ORIGINAL ARTICLE

A mixed finite-element, finite-volume, semi-implicit discretisation for atmospheric dynamics: Spherical geometry

Thomas Melvin[1] | Ben Shipway[1] | Nigel Wood[1] | Tommaso Benacchio[1†] | Thomas Bendall[1] | Ian Boutle[1] | Alex Brown[1] | Christine Johnson[1] | James Kent[1] | Stephen Pring[1] | Chris Smith[1] | Mohamed Zerroukat[1] | Colin Cotter[2] | John Thuburn[3]

[1]Met Office, Exeter, United Kingdom

[2]Imperial College, London, United Kingdom

[3]University of Exeter, United Kingdom

[†]Current Affiliation: Danish Meteorological Institute, Copenhagen, Denmark

Correspondence
Thomas Melvin, Met Office, FitzRoy Road, Exeter EX1 3PB, United Kingdom
Email: Thomas.Melvin@metoffice.gov.uk

Funding information
NERC through grants NE/K006762/1 & NE/K006789/1.
EPSRC through grants EP/L016613/1 & EP/R029423/1.

The reformulation of the Met Office's dynamical core for weather and climate prediction previously described by the authors is extended to spherical domains using a cubed-sphere mesh. This paper updates the semi-implicit mixed finite-element formulation to be suitable for spherical domains. In particular the finite-volume transport scheme is extended to take account of non-uniform, non-orthogonal meshes and uses an advective-then-flux formulation so that increment from the transport scheme is linear in the divergence. The resulting model is then applied to a standard set of dry dynamical core tests and compared to the existing semi-implicit semi-Lagrangian dynamical core currently used in the Met Office's operational model.

KEYWORDS
spatial discretisation; temporal discretisation; dynamical core; mimetic discretisation; cubed-sphere

1 | INTRODUCTION

At the centre of all weather and climate models lies the dynamical core. The dynamical core approximates the fluid dynamical motion that is resolved by the model mesh and is coupled to models for unresolved processes such as the boundary layer and non-fluid processes such as radiation. The dynamical core is required to be accurate, stable and efficient for the scales of motion that it simulates. Fundamental to achieving these properties is the choice of model mesh. This choice can result in a number of features that need to be addressed by the numerical scheme, such as resolution clustering, non-orthogonality, grid imprinting and computational modes; see Staniforth and Thuburn [1] for a more detailed discussion.

Modern supercomputers consist of a greatly increasing number of (increasingly heterogeneous) processors and in order to take advantage of this computational resource the dynamical core needs to make efficient use of memory management and communication processes (Lawrence et al. [2]). This has led to a shift away from the regular Latitude-Longitude mesh (which, due to convergence of the meridians at the poles, leads to computational bottlenecks) and towards some form of quasi-uniform horizontal mesh. Staniforth and Thuburn [1] detailed a number of desirable properties that any numerical scheme designed for dynamical cores should exhibit, including mimetic/compatible properties, at least 2nd order accuracy and minimal grid imprinting. Achieving these properties is non-trivial, particularly on a non-orthogonal meshes, such as is typically the case for quasi-uniform meshes on the sphere.

Cotter and Shipton [3], Cotter and Thuburn [4] and Thuburn and Cotter [5] developed a family of compatible mixed finite-element methods for the shallow-water equations on the sphere where orthogonality of the underlying mesh is not required to achieve good accuracy and hence they are well suited to a quasi-uniform mesh. This family of schemes was applied to a variety of icosahedral and cubed-sphere meshes. The mixed finite-element approach was extended into three-dimensions by Natale et al. [6] and Melvin et al. [7] who presented an application of this mixed finite-element model in Cartesian geometry using hexahedral elements and coupled to a finite-volume transport scheme and semi-implicit timestepping. Kent et al. [8] then extended the formulation of Melvin et al. [7] to the shallow water equations on a cubed-sphere horizontal mesh. See Cotter [9] for a survey of mixed finite-element methods for geophysical modelling.

The compatible mixed finite-element approach fulfils many of the desirable properties detailed in Staniforth and Thuburn [1] for the design of a dynamical core. Importantly it provides discrete analogues of certain continuous vector calculus identities (such as $\nabla \times \nabla\psi \equiv 0$ and $\nabla \cdot \nabla \times \mathbf{v} \equiv 0$ for all scalar ψ and vector $\mathbf{v}$) as well as sharing many of the good wave dispersion properties of the widely used Arakawa C-grid staggering (Arakawa and Lamb [10]). This scheme meets the necessary conditions for the absence of computational modes (such as a 2:1 ratio of horizontal velocity degrees of freedom to pressure degrees of freedom). Natale et al. [6] and Melvin et al. [11] showed how to create function spaces that mimic the Charney-Phillips grid staggering in the vertical direction which is desirable due to the absence of computational modes and good wave dispersion properties, (Thuburn and Woollings [12]).

In this paper the formulation used by Melvin et al. [7] and Kent et al. [8] is extended to three-dimensional spherical domains on a cubed-sphere mesh. Principal differences from these models are given in Section 2 and the governing equations that are used are revisited in Section 3. The wave dynamic components of the model are spatially discretised using the mixed finite-element method of Cotter and Shipton [3] and the temporal discretisation uses an iterated-semi-implicit scheme inspired by that of Wood et al. [13], seeking to maintain the temporal accuracy and long timestep stability of that model, Sections 4-5. Mappings from the computational space to the Equiangular cubed-sphere mesh used by the finite-element scheme in this study are given in Section 6. As in Melvin et al. [7] and Kent et al. [8] the finite-element wave dynamics model is coupled to an explicit finite-volume scheme for the transport terms, described in Section 7, which is applied to all model variables. In contrast to the scheme of Melvin et al. [7] and Kent et al. [8],

a directionally split method-of-lines scheme is used here to improve the efficiency of the transport scheme, and an advective-then-flux approach is used to improve stability of conservative transport in this context. Following Wood et al. [13] the iterative timestep is split into outer (transport) and inner (nonlinear) loops and at each iteration a linear system inspired by a semi-implicit formulation is solved as in Maynard et al. [14], outlined in Section 8. To assess the model's behaviour in a range of flow regimes it is applied to a number of dynamical core tests from the literature; the results are presented in Section 9 and compared to the semi-implicit semi-Lagrangian model of Wood et al. [13]. Finally, conclusions are summarised in Section 10.

2 | MODEL FORMULATION

The model formulation closely follows that of Melvin et al. [7] and the formulation is revisited in the following sections. However there are a number of differences that are highlighted here and discussed, together with their motivation, in more detail later:

1. Changes to the finite-volume transport scheme:
   - The momentum equation is reformulated in the advective form instead of the vector invariant form and the advection terms are handled by the explicit finite-volume advection scheme instead of the semi-implicit mixed finite-element method used by Melvin et al. [7] (Section 5)
   - The finite-volume transport scheme is temporally split between vertical and horizontal directions using a Strang splitting method (Section 7.2);
   - The conservative transport scheme uses an advective-then-flux formulation (Section 7.3).
   - The polynomial reconstruction used in the method-of-lines advection scheme uses a two-dimensional horizontal (Kent et al. [8]) and one-dimensional vertical reconstruction (Section 7.4);
2. The geopotential is placed in the $\mathbb{W}_3$ space instead of $\mathbb{W}_0$ (Section 5);
3. The Jacobian $\mathbf{J}$ mapping from the computational space to the physical space is computed with a semi-analytic expression via an intermediate spherical coordinate system (Section 6);
4. The equation of state is sampled at nodal points of the $\mathbb{W}_3$ degrees of freedom instead of being solved in the weak form. (Section 8.3)

3 | CONTINUOUS EQUATIONS

The Euler equations for a dry perfect gas in a rotating frame are

$$\frac{\partial \mathbf{u}}{\partial t} = -\left(\mathbf{u}\cdot\nabla\right)\mathbf{u} + \mathbf{S}, \tag{1}$$

$$\frac{\partial \rho}{\partial t} = -\nabla\cdot\left(\rho\mathbf{u}\right), \tag{2}$$

$$\frac{\partial \theta}{\partial t} = -\mathbf{u}\cdot\nabla\theta, \tag{3}$$

where $\mathbf{S} \equiv -2\boldsymbol{\Omega}\times\mathbf{u} - \nabla\Phi - c_p\theta\nabla\Pi$, together with the nonlinear equation of state

$$\Pi^{\left(\frac{1-\kappa}{\kappa}\right)} = \frac{R}{p_0}\rho\theta. \tag{4}$$

The velocity vector is $\mathbf{u}$; $\boldsymbol{\Omega}$ is the rotation vector; Φ is the geopotential; c_p is the specific heat at constant pressure; θ is the potential temperature, related to temperature through $T = \theta\Pi$; $\Pi = (p/p_0)^{\kappa}$ is the Exner pressure with p pressure and p_0 a constant reference pressure; R is the gas constant per unit mass; $\kappa \equiv R/c_p$; and ρ is the density.

These equations are solved on a spherical shell subject to the boundary condition of zero mass flux through the top and bottom boundaries of the domain.

4 | OVERVIEW OF THE SPATIO-TEMPORAL DISCRETISATION

The temporal discretisation of the equations follows that of Melvin et al. [7] and is inspired by an iterative-semi-implicit semi-Lagrangian discretisation such as that used in Wood et al. [13]. In this scheme the advective terms are handled using an explicit Eulerian scheme. This includes, in contrast to [7], the advection terms in the momentum equation. The scheme acts on an intermediate update of the wave dynamics terms for the variables (see Section 7.1), instead of the time level n terms in [7]. The discretization is in flux form for the continuity equation and advective form for potential temperature and momentum equations. All other terms are handled using an iterative-implicit temporal discretisation. The momentum equation is recast from the vector invariant form of Melvin et al. [7] to the advective form, resulting in a consistent discretisation of all transport terms using the finite-volume transport scheme of Section 7 due to the explicit presence of a transport term for the wind field. The principal advantage of this change is the ability to switch from the centred finite-element discretisation used by [7] to the high-order upwind finite volume discretisation used for the density and potential temperature fields. Alternatively a reconstructed flux type method [15] could be used for the vector invariant form instead of the centred finite-element discretisation.

With the addition of an implicit Rayleigh damping layer (with height dependent strength μ) applied to the vertical component of the velocity vector (1)-(4) are discretised to give

$$\delta_t \mathbf{u} = -\overline{-\mu\left(\frac{\mathbf{u}\cdot\mathbf{n}}{\mathbf{z}\cdot\mathbf{n}}\right)\mathbf{z}}^{1} - \mathcal{A}\left(\mathbf{u}^p, \overline{\mathbf{u}}^{1/2}\right) + \overline{\mathbf{S}}^{\alpha}, \tag{5}$$

$$\delta_t \rho = -\nabla\cdot\mathcal{F}\left(\rho^p, \overline{\mathbf{u}}^{1/2}\right), \tag{6}$$

$$\delta_t \theta = -\mathcal{A}\left(\theta^p, \overline{\mathbf{u}}^{1/2}\right), \tag{7}$$

$$\overline{\Pi^{\left(\frac{1-\kappa}{\kappa}\right)}}^{1} = \overline{\frac{R}{p_0}\rho\theta}^{1}, \tag{8}$$

where, for a generic scalar or vector variable F,

$$\delta_t F \equiv \frac{F^{n+1} - F^n}{\Delta t}, \qquad \overline{F}^{\alpha} \equiv \alpha F^{n+1} + (1-\alpha)\, F^n. \tag{9}$$

The parameter α is a temporal off-centring parameter and the superscripts n and $n+1$ indicate approximations at time $n\Delta t$ and $(n+1)\,\Delta t$ respectively. The advecting velocity $\overline{\mathbf{u}}^{1/2}$ is therefore a centred Eulerian average in time and, in contrast to Melvin et al. [7], the consistent metric modification of $\overline{\mathbf{u}}^{1/2}$ (their Section 5.3.3) is not used here. $\mathcal{F}\left(q^p, \overline{\mathbf{u}}^{1/2}\right)$ is the time-averaged flux and $\mathcal{A}\left(q^p, \overline{\mathbf{u}}^{1/2}\right)$ is the time-averaged advection tendency of a scalar q^p and $\mathcal{A}\left(\mathbf{v}^p, \overline{\mathbf{u}}^{1/2}\right)$ is the time-averaged advection tendency of a vector $\mathbf{v}^p$, where superscripts p indicate an intermediate wave dynamics state of a scalar q or vector field $\mathbf{v}$. See Section 7.1 for more details how the transported states q^p and $\mathbf{v}^p$ are computed.

All terms are discretised in space using the mixed finite-element scheme described in section 5, except for $\mathcal{F}$, $\mathcal{A}$

and $\mathcal{A}$ which are discretised using the finite-volume scheme described in section 7.

5 | THE MIXED FINITE-ELEMENT DISCRETISATION

The finite-element spaces used in the spatial discretisation are the same as those in Melvin et al. [7]. Each variable is represented in an appropriate function space in a hexahedral element:

- $\mathbf{u} \in \mathbb{W}_2$: The Raviart-Thomas RT_l space of vector functions of degree l tangential to and discontinuous across an element facet and degree $l+1$ normal to and continuous across an element facet;
- Π, ρ, $\Phi \in \mathbb{W}_3$: The Q_l^{DG} space of scalar functions built from the tensor product of degree l polynomials that are discontinuous at element boundaries.
- $\theta \in \mathbb{W}_\theta$: The space of scalar functions based on the vertical part of $\mathbb{W}_2$
- The components of the coordinate field $\chi_i \in \mathbb{W}_\chi$, $i = 1, 2, 3$: The Q_m^{DG} space of scalar functions.

As in Melvin et al. [7], and following [16], equations (5)-(8) are transformed from cells in the physical space to a single reference cell in the computational space. This is done so that a single set of basis function and quadrature points can be used in the mixed finite element scheme, instead of defining a different set for each physical cell. The physical space consists of a spherical shell divided into many different cells defined by the model mesh and the computational space consists of a single unit cell. As stated by [7] "It is important that the transformations between the physical and reference cells preserve the various geometric properties of the mixed finite-element discretization. This would happen automatically if the metric tensor of the reference cell were the transformation of the metric tensor of the physical cell, but this would reintroduce a dependency in the reference cell on the physical cell it is mapped with. Instead a Cartesian metric tensor is assumed for the reference cell independently of the physical cell. Therefore, preservation of the required properties is achieved by using a specific collection of transformations that are specific to each function space." These are the Piola transformations that are given below. Following the choice of a Cartesian coordinate system for the physical space the associated metric tensor for both the computational and physical spaces is then the identity and is therefore dropped in the following formulation. Additionally, the basis vectors of the vector wind $\mathbf{u}$ are then the standard Cartesian basis vectors $(\mathbf{e_x}, \mathbf{e_y}, \mathbf{e_z})$. In principle it would be possible to use a spherical coordinate system as the physical coordinates, however this would introduce a non-diagonal metric tensor in the mapping from physical to computational space and would modify the transformations used below.

Given a mapping, ϕ, between physical space (denoted by undressed variables) $\boldsymbol{\chi}$ and computational space (denoted by dressed, $\widehat{\ }$ variables) $\widehat{\boldsymbol{\chi}}$ such that $\boldsymbol{\chi} = \boldsymbol{\phi}(\widehat{\boldsymbol{\chi}})$, with a Jacobian $\mathbf{J} \equiv \partial\boldsymbol{\phi}(\widehat{\boldsymbol{\chi}}) / \partial\widehat{\boldsymbol{\chi}}$, variables in each of the four function spaces transform from computational space to physical space according to the following rules

- $\mathbb{W}_2$: $\mathbf{v} = \mathbf{J}\widehat{\mathbf{v}}/\det\mathbf{J}$;
- $\mathbb{W}_3$: $\sigma = \widehat{\sigma}$ and $\nabla\cdot\mathbf{v} = \widehat{\nabla}\cdot\widehat{\mathbf{v}}/\det\mathbf{J}$ for $\mathbf{v} \in \mathbb{W}_2$;
- $\mathbb{W}_\theta$: $w = \widehat{w}$;
- $\mathbb{W}_\chi$: $\zeta = \widehat{\zeta}$.

These transformations are designed to preserve various geometric properties of the mixed finite element discretisation following the assumption decision to use Cartesian metric for the computational space independent of the choice for the physical space (see Rognes et al. [16] and references within for more details). Note that, as in Melvin et al. [7], to

avoid problems with not being able to exactly integrate the weak form of divergence, the rehabilitation method of Bochev and Ridzal [17] is used. This modifies the mapping of $\sigma \in \mathbb{W}_3$ from $\sigma = \widehat{\sigma}/\det \mathbf{J}$ to $\sigma = \widehat{\sigma}$ which results in a weak divergence $\int \widehat{\sigma}\widehat{\nabla} \cdot \widehat{\mathbf{v}}$ that can be exactly integrated for general cell shapes, see Natale et al. [6] and Melvin et al. [7] for more details. In contrast to Melvin et al. [7], the $\mathbb{W}_0$ and $\mathbb{W}_1$ spaces are not used in this formulation. Placing $\Phi \in \mathbb{W}_3$ gives a more compact stencil for the geopotential gradient that matches that of the pressure gradient. This leads to a small improvement in the model's discrete representation of quasi-hydrostatic balance.

5.1 | Discrete equations using the computational cell

The discretisations of the continuity (6) and thermodynamic (7) equations follow Melvin et al. [7] and so are not repeated here. The momentum equation (5) is now cast in the advective form and therefore the discretisation of the transport terms has altered. The form used here is obtained by multiplying (5) by a test function $\mathbf{v} \in \mathbb{W}_2$, transforming to the computational space and integrating over the domain, giving

$$\left\langle \mathbf{J}\widehat{\mathbf{v}}, \frac{\mathbf{J}\delta_t \widehat{\mathbf{u}}}{\det \mathbf{J}} \right\rangle = -\overline{\left\langle \mathbf{J}\widehat{\mathbf{v}}, \mu \left(\frac{\widehat{\mathbf{u}} \cdot \widehat{\mathbf{n}}_b}{\widehat{\mathbf{z}}_b \cdot \widehat{\mathbf{n}}_b} \right) \frac{\mathbf{J}\widehat{\mathbf{z}}_b}{\det \mathbf{J}} \right\rangle}^{1} + R_u^A + \overline{R_u}^{\alpha}, \tag{10}$$

$$R_u^A \equiv -\left\langle \mathbf{J}\widehat{\mathbf{v}}, \mathscr{I}\left(\widehat{\mathbf{u}}^p, \overline{\widehat{\mathbf{u}}}^{1/2} \right) \right\rangle, \tag{11}$$

$$\begin{aligned} \overline{R_u}^{\alpha} \equiv & -\overline{\left\langle \mathbf{J}\widehat{\mathbf{v}}, 2\boldsymbol{\Omega} \times \frac{\mathbf{J}\widehat{\mathbf{u}}}{\det \mathbf{J}} \right\rangle}^{\alpha} + \overline{\left\langle \widehat{\nabla} \cdot \widehat{\mathbf{v}}, \widehat{\Phi} \right\rangle}^{\alpha} \\ & -\overline{\left\langle\!\left\langle [\![c_p \widehat{\theta} \widehat{\mathbf{v}}]\!], \left\{ \widehat{\Pi} \right\} \right\rangle\!\right\rangle}^{\alpha} + \overline{\left\langle c_p \widehat{\theta} \widehat{\nabla}_{\mathrm{C}} \cdot \widehat{\mathbf{v}} + \widehat{\mathbf{v}} \cdot \widehat{\nabla}_{\mathrm{C}} \left(c_p \widehat{\theta} \right), \widehat{\Pi} \right\rangle}^{\alpha}, \end{aligned} \tag{12}$$

where $\mathbf{z_b}$ and $\mathbf{n_b}$ are unit vectors parallel to gravity and normal to model layers respectively, see Figure 1 of [7] for details. The inner product over the domain D is

$$\langle f, g \rangle \equiv \int_{\mathrm{D}} f g dV, \tag{13}$$

and $\langle\langle \cdot \rangle\rangle$ denotes the surface integrals over the collection of all cell faces evaluated in the computational space and $[\![\cdot]\!]$ and $\{ \cdot \}$ indicate the jump in its argument and the value of its argument across a cell face, respectively (see Melvin et al. [7], section 4.4.1. for details).

The equation of state (8) is sampled at nodal points of the finite-element scheme rather than solved in its weak form as in Melvin et al. [7]. This is motivated by a desire for (8) to hold exactly as a diagnostic relationship between the Exner pressure, potential temperature and density.

6 | MESH MAPPINGS

The discretisation presented in the previous section is valid for a general three-dimensional hexahedral mesh and was applied to a uniform horizontally biperiodic domain in Melvin et al. [7]. Here spherical shell domains are considered. To complete the discretisation the mappings from the computational space to the physcial space introduced in Section 5 need to be specified.

The computational space consists of a single unit cell with coordinates $\widehat{\boldsymbol{\chi}} \in [0, 1]^3$ and a Cartesian (identity) metric tensor. Mappings $\boldsymbol{\chi} = \boldsymbol{\phi}(\widehat{\boldsymbol{\chi}})$ are introduced to map this computational cell to each cell in the physical mesh.

FIGURE 1 *C*12 cubed-sphere horizontal mesh with 6*x*12*x*12 cells.

The physical mesh coordinates are parametrised as a finite-element field $\boldsymbol{\chi} \in \mathbb{W}_\chi$ as in Melvin et al. [7] with a piecewise polynomial representation of degree m and have a Cartesian (identity) metric tensor. However, representing a spherical manifold with a piecewise polynomial introduces discretisation errors that depend upon the degree m and in order to accurately represent the surface of the sphere a high degree $\mathbb{W}_\chi$ space is required (Kent et al. [8]). To avoid this, an alternative approach used here is to map via an intermediate spherical coordinate system $\boldsymbol{\xi}$ such that the Jacobian is given by

$$\mathbf{J} \equiv \frac{\partial \boldsymbol{\phi}(\widehat{\boldsymbol{\chi}})}{\partial \widehat{\boldsymbol{\chi}}} \equiv \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \widehat{\boldsymbol{\chi}}}, \tag{14}$$

where $\boldsymbol{\xi}$ can be chosen such that the transformation of a cell from computational space to physical space, $\partial\boldsymbol{\xi}/\partial\widehat{\boldsymbol{\chi}}$, can be accurately represented by $\boldsymbol{\xi} \in \mathbb{W}_\chi$ with a low degree $\mathbb{W}_\chi$ space. The coordinate transformation from $\boldsymbol{\xi}$ to $\boldsymbol{\chi}$, $\partial\boldsymbol{\chi}/\partial\boldsymbol{\xi}$ is chosen such that it has a known analytic form that captures the spherical nature of the manifold. The form of $\mathbf{J}$ used in Melvin et al. [7] is recovered by setting $\boldsymbol{\xi} = \boldsymbol{\chi}$ in which case $\partial\boldsymbol{\chi}/\partial\boldsymbol{\xi}$ becomes the identity matrix.

The horizontal mesh used here is an equi-angular cubed-sphere (Ronchi et al. [18]), Figure 1. Vertically the mesh is extruded as described in Adams et al. [19]. The mesh resolution is denoted as $CnL\ell$ where n is the number of cells along one edge of a panel and the ℓ is the number of vertical layers such that there are $6n^2$ model columns and $6n^2\ell$ cells in the three-dimensional mesh.

A geocentric Cartesian coordinate system $\boldsymbol{\chi} \equiv (X, Y, Z)$ is used where $(X, Y, Z) = 0$ is the centre of the sphere of radius $a = \sqrt{X^2 + Y^2 + Z^2}$. Alongside the Cartesian coordinates $\boldsymbol{\chi}$ a spherical coordinate system $\boldsymbol{\xi} \equiv (\xi, \eta, r)$ is used with angular variables $[\xi, \eta] \in [-\pi/4, \pi/4]$ on each panel such that lines of constant ξ and η are angularly equidistant great circles on each panel and r is the radial distance from the centre of the sphere. As an example, with this choice of coordinates and mesh the components of the Jacobian $\mathbf{J} \equiv \frac{\partial\boldsymbol{\chi}}{\partial\boldsymbol{\xi}} \frac{\partial\boldsymbol{\xi}}{\partial\widehat{\boldsymbol{\chi}}}$ in each cell c with spacing $\Delta\xi_c$ and $\Delta\eta_c$ in the ξ and η direction respectively and with a constant slope in height above the surface of the sphere of $\delta_\xi \equiv \Delta r_c/\Delta\xi_c$ in

the $\xi-$direction and $\delta_\eta \equiv \Delta r_c/\Delta\eta_c$ in the $\eta-$direction are

$$\frac{\partial \boldsymbol{\xi}}{\partial \widehat{\boldsymbol{\chi}}} = \begin{pmatrix} \Delta\xi_c & 0 & 0 \\ 0 & \Delta\eta_c & 0 \\ \Delta\xi_c \tan\delta_\xi & \Delta\eta_c \tan\delta_\eta & \Delta r_c \end{pmatrix}. \tag{15}$$

In practice, to maintain flexibility of the scheme and to facilitate the inclusion of arbitrary orography, this component of the Jacobian is computed numerically with $\boldsymbol{\xi} \in \mathbb{W}_\chi$.

The second component of the Jacobian, $\partial\boldsymbol{\chi}/\partial\boldsymbol{\xi}$, transforms the spherical coordinate $\boldsymbol{\xi}$ into the Cartesian coordinate $\boldsymbol{\chi}$. Following Nair et al. [20] the basis vectors for a panel of the Equiangular cubed-sphere are

$$\mathbf{e}_\xi = (\frac{r}{\varrho^3})(1+t_\xi^2)\left[-t_\xi,\ (1+t_\eta^2),\ -t_\xi t_\eta\right], \tag{16}$$

$$\mathbf{e}_\eta = (\frac{r}{\varrho^3})(1+t_\eta^2)\left[-t_\eta,\ -t_\xi t_\eta,\ (1+t_\xi^2)\right], \tag{17}$$

$$\mathbf{e}_r = \frac{1}{\varrho}\left[1,\ t_\xi,\ t_\eta\right], \tag{18}$$

where

$$t_\xi = \tan(\xi), \quad t_\eta = \tan(\eta), \quad \varrho = \sqrt{1+t_\xi^2+t_\eta^2}. \tag{19}$$

The second component of the Jacobian mapping is then

$$\frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{\xi}} = R_i\left[\mathbf{e}_\xi^T,\ \mathbf{e}_\eta^T,\ \mathbf{e}_r^T\right], \tag{20}$$

where R_i is a rotation matrix for panel $i = 1,\ldots,6$ of the cubed-sphere that will translate and rotate $\left[\mathbf{e}_\xi^T,\ \mathbf{e}_\eta^T,\ \mathbf{e}_r^T\right]$ such that the union of all 6 panels form a spherical shell. The rotation matrix for each panel is given by

$$\begin{aligned} R_1 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad R_2 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad R_3 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \\ R_4 &= \begin{pmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad R_5 = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad R_6 = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{pmatrix}. \end{aligned} \tag{21}$$

7 | FINITE-VOLUME TRANSPORT DISCRETISATION

The transport scheme is an extension to the method-of-lines scheme used by Melvin et al. [7]. Solving the transport equation for an intermediate estimate s^p of wave dynamics terms of a scalar field s with a prescribed wind field $\mathbf{u}$ over a timestep either in advective form gives

$$s^{n+1} = s^p - \Delta t\,\mathcal{A}\left(s^p, \mathbf{u}\right), \tag{22}$$

or in flux form gives

$$s^{n+1} = s^p - \Delta t \nabla \cdot \mathcal{F}\left(s^p, \mathbf{u}\right). \tag{23}$$

The transport scheme is chosen such that it maintains a number of desirable properties:

1. The flexibility to alter the order of accuracy independently of the accuracy chosen for the finite-element, wave dynamics part of the model;
2. At least second order temporal accuracy;
3. Small dispersive errors;
4. Scale selective damping;
5. Stability for large CFL ($\mathbf{u}\Delta t/\Delta x > 1$) flows in any coordinate direction;
6. Flux-form variables should return increments that are linear in the divergence;
7. Computationally efficient.

Achieving the first property rules out using the native finite-element discretisation when using degree $l = 0$ spaces, where the transport scheme would be at best first-order, and instead motivates the use of a finite-volume method where the spatial accuracy can be linked to the polynomial reconstruction. The desired temporal accuracy is achieved through using an explicit multi-stage Runge-Kutta integration scheme for the transport terms in the same manner as Melvin et al. [7]. The third and fourth properties can be achieved through using an upwind, even-degree polynomial reconstruction of the field used in the advective $\mathcal{A}$ and flux $\mathcal{F}$ terms of (22) and (23) respectively. To achieve the fifth property the explicit Runge-Kutta scheme can be substepped within the transport scheme such that the effective CFL number for each substep is within the stability envelope of the desired Runge-Kutta scheme. The sixth property is motivated by a desire to maintain a constant density in non-divergent flows. To achieve this it is sufficient that the increment of a flux-form variable is linear in the divergence of the transporting wind field, here this is achieved by using an advective-then-flux scheme, Section 7.3.

As a step towards obtaining the seventh property the transport scheme is temporally split between the horizontal and vertical directions using a 2nd order Strang splitting (Strang [21]). Splitting can deliver significant computational performance benefits, by allowing different schemes or options to be used for the different directions. Horizontal transport requires data communication costs when using multi-processor computers, while vertical transport typically involves higher Courant numbers (due to the grid anisotropy). Separating the two allows these problems to be addressed separately.

The horizontal spatial reconstruction follows that described in Kent et al. [8] and the vertical reconstruction is extended from that used in Melvin et al. [7] as described in Section 7.4. The scheme defined in this section is applied in computational space, except for the spatial reconstructions which are computed in physical space. The finite-volume transport scheme used here is designed for a mesh with a single scalar degree of freedom per cell entity (i.e. in the cell centre or centre of a face). To couple the finite-element wave dynamics and the finite-volume transport, scalars need to be mapped from the finite-element spaces to the finite-volume space. The mapping from a degree $l = 0$ finite-element space to the finite-volume scheme is the identity operator while mapping from an degree $l > 0$ space requires a projection into the finite-volume space and is not considered here.

7.1 | Transported State

As in Kent et al. [8] the transport acts on an intermediate state $(\mathbf{u}^p, \rho^p, \theta^p)$ (termed predictors in Kent et al. [8], their Section 4.3) for the prognostic variables $(\mathbf{u}, \rho, \theta)$ that consists of an explicit half timestep estimate of the wave dynamics terms only. The use of this estimate (instead of just taking the start of timestep fields) allows the present method to mimic the stability properties of a semi-implicit semi-Lagrangian scheme while using an Eulerian average advecting velocity $\overline{\mathbf{u}}^{1/2}$ rather than the Lagrangian average advecting velocity used in a standard semi-implicit semi-Lagrangian scheme. A full analysis and discussion of reasons behind this choice is in preparation and will be given elsewhere. The fields to be transported are given by

$$\mathbf{u}^p \equiv [\mathbf{u} + (1-\alpha)\,\Delta t \mathbf{S}]^n, \tag{24}$$

$$\rho^p \equiv [\rho - (1-\alpha)\,\Delta t \rho \nabla \cdot \mathbf{u}]^n, \tag{25}$$

$$\theta^p \equiv \theta^n. \tag{26}$$

As in Kent et al. [8] the choice of these terms is motivated by capturing the explicit parts of the non-advective processes e.g. ρ^p contains the $\rho\nabla\cdot\mathbf{u}$ component of $\nabla\cdot(\rho\mathbf{u})$ but not the advective $\mathbf{u}\cdot\nabla\rho$ component.

7.2 | Temporal Splitting: Advective Form

The transport of a field s by a three-dimensional velocity field $\mathbf{u}$ is split into vertical and horizontal components using a second-order Strang splitting as follows. Noting that following McRae et al. [22] (their table 3, where the $\mathbb{W}_2$ space used here is referred to as NCF_r) the velocity space $\mathbb{W}_2 \equiv \mathbb{W}_2^h \oplus \mathbb{W}_2^v$ can be written as the composition of a space $\mathbb{W}_2^h$ of vectors in the horizontal direction and $\mathbb{W}_2^v$ of vectors in the vertical direction, (see Maynard et al. [14], their figure 1, for a illustrative example). The wind is split into horizontal $\mathbf{u}_H \in \mathbb{W}_2^h$ and vertical $\mathbf{u}_V \in \mathbb{W}_2^v$ components $(\mathbf{u} \equiv \mathbf{u}_H \oplus \mathbf{u}_V)$. The transport equation $\partial s/\partial t + \mathcal{A}(s, \mathbf{u}) = 0$ is then discretised across the timestep as

$$s_V = s^p - \frac{\Delta t}{2}\mathcal{A}_V\left(s^p, \mathbf{u}_V\right), \tag{27}$$

$$s_{HV} = s_V - \Delta t \mathcal{A}_H\left(s_V, \mathbf{u}_H\right), \tag{28}$$

$$s^{n+1} \equiv s_{VHV} = s_{HV} - \frac{\Delta t}{2}\mathcal{A}_H\left(s_{HV}, \mathbf{u}_V\right), \tag{29}$$

with the operator $\mathcal{A} \equiv \mathcal{A}_H \oplus \mathcal{A}_V$ split into its horizontal and vertical components. Each split step is then discretised using a multistage Runge-Kutta scheme as in Melvin et al. [7], Kent et al. [8], giving

$$s^{(i)} = s^p - \Delta t \sum_{j=1}^{i-1} a_{i,j}\mathcal{A}_D\left(s^{(j)}, \mathbf{u}_D\right), \quad i = 1, \ldots, m, \tag{30}$$

$$s^{n+1} = s^p - \Delta t \sum_{k=1}^{m} b_k \mathcal{A}_D\left(s^{(k)}, \mathbf{u}_D\right), \tag{31}$$

where D is the direction (either V or H), The coefficients $a_{i,j}$ and b_k are given by the Butcher tableau (Butcher [23]) associated with the chosen scheme. (30)-(31) are then further substepped to ensure stablility for large CFL numbers. The specific Runge–Kutta scheme used here, as in Melvin et al. [7] is the third-order, three-stage, strong stability preserving Runge–Kutta scheme [24].

7.3 | Temporal Splitting: Flux Form

It is tempting to formulate the flux form by replacing the $\mathcal{A}$'s in (27)-(29) with $\nabla \cdot \mathcal{F}$'s. However, in the presence of non-divergent flow this does not have the desirable property of preserving a constant (except for the trivial case where the components of the divergence in each direction are zero) as s^{n+1} is not linear in $\nabla \cdot \mathbf{u}$. Simply replacing only the $\mathcal{A}$ in (29) by $\nabla \cdot \mathcal{F}$ does not conserve the total scalar being transported. However, the following split form is both conservative and preserves a constant (Bendall et al. [25]).

Revisiting the Strang splitting, (27)-(29) are modified as follows

$$s_V = s^p - \frac{\Delta t}{2}\mathcal{A}_V\left(s^p, \mathbf{u}_V\right), \tag{32}$$

$$s_{HV} = s_V - \Delta t \mathcal{A}_H\left(s_V, \mathbf{u}_H\right), \tag{33}$$

$$s^{n+1} \equiv s_{VHV} = s^p - \frac{\Delta t}{2}\nabla_V \cdot \mathcal{F}_V\left(s^p, \mathbf{u}_V\right) - \Delta t \nabla_H \cdot \mathcal{F}_H\left(s_V, \mathbf{u}_H\right) - \frac{\Delta t}{2}\nabla_V \cdot \mathcal{F}_V\left(s_{HV}, \mathbf{u}_V\right), \tag{34}$$

with $\nabla\cdot \equiv \nabla_H \cdot \oplus \nabla_V \cdot$. When s^p is constant the final step is seen to reduce to $s^{n+1} = s^p\left(1 - \Delta t \nabla \cdot \mathbf{u}\right)$ which preserves s^p when $\nabla \cdot \mathbf{u} = 0$.

At any stage of the split step, where a flux-form equation is solved, such as (34), the advective-then-flux form of a Runge-Kutta scheme can be obtained from an $m-$stage scheme for the advective update (30)-(31) by setting

$$s^* = \sum_{k=1}^{m} b_k s^{(k)} \tag{35}$$

$$s^{n+1} = s^p - \Delta t \nabla_D \cdot \boldsymbol{\mathcal{F}}_{\boldsymbol{D}}\left(s^*, \mathbf{u}_D\right), \tag{36}$$

and again when s^p is constant the final step reduces to $s^{n+1} = s^p\left(1 - \Delta t \nabla \cdot \mathbf{u}\right)$.

7.4 | Spatial reconstruction of a scalar field

The transport scheme computes a high order reconstruction $\check{s}$ of a given scalar field s in physical space $\boldsymbol{\chi}$. For a scalar value in cell j the reconstructed field is computed at points staggered half a grid length $j \pm \Delta_i/2, \quad i = 1, 2, 3$ from the original field in all three directions, where Δ_i is defined to be the grid spacing in the $\widehat{\chi_i}$ direction. For example, for a field in $\mathbb{W}_3$ such as the density ρ_j, which is located at cell centres, then the reconstructed field $\check{\rho}_{j\pm\Delta_i/2}$ is computed at the centre of each cell face $j \pm \Delta_i/2$. The reconstruction is computed by fitting a polynomial through a number of cells and evaluating this polynomial at the staggered points. The reconstruction is given an upwind bias, determined by the wind direction ($\mathbf{u}_j$ for $\mathcal{A}$ and $\mathbf{u}_{j\pm\delta_i}$ for $\mathcal{F}$), by using even order polynomials for the reconstruction.

The horizontal spatial reconstruction is the same as Kent et al. [8] and is based on that used in Thuburn et al. [26] and the interested reader is referred to Baldauf [27] and Skamarock and Menchaca [28] for other results on these types of schemes. In brief, a two-dimensional polynomial in local Cartesian coordinates is fitted in a least squares sense to a region of cells around the reconstruction point. This polynomial is then evaluated at the reconstruction point to give the reconstructed field. This method results in a scheme that is accurate across discontinuities in the mesh (such as at panel boundaries) and reduces grid imprinting from the transport scheme.

The vertical reconstruction follows the same method as the horizontal scheme, except now a one-dimensional polynomial is used and the local coordinate system z can be taken to be aligned to the radial distance from the surface of the sphere and the origin of the local coordinate system is at the reconstruction point.

Near the top and bottom boundaries when there are not enough points to construct an upwind degree n polyno-

mial then the stencil is shifted so that the same polynomial is used as the first point where there are enough points for an upwind polynomial. The result is that the polynomial is no longer upwinded but the desired degree is maintained.

7.5 | Flux computation

The flux $\mathcal{F}$ is computed as in Kent et al. [8] by a pointwise multiplication of the computational wind field $\widehat{\mathbf{u}}$ sampled at the centre of each face $k = j \pm \delta_i$ by the reconstructed scalar $\check{s}$

$$\mathcal{F} \equiv \widehat{\mathbf{u}}\left(\widehat{\boldsymbol{\chi}}_k\right) \check{s}. \tag{37}$$

The finite-volume divergence operator for a flux is then given by

$$\nabla \cdot \mathcal{F} = \frac{1}{\det \mathbf{J}} \sum_{k=1,\ldots,6} \mathcal{F}\left(\widehat{\boldsymbol{\chi}}_k\right) \cdot \widehat{\mathbf{n}}_j, \tag{38}$$

where k is the index of each face of the cell j.

7.6 | Advective increment computation

To compute the advective tendency $\mathcal{A} \equiv \mathbf{u} \cdot \nabla s$ of a field s, given the reconstructed field $\check{s}$ located at points staggered half a grid length from s_j the advective update is

$$\widehat{\mathcal{A}}_j \equiv \frac{1}{\det \mathbf{J}} \widehat{\mathbf{u}}\left(\widehat{\boldsymbol{\chi}}_j\right) \cdot \widehat{\boldsymbol{\delta}} \check{s}_j, \tag{39}$$

where $\widehat{\mathbf{u}}\left(\widehat{\boldsymbol{\chi}}_j\right)$ is the computational velocity field sampled at $\boldsymbol{\chi}_j$ and $\widehat{\boldsymbol{\delta}}$ is the discrete gradient operator in the computational space. For example if $\check{s}$ are located on cell faces $\left[\check{s}_{j+\Delta_1/2}, \check{s}_{j-\Delta_1/2}, \check{s}_{j+\Delta_2/2}, \check{s}_{j-\Delta_2/2}, \check{s}_{j+\Delta_3/2}, \check{s}_{j-\Delta_3/2}\right]$ on the (East, West, North, South, Up, Down) sides of cell j respectively then

$$\delta \check{s}_j \equiv \left[\left(\check{s}_{j+\Delta_1/2} - \check{s}_{j-\Delta_1/2}\right), \left(\check{s}_{j+\Delta_2/2} - \check{s}_{j+\Delta_2/2}\right), \left(\check{s}_{j+\Delta_3/2} - \check{s}_{j-\Delta_3/2}\right)\right]. \tag{40}$$

Optionally monotonicity can be enforced on the advective update $\widehat{\mathcal{A}}$ through a simple clipping scheme. On the final stage of the Runge-Kutta scheme the update $\widehat{\mathcal{A}}$ is modified to ensure that $s^{min} \leq s^{n+1} \leq s^{max}$ where s^{min} and s^{max} are the minimum and maximum values of s^p used in the stencil to compute $\widehat{\mathcal{A}}$ respectively.

7.7 | Advection of vector fields

To compute the advective increment $\widehat{\mathcal{A}}\left(\widehat{\nu}, \widehat{\overline{\mathbf{u}}}^{1/2}\right)$ of a vector field $\widehat{\nu}$ the advected field is first transformed into physical space using the $\mathbb{W}_2$ transform

$$\dot{\boldsymbol{\chi}} \equiv \frac{\partial \boldsymbol{\chi}}{\partial t} = \frac{\mathbf{J}\widehat{\nu}}{\det \mathbf{J}}, \tag{41}$$

where each component of $\dot{\boldsymbol{\chi}}$ is placed in the $\mathbb{W}_3$ space $\dot{\chi}_i \in \mathbb{W}_3$ $i = 1, 2, 3$. Since $\boldsymbol{\chi}$ is a Cartesian coordinate system with basis vectors $(\mathbf{e_x}, \mathbf{e_y}, \mathbf{e_z})$ this means that the Cartesian components of the velocity vector are transported, avoiding

the computation of gradients of any basis vectors. Each component of the vector is transported as a scalar in the advective form by solving

$$\dot{\chi}_i^{n+1} = \dot{\chi}_i^p - \widehat{\mathcal{A}}\left(\dot{\chi}_i, \overline{\widehat{\mathbf{u}}}^{1/2}\right), \tag{42}$$

with $\widehat{\mathcal{A}}$ given by (39) using the method described in the previous sections. Once all three components have been advected the advective increment $\widehat{\mathcal{A}}$ is

$$\widehat{\mathcal{A}} \equiv \left[\widehat{\mathcal{A}}\left(\dot{\chi}_1, \overline{\widehat{\mathbf{u}}}^{1/2}\right), \widehat{\mathcal{A}}\left(\dot{\chi}_2, \overline{\widehat{\mathbf{u}}}^{1/2}\right), \widehat{\mathcal{A}}\left(\dot{\chi}_3, \overline{\widehat{\mathbf{u}}}^{1/2}\right)\right], \tag{43}$$

and is mapped back to the $\mathbb{W}_2$ space through (11).

8 | TIMESTEPPING

The timestepping algorithm closely follows that of Melvin et al. [7] but, inspired by the algorithm of Wood et al. [13], the Newton loop is split into (n_o) outer loops and (n_i) inner loops, such that the advection scheme is called n_o times in the outer loop, whilst updates to the residuals and the linear solver are called $n_o \times n_i$ times in the inner loop.

8.1 | Linear System

The solution procedure follows Melvin et al. [7] except that the linear system is modified to include the Coriolis terms. To recap, increments to the state variable $\mathbf{x}' \equiv \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ at iteration k are sought from the linear system

$$\mathcal{L}\left(\mathbf{x}^*\right)\mathbf{x}' = -\mathcal{R}\left(\mathbf{x}^{(k)}\right), \quad \mathbf{x} \equiv (\mathbf{u}, \rho, \theta, \Pi)^T. \tag{44}$$

The linear operator $\mathcal{L}$ is inspired by the linearisation of the set of residuals $\mathcal{R}$ (see section 8.3 below) about some reference state $\mathbf{x}^* \equiv (\mathbf{0}, \rho^*, \theta^*, \Pi^*)^T$ with relaxation factors $\tau_{u,\rho,\theta}$ to obtain $\mathcal{L}(\mathbf{x}^*)$. In spatially continuous form $\mathcal{L}$ from Melvin et al. [7] (their (42)) is augmented by the Coriolis terms to give

$$\mathcal{L}\left(\mathbf{x}^*\right)\mathbf{x}' = \begin{pmatrix} (1 + 2\tau_u \Delta t \boldsymbol{\Omega}\times)\,\mathbf{u}' - \mu\left(\frac{\mathbf{n}_b \cdot \mathbf{u}'}{\mathbf{n}_b \cdot \mathbf{z}_b}\right)\mathbf{z}_b + \tau_u \Delta t c_p\left(\theta' \mathbf{z}_b\left[\mathbf{n}_b \cdot \nabla\Pi^*\right] + \theta^* \nabla\Pi'\right), \\ \rho' + \tau_\rho \Delta t \nabla \cdot (\rho^* \mathbf{u}'), \\ \theta' + \tau_\theta \Delta t \mathbf{u}' \cdot \mathbf{z}_b\left(\mathbf{n}_b \cdot \nabla\theta^*\right), \\ \frac{1-\kappa}{\kappa}\frac{\Pi'}{\Pi^*} - \frac{\rho'}{\rho^*} - \frac{\theta'}{\theta^*}. \end{pmatrix} \tag{45}$$

Note, following Wood et al. [13] but in contrast to Melvin et al. [7], only the vertical component of the implicit buoyancy terms $(\theta' \nabla\Pi^*,\ \mathbf{u}' \cdot \nabla\theta^*)$ are retained in the linear system which improves the efficiency of the solver without detracting from the convergence properties.

8.2 | Inner Loop Convergence

The implicit terms in the continuity and thermodynamic equations contain corrections to the transport terms, $\nabla \cdot (\rho^* \mathbf{u}')$ and $\mathbf{u}' \cdot \nabla\theta^*$ respectively, where the velocity increment $\mathbf{u}' \equiv \mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}$ is defined relative to the latest (inner

loop) estimate for the velocity field $\mathbf{u}^{(k)}$. Since the transport terms are only updated in the outer loop the advecting wind is $\bar{\mathbf{u}}^{1/2} = 1/2\left(\mathbf{u}^{(o)} + \mathbf{u}^{n}\right)$ where $\mathbf{u}^{(o)}$ is the latest estimate of $\mathbf{u}^{n+1}$ available in the outer loop. If the inner loop iterate $n_i > 1$ then the two estimates for $\mathbf{u}^{n+1}$ do not agree $\left(\mathbf{u}^{(o)} \neq \mathbf{u}^{(k)}\right)$ and this can lead to an inconsistency in the discretisation. Taking the thermodynamic equation $\theta' + \tau_\theta \Delta t \mathbf{u}' \cdot \nabla\theta^* = R_\theta$ with $\tau_\theta = 1/2$ and $\theta^* = \theta^n$ this becomes

$$\begin{aligned} \theta^{(k+1)} \quad - \quad & \theta^{(k)} + \frac{\Delta t}{2}\left(\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\right) \cdot \nabla\theta^n \\ = \quad & R_\theta \equiv -\left[\theta^{(k)} - \theta^n + \frac{\Delta t}{2}\left(\mathbf{u}^{(o)} + \mathbf{u}^{n}\right) \cdot \nabla\theta^n\right], \end{aligned} \tag{46}$$

which can be rearranged to give

$$\begin{aligned} \theta^{(k+1)} - \theta^n \quad + \quad & \frac{\Delta t}{2}\left(\mathbf{u}^{(k+1)} + \mathbf{u}^{n}\right) \cdot \nabla\theta^n \\ = \quad & \frac{\Delta t}{2}\left(\mathbf{u}^{(k)} - \mathbf{u}^{(o)}\right) \cdot \nabla\theta^n, \end{aligned} \tag{47}$$

where the left hand side is the desired temporal discretisation of the equation (centred implicit). However the term on the right hand side only vanishes if $\mathbf{u}^{(k)} = \mathbf{u}^{(o)}$. The solution to this inconsistency is that in the inner loop when $n_i > 1$ the residuals of the thermodynamic (R_θ) and continuity (R_ρ) equations are set to zero. Returning to the above example (46) is replaced by

$$\theta^{(k+1)} - \theta^{(k)} + \frac{\Delta t}{2}\left(\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\right) \cdot \nabla\theta^n = 0, \tag{48}$$

and for k such that $\mathbf{u}^{(k-1)} = \mathbf{u}^{(o)}$ then replacing $\theta^{(k)}$ with (46) with all indices reduced by 1 yields the desired form

$$\theta^{(k+1)} - \theta^n + \frac{\Delta t}{2}\left(\mathbf{u}^{(k+1)} + \mathbf{u}^{n}\right) \cdot \nabla\theta^n = 0. \tag{49}$$

The same result then applies for larger k upon repeated use of (49). This inner loop correction to the residuals is not applied to the momentum equation or equation of state since in this case the linear corrections in (45) correspond to terms that are updated in the inner loop, i.e. there are no linear transport terms, and so there is no inconsistency.

8.3 | Discrete Linear System

Applying the mixed finite-element discretisation of Section 5 to (45) results in:

$$M_2^{\mu,C}\widetilde{u}' - P_{2v\theta}^{\Pi^*}\widetilde{\theta}' - G^{\theta^*}\widetilde{\Pi}' = -\mathcal{R}_{\mathbf{u}}, \tag{50}$$

$$M_3\widetilde{\rho}' + D\left(\widetilde{\rho}^*\widetilde{u}'\right) = -\mathcal{R}_\rho, \tag{51}$$

$$M_\theta\widetilde{\theta}' + P_{\theta 2v}^{\theta^*}\widetilde{u}' = -\mathcal{R}_\theta, \tag{52}$$

$$E^{\Pi^*}\widetilde{\Pi}' - E^{\rho^*}\widetilde{\rho}' - E^{\theta^*}\widetilde{\theta}' = -\mathcal{R}_\Pi, \tag{53}$$

with

$$\mathcal{R}_{\mathbf{u}} = \Delta t \left(M_2 \delta_t \widetilde{u} + M_\mu \overline{\widetilde{u}}^1 - \overline{R_u}^\alpha - R_u^A \right), \tag{54}$$

$$\mathcal{R}_\rho = \Delta t \left(M_3 \delta_t \widetilde{\rho} - R_\rho^F \right), \tag{55}$$

$$\mathcal{R}_\theta = \Delta t \left(M_\theta \delta_t \widetilde{\theta} - R_\theta^A \right), \tag{56}$$

$$\mathcal{R}_\Pi \equiv \left(1 - \frac{p_0 \Pi^{\frac{1-\kappa}{\kappa}}}{R \rho \theta} \right), \tag{57}$$

where $P_{2\nu\theta}^{\Pi^*}$ and $P_{\theta 2\nu}^{\theta^*}$ are the vertically restricted versions of the $P_{2\theta}^{\Pi^*}$ and $P_{\theta 2}^{\theta^*}$ operators given in Melvin et al. [7] (their (84) and (87)). $M_2^{\mu,C}$ is the operator formed by combining the $\mathbb{W}_2$ mass matrix with the operators arising from the Rayleigh damping and Coriolis terms:

$$M_2^{\mu,C} \equiv M_2 + \Delta t M_\mu + \tau_u \Delta t M_C, \tag{58}$$

with

$$(M_C)_{ij} \equiv \left\langle \mathbf{J}\widehat{\mathbf{v}}_i, 2\boldsymbol{\Omega} \times \frac{\mathbf{J}\widehat{\mathbf{v}}_j}{\det(\mathbf{J})} \right\rangle. \tag{59}$$

As, in contrast to Melvin et al. [7], the equation of state is now sampled the operators in (53) are given by

$$E^{\Pi^*} = \frac{1-\kappa}{\kappa} \left[\frac{p_0}{R} \frac{\left(\widehat{\Pi^*} \right)^{\frac{1-\kappa}{\kappa}}}{\widehat{\rho^*}\widehat{\theta^*}} \right] \frac{\widehat{\sigma}}{\widehat{\Pi^*}}, \tag{60}$$

$$E^{\rho^*} = \frac{\widehat{\sigma}}{\widehat{\rho^*}}, \tag{61}$$

$$E^{\theta^*} = \frac{\widehat{w}}{\widehat{\theta^*}}. \tag{62}$$

Each entry $E_{i,j}$ of an operator E is obtained by evaluating all variables at nodal points $\widehat{\chi_i}$ with basis function $\widehat{\sigma}_j$ or $\widehat{w}_j$. All other operators have the same form as given by Melvin et al. [7]. At convergence of the iterative procedure primed quantities vanish and $\mathcal{R}\left(\mathbf{x}^{(k)}\right) = 0$ is solved.

8.4 | Iterative Solver

The system of equations (50)-(53) is solved using the method presented in Maynard et al. [14]. This consists of an iterative Krylov method that is preconditioned by an approximate Schur complement of the equations for the pressure increment which is itself solved using a single v-cycle of a geometric multigrid method. The approximate Schur complement is achieved by a diagonal mass lumping of M_θ and M_2^μ (where the Coriolis terms have been dropped from the lumped approximation, equivalent to $\tau_u = 0$ in (58)).

8.5 | Mass Conservation

Mass conservation in the combined finite-element finite-volume discretisation is achieved by combining (51), (55) along with $\Delta t R_\rho^F = \tau_\rho{}^{-1} D\left(\mathcal{F}\left[\rho^p, \overline{\widehat{\mathbf{u}}}^{1/2}\right]\right)$ giving

$$M_3\widehat{\rho}^{(k+1)} + D\left(\widehat{\rho^*}\left[\widehat{\mathbf{u}}^{(k+1)} - \widehat{\mathbf{u}}^{(k)}\right]\right) = M_3\widehat{\rho}^n + \tau_\rho{}^{-1} D\left(\mathcal{F}\left[\rho^p, \overline{\widehat{\mathbf{u}}}^{1/2}\right]\right) \tag{63}$$

and summing over the entire domain D this reduces to $\sum_D M_3\left(\widehat{\rho}^{(k+1)} - \widehat{\rho}^n\right) = 0$ following definition of the discrete divergence operator from [7], their (82), giving $\sum_D D(\cdot) \equiv 0$.

9 | COMPUTATIONAL EXAMPLES

In order to assess the accuracy of the model it is run on a set of standard numerical tests for atmospheric dynamics drawn from the literature. These are used to ensure the model generates the correct response to forcing at different scales as well as maintaining the large scale balances important in the governing equations. Of particular concern with meshes, such as the cubed-sphere, that have discontinuities in their coordinate lines, is what impact those discontinuities have on the numerical solutions, referred to as grid imprinting.

The tests results presented in this section are:

9.1 Resting atmosphere over orography (Test 2.0.0 from the 2012 DCMIP project, Ullrich et al. [29])
9.2 Flow over a Gaussian mountain (Allen and Zerroukat [30])
9.3 Deep atmosphere baroclinic wave (Ullrich et al. [31])
9.4 Held-Suarez climate test (Held and Suarez [32])

Key test parameters for each of these examples are summarised in Table 1 where the average grid spacing has been taken to be the square root of the average cell area. For a Cn mesh this is given by

$$dC = \sqrt{\frac{4\pi a^2}{6n^2}}. \tag{64}$$

While in principle the finite-element methodology affords flexibility in the polynomial degree, as in Melvin et al. [7] the focus will again be on results in the lowest-degree $l = 0$ case. Additionally a number of simplifications and specifications to the formulation given in previous sections are made:

- The coordinate space is $\mathbb{W}_\chi = Q_1^{DG}$;
- The angular resolution of the cubed-sphere mesh is kept constant: $\Delta\xi_c = \Delta\eta_c = \frac{\pi}{2n}, \quad \forall c$;
- The semi-implicit scheme is centred in time: $\alpha = 1/2$;
- The relaxation parameters are $\tau_u = 1/2$ and $\tau_{\rho,\theta} = 1$ which is empirically found to improve convergence, consistent with Wood et al. [13];
- 2 outer (advection) and 2 inner (nonlinear) iterations are used: $n_o = n_i = 2$;
- As in Melvin et al. [7] the reference profiles $\mathbf{x}^*$ are taken to be the start of timestep fields $\mathbf{x}^* \equiv \mathbf{x}^n$ with no adjustment applied to these fields;

Test	Section	Resolution $CnLm$ (n cells per edge, m layers)	Approx grid spacing dC (km)	Model depth z_T (km)	Δt (s)
Resting atmosphere	9.1	C96L30	96.0	12	600
Gaussian mountain	9.2	C96L40	96.0	32	900
Baroclinic wave	9.3	C96L30	96.0	30	900
Baroclinic wave: grid imprinting	9.3.1	C448L30	20.6	30	225
		C896L30	10.3	30	225
Held-Suarez	9.4	C48L30	192.1	30	1800

TABLE 1 Model parameters for each test.

- A quadratic reconstruction of scalar fields is used in the advection scheme to compute fluxes $\widehat{\mathcal{F}}$ and advective updates $\widehat{\mathcal{A}}$;
- The damping layer is not required for any tests considered here and so $\mu \equiv 0$;
- The vertical mesh consists of n levels and the height of each level k is

$$z_k = z_T \epsilon_k + z_B \left(1 - \epsilon_k\right) \tag{65}$$

where $z = r - a$ is the height above the surface of the sphere and z_B is height of the domain surface above a ($z_B = 0$ or as defined by any orographic profile). The non-dimensional parameter ϵ is given by

$$\epsilon_k = (k/n)\,, \tag{66}$$

for a uniform vertical mesh or

$$\epsilon_k = \frac{\sqrt{\gamma\,(k/n)^2 + 1} - 1}{\sqrt{\gamma + 1} - 1}, \tag{67}$$

with $\gamma = 15$ for a quadratic stretching.

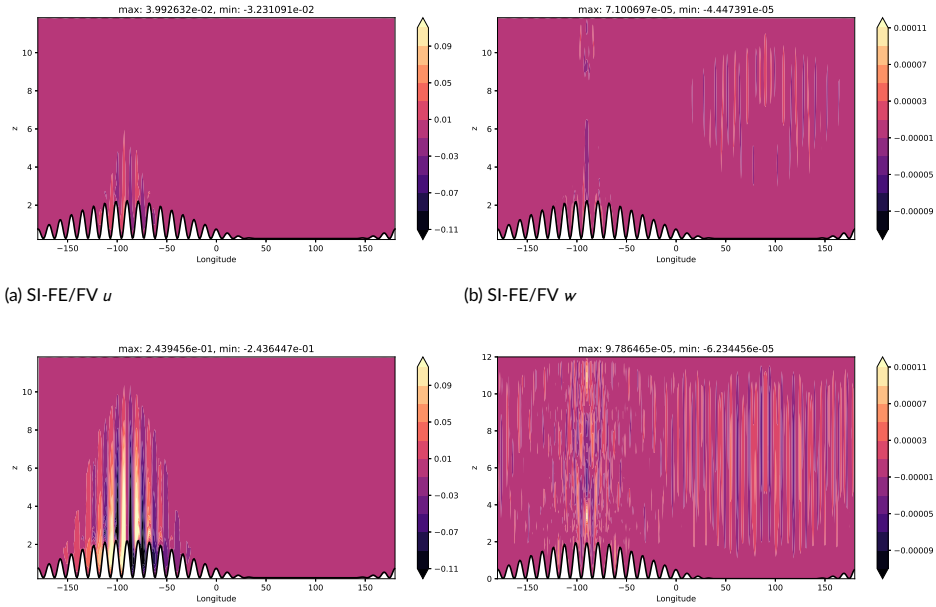
- All initial conditions are computed by sampling the field at degree of freedom locations and where required the density or initial pressure are obtained from the equation of state. Additionally, no discrete balance is applied to the initial conditions.

For the spherical domain used in these examples the coordinate $\boldsymbol{\chi}$ is replaced by the standard geocentric Cartesian coordinates $\mathbf{X}$ so that $(\chi_1, \chi_2, \chi_3) \equiv (X, Y, Z)$. Additionally the results are linearly interpolated into a regular lat-lon grid for presentation. In this section w is used to denote the vertical component of the velocity $\mathbf{u}$ in the radial direction (i.e. $w = Dr/Dt \neq D\chi_3/Dt$) and u is the zonal component of the velocity $\mathbf{u}$ (i.e $u = r\cos\phi D\lambda/Dt \neq D\chi_1/Dt$). Initial conditions are given in spherical coordinates with latitude $\phi \in (-\pi/2, \pi/2)$ and longitude $\lambda \in (-\pi, \pi)$.

9.1 | Resting atmosphere over orography

Orography is represented in the model formulation through the Piola transforms and in particular through the Jacobian $\mathbf{J}$. In the presence of orography the mapping $\partial\boldsymbol{\xi}/\partial\widehat{\boldsymbol{\chi}}$ (15) introduces a coupling between the horizontal components

of the velocity vector into the vertical component of the momentum equation (through the $\tan\delta_\xi$ and $\tan\delta_\eta$ terms in (15)). This is in contrast to most models (such as Wood et al. [13]) where the presence of orography and terrain following coordinates introduces a coupling of the vertical components of the pressure gradient term into the horizontal components of the momentum equation (for example (22) & (23) of Wood et al. [13]). It is therefore interesting to see how this different formulation can represent a balanced state over orography. To complement the orographic tests already presented in Melvin et al. [7] test 2.0.0 from the DCMIP2012 project Ullrich et al. [29] is used to simulate a resting atmosphere over large scale orography.



(a) SI-FE/FV u (b) SI-FE/FV w

(c) SISL u (d) SISL w

FIGURE 2 Zonal (left panels) and vertical (right panels) wind fields for the resting atmosphere case after 6 days on a C96L30 mesh with $\Delta t = 600$ s. The top row shows results from this paper and the bottom row shows results from the semi-implicit semi-Lagrangian ENDGame model with a 1 degree resolution and $\Delta t = 600$ s, (Wood et al. [13]).

The test is based upon earlier ideas by Lin [33]. The orographic profile is given by

$$z_B = \begin{cases} \frac{h_0}{2}\left[1+\cos\left(\frac{\pi r_m}{R_m}\right)\right]\cos^2\left(\frac{\pi r_m}{\zeta_m}\right), & \text{if } r_m < R_m, \\ 0, & \text{otherwise.} \end{cases} \tag{68}$$

The mountain height is $h_0 = 2000m$ and $R_m = 3\pi/4$, $\zeta_m = \pi/16$. The great circle distance from the mountain centre-point $(\lambda_m, \phi_m) = (3\pi/2, 0)$ is

$$r_m = \arccos\left[\sin\phi_m \sin\phi + \cos\phi_m \cos_\phi \cos\left(\lambda - \lambda_m\right)\right]. \tag{69}$$

The atmosphere is initialised at rest ($\mathbf{u} = 0$) and is non-rotating ($\boldsymbol{\Omega} = 0$). A constant lapse rate $\Gamma = 0.0065 K m^{-1}$ is used, giving the initial temperature as

$$T = T_0 - \Gamma (r - a), \tag{70}$$

with $T_0 = 300K$. This example tests the accuracy of the pressure gradient terms over orography. Since the initial state is in balance no motion should be generated, but, due to inaccuracies in the pressure gradient terms in terrain following coordinates, this balance will not be discretely maintained and motion will be generated. The size of the motion generated is related to the error in the pressure gradient terms. Figure 2 shows the zonal and vertical wind fields along the equator after 6 days simulation on a C96L30 mesh with a uniform vertical resolution and model top at $z_T = 12km$. This is compared to the semi-implicit semi-Lagrangian model of Wood et al. [13] run on a 1 degree Latitude-Longitude mesh with the same vertical mesh. Over the mountain a small amount of motion is generated, as shown in both the zonal and vertical velocities (Figure 2, top row). Compared to Wood et al. [13] (Figure 2, bottom row) the zonal velocity perturbations are approximately an order of magnitude smaller while the vertical velocity perturbations are of the same order but less widespread, particularly over the orography. Taken together this indicates the model is able to maintain balance over orography relatively well and there is no spuriously large growth of perturbations.

9.2 | Flow over a Gaussian hill

This test simulates the generation of Rossby waves from flow over orography and is based upon the test of Tomita and Satoh [34] and Jablonowski et al. [35] and further developed by Allen and Zerroukat [30]. The set up used here follows Allen and Zerroukat [30] and the mountain profile is given by

$$z_B = h_0 \exp\left[-\left(\frac{a}{\zeta_m} r_m\right)^2\right], \tag{71}$$

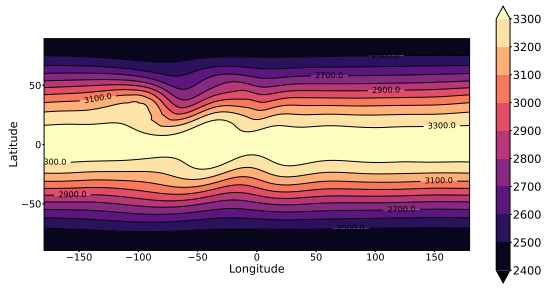
with $h_0 = 2000m$, $\zeta_m = 1500km$ and r_m given by (69) with the mountain centre at $(\lambda_m, \phi_m) = (-\pi/2, \pi/6)$. The initial wind is given by

$$u (r \cos\phi) = u_0 \frac{r}{a} \cos\phi, \tag{72}$$

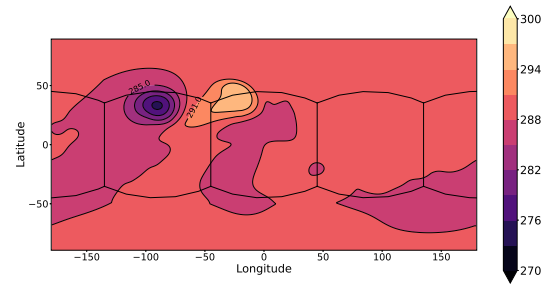
with $u_0 = 20ms^{-1}$ which is (5.7) from Allen and Zerroukat [30] with $\beta = 0$. The atmosphere is isothermal with $T = 288K$ and the surface pressure is given by (5.15) in Allen and Zerroukat [30]:

$$p_s = p_p \exp\left[(2\Omega a + u_0)\frac{u_0}{2RT}\cos^2\phi\right] \exp\left(-\frac{z_s g}{RT}\right), \tag{73}$$

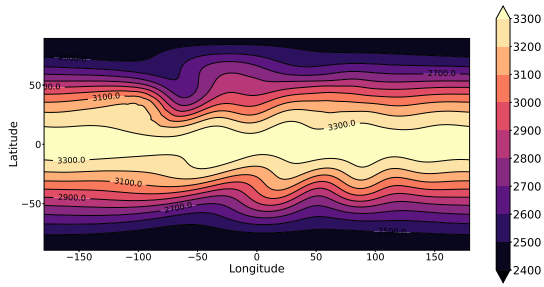
with $p_p = 930hPa$. The model is run at C96L40 resolution with a uniform vertical mesh and model top at $z_T = 32km$. The $700hPa$ geopotential height and temperature are shown at days 5, 10 and 15 of the simulation in Figure 3. These results are similar to both the Yin-Yang and ENDGame semi-implicit semi-Lagrangian results presented in Allen and Zerroukat [30] (their figures 8 and 9). The locations of the cubed-sphere panel boundaries are overlaid on the temperature figures and although the mountain is located across a panel edge there are no obvious indications of grid imprinting from the cubed-sphere on the solution profiles.



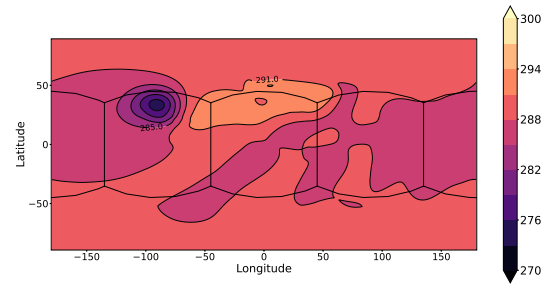
(a) Day 5 $z700$



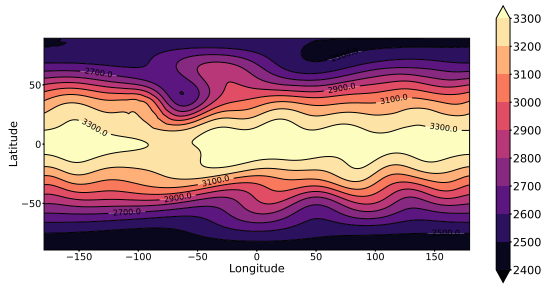
(b) Day 5 $T700$



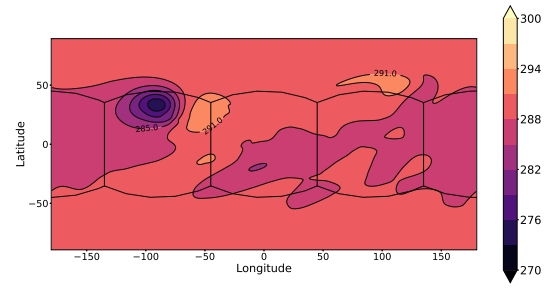
(c) Day 10 $z700$



(d) Day 10 $T700$



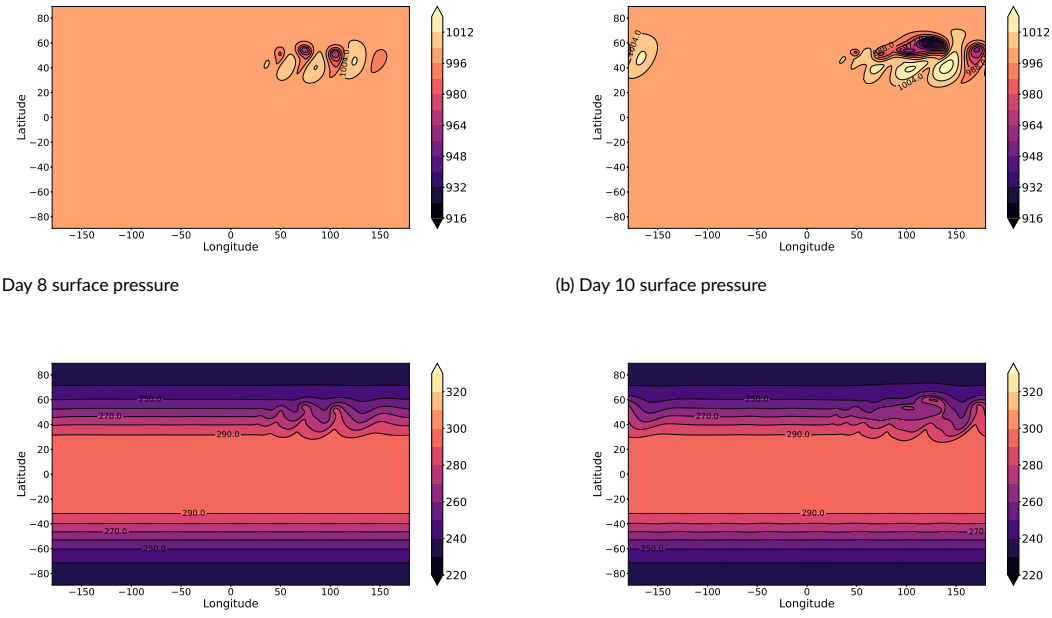
(e) Day 15 $z700$



(f) Day 15 $T700$

FIGURE 3 $700hPa$ geopotential height (left column) and temperature (right column) at days 5, 10 and 15 for the flow over a Gaussian hill test at C96L40 resolution with $\Delta t = 900s$. The location of the cubed-sphere panel boundaries are overlaid on the $T700$ plots.

9.3 | Baroclinic wave



(a) Day 8 surface pressure

(b) Day 10 surface pressure

(c) Day 8 $T850$

(d) Day 10 $T850$

FIGURE 4 Surface pressure (top row) and 850 hPa temperature (bottom row) for the Baroclinic wave test on a C96L30 mesh with $\Delta t = 900$ s. Left panels: after 8 days simulation and Right panels: after 10 days simulation.

The baroclinic wave test of Ullrich et al. [31] simulates the formation of a series of features typical of mid-latitude weather systems. The test is run for 15 days at C96L30 resolution with a quadratic vertical mesh and the same timestep $\Delta t = 900s$ as used in Wood et al. [13] with a monotonic filter applied to the transport of θ as described in Section 7.6. The surface pressure and $850hPa$ temperature at days 8 and 10 are shown in Figure 4. These compare well with the results shown in both Ullrich et al. [31] and Wood et al. [13]. Importantly there are no obvious signs of grid imprinting in the southern hemisphere in either the pressure or temperature fields. The minimum surface pressure throughout the simulation compared to Wood et al. [13] is shown in Figure 5 and both models show excellent agreement through the first 10 days of simulation before some divergence in model solutions over the last 5 days of simulation.

9.3.1 | Grid Imprinting

Following Ji [36] to further investigate the grid imprinting in the model the baroclinic wave test is repeated at high horizontal resolution (C448 and C896) for a single simulated day. The vertical velocity (with zonal mean removed) around a corner of the cubed-sphere in the southern hemisphere (away from the initial perturbation) is shown in Figure 6 and can be compared with Figures 2.6 & 2.7 of Ji [36]. At both resolutions there is some spurious motion

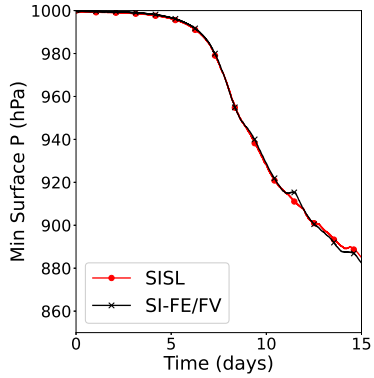
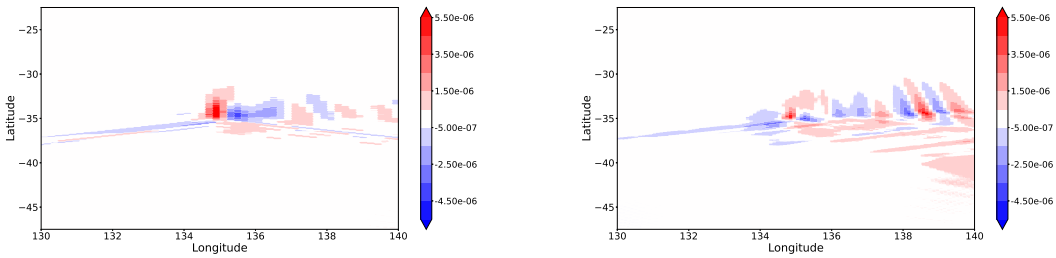


FIGURE 5 Minimum surface pressure for the Baroclinic wave test on a C96L30 mesh with $\Delta t = 900$ s compared with the SISL model of Wood et al. [13] on a 1 degree mesh with the same timestep.



(a) C448 (b) C896

FIGURE 6 Vertical velocity at a corner of the cubed-sphere after 1 day of simulation on a (left) C448L30 mesh with $\Delta t = 450$ and (right) C896L30 mesh with $\Delta t = 225$.

around the corner of the cubed-sphere, however the values are very small, of the order 10^{-3}mm/s, and smaller than both the MPAS and FV3 models considered in Ji [36], so this level of grid imprinting is considered acceptable. These results support the choice to use a lowest degree compatible finite-element method which simplifies a number of aspects of the model design (such as coupling to existing subgrid parametrisation schemes, Brown et al. [37]).

9.4 | Held-Suarez

The Held-Suarez idealised climate test (Held and Suarez [32]) simulates the evolution of an atmospheric state with relaxation towards a prescribed surface temperature and wind profile. The model is run for 1200 days and time-averaged fields (after the first 200 day spinup period) are shown in Figure 7. The initial state is taken to be the baroclinic wave initial state from Section 9.3. The time-average (sampled every day) of the zonal velocity field on a C48L30 mesh using a quadratic stretching in the vertical mesh is shown in Figure 7. The left panel shows the zonally averaged zonal velocity field and the right panel shows the zonal velocity on level 14 (approximately though the centre of the jets). These profiles are again similar to those produced by ENDGame (Tort et al. [38]). The horizontal cross section shows no obvious sign of grid imprinting from the cubed-sphere grid, indicating that even for long timescale runs there is no systematic error from the treatment of the cubed-sphere panel corners and edges.

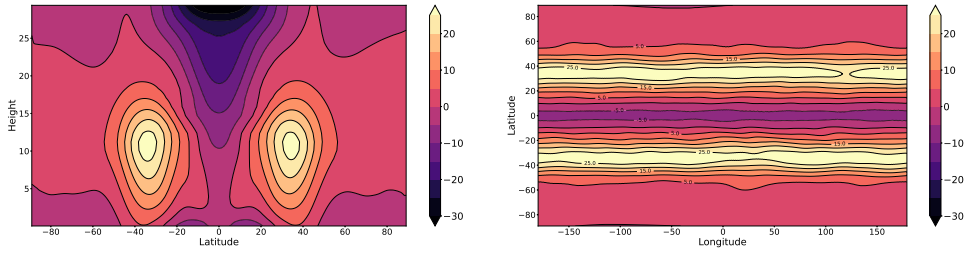


FIGURE 7 Zonally averaged zonal velocity (left) and zonal velocity on level 14 ($\approx 10.5km$) (right) fields. All fields are averaged over the last 1000 days of a 1200 day Held-Suarez run using a C48L30 mesh with $\Delta t = 1800$ s.

10 | SUMMARY

The mixed finite-element, finite-volume, semi-implicit model of Melvin et al. [7] has been extended to spherical geometry for atmospheric modelling. The finite-volume transport scheme has been extended to encompass the momentum advection terms and has been adapted to the non-uniform, non-orthogonal horizontal mesh by using polynomial reconstructions as in Kent et al. [8]. In order to maintain a constant field, an advective-then-flux formulation for conservatively transported variables is used (Bendall et al. [25]). The finite-element method presented in Melvin et al. [7] has required minimal modification for spherical geometries, highlighting the flexibility of this approach, and the only significant change here is the use of a semi-analytic mapping to the sphere to accurately represent a spherical shell domain. The temporal discretisation mimics that of Wood et al. [13] and is coupled to the multigrid solver of Maynard et al. [14] to give an efficient method of solving the linear semi-implicit system.

The semi-implicit finite-element finite-volume dynamical core has been applied to a number of standard dynamical tests. The model has been shown to produce results comparable to those in the literature and in particular to the existing semi-implicit semi-Lagrangian ENDGame dynamical core (Wood et al. [13]) used in operational models at the

Met Office. These results demonstrate that using $l = 0$ elements provides sufficient accuracy on a cubed-sphere mesh and so this formulation is planned to be used in future applications of this model. We note that the model can be run for simple dynamical core tests at higher $l > 0$ finite element order. The main missing block towards running more complex tests is the functionality to map the high order finite element spaces to low order spaces on a high resolution mesh in order to couple to the finite volume transport scheme and the physical parametrisations. This model has also recently been coupled to idealised physical parametrisations and chemical processes by Brown et al. [37] and to the simulation of Exoplanet atmospheres, Sergeev et al. [39], again producing results comparable to those in the literature.

Continual improvements are being made to optimise the computational performance of the model. There does not appear to be any fundamental barrier to achieving comparable throughput to the present operational model.

As a next step towards using this model for numerical weather and climate prediction the dynamical core has been extended to handle moist dynamic processes and coupled to the existing suite of subgrid physical parametrisations used by Walters et al. [40]. The formulation and code have been extended to limited area domains with forced boundaries and both of these developments will be reported upon in future work. In order to couple with a data assimilation system a tangent linear version of this model has also been developed and again will be reported upon in future work.

Data Availability

The data used to support the findings of this study were generated using the Met Office's LFRic-Atmosphere model. The LFRic-Atmosphere source code and configuration files are freely available from the Met Office Science Repository Service (https://code.metoffice.gov.uk) upon registration and completion of a software licence.

Acknowledgments

The authors would like to express their gratitude to the many people who have worked and contributed to the development of the LFRic atmosphere model, PSyclone and the GungHo dynamical core, without whose efforts this manuscript would not have been possible. The authors would also like to thank two anonymous reviewers for their useful comments on an earlier version of this manuscript.

The work of John Thuburn was supported by NERC under the GungHo Phase 2 programme, grant NE/K006762/1. The work of Colin Cotter was supported by NERC under the grant NE/K006789/1 and by EPSRC by the grants EP/L016613/1 & EP/R029423/1.

References

[1] Staniforth A, Thuburn J. Horizontal grids for global weather and climate prediction models: a review. Quarterly Journal of the Royal Meteorological Society 2012;138(662):1–26. https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.958.

[2] Lawrence BN, Rezny M, Budich R, Bauer P, Behrens J, Carter M, et al. Crossing the chasm: how to develop weather and climate models for next generation computers? Geoscientific Model Development 2018;11(5):1799–1821. https://gmd.copernicus.org/articles/11/1799/2018/.

[3] Cotter CJ, Shipton J. Mixed finite elements for numerical weather prediction. Journal of Computational Physics 2012;231(21):7076–7091. https://www.sciencedirect.com/science/article/pii/S0021999112002628.

[4] Cotter CJ, Thuburn J. A finite element exterior calculus framework for the rotating shallow-water equations. Journal of Computational Physics 2014;257:1506–1526. https://www.sciencedirect.com/science/article/pii/S0021999113006761, physics-compatible numerical methods.

[5] Thuburn J, Cotter CJ. A primal–dual mimetic finite element scheme for the rotating shallow water equations on polygonal spherical meshes. Journal of Computational Physics 2015;290:274–297. https://www.sciencedirect.com/science/article/pii/S0021999115001151.

[6] Natale A, Shipton J, Cotter CJ. Compatible finite element spaces for geophysical fluid dynamics. Dynamics and Statistics of the Climate System 2016;1(1).

[7] Melvin T, Benacchio T, Shipway B, Wood N, Thuburn J, Cotter C. A mixed finite-element, finite-volume, semi-implicit discretization for atmospheric dynamics: Cartesian geometry. Quarterly Journal of the Royal Meteorological Society 2019;145(724):2835–2853. https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3501.

[8] Kent J, Melvin T, Wimmer GA. A mixed finite-element discretisation of the shallow-water equations. Geoscientific Model Development 2023;16(4):1265–1276. https://gmd.copernicus.org/articles/16/1265/2023/.

[9] Cotter CJ. Compatible finite element methods for geophysical fluid dynamics. Acta Numerica 2023;32:291–393.

[10] Arakawa A, Lamb VR. Computational Design of the Basic Dynamical Processes of the UCLA General Circulation Model. In: CHANG J, editor. General Circulation Models of the Atmosphere, vol. 17 of Methods in Computational Physics: Advances in Research and Applications Elsevier; 1977.p. 173–265. https://www.sciencedirect.com/science/article/pii/B9780124608177500094.

[11] Melvin T, Benacchio T, Thuburn J, Cotter C. Choice of function spaces for thermodynamic variables in mixed finite-element methods. Quarterly Journal of the Royal Meteorological Society 2018;144(712):900–916. https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3268.

[12] Thuburn J, Woollings TJ. Vertical discretizations for compressible Euler equation atmospheric models giving optimal representation of normal modes. Journal of Computational Physics 2005;203(2):386–404. https://www.sciencedirect.com/science/article/pii/S0021999104003316.

[13] Wood N, Staniforth A, White A, Allen T, Diamantakis M, Gross M, et al. An inherently mass-conserving semi-implicit semi-Lagrangian discretization of the deep-atmosphere global non-hydrostatic equations. Quarterly Journal of the Royal Meteorological Society 2014;140(682):1505–1520.

[14] Maynard C, Melvin T, Müller EH. Multigrid preconditioners for the mixed finite element dynamical core of the LFRic atmospheric model. Quarterly Journal of the Royal Meteorological Society 2020;146(733):3917–3936. https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3880.

[15] Mullen P, McKenzie A, Pavlov D, Durant L, Tong Y, Kanso E, et al. Discrete Lie Advection of Differential Forms;11(2):131–149. https://doi.org/10.1007/s10208-010-9076-y.

[16] Rognes ME, Kirby RC, Logg A. Efficient Assembly of Hdiv and Hcurl Conforming Finite Elements. SIAM Journal on Scientific Computing 2010;31(6):4130–4151. https://doi.org/10.1137/08073901X.

[17] Bochev PB, Ridzal D. Rehabilitation of the lowest-order Raviart-Thomas element on quadrilateral grids. SIAM Journal on Numerical Analysis 2008;47(1):487–507. http://www.jstor.org/stable/25663133.

[18] Ronchi C, Iacono R, Paolucci PS. The "Cubed Sphere": A New Method for the Solution of Partial Differential Equations in Spherical Geometry. Journal of Computational Physics 1996;124(1):93–114. https://www.sciencedirect.com/science/article/pii/S0021999196900479.

[19] Adams SV, Ford RW, Hambley M, Hobson JM, Kavčič I, Maynard CM, et al. LFRic: Meeting the challenges of scalability and performance portability in Weather and Climate models. Journal of Parallel and Distributed Computing 2019;http://www.sciencedirect.com/science/article/pii/S0743731518305306.

[20] Nair RD, Thomas SJ, Loft RD. A Discontinuous Galerkin Transport Scheme on the Cubed Sphere. Monthly Weather Review 2005;133(4):814 – 828. https://journals.ametsoc.org/view/journals/mwre/133/4/mwr2890.1.xml.

[21] Strang G. On the Construction and Comparison of Difference Schemes. SIAM Journal on Numerical Analysis 1968;5(3):506–517. https://doi.org/10.1137/0705041.

[22] McRae ATT, Bercea GT, Mitchell L, Ham DA, Cotter CJ. Automated Generation and Symbolic Manipulation of Tensor Product Finite Elements. SIAM Journal on Scientific Computing 2016;38(5):S25–S47. https://doi.org/10.1137/15M1021167.

[23] Butcher JC. The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods. Wiley; 1987. https://books.google.co.uk/books?id=SjXvAAAAMAAJ.

[24] Gottlieb S. On High Order Strong Stability Preserving Runge–Kutta and Multi Step Time Discretizations;25(1):105–128. https://doi.org/10.1007/s10915-004-4635-5.

[25] Bendall TM, Wood N, Thuburn J, Cotter CJ. A solution to the trilemma of the moist Charney–Phillips staggering. Quarterly Journal of the Royal Meteorological Society 2023;149(750):262–276. https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.4406.

[26] Thuburn J, Cotter CJ, Dubos T. A mimetic, semi-implicit, forward-in-time, finite volume shallow water model: comparison of hexagonal–icosahedral and cubed-sphere grids. Geoscientific Model Development 2014;7(3):909–929. https://gmd.copernicus.org/articles/7/909/2014/.

[27] Baldauf M. Stability analysis for linear discretisations of the advection equation with Runge–Kutta time integration. Journal of Computational Physics 2008;227(13):6638–6659. https://www.sciencedirect.com/science/article/pii/S0021999108001769.

[28] Skamarock WC, Menchaca M. Conservative Transport Schemes for Spherical Geodesic Grids: High-Order Reconstructions for Forward-in-Time Schemes. Monthly Weather Review 2010;138(12):4497 – 4508. https://journals.ametsoc.org/view/journals/mwre/138/12/2010mwr3390.1.xml.

[29] Ullrich PA, Christiane Jablonowski C, Kent J, Lauritzen PH, Nair RD, Taylor MA, DCMIP2012 test case document; 2012. Accessed: 10-12-2021. http://www-personal.umich.edu/~cjablono/DCMIP-2012_TestCaseDocument_v1.7.pdf.

[30] Allen T, Zerroukat M. A deep non-hydrostatic compressible atmospheric model on a Yin-Yang grid. Journal of Computational Physics 2016;319:44–60. https://www.sciencedirect.com/science/article/pii/S0021999116301590.

[31] Ullrich PA, Melvin T, Jablonowski C, Staniforth A. A proposed baroclinic wave test case for deep- and shallow-atmosphere dynamical cores. Quarterly Journal of the Royal Meteorological Society 2014;140(682):1590–1602.

[32] Held IM, Suarez MJ. A Proposal for the Intercomparison of the Dynamical Cores of Atmospheric General Circulation Models. Bulletin of the American Meteorological Society 1994 10;75(10):1825–1830.

[33] Lin SJ. A finite-volume integration method for computing pressure gradient force in general vertical coordinates. Quarterly Journal of the Royal Meteorological Society 1997;123(542):1749–1762.

[34] Tomita H, Satoh M. A new dynamical framework of nonhydrostatic global model using the icosahedral grid. Fluid Dynamics Research 2004;34(6):357.

[35] Jablonowski C, Lauritzen P, Nair R, Taylor M. Idealized test cases for the dynamical cores of Atmospheric General Circulation Models: A proposal for the NCAR ASP 2008 summer colloquium. Manuscript: May/29/2008, NCAR Technical Report and Journal Paper 2008;https://public.websites.umich.edu/~cjablono/dycore_test_suite.html.

[36] Ji M. Dynamical Core Evaluation Test Report for NOAA's Next Generation Global Prediction System (NGGPS); 2016. https://www.weather.gov/media/sti/nggps/NGGPS%20Dycore%20Phase%202%20Test%20Report%20website.pdf.

[37] Brown A, Bendall TM, Boutle I, Melvin T, Shipway B, Physics-Dynamics-Chemistry Coupling Across Different Meshes in LFRic-Atmosphere: Formulation and Idealised Tests; 2023. https://arxiv.org/abs/2310.01255.

[38] Tort M, Dubos T, Melvin T. Energy-conserving finite-difference schemes for quasi-hydrostatic equations. Quarterly Journal of the Royal Meteorological Society 2015;141(693):3056–3075. https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.2590.

[39] Sergeev DE, Mayne NJ, Bendall T, Boutle IA, Brown A, Kavčič I, et al. Simulations of idealised 3D atmospheric flows on terrestrial planets using LFRic-Atmosphere. Geoscientific Model Development 2023;16(19):5601–5626. https://gmd.copernicus.org/articles/16/5601/2023/.

[40] Walters D, Baran AJ, Boutle I, Brooks M, Earnshaw P, Edwards J, et al. The Met Office Unified Model Global Atmosphere 7.0/7.1 and JULES Global Land 7.0 configurations. Geoscientific Model Development 2019;12(5):1909–1963. https://gmd.copernicus.org/articles/12/1909/2019/.