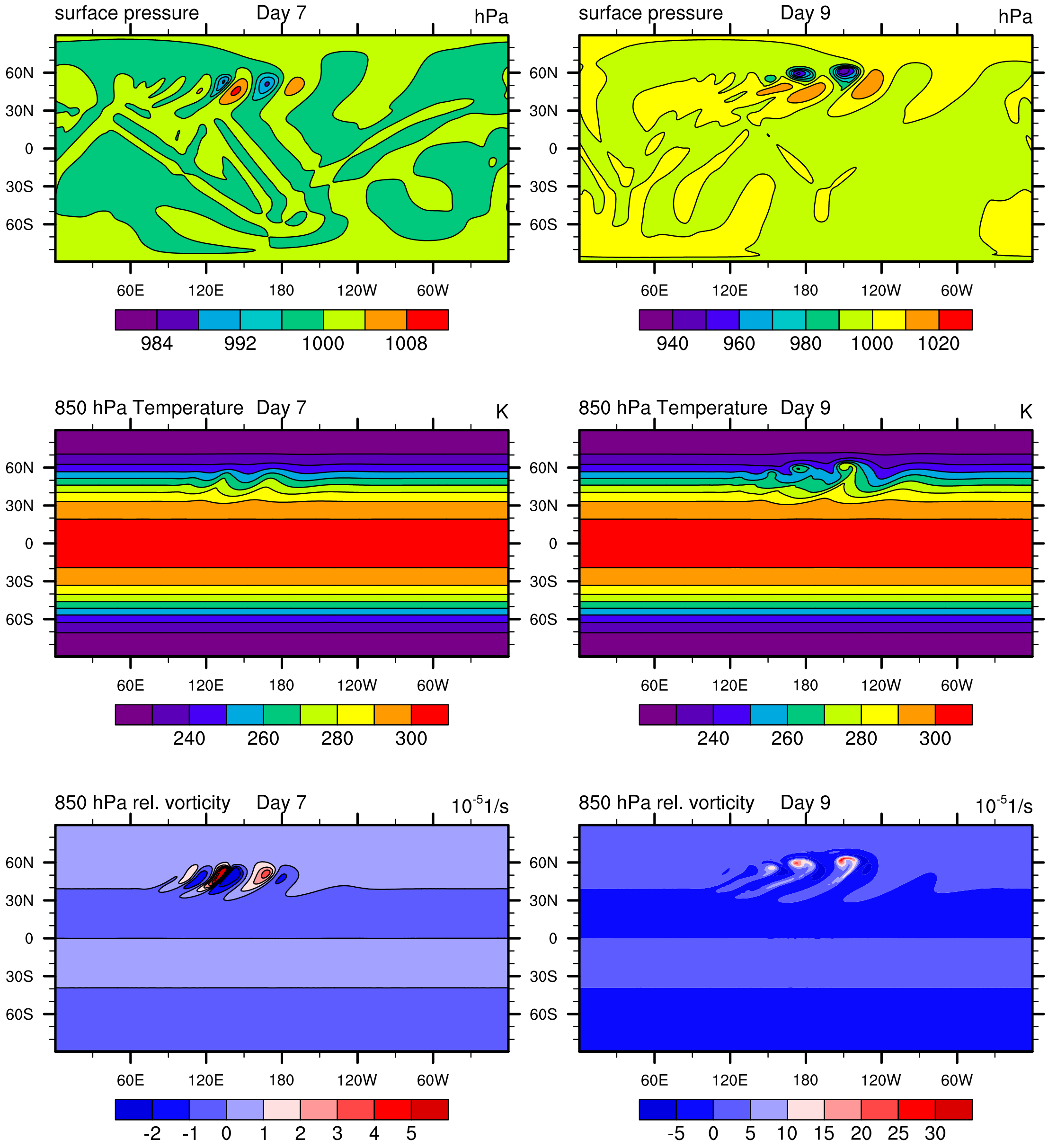

surface pressure Day 7 hPa
surface pressure Day 9 hPa
60N
30N
0
30S
60S
60E
120E
180
120W
60W
984
992
1000
1008
940
960
980
1000
1020
850 hPa Temperature Day 7 K
850 hPa Temperature Day 9 K
240
260
280
300
850 hPa rel. vorticity Day 7 10-5 1/s
850 hPa rel. vorticity Day 9 10-5 1/s
-2
-1
0
1
2
3
4
5
-5
0
5
10
15
20
25
30

**Fig. 4.** Snapshots from the baroclinic wave test case at day 7 and 9 simulated on a c90 grid with 26 vertical levels and 30 km model cap. The time step is chosen to be $\Delta t = 250$ s. Surface pressure is plotted in the upper row, 850 hPa temperature in the middle row and 850 hPa relative vorticity in the bottom row.

global domain. The RMS errors are plotted in Fig. 5. In general our results are competitive with other model simulations presented in [20]. As is typical for this test case, rough first-order convergence is observed in the error norms. The error norms for the c90 and c180 simulations lie well below the uncertainty range for this test case, as estimated in [20] and denoted by the gray shaded area in Fig. 5.

### *5.2. 3D Rossby–Haurwitz wave*

The Rossby–Haurwitz wave test case is a 3D extension of the 2D shallow-water Rossby–Haurwitz wave described by Williamson et al. [55]. The 3D test has also been described in Giraldo and Rosmond [12], although this reference specifies some test case parameters inaccurately. The form of this test used here is described in Jablonowski et al. [19]. The Rossby–Haurwitz wave is an analytical solution of the barotropic vorticity equation that features an unsteady wave that translates westward at a known velocity. The wave is not an analytic solution of the full non-hydrostatic equations of motion, but

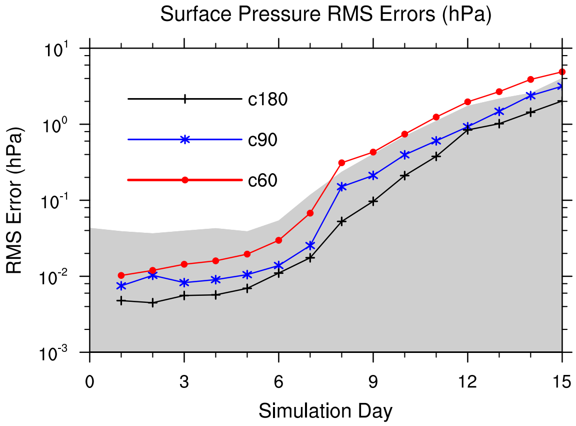


**Fig. 5.** RMS errors in the global surface pressure relative to a c360 baroclinic instability simulation over 15 days. In all cases we use 26 vertical levels and a 30 km model cap. The time step is chosen to be $\Delta t = 250$ s at c90 and is otherwise scaled to maintain a constant CFL number. The gray shaded region represents a measure of errors that arise due to uncertainties in the numerically computed solution as estimated in [20].

under these equations the solution still inherits many characteristic features. The wave still translates westward at a roughly constant velocity and approximately preserves its shape over time. As such, this test is very helpful in determining the ability of a model to maintain this shape over time.

The initial velocity field is non-divergent and defined by the streamfunction

$$\psi_s(\lambda, \varphi) = -a^2 M \sin\varphi + a^2 K \cos^n\varphi \sin\varphi \cos(n\lambda), \tag{107}$$

where the parameters $M$ and $K$ are chosen such that $M = K = u_0/(na)$ with $u_0 = 50$ m s$^{-1}$ and $n = 4$. For the non-divergent 2D barotropic model, Haurwitz [16] showed that this streamfunction moves in the zonal direction without change of shape with an angular velocity given by

$$\nu = \frac{n(3+n)M - 2\omega}{(1+n)(2+n)}, \tag{108}$$

which for the given choice of parameters corresponds to a westward propagation with a period of approximately 24 days. Simulations with 3D primitive equation models, however, instead yield motions with a period of approximately 26 days in practice [49]. The initial vertical velocity is set to zero and should ideally remain so.

The horizontal velocity components are vertically uniform and given by

$$u(\lambda, \varphi) = aM\cos\varphi + aK\cos^{n-1}\varphi\cos(n\lambda)(n\sin^2\varphi - \cos^2\varphi), \tag{109}$$

$$v(\lambda, \varphi) = -aKn\cos^{n-1}\varphi\sin\varphi\sin(n\lambda). \tag{110}$$

The vertical temperature profile is characterized by a constant lapse rate,

$$T = T_0 - \Gamma\tilde{z}, \tag{111}$$

with $T_0 = 288$ K and $\Gamma = 0.0065$ K m$^{-1}$. Here $\tilde{z}$ is the equivalent height, defined via

$$\tilde{z} = \frac{T_0}{\Gamma}\left(1 - \left(\frac{p}{p_{\text{ref}}}\right)^{\frac{\Gamma R_d}{g}}\right), \tag{112}$$

where $p_{\text{ref}}$ is a reference pressure, chosen to be $p_{\text{ref}} = 955$ hPa. The equivalent height and geopotential height are related via the formula

$$\Phi = gz = g\tilde{z} + \Phi'(\lambda, \varphi). \tag{113}$$

Solving (112) and (113) for pressure as a function of height then gives the functional relation

$$p = p_{\text{ref}}\left(1 - \frac{\Gamma(gz - \Phi'(\lambda, \varphi))}{gT_0}\right)^{\frac{g}{R_d\Gamma}}. \tag{114}$$

Here $\Phi'$ is the geopotential perturbation, defined by

$$\Phi' = a^2[A(\varphi) + B(\varphi)\cos(n\lambda) + C(\varphi)\cos(2n\lambda)], \tag{115}$$

where

$$A(\varphi) = \frac{M(2\omega + M)}{2}\cos^2\varphi + \frac{K^2}{4}\cos^{2n}\varphi\left[(n+1)\cos^2\varphi + (2n^2 - n - 2)\right] - \frac{n^2K^2}{2}\cos^{2(n-1)}\varphi, \tag{116}$$

$$B(\varphi) = \frac{2(\omega + M)K}{(n+1)(n+2)}\cos^n\varphi[(n^2 + 2n + 2) - (n+1)^2\cos^2\varphi], \tag{117}$$

$$C(\varphi) = \frac{K^2}{4}\cos^{2n}\varphi[(n+1)\cos^2\varphi - (n+2)]. \tag{118}$$

The density is recovered from (114) via the ideal gas law $\rho = p/(R_dT)$ and potential temperature $\theta$ via (7).

The hydrostatic background state is chosen to be a profile with constant lapse rate $\Gamma$ and constant surface temperature $T_0$. This test case is run for 15 days. No damping layer is used at the model top. In Fig. 6 we plot a snapshot of the Rossby–Haurwitz wave test case at day 15 showing the 850 hPa zonal and meridional wind, surface pressure, 850 hPa temperature,

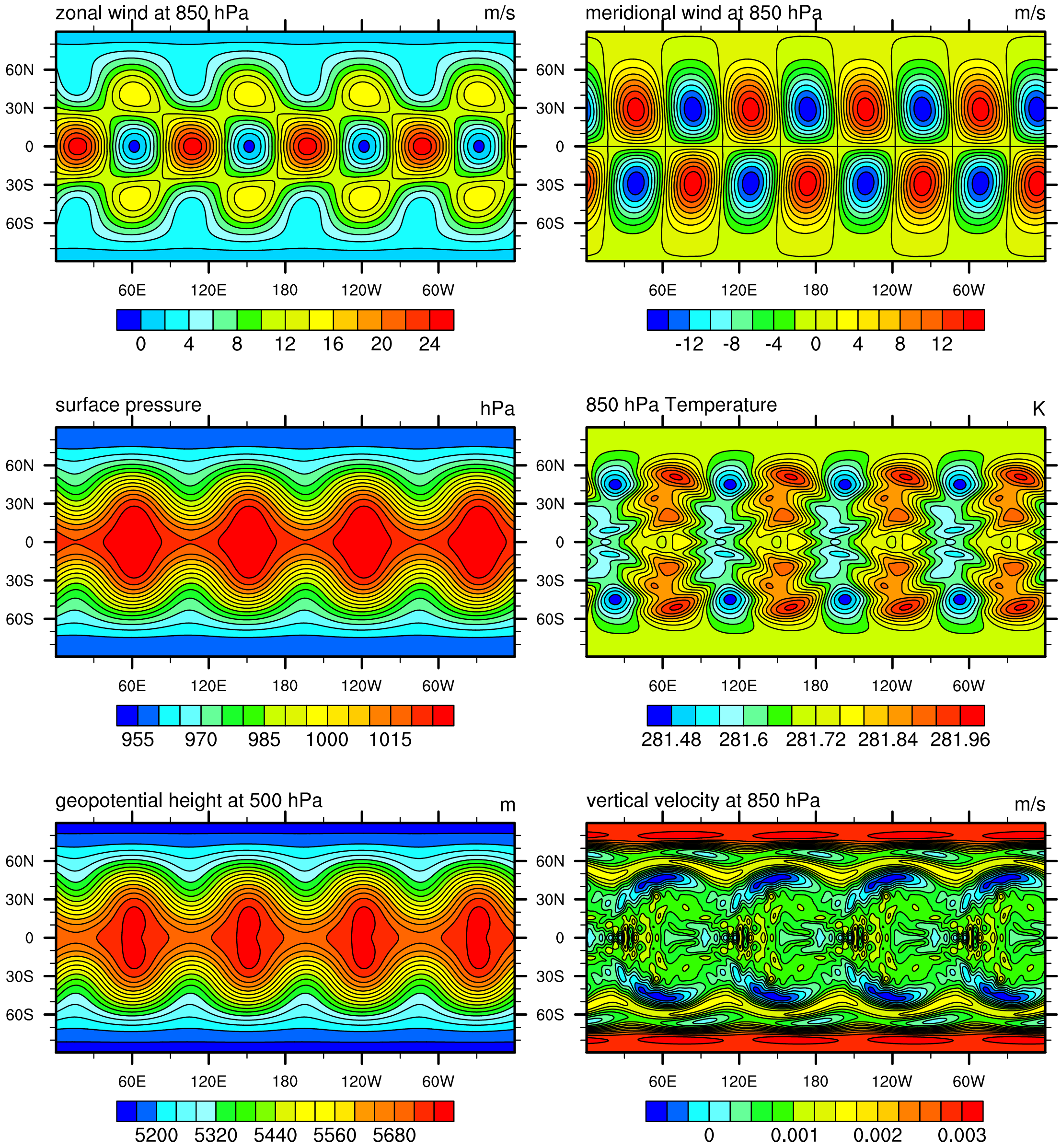


**Fig. 6.** Snapshots from the Rossby–Haurwitz wave at day 15 simulated on a c90 grid with 26 vertical levels and 30 km model cap. Zonal and meridional wind (both at 850 hPa) are plotted in the top row, surface pressure and temperature at 850 hPa are shown in the middle row and 500 hPa geopotential height and 850 hPa vertical velocity are plotted in the bottom row.

500 hPa geopotential height and 850 hPa vertical velocity. Using other dynamical cores for comparison [18], we note that the wave speed of the Rossby–Haurwitz wave is correctly captured by our method. Further, we observe only small variations in the vertical velocity field, no signs of north–south symmetry breakage and overall observe no obvious signs of instability by day 15.

### 5.3. Mountain-induced Rossby wave-train

The mountain-induced Rossby wave train is again an adaptation of a similar shallow-water test case from Williamson et al. [55]. The test case used in this paper most closely resemble that of Tomita and Satoh [44], and is described in detail in Jablonowski et al. [19]. This test begins with smooth isothermal initial conditions that are a balanced analytic solution of the primitive equation in the absence of topography. An idealized mountain then triggers the evolution of a Rossby wave train, which is modeled over a period of 30 days. This test produces significant dynamical motions and so is useful for testing the robustness of the model and its treatment of the model top. The presence of strong vertical motions in the vicinity of the mountain leads to wave reflection in most models at the upper boundary and so a sponge layer is generally needed. There is no known analytic solution to this test case, but several known numerical solutions are available [18]. Model solutions tend to diverge observably after day 15 as small-scale numerical oscillations are brought to the grid scale.

The initial components of the horizontal wind in spherical coordinates are

$$u(\lambda,\varphi)=u_0\cos\varphi,\quad v(\lambda,\varphi)=0, \tag{119}$$

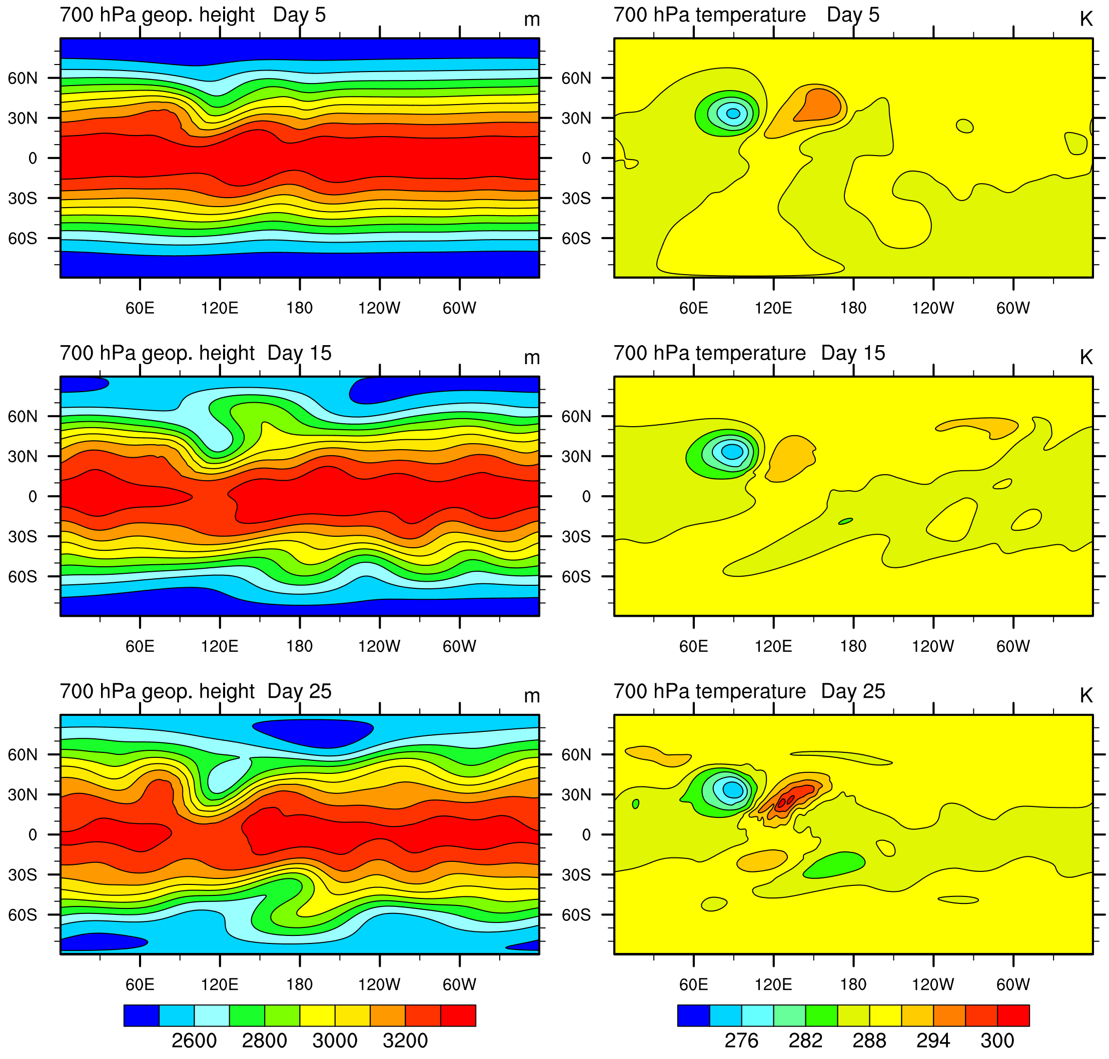

700 hPa geop. height Day 5
m
700 hPa temperature Day 5
K
700 hPa geop. height Day 15
700 hPa temperature Day 15
700 hPa geop. height Day 25
700 hPa temperature Day 25
60N
30N
0
30S
60S
60E
120E
180
120W
60W
2600
2800
3000
3200
276
282
288
294
300

**Fig. 7.** Snapshots from the mountain-induced Rossby-wave train wave at day 5 (top row), day 15 (middle row) and day 25 (bottom row) simulated on a c 90 grid with 26 vertical levels and 30 km model cap. Geopotential height and temperature at 700 hPa are shown in the left and right column, respectively.

where the maximum amplitude of the zonal wind $u_0$ is set to 20 m s$^{-1}$ and the vertical velocity is zero. The atmosphere is initially isothermal with $T_0 = 288$ K, which gives a constant Brunt–Väisälä frequency

$$N = \sqrt{\frac{g}{c_p T_0}} \approx 0.0182 \text{ s}^{-1}. \tag{120}$$

An idealized bell-shape mountain is introduced with surface height

$$z_s(\lambda, \varphi) = g h_0 \exp\left[-\left(\frac{r_c}{d}\right)^2\right], \tag{121}$$

where $h_0 = 2000$ m is the peak height of the mountain and $d = 1500$ km is the half-width of the Gaussian profile. Here $r_c$ denotes the great circle distance (105) with centerpoint $(\lambda_c, \varphi_c) = (\pi/2, \pi/6)$. The surface pressure $p_s$ is chosen to balance the initial conditions,

$$p_s(\lambda, \varphi) = p_p \exp\left[-\frac{a N^2 u_0}{2 g^2 \kappa}\left(\frac{u_0}{a} + 2\omega\right)(\sin^2\varphi - 1) - \frac{N^2}{g\kappa} z_s(\lambda, \varphi)\right], \tag{122}$$

with Pole pressure $p_p = 930$ hPa and $\kappa = R_d/c_p = 2/7$.

The hydrostatic background state is chosen to match the initial conditions. This test case is run for 30 days. Rayleigh friction is used at the model top. In Figs. 7 and 8 we plot the results over the simulation period, showing 700 hPa geopotential

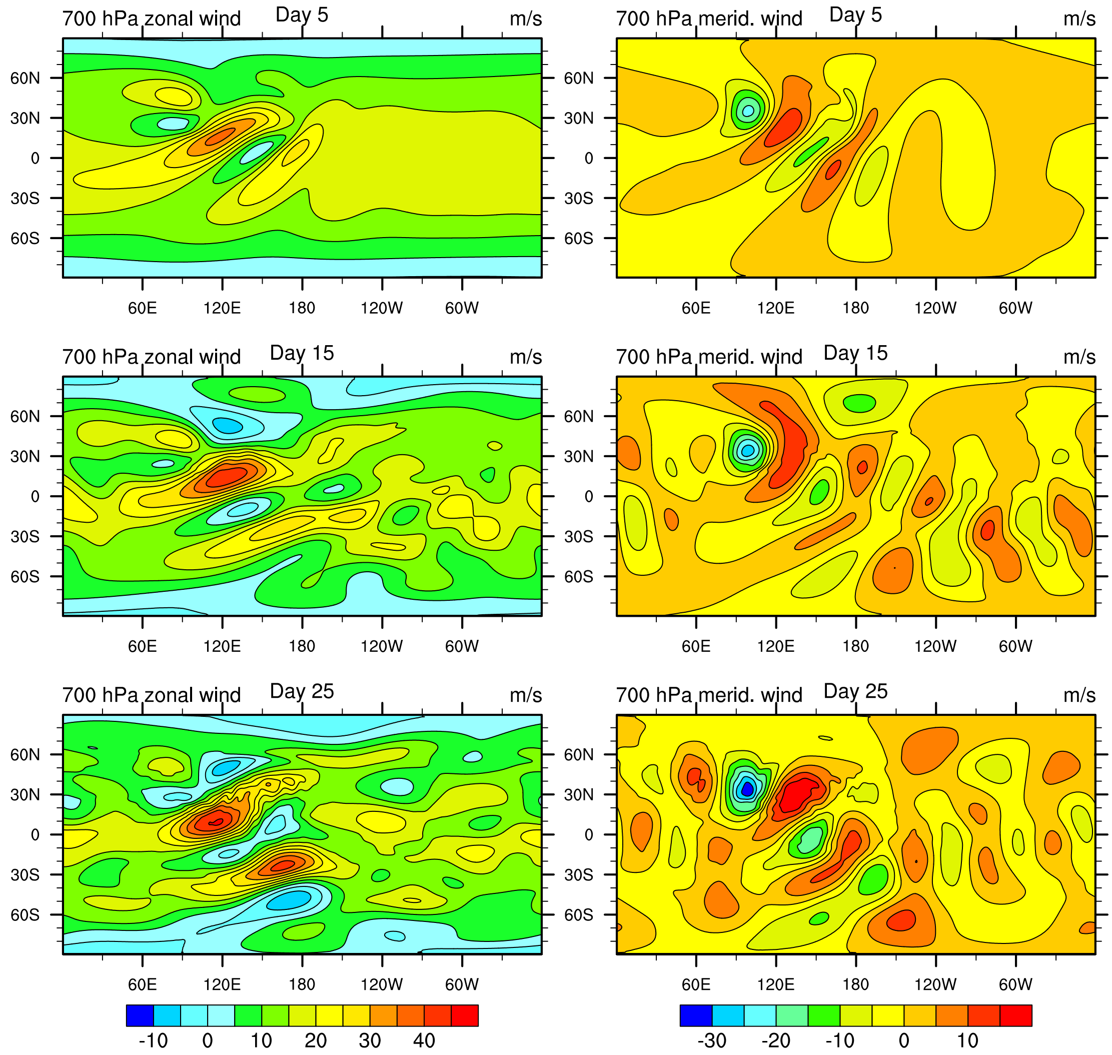


**Fig. 8.** Snapshots from the mountain-induced Rossby-wave train wave at day 5 (top row), day 15 (middle row) and day 25 (bottom row) simulated on a c90 grid with 26 vertical levels and 30 km model cap. Zonal and meridional wind at 700 hPa are shown in the left and right column, respectively.

height, 700 hPa temperature, 700 hPa zonal wind and 700 hPa meridional wind at day 5, 15 and 25. With the Rayleigh damping layer in place we observe good agreement with known results from hydrostatic models [18]. After 15 days uncertainties introduced by the numerical discretization lead to divergence in model solutions and so the flow becomes less predictable.

### 5.4. Gravity waves

This test case (test 6–0–0 in [19]) explores the propagation of gravity waves in a non-rotating ($\omega = 0$) domain which are triggered by an initial perturbation in the potential temperature field. The background field consists of a zero velocity field ($\mathbf{u} = \mathbf{0}$ m s$^{-1}$) and a constant Brunt–Väisälä frequency $\mathcal{N} = 0.01$ s$^{-1}$. This leads to a horizontally uniform pressure field defined by

$$p(z) = p_0\left[\left(1 - \frac{S}{T_0}\right) + \frac{S}{T_0}\exp\left(-\frac{\mathcal{N}^2 z}{g}\right)\right]^{c_p/R_d}, \tag{123}$$

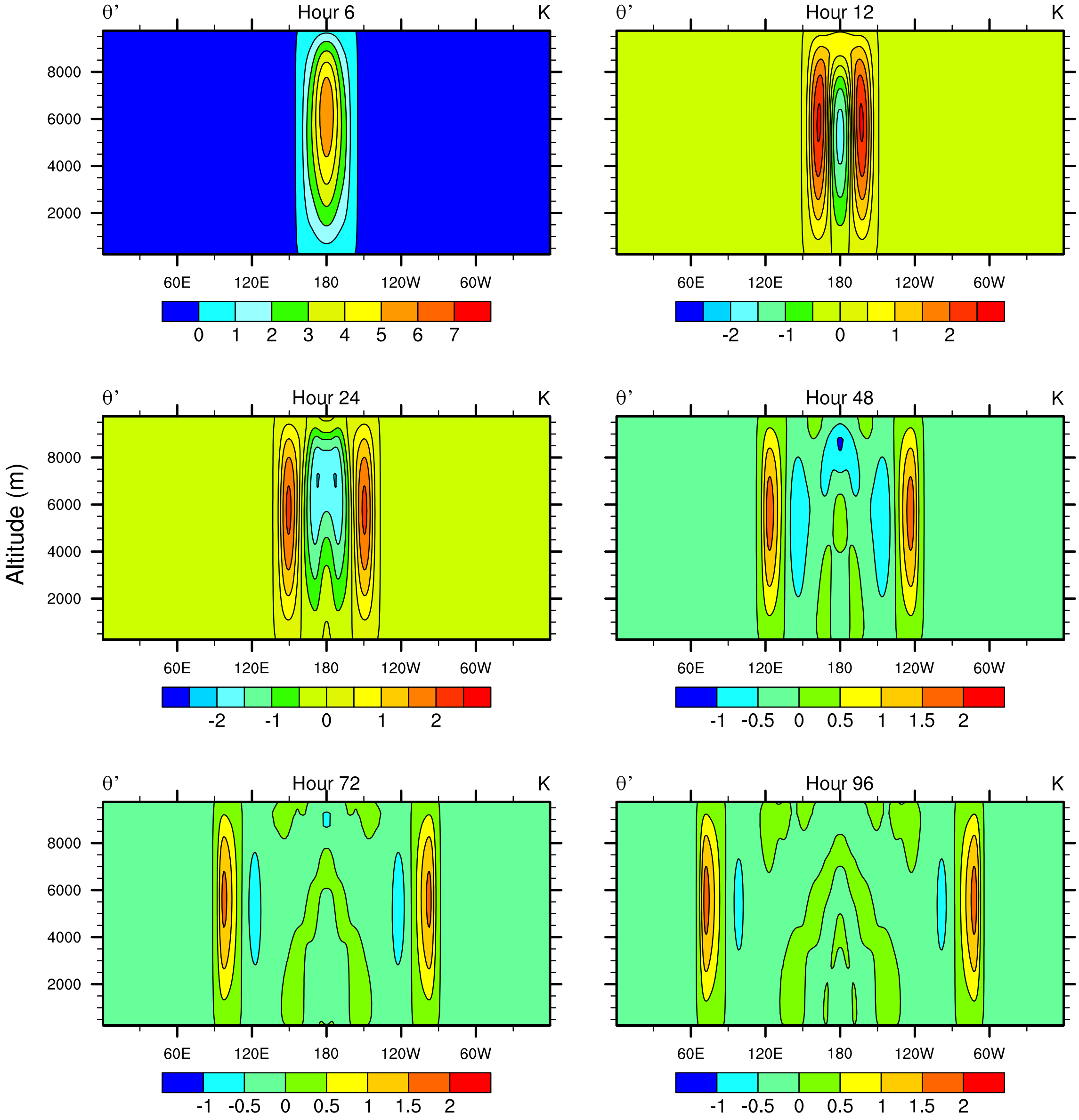


**Fig. 9.** Snapshots along the equator from the gravity wave test at hours 6, 12, 24, 48, 72 and 96 simulated on a c90 grid with 20 uniformly spaced vertical levels and a 10 km model cap. Potential temperature perturbation $\theta'$ at each time is depicted here along the equator.

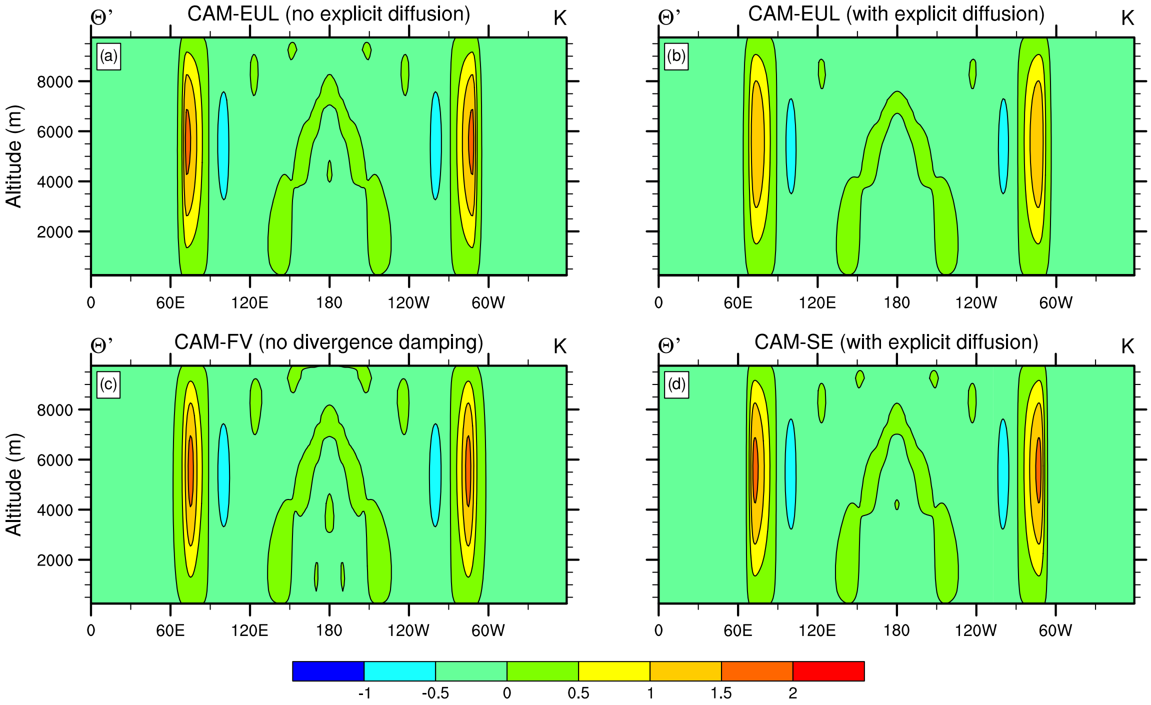


**Fig. 10.** Snapshots along the equator from the gravity wave test at hour 96 simulated using CAM-EUL with (a) no diffusion and (b) default diffusion, (c) CAM-FV with no divergence damping and (d) CAM-SE with default diffusion. All plots are at 96 h and so are comparable to hour 96 in Fig. 9. The model levels are approximately equally spaced in the vertical, since each of these models use a hydrostatic pressure-based coordinate in the vertical.

where $p_0 = 1000$ hPa, $S = g^2/(c_p\mathcal{N}^2)$ and $T_0 = 300$ K. The height-based potential temperature field $\theta = \bar{\theta} + \theta'$ is composed of a hydrostatically balanced mean state $\bar{\theta}$ and the perturbation $\theta'$. It is given by

$$\theta(\lambda, \varphi, z) = T_0 \exp\left(\frac{\mathcal{N}^2 z}{g}\right) + \Delta\theta s(\lambda, \varphi) \sin\left(\frac{2\pi z}{L_z}\right), \tag{124}$$

where $\Delta\theta = 10$ K is the maximum potential temperature perturbation and $L_z = 20$ km is the vertical wave length of the perturbation. The horizontal shape function $s(\lambda, \varphi)$ is defined as

$$s(\lambda, \varphi) = \begin{cases} \frac{1}{2}(1 + \cos(\pi r/R)) & \text{if } r < R, \\ 0 & \text{if } r \geqslant R, \end{cases} \tag{125}$$

with $R = a/3$ and $r$ the great circle distance (105) between $(\lambda, \varphi)$ and the center of the cosine bell, initially set to $(\lambda_c, \varphi_c) = (\pi, 0)$. The model top is chosen to be 10 km to correspond with half a wave length of the perturbation. No damping layer is used at the top of the atmosphere. The hydrostatic background state is given by (123) and (124) without the perturbation.

This test case is run for 4 days on a c90 grid with 20 equidistant ($\Delta z = 500$ m) vertical levels (L20) and 10 km model cap. In Fig. 9 we plot the longitude-height ($\lambda - z$) profile of the potential temperature perturbation from the initial basic state at each level along the equator over the four day period (96 h). The results along the equator at $t = 96$ h are also shown in Fig. 10 for the spectral-transform-based CAM Eulerian (CAM-EUL) dynamical core both without and with explicitly added fourth-order hyperdiffusion using the diffusion coefficient $K_4 = 5 \times 10^{14}$ m$^4$ s$^{-1}$, the CAM finite-volume (CAM-FV) [22] dynamical core without its divergence damping mechanism [54], and the spectral-element-based CAM-SE model with fourth-order hyperdiffusion and $K_4 = 9.6 \times 10^{14}$ m$^4$ s$^{-1}$ [27]. The CAM-FV and CAM-SE models are run with a corresponding 1° × 1°$L$20 grid spacing, which translates to a triangular truncation of T106 for the L20 CAM-EUL model. The results from MCore are competitive with these models. They show signicantly less diffusion than CAM-EUL with its default hyperdiffusion (Fig. 10(b)) and CAM-FV (Fig. 10(c)), while maintaining a nearly identical contour pattern with sharp gradients at the leading edges of the gravity wave (Fig. 10(a) and (d)). In addition, the phase speed of the wave is consistent in all model simulations.

### 5.5. Held–Suarez climatology

The Held–Suarez test [17] is a first step towards verification of a climate model for real-world climate simulations. This test makes use of a highly idealized physics forcing which is used to represent both diabatic heating/cooling and

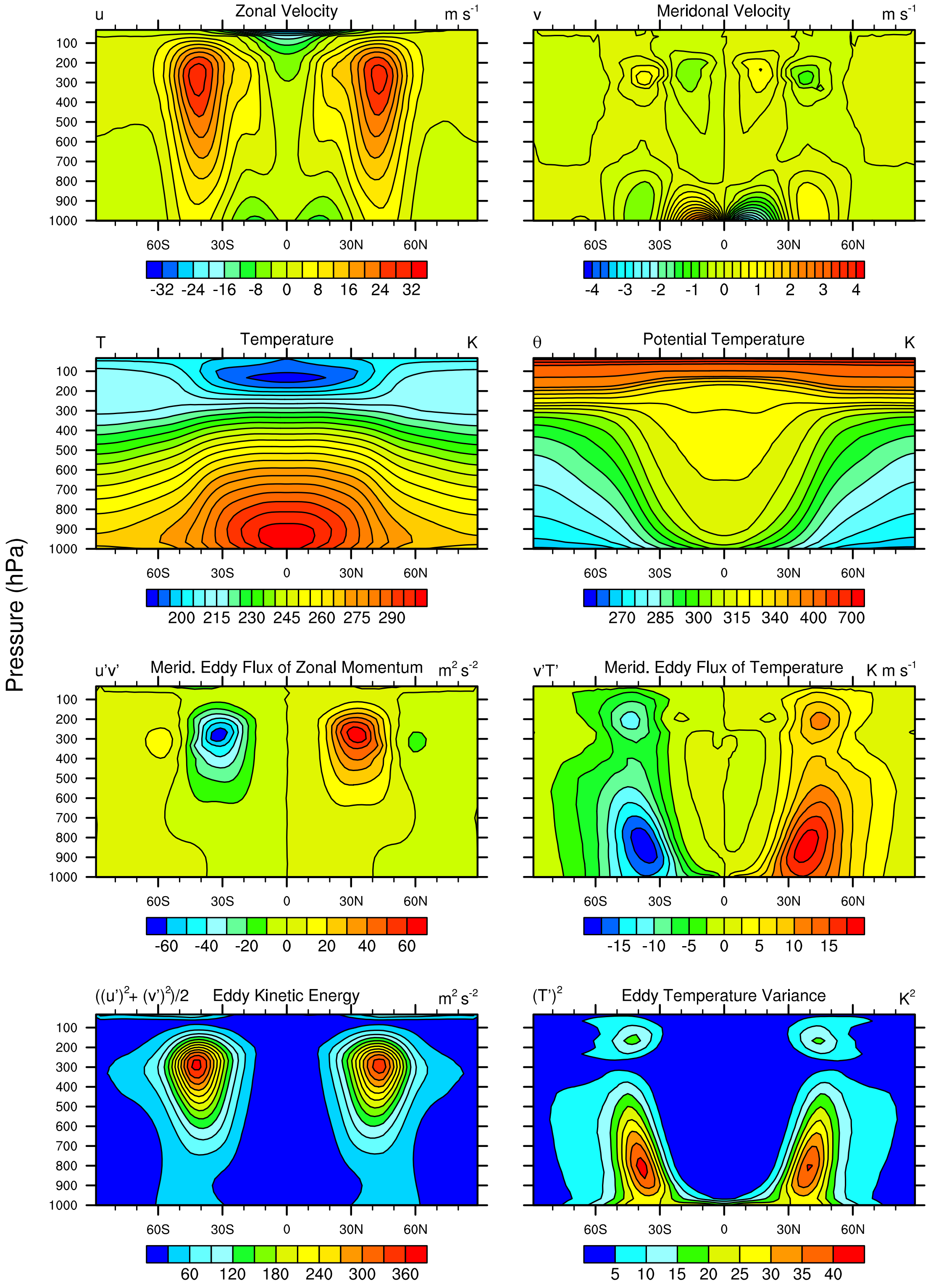


**Fig. 11.** 1000-day integrated climatology with Held–Suarez forcing from a c48 run with 30 vertical levels and 30 km model cap. The time step is chosen to be $\Delta t = 420$ s.

boundary-layer friction. Climate statistics are obtained from a long-term integration of the fluid equations under these conditions, which can then be compared against the solutions from other models.

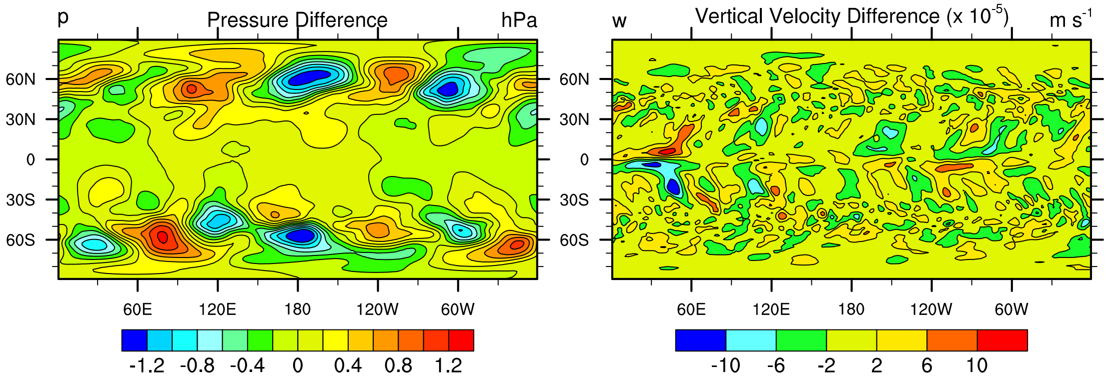


**Fig. 12.** 1000-day averages of pressure (left) and vertical velocity (right) with zonal means removed for the lowest model level from the Held–Suarez climatology test. The results are from a c48 run with 30 vertical levels and 30 km model cap with time step is chosen to be $\Delta t = 420$ s.

The additional Held–Suarez relaxation terms in the equations of motion are applied at the end of each time step following the dynamical update for time level $n + 1$ described in Section 4.4. The boundary layer friction is computed using a backward Euler step,

$$(\rho\mathbf{u})^* = (1 + k_v(\sigma)\Delta t)^{-1}(\rho\mathbf{u})^{n+1},$$

which is applied to all components of the velocity field, where $\sigma = p^{n+1}/p_s^{n+1}$ and $k_v$ is identical to the choice made in [17]. The temperature relaxation is applied to the potential temperature density $(\rho\theta)$ using a similar procedure, since modifications to the density would violate conservation of mass. Specifically, we compute

$$(\rho\theta)^* = (\rho\theta)^{n+1} + \Delta t(1 - \Delta t\partial\mathcal{H})^{-1}\mathcal{H},$$

where

$$\mathcal{H} = -\frac{k_T}{\gamma}\left(1 - \frac{T_{eq}}{T}\right)(\rho\theta),$$

and

$$\partial\mathcal{H} = -\frac{k_T}{\gamma}\left(1 + (\gamma - 1)\frac{T_{eq}}{T}\right).$$

The relaxation coefficient $k_T$ and equilibrium temperature $T_{eq}$ is again chosen in accordance with [17]. The temperature $T$ is computed in terms of the density and potential temperature density using (7) and the ideal gas law,

$$T = \frac{p_0}{\rho R_d}\left(\frac{R_d(\rho\theta)}{p_0}\right)^{c_p/c_v}$$

After application of the physics forcing the modified state variables $(\rho\mathbf{u})^*$ and $(\rho\theta)^*$ then replace the dynamic estimates $(\rho\mathbf{u})^{n+1}$ and $(\rho\theta)^{n+1}$.

A 1200-day simulation using the Held–Suarez forcing is run on a c48 grid (approximately 208 km equatorial grid spacing) with 30 vertical levels and $\Delta t = 420$ s. The background state is initially prescribed based on hydrostatic balance and the equilibrium temperature given in [17], but is reinitialized at day 200 to using a 50 day average of $\rho$ and $\rho\theta$ from between day 150 and 200. This choice allows the background state to more closely match the quasi-steady climatology one would expect from a Held–Suarez simulation. A temporal average from days 200 to 1200 is then computed using the zonal average of the zonal velocity $u$, meridional velocity $v$, temperature $T$, potential temperature $\theta$, meridional eddy flux of zonal momentum $u'v'$, meridional eddy flux of temperature $v'T'$, eddy kinetic energy $(\frac{1}{2}(u'^2 + v'^2))$ and eddy temperature variance $(T')^2$. The prime on each variable denotes the deviation from the zonal mean. The results of this 1000-days integrated run are then plotted in Fig. 11. Our results closely match with known results from the Held–Suarez climatology, such as those of [50]. To verify that grid imprinting from the cubed-sphere grid is not a dominant influence on the climatology, we have examined the perturbation of the pressure and vertical velocity from its zonal average in the lowest model level, as averaged over days 200 to 1200. These results are shown in Fig. 12. In particular, since vertical velocity is generally small compared to other variables, we expect that grid imprinting will be most apparent in this field. However, these results do not suggest any significant influence on either the pressure or vertical velocity from grid imprinting within the model.

## 6. Conclusions and future work

In this paper we have developed a new atmospheric dynamical core which uses high-order finite-volume methods for solving the non-hydrostatic equations of motion under the shallow-atmosphere approximation. The model is built on a cubed-sphere grid with a height-based vertical coordinate. Under the upwind finite-volume methodology, a sub-grid-scale reconstruction is built within each element using neighboring element values. The reconstruction we propose is novel, incorporating geometric terms over the minimal stencil required for fourth-order accuracy. Along edges where the reconstruction is discontinuous we make use of the AUSM$^+$-up Riemann solver of Liou [23] to compute fluxes in the horizontal, and a modified quasi-linear variant of the AUSM$^+$-up scheme in the vertical. To avoid restrictions due to fast vertically propagating wave modes, all terms responsible for vertical motion are computed using an iterative implicit step. The model additionally has the option for Rayleigh damping at the model top to damp out potential wave reflection at the upper boundary.

The resulting method has been tested on a variety of problems and has been shown to be stable, robust and accurate. In addition to the test results on the sphere given in this paper, the splitting strategy used by this model has also been thoroughly tested in Cartesian geometry in Ullrich and Jablonowski [45]. However, insofar all of the tests that have been applied have made use of an idealized setup, which does not necessarily test the whole range of real atmospheric motions. In the future, additional testing is necessary to ensure that the model is able to correctly simulate real atmospheric flows. In particular, we are currently investigating the non-hydrostatic test suite of Wedi and Smolarkiewicz [51], which will better allow us to evaluate the treatment of non-hydrostatic motions by MCore.

MCore is still in the experimental stages and so significant work remains to be done before it can be used operationally. A monotonicity preserving transport scheme still remains to be included in MCore, which will require evaluation and testing of a variety of transport algorithms. Further, physical parameterizations must be incorporated in operational versions of the model.

## Acknowledgments

Support for this work has been provided by the Office ofScience, US Department of Energy, Award No. DE-SC0003990 and a University of Michigan Rackham Predoctoral Fellowship. We would also like to thank the two anonymous reviewers for their constructive comments in improving this work.

## Appendix A. Geometric properties of cubed-sphere coordinates

In this appendix we present metric terms and coordinate transforms which arise from the use of cubed-sphere coordinates in formulating the non-hydrostatic equations of motion in the deep atmosphere. We will make ample use of Einstein summation notation, where repeated indices imply summation over that index. Under cubed-sphere coordinates the indices take on values $\alpha$, $\beta$ and $r$, which correspond to the first, second and third coordinate direction.

### *A.1. The metric*

The metric is identical on each panel of the cubed-sphere grid, but varies depending on the coordinates within each panel. Using either covariant and contravariant indices, the 2D metric on a surface of constant $r$ is defined as follows (note that $G^{ij} = (G_{ij})^{-1}$).

$$\widehat{G}_{ij} = \frac{r^2(1+Y^2)(1+X^2)}{\delta^4}\begin{pmatrix} 1+X^2 & -XY \\ -XY & 1+Y^2 \end{pmatrix}, \tag{A.1}$$

$$\widehat{G}^{ij} = \frac{\delta^2}{r^2(1+X^2)(1+Y^2)}\begin{pmatrix} 1+Y^2 & XY \\ XY & 1+X^2 \end{pmatrix}. \tag{A.2}$$

The radial basis vector is everywhere orthogonal to surfaces of constant $r$, as with spherical coordinates, and has unit length by construction. Hence, the complete metric in 3D can be decomposed into a 2D component along with a unit radial component,

$$G_{ij} = \begin{pmatrix} \widehat{G}_{ij} & 0 \\ 0 & 1 \end{pmatrix}, \quad G^{ij} = \begin{pmatrix} \widehat{G}^{ij} & 0 \\ 0 & 1 \end{pmatrix}. \tag{A.3}$$

In curvilinear coordinates, the metric is responsible for determining the length of basis vectors as well as the orthogonality properties of the coordinate system. The inner product of two vectors is defined as

$$\langle \mathbf{v}, \mathbf{w} \rangle = G_{ij} v^i w^j, \tag{A.4}$$

where $v^j$ and $w^j$ denote the components of $\mathbf{v}$ and $\mathbf{w}$ in the cubed-sphere basis. The magnitude of a vector in the cubed-sphere basis can be constructed via the inner product, $|\mathbf{v}| = \langle \mathbf{v}, \mathbf{v} \rangle$. The determinant of the covariant metric determines the infinitesimal volume element via

$$J = \sqrt{\det G_{ij}} = \frac{r^2(1+X^2)(1+Y^2)}{\delta^3}, \quad d\mathbf{V} = J d\alpha d\beta dr. \tag{A.5}$$

Similarly, the determinants of the cofactor matrices of $G_{ij}$ determine the infinitesimal areas along surfaces where one variable is held constant:

$$J_\alpha = \frac{r(1+Y^2)\sqrt{1+X^2}}{\delta^2} \quad dA_\alpha = J_\alpha d\beta dr, \tag{A.6}$$

$$J_\beta = \frac{r(1+X^2)\sqrt{1+Y^2}}{\delta^2} \quad dA_\beta = J_\beta d\alpha dr, \tag{A.7}$$

$$J_r = \frac{r^2(1+X^2)(1+Y^2)}{\delta^3} \quad dA_r = J_r d\alpha d\beta. \tag{A.8}$$

Other vector operations are similarly defined via the metric and its byproducts. The cross-product of two vectors is defined as

$$(\mathbf{u} \times \mathbf{v})^i = JG^{ij}\epsilon_{jk\ell}u^k v^\ell, \tag{A.9}$$

with third-order permutation symbol $\epsilon_{j\,k\ell}$, defined via

$$\epsilon_{jk\ell} = \begin{cases} +1 & \text{if } (j,k,\ell) \text{ is } (\alpha,\beta,r),(r,\alpha,\beta) \quad \text{or } (\beta,r,\alpha), \\ -1 & \text{if } \quad (j,k,\ell) \text{ is } (\alpha,r,\beta),(r,\beta,\alpha) \quad \text{or } (\beta,\alpha,r), \\ 0 & \text{otherwise.} \end{cases} \tag{A.10}$$

Under the natural basis the gradient operator is

$$\nabla^i \phi = G^{ij}\frac{\partial \phi}{\partial x^j}, \tag{A.11}$$

and the divergence operator takes the form

$$\nabla \cdot \mathbf{F} = \frac{1}{J}\frac{\partial}{\partial x^k}(JF^k), \tag{A.12}$$

with $x^k = (x^\alpha, x^\beta, x^r) = (\alpha, \beta, r)$.

### *A.2. Christoffel symbols of the second kind*

The Christoffel symbols of the second kind represent the effect of transport of a vector field along coordinate lines. They appear in certain derivative operations applied to tensor fields, such as the divergence of the two-index tensor flux of the momentum. In terms of the metric, they are

$$\Gamma^i_{jk} = \frac{1}{2}G^{im}\left(\frac{\partial G_{jm}}{\partial x^k} + \frac{\partial G_{km}}{\partial x^j} - \frac{\partial G_{jk}}{\partial x^m}\right). \tag{A.13}$$

In equiangular cubed-sphere coordinates we obtain the following expressions.

$$\Gamma^\alpha = \begin{pmatrix} \frac{2XY^2}{\delta^2} & \frac{-Y(1+Y^2)}{\delta^2} & \frac{1}{r} \\ \frac{-Y(1+Y^2)}{\delta^2} & 0 & 0 \\ \frac{1}{r} & 0 & 0 \end{pmatrix}, \tag{A.14}$$

$$\Gamma^\beta = \begin{pmatrix} 0 & \frac{-X(1+X^2)}{\delta^2} & 0 \\ \frac{-X(1+X^2)}{\delta^2} & \frac{2X^2Y}{\delta^2} & \frac{1}{r} \\ 0 & \frac{1}{r} & 0 \end{pmatrix}, \tag{A.15}$$

$$\Gamma^r = \frac{r(1+X^2)(1+Y^2)}{\delta^4}\begin{pmatrix} -(1+X^2) & XY & 0 \\ XY & -(1+Y^2) & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{A.16}$$

Using these definitions, we can write the divergence of a two-index tensor $T_{ij}$ as

$$\mathrm{div}_j T^{ij} = \frac{1}{J}\frac{\partial}{\partial x^j}(JT^{ij}) + \Gamma^i_{jk}T^{jk}. \tag{A.17}$$

### *A.3. Source terms*

In its most generic form, the gravitational source term of the momentum equation takes the form

$$\boldsymbol{\psi}_G = -\rho g \left(\frac{a}{r}\right)^2 \mathbf{e}_r, \tag{A.18}$$

with radial basis vector $\mathbf{e}_r$.

The full Coriolis source terms of the momentum equation is

$$\boldsymbol{\psi}_C = -2\boldsymbol{\Omega} \times (\rho \mathbf{u}), \quad \boldsymbol{\Omega} = \omega[\cos\varphi \mathbf{e}_\varphi + \sin\varphi \mathbf{e}_r], \tag{A.19}$$

at latitude $\varphi$ and with latitudinal unit basis vector $\mathbf{e}_\varphi$. In terms of cubed-sphere coordinates, the angular velocity vector $\boldsymbol{\Omega}$ is

$$\boldsymbol{\Omega} = \begin{cases} \omega\left[\frac{\delta}{r(1+Y^2)}\mathbf{g}_\beta + \frac{Y}{\delta}\mathbf{g}_r\right] & \text{for equatorial panels } (n_p < 5), \\ s\omega\left[-\frac{X\delta}{r(1+X^2)}\mathbf{g}_\alpha - \frac{Y\delta}{r(1+Y^2)}\mathbf{g}_\beta + \frac{1}{\delta}\mathbf{g}_r\right] & \text{for polar panels } \quad (n_p \geqslant 5) \end{cases} \tag{A.20}$$

### *A.4. Transformation matrices from spherical coordinates*

The change of variables matrices are defined via the relation

$$\begin{bmatrix} v^\alpha \\ v^\beta \end{bmatrix} = \mathbf{A}\begin{bmatrix} v^\lambda \\ v^\varphi \end{bmatrix}, \quad \begin{bmatrix} v^\lambda \\ v^\varphi \end{bmatrix} = \mathbf{A}^{-1}\begin{bmatrix} v^\alpha \\ v^\beta \end{bmatrix}, \tag{A.21}$$

where $\lambda$ is the longitude (chosen so that $\lambda = 0$ corresponds to $\alpha = 0$ on panel 1) and $\varphi$ is the latitude ($\varphi = 0$ here corresponds to the equator). Both sets of equations are defined in the natural coordinate basis.

#### *A.4.1. Equatorial panels*

Equatorial panels are denoted by an index $n_p \in \{1,2,3,4\}$, where $n_p = 1$ corresponds to the panel containing the longitude line $\lambda = 0$. The point coordinate transformation between cubed-sphere coordinates and spherical coordinates on these panels is defined by the following relations:

$$\alpha = \lambda - \frac{\pi}{2}(n_p - 1), \quad \beta = \arctan\left(\frac{\tan\varphi}{\cos\lambda}\right), \tag{A.22}$$

$$\lambda = \alpha + \frac{\pi}{2}(n_p - 1), \quad \varphi = \arctan(\tan\beta\cos\alpha). \tag{A.23}$$

On equatorial panels, the change of variables matrices are defined in terms of the gnomonic coordinate $(X, Y)$ as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ \frac{XY}{1+Y^2} & \frac{\delta^2}{(1+Y^2)\sqrt{1+X^2}} \end{bmatrix}, \tag{A.24}$$

$$\mathbf{A}^{-1} = \begin{bmatrix} 1 & 0 \\ \frac{-XY\sqrt{1+X^2}}{\delta^2} & \frac{(1+Y^2)\sqrt{1+X^2}}{\delta^2} \end{bmatrix}. \tag{A.25}$$

#### *A.4.2. Polar panels*

Polar panels are denoted by an index $n_p \in \{5,6\}$ where the index $p = 5$ corresponds to the north polar panel and the index $n_p = 6$ corresponds to the south polar panel. We define a panel indicator $s$ as

$$s = \begin{cases} 1 & \text{if } n_p = 5, \\ -1 & \text{if } n_p = 6. \end{cases} \tag{A.26}$$

The pointwise coordinate transforms then take the form:

$$\alpha = s\arctan(\cot\varphi\sin\lambda), \quad \beta = -\arctan(\cot\varphi\cos\lambda), \tag{A.27}$$

$$\lambda = -\arctan\left(\frac{\tan\alpha}{\tan\beta}\right), \quad \varphi = s\,\mathrm{arccot}(\sqrt{\tan^2\alpha + \tan^2\beta}). \tag{A.28}$$

The pointwise change of variables matrices are defined by:

$$\mathbf{A} = \begin{bmatrix} \frac{-sY}{1+X^2} & \frac{-s\delta^2 X}{(1+X^2)\sqrt{X^2+Y^2}} \\ \frac{sX}{1+Y^2} & \frac{-s\delta^2 Y}{(1+Y^2)\sqrt{X^2+Y^2}} \end{bmatrix}. \tag{A.29}$$

$$\mathbf{A}^{-1} = \begin{bmatrix} -\frac{sY(1+X^2)}{X^2+Y^2} & \frac{sX(1+Y^2)}{X^2+Y^2} \\ -\frac{sX(1+X^2)}{\delta^2\sqrt{X^2+Y^2}} & -\frac{sY(1+Y^2)}{\delta^2\sqrt{X^2+Y^2}} \end{bmatrix}. \tag{A.30}$$

## Appendix B. The shallow-atmosphere approximation

The shallow-atmosphere approximation has the effect of reducing the vertically varying structure of the atmosphere to a set of stacked layers. Layers are approximated to have radius $r = a$, which removes geometric terms associated with increasing area as altitude increases. This approximation was first introduced by Phillips [28]. It is described in conjunction with other consistent approximations of the full non-hydrostatic primitive equations in White et al. [53].

Following the metric formulation described in Appendix A, the shallow-atmosphere approximation follows by simply replacing all instances of $r$ in the deep-atmosphere metric (A.1) and (A.2) with the radius of the Earth $a$. As a result, the infinitesimal volume element becomes

$$J = \sqrt{\det G_{ij}} = \frac{a^2(1+X^2)(1+Y^2)}{\delta^3}, \quad d\mathbf{V} = Jd\alpha d\beta dr. \tag{B.1}$$

Infinitesimal areas along surfaces (A.6)–(A.8) likewise become

$$J_\alpha = \frac{a(1+Y^2)\sqrt{1+X^2}}{\delta^2} \quad dA_\alpha = J_\alpha d\beta dr, \tag{B.2}$$

$$J_\beta = \frac{a(1+X^2)\sqrt{1+Y^2}}{\delta^2} \quad dA_\beta = J_\beta d\alpha dr, \tag{B.3}$$

$$J_r = \frac{a^2(1+X^2)(1+Y^2)}{\delta^3} \quad dA_r = J_r d\alpha d\beta. \tag{B.4}$$

The Christoffel symbols are significantly affected by this change, now taking the form

$$\Gamma^\alpha = \begin{pmatrix} \frac{2XY^2}{\delta^2} & \frac{-Y(1+Y^2)}{\delta^2} & 0 \\ \frac{-Y(1+Y^2)}{\delta^2} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{B.5}$$

$$\Gamma^\beta = \begin{pmatrix} 0 & \frac{-X(1+X^2)}{\delta^2} & 0 \\ \frac{-X(1+X^2)}{\delta^2} & \frac{2X^2Y}{\delta^2} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{B.6}$$

$$\Gamma^r = 0. \tag{B.7}$$

The gravitational source term (A.18) further simplifies to

$$\psi_G = -\rho g \mathbf{e}_r. \tag{B.8}$$

Finally, conservation of energy requires that so-called “cosine Coriolis terms” be dropped from the momentum evolution equation. As a consequence, all $\mathbf{g}_\alpha$ and $\mathbf{g}_\beta$ dependence of (A.20) is dropped and (A.19) simplifies to

$$\psi_C = -f\mathbf{g}_r \times (\rho\mathbf{u}), \tag{B.9}$$

where the Coriolis parameter is

$$f = \frac{2\omega}{\delta}\begin{cases} Y & \text{for equatorial panels} \quad (n_p < 5) \\ s & \text{for polar panels} \quad (n_p \geqslant 5). \end{cases} \tag{B.10}$$

## References

[1] N. Ahmad, J. Lindeman, Euler solutions using flux-based wave decomposition, Int. J. Numer. Methods Fluids 54 (2007) 47–72.

[2] A. Arakawa, V. Lamb, Computational design and the basic dynamical processes of the UCLA general circulation model, Methods Comput. Phys. 17 (1977) 173–265.

[3] M. Barad, P. Colella, A fourth-order accurate local refinement method for Poisson's equation, J. Comput. Phys. 209 (2005) 1–18.

[4] R. Bubnová, G. Hello, P. Bénard, J.-F. Geleyn, Integration of the fully elastic equations cast in the hydrostatic pressure terrain-following coordinate in the framework of the ARPEGE/Aladin NWP system, Mon. Weather Rev. 123 (1995) 515–535.

[5] C. Chen, F. Xiao, Shallow water model on cubed-sphere by multi-moment finite volume method, J. Comput. Phys. 227 (2008) 5019–5044.

[6] T. Davies, M.J.P. Cullen, A.J. Malcolm, M.H. Mawson, A. Staniforth, A.A. White, N. Wood, A new dynamical core for the met office's global and regional modelling of the atmosphere, Quart. J. Roy. Meteor. Soc. 131 (608) (2005) 1759–1782.

[7] J. Dennis, J. Edwards, K.J. Evans, O.N. Guba, P.H. Lauritzen, A.A. Mirin, A. St-Cyr, M.A. Taylor, P.H. Worley, CAM-SE: a scalable spectral element dynamical core for the community atmosphere model, Int. J. High Perform. Comput. Appl. (2011).

[8] L.J. Donner, B.L. Wyman, R.S. Hemler, L.W. Horowitz, Y. Ming, M. Zhao, J.-C. Golaz, P. Ginoux, S.-J. Lin, M.D. Schwarzkopf, J. Austin, G. Alaka, W.F. Cooke, T.L. Delworth, S.M. Freidenreich, C.T. Gordon, S.M. Griffies, I.M. Held, W.J. Hurlin, S.A. Klein, T.R. Knutson, A.R. Langenhorst, H.-C. Lee, Y. Lin, B.I. Magi, S.L. Malyshev, P.C.D. Milly, V. Naik, M.J. Nath, R. Pincus, J.J. Ploshay, V. Ramaswamy, C.J. Seman, E. Shevliakova, J.J. Sirutis, W.F. Stern, R.J. Stouffer, R.J. Wilson, M. Winton, A.T. Wittenberg, F. Zeng, The Dynamical core, physical parameterizations, and basic simulation characteristics of the atmospheric component AM3 of the GFDL global coupled model CM3, J. Climate, 2011.

[9] A. Fournier, M.A. Taylor, J.J. Tribbia, The spectral element atmosphere model (SEAM): high-resolution parallel computation and localized resolution of regional dynamics, Mon. Weather Rev. 132 (3) (2004) 726–748.

[10] T. Gal-Chen, R.C.J. Somerville, On the use of a coordinate transformation for the solution of the Navier–Stokes equations, J. Comput. Phys. 17 (1975) 209–228.

[11] A. Gassmann, Non-hydrostatic modelling with ICON, in: Proceedings of the ECMWF Workshop on Non-hydrostatic Modelling, Reading, U.K., November 8–10, 2011, pp. 131–142. <http://www.ecmwf.int/publications/library/do/references/list/201010>.

[12] F.X. Giraldo, T.E. Rosmond, A scalable spectral element Eulerian atmospheric model (SEE-AM) for NWP: dynamical core tests, Mon. Weather Rev. 132 (2004) 133–153.

[13] S.K. Godunov, A difference scheme for numerical solution of discontinuous solution of hydrodynamic equations, Math. Sbornik 47 (1959) 271–306. translated US Joint Publ. Res. Service, JPRS 7226, 1969.

[14] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Rev. 43 (1) (2001) 89–112.

[15] M.W. Govett, J. Middlecoff, T. Henderson, Running the NIM next-generation weather model on GPUs, in: IEEE International Symposium on Cluster Computing and the Grid, 2010, pp. 792–796.

[16] B. Haurwitz, The motion of atmospheric disturbances on the spherical earth, J. Mar. Res. 3 (1940) 254–267.

[17] I.M. Held, M.J. Suarez, A proposal for the intercomparison of the dynamical cores of atmospheric general circulation models, B. Am. Meteorol. Soc. 75 (1994) 1825–1830.

[18] C. Jablonowski, P.H. Lauritzen, M.A. Taylor, R.D. Nair, An intercomparison of 10 atmospheric model dynamical cores, Eos Transactions AGU, Fall Meeting Supplement Abstract A33A-0214, vol. 89, no. 53, 2008a.

[19] C. Jablonowski, P.H. Lauritzen, M.A. Taylor, R.D. Nair, Idealized test cases for the dynamical cores of atmospheric general circulation models: a proposal for the NCAR ASP 2008 summer colloquium, Technical Report, 2008b. <http://esse.engin.umich.edu/admg/publications.php>.

[20] C. Jablonowski, D.L. Williamson, A baroclinic instabilitiy test case for atmospheric model dynamical cores, Quart. J. Roy. Meteor. Soc. 132 (621C) (2006) 2943–2975.

[21] P.H. Lauritzen, C. Jablonowski, M.A. Taylor, R.D. Nair, Rotated versions of the Jablonowski steady-state and baroclinic wave test cases: a dynamical core intercomparison, J. Adv. Model. Earth Syst. 2 (15) (2010) 34.

[22] S.-J. Lin, A "vertically Lagrangian" finite-volume dynamical core for global models, Mon. Weather Rev. 132 (2004) 2293–2307.

[23] M.-S. Liou, A sequel to AUSM, Part II: AUSM+-up for all speeds, J. Comput. Phys. 214 (2006) 137–170.

[24] D. Majewski, The new global icosahedral-hexagonal gridpoint model GME of the Deutscher Wetterdienst, in: Proceedings Seminar on Numerical Methods for Weather Prediction, ECMWF, Reading, United Kingdom, vol. 1, 1998, pp. 172–201.

[25] D. Majewski, D. Liermann, P. Prohl, B. Ritter, M. Buchhold, T. Hanisch, G. Paul, W. Wergen, J. Baumgardner, The operational global icosahedral hexagonal gridpoint model GME: description and high-resolution tests, Mon. Weather Rev. 130 (2002) 319–338.

[26] R.D. Nair, S.J. Thomas, R.D. Loft, A discontinuous Galerkin global shallow water model, Mon. Weather Rev. 133 (2005) 876–888.

[27] R.B. Neale, C.-C. Chen, A. Gettelman, P.H. Lauritzen, S. Park, D.L. Williamson, A.J. Conley, R. Garcia, D. Kinnison, J.-F. Lamarque, D. Marsh, M. Mills, A.K. Smith, S. Tilmes, F. Vitt, P. Cameron-Smith, W.D. Collins, M.J. Iacono, R.C. Easter, X. Liu, S.J. Ghan, P.J. Rasch, M.A. Taylor, Description of the NCAR community atmosphere model (CAM 5.0), NCAR technical note NCAR/TN-486+STR, National Center for Atmospheric Research, Boulder, Colorado, 2010. <http://www.cesm.ucar.edu/models/cesm1.0/cam/>.

[28] N.A. Phillips, The equations of motion for a shallow rotating atmosphere and the 'traditional approximation', J. Atmosp. Sci. 23 (1966) 626–627.

[29] W.M. Putman, S.-J. Lin, Finite-volume transport on various cubed-sphere grids, J. Comput. Phys. 227 (2007) 55–78.

[30] W.M. Putman, S.-J. Lin, A finite-volume dynamical core on the cubed-sphere grid, in: N.V. Pogorelov, E. Audit, P. Colella, G.P. Zank, (Eds.), Numerical modeling of space plasma flows: astronum-2008, Astronomical Society of the Pacific Conference Series, vol. 406, 2009, pp. 268–276.

[31] M. Rančić, J. Purser, F. Mesinger, A global shallow water model using an expanded spherical cube, Quart. J. Roy. Meteor. Soc. 122 (1996) 959–982.

[32] T.D. Ringler, R.P. Heikes, D.A. Randall, Modeling the atmospheric general circulation using a spherical geodesic grid: a new class of dynamical cores, Mon. Weather Rev. 128 (2000) 2471–2490.

[33] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357–372.

[34] C. Ronchi, R. Iacono, P.S. Paolucci, The "cubed sphere": a new method for the solution of partial differential equations in spherical geometry, J. Comput. Phys. 124 (1) (1996) 93–114.

[35] J.A. Rossmanith, A wave propagation method for hyperbolic systems on the sphere, J. Comput. Phys. 213 (2006) 629–658.

[36] V. Rusanov, Calculation of intersection of non-steady shock waves with obstacles, J. Comput. Math. Phys. USSR 1 (1961) 267–279.

[37] R. Sadourny, Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids, Mon. Weather Rev. 100 (1972) 136–144.

[38] M. Satoh, T. Matsuno, H. Tomita, H. Miura, T. Nasuno, S. Iga, Nonhydrostatic icosahedral atmospheric model (NICAM) for global cloud resolving simulations, J. Comput. Phys. 227 (2008) 3486–3514.

[39] W.C. Skamarock, J.B. Klemp, M. Duda, L. Fowler, S.-H. Park, Global non-hydrostatic modelling using Voronoi meshes: the MPAS model, in: Proceedings of the ECMWF Workshop on Non-hydrostatic Modelling, Reading, 2010, U.K. November 8–10, 2011, pp. 121–130. <http://www.ecmwf.int/publications/library/do/references/list/201010>.

[40] W.C. Skamarock, J.B. Klemp, A time-split nonhydrostatic atmospheric model for weather research and forecasting applications, J. Comput. Phys. 227 (2008) 3465–3485.

[41] A. Staniforth, N. Wood, Aspects of the dynamical core of a nonhydrostatic, deep-atmosphere, unified weather and climate-prediction model, J. Comput. Phys. 227 (2008) 3445–3464.

[42] M.A. Taylor, J. Tribbia, M. Iskandarani, The spectral element method for the shallow water equations on the sphere, J. Comput. Phys. 130 (1997) 92–108.

[43] M.A. Taylor, A. Fournier, A compatible and conservative spectral element method on unstructured grids, J. Comput. Phys. 229 (17) (2010) 5879–5895.

[44] H. Tomita, M. Satoh, A new dynamical framework of nonhydrostatic global model using the icosahedral grid, Fluid Dyn. Res. 34 (6) (2004) 357–400.

[45] P.A. Ullrich, C. Jablonowski, Operator-split Runge–Kutta–Rosenbrock (RKR) methods for non-hydrostatic atmospheric models, Mon. Weather Rev. 140 (2012) 1257–1284.

[46] P.A. Ullrich, C. Jablonowski, B.L. van Leer, High-order finite-volume models for the shallow-water equations on the sphere, J. Comput. Phys. 229 (2010) 6104–6134.

[47] P.A. Ullrich, P.H. Lauritzen, C. Jablonowski, Geometrically exact conservative remapping (GECoRe): regular latitude-longitude and cubed-sphere grids, Mon. Weather Rev. 137 (2009) 1721–1741.

[48] R.L. Walko, R. Avissar, The ocean-land-atmosphere model (OLAM). Part II: formulation and tests of the nonhydrostatic dynamic core, Mon. Weather Rev. 136 (2008) 4045–4062.

[49] H. Wan, Developing and testing a hydrostatic atmospheric dynamical core on triangular grids, Technical Report 65, Reports on Earth System Science, Max–Planck Institute for Meteorology, Hamburg, Germany, 2009, ISSN 1614-119.

[50] H. Wan, M.A. Giorgetta, L. Bonaventura, Ensemble Held Suarez test with a spectral transform model: variability, sensitivity, and convergence, Mon. Weather Rev. 136 (2008) 1075.

[51] N.P. Wedi, P.K. Smolarkiewicz, A framework for testing global non-hydrostatic models, Quart. J. Roy. Met. Soc. 135 (2009) 469–484.

[52] N.P. Wedi, K. Yessad, A. Untch, The nonhydrostatic global IFS/ARPEGE: model formulation and testing. Technical Memorandum, No. 594, European Centre for Medium-Range Weather Forecasts, Reading, U.K., 2009, p. 36.

[53] A.A. White, B.J. Hoskins, I. Roulstone, A. Staniforth, Consistent approximate models of the global atmosphere: shallow, deep, hydrostatic, quasi-hydrostatic and non-hydrostatic, Quart. J. Roy. Met. Soc. 131 (2005) 2081–2107.

[54] J.P. Whitehead, C. Jablonowski, R.B. Rood, P.H. Lauritzen, A stability analysis of divergence damping on a latitude-longitude grid, Mon. Weather Rev. 139 (2011) 2976–2993.

[55] D.L. Williamson, J. Drake, J. Hack, R. Jakob, P. Swarztrauber, A standard test set for numerical approximations to the shallow water equations in spherical geometry, J. Comput. Phys. 102 (1992) 211–224.

[56] D.L. Williamson, Equivalent finite volume and Eulerian spectral transform horizontal resolutions established from aqua-planet simulations, Tellus Ser. A 60 (2008) 839–847.

[57] K. Yeh, J. Côté, S. Gravel, A. Méthot, A. Patoine, M. Roch, A. Staniforth, The CMC MRB global environmental multiscale (GEM) model. Part III: Nonhydrostatic formulation, Mon. Weather Rev. 130 (2002) 339–356.