JAMES | Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1029/2025MS005014

Special Collection:
The CliMA Earth System Model

The Climate Modeling Alliance Atmosphere Dynamical Core: Concepts, Numerics, and Scaling

Dennis Yatunin[1], Simon Byrne[1,2], Charles Kawczynski[1], Sriharsha Kandala[1], Gabriele Bozzola[1], Akshay Sridhar[1], Zhaoyi Shen[1], Anna Jaruga[1], Julia Sloan[1], Jia He[1,3], Daniel Zhengyu Huang[1,4], Valeria Barra[1,5], Ray Chew[1], Anudhyan Boral[6], Yi-fan Chen[6], Oswald Knoth[1,7], Paul Ullrich[8,9], Cheikh Mbengue[1], and Tapio Schneider[1,6]

[1]California Institute of Technology, Pasadena, CA, USA, [2]Now at NVIDIA, Santa Clara, CA, USA, [3]Now at NextEra Analytics, St Paul, MN, USA, [4]Now at Beijing International Center for Mathematical Research, Center for Machine Learning Research, Peking University, Beijing, China, [5]Now at Department of Mathematics and Statistics, San Diego State University, San Diego, CA, USA, [6]Google LLC, Mountain View, CA, USA, [7]Leibniz-Institut für Troposphärenforschung, Leibniz, Germany, [8]Department of Land, Air and Water Resources, University of California, Davis, CA, USA, [9]Physical and Life Sciences Directorate, Lawrence Livermore National Laboratory, Livermore, CA, USA

Abstract This paper presents the dynamical core of the Climate Modeling Alliance (CliMA) atmosphere model, designed for efficient simulation of a wide range of atmospheric flows across scales. The core uses the nonhydrostatic equations of motion for a deep atmosphere, discretized with a hybrid approach that combines a spectral element method (SEM) in the horizontal and a staggered finite-difference method in a height-based, terrain-following coordinate in the vertical. This approach leverages the high-order accuracy and scalability of the SEM, while maintaining the computational efficiency and stability of finite differences on a staggered grid. The model's coordinate-independent equation set allows for simulations in a variety of geometries and planetary configurations. The use of the specific total energy of moist air as a prognostic variable, along with a consistent thermodynamic formulation, ensures the conservation of energy, air mass, and water mass, even in moist atmospheres and in the presence of subgrid-scale parameterizations, without ad hoc fixers. A horizontally explicit, vertically implicit (HEVI) timestepping strategy treats fast vertical processes implicitly and further enhances computational efficiency by allowing larger timesteps. The model demonstrates excellent strong and weak scaling on CPUs and GPUs, making it well-suited for high-resolution simulations on modern supercomputing architectures, including those on the cloud, which widens access to climate models.

Plain Language Summary We have developed a new model to simulate Earth's and other planet's atmospheres accurately and efficiently. The model can be used to predict weather and study climate change on scales from kilometers to hundreds of kilometers. The model's physical formulation and discretization are designed to ensure that important properties such as the atmosphere's total energy and mass are conserved, which is crucial for preventing model drift in long-term climate studies. It is also designed to work well on different types of computer architectures, including graphics processing units (GPUs) widely available for cloud computing. This democratizes access to core climate modeling tools.

Key Points:

- The paper introduces a nonhydrostatic dynamical core with consistent moist thermodynamics and total moist energy as prognostic variable
- The model uses a hybrid spectral element/finite difference discretization that conserves mass, total moist energy, and water
- Excellent CPU/GPU scaling makes the model suitable for cloud computing

Correspondence to:
T. Schneider,
tapio@caltech.edu

Citation:
Yatunin, D., Byrne, S., Kawczynski, C., Kandala, S., Bozzola, G., Sridhar, A., et al. (2026). The climate modeling alliance atmosphere dynamical core: Concepts, numerics, and scaling. *Journal of Advances in Modeling Earth Systems*, *18*, e2025MS005014. https://doi.org/10.1029/2025MS005014

Received 10 FEB 2025
Accepted 8 DEC 2025

Author Contributions:
Conceptualization: Tapio Schneider
Funding acquisition: Tapio Schneider
Methodology: Dennis Yatunin, Simon Byrne, Oswald Knoth, Paul Ullrich, Tapio Schneider
Project administration: Cheikh Mbengue, Tapio Schneider
Software: Dennis Yatunin, Simon Byrne, Charles Kawczynski, Sriharsha Kandala, Gabriele Bozzola, Zhaoyi Shen, Anna Jaruga, Julia Sloan, Jia He, Daniel Zhengyu Huang, Valeria Barra, Ray Chew, Anudhyan Boral, Yi-fan Chen, Oswald Knoth
Supervision: Tapio Schneider

1. Introduction

Dynamical cores are the engines of climate models. They simulate the motions of the atmosphere and other climate system components. The Climate Modeling Alliance (CliMA) has developed a new dynamical core for its atmosphere model, drawing on established numerical methods while incorporating innovative features. Implemented in Julia, a high-level programming language that facilitates performance portability and integration of machine learning components (e.g., Charbonneau et al., 2025; Christopoulos et al., 2024), the core is designed to be versatile, robust, and computationally efficient. The core's development was guided by several objectives: (a) adherence to a consistent and accurate thermodynamic framework, (b) flexibility for diverse applications, from large-eddy simulations to global climate simulations, including of extraterrestrial atmospheres, (c) conservation of energy, mass, and water, even in deep atmospheres, and (d) performance portability and scalability across CPU and GPU computing platforms. The core's modular design allows for the construction of other climate model components on top of the same computational infrastructure, as exemplified by its shared use in the CliMA land model (Deck et al., 2026).

Validation: Dennis Yatunin, Sriharsha Kandala, Akshay Sridhar, Zhaoyi Shen, Anna Jaruga, Julia Sloan, Jia He, Valeria Barra, Ray Chew, Oswald Knoth
Visualization: Dennis Yatunin, Akshay Sridhar, Zhaoyi Shen, Anna Jaruga, Julia Sloan, Jia He, Valeria Barra, Ray Chew, Oswald Knoth
Writing – original draft: Dennis Yatunin, Tapio Schneider
Writing – review & editing: Dennis Yatunin, Simon Byrne, Sriharsha Kandala, Akshay Sridhar, Zhaoyi Shen, Anna Jaruga, Paul Ullrich, Tapio Schneider

Numerically, the CliMA dynamical core employs the mimetic spectral element method (SEM) horizontally (Guba et al., 2014; M. A. Taylor & Fournier, 2010; M. A. Taylor et al., 2020), which has proven scalability on high-performance computing platforms, including GPUs (M. Taylor et al., 2023). The mimetic SEM has desirable conservation properties for idealized, adiabatic dynamics (M. A. Taylor et al., 2020); however, this conservation is not guaranteed when moist processes or subgrid-scale parameterizations such as hyperdiffusion are introduced, often necessitating energy fixers we seek to avoid. The CliMA dynamical core uses finite differences vertically, with a staggered grid to suppress computational modes and with a horizontally explicit and vertically implicit (HEVI) timestepping strategy (Colavolpe et al., 2017; Gardner et al., 2018; Thuburn & Woollings, 2005; Weller et al., 2013). The core uses the total energy of moist air as a prognostic variable, similar to what is discussed in Satoh (2003) and Satoh et al. (2008), enabling conservation of energy, air mass, and water mass even in the presence of moist processes. This design choice avoids the need for ad hoc fixers that are common in other atmosphere models and are viewed as challenging to remove (Jablonowski & Williamson, 2011; P. H. Lauritzen et al., 2022). The coordinate-independent, fully compressible equation set allows for simulations in various geometries, such as Cartesian geometry for large-eddy or cloud-resolving simulations and spherical geometry in global climate simulations. Additionally, the model employs an internally consistent set of thermodynamic approximations that integrates insights from previous studies (Bowen & Thuburn, 2022a, 2022b; Duarte et al., 2014; Romps, 2008) and is based on the concept of a calorically perfect fluid, that is, a fluid with constant specific heat capacities.

This paper describes the conceptual foundations of the dynamical core, its discretization strategies, and its performance in standard benchmark tests (Hughes & Jablonowski, 2023; Ullrich et al., 2014, 2016). Subsequent companion papers will describe the parameterizations used in the atmosphere model, examine the quality of climate simulations, and discuss a hierarchy of atmosphere models built using this core.

The structure of this paper is as follows. Section 2 reviews the thermodynamic formulation. Section 3 presents the dynamical equations in coordinate-independent form and discusses their properties where they extend beyond what is standard in atmosphere models. Section 4 summarizes the equations in a generalized coordinate system that accommodates terrain-following coordinates and is adaptable to both Cartesian and spherical domains. Section 5 details the discretization approach, building on recent advances in spectral element methods and HEVI timestepping. Section 6 presents benchmark test cases and demonstrates energy, air mass, and water conservation to within floating point precision. Section 7 illustrates the computational performance, including strong and weak scaling, of the dynamical core on CPUs and GPUs. Section 8 summarizes the conclusions. Additional technical aspects are provided in the appendices.

2. Thermodynamics

2.1. Working Fluid and Equation of State

Using thermodynamically consistent approximations, CliMA's atmospheric dynamical core simulates the dynamics and thermodynamics of moist air, a mixture of dry air, water vapor (both treated as ideal gases), and condensed water, both suspended in clouds and falling as precipitation. We assume that all constituents of the working fluid are in thermal equilibrium, maintaining the same temperature. While condensates may sediment or fall, they are assumed to remain in thermal equilibrium with the air. This means that we neglect temperature differences between the condensate and ambient air, for example, due to evaporative cooling of raindrops (Roy et al., 2023), when calculating fluid energetics (though not necessarily microphysical process rates). However, condensates are not required to be in full thermodynamic equilibrium: we allow for out-of-equilibrium phases such as supercooled liquid (cf. Bowen & Thuburn, 2022a, 2022b). By including both suspended cloud condensate and precipitation in the working fluid, we facilitate the use of microphysics schemes that represent these components through properties (e.g., moments) of a continuous particle size distribution (e.g., Hu & Igel, 2024).

The moist air density ρ contains contributions from all constituents, represented by mass fractions: dry air (q_d), water vapor (q_v), liquid water (q_l), and ice (q_i). We define the condensate mass fraction as $q_c = q_l + q_i$, and the total specific humidity as $q_t = q_v + q_c$. Since these account for all components of the working fluid, $q_t + q_d = 1$. Importantly, the model does not assume small specific humidities, enabling its application to planetary atmospheres where a condensable species, such as CO_2 on Mars, is a major atmospheric constituent and its condensation can substantially affect air mass and dynamics (e.g., Soto et al., 2015).

Table 1
Thermodynamic Constants

Symbol	Definition	Value
R_d	Gas constant of dry air	287.0 J (kg K)$^{-1}$
R_v	Gas constant of water vapor	461.5 J (kg K)$^{-1}$
c_{pd}	Isobaric specific heat of dry air	1004.5 J (kg K)$^{-1}$
c_{pv}	Isobaric specific heat of water vapor	1859.0 J (kg K)$^{-1}$
c_{pl}	Isobaric specific heat of liquid water	4181.0 J (kg K)$^{-1}$
c_{pi}	Isobaric specific heat of ice	2070.0 J (kg K)$^{-1}$
T_0	Reference temperature	273.16 K
$L_{v,0}$	Latent heat of vaporization	2.5008×10^6 J kg^{-1}
$L_{f,0}$	Latent heat of fusion	0.3336×10^6 J kg^{-1}
p_{tr}	Triple point vapor pressure	611.657 Pa
p_0	Reference pressure	10^5 Pa
T_{tr}	Triple point temperature	273.16 K

In many bulk microphysics schemes, precipitation is treated as a separate category of condensed water mass. In these cases, the liquid and ice specific humidities are further divided into contributions from suspended cloud condensate (q_l^{cl} and q_i^{cl}) and precipitation (q_l^{pr} and q_i^{pr}), so that $q_l = q_l^{\mathrm{cl}} + q_l^{\mathrm{pr}}$ and $q_i = q_i^{\mathrm{cl}} + q_i^{\mathrm{pr}}$. We will allow for that distinction in the equations of motion.

The pressure p of the working fluid is the sum of the partial pressures of dry air and water vapor, expressed as $p = \rho\left(R_d q_d + R_v q_v\right) T$. Here, R_d and R_v are the specific gas constants of dry air and water vapor, respectively. We have neglected the volume, but not the masses, of the condensed phases because the specific volume of the condensates is a factor 10^3 lower than that of the gas phases. The equation of state can then be written as

$$p = \rho R_m T, \tag{1}$$

where

$$R_m(q) = R_d\left(1 - q_t\right) + R_v q_v = R_d\left[1 + \left(\epsilon_{dv} - 1\right) q_t - \epsilon_{dv} q_c\right] \tag{2}$$

is the specific mixture gas constant of moist air, a variable dependent on the specific humidities $q = \left(q_t, q_l, q_i\right)$ and the ratio of the specific gas constants of water vapor and dry air, $\epsilon_{dv} = R_v / R_d$ (≈1.61). Table 1 lists the thermodynamic constants we use.

2.2. Heat Capacities

Our primary thermodynamic assumption is that the fluid is calorically perfect. This means that the isochoric specific heat capacities of the moist air constituents—dry air (c_{vd}), water vapor (c_{vv}), liquid water (c_{vl}), and ice (c_{vi})—are considered constant. Consequently, the isobaric specific heat capacities of these constituents—dry air ($c_{pd} = c_{vd} + R_d$), water vapor ($c_{pv} = c_{vv} + R_v$), liquid water ($c_{pl} = c_{vl}$), and ice ($c_{pi} = c_{vi}$)—are also constant. In reality, specific heat capacities vary slightly with temperature, but for atmospheric conditions, treating them as constant introduces an error of less than 1% for dry air, the major component of moist air, and at most of a few percent for the water phases.

The corresponding specific heat capacities of moist air then are the weighted sums of those of the constituents:

$$c_{\cdot m}(q) = \left(1 - q_t\right) c_{\cdot d} + q_v c_{\cdot v} + q_l c_{\cdot l} + q_i c_{\cdot i} \tag{3}$$

$$= c_{\cdot d} + \left(c_{\cdot v} - c_{\cdot d}\right) q_t + \left(c_{\cdot l} - c_{\cdot v}\right) q_l + \left(c_{\cdot i} - c_{\cdot v}\right) q_i \tag{4}$$

where $\cdot$ stands for v or p and we have used $q_v = q_t - q_l - q_i$. Substitution shows that the standard relation between the specific heat capacities of the constituents also holds for moist air as a whole:

$$c_{pm}(q) = c_{vm}(q) + R_m(q). \tag{5}$$

2.3. Latent Heats

Kirchhoff's relation states that the specific latent heat (enthalpy) L for phase changes varies with temperature T as

$$\frac{dL}{dT} = \Delta c_p, \tag{6}$$

where Δc_p is the difference in isobaric specific heat capacities between phases. Assuming constant specific heat capacities, this integrates to

$$L(T) = L_0 + \Delta c_p (T - T_0), \tag{7}$$

with T_0 as a reference temperature and L_0 as the specific latent heat at T_0. Thus, in a calorically perfect fluid, specific latent heats depend linearly on temperature. The model's energetics are invariant under shifts of the reference temperature; however, the quality of the linearization in Equation 7 depends on this choice, suggesting that T_0 should be chosen to be a value typical for the atmosphere (Ambaum, 2020). We choose the triple point temperature of water as the reference.

For the water phase transitions, this implies specifically:

1. $L_v(T) = L_{v,0} + (c_{pv} - c_{pl})(T - T_0)$: Latent heat of vaporization;
2. $L_f(T) = L_{f,0} + (c_{pl} - c_{pi})(T - T_0)$: Latent heat of fusion;
3. $L_s(T) = L_{s,0} + (c_{pv} - c_{pi})(T - T_0)$: Latent heat of sublimation.

Given $L_{s,0} = L_{v,0} + L_{f,0}$, it follows that $L_s(T) = L_v(T) + L_f(T)$, as expected.

2.4. Internal Energies

The specific internal energies of the moist air constituents are (Romps, 2008).

$$I_d(T) = c_{vd}(T - T_0) - R_d T_0, \tag{8a}$$
$$I_v(T) = c_{vv}(T - T_0) + I_{v,0}, \tag{8b}$$
$$I_l(T) = c_{vl}(T - T_0), \tag{8c}$$
$$I_i(T) = c_{vi}(T - T_0) - I_{i,0}. \tag{8d}$$

Here, $I_{v,0}$ and $I_{i,0}$ represent the reference specific internal energy differences at the reference temperature T_0 between vapor and liquid, and between ice and liquid, respectively. Since dry air and water are distinct constituents that cannot be converted into one another, we can independently specify their reference specific internal energies at T_0. For convenience in later enthalpy and enthalpy flux calculations, we set the reference specific internal energy for liquid water to 0 and that for dry air to $-R_d T_0$. (It can be verified that the equations of motion, to be discussed later, are invariant under addition of an arbitrary constant here, as they should be (Dubos, 2024).) The specific internal energy of moist air then is the weighted sum

$$\begin{aligned} I(T,q) &= (1 - q_t) I_d(T) + q_v I_v(T) + q_l I_l(T) + q_i I_i(T) \\ &= c_{vm}(q)(T - T_0) + (q_t - q_c) I_{v,0} - q_i I_{i,0} - (1 - q_t) R_d T_0, \end{aligned} \tag{9}$$

where we have used $q_v = q_t - q_c$. Temperature can be obtained from the specific internal energy and the specific humidities by

$$T = T_0 + \frac{I - (q_t - q_c) I_{v,0} + q_i I_{i,0} + (1 - q_t) R_d T_0}{c_{vm}(q)}. \tag{10}$$

This equation allows us to recover temperature when using specific internal energy and specific humidities as the primary thermodynamic state variables.

Furthermore, the reference specific internal energies $I_{v,0}$ and $I_{i,0}$ are related to the reference specific latent heats of vaporization and fusion ($L_{v,0}$ and $L_{f,0}$) at T_0 through the following expressions:

$$I_{v,0} = L_{v,0} - R_v T_0, \tag{11a}$$

$$I_{i,0} = L_{f,0}. \tag{11b}$$

These relations are obtained by subtracting the corresponding "p/ρ" term (which is zero for the condensed phases) from the specific latent heats.

2.5. Enthalpies

Enthalpy plays a crucial role in describing fluid transport thermodynamics, appearing in both grid-scale and subgrid-scale (SGS) transport terms.

The specific enthalpies of the moist air constituents are derived by adding p_μ/ρ_μ for each constituent μ to its specific internal energy (Equation 8). Neglecting the specific volumes ρ_l^{-1} and ρ_i^{-1} of the condensed phases and using the relations (11) between reference specific energies and latent heats, we arrive at the following expressions for specific enthalpies:

$$h_d(T) = I_d(T) + R_d T = c_{pd}(T - T_0), \tag{12a}$$

$$h_v(T) = I_v(T) + R_v T = c_{pv}(T - T_0) + L_{v,0}, \tag{12b}$$

$$h_l(T) = I_l(T) = c_{pl}(T - T_0), \tag{12c}$$

$$h_i(T) = I_i(T) = c_{pi}(T - T_0) - L_{f,0}. \tag{12d}$$

The enthalpy of moist air is the weighted sum of these constituent enthalpies:

$$\begin{aligned} h(T,q) &= (1 - q_t) h_d + q_v h_v + q_l h_l + q_i h_i \\ &= c_{pm}(q)(T - T_0) + (q_t - q_c) L_{v,0} - q_i L_{f,0} \\ &= I(T,q) + R_m T. \end{aligned} \tag{13}$$

2.6. Saturation Vapor Pressure

The Clausius-Clapeyron relation relates the saturation vapor pressure p_v^* of an ideal gas over a plane condensate surface to temperature,

$$\frac{d \log(p_v^*)}{dT} = \frac{L}{R_v T^2}, \tag{14}$$

where L is the specific latent heat of the phase transition: L_v over liquid, and L_s over ice. For a mixture of liquid and ice out of thermodynamic equilibrium (e.g., in mixed-phase clouds), a thermodynamically consistent saturation vapor pressure uses a weighted average of L_v and L_s,

$$L = \lambda_f L_v + (1 - \lambda_f) L_s, \tag{15}$$

where $\lambda_f = q_l/q_c$ is the liquid mass fraction of the condensate and $1 - \lambda_f = q_i/q_c$ is the ice mass fraction (Pressel et al., 2015).

Substituting the linear latent heat–temperature relation (7) and defining p_{tr} as the vapor pressure at the triple point at temperature T_{tr}, the Clausius-Clapeyron relation can be integrated to a closed-form expression for the vapor pressure, the so-called Rankine–Kirchhoff approximation (Duarte et al., 2014; Romps, 2021):

$$p_v^* = p_{\mathrm{tr}}\left(\frac{T}{T_{\mathrm{tr}}}\right)^{\alpha_v} \exp\left[\beta_v\left(\frac{1}{T_{\mathrm{tr}}} - \frac{1}{T}\right)\right]. \tag{16}$$

Here,

$$\alpha_v = \frac{c_{pv} - \lambda_f c_{pl} - \left(1 - \lambda_f\right) c_{pi}}{R_v} \tag{17}$$

and

$$\beta_v = \frac{\lambda_f L_{v,0} + \left(1 - \lambda_f\right) L_{s,0}}{R_v} - \alpha_v T_0. \tag{18}$$

For $\lambda_f = 1$, this gives the saturation vapor pressure over liquid, and for $\lambda_f = 0$, that over ice. This formulation is mathematically consistent, as the saturation vapor pressure (Equation 16) is invariant to the choice of reference temperature T_0: it does not change when the reference temperature is shifted from T_0 to $T_0 + \delta T_0$ provided the specific latent heats at the reference temperature are shifted correspondingly (e.g., $L_{v,0}$ is replaced by $L_{v,0} + (c_{pv} - c_{pl})\delta T_0$). However, the accuracy of the Rankine-Kirchhoff approximation depends on the chosen parameters. Using the triple point of water as the reference and the constants in Table 1, the approximation is highly accurate: the saturation vapor pressure over liquid between 248 and 325 K lies within 0.4% of measured values, and the ratio of the saturation vapor pressure over ice to that over liquid between 233 and 273 K lies within 0.6% of measured values (Ambaum, 2020).

2.7. Local Thermodynamic Equilibrium of Cloud Condensate

Optionally, the CliMA dynamical core can use the assumption of local thermodynamic equilibrium common in other atmosphere models, which takes the constituents of cloudy air (but excluding falling precipitation) to be not only at the same temperature but the water phases also to coexist in thermodynamic equilibrium. In that case, Gibbs' phase rule implies that it suffices to specify the thermodynamic state of cloudy air with only 3 thermodynamic state variables, such as a humidity, density, and an energy variable, from which the partitioning into water phases can then be uniquely determined. We may choose the specific total humidity of the cloudy air, that is, the moist air including cloud condensate but excluding precipitation,

$$q_t^{\mathrm{cl}} = q_v + q_l^{\mathrm{cl}} + q_i^{\mathrm{cl}} = q_t - q_l^{\mathrm{pr}} - q_i^{\mathrm{pr}}, \tag{19}$$

the corresponding cloudy air density,

$$\rho^{\mathrm{cl}} = \rho\left(1 - q_l^{\mathrm{pr}} - q_i^{\mathrm{pr}}\right), \tag{20}$$

and the specific internal energy of cloudy air,

$$I^{\mathrm{cl}} = I - \left(c_{vl} q_l^{\mathrm{pr}} + c_{vi} q_i^{\mathrm{pr}}\right)(T - T_0) + q_i^{\mathrm{pr}} I_{i,0} - \left(q_l^{\mathrm{pr}} + q_i^{\mathrm{pr}}\right) R_d T_0. \tag{21}$$

The condensate specific humidities q_l^{cl} and q_i^{cl} can then be obtained from these three quantities by saturation adjustment, while the precipitation humidities remain prognostic variables (e.g., Sridhar et al., 2022).

Thermodynamic equilibrium requires the liquid fraction to be a Heaviside function $\mathcal{H}$ of temperature,

$$\lambda_f(T) = \mathcal{H}\left(T - T_{\mathrm{freeze}}\right), \tag{22}$$

with $\lambda_f = 0$ below the freezing temperature T_{freeze} and $\lambda_f = 1$ above it. However, supercooled liquid, which is out of equilibrium, can exist between the homogeneous ice nucleation temperature T_{icenuc} and the freezing temperature T_{freeze}. Many climate models use a ramp function $\lambda_f(T)$ that transitions from 0 for $T \le T_{\text{icenuc}}$ to 1 for $T \ge T_{\text{freeze}}$ to simulate this, for example,

$$\lambda_f(T) = \begin{cases} 1 & T \ge T_{\text{freeze}}, \\ \left(\dfrac{T - T_{\text{icenuc}}}{T_{\text{freeze}} - T_{\text{icenuc}}}\right)^{n_f} & T_{\text{icenuc}} < T < T_{\text{freeze}}, \\ 0 & T \le T_{\text{icenuc}}, \end{cases} \tag{23}$$

with some exponent n_f (Kaul et al., 2015). Although this can be incorporated within our thermodynamic framework, and an option for it is implemented in our code, it is an approximation for modeling out-of-equilibrium phases such as supercooled liquids which, in reality, depend not solely on thermodynamic state variables but also on the history of air masses. Our default choice is to model the condensed phases with separate prognostic equations, with microphysics parameterizations that allow non-equilibrium phases to coexist (e.g., Morrison & Milbrandt, 2015).

3. Dynamics

3.1. Prognostic Variables

The CliMA atmosphere model employs the fully compressible, non-hydrostatic fluid dynamical equations. As prognostic variables for water, it uses the total water specific humidity q_t, liquid specific humidities q_l^σ, and ice specific humidities q_i^σ. Here σ can represent either cloud condensate ("cl") or precipitation ("pr") when these are treated as separate water mass categories, or it can be omitted if no such distinction is made. If local thermodynamic equilibrium is assumed, q_l^{cl} and q_i^{cl} can be diagnosed from the other thermodynamic variables through saturation adjustment (Section 2.7).

As prognostic variable for energy, the model employs specific total energy of moist air (Bott, 2008; Romps, 2008; Sridhar et al., 2022),

$$e_{\text{tot}} = I + \Phi + \kappa + \kappa_{\text{SGS}}. \tag{24}$$

Here, $\Phi = gz$ represents the geopotential in the approximation of Earth as a sphere (White et al., 2005), where z is the altitude above a reference level (geoid) and we additionally take the gravitational acceleration g to be constant (extension of the geopotential to non-constant g is straightforward if needed for deep planetary atmosphere simulations);

$$\kappa = \frac{1}{2}\|\boldsymbol{u}\|^2 \tag{25}$$

is the specific kinetic energy associated with the resolved three-dimensional velocity $\boldsymbol{u}$, which, following Romps (2008), is taken to be the velocity of dry air (i.e., the kinetic energy of the differential motion of water is neglected); and κ_{SGS} represents the parameterized SGS kinetic energy if that is available.

This choice, combined with a conservative discretization and the consistent thermodynamics described previously, guarantees the conservation of total energy, air mass, and water mass (up to roundoff) in the model. This resolves the challenge of achieving energy conservation in the presence of complex topography and parameterization schemes, without the ad-hoc fixers that are usually used in atmosphere models (P. H. Lauritzen et al., 2022). The trade-off is that a simple, discretely materially conserved thermodynamic variable such as potential temperature is no longer guaranteed to be conserved. We consider this advantageous, however, because potential temperature is conserved only under limited physical assumptions (e.g., for dry dynamics) and, unlike total energy, is not an extensive quantity that satisfies a volume-integrated conservation law.

3.2. Coordinate-Independent Equations of Motion

We begin by introducing the equations in a form that is not dependent on specific coordinates, before expanding them in a generalized coordinate system with terrain-following vertical coordinates later. For scalar quantities, we use governing equations in flux form to ensure discrete conservation. For momentum, we use the "vector invariant" form, where advection terms are expressed as vorticity terms and kinetic energy gradients, using the dot product identity $\boldsymbol{u}\cdot\nabla\boldsymbol{u} = \nabla\|\boldsymbol{u}\|^2/2 + (\nabla\times\boldsymbol{u})\times\boldsymbol{u}$. This approach accurately represents rotational wave modes in the discrete model (Ringler et al., 2010; M. A. Taylor & Fournier, 2010) and avoids the curvature terms (Christoffel symbols) that otherwise arise in advection terms from coordinate derivatives in non-orthogonal coordinates.

The governing equations for a general deep atmosphere are as follows (Romps, 2008; White et al., 2005):

1. Mass continuity

$$\frac{\partial}{\partial t}\rho + \nabla\cdot\left[\rho\left(\boldsymbol{u} - W_{q_t}\hat{\boldsymbol{k}}\right)\right] = \rho\,\hat{S}_{q_t} \tag{26a}$$

2. Momentum

$$\frac{\partial}{\partial t}\boldsymbol{u} + (2\boldsymbol{\Omega}+\boldsymbol{\omega})\times\boldsymbol{u} = -c_{pd}(\theta_v - \theta_{v,r})\nabla\Pi - \nabla(\Phi-\Phi_r) - \nabla\kappa + \boldsymbol{\mathcal{S}}_{\boldsymbol{u}} - \frac{1}{\rho}\nabla\cdot[\rho(\boldsymbol{\mathcal{T}}+\boldsymbol{\mathcal{H}}_{\boldsymbol{u}})] \tag{26b}$$

3. Total energy

$$\frac{\partial}{\partial t}(\rho e_{\text{tot}}) + \nabla\cdot\left[\rho\left(h_{\text{tot}}\boldsymbol{u} - W_h\hat{\boldsymbol{k}}\right)\right] = -\nabla\cdot[\rho(\boldsymbol{\mathcal{F}}_R + \boldsymbol{\mathcal{F}}_h + \boldsymbol{\mathcal{H}}_h + \boldsymbol{u}\cdot(\boldsymbol{\mathcal{T}}+\boldsymbol{\mathcal{H}}_{\boldsymbol{u}}))] \tag{26c}$$

4. Total water

$$\frac{\partial}{\partial t}(\rho q_t) + \nabla\cdot\left[\rho\left(q_t\boldsymbol{u} - W_{q_t}\hat{\boldsymbol{k}}\right)\right] = -\nabla\cdot[\rho(\boldsymbol{\mathcal{F}}_{q_t} + \boldsymbol{\mathcal{H}}_{q_t})] \equiv \rho\,\hat{S}_{q_t} \tag{26d}$$

5. Condensate/precipitation (for $\mu\in\{l,i\}$ and $\sigma\in\{\cdot,\text{cl},\text{pr}\}$ as needed)

$$\frac{\partial}{\partial t}(\rho q_\mu^\sigma) + \nabla\cdot\left[\rho q_\mu^\sigma\left(\boldsymbol{u} - w_\mu^\sigma\hat{\boldsymbol{k}}\right)\right] = \rho\mathcal{S}_{q_\mu^\sigma} - \nabla\cdot[\rho(\boldsymbol{\mathcal{F}}_{q_\mu^\sigma} + \boldsymbol{\mathcal{H}}_{q_\mu^\sigma})] \tag{26e}$$

6. Tracers

$$\frac{\partial}{\partial t}(\rho\chi) + \nabla\cdot\left[\rho\chi\left(\boldsymbol{u} - w_\chi\hat{\boldsymbol{k}}\right)\right] = \rho\mathcal{S}_\chi - \nabla\cdot[\rho(\boldsymbol{\mathcal{F}}_\chi + \boldsymbol{\mathcal{H}}_\chi)] \tag{26f}$$

7. Equation of state

$$p = \rho R_m T \tag{26g}$$

The variables in Equation 26 are defined in Table 2.

A few salient points about this equation set are worth pointing out:

1. *Conservation Properties*. The governing equations for mass, energy, and total water are written in a flux-conservative form. Their right-hand sides contain only flux-divergence terms, with no non-conservative sources or sinks. This structure is by design: when paired with a flux-conservative discretization, it guarantees that the volume-integrated air mass, total water, and total energy are conserved within the domain, with changes occurring only through fluxes across the boundaries. This holds even in the presence of parameterizations that are contained within the various subgrid-scale (SGS) fluxes on the right-hand sides, representing the effects of unresolved motions on the resolved scales.
2. *Equation Formulation and Closures*. The advective flux term in the total energy equation involves the specific total enthalpy, $h_{\text{tot}} = e_{\text{tot}} + p/\rho$, which includes kinetic and potential energy contributions. The pressure is determined by the ideal-gas equation of state (Equation 26g), which neglects pressure contributions from averaging over correlated subgrid-scale fluctuations.
3. *Dry-Air Reference Frame*. Following Romps (2008), the equations are formulated in a dry-air reference frame where $\boldsymbol{u}$ represents the velocity of dry air. The differential motion of water relative to dry air (e.g., hydrometeor fall) is explicitly accounted for as mass and energy flux terms in the balance equations. This choice is advantageous for applying boundary conditions, as the wall-normal velocity of dry air can be set to zero at a rigid boundary even with surface moisture fluxes.
4. *Model Simplifications*. For computational efficiency, the model disregards the transport of momentum and the kinetic energy associated with the differential motion of water relative to dry air (Romps, 2008). Neglecting the differential momentum transport by water is justified because the omitted terms describe an internal redistribution of momentum whose global integral is zero, and because achieving a precise local momentum balance is considered less critical for long-term stability than maintaining an exact balance for mass and energy, for which small spurious sources can cause significant climate drift (Thuburn, 2008).
5. *Temperature Computation*. Temperature is computed from the specific total energy e_{tot} via Equation 10. The necessary subtraction of kinetic energy from total energy does not result in catastrophic cancellation of roundoff errors because kinetic energy is a small fraction of the total energy, with the ratio scaling like the square of the Mach number U/c_s, where U is a velocity scale and c_s the speed of sound (Peixoto et al., 1991). In the atmosphere, this is small—typically, $(U/c_s)^2 \lesssim O\left(10^{-2}\right)$—so the kinetic energy contributes a small term to the total energy. Even when considering only perturbations, the ratio of kinetic energy perturbations to thermal perturbations scales like $0.5(\delta U)^2/(c_{vd}\delta T)$, which is still $O\left(10^{-2}\right)$ for $\delta U \sim 10$ m s^{-1} and $\delta T \sim 5$ K.
6. *Pressure Gradient Force*. Following M. A. Taylor et al. (2020), the pressure-gradient acceleration in the momentum Equation 26b is formulated using the Exner function Π and virtual potential temperature θ_v,

$$-\frac{1}{\rho}\nabla p = -c_{pd}\theta_v \nabla \Pi, \tag{27}$$

where $\Pi = \left(p/p_0\right)^{R_d/c_{pd}}$ is the Exner function defined with respect to dry-air constants (Table 1) and

$$\theta_v = \frac{R_m}{R_d}\frac{T}{\Pi} \tag{28}$$

is the corresponding virtual potential temperature. Using the Exner function is a well-established method that, when paired with a mimetic discretization such as the spectral element method, avoids spurious potential energy–kinetic energy conversions from the pressure work terms (M. A. Taylor et al., 2020). To further improve numerical accuracy and reduce discretization noise near topography, the force is expressed as a deviation from a hydrostatically balanced reference state (Golaz et al., 2022; Herrington et al., 2022). We choose the reference state with temperature $T_r(\Pi) = T_{\text{min}} + (T_{\text{sfc}} - T_{\text{min}})\Pi^{n_s}$, with surface temperature $T_{\text{sfc}} = 288$ K, minimum temperature $T_{\text{min}} = 215$ K, and $n_s = 7$, which ensures a smooth transition from T_{sfc} to T_{min} in the stratosphere. With $\theta_{v,r} = T_r/\Pi$ and

$$\Phi_r = -c_{pd}\left[T_{\text{min}}\log\Pi + \frac{T_{\text{sfc}} - T_{\text{min}}}{n_s}(\Pi^{n_s} - 1)\right], \tag{29}$$

Table 2
Variables and Parameters in Dynamical Equations

Variable	Definition	Units
e_{tot}	Specific total energy of moist air	J kg^{-1}
h	Specific enthalpy of moist air	J kg^{-1}
h_μ	Specific enthalpy of constituent $\mu \in \{d, v, l, i\}$	J kg^{-1}
h_{tot}	Specific total enthalpy $h_{\text{tot}} = e_{\text{tot}} + p/\rho$	J kg^{-1}
I_μ	Specific internal energy of constituent $\mu \in \{d, v, l, i\}$	J kg^{-1}
I	Specific internal energy of moist air	J kg^{-1}
$\hat{\boldsymbol{k}}$	Vertical unit vector	
p	Pressure	Pa
q_t	Total water specific humidity	kg/kg
q_l	Liquid water specific humidity	kg/kg
q_i	Ice specific humidity	kg/kg
q_μ^σ	Generic condensate specific humidity, $\mu \in \{l, i\}$, $\sigma \in \{\text{pr}, \text{cl}\}$	kg/kg
R_m	Gas "constant" of moist air (depends on specific humidities)	J kg^{-1} K^{-1}
s_d	Dry static energy $s_d = c_{pd}(T - T_0) + \Phi$	J kg^{-1}
T	Temperature	K
$\boldsymbol{u}$	Velocity (3D) of dry air	m s^{-1}
w_μ^σ	Sedimentation/fall velocity, $\mu \in \{l, i, \chi\}$, $\sigma \in \{\cdot, \text{cl}, \text{pr}\}$	m s^{-1}
W_h	Sedimentation flux of enthalpy	W m kg^{-1}
W_{q_t}	Sedimentation flux of total water	m s^{-1}
θ_v	Virtual potential temperature	K
$\theta_{v,r}$	Reference virtual potential temperature	K
κ_{SGS}	Specific kinetic energy of SGS motions	J kg^{-1}
Π	Exner function $\Pi = (p/p_0)^{R_d/c_{pd}}$	
ρ	Density of moist air	kg m^{-3}
Φ	Geopotential	m^2 s^{-2}
Φ_r	Reference geopotential	m^2 s^{-2}
χ	Generic tracer	kg/kg
$\boldsymbol{\omega}$	Relative vorticity (3D) $\boldsymbol{\omega} = \nabla \times \boldsymbol{u}$	s^{-1}
$\boldsymbol{\Omega}$	Angular velocity of planetary rotation	s^{-1}
$\boldsymbol{S_u}$	Specific momentum source (acceleration)	m s^{-2}
S_ψ	Source of scalar ψ	s^{-1}
$\hat{S}_{q_t}$	Effective source of total water specific humidity	s^{-1}
$\boldsymbol{\mathcal{F}}_h$	SGS flux of specific total enthalpy	W m kg^{-1}
$\boldsymbol{\mathcal{F}}_R$	Radiative energy flux	W m kg^{-1}
$\boldsymbol{\mathcal{F}}_\psi$	SGS flux of scalar ψ	m s^{-1}
$\boldsymbol{\mathcal{H}}_h$	Hyperdiffusive flux of specific total enthalpy	W m kg^{-1}
$\boldsymbol{\mathcal{H}}_{\boldsymbol{u}}$	Hyperdiffusive momentum flux tensor	m^2 s^{-2}
$\boldsymbol{\mathcal{H}}_\psi$	Hyperdiffusive flux of scalar ψ	m s^{-1}
$\boldsymbol{\mathcal{T}}$	SGS momentum flux tensor	m^2 s^{-2}

this satisfies the hydrostatic balance equation $c_{pd}\theta_{v,r}\nabla\Pi + \nabla\Phi_r = 0$ for any Π. It results in the combined pressure-gradient and geopotential terms seen in Equation 26b.

7. *Reference Temperature Invariance*. The thermodynamic framework is constructed such that the model's energetics are invariant under shifts of the reference temperature T_0. This ensures that the choice of T_0 does not affect physical results, except through the quality of the linearization of the temperature dependence of the specific latent heats (Ambaum, 2020). This invariance is a direct consequence of the model's thermodynamically consistent formulation. A shift in the reference temperature, $T_0 \to T_0 + \delta T_0$, and the corresponding consistent shift in the reference latent heats (e.g., $L_{v,0} \to L_{v,0} + (c_{pv} - c_{pl})\delta T_0$), induces constant offsets to the specific enthalpies of each constituent. Because the offsets for all water species are identical, the additional terms generated in the total energy equation sum to zero identically by virtue of the total water conservation law (Appendix A). This cancellation ensures that the model's energetics and dynamics are fundamentally independent of the arbitrary choice of reference temperature.

There are several terms in Equation 26 that are ultimately the result of parameterizations, to be discussed in subsequent papers on the atmosphere model; they are summarized in what follows.

3.3. Parameterized SGS Fluxes

The right-hand sides of the governing equations include the SGS momentum flux tensor $\boldsymbol{\mathcal{T}}$ and the SGS scalar fluxes $\boldsymbol{\mathcal{F}}_h$, $\boldsymbol{\mathcal{F}}_{q_t}$, $\boldsymbol{\mathcal{F}}_{q_\mu^c}$, and $\boldsymbol{\mathcal{F}}_\chi$. These terms represent the effects of unresolved subgrid-scale processes on the resolved flow. The functional forms of these fluxes are determined by parameterization schemes, which may have diffusive and advective components, the latter, for example, associated with convection closures (e.g., Christopoulos et al., 2024; Cohen et al., 2020; Lopez-Gomez et al., 2020, 2022).

For any portion of the SGS flux that is modeled with a diffusive closure, energetic consistency and invariance to the choice of reference temperature place strict constraints on the formulation. The total SGS enthalpy flux, $\boldsymbol{\mathcal{F}}_h$, is conceptually decomposed into two components: (a) the total enthalpy carried by the SGS water fluxes, and (b) a diffusive heat flux analogous to Fourier's law of heat conduction (Romps, 2008). While it can be computationally convenient to model the total flux as proportional to the gradient of a single combined quantity such as total enthalpy (h_{tot}), this approach is inconsistent in a multi-component fluid with a dry-air reference frame because it includes a component representing the enthalpy flux associated with dry-air diffusion ($\propto -h_{\text{tot},d}\nabla(1-q_t)$, which should not appear.

A consistent formulation requires modeling these two components explicitly. The first component, representing the enthalpy advected by the diffusion of water mass, is given by the sum of the enthalpies carried by the diffusive mass flux of each water species,

$$\boldsymbol{\mathcal{F}}_{h,\text{water}} = \sum_{\mu\in\{v,l,i\}} h_{\text{tot},\mu}\,\boldsymbol{\mathcal{F}}_{q_\mu} \tag{30}$$

where $\boldsymbol{\mathcal{F}}_{q_\mu} \approx -D_q\nabla q_\mu$ is the diffusive mass flux of water species μ with specific total enthalpy $h_{\text{tot},\mu}$ (Romps, 2008). The second component is the thermal diffusion of sensible heat. In a stratified fluid, this flux must be proportional to the gradient of a quantity that is at least approximately conserved under adiabatic displacements: here, the dry static energy,

$$s_d = h_d + \Phi = c_{pd}(T - T_0) + \Phi \tag{31}$$

is the natural quantity. This ensures that no spurious vertical heat flux is generated in a neutrally stratified atmosphere with $\partial s_d/\partial z = 0$. The thermal flux is therefore modeled as

$$\boldsymbol{\mathcal{F}}_{h,\text{thermal}} = -D_h\nabla s_d. \tag{32}$$

The total diffusive SGS enthalpy flux is the sum of these two components:

$$\boldsymbol{\mathcal{F}}_h = \boldsymbol{\mathcal{F}}_{h,\text{water}} + \boldsymbol{\mathcal{F}}_{h,\text{thermal}} = -D_q \sum_{\mu\in\{v,l,i\}} h_{\text{tot},\mu} \nabla q_\mu - D_h \nabla s_d. \tag{33}$$

This explicit decomposition ensures energetic consistency. The thermal flux term is invariant under shifts of the reference temperature T_0, and the transformation of the water-enthalpy flux is handled consistently by the total water conservation law (Appendix A). This consistency holds for any choice of the diffusivities D_h and D_q. A common and physically motivated choice is to set them equal ($D_h = D_q$), which corresponds to a turbulent Lewis number of unity and ensures that the net turbulent flux approximately vanishes in a moist-neutral atmosphere. We will discuss the partitioning of this total enthalpy flux in the context of boundary fluxes in Section 3.7.4.

3.4. Hyperdiffusion

Additionally, the SGS fluxes also include the horizontal hyperdiffusive scalar fluxes $\boldsymbol{\mathcal{H}}_h$, $\boldsymbol{\mathcal{H}}_{q_t}$, $\boldsymbol{\mathcal{H}}_{q_\mu^\sigma}$, $\boldsymbol{\mathcal{H}}_\chi$, as well as the hyperviscous specific momentum flux tensor $\boldsymbol{\mathcal{H}}_{\boldsymbol{u}}$. They are included to enhance numerical stability and suppress spurious oscillations, that is, as a numerical artifice; in our implementation, they also depend on the terrain-following coordinate system and its notion of what is meant by "horizontal." Nonetheless, we include them here for completeness.

For a generic scalar ψ, we use biharmonic hyperdiffusion, , where the flux is proportional to the third derivative of the field. The hyperdiffusive flux is given by

$$\boldsymbol{\mathcal{H}}_\psi = \nu_\psi \nabla_h \left[\nabla_h^2 (\psi - \psi_r)\right], \tag{34}$$

where ν_ψ is a hyperdiffusion coefficient and ∇_h is the horizontal del operator. In this context, "horizontal" refers to terrain-following horizontal coordinates, as detailed in Section 4. A vertically varying reference state ψ_r is subtracted from dry static energy and total specific humidity to reduce spurious vertical mixing that can arise from diffusion along coordinate surfaces that warp over steep topography (e.g., Doms & Baldauf, 2018; Herrington et al., 2022). The reference state is derived from the same temperature profile $T_r(\Pi)$ used for the pressure gradient force, with a constant relative humidity of 50% for specific humidity.

Hyperdiffusion is applied to the total specific humidity, q_t. However, a physical distinction is made in how the resulting tendency is applied to the individual water species. Applying horizontal diffusion to precipitating species (e.g., rain and snow) would artificially smear out features such as rain shafts whose evolution is dominated by vertical sedimentation and microphysics. Therefore, to preserve the structure of precipitation fields, the hyperdiffusive tendency for total water is partitioned exclusively among the suspended species (water vapor, cloud liquid, and cloud ice). The hyperdiffusive flux divergence for a specific condensate category q_μ^σ is thus defined as

$$-\frac{1}{\rho}\nabla_h \cdot (\rho \boldsymbol{\mathcal{H}}_{q_\mu^\sigma}) = \begin{cases} \dfrac{q_\mu^\sigma}{q_t^{\text{cl}}} \left(-\dfrac{1}{\rho}\nabla_h \cdot (\rho \boldsymbol{\mathcal{H}}_{q_t})\right) & \text{if } \sigma = \text{cl (cloud)}, \\ 0 & \text{if } \sigma = \text{pr (precipitation)}, \end{cases} \tag{35}$$

where $q_t^{\text{cl}} = q_t - q_l^{\text{pr}} - q_i^{\text{pr}}$ is the total specific humidity of the suspended components (vapor and cloud condensate). The corresponding tendency for water vapor is handled implicitly, as its specific humidity is diagnosed from the total and condensate specific humidities.

Consistent with the formulation for the turbulent flux $\boldsymbol{\mathcal{F}}_h$, the total hyperdiffusive enthalpy flux $\boldsymbol{\mathcal{H}}_h$ in the total energy equation is constructed from its components: a thermal flux and a water-enthalpy flux. The thermal flux, $\boldsymbol{\mathcal{H}}_{s_d}$, directly hyperdiffuses the dry static energy, $s_d = h_d + \Phi$. The water-enthalpy flux, $\boldsymbol{\mathcal{H}}_{h,\text{water}}$, accounts for the enthalpy transported by the hyperdiffusive flux of total water, $\boldsymbol{\mathcal{H}}_{q_t}$. Given the partitioning of the water tendency, this flux is exactly given by

$$\boldsymbol{\mathcal{H}}_{h,\text{water}} = h_{\text{tot,cl}} \boldsymbol{\mathcal{H}}_{q_t}, \tag{36}$$

where $h_{\mathrm{tot,cl}}$ is the mass-weighted average specific total enthalpy of the suspended water species:

$$h_{\mathrm{tot,cl}} = \frac{q_v h_{\mathrm{tot},v} + q_l^{\mathrm{cl}} h_{\mathrm{tot},l} + q_i^{\mathrm{cl}} h_{\mathrm{tot},i}}{q_t^{\mathrm{cl}}}. \tag{37}$$

The total hyperdiffusive enthalpy flux is the sum of these components:

$$\boldsymbol{\mathcal{H}}_h = \boldsymbol{\mathcal{H}}_{s_d} + \boldsymbol{\mathcal{H}}_{h,\mathrm{water}} = \nu_h \nabla_h \left[\nabla_h^2 (s_d - s_{d,r})\right] + h_{\mathrm{tot,cl}} \boldsymbol{\mathcal{H}}_{q_t}. \tag{38}$$

Consistent with the physically motivated assumption of a turbulent Lewis number of unity, the hyperdiffusion coefficients for dry static energy (ν_h) and total specific humidity (ν_{q_t}) are chosen to be equal. This consistent formulation, where the hyperdiffusive flux in the total energy equation explicitly accounts for the energy transported by the hyperdiffusive water flux, ensures that total energy and water are conserved. Because all hyperdiffusion is formulated as flux divergences along horizontal coordinate surfaces, the operators only redistribute quantities and do not create spurious global sources or sinks. This contrasts with formulations where hyperdiffusion is applied to potential temperature or other intensive thermodynamic variables, which can lead to unphysical sources or sinks of total energy that then must be compensated by energy fixers.

For the velocity field, we apply horizontal biharmonic hyperviscosity to the three-dimensional vector velocity $\boldsymbol{u}$,

$$\boldsymbol{\mathcal{H}}_{\boldsymbol{u}} = \nu_{\boldsymbol{u}} \nabla_h (\Delta_h \boldsymbol{u}), \tag{39}$$

with horizontal vector Laplacian defined as

$$\Delta_h \boldsymbol{u} = \nabla_h (\nabla_h \cdot \boldsymbol{u}) - \nabla_h \times (\nabla_h \times \boldsymbol{u}). \tag{40}$$

The specific momentum tendency due to hyperviscosity can then be expressed as

$$-\frac{1}{\rho} \nabla_h \cdot (\rho \boldsymbol{\mathcal{H}}_{\boldsymbol{u}}) = -\frac{1}{\rho} \nabla_h \cdot (\rho \nu_{\boldsymbol{u}} \nabla_h) \Delta_h \boldsymbol{u}. \tag{41}$$

Since computing the tensor divergence in non-orthogonal coordinates involves curvature terms (Christoffel symbols), we adopt a computationally more efficient form that is often employed in atmosphere models (e.g., Dennis et al., 2012; P. H. Lauritzen et al., 2018; Ullrich, 2014; Ullrich et al., 2018):

$$-\frac{1}{\rho} \nabla_h \cdot (\rho \boldsymbol{\mathcal{H}}_{\boldsymbol{u}}) \approx \boldsymbol{\mathcal{D}}_{\boldsymbol{u}}^4 = -\nu_{\boldsymbol{u}} [\delta_{\mathrm{div}} \nabla_h (\nabla_h \cdot \Delta_h \boldsymbol{u}) - \nabla \times (\nabla_h \times \Delta_h \boldsymbol{u})]. \tag{42}$$

When $\delta_{\mathrm{div}} = 1$ and the hyperviscosity $\nu_{\boldsymbol{u}}$ is constant, this approximation arises from neglecting density variations along horizontal coordinate surfaces. For $\delta_{\mathrm{div}} > 1$, the nondimensional parameter δ_{div} allows for increased damping of two-dimensional divergent flow components, which include gravity waves (Jablonowski & Williamson, 2011). Our default choice is $\delta_{\mathrm{div}} = 5$ (P. H. Lauritzen et al., 2018).

For numerical stability, the hyperviscosity $\nu_{\boldsymbol{u}}$ generally needs to increase with average horizontal grid spacing δx, typically like $(\delta x)^3$ (P. H. Lauritzen et al., 2018; Skamarock et al., 2014). The hyperdiffusivities $\nu_h = \nu_{q_t}$ are related to the hyperviscosity $\nu_{\boldsymbol{u}}$ by a turbulent Prandtl number

$$\mathrm{Pr}_t = \frac{\nu_{\boldsymbol{u}}}{\nu_h} \lesssim 1. \tag{43}$$

3.5. Sedimentation and Fall Velocities

In moist air, liquid water and ice within clouds can sediment, and precipitation falls. We denote the corresponding downward velocities relative to the dry-air velocity by w_l and w_i for liquid water and ice. For distinct cloud condensate and precipitation species, we use w_μ^σ (where $\mu \in \{l, i\}$ and $\sigma \in \{\mathrm{cl}, \mathrm{pr}\}$), and for any other tracer χ with a

differential vertical velocity relative to the dry air (e.g., a moment of a hydrometeor size distribution), we use w_χ. All sedimentation and fall velocities are defined as positive downward, implying that $-w_\mu^\sigma \hat{\boldsymbol{k}}$, with upward vertical unit vector $\hat{\boldsymbol{k}}$, points downward.

These sedimentation and fall velocities are parameterized, often by a microphysics scheme that relates them to the size distributions of hydrometeors. The total water sedimentation flux W_{q_t} is the weighted sum of the constituent velocities, considering that only condensates have motion relative to the dry air:

$$W_{q_t} = q_l w_l + q_i w_i. \tag{44}$$

The condensate velocities can themselves be weighted averages of separate condensate categories, for example:

$$w_l = \frac{q_l^{\mathrm{cl}} w_l^{\mathrm{cl}} + q_l^{\mathrm{pr}} w_l^{\mathrm{pr}}}{q_l}, \tag{45}$$

with an analogous expression for ice velocities.

Sedimenting/falling condensate carries total specific enthalpy downward, resulting in a sedimentation flux

$$W_h = q_l h_{\mathrm{tot},l} w_l + q_i h_{\mathrm{tot},i} w_i, \tag{46}$$

where $h_{\mathrm{tot},l}$ and $h_{\mathrm{tot},i}$ are the specific total enthalpies of cloud liquid and ice. (However, because condensates are assumed to be incompressible, energies and enthalpies do not need to be distinguished here).

3.6. Mass Sources and Sinks

The mass continuity Equation 26a explicitly accounts for the sedimentation of condensate relative to the dry-air velocity $\boldsymbol{u}$. In addition, on the right-hand side, it includes the effective moisture source/sink

$$\hat{S}_{q_t} = -\frac{1}{\rho}\nabla \cdot [\rho(\boldsymbol{\mathcal{F}}_{q_t} + \boldsymbol{\mathcal{H}}_{q_t})] \tag{47}$$

owing to SGS fluxes of water, $\boldsymbol{\mathcal{F}}_{q_t}$ and $\boldsymbol{\mathcal{H}}_{q_t}$ (Romps, 2008). Taking the difference between the continuity Equation 26a and the total water Equation 26d then yields the exact conservation law for dry air mass density $\rho(1 - q_t)$, without sources/sinks on the right-hand side.

In Earth's atmosphere, the effective moisture source/sink $\hat{S}_{q_t}$ is usually at least two orders of magnitude smaller than the other terms in the continuity Equation 26a. It is neglected in most models, and it does not usually have a significant effect in Earth-like climates (Abbott & O'Gorman, 2024). We retain it for generality, as it becomes dynamically important, for example, in Mars's atmosphere, where the condensable species (CO_2) is a major constituent of the atmosphere (Soto et al., 2015).

3.7. Surface Boundary Conditions

The surface boundary conditions are relatively standard. We enforce the vanishing of the normal component of the dry-air velocity at the surface,

$$\hat{\boldsymbol{n}} \cdot \boldsymbol{u}|_{\mathrm{sfc}} = 0, \tag{48}$$

where $\hat{\boldsymbol{n}}$ is the outward-pointing unit normal vector at the surface. As the velocity $\boldsymbol{u}$ is the dry-air velocity, this condition enforces an impermeable boundary for the dry-air component of the fluid. This is an advantage of the dry-air reference frame, as it allows for a simple no-normal-flow condition even in the presence of surface moisture fluxes (e.g., evaporation or sublimation), where the moist-air mean velocity would be non-zero.

To specify diffusive flux boundary conditions, we also require the near-surface velocity component parallel to the surface

Table 3
Variables in Boundary Conditions

Variable	Definition	Units
$\hat{\boldsymbol{n}}$	Unit normal vector pointing into flow domain	
C_d	Drag coefficient for momentum	
C_h	Exchange coefficient for enthalpy and water vapor	
g_{ae}	Aerodynamic conductance	m s^{-1}
g_v	Conductance for enthalpy and water vapor	m s^{-1}
q_v^*	Saturation specific humidity	kg/kg
E	Evaporation/sublimation rate	kg m^{-2} s^{-1}
P_μ	Precipitation rate ($\mu \in \{l, i\}$)	kg m^{-2} s^{-1}
EF	Total enthalpy flux	W m^{-2}
LHF	Latent heat flux	W m^{-2}
SHF	Sensible heat flux	W m^{-2}

$$\boldsymbol{u}_{p,\text{int}} = \left(\boldsymbol{I} - \hat{\boldsymbol{n}}\hat{\boldsymbol{n}}^T\right)\boldsymbol{u}|_{\text{int}}, \tag{49}$$

where $\boldsymbol{I}$ is the identity tensor. Here, the subscript "sfc" denotes quantities at the surface, while "int" refers to near-surface quantities within the atmosphere, typically evaluated at the model level closest to the surface.

Surface boundary conditions for the various SGS fluxes $\boldsymbol{\mathcal{F}}$. are generally formulated using bulk exchange laws, with exchange coefficients derived from Monin-Obukhov similarity theory (Foken, 2006). These bulk exchange laws are essentially first-order finite difference approximations of flux-gradient relationships. We introduce the notation

$$\Delta_s\psi = \psi_{\text{int}} - \psi_{\text{sfc}} \tag{50}$$

to represent differences in scalar quantities ψ between the near-surface atmosphere and the surface. See Table 3 for variables and notation used in boundary conditions.

The various hyperdiffusive fluxes $\boldsymbol{\mathcal{H}}$. are aligned with the model's "horizontal" coordinate surfaces, which conform to the terrain near the surface. This alignment eliminates the need to explicitly define boundary conditions for hyperdiffusive fluxes at the surface, although it may introduce spurious vertical mixing near steep topography.

Finally, fluxes associated with sedimenting or falling condensate are subject to outgoing conditions at the surface.

3.7.1. Momentum Flux

The momentum flux at the surface is modeled using a drag law,

$$\hat{\boldsymbol{n}} \cdot (\rho\boldsymbol{\mathcal{T}})|_{\text{sfc}} = -\rho_{\text{sfc}} C_d ||\boldsymbol{u}_{p,\text{int}}|| \boldsymbol{u}_{p,\text{int}}, \tag{51}$$

where C_d represents the drag coefficient (determined based on surface and flow properties using Monin-Obukhov similarity theory) and ρ_{sfc} is the air density at the surface, obtained by hydrostatic extrapolation:

$$\rho_{\text{sfc}} \approx \rho_{\text{int}} \left(\frac{R_m(q_{\text{int}})T_{\text{int}}}{R_m(q_{\text{sfc}})T_{\text{sfc}}}\right) \exp\left(\frac{g\Delta z}{R_m(q_{\text{int}})T_{\text{int}}}\right). \tag{52}$$

The surface velocity is assumed to be zero, so only the near-surface wind parallel to the surface, $\boldsymbol{u}_{p,\text{int}}$, appears and effectively captures the velocity difference between the near-surface air and the surface.

3.7.2. Evaporation and Sublimation

Evaporation and sublimation processes are modeled using a bulk exchange law similar to the momentum drag law,

$$E = \hat{\boldsymbol{n}} \cdot (\rho \boldsymbol{\mathcal{F}}_{q_v})\big|_{\text{sfc}} = -\rho_{\text{sfc}} g_h \Delta_s q_v, \tag{53}$$

where g_h is the effective conductance for water vapor. Over wet surfaces, such as oceans or lakes, this conductance reduces to the aerodynamic conductance,

$$g_h = g_{\text{ae}} = C_h \|\boldsymbol{u}_{p,\text{int}}\|, \tag{54}$$

with C_h analogous to C_d for momentum. The surface specific humidity $q_{v,\text{sfc}}$ appearing in $\Delta_s q_v = q_{v,\text{int}} - q_{v,\text{sfc}}$ in that case is set to the saturation specific humidity at the surface conditions,

$$q_{v,\text{sfc}} = q_v^*(T_{\text{sfc}}, \rho_{\text{sfc}}). \tag{55}$$

For land surfaces, modifications to conductance and surface specific humidity reflect the surface characteristics (Bonan, 2019).

3.7.3. Precipitation and Sedimentation of Cloud Condensate

Condensates with sedimentation/fall velocities w_l and w_i (and w_l^σ and w_i^σ if there are separate cloud and precipitation categories) satisfy outgoing boundary conditions at the surface; that is, for example,

$$\rho q_\mu^\sigma w_\mu^\sigma\big|_{\text{sfc}} = \rho q_\mu^\sigma w_\mu^\sigma\big|_{\text{int}}. \tag{56}$$

The fluxes

$$P_\mu = \sum_\sigma \rho q_\mu^\sigma w_\mu^\sigma (\hat{\boldsymbol{k}} \cdot \hat{\boldsymbol{n}})\big|_{\text{sfc}} \quad \text{for} \quad \mu \in \{l, i\},\ \sigma \in \{\cdot, \text{cl}, \text{pr}\} \tag{57}$$

all contribute to the respective precipitation rates.

Diffusive SGS fluxes of cloud condensate and precipitation are taken to vanish at the surface, for example,

$$\hat{\boldsymbol{n}} \cdot (\rho \boldsymbol{\mathcal{F}}_{q_\mu})\big|_{\text{sfc}} = 0 \quad \text{for} \quad \mu \in \{l, i\}. \tag{58}$$

3.7.4. Enthalpy Fluxes

The diffusive total enthalpy flux $\text{EF} = \hat{\boldsymbol{n}} \cdot (\rho \boldsymbol{\mathcal{F}}_h)\big|_{\text{sfc}}$ at the surface is the boundary condition for the interior SGS enthalpy flux $\boldsymbol{\mathcal{F}}_h$ defined in Section 3.3. It is the sum of two physically distinct components: a thermal diffusion flux and a flux of total enthalpy advected by the evaporating water mass. For consistency with conventional partitioning, this total flux is decomposed into a sensible heat flux (SHF) and a latent heat flux (LHF).

The latent heat flux is defined conventionally as the energy associated with the phase change of the evaporating water,

$$\text{LHF} = L_{v,0} E, \tag{59}$$

where $E = -\rho_{\text{sfc}} g_h \Delta_s q_v$ is the evaporation rate.

The sensible heat flux is then defined as the sum of all other non-latent energy fluxes. This includes the thermal diffusion flux, which is driven by the near-surface difference in dry static energy ($s_d = h_d + \Phi$), and the sensible heat and potential energy advected by the evaporating water mass, neglecting the (small) kinetic energy:

$$\mathrm{SHF} = -\rho_{\mathrm{sfc}} g_h \Delta_s(s_d) + (h_v + \Phi - L_{v,0})\, E. \tag{60}$$

With these definitions, the total surface enthalpy flux is given by the conventional sum:

$$\mathrm{EF} = \mathrm{SHF} + \mathrm{LHF}. \tag{61}$$

This formulation correctly accounts for the static energy carried by evaporating vapor, in addition to the dry static energy flux.

For this formulation to be consistent, the effective conductance g_h is used for both the thermal flux component of SHF and the water vapor flux (E). This is physically motivated by the assumption that the respective eddy diffusivities are equal (a turbulent Lewis number of unity, see Section 3.3). In the dry-air limit, this formulation is equivalent to classic closures based on the potential temperature difference because vertical dry static energy gradients approximate potential temperature gradients. Appendix A2 shows that this formulation of the surface enthalpy flux transforms such that the model physics remain invariant under shifts of the reference temperature T_0.

Finally, the downward enthalpy flux carried by sedimenting/falling condensate, $-\rho W_h \hat{\boldsymbol{k}}$, satisfies an outgoing condition at the surface, ensuring $\rho W_h|_{\mathrm{sfc}} = \rho W_h|_{\mathrm{int}}$. The sedimentation flux of enthalpy, W_h, is the mass-weighted sum of the constituent total enthalpies:

$$W_h = q_l h_{\mathrm{tot},l} w_l + q_i h_{\mathrm{tot},i} w_i. \tag{62}$$

To maintain a closed energy balance, this energy loss in the atmosphere must be accounted for as an energy gain at the surface.

3.8. Model Top and Sponge Layer

The model top is a rigid lid at fixed height z_t, with insulating and free-slip boundary conditions. To absorb upward-propagating waves and prevent their reflection, we apply damping to all prognostic variables via viscous dissipation, with additional Rayleigh damping for vertical velocities.

3.8.1. Viscous Sponge

We apply horizontal viscous damping to all prognostic variables within a sponge layer. This damps fluctuations near the upper boundary without, for example, imparting an artificial and unphysical zonal-mean torque, as Rayleigh damping of horizontal velocities would (Shepherd et al., 1996). The tendency for a scalar ψ due to viscous damping is

$$\rho \mathcal{S}_\psi = \cdots + \kappa_{\max} \beta_{\mathrm{sponge}} \nabla_h \cdot (\rho \nabla_h \psi), \tag{63}$$

where $\kappa_{\max} = 10^6\ \mathrm{m^2\ s^{-1}}$ is a turbulent diffusivity and $\beta_{\mathrm{sponge}}(z) \in [0,1]$ is a vertical ramp function ensuring a smooth onset of the sponge layer, and the dots indicate other tendencies. We use the following ramp function (J. B. Klemp & Lilly, 1978):

$$\beta_{\mathrm{sponge}}(z) = \begin{cases} 0 & z \le z_d, \\ \sin^2\left(\dfrac{\pi}{2} \dfrac{z - z_d}{z_t - z_d}\right) & z > z_d, \end{cases} \tag{64}$$

where z_d is the altitude where the sponge layer begins. The sponge layer's thickness should be about 1.5 times the vertical wavelength of the waves it aims to absorb, which is around 2 km–10 km in the stratosphere (Durran & Klemp, 1983; Ern et al., 2018; Fritts & Alexander, 2003; J. Klemp et al., 2008; Midgley & Liemohn, 1966; Tsuda, 2014). Our default choice is $z_d = z_t - 15$ km.

For the three-dimensional velocity vector $\boldsymbol{u}$, viscous damping is applied as

$$\boldsymbol{S}_{\boldsymbol{u}} = \cdots + \kappa_{\max}\beta_{\text{sponge}}\Delta_h \boldsymbol{u}, \tag{65}$$

with the horizontal vector Laplacian defined in Equation 40. For simplicity, we neglect horizontal density variations, similar to the hyperdiffusion approach in Equation 42.

3.8.2. Rayleigh Drag

We additionally damp the vertical velocity component $\hat{\boldsymbol{k}} \cdot \boldsymbol{u}$ using linear (Rayleigh) drag:

$$\boldsymbol{S}_{\boldsymbol{u}} = \cdots - \alpha_{\max}\beta_{\text{sponge}}\left(\hat{\boldsymbol{k}} \cdot \boldsymbol{u}\right)\hat{\boldsymbol{k}}. \tag{66}$$

This damping is exclusive to vertical velocities to avoid introducing an unphysical zonal-mean torque, which could lead to artifacts in the upper atmosphere circulation (e.g., Shepherd et al., 1996). We use the same ramp function (Equation 64) as for the viscous sponge. Effective maximum relaxation coefficients are typically around $\alpha_{\max} = 0.2\ \text{s}^{-1}$ (Jablonowski & Williamson, 2011; J. Klemp et al., 2008).

4. Generalized Coordinates

4.1. Coordinate System and Metrics

To enable both higher resolution near the surface and simulations with topography, we employ a height-based, terrain-following, and stretched vertical coordinate ξ^3 and horizontal coordinates $\left(\xi^1, \xi^2\right)$ that parameterize a cubed sphere for simulations on a sphere (Ronchi et al., 1996; Sadourny, 1972). This results in generalized coordinates $\left(\xi^1, \xi^2, \xi^3\right)$ that are curvilinear and non-orthogonal in the presence of topography.

We define the coordinate system through a mapping from generalized coordinates to position vectors $\boldsymbol{r}$ in an embedding Euclidean space,

$$\boldsymbol{r}\left(\xi^1, \xi^2, \xi^3\right) = r^i\left(\xi^1, \xi^2, \xi^3\right)\boldsymbol{x}_i, \tag{67}$$

where $(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)$ are Cartesian basis vectors, and we use Einstein notation for summation over repeated indices. This mapping allows us to compute the covariant basis vectors,

$$\boldsymbol{e}_i = \frac{\partial \boldsymbol{r}}{\partial \xi^i}, \tag{68}$$

which are tangent to the generalized coordinate surfaces (see Figure 1 for an illustration). Because the mapping must also be invertible, we can also obtain the contravariant basis vectors,

$$\boldsymbol{e}^i = \nabla \xi^i, \tag{69}$$

which are orthogonal to the generalized coordinate surfaces. As usual, we use upper indices for contravariant basis vector and components and lower indices for covariant basis vectors and components.

The basis vectors satisfy the orthogonality relation

$$\boldsymbol{e}^i \cdot \boldsymbol{e}_j = \boldsymbol{e}_j \cdot \boldsymbol{e}^i = \delta^i_j, \tag{70}$$

where δ^i_j is the Kronecker delta. The covariant and contravariant components of the metric tensor $\boldsymbol{g}$ are then given by

$$g_{ij} = \boldsymbol{e}_i \cdot \boldsymbol{e}_j \qquad \text{and} \qquad g^{ij} = \boldsymbol{e}^i \cdot \boldsymbol{e}^j. \tag{71}$$

We will also need the covariant and contravariant components of the alternating tensor ε, given by

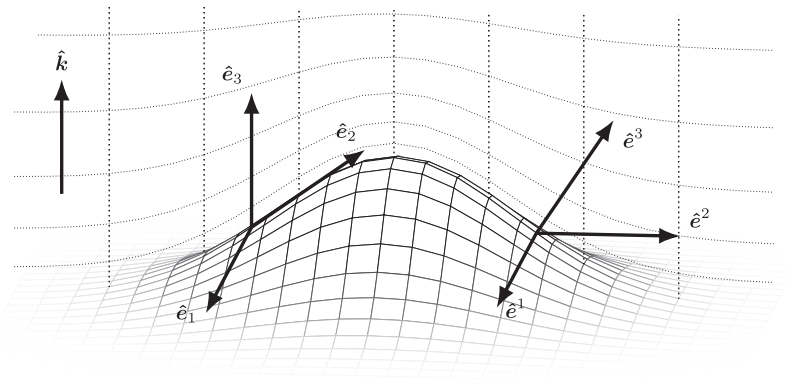

Figure 1. Unit covariant and contravariant basis vectors $\hat{\boldsymbol{e}}_i$ and $\hat{\boldsymbol{e}}^i$ in stretched terrain-following coordinates, with the vertical direction (direction of increasing gravitational potential) denoted by the unit vector $\hat{\boldsymbol{k}}$. The horizontal dotted lines indicate uniformly spaced values of ξ^3 in the plane where $\xi^1 = 0$, and the vertical dotted lines indicate uniformly spaced values of ξ^2 in this plane. As one moves upward, $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ become more orthogonal to gravity, while $\boldsymbol{e}_3$ increases in length.

$$\mathcal{E}_{ijk} = \boldsymbol{e}_i \cdot (\boldsymbol{e}_j \times \boldsymbol{e}_k) = J\varepsilon_{ijk} \qquad \text{and} \qquad \mathcal{E}^{ijk} = \boldsymbol{e}^i \cdot \left(\boldsymbol{e}^j \times \boldsymbol{e}^k\right) = \frac{1}{J}\varepsilon^{ijk}, \tag{72}$$

where $\varepsilon_{ijk} = \varepsilon^{ijk}$ is the Levi-Civita symbol, and $J = (\det(\boldsymbol{g}))^{1/2}$ is the Jacobian determinant (volume element in generalized coordinates).

Any vector $\boldsymbol{u}$ can be expressed in terms of its covariant (u_i) or contravariant (u^i) components,

$$\boldsymbol{u} = u_i \boldsymbol{e}^i = u^i \boldsymbol{e}_i, \tag{73}$$

which are related by

$$u_i = g_{ij} u^j \qquad \text{and} \qquad u^i = g^{ij} u_j. \tag{74}$$

Inner and cross products of vectors can be written as

$$\boldsymbol{u} \cdot \boldsymbol{v} = g^{ij} u_i v_j = g_{ij} u^i v^j, \tag{75}$$

and

$$\boldsymbol{u} \times \boldsymbol{v} = \mathcal{E}^{ijk} \boldsymbol{e}_i u_j v_k = \mathcal{E}_{ijk} \boldsymbol{e}^i u^j v^k. \tag{76}$$

The gradient, divergence, and curl operators can be expressed as

$$\nabla = \boldsymbol{e}^i \frac{\partial}{\partial \xi^i}, \tag{77a}$$

$$\nabla \cdot = \frac{1}{J} \frac{\partial}{\partial \xi^i} J \left(\boldsymbol{e}^i\right)^{\intercal}, \tag{77b}$$

$$\nabla \times = \mathcal{E}^{ijk} \boldsymbol{e}_i \frac{\partial}{\partial \xi^j} (\boldsymbol{e}_k)^{\intercal}, \tag{77c}$$

where $(\cdot)^{\intercal}$ denotes the transpose.

4.2. Equations of Motion

Leveraging standard tensor calculus (Kajishima & Taira, 2017; Appendix A), we express the governing Equation 26 in generalized coordinates $\left(\xi^1, \xi^2, \xi^3\right)$ as follows:

1. Mass continuity

$$\frac{\partial}{\partial t}\rho + \underbrace{\frac{1}{J}\frac{\partial}{\partial \xi^j}\left[\rho J\left(u^j - W_{q_t}\hat{k}^j\right)\right]}_{\text{VI}} = \rho\,\hat{\mathcal{S}}_{q_t} \tag{78a}$$

2. Momentum

$$\begin{aligned}\frac{\partial}{\partial t}u_i + \mathcal{E}_{ikl}\left(2\Omega^k + \omega^k\right)u^l = &\underbrace{-c_{pd}\left(\theta_v - \theta_{v,r}\right)\frac{\partial \Pi}{\partial \xi^i} - \frac{\partial}{\partial \xi^i}\left(\Phi - \Phi_r\right)}_{\text{VI}} \\ &- \frac{\partial}{\partial \xi^i}\kappa + \underbrace{\mathcal{S}_{\boldsymbol{u},i}}_{\text{VI}} - \mathcal{D}^4_{\boldsymbol{u},i} - \underbrace{\frac{1}{\rho J}\nabla_j\left(\rho J g_{ik}\mathcal{T}^{kj}\right)}_{\text{VI}}.\end{aligned} \tag{78b}$$

3. Total energy

$$\begin{aligned}\frac{\partial}{\partial t}\left(\rho e_{\text{tot}}\right) + \underbrace{\frac{1}{J}\frac{\partial}{\partial \xi^j}\left[\rho J\left(h_{\text{tot}}u^j - W_h\hat{k}^j\right)\right]}_{\text{VI}} = &\underbrace{-\frac{1}{J}\frac{\partial}{\partial \xi^j}\left(\rho J\mathcal{F}^j_h\right)}_{\text{VI}} \\ &- \frac{1}{J}\frac{\partial}{\partial \xi^j}\left[\rho J\left(\mathcal{F}^j_R + \mathcal{H}^j_h + \left(\boldsymbol{u}\cdot\boldsymbol{\mathcal{T}}\right)^j\right)\right]\end{aligned} \tag{78c}$$

4. Total water

$$\frac{\partial}{\partial t}\left(\rho q_t\right) + \underbrace{\frac{1}{J}\frac{\partial}{\partial \xi^j}\left[\rho J\left(q_t u^j - W_{q_t}\hat{k}^j\right)\right]}_{\text{VI}} = \underbrace{-\frac{1}{J}\frac{\partial}{\partial \xi^j}\left(\rho J\mathcal{F}^j_{q_t}\right)}_{\text{VI}} - \frac{1}{J}\frac{\partial}{\partial \xi^j}\left(\rho J\mathcal{H}^j_{q_t}\right) \tag{78d}$$

5. Condensate/precipitation (for $\mu \in \{l, i\}$ and $\sigma \in \{\cdot, \text{cl}, \text{pr}\}$ as needed)

$$\frac{\partial}{\partial t}\left(\rho q^\sigma_\mu\right) + \underbrace{\frac{1}{J}\frac{\partial}{\partial \xi^j}\left[\rho J q^\sigma_\mu\left(u^j - w^\sigma_\mu\hat{k}^j\right)\right]}_{\text{VI}} = \rho\mathcal{S}_{q^\sigma_\mu} - \underbrace{\frac{1}{J}\frac{\partial}{\partial \xi^j}\left(\rho J\mathcal{F}^j_{q^\sigma_\mu}\right)}_{\text{VI}} - \frac{1}{J}\frac{\partial}{\partial \xi^j}\left(\rho J\mathcal{H}^j_{q^\sigma_\mu}\right) \tag{78e}$$

6. Tracer transport

$$\frac{\partial}{\partial t}\left(\rho\chi\right) + \underbrace{\frac{1}{J}\frac{\partial}{\partial \xi^j}\left[\rho J\chi\left(u^j - w_\chi\hat{k}^j\right)\right]}_{\text{VI}} = \rho\mathcal{S}_\chi - \underbrace{\frac{1}{J}\frac{\partial}{\partial \xi^j}\left(\rho J\mathcal{F}^j_\chi\right)}_{\text{VI}} - \frac{1}{J}\frac{\partial}{\partial \xi^j}\left(\rho J\mathcal{H}^j_\chi\right) \tag{78f}$$

The following symbols appear in the equations:

- The specific kinetic energy of the resolved motions is

$$\kappa = \frac{g^{kl} u_k u_l}{2}; \tag{79}$$

- The contravariant vorticity component is

$$\omega^i = \mathcal{E}^{ijk} \frac{\partial u_k}{\partial \xi^j}. \tag{80}$$

- The contravariant components of the vertical unit vector are $\hat{k}^j = \delta^j_3$, so that the contravariant sedimentation/fall velocities only have contravariant vertical components.
- The operator ∇_j denotes the covariant derivative, which includes curvature terms (Christoffel symbols) that account for changes of the basis vectors in space.

To avoid explicitly computing the curvature terms, as is common in computational fluid dynamics, we evaluate the tensor divergence by transforming to Cartesian coordinates, computing the divergence there, and transforming back (Vinokur, 1974), which avoids undifferentiated terms (Christoffel symbols) that would make ensuring discrete conservation difficult. However, in the examples presented in this paper, the SGS momentum flux tensor only has contravariant vertical components.

Hyperdiffusive horizontal scalar fluxes are computed by repeated application of gradient and divergence operators; that is, for a generic scalar ψ,

$$\mathcal{H}^j_\psi = \nu_\psi g^{jk} \frac{\partial(\Delta_h \psi)}{\partial \xi^k}, \qquad j,k \in \{1,2\} \tag{81}$$

where

$$\Delta_h \psi = \frac{1}{J} \frac{\partial}{\partial \xi^j} \left(J g^{jk} \frac{\partial \psi}{\partial \xi^k} \right), \qquad j,k \in \{1,2\} \tag{82}$$

is the horizontal Laplacian of ψ on the terrain-following ξ^1 and ξ^2 coordinate surfaces. For the tensor hyperdiffusion in the horizontal components ($i = 1,2$) of the momentum equation, we use the simplified form (Equation 42), expressed in generalized coordinates.

We adopt the approach of Gardner et al. (2018) and use the covariant velocity components u_i as prognostic variables. This leads to an equation set where vertical ($i = 3$) pressure and geopotential gradients are isolated to the covariant vertical velocity equation. This simplifies the maintenance of hydrostatic balance and the implementation of implicit solves in the vertical direction, as only the $i = 3$ component of the momentum equations needs to be involved in those implicit solves. In anticipation of the description of the timestepping strategy in Section 5.7, the terms that are treated as vertically implicit (VI) are marked in the above Equation (78).

5. Discretization

This section details the discretization of the governing equations. Our approach is a hybrid one, combining a horizontal spectral element method (SEM) with a vertical finite-difference method on a staggered grid. While the foundational principles of the SEM are well-established in the literature (e.g., M. A. Taylor & Fournier, 2010; M. A. Taylor et al., 2020), our specific implementation presents unique challenges and solutions that merit a detailed description. In particular, the use of a height-based, terrain-following coordinate (as opposed to, e.g., mass-based coordinates) and the strict requirement to discretely conserve total energy, mass, and water necessitate a careful and specific treatment of the reconstructions and interpolations on the staggered grids. These details, while subtle, are crucial for the model's conservation properties and overall robustness. The following subsections therefore

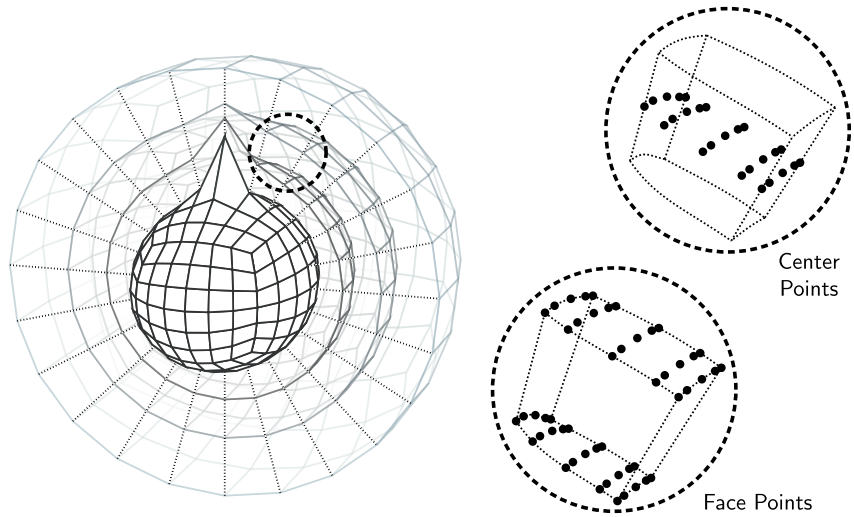

Figure 2. Finite element discretization for an atmosphere with a large mountain at one pole. The circled elements on the right show Gauss-Lobatto-Legendre points at layer centers and faces. Grid parameters: $N_p = 4$, $N_v = 3$, and $N_h = 6^3 = 216$ (6 panels of an equiangular cubed sphere with 6 elements along each side).

outline the necessary components of our hybrid scheme, focusing on the aspects that ensure its fidelity and conservative nature.

5.1. Cubed Sphere Mesh in a Deep Atmosphere

To decouple horizontal (ξ^1 and ξ^2) and vertical (ξ^3) operations, we discretize the domain with N_h horizontal elements, extruded vertically into N_v layers with a stretched, terrain-following grid (Appendix B), resulting in $N_h \times N_v$ total elements. For simulations in a spherical shell, we employ an equiangular cubed sphere grid (Ronchi et al., 1996; Sadourny, 1972); see Figure 2 for an illustration. For simulations in Cartesian geometry, we apply the same framework to a box domain.

Using a local coordinate system ($\xi^1, \xi^2 \in [-1, 1]$ and $\xi^3 \in [0, 1]$) in each element (indexed horizontally by n_h and vertically by n_v), the radial unit vector $\hat{\boldsymbol{k}}$ is given by

$$\hat{\boldsymbol{k}}\left(\xi^1, \xi^2; n_h\right) = \frac{\boldsymbol{k}\left(\xi^1, \xi^2; n_h\right)}{\left\|\boldsymbol{k}\left(\xi^1, \xi^2; n_h\right)\right\|}, \tag{83}$$

where the non-normalized radial vector $\boldsymbol{k}$ is a bilinear interpolation of the quadrilateral's vertex coordinates $\hat{\boldsymbol{k}}_{n_h}^{\mathrm{SE}}$, $\hat{\boldsymbol{k}}_{n_h}^{\mathrm{NE}}$, $\hat{\boldsymbol{k}}_{n_h}^{\mathrm{NW}}$, and $\hat{\boldsymbol{k}}_{n_h}^{\mathrm{SW}}$ on the cubed unit sphere (Guba et al., 2014):

$$\begin{aligned}\boldsymbol{k}\left(\xi^1, \xi^2; n_h\right) = \frac{1}{4}\big[&\left(1+\xi^1\right)\left(1-\xi^2\right)\hat{\boldsymbol{k}}_{n_h}^{\mathrm{SE}} + \left(1+\xi^1\right)\left(1+\xi^2\right)\hat{\boldsymbol{k}}_{n_h}^{\mathrm{NE}} \\ &+ \left(1-\xi^1\right)\left(1+\xi^2\right)\hat{\boldsymbol{k}}_{n_h}^{\mathrm{NW}} + \left(1-\xi^1\right)\left(1-\xi^2\right)\hat{\boldsymbol{k}}_{n_h}^{\mathrm{SW}}\big].\end{aligned} \tag{84}$$

Projecting the unit vector $\hat{\boldsymbol{k}}$ onto the element's bottom and top faces at elevations z_B and z_T, the elevation of points in the element is given by linear interpolation:

$$z\left(\xi^1, \xi^2, \xi^3; n_h, n_v\right) = \left(1-\xi^3\right) z_B\left(\xi^1, \xi^2; n_h, n_v\right) + \xi^3 z_T\left(\xi^1, \xi^2; n_h, n_v\right). \tag{85}$$

The transformation to physical coordinates is then

$$\boldsymbol{r}\left(\xi^1, \xi^2, \xi^3; n_h, n_v\right) = r\left(\xi^1, \xi^2, \xi^3; n_h, n_v\right) \hat{\boldsymbol{k}}\left(\xi^1, \xi^2; n_h\right), \tag{86}$$

where $r = R + z$ is the norm of the position vector $\boldsymbol{r}$, and R is the fixed radius of the reference sphere (e.g., Earth's mean radius). The horizontal covariant basis vectors $\boldsymbol{e}_i$ $(i = 1, 2)$ for this transformation are

$$\begin{aligned}\boldsymbol{e}_i(\xi^1,\xi^2,\xi^3;n_h,n_v) = \hat{\boldsymbol{k}}(\xi^1,\xi^2;n_h)\frac{\partial}{\partial\xi^i}z(\xi^1,\xi^2,\xi^3;n_h,n_v)\\ + r(\xi^1,\xi^2,\xi^3;n_h,n_v)\frac{\partial}{\partial\xi^i}\hat{\boldsymbol{k}}(\xi^1,\xi^2;n_h),\end{aligned} \tag{87}$$

and the vertical covariant basis vector $\boldsymbol{e}_3$ is

$$\boldsymbol{e}_3(\xi^1,\xi^2;n_h,n_v) = \left[z_T(\xi^1,\xi^2;n_h,n_v) - z_B(\xi^1,\xi^2;n_h,n_v)\right]\hat{\boldsymbol{k}}(\xi^1,\xi^2;n_h). \tag{88}$$

All horizontal derivatives, including those that appear in $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, are evaluated using the spectral element method, and discontinuities in the derivatives are eliminated through direct stiffness summation (Section 5.2).

We approximate Earth's geoid and geopotential surfaces as spherical, but we retain the full deep-atmosphere equations and geometry (White et al., 2005), without using the shallow-atmosphere approximation ($r \approx R$). (However, the model implementation also supports the shallow atmosphere approximation, with the corresponding adjustments to metric terms and equations of motion.)

5.2. Horizontal Spectral Elements

We use the continuous Galerkin spectral element method (SEM) for horizontal discretization (Karniadakis & Sherwin, 2005; M. A. Taylor & Fournier, 2010; M. A. Taylor et al., 2020). The SEM approximates a variable field $\psi(\boldsymbol{r})$ via the nodal point representation $\psi \approx \mathcal{I}\psi$, with

$$\mathcal{I}\psi\left[\boldsymbol{r}(\xi^1,\xi^2,\xi^3;n_h,n_v)\right] = \hat{\psi}_{n_h,n_v}(\xi^1,\xi^2,\xi^3), \tag{89}$$

where $\hat{\psi}_{n_h,n_v}$ is a polynomial of order N_p in ξ^1 and ξ^2. We represent $\hat{\psi}_{n_h,n_v}$ using tensor products of Lagrange polynomials $L^{n_i}_{N_p}$ through $N_p + 1$ Gauss-Lobatto-Legendre (GLL) quadrature points $\hat{\zeta}^{n_i}_{N_p}$ (Appendix C):

$$\hat{\psi}_{n_h,n_v}(\xi^1,\xi^2,\xi^3) = \sum_{n_1,n_2=1}^{N_p+1} L^{n_1}_{N_p}(\xi^1)\,L^{n_2}_{N_p}(\xi^2)\hat{\psi}_{n_h,n_v}(\hat{\zeta}^{n_1}_{N_p},\hat{\zeta}^{n_2}_{N_p},\xi^3). \tag{90}$$

The continuous Galerkin SEM interpolates between the values of ψ at nodal points,

$$\psi\left[\boldsymbol{r}(\hat{\zeta}^{n_1}_{N_p},\hat{\zeta}^{n_2}_{N_p},\xi^3;n_h,n_v)\right] = \hat{\psi}_{n_h,n_v}(\hat{\zeta}^{n_1}_{N_p},\hat{\zeta}^{n_2}_{N_p},\xi^3), \tag{91}$$

meaning that $\mathcal{I}\psi$ is continuous across element boundaries as long as ψ is continuous.

To ensure continuity at boundaries, we use the direct stiffness summation (DSS) operator $\mathcal{P}$ (Deville et al., 2002; M. A. Taylor & Fournier, 2010). This operator replaces the value at each boundary point with a volume-weighted average over its collocated neighborhood,

$$\mathcal{P}(\mathcal{I}\psi)\left[\boldsymbol{r}(\hat{\zeta}^{n_1}_{N_p},\hat{\zeta}^{n_2}_{N_p},\xi^3;n_h,n_v)\right] = \frac{\sum_{(m_1,m_2,m_h)\in C^{n_1,n_2}_{n_h}} \hat{\psi}_{m_h,n_v}(\hat{\zeta}^{m_1}_{N_p},\hat{\zeta}^{m_2}_{N_p},\xi^3)\delta V(\hat{\zeta}^{m_1}_{N_p},\hat{\zeta}^{m_2}_{N_p},\xi^3;m_h,n_v)}{\sum_{(m_1,m_2,m_h)\in C^{n_1,n_2}_{n_h}} \delta V(\hat{\zeta}^{m_1}_{N_p},\hat{\zeta}^{m_2}_{N_p},\xi^3;m_h,n_v)}. \tag{92}$$

The collocated neighborhood of $\boldsymbol{r}(\hat{\zeta}^{n_1}_{N_p},\hat{\zeta}^{n_2}_{N_p},\xi^3;n_h,n_v)$ is the set of indices in horizontally adjacent elements that correspond to the same physical location,

$$C^{n_1,n_2}_{n_h} = \left\{(m_1,m_2,m_h) \;\middle|\; \boldsymbol{r}(\hat{\zeta}^{m_1}_{N_p},\hat{\zeta}^{m_2}_{N_p},\xi^3;m_h,n_v) = \boldsymbol{r}(\hat{\zeta}^{n_1}_{N_p},\hat{\zeta}^{n_2}_{N_p},\xi^3;n_h,n_v)\right\}. \tag{93}$$

The volume differential

$$\delta V\left(\hat{\zeta}_{N_p}^{n_1}, \hat{\zeta}_{N_p}^{n_2}, \xi^3; n_h, n_v\right) = w_{N_p}^{n_1} w_{N_p}^{n_2} J\left(\hat{\zeta}_{N_p}^{n_1}, \hat{\zeta}_{N_p}^{n_2}, \xi^3; n_h, n_v\right) \tag{94}$$

is formed from the GLL quadrature weights $w_{N_p}^{n_i}$ of the nodal points $\hat{\zeta}_{N_p}^{n_i}$ and a discrete approximation of the metric Jacobian J (Appendix D). When applying $\mathcal{P}$ to vector fields, we first convert the vectors to Cartesian representations, apply $\mathcal{P}$ to each component, and then convert back. DSS is the only operation requiring horizontal communication between elements, thus enabling parallel computing efficiency.

For any continuous interpolation $\mathcal{I}\phi$, the DSS operator preserves the discrete inner product,

$$\mathscr{I}_{\Omega_h}[\mathcal{I}\phi, \mathcal{P}(\mathcal{I}\psi)] = \mathscr{I}_{\Omega_h}[\mathcal{I}\psi, \mathcal{P}(\mathcal{I}\phi)]. \tag{95}$$

This approximates the integral over the horizontal domain Ω_h (a volume of height $\delta\xi^3 = 1$ centered on coordinate ξ^3 in the layer of elements with vertical index n_v), using nodal point quadrature along the horizontal direction and the midpoint rule along the vertical direction:

$$\begin{aligned} \int_{\Omega_h\left(\xi^3; n_v\right)} (\mathcal{I}\phi)(\mathcal{I}\psi)\, dV &\approx \mathscr{I}_{\Omega_h\left(\xi^3; n_v\right)}(\mathcal{I}\phi, \mathcal{I}\psi) \\ &= \sum_{n_h=1}^{N_h} \sum_{n_1,n_2=1}^{N_p+1} \hat{\phi}_{n_h,n_v}\left(\hat{\zeta}_{N_p}^{n_1}, \hat{\zeta}_{N_p}^{n_2}, \xi^3\right) \hat{\psi}_{n_h,n_v}\left(\hat{\zeta}_{N_p}^{n_1}, \hat{\zeta}_{N_p}^{n_2}, \xi^3\right) \delta V\left(\hat{\zeta}_{N_p}^{n_1}, \hat{\zeta}_{N_p}^{n_2}, \xi^3; n_h, n_v\right). \end{aligned} \tag{96}$$

Preserving this inner product enables differential operators to discretely conserve air mass, energy, and other quantities (Appendix E).

5.3. SEM Differential Operators

The SEM provides strong and weak formulations of horizontal differential operators. The strong formulation approximates the derivative $\partial\psi/\partial\xi^i$ $(i = 1, 2)$ by differentiating the basis functions in the nodal point representation (Equation 90):

$$\frac{\partial}{\partial\xi^i}\hat{\psi}_{n_h,n_v}\left(\hat{\zeta}_{N_p}^{n_1}, \hat{\zeta}_{N_p}^{n_2}, \xi^3\right) = \begin{cases} \sum_{m_1=1}^{N_p+1} L_{N_p}^{m_1\prime}\left(\hat{\zeta}_{N_p}^{n_1}\right) \hat{\psi}_{n_h,n_v}\left(\hat{\zeta}_{N_p}^{m_1}, \hat{\zeta}_{N_p}^{n_2}, \xi^3\right) & \text{if } i = 1, \\ \sum_{m_2=1}^{N_p+1} L_{N_p}^{m_2\prime}\left(\hat{\zeta}_{N_p}^{n_2}\right) \hat{\psi}_{n_h,n_v}\left(\hat{\zeta}_{N_p}^{n_1}, \hat{\zeta}_{N_p}^{m_2}, \xi^3\right) & \text{if } i = 2. \end{cases} \tag{97}$$

The components of the three-dimensional differential operators (Equation 77) involving horizontal derivatives are then

$$\nabla_h = \boldsymbol{e}^1 \frac{\partial}{\partial\xi^1} + \boldsymbol{e}^2 \frac{\partial}{\partial\xi^2}, \tag{98a}$$

$$\nabla_h \cdot = \frac{1}{J}\frac{\partial}{\partial\xi^1} J\left(\boldsymbol{e}^1\right)^{\intercal} + \frac{1}{J}\frac{\partial}{\partial\xi^2} J\left(\boldsymbol{e}^2\right)^{\intercal}, \tag{98b}$$

$$\nabla_h \times = \left(\frac{\boldsymbol{e}_1}{J}\frac{\partial}{\partial\xi^2} - \frac{\boldsymbol{e}_2}{J}\frac{\partial}{\partial\xi^1}\right)(\boldsymbol{e}_3)^{\intercal} - \frac{\boldsymbol{e}_3}{J}\left(\frac{\partial}{\partial\xi^2}(\boldsymbol{e}_1)^{\intercal} - \frac{\partial}{\partial\xi^1}(\boldsymbol{e}_2)^{\intercal}\right), \tag{98c}$$

where J is a discrete approximation of the metric Jacobian (Appendix D). These operators satisfy the identities $\nabla_h \times \nabla_h = 0$ and $\nabla_h \cdot \nabla_h \times = 0$ (M. A. Taylor & Fournier, 2010).

The weak formulation (indicated by tilde accents) modifies the derivative to satisfy an integration by parts identity, ensuring that the weak derivative is the negative adjoint of the strong derivative:

$$\mathscr{I}_{\Omega_h}\left(\mathcal{I}\phi, \frac{1}{J}\frac{\tilde{\partial}}{\tilde{\partial}\xi^i}\mathcal{I}\psi\right) + \mathscr{I}_{\Omega_h}\left(\mathcal{I}\psi, \frac{1}{J}\frac{\partial}{\partial\xi^i}\mathcal{I}\phi\right) = 0. \tag{99}$$

The weak derivative can be computed as

$$\frac{\tilde{\partial}}{\tilde{\partial}\xi^i}\hat{\psi}_{n_h,n_v}\left(\hat{\zeta}^{n_1}_{N_p}, \hat{\zeta}^{n_2}_{N_p}, \xi^3\right) = \begin{cases} -\sum_{m_1=1}^{N_p+1} \frac{w^{m_1}_{N_p}}{w^{n_1}_{N_p}} L^{n_1\prime}_{N_p}\left(\hat{\zeta}^{m_1}_{N_p}\right)\hat{\psi}_{n_h,n_v}\left(\hat{\zeta}^{m_1}_{N_p}, \hat{\zeta}^{n_2}_{N_p}, \xi^3\right) & \text{if } i = 1, \\ -\sum_{m_2=1}^{N_p+1} \frac{w^{m_2}_{N_p}}{w^{n_2}_{N_p}} L^{n_2\prime}_{N_p}\left(\hat{\zeta}^{m_2}_{N_p}\right)\hat{\psi}_{n_h,n_v}\left(\hat{\zeta}^{n_1}_{N_p}, \hat{\zeta}^{m_2}_{N_p}, \xi^3\right) & \text{if } i = 2. \end{cases} \tag{100}$$

The corresponding weak differential operators, obtained by using weak derivatives in Equation 98, also satisfy the identities $\tilde{\nabla}_h \times \tilde{\nabla}_h = 0$ and $\tilde{\nabla}_h \cdot \tilde{\nabla}_h \times = 0$. Both formulations offer computational efficiency, computing horizontal derivatives at $\mathcal{O}(N_p^2)$ nodal points in $\mathcal{O}(N_p)$ operations per node (M. A. Taylor & Fournier, 2010).

As in the three-dimensional continuous Galerkin SEM (M. A. Taylor & Fournier, 2010), the integration by parts identity (Equation 99) leads to discrete versions of divergence and Stokes' theorems (Appendix E). To ensure global flux conservation for scalars and vorticity, we use the weak formulation for flux divergences and for curl terms in the momentum equation. (See Section 5.6 for further details.) Conversely, for gradients in the momentum equation, we use the strong formulation to conserve kinetic energy when pressure gradients and sources are absent.

We evaluate horizontal scalar and vector Laplacians using a mixed formulation,

$$\tilde{\Delta}_h \mathcal{I}\psi = \tilde{\nabla}_h \cdot \nabla_h \mathcal{I}\psi, \tag{101a}$$

$$\tilde{\Delta}_h \mathcal{I}\boldsymbol{u} = \nabla_h \tilde{\nabla}_h \cdot \mathcal{I}\boldsymbol{u} - \nabla_h \times \tilde{\nabla}_h \times \mathcal{I}\boldsymbol{u}, \tag{101b}$$

which satisfies a second-order analogue of integration by parts (Appendix E). For fourth-order hyperdiffusion tendencies, we apply the Laplacian $\tilde{\Delta}_h$ twice, with the DSS operator $\mathcal{P}$ applied in between. This eliminates discontinuities while preserving the inner product, ensuring global conservation for diffusion and hyperdiffusion terms.

5.4. Vertical Staggered Grid

To suppress computational modes in the vertical, we use a mass-weighted Lorenz-staggered finite difference method, where vertical velocities are defined on element faces and all other variables on element centers (Thuburn & Woollings, 2005; M. A. Taylor et al., 2020). While other vertical arrangements of prognostic variables exist, such as the Charney-Philips staggering, we chose the Lorenz staggering for its relative simplicity and proven robustness. The analysis by Thuburn and Woollings (2005) suggests that while the Charney-Philips staggering may offer marginal benefits for the numerical propagation of certain waves, the Lorenz staggering also performs well. Given our primary focus on the exact conservation of fundamental extensive quantities, the simplicity of the Lorenz grid was deemed a favorable trade-off against the small potential gains from a more complex staggering.

Staggered grids represent the nodal point interpolation $\mathcal{I}\psi(\boldsymbol{r})$ either using values ψ^{f} at $N_v + 1$ element faces (with $n_v = 1$ at the surface),

$$\psi^{\mathrm{f}}_{n_v}\left(\xi^1, \xi^2; n_h\right) = \begin{cases} \mathcal{I}\psi\left[\boldsymbol{r}\left(\xi^1, \xi^2, 0; n_h, n_v\right)\right] & \text{if } n_v \le N_v, \\ \mathcal{I}\psi\left[\boldsymbol{r}\left(\xi^1, \xi^2, 1; n_h, n_v - 1\right)\right] & \text{if } n_v = N_v + 1, \end{cases} \tag{102}$$

or values ψ^{c} at N_v element centers,

$$\psi^{\mathrm{c}}_{n_v}\left(\xi^1,\xi^2;n_h\right) = \mathcal{I}\psi\left[\boldsymbol{r}\left(\xi^1,\xi^2,1/2;n_h,n_v\right)\right]. \tag{103}$$

Lorenz staggering represents covariant vertical velocity components $\boldsymbol{u}_v = u_3\boldsymbol{e}^3$ by face values and horizontal velocity components $\boldsymbol{u}_h = u_1\boldsymbol{e}^1 + u_2\boldsymbol{e}^2$, together with all other variables, by center values. The vertical coordinate levels are uniformly spaced in the generalized coordinate ξ^3 but are generally stretched and follow the terrain in physical space (Appendix B).

To convert between faces and centers, we use an arithmetic average,

$$\langle\psi^{\mathrm{f}}\rangle^{\mathrm{c}}_{n_v} = \left(\psi^{\mathrm{f}}_{n_v} + \psi^{\mathrm{f}}_{n_v+1}\right)/2, \tag{104a}$$

$$\langle\psi^{\mathrm{c}}\rangle^{\mathrm{f}}_{n_v} = \begin{cases} \psi^{\mathrm{c}}_1 & \text{if } n_v = 1, \\ \left(\psi^{\mathrm{c}}_{n_v} + \psi^{\mathrm{c}}_{n_v-1}\right)/2 & \text{if } 2 \le n_v \le N_v, \\ \psi^{\mathrm{c}}_{N_v} & \text{if } n_v = N_v + 1, \end{cases} \tag{104b}$$

with constant extrapolation to the boundaries of the domain (M. A. Taylor et al., 2020), and a mass-weighted average,

$$\lceil\psi^{\mathrm{f}}\rfloor^{\mathrm{c}} = \frac{1}{(\rho J)^{\mathrm{c}}}\langle(\rho\psi J)^{\mathrm{f}}\rangle^{\mathrm{c}}, \tag{105a}$$

$$\lceil\psi^{\mathrm{c}}\rfloor^{\mathrm{f}} = \frac{1}{(\rho J)^{\mathrm{f}}}\langle(\rho\psi J)^{\mathrm{c}}\rangle^{\mathrm{f}}. \tag{105b}$$

We apply the averaging operators componentwise to vector fields; for example, the available averages of covariant vector fields on cell centers are

$$\langle(\boldsymbol{u}_i)^{\mathrm{c}}\rangle^{\mathrm{f}} = \langle(u_i)^{\mathrm{c}}\rangle^{\mathrm{f}}\left(\boldsymbol{e}^i\right)^{\mathrm{f}}, \tag{106a}$$

$$\lceil(\boldsymbol{u}_i)^{\mathrm{c}}\rfloor^{\mathrm{f}} = \lceil(u_i)^{\mathrm{c}}\rfloor^{\mathrm{f}}\left(\boldsymbol{e}^i\right)^{\mathrm{f}}. \tag{106b}$$

These operators satisfy a density-weighted averaging by parts identity for both scalar and vector fields,

$$\mathscr{I}^{\mathrm{c}}_{\Omega_v}\left[(\rho\phi)^{\mathrm{c}}, \lceil\psi^{\mathrm{f}}\rfloor^{\mathrm{c}}\right] - \mathscr{I}^{\mathrm{f}}_{\Omega_v}\left[(\rho\psi)^{\mathrm{f}}, \langle\phi^{\mathrm{c}}\rangle^{\mathrm{f}}\right] = 0, \tag{107a}$$

$$\mathscr{I}^{\mathrm{f}}_{\Omega_v}\left[(\rho\phi)^{\mathrm{f}}, \lceil\psi^{\mathrm{c}}\rfloor^{\mathrm{f}}\right] - \mathscr{I}^{\mathrm{c}}_{\Omega_v}\left[(\rho\psi)^{\mathrm{c}}, \langle\phi^{\mathrm{f}}\rangle^{\mathrm{c}}\right] = 0, \tag{107b}$$

where the inner product over the vertical domain Ω_v, to second-order accuracy, is

$$\int_{\Omega_v\left(\xi^{n_1}_{N_p}, \xi^{n_2}_{N_p}; n_h\right)} \phi^{\mathrm{c}}\psi^{\mathrm{c}}\,\delta A\,dz \approx \mathscr{I}^{\mathrm{c}}_{\Omega_v\left(\xi^{n_1}_{N_p}, \xi^{n_2}_{N_p}; n_h\right)}\left(\phi^{\mathrm{c}}, \psi^{\mathrm{c}}\right) = \sum_{n_v=1}^{N_v}\left(\phi\psi\delta V\right)^{\mathrm{c}}_{n_v}\left(\xi^{n_1}_{N_p}, \xi^{n_2}_{N_p}; n_h\right) \tag{108}$$

for center interpolations and

$$\begin{aligned}\int_{\Omega_v\left(\xi^{n_1}_{N_p}, \xi^{n_2}_{N_p}; n_h\right)} \phi^{\mathrm{f}}\psi^{\mathrm{f}}\,\delta A\,dz \approx \mathscr{I}^{\mathrm{f}}_{\Omega_v\left(\xi^{n_1}_{N_p}, \xi^{n_2}_{N_p}; n_h\right)}\left(\phi^{\mathrm{f}}, \psi^{\mathrm{f}}\right) = \sum_{n_v=1}^{N_v}\frac{1}{2}\Big[&\left(\phi\psi\delta V\right)^{\mathrm{f}}_{n_v}\left(\xi^{n_1}_{N_p}, \xi^{n_2}_{N_p}; n_h\right) \\ &+ \left(\phi\psi\delta V\right)^{\mathrm{f}}_{n_v+1}\left(\xi^{n_1}_{N_p}, \xi^{n_2}_{N_p}; n_h\right)\Big]\end{aligned} \tag{109}$$

for face interpolations, with $\delta A = \delta V/(z_T - z_B)$ denoting the area differential for each nodal point. Preserving this inner product allows combinations of averaging and differential operators to satisfy discrete conservation laws (Appendix E).

5.5. Vertical Differential Operators

We approximate vertical derivatives by centered differences between adjacent faces or centers,

$$\left(\frac{\partial}{\partial \xi^3}\psi^{\mathrm{f}}\right)^{\mathrm{c}}_{n_v} = \psi^{\mathrm{f}}_{n_v+1} - \psi^{\mathrm{f}}_{n_v}, \tag{110a}$$

$$\left(\frac{\partial}{\partial \xi^3}\psi^{\mathrm{c}}\right)^{\mathrm{f}}_{n_v} = \begin{cases} 0 & \text{if } n_v = 1 \text{ or } n_v = N_v + 1, \\ \psi^{\mathrm{c}}_{n_v} - \psi^{\mathrm{c}}_{n_v-1} & \text{if } 2 \le n_v \le N_v. \end{cases} \tag{110b}$$

At non-boundary points, these approximations commute with the arithmetic mean $\langle\cdot\rangle$ and satisfy a discrete product rule. For inputs on faces, this can be expressed as

$$\left\langle \frac{\partial}{\partial \xi^3}\psi^{\mathrm{f}} \right\rangle^{\mathrm{f}} = \frac{\partial}{\partial \xi^3}\langle \psi^{\mathrm{f}} \rangle^{\mathrm{c}}, \tag{111a}$$

$$\frac{\partial}{\partial \xi^3}(\phi\psi)^{\mathrm{f}} = \langle \phi^{\mathrm{f}} \rangle^{\mathrm{c}} \frac{\partial}{\partial \xi^3}\psi^{\mathrm{f}} + \langle \psi^{\mathrm{f}} \rangle^{\mathrm{c}} \frac{\partial}{\partial \xi^3}\phi^{\mathrm{f}}. \tag{111b}$$

The components of the differential operators Equation 77 involving vertical derivatives are

$$\nabla_v \psi^{\mathrm{f}} = \left(\boldsymbol{e}^3\right)^{\mathrm{c}} \frac{\partial}{\partial \xi^3}\psi^{\mathrm{f}}, \tag{112a}$$

$$\nabla_v \cdot \boldsymbol{u}^{\mathrm{f}} = \frac{1}{J^{\mathrm{c}}}\frac{\partial}{\partial \xi^3}\left(Ju^3\right)^{\mathrm{f}}, \tag{112b}$$

$$\nabla_v \times \boldsymbol{u}^{\mathrm{f}} = \frac{(\boldsymbol{e}_2)^{\mathrm{c}}}{J^{\mathrm{c}}}\frac{\partial}{\partial \xi^3}(u_1)^{\mathrm{f}} - \frac{(\boldsymbol{e}_1)^{\mathrm{c}}}{J^{\mathrm{c}}}\frac{\partial}{\partial \xi^3}(u_2)^{\mathrm{f}}. \tag{112c}$$

Swapping $^{\mathrm{c}}$ and $^{\mathrm{f}}$ superscripts yields analogous relations for inputs on centers.

The vertical derivatives satisfy a discrete form of integration by parts,

$$\mathscr{I}^{\mathrm{c}}_{\Omega_v}\left(\phi^{\mathrm{c}}, \frac{1}{J^{\mathrm{c}}}\frac{\partial}{\partial \xi^3}\psi^{\mathrm{f}}\right) + \mathscr{I}^{\mathrm{f}}_{\Omega_v}\left(\psi^{\mathrm{f}}, \frac{1}{J^{\mathrm{f}}}\frac{\partial}{\partial \xi^3}\phi^{\mathrm{c}}\right) = \hat{\mathcal{W}}\left(\phi^{\mathrm{c}}_{N_v}\psi^{\mathrm{f}}_{N_v+1} - \phi^{\mathrm{c}}_1\psi^{\mathrm{f}}_1\right), \tag{113a}$$

$$\mathscr{I}^{\mathrm{f}}_{\Omega_v}\left(\phi^{\mathrm{f}}, \frac{1}{J^{\mathrm{f}}}\frac{\partial}{\partial \xi^3}\psi^{\mathrm{c}}\right) + \mathscr{I}^{\mathrm{c}}_{\Omega_v}\left(\psi^{\mathrm{c}}, \frac{1}{J^{\mathrm{c}}}\frac{\partial}{\partial \xi^3}\phi^{\mathrm{f}}\right) = \hat{\mathcal{W}}\left(\phi^{\mathrm{f}}_{N_v+1}\psi^{\mathrm{c}}_{N_v} - \phi^{\mathrm{f}}_1\psi^{\mathrm{c}}_1\right). \tag{113b}$$

Where $\hat{\mathcal{W}} = \delta V/J$ is the product of GLL weights along the ξ^1 and ξ^2 directions. As with the horizontal discretization, this leads to discrete analogs of divergence and Stokes' theorems, which are crucial for conservation properties (Appendix E). Recursive application of the differential operators also yields higher-order analogs of these identities.

5.6. Velocity and Scalar Reconstructions

Evaluating differential operators such as flux divergences requires center values of vertical velocities, which are natively defined on faces, and face values of other variables that are natively defined on centers. Therefore, face values need to be reconstructed on centers and vice versa, in such a way that global conservation laws are satisfied. One possible way to satisfy these conservation laws, in some respects resembling Simmons and Burridge (1981), is with the following reconstructions:

1. *Density*. We reconstruct the density on faces from the density on centers with the Jacobian-weighted average

$$\rho^{\mathrm{f}} = \frac{1}{J^{\mathrm{f}}}\langle(\rho J)^{\mathrm{c}}\rangle^{\mathrm{f}}. \tag{114}$$

This quantity is needed to define the mass-weighted average $\lceil\cdot\rfloor$.

2. *Velocity*. The covariant components of velocity are reconstructed using unweighted averages,

$$(\boldsymbol{u}_h)^{\mathrm{f}} = \langle(\boldsymbol{u}_h)^{\mathrm{c}}\rangle^{\mathrm{f}}, \tag{115a}$$

$$(\boldsymbol{u}_v)^{\mathrm{c}} = \left\langle(\boldsymbol{u}_v)^{\mathrm{f}}\right\rangle^{\mathrm{c}}. \tag{115b}$$

With the contravariant subtensors of the symmetric metric tensor $\boldsymbol{g}$ given by

$$\boldsymbol{g}^{hh} = g^{11}\boldsymbol{e}_1(\boldsymbol{e}_1)^\intercal + g^{21}\boldsymbol{e}_1(\boldsymbol{e}_2)^\intercal + g^{21}\boldsymbol{e}_2(\boldsymbol{e}_1)^\intercal + g^{22}\boldsymbol{e}_2(\boldsymbol{e}_2)^\intercal, \tag{116a}$$

$$\boldsymbol{g}^{hv} = g^{31}\boldsymbol{e}_1(\boldsymbol{e}_3)^\intercal + g^{32}\boldsymbol{e}_2(\boldsymbol{e}_3)^\intercal, \tag{116b}$$

$$\boldsymbol{g}^{vh} = g^{31}\boldsymbol{e}_3(\boldsymbol{e}_1)^\intercal + g^{32}\boldsymbol{e}_3(\boldsymbol{e}_2)^\intercal, \tag{116c}$$

$$\boldsymbol{g}^{vv} = g^{33}\boldsymbol{e}_3(\boldsymbol{e}_3)^\intercal, \tag{116d}$$

the contravariant components are defined in terms of covariant components as

$$\left(\boldsymbol{u}^h\right)^{\mathrm{c}} = \left(\boldsymbol{g}^{hh}\boldsymbol{u}_h\right)^{\mathrm{c}} + \left(\boldsymbol{g}^{hv}\boldsymbol{u}_v\right)^{\mathrm{c}}, \tag{117a}$$

$$(\boldsymbol{u}^v)^{\mathrm{f}} = \lceil\left(\boldsymbol{g}^{vh}\boldsymbol{u}_h\right)^{\mathrm{c}}\rfloor^{\mathrm{f}} + (\boldsymbol{g}^{vv}\boldsymbol{u}_v)^{\mathrm{f}}, \tag{117b}$$

where a weighted average is used to reconstruct $\boldsymbol{u}^v$ on faces. We obtain $\boldsymbol{u}^h$ on faces with an unweighted average of its reconstruction on centers,

$$\left(\boldsymbol{u}^h\right)^{\mathrm{f}} = \left\langle\left(\boldsymbol{u}^h\right)^{\mathrm{c}}\right\rangle^{\mathrm{f}}, \tag{118}$$

and we do not have a unique reconstruction of $\boldsymbol{u}^v$ on centers.

3. *Velocity Boundary Conditions*. At the top and bottom faces, we enforce the boundary condition (Equation 48) of no normal flow, $(\boldsymbol{u}^v)^{\mathrm{f}}_1 = (\boldsymbol{u}^v)^{\mathrm{f}}_{N_v+1} = 0$, by setting u_3 at the boundaries to

$$(u_3)^{\mathrm{f}}_1 = -\frac{\left(g^{31}u_1 + g^{32}u_2\right)^{\mathrm{c}}_1}{(g^{33})^{\mathrm{f}}_1}, \tag{119a}$$

$$(u_3)^{\mathrm{f}}_{N_v+1} = -\frac{\left(g^{31}u_1 + g^{32}u_2\right)^{\mathrm{c}}_{N_v}}{(g^{33})^{\mathrm{f}}_{N_v+1}}. \tag{119b}$$

4. *Vorticity*. We compute the contravariant components of vorticity with a weak horizontal curl,

$$\left(\boldsymbol{\omega}^h\right)^{\mathrm{f}} = \tilde{\nabla}_h \times (\boldsymbol{u}_v)^{\mathrm{f}} + \nabla_v \times (\boldsymbol{u}_h)^{\mathrm{c}}, \tag{120a}$$

$$(\boldsymbol{\omega}^v)^{\mathrm{c}} = \tilde{\nabla}_h \times (\boldsymbol{u}_h)^{\mathrm{c}}. \tag{120b}$$

5. *Kinetic Energy*. The contribution of vertical velocity to $\boldsymbol{u}\cdot\boldsymbol{u}$ on centers is reconstructed using an unweighted average of the term involving $\boldsymbol{g}^{vv}$,

$$(\boldsymbol{u}^v \cdot \boldsymbol{u}_v)^{\mathrm{c}} = \left(\boldsymbol{g}^{vh}\boldsymbol{u}_h \cdot \boldsymbol{u}_v\right)^{\mathrm{c}} + \langle(\boldsymbol{g}^{vv}\boldsymbol{u}_v \cdot \boldsymbol{u}_v)^{\mathrm{f}}\rangle^{\mathrm{c}}. \tag{121}$$

The kinetic energy is then computed as

$$\kappa^{\mathrm{c}} = \frac{1}{2}(\boldsymbol{u}\cdot\boldsymbol{u})^{\mathrm{c}} = \frac{1}{2}\left[\left(\boldsymbol{u}^h \cdot \boldsymbol{u}_h\right)^{\mathrm{c}} + (\boldsymbol{u}^v \cdot \boldsymbol{u}_v)^{\mathrm{c}}\right]. \tag{122}$$

6. *Scalar Advection*. We reconstruct scalars ψ, including specific humidities q_μ, the specific total enthalpy h_{tot}, and tracers χ, on faces through

$$\psi^{\mathrm{f}} = \mathcal{U}\left[\psi^{\mathrm{c}}, (\boldsymbol{u}^v)^{\mathrm{f}}\right], \tag{123}$$

where $\mathcal{U}$ denotes an upwind-biased reconstruction (Leer, 1977; Lin et al., 1994) or a flux-corrected transport reconstruction (Zalesak, 1979). This enables discrete monotonicity preservation of ψ for strong stability-preserving timestepping schemes, and approximate monotonicity preservation for other types of time-stepping. Our default choice is to use the upwind-biased reconstruction of Lin et al. (1994), with a limiter based on local extrema of ψ^{c}. The flux divergence of $\rho\psi$ is discretized with a weak horizontal divergence,

$$\left(\partial_t \rho\psi|_{\mathrm{adv}}\right)^{\mathrm{c}} = -\tilde{\nabla}_h \cdot \left(\rho\psi\boldsymbol{u}^h\right)^{\mathrm{c}} - \nabla_v \cdot (\rho\psi\boldsymbol{u}^v)^{\mathrm{f}}. \tag{124}$$

Setting $\psi = 1$ yields the corresponding mass flux divergence, ensuring tracer-mass consistency.

7. *Momentum Advection*. Momentum advection is discretized in vector invariant form as

$$\left(\partial_t \boldsymbol{u}_h|_{\mathrm{adv}}\right)^{\mathrm{c}} = -\nabla_h \kappa^{\mathrm{c}} - \left(\boldsymbol{\omega}^v \times \boldsymbol{u}^h\right)^{\mathrm{c}} - \left\lceil\left(\boldsymbol{\omega}^h \times \boldsymbol{u}^v\right)^{\mathrm{f}}\right\rfloor^{\mathrm{c}}, \tag{125a}$$

$$\left(\partial_t \boldsymbol{u}_v|_{\mathrm{adv}}\right)^{\mathrm{f}} = -\nabla_v \kappa^{\mathrm{c}} - \left(\boldsymbol{\omega}^h \times \boldsymbol{u}^h\right)^{\mathrm{f}}, \tag{125b}$$

with a strong horizontal gradient of kinetic energy and a weighted average of the vorticity term involving $\boldsymbol{u}^v$.

With these reconstructions, momentum advection conserves kinetic energy and vorticity globally, and flux divergences conserve scalars globally (Appendix E). Because total energy is also discretely conserved, this ensures that any numerical conversion between kinetic and non-kinetic energy is performed solely by the discretized pressure gradient term and other physical sources/sinks, without spurious sources or sinks of energy from the momentum advection scheme itself. This property is crucial for maintaining a consistent energy pathway between the resolved kinetic scales and the thermodynamic reservoirs, even in the presence of moist processes.

5.7. Timestepping

We employ a HEVI timestepping strategy that steps horizontal tendencies explicitly and fast vertical tendencies implicitly. That is, tendencies are split into explicit and implicit parts, $\boldsymbol{T}^E$ and $\boldsymbol{T}^I$, so that the equations of motion (Equation 78) can be written as

$$\partial_t \boldsymbol{Y} = \boldsymbol{T}^E(\boldsymbol{Y}, t) + \boldsymbol{T}^I(\boldsymbol{Y}, t), \tag{126}$$

where $\boldsymbol{Y}$ is a block vector representation of the prognostic state (Appendix F). The implicit tendency, consisting of the vertical components of derivative and relaxation terms identified by "VI" in Equation (78), is responsible for sound and gravity waves, falling/sedimenting condensate, diffusion, and damping along the vertical direction.

To step the state $\boldsymbol{Y}$ forward in time, we use an additive Runge-Kutta method,

$$\boldsymbol{Y}|_{t=t_0+\delta t} = \mathcal{P}\left[\boldsymbol{Y}|_{t=t_0} + \delta t \sum_{s=1}^{N_s} b_s^E \boldsymbol{T}^E(\boldsymbol{U}_s, t_0 + \delta t) + \delta t \sum_{s=1}^{N_s} b_s^I \boldsymbol{T}^I(\boldsymbol{U}_s, t_0 + \delta t)\right], \tag{127}$$

defined in terms of N_s stages

$$\boldsymbol{U}_i = \mathcal{P}\left[\hat{\boldsymbol{U}}_i + a^I_{ii}\delta t\, \boldsymbol{T}^I\left(\boldsymbol{U}_i, t_0 + c^I_i \delta t\right)\right], \tag{128}$$

where the partially assembled stage values $\hat{\boldsymbol{U}}_i$ are given by

$$\hat{\boldsymbol{U}}_i = \mathcal{P}\left[\boldsymbol{Y}|_{t=t_0} + \delta t \sum_{j=1}^{i-1} a^E_{ij} \boldsymbol{T}^E\left(\boldsymbol{U}_j, t_0 + c^E_j \delta t\right) + \delta t \sum_{j=1}^{i-1} a^I_{ij} \boldsymbol{T}^I\left(\boldsymbol{U}_j, t_0 + c^I_j \delta t\right)\right]. \tag{129}$$

The DSS operator $\mathcal{P}$ is applied at the end of each stage and at the end of the timestep, avoiding the accumulation of round-off errors that are discontinuous across element boundaries. Our default explicit/implicit Butcher tableau coefficients $\boldsymbol{a}^{E/I}$, $\boldsymbol{b}^{E/I}$, and $\boldsymbol{c}^{E/I}$ are from the 3rd-order ARS343 scheme of Ascher et al. (1997), as this proved to be robust and provide the shortest time to solution in the benchmark tests of Gardner et al. (2018). This scheme has 4 stages, 3 of which include explicit terms.

To find a value of $\boldsymbol{U}_i$ that approximately satisfies Equation 128 on stages with $a^I_{ii} \neq 0$, we use Newton's method to iteratively solve the implicit equation

$$\boldsymbol{R}_i(\boldsymbol{U}_i) = \boldsymbol{U}_i - \hat{\boldsymbol{U}}_i - a^I_{ii}\delta t\, \boldsymbol{T}^I\left(\boldsymbol{U}_i, t_0 + c^I_i \delta t\right) = 0. \tag{130}$$

Since $\boldsymbol{T}^I$ does not involve horizontal derivatives, we can solve this equation independently in each column, without applying DSS and incurring horizontal communication costs. Starting with the initial guess $\boldsymbol{U}^{(0)}_i = \hat{\boldsymbol{U}}_i$, Newton's method performs N_n updates

$$\boldsymbol{U}^{(n)}_i = \boldsymbol{U}^{(n-1)}_i - \left[\boldsymbol{W}_i\left(\boldsymbol{U}^{(n-1)}_i\right)\right]^{-1} \boldsymbol{R}_i\left(\boldsymbol{U}^{(n-1)}_i\right), \tag{131}$$

where $\boldsymbol{W}_i$ is the approximate Jacobian of the residual,

$$\boldsymbol{W}_i(\boldsymbol{U}_i) \approx \frac{\partial}{\partial \boldsymbol{U}_i} \boldsymbol{R}_i(\boldsymbol{U}_i). \tag{132}$$

The Jacobian approximation focuses only on the most rapidly varying terms, and its sparsity structure allows us to compute $(\boldsymbol{W}_i)^{-1}\boldsymbol{R}_i$ through an efficient linear solver (Appendix G). After N_n iterations of Newton's method, we apply DSS to get the approximation

$$\boldsymbol{U}_i \approx \mathcal{P}\left(\boldsymbol{U}^{(N_n)}_i\right). \tag{133}$$

To ensure that Equation 128 is satisfied exactly, we approximate the implicit tendency of $\boldsymbol{U}_i$ as

$$\mathcal{P}\left[\boldsymbol{T}^I\left(\boldsymbol{U}_i, t_0 + c^I_i \delta t\right)\right] \approx \frac{\boldsymbol{U}_i - \hat{\boldsymbol{U}}_i}{a^I_{ii}\delta t}. \tag{134}$$

A single Newton iteration is sufficient for ARS343 when only terms responsible for sound and gravity waves are included in the implicit tendency (Gardner et al., 2018); 3 iterations are typically required when diffusion, damping, and/or sedimentation are treated implicitly.

The HEVI timestepping scheme avoids limitations on δt from vertically propagating sound and gravity waves, as well as limitations from damping terms that have short timescales, such as those in the sponge layer. This is computationally advantageous because, at typical resolutions for atmosphere models, the vertical resolution (tens of meters near the surface to kilometers near the model top) is much finer than the horizontal resolution (tens to hundreds of kilometers). The principal remaining timestep limitation is from sound waves propagating

horizontally with speed $c_s \approx 340$ m s^{-1}, resulting in maximum timesteps of order $\delta t \sim (\delta x)_{\min}/c_s$, where $(\delta x)_{\min}$ is the minimum horizontal distance between nodal points.

The Julia programming language facilitates performance portability across diverse GPU and CPU computing platforms. The code allows flexible but consistent use of different floating-point precisions, such as 32- or 64-bit floating-point arithmetic, throughout the model.

6. Benchmark Tests

We assess the accuracy and conservation properties of the dynamical core in a series of benchmark tests. These tests range from idealized baroclinic wave simulations to simulations of gravity waves generated by topography, providing an evaluation of the core's ability to capture a range of atmospheric phenomena. All simulations are performed at 32-bit floating point precision unless otherwise stated.

6.1. Baroclinic Waves

We first show results from a series of baroclinic wave tests of increasing complexity, beginning with a dry baroclinic wave, then a moist version, and culminating in a moist baroclinic wave with topography. These tests are designed to assess the dynamical core's ability to simulate the development of baroclinic instability, a crucial driver of mid-latitude weather systems.

For the baroclinic wave simulations, we use polynomial order $N_p = 3$ and $N_e = 120$ spectral elements across each edge of the cubed sphere, resulting in $N_h = 6N_e^2$ horizontal spectral elements and an average grid spacing of

$$\delta x \approx \left[4\pi a^2/\left(6N_e^2 N_p^2\right)\right]^{1/2}. \tag{135}$$

For $N_e = 120$ and with Earth's mean radius $a = 6371$ km, this implies an average grid spacing of $\delta x \approx 26$ km. We use $N_v = 43$ vertical levels that are unequally spaced using SLEVE stretching, for a resolution of 30 m near the surface that increases to around 2 km near the model top at $z_t = 30$ km (Appendix B). We apply horizontal hyperdiffusion with $\nu_{\boldsymbol{u}} = 3.1 \times 10^{12}$ m^4 s^{-1} for velocity and $\nu_\psi = 5\nu_{\boldsymbol{u}}$ for scalars, with damping of the divergent flow component enhanced by a factor $\delta_{\text{div}} = 5$. The timestep for all cases is $\delta t = 40$ s. This corresponds to an average horizontal Courant number for acoustic waves of about 40 s $\times$ 330 m s^{-1}/26 km ≈ 0.5, which is comparable to the stable Courant numbers reported for the ARS343 scheme in the dry simulations of Gardner et al. (2018), with the slightly smaller value in our case being consistent with the additional complexities of moist dynamics, phase transitions, and microphysics. Only the moist baroclinic wave with topography uses a sponge layer to absorb gravity waves at the model top; the other simulations do not require one.

6.1.1. Dry Baroclinic Wave

This test case uses a dry, deep atmosphere with a baroclinically unstable zonal jet (Ullrich et al., 2014). The initial conditions are based on an analytical steady-state solution to the dry fluid flow equations on a rotating sphere without topography. The initial state is zonally symmetric. A small-amplitude vorticity perturbation is inserted in the Northern Hemisphere, which triggers the development of a baroclinic wave. The model is then integrated forward in time for 10 days, and the wave's evolution is analyzed, with a focus on the development of a frontal zone. Any short-term wave development observed in the Southern Hemisphere can be attributed to discretization errors.

Figure 3 shows the surface pressure perturbation from the initial condition, the 850 hPa temperature, and the 850 hPa relative vorticity on days 8 and 10 of the simulation, focusing on the Northern Hemisphere. The qualitative features of the evolving baroclinic instability, including the pressure wave pattern and the onset of overturning temperature wavefronts on day 8, are similar to those obtained with other dynamical cores (Ullrich et al., 2014); however, the higher resolution of our simulations reveals more details, for example, about the frontal structures than were visible in those earlier, lower-resolution simulations. Nonlinear effects are visible through the appearance of breaking waves at day 10 in the air temperature and vorticity fields, and the formation of a distinct, detached low pressure zone in the surface pressure field. The minimum surface pressure at day 10 is

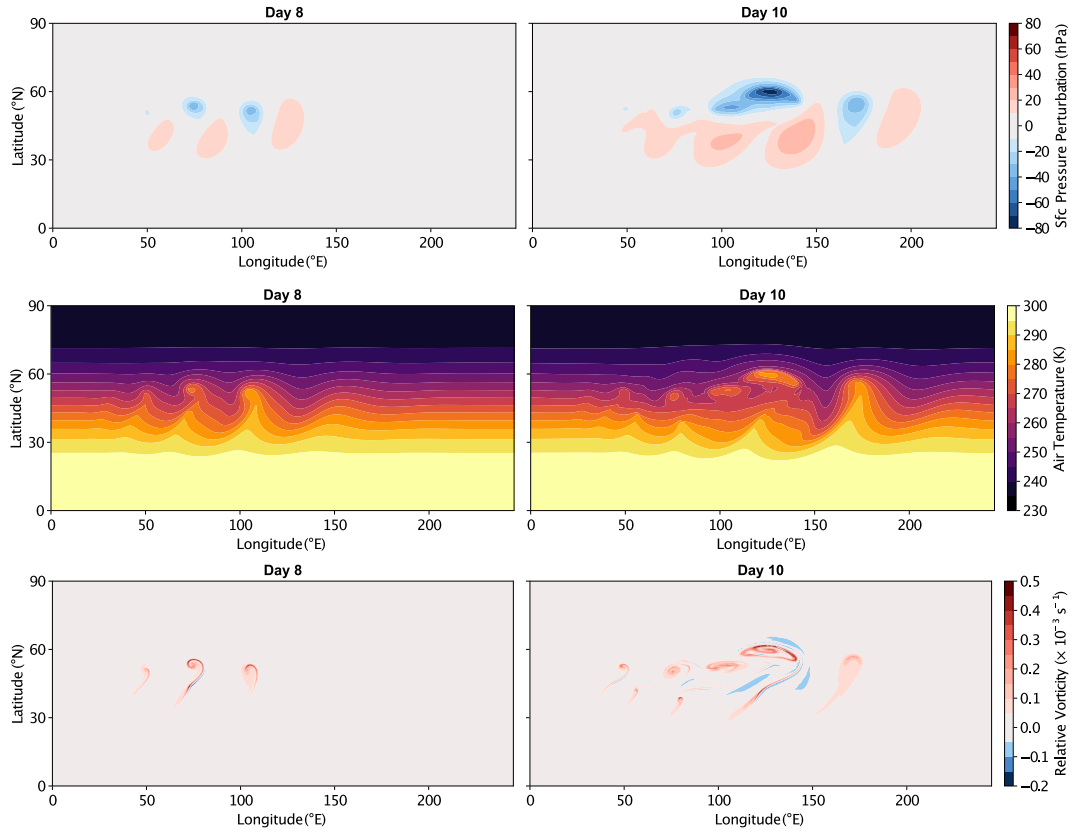

Figure 3. Dry baroclinic wave benchmark simulation at days 8 and 10: instantaneous surface pressure perturbation from initial condition (top row), air temperature at 850 hPa (middle row), and relative vorticity at 850 hPa (bottom row). Only a longitude sector in the Northern Hemisphere is shown; no perturbations would be visible in the Southern Hemisphere and in the longitude sector not shown here.

925 hPa in this simulation, which is consistent with the range of values around 930 hPa obtained with several other dynamical cores, including deep-atmosphere simulations, in Ullrich et al. (2014).

Figure 4 shows the pressure perturbation from its initial state along the 50°N latitude circle, capturing the growth of the baroclinic wave. The characteristic westward tilt with height of the growing baroclinic wave is clearly visible.

6.1.2. Moist Baroclinic Wave

This test case builds upon the dry baroclinic wave, adding moisture and simplified physics to examine the impact of moisture feedbacks on wave development (Ullrich et al., 2016). The initial conditions for this test case are similar to the dry case, but with the addition of a specific humidity field, which leads to a relative humidity maximum of around 85% at lower levels in midlatitudes. The moist test case uses a 1-moment microphysics parameterization to model precipitation processes, including prognostic equations for rain and snow, which resembles the Kessler parameterization commonly used in this benchmark test (Hughes & Jablonowski, 2023; Kessler, 1969); however, it differs from it in detail, such as the choice of parameters (Azimi et al., 2024) and the inclusion of ice-phase microphysics (Grabowski, 1998; Kaul et al., 2015). This test case assesses the model's ability to capture the interactions between subgrid-scale physical parameterizations and the dynamical evolution of baroclinic waves.

Figure 5 shows the instantaneous surface pressure perturbation from the initial condition and the 850-hPa temperature, relative vorticity, and specific humidity on days 8 and 10 of the simulation. The qualitative evolution of the surface pressure, air temperature, and relative vorticity fields resembles that of the dry test case. Sharp horizontal gradients and fine-scale features of moisture transport in the frontal zones, apparent in the specific

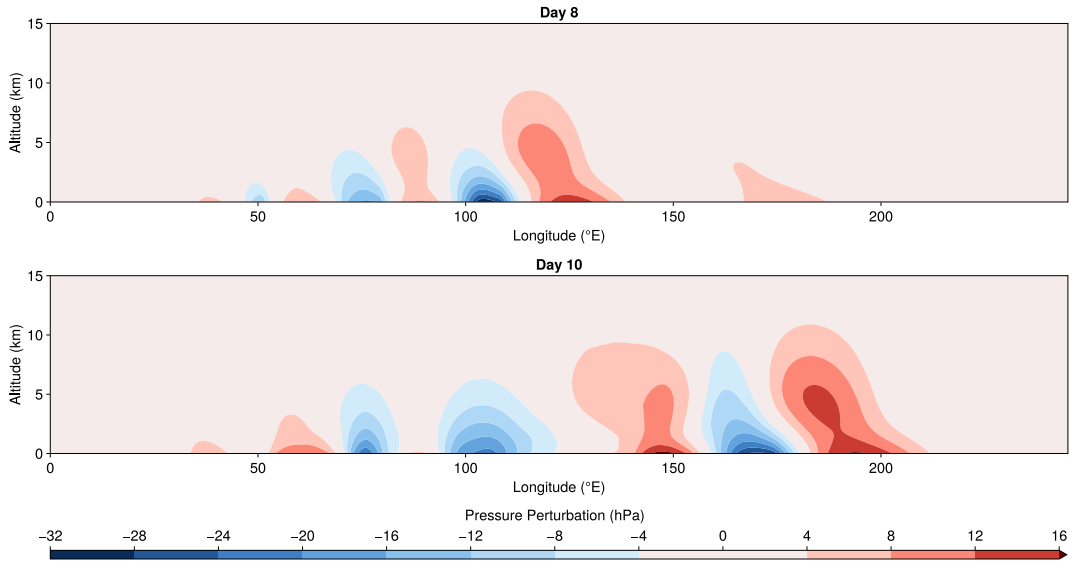

Figure 4. Instantaneous pressure perturbation from initial state in dry baroclinic wave benchmark simulation at days 8 and 10, shown along the 50°N latitude circle.

humidity field (represented on base-two logarithmic scale) at day 10, are retained at the horizontal resolution we use in this simulation, suggesting that the choice of horizontal hyperdiffusion parameters does not result in excessive diffusion. The model's robustness is further demonstrated by simulations at coarser resolutions (e.g., 103 km average horizontal grid spacing, not shown), which correctly capture the fundamental dynamics and large-scale evolution of the baroclinic wave, with finer-scale frontal structures being, as expected, less sharply defined.

6.1.3. Moist Baroclinic Wave With Topography

This test case further enhances the complexity by adding idealized topographic barriers to the moist baroclinic wave setup (Hughes & Jablonowski, 2023). It combines a baroclinically unstable base state with two analytically prescribed mountain ridges in midlatitudes, which trigger both baroclinic Rossby waves and inertia-gravity waves. This test case assesses the impact of mountains on baroclinic waves, examining orographically induced flow features, precipitation patterns, and the coupling between physics and dynamics. To stabilize these simulations with strong orographic gravity waves that propagate upward, we apply a Rayleigh sponge (Equation 66) that damps vertical velocities in the top 5 km of the domain. The resolution, simulation timestep, and precipitation timescale are otherwise identical to that of the moist baroclinic wave benchmark in Section 6.1.2. We use the SLEVE terrain-following coordinate described in Appendix B.

Figure 6 shows the evolution of the surface pressure perturbation from its initial state, the air temperature at 850 hPa, and the precipitation rate for the orographically triggered baroclinic wave test case at days 4 and 6. The surface pressure evolution compares favorably with Figure 4 in Hughes and Jablonowski (2023), with two identical pressure systems downstream of the mountains on day 4, followed by two distinct low-pressure systems triggered by the topographic obstacle. In a pattern similar to the pressure perturbation evolution, two nearly identical temperature and precipitation fronts develop at day 4 downstream of each mountain, followed by nonlinear wave breaking at day 6.

Figure 7 shows the evolution of the temperature front upstream of the topographic feature located at 140°E, inertia-gravity waves due to the terrain obstacle, and a distinct updraft upstream of the mountain. We recover key qualitative features of this test case noted by Hughes and Jablonowski (2023), such as the presence of the maximal temperature perturbation at altitudes of several kilometers on day 4 and strong updrafts at 125°E.

Figure 8 shows the instantaneous cloud condensate specific humidity along the 45°N latitude circle between 100°E and 240°E. Comparing with Figure 12 in Hughes and Jablonowski (2023), we note that our simulations

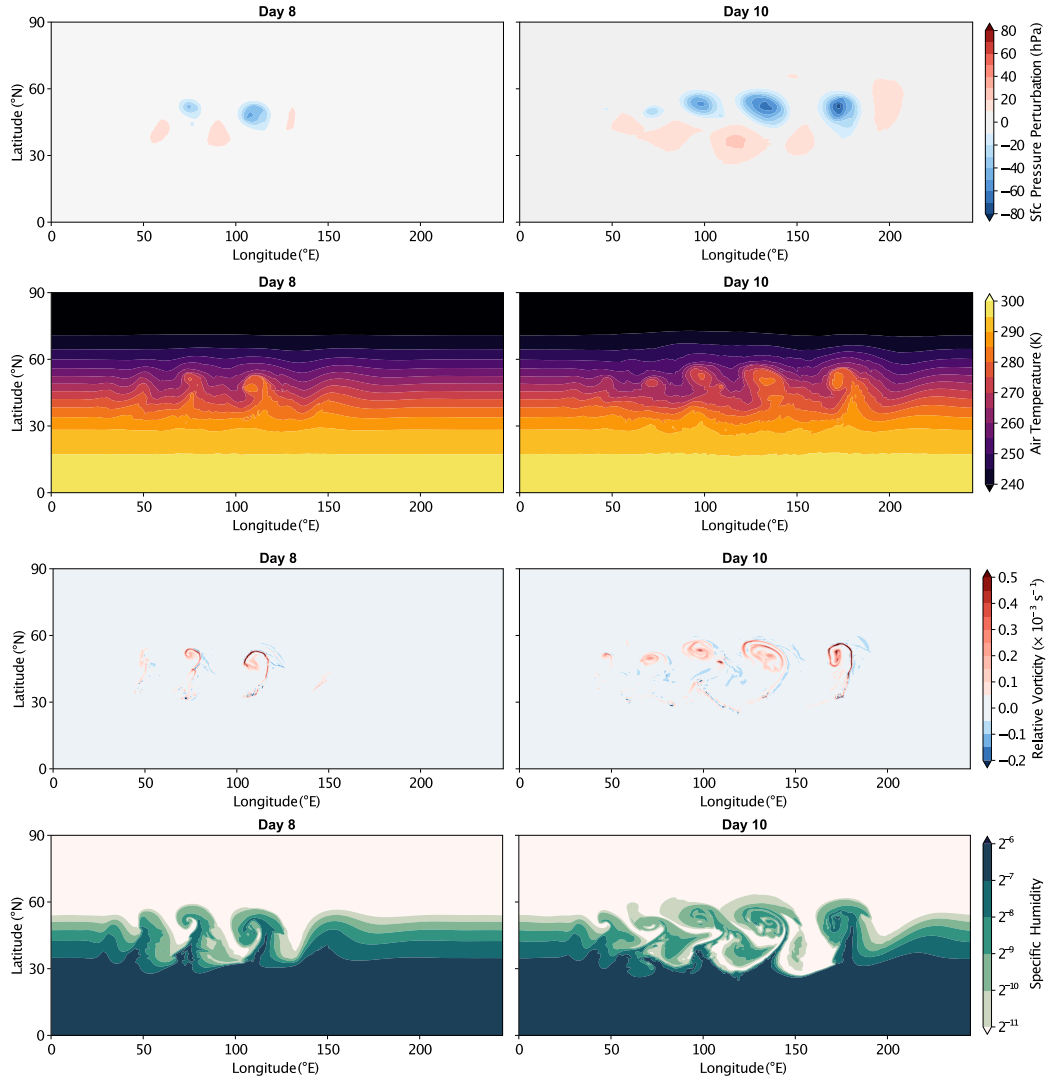

Figure 5. Moist baroclinic wave benchmark simulation at days 8 and 10: instantaneous surface pressure perturbation from initial state (first row), air temperature at 850 hPa (second row), relative vorticity at 850 hPa (third row), and specific humidity at 850 hPa (fourth row).

recover the most prominent cloud water features at approximately 140°E, where clouds are forced up to altitudes of 8 km due to the mountain, and at 210°E. Weaker cloud clusters are present at 160°E, in contrast to the results in Hughes and Jablonowski (2023), who see additional cloud water up to 6 km altitude at around 160°E. These differences are likely due to our different representation of microphysics.

To verify the conservation properties of the dynamical core, Figure 9 shows the evolution of globally integrated dry air mass, total water, and total energy over a 1-year simulation of the moist baroclinic wave with topography, at 103 km resolution ($N_e = 30$). We compare results from simulations using 32-bit and 64-bit floating-point precision to confirm that any deviations from exact conservation are attributable solely to numerical roundoff error.

As expected, dry air mass is conserved to the level of floating-point precision, with relative deviations of $O(10^{-4})$ in 32-bit precision reducing to $O(10^{-13})$ in 64-bit precision. The atmosphere loses water mass through precipitation over the course of the simulation; however, the sum of the water mass in the atmosphere and the cumulative mass of water precipitated to the surface remains constant, with deviations again controlled by floating-point precision ($O(10^{-4})$ for 32-bit and $O(10^{-13})$ for 64-bit). Similarly, total energy—the sum of the energy

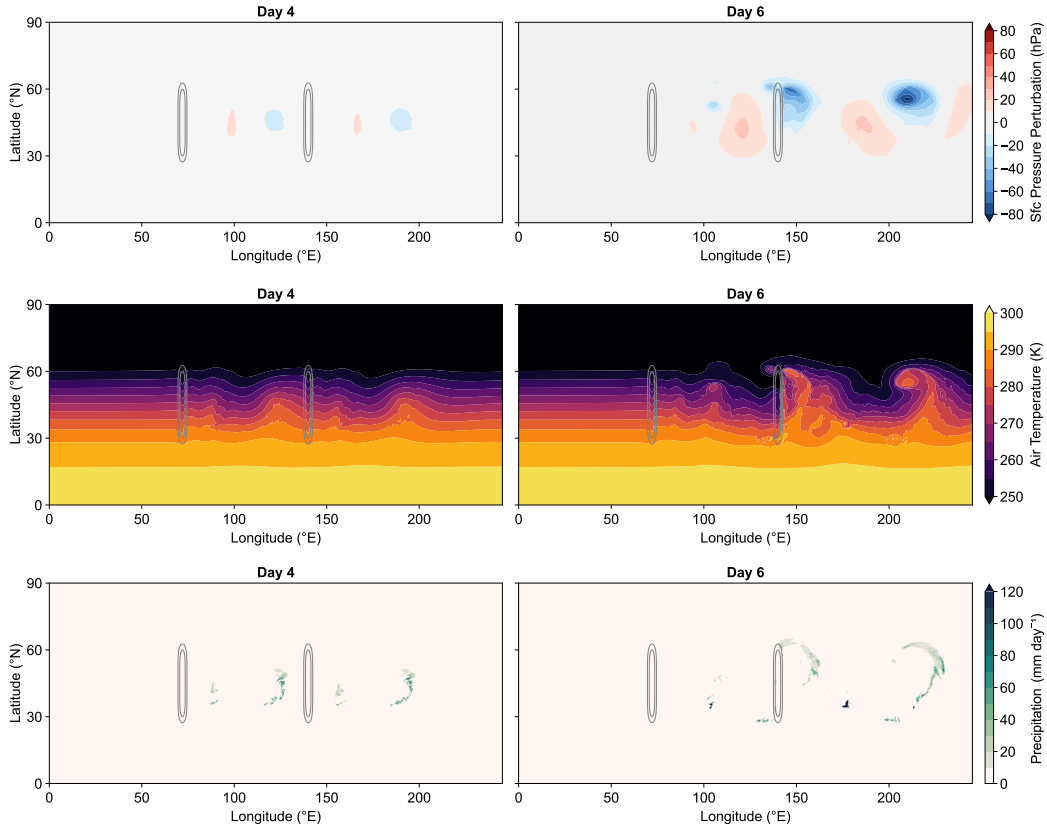

Figure 6. Instantaneous surface pressure perturbation from initial state (first row), 850 hPa air temperature (second row), and precipitation rate (third row) at days 4 (left panels) and 6 (right panels) for the orographically triggered baroclinic wave test case. The mountains are shown with solid contour lines.

within the atmosphere and the cumulative energy removed by precipitation—is conserved to within $O(10^{-1})$ at 32-bit and $O(10^{-8})$ at 64-bit precision, when normalized by the maximum kinetic energy. These normalized deviations show that the conservation error is small relative to the potential–kinetic energy conversion that occur; however, they appear relatively large only because the kinetic energy is five orders of magnitude smaller than the total energy, which itself depends on the arbitrary reference temperature T_0. These results demonstrate that even in a complex benchmark with moist processes and topography, the model conserves dry air mass, total water, and total energy without the need for ad hoc fixers.

6.2. Orographic Gravity Wave

To test the ability of the dynamical core to accurately simulate flow over topography, we consider the mountain wave benchmark test of Schär et al. (2002). This test simulates uniform flow in a horizontally periodic Cartesian domain that generates gravity waves when it impinges on a topographic obstacle.

For the numerical simulation, we discretize a two-dimensional domain with $x_{\max} = 300$ km and $z_t = 21$ km using the grid parameters $N_p = 3$ and $N_h = N_v = 100$. The linearly attenuated vertical coordinate is stretched so that element heights range from $\delta z|_s = 10$ m near the surface to $\delta z|_t = 1$ km at the model top. A Rayleigh sponge above $z_d = 13$ km damps vertical velocities with a maximum relaxation coefficient of $\alpha_{\max} = 0.1$ s^{-1}. For the initial condition, we use a velocity of $u = 10$ m s^{-1} along the x-direction and a hydrostatically balanced pressure profile. The buoyancy frequency has a uniform value of $N = 0.01$ s^{-1} from the surface (at $T_s = 288$ K) up to $T = 100$ K, and an isothermal profile extends from $T = 100$ K to the model top. With this configuration, the largest stable timestep for ARS343 is $\delta t = 0.7$ s.

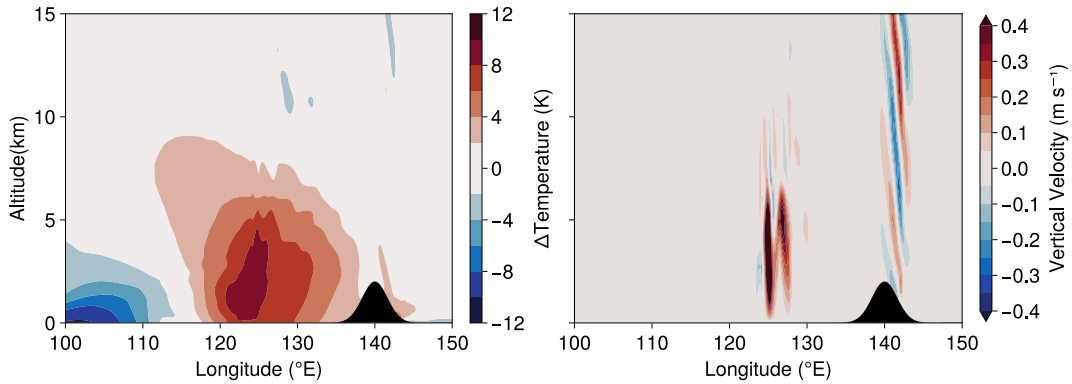

Figure 7. Slices along the 45°N latitude circle, between longitudes 100°E and 150°E, showing the temperature perturbation from the initial state (left panel) and the vertical velocity (right panel) on day 4 of the orographically triggered baroclinic wave test case. The mountain is depicted in black.

The methods discussed by J. B. Klemp et al. (2003) and Baldauf (2008), among others, can be used to obtain an approximate analytical solution in terms of Fourier transforms. The quasi-steady state established in the numerical simulation after a few days can be compared against the steady, first-order response to topography predicted analytically. For a Schär mountain of height $h = 25$ m, the simulated horizontal and vertical velocity fields closely match their predicted values, with deviations on the order of 0.001 m s^{-1} in the vertical velocity and 0.01 m s^{-1} in the horizontal velocity when using 64-bit floating-point precision (Figure 10).

Deviations between simulated and predicted velocity components can be attributed to numerical errors from the discretization and floating-point operations, as well as the first-order approximations used to compute the analytical solution. As the mountain height approaches $h = 0$ m, we observe linear convergence of the deviations when $h < 200$ m (Figure 11). The deviations do not converge for larger mountain heights because of missing higher-order terms in the analytical solution, and they do not converge for small mountain heights due to inadequate resolution and floating-point precision. Since $\boldsymbol{u}_h$ has a larger magnitude than $\boldsymbol{u}_v$, but its perturbations from gravity waves are similar to those of $\boldsymbol{u}_v$, its convergence is more strongly limited by floating-point precision.

7. Computational Performance

To assess the dynamical core's computational performance, we examine its weak and strong scaling on CPU and GPU architectures. Using the moist baroclinic wave test case from Section 6.1.2 as a benchmark, we vary the horizontal resolution by changing the number of horizontal spectral elements N_e along each cubed sphere edge, while keeping the vertical resolution fixed at $N_v = 43$ levels.

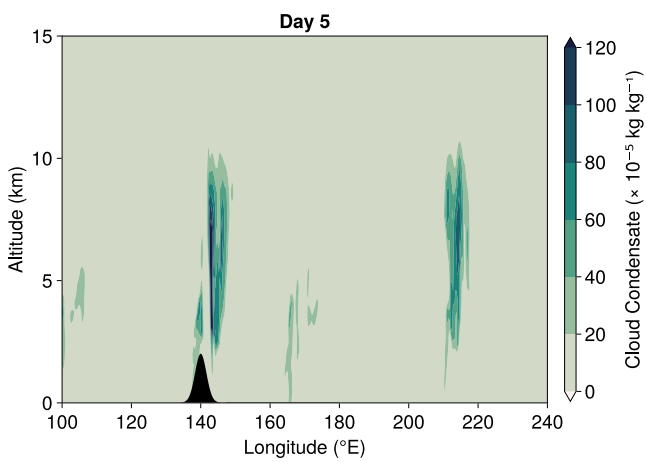

Figure 8. Slice along the 45°N latitude circle, between longitudes 100°E and 240°E, showing instantaneous cloud condensate specific humidity at day 5 of the orographically triggered baroclinic wave test case. The mountain is depicted in black.

The GPU scaling studies are conducted on the Derecho supercomputer at the NSF National Center for Atmospheric Research (NCAR), which consists of 82 GPU nodes, each with four NVIDIA A100 GPUs and 40 GB of memory per GPU (Computational and Information Systems Laboratory, 2023), and on the Google Cloud Platform (GCP) using NVIDIA H100 GPUs. We use between 1 and 512 GPUs in this scaling study.

The CPU scaling studies are conducted on Intel Icelake CPU nodes on Caltech's Resnick High-Performance Computing Cluster. Each node consists of two 32-core Intel Icelake 8352Y processors with 1.5 TB of memory per node. For higher-resolution simulations, with horizontal resolutions of 13 km or finer, memory constraints dictated use of at most 16 message passing interface (MPI) ranks per node. To maintain consistency in the CPU scaling studies, we use 16 MPI ranks per node—from 16 to 512 MPI ranks total—for all simulations, although this results in under-utilization of the available computing resources at lower resolutions.

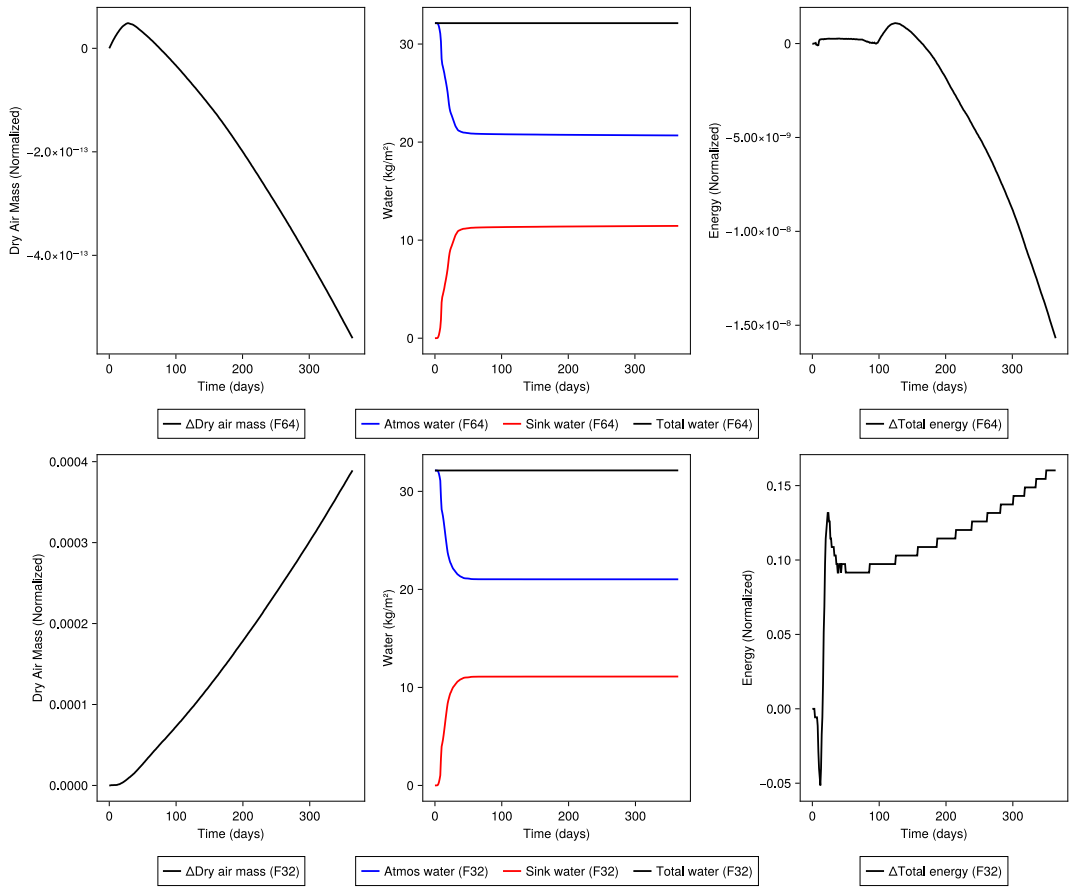

Figure 9. Conservation of dry air mass (left), total water mass (middle), and total energy (right) over a 1-year simulation of the moist baroclinic wave benchmark with topography and 1-moment microphysics at 103 km resolution. Results are shown for both 64-bit (F64, top row) and 32-bit (F32, bottom row) floating-point precision. Dry air mass is shown as the relative deviation from its initial value (5.23×10^{18} kg). For total water, the sum of atmospheric mass and cumulative precipitation is shown, expressed as an average mass per unit area of Earth's surface (initial total: 1.639×10^{16} kg). Total energy variations are shown as deviations from the initial value (2.479×10^{25} J), normalized by the maximum kinetic energy (4.030×10^{20} J) reached during the simulation. The discrete steps in the 32-bit total energy plot are artifacts of finite-precision arithmetic, not physical model processes.

7.1. Weak Scaling

Weak scaling refers to the model's ability to maintain constant time per model timestep as the resolution is refined and the processor count is increased in proportion to the problem size. To examine weak scaling, we refine the horizontal resolution from $\delta x \approx 103$ km ($N_e = 30$) to $\delta x \approx 6$ km ($N_e = 480$) by increasing N_e by factors of $\approx\sqrt{2}$ ($N_e = 30, 42, 60, 84, 120, 170, 240, 340, 480$) while doubling the processor count at each resolution increment. An integration time of one day (14,400 steps of 6 s) is used to estimate the time per model timestep for GPU simulations, and a minimum of 1,024 timesteps is used to estimate the time per model timestep for the CPU weak scaling runs. Perfect weak scaling would imply that the time per model timestep remains constant as the resolution is refined and the processor count increases like $(\delta x)^{-2}$.

Figure 12 shows the scaling results. We observe excellent weak scaling efficiencies, above 92% on the GPUs and above 98% on the CPUs, for the range of resolutions and process counts explored in our study. On GPUs, the time per timestep stays near its 1-GPU value of 223 ms at all resolutions, demonstrating the model's capability to efficiently handle increasing problem sizes with proportionally larger computational resources.

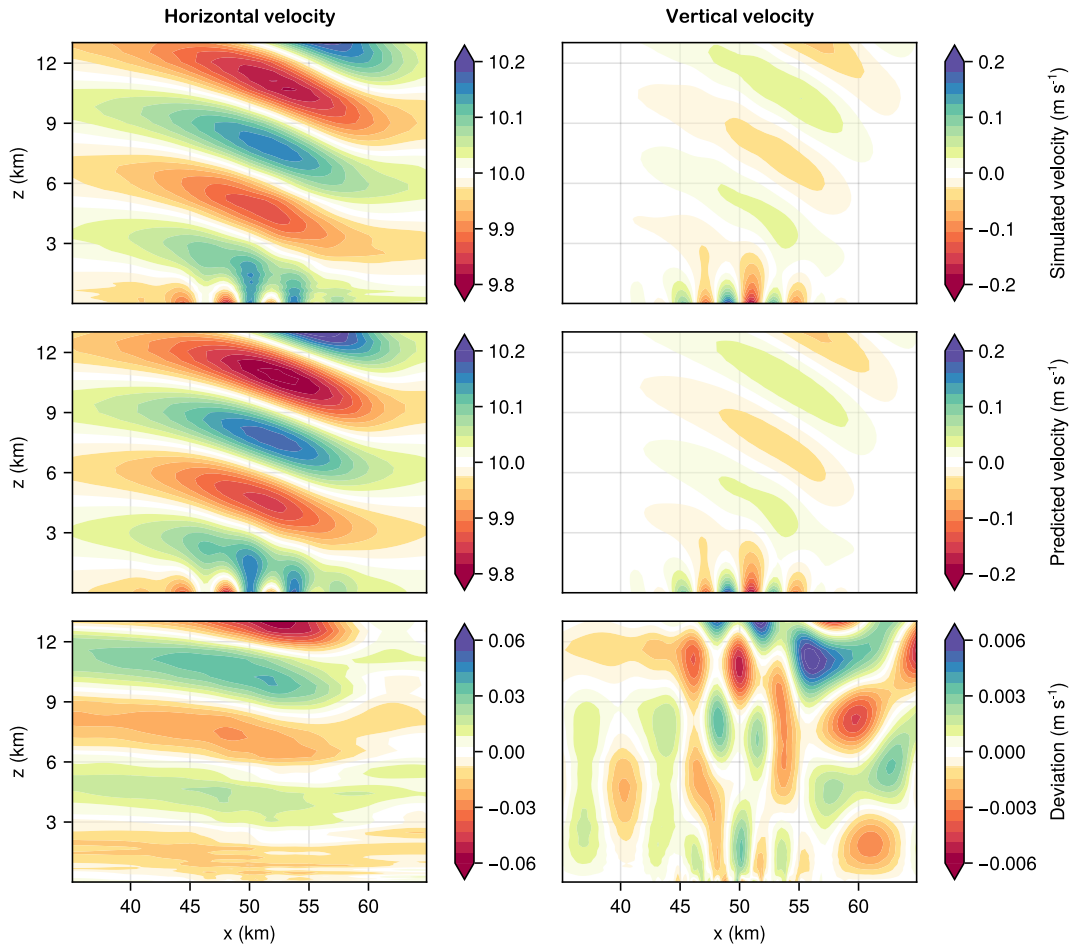

Figure 10. Horizontal (left) and vertical (right) velocity components in flow over a 25 m Schär mountain with 64-bit floating-point precision. (first row) Numerical simulation after 2 days of periodic flow. (second row) First-order, steady-state analytical solution. (third row) Difference between numerical simulation and analytical solution. All panels share the same spatial axes.

7.2. Strong Scaling

Strong scaling refers to the model's ability to reduce the time-to-solution as the number of processors is increased for a fixed problem size. To examine strong scaling, we increase the number of processors for several fixed resolutions δx from 6 km ($N_e = 480$) to 103 km ($N_e = 30$) and determine the time-to-solution in SYPD. The timestep δt is reduced roughly (but not exactly) like N_e^{-1} to satisfy Courant-Friedrichs-Lewy (CFL) bounds; we use $\delta t = 200\text{ s}, 100\text{ s}, 40\text{ s}, 15\text{ s}$, and 6 s for $N_e = 30, 60, 120, 240$, and 480. (The CFL bounds depend on the minimum distance between nodal points of the SEM method, which scales with N_e^{-1} for an equiangular cubed sphere, like the average grid spacing (Equation 135). However, stability of the simulations required somewhat shorter timesteps than implied by this scaling at high resolution.) The hyperdiffusivity is likewise reduced as the resolution increases, scaling like N_e^{-3}, as in P. H. Lauritzen et al. (2018). Each simulation is run for 1 simulated day on the GPUs. Since the time-to-solution is longer on CPUs, the integration times for estimating SYPD range from 1 day to a few minutes (≈ 16 steps) for the highest-resolution simulations running on single nodes.

Figure 13 shows the strong scaling results. For GPUs, there is an efficient strong scaling regime whose width increases as the resolution is refined, with scaling efficiencies above 95% achieved when there are $\gtrsim$5400 spectral elements per GPU. Below this threshold, the arithmetic intensity per GPU becomes small enough that communication costs begin to dominate. The figure compares scaling behavior on NVIDIA A100 GPUs on NCAR's Derecho supercomputer and on H100 GPUs on GCP. The differences in raw hardware capabilities are

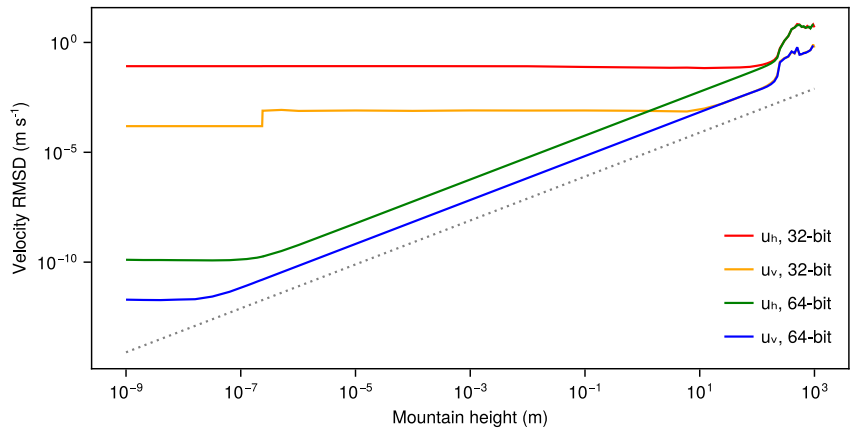

Figure 11. Average (root mean square) deviations between simulated and predicted velocity components in flow over the Schär mountain with varying mountain heights. Each average is computed over all points below the sponge layer.

apparent in the higher SYPD of the H100s at low GPU counts. However, the Derecho system's interconnect and configuration offer greater scaling efficiency, with SYPD on multiple A100s matching or outperforming the GCP H100s at sufficiently high GPU counts for each horizontal resolution down to 13 km. (Precise tuning and configuration of virtual machines on GCP may yield higher scaling efficiencies but is beyond the scope of the current scaling evaluation.)

The results show that simulations at moderately high resolutions (25–50 km) for the standards of climate models are feasible at >1 SYPD throughput on a dozen or a few dozen GPUs. Such compute resources, routinely available from cloud providers such as GCP, open up access to high-resolution atmosphere modeling to a broader range of users than those with access to supercomputing facilities. Extrapolating the scaling results also shows that simulations with around 10 km resolution at >1 SYPD are feasible on high-performance computing facilities with hundreds to thousands of GPUs. Although our primary focus is on GPU platforms, we also achieve good strong scaling efficiencies on CPUs up to 16 nodes (with 16 ranks per node), with efficiencies of about 80% for the highest-resolution runs.

7.3. Absolute Performance and Intercomparison

To place our results in the context of the field, we compare the absolute performance of the CliMA dynamical core with other state-of-the-art models.

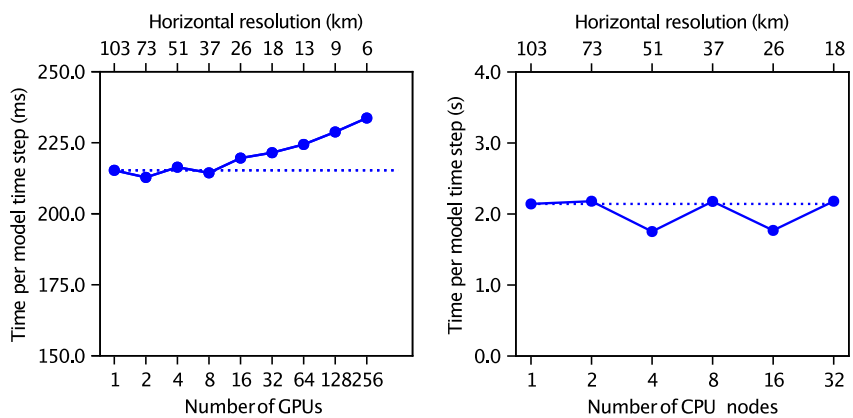

Figure 12. Time per model timestep versus number of GPUs (left) and CPUs (right) for different horizontal resolutions. The GPUs are NVIDIA A100 GPUs, and the CPUs are Intel Icelake 8352Y processors. We use 16 MPI ranks per CPU node.

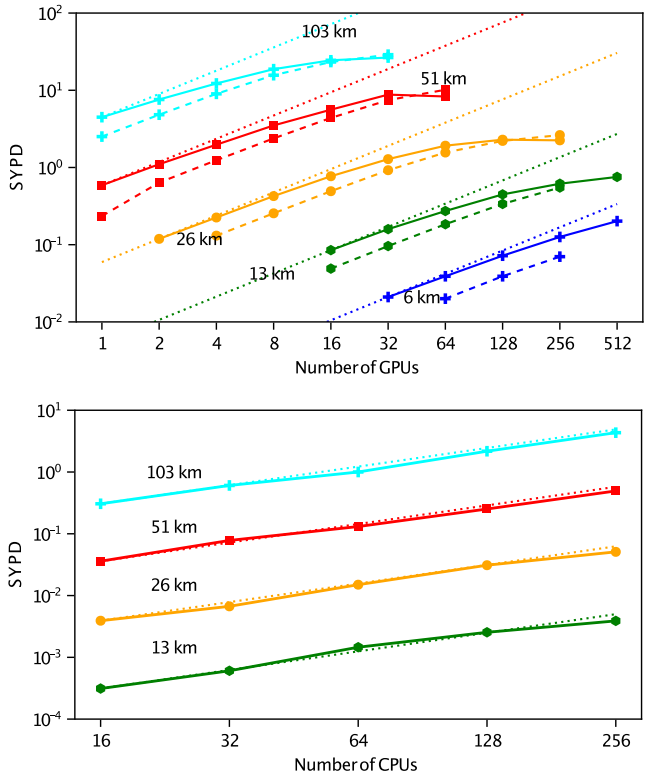

Figure 13. SYPD versus number of GPUs (top) and CPUs (bottom) for different horizontal resolutions when running the moist baroclinic wave benchmark. The GPU scaling runs were carried out on Google Cloud Platform NVIDIA H100's (solid lines) and NCAR Derecho NVIDIA A100's (dashed lines). The CPU scaling runs were carried out at Caltech's Resnick High Performance Computing Center.

A key benchmark is the C++/Kokkos-based Simple Cloud-Resolving E3SM Atmosphere Model (SCREAM). Donahue et al. (2024) report a throughput of 0.52 SYPD for their dynamical core on the Perlmutter GPU system, using 1536 NVIDIA A100 GPUs at a horizontal resolution of 3.25 km. In comparison, the CliMA dynamical core (including 1-moment microphysics) achieves 0.20 SYPD at a resolution of 6 km on 256 GCP H100 GPUs. A direct comparison is challenging due to substantial differences in model configuration, hardware, and resolution. However, accounting for the significant differences in resolution (a factor of $\sim$2, implying a factor of $\sim$8 in computational cost), hardware (1536 previous-generation A100s vs. 256 current-generation H100s), and dynamical core formulation, the performance of the two models is broadly comparable, demonstrating the competitiveness of our Julia-based framework.

Another relevant benchmark is the Python-based Pace model, which, like our model, is written in a high-level language to achieve performance portability and the ability to incorporate machine learning components. Dahm et al. (2023) report a time per timestep of 3.98 s for their dynamical core (a reimplementation of GFDL's FV3) at approximately 50 km resolution on 6 NVIDIA P100 GPUs. For comparison, our dynamical core (including 1-moment microphysics) achieves a time per timestep of approximately 0.22 s at a similar 51-km resolution on 4 A100 GPUs (Figure 12). While a direct comparison is challenging due to the substantial differences in hardware generation (P100 vs. A100), GPU count (6 vs. 4), and dynamical core configuration, the results strongly highlight the performance of our Julia-based implementation, which achieves a nearly order-of-magnitude faster time per timestep. This demonstrates that a modern high-level language such as Julia, when coupled with an appropriate performance-portability framework, can deliver performance competitive with or superior to traditional low-level languages on GPU architectures.

7.4. Performance Considerations

Within each timestep, performance is governed by the speed of kernel execution and the number of kernel launches. To maximize kernel speed, we parallelize pointwise and finite difference kernels so that each nodal point value is evaluated in a separate GPU thread, and we parallelize SEM operations so that each horizontal slice is evaluated in a separate thread. We also organize our data in global memory so that more frequent operations only read from nearby memory locations (Appendix F).

The Julia programming language offers syntactic support for inline fusion of array operations, which we extend to non-local operators such as vertical/horizontal derivatives, and we generalize this functionality to allow fusion across multiple functions. This allows us to have fewer separate kernel launches, reducing the number of expensive loads from global memory. However, overly aggressive fusion can lead to register spillover and running out of shared memory. We are currently exploring memory footprint reduction and heuristics for determining optimal kernel fusion to improve computational performance further.

8. Conclusions

We have presented the nonhydrostatic dynamical core of the CliMA atmosphere model. While individual components of this dynamical core are inspired by prior work—such as the use of horizontal spectral elements also found in the Community Atmosphere Model (CAM) and the Energy Exascale Earth System Model (E3SM) (Guba et al., 2014; P. H. Lauritzen et al., 2018; M. A. Taylor & Fournier, 2010; M. A. Taylor et al., 2020; M. Taylor et al., 2023) or a focus on energy conservation that is also central to the Nonhydrostatic Icosahedral Atmospheric Model (NICAM) (Satoh, 2003; Satoh et al., 2008)—our framework presents a novel synthesis.

In contrast to models that use dry internal energy or potential temperature as a prognostic variable, our choice of *moist total energy* enables the exact conservation of total energy, mass, and water simultaneously. This property holds under fully moist dynamics, even with subgrid-scale parameterizations such as hyperdiffusion, complex topography, and in a deep atmosphere, thereby overcoming many of the energy conservation challenges and the need for ad-hoc fixers detailed in P. H. Lauritzen et al. (2022). This is achieved through the combination of our consistent thermodynamic framework (Section 2) and hybrid discretization (Section 5 and Appendices). This choice prioritizes the conservation of total energy, which is fundamental for preventing climate drift. The trade-off is that other quantities, such as potential temperature, are not explicitly materially conserved. We consider this advantageous, as potential temperature is not physically conserved under the moist, diabatic conditions of the real atmosphere.

The conservation properties are achieved by using a consistent moist thermodynamic framework and specific total energy as a prognostic variable, along with a combination of a spectral element discretization in the horizontal and finite differences on a staggered grid in the vertical. To make the numerical methods conservative, we employed carefully chosen reconstructions of quantities at vertical grid faces and centers. The choice of governing equations and discretization strategy also ensures that the model achieves the chosen order of accuracy (e.g., 3rd order in the horizontal) even in the presence of moist processes and complex topography. Implicit timestepping of fast vertical terms removes timestep limitations from fast vertical dynamics and renders the horizontal propagation of sound waves as the limiting factor for the length of timesteps.

Benchmark tests showed the fidelity of the numerical simulations and demonstrated the conservation properties in practice. They also demonstrated the scalability and performance portability of the dynamical core on CPU and GPU platforms, including cloud GPUs. In particular, the GPU performance is of importance for the practical usability of the model for advancing fundamental climate science and climate modeling, as it enables atmosphere simulations at 25–50 km resolutions at throughputs of >1 SYPD on a few dozen GPUs, which are routinely available even in smaller-scale compute centers and from cloud providers. This opens higher-resolution atmosphere modeling both to the broader scientific community that can access modest-size GPU resources and to the generation of large ensembles of atmosphere simulations, for example, for climate risk assessments (Schneider et al., 2023).

The dynamical core is written in the high-level Julia language, which is performance portable and facilitates integration and possible online training of machine-learning components (e.g., Charbonneau et al., 2025; Christopoulos et al., 2024). This is achieved using gradient-free methods, which do not require the model to be backwards-differentiable and are well-suited for learning from the noisy statistics of turbulent flows (e.g., Dunbar et al., 2022; Huang, Huang, et al., 2022; Huang, Schneider, & Stuart, 2022; Kovachki & Stuart, 2019; Schneider et al., 2017; Schneider et al., 2023). While not required by our current methods, the model's implementation in Julia makes it amenable to automatic differentiation, providing a promising avenue for future work with gradient-based approaches. Subsequent companion papers will describe parameterizations used in the atmosphere model, including some incorporating machine-learning components.

The dynamical core presented here provides a robust and computationally efficient foundation for a new generation of atmosphere models. The successful validation of the core is the first step in a broader research program. Future work will proceed along several avenues:

- We have coupled the dynamical core with a comprehensive suite of physical parameterizations and are in the process of demonstrating the scientific fidelity of the resulting atmosphere model in long-term climate simulations, which will be the subject of forthcoming companion papers.
- To fully realize the model's versatility from global climate down to meter-scale large-eddy simulations, future development will target the implementation of fully implicit timestepping schemes. This will overcome the horizontal Courant number limitations of the current HEVI scheme and enable the efficient simulation of turbulent flows at very high resolutions.
- The model's implementation in Julia offers unique opportunities for novel scientific workflows. While our current machine learning integration relies on gradient-free methods, the framework's amenability to automatic differentiation provides a promising avenue for future exploration of gradient-based learning approaches for parameterization development.

Appendix A: Reference Temperature Invariance

The physics of the model must be independent of the arbitrary reference temperature T_0. A shift in the reference temperature, $T_0 \to T_0' = T_0 + \delta T_0$, requires a commensurate shift in the reference latent heats consistent with Kirchhoff's Law (Equation 7), for example, $L_{v,0} \to L_{v,0}' = L_{v,0} + (c_{pv} - c_{pl})\delta T_0$, and similarly for reference internal energies of phase changes. With these consistent definitions, it follows that under a reference temperature shift, the specific internal energies (I_μ) and specific enthalpies (h_μ) of each constituent μ transform by subtracting a constant offset, C_μ:

$$I_\mu \to I_\mu' = I_\mu - C_\mu, \tag{A1}$$

$$h_\mu \to h_\mu' = h_\mu - C_\mu. \tag{A2}$$

Crucially, while the offset for dry air ($\mu = d$) is distinct, the offsets for all three water species—vapor (v), liquid (l), and ice (i)—are identical:

$$C_d = c_{pd}\delta T_0, \tag{A3}$$

$$C_v = C_l = C_i = c_{pl}\delta T_0 \equiv C_{\text{water}}. \tag{A4}$$

This property is the key to the invariance of the governing equations.

A1. Invariance of the Governing Equations

The total energy Equation 26c is a conservation law for the specific total energy $e_{\text{tot}} = \sum_\mu q_\mu I_\mu + \Phi + \kappa + \kappa_{\text{SGS}}$. Under the reference temperature shift, the specific total energy and total enthalpy transform as:

$$e_{\text{tot}} \to e_{\text{tot}}' = e_{\text{tot}} - \sum_\mu q_\mu C_\mu, \tag{A5}$$

$$h_{\text{tot}} \to h_{\text{tot}}' = h_{\text{tot}} - \sum_\mu q_\mu C_\mu. \tag{A6}$$

The total energy equation involves advective and sedimentation fluxes of total enthalpy, as well as a subgrid-scale (SGS) diffusive flux of total enthalpy, $\boldsymbol{\mathcal{F}}_h$. As established in Section 3.3, the transformation of this flux under a shift in T_0 is given by the enthalpy carried by the SGS water fluxes, $\delta(\boldsymbol{\mathcal{F}}_h) = -\sum_{\mu\in\{v,l,i\}} C_\mu \boldsymbol{\mathcal{F}}_{q_\mu}$. To demonstrate invariance, we must show that the additional terms generated in the total energy equation by the transformation of all its components sum to zero.

Summing all extra terms generated in the total energy equation—from the time derivative, advection, sedimentation, and SGS fluxes—yields the requirement that

$$\sum_\mu C_\mu \left[\frac{\partial(\rho q_\mu)}{\partial t} + \nabla\cdot\left(\rho q_\mu \left(\boldsymbol{u} - w_\mu \hat{\boldsymbol{k}}\right)\right) + \nabla\cdot\left(\rho \boldsymbol{\mathcal{F}}_{q_\mu}\right)\right] = 0, \tag{A7}$$

where we have omitted the hyperdiffusive fluxes for notational convenience (they do not break the invariance). The expression in the brackets is the mass conservation equation for constituent μ. This sum is zero for both the dry air and water components. For dry air ($\mu = d$), the sedimentation velocity w_d and SGS flux $\boldsymbol{\mathcal{F}}_{q_d}$ are zero by definition, and the bracketed term vanishes by dry-air mass conservation. For the water phases ($\mu \in \{v, l, i\}$), the offset $C_\mu = C_{\text{water}}$ is identical for all species and can be factored out of the sum. The remaining sum over the bracketed terms,

$$\sum_{\mu\in\{v,l,i\}}\left[\frac{\partial(\rho q_\mu)}{\partial t}+\nabla\cdot\left(\rho q_\mu\left(\boldsymbol{u}-w_\mu\hat{\boldsymbol{k}}\right)\right)+\nabla\cdot\left(\rho\boldsymbol{\mathcal{F}}_{q_\mu}\right)\right], \tag{A8}$$

is equivalent to the total water conservation law (Equation 26d), using the definitions $q_t = \sum_{\mu\in\{v,l,i\}} q_\mu$, $W_{q_t} = \sum_{\mu\in\{v,l,i\}} q_\mu w_\mu$, and $\boldsymbol{\mathcal{F}}_{q_t} = \sum_{\mu\in\{v,l,i\}} \boldsymbol{\mathcal{F}}_{q_\mu}$. Since the total water conservation law has no physical source terms, Equation A8 is zero (up to the omitted hyperdiffusion terms). Thus, the contributions from both dry air and water vanish, and the governing equations are invariant to the choice of T_0.

A2. Invariance of Surface Boundary Conditions

The physical evolution of the model must be independent of the arbitrary reference temperature T_0. A shift in this reference temperature, $T_0 \to T_0' = T_0 + \delta T_0$, induces a redefinition of the prognostic total energy variable, $e_{\text{tot}} \to e_{\text{tot}}' = e_{\text{tot}} - \sum_\mu q_\mu C_\mu$. For the model to produce a physically invariant solution while prognosing this numerically different quantity, the boundary fluxes for total enthalpy must transform in a compensating manner. This section demonstrates that the model's formulation for the total non-radiative surface energy flux possesses the required transformation property.

The total parameterized non-radiative energy flux from the surface into the atmosphere, denoted $F_{\text{param,tot}}$, is composed of the upward turbulent enthalpy flux, EF, and the downward enthalpy flux carried by precipitation, F_{precip}:

$$F_{\text{param,tot}} = \text{EF} - F_{\text{precip}}. \tag{A9}$$

The precipitation flux is the sum of the total enthalpies carried by liquid and ice precipitation, $F_{\text{precip}} = P_l h_{\text{tot},l} + P_i h_{\text{tot},i}$. To prove invariance, we analyze the transformation of each component under a T_0 shift.

The transformation of the turbulent flux, $\delta(\text{EF})$, is derived from its decomposition into sensible and latent heat fluxes, EF = SHF + LHF. The transformation of the latent heat flux is:

$$\delta(\text{LHF}) = \delta(L_{v,0}E) = E\delta(L_{v,0}) = E(c_{pv}-c_{pl})\delta T_0. \tag{A10}$$

The sensible heat flux is defined as the sum of the thermal diffusion flux and the static energy carried by evaporation, $\text{SHF} = -\rho_{\text{sfc}} g_h \Delta_s(s_d) + E\cdot(h_v + \Phi - L_{v,0})$. Its transformation is the sum of the transformations of its components. The thermal diffusion term is invariant, as $\delta(\Delta_s(s_d)) = \Delta_s(\delta(h_d)) = \Delta_s(-C_d) = 0$. The transformation of the advected static energy is:

$$\delta(E\cdot(h_v+\Phi-L_{v,0})) = E\cdot(\delta(h_v)-\delta(L_{v,0})) = E\cdot(-C_{\text{water}}-(c_{pv}-c_{pl})\delta T_0) = -Ec_{pv}\delta T_0. \tag{A11}$$

Thus, $\delta(\text{SHF}) = -Ec_{pv}\delta T_0$. Combining the transformations, the total transformation for the turbulent component is:

$$\delta(\text{EF}) = \delta(\text{SHF}) + \delta(\text{LHF}) = -Ec_{pv}\delta T_0 + E(c_{pv}-c_{pl})\delta T_0 = -Ec_{pl}\delta T_0 = -EC_{\text{water}}. \tag{A12}$$

The transformation of the precipitation flux, $\delta(F_{\text{precip}})$, follows from the transformation of the liquid and ice total enthalpies, $\delta(h_{\text{tot},l}) = \delta(h_{\text{tot},i}) = -C_{\text{water}}$:

$$\delta(F_{\text{precip}}) = P_l\delta(h_{\text{tot},l}) + P_i\delta(h_{\text{tot},i}) = -(P_l+P_i)\,C_{\text{water}}. \tag{A13}$$

Combining these results, the transformation of the total parameterized surface energy flux is

$$\delta(F_{\text{param,tot}}) = \delta(\text{EF}) - \delta(F_{\text{precip}}) = -EC_{\text{water}} - (-(P_l+P_i)\,C_{\text{water}}) = -(E-P_l-P_i)\,C_{\text{water}}. \tag{A14}$$

This result is physically consistent. The change in the boundary flux must balance the change in the tendency of the atmosphere's total energy content. The change in the total energy content is $\Delta E_{\text{atm}} = -\int_V \rho \sum_\mu q_\mu C_\mu dV$. Since dry air mass is conserved, the change in the tendency is driven only by the change in total water mass:

$$\frac{d(\Delta E_{\text{atm}})}{dt} = -C_{\text{water}} \frac{dM_{t,\text{atm}}}{dt} = -C_{\text{water}} \int_A (E - P_l - P_i)\, dA. \tag{A15}$$

The global integral of our derived flux transformation, $\int_A \delta(F_{\text{param,tot}})\, dA$, is identical to this required change in the energy tendency. This demonstrates that the model's formulation of the complete surface energy exchange is consistent: the parameterized fluxes correctly track the redefinition of physical energy under a shift in reference temperature, ensuring the overall prognostic system remains physically invariant.

Appendix B: Stretched Terrain-Following Coordinates

The generalized vertical coordinate ξ^3 is obtained by a sequence of transformations, which include topographic smoothing, introduction of a terrain-following coordinate η, and stretching of η to the generalized coordinate ξ^3, which, when uniformly discretized, achieves finer resolution near the surface than aloft.

B1. Topography Smoothing

To avoid generating spurious grid-scale noise (P. Lauritzen et al., 2015), we smooth surface topography $z_s(\lambda, \varphi)$ (longitude λ and latitude φ) by "evolving" it according to the diffusion equation

$$\frac{\partial z_s}{\partial t} = \nu_{\text{top}} \Delta_h z_s, \tag{B1}$$

using the diffusivity $\nu_{\text{top}} = C_o(\delta x)^2/\delta t$, where δx is the SEM resolution, δt is the timestep, and $C_o < 0.5$ is a Courant number (default value $C_0 = 0.05$). For an attenuation of topography at the smallest resolved scale ($\sim 1/\delta x$) by a factor n_a (default value $n_a = 5$), topography needs to diffuse over an interval $(\Delta t)_{\text{top}} \approx [\ln(n_a)/C_o]\delta t$. We use the SEM and the forward Euler method to discretize the diffusion equation as

$$\mathcal{I} z_s|_{t+\delta t} = \mathcal{I} z_s|_t + \delta t\, \nu_{\text{top}} \mathcal{P}\left(\tilde{\Delta}_h \mathcal{I} z_s\right), \tag{B2}$$

where $\mathcal{I} z_s$ is the nodal point interpolation of z_s, $\tilde{\Delta}_h$ is the mixed formulation of the horizontal Laplacian, and $\mathcal{P}$ is the DSS operator, whose weights can be simplified to avoid requiring the value of z_s (Appendix D).

The result is a smoothed topography $\hat{z}_s(\lambda, \varphi)$ for defining terrain-following coordinates. However, for the smooth topography examples in this paper, additional topographic smoothing is unnecessary and is not used.

B2. Terrain-Following Coordinates

We introduce a nondimensional terrain-following coordinate η, with $\eta = 0$ at the surface at the smoothed elevation $z = \hat{z}_s(\lambda, \varphi)$ and $\eta = 1$ at the top of the model domain at a constant altitude $z = z_t$. Established choices of terrain-following coordinates in the CliMA model include:

B2.1. Linearly Attenuating Vertical Coordinate

Gal-Chen and Somerville (1975) offered a simple choice of a terrain-following coordinate,

$$z(\lambda, \varphi, \eta) = \eta z_t + (1 - \eta)\hat{z}_s(\lambda, \varphi), \tag{B3}$$

in which the effect of topography on the vertical coordinate attenuates linearly with altitude.

To ensure horizontal coordinate surfaces above a chosen nondimensional height η_h, a modified version is used (Doms & Baldauf, 2018):

$$z(\lambda,\varphi,\eta) = \begin{cases} \eta z_t + \hat{z}_s(\lambda,\varphi)(\eta_h - \eta)/\eta_h & \text{if } 0 \le \eta \le \eta_h, \\ \eta z_t & \text{if } \eta_h \le \eta \le 1. \end{cases} \tag{B4}$$

Setting $\eta_h = 1$ recovers the original formulation (Equation B3). A suitable value for atmosphere models is $\eta_h \approx 35\ \text{km}/z_t$ (e.g., $\eta_h = 0.58$ for a model top at $z_t = 60$ km), yielding horizontal coordinate surfaces in the stratosphere.

B2.2. Smooth-Level Vertical Coordinate (SLEVE)

Schär et al. (2002) proposed a smoother, multiscale terrain-following coordinate, which we present here in a single-scale version with vanishing warping above η_h (Doms & Baldauf, 2018):

$$z(\lambda,\varphi,\eta) = \begin{cases} \eta z_t + \hat{z}_s(\lambda,\varphi)\dfrac{\sinh\left[(\eta_h - \eta)/(s\eta_h)\right]}{\sinh(1/s)} & \text{if } 0 \le \eta \le \eta_h, \\ \eta z_t & \text{if } \eta_h \le \eta \le 1. \end{cases} \tag{B5}$$

The nondimensional parameter s determines the decay scale sz_t over which topographic influence diminishes. A suitable choice is $s \approx 15\ \text{km}/z_t$ (e.g., $s = 0.5$ for a model domain with $z_t = 30$ km). A consistent choice of where coordinate surfaces flatten then is $\eta_h \approx 3s$. For the coordinate to be well defined (z must increase monotonically with η), it is essential that s is sufficiently large.

While multiscale extensions of SLEVE exist (J. B. Klemp, 2011; Leuenberger et al., 2010; Schär et al., 2002), which may be advantageous in high-resolution models, they are not currently used in the CliMA model.

B3. Vertical Coordinate Stretching

In addition to warping coordinate surfaces over topography, we enhance vertical resolution near the surface through stretching. We introduce a non-dimensional stretched vertical coordinate ξ^3, which, when uniformly discretized, achieves finer resolution near the surface. We choose $\xi^3 \in [0, N_v]$, so that $\delta\xi^3 = 1$ is the uniform grid spacing for N_v vertical layers.

The stretching function is a hyperbolic tangent,

$$\eta = 1 - \frac{\tanh\left[\gamma\left(1 - \xi^3/N_v\right)\right]}{\tanh(\gamma)}, \tag{B6}$$

which monotonically maps $\xi^3 \in [0, N_v] \to \eta \in [0, 1]$. The stretching parameter γ controls the degree of refinement; higher values yield finer near-surface resolution.

The stretching parameter γ is chosen to achieve a given resolution $\delta z|_s$ at a flat surface. With N_v vertical layer centers, γ is obtained by iteratively solving

$$\frac{\delta z|_s}{z_t} = \delta\eta|_s = 1 - \frac{\tanh\left[\gamma(1 - 1/N_v)\right]}{\tanh(\gamma)}, \tag{B7}$$

where $\delta\xi^3 = 1$ corresponds to the level spacing in the stretched vertical coordinate. For the baroclinic wave benchmarks in this paper, we use $z_t = 30$ km, $N_v = 43$ and a target $\delta z|_s = 30$ m, which yields $\gamma = 2.8$. This results in a smooth transition of resolution from 30 m near the surface to 1 km at 9 km altitude, asymptoting to 2 km near the model top. (For our default atmosphere model configuration, to be described in a follow-on paper, we use higher vertical resolution.)

The parameters controlling vertical stretching (e.g., γ, N_v) and terrain-following warping (e.g., s, η_h) must be carefully chosen to ensure the vertical coordinate remains monotonic ($\delta z|_s > 0$), even over the highest topographic features where coordinate layers are most compressed.

Appendix C: GLL Quadrature and Lagrange Polynomials

The SEM horizontal discretization is defined in terms of the GLL quadrature points,

$$\tilde{\zeta}^n_{N_p} \in \{\zeta \mid (1-\zeta^2) P'_{N_p}(\zeta) = 0\}, \tag{C1}$$

where $P_{N_p}(\zeta)$ is the Legendre polynomial of order N_p, and $P'_{N_p}(\zeta)$ is its derivative. The set of Lagrange polynomials through any $N_p + 1$ points constitutes a basis for the space of all polynomials of order N_p. We use Lagrange polynomials through the $N_p + 1$ GLL points, where the n-th Lagrange polynomial is

$$L^n_{N_p}(\zeta) = \prod_{\substack{m=1 \\ m \neq n}}^{N_p+1} \frac{\zeta - \tilde{\zeta}^m_{N_p}}{\tilde{\zeta}^n_{N_p} - \tilde{\zeta}^m_{N_p}}. \tag{C2}$$

This satisfies the orthogonality relation

$$L^n_{N_p}\left(\tilde{\zeta}^m_{N_p}\right) = \delta^m_n, \tag{C3}$$

and its derivative at each GLL point is

$$L^n_{N_p}{}'\left(\tilde{\zeta}^m_{N_p}\right) = \begin{cases} -\frac{1}{4} N_p (N_p+1) & \text{if } n = m \text{ and } n = 1, \\ 0 & \text{if } n = m \text{ and } 2 \le n \le N_p, \\ \frac{1}{4} N_p (N_p+1) & \text{if } n = m \text{ and } n = N_p + 1, \\ \dfrac{1}{\tilde{\zeta}^m_{N_p} - \tilde{\zeta}^n_{N_p}} \dfrac{P_{N_p}\left(\tilde{\zeta}^m_{N_p}\right)}{P_{N_p}\left(\tilde{\zeta}^n_{N_p}\right)} & \text{if } n \neq m. \end{cases} \tag{C4}$$

We can exactly integrate any polynomial $F(\zeta)$ with a highest order no greater than $2N_p - 1$ over the interval $[-1, 1]$ using the quadrature rule

$$\int_{-1}^{1} F(\zeta)\, d\zeta = \int_{-1}^{1} \left(\sum_{n=1}^{N_p+1} L^n_{N_p}(\zeta) F\left(\tilde{\zeta}^n_{N_p}\right) \right) d\zeta = \sum_{n=1}^{N_p+1} w^n_{N_p} F\left(\tilde{\zeta}^n_{N_p}\right), \tag{C5}$$

where the weights are given by

$$w^n_{N_p} = \int_{-1}^{1} L^n_{N_p}(\zeta)\, d\zeta = \begin{cases} \dfrac{2}{N_p(N_p+1)} & \text{if } n = 1 \text{ or } n = N_p + 1, \\ \dfrac{2}{N_p(N_p+1) P_{N_p}\left(\tilde{\zeta}^n_{N_p}\right)^2} & \text{if } 2 \le n \le N_p. \end{cases} \tag{C6}$$

For higher-order polynomial functions, the quadrature rule can also be applied, leading to an approximate orthogonality relation for inner products of Lagrange polynomials,

$$\int_{-1}^{1} L_{N_p}^n(\zeta) L_{N_p}^m(\zeta)\, d\zeta \approx \sum_{q=1}^{N_p+1} w_{N_p}^q L_{N_p}^n\left(\hat{\zeta}_{N_p}^q\right) L_{N_p}^m\left(\hat{\zeta}_{N_p}^q\right) = \sum_{q=1}^{N_p+1} w_{N_p}^q \delta_n^q \delta_m^q = w_{N_p}^n \delta_n^m. \tag{C7}$$

With this integrand of order $2N_p$, the quadrature error is bounded by

$$\left| w_{N_p}^n \delta_n^m - \int_{-1}^{1} L_{N_p}^n(\zeta) L_{N_p}^m(\zeta)\, d\zeta \right| \leq \frac{(N_p+1)\, 2^{2N_p+1} (N_p!)^4}{N_p (2N_p+1)\, ((2N_p)!)^3} \left| L_{N_p}^{n\ (N_p)} L_{N_p}^{m\ (N_p)} \right|, \tag{C8}$$

where $L_{N_p}^{n\ (N_p)}$ denotes the N_p-th derivative

$$L_{N_p}^{n\ (N_p)} = \frac{\partial^{N_p}}{\partial \zeta^{N_p}} L_{N_p}^n(\zeta) = \prod_{\substack{m=1 \\ m \neq n}}^{N_p+1} \frac{1}{\zeta_{N_p}^n - \zeta_{N_p}^m}. \tag{C9}$$

We typically set $N_p = 3$, in which case this error bound is on the order of 10^{-4}.

Appendix D: Discrete Metric Terms

We express the radial unit vector $\hat{\boldsymbol{k}} = \hat{k}_x \hat{\boldsymbol{x}} + \hat{k}_y \hat{\boldsymbol{y}} + \hat{k}_z \hat{\boldsymbol{z}}$, where $\hat{\boldsymbol{z}}$ is aligned with the planetary spin axis, using generalized spherical coordinates as $(\lambda, \varphi, 1)$, with longitude $\lambda = \arctan\left(\hat{k}_y, \hat{k}_x\right)$ and latitude $\varphi = \arcsin \hat{k}_z$. Then, we can write the horizontal unit basis vectors as

$$\hat{\boldsymbol{\lambda}} = \hat{\lambda}_x \hat{\boldsymbol{x}} + \hat{\lambda}_y \hat{\boldsymbol{y}} + \hat{\lambda}_z \hat{\boldsymbol{z}} = -(\sin \lambda) \hat{\boldsymbol{x}} + (\cos \lambda) \hat{\boldsymbol{y}}, \tag{D1a}$$

$$\hat{\boldsymbol{\varphi}} = \hat{\varphi}_x \hat{\boldsymbol{x}} + \hat{\varphi}_y \hat{\boldsymbol{y}} + \hat{\varphi}_z \hat{\boldsymbol{z}} = -(\cos \lambda)(\sin \varphi) \hat{\boldsymbol{x}} - (\sin \lambda)(\sin \varphi) \hat{\boldsymbol{y}} + (\cos \varphi) \hat{\boldsymbol{z}}, \tag{D1b}$$

so that the horizontal metric Jacobian at $\boldsymbol{r} = R\hat{\boldsymbol{k}}$ becomes

$$\boldsymbol{J}_h^{\text{shallow}}\left(\xi^1, \xi^2; n_h\right) = R \begin{pmatrix} \hat{\lambda}_x & \hat{\lambda}_y & \hat{\lambda}_z \\ \hat{\varphi}_x & \hat{\varphi}_y & \hat{\varphi}_z \end{pmatrix} \begin{pmatrix} \frac{\partial \hat{k}_x}{\partial \xi^1} & \frac{\partial \hat{k}_x}{\partial \xi^2} \\ \frac{\partial \hat{k}_y}{\partial \xi^1} & \frac{\partial \hat{k}_y}{\partial \xi^2} \\ \frac{\partial \hat{k}_z}{\partial \xi^1} & \frac{\partial \hat{k}_z}{\partial \xi^2} \end{pmatrix}. \tag{D2}$$

The derivatives of $\hat{\boldsymbol{k}}$ are obtained through automatic differentiation of cubed sphere mesh coordinates. In the shallow-atmosphere approximation, this horizontal metric Jacobian is used at all points, and the planetary angular velocity is additionally replaced by its local vertical projection $\boldsymbol{\Omega} \approx (\Omega \sin \varphi) \hat{\boldsymbol{k}}$, where Ω denotes the planetary rotation frequency. In a deep atmosphere, we use the full angular velocity $\boldsymbol{\Omega} = \Omega \hat{\boldsymbol{z}}$ and the exact horizontal metric Jacobian at $\boldsymbol{r} = r\hat{\boldsymbol{k}}$,

$$\boldsymbol{J}_h^{\text{deep}}\left(\xi^1, \xi^2, \xi^3; n_h, n_v\right) = \frac{1}{R} r\left(\xi^1, \xi^2, \xi^3; n_h, n_v\right) \boldsymbol{J}_h^{\text{shallow}}\left(\xi^1, \xi^2; n_h\right). \tag{D3}$$

In a box domain, where the horizontal unit basis vectors are $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$, we define the horizontal metric Jacobian at $\boldsymbol{r} = x\hat{\boldsymbol{x}} + y\hat{\boldsymbol{y}} + z\hat{\boldsymbol{z}}$ as

$$\boldsymbol{J}_h^{\text{box}}\left(\xi^1,\xi^2;n_h\right)=\begin{pmatrix}\dfrac{\partial x}{\partial \xi_1} & 0\\ 0 & \dfrac{\partial y}{\partial \xi_2}\end{pmatrix}, \tag{D4}$$

with the derivatives of x and y specified as constant grid parameters. In general, the horizontal metric determinant is given by

$$J_h=\det \boldsymbol{J}_h=\begin{cases}R^2\left\|\dfrac{\partial \hat{\boldsymbol{k}}}{\partial \xi^1}\times\dfrac{\partial \hat{\boldsymbol{k}}}{\partial \xi^2}\right\| & \text{in a shallow atmosphere,}\\ r^2\left\|\dfrac{\partial \hat{\boldsymbol{k}}}{\partial \xi^1}\times\dfrac{\partial \hat{\boldsymbol{k}}}{\partial \xi^2}\right\| & \text{in a deep atmosphere,}\\ \dfrac{\partial x}{\partial \xi^1}\dfrac{\partial y}{\partial \xi^2} & \text{in a box.}\end{cases} \tag{D5}$$

The full metric Jacobian is defined in terms of the $\boldsymbol{J}_h$ components as

$$\boldsymbol{J}\left(\xi^1,\xi^2,\xi^3;n_h,n_v\right)=\begin{pmatrix}J_h^{11} & J_h^{12} & 0\\ J_h^{21} & J_h^{22} & 0\\ \dfrac{\partial z}{\partial \xi^1} & \dfrac{\partial z}{\partial \xi^2} & \dfrac{\partial z}{\partial \xi^3}\end{pmatrix}. \tag{D6}$$

We compute the derivatives of z using discrete approximations, calculating horizontal derivatives of z using the strong SEM formulation and the vertical derivative using finite differences, with one-sided differences on boundary faces:

$$\left(\frac{\partial z}{\partial \xi^3}\right)^{\text{f}}_{n_v}=\begin{cases}z_2^{\text{f}}-z_1^{\text{f}} & n_v=1,\\ z_{n_v}^{\text{c}}-z_{n_v-1}^{\text{c}} & 2\le n_v\le N_v,\\ z_{N_v+1}^{\text{f}}-z_{N_v}^{\text{f}} & n_v=N_v+1.\end{cases} \tag{D7}$$

The covariant and contravariant representations of the metric tensor are $\boldsymbol{g}_{ij}=\boldsymbol{J}^{\intercal}\boldsymbol{J}$ and $\boldsymbol{g}^{ij}=\left(\boldsymbol{g}_{ij}\right)^{-1}$, and the metric determinant is

$$J=\det \boldsymbol{J}=J_h\frac{\partial z}{\partial \xi^3}. \tag{D8}$$

Because the GLL weights $w_{N_p}^n$, the vertical derivative $\partial z/\partial \xi^3$, and the radial distance r are all continuous across element boundaries, we can express the DSS operator as

$$\mathcal{P}(\mathcal{I}\psi)\left[\boldsymbol{r}\left(\xi_{N_p}^{n_1},\xi_{N_p}^{n_2},\xi^3;n_h,n_v\right)\right]=\sum_{(m_1,m_2,m_h)\in C_{n_h}^{n_1,n_2}}\mathcal{W}\left(\xi_{N_p}^{m_1},\xi_{N_p}^{m_2},\xi^3;m_h,n_v\right)\hat{\psi}_{m_h,n_v}\left(\xi_{N_p}^{m_1},\xi_{N_p}^{m_2},\xi^3\right), \tag{D9}$$

where the continuity of $w_{N_p}^{n_1}w_{N_p}^{n_2}\partial z/\partial \xi^3$ and r^2 simplifies the DSS weights from normalized volume elements δV to normalized horizontal metric determinants J_h, with only the shallow-atmosphere determinant required on a sphere:

$$\mathcal{W}\left(\hat{\zeta}^{n_1}_{N_p}, \hat{\zeta}^{n_2}_{N_p}, \xi^3; n_h, n_v\right) = \frac{J_h^{\text{shallow/box}}\left(\hat{\zeta}^{n_1}_{N_p}, \hat{\zeta}^{n_2}_{N_p}; n_h\right)}{\sum\limits_{(m_1,m_2,m_h)\in C^{n_1,n_2}_{n_h}} J_h^{\text{shallow/box}}\left(\hat{\zeta}^{n_1}_{N_p}, \hat{\zeta}^{n_2}_{N_p}; m_h\right)}. \tag{D10}$$

Unlike the vertical derivatives, the horizontal derivatives of $\hat{\boldsymbol{k}}$ and z are discontinuous across element boundaries, so we must apply $\mathcal{P}$ after all operations involving J and $\boldsymbol{g}$ before we can horizontally differentiate the results of those operations.

Appendix E: Discrete Conservation Properties

We approximate integrals over the full domain Ω as

$$\int_\Omega \psi^{\mathrm{c}}\, dV \approx \mathscr{I}^{\mathrm{c}}_\Omega(\psi^{\mathrm{c}}) = \sum_{n_v=1}^{N_v}\sum_{n_h=1}^{N_h}\sum_{n_1,n_2=1}^{N_p+1} (\psi\delta V)^{\mathrm{c}}_{n_v}\left(\hat{\zeta}^{n_1}_{N_p}, \hat{\zeta}^{n_2}_{N_p}; n_h\right), \tag{E1a}$$

$$\int_\Omega \psi^{\mathrm{f}}\, dV \approx \mathscr{I}^{\mathrm{f}}_\Omega(\psi^{\mathrm{f}}) = \sum_{n_v=1}^{N_v}\sum_{n_h=1}^{N_h}\sum_{n_1,n_2=1}^{N_p+1} \frac{1}{2}\left[(\psi\delta V)^{\mathrm{f}}_{n_v}\left(\hat{\zeta}^{n_1}_{N_p}, \hat{\zeta}^{n_2}_{N_p}; n_h\right) + (\psi\delta V)^{\mathrm{f}}_{n_v+1}\left(\hat{\zeta}^{n_1}_{N_p}, \hat{\zeta}^{n_2}_{N_p}; n_h\right)\right], \tag{E1b}$$

which can be decomposed into horizontal and vertical inner products,

$$\mathscr{I}^{\mathrm{c}}_\Omega(\psi^{\mathrm{c}}) = \sum_{n_v=1}^{N_v} \mathscr{I}_{\Omega_h\left(\frac{1}{2};n_v\right)}(1,\psi^{\mathrm{c}}) = \sum_{n_h=1}^{N_h}\sum_{n_1,n_2=1}^{N_p+1} \mathscr{I}^{\mathrm{c}}_{\Omega_v\left(\hat{\zeta}^{n_1}_{N_p},\hat{\zeta}^{n_2}_{N_p};n_h\right)}(1,\psi^{\mathrm{c}}), \tag{E2a}$$

$$\begin{aligned}\mathscr{I}^{\mathrm{f}}_\Omega(\psi^{\mathrm{f}}) &= \sum_{n_v=1}^{N_v} \frac{1}{2}\left[\mathscr{I}_{\Omega_h(0;n_v)}(1,\psi^{\mathrm{f}}) + \mathscr{I}_{\Omega_h(1;n_v)}(1,\psi^{\mathrm{f}})\right] \\ &= \sum_{n_h=1}^{N_h}\sum_{n_1,n_2=1}^{N_p+1} \mathscr{I}^{\mathrm{f}}_{\Omega_v\left(\hat{\zeta}^{n_1}_{N_p},\hat{\zeta}^{n_2}_{N_p};n_h\right)}(1,\psi^{\mathrm{f}}).\end{aligned} \tag{E2b}$$

The horizontal inner product satisfies a discrete divergence theorem,

$$\mathscr{I}_{\Omega_h}\left(\mathcal{I}\psi, \tilde{\nabla}_h\cdot\mathcal{I}\boldsymbol{u}\right) + \mathscr{I}_{\Omega_h}(\mathcal{I}\boldsymbol{u}, \nabla_h\mathcal{I}\psi) = 0, \tag{E3a}$$

$$\mathscr{I}_{\Omega_h}(\mathcal{I}\psi, \nabla_h\cdot\mathcal{I}\boldsymbol{u}) + \mathscr{I}_{\Omega_h}\left(\mathcal{I}\boldsymbol{u}, \tilde{\nabla}_h\mathcal{I}\psi\right) = 0, \tag{E3b}$$

and a discrete Stokes theorem,

$$\mathscr{I}_{\Omega_h}\left(\mathcal{I}\boldsymbol{u}, \tilde{\nabla}_h\times\mathcal{I}\boldsymbol{v}\right) - \mathscr{I}_{\Omega_h}(\mathcal{I}\boldsymbol{v}, \nabla_h\times\mathcal{I}\boldsymbol{u}) = 0, \tag{E4}$$

as well as a higher-order identity for the Laplacian operator,

$$\mathscr{I}_{\Omega_h}\left(\mathcal{I}\phi, \tilde{\Delta}_h\mathcal{I}\psi\right) - \mathscr{I}_{\Omega_h}\left(\mathcal{I}\psi, \tilde{\Delta}_h\mathcal{I}\phi\right) = 0, \tag{E5a}$$

$$\mathscr{I}_{\Omega_h}\left(\mathcal{I}\boldsymbol{u}, \tilde{\Delta}_h\mathcal{I}\boldsymbol{v}\right) - \mathscr{I}_{\Omega_h}\left(\mathcal{I}\boldsymbol{v}, \tilde{\Delta}_h\mathcal{I}\boldsymbol{u}\right) = 0. \tag{E5b}$$

The vertical inner product also satisfies a discrete divergence theorem,

$$\mathscr{I}^{\mathrm{c}}_{\Omega_v}\left(\psi^{\mathrm{c}}, \nabla_v\cdot\boldsymbol{u}^{\mathrm{f}}\right) + \mathscr{I}^{\mathrm{f}}_{\Omega_v}\left(\boldsymbol{u}^{\mathrm{f}}, \nabla_v\psi^{\mathrm{c}}\right) = \psi^{\mathrm{c}}_{N_v}\left(u^3\delta V\right)^{\mathrm{f}}_{N_v+1} - \psi^{\mathrm{c}}_1\left(u^3\delta V\right)^{\mathrm{f}}_1, \tag{E6a}$$

$$\mathscr{I}^{\mathrm{f}}_{\Omega_v}\left(\psi^{\mathrm{f}}, \nabla_v\cdot\boldsymbol{u}^{\mathrm{c}}\right) + \mathscr{I}^{\mathrm{c}}_{\Omega_v}\left(\boldsymbol{u}^{\mathrm{c}}, \nabla_v\psi^{\mathrm{f}}\right) = \psi^{\mathrm{f}}_{N_v+1}\left(u^3\delta V\right)^{\mathrm{c}}_{N_v} - \psi^{\mathrm{f}}_1\left(u^3\delta V\right)^{\mathrm{c}}_1, \tag{E6b}$$

and a discrete Stokes' theorem,

$$\mathscr{I}^{\mathrm{c}}_{\Omega_v}\left(\boldsymbol{u}^{\mathrm{c}}, \nabla_v \times \boldsymbol{v}^{\mathrm{f}}\right) - \mathscr{I}^{\mathrm{f}}_{\Omega_v}\left(\boldsymbol{v}^{\mathrm{f}}, \nabla_v \times \boldsymbol{u}^{\mathrm{c}}\right) = \hat{\mathcal{W}}\Big[(u_2)^{\mathrm{c}}_{N_v}(v_1)^{\mathrm{f}}_{N_v+1} - (u_1)^{\mathrm{c}}_{N_v}(v_2)^{\mathrm{f}}_{N_v+1} - (u_2)^{\mathrm{c}}_{1}(v_1)^{\mathrm{f}}_{1} + (u_1)^{\mathrm{c}}_{1}(v_2)^{\mathrm{f}}_{1}\Big], \quad \text{(E7a)}$$

$$\mathscr{I}^{\mathrm{f}}_{\Omega_v}\left(\boldsymbol{u}^{\mathrm{f}}, \nabla_v \times \boldsymbol{v}^{\mathrm{c}}\right) - \mathscr{I}^{\mathrm{c}}_{\Omega_v}\left(\boldsymbol{v}^{\mathrm{c}}, \nabla_v \times \boldsymbol{u}^{\mathrm{f}}\right) = \hat{\mathcal{W}}\Big[(u_2)^{\mathrm{f}}_{N_v+1}(v_1)^{\mathrm{c}}_{N_v} - (u_1)^{\mathrm{f}}_{N_v+1}(v_2)^{\mathrm{c}}_{N_v} - (u_2)^{\mathrm{f}}_{1}(v_1)^{\mathrm{c}}_{1} + (u_1)^{\mathrm{f}}_{1}(v_2)^{\mathrm{c}}_{1}\Big], \quad \text{(E7b)}$$

with $\hat{\mathcal{W}} = \delta V / J$ denoting the product of GLL weights.

With zero-flux boundary conditions at the top and bottom faces, the divergence theorem guarantees that the advective tendency

$$\left(\partial_t \rho\psi|_{\mathrm{adv}}\right)^{\mathrm{c}} = -\tilde{\nabla}_h \cdot \left(\rho\psi \boldsymbol{u}^h\right)^{\mathrm{c}} - \nabla_v \cdot \left(\rho\psi \boldsymbol{u}^v\right)^{\mathrm{f}} \quad \text{(E8)}$$

does not contribute to the integral of $\rho\psi$,

$$\partial_t \mathscr{I}^{\mathrm{c}}_{\Omega}\left[(\rho\psi)^{\mathrm{c}}\right]|_{\mathrm{adv}} = -\sum_{n_v} \mathscr{I}_{\Omega_h}\left[1, \tilde{\nabla}_h \cdot \left(\rho\psi \boldsymbol{u}^h\right)^{\mathrm{c}}\right] - \sum_{n_h, n_1, n_2} \mathscr{I}^{\mathrm{c}}_{\Omega_v}\left[1, \nabla_v \cdot \left(\rho\psi \boldsymbol{u}^v\right)^{\mathrm{f}}\right] = 0. \quad \text{(E9)}$$

This means that all scalar prognostic variables—air mass, total energy, and tracers—are conserved by advection.

Similarly, Stokes' theorem guarantees that the advective tendency of vertical vorticity,

$$\left(\partial_t \boldsymbol{\omega}^v|_{\mathrm{adv}}\right)^{\mathrm{c}} = \tilde{\nabla}_h \times \left(\partial_t \boldsymbol{u}_h|_{\mathrm{adv}}\right)^{\mathrm{c}}, \quad \text{(E10)}$$

does not contribute to the integral of ω^3:

$$\partial_t \mathscr{I}^{\mathrm{c}}_{\Omega}\left[\left(\omega^3\right)^{\mathrm{c}}\right]|_{\mathrm{adv}} = \sum_{n_v} \mathscr{I}_{\Omega_h}\left[(\boldsymbol{e}_3)^{\mathrm{c}}, \tilde{\nabla}_h \times \left(\partial_t \boldsymbol{u}_h|_{\mathrm{adv}}\right)^{\mathrm{c}}\right] = 0. \quad \text{(E11)}$$

The same is true for the advective tendency of horizontal vorticity,

$$\left(\partial_t \boldsymbol{\omega}^h|_{\mathrm{adv}}\right)^{\mathrm{f}} = \tilde{\nabla}_h \times \left(\partial_t \boldsymbol{u}_v|_{\mathrm{adv}}\right)^{\mathrm{f}} + \nabla_v \times \left(\partial_t \boldsymbol{u}_h|_{\mathrm{adv}}\right)^{\mathrm{c}}, \quad \text{(E12)}$$

up to boundary terms associated with the vertical curl operator. For example, the integral of ω^1 can be expressed as

$$\begin{aligned}\partial_t \mathscr{I}^{\mathrm{f}}_{\Omega}\left[\left(\omega^1\right)^{\mathrm{f}}\right]|_{\mathrm{adv}} &= \sum_{n_v} \mathscr{I}_{\Omega_h}\left[(\boldsymbol{e}_1)^{\mathrm{f}}, \tilde{\nabla}_h \times \left(\partial_t \boldsymbol{u}_v|_{\mathrm{adv}}\right)^{\mathrm{f}}\right] + \sum_{n_h, n_1, n_2} \mathscr{I}^{\mathrm{f}}_{\Omega_v}\left[(\boldsymbol{e}_1)^{\mathrm{f}}, \nabla_v \times \left(\partial_t \boldsymbol{u}_h|_{\mathrm{adv}}\right)^{\mathrm{c}}\right] \\ &= \mathscr{I}_{\Omega_h\left(\frac{1}{2}; 1\right)}\left[\frac{\left(\boldsymbol{e}^2\right)^{\mathrm{c}}}{J^{\mathrm{c}}}, \left(\partial_t \boldsymbol{u}_h|_{\mathrm{adv}}\right)^{\mathrm{c}}\right] - \mathscr{I}_{\Omega_h\left(\frac{1}{2}; N_v\right)}\left[\frac{\left(\boldsymbol{e}^2\right)^{\mathrm{c}}}{J^{\mathrm{c}}}, \left(\partial_t \boldsymbol{u}_h|_{\mathrm{adv}}\right)^{\mathrm{c}}\right].\end{aligned} \quad \text{(E13)}$$

The advective tendency of kinetic energy is given by

$$\left(\partial_t \kappa|_{\mathrm{adv}}\right)^{\mathrm{c}} = \left(\boldsymbol{u}^h \cdot \partial_t \boldsymbol{u}_h|_{\mathrm{adv}}\right)^{\mathrm{c}} + \left(\boldsymbol{g}^{vh} \boldsymbol{u}_h\right)^{\mathrm{c}} \cdot \left\langle \left(\partial_t \boldsymbol{u}_v|_{\mathrm{adv}}\right)^{\mathrm{f}} \right\rangle^{\mathrm{c}} + \left\langle \left(\boldsymbol{g}^{vv} \boldsymbol{u}_v \cdot \partial_t \boldsymbol{u}_v|_{\mathrm{adv}}\right)^{\mathrm{f}} \right\rangle^{\mathrm{c}}. \quad \text{(E14)}$$

The corresponding density-weighted tendency can be expressed as

$$\begin{aligned}\left(\partial_t \rho\kappa|_{\text{adv}}\right)^{\text{c}} &= \left(\kappa\partial_t \rho|_{\text{adv}}\right)^{\text{c}} + \left(\rho\partial_t \kappa|_{\text{adv}}\right)^{\text{c}} \\ &= -\left(\mathcal{D}_h^{\kappa}\rho\kappa\right)^{\text{c}} - \left(\mathcal{D}_v^{\kappa}\rho\kappa\right)^{\text{c}} - \left(\mathcal{D}_h^{\omega}\rho\kappa\right)^{\text{c}} - \left(\mathcal{D}_v^{\omega}\rho\kappa\right)^{\text{c}},\end{aligned} \tag{E15}$$

with the contributions from kinetic energy and vorticity split into the horizontal and vertical terms

$$\left(\mathcal{D}_h^{\kappa}\rho\kappa\right)^{\text{c}} = \kappa^{\text{c}}\tilde{\nabla}_h \cdot \left(\rho \boldsymbol{u}^h\right)^{\text{c}} + \left(\rho \boldsymbol{u}^h\right)^{\text{c}} \cdot \nabla_h \kappa^{\text{c}}, \tag{E16a}$$

$$\left(\mathcal{D}_v^{\kappa}\rho\kappa\right)^{\text{c}} = \kappa^{\text{c}}\nabla_v \cdot \left(\rho \boldsymbol{u}^v\right)^{\text{f}} + \left(\rho \boldsymbol{g}^{vh}\boldsymbol{u}_h\right)^{\text{c}} \cdot \langle \nabla_v \kappa^{\text{c}}\rangle^{\text{c}} + \rho^{\text{c}}\langle \left(\boldsymbol{g}^{vv}\boldsymbol{u}_v\right)^{\text{f}} \cdot \nabla_v \kappa^{\text{c}}\rangle^{\text{c}}, \tag{E16b}$$

$$\begin{aligned}\left(\mathcal{D}_h^{\omega}\rho\kappa\right)^{\text{c}} &= \left(\rho \boldsymbol{u}^h\right)^{\text{c}} \cdot \left\lceil \left(\boldsymbol{\omega}^h \times \boldsymbol{u}^v\right)^{\text{f}} \right\rfloor^{\text{c}} + \left(\rho \boldsymbol{g}^{vh}\boldsymbol{u}_h\right)^{\text{c}} \cdot \left\langle \left(\boldsymbol{\omega}^h \times \boldsymbol{u}^h\right)^{\text{f}} \right\rangle^{\text{c}} \\ &\quad + \rho^{\text{c}}\left\langle \left(\boldsymbol{g}^{vv}\boldsymbol{u}_v\right)^{\text{f}} \cdot \left(\boldsymbol{\omega}^h \times \boldsymbol{u}^h\right)^{\text{f}} \right\rangle^{\text{c}},\end{aligned} \tag{E16c}$$

$$\left(\mathcal{D}_v^{\omega}\rho\kappa\right)^{\text{c}} = \left(\rho \boldsymbol{u}^h\right)^{\text{c}} \cdot \left(\boldsymbol{\omega}^v \times \boldsymbol{u}^h\right)^{\text{c}} = 0, \tag{E16d}$$

where the last term is 0 because $\boldsymbol{u}^h$ is orthogonal to $\boldsymbol{\omega}^v \times \boldsymbol{u}^h$. By the horizontal divergence theorem, the first term integrates to 0,

$$\mathcal{I}_{\Omega}^{\text{c}}\left[\left(\mathcal{D}_h^{\kappa}\rho\kappa\right)^{\text{c}}\right] = \sum_{n_v} \mathcal{I}_{\Omega_h}\left[\kappa^{\text{c}}, \tilde{\nabla}_h \cdot \left(\rho \boldsymbol{u}^h\right)^{\text{c}}\right] + \sum_{n_v} \mathcal{I}_{\Omega_h}\left[\left(\rho \boldsymbol{u}^h\right)^{\text{c}}, \nabla_h \kappa^{\text{c}}\right] = 0. \tag{E17}$$

Given the weighted reconstruction of $(\boldsymbol{u}^v)^{\text{f}}$ from $(\boldsymbol{u}_h)^{\text{c}}$ and $(\boldsymbol{u}_v)^{\text{f}}$, averaging by parts and the vertical divergence theorem ensure that the second term integrates to 0,

$$\begin{aligned}\mathcal{I}_{\Omega}^{\text{c}}\left[\left(\mathcal{D}_v^{\kappa}\rho\kappa\right)^{\text{c}}\right] &= \sum_{n_h,n_1,n_2} \mathcal{I}_{\Omega_v}^{\text{c}}\left[\kappa^{\text{c}}, \nabla_v \cdot \left(\rho \boldsymbol{u}^v\right)^{\text{f}}\right] + \sum_{n_h,n_1,n_2} \mathcal{I}_{\Omega_v}^{\text{c}}\left[\left(\rho \boldsymbol{g}^{vh}\boldsymbol{u}_h\right)^{\text{c}}, \langle \nabla_v \kappa^{\text{c}}\rangle^{\text{c}}\right] + \\ &\quad \sum_{n_h,n_1,n_2} \mathcal{I}_{\Omega_v}^{\text{c}}\left[\rho^{\text{c}}, \left\langle \left(\boldsymbol{g}^{vv}\boldsymbol{u}_v\right)^{\text{f}} \cdot \nabla_v \kappa^{\text{c}}\right\rangle^{\text{c}}\right] \\ &= \sum_{n_h,n_1,n_2} \mathcal{I}_{\Omega_v}^{\text{c}}\left[\kappa^{\text{c}}, \nabla_v \cdot \left(\rho \boldsymbol{u}^v\right)^{\text{f}}\right] + \sum_{n_h,n_1,n_2} \mathcal{I}_{\Omega_v}^{\text{f}}\left[\left(\rho \boldsymbol{u}^v\right)^{\text{f}}, \nabla_v \kappa^{\text{c}}\right] \\ &= 0.\end{aligned} \tag{E18}$$

Given that $\left(\boldsymbol{u}^h\right)^{\text{f}}$ is reconstructed with an unweighted average of $\left(\boldsymbol{u}^h\right)^{\text{c}}$, averaging by parts and the triple product identity also imply that

$$\begin{aligned}\mathcal{I}_{\Omega}^{\text{c}}\left[\left(\mathcal{D}_h^{\omega}\rho\kappa\right)^{\text{c}}\right] &= \sum_{n_h,n_1,n_2} \mathcal{I}_{\Omega_v}^{\text{c}}\left[\left(\rho \boldsymbol{u}^h\right)^{\text{c}}, \left\lceil \left(\boldsymbol{\omega}^h \times \boldsymbol{u}^v\right)^{\text{f}} \right\rfloor^{\text{c}}\right] \\ &\quad + \sum_{n_h,n_1,n_2} \mathcal{I}_{\Omega_v}^{\text{c}}\left[\left(\rho \boldsymbol{g}^{vh}\boldsymbol{u}_h\right)^{\text{c}}, \left\langle \left(\boldsymbol{\omega}^h \times \boldsymbol{u}^h\right)^{\text{f}}\right\rangle^{\text{c}}\right] \\ &\quad + \sum_{n_h,n_1,n_2} \mathcal{I}_{\Omega_v}^{\text{c}}\left[\rho^{\text{c}}, \left\langle \left(\boldsymbol{g}^{vv}\boldsymbol{u}_v\right)^{\text{f}} \cdot \left(\boldsymbol{\omega}^h \times \boldsymbol{u}^h\right)^{\text{f}}\right\rangle^{\text{c}}\right] \\ &= \sum_{n_h,n_1,n_2} \mathcal{I}_{\Omega_v}^{\text{f}}\left[\left(\rho \boldsymbol{u}^h\right)^{\text{f}}, \left(\boldsymbol{\omega}^h \times \boldsymbol{u}^v\right)^{\text{f}}\right] + \sum_{n_h,n_1,n_2} \mathcal{I}_{\Omega_v}^{\text{f}}\left[\left(\rho \boldsymbol{u}^v\right)^{\text{f}}, \left(\boldsymbol{\omega}^h \times \boldsymbol{u}^h\right)^{\text{f}}\right] \\ &= \sum_{n_h,n_1,n_2} \mathcal{I}_{\Omega_v}^{\text{f}}\left[\left(\rho \boldsymbol{\omega}^h\right)^{\text{f}}, \left(\boldsymbol{u}^v \times \boldsymbol{u}^h\right)^{\text{f}}\right] + \sum_{n_h,n_1,n_2} \mathcal{I}_{\Omega_v}^{\text{f}}\left[\left(\rho \boldsymbol{\omega}^h\right)^{\text{f}}, \left(\boldsymbol{u}^h \times \boldsymbol{u}^v\right)^{\text{f}}\right] \\ &= 0.\end{aligned} \tag{E19}$$

This shows that the advective tendency does not contribute to the density-weighted integral of kinetic energy:

$$\partial_t \mathscr{I}^{\mathrm{c}}_{\Omega}[(\rho\kappa)^{\mathrm{c}}]|_{\mathrm{adv}} = -\mathscr{I}^{\mathrm{c}}_{\Omega}\left[\left(\mathcal{D}^{\kappa}_{h}\rho\kappa\right)^{\mathrm{c}} + \left(\mathcal{D}^{\kappa}_{v}\rho\kappa\right)^{\mathrm{c}} + \left(\mathcal{D}^{\omega}_{h}\rho\kappa\right)^{\mathrm{c}} + \left(\mathcal{D}^{\omega}_{v}\rho\kappa\right)^{\mathrm{c}}\right] = 0. \tag{E20}$$

Therefore, in the absence of pressure gradients and sources/sinks, kinetic energy is conserved. Since total energy is also conserved, this implies that the sum of internal, potential, and latent energies is also separately conserved in the absence of pressure gradients and sources/sinks. As a consequence, the conversions between kinetic and internal/potential/latent energy due to pressure gradient-velocity correlations are accurately represented, even in the presence of moist processes.

Although the conservation of scalars and vorticity is a direct consequence of how we construct our spatial discretization and use weak horizontal derivatives, the conservation of kinetic energy relies on three additional properties of our reconstruction:

1. $\nabla \cdot \left(\rho \boldsymbol{u}^{h}\right)^{\mathrm{c}}$ and $\nabla\kappa^{\mathrm{c}}$ are computed with different horizontal derivatives,
2. $(\boldsymbol{u}_{v})^{\mathrm{c}}$ and $(\boldsymbol{u}^{v})^{\mathrm{f}}$ are computed with different vertical averages,
3. $\left(\boldsymbol{\omega}^{h} \times \boldsymbol{u}^{v}\right)^{\mathrm{c}}$ and $\left(\boldsymbol{u}^{h}\right)^{\mathrm{f}}$ are computed with different vertical averages.

Since the conservation of each scalar $\rho\psi$ requires a weak horizontal derivative in the divergence $\nabla \cdot \left(\rho\psi\boldsymbol{u}^{h}\right)^{\mathrm{c}}$, the first property implies that a strong horizontal derivative is needed in $\nabla\kappa^{\mathrm{c}}$. But the remaining two properties do not constrain which vertical averages should be used for specific terms, so our reconstruction is not necessarily unique. For example, an alternative reconstruction that uses an unweighted average to compute $\left(\boldsymbol{\omega}^{h} \times \boldsymbol{u}^{v}\right)^{\mathrm{c}}$ and a weighted average to compute $\left(\boldsymbol{u}^{h}\right)^{\mathrm{f}}$ has exactly the same conservation properties as ours. However, that particular reconstruction is unstable with our spatial and temporal discretizations, generating spurious vertical modes, even though similar reconstructions have been shown to be stable with other discretizations (M. A. Taylor et al., 2020). So, despite the fact that our reconstruction is stable for many configurations, changing any part of our discretization may require swapping weighted and unweighted averages to preserve stability.

Appendix F: Block Vector Representation

Interpolations of scalar fields are implemented as block vectors,

$$\psi^{\mathrm{f}} \in \mathbb{R}^{N_v+1} \otimes \mathbb{R}^{N_p+1} \otimes \mathbb{R}^{N_p+1} \otimes \mathbb{R}^{N_h}, \tag{F1a}$$

$$\psi^{\mathrm{c}} \in \mathbb{R}^{N_v} \otimes \mathbb{R}^{N_p+1} \otimes \mathbb{R}^{N_p+1} \otimes \mathbb{R}^{N_h}, \tag{F1b}$$

with the full atmospheric state given by

$$\boldsymbol{Y} = \begin{pmatrix} \rho^{\mathrm{c}} \\ (\rho e_{\mathrm{tot}})^{\mathrm{c}} \\ \left(\rho q_{t}\right)^{\mathrm{c}} \\ \left(\rho q_{\mu}^{\sigma}\right)^{\mathrm{c}} \\ (\rho\chi)^{\mathrm{c}} \\ (\boldsymbol{u}_{h})^{\mathrm{c}} \\ (\boldsymbol{u}_{v})^{\mathrm{f}} \end{pmatrix}. \tag{F2}$$

Non-scalar fields are represented using an additional block axis to store vector/tensor components. The order of block axes prioritizes operations as follows:

1. Vertical interpolations and derivatives,
2. Horizontal derivatives,

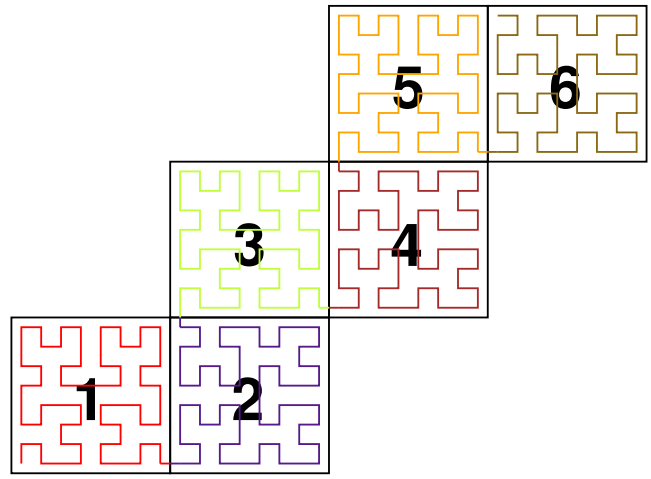

Figure F1. Panels of the cubed sphere with elements connected by a Hilbert space-filling curve.

3. Horizontal DSS.

This reflects how many operations of each type we perform—aside from pointwise computations, vertical interpolations and derivatives are our most common type of operation, while DSS is the least common. Operations along the first few axes of block vectors tend to be fastest because they operate on adjacent areas of memory. The last block axis represents the horizontal element index n_h and corresponds to two spatial dimensions. These are mapped to one dimension while approximately preserving locality by ordering the values of n_h along a generalized Hilbert space-filling curve, which, unlike traditional Hilbert curves, does not require the number of elements along each side to be a power of 2 (Červený, 2024). For simulations in a spherical shell, 6 connected space-filling curves are generated on panels of the cubed sphere (S2 Geometry Library, 2024), as shown in Figure F1.

Appendix G: Implicit Solver

Using the block vector representation of the atmospheric state (Equation F2), we can express the implicit tendency as

$$\boldsymbol{T}^I(\boldsymbol{Y}) = \boldsymbol{T}^I_{\text{adv}}(\boldsymbol{Y}) + \boldsymbol{T}^I_{\text{dif}}(\boldsymbol{Y}), \tag{G1}$$

with an advective component

$$\boldsymbol{T}^I_{\text{adv}}(\boldsymbol{Y},t) = \begin{pmatrix} -\nabla_v \cdot \left[\rho\left(\boldsymbol{u}^v - W_{q_t}\hat{\boldsymbol{k}}^v\right)\right]^{\text{f}} \\ -\nabla_v \cdot \left[\rho\left(h_{\text{tot}}\boldsymbol{u}^v - W_h\hat{\boldsymbol{k}}\right)\right]^{\text{f}} \\ -\nabla_v \cdot \left[\rho\left(q_t\boldsymbol{u}^v - W_{q_t}\hat{\boldsymbol{k}}^v\right)\right]^{\text{f}} \\ -\nabla_v \cdot \left[\rho q^\sigma_\mu\left(\boldsymbol{u}^v - w^\sigma_\mu\hat{\boldsymbol{k}}^v\right)\right]^{\text{f}} \\ -\nabla_v \cdot \left[\rho\chi\left(\boldsymbol{u}^v - w_\chi\hat{\boldsymbol{k}}^v\right)\right]^{\text{f}} \\ 0 \\ -\left[\nabla_v p^{\text{c}}\right]/\rho^f - \nabla_v\Phi^{\text{c}} \end{pmatrix}, \tag{G2}$$

and a diffusive/damping component

$$\boldsymbol{T}^I_{\mathrm{dif}}(\boldsymbol{Y},t) = \begin{pmatrix} 0 \\ -\nabla_v \cdot [\rho \boldsymbol{\mathcal{F}}^v_{h_{\mathrm{tot}}}]^{\mathrm{f}} \\ -\nabla_v \cdot [\rho \boldsymbol{\mathcal{F}}^v_{q_t}]^{\mathrm{f}} \\ -\nabla_v \cdot [\rho \boldsymbol{\mathcal{F}}^v_{q^\sigma_\mu}]^{\mathrm{f}} \\ -\nabla_v \cdot [\rho \boldsymbol{\mathcal{F}}^v_{\chi}]^{\mathrm{f}} \\ -\left[\nabla_v \cdot \left(\rho \boldsymbol{\mathcal{T}}^v_h\right)^{\mathrm{f}}\right]/\rho^c \\ -\left[\nabla_v \cdot \left(\rho \boldsymbol{\mathcal{T}}^v_v\right)^{\mathrm{c}}\right]/\rho^f + \left(\boldsymbol{S}_{\boldsymbol{u},v}\right)^{\mathrm{f}} \end{pmatrix}. \tag{G3}$$

The Jacobian of the i-th implicit Runge-Kutta stage is given by

$$\boldsymbol{W}_i(\boldsymbol{Y}) = \boldsymbol{I} - a^I_{ii}\delta t \frac{\partial}{\partial \boldsymbol{Y}} \boldsymbol{T}^I(\boldsymbol{Y},t), \tag{G4}$$

which we compute using a series of approximations that accelerate the solution of linear systems involving the Jacobian.

To approximate the advective flux contribution to the Jacobian, we assume that perturbations of grid-mean advection due to velocity components are much larger than those due to scalars (which is the case, e.g., when mean vertical velocities are negligible):

$$\frac{\partial \nabla_v \cdot (\rho\psi \boldsymbol{u}^v)^{\mathrm{f}}}{\partial (\boldsymbol{u}_h)^{\mathrm{c}}} \delta(\boldsymbol{u}_h)^{\mathrm{c}} \gg \frac{\partial \nabla_v \cdot (\rho\psi \boldsymbol{u}^v)^{\mathrm{f}}}{\partial (\rho\psi)^{\mathrm{c}}} \delta(\rho\psi)^{\mathrm{c}}, \tag{G5a}$$

$$\frac{\partial \nabla_v \cdot (\rho\psi \boldsymbol{u}^v)^{\mathrm{f}}}{\partial (\boldsymbol{u}_v)^{\mathrm{f}}} \delta(\boldsymbol{u}_v)^{\mathrm{f}} \gg \frac{\partial \nabla_v \cdot (\rho\psi \boldsymbol{u}^v)^{\mathrm{f}}}{\partial (\rho\psi)^{\mathrm{c}}} \delta(\rho\psi)^{\mathrm{c}}. \tag{G5b}$$

We assume the converse for the sedimentation fluxes, whose dependence on grid-mean velocities we neglect:

$$\frac{\partial \nabla_v \cdot \left(\rho\psi w_\psi \hat{\boldsymbol{k}}^v\right)^{\mathrm{f}}}{\partial (\rho\psi)^{\mathrm{c}}} \delta(\rho\psi)^{\mathrm{c}} \gg \frac{\partial \nabla_v \cdot \left(\rho\psi w_\psi \hat{\boldsymbol{k}}^v\right)^{\mathrm{f}}}{\partial (\boldsymbol{u}_h)^{\mathrm{c}}} \delta(\boldsymbol{u}_h)^{\mathrm{c}}, \tag{G6a}$$

$$\frac{\partial \nabla_v \cdot \left(\rho\psi w_\psi \hat{\boldsymbol{k}}^v\right)^{\mathrm{f}}}{\partial (\rho\psi)^{\mathrm{c}}} \delta(\rho\psi)^{\mathrm{c}} \gg \frac{\partial \nabla_v \cdot \left(\rho\psi w_\psi \hat{\boldsymbol{k}}^v\right)^{\mathrm{f}}}{\partial (\boldsymbol{u}_v)^{\mathrm{f}}} \delta(\boldsymbol{u}_v)^{\mathrm{f}}. \tag{G6b}$$

We also assume that perturbations due to density-weighted scalars are much larger than those due to density itself (i.e., scalar perturbations satisfy $\delta(\rho\psi) \gg \psi\,\delta\rho$, and advective velocities do not strongly depend on density),

$$\frac{\partial \nabla_v \cdot (\rho\psi \boldsymbol{u}^v)^{\mathrm{f}}}{\partial (\rho\psi)^{\mathrm{c}}} \delta(\rho\psi)^{\mathrm{c}} \gg \frac{\partial \nabla_v \cdot (\rho\psi \boldsymbol{u}^v)^{\mathrm{f}}}{\partial \rho^{\mathrm{c}}} \delta\rho^{\mathrm{c}}, \tag{G7a}$$

$$\frac{\partial \nabla_v \cdot \left(\rho\psi w_\psi \hat{\boldsymbol{k}}^v\right)^{\mathrm{f}}}{\partial (\rho\psi)^{\mathrm{c}}} \delta(\rho\psi)^{\mathrm{c}} \gg \frac{\partial \nabla_v \cdot \left(\rho\psi w_\psi \hat{\boldsymbol{k}}^v\right)^{\mathrm{f}}}{\partial \rho^{\mathrm{c}}} \delta\rho^{\mathrm{c}}, \tag{G7b}$$

and, since $q_t \ll 1$, we neglect the effect of falling/sedimenting condensate on density relative to grid-mean advection,

$$\frac{\partial \nabla_v \cdot (\rho \boldsymbol{u}^v)^{\mathrm{f}}}{\partial (\boldsymbol{u}_h)^{\mathrm{c}}} \delta(\boldsymbol{u}_h)^{\mathrm{c}} \gg \frac{\partial \nabla_v \cdot \left(\rho W_{q_t} \hat{\boldsymbol{k}}^v\right)^{\mathrm{f}}}{\partial (\rho q_t)^{\mathrm{c}}} \delta(\rho q_t)^{\mathrm{c}}, \tag{G8a}$$

$$\frac{\partial \nabla_v \cdot (\rho \boldsymbol{u}^v)^{\mathrm{f}}}{\partial (\boldsymbol{u}_v)^{\mathrm{f}}} \delta(\boldsymbol{u}_v)^{\mathrm{f}} \gg \frac{\partial \nabla_v \cdot \left(\rho W_{q_t} \hat{\boldsymbol{k}}^v\right)^{\mathrm{f}}}{\partial (\rho q_t)^{\mathrm{c}}} \delta(\rho q_t)^{\mathrm{c}}. \tag{G8b}$$

In addition, we assume that grid-mean velocities have a much larger effect on the grid-mean advection of enthalpy than on its sedimentation,

$$\frac{\partial \nabla_v \cdot (\rho h_{\mathrm{tot}} \boldsymbol{u}^v)^{\mathrm{f}}}{\partial (\boldsymbol{u}_h)^{\mathrm{c}}} \gg \frac{\partial \nabla_v \cdot \left(\rho W_h \hat{\boldsymbol{k}}^v\right)^{\mathrm{f}}}{\partial (\boldsymbol{u}_h)^{\mathrm{c}}}, \tag{G9a}$$

$$\frac{\partial \nabla_v \cdot (\rho h_{\mathrm{tot}} \boldsymbol{u}^v)^{\mathrm{f}}}{\partial (\boldsymbol{u}_v)^{\mathrm{f}}} \gg \frac{\partial \nabla_v \cdot \left(\rho W_h \hat{\boldsymbol{k}}^v\right)^{\mathrm{f}}}{\partial (\boldsymbol{u}_v)^{\mathrm{f}}}, \tag{G9b}$$

and, since $\|\boldsymbol{u}_v\| \ll \|\boldsymbol{u}_h\|$, that the effects of vertical velocity on enthalpy and pressure are negligible compared to the effects of horizontal velocity:

$$\frac{\partial (h_{\mathrm{tot}})^{\mathrm{f}}}{\partial (\boldsymbol{u}_h)^{\mathrm{c}}} \delta(\boldsymbol{u}_h)^{\mathrm{c}} \gg \frac{\partial (h_{\mathrm{tot}})^{\mathrm{f}}}{\partial (\boldsymbol{u}_v)^{\mathrm{f}}} \delta(\boldsymbol{u}_v)^{\mathrm{f}}, \tag{G10a}$$

$$\frac{\partial p^{\mathrm{c}}}{\partial (\boldsymbol{u}_h)^{\mathrm{c}}} \delta(\boldsymbol{u}_h)^{\mathrm{c}} \gg \frac{\partial p^{\mathrm{c}}}{\partial (\boldsymbol{u}_v)^{\mathrm{f}}} \delta(\boldsymbol{u}_v)^{\mathrm{f}} \tag{G10b}$$

This leads to the derivative approximation

$$\frac{\partial}{\partial \boldsymbol{Y}} \boldsymbol{T}^I_{\mathrm{adv}}(\boldsymbol{Y}, t) \approx$$

$$\begin{pmatrix}
0 & 0 & 0 & \frac{\partial \nabla_v \cdot (\rho \boldsymbol{u}^v)^{\mathrm{f}}}{\partial (\boldsymbol{u}_h)^{\mathrm{c}}} & \frac{\partial \nabla_v \cdot (\rho \boldsymbol{u}^v)^{\mathrm{f}}}{\partial (\boldsymbol{u}_v)^{\mathrm{f}}} \\
0 & -\frac{\partial \nabla_v \cdot \left(\rho W_h \hat{\boldsymbol{k}}^v\right)^{\mathrm{f}}}{\partial (\rho e_{\mathrm{tot}})^{\mathrm{c}}} & -\frac{\partial \nabla_v \cdot \left(\rho W_h \hat{\boldsymbol{k}}^v\right)^{\mathrm{f}}}{\partial (\rho q)^{\mathrm{c}}} & \frac{\partial \nabla_v \cdot (\rho h_{\mathrm{tot}} \boldsymbol{u}^v)^{\mathrm{f}}}{\partial (\boldsymbol{u}_h)^{\mathrm{c}}} & \frac{\partial \nabla_v \cdot (\rho h_{\mathrm{tot}} \boldsymbol{u}^v)^{\mathrm{f}}}{\partial (\boldsymbol{u}_v)^{\mathrm{f}}} \\
0 & 0 & -\frac{\partial \nabla_v \cdot \left(\rho q w_q \hat{\boldsymbol{k}}^v\right)^{\mathrm{f}}}{\partial (\rho q)^{\mathrm{c}}} & \frac{\partial \nabla_v \cdot (\rho q \boldsymbol{u}^v)^{\mathrm{f}}}{\partial (\boldsymbol{u}_h)^{\mathrm{c}}} & \frac{\partial \nabla_v \cdot (\rho q \boldsymbol{u}^v)^{\mathrm{f}}}{\partial (\boldsymbol{u}_v)^{\mathrm{f}}} \\
0 & 0 & 0 & 0 & 0 \\
\frac{\partial (\nabla_v p^{\mathrm{c}})/\rho^{\mathrm{f}}}{\partial \rho^{\mathrm{c}}} & \frac{\partial (\nabla_v p^{\mathrm{c}})/\rho^{\mathrm{f}}}{\partial (\rho e_{\mathrm{tot}})^{\mathrm{c}}} & \frac{\partial (\nabla_v p^{\mathrm{c}})/\rho^{\mathrm{f}}}{\partial (\rho q)^{\mathrm{c}}} & \frac{\partial (\nabla_v p^{\mathrm{c}})/\rho^{\mathrm{f}}}{\partial (\boldsymbol{u}_h)^{\mathrm{c}}} & 0
\end{pmatrix}, \tag{G11}$$

where q stands for the scalars q_t, q^σ_μ, and χ. We neglect the derivatives of h_{tot} and p with respect to q^σ_μ, as they are much smaller than the derivatives with respect to q_t, and the derivatives with respect to any passive tracer χ are always 0 as well.

To approximate the diffusive contribution, we assume that diffusive fluxes of scalars vary primarily because of changes in the scalars themselves (ignoring, e.g., the dependencies of diffusivities on the scalars),

$$\boldsymbol{\mathcal{F}}_\psi \sim \Delta \psi, \tag{G12}$$

and likewise for sources and diffusive fluxes of velocity components,

$$\boldsymbol{S}_{\boldsymbol{u},v} \sim \boldsymbol{u}_v, \tag{G13a}$$

$$\boldsymbol{\mathcal{T}}^v_h \sim \Delta \boldsymbol{u}_h, \tag{G13b}$$

$$\boldsymbol{\mathcal{T}}^v_v \sim \Delta \boldsymbol{u}_v. \tag{G13c}$$

We also assume that perturbations of scalar diffusion due to density are negligible,

$$\frac{\partial \nabla_v \cdot (\rho \boldsymbol{\mathcal{F}}^v_\psi)^{\mathrm{f}}}{\partial (\rho\psi)^{\mathrm{c}}} \delta(\rho\psi)^{\mathrm{c}} \gg \frac{\partial \nabla_v \cdot (\rho \boldsymbol{\mathcal{F}}^v_\psi)^{\mathrm{f}}}{\partial \rho^{\mathrm{c}}} \delta\rho^{\mathrm{c}}, \tag{G14}$$

and likewise for perturbations of velocity diffusion,

$$\frac{\partial \left[\nabla_v \cdot (\rho \boldsymbol{\mathcal{T}}^v_h)^{\mathrm{f}}\right]/\rho^{\mathrm{c}}}{\partial (\boldsymbol{u}_h)^{\mathrm{c}}} \delta(\boldsymbol{u}_h)^{\mathrm{c}} \gg \frac{\partial \left[\nabla_v \cdot (\rho \boldsymbol{\mathcal{T}}^v_h)^{\mathrm{f}}\right]/\rho^{\mathrm{c}}}{\partial \rho^{\mathrm{c}}} \delta\rho^{\mathrm{c}}, \tag{G15a}$$

$$\frac{\partial \left[\nabla_v \cdot (\rho \boldsymbol{\mathcal{T}}^v_v)^{\mathrm{f}}\right]/\rho^{\mathrm{c}}}{\partial (\boldsymbol{u}_v)^{\mathrm{c}}} \delta(\boldsymbol{u}_v)^{\mathrm{c}} \gg \frac{\partial \left[\nabla_v \cdot (\rho \boldsymbol{\mathcal{T}}^v_v)^{\mathrm{f}}\right]/\rho^{\mathrm{c}}}{\partial \rho^{\mathrm{c}}} \delta\rho^{\mathrm{c}}, \tag{G15b}$$

so that the diffusive derivative is given by

$$\frac{\partial}{\partial \boldsymbol{Y}} \boldsymbol{T}^I_{\mathrm{dif}}(\boldsymbol{Y}, t) \approx$$

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & \frac{\partial \nabla_v \cdot (\rho \boldsymbol{\mathcal{F}}^v_{h_{\mathrm{tot}}})^{\mathrm{f}}}{\partial (\rho e_{\mathrm{tot}})^{\mathrm{c}}} & \frac{\partial \nabla_v \cdot (\rho \boldsymbol{\mathcal{F}}^v_{h_{\mathrm{tot}}})^{\mathrm{f}}}{\partial (\rho q)^{\mathrm{c}}} & \frac{\partial \nabla_v \cdot (\rho \boldsymbol{\mathcal{F}}^v_{h_{\mathrm{tot}}})^{\mathrm{f}}}{\partial (\boldsymbol{u}_h)^{\mathrm{c}}} & 0 \\
0 & 0 & \frac{\partial \nabla_v \cdot (\rho \boldsymbol{\mathcal{F}}^v_q)^{\mathrm{f}}}{\partial (\rho q)^{\mathrm{c}}} & 0 & 0 \\
0 & 0 & 0 & \frac{\partial \left[\nabla_v \cdot (\rho \boldsymbol{\mathcal{T}}^v_h)^{\mathrm{f}}\right]/\rho^{\mathrm{c}}}{\partial (\boldsymbol{u}_h)^{\mathrm{c}}} & 0 \\
0 & 0 & 0 & 0 & \frac{\partial \left[\nabla_v \cdot (\rho \boldsymbol{\mathcal{T}}^v_v)^{\mathrm{c}}\right]/\rho^{\mathrm{f}}}{\partial (\boldsymbol{u}_v)^{\mathrm{f}}} - \frac{\partial (\boldsymbol{S}_{\boldsymbol{u},v})^{\mathrm{f}}}{\partial (\boldsymbol{u}_v)^{\mathrm{f}}}
\end{pmatrix}. \tag{G16}$$

With these approximations, $\boldsymbol{W}_i$ can be split into four blocks that have the sparsity structure

$$\boldsymbol{W}_i = \begin{pmatrix} \boldsymbol{W}^{11}_i & \boldsymbol{W}^{12}_i \\ \boldsymbol{W}^{21}_i & \boldsymbol{W}^{22}_i \end{pmatrix} \sim \begin{pmatrix} \begin{bmatrix} \boldsymbol{I} & \star & \star \\ \star & \mathbb{T} & \mathbb{T} \\ \star & \star & \mathbb{T} \end{bmatrix} & \begin{bmatrix} \mathbb{T} & \mathbb{B} \\ \mathbb{T} & \mathbb{B} \\ \mathbb{T} & \mathbb{B} \end{bmatrix} \\ \begin{bmatrix} \star & \star & \star \\ \mathbb{B} & \mathbb{B} & \mathbb{B} \end{bmatrix} & \begin{bmatrix} \mathbb{T} & \star \\ \mathbb{B} & \mathbb{T} \end{bmatrix} \end{pmatrix}, \tag{G17}$$

where $\mathbb{B}$ denotes a bidiagonal matrix and $\mathbb{T}$ denotes a tridiagonal matrix. The diagonal blocks $\boldsymbol{W}_i^{11}$ and $\boldsymbol{W}_i^{22}$ have a damping effect on $\boldsymbol{Y}$, while the off-diagonal blocks $\boldsymbol{W}_i^{12}$ and $\boldsymbol{W}_i^{21}$ introduce an oscillatory coupling between scalars and velocity, representing the effects of vertically propagating sound and gravity waves.

To avoid computing the dense inverse $(\boldsymbol{W}_i)^{-1}$ in each iteration of Newton's method, we evaluate the action $(\boldsymbol{W}_i)^{-1}\boldsymbol{R}_i$ using a Schur complement solver. With the full matrix equation expressed in block form as

$$\begin{pmatrix} \boldsymbol{W}_i^{11} & \boldsymbol{W}_i^{12} \\ \boldsymbol{W}_i^{21} & \boldsymbol{W}_i^{22} \end{pmatrix} \begin{pmatrix} \delta\boldsymbol{U}_i^1 \\ \delta\boldsymbol{U}_i^2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}_i^1 \\ \boldsymbol{R}_i^2 \end{pmatrix}, \tag{G18}$$

we first solve the simplified equation

$$\left(\boldsymbol{W}_i/\boldsymbol{W}_i^{11}\right)\delta\boldsymbol{U}_i^2 = \hat{\boldsymbol{R}}_i^2, \tag{G19}$$

where the Schur complement of $\boldsymbol{W}_i^{11}$ in $\boldsymbol{W}_i$ is

$$\boldsymbol{W}_i/\boldsymbol{W}_i^{11} = \boldsymbol{W}_i^{22} - \boldsymbol{W}_i^{21}\left(\boldsymbol{W}_i^{11}\right)^{-1}\boldsymbol{W}_i^{12}, \tag{G20}$$

and the right-hand side is

$$\hat{\boldsymbol{R}}_i^2 = \boldsymbol{W}_i^{21}\left(\boldsymbol{W}_i^{11}\right)^{-1}\boldsymbol{R}_i^1 - \boldsymbol{R}_i^2. \tag{G21}$$

After solving for $\delta\boldsymbol{U}_i^2$, we compute $\delta\boldsymbol{U}_i^1$ as

$$\delta\boldsymbol{U}_i^1 = \left(\boldsymbol{W}_i^{11}\right)^{-1}\left(\boldsymbol{R}_i^1 - \boldsymbol{W}_i^{12}\delta\boldsymbol{U}_i^2\right). \tag{G22}$$

When only sound and gravity waves are treated implicitly, $\boldsymbol{W}_i^{11}$ simplifies to the identity matrix, and its Schur complement has a block lower-triangular sparsity structure. This means that $\left(\boldsymbol{W}_i^{11}\right)^{-1}\boldsymbol{v} = \boldsymbol{v}$ for any vector $\boldsymbol{v}$, while $\left(\boldsymbol{W}_i/\boldsymbol{W}_i^{11}\right)^{-1}\boldsymbol{v}$ can be computed using an efficient block triangular matrix solver (with each block along the diagonal inverted using a tridiagonal matrix solver). To achieve comparable computational efficiency when treating diffusion, damping, and/or sedimentation implicitly, in which case the Schur complement is not guaranteed to be block lower-triangular, we use a stationary iterative solver that approximates the Schur complement as

$$\boldsymbol{W}_i/\boldsymbol{W}_i^{11} \approx \boldsymbol{S}_i = \boldsymbol{W}_i^{22} - \boldsymbol{W}_i^{21}\left[\mathrm{diag}\left(\boldsymbol{W}_i^{11}\right)\right]^{-1}\boldsymbol{W}_i^{12}, \tag{G23}$$

where $\mathrm{diag}\left(\boldsymbol{W}_i^{11}\right)$ is the Jacobi preconditioner of $\boldsymbol{W}_i^{11}$. Starting with the initial perturbation $\left(\delta\boldsymbol{U}_i^2\right)^{(0)} = 0$, the iterative solver updates

$$\left(\delta\boldsymbol{U}_i^2\right)^{(i)} = \left(\delta\boldsymbol{U}_i^2\right)^{(i-1)} + \left(\boldsymbol{S}_i\right)^{-1}\left[\hat{\boldsymbol{R}}_i^2 - \left(\boldsymbol{W}_i/\boldsymbol{W}_i^{11}\right)\left(\delta\boldsymbol{U}_i^2\right)^{(i-1)}\right]. \tag{G24}$$

Since $\boldsymbol{W}_i^{11}$ is block upper-triangular and $\boldsymbol{S}_i$ is block lower-triangular, we can compute $\left(\boldsymbol{W}_i^{11}\right)^{-1}\boldsymbol{v}$ and $\left(\boldsymbol{S}_i\right)^{-1}\boldsymbol{v}$ using efficient block matrix solvers. Two iterations of the stationary iterative solver are typically sufficient when nested within iterations of Newton's method in the ARS343 timestepping scheme.

Conflict of Interest

The authors declare no conflicts of interest relevant to this study.

Data Availability Statement

The code implementing the CliMA dynamical core is publicly available (Climate Modeling Alliance, 2025a). The benchmark tests are part of a broader test suite for the CliMA atmosphere model and are likewise publicly available (Climate Modeling Alliance, 2025b).

Acknowledgments

This research was supported by Schmidt Sciences, LLC, the U.S. National Science Foundation (Grant AGS-1835860), Office of Naval Research (Grant N00014-23-1-2654), and the Heising-Simons Foundation. We acknowledge high-performance computing support from Caltech's Resnick High Performance Computing Center, the Derecho system (Computational and Information Systems Laboratory, 2023) provided by the National Center for Atmospheric Research (NCAR), sponsored by the National Science Foundation, and Google Cloud Platform (GCP). We thank David Romps for helpful discussions on the formulation of the equations of motion; Carlos Boneti for his help in configuring the GCP virtual machine; Jake Bolewski for his contributions to the early phases of the code development; and the reviewers for insightful comments that led to considerable improvements of the paper.

References

Abbott, T. H., & O'Gorman, P. A. (2024). Impact of precipitation mass sinks on midlatitude storms over a wide range of climates. *Weather and Climate Dynamics*, *5*(1), 17–41. https://doi.org/10.5194/wcd-5-17-2024

Ambaum, M. H. P. (2020). Accurate, simple equation for saturated vapour pressure over water and ice. *Quarterly Journal of the Royal Meteorological Society*, *146*(733), 4252–5258. https://doi.org/10.1002/qj.3899

Ascher, U. M., Ruuth, S. J., & Spiteri, R. J. (1997). Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, *25*(2–3), 151–167. https://doi.org/10.1016/s0168-9274(97)00056-1

Azimi, S., Jaruga, A., de Jong, E. K., Arabas, S., & Schneider, T. (2024). Training warm-rain bulk microphysics schemes using super-droplet simulations. *Journal of Advances in Modeling Earth Systems*, *16*(7), e2023MS004028. https://doi.org/10.1029/2023MS004028

Baldauf, M. (2008). A linear solution for flow over mountains and its comparison with the COSMO model. *COSMO Newsletter*, *8*, 19–24.

Bonan, G. (2019). *Climate change and terrestrial ecosystem modeling*. Cambridge Univ. Press. https://doi.org/10.1017/9781107339217

Bott, A. (2008). Theoretical considerations on the mass and energy consistent treatment of precipitation in cloudy atmospheres. *Atmospheric Research*, *89*(3), 252–269. https://doi.org/10.1016/j.atmosres.2008.02.010

Bowen, P., & Thuburn, J. (2022a). Consistent and flexible thermodynamics in atmospheric models using internal energy as a thermodynamic potential. Part I: Equilibrium regime. *Quarterly Journal of the Royal Meteorological Society*, *148*(749), 3730–3755. https://doi.org/10.1002/qj.4385

Bowen, P., & Thuburn, J. (2022b). Consistent and flexible thermodynamics in atmospheric models using internal energy as a thermodynamic potential. Part II: Non-equilibrium regime. *Quarterly Journal of the Royal Meteorological Society*, *148*(749), 3540–3565. https://doi.org/10.1002/qj.4373

Červený, J. (2024). Generalized Hilbert space-filling curve. Retrieved from https://github.com/jakubcerveny/gilbert.

Charbonneau, A., Deck, K., & Schneider, T. (2025). A physics-constrained neural differential equation framework for data-driven snowpack simulation. *Artificial Intelligence for the Earth Systems*, *4*(3), e240040. https://doi.org/10.1175/AIES-D-24-0040.1

Christopoulos, C., Lopez-Gomez, I., Beucler, T., Cohen, Y., Kawczynski, C., Dunbar, O. R. A., & Schneider, T. (2024). Online learning of entrainment closures in a hybrid machine learning parameterization. *Journal of Advances in Modeling Earth Systems*, *16*, e2024MS004485. https://doi.org/10.1029/2024MS004485

Climate Modeling Alliance. (2025a). ClimaCore.jl [Software]. https://doi.org/10.5281/zenodo.5554759

Climate Modeling Alliance. (2025b). ClimaAtmos.jl [Software]. https://doi.org/10.5281/zenodo.17237816

Cohen, Y., Lopez-Gomez, I., Jaruga, A., He, J., Kaul, C. M., & Schneider, T. (2020). Unified entrainment and detrainment closures for extended eddy-diffusivity mass-flux schemes. *Journal of Advances in Modeling Earth Systems*, *12*(9), e2020MS002162. https://doi.org/10.1029/2020MS002162

Colavolpe, C., Voitus, F., & Bénard, P. (2017). RK-IMEX HEVI schemes for fully compressible atmospheric models with advection: Analyses and numerical testing. *Quarterly Journal of the Royal Meteorological Society*, *143*, 1336–1350. https://doi.org/10.1002/qj.3008

Computational and Information Systems Laboratory. (2023). Derecho: HPE Cray EX system [Hardware]. *University Community Computing*. https://doi.org/10.5065/qx9a-pg09

Dahm, J., Davis, E., Deconinck, F., Elbert, O., George, R., McGibbon, J., et al. (2023). Pace v0. 2: A Python-based performance-portable atmospheric model. *Geoscientific Model Development*, *16*(9), 2719–2736. https://doi.org/10.5194/gmd-16-2719-2023

Deck, K., Braghiere, R. K., Renchon, A. A., Sloan, J., Bozzola, G., Speer, E., et al. (2026). ClimaLand: A land surface model designed to enable data-driven parameterizations. *Journal of Advances in Modeling Earth Systems*, *18*(1), e2025MS005118. https://doi.org/10.1029/2025MS005118

Dennis, J. M., Edwards, J., Evans, K. J., Guba, O., Lauritzen, P. H., Mirin, A. A., et al. (2012). CAM-SE: A scalable spectral element dynamical core for the Community Atmosphere Model. *High Perf. Comp. Appl.*, *26*(1), 74–89. https://doi.org/10.1177/1094342011428142

Deville, M. O., Fischer, P. F., & Mund, E. H. (2002). *High-order methods for incompressible fluid flow*. Cambridge University Press.

Doms, G., & Baldauf, M. (2018). *A description of the nonhydrostatic regional COSMO-model (Tech. Rep.)*. Consortium for Small-Scale Modeling.

Donahue, A. S., Caldwell, P. M., Bertagna, L., Beydoun, H., Bogenschutz, P. A., Bradley, A., et al. (2024). To exascale and beyond—The simple cloud-resolving E3SM atmosphere model (SCREAM), a performance portable global atmosphere model for cloud-resolving scales. *Journal of Advances in Modeling Earth Systems*, *16*(7), e2024MS004314. https://doi.org/10.1029/2024MS004314

Duarte, M., Almgren, A. S., Balakrishnan, K., Bell, J. B., & Romps, D. M. (2014). A numerical study of methods for moist atmospheric flows: Compressible equations. *Monthly Weather Review*, *142*(11), 4269–4283. https://doi.org/10.1175/mwr-d-13-00368.1

Dubos, T. (2024). On the thermodynamic invariance of fine-grain and coarse-grain fluid models. *Quarterly Journal of the Royal Meteorological Society*, *150*(764), 4567–4579. https://doi.org/10.1002/qj.4830

Dunbar, O. R. A., Duncan, A. B., Stuart, A. M., & Wolfram, M.-T. (2022). Ensemble inference methods for models with noisy and expensive likelihoods. *SIAM Journal on Applied Dynamical Systems*, *21*(2), 1539–1572. https://doi.org/10.1137/21m1410853

Durran, D. R., & Klemp, J. B. (1983). A compressible model for the simulation of moist mountain waves. *Monthly Weather Review*, *111*(12), 2341–2361. https://doi.org/10.1175/1520-0493(1983)111<2341:acmfts>2.0.co;2

Ern, M., Trinh, Q. T., Preusse, P., Gille, J. C., Mlynczak, M. G., Russell, J. M., III., & Riese, M. (2018). GRACILE: A comprehensive climatology of atmospheric gravity wave parameters based on satellite limb soundings. *Earth System Science Data*, *10*(2), 857–892. https://doi.org/10.5194/essd-10-857-2018

Foken, T. (2006). 50 years of the Monin-Obukhov similarity theory. *Bound.-Layer Meteor.*, *119*(3), 431–447. https://doi.org/10.1007/s10546-006-9048-6

Fritts, D. C., & Alexander, M. J. (2003). Gravity wave dynamics and effects in the middle atmosphere. *Reviews of Geophysics*, *41*(1003). https://doi.org/10.1029/2001RG000106

Gal-Chen, T., & Somerville, R. C. (1975). On the use of a coordinate transformation for the solution of the navier-stokes equation. *Journal of Computational Physics*, *17*(2), 209–228. https://doi.org/10.1016/0021-9991(75)90037-6

Gardner, D. J., Guerra, J. E., Hamon, F. P., Reynolds, D. R., Ullrich, P. A., & Woodward, C. S. (2018). Implicit-explicit (IMEX) Runge-Kutta methods for non-hydrostatic atmospheric models. *Geoscientific Model Development*, *11*(4), 1497–1515. https://doi.org/10.5194/gmd-11-1497-2018

Golaz, J.-C., Van Roekel, L. P., Zheng, X., Roberts, A. F., Wolfe, J. D., Lin, W., et al. (2022). The DOE E3SM model version 2: Overview of the physical model and initial model evaluation. *Journal of Advances in Modeling Earth Systems*, *14*(12), e2022MS003156. https://doi.org/10.1029/2022MS003156

Grabowski, W. W. (1998). Toward cloud resolving modeling of large-scale tropical circulations: A simple cloud microphysics parameterization. *Journal of the Atmospheric Sciences*, *55*(21), 3283–3298. https://doi.org/10.1175/1520-0469(1998)055<3283:tcrmol>2.0.co;2

Guba, O., Taylor, M. A., Ullrich, P. A., Overfelt, J. R., & Levy, M. N. (2014). The spectral element method (SEM) on variable-resolution grids: Evaluating grid sensitivity and resolution-aware numerical viscosity. *Geoscientific Model Development*, *7*(6), 2803–2816. https://doi.org/10.5194/gmd-7-2803-2014

Herrington, A. R., Gettelman, A., Lauritzen, P. H., & Taylor, M. A. (2022). Impact of grids and dynamical cores in CESM2.2 on the surface mass balance of the Greenland ice sheet. *Journal of Advances in Modeling Earth Systems*, *14*, e2022MS003192. https://doi.org/10.1029/2022MS003192

Hu, A. Z., & Igel, A. L. (2024). Bulk microphysics schemes may perform better with a unified cloud-rain category. *Journal of Advances in Modeling Earth Systems*, *16*(7), e2023MS004068. https://doi.org/10.1029/2023MS004068

Huang, D. Z., Huang, J., Reich, S., & Stuart, A. M. (2022a). Efficient derivative-free Bayesian inference for large-scale inverse problems. *Inverse Problems*, *38*(12), 125006. https://doi.org/10.1088/1361-6420/ac99fa

Huang, D. Z., Schneider, T., & Stuart, A. M. (2022b). Iterated Kalman methodology for inverse problems. *Journal of Computational Physics*, *463*, 111262. https://doi.org/10.1016/j.jcp.2022.111262

Hughes, O. K., & Jablonowski, C. (2023). A mountain-induced moist baroclinic wave test case for the dynamical cores of atmospheric general circulation models. *Geoscientific Model Development*, *16*(22), 6805–6831. https://doi.org/10.5194/gmd-16-6805-2023

Jablonowski, C., & Williamson, D. L. (2011). The pros and cons of diffusion, filters and fixers in atmospheric general circulation models. In P. H. Lauritzen, C. Jablonowski, M. A. Taylor, & R. D. Nair (Eds.), *Numerical Techniques for Global Atmospheric Models* (Vol. 80, pp. 381–493). Springer-Verlag. https://doi.org/10.1007/978-3-642-11640-7_13

Kajishima, T., & Taira, K. (2017). *Computational fluid dynamics*. Springer.

Karniadakis, G. E., & Sherwin, S. J. (2005). *Spectral/hp element methods for computational fluid dynamics* (2nd ed.). Oxford University Press.

Kaul, C. M., Teixeira, J., & Suzuki, K. (2015). Sensitivities in large-eddy simulations of mixed-phase Arctic stratocumulus clouds using a simple microphysics approach. *Monthly Weather Review*, *143*(11), 4393–4421. https://doi.org/10.1175/MWR-D-14-00319.1

Kessler, E. (1969). On the distribution and continuity of water substance in atmospheric circulations. *American Meteorological Society*, *10*, 1–84. https://doi.org/10.1007/978-1-935704-36-2_1

Klemp, J., Dudhia, J., & Hassiotis, A. (2008). An upper gravity-wave absorbing layer for NWP applications. *Monthly Weather Review*, *136*(10), 3987–4004. https://doi.org/10.1175/2008MWR2596.1

Klemp, J. B. (2011). A terrain-following coordinate with smoothed coordinate surfaces. *Monthly Weather Review*, *139*(7), 2163–2169. https://doi.org/10.1175/MWR-D-10-05046.1

Klemp, J. B., & Lilly, D. K. (1978). Numerical simulation of hydrostatic mountain waves. *Journal of the Atmospheric Sciences*, *35*(1), 78–107. https://doi.org/10.1175/1520-0469(1978)035<0078:nsohmw>2.0.co;2

Klemp, J. B., Skamarock, W. C., & Fuhrer, O. (2003). Numerical consistency of metric terms in terrain-following coordinates. *Monthly Weather Review*, *131*(7), 1229–1239. https://doi.org/10.1175/1520-0493(2003)131<1229:ncomti>2.0.co;2

Kovachki, N. B., & Stuart, A. M. (2019). Ensemble Kalman inversion: A derivative-free technique for machine learning tasks. *Inverse Problems*, *35*(9), 095005. https://doi.org/10.1088/1361-6420/ab1c3a

Lauritzen, P., Bacmeister, J., Callaghan, P., & Taylor, M. (2015). Ncar_topo (v1. 0): Ncar global model topography generation software for unstructured grids. *Geoscientific Model Development*, *8*(12), 3975–3986. https://doi.org/10.5194/gmd-8-3975-2015

Lauritzen, P. H., Kevlahan, N.-R., Toniazzo, T., Eldred, C., Dubos, T., Gassmann, A., et al. (2022). Reconciling and improving formulations for thermodynamics and conservation principles in Earth System Models (ESMs). *Journal of Advances in Modeling Earth Systems*, *14*(9), e2022MS003117. https://doi.org/10.1029/2022MS003117

Lauritzen, P. H., Nair, R. D., Herrington, A., Callaghan, P., Goldhaber, S., Dennis, J., et al. (2018). NCAR release of CAM-SE in CESM2.0: A reformulation of the spectral element dynamical core in dry-mass vertical coordinates with comprehensive treatment of condensates and energy. *Journal of Advances in Modeling Earth Systems*, *10*(7), 1537–1570. https://doi.org/10.1029/2017ms001257

Leer, B. V. (1977). Towards the ultimate conservative difference scheme III. Upstream-centered finite-difference schemes for ideal compressible flow. *Journal of Computational Physics*, 263–275.

Leuenberger, D., Koller, M., Fuhrer, O., & Schär, C. (2010). A generalization of the SLEVE vertical coordinate. *Monthly Weather Review*, *138*(9), 3683–3689. https://doi.org/10.1175/2010mwr3307.1

Lin, S.-J., Chao, W. C., Sud, Y. C., & Walker, G. K. (1994). A class of the van leer-type transport schemes and its application to the moisture transport in a general circulation model. *Monthly Weather Review*, *122*, 1575–1593. https://doi.org/10.1175/1520-0493(1994)122⟨1575:ACOTVL⟩2.0.CO;2

Lopez-Gomez, I., Christopoulos, C., Ervik, H. L., Dunbar, O. R. A., Cohen, Y., & Schneider, T. (2022). Training physics-based machine-learning parameterizations with gradient-free ensemble Kalman methods. *Journal of Advances in Modeling Earth Systems*, *14*(8), e2022MS003105. https://doi.org/10.1029/2022MS003105

Lopez-Gomez, I., Cohen, Y., He, J., Jaruga, A., & Schneider, T. (2020). A generalized mixing length closure for eddy-diffusivity mass-flux schemes of turbulence and convection. *Journal of Advances in Modeling Earth Systems*, *12*(11), e2020MS002161. https://doi.org/10.1029/2020MS002161

Midgley, J., & Liemohn, H. (1966). Gravity waves in a realistic atmosphere. *Journal of Geophysical Research*, *71*(15), 3729–3748. https://doi.org/10.1029/jz071i015p03729

Morrison, H., & Milbrandt, J. A. (2015). Parameterization of cloud microphysics based on the prediction of bulk ice particle properties. Part I: Scheme description and idealized tests. *Journal of the Atmospheric Sciences*, *72*(1), 287–311. https://doi.org/10.1175/JAS-D-14-0065.1

Peixoto, J. P., Oort, A. H., de Almeida, M., & Tomé, A. (1991). Entropy budget of the atmosphere. *Journal of Geophysical Research*, *96*(D6), 10981–10988. https://doi.org/10.1029/91jd00721

Pressel, K. G., Kaul, C. M., Schneider, T., Tan, Z., & Mishra, S. (2015). Large-eddy simulation in an anelastic framework with closed water and entropy balances. *Journal of Advances in Modeling Earth Systems*, *7*(3), 1425–1456. https://doi.org/10.1002/2015MS000496

Ringler, T. D., Thuburn, J., B.Klemp, J., & C.Skamarock, W. (2010). A unified approach to energy conservation and potential vorticity dynamics for arbitrarily-structured C-grids. *Journal of Computational Physics*, *229*(9), 3065–3090. https://doi.org/10.1016/j.jcp.2009.12.007

Romps, D. M. (2008). The dry-entropy budget of a moist atmosphere. *Journal of the Atmospheric Sciences*, *65*(12), 3779–3799. https://doi.org/10.1175/2008JAS2679.1

Romps, D. M. (2021). The Rankine–Kirchhoff approximations for moist thermodynamics. *Quarterly Journal of the Royal Meteorological Society*, *147*(740), 3493–3497. https://doi.org/10.1002/qj.4154

Ronchi, C., Iacono, R., & Paolucci, P. (1996). The "cubed sphere": A new method for the solution of partial differential equations in spherical geometry. *Journal of Computational Physics*, *124*, 93–114. https://doi.org/10.1006/jcph.1996.004

Roy, P., Rauber, R. M., & Girolamo, L. D. (2023). A closer look at the evolution of supercooled cloud droplet temperature and lifetime in different environmental conditions with implications for ice nucleation in the evaporating regions of clouds. *Journal of the Atmospheric Sciences*, *80*, 2481–2501.

S2 Geometry Library. (2024). Earth cube. Retrieved from http://s2geometry.io/resources/earthcube

Sadourny, R. (1972). Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids. *Monthly Weather Review*, *100*(2), 136–144. https://doi.org/10.1175/1520-0493(1972)100<0136:cfaotp>2.3.co;2

Satoh, M. (2003). Conservative scheme for a compressible nonhydrostatic model with moist processes. *Monthly Weather Review*, *131*(6), 1033–1055. https://doi.org/10.1175/1520-0493(2003)131<1033:csfacn>2.0.co;2

Satoh, M., Matsuno, T., Tomita, H., Miura, H., Nasuno, T., & Iga, S. (2008). Nonhydrostatic icosahedral atmospheric model (NICAM) for global cloud resolving simulations. *Journal of Computational Physics*, *227*(7), 3486–3514. https://doi.org/10.1016/j.jcp.2007.02.006

Schär, C., Leuenberger, D., Fuhrer, O., Lüthi, D., & Girard, C. (2002). A new terrain-following vertical coordinate formulation for atmospheric prediction models. *Monthly Weather Review*, *130*(10), 2459–2480. https://doi.org/10.1175/1520-0493(2002)130<2459:antfvc>2.0.co;2

Schneider, T., Behera, S., Boccaletti, G., Deser, C., Emanuel, K., Ferrari, R., et al. (2023). Harnessing AI and computing to advance climate modeling and prediction. *Nature Climate Change*, *13*(9), 887–889. https://doi.org/10.1038/s41558-023-01769-3

Schneider, T., Lan, S., Stuart, A., & Teixeira, J. (2017). Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophysical Research Letters*, *44*(24), 12396–12417. https://doi.org/10.1002/2017GL076101

Shepherd, T. G., Semeniuk, K., & Koshyk, J. N. (1996). Sponge layer feedbacks in middle-atmosphere models. *Journal of Geophysical Research*, *101*(D18), 23447–23464. https://doi.org/10.1029/96JD01994

Simmons, A. J., & Burridge, D. M. (1981). An energy and angular-momentum conserving vertical finite-difference scheme and hybrid vertical coordinates. *Monthly Weather Review*, *109*(4), 758–766. https://doi.org/10.1175/1520-0493(1981)109<0758:aeaamc>2.0.co;2

Skamarock, W. C., Park, S.-H., Klemp, J. B., & Snyder, C. (2014). Atmospheric kinetic energy spectra from global high-resolution nonhydrostatic simulations. *Journal of the Atmospheric Sciences*, *71*(11), 4369–4381. https://doi.org/10.1175/JAS-D-14-0114.1

Soto, A., Mischna, M., Schneider, T., Lee, C., & Richardson, M. (2015). Martian atmospheric collapse: Idealized GCM studies. *Icarus*, *250*, 553–569. https://doi.org/10.1016/j.icarus.2014.11.028

Sridhar, A., Tissaoui, Y., Marras, S., Shen, Z., Kawczynski, C., Byrne, S., et al. (2022). Large-eddy simulations with ClimateMachine v0.2.0: A new open-source code for atmospheric simulations on GPUs and CPUs. *Geoscientific Model Development*, *15*, 6259–6284. https://doi.org/10.5194/gmd-15-6259-2022

Taylor, M., Caldwell, P. M., Bertagna, L., Clevenger, C., Donahue, A., Foucar, J., et al. (2023). The simple cloud-resolving E3SM atmosphere model running on the Frontier exascale system. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (pp. 1–11). https://doi.org/10.1145/3581784.3627044

Taylor, M. A., Guba, O., Steyer, A., Ullrich, P. A., Hall, D. M., & Eldred, C. (2020). An energy consistent discretization of the nonhydrostatic equations in primitive variables. *Journal of Advances in Modeling Earth Systems*, *12*(1), e2019MS001783. https://doi.org/10.1029/2019MS001783

Taylor, M. A., & Fournier, A. (2010). A compatible and conservative spectral element method on unstructured grids. *Journal of Computational Physics*, *229*(17), 5879–5895. https://doi.org/10.1016/j.jcp.2010.04.008

Thuburn, J. (2008). Some conservation issues for the dynamical cores of NWP and climate models. *Journal of Computational Physics*, *227*(7), 3715–3730. https://doi.org/10.1016/j.jcp.2006.08.016

Thuburn, J., & Woollings, T. J. (2005). Vertical discretizations for compressible Euler equation atmospheric models giving optimal representation of normal modes. *Journal of Computational Physics*, *203*(2), 386–404. https://doi.org/10.1016/j.jcp.2004.08.018

Tsuda, T. (2014). Characteristics of atmospheric gravity waves observed using the MU (Middle and Upper atmosphere) radar and GPS (Global Positioning System) radio occultation. *Proceedings of the Japan Academy, Series B*, *90*(1), 12–27. https://doi.org/10.2183/pjab.90.12

Ullrich, P. A. (2014). A global finite-element shallow-water model supporting continuous and discontinuous elements. *Geoscientific Model Development*, *7*(6), 3017–3035. https://doi.org/10.5194/gmd-7-3017-2014

Ullrich, P. A., Jablonowski, C., Reed, K. A., Zarzycki, C., Lauritzen, P. H., Nair, R. D., et al. (2016). *Dynamical core model intercomparison project (DCMIP2016) test case document (Tech. Rep.)*. University of Michigan. Retrieved from https://github.com/ClimateGlobalChange/DCMIP2016/blob/master/DCMIP2016-TestCaseDocument_v1.pdf

Ullrich, P. A., Melvin, T., Jablonowski, C., & Staniforth, A. (2014). A proposed baroclinic wave test case for deep-and shallow-atmosphere dynamical cores. *Quarterly Journal of the Royal Meteorological Society*, *140*(682), 1590–1602. https://doi.org/10.1002/qj.2241

Ullrich, P. A., Reynolds, D. R., Guerra, J. E., & Taylor, M. A. (2018). Impact and importance of hyperdiffusion on the spectral element method: A linear dispersion analysis. *Journal of Computational Physics*, *375*, 427–446. https://doi.org/10.1016/j.jcp.2018.06.035

Vinokur, M. (1974). Conservation equations of gasdynamics in curvilinear coordinate systems. *Journal of Computational Physics*, *14*(2), 105–125. https://doi.org/10.1016/0021-9991(74)90008-4

Weller, H., Lock, S.-J., & Wood, N. (2013). Runge–Kutta IMEX schemes for the horizontally explicit/vertically implicit (HEVI) solution of wave equations. *Journal of Computational Physics*, *252*, 365–381. https://doi.org/10.1016/j.jcp.2013.06.025

White, A. A., Hoskins, B. J., Roulstone, I., & Staniforth, A. (2005). Consistent approximate models of the global atmosphere: Shallow, deep, hydrostatic, quasi-hydrostatic and non-hydrostatic. *Quarterly Journal of the Royal Meteorological Society*, *131*(609), 2081–2107. https://doi.org/10.1256/qj.04.49

Zalesak, S. T. (1979). Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, *31*(3), 335–362. https://doi.org/10.1016/0021-9991(79)90051-2