IMPLICIT-EXPLICIT FORMULATIONS OF A THREE-DIMENSIONAL NONHYDROSTATIC UNIFIED MODEL OF THE ATMOSPHERE (NUMA)

F.X. GIRALDO *, J.F. KELLY †, AND E.M. CONSTANTINESCU ‡

Key words. cloud-resolving model; compressible flow; element-based Galerkin methods; Euler; global model; IMEX; Lagrange; Legendre; mesoscale model; Navier-Stokes; nonhydrostatic; semi-implicit; spectral elements; time-integration.

AMS subject classifications. 65M60, 65M70, 35L65, 86A10

Abstract. We derive an implicit-explicit (IMEX) formalism for the three-dimensional Euler equations that allow a unified representation of various nonhydrostatic flow regimes, including cloud-resolving and mesoscale (flow in a 3D Cartesian domain) as well as global regimes (flow in spherical geometries). This general IMEX formalism admits numerous types of methods including single-stage multi-step methods (e.g., Adams methods and backward difference formulas) and multi-stage single-step methods (e.g., additive Runge-Kutta methods). The significance of this result is that it allows a numerical model to reuse the same machinery for all classes of time-integration methods described in this work. We also derive two classes of IMEX methods, 1D and 3D, and show that they achieve their expected theoretical rates of convergence regardless of the geometry (e.g., 3D box or sphere) and introduce a new second-order IMEX Runge-Kutta method that performs better than the other second order methods considered. We then compare all the IMEX methods in terms of accuracy and efficiency for two types of geophysical fluid dynamics problems: buoyant convection and inertia-gravity waves. These results show that the high-order time-integration methods yield better efficiency particularly when high levels of accuracy are desired.

1. Introduction. In a previous article [20] we introduced the Nonhydrostatic Unified Model of the Atmosphere (NUMA) for use in limited-area modeling (i.e., mesoscale or regional flow), namely, applications in which the flows are in large, three-dimensional Cartesian domains (imagine flow in a 3D box where the grid resolutions are below 10 km); the emphasis of that paper was on the performance of the model on distributed-memory computers with a large number of processors. In that paper we showed that the explicit RK35 time-integrator (also used in this paper) was able to achieve strong linear scaling for processor counts on the order of 10^5. The emphasis of the present article is on the mathematical framework of the model dynamics (i.e., we are not considering the subgrid-scale parameterization at this point; moisture has already been included in a 2D version of the model, see [9]) that allows for a unification across various metrics. NUMA is unified in terms of spatial discretization methods and can use high-order continuous and discontinuous Galerkin methods [12, 20]; in this paper we only consider high-order continuous Galerkin methods. NUMA is also unified across multiple scales in that it has been designed as a cloud-resolving model (resolution of less than 1 km), mesoscale model (resolution of 1 km to tens of km), and global model (resolution of tens to hundreds of km) typical for climate and global weather prediction applications. To be unified across these disparate scales a model must understand the differences between flow taking place inside a 3D Cartesian domain as well as flow taking place in a domain comprised of concentric spheres as is required in global atmospheric modeling. The principal challenge is that the model must account for the direction in which gravity and Coriolis act. Additionally, the time-integrators must be specifically designed for efficiency due to the difference in the vertical and horizontal scales.

In this paper, we present the unified equations with a suite of time-integrators for the different types of simulations. We include explicit time-integrators, implicit-explicit (IMEX) methods developed for fast waves in all directions (three-dimensions), and for fast waves in the vertical direction (one dimension). These IMEX methods can be recast in the general framework of multirate methods (see, e.g., [28, 16]) where the operators are partitioned into fast and slow moving processes.

The remainder of the paper is organized as follows. In Sec. 2 the form of the governing equations used is described, including the splitting of the variables into reference and perturbation states that simplifies the separation of the slow and fast waves. Section 3 is the heart of this paper and is where we describe the general implicit-explicit (IMEX) time-integration strategy that allows us to include any type of IMEX method into our formulation (and model), including 1D and 3D IMEX methods,

*Department of Applied Mathematics, Naval Postgraduate School, Monterey, CA 93943,
†Exa Corp., Burlington MA 01803,
‡Mathematics and Computer Science Division Argonne National Laboratory, Argonne IL 60439

as well as multi-step and multi-stage methods. In Sec. 4 we show numerical results of our model using the suite of IMEX time-integrators described in Sec. 3. We use three test cases that cover the range of problems of particular interest to us: cloud-resolving, mesoscale, and global simulations. In Sec. 5 we present a summary of our findings and discuss directions for future work.

We begin by describing the governing equations used in our study and discuss in detail the separation of the multi-scale processes (i.e., fast and slow waves).

2. Governing Equations. The Euler equations can be written in a various ways (see [14] for other possibilities) but, based on [14], we have chosen to use the following form:

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) = 0 \tag{2.1a}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} + \frac{1}{\rho} \boldsymbol{\nabla} P + g \bar{\boldsymbol{r}} + f \bar{\boldsymbol{r}} \times \boldsymbol{u} = \boldsymbol{0} \tag{2.1b}$$

$$\frac{\partial \theta}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \theta = 0 \tag{2.1c}$$

where the prognostic variables are $(\rho, \boldsymbol{u}^{\mathcal{T}}, \theta)^{\mathcal{T}}$ and ρ is the density, $\boldsymbol{u} = (u, v, w)^{\mathcal{T}}$ is the Cartesian velocity field, θ is the potential temperature, $\boldsymbol{\nabla} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)^{\mathcal{T}}$ is the three-dimensional gradient operator, $\bar{\boldsymbol{r}} = (r_x, r_y, r_z)^{\mathcal{T}}$ is the unit vector pointing in the radial direction, f is the Coriolis parameter, and $\boldsymbol{0} \in \mathcal{R}^3$ is the zero-vector of $\dim(\boldsymbol{u}) = 3$. In mesoscale mode (i.e., flow in a box) $\bar{\boldsymbol{r}} = \bar{\boldsymbol{k}}$, the unit vector along the z direction, and in global mode (i.e., flow on a spherical volume) $\bar{\boldsymbol{r}} = \frac{\boldsymbol{x}}{||\boldsymbol{x}||_2}$, where $\boldsymbol{x}$ is the grid point coordinate in Cartesian space and $||\cdot||_2$ is the 2-norm. The pressure P that appears in the momentum equation is obtained from the equation of state

$$P = P_A \left(\frac{\rho R \theta}{P_A} \right)^{\gamma}$$

where P_A is the atmospheric pressure at the ground. We note that we define the governing equations in 3D Cartesian coordinates regardless of the type of geometry we use (i.e., whether the domain is a 3D box or spherical).

Introducing the following splitting of the density $\rho(\boldsymbol{x}, t) = \rho_0(\boldsymbol{x}) + \rho'(\boldsymbol{x}, t)$, potential temperature $\theta(\boldsymbol{x}, t) = \theta_0(\boldsymbol{x}) + \theta'(\boldsymbol{x}, t)$, and pressure $P(\boldsymbol{x}, t) = P_0(\boldsymbol{x}) + P'(\boldsymbol{x}, t)$ where the reference values are in hydrostatic balance, i.e., $\frac{\partial P_0}{\partial r} = -\rho_0 g$, we can rewrite Eq. (2.1) as

$$\frac{\partial \rho'}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \rho' + \boldsymbol{u} \cdot \boldsymbol{\nabla} \rho_0 + (\rho' + \rho_0) \boldsymbol{\nabla} \cdot \boldsymbol{u} = 0 \tag{2.2a}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} + \frac{1}{\rho' + \rho_0} \left(\boldsymbol{\nabla} P' + \boldsymbol{\mathcal{H}} \boldsymbol{\nabla} P_0 \right) + \frac{\rho'}{\rho' + \rho_0} g \bar{\boldsymbol{r}} + f \bar{\boldsymbol{r}} \times \boldsymbol{u} = \boldsymbol{0} \tag{2.2b}$$

$$\frac{\partial \theta'}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \theta' + \boldsymbol{u} \cdot \boldsymbol{\nabla} \theta_0 = 0, \tag{2.2c}$$

where

$$\boldsymbol{\mathcal{H}} = \boldsymbol{I} - \bar{\boldsymbol{r}} \bar{\boldsymbol{r}}^{\mathcal{T}}$$

is an orthogonal projector (it is both idempotent and self-adjoint) that enforces the *hydrostatic* balance by eliminating the term in $\boldsymbol{\nabla} P_0$ that is along the $\bar{\boldsymbol{r}}$ direction, which cancels the buoyancy term $\rho_0 g \bar{\boldsymbol{r}}$ (where $\boldsymbol{I}$ in $\boldsymbol{\mathcal{H}}$ is the rank-3 identity matrix). If the reference pressure P_0 is defined to be in perfect hydrostatic balance, then the reference pressure gradient in Eq. (2.2b) will vanish. The reason for maintaining this term is in case a different reference field is used (e.g., one that enforces both hydrostatic *AND* geostrophic balance). The geometric interpretation of the projector $\boldsymbol{\mathcal{H}}$ is that of only taking into the account the shadow (i.e., projection) of the vector $\boldsymbol{\nabla} P_0$ formed by shining a light along the $\bar{\boldsymbol{r}}$ direction; the derivation of these equations is described in Appendix A. Having described the form of the governing equations that we use, let us now turn to the construction of the implicit-explicit time-integration strategy.

3. Implicit-Explicit Time-Integration. The governing equations can be written in the compact vector form

$$\frac{\partial \boldsymbol{q}}{\partial t} = S(\boldsymbol{q}), \tag{3.1}$$

where $\boldsymbol{q} = (\rho', \boldsymbol{u}^{\mathcal{T}}, \theta')^{\mathcal{T}}$ and the right-hand side $S(\boldsymbol{q})$ represents the remaining terms in the equations apart from the time derivatives. To obtain the implicit-explicit (IMEX) time-discretization of Eq. (3.1), we introduce a linear operator $L(\boldsymbol{q})$ that approximates $S(\boldsymbol{q})$ and contains the terms responsible for the acoustic and gravity waves (the precise form is defined in Sect. 3.5.1). We then rewrite Eq. (3.1) as

$$\frac{\partial \boldsymbol{q}}{\partial t} = \{S(\boldsymbol{q}) - \delta L(\boldsymbol{q})\} + [\delta L(\boldsymbol{q})] \tag{3.2}$$

and discretize explicitly in time the terms in curly brackets and implicitly those in square brackets. The parameter δ is introduced in Eq. (3.2) in order to obtain a unified formalism for IMEX discretizations: implicit-explicit for $\delta = 1$ and fully explicit for $\delta = 0$.

To advance (3.2) in time, we consider IMEX linear multi-step [2, 18] and multi-stage schemes [1, 21, 26].

3.1. IMEX Linear Multi-step Methods. As was done in [10, 11] we now consider a generic K-step (multi-step method) discretization of Eq. (3.2) of the form

$$\boldsymbol{q}^{n+1} = \sum_{k=0}^{K-1} \check{\alpha}_k \boldsymbol{q}^{n-k} + \chi \Delta t \sum_{k=0}^{K-1} \check{\beta}_k [S(\boldsymbol{q}^{n-k}) - \delta L(\boldsymbol{q}^{n-k})] + \chi \Delta t \delta L(\boldsymbol{q}^{n+1}), \tag{3.3}$$

where Δt is the time step, assumed to be constant for simplicity, and $\boldsymbol{q}^n$ denotes the solution at time $n\Delta t$, for $n = 0, 1, \ldots$ To simplify the discussion of the IMEX formulation, we now introduce the following variables:

$$\boldsymbol{q}_{tt} = \boldsymbol{q}^{n+1} - \sum_{k=0}^{K-1} \check{\beta}_k \boldsymbol{q}^{n-k}, \;\; \widehat{\boldsymbol{q}} = \boldsymbol{q}^E - \sum_{k=0}^{K-1} \check{\beta}_k \boldsymbol{q}^{n-k}, \;\; \boldsymbol{q}^E = \sum_{k=0}^{K-1} \check{\alpha}_k \boldsymbol{q}^{n-k} + \chi \Delta t \sum_{k=0}^{K-1} \check{\beta}_k S(\boldsymbol{q}^{n-k}). \tag{3.4}$$

These then allow us to write Eq. (3.2) as

$$\boldsymbol{q}_{tt} = \widehat{\boldsymbol{q}} + \lambda L(\boldsymbol{q}_{tt}), \tag{3.5}$$

where $\lambda = \chi \Delta t \delta$. For example, the coefficients for the second-order backward-difference-formula (BDF2) method, assuming constant time-stepping, are $\check{\alpha}_0 = 4/3$, $\check{\alpha}_1 = -1/3$, $\chi = 2/3$, $\check{\beta}_0 = 2$, and $\check{\beta}_1 = -1$ (see [13] for BDF-K methods of orders one through six); in this work we use BDF2 as one of the multi-step methods for our study. Using the fact that L is a linear operator, one can write any IMEX multi-step scheme [7, 18] as (3.3). For example, the other multi-step method that we use for our study is the AI2*/AB3 scheme (which we denote as AI2) of Durran and Blossey [7] defined by $\check{\alpha}_0 = 1$, $\check{\alpha}_1 = 0$, $\chi = 1$, $\check{\beta}_0 = 23/12$, $\check{\beta}_1 = -16/12$, and $\check{\beta}_2 = 5/12$. Although we only consider two multi-step IMEX methods we note that any other multi-step method can be included in our formulation described above.

Ideally, one would like to balance the errors between space and time (and boundary conditions), as we show in [24] for a simple equation. We do not use BDFs of higher order than 2 because they are not A-stable (e.g., see [13]); therefore, this means that the time-integrator will likely dominate the solution error because we tend to use much higher order in space (e.g., 4th through 8th order) in the continuous/discontinuous Galerkin methods. Hence, one of the challenges in the development of time-integrators for higher spatial discretization methods is to design high-order time-integrators that are accurate, at least A-stable and efficient under some metric. Toward this goal, we also consider high-order (up to 4th order) IMEX Runge-Kutta methods.

The crux of the IMEX method, as is evident in Eq. (3.2), is the derivation of the linear operator L. The success of the method depends on this operator which must be chosen such that the fastest

waves in the system are retained, albeit in their linearized form. If the correct operator L is not obtained, the method will not work effectively. Fortunately, deriving the linear operator is rather straightforward; we show how to derive such an operator in [13].

Moving from multi-step to multi-stage methods allows us to use high-order L- and A-stable time-integrators (for a discussion on A- and L-stability, see, e.g., [22, 17]). In Sec. 3.2 we show that our generalized IMEX formalism also accommodates multi-stage methods. Although not shown, our formalism can also be used to include combinations of multi-step and mult-stage such as the method used in, e.g., [31].

3.2. IMEX Linear Multi-stage Methods. Implicit-explicit multi-stage schemes, such as Runge-Kutta, have been developed in the same fashion as the IMEX linear multi-step methods [1, 21, 26]. When applied to such partitioned problems as Eq. (3.2), Runge-Kutta methods are sometimes referred to as additive Runge-Kutta (ARK). The idea is to use two different integrators for the nonstiff and the stiff terms, respectively. An implicit integrator will be used for the stiff part (square brackets in Eq. (3.2)) that represents the acoustic and gravity waves, whereas an explicit one will be used for the nonstiff part (curly brackets in Eq. (3.2)) that represents the advective terms. Singly Diagonally implicit s-stage ARK methods (or SDIRK) can be represented as

$$Y^{(i)} = y^n + \Delta t \sum_{j=1}^{i-1} a_{ij} f\left(Y^{(j)}\right) + \Delta t \sum_{j=1}^{i} \tilde{a}_{ij} g\left(Y^{(j)}\right), \qquad i = 1, \ldots, s \tag{3.6a}$$

$$y^{n+1} = y^n + \Delta t \sum_{i=1}^{s} b_i f\left(Y^{(i)}\right) + \Delta t \sum_{i=1}^{s} \tilde{b}_i g\left(Y^{(i)}\right), \tag{3.6b}$$

where $f(\boldsymbol{q}) = S(\boldsymbol{q}) - \delta L(\boldsymbol{q})$ is the explicitly treated nonstiff part with coefficients $A = \{a_{ij}\}$, $b = \{b_i\}$ and $g(\boldsymbol{q}) = \delta L(\boldsymbol{q})$ is the implicit stiff part with coefficients $\tilde{A} = \{\tilde{a}_{ij}\}$, $\tilde{b} = \{\tilde{b}_i\}$. The two integrators defined by (A, b) and ($\tilde{A}$, $\tilde{b}$) are constructed so that both have the same order of consistency by themselves just as well as the compound method (A, $\tilde{A}$, b, $\tilde{b}$). ARK methods are represented compactly by the following two Butcher tableaux [3]:

$$\begin{array}{c|c} c & A \\ \hline & b^{\mathcal{T}} \end{array} \qquad \begin{array}{c|c} \tilde{c} & \tilde{A} \\ \hline & \tilde{b}^{\mathcal{T}} \end{array},$$

where the abscissas $c_i = \sum_j a_{ij}$ and $\tilde{c}_i = \sum_j \tilde{a}_{ij}$ represent the time when f and g are evaluated, respectively; that is, at each stage the functions are evaluated at $t + c_i \Delta t$ and $t + \tilde{c}_i \Delta t$.

In contrast with linear multi-step schemes, ARK methods require a few implicit solves per step, which is equal to the cardinality of $\{\tilde{a}_{ii} : \tilde{a}_{ii} \neq 0, i = 1, \ldots, s\}$. However, the implicit part of ARK schemes can achieve A- and L-stability properties of arbitrary (high) order and are no longer subject to the stability barriers of the linear multi-step methods.

If the stiff component is linear, when solving Eq. (3.2), one can formulate an ARK scheme by using a similar formulation to that in Eq. (3.3). An s-stage ARK scheme applied to Eq. (3.2) has the following form:

$$\begin{aligned} \boldsymbol{Q}^{(i)} =& \boldsymbol{q}^n + \Delta t \sum_{j=1}^{i-1} a_{ij} \left(S(\boldsymbol{Q}^{(j)}) - \delta L(\boldsymbol{Q}^{(j)})\right) + \Delta t \sum_{j=1}^{i-1} \tilde{a}_{ij} \left(\delta L(\boldsymbol{Q}^{(j)})\right) \\ &+ \Delta t\, \tilde{a}_{ii} \left(\delta L(\boldsymbol{Q}^{(i)})\right), \; i = 1, \ldots, s, \end{aligned} \tag{3.7a}$$

$$\boldsymbol{q}^{n+1} = \boldsymbol{q}^n + \Delta t \sum_{i=1}^{s} b_i S(\boldsymbol{Q}^{(i)}), \tag{3.7b}$$

where we assume that $b = \tilde{b}$ which is a necessary condition for the conservation of linear invariants; this will be shown to be important in Sec. 4.4 [1]

[1] It should be mentioned that none of the methods presented in this work conserve quadratic invariants which is

To write the IMEX form as in Eq. (3.5) requires us to define for each stage $i = 1, \ldots, s$ and $\tilde{a}_{ii} \neq 0$ the following IMEX variables (defined for the multi-step methods in Eq. (3.4))

$$\boldsymbol{q}_{tt} = \boldsymbol{Q}^{(i)} + \sum_{j=1}^{i-1} \frac{\tilde{a}_{ij} - a_{ij}}{\tilde{a}_{ii}} \boldsymbol{Q}^{(j)} \,, \tag{3.8a}$$

$$\widehat{\boldsymbol{q}} = \boldsymbol{q}^{E} + \sum_{j=1}^{i-1} \frac{\tilde{a}_{ij} - a_{ij}}{\tilde{a}_{ii}} \boldsymbol{Q}^{(j)} \,, \tag{3.8b}$$

$$\boldsymbol{q}^{E} = \boldsymbol{q}^{n} + \Delta t \sum_{j=1}^{i-1} a_{ij} S(\boldsymbol{Q}^{(j)}) \,. \tag{3.8c}$$

Then the following linear system is solved (similar to Eq. (3.5)):

$$\boldsymbol{q}_{tt} = \widehat{\boldsymbol{q}} + \Delta t \, \tilde{a}_{ii} \delta L \left(\boldsymbol{q}_{tt} \right) \,. \tag{3.8d}$$

The stage value is obtained from Eq. (3.8a):

$$\boldsymbol{Q}^{(i)} = \boldsymbol{q}_{tt} - \sum_{j=1}^{i-1} \frac{\tilde{a}_{ij} - a_{ij}}{\tilde{a}_{ii}} \boldsymbol{Q}^{(j)} \,. \tag{3.8e}$$

In the case of explicit stages ($\tilde{a}_{ii} = 0$), $\boldsymbol{Q}^{(i)}$ from Eq. (3.7a) is obtained by

$$\boldsymbol{Q}^{(i)} = \boldsymbol{q}^{n} + \Delta t \sum_{j=1}^{i-1} a_{ij} S(\boldsymbol{Q}^{(j)}) + \Delta t \delta L \sum_{j=1}^{i-1} (\tilde{a}_{ij} - a_{ij}) \boldsymbol{Q}^{(j)} \,. \tag{3.9}$$

The solution at the next step is obtained from Eq. (3.7b).

In this study we develop a new second-order ARK method and also consider the ARK schemes of orders 3 and 4 developed by Kennedy and Carpenter [21]. All ARK schemes are singly diagonal, first-stage explicit (*i.e.*, $\tilde{a}_{ii} = \tilde{a}_{jj}$, $2 \leq i, j \leq s$). Having the same $\tilde{a}$ on the tableau diagonal benefits the linear solves with direct methods because the factorization of $(I - \Delta t \, \tilde{a}_{ii} L)$ in Eq. (3.8d) needs to be computed only once. They also have L-stable implicit parts and second stage-order that limits the order reduction when applied to stiff problems.

We now introduce the (new) second-order ARK scheme. L-stable second-order ARK methods and second-stage order (i.e., all internal stage values are second-order approximations of the solution) with minimal cost per step have at least three stages with the first-stage being explicit. By applying the order conditions and stability constraints, we obtain the following ARK Butcher tableaux [3]:

$$\begin{array}{c|ccc} 0 & 0 & & \\ 2 \mp \sqrt{2} & 2 \mp \sqrt{2} & 0 & \\ 1 & 1 - a_{32} & a_{32} & 0 \\ \hline & \pm \frac{1}{2\sqrt{2}} & \pm \frac{1}{2\sqrt{2}} & 1 \mp \frac{1}{\sqrt{2}} \end{array} \qquad \begin{array}{c|ccc} 0 & 0 & & \\ 2 \mp \sqrt{2} & 1 \mp \frac{1}{\sqrt{2}} & 1 \mp \frac{1}{\sqrt{2}} & \\ 1 & \pm \frac{1}{2\sqrt{2}} & \pm \frac{1}{2\sqrt{2}} & 1 \mp \frac{1}{\sqrt{2}} \\ \hline & \pm \frac{1}{2\sqrt{2}} & \pm \frac{1}{2\sqrt{2}} & 1 \mp \frac{1}{\sqrt{2}} \end{array} \,. \tag{3.10}$$

The family of schemes defined by Eq. (3.10) results in two choices for the implicit part driven by the diagonal element $(1 \mp \frac{1}{\sqrt{2}})$. We choose $1 - \frac{1}{\sqrt{2}}$ because this implies that all function evaluations are taken inside the time-step interval, i.e., at the specific times $\{t_n, t_n + (2 - \sqrt{2})\Delta t, t_n + \Delta t\}$. Choosing $1 - \frac{1}{\sqrt{2}}$ for the implicit diagonal component also implies that the top plus/minus signs are used throughout the tableaux; note that this results in positive coefficients only. The only free parameter

necessary for the conservation of energy. Nonetheless, our numerical results show that the energy loss is relatively small for long time-integrations.

that remains is a_{32}, which can be adjusted to provide particular stability and accuracy properties. In our experiments we choose $a_{32} = \frac{1}{6}\left(3+2\sqrt{2}\right)$, which confers a relatively large stability region along the imaginary axis as well as eliminates the explicit second order error while minimizes some third-order error components. We denote this scheme by ARK2 and note that the implicit part is the same as the method found by Butcher and Chen [4]. To complete the formulation of ARK2, we give the b vectors for a first-order embedded method as $\widehat{b} = \left[(4-\sqrt{2})/8, (4-\sqrt{2})/8, 1/(2\sqrt{2})\right]^{\mathcal{T}}$ and a second-order, dense output formula

$$\mathbb{B}^* = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 1-\sqrt{2} \\ -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}^{\mathcal{T}},$$

which can be used for stable second-order interpolation within one time step by

$$\boldsymbol{q}^*(t_n + \vartheta\Delta t) := \boldsymbol{q}_n + \Delta t \sum_{i=1}^{3} b_i^*(\vartheta)\left(f(\boldsymbol{Q}^{(i)}) + g(\boldsymbol{Q}^{(i)})\right),$$

where $\vartheta \in [0,1]$, $b_i^*(\vartheta) = \sum_{j=1}^{2} \mathbb{B}_{ij}^* \vartheta^j$ is a vector of computed weights for a given "target" time, and $\boldsymbol{q}^*(t_n + \vartheta\Delta t) - \boldsymbol{q}(t_n + \vartheta\Delta t) = \mathcal{O}(\Delta t^3)$.

High-order ARK methods are difficult to construct, and for this study we consider schemes available from the literature. Methods of orders three (four stages), four (six stages), and five (eight stages) have been developed in [21]. They are all explicit first-stage, singly diagonal, second-stage order, L-stable methods. In our experiments we use the third- and fourth-order methods, which we denote by ARK3 and ARK4.

3.3. Boundary Conditions. In this paper, we only consider no-flux (i.e., reflecting) boundary conditions; however, we include both no-flux and non-reflecting boundary conditions in order to show how to include both types of boundary conditions within the IMEX formulation. For the no-flux boundary conditions, we apply the condition $\bar{\boldsymbol{n}}_\Gamma \cdot \boldsymbol{u} = 0$, where $\bar{\boldsymbol{n}}_\Gamma$ is the outward pointing unit normal vector of the boundary Γ. Since $\boldsymbol{u}$ and $\bar{\boldsymbol{n}}_\Gamma$ both live in R^3, we can define an augmented normal vector

$$\bar{\boldsymbol{n}}_\Gamma = \left(0, \bar{\boldsymbol{n}}_\Gamma^{\mathcal{T}}, 0\right)^{\mathcal{T}} \in R^5$$

that then allows us to satisfy no-flux boundary conditions as follows: $\bar{\boldsymbol{n}}_\Gamma \cdot \boldsymbol{q} = 0$. We will use $\bar{\boldsymbol{n}}_\Gamma$ as either a vector in R^3 or R^5, but this should be self-evident by virtue of the dimensions of the vector we operate on with $\bar{\boldsymbol{n}}_\Gamma$. For explicit time-integration methods, one can apply all boundary conditions in an *a posteriori* fashion, but this is not correct for an implicit method; for such methods, all boundary conditions need to be applied differently.

For implicit (i.e, the implicit part of IMEX) time-integrators, we apply the boundary conditions through Lagrange multipliers as follows:

$$\frac{\partial \boldsymbol{q}}{\partial t} = S(\boldsymbol{q}) + \tau_{nf}\bar{\boldsymbol{n}}_\Gamma + \tau_{nr}(\boldsymbol{q} - \boldsymbol{q}_b) \tag{3.11}$$

where τ_{nf} and τ_{nr} are the Lagrange multipliers for the *no-flux* and *non-reflecting* boundary conditions, respectively, and $\boldsymbol{q}_b$ is the free-stream (boundary) values of the state variable $\boldsymbol{q}$.

The simplest way of imposing non-reflecting boundary conditions (NRBC) is through the use of sponge layers. However, these are not by any means the best way of imposing such NRBCs but is used extensively in many operational weather models (for an example of proper NRBCs, see, e.g., [6, 24, 23]). To impose a sponge layer boundary condition, one can write the semi-discrete (in time) equations as follows

$$\boldsymbol{q}_{tt} = \alpha\left(\widehat{\boldsymbol{q}} + \lambda L(\boldsymbol{q}_{tt})\right) + \beta\widehat{\boldsymbol{q}}_b$$

where α and β are Newtonian relaxation coefficients that drive the solution towards the boundary reference value such that $\alpha \to 1$, $\beta \to 0$ in the interior and $\alpha \to 0$, $\beta \to 1$ as the non-reflecting boundaries are approached; this boundary condition is applied to the entire solution vector $\boldsymbol{q}$.

To impose no-flux boundaries, one need only apply a constraint on the velocity field $\boldsymbol{u}$. In this case, we rewrite the momentum equations as

$$\boldsymbol{u}_{tt} = \alpha\left(\widehat{\boldsymbol{u}} + \lambda L(\boldsymbol{q}_{tt})\right) + \beta \boldsymbol{u}_b + \tau_{nf}\bar{\boldsymbol{n}}_\Gamma.$$

Taking the scalar product of this equation with $\bar{\boldsymbol{n}}_\Gamma$ and rearranging results in the following equivalent system

$$\boldsymbol{u}_{tt} = \boldsymbol{\mathcal{P}}\left[\alpha\left(\widehat{\boldsymbol{u}} + \lambda L(\boldsymbol{q}_{tt})\right) + \beta \boldsymbol{u}_b\right]$$

where

$$\boldsymbol{\mathcal{P}} = \begin{cases} \boldsymbol{I} - \bar{\boldsymbol{n}}_\Gamma \bar{\boldsymbol{n}}_\Gamma^{\mathcal{T}} & \text{in} \quad \Gamma \\ \boldsymbol{I} & \text{in} \quad \Omega - \Gamma \end{cases} \tag{3.12}$$

is the orthogonal projector that imposes the no-flux boundary condition, where $\boldsymbol{I}$ denotes the rank-3 identity matrix.

3.4. Stabilization. It is well understood that continuous Galerkin (CG) methods require stabilization for classes of differential operators where advection plays a significant role (e.g., see [25]). This is especially true when inexact integration is used in the inner products of the spatial discretization method since the numerical representation will not preserve the skew-symmetry of the continuous differential operator. For this reason, CG methods are used with either filters or artificial viscosity. In this paper, we add a minimal amount of artificial viscosity through a Laplacian operator applied to the momentum and temperature equations as such $\mu\boldsymbol{\nabla}^2 q$ where $\mu = 0.1$ m^2/sec for all simulations. Furthermore, the artificial viscosity is applied only to the explicit part of the IMEX time-integrators. In Appendix B, we discuss the issues which arise with using *a posteriori* filters, as is often done in high-order finite element methods.

3.5. IMEX in All Directions. In this section, we describe the application of the IMEX method where the implicit linear operator is defined in all three spatial dimensions.

3.5.1. No Schur Form. The linear operator for the IMEX method applied to all three spatial dimensions is

$$L(\boldsymbol{q}) = -\begin{pmatrix} \boldsymbol{u}\cdot\boldsymbol{\nabla}\rho_0 + \rho_0\boldsymbol{\nabla}\cdot\boldsymbol{u} \\ \frac{1}{\rho_0}\boldsymbol{\nabla}P' + g\frac{\rho'}{\rho_0}\bar{\boldsymbol{r}} \\ \boldsymbol{u}\cdot\boldsymbol{\nabla}\theta_0 \end{pmatrix}, \tag{3.13}$$

with the (linearized) pressure defined as

$$P' = \frac{\gamma P_0}{\rho_0}\rho' + \frac{\gamma P_0}{\theta_0}\theta'. \tag{3.14}$$

Applying the IMEX method yields

$$\rho_{tt} = (\alpha\widehat{\rho} + \beta\widehat{\rho}_b) - \alpha\lambda\left(\boldsymbol{u}_{tt}\cdot\boldsymbol{\nabla}\rho_0 + \rho_0\boldsymbol{\nabla}\cdot\boldsymbol{u}_{tt}\right) \tag{3.15a}$$

$$\boldsymbol{u}_{tt} = (\alpha\widehat{\boldsymbol{u}} + \beta\widehat{\boldsymbol{u}}_b) - \frac{\alpha\lambda}{\rho_0}\left(\boldsymbol{\nabla}P_{tt} + g\rho_{tt}\bar{\boldsymbol{r}}\right) \tag{3.15b}$$

$$\theta_{tt} = \left(\alpha\widehat{\theta} + \beta\widehat{\theta_b}\right) - \alpha\lambda\left(\boldsymbol{u}_{tt}\cdot\boldsymbol{\nabla}\theta_0\right) \tag{3.15c}$$

$$P_{tt} = G_0\rho_{tt} + H_0\theta_{tt}, \tag{3.15d}$$

where

$$G_0 = \frac{\gamma P_0}{\rho_0}, \quad H_0 = \frac{\gamma P_0}{\theta_0}. \tag{3.16}$$

The system represented by Eqs. (3.15a)-(3.15d) is the *No Schur* IMEX form.

3.5.2. Schur Form. Substituting Eq. (3.15c) into Eq. (3.15d) yields

$$\rho_{tt} = \frac{1}{G_0}\left\{P_{tt} - H_0\left[\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right) - \alpha\lambda\left(\boldsymbol{u}_{tt}\cdot\boldsymbol{\nabla}\theta_0\right)\right]\right\}. \tag{3.17}$$

We can now substitute Eq. (3.17) into Eq. (3.15b) in order to express the momentum as a function of pressure only. Upon applying this substitution, we get

$$\boldsymbol{u}_{tt} = \boldsymbol{\mathcal{P}}_C\left[(\alpha\widehat{\boldsymbol{u}} + \beta\widehat{\boldsymbol{u}}_b) + \alpha\lambda\frac{gH_0}{\rho_0 G_0}\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right)\bar{\boldsymbol{r}} - \frac{\alpha\lambda}{\rho_0}\left(\boldsymbol{\nabla}P_{tt} + \frac{g}{G_0}P_{tt}\bar{\boldsymbol{r}}\right)\right] \tag{3.18}$$

where no-flux boundary conditions are enforced through the application of the orthogonal projector $\boldsymbol{\mathcal{P}}$ given in Eq. (3.12), $\boldsymbol{\mathcal{P}}_C = \boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{C}}$, $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{A}}^{-1}$, where the matrix $\boldsymbol{\mathcal{A}}$ is obtained by isolating the momentum equation in terms of its variables and is defined as

$$\boldsymbol{\mathcal{A}} = \mathbf{I} + c\bar{\boldsymbol{r}}\left(\nabla\theta_0\right)^{\mathcal{T}},$$

where $\bar{\boldsymbol{r}} = (r_x, r_y, r_z)^{\mathcal{T}}$ and

$$c = (\alpha\lambda)^2\frac{g}{\theta_0}. \tag{3.19}$$

Substituting Eqs. (3.15a) and (3.15c) into Eq. (3.15d) yields

$$P_{tt} = G_0\left(\alpha\widehat{\rho} + \beta\widehat{\rho}_b\right) + H_0\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right) - \alpha\lambda\boldsymbol{F}_0\cdot\boldsymbol{u}_{tt} - \alpha\lambda\rho_0 G_0\boldsymbol{\nabla}\cdot\boldsymbol{u}_{tt}, \tag{3.20}$$

where $\boldsymbol{F}_0 = G_0\boldsymbol{\nabla}\rho_0 + H_0\boldsymbol{\nabla}\theta_0$. The last step is to substitute Eq. (3.18) into Eq. (3.20), which yields the *Schur* form

$$\begin{aligned}
&P_{tt} - (\alpha\lambda)^2\boldsymbol{F}_0\cdot\left[\boldsymbol{\mathcal{P}}_C\left(\frac{1}{\rho_0}\boldsymbol{\nabla}P_{tt} + \frac{g}{\rho_0 G_0}P_{tt}\bar{\boldsymbol{r}}\right)\right] \\
&\quad - (\alpha\lambda)^2 G_0\rho_0\boldsymbol{\nabla}\cdot\left[\boldsymbol{\mathcal{P}}_C\left(\frac{1}{\rho_0}\boldsymbol{\nabla}P_{tt} + \frac{g}{\rho_0 G_0}P_{tt}\bar{\boldsymbol{r}}\right)\right] \\
&\quad = G_0\left(\alpha\widehat{\rho} + \beta\widehat{\rho}_b\right) + H_0\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right) \\
&\quad - \alpha\lambda\boldsymbol{F}_0\cdot\left[\boldsymbol{\mathcal{P}}_C\left((\alpha\widehat{\boldsymbol{u}} + \beta\widehat{\boldsymbol{u}}_b) + \alpha\lambda\frac{gH_0}{\rho_0 G_0}\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right)\bar{\boldsymbol{r}}\right)\right] \\
&\quad - \alpha\lambda G_0\rho_0\boldsymbol{\nabla}\cdot\left[\boldsymbol{\mathcal{P}}_C\left((\alpha\widehat{\boldsymbol{u}} + \beta\widehat{\boldsymbol{u}}_b) + \alpha\lambda\frac{gH_0}{\rho_0 G_0}\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right)\bar{\boldsymbol{r}}\right)\right].
\end{aligned} \tag{3.21}$$

3.5.3. Schur Form in Cloud-Resolving/Mesoscale Mode. For cloud-resolving/mesoscale mode (i.e., flow in a box) the following simplifications occur: $\bar{\boldsymbol{r}} = \bar{\boldsymbol{k}}$ and $\boldsymbol{q}_0 = \boldsymbol{q}_0(z)$. These two changes vastly simplify the Schur form. For example, the matrix $\boldsymbol{\mathcal{A}}$ becomes diagonal and is defined as $diag(\boldsymbol{\mathcal{A}}) = \left(1, 1, 1 + c\frac{d\theta_0}{dz}\right)$ and $\boldsymbol{\mathcal{C}}$ becomes $diag(\boldsymbol{\mathcal{C}}) = \left(1, 1, 1/\left(1 + c\frac{d\theta_0}{dz}\right)\right)$, which is the three-dimensional generalization of the two-dimensional matrix $\boldsymbol{\mathcal{C}}$ given in [14]. Equation (3.21) simplifies to

$$\begin{aligned}
&P_{tt} - (\alpha\lambda)^2 F_0\bar{\boldsymbol{k}}\cdot\left[\boldsymbol{\mathcal{P}}_C\left(\frac{1}{\rho_0}\boldsymbol{\nabla}P_{tt} + \frac{g}{\rho_0 G_0}P_{tt}\bar{\boldsymbol{k}}\right)\right] \\
&\quad - (\alpha\lambda)^2 G_0\rho_0\boldsymbol{\nabla}\cdot\left[\boldsymbol{\mathcal{P}}_C\left(\frac{1}{\rho_0}\boldsymbol{\nabla}P_{tt} + \frac{g}{\rho_0 G_0}P_{tt}\bar{\boldsymbol{k}}\right)\right] \\
&\quad = G_0\left(\alpha\widehat{\rho} + \beta\widehat{\rho}_b\right) + H_0\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right) \\
&\quad - \alpha\lambda F_0\bar{\boldsymbol{k}}\cdot\left[\boldsymbol{\mathcal{P}}_C\left((\alpha\widehat{\boldsymbol{u}} + \beta\widehat{\boldsymbol{u}}_b) + \alpha\lambda\frac{gH_0}{\rho_0 G_0}\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right)\bar{\boldsymbol{k}}\right)\right] \\
&\quad - \alpha\lambda G_0\rho_0\boldsymbol{\nabla}\cdot\left[\boldsymbol{\mathcal{P}}_C\left((\alpha\widehat{\boldsymbol{u}} + \beta\widehat{\boldsymbol{u}}_b) + \alpha\lambda\frac{gH_0}{\rho_0 G_0}\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right)\bar{\boldsymbol{k}}\right)\right]
\end{aligned} \tag{3.22}$$

where $F_0 = G_0\frac{d\rho_0}{dz} + H_0\frac{d\theta_0}{dz}$.

3.6. IMEX in One Direction. The IMEX method defined in all spatial dimensions as described in Sec. 3.5 is general and applicable to many problems in atmospheric modeling. However, that formulation requires the solution of a single, large, sparse global matrix that represents the underlying 3D problem and can be costly even with the use of the most sophisticated iterative solvers and preconditioners. For problems where the domain has different scales in the vertical and horizontal direction it may be advantageous to employ an IMEX method in the vertical dimension only. This is the case in global atmospheric modeling where the vertical direction is less than 40 km while the horizontal direction is a thousand times larger. In such a case, the time-step restriction will be solely dominated by the vertical direction, and so it is prudent to develop an IMEX approach whereby the horizontal direction is solved fully explicitly but the vertical direction is solved using IMEX methods; this strategy then results in the solution of a collection of small banded (one-dimensional) matrices that are stored on-processor and are decoupled from each other. Besides being much faster to solve, this approach has the added advantage that the method will scale exactly as the underlying explicit method because no MPI communications are required to solve the implicit problem precisely because each column of data is completely independent from all other columns. This solution strategy requires using a 2D domain decomposition whereby the vertical direction is entirely on-processor, resulting in an embarrassingly parallel solution strategy. Furthermore, additional concurrency may be extracted from the solution of these independent columns through fine-grained parallelism (e.g., through either multi-threading using OpenMP or CUDA/OpenCL within GPUs).

To construct the IMEX method in the vertical (in cloud-resolving/mesoscale mode) or radial (in global) direction requires first mapping the Cartesian coordinates to a local radial-tangent space. We refer to this mapping as follows. Let $\boldsymbol{\mathcal{R}} : \mathcal{C} \rightarrow R$ where $\boldsymbol{\mathcal{R}}$ is the map that takes the standard Cartesian space (i.e., R^3) to the rotated space R defined by the vectors $(\bar{\boldsymbol{s}}, \bar{\boldsymbol{t}}, \bar{\boldsymbol{r}})^{\mathcal{T}}$, which we define below. The first step is to map the velocity field $\boldsymbol{u} = (u, v, w)^{\mathcal{T}}$ as follows:

$$\boldsymbol{u}^R = \boldsymbol{\mathcal{R}}\boldsymbol{u} \tag{3.23}$$

where $\boldsymbol{u}^R = \left(u^{(s)}, u^{(t)}, u^{(r)}\right)^{\mathcal{T}}$ is the rotated velocity field,

$$\boldsymbol{\mathcal{R}} = (\bar{\boldsymbol{s}} \qquad \bar{\boldsymbol{t}} \qquad \bar{\boldsymbol{r}})^{\mathcal{T}} \tag{3.24}$$

is the map, and $\bar{\boldsymbol{r}} = \frac{\boldsymbol{x}}{||\boldsymbol{x}||} = (r_x, r_y, r_z)^{\mathcal{T}}$, $\bar{\boldsymbol{s}} = \boldsymbol{\mathcal{Q}}_v \bar{\boldsymbol{r}} \times \bar{\boldsymbol{v}}$, and $\bar{\boldsymbol{t}} = \bar{\boldsymbol{r}} \times \bar{\boldsymbol{s}}$ are normalized vectors. The vector $\bar{\boldsymbol{s}}$ is guaranteed to be orthogonal to $\bar{\boldsymbol{r}}$ by virtue of the projection $\boldsymbol{\mathcal{Q}}_v \in R^{3\times 3}$ and then taking the vector product with $\bar{\boldsymbol{v}}$. The vector $\bar{\boldsymbol{v}} \in R^3$ is chosen to be along the $\bar{\boldsymbol{i}}$, $\bar{\boldsymbol{j}}$, or $\bar{\boldsymbol{k}}$ directions depending on which component of $\bar{\boldsymbol{r}}$ is a minimum; that is, $\bar{\boldsymbol{v}} = \bar{\boldsymbol{i}}$ if $|r_x| = min(|r_x|, |r_y|, |r_z|)$, and so on. This is done to avoid aligning the vector $\bar{\boldsymbol{v}}$ with the null space of $\bar{\boldsymbol{r}}$. The matrix is defined as $\boldsymbol{\mathcal{Q}}_v = \delta_{ij}(1 - \delta_{ijk})$, where δ_{ij} and δ_{ijk} are the Kronecker delta functions and $i, j, k = 1, ..., 3$ are the indices of $\boldsymbol{\mathcal{Q}}_v$ and $k = 1, 2, 3$ for $\bar{\boldsymbol{v}} = \bar{\boldsymbol{i}}$, $\bar{\boldsymbol{j}}$, $\bar{\boldsymbol{k}}$, respectively. The matrix $\boldsymbol{\mathcal{Q}}_v$ is constructed in order to project $\bar{\boldsymbol{r}}$ along a subspace of R^3 in a direction orthogonal to $\bar{\boldsymbol{v}}$. This approach guarantees that $\bar{\boldsymbol{s}}$ and $\bar{\boldsymbol{t}}$ form a tangent plane passing through the radial vector $\bar{\boldsymbol{r}}$; note that they form an orthogonal (local) coordinate system that is independent of the geometry of the problem. This is critical because it means that this approach is applicable to not just a box (i.e., cloud-resolving/mesoscale flow) or a sphere (i.e., global flow) but also to any other geometry including oblate spheroids (for use in more realistic geometric representations of the Earth because no specific geometry is assumed). The mapping described in essence is similar to a modified Gram-Schmidt orthogonalization; the key difference is that this orthogonal mapping also works naturally even when one of the new vectors is aligned with the original Cartesian directions.

3.6.1. No Schur Form. Upon applying the rotation transformation given in Eq. (3.24), we obtain the rotated variables

$$\boldsymbol{q}^R = \begin{pmatrix} \rho' \\ u^{(s)} \\ u^{(t)} \\ u^{(r)} \\ \theta' \end{pmatrix}. \tag{3.25}$$

The linear operator for the IMEX method applied along this rotated system for either the vertical (in cloud-resolving/mesoscale mode) or radial (in global mode) is

$$L(\boldsymbol{q}) = -\begin{pmatrix} u^{(r)}\frac{d\rho_0}{dr} + \rho_0\frac{\partial u^{(r)}}{\partial r} \\ 0 \\ 0 \\ \frac{1}{\rho_0}\frac{\partial P'}{\partial r} + g\frac{\rho'}{\rho_0} \\ u^{(r)}\frac{d\theta_0}{dr} \end{pmatrix} \tag{3.26}$$

with the pressure defined as in Eq. (3.14). Applying the IMEX method yields

$$\rho_{tt} = (\alpha\widehat{\rho} + \beta\widehat{\rho}_b) - \alpha\lambda\left(u_{tt}^{(r)}\frac{d\rho_0}{dr} + \rho_0\frac{\partial u_{tt}^{(r)}}{\partial r}\right) \tag{3.27a}$$

$$u_{tt}^{(s)} = \left(\alpha\widehat{u}^{(s)} + \beta\widehat{u}_b^{(s)}\right) \tag{3.27b}$$

$$u_{tt}^{(t)} = \left(\alpha\widehat{u}^{(t)} + \beta\widehat{u}_b^{(t)}\right) \tag{3.27c}$$

$$u_{tt}^{(r)} = \left(\alpha\widehat{u}^{(r)} + \beta\widehat{u}_b^{(r)}\right) - \frac{\alpha\lambda}{\rho_0}\left(\frac{\partial P_{tt}}{\partial r} + g\rho_{tt}\right) \tag{3.27d}$$

$$\theta_{tt} = \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right) - \alpha\lambda\left(u_{tt}^{(r)}\frac{d\theta_0}{dr}\right) \tag{3.27e}$$

$$P_{tt} = G_0\rho_{tt} + H_0\theta_{tt}, \tag{3.27f}$$

where G_0 and H_0 are defined in Eq. (3.16); the system represented by Eqs. (3.27a)-(3.27f) is the *No Schur* IMEX form.

3.6.2. Schur Form. Substituting Eq. (3.27e) into Eq. (3.27f) yields

$$\rho_{tt} = \frac{1}{G_0}\left\{P_{tt} - H_0\left[\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right) - \alpha\lambda\left(u_{tt}^{(r)}\frac{d\theta_0}{dr}\right)\right]\right\}. \tag{3.28}$$

We can now substitute Eq. (3.28) into Eq. (3.27d) in order to express the momentum as a function of pressure only. Upon applying this substitution, we get

$$\boldsymbol{u}_{tt}^R = \boldsymbol{\mathcal{P}}_C^R\left[\left(\alpha\widehat{\boldsymbol{u}}^R + \beta\widehat{\boldsymbol{u}}_b^R\right) + \alpha\lambda\frac{gH_0}{\rho_0 G_0}\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right)\bar{\boldsymbol{r}}_R - \frac{\alpha\lambda}{\rho_0}\left(\frac{\partial P_{tt}}{\partial r} + \frac{g}{G_0}P_{tt}\right)\bar{\boldsymbol{r}}_R\right] \tag{3.29}$$

where $\boldsymbol{u}_{tt}^R = \left(u_{tt}^{(r)}, u_{tt}^{(s)}, u_{tt}^{(t)}\right)^{\mathcal{T}}$ and similarly for $\widehat{\boldsymbol{u}}^R$ and $\widehat{\boldsymbol{u}}_b^R$, and $\bar{\boldsymbol{r}}_R = \bar{\boldsymbol{r}}$ because the implicit correction should only act along the direction $\bar{\boldsymbol{r}}$.

The no-flux boundary conditions are enforced through the application of the orthogonal projector $\boldsymbol{\mathcal{P}}_C^R = \frac{1}{c}\boldsymbol{\mathcal{P}}^R$ with

$$c = 1 + (\alpha\lambda)^2\frac{g}{\theta_0}\frac{d\theta_0}{dr} \tag{3.30}$$

and

$$\boldsymbol{\mathcal{P}}^R = \boldsymbol{I} - \bar{\boldsymbol{n}}_R\bar{\boldsymbol{n}}_R^{\mathcal{T}}, \tag{3.31}$$

where the vector $\bar{\boldsymbol{n}}_R = n_s\bar{\boldsymbol{s}} + n_t\bar{\boldsymbol{t}} + n_r\bar{\boldsymbol{r}}$ is the projection of $\bar{\boldsymbol{n}}_\Gamma \in R^3$ (the unit normal outward pointing vector to the domain boundary Γ) in the direction of the new rotated coordinate system with components defined as $n_s = \bar{\boldsymbol{n}}_\Gamma\cdot\bar{\boldsymbol{s}}$, $n_t = \bar{\boldsymbol{n}}_\Gamma\cdot\bar{\boldsymbol{t}}$, and $n_r = \bar{\boldsymbol{n}}_\Gamma\cdot\bar{\boldsymbol{r}}$.

Substituting Eqs. (3.27a) and (3.27e) into Eq. (3.27f) yields

$$P_{tt} = G_0\left(\alpha\widehat{\rho} + \beta\widehat{\rho}_b\right) + H_0\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right) - \alpha\lambda F_0 u_{tt}^{(r)} - \alpha\lambda\rho_0 G_0\frac{\partial u_{tt}^{(r)}}{\partial r}, \tag{3.32}$$

where $F_0 = G_0 \frac{d\rho_0}{dr} + H_0 \frac{d\theta_0}{dr}$. The last step is to substitute Eq. (3.29) into Eq. (3.32), which yields the *Schur* form

$$\begin{aligned}
P_{tt} &- (\alpha\lambda)^2 F_0 \bar{\boldsymbol{r}}_R \cdot \left[\boldsymbol{\mathcal{P}}_C^R \left(\frac{1}{\rho_0} \frac{\partial P_{tt}}{\partial r} + \frac{g}{\rho_0 G_0} P_{tt} \right) \bar{\boldsymbol{r}}_R \right] \\
&- (\alpha\lambda)^2 G_0 \rho_0 \frac{\partial}{\partial r} \left[\bar{\boldsymbol{r}}_R \cdot \boldsymbol{\mathcal{P}}_C \left(\frac{1}{\rho_0} \frac{\partial P_{tt}}{\partial r} + \frac{g}{\rho_0 G_0} P_{tt} \right) \bar{\boldsymbol{r}}_R \right] \\
&= G_0 \left(\alpha\widehat{\rho} + \beta\widehat{\rho}_b \right) + H_0 \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b \right) \\
&- \alpha\lambda F_0 \bar{\boldsymbol{r}}_R \cdot \left[\boldsymbol{\mathcal{P}}_C \left\{ \left(\alpha\widehat{\boldsymbol{u}}^R + \beta\widehat{\boldsymbol{u}}_b^R \right) + \alpha\lambda \frac{g H_0}{\rho_0 G_0} \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b \right) \bar{\boldsymbol{r}}_R \right\} \right] \\
&- \alpha\lambda G_0 \rho_0 \frac{\partial}{\partial r} \left[\bar{\boldsymbol{r}}_R \cdot \boldsymbol{\mathcal{P}}_C \left\{ \left(\alpha\widehat{\boldsymbol{u}}^R + \beta\widehat{\boldsymbol{u}}_b^R \right) + \alpha\lambda \frac{g H_0}{\rho_0 G_0} \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b \right) \bar{\boldsymbol{r}}_R \right\} \right]. \qquad (3.33)
\end{aligned}$$

3.6.3. Schur Form in Cloud-Resolving/Mesoscale Mode. For the case of cloud-resolving or mesoscale mode (i.e., flow in a box) the simplifications $\bar{\boldsymbol{r}} = \bar{\boldsymbol{k}}$ and $\boldsymbol{q}_0 = \boldsymbol{q}_0(z)$ affect the Schur form as follows. First we note that the rotation matrix becomes the identity matrix $\boldsymbol{\mathcal{R}} = \boldsymbol{I}$. This mapping says that $u^{(r)} = w$, as it should. Equations (3.27a)-(3.27f) simplify to

$$\rho_{tt} = (\alpha\widehat{\rho} + \beta\widehat{\rho}_b) - \alpha\lambda \left(w \frac{d\rho_0}{dz} + \rho_0 \frac{\partial w}{\partial z} \right) \qquad (3.34a)$$

$$u_{tt} = (\alpha\widehat{u} + \beta\widehat{u}_b) \qquad (3.34b)$$

$$v_{tt} = (\alpha\widehat{v} + \beta\widehat{v}_b) \qquad (3.34c)$$

$$w_{tt} = (\alpha\widehat{w} + \beta\widehat{w}_b) - \frac{\alpha\lambda}{\rho_0} \left(\frac{\partial P_{tt}}{\partial z} + g\rho_{tt} \right) \qquad (3.34d)$$

$$\theta_{tt} = \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b \right) - \alpha\lambda \left(w_{tt} \frac{d\theta_0}{dz} \right), \qquad (3.34e)$$

and Eq. (3.30) simplifies to

$$c = 1 + (\alpha\lambda)^2 \frac{g}{\theta_0} \frac{d\theta_0}{dz}$$

with $\boldsymbol{\mathcal{P}}_C^R = \boldsymbol{\mathcal{P}}$ and $\bar{\boldsymbol{r}}_R = \bar{\boldsymbol{k}}$, which defines a classical IMEX formulation for a mesoscale model and is the three-dimensional version of the IMEX (i.e., semi-implicit) formulation described in [14]. All these simplifications result in the new form of Eq. (3.33):

$$\begin{aligned}
P_{tt} &- (\alpha\lambda)^2 F_0 \bar{\boldsymbol{k}} \cdot \left[\boldsymbol{\mathcal{P}}_C \left(\frac{1}{\rho_0} \frac{\partial P_{tt}}{\partial z} + \frac{g}{\rho_0 G_0} P_{tt} \right) \bar{\boldsymbol{k}} \right] \\
&- (\alpha\lambda)^2 G_0 \rho_0 \frac{\partial}{\partial z} \left[\bar{\boldsymbol{k}} \cdot \boldsymbol{\mathcal{P}}_C \left(\frac{1}{\rho_0} \frac{\partial P_{tt}}{\partial z} + \frac{g}{\rho_0 G_0} P_{tt} \right) \bar{\boldsymbol{k}} \right] \\
&= G_0 \left(\alpha\widehat{\rho} + \beta\widehat{\rho}_b \right) + H_0 \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b \right) \\
&- \alpha\lambda F_0 \bar{\boldsymbol{k}} \cdot \left[\boldsymbol{\mathcal{P}}_C \left\{ (\alpha\widehat{\boldsymbol{u}} + \beta\widehat{\boldsymbol{u}}_b) + \alpha\lambda \frac{g H_0}{\rho_0 G_0} \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b \right) \bar{\boldsymbol{k}} \right\} \right] \\
&- \alpha\lambda G_0 \rho_0 \frac{\partial}{\partial z} \left[\bar{\boldsymbol{k}} \cdot \boldsymbol{\mathcal{P}}_C \left\{ (\alpha\widehat{\boldsymbol{u}} + \beta\widehat{\boldsymbol{u}}_b) + \alpha\lambda \frac{g H_0}{\rho_0 G_0} \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b \right) \bar{\boldsymbol{k}} \right\} \right]. \qquad (3.35)
\end{aligned}$$

4. Results. In this section, we present three types of results for our unified atmospheric model NUMA using continuous Galerkin methods; the order of the spatial discretization is determined by the polynomial order (plus one) used for constructing the grid. The types of problems considered represent the class of problems we expect to solve with our model, including cloud-resolving simulations in order to understand fine-scale structures such as turbulence; mesoscale problems typical

of regional or limited-area numerical weather prediction problems; and global problems representing the general circulation of atmospheric dynamics typical in either climate simulations or global numerical weather prediction. We note that the goal of using these three types of problems is not to verify, validate, or benchmark NUMA but rather to introduce the possible applications that NUMA can be used for and to quantify which type of IMEX time-integrator (e.g., 1D or 3D decomposition) is more efficient depending on the type of problem being solved (i.e., cloud-resolving, mesoscale, or global). We quantify the accuracy and efficiency of each of the time-integrators in order to understand the order of magnitude of the errors committed by low-order versus high-order time-integrators in atmospheric models.

To compare the various time-integrators, we use the explicit (RK35) time-integrator [29] with a small time-step as the *exact* solution. We then compute the (absolute) L^2 norm:

$$L^2 \text{ error} = \| \left(\boldsymbol{q}^{num} - \boldsymbol{q}^{exact} \right) \|_2$$

for $\boldsymbol{q}^{num},\ \boldsymbol{q}^{exact} \in \mathcal{R}^{N_{dof}}$ where $N_{dof} = 5N_{points}$, with N_{points} being the number of gridpoints in the domain and the scalar 5 the dimension of the solution vector at each gridpoint. In other words, we compute the norm of the solution vector $\boldsymbol{q}$ taking it as a column vector of $\dim \boldsymbol{q} = N_{dof}$, i.e., the Frobenius norm of the matrix $\boldsymbol{q} \in \mathcal{R}^{5 \times N_{points}}$.

The linear system resulting from the 3D IMEX approach is solved using GMRES with an element-based spectrally optimized approximate inverse preconditioner [5]. However, the preconditioners do not have a significant impact on the efficiency study because the results shown below are derived for time-step sizes that are relatively small; that is, GMRES converges to a solution with a relatively small number of iterations (less than 10). For the 1D IMEX approach, the linear system is solved using a direct solver (LU decomposition). While both iterative and direct solvers are included within NUMA, we have chosen to use a direct solver for the 1D IMEX approach because it is a more robust solution strategy since a stopping criterion is not required although this may mean that the direct solver will require more operations than an iterative approach.

The number of gridpoints in each simulation is determined by the number of elements and the polynomial order of the continuous Galerkin method. For instance, for the cloud-resolving and mesoscale simulations the number of gridpoints is defined as $N_{points} = (N_E N + 1)^3$ where N_E and N denotes the number of elements and the polynomial order in each Cartesian direction. For the global simulation since we use a cubed-sphere grid, the number of gridpoints is defined to be $N_{points} = \left(6(N_E N)^2 + 2\right)(N_E N + 1)$ where the first term in parentheses denotes the number of points on a spherical shell (see, e.g., [15, 8, 10] while the second term represents the number of points along a radial component. We note that currently NUMA only admits hexahedral elements.

A brief discussion on the goals of this study and how this translate to the use of IMEX methods for operational models is in order. The goal of IMEX methods is to accelerate the speed of a simulation. In other words, since a larger time-step is used then it is expected that the wallclock time of the simulation will decrease. However, IMEX methods remain stable precisely by damping the waves that are treated implicitly; in most operational models, these wave are the acoustic waves which are deemed unimportant to the rest of the governing dynamics. Therefore, one would like to use as large a time-step as possible in order to extract the maximum level of efficiency. However, in this study we will not use extremely large time-steps for the following reason. Since the simulations studied below do not have analytic solutions, we use an explicit time-integrator with a very small time-step as our *exact* solution. This *exact* solution represents all waves properly including the acoustic waves. However, the IMEX solutions damp the acoustic waves and as we increase the time-steps of these simulations, the acoustic waves will not be represented properly (this is explained in more detail in Fig. 4.3). For this reason we cannot use the explicit solution as the "truth" solution when comparing IMEX methods at very large time-steps. Instead, we shall use small time-step simulations to compute the convergence rates of the various IMEX methods.

4.1. Cloud-Resolving Mode: Rising Thermal Bubble. This test case uses a hydrostatically balanced reference state with a thermally neutral atmosphere; that is, the reference potential temperature is taken to be $\theta_0 = 300$ Kelvin (K); this is the three-dimensional extension of the two-dimensional test case proposed in [12] . The initial conditions are augmented by the following

perturbation

$$\Delta\theta = \begin{cases} 0 & \text{for} \quad R > R_c \\ \theta_c \left[1 + \cos\left(\frac{\pi R}{R_c}\right)\right] & \text{for} \quad R \leq R_c, \end{cases}$$

where R is the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{x}_c$, $\boldsymbol{x}_c = (500, 500, 260)$, $R_c = 250$ meters (m), and $\theta_c = 0.5$ is a constant. The domain for this problem is $(x, y, z) \in [0, 1000]^3$ m. Note that cloud-resolving simulations are usually carried out with grid resolutions less than 1000 m. Since for this test case we use grid resolutions of 10 to 20 m, we refer to it as cloud-resolving. This test case does not have an analytic solution, but the proper behavior of buoyant convection is well understood and can be used to verify the model.

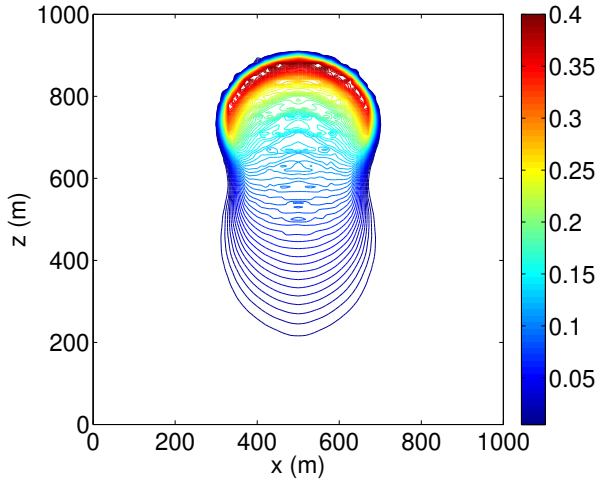


FIGURE 4.1. *Cloud-Resolving Mode: Rising Thermal Bubble. A slice of the potential temperature perturbation (at y=500 m) after 400 seconds (s) for* 24^3 *elements with 4th order polynomials. The contour lines are from 0.005 to 0.5 with an interval of 0.005.*

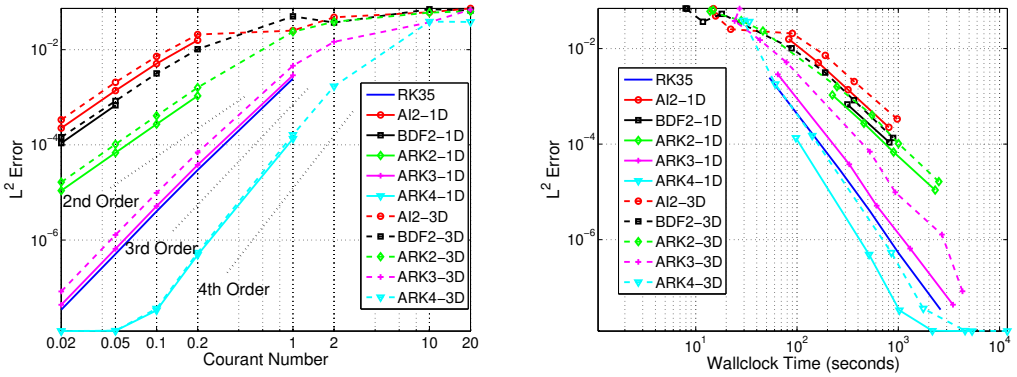
Figure 4.1 shows the potential temperature perturbation after 400 s for a grid resolution of 24^3 elements each with 4th-order polynomials (which yields a grid resolution of 10.3 m and 912673 gridpoints). Note that the initial condition is a cosine bubble (in three-dimensions) that, after 400 s, evolves into a bubble that folds in on itself because of the buoyancy of the hotter fluid positioned in the center of the bubble. This problem is similar to the classical Rayleigh-Taylor instability fluid dynamics problem.

To compare the accuracy and efficiency of the time-integrators, we run this test case using a grid consisting of 10^3 elements each with 4th order polynomials, which yields a resolution of 20 m with 68921 gridpoints; 10 MPI (Message-Passing Interface) processes are used for timing the simulations. In Fig. 4.2 we report the accuracy (panel a) and wallclock time (panel b) as a function of the Courant number (Courant numbers reported are always the maximum associated with the fast waves). The simulations are run for 100 s, where the L^2 norm is computed using the explicit (RK35) solution with a Courant number of 0.002.

Figure 4.2a shows that all the time-integrators yield the theoretically expected convergence rates (this is evident by comparing the results of the various order time-integrators with the theoretical convergence rates for order 2, 3, and 4). Furthermore, we note that all the second-order methods yield the same convergence rates (all the slopes are the same) regardless of whether the method uses a 1D-IMEX or a 3D-IMEX approach. The same is also true for the third- and fourth-order methods. In addition, while all second order methods yield the same order of accuracy, ARK2 is an order of magnitude more accurate than the other two second order methods. Note that constructing a 3D-IMEX method that achieves the theoretical rate of convergence rate is relatively straightforward, but this is not the case for the 1D-IMEX method because its derivation is more involved. Therefore

the results of this figure confirm that the 1D-IMEX methods have been derived correctly since they are behaving as theoretically expected.

Figure 4.2b shows the error versus wallclock time; the results of this figure can be summarized as follows. For accuracy levels between 10^{-1} to 10^{-2}, the 3D IMEX methods dominate; however, below errors of 10^{-3} the 3rd and 4th order methods dominate with ARK4 being the fastest (especially the 1D IMEX method). Focusing on 2nd order methods, we see that ARK2 performs very well. In fact, for accuracy levels below 10^{-3} ARK2 is the most efficient 2nd order method.



a) b)

FIGURE 4.2. *Cloud-Resolving Mode: Rising Thermal Bubble. The a) accuracy and b) efficiency for the explicit (RK35), 1D-IMEX, and 3D-IMEX time-integrators. Results are shown for a final time of 100 s.*

For this discussion and what follows, let us define the grid resolution (GR) ratio between the horizontal and vertical directions as follows

$$R_{GR} = \frac{H_{GR}}{V_{GR}}.$$

For the Cloud-Resolving Mode (CRM) simulations of the thermal bubble, $R_{GR} = 1$ (i.e., the nonhydrostatic regime), which means that the only way to increase the maximum allowable time-step is to use an IMEX method in all three dimensions; in other words, in this regime, the 1D IMEX methods will not offer any advantages over a fully explicit method. In this regime, in terms of pure speed (i.e., the least amount of wallclock time regardless of accuracy) the 1D IMEX methods do not perform as well as the 3D IMEX methods because, at this regime, the 1D IMEX methods are behaving exactly like fully explicit methods (top left corner of Fig. 4.2a). However, wallclock time alone should not be the only measure of the efficiency of a time-integrator because, as we show here, one should also take into the account the quality of the solution.

One further comment on Fig. 4.2a: the accuracy of all the time-integrators begin to converge toward a similar value at very large Courant numbers. The reason is that the small time-step simulation that we are calling the "exact" solution is representing the fast waves (e.g., acoustic waves) accurately while the IMEX simulations are stepping over these stiff components. This may seem to be a problem at first glance; but since we are not interested in the acoustic waves (they are believed to play no role in atmospheric modeling), it does not matter. Below we explain this phenomenon more rigorously.

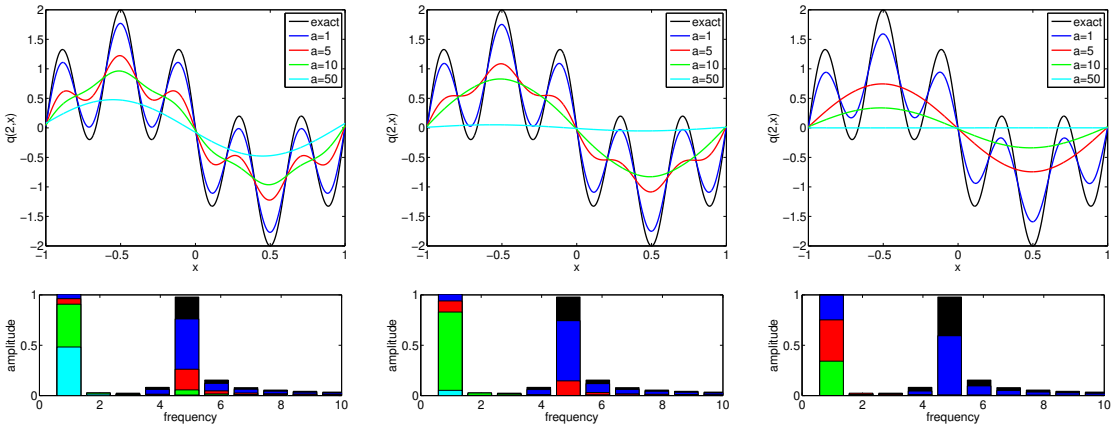
Large Time-Step Behavior. In Fig. 4.2a we observe that at large time steps the accuracy given by different methods is relatively similar. In this regime the methods are still stable, but because of the large time steps, the implicit part of the time-integrator attenuates the high frequency solution components and fast wave speed components. To illustrate this effect, we consider the simple one-dimensional wave equation

$$\frac{\partial q}{\partial t} + a\frac{\partial q}{\partial x} = 0\,, \quad q(0,x) = \sin(2\pi(x+1)) + \sin(10\pi(x+1))\,,\ x \in [-1,1]\,, \tag{4.1}$$

where a the wave speed on a periodic domain. The exact solution is the same as the initial condition with a phase shift, and in particular $q(2aT, x) = q(0, x)$, $T = 1, 2, 3, \ldots$. For illustrative purposes, let us discretize this equation using the unconditionally stable first-order upwind in space and backward Euler in time. In this setting we use only an implicit scheme in order to avoid stability issues and to be in the position to replicate the error behavior observed in Fig. 4.2a. By applying a Fourier analysis (i.e., von Neumann) we obtain the following amplification factor for the Fourier modes $\widehat{q}_{n+1} = r(\xi)\widehat{q}_n$ where

$$|r(\xi)| = (1 + 2a\lambda(1 + a\lambda)(1 - \cos(\xi)))^{-\frac{1}{2}} , \tag{4.2}$$

and $\lambda = \Delta t/\Delta x$ and $\xi = k\Delta x$ with the harmonic wave $k = 2\pi/T$, $k \in [-\pi/\Delta x, \pi\Delta x]$. From this analysis we observe that increasing Δt or a results in general in an increased attenuation of the solution component. After N time steps the attenuation is proportional to $|r(\xi)|^N$. In Fig. 4.3 we illustrate the solution and its spectrum after 2 s (one period for $a = 1$), which we denote in Fig. 4.3 as $q(2, x)$, with different wave speeds and time steps.



a) $\Delta t = \Delta x/100 = 10^{-5}$ b) $\Delta t = \Delta x/10 = 10^{-4}$ c) $\Delta t = \Delta x = 10^{-3}$

FIGURE 4.3. *Exact and numerical solution of the wave equation with different propagation speeds a, and using different time steps along with their corresponding spectra. The final time is the same for all solutions, the difference being that the solution given by setting $a = 1$ travels once across the domain, whereas using $a = 5$ results in five domain traversals by the solution profile. The spectrum indicates how well the 1 Hz and 5 Hz solution components are represented. The color spectrum color scheme is the same as for the solutions (e.g., black is the exact, blue is for $a = 1$).*

The spectrum (in space) associated with the solutions is displayed in the lower panels of Fig. 4.3. The initial and (evolved) exact solution indicate contributions at 1 Hz and 5 Hz. As expected, a quick inspection of Eq. (4.2) reveals that by keeping the time step constant and increasing the wave speed, only the low-frequency components are preserved. We can see this result, for instance, in Fig. 4.3a, where for $a = 1, 5$ some energy in the 5 Hz signal is still present (blue and red bars), but not for $a = 10, 50$ (green and cyan bars). Moreover, the same effect is observed by increasing the time step and keeping the same wave speed. In particular, we see that the components with large wave speed are almost completely damped by changing the time step from $\Delta x/100$ to Δx, whereas the lower wave speeds still retain some energy in the high-frequency domain region. More to the point, we note that component $a = 50$ is completely attenuated at $\Delta t = \Delta x$. This is precisely the effect that we observe in Figure 4.2a, where the time step is increased to a point at which a significant part of the fast dynamics is completely attenuated, resulting in errors that remain relatively constant for all the time-integrators; fortunately these fast dynamics comprise mostly the acoustic waves that we are not so interested in resolving exactly but the point is that our error metrics pick up this difference in the solution between the small time-step "exact" solution and the large time-step IMEX solutions. This means that we cannot state with certainty what is the true error at large time-step for the IMEX methods and for this reason we should not measure error norms beyond the Courant numbers (time-steps) that we report in Fig. 4.2a.

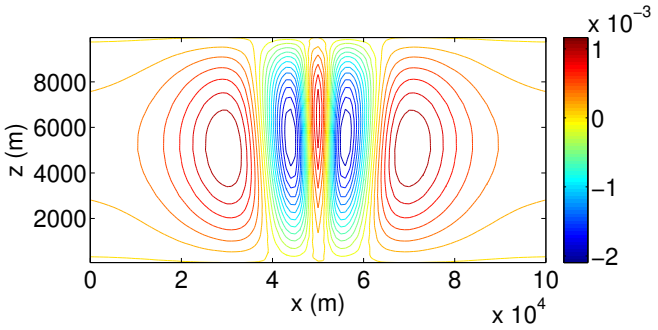


FIGURE 4.4. *Mesoscale Mode: 3D Inertia-Gravity Wave. A slice of the potential temperature perturbation (at y=50 km) after 700 s for* $30 \times 30 \times 5$ *elements with 4th-order polynomials. The contour lines are from* -1×10^{-3} *to* 1×10^{-3} *with an interval of* 1×10^{-4}.

4.2. Mesoscale Mode: 3D Inertia-Gravity Wave. This test case is similar to the two-dimensional inertia-gravity wave test proposed in [27]. For completeness, we now define the statement of the problem. The initial state of the atmosphere is taken to have no mean flow ($\boldsymbol{u}_0 = 0$) in a uniformly stratified atmosphere with Brunt-Väisälä frequency of $\mathcal{N} = 0.01$/s. The definition of Brunt-Väisälä frequency $\mathcal{N}^2 = g\frac{d}{dz}(\ln\theta_0)$ yields

$$\theta_0 = \theta_{00} \mathrm{e}^{\frac{\mathcal{N}^2 z}{g}}$$

where $\theta_{00} = 300$ K. Finally, the potential temperature perturbation is given as

$$\theta' = \theta_c \frac{\sin\left(\frac{\pi z}{h_c}\right)}{1 + \left(\frac{x - x_c}{a_c}\right)^2 + \left(\frac{y - y_c}{a_c}\right)^2}$$

where $\theta_c = 0.01$ K, $h_c = 10000$ m, $a_c = 5000$ m, $x_c = y_c = 50000$ m and the domain is defined as $(x, y, z) \in [0, 100] \times [0, 100] \times [0, 10]$ km with $t \in [0, 700]$ s.

This test case is quite similar to its two-dimensional analog except that the initial condition (the initial perturbation in θ') is perturbed equally in both the x and y directions. The advantage of doing this is that the solution remains symmetric with respect to the x-y directions and, therefore, offers a quick check on the proper behavior of the solution. No mean flow is given in order to simplify the boundary conditions to no-flux (all 6 faces of the three-dimensional cube are hard walls) and to maintain symmetry with respect to the center of the domain.

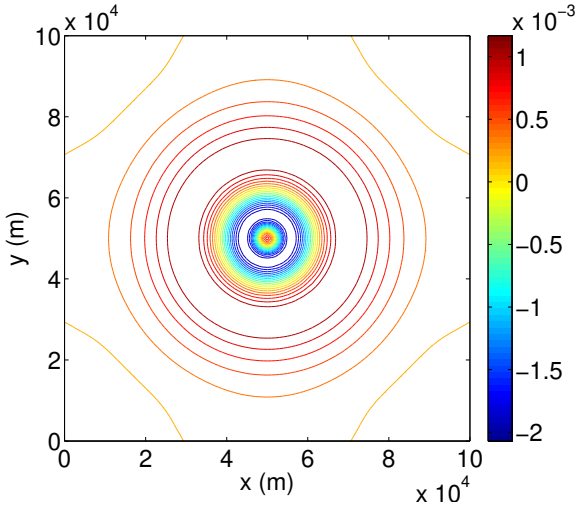


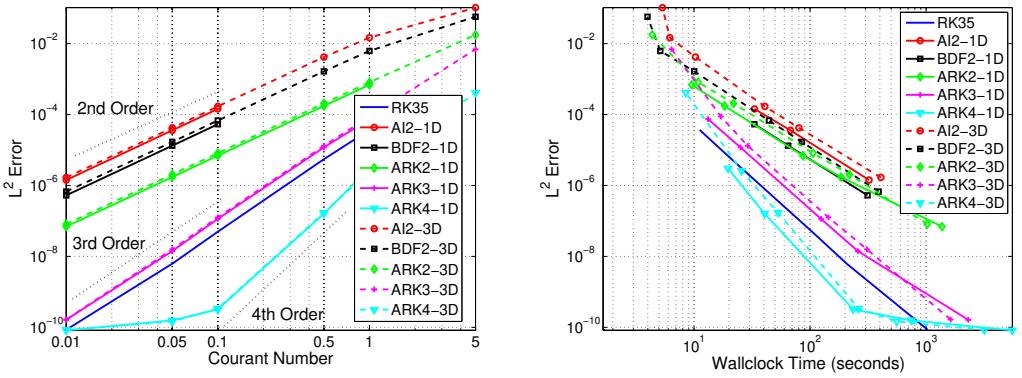
FIGURE 4.5. *Mesoscale Mode: 3D Inertia-Gravity Wave. A slice of the potential temperature perturbation (at z=5 km) after 700 s for* $30 \times 30 \times 5$ *elements with 4th-order polynomials. The contour lines are from* -1×10^{-3} *to* 1×10^{-3} *with an interval of* 1×10^{-4}.

Figure 4.4 shows a slice of the potential temperature perturbation (at $y = 50$ km) at 700 s into the simulation. The initial temperature perturbation has expanded from the initial position at $(x, y) = (50, 50)$ km. Figure 4.5 shows the symmetry of the solution with respect to the xy-plane (taken at $z = 5$ km). Note that perfect symmetry with respect to the center of the domain is maintained. At 800 s, the outer ring of the initial condition hits the boundary and for this reason we run the simulation for fewer than 700 s for the convergence study.

Figure 4.6 shows the accuracy (panel a) and efficiency (panel b) for the various time-integrators considered. For these simulations the grid is comprised of $30 \times 30 \times 5$ elements of 4th-order that results in a grid resolution of 826×476 m with 307461 gridpoints which yields a grid resolution ratio of $R_{GR} = 1.7$. The simulations are run for 200 s, where the L^2 norm is computed using the explicit (RK35) solution with a Courant number of 0.001. For the efficiency study, 96 MPI processes are used for all the simulations.

Figure 4.6a shows that all the time-integrators yield the theoretically expected convergence rates; this is yet another test confirming that the 1D-IMEX time-integrators are functioning properly. In addition, ARK2 yields solutions an order of magnitude more accurate than the other two second order methods and the 1D-IMEX ARK2 method remains stable for larger Courant numbers than BDF2 and AI2.

Figure 4.6b shows the error versus wallclock time. For achieving accuracy levels between 10^{-2} and 10^{-4}, the most efficient time-integrators are the 2nd order methods, in particular ARK2 performs well for accuracy levels above 10^{-4}. For achieving accuracy levels below 10^{-4}, the most efficient time-integrators are the 3rd and 4th order methods. For this simulation, the ratio of horizontal to vertical grid resolution is $R_{GR} = 1.7$ which means that the 1D IMEX methods will offer an advantage over fully explicit methods. An interesting result from this figure is that the 1D and 3D IMEX methods of the same order are equally efficient. For example, for accuracy levels below 10^{-6} we see that the ARK4 dominates with both the 1D and 3D IMEX methods yielding comparable efficiency.



a) b)

FIGURE 4.6. *Mesoscale Mode: Inertia-Gravity Wave. The a) accuracy and b) efficiency for the explicit (RK35), 1D-IMEX, and 3D-IMEX time-integrators. Results are shown for a final time of 200 s.*

4.3. Global-Scale Mode: Inertia-Gravity Wave on the Sphere. The global scale problem we consider is that of inertia-gravity waves traveling around the entire planet [30]. We begin with a hydrostatically balanced initial state with a potential temperature perturbation. The initial condition is defined as a hydrostatically balanced atmosphere with background (reference) potential temperature defined as in the previous case

$$\theta_0 = \theta_{00} \mathrm{e}^{\frac{\mathcal{N}^2 r}{g}}$$

with $\theta_{00} = 300$ K, where z in the previous case has been replaced by the radial component r. The other difference is that the potential temperature perturbation is now given as

$$\theta' = \theta_c f(\lambda, \phi) g(r),$$

where $\theta_c = 10$ K,

$$f(\lambda,\phi) = \begin{cases} 0 & \text{for} \quad R > R_c \\ \frac{1}{2}\left[1 + \cos\left(\frac{\pi R}{R_c}\right)\right] & \text{for} \quad R \leq R_c, \end{cases}$$

and

$$g(r) = \sin\left(\frac{n_v \pi r}{r_T}\right),$$

where $R = R_E \cos^{-1}\left[\sin\phi_0 \sin\phi + \cos\phi_0 \cos\phi \cos(\lambda - \lambda_0)\right]$ is the geodesic distance between the spherical coordinate pairs (λ_0, ϕ_0) and (λ, ϕ), $R_c = R_E/3$, and $n_v = 1$ with $\mathcal{N} = 0.02/s$. The domain for this problem is comprised of the surface of the Earth with a radius of $R_E = 6371$ km and a model altitude of $r_T = 10$ km. According to linear theory, the phase speed of the gravity wave should move as

$$c_{gw} = \frac{\mathcal{N} r_T}{\pi n_v}$$

which, for this problem setup would be $c_{gw} = 63.66$ m/s. Figure 4.7 shows the results after 48 hours for a grid resolution of 138 km in the horizontal by 0.24 km in the vertical ($R_{GR} = 575$) for a total of 566866 gridpoints. This coarse resolution gives a gravity wave phase speed of 66.68 (less than 5% error).

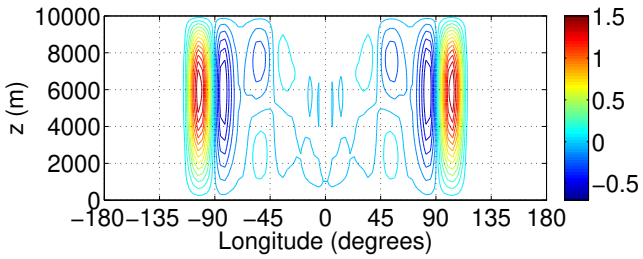


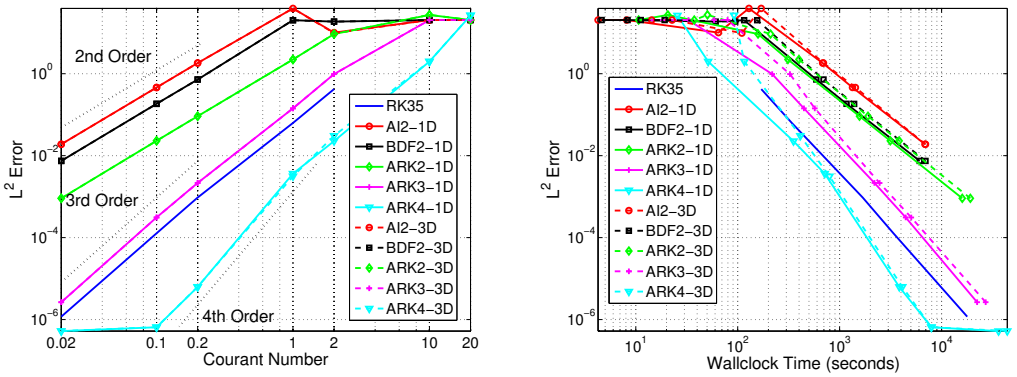
FIGURE 4.7. *Global-Scale Mode: Inertia-Gravity Wave on the Sphere. The potential temperature perturbation after 48 hours for* 864 *elements in the horizontal (spherical surface) and* 10 *elements in the vertical with 4th-order polynomials. Results are shown along the equator (latitude is 0 degrees). The contours lines are from* −0.6 *to* 1.3 *with an interval of* 0.1.

The challenge posed by such a grid ratio is that while the horizontal grid resolution is quite coarse (276 km), the vertical grid resolution is rather fine (0.47 km). The grid resolution ratio of $R_{GR} = 575$ is somewhat typical of the value found in climate applications. Currently, climate models use the hydrostatic equations which do not have vertical acoustic modes. However, as grid resolutions become finer, many global-scale weather models will move towards the nonhydrostatic equations and therefore will face such grid resolution issues. In such future weather models, the vertical direction will be much better resolved than the horizontal and therefore a 1D IMEX method should offer significant savings over a 3D IMEX method. Furthermore, the presence of these "multi-scales" makes this class of problem challenging and representative of the applications that must be properly modeled in large-scale atmospheric dynamics applications of the future (e.g., nonhydrostatic weather prediction).

Figure 4.8 shows the error versus Courant number (left panel) and the error versus wallclock time (right panel) for the various time-integrators used. For this case we use a (cubed-sphere) grid consisting of $216 \times 5 = 1080$ elements (horizontal x vertical) each with 4th-order polynomials for a total of 72000 gridpoints. The model is integrated for 10000 s, where the explicit RK35 solution with a Courant number of 0.002 is used as the exact solution. For all the simulations, 96 MPI processes are used. Figure 4.8a confirms that the 1D IMEX methods (solid lines) yield the same accuracy as their 3D IMEX counterparts (the dashed and solid lines are on top of each other for all 1D and 3D IMEX methods). This shows that the derivation of the generalized 1D IMEX approach has been derived and implemented correctly for spherical geometries as well as Cartesian geometries (two previous simulations).

Turning now to the efficiency of the time-integrators, Fig. 4.8b shows that the most efficient time-integrators for accuracy levels above 10^0 are the 2nd order methods (both 1D and 3D IMEX methods). For accuracy levels below 10^0 the high-order methods are more efficient than the low-order methods, that is, the 3rd and 4th order methods are more efficient than the 2nd order methods. In summary, these results show the value of high-order time-accuracy because the dominant methods for the highest levels of accuracy are the ARK3, RK35, and ARK4 methods. Note that the RK35 (explicit results) will always yield more accurate results than the IMEX methods for any stable time-step. Recall that the RK35 is what we use to compute the *exact* solution and therefore treats all waves accurately. Therefore, as the time-step is increased it is expected that this method will yield more accurate results than an IMEX method of the same order (that does not handle the acoustic waves accurately). We include the results of the RK35 method nonetheless although it is not completely a fair comparison. Additionally, these results show that the 1D IMEX methods are

only slightly more efficient than the 3D IMEX methods even for grid resolution regimes $R_{GR} >> 1$. This may seem surprising at first glance since one might expect the 1D IMEX methods to be faster for the same time-step size (since no 3D matrix problem needs to be inverted). However, because the time-step sizes reported in Fig. 4.8b are relatively small, the iterative solvers in the 3D IMEX methods do not require many iterations. Nonetheless, the fact that the 1D and 3D IMEX methods are costing the same is a tribute to the good design of the 3D IMEX methods and their associated machinery (solvers and preconditioners, which are beyond the scope of the current work).



a) b)

FIGURE 4.8. *Global-Scale Mode: Inertia-Gravity Wave on the Sphere. The a) accuracy and b) efficiency for the explicit (RK35), 1D-IMEX, and 3D-IMEX time-integrators. Results are shown for a final time of 10000 s.*

4.4. Conservation. The last comparison we show concerns the conservation properties of the time-integrators. We choose the global-scale problem because it represents the longest simulation of all the three test cases considered. Another reason is due to the fact that for this problem the stiffness is unidirectional (along the radial direction), and so both the 1D and 3D IMEX methods allow for very large time-steps (Courant numbers) with respect to the radial direction. We also use this test case to highlight the conservation measures because it is deemed a more difficult problem due to the spherical geometry.

For this comparison we define the mass and energy loss as

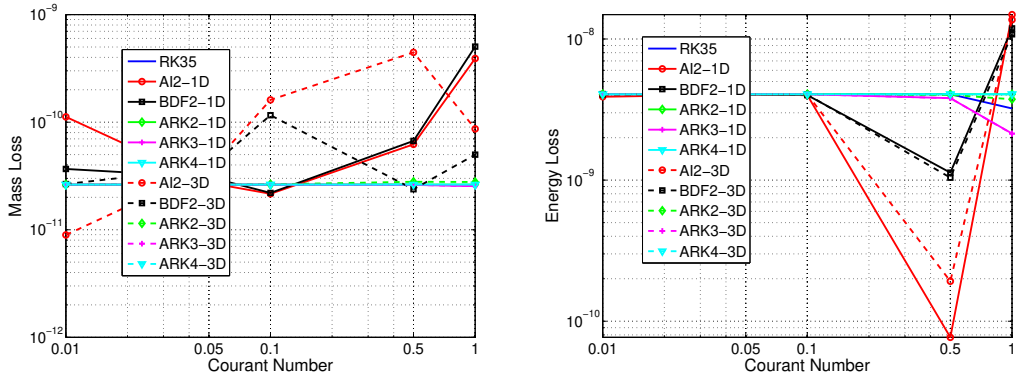
$$\text{Mass Loss} = \left| \frac{\text{Mass}(t) - \text{Mass}(0)}{\text{Mass}(0)} \right|, \quad \text{Energy Loss} = \left| \frac{\text{Energy}(t) - \text{Energy}(0)}{\text{Energy}(0)} \right|,$$

where Mass(t) and Energy(t) is the mass/energy at time t, where we compare the difference between the initial mass, Mass(0), and energy, Energy(0).

The mass and energy are defined as

$$\text{Mass(t)} = \int_\Omega \rho d\Omega, \quad \text{Energy(t)} = \int_\Omega \rho e d\Omega,$$

where ρ and e are the total density and energy, with the total energy defined as $e(t) = c_v T(t) + \frac{\boldsymbol{u}\cdot\boldsymbol{u}}{2} + g(R - R_E)$ (internal, kinetic, and potential energies, respectively), with R being the radial distance from the center of the Earth and R_E being the radius of the Earth.

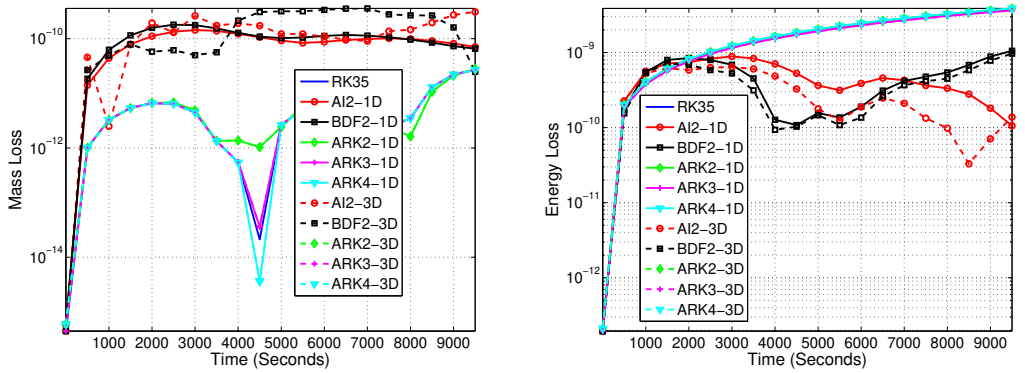


a) b)

FIGURE 4.9. *Conservation. The a) mass loss and b) energy loss for the explicit (RK35), 1D-IMEX, and 3D-IMEX time-integrators as a function of Courant number. Results are shown for the global-scale problem at a final time of 10000 s.*

Figure 4.9a shows that all the Runge-Kutta time-integrators maintain the same level of mass conservation regardless of the time step. However, the BDF2 and AI2 do not, although on average they conserve mass to the same level as the Runge-Kutta methods (in Fig. 4.9a we see that the mass conservation measures oscillate for both BDF2 and AI2 approximately about the mass conservation level of the Runge-Kutta methods). Similarly, the energy conservation remains constant with time-step for the Runge-Kutta methods but oscillates for both BDF2 and AI2. In sum, the Runge-Kutta methods (explicit and IMEX) are more consistent with respect to time-step size and conservation metrics than BDF2 and AI2. One should insist on the time-integrator to yield the same mass and energy conservation independent of the time-step used and this is clearly provided by the Runge-Kutta methods.

To better understand the behavior we see in Fig. 4.9 we now show the conservation measures of mass and energy throughout the 10000 s simulation in Fig. 4.10.



a) b)

FIGURE 4.10. *Time Series of Conservation. The a) mass loss and b) energy loss for the explicit (RK35), 1D-IMEX, and 3D-IMEX time-integrators as a function of simulation time. Results are shown for the global-scale problem throughout a 10000 s simulation for a Courant number of* 0.5.

For this time series analysis, we use a Courant number of 0.5 because this is the largest value for which all the time-integrators are stable (the explicit RK35 method becomes unstable for larger Courant numbers). Figure 4.10a shows that the mass loss for all the Runge-Kutta methods are lower than those for the BDF2 and AI2 methods. Figure 4.10b shows the energy loss for the Runge-Kutta methods to be higher than those for BDF2 and AI2. However, in Fig. 4.9b we see that at a

Courant number of 0.5, the BDF2 and AI2 yield a minimum energy loss but this is not sustained for all Courant numbers. In contrast, both the mass and energy loss are exactly the same for all Courant numbers for the Runge-Kutta methods. It is desirable to use a time-integrator that yields consistent metrics regardless of the Courant number used; all the Runge-Kutta methods behave in this desirable fashion.

To try to explain the consistent behavior of conservation with time-step size for Runge-Kutta methods, let us take a closer look at the ARK and linear multistep calculations. Consider a weight vector $\mathbf{e}$. If $S(\boldsymbol{q})$ is a conservative discretization (with linear invariant $\boldsymbol{q}$) then $\mathbf{e}^{\mathcal{T}} S(\boldsymbol{q}) = 0$ so that $\mathbf{e}^{\mathcal{T}}\,\boldsymbol{q} = \text{constant}$. Then, following Eq. (3.6b), we obtain

$$\begin{aligned}\mathbf{e}^{\mathcal{T}}\boldsymbol{q}^{n+1} &= \mathbf{e}^{\mathcal{T}}\boldsymbol{q}^{n} + \Delta t \sum_{i=1}^{s} b_i \mathbf{e}^{\mathcal{T}} \left(S(\boldsymbol{Q}^{(i)}) - \delta L(\boldsymbol{Q}^{(i)}) \right) + \Delta t \sum_{i=1}^{s} \tilde{b}_i \mathbf{e}^{\mathcal{T}} \left(\delta L(\boldsymbol{Q}^{(i)}) \right) \\ &= \mathbf{e}^{\mathcal{T}}\boldsymbol{q}^{n} + \Delta t \sum_{i=1}^{s} b_i \left(\mathbf{e}^{\mathcal{T}} S(\boldsymbol{Q}^{(i)}) \right) = \mathbf{e}^{\mathcal{T}}\boldsymbol{q}^{n}, \qquad \text{because} \;\; b = \tilde{b}. \end{aligned} \tag{4.3}$$

Therefore, the ARK methods behave in this regard like an explicit method and preserve all linear invariants to machine precision (assuming that the function S is conservative at every stage). This will be the case for both the 1D and 3D IMEX methods which we see to be true in Figs. 4.9 and 4.10. In other words, the accuracy to which the linear system (3.8d) is solved at every stage does not play a role in the linear conservation measures, which is a property resulting from (4.3). On the other hand, linear multistep methods evolve subject to the linear solve in Eq. (3.5). That is, from Eq. (3.4) at each step we have

$$\mathbf{e}^{\mathcal{T}}\boldsymbol{q}^{n+1} = \mathbf{e}^{\mathcal{T}}(I - \lambda L)^{-1} \left(\widehat{\boldsymbol{q}} + \sum_{k=0}^{K-1} \check{\beta}_k \boldsymbol{q}^{n-k} - \lambda L \sum_{k=0}^{K-1} \check{\beta}_k \boldsymbol{q}^{n-k} \right), \tag{4.4}$$

which is also conservative if system (3.5) is solved exactly. However, this is not the case if, for instance, iterative solvers are used and stopped before reaching machine precision; therefore, the preservation of linear invariants for the linear multistep methods presented here is correct only asymptotically. On the other hand, the 1D IMEX method should, in principle, conserve as long as the function S is conservative at every instance of the multi-step method; this is due to the fact that linear systems (4.4) or (3.5) are solved exactly. However, because $S(\boldsymbol{q})$ is not conservative we still see a degradation of the conservation measures.

4.5. Linear Stability Analysis. Although not entirely appropriate for systems of nonlinear partial differential equations, it is revealing to apply linear stability analysis to time-integration methods. The following analysis will not only tell us something about the time-integration methods used in this work but it will also help explain the nature of the IMEX strategy.

We begin by writing the following ordinary differential equation

$$\frac{dq}{dt} = \{ik_s q\} + [ik_f q] \tag{4.5}$$

where q is our solution variable, $i = \sqrt{-1}$, and k_s and k_f are the wave speeds due to the *slow* (k_s) and *fast* (k_f) modes in the system. This equation originates from first writing the general wave equation and then introducing a Fourier (exact) solution of the spatial derivatives in order to isolate time from the original partial differential equation. In an IMEX approach the strategy is then to linearize the original stiff nonlinear operator such that we are able to extract a system that looks like Eq. (4.5); this is achieved by first discretizing in space and linearizing in time (about the fast waves). In Eq. (4.5) we have added the curly and square brackets to remind the reader which terms are handled explicitly $\{k_s\}$ and which implicitly $[k_f]$. We note that this analysis also assumes that the linearized implicit and explicit terms can be diagonalized simultaneously, which is not typically the case; however, this analysis is still useful to understand the stability behavior of the IMEX form beyond the scalar case.

Rather than showing the stability analysis of all the methods used in this paper, we concentrate on the 2nd order methods because the ARK2 is the only new method created in this work. Let us begin with the other two 2nd order methods. The stability analysis for all the methods presented use contour levels for the amplification factors from 0 to 1 with a contour interval of 0.05.

The stability analysis of multi-step methods requires extracting the M roots of the Mth order polynomial in q (where M denotes the maximum order of all the components of the IMEX time-integrator). For the case of BDF2, $M = 2$ which yields the two roots given in Fig. 4.11 where one root is the physical mode (first mode) and the other is the computational mode (second mode). As was shown in [10] the attractive property of BDF2 is that its computational mode is well damped.

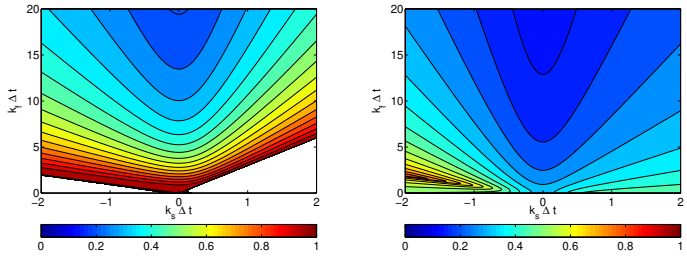


FIGURE 4.11. *Stability Analysis: Amplification factors for BDF2 for a) first mode and b) second mode, as functions of the slow (horizontal) and fast (vertical) Courant numbers. Contour levels range from 0 to 1 with a contour interval of 0.05.*

In Fig. 4.12 we show the stability analysis for the AI2 method. For this method $M = 3$ since it is a combination of a 2nd order Adams (Moulton) implicit component with a 3rd order Adams (Bashforth) explicit part.

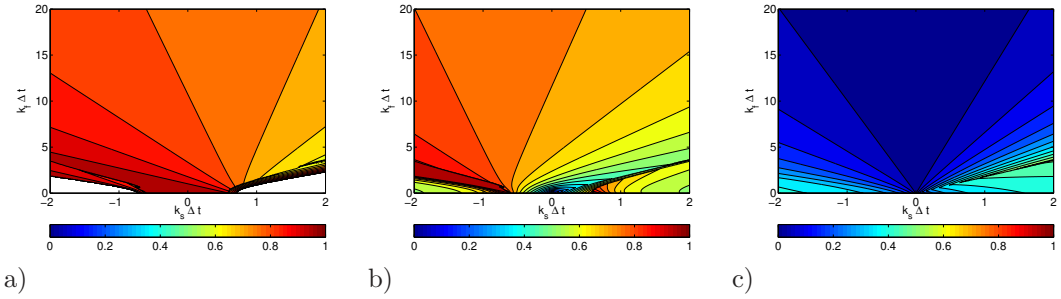


FIGURE 4.12. *Stability Analysis: Amplification factors for AI2 for a) first mode, b) second mode, and c) third mode, as functions of the slow (horizontal) and fast (vertical) Courant numbers. Contour levels range from 0 to 1 with a contour interval of 0.05.*

For single-step multi-stage methods, the stability analysis is much more straightforward since the stability condition arises from the solution of a polynomial (linear in q) in Δt of order $M = S$ where S denotes the number of stages. This solution can also be represented as a rational function in $k_s\Delta t$ and $k_f\Delta t$. The most favorable interpretation of the stability analyses for both BDF2 (Fig. 4.11) and AI2 (Fig. 4.12) is that the first mode represents the physical mode and the remaining modes are the computational ones. In this interpretation it is clear that the computational modes are all well damped and so we only need to compare the physical modes with the only mode (physical) obtained from ARK2. Under this assumption we can now discuss the plots in Fig. 4.13.

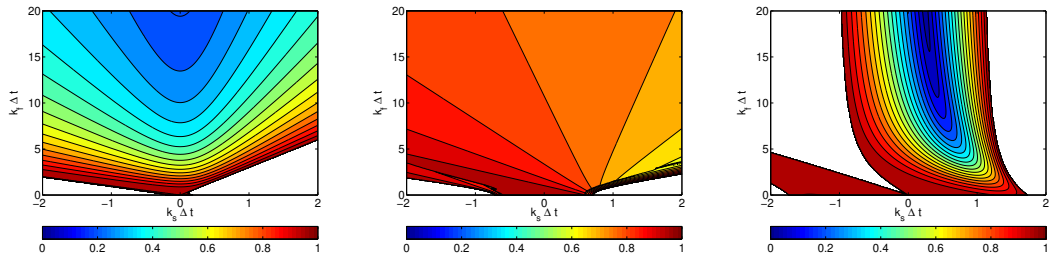


FIGURE 4.13. *Stability Analysis: Amplification factors for a) BDF2, b) AI2, and c) ARK2 as functions of the slow (horizontal) and fast (vertical) Courant numbers. Contour levels range from 0 to 1 with a contour interval of 0.05.*

Figure 4.13 shows the stability regions for BDF2, AI2, and ARK2 for the Courant number ranges used in all the simulations (maximum Courant numbers of 20 for the fast waves and below 2 for the slow waves). This figure shows that for fully explicit time-integration, ARK2 has a much larger stability region than either BDF2 or AI2 (bottom of the figures for which $k_f \Delta t = 0$). However, as we consider the maximum Courant number for the slow waves (maximum value of $k_s \Delta t$) and increase the fast wave Courant number ($k_f \Delta t$) we see that both BDF2 and AI2 damp the solution whereas ARK2 does not. On the other hand for slow waves that are $k_s \Delta t = 0.5$, ARK2 and BDF2 strongly damp the solution for all $k_f \Delta t > 1$ while AI2 does not (AI2 does damp the solution but not as strongly). Looking again at Fig. 4.13 it is no surprise why BDF2 is such a successful method, i.e., it is extremely stable because it strongly damps the solution.

These results also indicate that BDF2 and AI2 yield stable solutions with the IMEX method for Courant numbers for which using just the explicit part (k_s) would result in an unstable solution. In other words, the implicit part stabilizes the explicit part; moreover, stability would be guaranteed by increasing the time step. However, given fixed k_s and k_f (which are problem and spatial discretization dependent) we note that for this situation to occur requires k_s and k_f to be roughly of the same size. Moreover, k_f must be clearly separated from the origin so that k_s remains well inside the stable domain. Furthermore, as explained in Sec. 4.1 (Large Time-Step Behavior), the time-step should be small enough to resolve the waves in the term treated explicitly. Therefore, as illustrated by the numerical experiments presented here, the dominant unstable modes affect the area close to small k_f, which is largest for ARK2. We also note that ARK2 can be made to have the same "fanning out" as BDF2 and AI2 of the stability region along the imaginary axis if one sets $a_{32} = 1/2$; however, this did not bring any additional stability benefits on experiments carried out on the rising thermal bubble problem. Therefore, what is important to emphasize is that while this stability analysis gives us insight into the stability of a specific method, it will tell us nothing about the accuracy or the efficiency of the method. For this reason, one must be careful to perform both the stability analysis and numerical experiments as we have done here. On that note, it is worth discussing the performance of the time-integrators on the numerical test cases in order to complement the stability analysis presented.

For the Cloud-resolving problem in Fig. 4.2a we note that in practice, the stability regions of both ARK2 and AI2 are indeed larger than that for BDF2 (where we see that the solid black/square line is shorter than the green/diamond and red/circle lines). Turning now to the Mesoscale problem, Fig. 4.6a shows that the stability region of ARK2 is significantly larger than those for either BDF2 and AI2 (an order of magnitude). Finally, for the Global-scale problem, Fig. 4.8a shows that all three 2nd order methods are able to run with the entire range of Courant numbers used (this is because this is a very stiff problem where the stiffness is unidirectional and arising through the vertical acoustic waves which are being handled implicitly and so all IMEX methods should be able to run stably with much larger Courant numbers than shown here). To supplement the discussion on accuracy and stability we have also shown efficiency plots which show that the BDF2 method is actually quite efficient when considering the fastest time to a given level of accuracy. In all three simulations, the BDF2 and ARK2 methods outperformed the AI2 method. This result along with

the more robust conservation measures obtained by ARK2 lead us to recommend ARK2 among the 2nd order time-integrators studied.

5. Conclusions. We have derived implicit-explicit (IMEX) formulations for the 3D Euler equations with a unified representation of various nonhydrostatic flow regimes including cloud-resolving and mesoscale (flow in a 3D Cartesian domain) as well as global regimes (flow in spherical geometry). This general IMEX formulation admits numerous types of methods including single-stage multi-step methods (e.g., AI2 and BDF2) and multi-stage single-step methods (e.g., the additive Runge-Kutta methods). The significance of this general IMEX formulation is that it allows a numerical model to reuse the same machinery for every time-integration method; for example, the calls to the spatial discretization are exactly the same for all the time-integration methods studied in this paper. Moreover, we have introduced and tested a new L-stable second-order additive Runge-Kutta method and have shown it to be the best second order method studied in this work. In addition, we compared two classes of IMEX methods: 1D and 3D. The 3D IMEX approach is more straightforward to implement and performs well although it relies heavily on good preconditioners and iterative solvers. However, the 3D IMEX methods should be at a disadvantage when the problem has stiffness along only one of the spatial directions. For this type of unidirectional stiffness, the 1D IMEX methods should be the clear winners but we did not observe this for the Courant numbers that were used (less than 20). It is quite possible that for very large Courant numbers, the 3D IMEX methods may not compete with the 1D IMEX methods due to the number of iterations they require for convergence - the 1D IMEX methods do not require iterative solvers.

For problems where the stiffness is multi-directional, the 3D IMEX methods should perform best. Therefore, it is important to include various choices of time-integrators into a model if one wishes to use it for various applications with particular grid resolution characteristics that may exacerbate the stiffness of the problem. In summary, the choice of which method to use to achieve the fastest integration depends on the grid resolution ratio (horizontal to vertical). All the grid resolution regimes showed that the maximum efficiency (fastest time to achieve an accurate solution) is best achieved by the use of high-order time-integration methods. Even if one is not willing to pay the price of additional computational time to achieve such levels of accuracy, one must be mindful of the quality of the solution that one should expect by using more efficient yet lower-order time-integration methods.

The next step in this research is to perform a detailed study of the scalability of the 1D and 3D IMEX methods on massively parallel computers. We have previously demonstrated that the explicit RK35 time-integrator exhibits strong linear scaling for processor counts of the order 10^5 (see [20]); we expect that the 1D IMEX methods will perform the same because they have the same communication footprint as an explicit method, that is, they only require communication across vertex neighbors. On the other hand, constructing perfectly scalable 3D IMEX methods remains a challenge because these methods rely on iterative solvers and preconditioners (too many iterations will destroy perfect scalability because iterative solvers require all-to-all communication). For the past few years we have been constructing scalable preconditioners (see [5]) and have made advances, but this remains an open topic. Upon completing our work on preconditioners we will report the scalability of the 1D and 3D IMEX methods for large processor counts.

Acknowledgements. FXG and JFK gratefully acknowledge the support of the Office of Naval Research through program element PE-0602435N and the National Science Foundation (Division of Mathematical Sciences) through program element 121670. FXG and EMC gratefully acknowledge the support of the Air Force Office of Scientific Research through the Computational Mathematics program. EMC was also supported in part by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357. Discussions with Sarah-Jane Lock and Hilary Weller on IMEX time-integrators during the "Multiscale Numerics of Atmosphere and Ocean" program at the Newton Institute, Cambridge University, are gratefully acknowledged.

Appendix A. Deriving Unified Balanced Equations. In this appendix, we prove that the Euler equations can be written in a unified way for use in any type of geometry using Cartesian coordinates where gravity acts along a specified direction denoted by the vector $\bar{\boldsymbol{r}}$. There exists work

in the literature on deriving forms of the Euler equations that are valid for, say, spherical geometry under a specified coordinate invariant form. In our case we can represent the Euler equations in any form and, while using Cartesian coordinates (or other coordinate systems), can represent a diverse set of forms of the governing equations. We only show a simple derivation of the form used in Eq. (2.2) but the procedure is the same for other forms (e.g., conservation forms of density - density potential temperature or density-energy).

THEOREM A.1. *The equations*

$$\begin{aligned} \frac{\partial \rho'}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\rho' + \boldsymbol{u}\cdot\boldsymbol{\nabla}\rho_0 + (\rho'+\rho_0)\boldsymbol{\nabla}\cdot\boldsymbol{u} &= 0 \\ \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} + \frac{1}{\rho'+\rho_0}\left(\boldsymbol{\nabla}P' + \boldsymbol{\mathcal{H}}\boldsymbol{\nabla}P_0\right) + \frac{\rho'}{\rho'+\rho_0}g\bar{\boldsymbol{r}} + f\bar{\boldsymbol{r}}\times\boldsymbol{u} &= \boldsymbol{0} \\ \frac{\partial \theta'}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\theta' + \boldsymbol{u}\cdot\boldsymbol{\nabla}\theta_0 &= 0, \end{aligned} \tag{A.1}$$

represent the balanced Euler equations written in Cartesian coordinates and are valid for any geometry with gravity acting along the $\bar{\boldsymbol{r}}$ *direction*

Proof. We begin by writing the Euler equations in their original (density-potential temperature) form

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}) &= 0 \\ \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} + \frac{1}{\rho}\boldsymbol{\nabla}P + g\bar{\boldsymbol{r}} + f\bar{\boldsymbol{r}}\times\boldsymbol{u} &= \boldsymbol{0} \\ \frac{\partial \theta}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\theta &= 0 \end{aligned} \tag{A.2}$$

where, upon, introducing the splitting $\rho(\boldsymbol{x},t) = \rho_0(\boldsymbol{x}) + \rho'(\boldsymbol{x},t)$, $\theta(\boldsymbol{x},t) = \theta_0(\boldsymbol{x}) + \theta'(\boldsymbol{x},t)$, $P(\boldsymbol{x},t) = P_0(\boldsymbol{x}) + P'(\boldsymbol{x},t)$ yields the equations for mass (ρ') and energy (θ') given in Eq. (A.1). Next, we expand the momentum equation ($\boldsymbol{u}$) in Eq. (A.2) using this variable splitting to arrive at

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} + \frac{1}{(\rho_0+\rho')}\left[\boldsymbol{\nabla}P_0 + \rho_0 g\bar{\boldsymbol{r}} + \boldsymbol{\nabla}P' + \rho' g\bar{\boldsymbol{r}}\right] + f\bar{\boldsymbol{r}}\times\boldsymbol{u} = \boldsymbol{0}. \tag{A.3}$$

If the reference fields are required to be hydrostatically balanced then we require that the first two terms in square brackets satisfy the condition

$$\bar{\boldsymbol{r}}^{\mathcal{T}}\boldsymbol{\nabla}P_0 + \rho_0 g = 0$$

where we have used the fact that $||\bar{\boldsymbol{r}}||_2 = 1$. Using this condition for $\rho_0 g$ allows us to write the first two terms in square brackets as

$$\boldsymbol{\nabla}P_0 + \rho_0 g\bar{\boldsymbol{r}} \equiv \boldsymbol{\nabla}P_0 - \bar{\boldsymbol{r}}^{\mathcal{T}}\boldsymbol{\nabla}P_0\bar{\boldsymbol{r}} = (\boldsymbol{I} - \bar{\boldsymbol{r}}\bar{\boldsymbol{r}}^{\mathcal{T}})\boldsymbol{\nabla}P_0$$

where, when we define $\boldsymbol{\mathcal{H}} = (\boldsymbol{I} - \bar{\boldsymbol{r}}\bar{\boldsymbol{r}}^{\mathcal{T}})$ recovers Eq. (A.1) which is valid for any geometry. □

Note that for the special case $\bar{\boldsymbol{r}} = (0,0,1)^{\mathcal{T}}$ (e.g., flow in a box), the term $\boldsymbol{\mathcal{H}}\boldsymbol{\nabla}P_0$ simplifies to $(\frac{\partial P_0}{\partial x}, \frac{\partial P_0}{\partial y}, 0)^{\mathcal{T}}$ where, if $P_0 = P_0(z)$, then the entire term $\boldsymbol{\mathcal{H}}\boldsymbol{\nabla}P_0$ vanishes.

The value of the formulation described above is that a balanced reference field can be built into the governing equations for a variety of reference states. For example, the condition used to derive $\boldsymbol{\mathcal{H}}$ was based on hydrostatic balance but the remaining terms in the reference field can satisfy other balance conditions as well, including, e.g., geostrophic balance.

Appendix B. Effects of Stabilization on Time Convergence Rates. Two popular mechanisms for stabilizing Galerkin methods are: 1) low-pass filters and 2) artificial viscosity. Our intent here is not to present an exhaustive discussion on these mechanisms but rather to discuss the choice of using artificial viscosity for this study instead of filters. To see how these two mechanisms could affect the time convergence rates, let us begin with the linear system of equations written as follows:

$$\frac{d\boldsymbol{q}}{dt} = S(\boldsymbol{q})$$

where $S(\boldsymbol{q})$ denotes the spatially discretized differential operators. If we now apply a forward Euler method in time, results in the following fully discrete form

$$\boldsymbol{q}^{n+1} = \boldsymbol{q}^n + \Delta t S(\boldsymbol{q}^n)$$

where n and $n+1$ denote the times t^n and $t^{n+1} = t^n + \Delta t$. From this last equation, we can see that if S is a linear operator with respect to $\boldsymbol{q}$ then the evolution of this equation in time becomes

$$\begin{aligned}
\boldsymbol{q}^1 &= \boldsymbol{q}^0 + \Delta t S \boldsymbol{q}^0 \\
\boldsymbol{q}^2 &= \boldsymbol{q}^1 + \Delta t S \boldsymbol{q}^1 = \boldsymbol{q}^0 + \Delta t S \boldsymbol{q}^0 + \Delta t S \boldsymbol{q}^0 + (\Delta t S)^2 \boldsymbol{q}^0 \\
&\vdots \\
\boldsymbol{q}^k &= (I + \Delta t S)^k \boldsymbol{q}^0
\end{aligned}$$

where we can see that at time-step k the solution is related to the initial condition through the operator $R = (I + \Delta t S)^k$. If we now consider applying a low-pass filter through the filter matrix F in the usual *a posteriori* fashion, this results in the following form

$$\widetilde{\boldsymbol{q}}^{n+1} = F\left(\boldsymbol{q}^n + \Delta t S(\boldsymbol{q}^n)\right) \cdots \widetilde{\boldsymbol{q}}^k = (F + \Delta t F S)^k \boldsymbol{q}^0$$

where $\widetilde{\boldsymbol{q}}$ denotes the filtered solution. This simple derivation reveals that if F is not idempotent (i.e., $F = F^2$) the amount of filtering is dependent on the number of time-steps (k) we require to reach a final time (the first term on the right-hand-side). This means that when using a very small time-step, which we call the "exact" solution, will have a different amount of filtering and thereby represent a different solution than the IMEX simulations that use a larger time-step with a corresponding smaller number of time-steps (k). This will hinder obtaining the correct time rates of convergence. Note, however, that this will not affect the spatial rates of convergence because a very small time-step is used which removes the time error from the (spatial) convergence rates (see, e.g., [19] for a discussion on how to circumvent the issues with non-idempotent filters).

To see the difference with artificial viscosity, we note that the discretized operators are written as follows:

$$\frac{d\boldsymbol{q}}{dt} = S(\boldsymbol{q}) + \mu L(\boldsymbol{q})$$

where μ is the viscosity parameter and L is the linear hyper-viscosity operator. This equation fully discretized (in both space and time) yields the following form

$$\boldsymbol{q}^k = (I + \Delta t S + \Delta t \mu L)^k \boldsymbol{q}^0$$

where we can now define $S' = S + \mu L$ to be a new operator that now yields

$$\boldsymbol{q}^k = (I + \Delta t S')^k \boldsymbol{q}^0$$

that now looks like the original equation from above. In essence, we have changed the original operator but see that there is no difficulty with obtaining a convergent solution.

REFERENCES

[1] U. Ascher, S. Ruuth, and R. Spiteri, *Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations*, Applied Numerical Mathematics, 25 (1997), pp. 151–167.

[2] U. Ascher, S. Ruuth, and B. Wetton, *Implicit-explicit methods for time-dependent partial differential equations*, SIAM J. Numer. Anal., 32 (1995), pp. 797–823.

[3] J. Butcher, *Numerical Methods for Ordinary Differential Equations*, Wiley, second ed., June 2008.

[4] J. Butcher and D. Chen, *A new type of singly-implicit Runge-Kutta method*, Applied numerical mathematics, 34 (2000), pp. 179–188.

[5] L. E. Carr, III, C. F. Borges, and F. X. Giraldo, *An Element-based Spectrally Optimized Approximate Inverse Preconditioner for the Euler Equations*, SIAM Journal on Scientific Computing, 34 (2012), pp. B392–B420.

[6] J. R. Dea, F. X. Giraldo, and B. Neta, *High-order non-reflecting boundary conditions for the linearized 2-D Euler equations: No mean flow case*, Wave Motion, 46 (2009), pp. 210–220.
[7] D. Durran and P. Blossey, *Implicit-explicit multistep methods for fast-wave slow-wave problems*, Monthly Weather Review, 140 (2012), pp. 1307–1325.
[8] A. Fournier, M. Taylor, and J. Tribbia, *The spectral element atmosphere model (SEAM): High-resolution parallel computation and localized resolution of regional dynamics*, Monthly Weather Review, 132 (2004), pp. 726–748.
[9] S. Gabersek, F. X. Giraldo, and J. D. Doyle, *Dry and Moist Idealized Experiments with a Two-Dimensional Spectral Element Model*, Monthly Weather Review, 140 (2012), pp. 3163–3182.
[10] F. X. Giraldo, *Semi-implicit time-integrators for a scalable spectral element atmospheric model*, Q. J. R. Meteorol. Soc., 131 (2005), pp. 2431–2454.
[11] ———, *Hybrid Eulerian-Lagrangian semi-implicit time-integrators*, Computers & Mathematics with applications, 52 (2006), pp. 1325–1342.
[12] F. X. Giraldo and M. Restelli, *A study of spectral element and discontinuous Galerkin methods for the Navier–Stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases*, J. Comp. Phys., 227 (2008), pp. 3849–3877.
[13] ———, *High-order semi-implicit time-integrators for a triangular discontinuous galerkin oceanic shallow water model*, Int. J. Numer. Meth. Fl., 63 (2010), pp. 1077–1102.
[14] F. X. Giraldo, M. Restelli, and M. Laeuter, *Semi-implicit formulations of the Navier-Stokes equations: application to nonhydrostatic atmospheric modeling*, SIAM Journal on Scientific Computing, 32 (2010), pp. 3394–3425.
[15] F. X. Giraldo and T. E. Rosmond, *A scalable spectral element eulerian atmospheric model (SEE-AM) for NWP: Dynamical core tests*, Monthly Weather Review, 132 (2004), pp. 133–153.
[16] M. Günther, A. Kværnø, and P. Rentrop, *Multirate partitioned Runge-Kutta methods*, BIT, 41 (2001), pp. 504–514.
[17] E. Hairer, S. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer, 1993.
[18] W. Hundsdorfer and S. Ruuth, *IMEX extensions of linear multistep methods with general monotonicity and boundedness properties*, Journal of Computational Physics, 225 (2007), pp. 2016–2042.
[19] A. Kanevsky, M. H. Carpenter, and J. S. Hesthaven, *Idempotent filtering in spectral and spectral element methods*, Journal of Computational Physics, 220 (2006), pp. 41–58.
[20] J. F. Kelly and F. X. Giraldo, *Continuous and discontinuous Galerkin methods for a scalable three-dimensional nonhydrostatic atmospheric model: Limited-area mode*, Journal of Computational Physics, 231 (2012), pp. 7988–8008.
[21] C. Kennedy and M. Carpenter, *Additive Runge-Kutta schemes for convection-diffusion-reaction equations*, Appl. Numer. Math., 44 (2003), pp. 139–181.
[22] J. Lambert, *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*, Wiley, 1991.
[23] J. M. Lindquist, F. X. Giraldo, and B. Neta, *Klein-Gordon equation with advection on unbounded domains using spectral elements and high-order non-reflecting boundary conditions*, Applied Mathematics and Computation, 217 (2010), pp. 2710–2723.
[24] J. M. Lindquist, B. Neta, and F. X. Giraldo, *A spectral element solution of the Klein-Gordon equation with high-order treatment of time and non-reflecting boundary*, Wave Motion, 47 (2010), pp. 289–298.
[25] S. Marras, J. F. Kelly, F. X. Giraldo, and M. Vazquez, *Variational multiscale stabilization of high-order spectral elements for the advection-diffusion equation*, Journal of Computational Physics, 231 (2012), pp. 7187–7213.
[26] L. Pareschi and G. Russo, *Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation*, Journal of Scientific Computing, 25 (2005), pp. 129–155.
[27] W. Skamarock and J. Klemp, *Efficiency and accuracy of the Klemp–Wilhelmson time-splitting technique*, Mon. Wea. Rev., 122 (1994), pp. 2623–2630.
[28] S. Skelboe, *Stability properties of backward differentiation multirate formulas*, Applied Numerical Mathematics, 5 (1989), pp. 151–160.
[29] R. Spiteri and S. Ruuth, *A new class of optimal high-order strong-stability-preserving time discretization methods*, SIAM J. Numer. Anal., 40 (2002), pp. 469–491.
[30] H. Tomita and M. Satoh, *A new dynamical framework of nonhydrostatic global model using the icosahedral grid*, Fluid Dynamics Research, 34 (2004), pp. 357–400.
[31] P. Ullrich and C. Jablonowski, *Operator-Split Runge-Kutta-Rosenbrock Methods for Nonhydrostatic Atmospheric Models*, Monthly Weather Review, 140 (2012), pp. 1257–1284.