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 implicit-explicit (IMEX) formulations 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 formulation 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). This is an important result because it allows a numerical model to reuse the same machinery for all 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. We then compare all the IMEX methods in terms of accuracy and efficiency for three types of geophysical fluid dynamics problems: buoyant convection, density current, and a planetary acoustic wave.

1. Introduction. In a previous article [18] 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 kilometers); 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 perfect scalability 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 [8]) that allows for a unification across various metrics. NUMA is unified in terms of spatial discretization methods that it can use which include high-order continuous and discontinuous Galerkin methods [11, 18]; 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 kilometer), mesoscale model (resolution of 1 kilometer to tens of kilometers), and global model (resolution of tens to hundreds of kilometers) 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 depending on the application.

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., [23, 15]) 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 [12] for other possibilities) but, based on [12], we have chosen to use the following form:

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

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

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

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, $\widehat{\boldsymbol{r}} = (r_x, r_y, r_z)^{\mathcal{T}}$ is the unit vector pointing in the radial direction, and f is the Coriolis parameter. In mesoscale mode (i.e., flow in a box) $\widehat{\boldsymbol{r}} = \widehat{\boldsymbol{k}}$, the unit vector along the z direction, and in global mode (i.e., flow on a spherical volume) $\widehat{\boldsymbol{r}} = \frac{\boldsymbol{x}}{||\boldsymbol{x}||_2}$, 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., $\frac{\partial P_0}{\partial r} = -\rho_0 g \widehat{\boldsymbol{r}}$, we can rewrite Eq. (2.1) as

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

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

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

where

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

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 $\widehat{\boldsymbol{r}}$ direction, which cancels the buoyancy term $\rho_0 g \widehat{\boldsymbol{r}}$ (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 $\widehat{\boldsymbol{r}}$ direction. 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. 3.4.1). 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, 17] and multi-stage schemes [1, 19, 22].

3.1. IMEX Linear Multi-step Methods. As was done in [9, 10] we now consider a generic K-step (multi-step method) discretization of Eq. (3.2) of the form

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

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:

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

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

$$\boldsymbol{q}_{tt} = \widehat{\boldsymbol{q}} + \lambda L(\boldsymbol{q}_{tt}), \tag{3.4b}$$

where $\lambda = \chi \Delta t \delta$. For example, the coefficients for the second-order backward-difference-formula (BDF2) method, assuming constant time-stepping, are $\alpha_0 = 4/3$, $\alpha_1 = -1/3$, $\chi = 2/3$, $\beta_0 = 2$, and $\beta_1 = -1$ (see [12] 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 [6, 17] as (3.4). 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 [6] defined by $\alpha_0 = 1$, $\alpha_1 = 0$, $\chi = 1$, $\beta_0 = 23/12$, $\beta_1 = -16/12$, and $\beta_2 = 5/12$. 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 [21] for a simple equation. We do not use BDFs of higher order than 2 because they are not A-stable (e.g., see [12]); 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 [12].

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., [20, 16]). In fact, in Sec. 3.2 we show that our generalized IMEX formalism also accommodates multi-stage methods.

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, 19, 22]. When applied to such partitioned problems as (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 advective term, whereas an explicit one will be used for the nonstiff part (curly brackets in Eq. (3.2)) that represents the acoustic and gravity wave terms, such that the compound method is of a given order of consistency. Diagonally implicit s-stage ARK methods (or SDIRK) can be represented as

$$Y^{(i)} = y^n + \Delta t \sum_{j=1}^{i-1} a_{ij} f\left(Y^{(j)}\right) + \Delta t \sum_{j=1}^{i} \tilde{a}_{ij} g\left(Y^{(j)}\right), \qquad i = 1, \ldots, s \tag{3.5a}$$

$$y^{n+1} = y^n + \Delta t \sum_{i=1}^{s} b_i f\left(Y^{(i)}\right) + \Delta t \sum_{i=1}^{s} \tilde{b}_i g\left(Y^{(i)}\right), \tag{3.5b}$$

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 $\tilde{A} = \{\tilde{a}_{ij}\}$, $\tilde{b} = \{\tilde{b}_i\}$. 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 abscisas $c_i = \sum_j a_{ij}$ and $\tilde{c}_i = \sum_j \tilde{a}_{ij}$ represent the time when f and g are evaluated, respectively.

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 (3.2), one can formulate an ARK scheme by using a similar formulation to that in (3.3). An s-stage ARK scheme applied to (3.3) has the following form:

$$\begin{aligned} \boldsymbol{Q}^{(i)} =& \boldsymbol{q}^n + \Delta t \sum_{j=1}^{i-1} a_{ij} \left(S(\boldsymbol{Q}^{(j)}) - \delta L(\boldsymbol{Q}^{(j)})\right) + \Delta t \sum_{j=1}^{i-1} \tilde{a}_{ij} \left(\delta L(\boldsymbol{Q}^{(j)})\right) + \\ & + \Delta t\, \tilde{a}_{ii} \left(\delta L(\boldsymbol{Q}^{(i)})\right), \; i = 1, \ldots, s, \end{aligned} \tag{3.6a}$$

$$\boldsymbol{q}^{n+1} = \boldsymbol{q}^n + \Delta t \sum_{i=1}^{s} b_i S(\boldsymbol{Q}^{(i)}), \tag{3.6b}$$

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.

It follows that for each stage $i = 1, \ldots, s$ and $\tilde{a}_{ii} \neq 0$ we have

$$\boldsymbol{q}_{tt} = \boldsymbol{Q}^{(i)} + \sum_{j=1}^{i-1} \frac{\tilde{a}_{ij} - a_{ij}}{\tilde{a}_{ii}} \boldsymbol{Q}^{(j)} , \tag{3.7a}$$

$$\widehat{\boldsymbol{q}} = \boldsymbol{q}^E + \sum_{j=1}^{i-1} \frac{\tilde{a}_{ij} - a_{ij}}{\tilde{a}_{ii}} \boldsymbol{Q}^{(j)} , \tag{3.7b}$$

$$\boldsymbol{q}^E = \boldsymbol{q}^n + \Delta t \sum_{j=1}^{i-1} a_{ij} S(\boldsymbol{Q}^{(j)}) . \tag{3.7c}$$

Then the following linear system is solved (similar to (3.4b)):

$$\boldsymbol{q}_{tt} = \widehat{\boldsymbol{q}} + \Delta t \, \tilde{a}_{ii} \delta L \left(\boldsymbol{q}_{tt} \right) . \tag{3.7d}$$

The stage value is obtained from (3.7a):

$$\boldsymbol{Q}^{(i)} = \boldsymbol{q}_{tt} - \sum_{j=1}^{i-1} \frac{\tilde{a}_{ij} - a_{ij}}{\tilde{a}_{ii}} \boldsymbol{Q}^{(j)} . \tag{3.7e}$$

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

$$\boldsymbol{Q}^{(i)} = \boldsymbol{q}^n + \Delta t \sum_{j=1}^{i-1} a_{ij} S(\boldsymbol{Q}^{(j)}) + \Delta t L \sum_{j=1}^{i-1} (a_{ij} - \tilde{a}_{ij}) \boldsymbol{Q}^{(j)} . \tag{3.8}$$

The solution at the next step is obtained from (3.6b).

In this study we develop a new second-order ARK method and further consider the ARK schemes of orders 3 and 4 developed by Kennedy and Carpenter [19]. All ARK schemes are singly diagonal, first-stage explicit ($\tilde{a}_{ii} = \tilde{a}_{jj}$ if $\tilde{a}_{ii} \neq 0 \wedge \tilde{a}_{jj} \neq 0$). 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 (3.7d) 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]:

$$\begin{array}{c|ccc} 0 & 0 & & \\ 2 \pm \sqrt{2} & 2 \pm \sqrt{2} & 0 & \\ 1 & 1 - a_{32} & a_{32} & 0 \\ \hline & \pm \frac{1}{2\sqrt{2}} & \pm \frac{1}{2\sqrt{2}} & 1 \pm \frac{1}{\sqrt{2}} \end{array} \qquad \begin{array}{c|ccc} 0 & 0 & & \\ 2 \pm \sqrt{2} & 1 \pm \frac{1}{\sqrt{2}} & 1 \pm \frac{1}{\sqrt{2}} & \\ 1 & \pm \frac{1}{2\sqrt{2}} & \pm \frac{1}{2\sqrt{2}} & 1 \pm \frac{1}{\sqrt{2}} \\ \hline & \pm \frac{1}{2\sqrt{2}} & \pm \frac{1}{2\sqrt{2}} & 1 \pm \frac{1}{\sqrt{2}} \end{array} . \tag{3.9}$$

The family of schemes defined by (3.9) results in two choices for the implicit part driven by the diagonal element ($1 \pm \frac{1}{\sqrt{2}}$). We choose $1 - \frac{1}{\sqrt{2}}$ because this yields a monotonic abscissa. The only free parameter 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 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 [19]. 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 all boundary conditions within the IMEX formulation. For the no-flux boundary conditions, we apply the condition $\widehat{\boldsymbol{n}}_\Gamma \cdot \boldsymbol{u} = 0$, where $\widehat{\boldsymbol{n}}_\Gamma$ is the outward pointing unit normal vector of the boundary Γ. Since $\boldsymbol{u}$ and $\widehat{\boldsymbol{n}}_\Gamma$ both live in R^3, we can define an augmented normal vector

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

that then allows us to satisfy no-flux boundary conditions as follows: $\widehat{\boldsymbol{n}}_\Gamma \cdot \boldsymbol{q} = 0$. We will use $\widehat{\boldsymbol{n}}_\Gamma$ 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 $\widehat{\boldsymbol{n}}_\Gamma$. 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:

$$\frac{\partial \boldsymbol{q}}{\partial t} = S(\boldsymbol{q}) + \tau_{nf} \widehat{\boldsymbol{n}}_\Gamma + \tau_{nr}(\boldsymbol{q} - \boldsymbol{q}_b) \tag{3.10}$$

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 $\boldsymbol{q}$.

To impose the non-reflecting boundary conditions (NRBC), 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

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

Taking the scalar product of this equation with $\widehat{\boldsymbol{n}}_\Gamma$ 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

$$\boldsymbol{\mathcal{P}} = \begin{cases} \boldsymbol{I} - \widehat{\boldsymbol{n}}_\Gamma \widehat{\boldsymbol{n}}_\Gamma^{\mathcal{T}} & \text{in} \quad \Gamma \\ \boldsymbol{I} & \text{in} \quad \Omega - \Gamma \end{cases} \tag{3.11}$$

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

3.4. 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.

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

$$L(\boldsymbol{q}) = -\begin{pmatrix} \boldsymbol{u}\cdot\boldsymbol{\nabla}\rho_0 + \rho_0\boldsymbol{\nabla}\cdot\boldsymbol{u} \\ \frac{1}{\rho_0}\boldsymbol{\nabla}P' + g\frac{\rho'}{\rho_0}\widehat{\boldsymbol{r}} \\ \boldsymbol{u}\cdot\boldsymbol{\nabla}\theta_0 \end{pmatrix}, \tag{3.12}$$

with the (linearized) pressure defined as

$$P' = \frac{\gamma P_0}{\rho_0}\rho' + \frac{\gamma P_0}{\theta_0}\theta'. \tag{3.13}$$

Applying the IMEX method yields

$$\rho_{tt} = (\alpha\widehat{\rho} + \beta\widehat{\rho}_b) - \alpha\lambda\left(\boldsymbol{u}_{tt}\cdot\boldsymbol{\nabla}\rho_0 + \rho_0\boldsymbol{\nabla}\cdot\boldsymbol{u}_{tt}\right) \tag{3.14a}$$

$$\boldsymbol{u}_{tt} = (\alpha\widehat{\boldsymbol{u}} + \beta\widehat{\boldsymbol{u}}_b) - \frac{\alpha\lambda}{\rho_0}\left(\boldsymbol{\nabla}P_{tt} + g\rho_{tt}\widehat{\boldsymbol{r}}\right) \tag{3.14b}$$

$$\theta_{tt} = \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right) - \alpha\lambda\left(\boldsymbol{u}_{tt}\cdot\boldsymbol{\nabla}\theta_0\right) \tag{3.14c}$$

$$P_{tt} = G_0\rho_{tt} + H_0\theta_{tt}, \tag{3.14d}$$

where

$$G_0 = \frac{\gamma P_0}{\rho_0} \tag{3.15a}$$

$$H_0 = \frac{\gamma P_0}{\theta_0}; \tag{3.15b}$$

the system represented by Eqs. (3.14a)-(3.14d) is the *No Schur* IMEX form.

3.4.2. Schur Form. Substituting Eq. (3.14c) into Eq. (3.14d) yields

$$\rho_{tt} = \frac{1}{G_0}\left\{P_{tt} - H_0\left[\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right) - \alpha\lambda\left(\boldsymbol{u}_{tt}\cdot\boldsymbol{\nabla}\theta_0\right)\right]\right\}. \tag{3.16}$$

We can now substitute Eq. (3.16) into Eq. (3.14b) in order to express the momentum as a function of pressure only. Upon applying this substitution, we get

$$\boldsymbol{u}_{tt} = \boldsymbol{\mathcal{P}}_C\left[(\alpha\widehat{\boldsymbol{u}} + \beta\widehat{\boldsymbol{u}}_b) + \alpha\lambda\frac{gH_0}{\rho_0 G_0}\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right)\widehat{\boldsymbol{r}} - \frac{\alpha\lambda}{\rho_0}\left(\boldsymbol{\nabla}P_{tt} + \frac{g}{G_0}P_{tt}\widehat{\boldsymbol{r}}\right)\right] \tag{3.17}$$

where no-flux boundary conditions are enforced through the application of the orthogonal projector $\boldsymbol{\mathcal{P}}$ given in Eq. (3.11), $\boldsymbol{\mathcal{P}}_C = \boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{C}}$, $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{A}}^{-1}$, where the matrix $\boldsymbol{\mathcal{A}}$ is obtained by isolating the momentum equation in terms of its variables and is defined as

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

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

$$c = (\alpha\lambda)^2\frac{g}{\theta_0}. \tag{3.18}$$

Substituting Eqs. (3.14a) and (3.14c) into Eq. (3.14d) yields

$$P_{tt} = G_0\left(\alpha\widehat{\rho} + \beta\widehat{\rho}_b\right) + H_0\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right) - \alpha\lambda\boldsymbol{F}_0\cdot\boldsymbol{u}_{tt} - \alpha\lambda\rho_0 G_0\boldsymbol{\nabla}\cdot\boldsymbol{u}_{tt}, \tag{3.19}$$

where $\boldsymbol{F}_0 = G_0 \boldsymbol{\nabla} \rho_0 + H_0 \boldsymbol{\nabla} \theta_0$. The last step is to substitute Eq. (3.17) into Eq. (3.19), which yields the *Schur* form

$$
\begin{aligned}
P_{tt} &- (\alpha\lambda)^2 \boldsymbol{F}_0 \cdot \left[\boldsymbol{\mathcal{P}}_C \left(\frac{1}{\rho_0} \boldsymbol{\nabla} P_{tt} + \frac{g}{\rho_0 G_0} P_{tt} \widehat{\boldsymbol{r}} \right) \right] \\
&- (\alpha\lambda)^2 G_0 \rho_0 \boldsymbol{\nabla} \cdot \left[\boldsymbol{\mathcal{P}}_C \left(\frac{1}{\rho_0} \boldsymbol{\nabla} P_{tt} + \frac{g}{\rho_0 G_0} P_{tt} \widehat{\boldsymbol{r}} \right) \right] \\
&= G_0 \left(\alpha \widehat{\rho} + \beta \widehat{\rho}_b \right) + H_0 \left(\alpha \widehat{\theta} + \beta \widehat{\theta}_b \right) \\
&- \alpha\lambda \boldsymbol{F}_0 \cdot \left[\boldsymbol{\mathcal{P}}_C \left((\alpha \widehat{\boldsymbol{u}} + \beta \widehat{\boldsymbol{u}}_b) + \alpha\lambda \frac{g H_0}{\rho_0 G_0} \left(\alpha \widehat{\theta} + \beta \widehat{\theta}_b \right) \widehat{\boldsymbol{r}} \right) \right] \\
&- \alpha\lambda G_0 \rho_0 \boldsymbol{\nabla} \cdot \left[\boldsymbol{\mathcal{P}}_C \left((\alpha \widehat{\boldsymbol{u}} + \beta \widehat{\boldsymbol{u}}_b) + \alpha\lambda \frac{g H_0}{\rho_0 G_0} \left(\alpha \widehat{\theta} + \beta \widehat{\theta}_b \right) \widehat{\boldsymbol{r}} \right) \right]. \qquad (3.20)
\end{aligned}
$$

3.4.3. Schur Form in Cloud-Resolving/Mesoscale Mode. For cloud-resolving/mesoscale mode (i.e., flow in a box) the following simplifications occur: $\widehat{\boldsymbol{r}} = \widehat{\boldsymbol{k}}$ 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 [13]. Equation (3.20) simplifies to

$$
\begin{aligned}
P_{tt} &- (\alpha\lambda)^2 F_0 \widehat{\boldsymbol{k}} \cdot \left[\boldsymbol{\mathcal{P}}_C \left(\frac{1}{\rho_0} \boldsymbol{\nabla} P_{tt} + \frac{g}{\rho_0 G_0} P_{tt} \widehat{\boldsymbol{k}} \right) \right] \\
&- (\alpha\lambda)^2 G_0 \rho_0 \boldsymbol{\nabla} \cdot \left[\boldsymbol{\mathcal{P}}_C \left(\frac{1}{\rho_0} \boldsymbol{\nabla} P_{tt} + \frac{g}{\rho_0 G_0} P_{tt} \widehat{\boldsymbol{k}} \right) \right] \\
&= G_0 \left(\alpha \widehat{\rho} + \beta \widehat{\rho}_b \right) + H_0 \left(\alpha \widehat{\theta} + \beta \widehat{\theta}_b \right) \\
&- \alpha\lambda F_0 \widehat{\boldsymbol{k}} \cdot \left[\boldsymbol{\mathcal{P}}_C \left((\alpha \widehat{\boldsymbol{u}} + \beta \widehat{\boldsymbol{u}}_b) + \alpha\lambda \frac{g H_0}{\rho_0 G_0} \left(\alpha \widehat{\theta} + \beta \widehat{\theta}_b \right) \widehat{\boldsymbol{k}} \right) \right] \\
&- \alpha\lambda G_0 \rho_0 \boldsymbol{\nabla} \cdot \left[\boldsymbol{\mathcal{P}}_C \left((\alpha \widehat{\boldsymbol{u}} + \beta \widehat{\boldsymbol{u}}_b) + \alpha\lambda \frac{g H_0}{\rho_0 G_0} \left(\alpha \widehat{\theta} + \beta \widehat{\theta}_b \right) \widehat{\boldsymbol{k}} \right) \right] \qquad (3.21)
\end{aligned}
$$

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

3.5. IMEX in One Direction. The IMEX method defined in all spatial dimensions as described in Sec. 3.4 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 a certain class of problems (say, when the domain to be solved has different scales in the vertical versus the two horizontal directions) it may be advantageous to employ an IMEX method in only one of the directions. This is the case in global atmospheric modeling where the vertical direction is less than 40 kilometers 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 entirely on-processor and also completely 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. Note that to make this solution strategy work requires using a 2D domain decomposition whereby the vertical direction is entirely on-processor. This results in an embarrasingly 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 $(\widehat{\boldsymbol{s}}, \widehat{\boldsymbol{t}}, \widehat{\boldsymbol{r}})^{\mathcal{T}}$, which we define below. The first step is to map the velocity field $\boldsymbol{u} = (u, v, w)^{\mathcal{T}}$ as follows:

$$\boldsymbol{u}_R = \boldsymbol{\mathcal{R}}\boldsymbol{u} \tag{3.22}$$

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

$$\boldsymbol{\mathcal{R}} = \begin{pmatrix} \widehat{\boldsymbol{s}} & \widehat{\boldsymbol{t}} & \widehat{\boldsymbol{r}} \end{pmatrix}^{\mathcal{T}} \tag{3.23}$$

is the map, and $\widehat{\boldsymbol{r}} = \frac{\boldsymbol{x}}{||\boldsymbol{x}||} = (r_x, r_y, r_z)^{\mathcal{T}}$, $\widehat{\boldsymbol{s}} = \boldsymbol{\mathcal{Q}}_v \widehat{\boldsymbol{r}} \times \widehat{\boldsymbol{v}}$, and $\widehat{\boldsymbol{t}} = \widehat{\boldsymbol{r}} \times \widehat{\boldsymbol{s}}$ are normalized vectors. The vector $\widehat{\boldsymbol{s}}$ is guaranteed to be orthogonal to $\widehat{\boldsymbol{r}}$ by virtue of the projection $\boldsymbol{\mathcal{Q}}_v \in R^{3\times 3}$ and then taking the vector product with $\widehat{\boldsymbol{v}}$. The vector $\widehat{\boldsymbol{v}} \in R^3$ is chosen to be along the $\widehat{\boldsymbol{i}}$, $\widehat{\boldsymbol{j}}$, or $\widehat{\boldsymbol{k}}$ directions depending on which component of $\widehat{\boldsymbol{r}}$ is a minimum; that is, $\widehat{\boldsymbol{v}} = \widehat{\boldsymbol{i}}$ if $|r_x| = min(|r_x|, |r_y|, |r_z|)$, and so on. This is done to avoid aligning the vector $\widehat{\boldsymbol{v}}$ with the null space of $\widehat{\boldsymbol{r}}$. 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 $\widehat{\boldsymbol{v}} = \widehat{\boldsymbol{i}}$, $\widehat{\boldsymbol{j}}$, $\widehat{\boldsymbol{k}}$, respectively. The matrix $\boldsymbol{\mathcal{Q}}_v$ is constructed in order to project $\widehat{\boldsymbol{r}}$ along a subspace of R^3 in a direction orthogonal to $\widehat{\boldsymbol{v}}$. This approach guarantees that $\widehat{\boldsymbol{s}}$ and $\widehat{\boldsymbol{t}}$ form a tangent plane passing through the radial vector $\widehat{\boldsymbol{r}}$; 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.

3.5.1. No Schur Form. Upon applying the rotation transformation given in Eq. (3.23), we obtain the rotated variables

$$\boldsymbol{q}_R = \begin{pmatrix} \rho' \\ u^{(s)} \\ u^{(t)} \\ u^{(r)} \\ \theta' \end{pmatrix}. \tag{3.24}$$

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

$$L(\boldsymbol{q}) = - \begin{pmatrix} u^{(r)} \frac{d\rho_0}{dr} + \rho_0 \frac{\partial u^{(r)}}{\partial r} \\ 0 \\ 0 \\ \frac{1}{\rho_0} \frac{\partial P'}{\partial r} + g \frac{\rho'}{\rho_0} \\ u^{(r)} \frac{d\theta_0}{dr} \end{pmatrix} \tag{3.25}$$

with the pressure defined as in Eq. (3.13). Applying the IMEX method yields

$$\rho_{tt} = (\alpha\widehat{\rho} + \beta\widehat{\rho}_b) - \alpha\lambda\left(u_{tt}^{(r)}\frac{d\rho_0}{dr} + \rho_0\frac{\partial u_{tt}^{(r)}}{\partial r}\right) \tag{3.26a}$$

$$u_{tt}^{(s)} = \left(\alpha\widehat{u}^{(s)} + \beta\widehat{u}_b^{(s)}\right) \tag{3.26b}$$

$$u_{tt}^{(t)} = \left(\alpha\widehat{u}^{(t)} + \beta\widehat{u}_b^{(t)}\right) \tag{3.26c}$$

$$u_{tt}^{(r)} = \left(\alpha\widehat{u}^{(r)} + \beta\widehat{u}_b^{(r)}\right) - \frac{\alpha\lambda}{\rho_0}\left(\frac{\partial P_{tt}}{\partial r} + g\rho_{tt}\right) \tag{3.26d}$$

$$\theta_{tt} = \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right) - \alpha\lambda\left(u_{tt}^{(r)}\frac{d\theta_0}{dr}\right) \tag{3.26e}$$

$$P_{tt} = G_0\rho_{tt} + H_0\theta_{tt}, \tag{3.26f}$$

where G_0 and and H_0 are defined in Eq. (3.15); the system represented by Eqs. (3.26a)-(3.26f) is the *No Schur* IMEX form.

3.5.2. Schur Form. Substituting Eq. (3.26e) into Eq. (3.26f) yields

$$\rho_{tt} = \frac{1}{G_0}\left\{P_{tt} - H_0\left[\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right) - \alpha\lambda\left(u_{tt}^{(r)}\frac{d\theta_0}{dr}\right)\right]\right\}. \tag{3.27}$$

We can now substitute Eq. (3.27) into Eq. (3.26d) in order to express the momentum as a function of pressure only. Upon applying this substitution, we get

$$\boldsymbol{u}_{tt}^R = \boldsymbol{\mathcal{P}}_C^R\left[\left(\alpha\widehat{\boldsymbol{u}}^R + \beta\widehat{\boldsymbol{u}}_b^R\right) + \alpha\lambda\frac{gH_0}{\rho_0 G_0}\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right)\widehat{\boldsymbol{r}}_R - \frac{\alpha\lambda}{\rho_0}\left(\frac{\partial P_{tt}}{\partial r} + \frac{g}{G_0}P_{tt}\right)\widehat{\boldsymbol{r}}_R\right] \tag{3.28}$$

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 $\widehat{\boldsymbol{r}}_R = \widehat{\boldsymbol{r}}$ because the implicit correction should only act along the direction $\widehat{\boldsymbol{r}}$.

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

$$c = 1 + (\alpha\lambda)^2\frac{g}{\theta_0}\frac{d\theta_0}{dr} \tag{3.29}$$

and

$$\boldsymbol{\mathcal{P}}^R = \boldsymbol{I} - \widehat{\boldsymbol{n}}_R\widehat{\boldsymbol{n}}_R^{\mathcal{T}}, \tag{3.30}$$

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

Substituting Eqs. (3.26a) and (3.26e) into Eq. (3.26f) yields

$$P_{tt} = G_0\left(\alpha\widehat{\rho} + \beta\widehat{\rho}_b\right) + H_0\left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b\right) - \alpha\lambda F_0 u_{tt}^{(r)} - \alpha\lambda\rho_0 G_0\frac{\partial u_{tt}^{(r)}}{\partial r}, \tag{3.31}$$

where $F_0 = G_0\frac{d\rho_0}{dr} + H_0\frac{d\theta_0}{dr}$. The last step is to substitute Eq. (3.28) into Eq. (3.31), which yields

the *Schur* form

$$
\begin{aligned}
P_{tt} &- (\alpha\lambda)^2 F_0 \widehat{\boldsymbol{r}}_R \cdot \left[\boldsymbol{\mathcal{P}}_C^R \left(\frac{1}{\rho_0} \frac{\partial P_{tt}}{\partial r} + \frac{g}{\rho_0 G_0} P_{tt} \right) \widehat{\boldsymbol{r}}_R \right] \\
&- (\alpha\lambda)^2 G_0 \rho_0 \frac{\partial}{\partial r} \left[\widehat{\boldsymbol{r}}_R \cdot \boldsymbol{\mathcal{P}}_C \left(\frac{1}{\rho_0} \frac{\partial P_{tt}}{\partial r} + \frac{g}{\rho_0 G_0} P_{tt} \right) \widehat{\boldsymbol{r}}_R \right] \\
&= G_0 \left(\alpha\widehat{\rho} + \beta\widehat{\rho}_b \right) + H_0 \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b \right) \\
&- \alpha\lambda F_0 \widehat{\boldsymbol{r}}_R \cdot \left[\boldsymbol{\mathcal{P}}_C \left\{ \left(\alpha\widehat{\boldsymbol{u}}^R + \beta\widehat{\boldsymbol{u}}_b^R \right) + \alpha\lambda \frac{g H_0}{\rho_0 G_0} \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b \right) \widehat{\boldsymbol{r}}_R \right\} \right] \\
&- \alpha\lambda G_0 \rho_0 \frac{\partial}{\partial r} \left[\widehat{\boldsymbol{r}}_R \cdot \boldsymbol{\mathcal{P}}_C \left\{ \left(\alpha\widehat{\boldsymbol{u}}^R + \beta\widehat{\boldsymbol{u}}_b^R \right) + \alpha\lambda \frac{g H_0}{\rho_0 G_0} \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b \right) \widehat{\boldsymbol{r}}_R \right\} \right]. \qquad (3.32)
\end{aligned}
$$

3.5.3. Schur Form in Cloud-Resolving/Mesoscale Mode. For the case of cloud-resolving or mesoscale mode (i.e., flow in a box) the simplifications $\widehat{\boldsymbol{r}} = \widehat{\boldsymbol{k}}$ 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 (3.26a)-(3.26f) simplify to

$$
\begin{aligned}
\rho_{tt} &= \left(\alpha\widehat{\rho} + \beta\widehat{\rho}_b \right) - \alpha\lambda \left(w \frac{d\rho_0}{dz} + \rho_0 \frac{\partial w}{\partial z} \right) && (3.33a) \\
u_{tt} &= \left(\alpha\widehat{u} + \beta\widehat{u}_b \right) && (3.33b) \\
v_{tt} &= \left(\alpha\widehat{v} + \beta\widehat{v}_b \right) && (3.33c) \\
w_{tt} &= \left(\alpha\widehat{w} + \beta\widehat{w}_b \right) - \frac{\alpha\lambda}{\rho_0} \left(\frac{\partial P_{tt}}{\partial z} + g\rho_{tt} \right) && (3.33d) \\
\theta_{tt} &= \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b \right) - \alpha\lambda \left(w_{tt} \frac{d\theta_0}{dz} \right), && (3.33e)
\end{aligned}
$$

and Eq. (3.29) 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 $\widehat{\boldsymbol{r}}_R = \widehat{\boldsymbol{k}}$, 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 [13]. All these simplifications result in the new form of Eq. (3.32):

$$
\begin{aligned}
P_{tt} &- (\alpha\lambda)^2 F_0 \widehat{\boldsymbol{k}} \cdot \left[\boldsymbol{\mathcal{P}}_C \left(\frac{1}{\rho_0} \frac{\partial P_{tt}}{\partial z} + \frac{g}{\rho_0 G_0} P_{tt} \right) \widehat{\boldsymbol{k}} \right] \\
&- (\alpha\lambda)^2 G_0 \rho_0 \frac{\partial}{\partial z} \left[\widehat{\boldsymbol{k}} \cdot \boldsymbol{\mathcal{P}}_C \left(\frac{1}{\rho_0} \frac{\partial P_{tt}}{\partial z} + \frac{g}{\rho_0 G_0} P_{tt} \right) \widehat{\boldsymbol{k}} \right] \\
&= G_0 \left(\alpha\widehat{\rho} + \beta\widehat{\rho}_b \right) + H_0 \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b \right) \\
&- \alpha\lambda F_0 \widehat{\boldsymbol{k}} \cdot \left[\boldsymbol{\mathcal{P}}_C \left\{ \left(\alpha\widehat{\boldsymbol{u}} + \beta\widehat{\boldsymbol{u}}_b \right) + \alpha\lambda \frac{g H_0}{\rho_0 G_0} \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b \right) \widehat{\boldsymbol{k}} \right\} \right] \\
&- \alpha\lambda G_0 \rho_0 \frac{\partial}{\partial z} \left[\widehat{\boldsymbol{k}} \cdot \boldsymbol{\mathcal{P}}_C \left\{ \left(\alpha\widehat{\boldsymbol{u}} + \beta\widehat{\boldsymbol{u}}_b \right) + \alpha\lambda \frac{g H_0}{\rho_0 G_0} \left(\alpha\widehat{\theta} + \beta\widehat{\theta}_b \right) \widehat{\boldsymbol{k}} \right\} \right]. \qquad (3.34)
\end{aligned}
$$

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) works best 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 [24] with a small time-step as the analytic solution. We then compute the L^2 norm:

$$L^2 \text{ error} = \sqrt{\sum_{i=1}^{N_{dof}} (\boldsymbol{q}_i^{num} - \boldsymbol{q}_i^{exact})^2},$$

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 $\dim \boldsymbol{q} = N_{dof}$.

The linear system resulting from the 3D IMEX approach is solved using GMRES with a particular class of preconditioner derived specifically for our methods (see [5] for details). 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 (always less than 10 for these cases and choices of time-step size). 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., [14, 7, 9] while the second term represents the number of points along a radial component. We note that currently NUMA only admits hexahedral elements.

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) [26]. 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 $\boldsymbol{x}_c$ where $\boldsymbol{x}_c = (500, 500, 260)$, $R_c = 250$ meters, and $\theta_c = \frac{1}{1-\frac{6}{\pi^2}}$ is a constant. The domain for this problem is $(x, y, z) \in [0, 1000]^3$ meters. Note that cloud-resolving simulations are usually carried out with grid resolutions less than 1000 meters. Since for this test case we use grid resolutions of 10 to 20 meters, we refer to it as cloud-resolving. This test case does not have an analytic solution, but the proper behavior of the wave dynamics is well understood and can be used to verify the model.

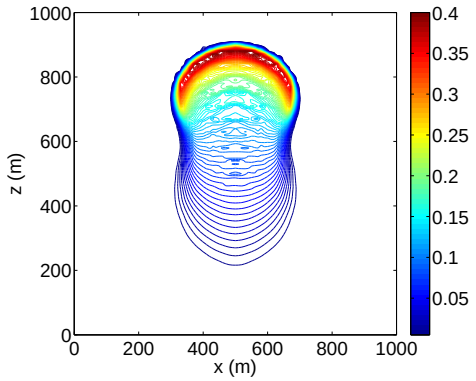
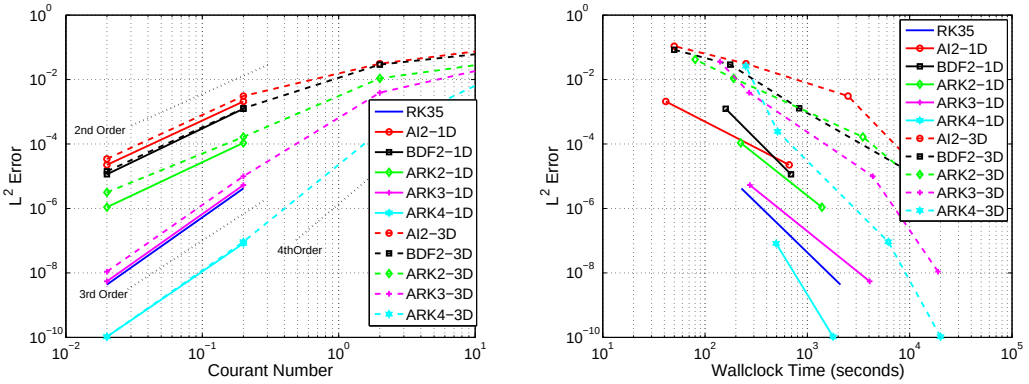


FIGURE 4.1. *Cloud-Resolving Mode: Rising Thermal Bubble. A slice of the potential temperature perturbation (at y=500 meters) after 400 seconds for 24^3 elements with 4th order polynomials. The contour lines are from 0.005 to 0.5 with an interval of 0.005.*

Figure 4.1 shows the potential temperature pertubation after 400 seconds for a grid resolution of 24^3 elements each with 4th-order polynomials (which yields a grid resolution of 10.3 meters and 912,673 gridpoints). Note that the initial condition is a cosine bubble (in three-dimensions) that, after 400 seconds, 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 meters and 68,921 gridpoints; 10 MPI (Message-Passing Interface) processors 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. The simulations are run for 10 seconds, where the L^2 norm is computed using the explicit (RK35) solution with a Courant number of 0.0001 seconds.



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 10 seconds.*

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. This is the main result that can be extracted from Fig. 4.2a; note that constructing a 3D-IMEX method with the expected convergence rate is trivial, but this is not the case for the 1D-IMEX

method because its derivation is much 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 10^0 and 10^{-3} the 3D-IMEX methods are the most efficient. For accuracy levels between 10^{-3} and 10^{-5}, the 1D-IMEX methods are the most efficient, with AI2 and ARK2 being the fastest. For accuracy levels between 10^{-5} and 10^{-7} the explicit RK35 is the fastest with ARK3 1D-IMEX being close. For accuracy levels below 10^{-7} the ARK4 1D-IMEX is the most efficient; this last result is indicative of the fact that at high-order accuracy levels, high-order methods will win.

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

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

For this simulation, $R_{GR} = 1$, which means that the only way to increase the maximum time-step is to use an IMEX method in all three dimensions. 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. 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 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 and faster wave speed components. To illustrate this effect, we consider the simple one-dimensional wave equation

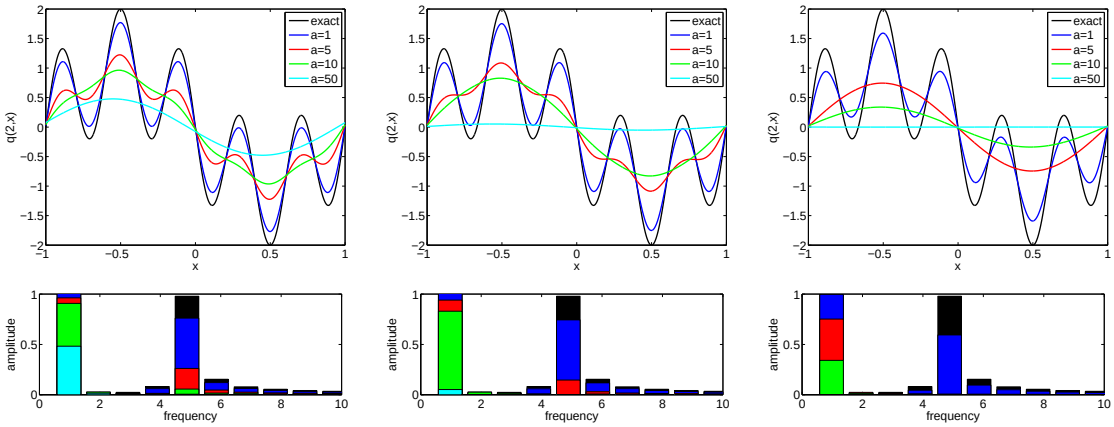
$$\frac{\partial q}{\partial t} + a\frac{\partial q}{\partial x} = 0\,, \quad q(0,x) = \sum_{i=1}^{2} \sin(2\pi\omega_i(x+1))\,,\ \omega = [1,5]\,,\ x \in [-1,1]\,, \tag{4.1}$$

where ω is the frequency component and a the wave speed. The exact solution is the same as the initial condition with a phase shift, and in particular $q(2aT,x) = q(0,x)$, $T = 1, 2, 3, \ldots$. We discretize this equation using the unconditionally stable first-order upwind in space and backward Euler in time. In this setting we isolate the implicit scheme in order 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 after one step $q_{n+1} = r(\xi,k)q_n$ where

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

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 $|r(\xi,k)|^N$. In Fig. 4.3 we illustrate the solution and its spectrum after two seconds (one period for $a = 1$) with different wave speeds and time steps. The initial and shifted spectrum of the exact solution indicate a large contribution at 1 Hz and 5 Hz that correspond to ω in (4.1). As expected, a quick inspection of (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, but not for $a = 10, 50$. 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 10^{-5} to 10^{-3}, 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 $\Delta t = 10^{-3}$. 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.



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.*

4.2. Mesoscale Mode: 3D Density Current. This test case is similar in many respects to the cloud-resolving simulation in that it also uses a hydrostatically balanced reference state with a thermally neutral atmosphere comprised of a reference potential temperature of $\theta_0 = 300$ Kelvin (K). The initial conditions are augmented by the following perturbation:

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

where

$$R = \left(\frac{x - xc}{r_x}\right)^2 + \left(\frac{y - yc}{r_y}\right)^2 + \left(\frac{z - zc}{r_z}\right)^2$$

is the ellipsoidal radius, where $(x_c, y_c, z_c) = (0, 12800, 3000)$ meters,

$$(r_x, r_y, r_z) = (8763.5, 8763.5, 2000)$$

meters, and $\theta_c = -15$, which defines a cold bubble instead of a warm one as in the previous test. The domain for this problem is much larger and has dimensions $(x, y, z) \in ([0, 25600], [0, 25600], [0, 6400])$ meters. Note that typical mesoscale simulations use resolutions varying from 100 meters to 10 kilometers. A physical viscosity of $\mu = 75$ meters^2/seconds is used. In other words, the operators $\mu\boldsymbol{\nabla}^2\boldsymbol{u}$ and $\mu\boldsymbol{\nabla}^2\theta'$ are added to the right-hand sides of Eqs. (2.2b) and (2.2c), respectively.

This test case is not typically found in the literature, but we use it nonetheless for the following reason. It is similar to the classical density current test case of Straka et al. [25] except that an

ellipsoidal (3D) initial condition is used instead of the usual elliptical (2D) initial condition. The radius of the bubble is computed to give the same initial volume for the perturbation as in the Straka et al. ellipse when it is extended fully in the y-direction (i.e., if the original Straka test case is run using the 3D model, the initial condition is the original one used by Straka et al. except that it is an elliptical cylinder since it extends in the y-direction with no variation). Therefore, since this test case is similar to the Straka et al. 2D test case, the dynamics of the problem is well understood (it produces Kelvin-Helmholtz instabilities). Note that since we use a symmetric initial condition (about $y = 12800$), the solution should remain symmetric throughout the entire simulation.

Figure 4.4 shows a slice of the potential temperature perturbation (at $x = 0$) at 200 seconds into the simulation. The cold bubble has begun to drop because of the colder air at the core, and it begins to shear because of the difference in temperature between the core and outer radius of the perturbation. The y-z slice shows that the bubble is perfectly symmetric with respect to the center of the y-direction, and this symmetry persists for the entire simulation.

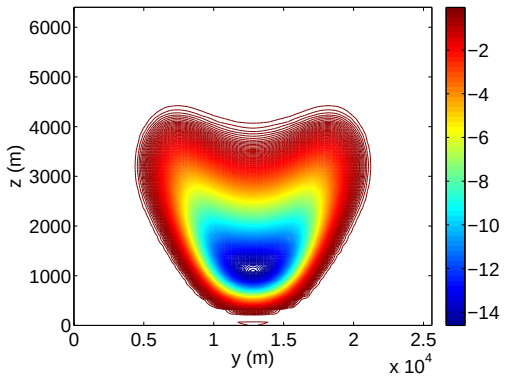
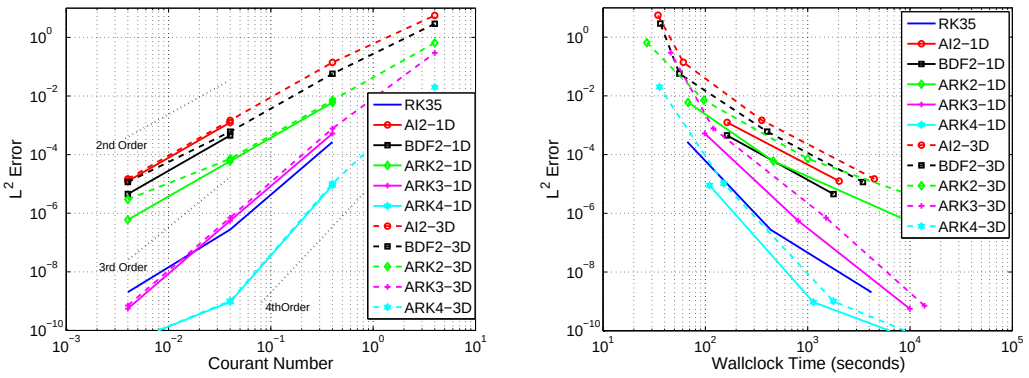


FIGURE 4.4. *Mesoscale Mode: Density Current. A slice of the potential temperature perturbation (at x=0 meters) after 200 seconds for* $24 \times 24 \times 6$ *elements with 8th-order polynomials. The contour lines are from -14 to -0.05 with an interval of 0.05.*

Figure 4.5 shows the accuracy (panel a) and efficiency (panel b) for the various time-integrators used in this simulation. For these simulations the grid is comprised of $16 \times 16 \times 4$ elements of 8th-order that results in a grid resolution of 200 meters with 549,153 gridpoints. The simulations are run for 10 seconds, where the L^2 norm is computed using the explicit (RK35) solution with a time-step of $\Delta t = 0.0001$ seconds. For the efficiency study, 16 MPI processors are used for all the simulations.



a) b)

FIGURE 4.5. *Mesoscale Mode: Density Current. 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 10 seconds.*

Figure 4.5a shows that all the time-integrators, once again, yield the theoretically expected convergence rates; this is yet another test confirming that the 1D-IMEX time-integrators are functioning properly. Figure 4.5b shows the error versus wallclock time. For achieving accuracy levels between 10^0 and 10^{-4}, the most efficient time-integrators are the 3D-IMEX methods, in particular ARK2 and ARK4 perform particularly well. For achieving accuracy levels below 10^{-4}, the most efficient time-integrators are the explicit RK35 and the 4th-order methods (ARK4 1D and 3D). Note that for this simulation, the ratio of horizontal to vertical grid resolution is $R_{GR} = 1$, which again means that the 1D IMEX methods will not be competitive with the 3D IMEX methods in terms of pure efficiency.

4.3. Global Mode: Planetary Acoustic Wave. The global scale problem we consider is that of an acoustic wave traveling around the entire planet [27]. We begin with a hydrostatically balanced initial state with a pressure perturbation. This initial pressure perturbation causes an acoustic wave to travel to the antipode, coalesce, and return to the initial position. The exact solution of this test case is simple in that the (linear) acoustic theory allows one to compute the analytic speed of sound based on the thermodynamics variables. In fact, for the background temperature field of $T_0 = 300$ Kelvin, the resulting speed of sound ($a = \sqrt{\gamma R_d T_0}$) is 347 meters/second.

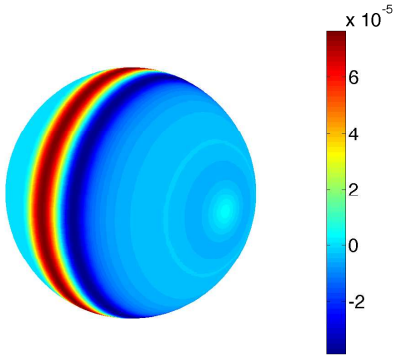


FIGURE 4.6. *Global Mode: Acoustic Wave Propagation. The density perturbation after 8 hours for* 384 *elements in the horizontal (spherical surface) and* 8 *elements in the vertical with 8th-order polynomials. The plot shown is at the first spherical level (i.e., surface of the planet); however the flow is fully three-dimensional (formed by concentric spheres).*

The initial condition is comprised of the pressure perturbation

$$p' = \Delta P f(\lambda, \phi) g(r),$$

where $\Delta P = 100$ Pascals,

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

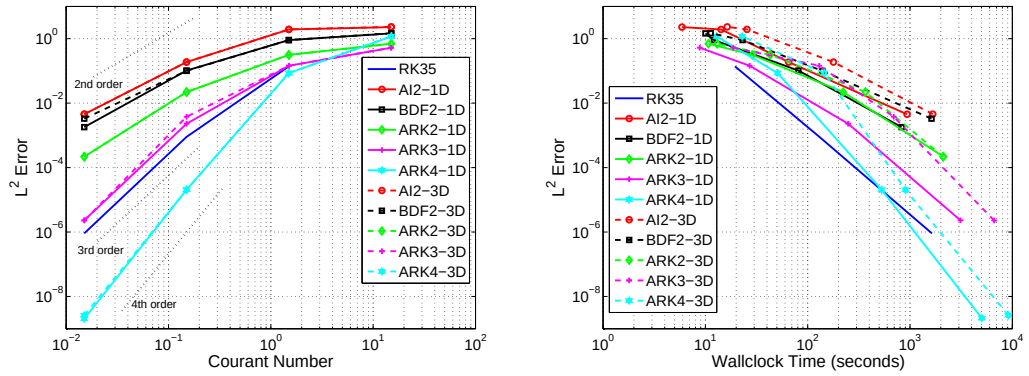
and

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

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 $n_v = 1$. The domain for this problem is comprised of the surface of the Earth with a radius of $R_E = 6371$ kilometers and a model altitude of $r_T = 10$ kilometers.

For this case we use a model grid resolution of $H_{GR} = 208$ kilometers by $V_{GR} = 300$ meters yielding a grid resolution ratio of $R_{GR} = 693$. Although the horizontal resolution used for this numerical experiment falls well within the hydrostatic scale (> 10 kilometers), the vertical resolution

is in the nonhydrostatic regime. The presence of these "multi-scales" makes this test case a challenging problem and representative of the applications that must be properly modeled in large-scale atmospheric dynamics applications of the future (e.g., nonhydrostatic weather prediction).



a) b)

FIGURE 4.7. *Global Mode: Acoustic Wave Propagation. 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 seconds.*

Figure 4.7 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 $6 * 8^3$ elements each with 4th-order polynomials for a total of 202,818 gridpoints. The model is integrated for 100 seconds, where the explicit RK35 solution with a time-step of $\Delta t = 0.001$ is used as the exact solution. For all the simulations, 12 MPI processors are used. Figure 4.7a 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 is an important result because it 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. 4.7b shows that the most efficient time-integrators for accuracy levels between 10^0 to 10^{-1} are the 1D IMEX methods. For accuracy levels between 10^{-1} to 10^{-5} the explicit RK35 method is the most efficient. For accuracy levels below 10^{-5} the ARK4 1D IMEX is the most efficient with the ARK4 3D IMEX surpassing the RK35 method beyond levels of 10^{-6}. These results show that the 1D IMEX methods are the clear winners over the 3D IMEX methods for grid resolution regimes $R_{GR} >> 1$. However, these results also show the value of high-order time-accuracy because the dominant methods for the largest range of levels of accuracy are the ARK3, RK35, and ARK4 methods.

4.4. Conservation. The last comparison we show concerns the conservation properties of the time-integrators. We chose the planetary acoustic wave to highlight the conservation measures because this case has more data points to compare the 1D and 3D IMEX methods; the reason is that 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 because of the spherical geometry.

For this comparison we define the mass loss as

$$\text{Mass Loss} = \text{Mass}(t) - \text{Mass}(0)$$

and the energy loss in a similar fashion

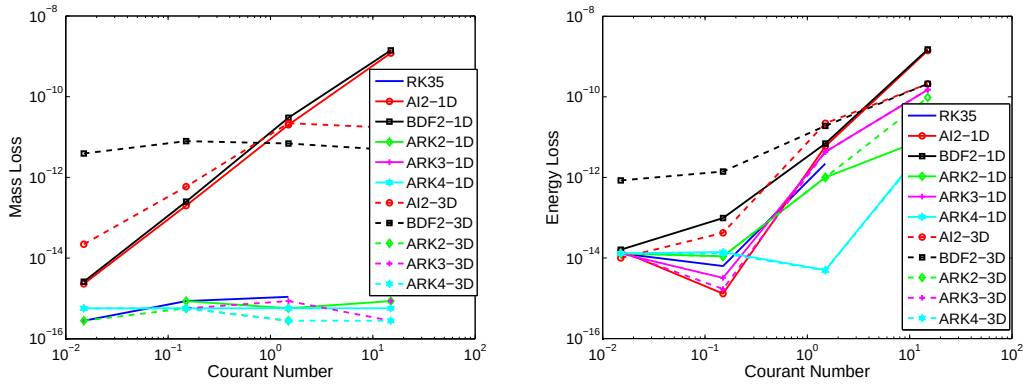
$$\text{Energy Loss} = \text{Energy}(t) - \text{Energy}(0),$$

where Mass(t) and Energy(t) is the mass/energy at time "t", where we compare the difference between the inital 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 4.8. *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 a final time of 100 seconds.*

The mass should be conserved exactly, but the energy can only be conserved up to the time-truncation error ($O(\Delta t^k)$, where k is the order of the time-integrator). Figure 4.8a shows that all the time-integrators conserve mass up to machine precision except for the BDF2 and AI2 methods. To explain this behavior, let us take a closer look at the ARK and linear multistep calculations. Consider a weight vector $\mathbf{e}$ such that $\mathbf{e}^\mathcal{T} S(\boldsymbol{q}) = 0$ so that $\mathbf{e}^\mathcal{T} \frac{\partial \boldsymbol{q}}{\partial t} = \text{const}$. Then, following (3.5b), we obtain

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

Therefore, the ARK methods behave in this regard like an explicit method and preserve all linear invariants to machine precision. On the other hand, linear multistep methods evolve subject to the linear solve in (3.4b) and therefore depend on the accuracy of the linear solver and the propagation of these errors in time.

Figure 4.8b shows the energy loss. Note that while none of the methods conserve energy (they cannot since energy is the result of a quadratic form and not a prognostic variable in the formulation used) the ARK4 method does allow for better energy conservation. This analysis informs us that in terms of mass conservation not all second-order methods are created equally, since ARK2 conserves mass as it should, whereas BDF2 and AI2 do not. In terms of energy conservation, if one is interested in reducing the leakage of energy (say, for a climate simulation) then it makes sense to use either a small time-step or a method that is designed to preserve all linear invariants.

5. Conclusions. We have derived implicit-explicit (IMEX) formulations for the 3D 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

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). This is an important result because it allows a numerical model to reuse the exact same machinery for every time-integration method used in this work; 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. 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 are 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 are the clear winners. For problems where the stiffness is multi-directional, the 3D IMEX methods perform well. 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 scales close to perfectly for processor counts of order 10^5 (see [18]); we expect that the 1D IMEX methods will perform the same because they have the same communication footprint as an explicit method. 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). For the past few years we have been constructing scalable preconditioners (see [5]) and have made advances, but this work remains incomplete. Upon completing our work on preconditioners we will report the scalability of the 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. FXG and EMC gratefully acknowledge the support of the Air Force Office of Scientific Research through the Computational Mathematics program.

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, C. F. Borges, and F. X. Giraldo, *An element-based spectrally-optimized approximate inverse preconditioner for the euler equations*, SISC, in press (2012).
[6] D. Durran and P. Blossey, *Implicit-explicit multistep methods for fast-wave slow-wave problems*, Monthly Weather Review, 140 (2012), pp. 1307–1325.
[7] 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.
[8] S. Gaberšek, F. X. Giraldo, and J. D. Doyle, *Dry and moist idealized experiments with a two-dimensional spectral element model*, Monthly Weather Review, in press (doi: 10.1175/MWR-D-11-00144.1) (2012).
[9] F. X. Giraldo, *Semi-implicit time-integrators for a scalable spectral element atmospheric model*, Q. J. R. Meteorol. Soc., 131 (2005), pp. 2431–2454.
[10] ———, *Hybrid Eulerian-Lagrangian semi-implicit time-integrators*, Computers & Mathematics with applications, 52 (2006), pp. 1325–1342.
[11] 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.

[12] ———, *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.

[13] 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.

[14] 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.

[15] M. Günther, A. Kværnø, and P. Rentrop, *Multirate partitioned Runge-Kutta methods*, BIT, 41 (2001), pp. 504–514.

[16] E. Hairer, S. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer, 1993.

[17] 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.

[18] J. F. Kelly and F. X. Giraldo, *Continuous and discontinuous galerkin methods for a scalable three-dimensional nonhydrostatic atmosphere model: limited-area mode*, J. Comp. Phys., in review (2011).

[19] C. Kennedy and M. Carpenter, *Additive Runge-Kutta schemes for convection-diffusion-reaction equations*, Appl. Numer. Math., 44 (2003), pp. 139–181.

[20] J. Lambert, *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*, Wiley, 1991.

[21] J. 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.

[22] 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.

[23] S. Skelboe, *Stability properties of backward differentiation multirate formulas*, Applied Numerical Mathematics, 5 (1989), pp. 151–160.

[24] 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.

[25] J. Straka, R. Wilhelmson, L. Wicker, J. Anderson, and K. Droegemeier, *Numerical solutions of a non–linear density current: A benchmark solution and comparisons*, Int. J. Numer. Meth. Fl., 17 (1993), pp. 1–22.

[26] S. J. Thomas, J. P. Hacker, and P. K. Smolarkiewicz, *Spectral preconditioners for nonhydrostatic atmospheric models*, Mon. Wea. Rev., 131 (2003), pp. 2464–2478.

[27] 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.