Operator-Split Runge–Kutta–Rosenbrock Methods for Nonhydrostatic Atmospheric Models

PAUL ULLRICH AND CHRISTIANE JABLONOWSKI

Department of Atmospheric, Oceanic and Space Sciences, University of Michigan, Ann Arbor, Ann Arbor, Michigan

(Manuscript received 6 December 2010, in final form 18 August 2011)

ABSTRACT

This paper presents a new approach for discretizing the nonhydrostatic Euler equations in Cartesian geometry using an operator-split time-stepping strategy and unstaggered upwind finite-volume model formulation. Following the method of lines, a spatial discretization of the governing equations leads to a set of coupled nonlinear ordinary differential equations. In general, explicit time-stepping methods cannot be applied directly to these equations because the large aspect ratio between the horizontal and vertical grid spacing leads to a stringent restriction on the time step to maintain numerical stability. Instead, an A-stable linearly implicit Rosenbrock method for evolving the vertical components of the equations coupled to a traditional explicit Runge–Kutta formula in the horizontal is proposed. Up to third-order temporal accuracy is achieved by carefully interleaving the explicit and linearly implicit steps. The time step for the resulting Runge–Kutta–Rosenbrock–type semi-implicit method is then restricted only by the grid spacing and wave speed in the horizontal. The high-order finite-volume model is tested against a series of atmospheric flow problems to verify accuracy and consistency. The results of these tests reveal that this method is accurate, stable, and applicable to a wide range of atmospheric flows and scales.

1. Introduction

One reason explicit time-stepping schemes are desirable in atmospheric models is locality of data, which allows the equations of motion to be evaluated with minimal communication between neighboring elements. In the context of large-scale parallel systems, this benefit means that communication between processors is significantly less than with equivalent implicit methods. Nonetheless, purely explicit methods have strict time-step restrictions that are required for stability. The fast-moving waves in the governing equations largely determine this time-step restriction, although they may possess little physical significance. In general, the maximum stable time step for explicit time discretizations is determined by the dimensionless Courant–Friedrichs–Lewy (CFL) number, which takes the form

$$\nu = \frac{c_{\max}\Delta t}{\Delta x}, \tag{1}$$

where $c_{\max}$ denotes the maximum wave speed of the system, Δx is the minimum grid spacing, and Δt is the maximum stable time step. Most explicit time-stepping methods are limited to $\nu \lesssim 1$. In atmospheric flows, sound waves are the fastest propagating wave modes, with an average speed of 340 m s^{-1} at sea level. At $\Delta x \approx$ 110-km resolution [which corresponds to about a 1° grid spacing at the equator or the spectral triangular truncation T85, as argued by Williamson (2008)], this leads to an explicit time-step restriction of about 5 min. Contrasting this against a vertical discretization with a minimum near-surface grid spacing of about 100 m, the maximum time-step restriction is merely $\Delta t \lesssim 0.3$ s, 1000 times smaller than the horizontal time step. With our current computing power, atmospheric modeling would be effectively impossible at such a time step.

Various methods have been developed to deal with the computational restrictions introduced by fast waves in the atmosphere. The first method arises from a modification of the equation set to ''filter'' out fast-moving waves. At large horizontal scales, hydrostatic models [including Taylor et al. (2008) and Lin (2004)] are usually employed to remove the acceleration of the vertical velocity from the system. This approach eliminates vertically

Corresponding author address: Paul Ullrich, Department of Atmospheric, Oceanic and Space Sciences, Space Research Building, University of Michigan, 2455 Hayward St., Ann Arbor, MI 48109.
E-mail: paullric@umich.edu

DOI: 10.1175/MWR-D-10-05073.1

propagating sound waves but reaches its limitations when the grid size is reduced to nonhydrostatic scales around 10 km. Motions at this scale can be dominated by large vertical velocities, and so the vertical acceleration term cannot be neglected. Further, the dispersion relation of the hydrostatic equation set reveals that the phase speed of gravity wave modes can be overestimated at these scales, as compared to the full nonhydrostatic equations (Durran 1999). Alternatively, so-called sound-proof systems of equations, including the Boussinesq equations, the anelastic system of Ogura and Phillips (1962), the pseudo-incompressible system of Durran (1989), and the unified approximation of Arakawa and Konor (2009), have been successfully used in models by removing sound waves from the governing equations. Nonetheless, it remains an open question as to whether these modified systems are valid on all scales [see, e.g., the discussion in Davies et al. (2003) and Klein et al. (2010)].

The second method for dealing with computationally fast waves relies on numerical methods that treat these modes in a stable manner. In particular, we focus on methods that integrate the full nonhydrostatic equation set but introduce a splitting strategy to deal with fast wave modes. These approaches are generally referred to as ''operator-split methods.'' Operator-split methods have been in use for atmospheric models for quite some time, beginning with the first semi-implicit methods of Kwizak and Robert (1971). Since then, semi-implicit methods have been used in atmospheric models at practically all scales (see, e.g., Bonaventura 2000; Giraldo 2005; Restelli and Giraldo 2009). Closely related to semi-implicit methods are split-explicit and fractional step techniques. These methods are similar but instead combine explicit operators, generally splitting on the slow and fast waves. These methods were originally developed by Gadd (1978) for atmospheric models, but they continue to be in use today. The Weather Research and Forecasting model (WRF; Skamarock and Klemp 2008), for example, uses both semi-implicit and explicit splitting, utilizing implicit integration for vertically propagating waves and a split-explicit technique for fast waves in the horizontal.

In this paper, we introduce a new time discretization for models that split the temporal and spatial derivatives using the method of lines (Schiesser 1991; Schiesser and Griffiths 2009). The proposed method offers a simple framework for achieving up to third-order temporal accuracy in a semi-implicit scheme while maintaining computational efficiency. In particular, this approach is designed to outperform a purely explicit formulation for models with a horizontal–vertical aspect ratio greater than 5. Following an operator-split Runge–Kutta–Rosenbrock (RKR) strategy, which combines an explicit Runge–Kutta (RK) method with a linearly implicit Rosenbrock step (Rosenbrock 1963), we obtain a method whose maximum stable time step is constrained only by the horizontal CFL number. To maximize efficiency on parallel systems, the splitting is performed on the horizontal and vertical components of the governing equations so that the implicit solve occurs without requiring off-processor communication (here we assume that vertical columns are not distributed among multiple processors). This strategy differs from the approach of St-Cyr and Neckels (2009), for instance, who apply the implicit solve over the entire domain. In this paper, the RKR time discretization strategy is demonstrated using a high-order finite-volume method in 2D and 3D so as to verify accuracy and stability, but this approach is easily extended to other methods that independently discretize space and time. In particular, discontinuous Galerkin methods, which have been growing in popularity in recent years, would be excellent candidates for use with these time integrators. The approach presented herein is valid for all horizontal scales and hence may be especially well suited for models that utilize adaptively refined meshes with scale differences.

In section 2, we introduce the full nonhydrostatic fluid equations and explain how to incorporate terrain-following coordinates. The RKR discretization is introduced in section 3, wherein we present a first-order, a second-order, and a third-order temporal discretization that is stable for high-order spatial discretizations. We will demonstrate these techniques using high-order finite-volume spatial discretizations, which are explained in section 4, followed by numerical results in section 5. Our conclusions and future work are given in section 6.

2. The nonhydrostatic fluid equations in Cartesian coordinates

We utilize the shallow-atmosphere nonhydrostatic fluid equations written in terms of the conservative variables density ρ, momentum $\rho\mathbf{u}$ (where $\mathbf{u}$ is the 3D velocity vector), and potential temperature density $\rho\theta$. In vector form, these equations are written as follows:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{2}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p\mathcal{I}) = -\rho g \mathbf{e}_z - f \mathbf{e}_z \times (\rho \mathbf{u}), \quad \text{and} \tag{3}$$

$$\frac{\partial \rho \theta}{\partial t} + \nabla \cdot (\rho \theta \mathbf{u}) = 0. \tag{4}$$

Here, $\otimes$ denotes the tensor (outer) product, $\mathcal{I}$ denotes the identity matrix, $\mathbf{e}_z$ is the basis vector in the z direction, g is

TABLE 1. List of parameters and physical constants used in this paper.

Parameter	Description	Control value
A	Radius of the earth	$6.371\ 22 \times 10^6$ m
Ω	Rotational speed of the earth	7.292×10^{-5} s^{-1}
G	Gravity	9.806 16 m s^{-2}
p_0	Background surface pressure	1000 hPa
c_p	Specific heat capacity of dry air at constant pressure	1004.5 J kg^{-1} K^{-1}
c_v	Specific heat capacity of dry air at constant volume	717.5 J kg^{-1} K^{-1}
R_d	Ideal gas constant of dry air	287.0 J kg^{-1} K^{-1}
γ	Ratio of specific heats c_p/c_v	1.4

gravity, and f is the Coriolis parameter. The pressure p in the momentum equation is related to the potential temperature density via the equation of state,

$$p = p_0 \left[\frac{R_d(\rho\theta)}{p_0} \right]^{c_p/c_v}. \tag{5}$$

A list and explanation of the constants used in this paper and their corresponding values can be found in Table 1. The second terms on the left-hand side of (2)–(4) are referred to as flux terms, because they determine the flow rate of the conservative state variables through the edges of a spatial region. The terms on the right-hand side of these equations are source terms. Nonhydrostatic mesoscale models that use a closely related equation set include the WRF (see, e.g., Skamarock and Klemp 2008) and the model by Ahmad and Lindeman (2007).

a. Reference profile splitting

A splitting of the hydrostatic and nonhydrostatic components of the thermodynamic variables is used that takes the form

$$\rho(\mathbf{x}, t) = \rho^h(\mathbf{x}) + \rho'(\mathbf{x}, t), \tag{6}$$

$$p(\mathbf{x}, t) = p^h(\mathbf{x}) + p'(\mathbf{x}, t), \quad \text{and} \tag{7}$$

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

where the quantities denoted by the superscript h are in local hydrostatic balance,

$$\frac{\partial p^h}{\partial z} = -\rho^h g, \tag{9}$$

and satisfy (5). This choice is desirable to remove errors in approximating the hydrostatic state of the atmosphere that could be responsible for significant generation of spurious vertical momentum. Under this splitting, the nonhydrostatic fluid equations (2)–(4) take the form

$$\frac{\partial \rho'}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) + \frac{\partial}{\partial z}(\rho w) = 0, \tag{10}$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial}{\partial x}(\rho u^2 + p') + \frac{\partial}{\partial y}(\rho uv) + \frac{\partial}{\partial z}(\rho uw) = -\frac{\partial p^h}{\partial x} + fv, \tag{11}$$

$$\frac{\partial \rho v}{\partial t} + \frac{\partial}{\partial x}(\rho uv) + \frac{\partial}{\partial y}(\rho v^2 + p') + \frac{\partial}{\partial z}(\rho vw) = -\frac{\partial p^h}{\partial y} - fu, \tag{12}$$

$$\frac{\partial \rho w}{\partial t} + \frac{\partial}{\partial x}(\rho uw) + \frac{\partial}{\partial y}(\rho vw) + \frac{\partial}{\partial z}(\rho w^2 + p') = -\rho' g, \tag{13}$$

and

$$\frac{\partial (\rho\theta)'}{\partial t} + \frac{\partial}{\partial x}(\rho\theta u) + \frac{\partial}{\partial y}(\rho\theta v) + \frac{\partial}{\partial z}(\rho\theta w) = 0. \tag{14}$$

Note that the hydrostatic background state can be chosen to vary spatially in the horizontal. This variation is included as a source term in the momentum evolution equations. This choice is made to avoid numerical errors in the calculation of the horizontal pressure gradient in the presence of steep topography.

b. Incorporating topography

When topography is present, terrain-following coordinates as introduced by Gal-Chen and Somerville (1975) (GS coordinates) are used to deform the computational domain to match the physical space. However, we do not modify the governing equations from the form (10)–(14) but instead make use of orthonormalization and deorthonormalization operators to accurately compute fluxes in the presence of topography (see section 4d). This approach is analogous to the treatment of edge fluxes that arises on unstructured grids. In addition to simplifying the arithmetic, this approach also avoids problems that may arise from the explicit computation of metric terms (Klemp et al. 2003).

A possible alternative to terrain-following coordinates are so-called shaved-cell methods (see, e.g., Adcroft et al.

1997), which remove the portions of a cell occupied by topography. Unfortunately, if terrain is accurately resolved, this approach will reduce the horizontal extent of an element, and hence the maximum allowable horizontal time step. Modified shaved cells that do not reduce the horizontal extent of cells could also be used, but this technique may significantly degrade the accuracy of the terrain discretization.

As has been shown by Schär et al. (2002), GS coordinates are suboptimal for atmospheric motions, because they tend to introduce spurious grid artifacts above rough terrain. More accurate numerical methods may assist in reducing these errors; in particular, for the tests presented in this paper, these coordinates have been shown to be sufficiently smooth. In the future, this choice of terrain-following coordinate in our model will likely be revisited.

3. RKR schemes

The method of lines approach is one of the most popular methods for constructing high-order finite-volume methods that are applicable to general systems of partial differential equations (PDEs) (Jameson et al. 1981; McCorquodale and Colella 2011). Under this framework, the spatial terms, including the flux and source terms, are discretized first, leading to a system of ordinary differential equations (ODEs) for the state variables within each grid cell. This system is then discretized by means of choosing an appropriate time-stepping scheme. The time-stepping scheme must be chosen so that the eigenvalues of the spatial operator fit within the scheme's stability region. Explicit schemes are generally computationally inexpensive but possess a restricted stability region, whereas implicit schemes are more costly but possess a large stability region. However, different physical processes can have eigenvalues that have dramatically different structure, and so it may not be appropriate to use a single time-stepping method to integrate all terms of the ODE system.

a. The Runge–Kutta–Rosenbrock approach

Implicit–explicit (IMEX) methods represent a category of general-purpose schemes for ODEs that couple implicit and explicit time integration methods. These methods have been in use as early as the 1970s (e.g., Crouzeix 1980; Varah 1980). More recently, a family of implicit–explicit Runge–Kutta (IMEX-RK) schemes was collected into a general framework by Ascher et al. (1997) in their seminal paper. They showed that it is possible to achieve an essentially arbitrary order of accuracy by correctly interleaving explicit and implicit steps, although with increasing computational expense.

IMEX methods are usually applied to an ODE of the form

$$\frac{\partial \mathbf{q}}{\partial t} = \mathbf{f}(\mathbf{q}) + \mathbf{g}(\mathbf{q}), \tag{15}$$

where $\mathbf{q}$ is some state vector. In particular, we assume that the terms $\mathbf{f}(\mathbf{q})$ are not stiff; mathematically, one can think of this as saying that the eigenvalues of the operator $\mathbf{f}$ are close to the origin. On the other hand, the terms grouped under the $\mathbf{g}(\mathbf{q})$ operator are assumed stiff, containing eigenvalues that are potentially unbounded. This stiffness can originate from short time-scale behavior, such as chemistry or, as in our case, from geometric stiffness due to a discrepancy in grid spacing in different coordinate directions. Because we focus on the dynamical aspects of the atmosphere, $\mathbf{f}(\mathbf{q})$ is assumed to be some horizontal spatial discretization of the fluid equations with grid spacing Δx and $\mathbf{g}(\mathbf{q})$ is a vertical discretization with grid spacing $\Delta z \ll \Delta x$. The much smaller vertical scale leads to eigenvalues of $\mathbf{g}$ that are typically far from the origin. Herein, we will see that the terms $\mathbf{f}(\mathbf{q})$ and $\mathbf{g}(\mathbf{q})$ arise naturally out of certain spatial discretizations of the nonhydrostatic model equations (10)–(14).

To improve the performance of the IMEX methods, we focus on the family of operator-split RKR methods, which are identical to IMEX schemes, except replacing the computationally expensive implicit step with a so-called Rosenbrock step. Rosenbrock methods were originally developed by Rosenbrock (1963) in the 1960s and later refined by Nørsett and Wolfbrandt (1979). In the atmospheric science community, Rosenbrock-type methods have been used by Lanser et al. (2001) for solving the shallow-water equations on the sphere and adopted by St-Cyr and Neckels (2009) in the development of a fully implicit discontinuous Galerkin mesoscale model. More recently, a framework for high-order RKR time-stepping methods has been presented by Jebens et al. (2011) for use in atmospheric models utilizing cut cells. These methods are also popular in atmospheric chemistry modeling (Sandu et al. 1997; Verwer et al. 1999), because reaction equations tend to operate on very fast time scales.

In an implicit approach, one usually ends up with a nonlinear system of equations of the form

$$\mathbf{F}(\mathbf{x}) = 0 \tag{16}$$

that must be solved numerically for some vector $\mathbf{x}$. Perhaps the most well-known and robust technique for solving this system is the Newton–Krylov algorithm, which is an iterative approach defined by

$$\mathbf{x}^{(i)} = \mathbf{x}^{(i-1)} - \left[\frac{d\mathbf{F}}{d\mathbf{x}}\bigg|_{\mathbf{x}=\mathbf{x}^{(i-1)}}\right]^{-1} \mathbf{F}(\mathbf{x}^{(i-1)}). \tag{17}$$

When applied to systems of ODEs obtained from time-split PDE systems, the Newton–Krylov method is usually initialized by taking $\mathbf{x}^{(0)}$ to be the value of $\mathbf{x}$ obtained at the previous time step. Although this method converges quadratically, the Jacobian $d\mathbf{F}/d\mathbf{x}$ must nonetheless be computed (numerically or analytically) at every iteration. Under the Rosenbrock method, one does not require nonlinear convergence of (16) and instead only takes one step of (17). This significantly reduces the costs associated with the implicit step and restricts the number of Jacobian evaluations to one per Rosenbrock step.

To obtain high-order accuracy in time with our scheme, we consider a general Runge–Kutta method with interleaved explicit and linearly implicit Rosenbrock steps. In the general RKR framework, we define the initial data by

$$\mathbf{q}^{(0)} = \mathbf{q}^n \tag{18}$$

and then each subsequent step by

$$\mathbf{q}^{(i)} = \sum_{k=0}^{i-1} \alpha_{i,k}\mathbf{q}^{(k)} + \varepsilon_i\beta_i\Delta t\mathbf{f}(\mathbf{q}^{(i-1)}) + (1-\varepsilon_i)\beta_i\Delta t\{\mathbf{I} - \beta_i\Delta t\mathbf{Dg}(\mathbf{q}^{(i-1)})\}^{-1}\mathbf{g}(\mathbf{q}^{(i-1)}), \tag{19}$$

where $\alpha_{i,k}$ and β_i are arbitrary coefficients; Δt is the time step; $\mathbf{Dg}$ denotes the Jacobian of $\mathbf{g}$; and ε_i is a binary indicator variable,

$$\varepsilon_i = \begin{cases} 1, & \text{for an explicit step,} \\ 0, & \text{for a Rosenbrock step.} \end{cases} \tag{20}$$

This scheme description is interchangeable with the dual Butcher tableau approach of Ascher et al. (1997) but is preferred in this paper because it is more closely linked with the method's actual implementation.

b. Computation of the Jacobian

The main cost of the Rosenbrock method is in the implicit vertical step, which consists first of the construction of the Jacobian matrix and second of the matrix solve. Under our current implementation, the matrix solve is handled by the Linear Algebra Package (LAPACK) banded matrix solver. These routines are roughly twice as fast as the general matrix solver routines but nonetheless contribute significantly to the total wall-clock time of the implicit solve. No additional effort has been made to optimize these routines for our problem. The bulk of optimization efforts have instead focused on the construction of the Jacobian matrix.

Perhaps the simplest method of constructing the Jacobian matrix is via numerical differentiation. Consider a vertical array of state variables $\tilde{\mathbf{q}}$ that consists of concatenating all element-wise state variables in a vertical column. The vector $\tilde{\mathbf{q}} = (\tilde{q}_1, \ldots, \tilde{q}_N)$ will have length $N = N_z \times N_c$, where N_z is the number of vertical elements and N_c is the number of degrees of freedom per element. Under numerical differencing the Jacobian is written as a concatenation of vectors

$$\mathbf{Dg}(\tilde{\mathbf{q}}) = \left[\frac{\partial \mathbf{g}}{\partial \tilde{q}_1}\middle| \ldots \middle| \frac{\partial \mathbf{g}}{\partial \tilde{q}_N}\right], \tag{21}$$

where

$$\frac{\partial \mathbf{g}}{\partial \tilde{q}_i}(\mathbf{q}) \approx \frac{\mathbf{g}(\tilde{\mathbf{q}} + \boldsymbol{\delta}_i\epsilon) - \mathbf{g}(\tilde{\mathbf{q}})}{\epsilon}. \tag{22}$$

Here, ϵ is a small positive constant (here chosen to be 10^{-5}) and $\boldsymbol{\delta}_i$ is the vector form of the Kronecker delta,

$$(\boldsymbol{\delta}_i)_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases} \tag{23}$$

Because the Jacobian is sparse, only the nonzero components of each vector must be computed at each stage. Nonetheless, this method can become quite expensive because of repeated evaluation of the vector-valued function $\mathbf{g}$.

If the function $\mathbf{g}$ is simple enough, it may be possible to instead formulate the Jacobian analytically. This strategy significantly reduces the computation time required for constructing the Jacobian and so will be our method of choice in this paper. Under our finite-volume formulation, a quasi-linear flux function in the vertical (see section 4f) is chosen in order to facilitate an easy formulation of the Jacobian matrix. It will be shown (see section 5a) that the effort in formulating an analytic Jacobian significantly reduces computation time for the implicit solve. The analytic expressions for each term of the Jacobian have been omitted here because they are quite lengthy.

c. A crude splitting scheme

Perhaps the simplest stable scheme for splitting the explicit and linearly implicit components of the evolution equations involves simply applying a high-order explicit Runge–Kutta operator to the explicit terms followed by the Rosenbrock operator applied at the full time step. If we use the third-order strong-stability-preserving (SSP) three-stage third-order Runge–Kutta (RK3) scheme of Gottlieb et al. (2001) for the explicit component, this scheme proceeds according to

$$\mathbf{q}^{(1)} = \mathbf{q}^n + \Delta t\mathbf{f}(\mathbf{q}^n), \tag{24}$$

$$\mathbf{q}^{(2)} = \frac{3}{4}\mathbf{q}^n + \frac{1}{4}\mathbf{q}^{(1)} + \frac{\Delta t}{4}\mathbf{f}(\mathbf{q}^{(1)}), \tag{25}$$

$$\mathbf{q}^{(3)} = \frac{1}{3}\mathbf{q}^n + \frac{2}{3}\mathbf{q}^{(2)} + \frac{2\Delta t}{3}\mathbf{f}(\mathbf{q}^{(2)}), \quad \text{and} \tag{26}$$

$$\mathbf{q}^{n+1} \equiv \mathbf{q}^{(4)} = \mathbf{q}^{(3)} + \Delta t\{\mathbf{I} - \Delta t\mathbf{D}\mathbf{g}(\mathbf{q}^{(3)})\}^{-1}\mathbf{g}(\mathbf{q}^{(3)}). \tag{27}$$

Here the indices n and $n + 1$ denote the current and future time step. This scheme is first-order accurate in time and extremely diffusive (as demonstrated in section 5a). As a consequence, we do not recommend using this scheme in practice.

d. The Strang carryover scheme

To achieve better accuracy in time, we suggest a splitting scheme pointed out by S. Ruuth (2010, personal communication) and based on Strang splitting of the explicit and implicit operators. This scheme combines a third-order SSP RK3 step with one implicit solve per time step. However, at the first time step, we must perform one additional implicit operation, storing

$$\mathcal{G}^0 = \left[\mathbf{I} - \frac{\Delta t}{2}\mathbf{D}\mathbf{g}(\mathbf{q}^0)\right]^{-1}\mathbf{g}(\mathbf{q}^0). \tag{28}$$

After this initialization step, the algorithm proceeds as follows:

$$\mathbf{q}^{(1)} = \mathbf{q}^n + \frac{\Delta t}{2}\mathcal{G}^n, \tag{29}$$

$$\mathbf{q}^{(2)} = \mathbf{q}^{(1)} + \Delta t\mathbf{f}(\mathbf{q}^{(1)}), \tag{30}$$

$$\mathbf{q}^{(3)} = \frac{3}{4}\mathbf{q}^{(1)} + \frac{1}{4}\mathbf{q}^{(2)} + \frac{\Delta t}{4}\mathbf{f}(\mathbf{q}^{(2)}), \tag{31}$$

$$\mathbf{q}^{(4)} = \frac{1}{3}\mathbf{q}^{(1)} + \frac{2}{3}\mathbf{q}^{(3)} + \frac{2\Delta t}{3}\mathbf{f}(\mathbf{q}^{(3)}), \tag{32}$$

$$\mathcal{G}^{n+1} = \left\{\mathbf{I} - \frac{\Delta t}{2}\mathbf{D}\mathbf{g}(\mathbf{q}^{(4)})\right\}^{-1}\mathbf{g}(\mathbf{q}^{(4)}), \quad \text{and} \tag{33}$$

$$\mathbf{q}^{n+1} = \mathbf{q}^{(4)} + \frac{\Delta t}{2}\mathcal{G}^{n+1}. \tag{34}$$

This scheme can achieve third-order linear and nonlinear accuracy in **f** plus second-order accuracy in **g**. Comparing this method against the crude scheme, the only significant difference is in the form of the implicit step: that is, in using (27) versus (33). In fact, the use of $\Delta t/2$ within the Jacobian term leads to an implicit step that is analogous to a linear Crank–Nicolson step and so gives us second-order accuracy.

e. The Ascher–Ruuth–Spiteri (2, 3, 3) scheme

A third-order operator-split RKR scheme can be obtained from the Ascher–Ruuth–Spiteri (2, 3, 3) [ARS(2, 3, 3)] scheme of Ascher et al. (1997). If we simply replace the implicit solve with a Rosenbrock step, the resulting scheme takes the form

$$\mathbf{q}^{(1)} = \mathbf{q}^n + \gamma_c\Delta t\mathbf{f}(\mathbf{q}^n), \tag{35}$$

$$\mathbf{q}^{(2)} = \mathbf{q}^{(1)} + \gamma_c\Delta t\{\mathbf{I} - \gamma_c\Delta t\mathbf{D}\mathbf{g}(\mathbf{q}^{(1)})\}^{-1}\mathbf{g}(\mathbf{q}^{(1)}), \tag{36}$$

$$\mathbf{q}^{(3)} = \frac{1}{\gamma_c}\mathbf{q}^n + \frac{(3\gamma_c - 2)}{\gamma_c}\mathbf{q}^{(1)} + \frac{(1 - 2\gamma_c)}{\gamma_c}\mathbf{q}^{(2)} + 2(1 - \gamma_c)\Delta t\mathbf{f}(\mathbf{q}^{(2)}), \tag{37}$$

$$\mathbf{q}^{(4)} = \mathbf{q}^{(3)} + \gamma_c\Delta t\{\mathbf{I} - \gamma_c\Delta t\mathbf{D}\mathbf{g}(\mathbf{q}^{(3)})\}^{-1}\mathbf{g}(\mathbf{q}^{(3)}), \quad \text{and} \tag{38}$$

$$\mathbf{q}^{n+1} = -\frac{1}{2}\mathbf{q}^n - \frac{3\gamma_c}{2}\mathbf{q}^{(1)} + \frac{3}{2}\mathbf{q}^{(2)} + \frac{3(3\gamma_c - 2)}{2}\mathbf{q}^{(3)} + \frac{1}{2\gamma_c}\mathbf{q}^{(4)} + \frac{\Delta t}{2}\mathbf{f}(\mathbf{q}^{(4)}), \tag{39}$$

where

$$\gamma_c = \frac{3 + \sqrt{3}}{6}. \tag{40}$$

This scheme is linearly third-order accurate in both **f** and **g** and any cross-terms that arise from the integration procedure, but it is only nonlinearly third-order accurate in **f**. In fact, when $\mathbf{g} = 0$ the stability region for this scheme is exactly the stability region of the usual three-stage third-order-accurate Runge–Kutta operator. This scheme requires three explicit steps per time step and two Rosenbrock steps, with each Rosenbrock step consisting of a single evaluation of the Jacobian and a single linear solve. As a consequence, the overall computational cost of this method is approximately twice that of the Strang carryover scheme.

4. Spatial discretization

In this section, we turn our attention to the spatial discretization of the 3D nonhydrostatic governing equations (10)–(14) using a high-order finite-volume scheme. Here, we describe the key components of our algorithm for discretely evolving the state vector over time.

a. Finite-volume approach

In the full finite-volume approach, we first integrate the Euler equations in the form (10)–(14) over an element $\mathcal{Z}$ (with volume $|\mathcal{Z}|$) and make use of Gauss's divergence theorem to write the flux term as an integral around the boundary $\partial\mathcal{Z}$, giving

$$\frac{\partial}{\partial t}\overline{\overline{\mathbf{q}}} + \frac{1}{|\mathcal{Z}|}\int\int_{\partial\mathcal{Z}} \boldsymbol{\mathcal{F}}\cdot\mathbf{n}\, dS = \overline{\overline{\psi}}_C + \overline{\overline{\psi}}_G + \overline{\overline{\psi}}_H. \quad (41)$$

Here, the flux integral is taken over the surface with normal vector $\mathbf{n}$ and infinitesimal area element dS. The term $\boldsymbol{\mathcal{F}}\cdot\mathbf{n}$ is a vector quantity that denotes the outward flux of each of the state variables perpendicular to the boundary. The double overline denotes either a 2D or 3D average of the form

$$\overline{\overline{\phi}} = \frac{1}{|\mathcal{Z}|}\int_{\mathcal{Z}} \phi\, dV, \quad (42)$$

where the term $dV = dxdydz$ denotes the infinitesimal volume element in 3D (or $dV = dxdz$ for 2D $x - z$ slice configurations). Here, $\overline{\overline{\mathbf{q}}}$ again denotes the averaged state vector in cell $\mathcal{Z}$. Likewise, $\overline{\overline{\psi}}_C$, $\overline{\overline{\psi}}_G$, and $\overline{\overline{\psi}}_H$ denote the source terms due to the Coriolis force, gravity, and hydrostatic background pressure. Note that the volume-averaged formulation (41) is exactly equivalent to the original nonhydrostatic equations, and it is left to us to define an appropriate discretization over each of the terms in this expression.

The computational domain for our model is defined over a regular Cartesian grid. Elements are equally spaced with grid spacing ΔX in the X direction, ΔY in the Y direction, and ΔZ in the Z direction. Cartesian coordinates (x, y, z) are related to the coordinates in the computational domain (X, Y, Z) via a transform

$$x = X, \quad y = Y, \quad z = z[Z, h_T(x,y)], \quad (43)$$

where z is a function of the bottom topography $h_T(x, y)$. Under the terrain-following GS coordinate, this transform takes the form

$$z = h_T(x,y) + \frac{Z}{H}[H - h_T(x,y)], \quad (44)$$

where H is the altitude of the model cap. Element centroids $\mathbf{X}_{i,j,k} = (X_i, Y_j, Z_k)$ are located at

$$X_i = \left(i - \frac{1}{2}\right)\Delta X, \quad Y_j = \left(j - \frac{1}{2}\right)\Delta Y,$$
$$Z_k = \left(k - \frac{1}{2}\right)\Delta Z, \quad (45)$$

with spatial indices (i, j, k) that start from $(1, 1, 1)$. Edges (or faces) are midway between neighboring element centroids and so are defined by half-indices and denoted by $\partial\mathcal{Z}$ with initial index ½. For example, the edge $\partial\mathcal{Z}_{i+1/2,j,k}$ is at the interface between element (i, j, k) and $(i + 1, j, k)$ and defines a plane that is constant in both Y and Z. Quantities which are defined as edge averages will be denoted by a single overline. Average edge fluxes are defined at element edges, which are denoted here by $\overline{\mathbf{F}}_{i+1/2,j,k}$ for a flux across edge $(i + ½, j, k)$ and defined by

$$\overline{\mathbf{F}}_{i+1/2,j,k} = \frac{1}{|\mathcal{Z}|_{i,j,k}}\int\int_{\partial\mathcal{Z}_{i+1/2,j,k}} \boldsymbol{\mathcal{F}}\cdot\mathbf{n}\, dS. \quad (46)$$

Hence, the volume-averaged formulation (41) can be written as

$$\frac{\partial}{\partial t}\overline{\overline{\mathbf{q}}}_{i,j,k} = \mathbf{H}(\mathbf{q}) + \mathbf{V}(\mathbf{q}), \quad (47)$$

where

$$\mathbf{H}(\mathbf{q}) = \overline{\mathbf{F}}_{i-1/2,j,k} - \overline{\mathbf{F}}_{i+1/2,j,k} + \overline{\mathbf{F}}_{i,j-1/2,k} - \overline{\mathbf{F}}_{i,j+1/2,k} + \overline{\overline{\psi}}_C + \overline{\overline{\psi}}_H \quad \text{and} \quad (48)$$

$$\mathbf{V}(\mathbf{q}) = \overline{\mathbf{F}}_{i,j,k-1/2} - \overline{\mathbf{F}}_{i,j,k+1/2} + \overline{\overline{\psi}}_G. \quad (49)$$

The choice of splitting into $\mathbf{H}(\mathbf{q})$ (horizontal) and $\mathbf{V}(\mathbf{q})$ (vertical) is analogous to (15), with $\mathbf{H}(\mathbf{q})$ denoting terms that are usually significantly less stiff than terms of $\mathbf{V}(\mathbf{q})$.

For each explicit substage, the 3D finite-volume algorithm proceeds as follows:

1) Construct the left and right edge-averaged state vector from the subgrid-scale reconstruction.
2) Deconvolve the edge averages to give a fourth-order-accurate edge-centered approximation of the state vector.
3) Transform vector quantities into the orthogonal frame at the edge center point.
4) Compute the pointwise flux across the edge using a Riemann solver.
5) Transform vector fluxes into the Cartesian frame using the deorthonormalization matrix.
6) Convolve the pointwise fluxes to obtain a fourth-order-accurate edge-averaged approximation of the flux.
7) Compute element-averaged source terms.
8) Update the element-averaged state vector within each element.

The algorithm is similar in the vertical, except the deconvolution and convolution stages are ignored, which

reduces the maximum formal order of accuracy of the spatial discretization in the vertical to second order. Each of the substages of this algorithm will be described in the following sections.

b. The subgrid-scale reconstruction

In the upwind finite-volume framework, we require a subgrid-scale reconstruction within each element that is used to capture the continuous behavior of state variables below the grid scale. The finite-volume subgrid-scale reconstruction is composed of a polynomial basis whose coefficients are reconstructed from the state information of neighboring elements. This is in contrast to discontinuous Galerkin methods, for instance, which instead store the polynomial coefficients as independent degrees of freedom. The subgrid-scale reconstruction is then used for computing the approximate averaged flux across each edge. Because reconstructions are inherently discontinuous across edges, a Riemann problem must be solved at each interface to obtain averaged edge fluxes. So-called approximate Riemann solvers that are typically used to solve each Riemann problem are pointwise operators that take as input the approximate state vector on each side of the interface (for simplicity referred to as left and right states). Hence, the simplest form for an approximate Riemann solver is

$$\mathbf{F}^* = \mathbf{F}^*(\mathbf{q}^L, \mathbf{q}^R), \tag{50}$$

where $\mathbf{q}^L$ is the left state and $\mathbf{q}^R$ is the right state. In the vertical, left and right instead correspond to bottom and top, respectively.

In the X direction, the subgrid-scale reconstruction is obtained by fitting a quartic polynomial through elements $\{(i-2, j, k), \ldots, (i+2, j, k)\}$. Evaluating the polynomial at the edge $(i+\frac{1}{2}, j, k)$ leads to reconstructed edge values, which are computed as a linear combination of element averages via

$$\overline{\mathbf{q}}^L_{i+1/2,j,k} = \frac{1}{30}\overline{\overline{\mathbf{q}}}_{i-2,j,k} - \frac{13}{60}\overline{\overline{\mathbf{q}}}_{i-1,j,k} + \frac{47}{60}\overline{\overline{\mathbf{q}}}_{i,j,k} + \frac{9}{20}\overline{\overline{\mathbf{q}}}_{i+1,j,k} - \frac{1}{20}\overline{\overline{\mathbf{q}}}_{i+2,j,k} \quad \text{and} \tag{51}$$

$$\overline{\mathbf{q}}^R_{i+1/2,j,k} = -\frac{1}{20}\overline{\overline{\mathbf{q}}}_{i-1,j,k} + \frac{9}{20}\overline{\overline{\mathbf{q}}}_{i,j,k} + \frac{47}{60}\overline{\overline{\mathbf{q}}}_{i+1,j,k} - \frac{13}{60}\overline{\overline{\mathbf{q}}}_{i+2,j,k} + \frac{1}{30}\overline{\overline{\mathbf{q}}}_{i+3,j,k}. \tag{52}$$

The magnitude of the discontinuity $\overline{\mathbf{q}}^R_{i+1/2,j,k} - \overline{\mathbf{q}}^L_{i+1/2,j,k}$ can be interpreted as a measure of the smoothness of the underlying field. This property is used by the Riemann solver in computing edge fluxes, so that diffusion can be targeted at regions where the underlying state variables are not smooth. The reconstruction formula for the Y direction is identical, except with the j indices varied. At vertical edges, we make use of either (51) and (52) with k indices varied or a more compact three-point stencil given by

$$\overline{\mathbf{q}}^L_{i,j,k+1/2} = -\frac{1}{6}\overline{\overline{\mathbf{q}}}_{i,j,k-1} + \frac{5}{6}\overline{\overline{\mathbf{q}}}_{i,j,k} + \frac{2}{6}\overline{\overline{\mathbf{q}}}_{i,j,k+1} \quad \text{and} \tag{53}$$

$$\overline{\mathbf{q}}^R_{i,j,k+1/2} = \frac{2}{6}\overline{\overline{\mathbf{q}}}_{i,j,k} + \frac{5}{6}\overline{\overline{\mathbf{q}}}_{i,j,k+1} - \frac{1}{6}\overline{\overline{\mathbf{q}}}_{i,j,k+2}. \tag{54}$$

It should be emphasized the aforementioned reconstructions produce either fifth-order-accurate (using the five-point stencil) or third-order-accurate (using the three-point stencil) approximations to the edge-averaged state vector. If we were to directly evaluate the flux using these reconstructed edge averages, the resulting edge-averaged flux would only be second-order accurate. This result arises because the flux function is not a linear operator, and so the average of the flux is not the flux of the average. Hence, without further application of a convolution/deconvolution operator, both of these reconstructed edge averages will lead to a scheme that is formally second-order accurate. Nonetheless, the stencil (51) and (52) generally produces less diffusive and more accurate results, whereas using the three-point stencil (53) and (54) reduces the bandwidth of the Jacobian matrix and so generally results in a faster vertical solve.

Near no-flux boundaries, the reconstruction formula is modified to only use one-sided information. For the five-point stencil near the bottom edge, we use

$$\overline{\mathbf{q}}^R_{i,j,1/2} = \frac{11}{6}\overline{\overline{\mathbf{q}}}_{i,j,1} - \frac{7}{6}\overline{\overline{\mathbf{q}}}_{i,j,2} + \frac{2}{6}\overline{\overline{\mathbf{q}}}_{i,j,3}, \tag{55}$$

$$\overline{\mathbf{q}}^L_{i,j,3/2} = \frac{2}{6}\overline{\overline{\mathbf{q}}}_{i,j,1} + \frac{5}{6}\overline{\overline{\mathbf{q}}}_{i,j,2} - \frac{1}{6}\overline{\overline{\mathbf{q}}}_{i,j,3}, \tag{56}$$

$$\overline{\mathbf{q}}^R_{i,j,3/2} = \frac{1}{4}\overline{\overline{\mathbf{q}}}_{i,j,1} + \frac{13}{12}\overline{\overline{\mathbf{q}}}_{i,j,2} - \frac{5}{12}\overline{\overline{\mathbf{q}}}_{i,j,3} + \frac{1}{12}\overline{\overline{\mathbf{q}}}_{i,j,4}, \quad \text{and} \tag{57}$$

$$\overline{\mathbf{q}}^L_{i,j,5/2} = -\frac{1}{12}\overline{\overline{\mathbf{q}}}_{i,j,1} + \frac{7}{12}\overline{\overline{\mathbf{q}}}_{i,j,2} + \frac{7}{12}\overline{\overline{\mathbf{q}}}_{i,j,3} - \frac{1}{12}\overline{\overline{\mathbf{q}}}_{i,j,4}. \tag{58}$$

For the three-point stencil, we use

$$\overline{\mathbf{q}}^R_{i,j,1/2} = \frac{3}{2}\overline{\overline{\mathbf{q}}}_{i,j,1} - \frac{1}{2}\overline{\overline{\mathbf{q}}}_{i,j,2} \quad \text{and} \tag{59}$$

$$\overline{\mathbf{q}}^L_{i,j,3/2} = \frac{1}{2}\overline{\overline{\mathbf{q}}}_{i,j,1} + \frac{1}{2}\overline{\overline{\mathbf{q}}}_{i,j,2}. \tag{60}$$

Near the upper boundary, the reconstructions are analogous, except with indices reordered.

c. Fourth-order convolution and deconvolution operators in 3D

As mentioned previously, the problem with using edge averages as inputs to the Riemann solver is that this operator is inherently pointwise, whereas edge averages are only second-order approximations to the pointwise edge-center value of the state vector. However, one may apply a deconvolution operator (Barad and Colella 2005) to convert edge averages to pointwise values [here denoted with a subscript (0)]. For example, along an edge of constant X, it can be verified that

$$\mathbf{q}_{(0)i+1/2,j,k} = \overline{\mathbf{q}}_{i+1/2,j,k} - \frac{\Delta Y^2}{24}\left(\frac{\partial^2 \mathbf{q}}{\partial Y^2}\right)_{(0)i+1/2,j,k} \tag{61}$$

is a fourth-order approximation to $\mathbf{q}_{(0)i+1/2,j,k}$, the value of the state vector at the center point of edge $\partial \mathcal{Z}_{i+1/2,j,k}$. Here, $\overline{\mathbf{q}}_{i+1/2,j,k}$ is the edge average of the state vector and $(\partial^2 \mathbf{q}/\partial Y^2)_{(0)i+1/2,j,k}$ is the second derivative in Y of the state vector evaluated at the edge center point. A second-order approximation to $(\partial^2 \mathbf{q}/\partial Y^2)_{(0)i+1/2,j,k}$ can be used, such as

$$\left(\frac{\partial^2 \mathbf{q}}{\partial Y^2}\right)_{(0)i+1/2,j,k} \approx \frac{\overline{\mathbf{q}}_{i+1/2,j-1,k} - 2\overline{\mathbf{q}}_{i+1/2,j,k} + \overline{\mathbf{q}}_{i+1/2,j+1,k}}{\Delta Y^2}, \tag{62}$$

which, in combination with (61), gives a fourth-order approximation to $\mathbf{q}_{(0)i+1/2,j,k}$. The Riemann solver is then applied pointwise to $\mathbf{q}^L_{(0)i+1/2,j,k}$ and $\mathbf{q}^R_{(0)i+1/2,j,k}$, and a convolution operation is applied to retrieve the edge average of the flux vector $\overline{\mathbf{F}}_{i+1/2,j,k}$,

$$\overline{\mathbf{F}}_{i+1/2,j,k} = \mathbf{F}_{(0)i+1/2,j,k} + \frac{\Delta Y^2}{24}\left(\frac{\partial^2 \mathbf{F}}{\partial Y^2}\right)_{(0)i+1/2,j,k}. \tag{63}$$

In this case, the edge-average of the flux vector is a fourth-order-accurate estimate of the flux across the given edge.

To obtain fourth-order accuracy overall, the source terms of the horizontal momentum equations must also be evaluated with at least third-order accuracy. Simply evaluating source terms using cell averages $\overline{\overline{\mathbf{q}}}_{i,j,k}$ only leads to second-order-accurate approximations of these terms. Hence, to obtain high-order accuracy, we again apply a deconvolution procedure to obtain an approximation to the state vector at the element center point,

$$\mathbf{q}_{(0)i,j,k} = \overline{\overline{\mathbf{q}}}_{i,j,k} - \frac{\Delta X^2}{24}\left(\frac{\partial^2 \mathbf{q}}{\partial X^2}\right)_{(0)i,j,k} - \frac{\Delta Y^2}{24}\left(\frac{\partial^2 \mathbf{q}}{\partial Y^2}\right)_{(0)i,j,k}. \tag{64}$$

Source terms are then evaluated at the element center point [$\psi_{(0)i,j,k} = \psi(\mathbf{q}_{(0)i,j,k})$], and a convolution procedure is applied to reaverage the source term over each element,

$$\overline{\overline{\psi}}_{i,j,k} = \psi_{(0)i,j,k} + \frac{\Delta X^2}{24}\left(\frac{\partial^2 \psi}{\partial X^2}\right)_{(0)i,j,k} + \frac{\Delta Y^2}{24}\left(\frac{\partial^2 \psi}{\partial Y^2}\right)_{(0)i,j,k}. \tag{65}$$

Again, all second derivatives are approximated to second-order accuracy using (62),

$$\left(\frac{\partial^2 \psi}{\partial X^2}\right)_{(0)i,j,k} \approx \frac{\overline{\overline{\psi}}_{i-1,j,k} - 2\overline{\overline{\psi}}_{i,j,k} + \overline{\overline{\psi}}_{i+1,j,k}}{\Delta X^2} \quad \text{and} \tag{66}$$

$$\left(\frac{\partial^2 \psi}{\partial Y^2}\right)_{(0)i,j,k} \approx \frac{\overline{\overline{\psi}}_{i,j-1,k} - 2\overline{\overline{\psi}}_{i,j,k} + \overline{\overline{\psi}}_{i,j+1,k}}{\Delta Y^2}. \tag{67}$$

In our treatment, the edge averages in the vertical direction are used as direct input to the Riemann solver. This implies that this method is only second-order accurate in the vertical. However, these errors are typically much smaller than the associated truncation errors in the horizontal for two reasons: First, in operational global atmospheric models the grid spacing in the vertical is typically much smaller than the grid spacing in the horizontal. Second, vertical wind speeds are typically much less than the corresponding horizontal wind speeds.

d. Orthonormalization

In this paper, we make use of approximate Riemann solvers for computing the flux $\overline{\mathbf{F}}(\mathbf{q})$ across each edge. However, generic Riemann solvers are purely one-dimensional operators. To apply this class of solvers to multidimensional problems, we must first isolate the vector components that are perpendicular and parallel to each edge. In 3D, the orthogonal frame consists of one basis vector that is orthogonal to the active edge (denoted $\mathbf{e}_\perp$) and two components that are parallel to the edge (denoted $\mathbf{e}_1$ and $\mathbf{e}_2$). Hence, the nonorthogonal components define a tangent plane to the active edge but are not necessarily orthogonal to one another. At the point where the orthogonal basis is defined, an arbitrary vector $\mathbf{v}$ can be written in either the Cartesian basis,

$$\mathbf{v} = v^x \mathbf{e}_x + v^y \mathbf{e}_y + v^z \mathbf{e}_z, \tag{68}$$

with unit vectors $\mathbf{e}_x$, $\mathbf{e}_y$, and $\mathbf{e}_z$, or in the orthonormal basis

$$\mathbf{v} = v^\perp \mathbf{e}_\perp + v^1 \mathbf{e}_1 + v^2 \mathbf{e}_2, \tag{69}$$

with unit vectors $\mathbf{e}_\perp$, $\mathbf{e}_1$, and $\mathbf{e}_2$. The transformation between these two systems is accomplished via the orthonormalization matrix $\mathcal{O}_d(x, y, z)$,

$$\begin{pmatrix} v^\perp \\ v^1 \\ v^2 \end{pmatrix} = \mathcal{O}_d(x, y, z) \begin{pmatrix} v^x \\ v^y \\ v^z \end{pmatrix}. \tag{70}$$

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

$$\begin{pmatrix} v^x \\ v^y \\ v^z \end{pmatrix} = \mathcal{O}_d^{-1}(x, y, z) \begin{pmatrix} v^\perp \\ v^1 \\ v^2 \end{pmatrix}. \tag{71}$$

In the absence topography, the orthonormalization matrices reduce to permutation matrices of the form

$$\mathcal{O}_x(x, y, z) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\mathcal{O}_y(x, y, z) = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\mathcal{O}_z(x, y, z) = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \tag{72}$$

In the presence of topography, the coordinate system has been constructed such that edges of constant X and Y are also constant in x and y, respectively. Hence, the matrices $\mathcal{O}_x$ and $\mathcal{O}_y$ are identical to those in (72). However, edges of constant Z (in computational space) no longer correspond to edges of constant z (in Cartesian space). Hence, an alternative orthonormalization matrix must be constructed for these edges.

By definition, the GS coordinates define a mapping between computational space and Cartesian space of the form

$$z(Z) = Z + h_T(x, y)\left(1 - \frac{Z}{H}\right), \tag{73}$$

where h_T is the height of the topography and H is the height of the model cap. The coordinates z and Z are defined in the range $z \in [h_T, H]$ and $Z \in [0, H]$. Hence, surfaces of constant Z in Cartesian space take the form

$$\phi(x, y, z) \equiv z - Z - h_T(x, y)\left(1 - \frac{Z}{H}\right). \tag{74}$$

The perpendicular vector to the surface is obtained by differentiating ϕ, leading to

$$\mathbf{q}_\perp = \nabla\phi = \left[-\frac{\partial h_T}{\partial x}\left(1 - \frac{Z}{H}\right)\right]\mathbf{e}_x + \left[-\frac{\partial h_T}{\partial y}\left(1 - \frac{Z}{H}\right)\right]\mathbf{e}_y + \mathbf{e}_z, \tag{75}$$

and so at this edge we have

$$\mathbf{e}_\perp = \frac{1}{|\mathbf{q}_\perp|}\mathbf{q}_\perp. \tag{76}$$

To obtain basis vectors tangent to the edge we use Gram–Schmidt orthonormalization to define

$$\mathbf{q}_1 = \mathbf{e}_x - (\mathbf{e}_x \cdot \mathbf{e}_\perp)\mathbf{e}_\perp, \quad \mathbf{q}_2 = \mathbf{e}_y - (\mathbf{e}_y \cdot \mathbf{e}_\perp)\mathbf{e}_\perp \tag{77}$$

and then normalize accordingly,

$$\mathbf{e}_1 = \frac{1}{|\mathbf{q}_1|}\mathbf{q}_1, \quad \mathbf{e}_2 = \frac{1}{|\mathbf{q}_2|}\mathbf{q}_2. \tag{78}$$

The deorthonormalization matrix at this edge is then the composition of the three basis vectors,

$$\mathcal{O}_z^{-1} = (\mathbf{e}_\perp \mathbf{e}_1 \mathbf{e}_2), \tag{79}$$

with $\mathcal{O}_z$ obtained via numerical inversion of $\mathcal{O}_z^{-1}$.

e. The AUSM$^+$-up Riemann solver

The Advection Upstream Splitting Method (AUSM$^+$-up) approximate Riemann solver of Liou (2006) was recently developed with the goal of enhancing the accuracy of the Riemann solution in the low Mach number regime. Many other commonly used Riemann solvers, including the popular solver of Rusanov (1961) and the solver of Roe (1981), do a poor job in the very low Mach number regime because they introduce a significant amount of numerical diffusion that can smear out the solution (see Ullrich et al. 2010). Here, we give a short overview of the algorithmic implementation of this solver without delving into the mathematical details.

In general, standalone Riemann solvers require that the velocity components of the input state vector be written in an orthogonal frame. Orthonormalization is performed by multiplying the velocity vector by the orthonormalization matrix, which is described in section 4d and yields velocity components $(v^\perp, v^1, v^2)$ in the orthogonal frame. The momentum flux, which is computed in the orthogonal frame, must similarly be transformed back into the Cartesian frame, which can be obtained by multiplying the Riemann flux by the inverse of the orthonormalization matrix.

Given a left state vector $\mathbf{q}_L = [\rho_L, (\rho v^\perp)_L, (\rho v^1)_L, (\rho v^2)_L, (\rho\theta)_L]$ and right state vector $\mathbf{q}_R = [\rho_R, (\rho v^\perp)_R, (\rho v^1)_R, (\rho v^2)_R, (\rho\theta)_R]$, we define the averaged density,

$$\rho_{1/2} = \frac{1}{2}(\rho_L + \rho_R); \tag{80}$$

averaged sound speed,

$$a_{1/2} = \frac{1}{2}\left[\sqrt{\frac{\gamma p_R}{\rho_R}} + \sqrt{\frac{\gamma p_L}{\rho_L}}\right]; \tag{81}$$

perpendicular Mach numbers at the interface,

$$M_L = \frac{v_L^\perp}{a_{1/2}}, \quad \text{and} \quad M_R = \frac{v_R^\perp}{a_{1/2}}; \tag{82}$$

and mean local Mach number,

$$\overline{M}^2 = \frac{(v_L^\perp)^2 + (v_R^\perp)^2}{2a_{1/2}^2}. \tag{83}$$

Here, p_L and p_R are the corresponding left and right pressures, which are calculated from the state vector via (5). Similarly, $v_L^\perp = (\rho v^\perp)_{L/\rho L}$ and $v_R^\perp = (\rho v^\perp)_{R/\rho R}$ are the interfacial velocities.

The interface Mach number is then defined as

$$M_{1/2} = \mathcal{M}_{(4)}^+(M_L) + \mathcal{M}_{(4)}^-(M_R) - K_p \max(1 - \sigma\overline{M}^2, 0)\frac{p_R - p_L}{\rho_{1/2}a_{1/2}^2}, \tag{84}$$

where

$$\mathcal{M}_{(2)}^\pm(M) = \frac{1}{4}(M \pm 1)^2,$$

$$\mathcal{M}_{(4)}^\pm(M) = \begin{cases} \frac{1}{2}(M \pm |M|) & \text{if} \quad |M| \geq 1, \\ \mathcal{M}_{(2)}^\pm(M)[1 \mp 16\tilde{\beta}\mathcal{M}_{(2)}^\mp(M)] & \text{otherwise.} \end{cases} \tag{85}$$

The advective component of the flux is then defined by

$$\dot{m}_{1/2} = a_{1/2}M_{1/2}\begin{cases} \rho_L & \text{if} \quad M_{1/2} > 0, \\ \rho_R & \text{otherwise.} \end{cases} \tag{86}$$

To obtain the pressure-driven component of the flux, we make use of the definition

$$\mathcal{P}_{(5)}^\pm(M) = \begin{cases} \frac{1}{2}[1 \pm \text{sign}(M)] & \text{if} \quad |M| \geq 1, \\ \mathcal{M}_{(2)}^\pm(M)[(\pm 2 - M) \mp 16\tilde{\alpha}M\mathcal{M}_{(2)}^\mp(M)] & \text{otherwise.} \end{cases} \tag{87}$$

The interface pressure-driven flux is then given by

$$p_{1/2} = \mathcal{P}_{(5)}^+(M_L)p_L + \mathcal{P}_{(5)}^-(M_R)p_R - K_u\mathcal{P}_{(5)}^+(M_L)\mathcal{P}_{(5)}^-(M_R)(\rho_L + \rho_R)a_{1/2}(v_R^\perp - v_L^\perp). \tag{88}$$

Combining (84)–(86) and (88), we obtain that the total numerical flux across the interface is then given by

$$\mathbf{F}^* = \left[\dot{m}_{1/2}\begin{cases} \mathbf{q}_L/\rho_L & \text{if} \quad \dot{m}_{1/2} > 0, \\ \mathbf{q}_R/\rho_R & \text{otherwise} \end{cases}\right] + \mathbf{p}_{1/2}, \tag{89}$$

with

$$\mathbf{q}_L = \begin{bmatrix} \rho_L \\ (\rho v^\perp)_L \\ (\rho v^1)_L \\ (\rho v^2)_L \\ (\rho\theta)_L \end{bmatrix}, \quad \mathbf{q}_R = \begin{bmatrix} \rho_R \\ (\rho v^\perp)_R \\ (\rho v^1)_R \\ (\rho v^2)_R \\ (\rho\theta)_R \end{bmatrix}, \quad \mathbf{P}_{1/2} = \begin{bmatrix} 0 \\ p_{1/2} \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{90}$$

Several free parameters are available in this scheme. For simplicity, we follow Liou (2006) by choosing

$$\tilde{\alpha} = \frac{3}{16}, \quad \tilde{\beta} = \frac{1}{8}, \quad K_u = \frac{3}{4}, \quad K_p = \frac{1}{4}, \quad \sigma = 1. \tag{91}$$

f. The modified AUSM+-up Riemann solver

The AUSM$^+$ solver can be simplified dramatically for approximately smooth flows with $M \ll 1$. This simplification may be desirable to improve the computational performance of the method and significantly reduce the complexity of the analytic Jacobian needed in the implicit step. This modified Riemann solver is then applied in the vertical, where computing the Jacobian requires multiple evaluations of the Riemann solver at each time step. To begin, we assume that the difference between the left and right states is small relative to the magnitude of the hydrostatic background. In this case, the speed of sound at the interface, which primarily comes into play in the diffusion terms, can be approximated as

$$a_{1/2} = \sqrt{\frac{\gamma \rho^h_{1/2}}{\rho^h_{1/2}}}, \tag{92}$$

where the pressure and density field are taken to be the quantities defined by the hydrostatic background. Further, by making use of (5) the diffusive term in (84), which involves the pressure difference across the interface, can be approximated as

$$\begin{aligned} p_R - p_L &= p_0 \left\{ \frac{R_d}{p_0} [(\rho\theta)^h_{1/2} + (\rho\theta)'_R] \right\}^{c_p/c_v} \\ &\quad - p_0 \left\{ \frac{R_d}{p_0} [(\rho\theta)^h_{1/2} + (\rho\theta)'_L] \right\}^{c_p/c_v}, \\ &\approx \frac{p_0 c_p}{c_v} [(\rho\theta)^h]^{c_p/c_v - 1} [(\rho\theta)'_R - (\rho\theta)'_L], \\ &= \frac{a^2_{1/2} \rho^h_{1/2}}{(\rho\theta)^h_{1/2}} [(\rho\theta)'_R - (\rho\theta)'_L]. \end{aligned}$$

The interfacial Mach number then takes the form

$$M_{1/2} = \frac{\rho_L v^\perp_L + \rho_R v^\perp_R}{a_{1/2}(\rho_L + \rho_R)} - \frac{K_p}{(\rho\theta)^h_{1/2}} [(\rho\theta)'_R - (\rho\theta)'_L]. \tag{93}$$

Similarly, in the limit of small Mach number, the interfacial pressure term (88) takes the simplified form

$$\begin{aligned} p_{1/2} &= p_0 \left[\frac{R_d}{p_0} \frac{(\rho\theta)_L + (\rho\theta)_R}{2} \right]^{c_p/c_v} \\ &\quad - \frac{K_u a_{1/2}}{2} (\rho_R v^\perp_R - \rho_L v^\perp_L). \end{aligned} \tag{94}$$

The flux calculation in this case otherwise follows (86) and (89).

g. Nonreflecting boundary conditions

When required in the test cases of section 5, nonreflecting boundary conditions are imposed in the form of a sponge layer along the model top and at outflow boundaries. These boundary conditions are imposed by adding damping to the momentum and potential temperature evolution equations. This damping takes the form

$$\frac{\partial \overline{\overline{q}}}{\partial t} = -\tau (\overline{\overline{q}} - \overline{\overline{q}}_b), \tag{95}$$

where $\tau = \tau(x, z)$ is the inverse time scale of the damping, $q \in \{\rho u, \rho v, \rho w, (\rho\theta)'\}$, and the subscript b denotes some specified background state. Here, $\tau = \tau(x, z)$ denotes the local strength of the Rayleigh damping. Damping is applied over a finite interval in the interior of the domain, spanning the range $[s_0 - s_T, s_0]$, where $s \in \{x, z\}$ for 2D $x - z$ slice simulations, s_0 denotes the location of the boundary, and s_T denotes the thickness of the damping layer. The strength of the damping within the damping layer is determined by τ, which, following Giraldo and Restelli (2008), is defined as

$$\tau(s) = \begin{cases} 0 & \text{if} \quad s < s_0 - s_T \\ \tau_0 \left[\dfrac{s - (s_0 - s_T)}{s_T} \right]^4 & \text{otherwise} \end{cases}, \tag{96}$$

for mountain wave test cases (sections 5c and 5d) where $\tau_0 = 2.0 \times 10^{-2}\ \mathrm{s}^{-1}$. If two boundary layers coincide, as with the outflow lateral boundary and the upper boundary, the strength of the Rayleigh friction is taken to be the maximum of the two coefficients.

h. Explicit diffusion

Although the Riemann solver provides a mechanism for maintaining stability via the addition of implicit diffusion, in some circumstances it may be desirable to specify an explicit viscous forcing. Traditionally, a parameterization of the viscosity appears in the form of a Laplacian-type viscous term on the right-hand side of the nonhydrostatic equations of motion (10)–(14). However, as argued by Giraldo and Restelli (2008), more physical parameterizations of dynamic viscosity exist. Nonetheless, for simplicity we will use this form of diffusion when needed for selected test cases in section 5.

The viscosity augments the nonhydrostatic equations by adding a term of the form $\nabla \cdot (\nu\rho\nabla\mathbf{u})$ to the right-hand side of the momentum equations and $\nabla \cdot (\nu\rho\nabla\theta)$ to the potential temperature density equation, for a given viscosity coefficient ν. For the case of zero topography, these terms are incorporated into the finite-volume discretization by adjusting the computed Riemann edge flux according to

$$[\mathbf{F}_{(0)}]^{\mathrm{diff}} = \mathbf{F}_{(0)} + \nu\rho\nabla_d \mathbf{q}_p, \tag{97}$$

where $\mathbf{F}_{(0)}$ is the output of the Riemann solver and $\mathbf{q}_p = (0, u, v, w, \theta)$. Here, ∇_d denotes the derivative across the edge, which is discretized in the X direction as

$$\begin{aligned} (\nabla_d \mathbf{q})_{i+1/2,j,k} &= \frac{1}{\Delta X} \left[\frac{1}{12} \overline{\overline{\mathbf{q}}}_{i-1,j,k} - \frac{15}{12} \overline{\overline{\mathbf{q}}}_{i,j,k} \right. \\ &\quad \left. + \frac{15}{12} \overline{\overline{\mathbf{q}}}_{i+1,j,k} - \frac{1}{12} \overline{\overline{\mathbf{q}}}_{i+2,j,k} \right]. \end{aligned} \tag{98}$$

For the y and z directions, the discretization is analogous. Under the three-point vertical stencil (53) and (54), we

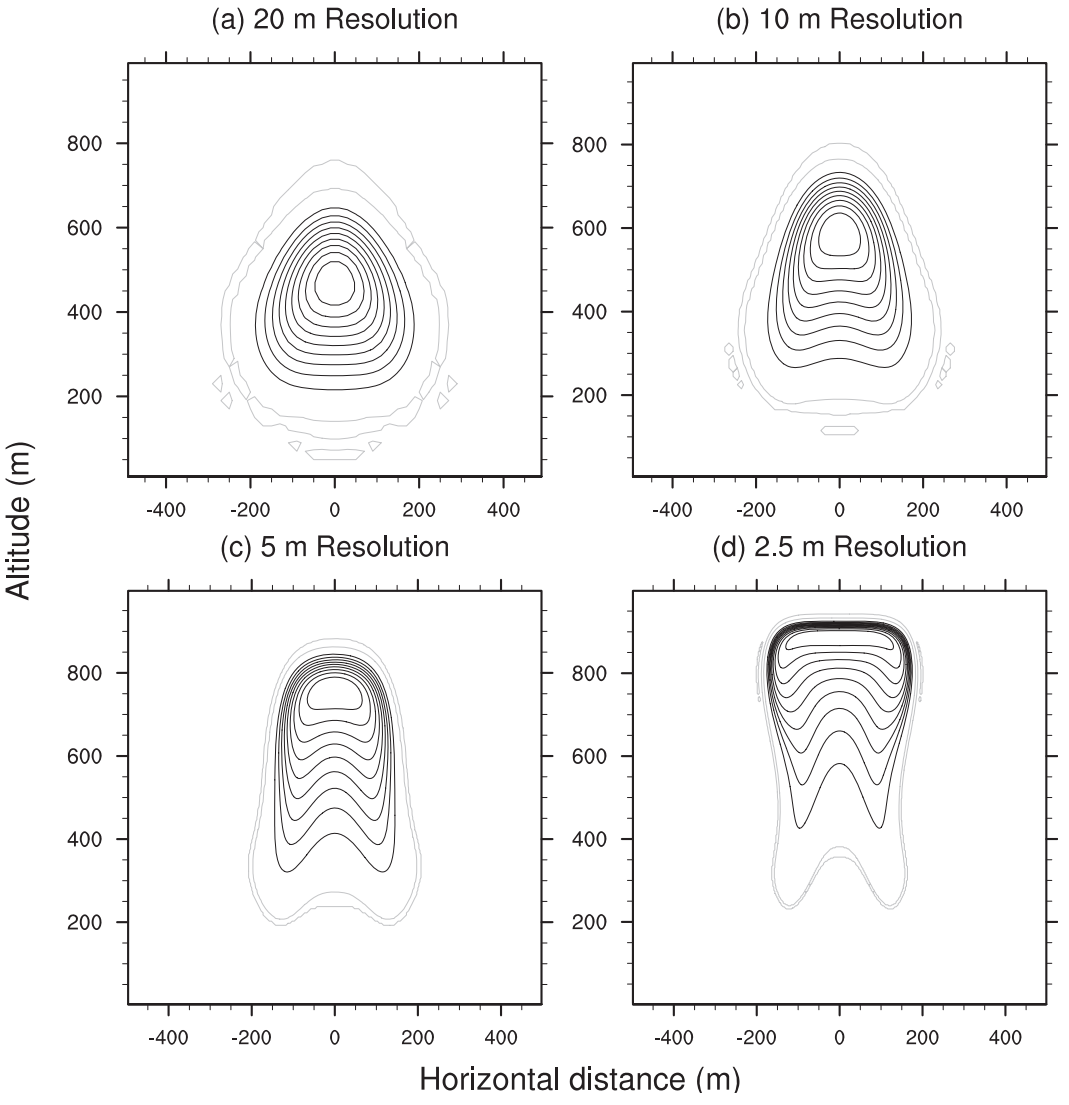

FIG. 1. Plots of the potential temperature perturbation for the rising thermal bubble test case with crude splitting and three-point vertical stencil at time $t = 700$ s and four choices of resolution. The time step is chosen to be 0.05 s at the coarsest resolution and is otherwise proportional to the grid spacing. Contour lines are from 300 to 300.5 K with a contour interval of 0.05 K. The 300-K contour line is shown in light gray to emphasize numerical oscillations due to undershoots and overshoots.

instead use the central approximation to the vertical derivative in (97), given by

$$(\nabla_d q)_{i,j,k+1/2} = \frac{1}{\Delta Z}[\bar{\bar{\mathbf{q}}}_{i,j,k+1} - \bar{\bar{\mathbf{q}}}_{i,j,k}]. \qquad (99)$$

5. Numerical results

In this section, we present a selection of 2D (x–z) and 3D numerical results in order to verify the convergence and accuracy properties of the schemes discussed in this paper. In section 5a, we look at a rising thermal bubble in order to verify that our scheme is consistent with other models and to show the effect of the first-, second-, and third-order-accurate time-stepping schemes. In this section, we also compare the three- and five-point vertical stencils and show the timing results for each of the available model configurations. In section 5b, we study the spatial and temporal convergence properties of these schemes using a density current test case with explicit viscous forcing. The performance of our model for flow over topography is studied in sections 5c and 5d. The latter case is used to verify stability of the time integrators, even for a large horizontal–vertical aspect ratio. The problem of a 3D geostrophically balanced flow in a channel is studied in section 5e, again using a large

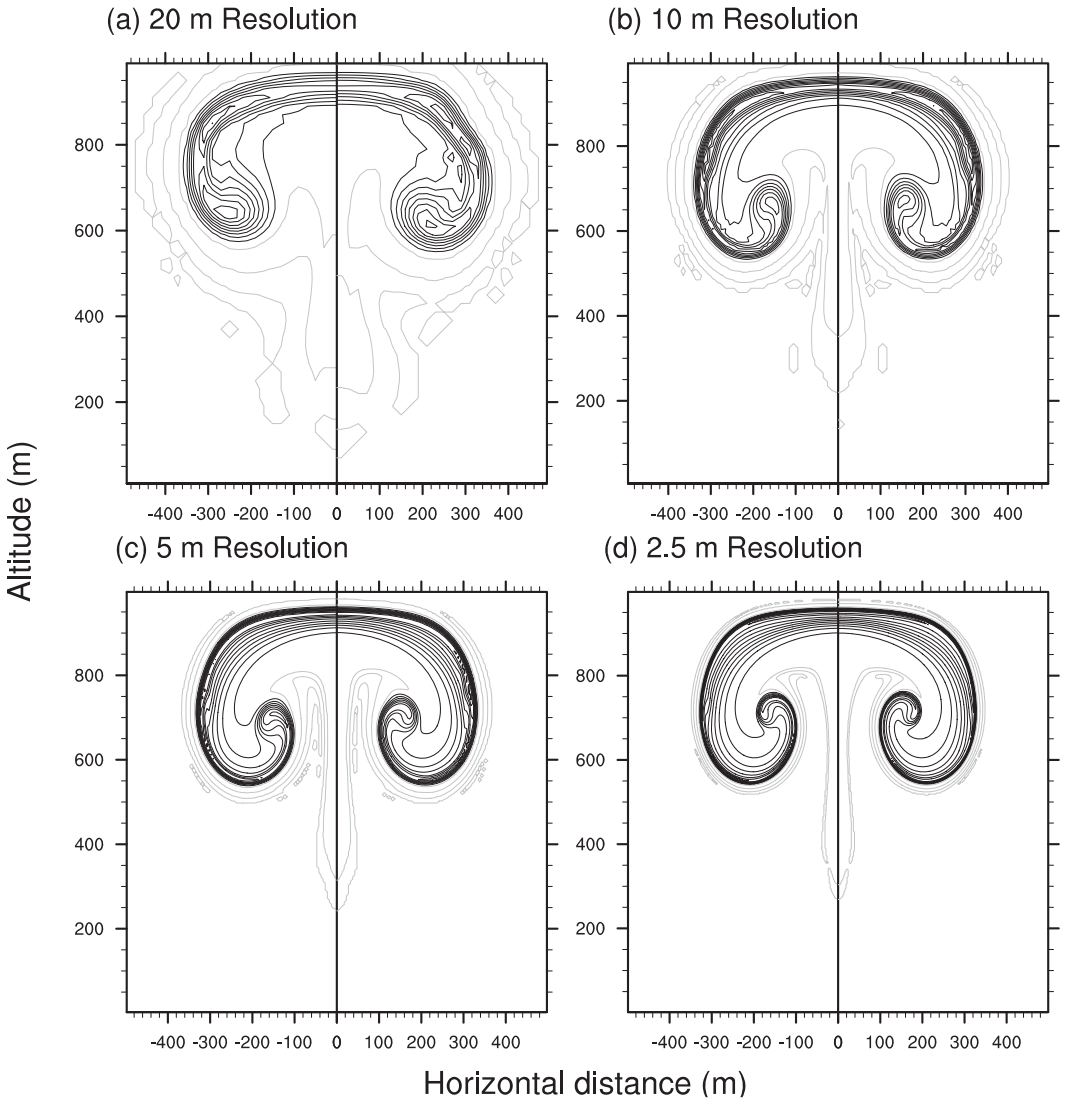

FIG. 2. As in Fig. 1, but showing (left) the Strang carryover scheme and (right) the ARS(2, 3, 3) scheme on the same axes with three-point vertical stencil given by (53) and (54). We have exploited the symmetry of the rising bubble to plot these schemes side by side for comparison.

horizontal–vertical aspect ratio. This test case is used to verify fourth-order horizontal convergence of our numerical method. Finally, in section 5f, we look at the evolution of a baroclinic wave in the channel on both a constant f plane and a constant β plane. These tests further evaluate our scheme on a wide range of possible scales, ranging from the microscale with the rising thermal bubble test to the global scale with the baroclinic instability.

a. Rising thermal bubble

The 2D (x, z) rising thermal bubble test case is essentially ubiquitous in the study of nonhydrostatic mesoscale models. This test follows the evolution of a warm bubble in a constant potential temperature environment. The warm bubble leads to a positive perturbation in the vertical velocity field, which acts to carry the bubble upward. As the bubble moves upward, shearing quickly deforms the circular bubble into a mushroom cloud. Here, we follow the initialization procedure described by Giraldo and Restelli (2008), which is a variation of the bubble experiments of Robert (1993). Because no explicit diffusion is added to this test, we do not anticipate that the solution will converge as spatial resolution is refined. However, at finer resolutions we do observe more finescale features of the thermal bubble, including tighter winding of the trailing edges at later times and sharper spatial gradients. Nonetheless, our comparisons for this test case are purely qualitative.

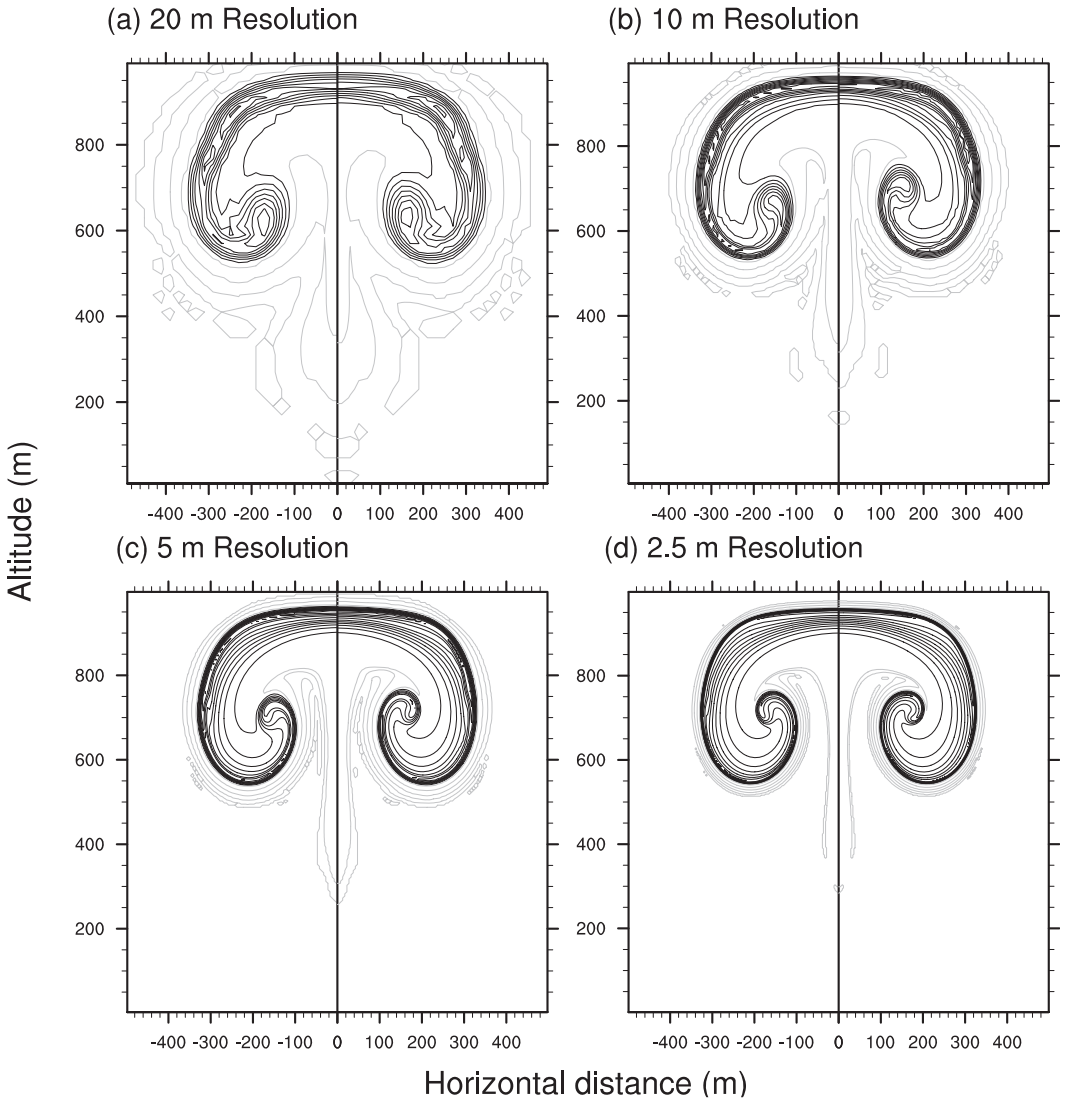

FIG. 3. As in Fig. 2, but using the five-point vertical stencil given by (51) and (52), showing (left) the Strang carryover scheme and (right) the ARS(2, 3, 3) scheme on the same axes.

The background consists of a constant potential temperature field $\bar{\theta} = 300$K, with a small perturbation of the form

$$\theta' = \begin{cases} 0 & \text{for} \quad r > r_c \\ \dfrac{\theta c}{2}\left[1 + \cos\left(\dfrac{\pi r}{rc}\right)\right] & \text{for} \quad r > r_c \end{cases}, \tag{100}$$

where

$$r = \sqrt{(x - x_c)^2 + (z - z_c)^2}. \tag{101}$$

Here, we choose the amplitude and radius of the perturbation to be $\theta_c = 0.5$ K and $r_c = 250$ m, respectively. The domain consists of a square region $(x, z) \in [-500, 500] \times [0, 1000]$ m with $t \in [0, 700]$ s. The center point of the bubble is located at $x_c = 500$ m and $z_c = 350$ m. The boundary conditions are no flux along all boundaries. No sponge layer or viscous forcing is used, and Coriolis forces are set to zero.

The potential temperature perturbation at $t = 700$ s with crude splitting is plotted in Fig. 1. Four different resolutions $\Delta X = \Delta Z$ are shown that range between 20 and 2.5 m. At even the finest resolution, this scheme performs exceptionally poorly, unable to even resolve the correct convection velocity at low resolutions or the anticipated winding of the bubble's leading edges at higher resolutions. This approach is equivalent to low-order time splitting of the horizontal and vertical motions; its poor behavior may be indicative of problems with other low-order time-split approaches.

TABLE 2. Timing results from the rising thermal bubble test on a 50 × 50 grid (20-m resolution) with $\Delta t = 0.05$ s until $t = 700$ s using a variety of model configurations. Timing is normalized to the fastest configuration, which uses an explicit RK3 method for time stepping in both the horizontal and vertical. On a recent MacBook Pro with 2.2-GHz Intel Core i7 chip and two processors, this parallel configuration required 32 s to run.

Scheme	Jacobian	Vertical stencil	Vertical solver	Normalized time
Explicit RK3	—	3 point	—	1.000
	—	5 point	—	1.044
Strang carryover	Analytic	3 point	Rosenbrock	2.325
			Fully implicit	4.052
		5 point	Rosenbrock	2.911
			Fully implicit	5.204
	Numerical	3 point	Rosenbrock	5.471
			Fully implicit	10.435
		5 point	Rosenbrock	7.241
			Fully implicit	13.977
ARS(2, 3, 3)	Analytic	3 point	Rosenbrock	4.043
			Fully implicit	7.495
		5 point	Rosenbrock	5.211
			Fully implicit	9.812
	Numerical	3 point	Rosenbrock	10.360
			Fully implicit	20.260
		5 point	Rosenbrock	13.936
			Fully implicit	27.226

The Strang carryover and ARS(2, 3, 3) schemes perform significantly better. The results with these two methods are plotted in Fig. 2 for the three-point vertical stencil (53) and (54) and in Fig. 3 for the five-point vertical stencil (51) and (52). These plots show both time-stepping schemes on the same axes, because the symmetry of the bubble allows us to only plot half the domain. As a consequence, we can clearly see differences between these

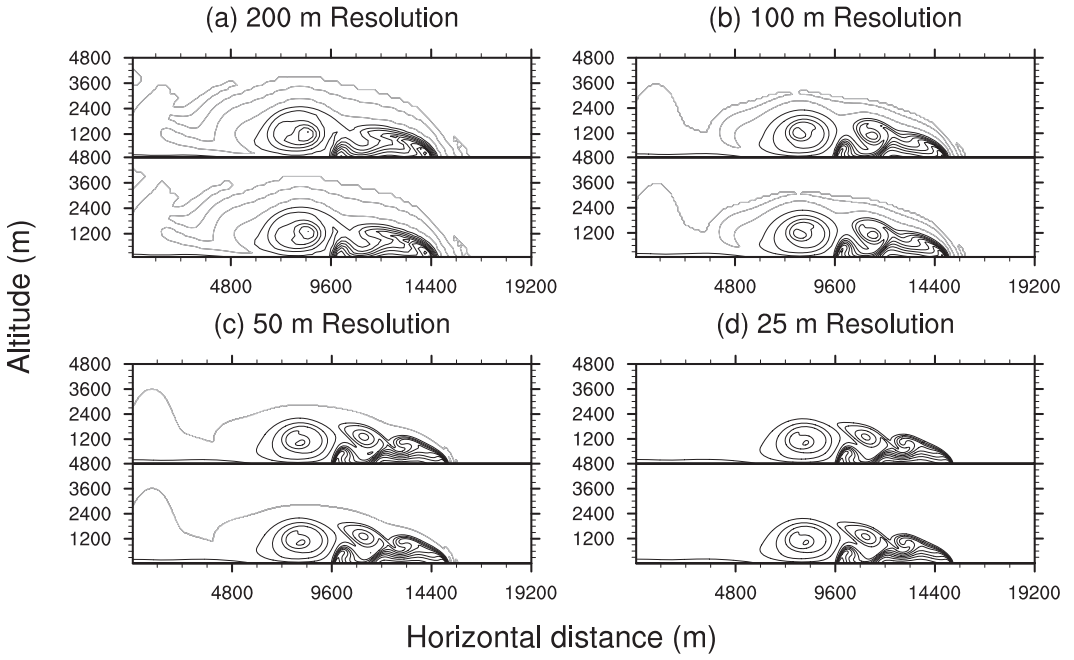

FIG. 4. Plots of the potential temperature perturbation for the Straka density current test case with three-point vertical stencil at time $t = 900$ s and four choices of resolution. The time step is 0.5 s for a grid spacing of 200 m and scales with the spatial resolution. Contour lines are from 291 to 300 K with a contour interval of 1 K. The 300-K contour line is shown in light gray to emphasize numerical oscillations. At each resolution, we show (top) the Strang carryover scheme and (bottom) the ARS(2, 3, 3) scheme for comparison.

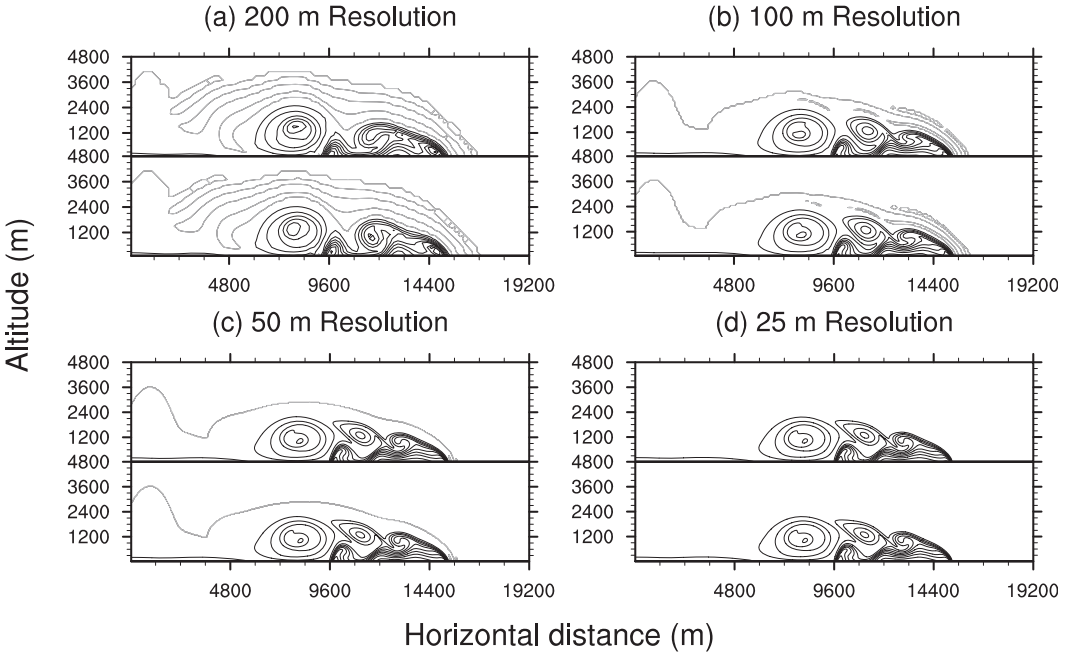

FIG. 5. As in Fig. 4, but with a five-point vertical stencil. Again, we plot (top) the Strang carryover scheme and (bottom) the ARS(2, 3, 3) scheme for comparison.

two time-stepping schemes in the final results. The plots are again made at four resolutions after 700 s. In both sets of plots, the ARS(2, 3, 3) scheme appears to perform better than the Strang carryover scheme, exhibiting slightly more winding and enhanced gradients, especially at lower resolutions. The results from using the five-point stencil are observably better than those with the three-point stencil, and so this wider stencil may be desirable in improving accuracy in real applications. Both of these schemes match the results reported in Giraldo and Restelli (2008) very closely.

Timing results for the rising thermal bubble test on a grid with 20-m horizontal and vertical resolution with $\Delta t = 0.05$ s are given in Table 2 for a variety of model configurations. We have tested out the Strang carryover and ARS(2, 3, 3) schemes with either a numerical or an analytic Jacobian, either a three-point or a five-point vertical stencil, and either a Rosenbrock solve or a fully implicit solve in the vertical. The timing results from the crude splitting scheme were not reported here because they very closely match the timing results from the Strang carryover scheme. As a baseline, we have run a fully explicit scheme that uses the RK3 method of Gottlieb et al. (2001) for the full set of nonhydrostatic equations. We observe that the cheapest RKR method is the Strang carryover scheme with analytic Jacobian, three-point vertical stencil, and Rosenbrock solver. This scheme is just over twice as expensive as the corresponding explicit method. The more accurate five-point stencil adds an overhead of approximately 25%. The fully implicit method approximately doubles the computation time, implying that roughly twice as many implicit solves are computed at each time step, but does not significantly increase the accuracy of the results. The ARS(2, 3, 3) scheme leads to a similar overhead, because it explicitly requires two implicit solves per time step.

b. Straka density current

The 2D (x–z) density current test proposed by Straka et al. (1993) considers the evolution of a cold bubble in a neutrally stratified atmosphere. In this setting, the bubble sinks and hits the ground, forming shearing currents, which then lead to the generation of Kelvin–Helmholtz rotors. Explicit viscosity with the coefficient $\nu = 75\ \mathrm{m^2\ s^{-1}}$ is added in order to obtain a grid-converged solution. Again, no sponge layers are used and Coriolis forcing is set to zero. The initial conditions are similar to the rising thermal bubble and consist of a constant potential temperature background with $\theta = 300$ K. The perturbation 712 in the temperature field is defined by

$$T' = \begin{cases} 0 & \text{if} \quad r > 1.0, \\ -T_c[1 + \cos(\pi r)] & \text{if} \quad r \le 1.0, \end{cases} \tag{102}$$

where $T_c = -15°\mathrm{C}$ and

$$r = \sqrt{\left(\frac{x - x_c}{x_r}\right)^2 + \left(\frac{z - z_c}{z_r}\right)^2}. \tag{103}$$

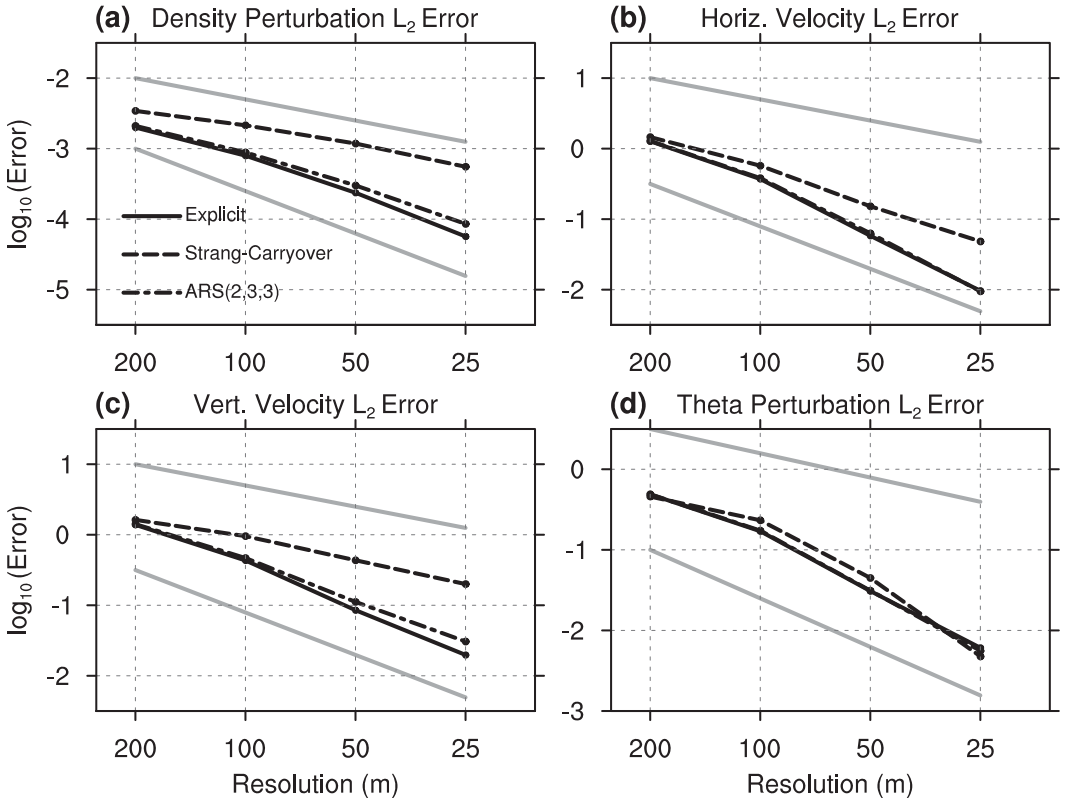

FIG. 6. Spatial convergence plots showing the L_2 difference in the density perturbation, horizontal velocity, vertical velocity, and potential temperature perturbation as a function of resolution and three-point vertical stencil and using the three-point vertical stencil. The solid line denotes the fully explicit scheme with RK3 time step, whereas the dashed line and dotted–dashed line correspond to the Strang carryover and ARS(2, 3, 3) schemes, respectively. The time step at 200 m is 0.5 s and is otherwise proportional to the grid spacing so as to maintain a constant CFL number. The gray lines denote perfect (top) first- and (bottom) second-order scaling rates as a function of grid spacing.

Note that following Straka et al. (1993) we apply the perturbation to the temperature field, whereas in Giraldo and Restelli (2008) the potential temperature is instead perturbed. This leads to an initial maximum potential temperature perturbation of −16.632 K when assuming a surface pressure of $p_0 = 1000$ hPa and hydrostatic background conditions. The domain is defined as $(x, z) \in [0, 25\ 600] \times [0, 6400]$ m with $t \in [0, 900]$ s. The center of the bubble is positioned at $(x_c, z_c) = (0, 3000)$ m with radius $(x_r, z_r) = (4000, 2000)$ m. The boundary conditions at all four boundaries are no flux. By defining the domain in this manner, we take advantage of the problem's symmetry by simulating only half of the problem domain.

The results of our model with the Strang carryover and ARS(2, 3, 3) schemes are plotted at $t = 900$ s in Fig. 4 for the three-point vertical stencil and in Fig. 5 for the five-point vertical stencil. Four resolutions $\Delta X = \Delta Z$ are shown. As with the rising thermal bubble, we observe a very slight improvement from using the ARS(2, 3, 3) scheme, which is more prominent at lower resolutions. Also, we observe a significant improvement at lower resolutions when using the five-point vertical stencil with both time-stepping schemes. The appearance of three well-defined Kelvin–Helmholtz rotors is not obvious until 50-m resolution with the three-point vertical stencil but is clear at 100-m resolution with the five-point vertical stencil. Both methods otherwise perform well with grid refinement.

A spatial convergence study has been performed to test the convergence properties of each of the schemes. The results are shown in Fig. 6 for the three-point vertical stencil and in Fig. 7 for the five-point vertical stencil. Here, we define the L_2 error analogous to Straka et al. (1993),

$$L_2(q) = \sqrt{\frac{1}{N_x N_z} \sum_{i=1}^{N_x} \sum_{k=1}^{N_z} [q_{i,k} - q_{i,k}^{(\mathrm{ref})}]^2}\ , \quad (104)$$

where N_x and N_z are the number of grid points in the horizontal and vertical directions, respectively. Here, $q_{i,k}^{(\mathrm{ref})}$ denotes a reference solution that is obtained from running this test case with the fully explicit scheme at

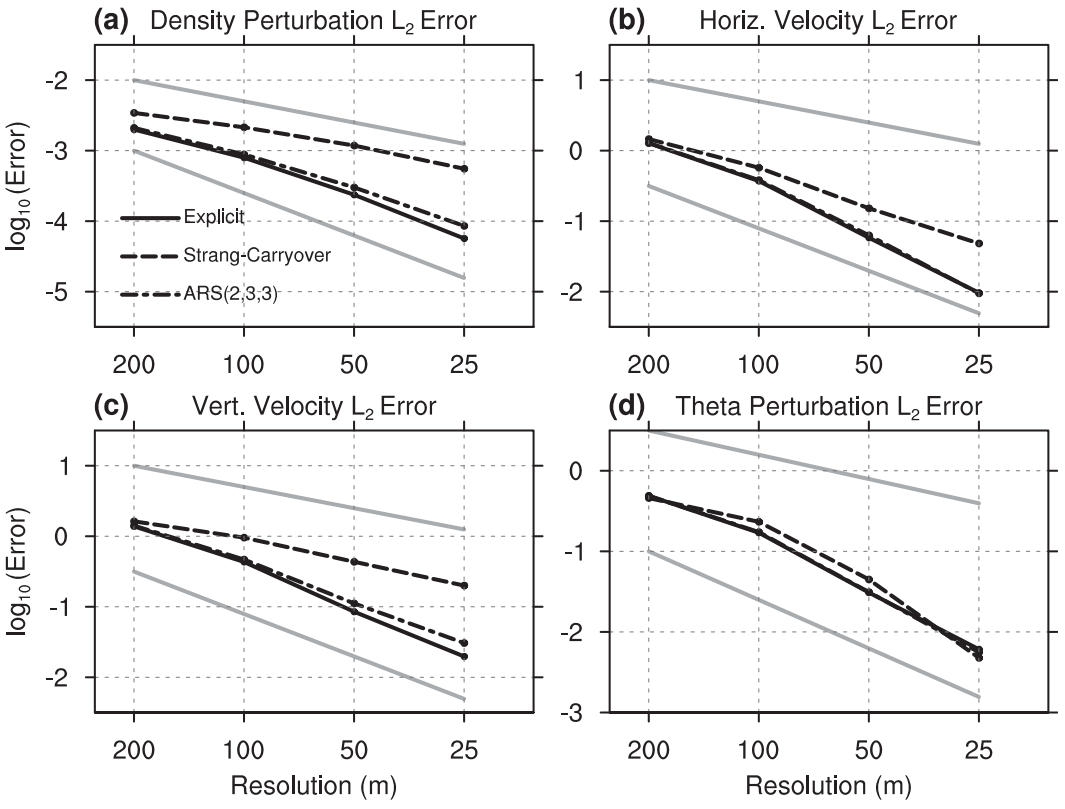

FIG. 7. As in Fig. 6, but for the five-point vertical stencil.

a resolution of 12.5 m. The variable q corresponds to each of $\{\rho', u, w, \theta'\}$.

We observe significantly different convergence behavior results for each of the four choices q. In fact, the fully explicit method, Strang carryover method, and ARS(2, 3, 3) method all seem to behave similarly when purely considering the perturbation in the potential temperature. Because the potential temperature perturbation is not a prognostic variable, its L_2 errors are a function of errors in both the density perturbation and the potential temperature density perturbation. In computing (104), these errors approximately cancel and so lead to very similar behavior among all time-stepping schemes. However, for each of the variables ρ', u, and w and for $(\rho\theta)'$ (not shown) the observed errors are significantly different between the Strang carryover and ARS(2, 3, 3) schemes. Although the Strang carryover scheme generally performs more poorly with the three-point vertical stencil, its spatial convergence rate roughly matches that of the other schemes. However, with the five-point vertical stencil (Fig. 5), we observe that the Strang carryover scheme seems to converge at least one order of accuracy more slowly than the other two schemes. The ARS(2, 3, 3) scheme closely mirrors the fully explicit scheme, suggesting better overall performance of this method.

The results from performing a temporal convergence analysis on this problem are given in Fig. 8. In this case, we set $\Delta X = \Delta Z = 100$ m = constant and adjust the time step size Δt to observe the rate of temporal convergence for each of the methods. The error norms are obtained by computing the L_2 norm (104) with a reference solution obtained by running the test case with $\Delta X = \Delta Z = 100$ m and $\Delta t = 1.5625 \times 10^{-2}$ s. For the density perturbation, horizontal velocity, and vertical velocity, we observe second-order convergence in Δt for the Strang carryover scheme and near-third-order convergence in Δt for the ARS(2, 3, 3) scheme. The potential temperature perturbation again shows different behavior because of cancelation in the computation of the L_2 errors but nonetheless yields improved errors under the ARS(2, 3, 3) scheme.

c. *Schär mountain*

The 2D ($x-z$) Schär et al. (2002) mountain wave test is characterized by a large hydrostatic wave with deep vertical propagation, which is overlaid by smaller-scale nonhydrostatic waves, which decay rapidly with height. These gravity waves are triggered by a wavelet-like mountain chain. It has been pointed out by Klemp et al. (2003) that a consistent treatment of metric terms is

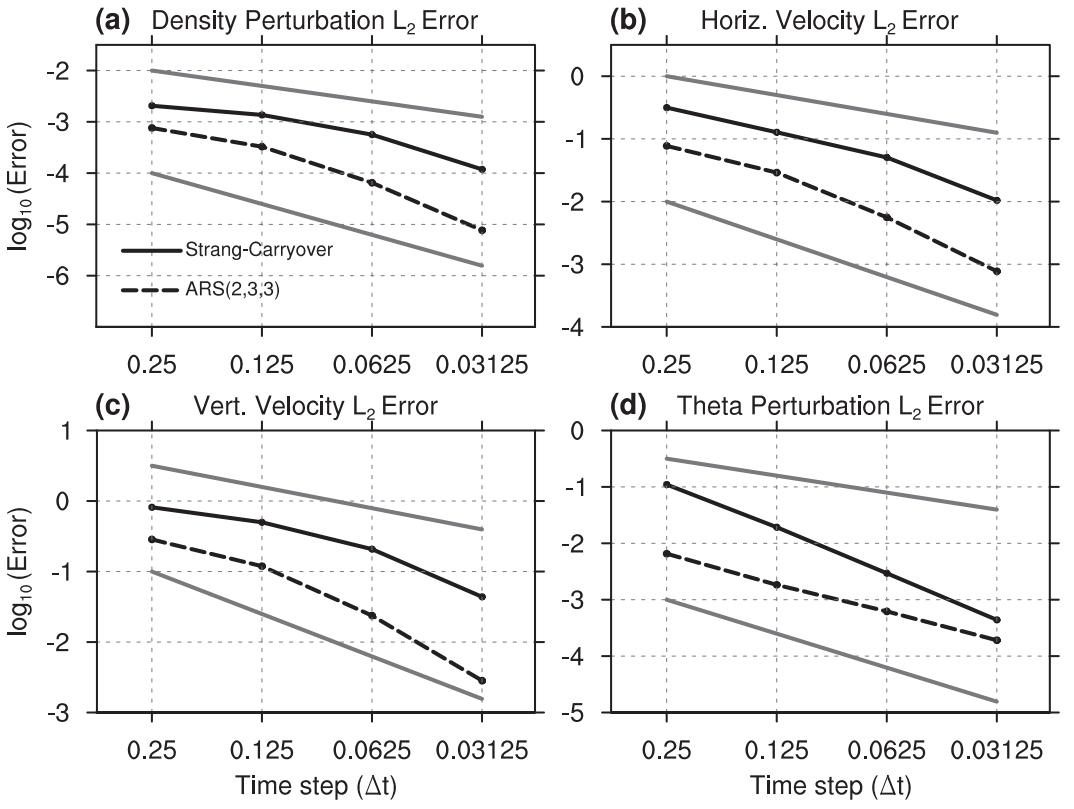

FIG. 8. Temporal convergence plots showing the L_2 error in the density perturbation, horizontal velocity, vertical velocity, and potential temperature perturbation as a function of time step Δt for a fixed grid spacing of $\Delta X = 100$ m. The solid line and dashed line correspond to the Strang carryover and ARS(2, 3, 3) schemes, respectively. The gray lines denote perfect (top) first- and (bottom) second-order scaling rates as a function of time step.

necessary to avoid triggering spurious waves that arise in this problem at coarse grid resolutions. As a consequence, this test has become standard in the mesoscale modeling community. The initial hydrostatic state of the atmosphere consists of a constant mean flow of $\overline{u} = 10\ \mathrm{m\ s^{-1}}$ in a uniformly stratified atmosphere with constant Brunt–Väisälä frequency of $\mathcal{N} = 0.01\ \mathrm{s^{-1}}$. The pressure and temperature at $z = 0$ are $p_0 = 1000$ hPa and $T_0 = 280$ K, respectively. The domain is defined as $(x, Z) \in [-25, 25] \times [0, 21]$ km with $t \in [0, 10]$ h. The mountain profile is defined by

$$h_T(x) = h_c \exp\left[-\left(\frac{x}{a_c}\right)^2\right] \cos^2\left(\frac{\pi x}{\lambda}\right), \qquad (105)$$

with parameters $h_c = 250$ m, $\lambda = 4000$ m, and $a_c = 5000$ m. A plot of the mountain profile is shown in Fig. 9, where the axes have been magnified to show the domain $(x, z) \in [-10\ 000, 10\ 000] \times [0, 4000]$ m. No-flux boundary conditions are used along the bottom surface, whereas nonreflecting boundary conditions are used along the top and lateral outflow boundary. The sponge extends upward from a height of 12 000 m and at the lateral outflow boundary with 10 000-m thickness. The background state $\mathbf{q}_b$ is defined by the hydrostatic background and constant mean flow. No explicit viscosity is used, and the Coriolis forcing is set to zero.

This test case is run at $\Delta X = 250$ m and $\Delta Z = 210$ m using the Strang carryover scheme and five-point vertical stencil. The observed horizontal and vertical wind speeds after 10 h are plotted in Fig. 10. The fields are smooth and agree well with published results in Klemp et al. (2003) and Giraldo and Restelli (2008). In particular, we do not

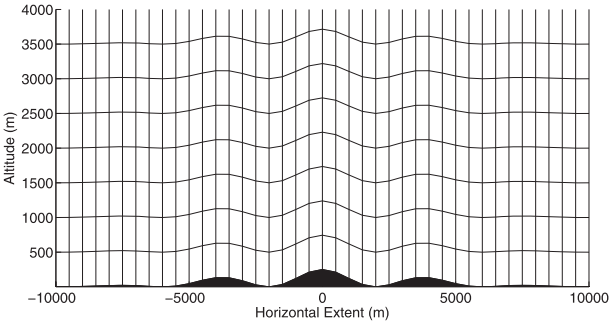

FIG. 9. The Schär mountain profile with $\Delta X = \Delta Z = 500$ m and magnified such that $(x, z) \in [-10\ 000, 10\ 000] \times [0, 4000]$ m.

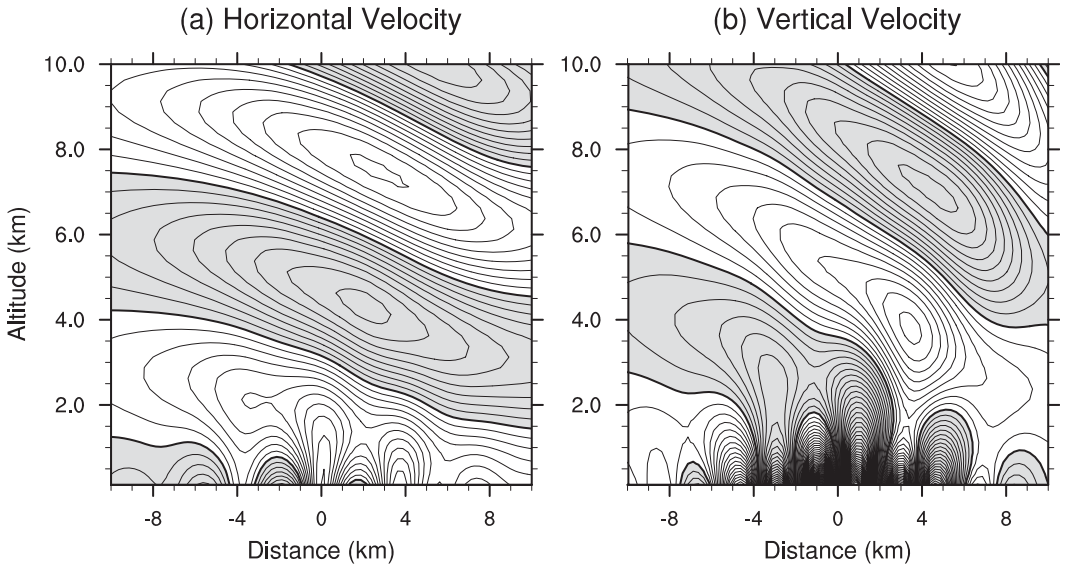

FIG. 10. Steady-state flow over the Schär mountain after 10 h with $\Delta X = 250$-m- and $\Delta Z = 210$-m resolution, 5-point vertical stencil, and Strang carryover scheme. The simulation is run to $t = 10$ h with a time step of $\Delta t = 0.6$ s. (a) The horizontal velocity has contour values between 8 m s^{-1} and 12 m s^{-1} with a contour interval of 0.2 m s^{-1} with emphasis on the 10 m s^{-1} contour. (b) The vertical velocity has contour values between -2 and 2 m s^{-1} with a contour interval of 0.05 m s^{-1} with emphasis on the 0 m s^{-1} contour. In (a), values less than 10 m s^{-1} are shaded. In (b), negative values are shaded.

observe any artifacts from the inhomogeneous terms described by Klemp et al. (2003), which would introduce spurious noise at coarse grid resolutions. As expected for steady-state solutions, the ARS(2, 3, 3) scheme produces qualitatively similar results.

d. Flow over an isolated mountain

The 2D (x, z) mountain wave test suite of Dudhia (1993) considers a variety of flow regimes in the presence of a smooth single-peaked mountain with constant inflow and outflow boundary conditions. The background consists of an atmospheric profile with constant Brunt–Väisälä frequency $\mathcal{N} = 0.01$ s^{-1} and nonzero Coriolis parameter $f = 10^{-4}$ s^{-1}. The pressure and temperature at $z = 0$ are $p_0 = 1000$ hPa and $T_0 = 300$ K, respectively. Topography is added in the form of a witch of Agnesi mountain,

$$h(x) = \frac{h_c}{1 + \left(\dfrac{x - x_c}{a_c}\right)^2}, \tag{106}$$

where the maximum height is $h_c = 400$ m, the center position is $x_c = 0$ m, and the mountain half-width a_c is varied. A constant mean flow $\overline{u} = 10$ m s^{-1} is then imposed in all flow regimes. The initial profile is assumed to be in hydrostatic and geostrophic balance and incorporates a meridional pressure gradient to exactly balance the Coriolis forcing due to the mean flow. The flow is evolved until $t\overline{u}/a_c = 21.6$.

The simulation domain is taken to be $(x, Z) \in [-14a_c, 14a_c] \times [0, 21\,000]$ m, and the time step is chosen to be $\Delta t = 0.006a_c/\overline{u}$, which requires 3600 time steps at all resolutions with $\Delta X = a_c/5$ and $\Delta Z = 420$ m. Nonreflecting boundary conditions are applied at the upper boundary above 16 km and at the outflow boundary with thickness 2 km. The background state is given by the hydrostatic background with prescribed mean flow. No explicit viscosity is used.

The results of these simulations using the Strang carryover time-stepping scheme are plotted in Fig. 11 for four choices of a_c: $a_c = 1$ km (Fig. 11a), $a_c = 10$ km (Fig. 11b), $a_c = 100$ km (Fig. 11c), and $a_c = 1000$ km (Fig. 11d). We observe agreement with the results of Dudhia (1993) in all four flow regimes, suggesting our model is correctly capturing the dynamics of these regimes. Further, our model appears stable even with a vertical CFL number of nearly 500 as in Fig. 11d. The total computation time is also observed to be roughly equivalent in all cases.

e. Steady-state geostrophically balanced flow in a channel

Our last test addresses a large-scale 3D flow field, to evaluate the full 3D response of the channel model (see also section 5f). The flow field for steady-state

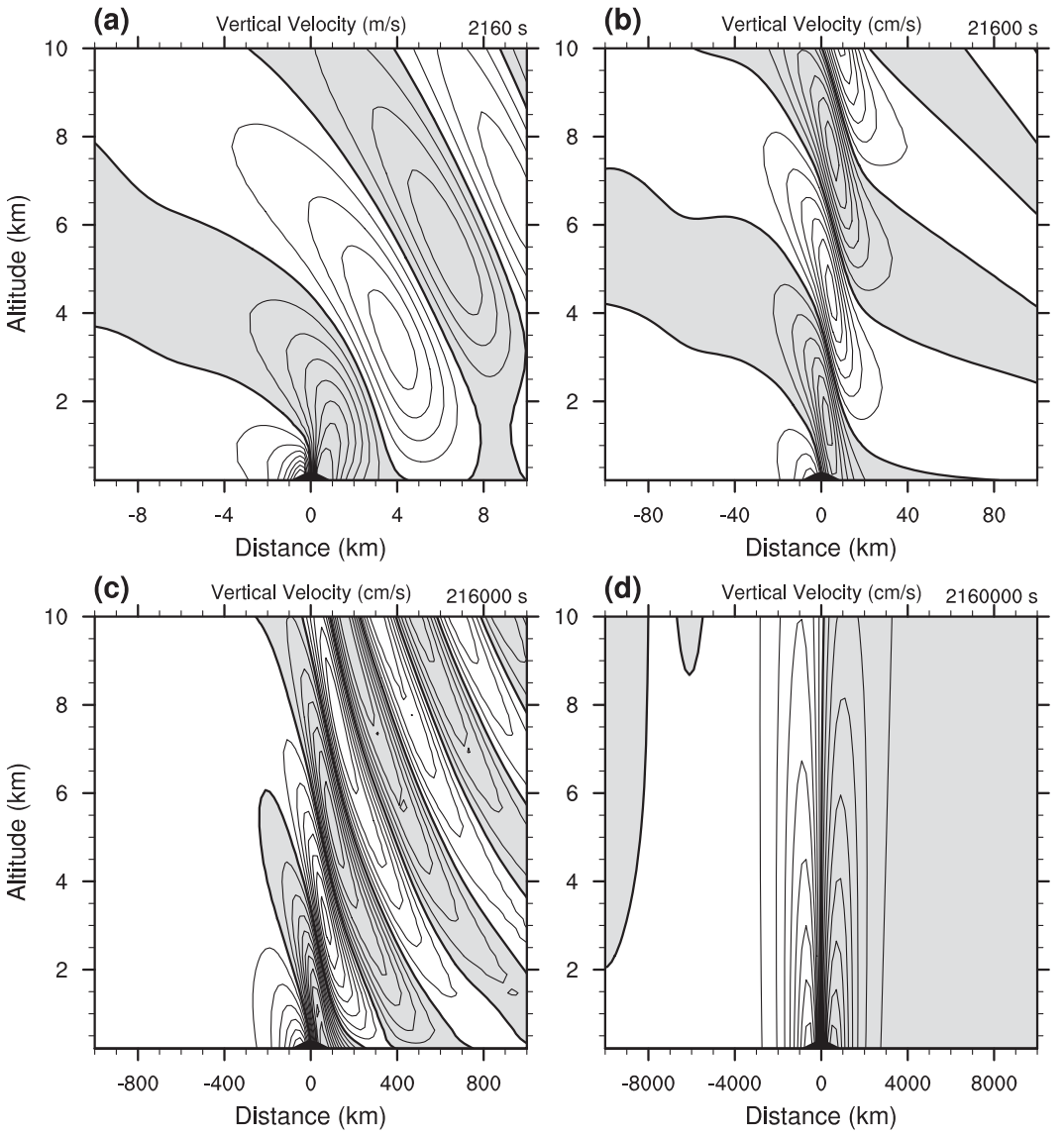

FIG. 11. Plots of vertical velocity for the Dudhia (1993) mountain test case with background flow speed u = 10 m s^{-1} and mountain half-width a_c given by (a) 1, (b) 10, (c) 100, and (d) 1000 km. Grid spacing is taken to be $\Delta X = a_c/5$ and $\Delta Z = 420$ m, leading to a maximum aspect ratio in (d) of $\Delta X/\Delta Z = 476$. The simulation is run up to $t = 21.6a_c/u$ with a fixed time step of $\Delta t = 0.006a_c/\bar{u}$. Contour lines are (a) from −2.1 to 1.2 m s^{-1} in intervals of 0.3 m s^{-1}, (b) from −42 to 42 cm s^{-1} in intervals of 7 cm s^{-1}, (c) from 3.6 to 2.4 cm s^{-1} in intervals of 0.4 cm s^{-1}, and (d) from −0.24 to 0.24 cm s^{-1} in units of 0.03 cm s^{-1}. Negative values are indicated by shaded regions.

geostrophically balanced flow in a channel is based on a new test case defined by C. Jablonowski et al. (2011, unpublished manuscript). The domain is a channel of dimensions $L_x \times L_y \times L_z$ with periodic boundaries in the x direction and no-flux conditions at all other interfaces. In this case, we choose L_x = 40 000 km, L_y = 6000 km, and L_z = 30 km. No viscosity or sponge layers have been applied. The initial flow is comprised of a zonally symmetric midlatitudinal jet, defined in terms of vertical pressure-based η coordinates (see the appendix) as

$$u(x, y, \eta) = -u_0 \sin^2\left(\frac{\pi y}{L_y}\right) \ln\eta \exp\left[-\left(\frac{\ln\eta}{b}\right)^2\right], \qquad (107)$$

so that the wind is zero at the surface and along the y boundary. The vertical half-width is set to $b = 2$, and u_0 is chosen to be 35 m s^{-1}. The meridional wind velocity v and vertical wind velocity w are both set to zero. We take the surface pressure to be constant with $p_s = p_0 = 10^5$ Pa. This formulation can be on either an f plane or a β plane, which have Coriolis parameters

TABLE 3. Relative errors in the vertical momentum field $\rho\omega$ and potential temperature density field $\rho\theta$ for the geostrophically balanced flow in a channel test with an *f*-plane approximation and Strang carryover time-stepping scheme. Errors are represented in scientific notation using the form *a* (*b*) for mantissa *a* and exponent *b*, under base 10. A convergence study is performed by varying the horizontal resolution. The computed order of accuracy is obtained from a least squares fit through the data.

Z momentum field $\rho\omega$			
Horizontal resolution	L_1 error	L_2 error	L_∞ error
400 km	1.821 (−8)	5.932 (−8)	4.248 (−7)
200 km	1.819 (−9)	6.054 (−9)	5.131 (−8)
100 km	1.238 (−10)	4.124 (−10)	3.483 (−10)
50 km	7.916 (−12)	2.637 (−11)	2.381 (−10)
Order	3.738	3.728	3.628
Potential temperature density field $\rho\theta$			
Horizontal resolution	L_1 error	L_2 error	L_∞ error
400 km	4.114 (−6)	1.213 (−5)	6.682 (−5)
200 km	2.039 (−7)	5.840 (−7)	3.052 (−6)
100 km	1.245 (−8)	3.474 (−8)	1.725 (−7)
50 km	7.798 (−10)	2.173 (−9)	1.113 (−8)
Order	4.113	4.141	4.180

$$f = f_0, \quad \text{and} \quad \beta = f_0 + \beta_0(y - y_0), \tag{108}$$

respectively, where $f_0 = 2\Omega \sin\varphi_0$ and $\beta_0 = 2a^{-1}\Omega\cos\varphi$ at latitude $\varphi_0 = 45°$N. Here, the radius of the earth is $a = 6371.229 \times 10^3$ m, its angular velocity is $\Omega = 7.292 \times 10^{-5}\ \mathrm{s}^{-1}$, and $y_0 = L_y/2$ is the center point of the domain in the y direction. The background geopotential field is again defined in terms of η coordinates is

$$\Phi(x, y, \eta) = \langle\Phi(\eta)\rangle + \Phi'(x, y,) \ln\eta \exp\left[-\left(\frac{\ln\eta}{b}\right)^2\right], \tag{109}$$

with the horizontal-mean geopotential

$$\langle\Phi(\eta)\rangle = \frac{T_0 g}{\Gamma}(1 - \eta^{R_d\Gamma/g}) \tag{110}$$

and variation

$$\Phi'(x, y) = \frac{u_0}{2}\left\{(f_0 - \beta_0 y_0)\left[y - \frac{L_y}{2} - \frac{L_y}{2\pi}\sin\left(\frac{2\pi y}{L_y}\right)\right] + \frac{\beta_0}{2}\left[y^2 - \frac{L_y y}{\pi}\sin\left(\frac{2\pi y}{L_y}\right) - \frac{L_y^2}{2\pi^2}\cos\left(\frac{2\pi y}{L_y}\right) - \frac{L_y^2}{3}\frac{L_y^2}{2\pi^2}\right]\right\}. \tag{111}$$

The reference temperature is $T_0 = 288$ K and the lapse rate is chosen to be $\Gamma = 0.005\ \mathrm{K\ m^{-1}}$. The corresponding temperature distribution is given by

TABLE 4. As in Table 3, but for the β-plane approximation.

Z momentum field ρw			
Horizontal resolution	L_1 error	L_2 error	L_∞ error
400 km	1.489 (−8)	4.763 (−8)	4.473 (−7)
200 km	1.799 (−9)	5.024 (−9)	3.496 (−8)
100 km	1.356 (−10)	3.851 (−10)	2.953 (−9)
50 km	9.125 (−12)	2.659 (−11)	2.054 (−10)
Order	3.575	3.613	3.683
Potential temperature density field $\rho\theta$			
Horizontal resolution	L_1 error	L_2 error	L_∞ error
400 km	1.339 (−5)	3.912 (−5)	2.156 (−4)
200 km	8.803 (−7)	2.601 (−6)	1.570 (−5)
100 km	5.525 (−8)	1.642 (−7)	1.015 (−6)
50 km	3.456 (−9)	1.029 (−8)	6.367 (−8)
Order	3.975	3.966	3.913

$$T(x, y, \eta) = \langle T(\eta)\rangle + \frac{\Phi'(x, y)}{R_d}\left[\frac{2}{b^2}(\ln\eta)^2 - 1\right]\exp\left[-\frac{\ln\eta}{b}\right], \tag{112}$$

with horizontal-mean temperature

$$\langle T(\eta)\rangle = T_0\eta^{R_d\Gamma/g}. \tag{113}$$

This test considers the steady-state problem where the solution is the initial state. Hence, error measures are calculated in the height field via the usual global error norms,

$$L_1(q) = \frac{I[|q - q_T|]}{I[|q_T|]}, \tag{114}$$

$$L_2(q) = \sqrt{\frac{I[(q - q_T)^2]}{I[q_T^2]}}, \quad \text{and} \tag{115}$$

$$L_\infty(q) = \frac{\max|q - q_T|}{\max|q_T|}, \tag{116}$$

where q_T is the field at the initial time and I denotes an approximation to the global integral, which is given by

$$I[x] = \sum_{\text{allcells}k} q_k V_k, \tag{117}$$

with V_k denoting the volume of element k.

Error norms are given in Table 3 for an *f*-plane approximation and Table 4 for the β-plane approximation. The simulations are run with a variable horizontal resolution of 400, 200, 100, and 50 km and a uniform vertical resolution of 1 km (30 equally spaced vertical levels) for one day. The time step at 400-km resolution is 960 s and otherwise scales with resolution. We observe convergence

that is slightly less than fourth order for the vertical momentum field and slightly better than fourth order for the potential temperature field. Because hydrostatic balance is guaranteed by the background splitting technique described in section 2, errors are only accumulated because of an imbalance in the geostrophically balanced components, and hence increasing the number of vertical levels does not have a significant impact on the error norms. Discrepancies in these errors at each vertical level trigger the slight imbalances in the vertical velocity. Because this is a steady-state test case, the errors in this analysis are dominated by errors in the spatial reconstruction, and so similar error norms are observed with both the crude and ARS(2, 3, 3) time-stepping schemes.

f. Baroclinic instability in a channel

This test case uses the same geostrophically balanced background as in section 5e. However, we additionally introduce a confined perturbation in the zonal wind field of the form

$$u'(x, y, \eta) = u_p \exp\left\{-\left[\frac{(x - x_c)^2 + (y - y_c)^2}{L_p^2}\right]\right\}, \quad (118)$$

with radius $L_p = 600$ km, maximum amplitude $u_p = 1$ m s^{-1}, and center point $(x_c, y_c) = (2000 \text{ km}, 2500 \text{ km})$. This perturbation is superimposed on the zonal wind field (107) so that the total zonal wind field reads

$$u_{\text{new}}(x, y, \eta) = u(x, y, \eta) + u'(x, y, \eta). \quad (119)$$

The setup resembles the baroclinic wave experiments on the sphere suggested by Jablonowski and Williamson (2006). The unbalanced perturbation acts as a trigger for baroclinic waves that grow explosively over a 10–12-day simulation period. Such a flow is characteristic for the midlatitudes. The channel test thereby assesses how well the finite-volume scheme simulates large-scale flow fields with large aspect ratios. All simulations are run with the ARS(2, 3, 3) scheme and utilize a 100-km horizontal grid spacing with 30 equally spaced vertical levels and a model top at 30 km. The Strang carryover scheme yields results that are qualitatively similar at this resolution. Again, no explicit viscosity or sponge layer is used. The time step is $\Delta t = 240$ s. Timing results for this test case are given in Table 5 and demonstrate the benefits of using the RKR methods over a purely explicit approach for this problem.

Snapshots of the simulation for the *f*-plane approximation at day 12 are plotted in Fig. 12. The figure depicts the horizontal cross sections of the pressure, temperature, and relative vorticity at 500 m. This vertical position corresponds to the height of the lowermost model level. We observe that the baroclinic wave has almost broken, which takes place around day 13.5. The flow has formed distinct low and high pressure systems that are associated with sharp temperature fronts and sharp gradients in the relative vorticity field.

TABLE 5. Timing results from the baroclinic instability in a channel on an $80 \times 12 \times 30$ grid ($\Delta X = 500$ km, $\Delta Z = 1$ km) and three-point vertical stencil. Timing is normalized to the fastest configuration, which uses the Strang carryover scheme and analytic Jacobian. On a recent MacBook Pro with 2.2-GHz Intel Core i7 chip and two processors, this parallel configuration required 6.8 s day^{-1}.

Scheme	Time step (Δt)	Normalized Time
Explicit RK3	3.0 s	214.777
Strang carryover	1200.0 s	1.000
ARS(2, 3, 3)	1200.0 s	1.569

The corresponding simulation results on the β plane are plotted in Fig. 13. Here, we show the identical fields, but now at day 10 before wave breaking events set in (around day 11). The presence of the planetary vorticity gradient has sped up the evolution of the baroclinic wave. Again, the low and high pressure systems are connected to sharp frontal zones in the temperature and vorticity fields that resemble realistic flow conditions. It is interesting to note that the β-plane simulation leads to a more confined flow field that has not spread to the northern and southern edges of the domain by day 10. These differences between the *f*-plane and β-plane simulations will be discussed in greater detail in C. Jablonowski et al. (2011, unpublished manuscript) alongside a comparison to other nonhydrostatic channel models. The main focus of this test is to demonstrate that the RKR schemes reliably simulate the evolution of atmospheric flow fields that are relevant for the large (midlatitudinal) portion of global atmospheric general circulation models. To verify correctness of our results, this test has also been run using the model of Norman et al. (2011) and qualitatively similar results were observed.

6. Conclusions

In this paper, we have presented a new approach for discretizing the nonhydrostatic Euler equations in Cartesian geometry using high-order finite-volume methods and a horizontal–vertical splitting strategy based on Runge–Kutta–Rosenbrock (RKR) time integration schemes. For atmospheric problems where the vertical grid spacing is usually much smaller than the horizontal, this strategy allows us to simulate the full Euler equations while only constraining the time step by the horizontal grid spacing. We have presented time-stepping schemes based on a crude strategy, a scheme that uses

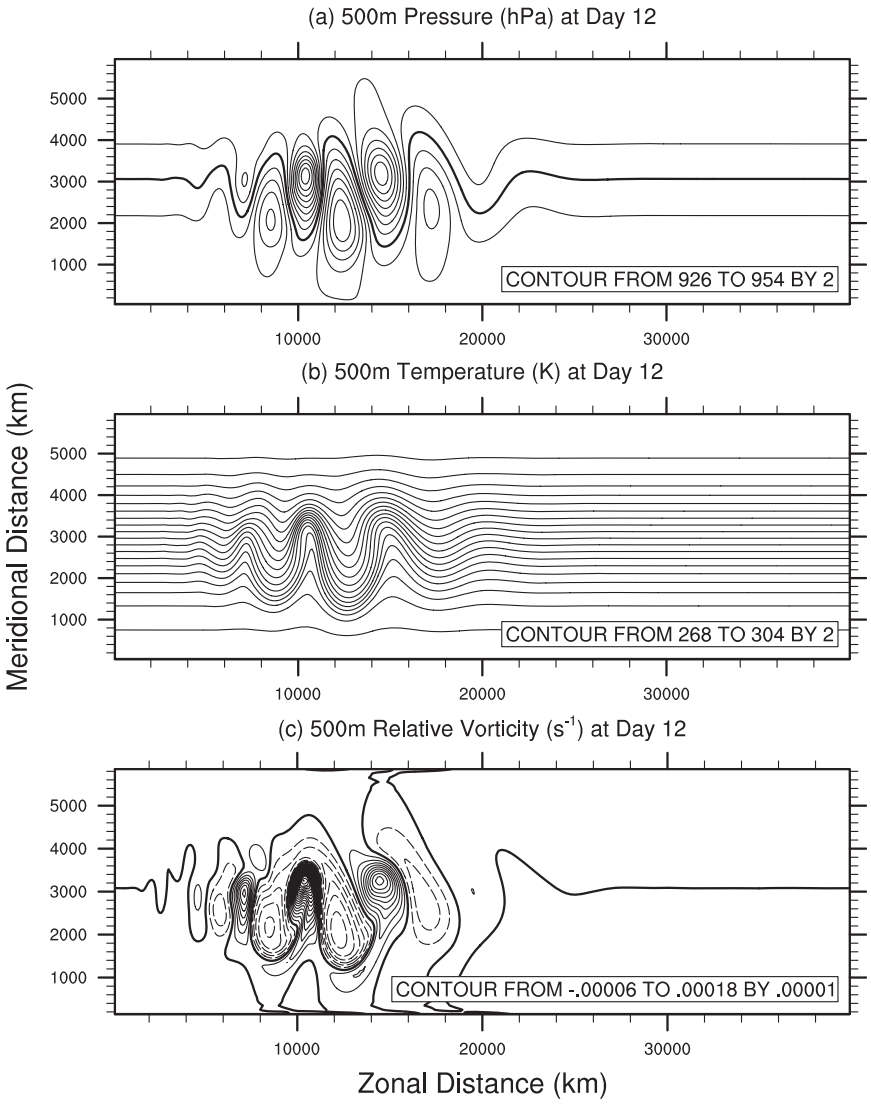

FIG. 12. Simulation results from the baroclinic instability in a channel computed at day 12 using the ARS(2, 3, 3) scheme with the *f*-plane approximation. The simulation is run at a horizontal resolution of 100 km and a vertical resolution of 1 km with a time step of 240 s. Contour lines are as indicated on each plot. The 942-hPa line is enhanced in the pressure plot. The zero line in the relative vorticity plot is enhanced, and negative values are plotted using dashed lines.

a Strang splitting and carryover strategy, and a higher-order scheme based on an approach attributed to Ascher et al. (1997). These time-stepping schemes have been implemented in a mesoscale atmospheric model that utilizes a high-order finite-volume-based approach. The crude time-stepping scheme is shown to be highly diffusive for thermal bubble experiments and shows no benefit over the Strang carryover scheme, which requires the same number of explicit and implicit steps per time step. The ARS(2, 3, 3) scheme shows a mild improvement over the Strang carryover approach but requires two implicit steps per time step. However, the higher-order accuracy in time this scheme affords may be desirable. Further, we compared a three-point and a five-point vertical reconstruction stencil within the model and observed significantly better results with the five-point stencil, but with a slight computational overhead.

Numerical results have shown our approach to be accurate, stable, and applicable to a range of atmospheric flows and horizontal–vertical aspect ratios. By using a fourth-order reconstruction strategy in the horizontal, we observe clear fourth-order convergence for flows with a small vertical velocity. Horizontal–vertical aspect ratios up to 500:1 have been tested under our scheme and verified to be stable up to a horizontal CFL number of 1.0.

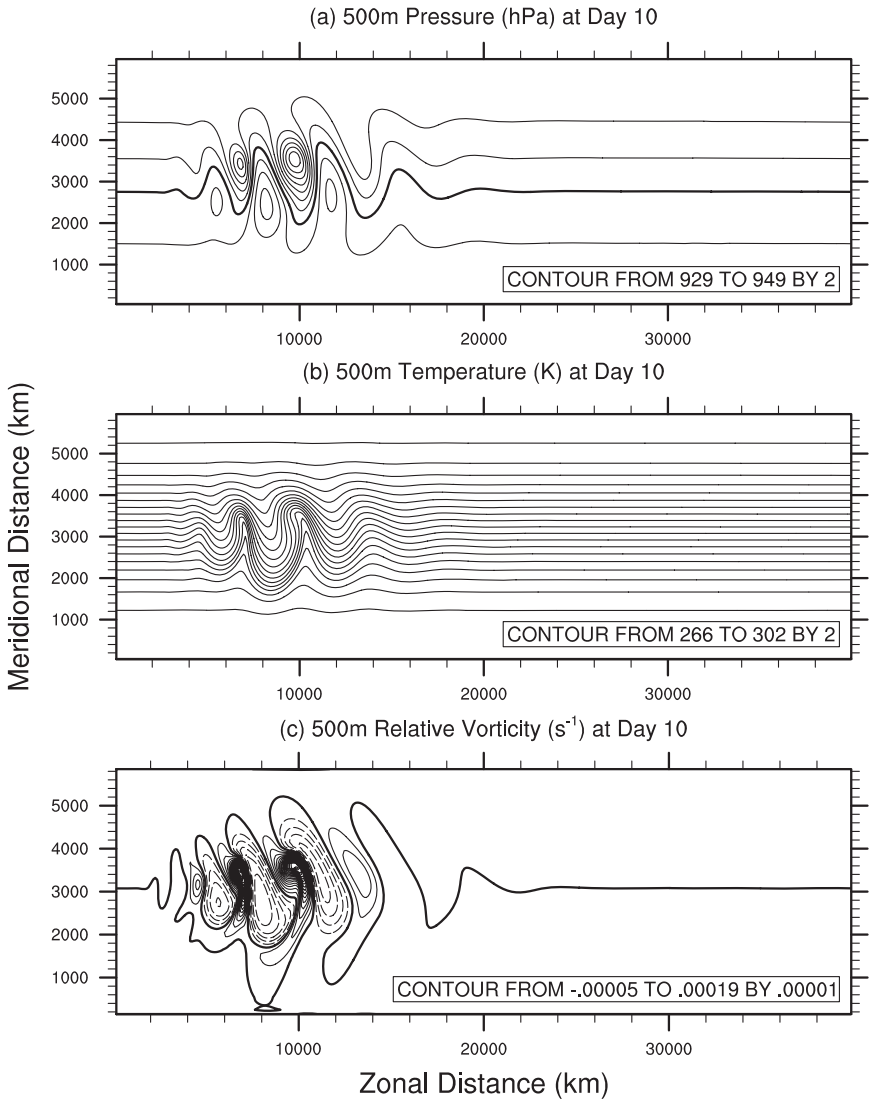

FIG. 13. As in Fig. 12, but for the β-plane approximation at day 10. The 943-hPa line is enhanced in the pressure plot. The zero line in the relative vorticity plot is enhanced, and negative values are plotted using dashed lines.

The results in this paper suggest that our horizontal–vertical dimension splitting strategy is a promising option for any high-order finite-volume or discontinuous Galerkin-based method. This model has already been extended to a full global nonhydrostatic dynamical core on a cubed-sphere grid utilizing high-order finite-volume methods in Ullrich and Jablonowski (2012, manuscript submitted to *J. Comput. Phys.*).

Acknowledgments. The authors would like to acknowledge the contributions of Steve Ruuth, whose fruitful comments helped in developing the Strang carryover approach for this paper. We would also like to thank Matthew Norman for his work implementing the baroclinic instability test and providing a comparison for our results. Finally, we thank the two anonymous reviewers for their very helpful commentary in improving this work. This work was supported by the Office of Science, U.S. Department of Energy, Award DE-SC0003990 and a University of Michigan Rackham Predoctoral fellowship.

APPENDIX

Converting between η and z Coordinates

The initial conditions required for the geostrophically balanced flow (section 5e) and baroclinic instability test cases (section 5f) are given in terms of pressure-based

vertical coordinates with $\eta = p/p_0$ with $p_0 = 1000$ hPa. To convert these coordinates to height-based z coordinates, we must implicitly solve a nonlinear equation relating η and z. Here, we choose to use the iterative Newton–Raphson strategy, which is given by

$$\eta^{n+1} = \eta^n - \left[\frac{\partial F}{\partial \eta}(x, y, \eta^n)\right]^{-1} F(x, y, \eta^n), \qquad \text{(A1)}$$

where $n = 0, 1, 2, \ldots$ is the iteration count. Here, the functions F and $\partial F/\partial \eta$ are determined by

$$F(x, y, \eta^n) = -gz + \Phi(x, y, \eta^n) \quad \text{and} \qquad \text{(A2)}$$

$$\frac{\partial F}{\partial \eta}(x, y, \eta^n) = -\frac{R_d}{\eta^n} T(x, y, \eta^n). \qquad \text{(A3)}$$

Here, Φ and T are given by (109) and (112), respectively. The starting value of $\eta^0 = 10^{-7}$ is used for all Newton iterations, corresponding to a model top of about 100 km. If a higher model top is required, the value of η^0 needs to be decreased. Convergence is deemed to have occurred if $|\eta^{n+1} - \eta^n| \leq 10^{-14}$ and usually takes about 10 iterations in most cases.

To compute pressure p, density ρ, and potential temperature θ as a function of position (x, y, z), we first solve for $\eta(x, y, z)$ via the iterative technique and then apply

$$p(x, y, \eta) = \eta(x, y, z) p_0, \qquad \text{(A4)}$$

$$\rho(x, y, \eta) = \frac{p(x, y, \eta)}{R_d T(x, y, \eta)}, \quad \text{and} \qquad \text{(A5)}$$

$$\theta(x, y, \eta) = T(x, y, \eta)\left[\frac{p_0}{p(x, y, \eta)}\right]^{R_d/c_p}. \qquad \text{(A6)}$$

REFERENCES

Adcroft, A., C. Hill, and J. Marshall, 1997: Representation of topography by shaved cells in a height coordinate ocean model. *Mon. Wea. Rev.,* **125,** 2293–2315.

Ahmad, N., and J. Lindeman, 2007: Euler solutions using flux-based wave decomposition. *Int. J. Numer. Methods Fluids,* **54,** 47–72, doi:10.1002/fld.1392.

Arakawa, A., and C. S. Konor, 2009: Unification of the anelastic and quasi-hydrostatic systems of equations. *Mon. Wea. Rev.,* **137,** 710–726.

Ascher, U. M., S. J. Ruuth, and R. J. Spiteri, 1997: Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Appl. Numer. Math.,* **25** (2–3), 151–167, doi:10.1016/S0168-9274(97)00056-1.

Barad, M., and P. Colella, 2005: A fourth-order accurate local refinement method for Poisson's equation. *J. Comput. Phys.,* **209,** 1–18, doi:10.1016/j.jcp.2005.02.027.

Bonaventura, L., 2000: A semi-implicit semi-Lagrangian scheme using the height coordinate for a nonhydrostatic and fully elastic model of atmospheric flows. *J. Comput. Phys.,* **158,** 186–213, doi:10.1006/jcph.1999.6414.

Crouzeix, M., 1980: Une méthode multipas implicite-explicite pour l'approximation des équations d'évolution paraboliques. *Numerische Math.,* **35,** 257–276, doi:10.1007/BF01396412.

Davies, T., A. Staniforth, N. Wood, and J. Thuburn, 2003: Validity of anelastic and other equation sets as inferred from normal-mode analysis. *Quart. J. Roy. Meteor. Soc.,* **129,** 2761–2775, doi:10.1256/qj.02.1951.

Dudhia, J., 1993: A nonhydrostatic version of the Penn State NCAR Mesoscale Model: Validation tests and simulation of an Atlantic cyclone and cold front. *Mon. Wea. Rev.,* **121,** 1493–1513.

Durran, D. R., 1989: Improving the anelastic approximation. *J. Atmos. Sci.,* **46,** 1453–1461.

——, 1999: *Numerical Methods for Wave Equations in Geophysical Fluid Dynamics*. Springer-Verlag, 489 pp.

Gadd, A. J., 1978: A split explicit integration scheme for numerical weather prediction. *Quart. J. Roy. Meteor. Soc.,* **104,** 569–582, doi:10.1002/qj.49710444103.

Gal-Chen, T., and R. C. J. Somerville, 1975: On the use of a coordinate transformation for the solution of the Navier-Stokes equations. *J. Comput. Phys.,* **17,** 209–228, doi:10.1016/0021-9991(75)90037-6.

Giraldo, F. X., 2005: Semi-implicit time-integrators for a scalable spectral element atmospheric model. *Quart. J. Roy. Meteor. Soc.,* **131,** 2431–2454, doi:10.1256/qj.03.218.

——, and M. Restelli, 2008: A study of spectral element and discontinuous Galerkin methods for the Navier–Stokes equations in nonhydrostatic mesoscale atmo spheric modeling: Equation sets and test cases. *J. Comput. Phys.,* **227,** 3849–3877, doi:10.1016/j.jcp.2007.12.009.

Gottlieb, S., C.-W. Shu, and E. Tadmor, 2001: Strong stability-preserving high-order time discretization methods. *SIAM Rev.,* **43,** 89–112, doi:10.1137/S003614450036757X.

Jablonowski, C., and D. L. Williamson, 2006: A baroclinic instabilitiy test case for atmospheric model dynamical cores. *Quart. J. Roy. Meteor. Soc.,* **132,** 2943–2975.

Jameson, A., W. Schmidt, and E. Turkel, 1981: Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes. *Proc. 14th Fluid and Plasma Dynamics Conf.,* Palo Alto, CA, AIAA, 15 pp.

Jebens, S., O. Knoth, and R. Weiner, 2011: Partially implicit peer methods for the compressible Euler equations. *J. Comput. Phys.,* **230,** 4955–4974, doi:10.1016/j.jcp.2011.03.015.

Klein, R., U. Achatz, D. Bresch, O. M. Knio, and P. K. Smolarkiewicz, 2010: Regime of validity of soundproof atmospheric flow models. *J. Atmos. Sci.,* **67,** 3226–3237.

Klemp, J. B., W. C. Skamarock, and O. Fuhrer, 2003: Numerical consistency of metric terms in terrain-following coordinates. *Mon. Wea. Rev.,* **131,** 1229–1239.

Kwizak, M., and A. J. Robert, 1971: A semi-implicit scheme for grid point atmospheric models of the primitive equations. *Mon. Wea. Rev.,* **99,** 32–36.

Lanser, D., J. G. Blom, and J. G. Verwer, 2001: Time integration of the shallow water equations in spherical geometry. *J. Comput. Phys.,* **171,** 373–393, doi:10.1006/jcph.2001.6802.

Lin, S., 2004: A "vertically Lagrangian" finite-volume dynamical core for global models. *Mon. Wea. Rev.,* **132,** 2293–2307.

Liou, M.-S., 2006: A sequel to AUSM. Part II: AUSM$^+$-up for all speeds. *J. Comput. Phys.,* **214,** 137–170, doi:10.1016/j.jcp.2005.09.020.

McCorquodale, P., and P. Colella, 2011: A high-order finite-volume method for conservation laws on locally refined grids. *Commun. Appl. Math. Comput. Sci.,* **6,** 1–25.

Norman, M. R., R. D. Nair, and F. H. M. Semazzi, 2011: A low communication and large time step explicit finite-volume solver for non-hydrostatic atmospheric dynamics. *J. Comput. Phys.,* **230,** 1567–1584, doi:10.1016/j.jcp.2010.11.022.

Nørsett, S. P., and A. Wolfbrandt, 1979: Order conditions for Rosenbrock type methods. *Numerische Math.,* **32,** 1–15, doi:10.1007/BF01397646.

Ogura, Y., and N. A. Phillips, 1962: Scale analysis of deep and shallow convection in the atmosphere. *J. Atmos. Sci.,* **19,** 173–179.

Restelli, M., and F. X. Giraldo, 2009: A conservative discontinuous Galerkin semi-implicit formulation for the Navier–Stokes equations in nonhydrostatic mesoscale modeling. *SIAM J. Sci. Comput.,* **31,** 2231–2257, doi:10.1137/070708470.

Robert, A., 1993: Bubble convection experiments with a semi-implicit formulation of the Euler equations. *J. Atmos. Sci.,* **50,** 1865–1873.

Roe, P. L., 1981: Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.,* **43,** 357–372, doi:10.1016/0021-9991(81)90128-5.

Rosenbrock, H., 1963: Some general implicit processes for the numerical solution of differential equations. *Comput. J.,* **5,** 329–330.

Rusanov, V., 1961: Calculation of intersection of non-steady shock waves with obstacles. *J. Comput. Math. Phys. USSR,* **1,** 267–279.

Sandu, A., J. G. Verwer, J. G. Blom, E. J. Spee, G. R. Carmichael, and F. A. Potra, 1997: Benchmarking stiff ode solvers for atmospheric chemistry problems II: Rosenbrock solvers. *Atmos. Environ.,* **31,** 3459–3472, doi:10.1016/S1352-2310(97)83212-8.

Schär, C., D. Leuenberger, O. Fuhrer, D. Lüthi, and C. Girard, 2002: A new terrain-following vertical coordinate formulation for atmospheric prediction models. *Mon. Wea. Rev.,* **130,** 2459–2480.

Schiesser, W. E., 1991: *The Numerical Method of Lines: Integration of Partial Differential Equations*. Academic Press, 326 pp.

——, and G. W. Griffiths, 2009: *A Compendium of Partial Differential Equation Models: Method of Lines Analysis with Matlab*. Cambridge University Press, 490 pp.

Skamarock, W. C., and J. B. Klemp, 2008: A time-split nonhydrostatic atmospheric model for weather research and forecasting applications. *J. Comput. Phys.,* **227,** 3465–3485, doi:10.1016/j.jcp.2007.01.037.

St-Cyr, A., and D. Neckels, 2009: A fully implicit Jacobian-free high-order discontinuous Galerkin mesoscale flow solver. *Proc. Ninth Int. Conf. on Computational Science,* Baton Rouge, LA, Intel, 243–252.

Straka, J. M., R. B. Wilhelmson, L. J. Wicker, J. R. Anderson, and K. K. Droegemeier, 1993: Numerical solutions of a nonlinear density current: A benchmark solution and comparisons. *Int. J. Numer. Methods Fluids,* **17,** 1–22, doi:10.1002/fld.1650170103.

Taylor, M. A., J. Edwards, and A. St Cyr, 2008: Petascale atmospheric models for the Community Climate System Model: New developments and evaluation of scalable dynamical cores. *J. Phys. Conf. Ser.,* **125,** 012023, doi:10.1088/1742-6596/125/1/012023.

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

Varah, J. M., 1980: Stability restrictions on second order, three level finite difference schemes for parabolic equations. *SIAM J. Numer. Anal.,* **17,** 300–309, doi:10.1137/0717025.

Verwer, J. G., E. J. Spee, J. G. Blom, and W. Hundsdorfer, 1999: A second-order Rosenbrock method applied to photochemical dispersion problems. *SIAM J. Sci. Comput.,* **20,** 1456–1480, doi:10.1137/S1064827597326651.

Williamson, D. L., 2008: Equivalent finite volume and Eulerian spectral transform horizontal resolutions established from aqua-planet simulations. *Tellus,* **60A,** 839–847, doi:10.1111/j.1600-0870.2008.00340.x.