

A Performance Study of Horizontally Explicit Vertically Implicit (HEVI) Time-Integrators for Non-Hydrostatic Atmospheric Models

Francis X. Giraldo^{a,*}, Felipe A. V. de Bragança Alves^{a,**}, James F. Kelly^b, Soonpil Kang^a, P. Alex Reinecke^c

^a*Department of Applied Mathematics, Naval Postgraduate School, Monterey, CA, U.S.*

^b*Space Science Division, U.S. Naval Research Laboratory, Washington, DC, U.S.*

^c*Marine Meteorology Division, U.S. Naval Research Laboratory, Monterey, CA, U.S.*

Abstract

We conduct a thorough study of different forms of horizontally explicit and vertically implicit (HEVI) time-integration strategies for the compressible Euler equations on spherical domains typical of nonhydrostatic global atmospheric applications. We compare the computational time and complexity of two nonlinear variants (NHEVI-GMRES and NHEVI-LU) and a linear variant (LHEVI). We report on the performance of these three variants for a number of additive Runge-Kutta Methods ranging in order of accuracy from second through fifth, and confirm the expected order of accuracy of the HEVI methods for each time-integrator. To gauge the maximum usable time-step of each HEVI method, we run simulations of a nonhydrostatic baroclinic instability for 100 days and then use this time-step to compare the time-to-solution of each method. The results show that NHEVI-LU is 2x faster than NHEVI-GMRES, and LHEVI is 5x faster than NHEVI-LU, for the idealized cases tested. The baroclinic instability and inertia-gravity wave simulations indicate that the optimal choice of time-integrator is LHEVI with either second or third order schemes, as both schemes yield similar time to solution and relative L2 error at their maximum usable time-steps. In the future, we will report on whether these results hold for more complex problems using, e.g., real atmospheric data and/or a higher model top typical of space weather applications.

Keywords: compressible Euler equations, flux-differencing, HEVI, IMEX, numerical weather prediction, Runge-Kutta method, space weather, spectral element method, time-integration

*Corresponding author

**Now at Instituto de Matemática e Estatística, Universidade de São Paulo, Brazil

Email addresses: fxgirald@nps.edu (Francis X. Giraldo), favbalves@gmail.com (Felipe A. V. de Bragança Alves), james.kelly@nrl.navy.mil (James F. Kelly), soonpil.kang.ks@nps.edu (Soonpil Kang), alex.reinecke@nrlmry.navy.mil (P. Alex Reinecke)

1. Introduction

All problems in the numerical solution of time-dependent partial differential equations (PDEs) require, at the very least, two key ingredients for accurate, stable, and efficient solutions. These ingredients are the selection of the methods for approximating the (i) spatial and (ii) temporal (time) derivatives. The focus of this paper is on time-integration methods specifically designed for geophysical fluid dynamics (GFD) where the stiffness of the PDEs results from a finer grid spacing in the vertical relative to the horizontal dimension. We are interested in solving the nonhydrostatic form of the compressible Euler equations, where explicit time-integration is impractical due to the small time-steps dictated by the acoustic speed and reflected by the Courant-Friedrichs-Lewy condition [9]. For this reason, recent work has focused on nonlinear horizontally explicit and vertically implicit methods (HEVI) where, as the name suggests, the horizontal flow along the surface of the earth is handled explicitly in time, whereas the flow normal to the surface of the earth along the radial (or vertical) direction is handled implicitly. The work by Gardner et al. [12] describes a nonlinear HEVI method that uses a GMRES-based Krylov solver, which we call nonlinear HEVI (or NHEVI-GMRES). Since the publication of that paper, others have contributed to this discussion and we mention in particular the work by Steyer et al. [51] and Vogl et al. [58]. The goal of our current work is to analyze NHEVI and seek improvements to it such that (i) the stability and accuracy of the original method is not degraded and (ii) its efficiency is enhanced. Some researchers have reached the conclusion that NHEVI-GMRES may be too expensive and modifications have been studied but perhaps not published (e.g., work on the U.S. Navy’s NEPTUNE model), while other approaches have already appeared in the literature such as LHEVI (e.g., see [16, 60, 33, 8, 6, 32, 49, 59, 48] for IMEX-LHEVI and [44, 28] for split-explicit LHEVI) but in these works it was not compared to the NHEVI method. Therefore, the motivation for the current work is to quantitatively compare the NHEVI method relative to linear HEVI (LHEVI); where we use the Nonhydrostatic Unified Model of the Atmosphere (NUMA) [16] for the numerical experiments.

NUMA is a research CFD code that uses Element-Based Galerkin (EBG) methods [34, 15] for spatial discretization and a suite of explicit, IMEX, and fully implicit time-integrators. NEPTUNE is primarily a low-altitude NWP model, although we are also developing a high-altitude, whole atmosphere version [3] of both NUMA and NEPTUNE for space weather applications. Early versions of NUMA [16] and NEPTUNE [39] relied primarily on 1D linear IMEX schemes, where each column of fluid was rotated to be aligned with the north pole during the implicit step. In addition, linear IMEX relies on an *a priori* reference state to construct the linear operator. Both of these constraints are problematic for an NWP model. The rotations add additional computational cost to the solver, while pre-defined reference states are typically not available for simulations with real data. In addition, in whole atmosphere applications, the upper atmosphere undergoes very large day/night transitions in temperature (hundreds of degrees Kelvin), which makes a predefined reference state numerically unstable. For these reasons, we decided to implement and evaluate various HEVI schemes for both the low-altitude and high-altitude versions of NUMA.

The complexity and numerical efficiency of each of these schemes are evaluated along with their accuracy with idealized benchmarks, allowing us to determine the optimal configuration for NUMA and then make a recommendation for NEPTUNE.

The remainder of the paper is organized as follows. In Sec. 2, we describe the form of the governing equations used in our study along with the contravariant formulation used in its discretization. Section 3 details the spatial discretization scheme based on EBG methods [15], including special treatment of the metric terms using curl-invariant metrics [30]. Section 4 summarizes the three different flavors of the HEVI methods: nonlinear HEVI based on an iterative Krylov solver (NHEVI-GMRES), nonlinear HEVI using a direct LU solver (NHEVI-LU), and linear HEVI (LHEVI). Section 5 provides a complexity analysis of the three HEVI variants, while Sec. 6 compares five additive Runge-Kutta (ARK) time-integrators that may be used with any of the three HEVI variants. Numerical results using two standard global atmospheric test cases are shown in Sec. 7, followed by conclusions in Sec. 8. In Appendix A we discuss two other forms of the nonhydrostatic equations found in the literature and include the necessary ingredients for solving them using the HEVI time-integration methods described in this work.

2. Governing Equations

In this study we use the compressible Euler equations (with an artificial hyper-diffusion operator). According to the convention we defined in [17, 19] we refer to this particular form of the Euler equations as set 2NC, where the "NC" refers to a non-conservative set. In Appendix A.1, we include two other equation sets: 2C which is the conservation form of 2NC and 3C which uses the total energy as a prognostic variable. All three sets are valid for a deep-atmosphere [61]. In a forthcoming paper, we present an equation set which uses internal energy for the thermodynamic variable [22] that is valid for space weather applications.

2.1. Set 2NC

This is perhaps the most widely used form of the equations in atmospheric sciences and is written as follows

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (1a)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\rho} \nabla P + \nabla \Phi + 2\omega \hat{\mathbf{r}} \times \mathbf{u} = \mathbf{0} \quad (1b)$$

$$\frac{\partial \theta}{\partial t} + \mathbf{u} \cdot \nabla \theta = 0 \quad (1c)$$

$$P = P_A \left(\frac{\rho R \theta}{P_A} \right)^\gamma. \quad (1d)$$

In these equations, ρ is density, $\mathbf{u} = (u, v, w)^\mathcal{T}$ is the Cartesian velocity field, θ is the potential temperature, P is pressure, $\Phi = g|\mathbf{r}|$ is geopotential, \mathbf{r} is the position vector (measured from sea level), ω is the angular rotation of the earth, $\hat{\mathbf{r}} = \frac{\mathbf{r}}{|\mathbf{r}|}$ is the unit vector along the direction of \mathbf{r} , R is the specific gas constant, $\gamma = \frac{c_p}{c_v}$ is the ratio

of specific heats with respect to constant pressure and volume, $P_A = 10^5$ Pa is the reference pressure at sea level, and the superscript \mathcal{T} is the transpose operator. Note that the momentum (1b) and thermodynamic (1c) equations are in non-conservation (advective) form while the continuity (1a) equation is in conservation (flux) form; with the proper numerics (as the EBG method that we use in this work) this equation set will globally conserve mass as long as we do not use the product rule in (1a). Set 2NC, or variants thereof, are used in, e.g., E3SM [53] and NEPTUNE [39].

2.2. Contravariant Formulations

NUMA uses a Cartesian coordinate system, which while natural for flow in a box (where the grid is aligned with the coordinate axes), appears out of place for global atmospheric simulations on the sphere where the coordinate axes are no longer aligned with a spherical grid. This issue can be ameliorated by using contravariant vector quantities that *are* aligned with the grid. This requires mapping the Cartesian coordinates to the reference coordinates such that vectors (e.g., velocity) are transformed as follows: $(u, v, w) \rightarrow (u^\xi, u^\eta, u^\zeta)$ where (ξ, η) are aligned with the spherical manifold and ζ is aligned with the radial component, i.e., the direction along which the HEVI method is applied.

To express the transformation from covariant to contravariant vector quantities, we make use of Einstein notation whereby u_i and u^i denote covariant and contravariant quantities, respectively. In what follows, we refer to the covariant vectors as the vectors in Cartesian space which, within an element, represent a curvilinear representation, while the contravariant vectors are those that are aligned with the manifold and the crux of the problem is to construct a local (element-wise) map that takes each element defined in Cartesian space to its corresponding reference space.

It is easy to show (via the chain rule) that the advective terms in covariant form can be replaced by their contravariant form as follows

$$\sum_{i=1}^3 u_i \frac{\partial q}{\partial x_i} = \sum_{i=1}^3 u^i \frac{\partial q}{\partial \xi^i} \quad (2)$$

where $\mathbf{u} = (u, v, w) = (u_1, u_2, u_3)$,

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right) = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3} \right) \quad (3)$$

are the covariant vector quantities and $\mathbf{u}^\xi = (u^\xi, u^\eta, u^\zeta) = (u^1, u^2, u^3)$,

$$\nabla_\xi = \left(\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta}, \frac{\partial}{\partial \zeta} \right) = \left(\frac{\partial}{\partial \xi^1}, \frac{\partial}{\partial \xi^2}, \frac{\partial}{\partial \xi^3} \right) \quad (4)$$

are the contravariant vector quantities. It is less easy to show that divergence (e.g., see [31]) can be transformed as follows

$$\sum_{i=1}^3 \frac{\partial}{\partial x_i} (q u_i) = \frac{1}{J} \sum_{i=1}^3 \frac{\partial}{\partial \xi^i} (J q u^i) \quad (5)$$

where $u^i = \mathbf{u} \cdot \nabla \xi^i$ for any vector \mathbf{u} and scalar q with metric Jacobian J (see Eq. (13)) and contravariant metric terms $\nabla \xi^i$ given by (14) where ξ^i is defined as $(\xi^1, \xi^2, \xi^3) = (\xi, \eta, \zeta)$. With all these transformations defined, we can proceed to the spatial discretization of the equation sets. Note that discrete mass conservation requires the continuity equation (1a) to be discretized in the divergence form shown in (5). We cannot use the product rule and expand the divergence operator but must maintain it in divergence form; it can be shown that the global integral of this term vanishes when using EBG methods (e.g., see [52, 15]) that are presented in Sec. 3. The same argument also holds for sets 2C and 3C described in Appendix A.1 that are able to conserve quantities written in this form (e.g., the mass for both and density potential temperature and total energy, respectively).

3. Spatial Discretization

Although the focus of this work is on time-integration, let us first describe the spectral element method used for the spatial discretization which will assist in describing the construction of the operator Jacobian in the HEVI methods. The basis functions $\psi(\xi, \eta, \zeta)$ are constructed as a tensor-product \otimes of the one-dimensional (1D) Lagrange polynomials $h(\xi^i)$ as follows

$$\psi_I(\xi, \eta, \zeta) = h_i(\xi) \otimes h_j(\eta) \otimes h_k(\zeta) \quad (6)$$

where $I = i + j(N_\xi + 1) + k(N_\xi + 1)(N_\eta + 1)$ is the map from the tensor-product to monolithic space, with corresponding quadrature weights $\varpi_I = \varpi_i^\xi \varpi_j^\eta \varpi_k^\zeta$ with (ξ_i, η_j, ζ_k) , $i = 0, \dots, N_\xi$, $j = 0, \dots, N_\eta$, $k = 0, \dots, N_\zeta$ denoting the Lobatto points of order (N_ξ, N_η, N_ζ) , respectively, that are used for both interpolation and integration, and $\psi \in \mathbb{H}^1$ (Sobolev space) (see, e.g., [38, 10, 31, 15]). Because we use the Lobatto points for both interpolation and integration, the 1D basis functions satisfy $h_i(\xi_j^k) = \delta_{ij}$ for any of the contravariant coordinates ξ^k where δ is the Kronecker function; this is due to the *Cardinality property* of Lagrange polynomials. Differentiating (6) yields

$$\begin{aligned} \frac{\partial}{\partial \xi} \psi_I(\xi, \eta, \zeta) &= \frac{dh_i(\xi)}{d\xi} \otimes h_j(\eta) \otimes h_k(\zeta) \\ \frac{\partial}{\partial \eta} \psi_I(\xi, \eta, \zeta) &= h_i(\xi) \otimes \frac{dh_j(\eta)}{d\eta} \otimes h_k(\zeta) \\ \frac{\partial}{\partial \zeta} \psi_I(\xi, \eta, \zeta) &= h_i(\xi) \otimes h_j(\eta) \otimes \frac{dh_k(\zeta)}{d\zeta} \end{aligned} \quad (7)$$

where, e.g., we define the 1D differentiation matrices as $D_{ij}^{\xi^k} = \frac{dh_i(\xi_j^k)}{d\xi^k}$, $\forall i, j = 0, \dots, N_{\xi^k}$. To simplify the exposition in Sec. 4, we exploit the *row-sum property* of D^{ξ^k} defined as $\sum_{i=0}^{N_{\xi^k}} D_{ij}^{\xi^k} = 0$ that can be used to show that for two functions p and q we satisfy the identity

$$p_i D_{ij}^{\xi^k} q_j = D_{ij}^{\xi^k} (p_i q_j), \quad (8)$$

which means that we can write advective terms, such as $u \frac{\partial u}{\partial \zeta}$ as $D_{ij}^\zeta (u_i u_j)$. This *flux-differencing* approach is used extensively in the literature addressing the construction of kinetic-energy preserving and entropy-stable discontinuous Galerkin methods (e.g., see [13, 59, 48]).

Using the basis functions (6) we can construct the mass matrix on each element Ω_e as such

$$M_{ij}^{(e)} = \int_{\Omega_e} \psi_i(x, y, z) \psi_j(x, y, z) d\Omega_e \equiv \int_{\widehat{\Omega}} \psi_i(\xi, \eta, \zeta) \psi_j(\xi, \eta, \zeta) J d\widehat{\Omega} \quad (9)$$

where we now use the monolithic space for the subscripts (i, j) , J is the metric Jacobian defined in the next section and $\widehat{\Omega}$ is the reference element defined by the cube $(\xi, \eta, \zeta) \in [-1, +1]^3$; note that we never need the basis functions $\psi(x, y, z)$ defined in the physical coordinates, only those in the reference coordinates. Applying co-located Lobatto integration to (9) yields the diagonal matrix

$$M_{ij}^{(e)} = \varpi_i J_i \delta_{ij} \quad (10)$$

due to Cardinality but is inexact since N Lobatto points integrate $\mathcal{O}(2N - 1)$ polynomials exactly (the mass matrix represents a $2N$ polynomial). To denote the diagonal nature of the mass matrix, we write it as $M_i^{(e)}$. To construct the global mass matrix we apply *direct-stiffness summation* (DSS) as follows

$$M_I = \bigwedge_{e=1}^{N_e} M_i^{(e)} \quad (11)$$

where $(i, e) \rightarrow I$ denotes the map from local element-wise to global gridpoint notation (see [23, 15]) and N_e denotes the number of elements such that

$$\Omega = \bigcup_{e=1}^{N_e} \Omega_e$$

where Ω is the global domain. Below we drop the subscript and write the global gridpoint mass matrix as M where its inverse is simply defined and denoted as M^{-1} .

3.1. Metric Terms

To transform from the physical Cartesian coordinates (x, y, z) to the reference element coordinates (ξ, η, ζ) requires us to discuss the construction of the metric terms. The standard approach to constructing them are the *cross-product* metrics (e.g., see [15]) written as

$$\nabla^{\xi^i} = \frac{1}{J} \left(\frac{\partial \mathbf{x}}{\partial \xi^j} \times \frac{\partial \mathbf{x}}{\partial \xi^k} \right) \quad (12)$$

where the metric Jacobian is defined as

$$J = \frac{\partial \mathbf{x}}{\partial \xi} \cdot \left(\frac{\partial \mathbf{x}}{\partial \eta} \times \frac{\partial \mathbf{x}}{\partial \zeta} \right) \quad (13)$$

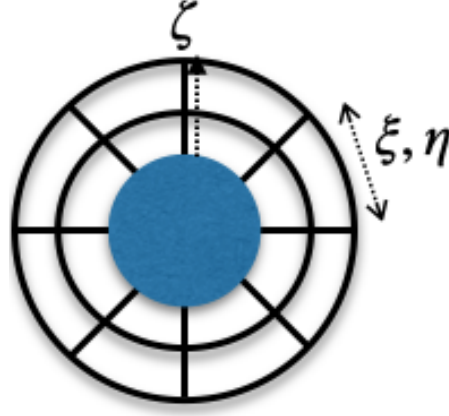


Figure 1: Alignment of grid such that $\xi - \eta$ define the spherical manifold while ζ defines the radial direction.

and i, j, k are defined cyclically such that if $i = 1$, then $j = 2$, and $k = 3$, etc. The cross-product metric terms can maintain constant stream preservation in 2D but do not have this property in 3D (see [30]). To satisfy this condition in 3D requires the use of the *curl-invariant* metrics (e.g., see [30]) that are written as

$$\nabla_{\xi^i} = \frac{1}{2J} \left[\frac{\partial}{\partial \xi^k} \left(\frac{\partial \mathbf{x}}{\partial \xi^j} \times \mathbf{x} \right) - \frac{\partial}{\partial \xi^j} \left(\frac{\partial \mathbf{x}}{\partial \xi^k} \times \mathbf{x} \right) \right] \quad (14)$$

where, once again, the indices (i, j, k) are cyclic, and the metric Jacobian is computed as described in [30].

In previous work [59, 48], we used the curl-invariant metrics but did not compare them to the cross-product metrics. We rectify this by demonstrating the superiority of the curl-invariant metrics to the cross-product form and then use the curl-invariant metrics throughout the paper because they are guaranteed to maintain constant-stream-preservation under all conditions, including when the boundaries are under-resolved, whether topography is present or not, and under inexact integration of the integrals required in efficient EBG methods. In an upcoming paper, we quantitatively compare various metric terms using a suite of tropospheric and high-altitude atmospheric problems in [22], further demonstrating the advantages of the curl-invariant metrics.

3.2. Metric Terms along Columns

By constructing our grids in NUMA such that the ζ direction is aligned with the vertical columns, this means that the HEVI methods only have to consider the ζ direction; this is depicted in Fig. 1 where $\xi - \eta$ define the spherical manifold (which we refer to as the horizontal direction), while ζ is aligned with the radial direction (referred to as the vertical direction) and is the direction along which the HEVI time-integrators are constructed.

However, the metric terms defined in (14) and (13) are uniquely defined for each element (e) and DOF (i, j, k) which implies that along a vertical column that shares two or more element columns (on the cubed-sphere, this can be 1, 2, 3, or 4 but much higher on an icosahedral grid) the metric terms as defined are C^0 continuous. For two of the three methods/storage schemes used in NUMA, this poses no problem. However, for a gridpoint-based continuous Galerkin method (what we call CGc in [2, 15]) we need to modify the metric terms. This procedure requires the following steps: (i) expand a given metric term \mathcal{M} in tensor-product Lagrange polynomials, (ii) evaluate the weak form within an element, (iii) apply DSS, and (iv) weight the result by the inverse mass matrix. This procedure ensures that the metric terms are continuous across elements. Steps (i) through (iv) outlined above can be defined concisely by the following L^2 projection

$$\mathcal{M}_I = M^{-1} \bigwedge_{e=1}^{N_e} \int_{\Omega_e} h_i(\xi) h_j(\eta) h_k(\zeta) \left(\sum_{l=0}^{N_\xi} \sum_{m=0}^{N_\eta} \sum_{n=0}^{N_\zeta} h_l(\xi) h_m(\eta) h_n(\zeta) \mathcal{M}_{l,m,n}^{(e)} \right) d\Omega_e \quad (15)$$

where h are the 1D basis functions previously defined, \mathcal{M}_I represents the metric terms at the global gridpoint I , $\mathcal{M}_{i,j,k}^{(e)}$ represents the metric terms for the element e at the local gridpoint i, j, k ; e.g., $\mathcal{M}_{i,j,k}^{(e)}$ could represent the components of $\nabla \zeta_{i,j,k}^{(e)}$. Once we apply this operation to J and $\nabla \zeta$, each vertical column can be integrated independently without the need to DSS along the (ξ, η) directions; however, DSS along the ζ direction is still required but this poses no issue if we partition the grid such that all points of a column are on the same memory space (either an MPI rank or GPU card). Note that this process is a one-time cost performed at initialization (or whenever the grid is modified as in dynamically adaptive mesh refinement), so this L^2 projection does not negatively impact performance.

3.3. Testing the Metric Terms

Mass conservation is a fundamental property of a dynamical core and important for accurate forecasts, such a surface pressure prediction [54]. Properly constructing metric terms is a prerequisite for mass conservation. To evaluate the curl-invariant metric terms, let us use a test case similar to the mountain case from [55], and compare them with the cross-product metrics. We modify the size of the mountain and apply no initial wind velocity. We define the profile of the axisymmetric mountain as follows

$$\text{height} = \frac{h_m}{1 + \left(\frac{r}{a_m} \right)^2} \quad (16)$$

where $r = r_e \cos^{-1}(\cos \phi \cos \lambda)$, r_e is the radius of the earth, with $h_m = 15$ km and $a_m = 5000$ km, with the model top at $z_T = 30$ km. The profile of the mountain has been made artificially large (almost twice the height of Mount Everest) in order to stress the metric terms. For the mountain case, we use 5×5 elements on each panel of the cubed-sphere grid [41] along with 6 elements in the vertical using polynomial degree $N = 4$ in all directions which results in an average grid resolution of 500 km

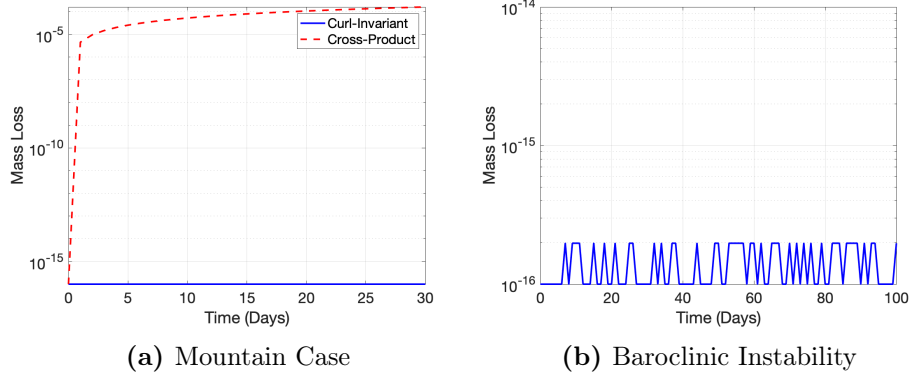


Figure 2: Mass loss for set 2NC for the (a) static balanced mountain for a 30-day simulation comparing the curl-invariant and cross-product metrics and (b) baroclinic instability for a 100-day simulation using the curl-invariant metrics.

in the horizontal and 1.25 km in the vertical where a terrain-following coordinate is used [11]. The grid is purposely coarse to test the performance of the metric terms in such resolution regimes. Figure 2a shows the mass loss for set 2NC for a 30-day simulation with data at a daily cadence; the mass loss is defined as

$$\text{Mass Loss} = \frac{|M(t) - M(0)|}{M(0)} \quad (17)$$

where

$$M(t) = \int_{\Omega} \rho(t) d\Omega \quad (18)$$

for any time t integrating along the global domain Ω . The dashed red line in Fig. 2a representing the cross-product metrics (12) hovers near $\mathcal{O}(10^{-5})$ mass loss. In contrast, the solid blue line for the curl-invariant metrics (14), evaluated with (15), conserves mass to machine double precision; the results for sets 2C and 3C are similar, the only difference is that 3C also conserves total energy. It is for this reason that from here on we use the curl-invariant metrics exclusively. Figure 2b shows the mass loss of the curl-invariant metrics for the baroclinic instability (presented in Sec. 7.1) for a 100-day simulation using a grid resolution of 104×1.25 km, where machine double-precision is achieved for the entirety of the 100-day simulation.

3.4. Tensor-based Hyper-Diffusion

We introduce numerical diffusion/hyper-diffusion by adding the following term to the right-hand side of the continuous governing equations

$$\frac{\partial \mathbf{q}}{\partial t} = S(\mathbf{q}) + (-1)^{\alpha+1} (\nabla \cdot \boldsymbol{\tau} \nabla)^{\alpha} \mathbf{q}, \quad (19)$$

where $S(\mathbf{q})$ represents the spatial operators of the compressible Euler equations, \mathbf{q} the state variable, $\boldsymbol{\tau}$ the viscosity tensor, and α is the order of the hyper-diffusion

operator (e.g., $\alpha = 1$ is second order standard diffusion and $\alpha = 2$ is fourth order hyper-diffusion). In this formulation, we adopt the tensor form of viscosity proposed in [21], but modify the original formulation by multiplying the viscosity tensor $\boldsymbol{\tau}$ by each application of Laplacian operator. Applying an EBG discretization to (19) yields the semi-discrete form

$$M \frac{d\mathbf{q}}{dt} = \mathcal{S}(\mathbf{q}) + \mathcal{H}_\nu(\mathbf{q}), \quad (20)$$

with

$$\mathcal{H}_\nu(\mathbf{q}) = (-1)^{\alpha+1} (\mathbf{L}_\nu)^\alpha \mathbf{q}, \quad (21)$$

where M is the global mass matrix, \mathcal{H}_ν is the hyper-diffusion operator, \mathbf{L}_ν is the Laplacian matrix with viscosity, and \mathcal{S} is the discrete EBG representation of S . Repeated application of the Laplacian operator gives rise to higher even-order derivatives [14, 27, 15]. The Laplacian matrix \mathbf{L}_ν is calculated using the contravariant form in terms of the reference coordinates $\boldsymbol{\xi}$ on the element Ω_e as follows

$$\mathbf{L}_\nu = \int_{\Omega_e} \nabla \psi_i \cdot \boldsymbol{\tau} \nabla \psi_j d\Omega_e \quad (22)$$

where ∇ is defined by (3). Since $\nabla \psi = \mathbf{J}^{-\mathcal{T}} \nabla_\xi \psi$ with ∇_ξ defined by (4), $\mathbf{J} = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}$, and $\mathbf{J}^{-\mathcal{T}} = (\mathbf{J}^\mathcal{T})^{-1}$, we rewrite (22) as

$$\begin{aligned} \mathbf{L}_\nu &= \int_{\hat{\Omega}} (\nabla_\xi \psi_i)^\mathcal{T} (\mathbf{J}^{-1} \boldsymbol{\tau} \mathbf{J}^{-\mathcal{T}}) \nabla_\xi \psi_j J d\hat{\Omega} \\ &= \int_{\hat{\Omega}} (\nabla_\xi \psi_i)^\mathcal{T} \mathbf{G}_\nu \nabla_\xi \psi_j J d\hat{\Omega}, \end{aligned} \quad (23)$$

where $\hat{\Omega}$ denotes the reference element, J the metric Jacobian defined by (13), and the viscous metric tensor \mathbf{G}_ν is defined as

$$\mathbf{G}_\nu = \mathbf{J}^{-1} \boldsymbol{\tau} \mathbf{J}^{-\mathcal{T}}. \quad (24)$$

Formulating hyper-diffusion using (20), (21), and (22) only requires storing a single metric tensor \mathbf{G}_ν instead of both a viscosity tensor $\boldsymbol{\tau}$ and the viscous metric tensors, thereby reducing memory consumption. We employ the following eigendecomposition of the viscosity tensor $\boldsymbol{\tau}$ from [21]

$$\boldsymbol{\tau} = \mathbf{J} \mathbf{E} \begin{pmatrix} \nu_1 \lambda_1^{-1} & 0 & 0 \\ 0 & \nu_2 \lambda_2^{-1} & 0 \\ 0 & 0 & \nu_3 \lambda_3^{-1} \end{pmatrix} (\mathbf{J} \mathbf{E})^\mathcal{T}, \quad (25)$$

where (ν_1, ν_2, ν_3) are the viscosity parameters that act in the principal axes of the metric tensor $\mathbf{G} = \mathbf{J}^{-1} \mathbf{J}^{-\mathcal{T}}$, and $(\lambda_1^{-1}, \lambda_2^{-1}, \lambda_3^{-1})$ are the eigenvalues of \mathbf{G} in ascending order ($\lambda_1 > \lambda_2 > \lambda_3$). The matrix \mathbf{E} contains the normalized eigenvectors $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ of \mathbf{G} in its columns, i.e., $\mathbf{E} = [\mathbf{e}_1 | \mathbf{e}_2 | \mathbf{e}_3]$. \mathbf{E} is orthonormal because \mathbf{G} is symmetric. The square root $\sqrt{\lambda}$ can be interpreted as the stretching from the reference to the

physical element, i.e., $\Delta x = \sqrt{\lambda}\Delta\xi = 2\sqrt{\lambda}$ for $\xi \in [-1, 1]$, which implies that $\sqrt{\lambda}$ characterizes the length scale of an element.

Substituting (25) into (24) yields the eigendecomposition

$$\begin{aligned}\mathbf{G}_\nu &= \mathbf{E} \begin{pmatrix} \nu_1 \lambda_1^{-1} & 0 & 0 \\ 0 & \nu_2 \lambda_2^{-1} & 0 \\ 0 & 0 & \nu_3 \lambda_3^{-1} \end{pmatrix} \mathbf{E}^\mathcal{T} \\ &= \nu_1 \lambda_1^{-1}(\mathbf{e}_1 \otimes \mathbf{e}_1) + \nu_2 \lambda_2^{-1}(\mathbf{e}_2 \otimes \mathbf{e}_2) + \nu_3 \lambda_3^{-1}(\mathbf{e}_3 \otimes \mathbf{e}_3).\end{aligned}\quad (26)$$

The viscosity parameters ν_i have physical dimensions $L^2/T^{1/\alpha}$, while kinematic (physical) hyperviscosity ν_p has dimensions $L^{2\alpha}/T$. The viscosity parameters can be scaled by the element size Δx and time-step Δt as follows

$$\nu_i \equiv (c_i)^{1/\alpha} \frac{(\Delta x_i)^2}{(\Delta t)^{1/\alpha}} = (c_i)^{1/\alpha} \frac{4\lambda_i}{N^2(\Delta t)^{1/\alpha}}, \quad i = 1, \dots, 3 \quad (27)$$

where $c_i = (c_1, c_2, c_3)$ are the dimensionless coefficients that independently control the amount of numerical hyper-diffusion in the principal directions and $\Delta x_i = 2\sqrt{\lambda_i}/N$ is the characteristic length for an element of polynomial order N (assumed constant to simplify the discussion); to recover the physical viscosity ν_p we need to exponentiate (27) by α .

In many atmospheric applications, a 3D mesh is generated using thin elements. These elements are characterized by two significantly large eigenvalues and a much smaller eigenvalue, $\lambda_1 \approx \lambda_2 \gg \lambda_3$, due to their high aspect ratio. In this case, we treat viscosity anisotropically by setting $\nu_H = \nu_1 = \nu_2$ for the horizontal direction and $\nu_V = \nu_3$ for the vertical direction, assuming that $\lambda_H = \lambda_1 = \lambda_2$ and $\lambda_V = \lambda_3$. This setting simplifies the expression for \mathbf{G}_ν in (26) as a combination of horizontal and vertical contributions, as such

$$\mathbf{G}_\nu = \nu_H \lambda_H^{-1}(\mathbf{e}_1 \otimes \mathbf{e}_1 + \mathbf{e}_2 \otimes \mathbf{e}_2) + \nu_V \lambda_V^{-1}(\mathbf{e}_3 \otimes \mathbf{e}_3). \quad (28)$$

4. Horizontally Explicit Vertically Implicit (HEVI)

With the contravariant form of the governing equations and spatial discretization both defined, we are now in a position to describe the HEVI time-integration strategy. We base the HEVI approach on Additive Runge-Kutta (ARK) methods, although other time-integration approaches could also be used (e.g., multi-step methods). We describe three forms of HEVI methods: two nonlinear (NHEVI-GMRES and NHEVI-LU) methods and one linear (LHEVI) method. The difference between linear and nonlinear HEVI is that in nonlinear HEVI (NHEVI), all the vertical terms are solved nonlinearly in an implicit fashion, as opposed to linearly in linear HEVI (LHEVI); this is described in detail below.

4.1. Implicit-Explicit (IMEX) Runge-Kutta Method

Since our HEVI method is based on additive Runge-Kutta methods, let us first describe the IMEX Runge-Kutta method. Consider solving the system of ordinary

differential equations (ODEs)

$$\frac{d\mathbf{q}}{dt} = \mathbf{E}(\mathbf{q}) + \mathbf{I}(\mathbf{q}) \quad (29)$$

where \mathbf{E} denotes the operators that are handled explicitly and \mathbf{I} those that are handled implicitly.

For the state vector \mathbf{q} , the stage values \mathbf{Q}_i are defined as follows

$$\mathbf{Q}_i = \mathbf{q}^n + \Delta t \sum_{j=1}^{i-1} a_{ij} \mathbf{E}(\mathbf{Q}_j) + \Delta t \sum_{j=1}^i \tilde{a}_{ij} \mathbf{I}(\mathbf{Q}_j), \quad (30)$$

for each stage $i = 1, \dots, s$. The updated solution is obtained in the following manner

$$\mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t \sum_{i=1}^s \left[b_i \mathbf{E}(\mathbf{Q}_i) + \tilde{b}_i \mathbf{I}(\mathbf{Q}_i) \right] \quad (31)$$

where a_{ij}, b_i and $\tilde{a}_{ij}, \tilde{b}_i$ are the Butcher tableaux coefficients for the explicit and implicit terms, respectively, and the superscripts n and $n+1$ denote the values at the current and next time-level. In this work, we only consider IMEX methods such that \mathbf{A} is strictly lower triangular and $\tilde{\mathbf{A}}$ is lower triangular, which makes them diagonally-implicit Runge-Kutta (DIRK) methods; to be more precise, we only consider DIRK methods that have the same (singly) diagonal implicit coefficient $\tilde{a}_{ii} = \text{constant}$ (SDIRK) and whereby the first stage is explicit (ESDIRK). The Butcher tableaux are given in Table 1 for a general Runge-Kutta method with $\mathbf{b} = \tilde{\mathbf{b}}$ which is a necessary condition for conserving linear invariants (e.g., see [16]).

Table 1: Butcher tableaux for the horizontal (left) and vertical (right) terms.

\mathbf{c}	\mathbf{A}	$\tilde{\mathbf{c}}$	$\tilde{\mathbf{A}}$
	\mathbf{b}^T		$\tilde{\mathbf{b}}^T$

We explored a range of SDIRK and DIRK methods from the literature including those in [24, 16, 51, 58, 25, 20] and only present those methods that resulted in the most efficient time-to-solution. To analyze the results of the study, we consider the ordinary differential equation

$$\frac{dq}{dt} = \{ik_s q\} + [ik_f q] \quad (32)$$

where $i = \sqrt{-1}$, k_s and k_f are the frequencies of the slow and fast waves. If we now analyze the stability region (as we showed in [16]) for implicit fast processes (order ≈ 40) versus explicit slow processes (order ≈ 4), for the ARK(2,3,2)¹ method

¹We use the IMEX designation ARK(i,e,o) from [5, 37] where i,e,o denote the number of implicit and explicit stages, and the order of the method, respectively.

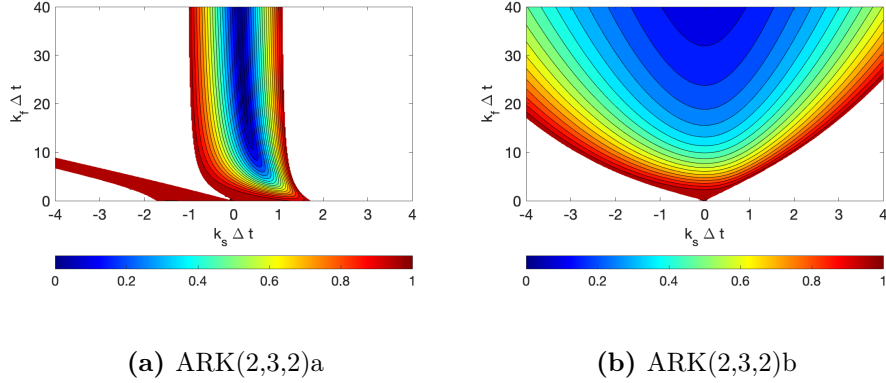


Figure 3: IMEX stability region for implicit fast processes (k_f) versus explicit slow processes (k_s) for the ARK(2,3,2) method.

presented in [16] we arrive at the stability regions illustrated in Fig. 5a; ARK(2,3,2)a was proposed in [16] ($a_{32} = \frac{1}{6}(3 + 2\sqrt{2})$) due to the large explicit stability region along the imaginary axis and for its accuracy (eliminates the second-order error for the explicit components). However, the choice $a_{32} = \frac{1}{2}$ has been the *de facto* method used in NUMA (see [15], Ch. 20, p. 464) due to its larger allowable time-step. The conclusion from our time-integrator study is that those methods that have the wedge-shape stability region as in Fig. 3b admit larger time-steps than those methods with the shape shown in Fig. 3a. It is for this reason that we limit our study to IMEX methods with stability regions similar to those shown in Fig. 3b.

Below, we describe two types of HEVI methods; in NHEVI at each stage value Q_i calculation, a nonlinear implicit system of equations must be solved. Within NHEVI, we describe two variants: NHEVI-GMRES whereby the linear problem is solved using GMRES and NHEVI-LU where the operator Jacobian (defined in Sec. 4.2.2) is constructed and then solved using direct (LU) solvers. In contrast, in LHEVI, only a linear system needs to be solved; we can decompose LHEVI into two separate classes: linearization over a time-independent reference state (RS) and linearization over a time-dependent previous solution (PS), although we only focus on LHEVI-PS. Let us now describe the NHEVI and LHEVI methods.

4.2. Nonlinear HEVI (NHEVI)

To motivate the HEVI methods, let us consider a time-integration strategy whereby the stiffness of the PDE is derived from the vertical processes as in the case of global atmospheric simulations where the vertical resolution is much finer than the horizontal. This arises because the spatial scales are so different in these directions where the horizontal dimension is on the order of $2\pi r_e = 2\pi \cdot 6.371 \times 10^3 \text{ km} \approx 40,000 \text{ km}$, whereas the vertical domain (in earth weather simulations) is $\approx 40 \text{ km}$ which gives a ratio of 1000:1. For this reason, the grid resolution in the vertical direction is much finer than in the horizontal. For these types of domains, it behooves us to treat the vertical direction differently and so let us consider treating it implicitly and fully nonlinearly. We now describe the details of the NHEVI approach.

Let \mathbf{q} be the state vector and consider splitting the right-hand side of the PDE into horizontal and vertical terms as follows

$$\frac{\partial \mathbf{q}}{\partial t} = \mathbf{H}(\mathbf{q}) + \mathbf{V}(\mathbf{q}) \quad (33)$$

where we now seek to solve

$$\mathbf{Q}_i = \mathbf{q}^n + \Delta t \sum_{j=1}^{i-1} a_{ij} \mathbf{H}(\mathbf{Q}_j) + \Delta t \sum_{j=1}^i \tilde{a}_{ij} \mathbf{V}(\mathbf{Q}_j), \quad (34)$$

where \mathbf{H} denotes the horizontal terms that are handled explicitly and \mathbf{V} the vertical terms that are handled implicitly. Even though we use Cartesian coordinates to represent the velocity field, we are able to decompose the horizontal and vertical directions; we make use of the transformations described in Sec. 2.2. Before continuing, we need to define the continuous horizontal and vertical operators.

4.2.1. HEVI Horizontal and Vertical Operators

In order to define the necessary horizontal and vertical operators for set 2NC, we first expand the divergence operator in the continuity equation (1a) using the contravariant formulation given by (5) and the gradient operators in (1b) and (1c) using (2). The horizontal components given by the ξ and η derivatives are collected in

$$\mathbf{H}(\mathbf{q}) = - \left(\mathbf{u}^H \cdot \nabla_H \mathbf{u} + \frac{1}{\rho} \left(\frac{\partial P}{\partial \xi} \nabla \xi + \frac{\partial P}{\partial \eta} \nabla \eta \right) + \left(\frac{\partial \Phi}{\partial \xi} \nabla \xi + \frac{\partial \Phi}{\partial \eta} \nabla \eta \right) + 2\omega \hat{\boldsymbol{\zeta}} \times \mathbf{u} \right) + \mathcal{H}_\nu(\mathbf{q}) \quad (35)$$

where $\mathbf{u}^H = (u^\xi, u^\eta, 0)^\top$, $\nabla_H = (\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta}, 0)^\top$, and $\hat{\boldsymbol{\zeta}}$ is the unit vector in the $\boldsymbol{\zeta}$ direction; we include the hyper-diffusion operator in the explicit operator even though it includes both horizontal and vertical components. The vertical components given by the ζ derivatives are

$$\mathbf{V}(\mathbf{q}) = - \left(u^\zeta \frac{\partial \mathbf{u}}{\partial \zeta} + \frac{1}{\rho} \frac{\partial P}{\partial \zeta} \nabla \zeta + \frac{\partial \Phi}{\partial \zeta} \nabla \zeta \right). \quad (36)$$

Since all the vertical operators, including vertical advection, are included in \mathbf{V} , our scheme is similar to the HEVI-A approach proposed in [12]. The horizontal and vertical operators for sets 2C and 3C are defined in Appendix A.2.

4.2.2. NHEVI-GMRES

In order to compute the stage value \mathbf{Q}_i , we reformulate the problem similarly to the multidimensional root-finding problem: $\mathbf{F}(\mathbf{Q}_i) = \mathbf{0}$, where the nonlinear function \mathbf{F} is defined as

$$\mathbf{F}(\mathbf{q}) = \mathbf{q} - \Lambda \mathbf{V}(\mathbf{q}) - \mathbf{R}, \quad (37)$$

with $\Lambda = \tilde{a}_{ii}\Delta t$ and

$$\mathbf{R} = \mathbf{q}^n + \Delta t \sum_{j=1}^{i-1} [a_{ij} \mathbf{H}(\mathbf{Q}_j) + \tilde{a}_{ij} \mathbf{V}(\mathbf{Q}_j)].$$

Given an initial guess \mathbf{q}_0 , we use the Newton-Raphson method to generate a better approximation to the solution through successive iterations m

$$\mathbf{q}_{m+1} = \mathbf{q}_m + \delta \mathbf{q}_m \quad (38)$$

where $\delta \mathbf{q}_m$ is computed by solving the linear system

$$\mathbb{J}(\mathbf{q}_m) \delta \mathbf{q}_m = -\mathbf{F}(\mathbf{q}_m) \quad (39)$$

with

$$\mathbb{J}(\mathbf{q}_m) = \frac{\partial \mathbf{F}}{\partial \mathbf{q}}(\mathbf{q}_m)$$

defining the operator Jacobian of the function \mathbf{F} evaluated at \mathbf{q}_m . In order to avoid building the operator Jacobian matrix (see, e.g., [40, 29]), we use an approximation for the action of the operator Jacobian on the vector $\delta \mathbf{q}$ using a Fréchet derivative, defined as

$$\mathbb{J}(\mathbf{q}_m) \delta \mathbf{q}_m \approx \frac{\mathbf{F}(\mathbf{q}_m + \boldsymbol{\mathcal{E}} \delta \mathbf{q}_m) - \mathbf{F}(\mathbf{q}_m)}{\boldsymbol{\mathcal{E}}} \quad (40)$$

where $\boldsymbol{\mathcal{E}}$ is some small perturbation. $\boldsymbol{\mathcal{E}}$ can be a small scalar value, relative to the norm of \mathbf{q}_m , but we use different magnitudes for each of the state variables \mathbf{q} as described in [58]. In this case, $\boldsymbol{\mathcal{E}}$ is defined as a diagonal matrix.

Substituting (40) into (39) and rearranging yields the final form for the Jacobian-free Newton-Krylov (JFNK) method

$$\mathbf{res} \equiv \frac{\mathbf{F}(\mathbf{q}_m + \boldsymbol{\mathcal{E}} \delta \mathbf{q}_m) + (\boldsymbol{\mathcal{E}} - \mathbf{I}) \mathbf{F}(\mathbf{q}_m)}{\boldsymbol{\mathcal{E}}} \approx \mathbf{0} \quad (41)$$

that is written in residual form that is suitable for the application of the generalized minimal residual (GMRES) [43] method. The evaluation of the residual in (41) only requires one evaluation of \mathbf{F} , which has a similar cost of evaluating the right-hand-side (RHS) in an explicit time-step; we say one and not two because $\mathbf{F}(\mathbf{q}_m)$ is already known. This is a key concept to the power and popularity of JFNK methods. Finally, since only the vertical terms are solved implicitly, we can solve the system for each column of the mesh independently, which avoids unnecessary off-process data communication and data storage. Because this approach uses a nonlinear HEVI method with GMRES, we refer to it as NHEVI-GMRES.

For set 2NC since the state vector is $\mathbf{q} = (\rho, \mathbf{u}^\tau, \theta)^\tau$, then we seek the root of the function

$$\mathbf{F}(\mathbf{q}) = \begin{pmatrix} \rho + \Lambda \left(\frac{1}{J} \frac{\partial (J \rho u^\zeta)}{\partial \zeta} \right) + R_\rho \\ \mathbf{u} + \Lambda \left(u^\zeta \frac{\partial \mathbf{u}}{\partial \zeta} + \frac{1}{\rho} \frac{\partial P}{\partial \zeta} \nabla \zeta + \frac{\partial \Phi}{\partial \zeta} \nabla \zeta \right) + \mathbf{R}_u \\ \theta + \Lambda \left(u^\zeta \frac{\partial \theta}{\partial \zeta} \right) + R_\theta \end{pmatrix} \quad (42)$$

where u^ζ is the third component of the contravariant velocity $\mathbf{u}^\zeta = (u^\xi, u^\eta, u^\zeta)$. Recall that we build the grid such that the ζ direction of each element is aligned with the vertical component of the grid, regardless of the geometry (e.g., flow on a sphere or in a box). For sets 2C and 3C, the function \mathbf{F} is defined by (37) with the corresponding state vector \mathbf{q} and right-hand side function \mathbf{R} .

The NHEVI-GMRES method presents subtle issues including a heavy computational cost. To reduce the cost of this algorithm we may consider the following strategies. The number of Newton iterations can be reduced by increasing the value of the stopping criterion, but this strategy is limited since the solution needs to be accurate. The number of GMRES iterations can be reduced by preconditioning which we did not implement. The strategies that we did pursue include: (1) replacing the GMRES solver in the solution of the linear problem with LU factorization and (2) removing the Newton solver in the nonlinear loop with frequent linearizations; both of these strategies are described in the sections below where we refer to them as NHEVI-LU in Sec. 4.2.3 and LHEVI in Sec. 4.3.

4.2.3. NHEVI-LU

An alternative to NHEVI-GMRES requires computing the operator Jacobian analytically. In this approach, we rebuild the Jacobian at each Newton step and then solve the resulting matrix problem using an LU factorization; for this reason we refer to it as NHEVI-LU.

Analytical Operator Jacobian for Set2NC. The function $\mathbf{F}(\mathbf{q})$ for which we need to find the roots in the NHEVI method is given in (42). The analytical operator Jacobian of this function, evaluated at a known state vector $\mathbf{q}_0 = (\rho_0, \mathbf{u}_0^\tau, \theta_0)^\tau$ and applied to the state vector $\mathbf{q} = (\rho, \mathbf{u}^\tau, \theta)^\tau$, is given by

$$\mathbb{J}_{2NC}(\mathbf{q}) = \mathbb{I} + \Lambda \mathbb{D}^\zeta \mathbb{K}_{2NC}(\mathbf{q}) \quad (43)$$

where

$$\mathbb{D}_{ij}^\zeta = \left(-\frac{(\varpi J)_j}{(\varpi J)_i} D_{ji}^\zeta, D_{ij}^\zeta, D_{ij}^\zeta, D_{ij}^\zeta, D_{ij}^\zeta \right)$$

for the 5 prognostic variables with \mathbb{I} denoting an identity matrix and

$$\mathbb{K}_{2NC}(\mathbf{q}) = \begin{pmatrix} (u^\zeta)_j & (\rho \zeta_x)_j & (\rho \zeta_y)_j & (\rho \zeta_z)_j & 0 \\ -\left(\frac{\zeta_x}{\rho^2}\right)_i P_j + \left(\frac{\zeta_x}{\rho}\right)_i \frac{\partial P_j}{\partial \rho} & (\zeta_x)_i u_j + u_i^\zeta & (\zeta_y)_i u_j & (\zeta_z)_i u_j & \left(\frac{\zeta_x}{\rho}\right)_i \frac{\partial P_j}{\partial \theta} \\ -\left(\frac{\zeta_y}{\rho^2}\right)_i P_j + \left(\frac{\zeta_y}{\rho}\right)_i \frac{\partial P_j}{\partial \rho} & (\zeta_x)_i v_j & (\zeta_y)_i v_j + u_i^\zeta & (\zeta_z)_i v_j & \left(\frac{\zeta_y}{\rho}\right)_i \frac{\partial P_j}{\partial \theta} \\ -\left(\frac{\zeta_z}{\rho^2}\right)_i P_j + \left(\frac{\zeta_z}{\rho}\right)_i \frac{\partial P_j}{\partial \rho} & (\zeta_x)_i w_j & (\zeta_y)_i w_j & (\zeta_z)_i w_j + u_i^\zeta & \left(\frac{\zeta_z}{\rho}\right)_i \frac{\partial P_j}{\partial \theta} \\ 0 & (\zeta_x)_i \theta_j & (\zeta_y)_i \theta_j & (\zeta_z)_i \theta_j & u_i^\zeta \end{pmatrix} \quad (44)$$

where J is the metric Jacobian defined in (13), and $(\zeta_x, \zeta_y, \zeta_z)$ are the metric terms along the vertical direction. In (43), we only show the entries at a given gridpoint $i = 0, \dots, N_\zeta$ on one element Ω_e where the use of D_{ij}^ζ represents a strong form derivative while $-D_{ji}^\zeta$ denotes a weak form derivative obtained from integration by parts (e.g., see [15]); our experience shows we need the weak form in the continuity

equation to formally conserve mass. Note that since \mathbb{D}^ζ is defined for all gridpoints $(i, j) = 0, \dots, N_\zeta$ then upon applying the differentiation \mathbb{D}^ζ makes these matrices $\dim(\mathbb{J}_{2NC}, \mathbb{I}, \mathbb{K}_{2NC}) = n_{var}(N_\zeta + 1) \times n_{var}(N_\zeta + 1)$.

Applying numerical integration (multiplying by $\varpi_i J_i$) and DSS along the ζ direction along all elements N_e^ζ , results in the global representation $\mathcal{G}(\mathbb{J}_{2NC})$ defined in the vector space $\mathbb{R}^{M \times M}$ per vertical column (with $6[N_e^\xi N_\xi + 1][N_e^\eta N_\eta + 1]$ columns on a cubed-sphere grid), where $M = n_{var} n_z$, with $n_z = N_e^\zeta (N_\zeta + 1)$ denoting the number of gridpoints along a column, and \mathcal{G} represents the operator that takes \mathbb{J}_{2NC} and constructs its global representation. We do not present the global matrix because its construction will vary depending on the spatial discretization used, however, the operator at a given gridpoint presented in (43) will not (only the definition of \mathbb{D}^ζ).

The cost of building $\mathcal{G}(\mathbb{J})_{2NC}$ is $\mathcal{O}(n_{var} N_e^\zeta (N_\zeta + 1)^3)$, with similar costs for constructing the Jacobians for sets 2C and 3C; these Jacobians are defined in Appendix A.3.

Lapack Banded Matrix Routines. We build the Jacobian matrix directly in LAPACK's band storage matrix format and use subroutines to perform LU factorization and linear system solves that are specialized to banded matrices [4]. In order to use LAPACK's band storage, the user must specify the number of sub-diagonals k_l and super-diagonals k_u of the matrix. For our methods, the number of sub-diagonals is always equal to the number of super-diagonals and is given by

$$k_u = k_l = n_{var}(N_\zeta + 1) - 1, \quad (45)$$

where $n_{var} = 5$ is the number of state variables.

We already saw that the dimension of the sparse representation of $\mathcal{G}(\mathbb{J})$ for the NHEVI method is a matrix with size $M \times M$; however, the size of its banded storage form is $[3n_{var}(N_\zeta + 1) - 2] \times M$ and its entries are related to that of the original global operator Jacobian matrix $\mathcal{G}(\mathbb{J})$ through

$$\mathcal{G}(\mathbb{J}_B)_{2k_u+1+i-j, j} = \mathcal{G}(\mathbb{J})_{i,j} \quad (46)$$

for $\max(1, j - 2k_u) \leq i \leq \min(n_{var} \times n_z, j + k_u)$.

4.3. Linear HEVI (LHEVI)

An alternative approach to NHEVI is the linear HEVI (LHEVI) method whereby we solve the linear problem

$$\frac{\partial \mathbf{q}}{\partial t} = \{\mathbf{S}(\mathbf{q}) - \mathbb{L}(\mathbf{q})\}_{EX} + [\mathbb{L}(\mathbf{q})]_{IM} \quad (47)$$

where \mathbf{S} denotes the original right-hand side operator of the PDE while \mathbb{L} represents a linear approximation of the stiff terms; the curly brackets denote the terms that are solved explicitly (EX) whereas the square brackets are those that are solved implicitly (IM). We must consider how to create the linear operator \mathbb{L} ; this approach is akin to a Rosenbrock [42] method since \mathbb{L} is an approximation to the fully implicit nonlinear

operator Jacobian \mathbb{J} . The simplicity and efficiency of LHEVI can be illustrated by discretizing (47) in time using a forward/backward Euler method

$$\mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t \mathbf{S}(\mathbf{q}^n) + \Delta t \mathbb{L}(\mathbf{q}^{n+1} - \mathbf{q}^n) \quad (48)$$

$$= \mathbf{q}^E + \Delta t \mathbb{L}(\mathbf{q}^{n+1} - \mathbf{q}^n), \quad (49)$$

subtracting both sides by \mathbf{q}^n

$$\mathbf{q}^{n+1} - \mathbf{q}^n = \mathbf{q}^E - \mathbf{q}^n + \Delta t \mathbb{L}(\mathbf{q}^{n+1} - \mathbf{q}^n), \quad (50)$$

and letting $\mathbf{q}_{tt} = \mathbf{q}^{n+1} - \mathbf{q}^n$, $\hat{\mathbf{q}} = \mathbf{q}^E - \mathbf{q}^n$, and $\Lambda = \Delta t$ yields the final LHEVI form

$$\mathbf{q}_{tt} = \hat{\mathbf{q}} + \Lambda \mathbb{L} \mathbf{q}_{tt} \quad (51)$$

which can be rearranged to more easily illustrate the implicit operator as follows

$$(\mathbb{I} - \Lambda \mathbb{L}) \mathbf{q}_{tt} = \hat{\mathbf{q}}. \quad (52)$$

The extension to an arbitrary order IMEX additive Runge-Kutta (ARK) method is described in [16] which we summarize now for completeness. We employ (51) with the following definitions

$$\begin{aligned} \mathbf{q}_{tt} &= \mathbf{Q}^{(i)} + \sum_{j=1}^{i-1} \frac{\tilde{a}_{ij} - a_{ij}}{\tilde{a}_{ii}} \mathbf{Q}^{(j)} \\ \hat{\mathbf{q}} &= \mathbf{q}^E + \sum_{j=1}^{i-1} \frac{\tilde{a}_{ij} - a_{ij}}{\tilde{a}_{ii}} \mathbf{Q}^{(j)} \\ \mathbf{q}^E &= \mathbf{q}^n + \Delta t \sum_{j=1}^{i-1} a_{ij} \mathbf{S}(\mathbf{Q}^{(j)}). \end{aligned} \quad (53)$$

4.3.1. LHEVI Linear Operator

To construct a semi-implicit IMEX method for the nonhydrostatic equations we construct an approximation \mathbb{L} by applying the expansion $\mathbf{q}(\mathbf{x}, t) = \mathbf{q}_0(\mathbf{x}, t) + \mathbf{q}'(\mathbf{x}, t)$ in \mathbf{S} [6, 32, 49, 59, 48], where $\mathbf{q}_0(\mathbf{x}, t)$ represents the solution at a previous time-level t^n , and only retaining terms linear in $\mathbf{q}'(\mathbf{x}, t)$. Doing so, results in the linear operator

$$\mathbb{L}(\mathbf{q}) = - \left(\begin{array}{c} \frac{1}{J} \frac{\partial}{\partial \zeta} \left[J \left(\rho_0 u^\zeta + \rho u_0^\zeta \right) \right] \\ u_0^\zeta \frac{\partial \mathbf{u}}{\partial \zeta} + u^\zeta \frac{\partial \mathbf{u}_0}{\partial \zeta} + \frac{\nabla \zeta}{\rho_0} \left[\frac{\partial}{\partial \zeta} (F_0 \rho + G_0 \theta) - \frac{\rho}{\rho_0} \frac{\partial P_0}{\partial \zeta} \right] \\ u^\zeta \frac{\partial \theta_0}{\partial \zeta} + u_0^\zeta \frac{\partial \theta}{\partial \zeta} \end{array} \right) \quad (54)$$

which we call LHEVI-PS (previous solution). A special case of LHEVI-PS arises when we apply a time-independent linearization defined as $\mathbf{q}(\mathbf{x}, t) = \mathbf{q}_0(\mathbf{x}) + \mathbf{q}'(\mathbf{x}, t)$; linearizing only about the scalar variables results in the semi-implicit IMEX methods described in [18, 16, 60, 33, 1, 36] which we call LHEVI-RS (reference state).

The reference state $\mathbf{q}_0(\mathbf{x})$ is selected to satisfy either a hydrostatic balance and/or geostrophic balance.

In what follows, we build on the NHEVI-LU method described in Sec. 4.2.3 and use $\mathbb{L} = \mathbb{I} - \mathbb{J}$, where \mathbb{J} is the operator Jacobian of \mathbf{F} for each equation set; we can in fact see this relationship between \mathbb{L} and \mathbb{J} in (52). We then use LAPACK’s banded routines to perform the LU factorization and back substitution as described in Sec. 4.2.3. Therefore, in the LHEVI-PS method, we essentially apply the NHEVI-LU method but for only one Newton iteration. In order to avoid the creation, factorization, and solution of the linear system at every time step, we explore an option to set how often the reference state is modified; this is the reason why we refer to this state as a *previous solution*. This reference state can be any previous state including the current one as in NHEVI. Unless otherwise specified, we recompute the operator Jacobian every 5 time-steps and show in Sec. 7 that some time-integration methods allow for much larger time-steps between operator Jacobian updates (referred to as the update time-step).

5. Complexity Analysis

The number of vertical levels n_z may need to be increased in an atmospheric model for several applications, such as studying the planetary boundary layer (PBL) or extending a model into the upper atmosphere (e.g., whole-atmosphere modeling) [22]. Hence, knowledge of the computational complexity of the time-integration scheme with respect to n_z is an important consideration. Let us now describe the complexity of each of the three methods: NHEVI-GMRES, NHEVI-LU, and LHEVI (PS).

5.1. NHEVI-GMRES

In NHEVI-GMRES, we must solve the JFNK problem for all $n_{var} = 5$ prognostic variables (density, three velocities, and thermodynamic variable) so that n_z grid-points along a vertical column require the solution of a $5n_z \times 5n_z$ system of nonlinear equations to be solved with complexity

$$25n_z^2 N_{Newton}(col) N_{GMRES}(col)^2 \quad (55)$$

per column col per RK stage, where N_{Newton} denotes the number of Newton iterations, and N_{GMRES} the number of GMRES iterations. Because the number of Newton and GMRES iterations vary for each column, the cost per iteration will be that of the most expensive column. Recall that the number of points along a vertical column is defined as $n_z = N_e^\zeta N_\zeta + 1$, where N_ζ and N_e^ζ represent the polynomial order and number of elements along the ζ direction, which is aligned with the radial component of the sphere (recall Fig. 1).

5.2. NHEVI-LU

Next, let us consider reducing the complexity (and computational cost) of NHEVI-GMRES by obviating the need for the GMRES iterations. We achieve this by replacing the iterative solution with a direct one such as LU, although other direct solvers

can be substituted. A standard LU factorization requires $\mathcal{O}(50n_z^3)$ for the forward reduction and $\mathcal{O}(25n_z^2)$ for the back substitution. Hence the complexity of NHEVI-LU is

$$(50n_z^3 + 25n_z^2) N_{Newton}(col) \quad (56)$$

per column per RK stage. The complexity of this approach can be further reduced by storing the matrix in banded form. For a matrix with bandwidth B the per column complexity of the forward-reduction and back-substitution is

$$10n_z B(B + 1) \quad (57)$$

where $B = 2n_{var}(N_\zeta + 1) \approx 10N_\zeta$. Therefore, the total per column per RK stage complexity of NHEVI-LU is

$$100n_z N_\zeta (10N_\zeta + 1) N_{Newton}(col). \quad (58)$$

Comparing the complexity of NHEVI-LU (58) with NHEVI-GMRES (55), we see that the cost of NHEVI-GMRES scales quadratically with n_z and N_{GMRES} whereas NHEVI-LU scales linearly with n_z and does not require GMRES. So, as the number of vertical levels n_z increases, NHEVI-LU requires fewer FLOPS than NHEVI-GMRES, making NHEVI-LU more efficient.

5.3. LHEVI

The advantage of LHEVI can be described by reconsidering the total per column complexity of NHEVI-LU, which we modify as follows for LHEVI

$$1000n_z N_\zeta^2 \quad (59a)$$

$$100n_z N_\zeta \quad (59b)$$

where (59a) is the cost of the forward-reduction and is only incurred whenever we update the operator Jacobian, while (59b) is the cost of the back-substitution that is incurred per RK stage. Comparing the NHEVI-LU and LHEVI complexity, we see that LHEVI amortizes the cost of constructing the forward-reduction across all stages and numerous time-steps so that we mostly pay the price of the less computationally expensive back-substitution.

5.4. Summary of Complexity

To summarize, the complexity of both NHEVI-LU and LHEVI scale linearly with n_z while NHEVI-GMRES scales quadratically with n_z . To illustrate this behavior, we ran four simulations with n_z ranging from 13 to 97 with each of the three HEVI variants for a short period of time (0.1 days at 104 km horizontal resolution for the baroclinic instability) and only report the simulation time associated with dynamics (and discount initialization time). The results of this study shown in Fig. 4 clearly show the super-quadratic dependence of n_z for NHEVI-GMRES and the approximately linear dependence for NHEVI-LU and LHEVI. In Sec. 7 we discuss a more

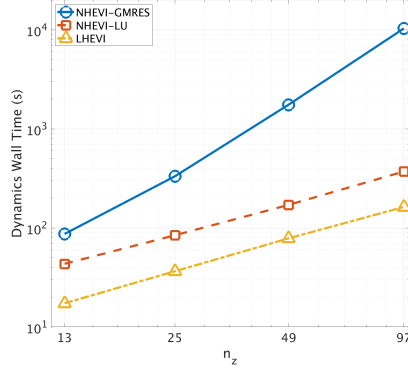


Figure 4: Time-to-solution for 0.1 simulation days as a function of vertical gridpoints n_z .

detailed cost analysis for the three methods. It should be added that increasing n_z has an adverse effect on the time-to-solution for NHEVI-GMRES because satisfying the stopping criterion on the residual \mathbf{res} is dependent on the condition number [56]. This increases the computation cost and also the memory consumption, since more Krylov vectors need to be stored.

6. Time-Integrators

In this section we explore five additive Runge-Kutta (ARK) time-integrators taken from the literature covering a range of orders from second through fifth. Our intent here is not to perform an exhaustive study of IMEX methods but rather to evaluate a representative sample of time-integrators that gave us the fastest time-to-solution. The methods that we showcase below are: ARK2 which is the ARK(2,3,2)b method defined in [16, 15], ARK3 which is the ARK(3,4,3) method from [24], ARS3 which is the ARK(3,4,3) method from [5], ARK4 which is the ARK(5,6,4) from [24], and ARK5 which is the ARK(7,8,5) from [25]. Because ARS3 and ARK5 do not possess a dense (or continuous) output formula (the rest of the methods do), we did not use it for constructing an initial guess for the nonlinear solver; this way we perform a fair comparison across all the methods. Dense output only impacts the performance of the NHEVI methods.

Figure 5 shows the stability region of the five methods we consider. Note that they all have the wedge-shape stability region which confers a larger explicit stability region with increasing implicit time-step size; this is the advantage of these types of time-integrators. The disadvantage is that it is difficult to know *a priori* how diffusive the method is with increasing implicit time-step. Comparing Figs. 3a and 3b, we see that for those methods where the implicit time-step does not alter the stability region of the explicit part, we can always be sure to avoid the diffusive part of the region by pushing the time-step to its maximum explicit component (red region in Fig. 3a). However, this means that these methods will always require a smaller time-step; these are the sorts of trade-offs that need to be considered when choosing a method. In

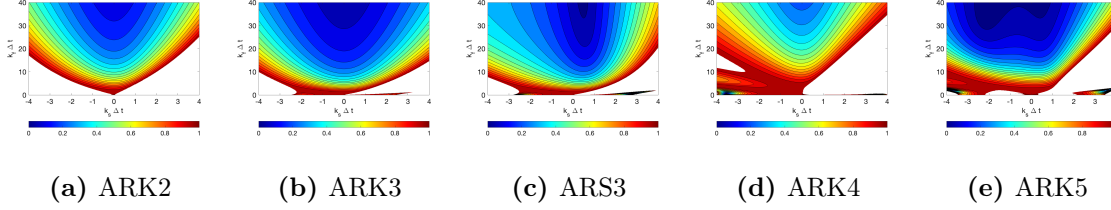


Figure 5: IMEX stability region for implicit fast processes (k_f) versus explicit slow processes (k_s) for a variety of ESDIRK methods.

what follows, we aim to use methods that allow for a faster time-to-solution regardless of the dissipation introduced.

Comparing the performance of the various methods using complexity is possible except for the NHEVI-GMRES method. The reason for this is because NHEVI-GMRES requires outer Newton iterations in addition to inner GMRES iterations; the main complication is in the inner GMRES iterations. In the baroclinic instability test below, we require on average 2-7 Newton iterations per time-step per column and ≈ 45 GMRES iterations per time-step per column (this number includes the iterations for all implicit stages and held constant across all time-integrators). For NHEVI-LU, computing the cost is more straightforward since the cost of the LU decomposition and backsolve is well-defined; this is also true for LHEVI since there are no Newton iterations. Therefore, in LHEVI, the cost of the method is directly proportional to the effective time-step of the method which we define as

$$\Delta t_{eff} = \frac{\Delta t_{max}}{s_{im}}$$

where Δt_{max} is the maximum time-step that the method can use and s_{im} denotes the number of implicit stages. This means that the number of evaluations of the left- and right-hand side functions are on the order

$$\text{Cost}_{LHEVI} = \frac{T_{final}}{\Delta t_{eff}}.$$

To compare whether method 1 is faster than method 2, we compute the ratio

$$R_{m1 \rightarrow m2} = \frac{\Delta t_{eff,m1}}{\Delta t_{eff,m2}} \begin{cases} > 1 & \text{faster} \\ < 1 & \text{slower} \end{cases} . \quad (60)$$

We use these ratios in Sec. 7 to compare the cost of the time-integrators.

7. Numerical Results

In this section, we present the numerical results for two dry test cases on the sphere. We begin with the nonhydrostatic baroclinic instability [57] to discern the performance of all the methods and to gauge the maximum usable time-steps for each time-integrator. Finally, we use the inertia-gravity wave problem [55] to compute convergence rates for the time-integrators.

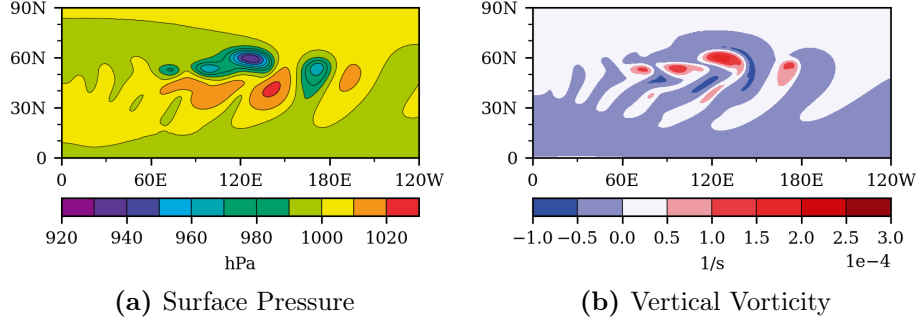


Figure 6: Baroclinic Instability. Set 2NC with ARK2-LHEVI at grid resolution 104×1.25 km resolution at 10 days showing: (a) Surface pressure and (b) vertical vorticity at 850 hPa.

7.1. Baroclinic Instability

The nonhydrostatic baroclinic instability proposed by Ullrich et al. [57] is a useful idealized test case for evaluating global, deep-atmosphere, non-hydrostatic dynamical cores. In particular, this case has been used for previous HEVI studies [12, 58]. The background state for this case is an exact solution to the deep-atmosphere Euler equations that is in both hydrostatic and geostrophic balance [50]. A stream-function perturbation is applied, triggers a baroclinic instability, and wave breaking occurs after day 7.

Figure 6 shows contours for the surface pressure and vertical vorticity at 850 hPa for set 2NC using a grid resolution of 104×1.25 km; the results for the other two equation sets are similar (not shown). The grid resolution corresponds to using polynomial order $N_\xi = N_\eta = N_\zeta = 4$ with number of elements $N_e^\xi = N_e^\eta = 24$ and $N_e^\zeta = 6$. The surface pressure contours shown in Fig. 6a and vertical vorticity shown in Fig. 6b compare favorably with results from the literature [57, 59, 47]. We also show the energy budget and minimum surface pressure for the three equation sets with data at a 6-hour cadence in Fig. 7. Figure 7a shows the deviations in internal, kinetic, potential, and total energy from their initial values and are computed in a similar fashion to the mass loss in (17) and (18) (without normalizing by the initial values). For sets 2NC and 2C, total energy is constructed from the other three components whereas for set 3C, internal energy is diagnosed (since total energy is a prognostic variable). Because the hyperdiffusion applied to the momentum and thermodynamic equations dissipates energy, we do not expect exact conservation for any of the three equation sets; however sets 2NC and, especially, 3C do very well and exhibit low total energy loss. Figure 7b shows that all equation sets yield similar minimum surface pressure for most of the simulation. The results for both the energy budget and minimum surface pressure compare well with results from the literature (see, e.g., [47]). With the verification confirmed, let us now turn to the discussion of the performance of the various HEVI methods.

Figure 8 shows the time-to-solution for the baroclinic instability test at 104×1.25 km resolution for set 2NC on 512 AMD Epyc (Rome) cores. All simulations

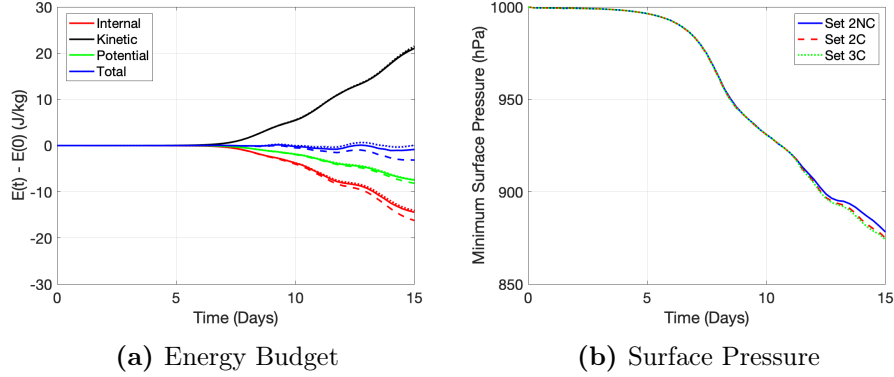


Figure 7: Baroclinic Instability. (a) Energy budget (solid line is 2NC, dashed is 2C, and dotted is 3C) and (b) minimum surface pressure for three equation sets. All simulations use ARK2-LHEVI at grid resolution 104×1.25 km resolution through 15 days.

Time-Integrator	HEVI Method	Δt (secs)	Update Δt	Time-to-Solution (secs)
ARK2	NHEVI-GMRES	135	-	1793
ARS343	NHEVI-GMRES	195	-	1884
ARK4	NHEVI-GMRES	205	-	2919
ARK5	NHEVI-GMRES	190	-	4399
ARK2	NHEVI-LU	135	-	502
ARS343	NHEVI-LU	195	-	515
ARK4	NHEVI-LU	205	-	791
ARK5	NHEVI-LU	190	-	1210
ARK2	LHEVI	135	50	163
ARS343	LHEVI	195	20	167
ARK4	LHEVI	205	50	245
ARK5	LHEVI	190	50	373

Table 2: Baroclinic Instability. Configurations for the timing comparisons in Fig. 8 for set 2NC. NHEVI uses $\epsilon = 0.05$, with Newton tolerance 10^{-5} and GMRES tolerance 10^{-9} .

use the 4th order tensor viscosity with non-dimensional parameters $(c_1, c_2, c_3) = (0.0045, 0.0045, 0.0001)$ with the viscosity computed at each stage of the IMEX method. Each IMEX method uses the optimal configuration that admits the maximum usable time-step. We define the maximum usable time-step as the time-step that (i) yields the expected solution of the test case at 10 days and (ii) remains stable for 100 days. The figure shows that NHEVI-LU is 2x faster than NHEVI-GMRES and that LHEVI is much faster than either NHEVI method; 5x faster than NHEVI-LU and 10x faster than NHEVI-GMRES. In addition, the results show that ARK2 and ARS3 are always the fastest methods.

To help make sense of the time-to-solution for LHEVI, it is convenient to look at the ratio R defined in (60). Comparing ARK2 to the other methods we see that the ratio is $R_{2 \rightarrow 3} = 1.04$, $R_{2 \rightarrow 4} = 1.31$, $R_{2 \rightarrow 5} = 2.48$, where the subscripts 2, 3, 4, 5 denote the order of the ARK methods; this ratio confirms that ARK2 and ARS3

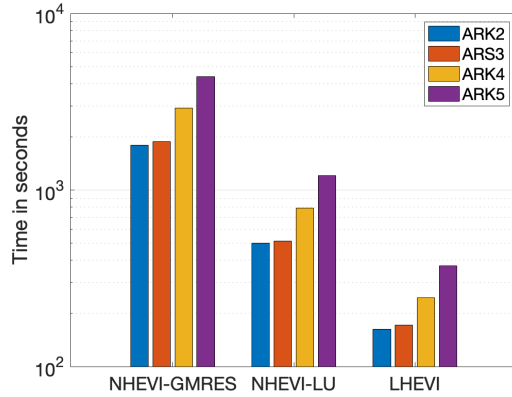


Figure 8: Baroclinic Instability. Time-to-solution (in seconds) for set 2NC for a 10-day simulation at 104×1.25 km resolution. Each IMEX method uses its maximum usable time-step configuration presented in Table 2.

result in a similar performance. The cost of all the methods can be reduced if we only apply hyper-diffusion at the final stage. This reduces both the computational and communication costs substantially in a per time-step basis. We save this option for future work where we shall discuss the role of this strategy within multi-rate HEVI methods (see, e.g., [35]).

Now that we have compared the performance of each of the three HEVI methods and shown that LHEVI is, by far, the most efficient, we next need to quantify the accuracy of LHEVI compared to NHEVI. We could use this test case (baroclinic instability) for the time-convergence study but we choose a different case (inertia-gravity) presented below.

7.2. Inertia-Gravity Wave

This section investigates the accuracy of the proposed LHEVI and NHEVI methods using the inertia-gravity wave from [55] (case 2-#2). We use this test case in keeping with recent papers that use the two-dimensional inertia-gravity wave in a square domain [45, 7, 6, 59]; in our case, we use its extension to three-dimensional global spherical domains suitable for verifying global atmospheric simulations.

The wave propagation is initiated by adding the following potential temperature perturbation defined in spherical coordinates (λ, ϕ, z)

$$\theta' = \Delta\theta f(\lambda, \phi)g(z), \quad (61)$$

where $\Delta\theta = 10$ K is the amplitude of the potential temperature perturbation. The horizontal and vertical distribution functions, $f(\lambda, \phi)$ and $g(z)$, are defined as

$$f(\lambda, \phi) = \begin{cases} \frac{1}{2}(1 + \cos(\pi r/r_{pert})) & r < r_{pert}, \\ 0 & r \geq r_{pert}, \end{cases} \quad (62)$$

$$g(z) = \sin\left(\frac{\pi z}{z_{top}}\right) \quad (63)$$

where

$$r = r_e \cos^{-1} (\sin \phi + \cos \phi_0 \cos (\lambda)) \quad (64)$$

with $z_{top} = 10$ km, $r_{pert} = r_e/3$ km, and λ and ϕ are the longitude and latitude, respectively. The global domain is discretized using 4th-order polynomials in all three directions. The grid resolution in the horizontal and vertical directions are approximately 104 km and 417 m, respectively; this grid resolution corresponds to the same polynomial order and number of elements used in the baroclinic instability. For temporal refinement, we test a series of time-step sizes, $\Delta t = 100 \times 2^n$ seconds, with $n \in [-9, -8, \dots, 0]$. Figure 9 displays the potential temperature perturbation along the equator at the final time $t = 48$ hr, calculated using ARK5-LHEVI. The potential temperature perturbations qualitatively agree with Fig. 5b in [55], indicating that the time-integrator can handle horizontally propagating gravity waves.

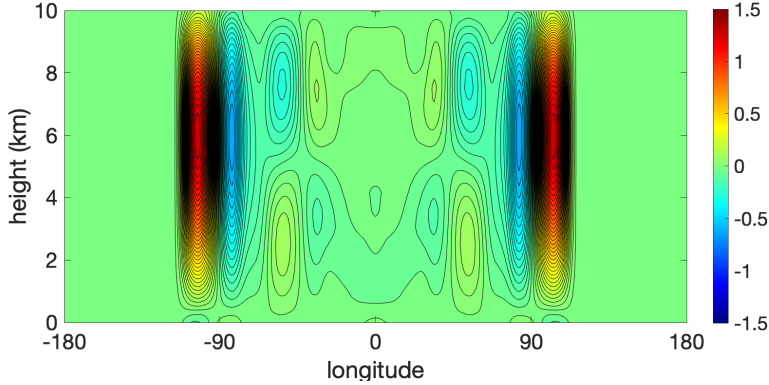


Figure 9: Inertia-gravity wave. Contours of potential temperature perturbation on the equator ($\phi = 0$) at $t = 48$ hours using set 2NC with ARK5-LHEVI.

We assess the accuracy of the method using a normalized L_2 -norm of the relative error in the potential temperature field, which is defined as

$$L_2 \text{ error} = \frac{\sqrt{\int_{\Omega} (\theta - \theta_{true})^2 d\Omega}}{\sqrt{\int_{\Omega} (\theta_{true})^2 d\Omega}}. \quad (65)$$

In this test, we consider the solution obtained using a five-stage fourth-order low-storage RK method [26] with $\Delta t = 100 \times 2^{-12}$ seconds as the true solution and measure the relative errors at $t = 3600$ seconds. Figure 10 shows that the LHEVI and NHEVI methods achieve optimal convergence rates in the L_2 -norm; the dashed lines denote the optimal convergence rates, which have slopes equal to their order on a log-log plot. Results are shown for set 2NC but similar results are obtained with the other two equation sets (not shown). In the figure, the curves from ARK3 and ARS3 overlap, since both produce almost identical results. The optimal convergence rates demonstrate that the formulations of the LHEVI and NHEVI methods are consistent and yield accurate solutions; in fact, the accuracy of the LHEVI and NHEVI solutions

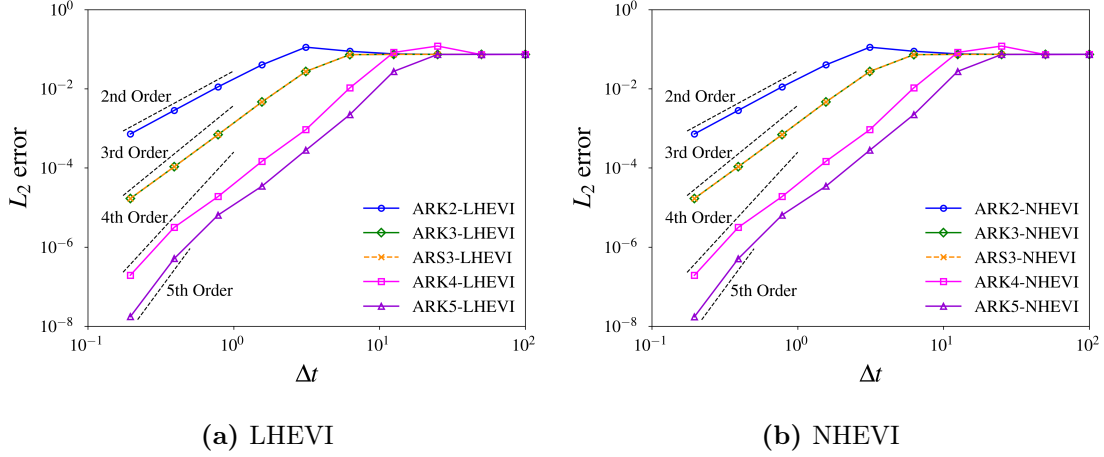


Figure 10: Inertia-gravity wave. Convergence of the normalized L_2 -norm of the relative error with respect to time-step Δt in the potential temperature field for LHEVI and NHEVI with set 2NC.

are almost identical. The plateauing of the error norms for large Δt was discussed in [16] and is due to the acoustic waves being entirely damped.

In Sec. 7.1, we used the tensor-based hyper-diffusion to ensure the long-time stability of the methods. While the artificial hyper-diffusion parameters are designed to automatically scale with the grid and time-step sizes, as shown in (27), for this particular test, we opt for the constant viscosities, $\nu_1 = \nu_2 = 5.0 \times 10^7$ and $\nu_3 = 150$, to eliminate the adaptive effects of time refinement and reach convergence (recall that to recover the physical viscosities requires squaring ν_i).

8. Conclusions

We compared three versions of HEVI methods (NHEVI-GMRES, NHEVI-LU, and LHEVI) using a number of different IMEX methods found in the literature ranging from second to fifth order in accuracy. We confirmed that all methods obtained the expected rate of convergence for LHEVI and NHEVI-LU. The LHEVI method performed the best regarding time-to-solution while achieving similar accuracy and stability characteristics to the NHEVI methods; we confirm these results by timing the simulations but also confirm the numerical results with the complexity analysis for each of the three HEVI methods. We tested the methods on the baroclinic instability on the sphere using 100-day simulations to confirm that the configurations for each HEVI method and all time-integrators were stable for long-time simulations. The baroclinic instability and inertia-gravity wave simulations indicate that the optimal choice of time-integrator is LHEVI with either ARK2 or ARS3, as both methods yield similar time-to-solution and relative L_2 error at their maximum usable time-steps. All time-integrators conserve mass with any choice of equation set; in addition, total energy is conserved up to dissipation due to hyper-diffusion for set 3C. In the future, we will test LHEVI with ARK2 and ARS3 on more stringent flow problems that produce steeper vertical gradients to test whether the same conclusions

hold. However, the results of this study offer new possibilities for constructing yet more efficient time-integrators, e.g., extending the HEVI methods to fully implicit methods in all dimensions that then employ the corresponding version of LHEVI as preconditioners. We will report on this class of methods in the future and test them on more realistic NWP and space weather problems.

Appendix A. Sets 2C and 3C

For completeness, below we include the governing equations for sets 2C and 3C in Appendix A.1, the horizontal and vertical operators in HEVI in Appendix A.2, and the HEVI operator Jacobians in Appendix A.3.

Appendix A.1. Governing Equations

A simple manipulation of (1) results in the conservation form

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{U} = 0 \quad (\text{A.1a})$$

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \left(\frac{\mathbf{U} \otimes \mathbf{U}}{\rho} + P \mathbf{I}_d \right) + \rho \nabla \Phi + 2\omega \hat{\mathbf{r}} \times \mathbf{U} = \mathbf{0} \quad (\text{A.1b})$$

$$\frac{\partial \Theta}{\partial t} + \nabla \cdot \left(\frac{\Theta \mathbf{U}}{\rho} \right) = 0 \quad (\text{A.1c})$$

$$P = P_A \left(\frac{R\Theta}{P_A} \right)^\gamma, \quad (\text{A.1d})$$

where $\mathbf{U} = \rho \mathbf{u}$ is momentum, $\Theta = \rho \theta$ is density potential temperature, and \mathbf{I}_d is the rank- d identity matrix where d is the spatial dimension; with the proper numerics (such as the EBG methods we use in this work) mass and integrated potential temperature are globally conserved. Set 2C, or a variant thereof, is used in, e.g., ICON [62] and MPAS [46].

Set 3C is the most commonly used form of the compressible Euler equations in computational fluid dynamics (CFD) and is written as follows

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{U} = 0 \quad (\text{A.2a})$$

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \left(\frac{\mathbf{U} \otimes \mathbf{U}}{\rho} + P \mathbf{I}_d \right) + \rho \nabla \Phi + 2\omega \hat{\mathbf{r}} \times \mathbf{U} = \mathbf{0} \quad (\text{A.2b})$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \left(\frac{(E + P)\mathbf{U}}{\rho} \right) = 0 \quad (\text{A.2c})$$

$$P = (\gamma - 1) \left(E - \frac{\mathbf{U} \cdot \mathbf{U}}{2\rho} - \rho \Phi \right), \quad (\text{A.2d})$$

where $E = \rho e$ is total energy, with $e = c_v T + \frac{\mathbf{u} \cdot \mathbf{u}}{2} + \Phi$ denoting the total specific energy (internal, kinetic, and potential). This set will conserve both mass and total energy. Unlike sets 2NC and 2C, 3C does not make any assumptions on the composition of the atmosphere and is valid for a whole atmosphere model. Set 3C, to our knowledge, has not been used in any atmospheric model except for NUMA [17, 18], an early version of CLIMA [49], and Atum [59, 48].

Appendix A.2. HEVI Operators

The horizontal and vertical HEVI operators for set 2C are

$$\mathbf{H}(\mathbf{q}) = - \begin{pmatrix} \frac{1}{J} \nabla_H \cdot (J \mathbf{F}_\rho^H) \\ \frac{1}{J} \nabla_H \cdot (J \mathbf{F}_U^H) + \rho \left(\frac{\partial \Phi}{\partial \xi} \nabla \xi + \frac{\partial \Phi}{\partial \eta} \nabla \eta \right) + 2\omega \hat{\boldsymbol{\zeta}} \times \mathbf{U} \\ \frac{1}{J} \nabla_H \cdot (J \mathbf{F}_T^H) \end{pmatrix} + \mathcal{H}_\nu(\mathbf{q}) \quad (\text{A.3})$$

$$\mathbf{V}(\mathbf{q}) = - \begin{pmatrix} \frac{1}{J} \frac{\partial}{\partial \zeta} (J \mathbf{F}_\rho^\zeta) \\ \frac{1}{J} \frac{\partial}{\partial \zeta} (J \mathbf{F}_U^\zeta) + \rho \frac{\partial \Phi}{\partial \zeta} \nabla \zeta \\ \frac{1}{J} \frac{\partial}{\partial \zeta} (J \mathbf{F}_T^\zeta) \end{pmatrix} \quad (\text{A.4})$$

where \mathbf{F}_T denotes the flux for the thermodynamic variable and the contravariant flux components are defined as $\mathbf{F}_q^{\xi, \eta} = (\mathbf{F}_q^\xi, \mathbf{F}_q^\eta, 0)^\top$, \mathbf{F}_q^ζ , for the state vector $\mathbf{q} = (\rho, \mathbf{U}^\top, \Theta)^\top$, with the contravariant fluxes computed as $\mathbf{F}_q^i = \mathbf{F}_q \cdot \nabla \xi^i$. Set 3C has a similar form except that the thermodynamic variable and corresponding flux are different, along with the equation of state (compare Eqs. (A.1) and (A.2)).

Appendix A.3. HEVI Operator Jacobians

For set 2C we arrive at the following operator Jacobian

$$\mathbb{J}_{2C}(\mathbf{q}) = \mathbb{I} + \Lambda \mathbb{D}^\zeta \mathbb{K}_{2C}(\mathbf{q}_j) + \Lambda \mathbb{P}$$

where \mathbb{D}^ζ can be either $\mathbb{D}_{strong}^\zeta$ or \mathbb{D}_{weak}^ζ , denoting the strong or weak forms, respectively. These are defined as follows

$$\left(\mathbb{D}_{strong}^\zeta \right)_{ij} = \frac{J_j}{J_i} D_{ij}^\zeta \quad (\text{A.5a})$$

$$\left(\mathbb{D}_{weak}^\zeta \right)_{ij} = - \frac{(\varpi J)_j}{(\varpi J)_i} D_{ji}^\zeta, \quad (\text{A.5b})$$

where

$$\mathbb{K}_{2C}(\mathbf{q}_j) = \begin{pmatrix} 0 & \zeta_x & \zeta_y & \zeta_z & 0 \\ -\frac{U}{\rho^2} U^\zeta + \zeta_x \frac{\partial P}{\partial \rho} & \frac{1}{\rho} U^\zeta + \zeta_x \left(\frac{U}{\rho} + \frac{\partial P}{\partial U} \right) & \frac{1}{\rho} \zeta_y \frac{U}{\rho} + \zeta_x \frac{\partial P}{\partial V} & \frac{1}{\rho} \zeta_z \frac{U}{\rho} + \zeta_x \frac{\partial P}{\partial W} & \frac{1}{\rho} \zeta_x \frac{\partial P}{\partial E} \\ -\frac{V}{\rho^2} U^\zeta + \zeta_y \frac{\partial P}{\partial \rho} & \frac{1}{\rho} \zeta_x \frac{V}{\rho} + \zeta_y \frac{\partial P}{\partial U} & \frac{1}{\rho} U^\zeta + \zeta_y \left(\frac{V}{\rho} + \frac{\partial P}{\partial V} \right) & \frac{1}{\rho} \zeta_z \frac{V}{\rho} + \zeta_y \frac{\partial P}{\partial W} & \frac{1}{\rho} \zeta_y \frac{\partial P}{\partial E} \\ -\frac{W}{\rho^2} U^\zeta + \zeta_z \frac{\partial P}{\partial \rho} & \frac{1}{\rho} \zeta_x \frac{W}{\rho} + \zeta_z \frac{\partial P}{\partial U} & \frac{1}{\rho} \zeta_y \frac{W}{\rho} + \zeta_z \frac{\partial P}{\partial V} & \frac{1}{\rho} U^\zeta + \zeta_z \left(\frac{W}{\rho} + \frac{\partial P}{\partial W} \right) & \frac{1}{\rho} \zeta_z \frac{\partial P}{\partial E} \\ -\frac{U^\zeta}{\rho^2} & \zeta_x \frac{\Theta}{\rho} & \zeta_y \frac{\Theta}{\rho} & \zeta_z \frac{\Theta}{\rho} & \frac{U^\zeta}{\rho} \end{pmatrix} \quad (\text{A.6})$$

with all terms evaluated at j , and

$$\mathbb{P} = \text{diag} \begin{pmatrix} 0 \\ \zeta_x \frac{\partial \Phi}{\partial \zeta} \\ \zeta_y \frac{\partial \Phi}{\partial \zeta} \\ \zeta_z \frac{\partial \Phi}{\partial \zeta} \\ 0 \end{pmatrix}, \quad (\text{A.7})$$

with all terms evaluated at i .

For set 3C we get

$$\mathbb{J}_{3C}(\mathbf{q}) = \mathbb{I} + \Lambda \mathbb{D}^\zeta \mathbb{K}_{3C}(\mathbf{q}_j) + \Lambda \mathbb{P}$$

with \mathbb{D}^ζ defined as in Eq. (A.5) and \mathbb{P} as in Eq. (A.7), with

$$\mathbb{K}_{3C}(\mathbf{q}_j) = \begin{pmatrix} 0 & \zeta_x & \zeta_y & \zeta_z & 0 \\ -\frac{U}{\rho^2} U^\zeta + \zeta_x \frac{\partial P}{\partial \rho} & \frac{1}{\rho} U^\zeta + \zeta_x \left(\frac{U}{\rho} + \frac{\partial P}{\partial U} \right) & \frac{1}{\rho} \zeta_y \frac{U}{\rho} + \zeta_x \frac{\partial P}{\partial V} & \frac{1}{\rho} \zeta_z \frac{U}{\rho} + \zeta_x \frac{\partial P}{\partial W} & \frac{1}{\rho} \zeta_x \frac{\partial P}{\partial E} \\ -\frac{V}{\rho^2} U^\zeta + \zeta_y \frac{\partial P}{\partial \rho} & \frac{1}{\rho} \zeta_x \frac{V}{\rho} + \zeta_y \frac{\partial P}{\partial U} & \frac{1}{\rho} U^\zeta + \zeta_y \left(\frac{V}{\rho} + \frac{\partial P}{\partial V} \right) & \frac{1}{\rho} \zeta_z \frac{V}{\rho} + \zeta_y \frac{\partial P}{\partial W} & \frac{1}{\rho} \zeta_y \frac{\partial P}{\partial E} \\ -\frac{W}{\rho^2} U^\zeta + \zeta_z \frac{\partial P}{\partial \rho} & \frac{1}{\rho} \zeta_x \frac{W}{\rho} + \zeta_z \frac{\partial P}{\partial U} & \frac{1}{\rho} \zeta_y \frac{W}{\rho} + \zeta_z \frac{\partial P}{\partial V} & \frac{1}{\rho} U^\zeta + \zeta_z \left(\frac{W}{\rho} + \frac{\partial P}{\partial W} \right) & \frac{1}{\rho} \zeta_z \frac{\partial P}{\partial E} \\ U^\zeta \left(\frac{1}{\rho} \frac{\partial P}{\partial \rho} - \frac{E+P}{\rho^2} \right) & \frac{U^\zeta}{\rho} \frac{\partial P}{\partial U} + \frac{E+P}{\rho} \zeta_x & \frac{U^\zeta}{\rho} \frac{\partial P}{\partial V} + \frac{E+P}{\rho} \zeta_y & \frac{U^\zeta}{\rho} \frac{\partial P}{\partial W} + \frac{E+P}{\rho} \zeta_z & \frac{U^\zeta}{\rho} \left(1 + \frac{\partial P}{\partial E} \right) \end{pmatrix}. \quad (\text{A.8})$$

with all terms evaluated at j .

Upon differentiation by \mathbb{D}^ζ , numerical integration (multiplying by $\varpi_i J_i$), and applying DSS along ζ , results in the global Jacobian matrices $\mathcal{G}(\mathbb{J}) \in \mathbb{R}^{M \times M}$, where $M = n_{var} N_e^\zeta (N_\zeta + 1)$, and \mathcal{G} represents the operator that takes \mathbb{J} and constructs its global representation, as presented in Sec. 4.2.3.

Acknowledgments

Felipe Alves was funded by the Defense Sciences Office of the Defense Applied Research and Projects Agency (DARPA DSO) through the Space Environment Exploitation (SEE) program. F. X. Giraldo, Soonpil Kang, James F. Kelly, and P. Alex Reinecke gratefully acknowledge the support of the Office of Naval Research under grant # N0001419WX00721. F. X. Giraldo was also supported by the National Science Foundation under grant AGS-1835881. This work was performed when Felipe Alves and Soonpil Kang held National Academy of Sciences' National Research Council (NRC) Fellowships at the Naval Postgraduate School.

References

- [1] D. ABDI, F. GIRALDO, E. CONSTANTINESCU, L. CARR, L. WILCOX, AND T. WARBURTON, *Acceleration of the implicit-explicit nonhydrostatic unified model of the atmosphere on manycore processors*, The International Journal of High Performance Computing Applications, 33 (2017), pp. 242–267.
- [2] D. ABDI AND F. X. GIRALDO, *Efficient construction of unified continuous and discontinuous Galerkin formulations for the 3D Euler Equations*, Journal of Computational Physics, 320 (2016), pp. 46–68.
- [3] R. AKMAEV, *Whole atmosphere modeling: Connecting terrestrial and space weather*, Reviews of Geophysics, 49 (2011).
- [4] E. ANDERSON, Z. BAI, C. BISCHOF, L. S. BLACKFORD, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, AND D. SORENSEN, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, third ed., 1999.

- [5] 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.
- [6] M. BALDAUF, *A horizontally explicit, vertically implicit (HEVI) discontinuous Galerkin scheme for the 2-dimensional euler and Navier-Stokes equations using terrain-following coordinates*, Journal of Computational Physics, 446 (2021).
- [7] M. BALDAUF AND S. BRDAR, *3d diffusion in terrain-following coordinates: testing and stability of horizontally explicit, vertically implicit discretizations*, Quarterly Journal of the Royal Meteorological Society, 142 (2016), pp. 2087–2101.
- [8] L. BAO, R. KLÖFKORN, AND R. NAIR, *Horizontally explicit and vertically implicit (HEVI) time discretization scheme for a discontinuous Galerkin nonhydrostatic model*, Monthly Weather Review, 143 (2015), pp. 972–990.
- [9] R. COURANT, K. FRIEDRICHS, AND H. LEWY, *On the partial difference equations of mathematical physics*, IBM Journal: translation from the original paper in Mathematische Annale, 100,32-24, 100 (1928), pp. 215–234.
- [10] M. DEVILLE, P. FISCHER, AND E. MUND, *High-Order Methods for Incompressible Flow*, Cambridge University Press, New York, 2002.
- [11] T. GALCHEN AND R. SOMERVILLE, *Use of a coordinate transformation for solution of Navier-Stokes equations*, Journal of Computational Physics, 17 (1975), pp. 209–228.
- [12] D. J. GARDNER, J. E. GUERRA, F. P. HAMON, D. R. REYNOLDS, P. A. ULLRICH, AND C. S. WOODWARD, *Implicit-explicit (IMEX) Runge-Kutta methods for non-hydrostatic atmospheric models*, Geoscientific Model Development, 11 (2018), pp. 1497–1515.
- [13] G. J. GASSNER, A. R. WINTERS, AND D. A. KOPRIVA, *Split form nodal discontinuous Galerkin schemes with summation-by-parts property for the compressible Euler equations*, Journal of Computational Physics, 327 (2016), pp. 39–66.
- [14] F. X. GIRALDO, *Trajectory calculations for spherical geodesic grids in Cartesian space*, Monthly Weather Review, 127 (1999), pp. 1651–1662.
- [15] —, *An Introduction to Element-based Galerkin Methods on Tensor-Product Bases - Analysis, Algorithms, and Applications*, vol. 24, Springer, 2020.
- [16] F. X. GIRALDO, J. F. KELLY, AND E. M. CONSTANTINESCU, *Implicit-explicit formulations of a three-dimensional nonhydrostatic unified model of the atmosphere (NUMA)*, SIAM Journal on Scientific Computing, 35 (2013), pp. B1162–B1194.

- [17] 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*, Journal of Computational Physics, 227 (2008), pp. 3849–3877.
- [18] —, *High-order semi-implicit time-integrators for a triangular discontinuous Galerkin oceanic shallow water model*, International Journal for Numerical Methods in Fluids, 63 (2010), pp. 1077–1102.
- [19] F. X. GIRALDO, M. RESTELLI, AND M. LÄUTER, *Semi-implicit formulations of the Navier–Stokes equations: application to nonhydrostatic atmospheric modeling*, SIAM Journal on Scientific Computing, 32 (2010), pp. 3394–3425.
- [20] O. GUBA, M. A. TAYLOR, A. M. BRADLEY, P. A. BOSLER, AND A. STEYER, *A framework to evaluate IMEX schemes for atmospheric models*, Geoscientific Model Development, 13 (2020), pp. 6467–6480.
- [21] O. GUBA, M. A. TAYLOR, P. A. ULLRICH, J. R. OVERFELT, AND M. N. LEVY, *The spectral element method (SEM) on variable-resolution grids: evaluating grid sensitivity and resolution-aware numerical viscosity*, Geoscientific Model Development, 7 (2014), pp. 2803–2816.
- [22] J. F. KELLY, F. A. V. D. B. A. ALVES, J. T. EMMERT, S. D. ECKERMANN, F. X. GIRALDO, AND P. A. REINECKE, *A nonhydrostatic and mass conserving ground-to-thermosphere dynamical core based on specific internal energy*, (in preparation), 1 (2023), pp. 1–X.
- [23] 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.
- [24] C. A. KENNEDY AND M. H. CARPENTER, *Additive Runge–Kutta schemes for convection–diffusion–reaction equations*, Applied Numerical Mathematics, 44 (2003), pp. 139–181.
- [25] C. A. KENNEDY AND M. H. CARPENTER, *Higher-order additive Runge–Kutta schemes for ordinary differential equations*, Applied Numerical Mathematics, 136 (2019), pp. 183–205.
- [26] C. A. KENNEDY, M. H. CARPENTER, AND R. M. LEWIS, *Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations*, Applied Numerical Mathematics, 35 (2000), pp. 177–219.
- [27] Y.-J. KIM, F. X. GIRALDO, M. FLATAU, C.-S. LIOU, AND M. S. PENG, *A sensitivity study of the Kelvin wave and the Madden-Julian Oscillation in aquaplanet simulations by the Naval Research Laboratory Spectral Element Atmospheric Model*, Journal of Geophysical Research-Atmospheres, 113 (2008).

- [28] J. B. KLEMP, W. C. SKAMAROCK, AND S. HA, *Damping acoustic modes in compressible horizontally explicit vertically implicit (HEVI) and split-explicit time integration schemes*, Monthly Weather Review, 146 (2018), pp. 1911–1923.
- [29] D. KNOLL AND D. KEYES, *Jacobian-free Newton-Krylov methods: a survey of approaches and applications*, Journal of Computational Physics, 193 (2004), pp. 357–397.
- [30] D. KOPRIVA, *Metric identities and the discontinuous spectral element method on curvilinear meshes*, Journal of Scientific Computing, 26 (2006), pp. 301–327.
- [31] D. KOPRIVA, *Implementing Spectral Methods for Partial Differential Equations*, Springer, New York, 2009.
- [32] D. LEE AND A. PALHA, *Exact spatial and temporal balance of energy exchanges within a horizontally explicit/vertically implicit non-hydrostatic atmosphere*, Journal of Computational Physics, 440 (2021).
- [33] S.-J. LOCK, N. WOOD, AND H. WELLER, *Numerical analyses of Runge–Kutta implicit–explicit schemes for horizontally explicit, vertically implicit solutions of atmospheric models*, Quarterly Journal of the Royal Meteorological Society, 140 (2014), pp. 1654–1669.
- [34] S. MARRAS, J. F. KELLY, M. MORAGUES, A. MÜLLER, M. A. KOPERA, M. VAZQUEZ, F. X. GIRALDO, G. HOUZEAUX, AND O. JORBA, *A review of element-based galerkin methods for numerical weather prediction: finite elements, spectral elements, and discontinuous Galerkin*, Archives of Computational Methods in Engineering, (2015), pp. 1–50.
- [35] P. R. MUGG, *Extrapolated Multirate Methods for Hyperbolic Partial Differential Equations*, PhD thesis, Naval Postgraduate School, Monterey, CA, June 2021.
- [36] A. MÜLLER, M. KOPERA, S. MARRAS, L. WILCOX, T. ISAAC, AND F. GIRALDO, *Strong scaling for numerical weather prediction at petascale with the atmospheric model numa*, The International Journal of High Performance Computing Applications, 0 (2018), p. 1094342018763966.
- [37] 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.
- [38] A. T. PATERA, *A spectral method for fluid dynamics: Laminar flow in a channel expansion*, Journal of Computational Physics, 54 (1984), pp. 468–488.
- [39] P. A. REINECKE, K. VINER, J. D. DOYLE, S. GABERSEK, M. MARTINI, J. MICHALAKES, D. RYGLICKI, D. FLAGG, AND F. X. GIRALDO, *Development and testing of a next generation spectral element model for the U.S. Navy*, in 17th Workshop on High Performance Computing (ECMWF), 24–28 October, ECMWF, Reading UK, 2016.

- [40] J. REISNER, A. WYSZOGRODZKI, V. MOUSSEAU, AND D. KNOLL, *An efficient physics-based preconditioner for the fully implicit solution of small-scale thermally driven atmospheric flows*, Journal of Computational Physics, 189 (2003), pp. 30–44.
- [41] C. RONCHI, R. IACONO, AND P. PAOLUCCI, *The ‘cubed sphere’: A new method for the solution of partial differential equations in spherical geometry*, Journal of Computational Physics, 124 (1996), pp. 93–114.
- [42] H. H. ROSENBROCK, *Some general implicit processes for the numerical solution of differential equations*, The Computer Journal, 5 (1963), pp. 329–330.
- [43] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, 1996.
- [44] M. SATOH, *Conservative scheme for the compressible nonhydrostatic models with the horizontally explicit and vertically implicit time integration scheme*, Monthly Weather Review, 130 (2002), pp. 1227–1245.
- [45] W. SKAMAROCK AND J. KLEMP, *Efficiency and accuracy of the Klemp–Wilhelmson time-splitting technique*, Monthly Weather Review, 122 (1994), pp. 2623–2630.
- [46] W. C. SKAMAROCK, J. B. KLEMP, M. G. DUDA, L. D. FOWLER, S.-H. PARK, AND T. D. RINGLER, *A multiscale nonhydrostatic atmospheric model using centroidal Voronoi tessellations and C-grid staggering*, Monthly Weather Review, 140 (2012), pp. 3090–3105.
- [47] W. C. SKAMAROCK, H. ONG, AND J. B. KLEMP, *A fully compressible nonhydrostatic deep-atmosphere equations solver for mpas*, Monthly Weather Review, 149 (2021), pp. 571–583.
- [48] A. N. SOUZA, J. HE, T. BISCHOFF, M. WARUSZEWSKI, L. NOVAK, V. BARRA, T. GIBSON, A. SRIDHAR, S. KANDALA, S. BYRNE, L. C. WILCOX, J. E. KOZDON, F. X. GIRALDO, O. KNOTH, J. MARSHALL, R. FERRARI, AND T. SCHNEIDER, *The flux-differencing discontinuous Galerkin method applied to an idealized fully compressible nonhydrostatic dry atmosphere*, Journal of Advances in Modeling Earth Systems, 15 (2023), p. e2022MS003527.
- [49] A. SRIDHAR, Y. TISSAOUI, S. MARRAS, Z. SHEN, C. KAWCZYNSKI, S. BYRNE, K. PAMNANY, M. WARUSZEWSKI, T. H. GIBSON, J. E. KOZDON, V. CHURAVY, L. C. WILCOX, F. X. GIRALDO, AND T. SCHNEIDER, *Large-eddy simulations with ClimateMachine v0.2.0: a new open-source code for atmospheric simulations on gpus and cpus*, Geoscientific Model Development, 15 (2022), pp. 6259–6284.
- [50] A. STANFORTH AND A. WHITE, *Further non-separable exact solutions of the deep-and shallow-atmosphere equations*, Atmospheric Science Letters, 12 (2011), pp. 356–361.

- [51] A. STEYER, C. J. VOGL, M. TAYLOR, AND O. GUBA, *Efficient IMEX Runge-Kutta methods for nonhydrostatic dynamics*, 2019. arXiv:1906.07219.
- [52] M. A. TAYLOR AND A. FOURNIER, *A compatible and conservative spectral element method on unstructured grids*, Journal of Computational Physics, 229 (2010), pp. 5879–5895.
- [53] M. A. TAYLOR, O. GUBA, A. STEYER, P. A. ULLRICH, D. M. HALL, AND C. ELDRED, *An energy consistent discretization of the nonhydrostatic equations in primitive variables*, Journal of Advances in Modeling Earth Systems, 12 (2020), p. e2019MS001783. e2019MS001783 10.1029/2019MS001783.
- [54] J. THUBURN, *Some conservation issues for the dynamical cores of nwp and climate models*, Journal of Computational Physics, 227 (2008), pp. 3715–3730.
- [55] 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.
- [56] L. TREFETHEN AND D. BAU III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- [57] P. A. ULLRICH, T. MELVIN, C. JABLONOWSKI, AND A. STANFORTH, *A proposed baroclinic wave test case for deep- and shallow-atmosphere dynamical cores*, Quarterly Journal of the Royal Meteorological Society, 140 (2014), pp. 1590–1602.
- [58] C. J. VOGL, A. STEYER, D. R. REYNOLDS, P. A. ULLRICH, AND C. S. WOODWARD, *Evaluation of implicit-explicit additive Runge-Kutta integrators for the HOMME-NH dynamical core*, Journal of Advances in Modeling Earth Systems, 11 (2019), pp. 4228–4244.
- [59] M. WARUSZEWSKI, T. H. GIBSON, L. C. WILCOX, J. E. KOZDON, AND F. X. GIRALDO, *Entropy stability discontinuous Galerkin methods for balance laws in non-conservative form. Applications to Euler with gravity*, Journal of Computational Physics, 468 (2022), p. 111507.
- [60] H. WELLER, S.-J. LOCK, AND N. WOOD, *Runge-Kutta IMEX schemes for the horizontally explicit/vertically implicit (HEVI) solution of wave equations*, Journal of Computational Physics, 252 (2013), pp. 365–381.
- [61] A. A. WHITE, B. J. HOSKINS, I. ROULSTONE, AND A. STANFORTH, *Consistent approximate models of the global atmosphere: shallow, deep, hydrostatic, quasi-hydrostatic and non-hydrostatic*, Quarterly Journal of the Royal Meteorological Society, 131 (2005), pp. 2081–2107.
- [62] G. ZÄNGL, D. REINERT, P. RÍPODAS, AND M. BALDAUF, *The ICON (ICOsa-hedral Non-hydrostatic) modelling framework of DWD and MPI-M: Description*

of the non-hydrostatic dynamical core, Quarterly Journal of the Royal Meteorological Society, 141 (2015), pp. 563–579.