RESEARCH ARTICLE

A compatible finite-element discretisation for the moist compressible Euler equations

Thomas M. Bendall[1,2] | Thomas H. Gibson[3] | Jemma Shipton[4] | Colin J. Cotter[2] |
Ben Shipway[1]

[1]Dynamics Research, Met Office, Exeter, UK

[2]Mathematics Department, Imperial College London, UK

[3]Department of Applied Mathematics, Naval Postgraduate School, Monterey, California

[4]Mathematics Department, University of Exeter, Exeter, UK

Correspondence
Thomas Bendall, Met Office, FitzRoy Road, Exeter EX1 3PB, UK.

A promising development of the last decade in the numerical modelling of geophysical fluids has been the compatible finite-element framework. Indeed, this will form the basis for the next-generation dynamical core of the Met Office. For this framework to be useful for numerical weather prediction models, it must be able to handle descriptions of unresolved and diabatic processes. These processes offer a challenging test for any numerical discretisation, and have not yet been described within the compatible finite-element framework. The main contribution of this article is to extend a discretisation using this new framework to include moist thermodynamics. Our results demonstrate that discretisations within the compatible finite-element framework can be robust enough also to describe moist atmospheric processes. We describe our discretisation strategy, including treatment of moist processes, and present two configurations of the model using different sets of function spaces with different degrees of finite element. The performance of the model is demonstrated through several test cases. Two of these test cases are new cloudy-atmosphere variants of existing test cases: inertia–gravity waves in a two-dimensional vertical slice and a three-dimensional rising thermal.

KEYWORDS

dynamical core, finite-element method, mimetic discretisation, moisture

1 | INTRODUCTION AND MOTIVATION

In the dynamical cores of global numerical weather prediction models, latitude–longitude grids have long been popular due to the orthogonality of the meridians and circles of latitude. This orthogonality can be exploited to gain a number of desirable numerical properties, as discussed in Staniforth and Thuburn (2012). However, as these grids are refined, the grid points around the poles converge. On the massively parallel computers that are expected to dominate the future of numerical weather prediction, this leads to a bottleneck in the data communication around the poles. This compromises the scalability of the model:

as more computational resources become available, there will be diminishing returns to the resolution at which the model can be run within a given wall-clock time. There has therefore been a search for alternative grids that will avoid the pole problem and deliver better scaling, while still providing the desirable numerical properties described in Staniforth and Thuburn (2012).

Compatible finite-element methods offer a promising solution to this problem. These are finite-element methods in which the variables lie in different function spaces, so that the discrete equations replicate the vector calculus identities of the continuous equations, such as $\nabla \times \nabla f = \mathbf{0}$ and $\nabla \cdot \nabla \times \mathbf{f} = 0$. These methods can facilitate the use of nonorthogonal meshes, for instance those that are quasiuniform over the sphere, so the scalability bottleneck can be avoided. Cotter and Shipton (2012) showed, for the linear rotating shallow-water equations, that a compatible finite-element discretisation still maintains many of the properties that are argued for by Staniforth and Thuburn (2012). More recent work using compatible finite-element methods for geophysical fluids includes Cotter and Thuburn (2014), Natale *et al.*, (2016), and Yamazaki *et al.*, (2017). The Met Office is therefore developing a dynamical core using such a discretisation; see Melvin *et al.*, (2019) for recent developments.

As of yet, the framework has not been combined with "physics" (descriptions of unresolved and diabatic physics processes). This is a vital step for such discretisations to be used in weather prediction or climate modelling. However, these processes can be challenging for the discretisation.

The main result of this article is the first extension of this framework to include moist thermodynamics, so introducing a basic form of "physics". We present the details of discretisations for two sets of compatible spaces, with elements of different approximation orders. Then we demonstrate these discretisations through a series of test cases featuring moisture.

The structure of this article is as follows. First we present the form of the continuous equations that we aim to solve in Section 2. Then Section 3 describes the spatial and temporal discretisation of the dynamical part of the model, presenting the function spaces for the variables and the strategy for solving the equations of motion. Section 4 details the specifics of the moisture parametrisations, including a discussion of how to combine fields from different function spaces within a parametrisation. We then demonstrate its use in a number of test cases in Section 5.

Finally, the model presented in this article is written as part of Gusto, a library for dynamical cores using compatible finite element discretisations. Gusto itself is based on the Firedrake software of Rathgeber *et al.*, (2017), the development of which is based at Imperial College London. This software provides automated code generation for finite-element methods. Firedrake also has the functionality for tensor product element and extruded mesh functionality, as described in McRae *et al.*, (2016), Bercea *et al.*, (2016), and Homolya and Ham (2016), which we used substantially.

2 | GOVERNING EQUATIONS

We solve the compressible Euler equations featuring three species of moisture: water vapour, cloud water, and rain. The ice phase is neglected. Motivated by the Met Office's most recent dynamical core, ENDGame, described in Walters *et al.*, (2014), our prognostic variables are the density of dry air ρ_d and the dry virtual potential temperature θ_vd, alongside the wind velocity $\mathbf{v}$ and the mixing ratios r_v, r_c, and r_r. Here the subscripts respectively denote water vapour, cloud water, and rain, whilst the mixing ratio r_i is the ratio of the density by volume of the ith substance to that of dry air, that is, $r_i := \rho_i / \rho_\mathrm{d}$. The total mixing ratio of water species is $r_\mathrm{t} := r_\mathrm{v} + r_\mathrm{c} + r_\mathrm{r}$. The dry potential temperature θ_vd is defined for r_v, temperature T, and air pressure p by

$$\theta_\mathrm{vd} := T \left(\frac{p_\mathrm{R}}{p} \right)^{R_\mathrm{d}/c_\mathrm{pd}} (1 + r_\mathrm{v}/\epsilon) = \theta_\mathrm{d}(1 + r_\mathrm{v}/\epsilon), \quad (1)$$

where c_pd is the specific heat capacity of dry air, p_R is a reference pressure, θ_d is the dry potential temperature, and $\epsilon := R_\mathrm{d}/R_\mathrm{v}$ is the ratio of specific gas constants of dry air and water vapour. The choice of θ_vd is motivated in Walters *et al.*, (2014), which notes that it is the more natural choice of variable to complement ρ_d, and claims that it may be smoother than the dry potential temperature θ_d.

The full equation set that we use is

$$\frac{D\mathbf{v}}{Dt} + \mathbf{f} \times \mathbf{v} + \frac{c_\mathrm{pd}\theta_\mathrm{vd}}{1 + r_\mathrm{t}} \nabla \Pi + \nabla \Phi = \mathbf{0}, \quad (2a)$$

$$\frac{D\theta_\mathrm{vd}}{Dt} + \theta_\mathrm{vd} \left(\frac{R_\mathrm{m}}{c_\mathrm{vml}} - \frac{R_\mathrm{d} c_\mathrm{pml}}{c_\mathrm{pd} c_\mathrm{vml}} \right) \nabla \cdot \mathbf{v}$$
$$= -\theta_\mathrm{vd} \left[\frac{c_\mathrm{vd} L_\mathrm{v}(T)}{c_\mathrm{vml} c_\mathrm{pd} T} - \frac{R_\mathrm{v}}{c_\mathrm{vml}} \left(1 - \frac{R_\mathrm{d} c_\mathrm{vml}}{R_\mathrm{m} c_\mathrm{pd}} \right) - \frac{R_\mathrm{v}}{R_\mathrm{m}} \right] \frac{Dr_\mathrm{v}}{Dt}, \quad (2b)$$

$$\frac{D\rho_\mathrm{d}}{Dt} + \rho_\mathrm{d} \nabla \cdot \mathbf{v} = 0, \quad (2c)$$

$$\frac{Dr_\mathrm{v}}{Dt} = \dot{r}^\mathrm{r}_\mathrm{evap} - \dot{r}^\mathrm{c}_\mathrm{cond}, \quad (2d)$$

$$\frac{Dr_c}{Dt} = \dot{r}_{cond}^c - \dot{r}_{accr} - \dot{r}_{accu}, \tag{2e}$$

$$\frac{Dr_r}{Dt} = \dot{r}_{accr} - \dot{r}_{accu} - \dot{r}_{evap}^r - S. \tag{2f}$$

Here, Π is the Exner pressure function and Φ represents the geopotential, whilst $\boldsymbol{f} = f\hat{\boldsymbol{k}}$ represents the Coriolis parameter multiplied by the vertical unit vector. The specific heat capacities c_{pd}, c_{vd}, c_{pml}, and c_{vml}, are (respectively) for dry air at constant pressure, dry air at constant volume, moist air at constant pressure, and moist air at constant volume. For these, as well as the specific gas constant R_m and also the latent heat of vaporization $L_v(T)$, we follow closely the values used in Bryan and Fritsch (2002), which are also listed in the Appendix. The advective derivative is given by

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla}. \tag{3}$$

The equation of state can be written as

$$\Pi = \left(\frac{p}{p_R}\right)^\kappa \equiv \left(\frac{\rho_d R_d \theta_{vd}}{p_R}\right)^{\kappa/(1-\kappa)}, \tag{4}$$

with $\kappa := R_d/c_{pd}$.

The terms on the left-hand sides of Equation 2 represent the dynamics, whilst the right-hand sides are considered to be the physics. The processes $\dot{r}_{cond}^c$, $\dot{r}_{accr}$, $\dot{r}_{evap}^r$, $\dot{r}_{accu}$, and S are the microphysics parametrisations and are described in Section 4.

The momentum equation can also be recast in vector invariant form:

$$\begin{aligned} &\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{\nabla} \times \boldsymbol{v}) \times \boldsymbol{v} + \boldsymbol{f} \times \boldsymbol{v} + \frac{1}{2}\boldsymbol{\nabla}v^2 \\ &+ \frac{c_{pd}\theta_{vd}}{1+r_t}\boldsymbol{\nabla}\Pi + \boldsymbol{\nabla}\Phi = \boldsymbol{0}. \end{aligned} \tag{5}$$

The final thing to note is the extra term proportional to $\boldsymbol{\nabla} \cdot \boldsymbol{v}$ appearing on the left-hand side of Equation 2b. This term is neglected in many models, but is mentioned in Thuburn (2017) and Bryan and Fritsch (2002) to be important in capturing convection fully, particularly in a saturated atmosphere. In our model, it appears in the forcing step of the dynamical core.

3 | DYNAMICS DISCRETISATION

3.1 | Function spaces

One of the main results of Cotter and Shipton (2012) was the suggestion of combinations of function spaces for the velocity $\boldsymbol{v}$ and the (dry) air density ρ_d within the compatible finite-element framework that will satisfy the properties described in Staniforth and Thuburn (2012). These spaces mimic the Arakawa C-grid staggering, which allowed for steady geostrophic modes, avoidance of spurious pressure modes, and more accurate representation of the dispersion relation for Rossby and inertia–gravity waves. Compatible sets of finite-element spaces for $\boldsymbol{v}$ and ρ_d will form part of a discrete de Rham complex, so that $\boldsymbol{\nabla} \cdot \boldsymbol{v}$ is in the same space as ρ_d. Some such families of spaces can be found in the periodic table of finite elements, described in Arnold and Logg (2014).

In this section, we will expand on this to list the finite-dimensional function spaces $(V_\rho, V_{\boldsymbol{v}}, V_\theta)$ that we use in our model for velocity, density, and potential temperature, respectively. We consider both two-dimensional vertical slices with quadrilateral cells and three-dimensional domains using hexahedral elements with quadrilateral faces. The stratification of geophysical fluids, along with their high aspect ratio, motivates using an extruded mesh in the vertical, meaning that the mesh has regular layers. The finite elements on such a mesh can be constructed as tensor product elements, as described in McRae *et al.*, (2016). We will present two configurations of the model, using two different sets of spaces from the same family of finite-element spaces. We label these two sets as the *lowest-order $k=0$ spaces* and *next-to-lowest-order $k=1$ spaces*. More explanation on these can be found in Bendall *et al.*, (2019). All of these spaces are illustrated in Figure 1.

The density ρ_d lies in a space that is discontinuous between elements. For $k=0$ spaces, the density is constant within each cell, while it is linear in each cell for $k=1$ spaces. These are the discontinuous Galerkin spaces, denoted by dQ_k.

The velocity $\boldsymbol{v}$ has continuous normal components between cells but discontinuous tangential components. The spaces that we use are the Raviart–Thomas spaces, RT_k, described in Raviart and Thomas (1977). For the $k=0$ configuration, this means that for each component the field is linear in the direction of that component and continuous between cells in that direction, but constant within the cell in other directions. This becomes continuous and quadratic in the direction of the component for the $k=1$ spaces, but discontinuous and linear in other directions.

To replicate the Charney–Philips grid, the potential temperature θ_{vd} is co-located with the vertical component of velocity. This is the tensor product element of the discontinuous Galerkin element dQ_k over the horizontal part of the domain with the continuous Galerkin element Q_{k+1} on the vertical part of the domain. We therefore denote this space by $dQ_k \otimes Q_{k+1}$. Moisture variables also lie in this space, a choice that facilitates the latent heat transfer associated with changes of phase of water.
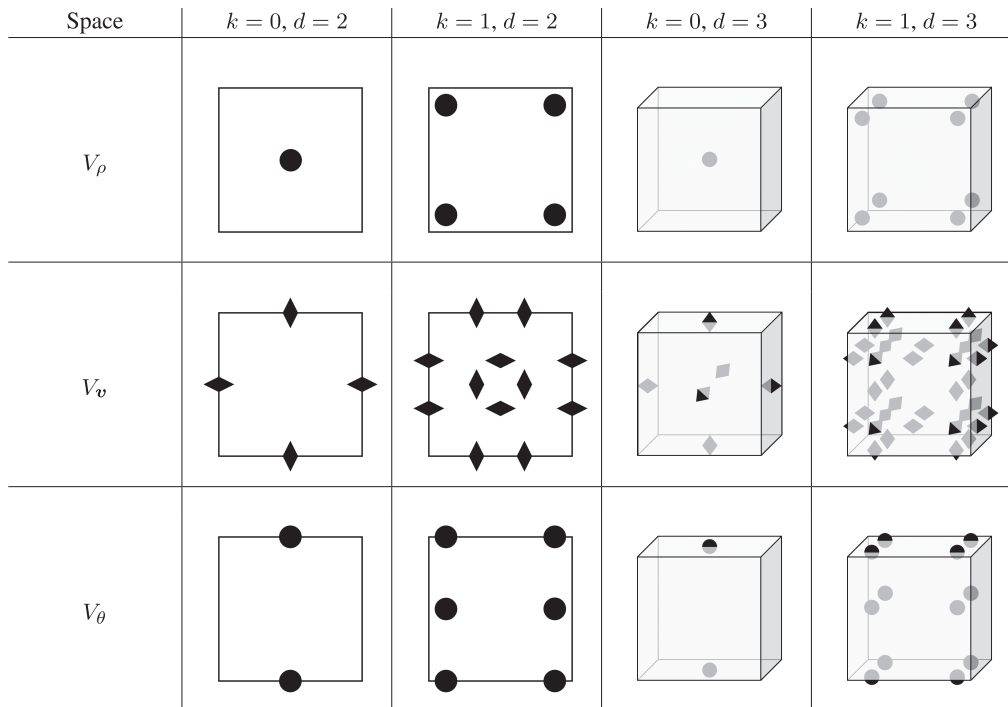
Space	$k = 0, d = 2$	$k = 1, d = 2$	$k = 0, d = 3$	$k = 1, d = 3$
V_ρ				
V_v				
V_θ				

FIGURE 1 An illustration of the function spaces that we use for density, velocity and potential temperature variables in vertical slice ($d = 2$) and three-dimensional ($d = 3$) simulations. We have have two configurations for each case: with vertical and horizontal degree of $k = 0$ and $k = 1$. Degrees of freedom on cell facets are shared between cells and denote the continuity of the field between them

3.2 | Overview

Here we present an outline of our model structure. It is based on the Met Office ENDGame model, described in Walters *et al.*, (2014), in terms of the semi-implicit formulation and physics–dynamics coupling, with the exception that Eulerian advection schemes are used with Eulerian averages in the semi-implicit time discretisation. This is in contrast to the ENDGame semi-Lagrangian approach.

We illustrate this with some pseudocode that is similar to that presented in Bendall *et al.*, (2019). The pseudocode describes how we evolve our prognostic variables, which we denote by the single vector $\chi = (v, \rho_\mathrm{d}, \theta_\mathrm{vd}, r_\mathrm{v}, r_\mathrm{c}, r_\mathrm{r})$. Overall, the model is a mixture of explicit and semi-implicit treatments. Terms in Equation 2 are divided into forcings $\mathcal{F}(\chi)$, advection terms $\mathcal{A}(\chi)$, and physics terms $\mathcal{P}(\chi)$, with forcing being semi-implicit while advection and physics are explicit. This pseudocode is as follows.

1 EXPLICIT FORCING: $\chi^* = \chi^n + \frac{1}{2}\Delta t \mathcal{F}(\chi^n)$
2 SET: $\chi_\mathrm{p}^{n+1} = \chi^n$
3 OUTER:
   a UPDATE: $\bar{u} = \frac{1}{2}(v_\mathrm{p}^{n+1} + v^n)$
   b ADVECT: $\chi_\mathrm{p} = \mathcal{A}_{\bar{u}}(\chi^*)$
   c INNER:

    (i) FIND RESIDUAL (IMPLICIT FORCING):
      $\Delta\chi = \chi_\mathrm{p} + \frac{1}{2}\Delta t \mathcal{F}(\chi_\mathrm{p}^{n+1}) - \chi_\mathrm{p}^{n+1}$
    (ii) SOLVE: $S(\chi') = \Delta\chi$ for χ'
    (iii) INCREMENT: $\chi_\mathrm{p}^{n+1} = \chi_\mathrm{p}^{n+1} + \chi'$

4 PHYSICS: $\chi_\mathrm{p}^{n+1} = \mathcal{P}(\chi_\mathrm{p}^{n+1})$
5 ADVANCE TIME STEP: $\chi^n = \chi_\mathrm{p}^{n+1}$

Given χ^n, the state at the nth time step, the first stage of the time step is to apply the explicit component of the forcing terms, returning χ^*. The action of the forcing is discussed in Section 3.3.

We then enter two loops. In the outer loop, the advecting velocity $\bar{u}$ is determined to be the average of v^n and the best guess of the velocity at the time step, v_p^{n+1}. The variables are then advected explicitly by an advection operator $\mathcal{A}_{\bar{u}}$, returning χ_p. The advection operators are described in Section 3.4. We use different advection operators for the $k = 0$ and $k = 1$ configurations.

To solve the implicit component of the forcing terms, we enter the inner loop. The residual $\Delta\chi$ is calculated between χ_p and the implicit part of the forcing $\mathcal{F}(\chi_\mathrm{p}^{n+1})$, so that the implicit part of the model is solved if $\Delta\chi = 0$. The forcing operator is the same as used in the explicit forcing step, and is described in Section 3.3. We iterate towards the

solution by solving a linearised system of equations for the perturbation χ':

$$S(\chi') = \Delta\chi. \tag{6}$$

The operator $S(\chi')$ represents a linearisation of Equation 2 about a reference state. The solution χ' is then used to increment χ_p^{n+1}. This is described further in Section 3.5.

Once the outer loop is completed, the physics processes are treated explicitly in their own stage.

3.3 | Forcing

The "forcing" operation constitutes the sum of the nonadvective terms in Equations 2a and 2b, which are written in weak form. Our goal is to find $\mathcal{F}(\boldsymbol{v})$ and $\mathcal{F}(\theta_{vd})$. The same operation is used here for both explicit and implicit stages of the model, as illustrated by the pseudocode at the start of Section 3.2.

For both $k = 0$ and $k = 1$ configurations, the forcing that we apply to $\boldsymbol{v}$ is the solution for $\boldsymbol{v}_{\text{trial}} \in V_{\boldsymbol{v}}$, for all $\boldsymbol{\psi} \in V_{\boldsymbol{v}}$, to

$$\int_\Omega \boldsymbol{\psi} \cdot \boldsymbol{v}_{\text{trial}} \, dx = -\int_\Omega [\boldsymbol{\psi} \cdot (\boldsymbol{f} \times \boldsymbol{v}) + g(\boldsymbol{\psi} \cdot \hat{\boldsymbol{k}})] \, dx$$
$$- \int_\Omega c_{pd} \Pi \boldsymbol{\nabla} \cdot \left(\frac{\theta_{vd} \boldsymbol{\psi}}{1 + r_t} \right) dx - \int_\Gamma c_{pd} \left[\!\!\left[\frac{\theta_{vd} \boldsymbol{\psi}}{1 + r_t} \right]\!\!\right]_{\boldsymbol{n}} \langle \Pi \rangle \, dS, \tag{7}$$

where Ω is the domain, Γ is the set of all interior facets, and the angled brackets $\langle \cdot \rangle$ denote the average value on either side of a facet. The double square brackets $[\![\boldsymbol{f}]\!]_{\boldsymbol{n}} = \boldsymbol{f}^+ \cdot \hat{\boldsymbol{n}}^+ + \boldsymbol{f}^- \cdot \hat{\boldsymbol{n}}^-$ denote the jump in values between either side of the facet, arbitrarily labelled with $+$ and $-$. To solve equations such as Equation 7 numerically, we expand the variables as finite sums of coefficients multiplying the basis functions that span the relevant finite-element function spaces. This yields a set of equations for the coefficients of the trial function, which can be solved by standard numerical techniques.

The forcing $\mathcal{F}(\theta_{vd})$ applied to θ_{vd} is the solution θ_{trial}, for all $\gamma \in V_\theta$, to

$$\int_\Omega \gamma \theta_{\text{trial}} \, dx = -\int_\Omega \gamma \theta_{vd} \left(\frac{R_m}{c_{vml}} - \frac{R_d c_{pml}}{c_{pd} c_{vml}} \right) (\boldsymbol{\nabla} \cdot \boldsymbol{v}) \, dx. \tag{8}$$

3.4 | Advection

In the advection stage, each of the variables is translated by the velocity $\bar{\boldsymbol{u}}$. We represent this action by the operator $\mathcal{A}_{\bar{\boldsymbol{u}}}$. Here we briefly provide details of the advection schemes used. We use finite-element transport schemes,

as these are most easily implemented in the Firedrake software that we use.

In the advection step, the $k = 0$ and $k = 1$ configurations differ. This is because the advection schemes that we use for the $k = 1$ spaces do not have second-order accuracy in space when applied to the $k = 0$ spaces. In the $k = 1$ configuration, we transport ρ_d using an upwind discontinuous Galerkin scheme, and for θ_{vd} and the moisture variables we use the embedded version of this scheme presented by Cotter and Kuzmin (2016). These schemes use the three-step Runge–Kutta time-stepping procedure (SSPRK-3) from Shu and Osher (1988). For the transport of $\boldsymbol{v}$, we write the equation in vector-invariant form and use the theta time-stepping method. The remainder of this section details these methods.

The schemes used for the $k = 0$ spaces, called the *recovered advection schemes*, were presented in Bendall *et al.*, (2019). They involve obtaining a representation of the fields in a higher-order space, where the finite-element transport methods will have second-order accuracy. This process is explained in Section 3.4.3.

3.4.1 | Discontinuous Galerkin upwind advection

First, we define a single forward-Euler step of discontinuous Galerkin upwinding, which we describe as the operation $\mathcal{L}_{\bar{\boldsymbol{u}}}$ upon a scalar quantity q. For the advective form of the transport equation, this involves finding the solution q_{trial}, for all $p \in V_q$:

$$\int_\Omega \psi q_{\text{trial}} \, dx - \int_\Omega \psi q \, dx - \Delta t \int_\Omega q[\boldsymbol{\nabla} \cdot (p\bar{\boldsymbol{u}})] \, dx$$
$$+ \Delta t \int_\Gamma (\boldsymbol{u} \cdot \hat{\boldsymbol{n}}^+) q^* [\![p]\!]_+ \, dS = 0, \tag{9}$$

where the jump is $[\![p]\!]_+ = p^+ - p^-$. The outward normal on the $+$ side of the facet is denoted by $\hat{\boldsymbol{n}}^+$ and q^* is the upwind value, identified by

$$q^* = \begin{cases} q^+, & \text{if } \bar{\boldsymbol{u}} \cdot \hat{\boldsymbol{n}}^+ > 0, \\ q^-, & \text{otherwise.} \end{cases} \tag{10}$$

This equation can also be cast in continuity form, as it will be for the transport of ρ_d. The SSPRK-3 time-stepping scheme presented in Shu and Osher (1988) involves composing these steps as follows:

$$q^{(1)} := q + \mathcal{L}_{\bar{\boldsymbol{u}}} q, \tag{11a}$$

$$q^{(2)} := \frac{3}{4} q + \frac{1}{4} (q^{(1)} + \mathcal{L}_{\bar{\boldsymbol{u}}} q^{(1)}), \tag{11b}$$

$$\mathcal{A}_{\overline{\boldsymbol{u}}}q := \frac{1}{3}q + \frac{2}{3}(q^{(2)} + \mathcal{L}_{\overline{\boldsymbol{u}}}q^{(2)}). \qquad (11c)$$

This is the scheme used for transport for ρ_d in the $k=1$ configuration, and is the basis for the embedded DG and recovered advection schemes. The discontinuous Galerkin upwinding method provides some stability, and is not difficult to use with slope limiters to ensure monotonicity.

3.4.2 | Embedded DG advection

The embedded DG transport scheme introduced in Cotter and Kuzmin (2016) is used for advecting θ_vd and the moisture variables in the $k=1$ configuration. The fields are first interpolated into fully discontinuous analogues of their spaces, where the advection takes place. The fields are then returned to their original spaces using a Galerkin projection. See Cotter and Kuzmin (2016) for more details.

3.4.3 | Recovery operator and recovered advection

The advection schemes used for the $k=0$ spaces were presented in Bendall *et al.*, (2019). The motivation for using these schemes is that the methods used for the $k=1$ spaces do not have second-order accuracy when used with the $k=0$ spaces. Other approaches, such as finite-volume schemes with large stencils, are not currently available through the Firedrake software. The recovered schemes extend the embedded DG scheme described in the previous section: the fields are converted from the original space to spaces with higher degrees of polynomial. The transport then occurs in a discontinuous higher-degree space. As in the embedded scheme, this field is projected back to the original space to complete the advection. The spaces used as part of this scheme were described in Bendall *et al.*, (2019).

The operator that we use to obtain the higher-order representation is called the *recovery operator*. It involves finding the gradients within cells by converting to a higher-order continuous function space. The operator determines the values of degrees of freedom shared between elements to be the average from neighbouring elements. These values are then adjusted to ensure that the mass of the field in each cell is preserved. At the exterior boundaries of the domain, this process may not have second-order numerical accuracy and a further extrapolation step is required to represent gradients accurately. For a full description, see Bendall *et al.*, (2019). The recovery operator is also used in physics–dynamics coupling, which

is described in Section 4. This is because it can be used to obtain a field in V_θ from a field in V_ρ.

3.4.4 | Vector invariant advection

In the $k=0$ configuration, we solved the transport equation for $\boldsymbol{v}$ in advective form:

$$\frac{\partial \boldsymbol{v}}{\partial t} + \overline{\boldsymbol{u}} \cdot \nabla \boldsymbol{v} = \boldsymbol{0}. \qquad (12)$$

However, for the $k=1$ spaces, we write the advection equation in vector-invariant form, discretising

$$\frac{\partial \boldsymbol{v}}{\partial t} + (\nabla \times \boldsymbol{v}) \times \overline{\boldsymbol{u}} + \frac{1}{2}\nabla(\boldsymbol{v} \cdot \overline{\boldsymbol{u}}) = \boldsymbol{0}. \qquad (13)$$

The action of the advection operator $\mathcal{L}_{\overline{\boldsymbol{u}}}$ gives the solution $\boldsymbol{v}_\mathrm{trial} \in V_{\boldsymbol{v}}$ for all $\boldsymbol{\psi} \in V_{\boldsymbol{v}}$ to

$$\int_\Omega \boldsymbol{\psi} \cdot (\boldsymbol{v}_\mathrm{trial} - \boldsymbol{v})\, dx - \int_\Gamma \boldsymbol{v}^* \cdot [\hat{\boldsymbol{n}}^+ \times [\![\overline{\boldsymbol{u}} \times \boldsymbol{\psi}]\!]_+]\mathrm{d}S$$
$$+ \int_\Omega \boldsymbol{v} \cdot [\nabla \times (\overline{\boldsymbol{u}} \times \boldsymbol{\psi})]\mathrm{d}x - \frac{1}{2}\int_\Omega (\boldsymbol{v} \cdot \overline{\boldsymbol{u}})(\nabla \cdot \boldsymbol{\psi})\mathrm{d}x = 0,$$
$$(14)$$

with $\boldsymbol{v}^*$ as the upwind value of $\boldsymbol{v}$ and $[\![\cdot]\!]_+$ taking the same definition as in Section 3.4.1.

The implicit midpoint rule is used, so that the velocity at the $(n+1)$th time step is found by solving

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + \Delta t \left[\frac{1}{2}\mathcal{L}_{\overline{\boldsymbol{u}}}\boldsymbol{v}_n + \frac{1}{2}\mathcal{L}_{\overline{\boldsymbol{u}}}\boldsymbol{v}_{n+1}\right]. \qquad (15)$$

3.4.5 | Limiting

The advection schemes that we use do not preserve the monotonicity property of the continuous transport equation. This can result in unphysical negative concentrations when advecting moisture variables. A way of preventing this is to apply a slope limiter.

We provide the option of adding a limiter to the advection of variables in V_θ. The vertex-based limiter of Kuzmin (2010) is designed for use on fields that are piecewise-linear and discontinuous between cells. It prevents the formation of new maxima and minima by separating the field into a constant mean part and adjusting the perturbed part in each cell. It is applied to the field before the advection operator $\mathcal{V}_{\overline{\boldsymbol{u}}}$ first acts at each time step and after each stage of the three-step Runge–Kutta method.

For advection of moisture variables in the $k=0$ set of spaces, we use the recovered space scheme and apply the vertex-based limiter of Kuzmin (2010). When returning
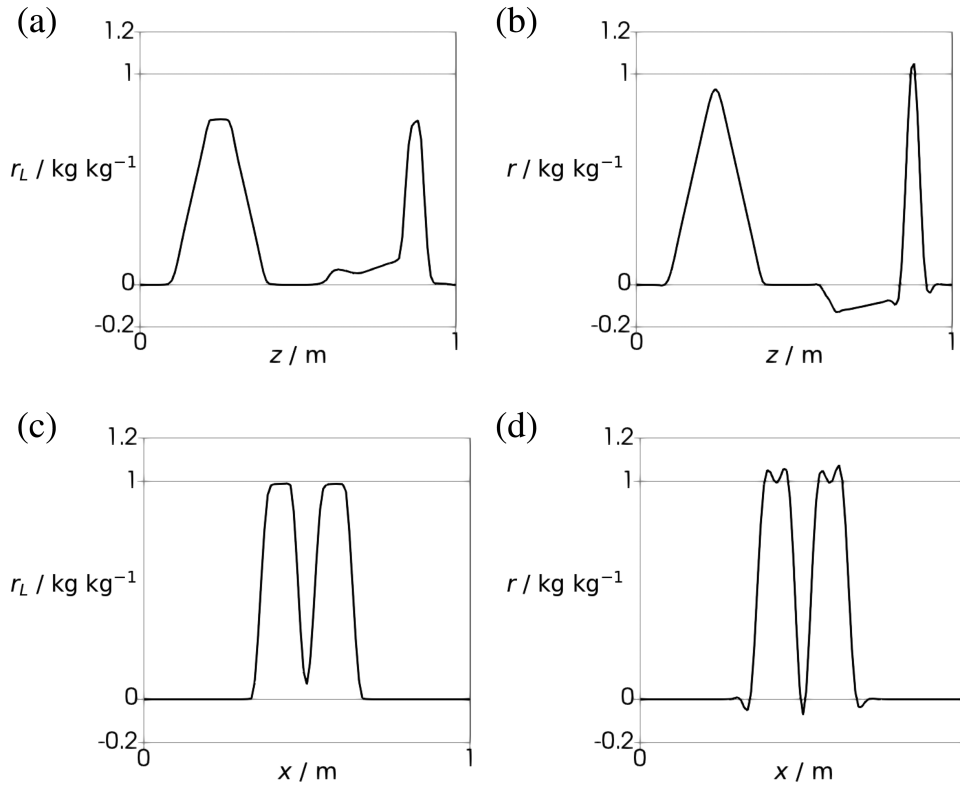
FIGURE 2 Slices of the final fields, following one rotation around the domain, of the slotted-cylinder, hump, and cone function of LeVeque (1996). The fields represent moisture variables in the two-dimensional $dQ_1 \otimes Q_2$, (a,c) with the limiter applied, and (b,d) without the limiter. (a,b) Slices along $x = 0.5$ m and (c,d) slices through the cylinder along $z = 0.8$ m. The left plots (with the limiter) show that the limiter prevents formation of overshoots and undershoots without damping the solution excessively, in contrast to the plots on the right, which do show overshoots and undershoots

the field back from the higher order space, we first perform a Galerkin projection into a fully discontinuous version of the original space and restore continuity using the appropriate recovery operator. For more details, see Bendall *et al.*, (2019).

With the $k = 1$ spaces, the vertex-based limiter of Kuzmin (2010) is again used. However, this field is quadratic in the vertical and the values of the field at the vertical midpoint of the cell are unchanged by the vertex-based limiter. We bound these values by restricting them to the average of the values at the adjacent vertices if this falls outside the range spanned by them. In other words, if $r_{i+1/2}$ is the value of the field at a degree of freedom halfway up the ith cell, then the new value is

$$
r_{i+1/2} = \begin{cases} r_{i+1/2}, & \text{if } \min(r_i, r_{i+1}) \leq r_{i+1/2} \\ & \text{and } r_{i+1/2} \leq \max(r_i, r_{i+1}), \\ \frac{1}{2}(r_i + r_{i+1}), & \text{otherwise.} \end{cases} \tag{16}
$$

To illustrate the effectiveness of this limiter, we performed the rotation of the slotted-cylinder, hump, and cone of LeVeque (1996) in a 2D vertical slice. Figure 2 shows the fields in V_θ space for the $k = 1$ configuration, rotated once around the domain by a velocity defined by

the stream function

$$
\psi = \pi(z(z - 1) + x(x - 1)). \tag{17}
$$

The domain used was the unit square, with $\Delta x = \Delta z = 0.01$, whilst the time step was $\Delta t = 10^{-4}$. The final fields were recorded at $t = 1$. Figure 2 demonstrates that the limiter does indeed prevent overshoots and undershoots from developing.

3.5 | Solving

We now present the strategy for the linear solve step of the model. When initialising the model, we set mean ρ_d and θ_{vd} states, which we denote respectively as $\bar{\rho}_d$ and $\bar{\theta}_{vd}$. The residual $\Delta \chi$ between the predicted explicit χ_p and the latest guess χ_p^{n+1} of the prognostic variables is computed first. Reducing this residual to zero solves the implicit part of the model. This is done by solving a linear set of equations, with the residual on the right-hand side, to calculate increments that are added to χ_p^{n+1}, updating it.

The linear equations that we solve are arrived at by linearising the full Equations 2 around the background state of $\bar{\rho}_d, \bar{\theta}_{vd}$, and zero velocity. Discretised in time, these

linear equations are:

$$\boldsymbol{v}' + \frac{\Delta t c_{\mathrm{pd}}}{2(1+r_{\mathrm{t}})}(\theta'_{\mathrm{vd}}\boldsymbol{\nabla}\overline{\Pi} + \overline{\theta}_{\mathrm{vd}}\boldsymbol{\nabla}\Pi') = \Delta\boldsymbol{v}, \tag{18a}$$

$$\rho'_{\mathrm{d}} + \frac{1}{2}\Delta t \boldsymbol{\nabla} \cdot (\overline{\rho}_{\mathrm{d}}\boldsymbol{v}') = \Delta\rho_{\mathrm{d}}, \tag{18b}$$

$$\theta'_{\mathrm{vd}} + \frac{1}{2}\Delta t(\hat{\boldsymbol{k}}\cdot\boldsymbol{v}')(\hat{\boldsymbol{k}}\cdot\boldsymbol{\nabla}\overline{\theta}_{\mathrm{vd}}) = \Delta\theta_{\mathrm{vd}}, \tag{18c}$$

where the primed quantities represent the perturbations to be found.

The values of the perturbations are found by solving a mixed finite-element problem, with more than one test and trial function. First we simplify the equations by eliminating the potential temperature perturbation θ'_{vd}. This allows us to produce a mixed system coupling the velocity and density perturbations, $\boldsymbol{v}'$ and ρ'_{d}. Taking $\hat{\boldsymbol{k}}$ as the unit upward normal, we introduce

$$\Theta' = \Delta\theta_{\mathrm{vd}} - \frac{1}{2}\Delta t(\hat{\boldsymbol{k}}\cdot\boldsymbol{\nabla}\overline{\theta}_{\mathrm{vd}})(\hat{\boldsymbol{k}}\cdot\boldsymbol{v}'_{\mathrm{trial}}). \tag{19}$$

The Exner pressure perturbation is then approximated as

$$\Pi' = \frac{\kappa}{1-\kappa}\overline{\Pi}\left(\frac{\Theta'}{\overline{\theta}_{\mathrm{vd}}} + \frac{\rho'_{\mathrm{trial}}}{\overline{\rho}_{\mathrm{d}}}\right),$$

$$\text{where } \overline{\Pi} = \left(\frac{\overline{\rho}_{\mathrm{d}}\overline{\theta}_{\mathrm{vd}}R_{\mathrm{d}}}{p_{\mathrm{R}}}\right)^{\kappa/(1-\kappa)}. \tag{20}$$

To solve the mixed finite-element problem for $\boldsymbol{v}'_{\mathrm{trial}}$ and ρ'_{trial}, we use a technique known as *hybridisation*. This involves relaxing the continuity requirement on the normal components of functions in $V_{\boldsymbol{v}}$. This new space, $\hat{V}_{\boldsymbol{v}}$, is a discontinuous variant of $V_{\boldsymbol{v}}$, since we are no longer requiring $\boldsymbol{v}' \cdot \hat{\boldsymbol{n}}$ to be continuous on cell interfaces. We then introduce an auxiliary field ℓ' in a function space V_{trace}, which approximates the average values of Π on cell faces. The auxiliary space V_{trace} consists of scalar functions, which are represented as polynomials on cell facets. In order to ensure that the discretisation remains within the compatible finite-element framework, we require the trial function $\boldsymbol{v}'_{\mathrm{trial}}$ to satisfy the condition

$$\int_{\Gamma}\lambda[\![\boldsymbol{v}'_{\mathrm{trial}}]\!]_{\boldsymbol{n}}\,dS + \int_{\partial\Omega}\lambda(\boldsymbol{v}'_{\mathrm{trial}}\cdot\hat{\boldsymbol{n}})\,ds = 0 \tag{21}$$

for all $\lambda \in V_{\mathrm{trace}}$. Equation 21 ensures that, while we appear to be searching for a velocity perturbation in

the discontinuous $\hat{V}_{\boldsymbol{v}}$, it is actually in $V_{\boldsymbol{v}}$. In other words, satisfying Equation 21 is equivalent to having $\boldsymbol{v}'_{\mathrm{trial}} \in V_{\boldsymbol{v}}$.

After substituting Equation 19 into Equation 18a, we multiply the result and Equation 18b by test functions in $\hat{V}_{\boldsymbol{v}}$ and V_{ρ} and integrate. Adding the condition of Equation 21, the resulting mixed system for $(\boldsymbol{v}'_{\mathrm{trial}}, \rho'_{\mathrm{trial}}, \ell'_{\mathrm{trial}}) \in (\hat{V}_{\boldsymbol{v}}, V_{\rho}, V_{\mathrm{trace}})$, which holds for all $\boldsymbol{\psi} \in \hat{V}_{\boldsymbol{v}}$, $\phi \in V_{\rho}$, and $\lambda \in V_{\mathrm{trace}}$, can be written as

$$\begin{aligned}
&\int_{\Omega}\boldsymbol{\psi}\cdot(\boldsymbol{v}'_{\mathrm{trial}} - \Delta\boldsymbol{v})\,dx \\
&-\frac{\Delta t}{2}c_{\mathrm{pd}}\left(\int_{\Omega}\boldsymbol{\nabla}\cdot\left[\frac{\Theta'w}{1+r_{\mathrm{t}}}\hat{\boldsymbol{k}}\right]\overline{\Pi}\,dx\right. \\
&-\int_{\Gamma}[\![\frac{\Theta'w}{1+r_{\mathrm{t}}}\hat{\boldsymbol{k}}]\!]_{\boldsymbol{n}}\langle\overline{\Pi}\rangle\,dS\Bigg) \\
&+\frac{\Delta t}{2}c_{\mathrm{pd}}\left(\int_{\partial\Omega}\frac{\Theta'w}{1+r_{\mathrm{t}}}\hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{n}}\langle\overline{\Pi}\rangle\,ds\right. \\
&-\int_{\Omega}\boldsymbol{\nabla}\cdot\left[\frac{\overline{\theta}_{\mathrm{vd}}}{1+r_{\mathrm{t}}}\boldsymbol{\psi}\right]\Pi'\,dx\Bigg) \\
&+\frac{\Delta t}{2}c_{\mathrm{pd}}\int_{\Gamma}[\![\frac{\overline{\theta}_{\mathrm{vd}}}{1+r_{\mathrm{t}}}\boldsymbol{\psi}]\!]_{\boldsymbol{n}}\ell'_{\mathrm{trial}}\,dS \\
&+\frac{\Delta t}{2}\int_{\partial\Omega}\frac{\overline{\theta}_{\mathrm{vd}}}{1+r_{\mathrm{t}}}(\boldsymbol{\psi}\cdot\hat{\boldsymbol{n}})\ell'_{\mathrm{trial}}\,ds \\
&+\int_{\Omega}\phi(\rho'_{\mathrm{trial}} - \Delta\rho_{\mathrm{d}})\,dx - \frac{\Delta t}{2}\int_{\Omega}(\boldsymbol{\nabla}\phi\cdot\boldsymbol{v}'_{\mathrm{trial}})\overline{\rho}_{\mathrm{d}}\,dx \\
&+\frac{\Delta t}{2}\left(\int_{\Gamma}[\![\phi\boldsymbol{v}'_{\mathrm{trial}}]\!]_{\boldsymbol{n}}\langle\rho_{\mathrm{d}}\rangle\,dS\int_{\partial\Omega}\phi\boldsymbol{v}'_{\mathrm{trial}}\cdot\hat{\boldsymbol{n}}\langle\rho_{\mathrm{d}}\rangle\,ds\right) \\
&+\int_{\Gamma}\lambda[\![\boldsymbol{v}'_{\mathrm{trial}}]\!]_{\boldsymbol{n}}\,dS\int_{\partial\Omega}\lambda(\boldsymbol{v}'_{\mathrm{trial}}\cdot\hat{\boldsymbol{n}})\,ds = 0,
\end{aligned} \tag{22}$$

where $w = \hat{\boldsymbol{k}}\cdot\boldsymbol{\psi}$ is the vertical component of the velocity test function and $\partial\Omega$ is the external boundary of the domain. This mixed system can be manipulated algebraically into a single system for $\ell\mathrm{trial}'$. This is the only globally coupled system requiring iterative inversion. Once ℓ' has been determined, ρ'_{d} and $\boldsymbol{v}'$ are reconstructed locally using values of ℓ'. Finally, the value of θ_{vd} is found as the θ'_{trial} that solves, for all $\gamma \in V_{\theta}$,

$$\int_{\Omega}\gamma\left[\theta'_{\mathrm{trial}} - \Delta\theta_{\mathrm{vd}} + \frac{\Delta t}{2}(\hat{\boldsymbol{k}}\cdot\boldsymbol{v})(\hat{\boldsymbol{k}}\cdot\boldsymbol{\nabla}\overline{\theta}_{\mathrm{vd}}]\right) = 0. \tag{23}$$

Further information on hybridisation techniques and their implementation can be found in Gibson *et al.* (2019; 2020). More on this particular linear solver and its performance will be detailed in future work.

4 | PHYSICS PARAMETRISATIONS

In this section, we consider the discretisation of the "physics" terms in Equations 2. The term labelled $\dot{r}^c_{cond}$ represents the condensation of water vapour into cloud water and the evaporation of cloud water into water vapour. We use the rate given in Rutledge and Hobbs (1983) and Bryan and Fritsch (2002), with the saturation mixing ratio from Tetens (1930), giving

$$r_{sat}(\tilde{p}, T) = \frac{\epsilon C_0^{sat}}{\tilde{p} \exp\left[-\frac{C_1^{sat}(T-T_R)}{T-C_2^{sat}}\right] - C_0^{sat}}, \quad (24)$$

where C_0^{sat}, C_1^{sat}, and C_2^{sat} are constants, the values of which are given in the Appendix. The pressure $\tilde{p}$ will be explained in the following section.

The similar term $\dot{r}^r_{evap}$ describes the evaporation of rain water into water vapour, with the value taken from Klemp and Wilhelmson (1978).

The coalescence of cloud water into rain droplets is described by $\dot{r}_{accr}$ and $\dot{r}_{accu}$, the accretion and auto-accumulation processes respectively. The formulas that we use are also those described in Klemp and Wilhelmson (1978), with values from Soong and Ogura (1973).

The sedimentation of rain is given by S. Our approach to determining S is similar to the single-moment scheme described in Milbrandt and McTaggart-Cowan (2010). We assume that the number $n_r(D)$ of raindrops of diameter D forms a spectrum described by a Gamma distribution. This is used to determine the terminal velocity of raindrops averaged over D. The rain field r_r is then advected by this averaged velocity using the same advection scheme used by the dynamics for θ_{vd} and the moisture variables described in Section 3.

4.1 | Combining fields from different function spaces

Many diagnostic fields, such as temperature T or saturation mixing ratio of water vapour r_{sat}, must be determined from prognostic fields in different function spaces, usually ρ_d, θ_{vd}, and r_v.

As ρ_d and θ_{vd} lie in different function spaces, when determining some diagnostic variable q there are a number of different choices that can be made. With our emphasis on parametrisations of moist processes, we choose $q \in V_\theta$, so that it is also in the same function space as the moisture variables. Our approach to combining these fields is to recover ρ_d is into V_θ using the recovery operator outlined in Section 3.4.3 to give a result that we denote as

$\tilde{\rho}_d$. For the lowest-order spaces, the values of $\tilde{\rho}_d$ (which are shared on the interfaces between layers in the mesh) are simply the average of the values in the cells above and below. Then q can be calculated algebraically within V_θ. This approach can be used with both the $k = 0$ and $k = 1$ sets of spaces, and with the $k = 0$ spaces it has second-order numerical accuracy (including at the boundaries of the domain). For example, the temperature T and pressure p used in parametrisations are calculated as

$$T = \frac{\theta_{vd}\tilde{\Pi}}{1 + r_v/\epsilon}, \quad \tilde{p} = p_R \tilde{\Pi}^{1/\kappa}, \quad (25)$$

where

$$\tilde{\Pi} = \left(\frac{\tilde{\rho}_d R_d \theta_{vd}}{p_R}\right)^{\kappa/(1-\kappa)}. \quad (26)$$

4.2 | Time discretisation

At present, we are performing simple explicit first-order integration in time for the physics routines. In other words, for a process $\dot{r}$ affecting a variable r, the new value r_{new} will be related to the old r_{old} by

$$r_{new} = r_{old} + \Delta t \, \dot{r}(r_{old}). \quad (27)$$

The exception here is the treatment of the sedimentation of rainfall, which was described in the previous section. The value of $\dot{r}$ comes from the state of the model just before this physics routine is called. This is the state of the model after either the dynamics or the preceding physics routine has been completed. The physics routines are executed consecutively (this is commonly known as "sequential splitting"), in the following order:

1. accretion of cloud water,
2. auto-accumulation of rain water,
3. sedimentation of rain,
4. evaporation of rain water, and
5. evaporation of cloud water/condensation of water vapour.

We choose to perform the evaporation/condensation step last so as to prevent any supersaturation at the end of the time step.

5 | TEST CASES

In this section, we demonstrate the discretisation detailed in previous sections through a series of test cases, with some comparison of the $k = 0$ and $k = 1$ configurations

of the model. Two new variants of test cases are presented, featuring a gravity wave in a saturated atmosphere and a three-dimensional rising thermal in a saturated atmosphere.

Throughout this section, x and y are the horizontal coordinates, and z is the vertical coordinate. For the two-dimensional tests, r is given by

$$r = \sqrt{(x - x_c)^2 + (z - z_c)^2}, \tag{28}$$

while for the three-dimensional thermal test it is

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2}, \tag{29}$$

where x_c, y_c, and z_c represent the centre of some perturbation.

5.1 | Bryan and Fritsch moist benchmark

The first demonstration of our discretisation is through the moist benchmark test case of Bryan and Fritsch (2002), which simulates a rising thermal through a cloudy atmosphere. The domain is a vertical slice of width $L = 20$ km and height $H = 10$ km. Periodic boundary conditions are applied at the vertical boundaries, but the top and bottom boundaries are rigid, so that $\boldsymbol{v} \cdot \hat{\boldsymbol{n}} = 0$ along them. As in Bryan and Fritsch (2002), we include no rain microphysics and no Coriolis force.

The initial conditions defined in Bryan and Fritsch (2002) specify a background state with constant $r_t = 0.02$ kg·kg^{-1} and constant wet-equivalent potential temperature $\theta_e = 320$ K, which is defined in Equation B2. Along with these, the background state is given by the requirements of hydrostatic balance, $r_v = r_{sat}$, and $p = 10^5$ Pa at the bottom boundary. The procedure described in Appendix B2 allows us to obtain the prognostic variables θ_{vd}, ρ_d, r_v, and r_c that satisfy these conditions approximately.

The following perturbation is then applied to θ_{vd}:

$$\theta'_{vd} = \begin{cases} \Delta\Theta\cos^2\left(\frac{\pi r}{2r_c}\right), & r < r_c, \\ 0, & \text{otherwise}, \end{cases} \tag{30}$$

where $\Delta\Theta = 2$ K, $l_c = 2$ km, $x_c = L/2$, and $z_c = 2$ km. The initial θ_{vd} field is given in terms of the background field $\overline{\theta}_{vd}$:

$$\theta_{vd} = \overline{\theta}_{vd}\left(1 + \frac{\theta'_{vd}}{300 \text{ K}}\right). \tag{31}$$

In the test case of Bryan and Fritsch (2002), the pressure field is unchanged by the perturbation. To replicate this with our prognostic variables, we find ρ_d such that, for all $\zeta \in V_\rho$,

$$\int_\Omega \zeta \rho_d \theta_{vd} \, dx = \int_\Omega \zeta \overline{\rho}_d \overline{\theta}_{vd} \, dx, \tag{32}$$

where $\overline{\rho}_d$ and $\overline{\theta}_{vd}$ are the background states for ρ_d and θ_{vd}. The system is returned to saturation by finding the r_v that solves, for all $\phi \in V_\theta$,

$$\int_\Omega \phi r_v \, dx = \int_\Omega \phi r_{sat}(\tilde{\rho}_d, \theta_{vd}, r_v) \, dx, \tag{33}$$

where $r_{sat}(\tilde{\rho}_d, \theta_{vd}, r_v)$ is the expression for the saturation mixing ratio. This expression is given in terms of the initial ρ_d recovered into V_θ and the initial θ_{vd} field that has already been found, and also the unknown r_v to be solved for. Finally, r_c is found from applying $r_c = r_t - r_v$. The initial velocity field is zero in each component.

Figures 3 and 4 show the θ_e and vertical velocity w fields at $t = 1{,}000$ s. Figure 3 shows these fields for the configuration using the $k = 0$ lowest-order spaces, with the $k = 1$ spaces shown in Figure 4. Both simulations used $\Delta x = \Delta z = 100$ m and $\Delta t = 1$ s. These final states are visibly different: whilst the $k = 0$ solutions resemble those of Bryan and Fritsch (2002), the $k = 1$ solution displays an extra plume forming at the top of the rising thermal. We believe this to be a manifestation of a physical instability that is damped by numerical diffusion in the $k = 0$ case. The $k = 1$ solution appears highly sensitive to the choice of mesh, as, at higher resolution, the θ_e field does not appear to converge to a single solution. Indeed, if the domain is spanned horizontally by an odd number of cells, rather than a secondary plume emerging, the top of the primary plume appears to collapse. This behaviour is also observed in the absence of moisture.

5.2 | Inertia–gravity waves in a saturated atmosphere

We present here a new test case, a moist version of the nonhydrostatic gravity-wave test of Skamarock and Klemp (1994), but in a saturated atmosphere like that of the moist benchmark in Bryan and Fritsch (2002). The final state of this test is spatially smooth, making this test appropriate for convergence tests. No rain physics is used in this test case, and there is also no Coriolis force.

The problem is set up in a two-dimensional vertical slice of length $L = 300$ km and height $H = 10$ km. The dry gravity-wave setup used in Skamarock and Klemp (1994) applies a perturbation to θ_d, which has a stratified background profile and is in hydrostatic balance. Our variation
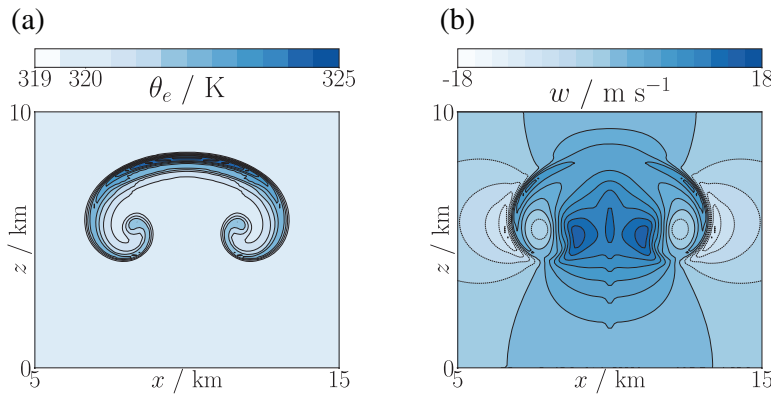
(a)

(b)

FIGURE 3 (a) θ_e field contoured every 0.5 K and (b) vertical velocity w field contoured every 2 m·s^{-1}, with both fields plotted at $t = 1{,}000$ s for a simulation of the moist benchmark case Bryan and Fritsch (2002) representing a thermal rising through a saturated atmosphere. The 320-K contour has been omitted for clarity in the θ_e field. This simulation is with the $k = 0$ lowest-order set of spaces, with grid spacing $\Delta x = \Delta z = 100$ m and a time step of $\Delta t = 1$ s. These solutions are visibly similar to those presented in Bryan and Fritsch (2002)

(a)

(b)
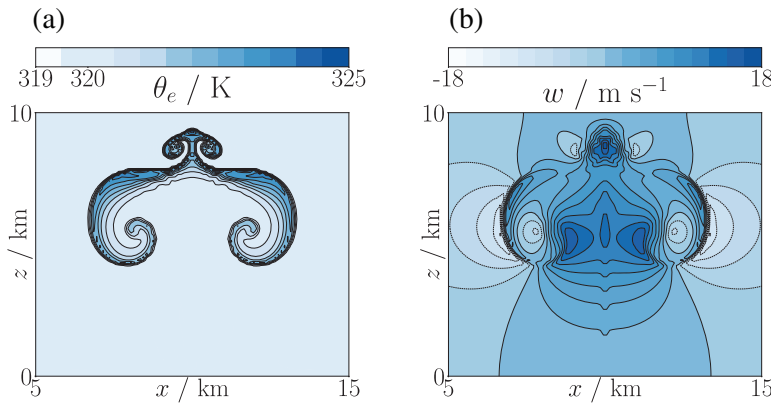
FIGURE 4 Fields output from the $k = 1$ next-to-lowest degree space simulation at $t = 1{,}000$ s of the moist benchmark from Bryan and Fritsch (2002). (a) θ_e with contours spaced by 0.5 K and (b) vertical velocity w contoured every 2 m·s^{-1}. The simulation used grid spacing $\Delta x = \Delta z = 100$ m and a time step of $\Delta t = 1$ s. The 320-K contour has been omitted for clarity in the θ_e field. A second plume can be seen forming at the top of the primary plume

on this is to apply the perturbation to the stratified background profile of θ_e in hydrostatic balance. Using (x,z) as the horizontal and vertical coordinates, the specified θ_e profile is

$$\bar{\theta}_e = \Theta_0 e^{N^2 z/g}, \tag{34}$$

where $\Theta_0 = 300$ K and $N^2 = 10^{-4}$ s^{-2}. With $r_t = 0.02$ kg·kg^{-1} everywhere and the boundary condition of $p = 10^5$ Pa at $z = 0$, we use the hydrostatic balance procedure laid out in Appendix B2 to find the θ_{vd}, ρ_d, r_v, and r_c fields that correspond to these initial conditions with the requirements of hydrostatic balance and $r_v = r_{sat}$ everywhere. The initial velocity applied is $\mathbf{v} = (U,0)$, with $U = 20$ m·s^{-1} describing uniform flow in the x-direction. This defines each of the mean fields.

A perturbation is then added, which is specified as

$$\theta'_e = \frac{\Delta\Theta}{1 + a^{-2}(x - L/2)^2} \sin\left(\frac{\pi z}{H}\right), \tag{35}$$

with $a = 5 \times 10^3$ m and $\Delta\Theta = 0.01$ K. The perturbed initial condition is then given by $\theta_e = \bar{\theta}_e + \theta'_e$. Setting the new requirements that both r_t and the pressure are unchanged by the addition of the perturbation and that we still have $r_v = r_{sat}$ defines the problem necessary to solve in order to find the initial prognostic fields. We do this via a nested

iterative process related to that described in Appendix B2. In the outer loop, we find ρ_d^h such that, for all $\zeta \in V_\rho$,

$$\int_\Omega \zeta \rho_d^h \theta_{vd}^n \, dx = \int_\Omega \zeta \bar{\rho}_d \bar{\theta}_{vd} \, dx, \tag{36}$$

which is combined with the previous best estimate of ρ_d^n to give $\rho_d^{n+1} = (1 - \delta)\rho_d^n + \delta\rho_d^h$, where $\delta = 0.8$. Nested inside this process are more damped iterations to find θ_{vd} and r_v, exactly as in Appendix B2.

Figure 5 shows the perturbation to the final diagnostic θ_e field at $t = 3{,}600$ s for simulations with the $k = 0$ lowest-degree spaces with $\Delta x = \Delta z = 500$ m and the $k = 1$ set of spaces, where $\Delta x = \Delta z = 1{,}000$ m, both with $\Delta t = 1.2$ s. These different cases are not visibly different from one another, and closely resemble the final state of the dry case from Skamarock and Klemp (1994).

To measure the spatial accuracy of our model, we ran this test case at different resolutions, each using a time step of $\Delta t = 1.2$ s. The error is measured looking at the θ_e diagnostic in V_θ at $t = 3{,}600$ s. The θ_e fields are interpolated on to the finest mesh, which has $\Delta x = 100$ m for the $k = 0$ case but $\Delta x = 200$ m for the $k = 1$ case. The error between the high-resolution solution for θ_e and those run at coarser resolutions is plotted in Figure 6, which indicates that in both the $k = 0$ and $k = 1$ cases the model has

FIGURE 5 The perturbations to the θ_e fields at $t = 3{,}600$ s for the moist gravity-wave test case. (a) The $k = 0$ lowest-order spaces set up using $\Delta x = 500$ m and (b) the $k = 1$ spaces with $\Delta x = 1{,}000$ m. Both cases used $\Delta t = 1.2$ s. Contours are spaced every 5×10^{-4} K. The solutions in both configurations are visibly similar
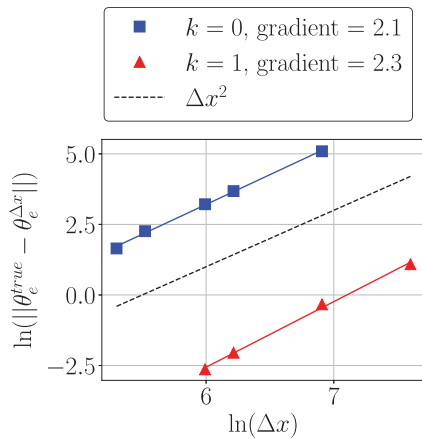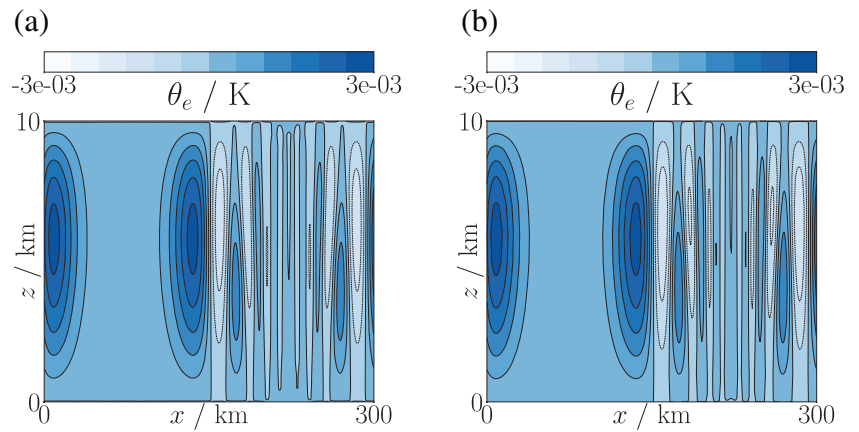
FIGURE 6 A convergence plot showing the error as a function of resolution in the final state from the moist gravity-wave test of Section 5.2. The true solution was taken from a high-resolution simulation. Both the $k = 0$ and $k = 1$ configurations of the model have second-order or better accuracy

second-order spatial accuracy, with the error proportional to $(\Delta x)^2$.

5.3 | Rising thermal with rain

This test case is based upon one described in Grabowski and Clark (1991). This involves a thermal rising in an unsaturated atmosphere, forming a cloud, which rains out. This is another two-dimensional vertical slice test, this time with a domain of height $H = 2.4$ km and length $L = 3.6$ km, again with periodic conditions at the vertical sides. The Coriolis force is neglected.

In contrast to the saturated atmosphere initial conditions of Section 5.1, the initial state is defined by the dry potential temperature θ_d and a relative humidity field $\mathcal{H}$. The background fields are $\mathcal{H} = 0.2$ everywhere and

$$\theta_d = \Theta e^{Sz}, \tag{37}$$

where Θ is the dry potential temperature corresponding to $T_{surf} = 283$ K and $p = 8.5 \times 10^4$ Pa, which also provides the pressure condition at the boundary. The stratification is given by $S = 1.3 \times 10^{-5} \cdot \mathrm{m}^{-1}$. We then use the procedure outlined in Appendix B3 to find the background θ_{vd}, ρ_d, and r_v fields that satisfy hydrostatic balance. The initial r_c and r_r fields are zero.

The perturbation is then applied to the relative humidity field $\mathcal{H}$, with a circular bubble that is just saturated, with an outer disc smoothing the perturbation into the background state. This initial relative humidity field is given by

$$\mathcal{H} = \begin{cases} \overline{\mathcal{H}}, & r \geq r_1, \\ \overline{\mathcal{H}} + (1 - \overline{\mathcal{H}})\cos^2\left(\frac{\pi(r - r_2)}{2(r_1 - r_2)}\right), & r_2 \leq r < r_1, \\ 1, & r < r_2, \end{cases} \tag{38}$$

where $\overline{\mathcal{H}} = 0.2$, with $x_c = L/2$, $z_c = 800$ m, $r_1 = 300$ m, and $r_2 = 200$ m. The r_v and θ_{vd} that correspond to this $\mathcal{H}$ are found via a fixed-point iterative method.

Fields are displayed in Figures 7 and 8 for θ'_{vd} at $t = 300$ and 600 s. Both simulations use the $k = 0$ lowest-order space set-up, with $\Delta x = 20$ m and $\Delta t = 1$ s. Figure 7 shows the results with no limiter applied to the advected moisture fields, whilst Figure 8 shows the same set-up but with a limiter applied to all the moisture variables.

The length-scales of the simulation are small enough for it to be highly turbulent, with the final state dependent on the resolution in the absence of a turbulence parametrisation. Indeed, the lack of turbulence parametrisation in our model explains why these results look significantly different from those of Grabowski and Clark (1991). Comparing Figures 7 and 8 demonstrates the effect of limiting the transport of moisture species. In the absence of the limiter, the mass of water vapour depreciates less and so more cloud is formed, associated with a greater release of latent heat and a stronger updraught. We found that rain
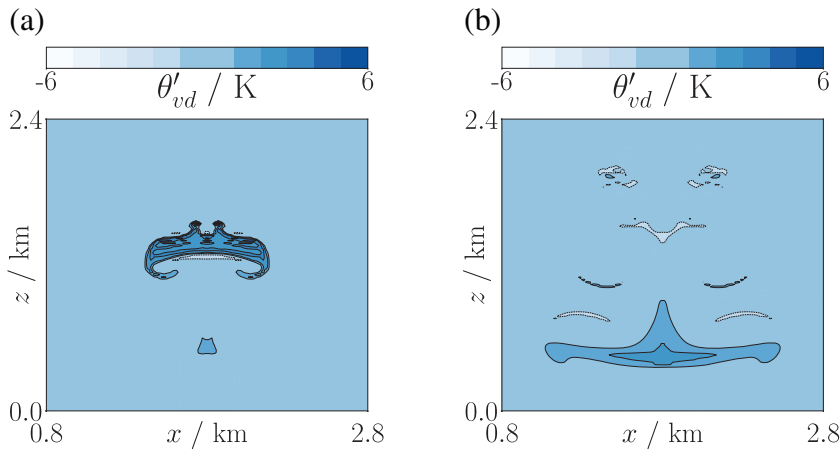
(a)

θ'_{vd} / K

(b)

θ'_{vd} / K

FIGURE 7 The perturbation to θ_{vd}, contoured every 0.5 K with zero contour omitted, at (a) $t = 300$ s and (b) $t = 600$ s from the rising thermal test with rain. These were produced using the $k = 0$ spaces with $\Delta x = \Delta z = 20$ m and a time step of $\Delta t = 1$ s
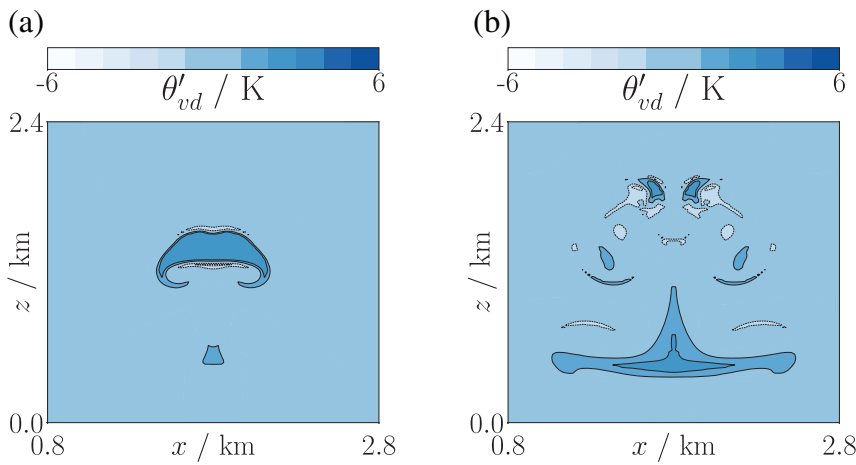

(a)

θ'_{vd} / K

(b)

θ'_{vd} / K

FIGURE 8 The perturbation to θ_{vd}, contoured every 0.5 K with zero contour omitted, at (a) $t = 300$ s and (b) $t = 600$ s from the rising thermal test with rain. This simulation is like that of Figure 7, but a limiter was applied to the transport of the moisture variables. The appearance of the bubble is noticeably different here, as less cloud is formed, with less associated latent heat release

forms earlier in the absence of a limiter. However, some negative moisture values do form, which are absent from the limited case.

5.4 | Three-dimensional thermal in a saturated atmosphere

We now demonstrate the use of our discretisation upon small-scale dynamics in three dimensions. This test case is a three-dimensional version of the moist benchmark of Bryan and Fritsch (2002) that was described in Section 5.1. Rain and the effects of planetary rotation are not included.

The domain is now periodic in the horizontal directions, with length, width, and height 10 km. The background state set-up is the same as that in Section 5.1, with $\theta_e = 320$ K, $r_t = 0.02$ kg·kg^{-1}, and pressure $p = 10^5$ Pa on the bottom surface. Using the initialisation procedure that was outlined in Appendix B2 generates the values of the prognostic variables such that the model is in hydrostatic balance and saturated with respect to water vapour.

Using $x_c = y_c = 5$ km and $z_c = 2$ km, we apply the perturbation

$$\theta'_{vd} = \begin{cases} \Delta\Theta\cos^2\left(\frac{\pi r}{2l_c}\right), & r < l_c, \\ 0, & \text{otherwise,} \end{cases} \quad (39)$$

with $\Delta\Theta = 1$ K and $l_c = 2$ km. As in Section 5.1, the perturbation is applied using the background field $\bar{\theta}_{vd}$:

$$\theta_{vd} = \bar{\theta}_{vd}\left(1 + \frac{\theta'_{vd}}{300 \text{ K}}\right). \quad (40)$$

The same routine as used in Section 5.1 is then applied to obtain the initial ρ_d, r_v, and r_c fields, ensuring that the atmosphere is exactly saturated and that the initial pressure field is equal to the background pressure field.

Cross-sections of the θ_e and vertical velocity w fields at $t = 1000$ s and $y = 5$ km are shown in Figures 9 and 10, for both the setups, using $k = 0$ lowest-order spaces (which had $\Delta x = \Delta y = \Delta z = 100$ m) and $k = 1$ spaces (which had $\Delta x = \Delta y = \Delta z = 200$ m). Both simulations had $\Delta t = 1$ s.

FIGURE 9 (a) θ_e and (b) vertical velocity w fields at $t = 1,000$ s for the three-dimensional simulation of a thermal rising through a saturated atmosphere. The θ_e field is contoured every 0.25 K with the 320-K contour omitted, whilst the contour spacing for the w field is 1 m·s^{-1}. Cross-sections are shown at $y = 5$ km, with values plotted on the lower side of the plane. This simulation is with the $k = 0$ lowest-order set of spaces, with grid spacing $\Delta x = 100$ m and a time step of $\Delta t = 1$ s
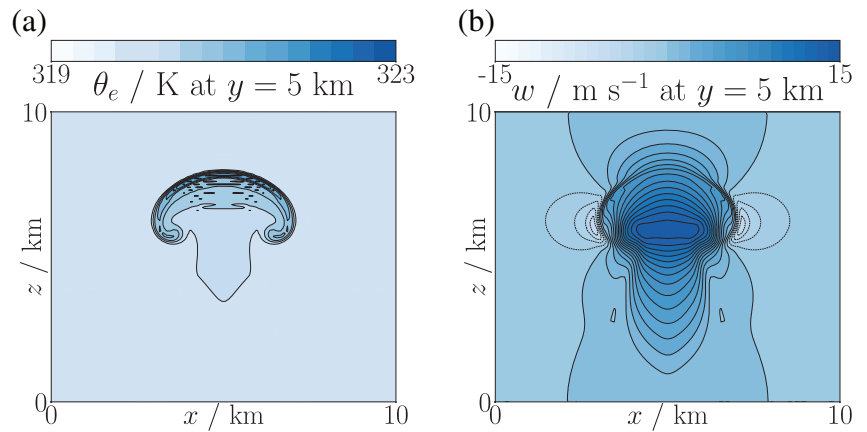
FIGURE 10 The fields output along $y = 5$ km from the $k = 1$ next-to-lowest order space simulation at $t = 1,000$ s of the three-dimensional simulation of a rising thermal in a saturated atmosphere. (a) θ_e and (b) vertical velocity w fields for a simulation using grid spacing $\Delta x = 200$ m and a time step $\Delta t = 1$ s. The θ_e field is contoured every 0.25 K with the 320-K contour omitted, whilst the contour spacing for the w field is 1 m·s^{-1}. Cross-sections are shown at $y = 5$ km, with values plotted on the lower side of the plane. As with the two-dimensional case, a second plume can be seen forming at the top of the primary plume
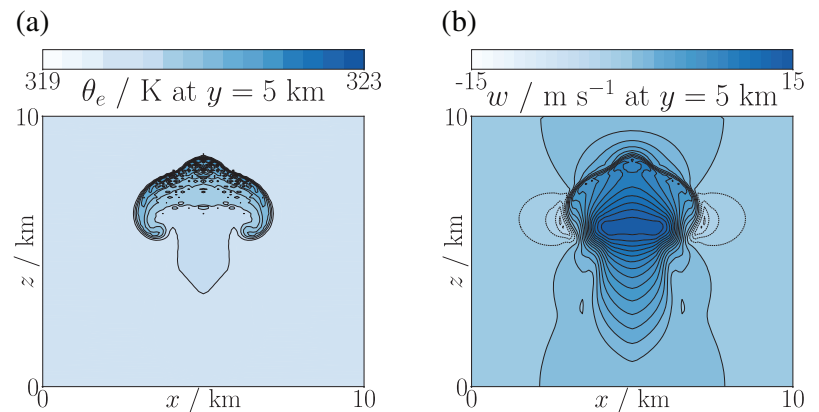
As in Section 5.1, we see a secondary plume beginning to form at the top of the rising thermal in the $k = 1$ case.

5.5 | Moist baroclinic wave

The final test case that we present is the moist baroclinic wave outlined in Ullrich *et al.*, (2015). This is the only test featuring the Coriolis force, although rain is again neglected. A three-dimensional channel of length $L = 40,000$ km in the x-direction (in which the domain is periodic), width $W = 6,000$ km in the y-direction, and height $H = 30$ km in the z-direction is used. The walls at the y and z boundaries of the domain are rigid, with no flow through them.

This test case uses initial conditions that are analytically in thermal wind balance. In Ullrich *et al.*, (2015), the vertical coordinate used is a pressure coordinate η. This is used to define the background zonal wind u, the geopotential Φ, the virtual temperature T_v, and the specific humidity q according to the following equations:

$$u = -u_0 \sin^2\left(\frac{\pi y}{W}\right) \ln \eta \exp\left[-\left(\frac{\ln \eta}{b}\right)^2\right], \quad (41a)$$

$$\Phi = \frac{T_0 g}{\Gamma}(1 - \eta^{R_d \Gamma/g}) + \frac{f_0 u_0}{2}$$
$$\times \left[y - \frac{W}{2} - \frac{W}{2\pi} \sin\left(\frac{2\pi y}{W}\right)\right] \ln \eta \exp\left[-\left(\frac{\ln \eta}{b}\right)^2\right], \quad (41b)$$

$$T_v = T_0 \eta^{R_d \Gamma/g} + \frac{f_0 u_0}{2R_d}\left[y - \frac{W}{2} - \frac{W}{2\pi} \sin\left(\frac{2\pi y}{W}\right)\right]$$
$$\times \left[\frac{2}{b^2}(\ln \eta)^2 - 1\right] \exp\left[-\left(\frac{\ln \eta}{b}\right)^2\right], \quad (41c)$$

$$q = \frac{q_0}{2} \exp\left[-\left(\frac{y}{\Delta y_w}\right)^4\right]\begin{cases} 1 + \cos\left[\frac{\pi(1-\eta)}{1-\eta_w}\right] & \eta \geq \eta_w, \\ 0 & \text{otherwise.} \end{cases}$$
$$(41d)$$

The constants take the values $\Gamma = 0.005$ K·m^{-1}, $f_0 = 2.00 \times 10^{-6}$·s^{-1}, $a = 6.37 \times 10^6$ m, $T_0 = 288$ K, $u_0 = 35$ m·s^{-1}, $b = 2$, $\Delta y_w = 3.2 \times 10^6$ m, and $\eta_w = 0.3$. We use a slightly lower value than Ullrich *et al.*, (2015) of $q_0 = 0.016$ to prevent our model being too close to saturation initially.

This initial condition is converted to our prognostic variables using the Newton iteration procedure suggested in Ullrich *et al.*, (2015) to find η. Using the requirement

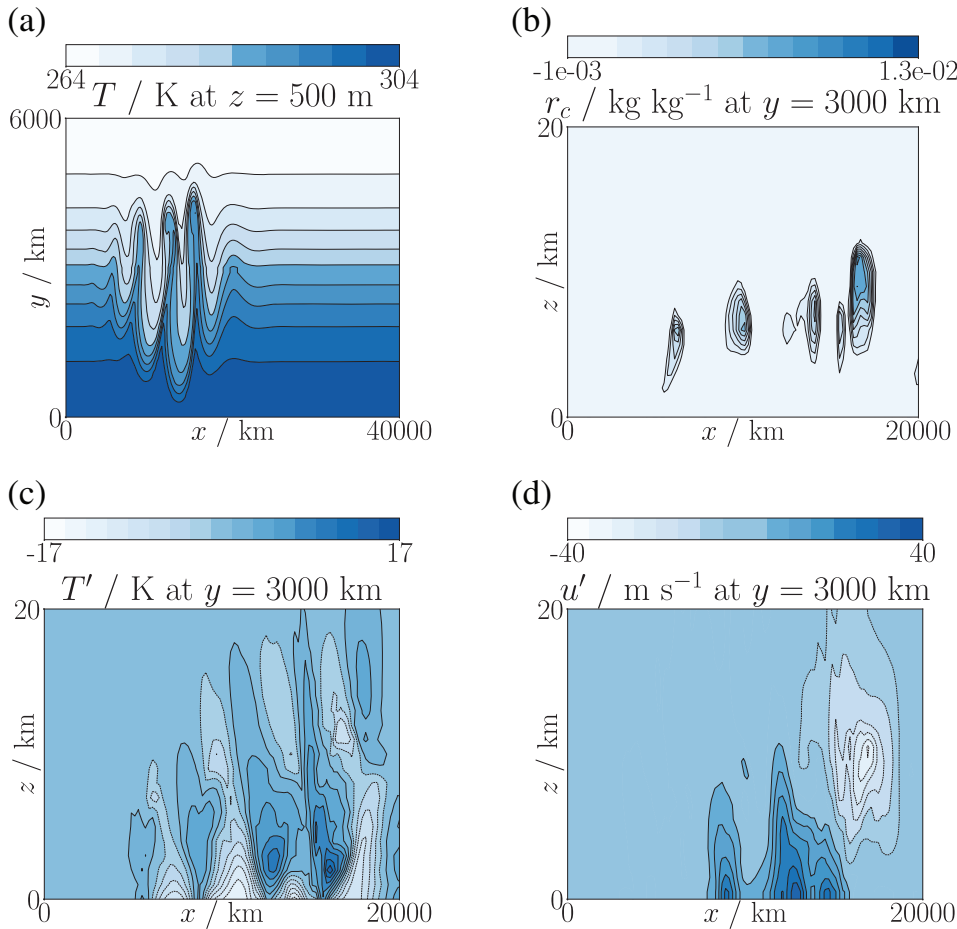(a)

(b)

(c)

(d)

FIGURE 11   Cross-sections of fields at $t = 12$ days from the moist baroclinic wave test case using the $k = 0$ spaces. The grid sizes used were $\Delta x = \Delta y = 200$ km and $\Delta z = 1$ km, with $\Delta t = 300$ s. Shown are (a) the T field on $z = 500$ m contoured every 4 K, (b) the r_c field on $y = 3{,}000$ km contoured every 10^{-3} kg·kg^{-1} with the zero contour omitted, (c) the perturbed temperature field on $y = 3{,}000$ km with contours every 2 K, and (d) the perturbed zonal wind u', with contours spaced by 5 m·s^{-1} and the zero contour omitted. Values shown on the $y = 3{,}000$ km plane are computed from the lower side of the plane

that $\Phi = gz$ and taking $\eta \in V_\theta$, the procedure uses

$$\eta^{n+1} = \eta^n - \frac{\Phi(\eta^n) - gz}{T_v(\eta^n) - R_d/\eta^n}. \tag{42}$$

This η field is then used to compute u, T_v, and q. In order to convert T_v and q into θ_{vd} and r_v, we use the definitions of θ_{vd}, $\eta = p/p_s$ for $p_s = 10^5$ Pa, and the equations

$$T = \frac{T_v}{1 + q(R_v/R_d - 1)} \tag{43}$$

and

$$r_v = \frac{q}{1 - q}. \tag{44}$$

These θ_{vd} and r_v fields are used to compute the ρ_d field that provides hydrostatic balance for the background state using the procedure outlined in Appendix B.

The perturbation is added to the x-component of the background velocity, which we will denote as $\bar{u}$, so that $u = \bar{u} + u'$. With $x_c = 2 \times 10^6$ m, $y_c = 2.5 \times 10^6$, $L_p = 6 \times 10^5$, $u_p = 1$ m·s^{-1}, we use

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}, \tag{45}$$

to obtain

$$u' = u_p \exp\left[-\left(\frac{r}{L_p}\right)^2\right]. \tag{46}$$

Figures 11 and 12 show fields from this test case at $t = 12$ days for the $k = 0$ and $k = 1$ spaces, respectively. For the $k = 0$ configuration, $\Delta x = \Delta y = 200$ km and $\Delta z = 1$ km, whilst for $k = 1$ these were $\Delta x = \Delta y = 250$ km and $\Delta z = 1.5$ km. For both simulations, $\Delta t = 300$ s. In the $k = 1$ simulation, the baroclinic wave becomes much stronger than in the $k = 0$ simulation. As the wave develops, the maxima and minima in the temperatures in the $k = 1$ case are higher than for the $k = 0$ lowest-order spaces. When these minima coincide with regions close to water-vapour saturation, more condensation occurs. This releases latent heat and strengthens the baroclinic wave, reinforcing the behaviour.

6 | CONCLUSIONS

This article represents the first application of the compatible finite-element framework of Cotter and Shipton (2012) to the moist atmosphere. These methods are promising

FIGURE 12 Cross-sections of fields at $t = 12$ days from the moist baroclinic wave test case using the $k = 1$ spaces. The grid sizes used were $\Delta x = \Delta y = 250$ km and $\Delta z = 1.5$ km, with $\Delta t = 300$ s. Plots shown are the same as in Figure 11, but note the different scale in the zonal wind perturbation plot (panel d). With the $k = 1$ spaces, the baroclinic instability is stronger than with the $k = 0$ spaces, more cloud appears to form, and the wave is less linear
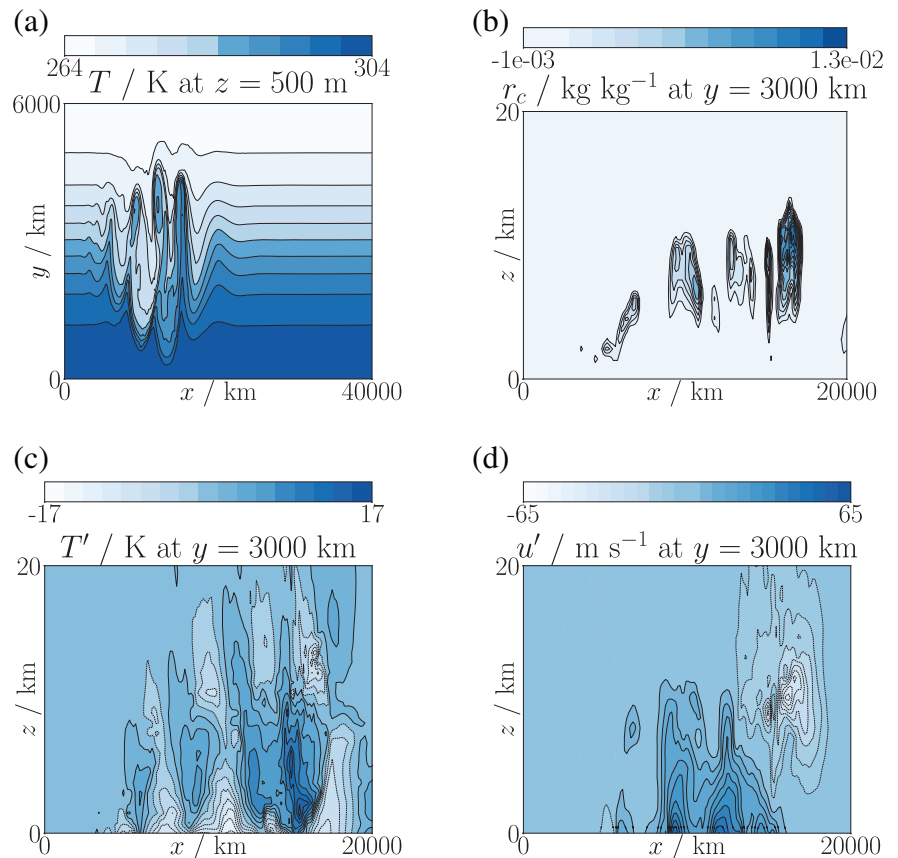
for avoiding the data-communication bottleneck associated with the convergence of points at the poles in a latitude–longitude mesh of the sphere. The work presented here marks a significant step in the development of these methods for use in numerical weather prediction.

We applied a discretisation to the moist compressible Euler equations, with configurations for two sets of function spaces with different finite-element orders. This builds upon the work of Cotter and Shipton (2012), Natale *et al.*, (2016), Yamazaki *et al.*, (2017), and Bendall *et al.*, (2019). The model configuration for each of the sets of spaces has been described, detailing the discrete equations that are solved and the transport schemes used, including slope limiters that can be used with moisture variables. A discussion is given of the parametrisation of the moist processes and how these are coupled to the description of the resolved flow. The performance of the model is displayed through the presentation of five test cases, including two new ones which we have introduced.

Our results demonstrate that this framework and our discretisation strategies are robust enough to deal with the more strenuous demands of the diabatic atmosphere, which is promising for the future development of such models. The results also offer some insight into how such discretisations may depend upon the order of the finite element.

Planned future work will extend this discretisation to spherical domains, and test the numerical properties of the different choices of finite-element spaces. More physics parametrisations should be included and the procedure used in the linear solve stage of the model, will be thoroughly detailed along with its performance.

ORCID
Thomas M. Bendall https://orcid.org/0000-0002-6546-8400
Colin J. Cotter https://orcid.org/0000-0001-7962-8324

REFERENCES
Arnold, D.N. and Logg, A. (2014) Periodic table of the finite elements. *SIAM News*, 47(9), 212.
Balay, S., Gropp, W.D., McInnes, L.C. and Smith, B.F. (1997). Efficient management of parallelism in object-oriented numerical

software libraries, *Modern Software Tools for Scientific Computing*, pp. 163–202. Boston, MA: Birkhäuser.

Bendall, T.M., Cotter, C.J. and Shipton, J. (2019) The 'recovered space' advection scheme for lowest-order compatible finite-element methods. *Journal of Computational Physics*, 390, 342–358.

Bercea, G.-T., McRae, A.T.T., Ham, D.A., Mitchell, L., Rathgeber, F., Nardi, L., Luporini, F. and Kelly, P.H.J. (2016) A structure-exploiting numbering algorithm for finite elements on extruded meshes, and its performance evaluation in Firedrake. *Geoscientific Model Development*, 9(10), 3803–3815.

Bryan, G.H. and Fritsch, J.M. (2002) A benchmark simulation for moist nonhydrostatic numerical models. *Monthly Weather Review*, 130(12), 2917–2928.

Cotter, C.J. and Kuzmin, D. (2016) Embedded discontinuous Galerkin transport schemes with localised limiters. *Journal of Computational Physics*, 311, 363–373.

Cotter, C.J. and Shipton, J. (2012) Mixed finite elements for numerical weather prediction. *Journal of Computational Physics*, 231(21), 7076–7091.

Cotter, C.J. and Thuburn, J. (2014) A finite element exterior calculus framework for the rotating shallow-water equations. *Journal of Computational Physics*, 257, 1506–1526.

Emanuel, K.A. (1994) *Atmospheric Convection*. Oxford: Oxford University Press.

Gibson, T.H., McRae, A.T.T., Cotter, C.J., Mitchell, L. and Ham, D.A. (2019) *Compatible Finite Element Methods for Geophysical Flows: Automation and Implementation Using Firedrake.* Cham, Switzerland: Springer Nature.

Gibson, T.H., Mitchell, L., Ham, D.A. and Cotter, C.J. (2020) Slate: extending Firedrake's domain-specific abstraction to hybridized solvers for geoscience and beyond. *Geoscientific Model Development.*, 13(2), 735–761.

Grabowski, W.W. and Clark, T.L. (1991) Cloud–environment interface instability: rising thermal calculations in two spatial dimensions. *Journal of the Atmospheric Sciences*, 48(4), 527–546.

Homolya, M. and Ham, D.A. (2016) A parallel edge orientation algorithm for quadrilateral meshes. *SIAM Journal on Scientific Computing*, 38(5), S48–S61.

Klemp, J.B. and Wilhelmson, R.B. (1978) The simulation of three-dimensional convective storm dynamics. *Journal of the Atmospheric Sciences*, 35(6), 1070–1096.

Kuzmin, D. (2010) A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods. *Journal of Computational and Applied Mathematics*, 233(12), 3077–3085.

LeVeque, R.J. (1996) High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis*, 33(2), 627–665.

McRae, A.T.T., Bercea, G.-T., Mitchell, L., Ham, D.A. and Cotter, C.J. (2016) Automated generation and symbolic manipulation of tensor product finite elements. *SIAM Journal on Scientific Computing*, 38(5), S25–S47.

Melvin, T., Benacchio, T., Shipway, B., Wood, N., Thuburn, J. and Cotter, C. (2019) A mixed finite-element, finite-volume, semi-implicit discretization for atmospheric dynamics: Cartesian geometry. *Quarterly Journal of the Royal Meteorological Society*, 145(724), 2835–2853.

Milbrandt, J.A. and McTaggart-Cowan, R. (2010) Sedimentation-induced errors in bulk microphysics schemes. *Journal of the Atmospheric Sciences*, 67(12), 3931–3948.

Natale, A., Shipton, J. and Cotter, C.J. (2016) Compatible finite-element spaces for geophysical fluid dynamics. *Dynamics and Statistics of the Climate System*, 1(1), 1–31.

Paluch, I.R. (1979) The entrainment mechanism in Colorado cumuli. *Journal of the Atmospheric Sciences*, 36(12), 2467–2478.

Petty, G.W. and Huang, W. (2011) The modified gamma size distribution applied to inhomogeneous and nonspherical particles: key relationships and conversions. *Journal of the Atmospheric Sciences*, 68(7), 1460–1473.

Rathgeber, F., Ham, D.A., Mitchell, L., Lange, M., Luporini, F., McRae, A.T.T., Bercea, G.-T., Markall, G.R. and Kelly, P.H.J. (2017) Firedrake: automating the finite-element method by composing abstractions. *ACM Transactions on Mathematical Software (TOMS)*, 43(3), 24.

Raviart, P. and Thomas, J. (1977). A mixed finite-element method for 2nd order elliptic problems. In:, *Mathematical Aspects of Finite Element Methods*, pp. 292–315. Berlin, Heidelberg: Springer.

Rutledge, S.A. and Hobbs, P. (1983) The mesoscale and microscale structure and organization of clouds and precipitation in mid-latitude cyclones. VIII: a model for the "seeder-feeder" process in warm-frontal rainbands. *Journal of the Atmospheric Sciences*, 40(5), 1185–1206.

Shu, C.-W. and Osher, S. (1988) Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2), 439–471.

Skamarock, W.C. and Klemp, J.B. (1994) Efficiency and accuracy of the Klemp–Wilhelmson time-splitting technique. *Monthly Weather Review*, 122(11), 2623–2630.

Soong, S.-T. and Ogura, Y. (1973) A comparison between axisymmetric and slab-symmetric cumulus cloud models. *Journal of the Atmospheric Sciences*, 30(5), 879–893.

Staniforth, A. and Thuburn, J. (2012) Horizontal grids for global weather and climate prediction models: a review. *Quarterly Journal of the Royal Meteorological Society*, 138(662), 1–26.

Tetens, O. (1930) Uber einige meteorologische Begriffe. *Zeitschrift fur Geophysik*, 6, 297–309.

Thuburn, J. (2017) Use of the Gibbs thermodynamic potential to express the equation of state in atmospheric models. *Quarterly Journal of the Royal Meteorological Society*, 143(704), 1185–1196.

Ullrich, P.A., Reed, K.A. and Jablonowski, C. (2015) Analytical initial conditions and an analysis of baroclinic instability waves in f- and β-plane 3D channel models. *Quarterly Journal of the Royal Meteorological Society*, 141(693), 2972–2988.

Walters, D., Wood, N., Vosper, S. and Milton, S. (2014) *ENDGame: A New Dynamical Core for Seamless Atmospheric Prediction*. Exeter, UK: Met Office. Technical Report.

Yamazaki, H., Shipton, J., Cullen, M.J.P., Mitchell, L. and Cotter, C.J. (2017) Vertical slice modelling of nonlinear Eady waves using a compatible finite-element method. *Journal of Computational Physics*, 343, 130–149.

APPENDIX

Notation, thermodynamic variables, and constants

The values or definitions of some thermodynamic constants or variables used in the text are shown in Table A1.

HYDROSTATIC BALANCE ROUTINES

For many test cases, the background or initial state of the model will be in hydrostatic balance. The procedure that we use to obtain a discrete hydrostatic balance is based on that presented in Natale et al., (2016). Given a boundary condition for the pressure and a θ_d field, this procedure finds the ρ_d that gives rise to zero vertical accelerations. Here, we describe two developments to this: the extension of the routine of Natale et al., (2016) to use a hybridisation technique and the treatment of cases in which thermodynamic and moist variables and the prognostic variables need to be found.

Hybridised hydrostatic balance

The hybridised hydrostatic balance system uses a similar approach to that described in Section 3.5. As in that case, the variables are expressed in discontinuous spaces and Lagrange multipliers ℓ are introduced in a trace space on the horizontal facets to provide continuity constraints. We also introduce $\mathring{V}_v$, the subspace of the fully discontinuous velocity space $\hat{V}_v$ with zero flow normal to the boundaries. The problem is then to find $(v_{\text{trial}}, \rho_{\text{trial}}, \ell_{\text{trial}}) \in (\mathring{V}_v, V_\rho, V_{\text{trace}})$, such that, for all $(\psi, \phi, \lambda) \in (\mathring{V}_v, V_\rho, V_{\text{trace}})$,

$$\int_\Omega \psi \cdot v_{\text{trial}} \, dx$$
$$-\int_\Omega c_{\text{pd}} \hat{\Pi}(\rho_d, \theta_{\text{vd}}) \nabla \cdot \left(\frac{\theta_{\text{vd}} \psi}{1+r_t} \right) dx + \int_\Gamma [\![\psi]\!]_n \, dS \quad \text{(B1a)}$$
$$= -\int_{\partial \Omega_0} \frac{c_{\text{pd}} \theta_{\text{vd}}}{1+r_t} (\psi \cdot \hat{n}) \Pi_0 \, dS - \int_\Omega g \psi \cdot \hat{k} \, dx,$$

$$\int_\Omega \phi c_{\text{pd}} \nabla \cdot \left(\frac{\theta_{\text{vd}} v_{\text{trial}}}{1+r_t} \right) dx = 0, \quad \text{(B1b)}$$

$$\int_\Gamma \lambda [\![v_{\text{trial}}]\!]_n \, dS = 0, \quad \text{(B1c)}$$

where $\partial \Omega_0$ is the boundary at which the condition that $\Pi = \Pi_0$ is to be satisfied and $\hat{\Pi}$ is a linearisation of Π. Equation B1a can be manipulated into a single elliptic equation for ℓ, which can be inverted efficiently in each column of the mesh. In fact, it turns out that ℓ is an approximation for $c_{\text{pd}} \theta_{\text{vd}} \Pi / (1 + r_t)$ on the horizontal facets of the mesh. Once ℓ is determined, v and ρ_d can be solved for via local back-substitution in each cell of the mesh. Continuity is restored in v again by using the recovery operator. By hybridising, we avoid solving a larger, more ill-conditioned mixed system and instead invert a condensed elliptic problem. The implementation of this procedure is described by Gibson et al., (2020), with the application taking the form of a custom Python-base preconditioner conforming to standard PETSc library options (Balay et al., 1997; Petty and Huang, 2011), as described in Gibson et al., (2020). The advantage is that the solver options for solving the condensed system arising from reducing Equation B1a can be easily updated, even when nested inside a nonlinear method.

Saturated conditions

This setup involves initial conditions like those in the moist benchmark of Bryan and Fritsch (2002). The problem is to find ρ_d, θ_{vd}, and r_v given θ_e, r_t, and a boundary condition on the pressure. Assuming the absence of rain, $r_v = r_{\text{sat}}$, and $r_c = r_t - r_v$. The wet-equivalent potential temperature, θ_e, is a conserved quantity in reversible, moist adiabatic processes, that is, $D\theta_e / Dt = 0$. Following Emanuel (1994), for our equation set (Equation 2), θ_e can be written as

$$\theta_e := T \left(\frac{p_0}{p} \right)^{\frac{R_d}{c_{\text{pd}} + c_{\text{pl}} r_t}} (\mathcal{H})^{\frac{-r_v R_v}{c_{\text{pd}} + c_{\text{pl}} r_t}} e^{\frac{L_v(T) r_v}{(c_{\text{pd}} + c_{\text{pl}} r_t) T}}, \quad \text{(B2)}$$

which, with $\mathcal{H} = 1$ for a saturated atmosphere, is the same as used by Bryan and Fritsch (2002) and derived in the appendix of Paluch (1979) from the second law of thermodynamics. The saturation mixing ratio is given by Equation 24.

The challenge is to obtain the θ_{vd} and ρ_d fields that satisfy the specified θ_e field whilst ensuring that $r_v = r_{\text{sat}}$. We use an initial guess for θ_{vd} and feed it to Equation B1a to generate a guess for ρ_d. This density is then converted into V_θ, before a nested fixed-point iteration-style procedure is used to invert $\theta_e(\theta_{\text{vd}}, \tilde{\rho}_d, r_v)$ and $r_{\text{sat}}(\theta_{\text{vd}}, \tilde{\rho}_d, r_v)$, to obtain θ_{vd} and r_v.

Let l, m, and n count the number of iterations to find ρ_d, θ_{vd}, and r_v respectively. These form nested loops, such that the outer loop uses the latest approximations of θ_{vd} and r_v with Equation B1a to obtain $\rho_d^{(l+1/2)}$. We then combine this with the previous value to update the approximation to ρ_d, using

$$\rho_d^{(l+1)} = (1 - \delta) \rho_d^{(l)} + \delta \rho_d^{(l+1/2)}, \quad \text{(B3)}$$

where $\delta = 0.8$. The next loop finds θ_{vd} using

$$\theta_{\text{vd}}^{(m+1/2)} = \frac{\theta_{\text{vd}}^{(m)} \theta_e}{\theta_e(\theta_{\text{vd}}^{(m)}, \tilde{\rho}_d^{(l)}, r_v^{(n)})}, \quad \text{(B4)}$$

$$\theta_{\text{vd}}^{(m+1)} = (1 - \delta) \theta_{\text{vd}}^{(m)} + \delta \theta_{\text{vd}}^{(m+1/2)}, \quad \text{(B5)}$$

Description	Variable	Value or definition
Specific heat capacity of dry air at constant V	c_{vd}	$717 \, \text{J·kg}^{-1} \, \text{·K}^{-1}$
Specific heat capacity of dry air constant p	c_{pd}	$1004.5 \, \text{J·kg}^{-1} \, \text{·K}^{-1}$
Specific heat capacity of water vapour constant V	c_{vv}	$1424 \, \text{J·kg}^{-1} \, \text{·K}^{-1}$
Specific heat capacity of water vapour constant p	c_{pv}	$1885 \, \text{J·kg}^{-1} \, \text{·K}^{-1}$
Specific heat capacity of liquid water constant p	c_{pl}	$4186 \, \text{J·kg}^{-1} \, \text{·K}^{-1}$
Specific heat capacity of moist air constant V	c_{vml}	$c_{vd} + r_v c_{vv} + (r_c + r_r)c_{pl}$
Specific heat capacity of moist air constant p	c_{pml}	$c_{pd} + r_v c_{pv} + (r_c + r_r)c_{pl}$
Specific gas constant for dry air	R_d	$287 \, \text{J·kg}^{-1} \, \text{·K}^{-1}$
Specific gas constant for water vapour	R_v	$461 \, \text{J·kg}^{-1} \, \text{·K}^{-1}$
Specific gas constant for moist air	R_m	$R_d + r_v R_v$
Reference Latent heat of vaporisation of water at T_R	L_{vR}	$2.5 \times 10^6 \, \text{J·kg}^{-1}$
Latent heat of vaporisation of water	L_v	$L_{vR} - (c_{pl} - c_{pv})(T - T_R)$
Reference temperature	T_R	$273.15 \, \text{K}$
Reference pressure	p_R	$10^5 \, \text{Pa}$
Constant in Tetens' formula	C_0^{sat}	$610.9 \, \text{Pa}$
Constant in Tetens' formula	C_1^{sat}	-17.27
Constant in Tetens' formula	C_2^{sat}	$35.86 \, \text{K}$
Ratio of R_d to R_v	ϵ	0.623
Ratio of R_d to c_{pd}	κ	$2/7$

TABLE A1 Thermodynamic values and constants used in the text

whilst the inner loop computes

$$r_v^{(n+1/2)} = r_{sat}(\theta_{vd}^{(m)}, \tilde{\rho}_d^{(l)}, r_v^{(n)}), \quad \text{(B6)}$$

$$r_v^{(n+1)} = (1 - \delta)r_v^{(m)} + \delta r_v^{(m+1/2)}, \quad \text{(B7)}$$

where θ_e without arguments denotes the specified value. These loops are iterated until $\theta_e(\theta_{vd}, \tilde{\rho}_d, r_v)$ converges to the specified value to some tolerance.

Unsaturated conditions

We now discuss how to find the prognostic thermodynamic variables given θ_d and the relative humidity $\mathcal{H}$, such as the case in Section 5.3 or Grabowski and Clark (1991). The relative humidity is related to r_v by

$$\mathcal{H} = \frac{r_v}{r_{sat}} \left(\frac{1 + r_{sat}/\epsilon}{1 + r_v/\epsilon} \right). \quad \text{(B8)}$$

As in Appendix B2, we use what can be thought of as a nested iterative procedure. Counting the latest approximations of ρ_d, θ_{vd}, and r_v with l, m, and n, the outer loop uses $\theta_{vd}^{(m)}$ and $r_v^{(n)}$ in the hydrostatic balance

equation (Equation B1a) to determine $\rho_d^{(l+1/2)}$. Again, the next value of ρ_d is given by

$$\rho_d^{(l+1)} = (1 - \delta)\rho_d^{(l)} + \delta\rho_d^{(l+1/2)}. \quad \text{(B9)}$$

There is only one inner loop in this case, so that $m = n$. The new value of r_v is found by rearranging Equation B8 so that

$$r_v^{(m+1/2)} = \frac{\mathcal{H} r_{sat}(\theta_{vd}^{(m)}, \tilde{\rho}_d^{(l)}, r_v^{(m)})}{1 + (1 - \mathcal{H})r_{sat}(\theta_{vd}^{(m)}, \tilde{\rho}_d^{(l)}, r_v^{(m)})/\epsilon}, \quad \text{(B10)}$$

$$r_v^{(m+1)} = (1 - \delta)r_v^{(m)} + \delta r_v^{(m+1/2)}, \quad \text{(B11)}$$

where $\mathcal{H}$ is the specified value of the relative humidity. The final step is to use the specified value of θ_d to obtain

$$\theta_{vd}^{(m)} = \theta_d(1 + r_v^{(m)}/\epsilon). \quad \text{(B12)}$$

This iterative process continues until $\mathcal{H}(\theta_{vd}^{(m)}, \rho_d^{(l)}, r_v^{(m)})$ has converged to its specified value to some tolerance.