https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

TerraFlow (version 1.0): a differentiable multi-backend nonhydrostatic atmospheric dynamical core on unstructured spherical meshes

Tiange Huang[1,2], Li Dong[2], Ziniu Xiao[2,1], Baoxiang Pan[2], and Gaopeng Lu[1]

[1]School of Earth and Space Sciences, University of Science and Technology of China, Hefei 230026, China
[2]State Key Laboratory of Earth System Numerical Modelling and Application, Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing 100029, China

Correspondence: Ziniu Xiao (xiaozn@lasg.iap.ac.cn)

Abstract. Atmospheric modeling increasingly requires dynamical cores that combine efficient simulation with gradient computation for sensitivity analysis and parameter estimation. To address these needs, we present TerraFlow (version 1.0), a differentiable, multi-backend nonhydrostatic atmospheric dynamical core that supports both JAX and PyTorch on unstructured spherical meshes. Built on a layer-averaged finite-volume formulation in a dry-mass vertical coordinate, TerraFlow uses a common array and state-update interface to execute the same numerical operators with either backend. This design brings automatic differentiation, just-in-time compilation, and graphics processing unit (GPU) acceleration across both backends together with distributed forward integration through the Message Passing Interface in a unified framework. Five idealized tests evaluate the model's numerical performance, demonstrating that it reproduces the principal features of passive-tracer transport, large-scale rotational flow, and nonhydrostatic gravity waves. To assess local derivative accuracy, we use finite-difference comparisons and Taylor remainder tests in short nonhydrostatic integrations. For both backends, directional derivatives computed by automatic differentiation agree with centered finite-difference estimates to relative errors below 10^{-4} for all tested output fields over suitable perturbation ranges, and the loss Taylor remainders exhibit second-order convergence. The resulting gradients support a successful inversion that recovers two prescribed initial potential-temperature perturbation amplitudes from noise-free synthetic observations in six updates. Computational efficiency is assessed separately through double-precision hydrostatic benchmarks, in which one NVIDIA RTX 4090 achieves forward-integration speedups of approximately 2.3–7.0 relative to corresponding runs on 10 central processing unit (CPU) nodes with 2560 cores in total, excluding compilation, model construction, and output. By bringing efficient forward simulation and verified local differentiation into a shared numerical implementation, TerraFlow provides a common framework for atmospheric sensitivity analysis and gradient-based parameter estimation across two computational backends.

1 Introduction

Atmospheric dynamical cores solve the equations governing resolved motion in weather and climate models (Bauer et al., 2015). At kilometer scales, nonhydrostatic formulations become increasingly important, while finer meshes and shorter time

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

steps raise computational demands (Skamarock et al., 2012; Zängl et al., 2015; Satoh et al., 2019; Schär et al., 2020). Efficient simulations therefore depend on numerical methods and software architectures that exploit parallelism across heterogeneous hardware, including central processing units (CPUs) and graphics processing units (GPUs) (Fuhrer et al., 2018; Bauer et al., 2021). Gradients with respect to initial conditions or model parameters also support sensitivity analysis, conventional four-dimensional variational data assimilation, and gradient-based parameter estimation (Errico, 1997; Bannister, 2017; Gelbrecht et al., 2023). These requirements motivate dynamical cores that support efficient forward integration and differentiation with respect to initial conditions and model parameters.

Quasi-uniform spherical meshes avoid the polar convergence of latitude–longitude grids, although accuracy also depends on staggering and discrete wave representation (Staniforth and Thuburn, 2012). Spherical centroidal Voronoi tessellations (SCVTs) support both quasi-uniform and locally refined meshes through a prescribed mesh-density function (Ringler et al., 2008). On such meshes, carefully designed C-grid discretizations can maintain stationary geostrophic modes in the linearized shallow-water equations (Thuburn et al., 2009). For the nonlinear shallow-water equations, this approach also supports energy conservation (Ringler et al., 2010). Building on this approach, Zhang et al. (2019) developed a layer-averaged nonhydrostatic framework as a prototype of the Global–Regional Integrated forecast SysTem (GRIST), combining horizontal C-grid staggering, a hybrid mass coordinate, and an implicit treatment of vertical fast waves. Its mass-based formulation supports closely related hydrostatic and nonhydrostatic solvers, while the implicit treatment relaxes the acoustic time-step restriction associated with fine vertical spacing. Zhang et al. (2020) extended it to moist dynamics in a dry-mass coordinate, with coupling among dynamics, tracer transport, and physical parameterizations.

Efficient implementation of these methods requires control over data movement and hardware-specific execution, as illustrated by near-global kilometer-scale simulations on thousands of GPUs (Fuhrer et al., 2018). JAX and PyTorch provide high-level array operations, automatic differentiation (AD), and compilation for accelerator execution (Bradbury et al., 2018; Paszke et al., 2019; Ansel et al., 2024). In atmospheric dynamics, the Python-based LMARSpy core combines GPU execution with a sharpness-preserving monotonicity limiter and a conservative vertical implicit solver (Zhang and Chen, 2025). Applying these approaches on unstructured meshes also requires efficient indirect neighbor access, local reconstruction, and flux accumulation.

AD applies the chain rule to program operations, yielding directional derivatives in forward mode and scalar-objective gradients in reverse mode (Giering and Kaminski, 1998; Baydin et al., 2018). Reverse mode is particularly useful when a scalar loss depends on many controls. AD can reduce the effort required to keep derivative calculations consistent with an evolving model (Moses et al., 2026), supporting parameter calibration and hybrid Earth system modeling (Gelbrecht et al., 2023). Rasp (2020) proposed coupled online learning with training targets from a concurrently running high-resolution model, demonstrating the method with Lorenz 96. NeuralGCM trains neural-network parameterizations end to end through multistep integrations of a differentiable atmospheric core, minimizing discrepancies between predictions and reanalysis data (Kochkov et al., 2024). The JAX Circulation Model (JCM) combines differentiable dynamics with modular physical parameterizations for sensitivity analysis and hybrid-model development (Davenport et al., 2026).

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

A differentiable atmospheric core must provide accurate forward simulations and verified derivatives at a practical computational cost. Derivative calculations must account for the full sequence of model operations, including flux reconstruction, implicit vertical solves, and state updates. Limiters and conditional branches can introduce nonsmoothness, making derivative checks necessary for the numerical configurations being evaluated. Computational cost is another concern: reverse-mode differentiation requires intermediate values from the forward integration, which can be expensive to retain in memory. Checkpointing reduces this memory demand by retaining selected states and recomputing intermediate values when needed (Griewank and Walther, 2000). DJ4Earth illustrates how automatic differentiation, compilation, and checkpointing can be combined to address these implementation challenges across several Earth system model components (Moses et al., 2026). These considerations motivate separate assessments of forward-simulation accuracy, derivative consistency, and computational efficiency.

Here we present TerraFlow (version 1.0), a differentiable, multi-backend nonhydrostatic atmospheric dynamical core on unstructured spherical meshes. Its numerical formulation builds on the layer-averaged nonhydrostatic framework of Zhang et al. (2019, 2020). TerraFlow supports both JAX and PyTorch backends, enabling automatic differentiation and efficient execution on CPUs and GPUs through just-in-time compilation. Together with support for distributed forward integration, these capabilities provide a unified environment for atmospheric simulation, sensitivity analysis, and gradient-based parameter estimation.

The remainder of this paper is organized as follows. Section 2 introduces the vertical coordinate and governing equations. Section 3 describes the numerical methods and software implementation. Section 4 evaluates the model using idealized dynamical tests. Section 5 presents derivative verification and a two-parameter inversion experiment. Section 6 assesses forward-integration performance on CPUs and GPUs. Finally, Sect. 7 summarizes the findings and outlines future work.

2 Model description

TerraFlow uses a finite-volume dynamical framework with unstructured spherical meshes in the horizontal and a hybrid vertical coordinate based on dry-air mass. The governing equations are formulated under the shallow-atmosphere approximation, with constant gravitational acceleration g and horizontal spherical geometry that is independent of height. This section first defines the vertical coordinate and then presents the governing equations. Appendix A summarizes the principal symbols used in the governing equations and their numerical implementation, together with their definitions and units.

2.1 Dry-mass hybrid coordinate

TerraFlow uses a hybrid terrain-following coordinate based on dry-air mass. Following Simmons and Burridge (1981) and Laprise (1992), the dry-mass variable π, expressed in pressure units, is written as

$$\pi(\eta) = \eta_A(\eta)p_0 + \eta_B(\eta)\pi_s, \qquad \eta = \eta_A + \eta_B, \tag{1}$$

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

where η_A and η_B are prescribed dimensionless coefficients, η is the generalized vertical coordinate, $p_0 = 10^5$ Pa is the reference pressure, and π_s is the surface value of π. The coefficients define surfaces that follow the terrain near the ground and approach surfaces of constant π aloft. At the model top, $\eta_B = 0$, so the top value π_t remains constant in time.

The relation between this coordinate and dry-air mass is specified by

$$\frac{\partial \pi}{\partial z} = -\rho_d g, \qquad \frac{\partial \phi}{\partial \pi} = -\alpha_d, \qquad \alpha_d = \rho_d^{-1}. \tag{2}$$

where z is geometric height, $\phi = gz$ is geopotential, and ρ_d and α_d are dry-air density and specific volume. In terms of η, the corresponding mass-coordinate metric is $\mu = \frac{\partial \pi}{\partial \eta}$, so that μ/g represents dry-air mass per unit horizontal area per unit η.

Air motion across the coordinate surfaces is described by the generalized vertical velocity $\dot{\eta} = D\eta/Dt$, which gives the vertical mass flux $F_\eta = \mu\dot{\eta}$. Here $D/Dt = \partial/\partial t + \boldsymbol{V}\cdot\nabla_\eta + \dot{\eta}\,\partial/\partial\eta$ is the material derivative, $\boldsymbol{V}$ is the horizontal velocity vector, and ∇_η denotes horizontal differentiation along constant-η surfaces. Impermeable boundary conditions are imposed at the model top and bottom by setting $F_\eta(\eta_t) = F_\eta(\eta_s) = 0$, where η_t and η_s denote the vertical coordinate values at the model top and surface, respectively.

2.2 Governing equations

In the dry-mass hybrid coordinate, the fully compressible nonhydrostatic governing equations take the form:

$$\frac{\partial \mu}{\partial t} + \nabla_\eta \cdot (\mu \boldsymbol{V}) + \frac{\partial F_\eta}{\partial \eta} = 0, \tag{3}$$

$$\frac{\partial (\mu\theta_m)}{\partial t} + \nabla_\eta \cdot (\mu \boldsymbol{V}\theta_m) + \frac{\partial (F_\eta \theta_m)}{\partial \eta} = \mu S(\theta_m), \tag{4}$$

$$\frac{\partial \boldsymbol{V}}{\partial t} + \zeta_a \boldsymbol{k}\times\boldsymbol{V} + \nabla_\eta KE + \dot{\eta}\frac{\partial \boldsymbol{V}}{\partial \eta} = -\mathcal{L}\left(\alpha_d \nabla_\eta p + \frac{1}{\mu}\frac{\partial p}{\partial \eta}\nabla_\eta \phi\right) + S(\boldsymbol{V}), \tag{5}$$

$$\frac{\partial w}{\partial t} + \boldsymbol{V}\cdot\nabla_\eta w + \dot{\eta}\frac{\partial w}{\partial \eta} = g\left(\frac{\mathcal{L}}{\mu}\frac{\partial p}{\partial \eta} - 1\right) + S(w), \tag{6}$$

$$\frac{\partial \phi}{\partial t} + \boldsymbol{V}\cdot\nabla_\eta \phi + \dot{\eta}\frac{\partial \phi}{\partial \eta} = gw + S(\phi), \tag{7}$$

$$\frac{\partial (\mu q_i)}{\partial t} + \nabla_\eta \cdot (\mu \boldsymbol{V} q_i) + \frac{\partial (F_\eta q_i)}{\partial \eta} = \mu S(q_i). \tag{8}$$

Here p is actual pressure including nonhydrostatic perturbations, and $w = Dz/Dt$ is the geometric vertical velocity, positive upward. The thermodynamic variable θ_m is modified potential temperature. For water species i, the dry-air mixing ratio is $q_i = \rho_i/\rho_d$, where ρ_i is the mass of that species per unit volume. The factor $\mathcal{L} = \rho_d/\rho_m$ is the ratio of dry-air density to total moist-air density ρ_m. The term $S(\chi)$ denotes the non-advective source or sink tendency of variable χ due to physical processes or numerical dissipation. The horizontal momentum equation, Eq. (5), is written in vector-invariant form in terms of the horizontal kinetic energy per unit mass $KE = \frac{1}{2}|\boldsymbol{V}|^2$ and the vertical absolute vorticity $\zeta_a = \zeta_r + f$. The vertical relative vorticity is $\zeta_r = \boldsymbol{k}\cdot(\nabla_\eta \times \boldsymbol{V})$, where $\boldsymbol{k}$ is the local upward unit vector. The Coriolis parameter is $f = 2\Omega\sin\varphi$, with Ω denoting the planetary rotation rate and φ the latitude. The modified potential temperature θ_m (Klemp et al., 2007) is defined by

$$\theta = T\left(\frac{p_0}{p}\right)^\kappa, \qquad \theta_m = \theta(1+\epsilon_v q_v), \qquad \kappa = \frac{R_d}{c_{pd}}, \qquad \epsilon_v = \frac{R_v}{R_d} \simeq 1.608. \tag{9}$$

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

Here T is temperature, θ is potential temperature, R_d and R_v are the gas constants of dry air and water vapor, respectively, and c_{pd} is the dry-air specific heat at constant pressure. The quantities q_v, q_c, and q_r denote the dry-air mixing ratios of water vapor, cloud water, and rainwater, respectively. The corresponding modified temperature is $T_m = T(1 + \epsilon_v q_v)$, giving the ideal-gas equation $p = \rho_d R_d T_m$. Modified potential temperature differs from virtual potential temperature: virtual temperature, defined using total moist-air density, satisfies $T_v = \mathcal{L} T_m$. The thermodynamic closure uses dry-air constants in the Exner function $(p/p_0)^\kappa$, while retaining the water-vapor contribution to the equation of state and water loading in momentum. In terms of the water-species mixing ratios, ρ_m and $\mathcal{L}$ are given by

$$\rho_m = \rho_d(1 + q_v + q_c + q_r + \cdots), \qquad \mathcal{L} = \frac{\rho_d}{\rho_m} = \frac{1}{1 + q_v + q_c + q_r + \cdots}. \tag{10}$$

The ellipsis denotes any additional water species included in total mass. Passive tracers may use the same mass-weighted transport equation but are excluded from the water-species sum in Eq. (10) and do not affect the equation of state. The system is closed by the diagnostic specific-volume relation in Eq. (2) and the equation of state (Klemp et al., 2007)

$$p = p_0 \left(\frac{R_d \theta_m}{p_0 \alpha_d} \right)^\gamma, \qquad \gamma = \frac{c_{pd}}{c_{vd}} = \frac{1}{1 - \kappa}, \tag{11}$$

where c_{vd} is the dry-air specific heat at constant volume and $R_d = c_{pd} - c_{vd}$.

TerraFlow's nonhydrostatic dynamical core (NDC) solves the governing equations presented above. The corresponding hydrostatic dynamical core (HDC) is obtained by replacing the vertical momentum equation with the hydrostatic balance condition $\mathcal{L}\partial p/\partial \pi = 1$. In this configuration, w and ϕ are no longer advanced as independent prognostic variables. Geopotential is instead diagnosed by integrating Eq. (2) upward from the surface boundary $\phi(\eta_s) = \phi_s = g z_s$, where ϕ_s is surface geopotential and z_s is terrain height. For dry air, $\mathcal{L} = 1$ and $\theta_m = \theta$, so hydrostatic balance gives $p = \pi$ provided that their values coincide at the model top. This equality generally does not hold in moist air, because π is defined in terms of dry-air mass whereas hydrostatic pressure reflects the total moist-air mass.

3 Numerical methods and implementation

3.1 Spatial discretization

TerraFlow implements spatial operators on the unstructured horizontal mesh using indirect addressing. Each operator retrieves the required indices of neighboring cells, edges, and vertices from mesh adjacency tables and uses them to access the corresponding state and geometry arrays (MacDonald et al., 2011). Separating mesh connectivity from array storage order provides flexibility in mesh geometry and topology, allowing the same operators to accommodate cells with different numbers of neighbors and to operate on both quasi-uniform and variable-resolution meshes. This flexibility allows the operator code to be reused on other mesh geometries, including cubed-sphere meshes, provided that they satisfy the geometric requirements of the discretization (Ringler et al., 2010). Indirect addressing further enables mesh reindexing schemes to adjust the numbering and storage order of cells, edges, and vertices, potentially improving memory locality and computational efficiency without changing mesh connectivity or the operator code.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

These operators act on variables arranged using horizontal C-grid staggering on a primal–dual mesh pair and Lorenz staggering in the vertical (Fig. 1). On the SCVT meshes considered here, the primal control volumes are spherical polygons, and the dual mesh consists of Delaunay triangles formed by connecting neighboring primal-cell centers (Ringler et al., 2008). The vertical discretization of the coordinate defined in Sect. 2.1 uses N_z layers numbered from the model top toward the surface, with full levels (layer centers) indexed by k and half levels (layer interfaces) by $k \pm \frac{1}{2}$. Interface coordinates follow directly from Eq. (1), and full-level coordinates are arithmetic means of adjacent half-level values. Integrating the continuous mass-coordinate metric μ over layer k gives its dry-mass thickness,

$$\delta\pi_k = \int_{\eta_{k-1/2}}^{\eta_{k+1/2}} \mu \, d\eta = \pi_{k+1/2} - \pi_{k-1/2}, \tag{12}$$

so that $\delta\pi_k/g$ is the dry-air mass per unit horizontal area in that layer. Layer dry-mass thickness $\delta\pi$, modified potential temperature θ_m, pressure p, dry-air specific volume α_d, and tracer mixing ratios q_i are located at primal-cell centers on full levels. At the same vertical levels, normal horizontal velocity u_n is located on primal-cell edges and vertical relative vorticity ζ_r at dual-cell centers. The edge-normal and tangential velocity components are $u_n = \boldsymbol{V}\cdot\boldsymbol{n}_e$ and $u_t = \boldsymbol{V}\cdot\boldsymbol{t}_e$, respectively, where $\boldsymbol{n}_e$ is the prescribed unit normal and $\boldsymbol{t}_e = \boldsymbol{n}_e \times \boldsymbol{k}$, with $\boldsymbol{k}$ the local upward unit vector. Geometric vertical velocity w, geopotential ϕ, and vertical mass flux F_η are defined at half levels.

On this staggered grid, the momentum equations are expressed in layer-averaged form, while the mass continuity, thermodynamic, and tracer transport equations are integrated over each layer. Related formulations are used in isopycnal ocean models (Oberhuber, 1993; Bleck, 1998), arbitrary Lagrangian–Eulerian ocean frameworks (Adcroft and Campin, 2004; Ringler et al., 2013), and atmospheric models with vertically Lagrangian (Lin, 2004; Chen et al., 2013) or quasi-Lagrangian coordinates (Bleck et al., 2015). In the Eulerian vertical discretization adopted here, the coordinate coefficients and η levels remain fixed, while layer mass thicknesses vary with π_s. The vertical mass flux F_η represents transport across these layer interfaces. Summing the layer-integrated form of the continuity equation, Eq. (3), over the atmospheric column yields a prognostic equation for π_s, with vanishing mass fluxes at the top and bottom. Layer integration of the thermodynamic and tracer transport equations, Eqs. (4) and (8), yields prognostic equations for the mass-weighted quantities $\delta\pi\theta_m$ and $\delta\pi q_i$, respectively. The horizontal momentum equation, Eq. (5), is projected onto the edge normal to obtain a prognostic equation for u_n, while the tangential component u_t is reconstructed from normal velocities on neighboring edges. The resulting spatially discrete system, including the diagnostic relations for α_d and p, can be written as:

$$\frac{\partial \pi_s}{\partial t} + \sum_{k=1}^{N_z} \nabla_\eta \cdot (\delta\pi \boldsymbol{V})_k = 0, \tag{13}$$

$$\frac{\partial (\delta\pi\theta_m)}{\partial t} + \nabla_\eta \cdot (\delta\pi \boldsymbol{V}\theta_m) + \delta(F_\eta\theta_m) = \delta\pi S(\theta_m), \tag{14}$$

$$\frac{\partial u_n}{\partial t} + \zeta_p \delta\pi u_t + \frac{\partial KE}{\partial n} + \dot{\eta}\frac{\partial u_n}{\partial \eta} = -\mathcal{L}\left(\alpha_d \frac{\partial p}{\partial n} + \frac{\delta p}{\delta\pi}\frac{\partial \phi}{\partial n}\right) + S(u_n), \tag{15}$$

$$\frac{\partial w}{\partial t} + \boldsymbol{V}\cdot\nabla_\eta w + \dot{\eta}\frac{\partial w}{\partial \eta} = \overline{g\mathcal{L}D_\pi p - g} + S(w), \tag{16}$$

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

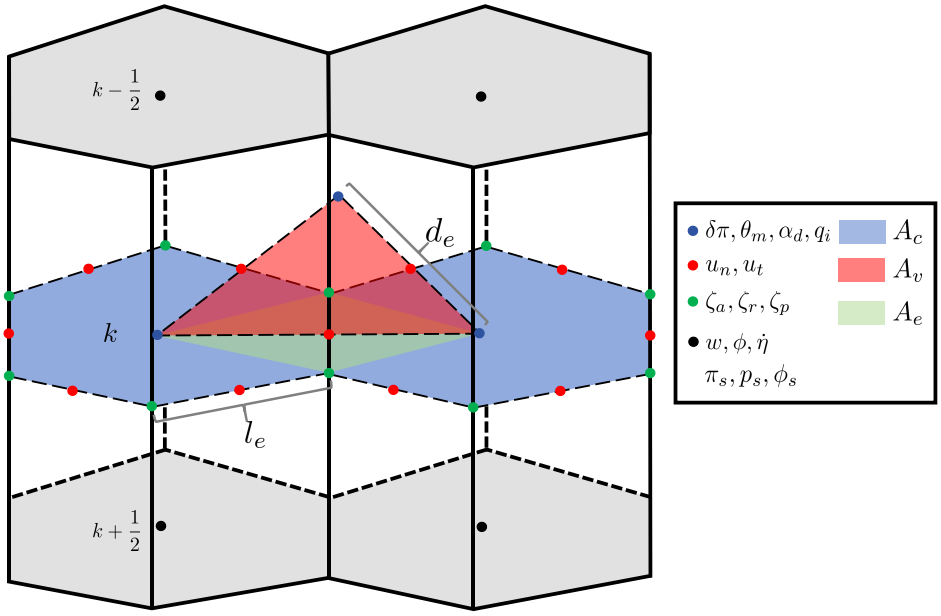

Figure 1. Horizontal C-grid and vertical Lorenz staggering in TerraFlow. Blue markers denote full-level scalars at primal-cell centers: layer dry-mass thickness $\delta\pi$, modified potential temperature θ_m, dry-air specific volume α_d, and tracer mixing ratios q_i. Red markers denote edge-normal velocity u_n and reconstructed tangential velocity u_t. Green markers denote dual-cell centers, where absolute vorticity ζ_a, relative vorticity ζ_r, and potential vorticity ζ_p are defined. Black markers denote layer-interface locations for vertical velocity w, geopotential ϕ, and coordinate velocity $\dot{\eta}$. The quantities π_s, p_s, and ϕ_s refer to the surface. Shaded areas A_c, A_v, and A_e represent a primal cell, a dual cell, and an edge diamond, respectively. The quantities d_e and l_e denote dual- and primal-edge lengths. Only representative variables are shown.

$$\frac{\partial \phi}{\partial t} + \boldsymbol{V} \cdot \nabla_\eta \phi + \dot{\eta} \frac{\partial \phi}{\partial \eta} = \overline{gw} + S(\phi), \tag{17}$$

$$\frac{\delta \phi}{\delta \pi} = -\alpha_d, \tag{18}$$

$$p = p_0 \left(-\frac{R_d \delta\pi \theta_m}{p_0 \delta\phi} \right)^\gamma, \tag{19}$$

$$\frac{\partial (\delta\pi q_i)}{\partial t} + \nabla_\eta \cdot (\delta\pi \boldsymbol{V} q_i) + \delta(F_\eta q_i) = \delta\pi S(q_i). \tag{20}$$

The spatial derivative symbols in this system denote discrete operators on the staggered grid. Horizontal derivatives, including the edge-normal derivative $\partial/\partial n$, are taken along surfaces of constant η, while $\delta\chi_k = \chi_{k+1/2} - \chi_{k-1/2}$ denotes the lower-minus-upper difference across a layer. The ratio $\delta p/\delta\pi$ represents the vertical pressure derivative at full levels, with δp denoting the difference between reconstructed interface pressures. Its counterpart at interior half levels is defined by $(D_\pi p)_{k+1/2} = (p_{k+1} - p_k)/(\pi_{k+1} - \pi_k)$. Geopotential is prognostic at layer interfaces in Eq. (17), with full-level values appearing in the horizontal momentum equation, Eq. (15). Equations (18) and (19) provide the diagnostic dry-air specific volume and pressure. The overbars in Eqs. (16) and (17) denote the semi-implicit time averages defined in Sect. 3.2.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

The horizontal terms are evaluated using discrete divergence, gradient, and curl operators on the primal–dual mesh pair (Ringler et al., 2010; Zhang et al., 2019). At each vertical level, the divergence theorem gives the divergence from the net outward flux across a primal-cell boundary. The edge-normal gradient is evaluated from differences between adjacent primal-cell values, and Stokes' theorem gives the vertical component of the curl from circulation around a dual cell:

$$(\nabla_\eta \cdot \boldsymbol{F})_c = \frac{1}{A_c} \sum_{e \in E(c)} s_{c,e}\, l_e F_e, \tag{21}$$

$$\left(\frac{\partial \chi}{\partial n}\right)_e = \frac{\chi_{c_2} - \chi_{c_1}}{d_e}, \tag{22}$$

$$[\boldsymbol{k} \cdot (\nabla_\eta \times \boldsymbol{V})]_v = \frac{1}{A_v} \sum_{e \in E(v)} \sigma_{v,e}\, d_e u_{n,e}. \tag{23}$$

Here $F_e = \boldsymbol{F}_e \cdot \boldsymbol{n}_e$ denotes an edge-normal flux and χ denotes a scalar at primal-cell centers, with vertical indices omitted for brevity. The geometric quantities A_c and A_v are the primal- and dual-cell areas, respectively, and l_e and d_e are the corresponding edge lengths. The sets $E(c)$ and $E(v)$ contain the indices of the edges bounding primal cell c and dual cell v, respectively. Each primal edge and its corresponding dual edge share the same index e. For each edge, the adjacent primal cells c_1 and c_2 are labeled so that the prescribed normal $\boldsymbol{n}_e$ points from c_1 to c_2. The sign $s_{c,e}$ is $+1$ when this normal points outward from cell c and -1 when it points inward. In the circulation sum, $\sigma_{v,e}$ is $+1$ when $\boldsymbol{n}_e$ points counterclockwise around the boundary of cell v as viewed from above, and -1 otherwise. For the horizontal transport terms in Eqs. (13), (14), and (20), the divergence operator uses $F_e = M_e$, $F_e = M_e \theta_{m,e}$, and $F_e = M_e q_{i,e}$, respectively. The mass flux is $M_e = \delta\pi_e u_{n,e}$, where $\delta\pi_e$ is the layer thickness interpolated to the edge. The scalar edge values $\theta_{m,e}$ and $q_{i,e}$ are reconstructed using the transport schemes described below. Each interior edge carries a shared flux that enters its two adjacent cells with opposite signs, so its contributions cancel in the area-weighted sum of the discrete divergence.

For the horizontal momentum equation, Eq. (15), the rotational contribution is evaluated using the transported potential vorticity (PV) $\zeta_p = \zeta_a / \delta\pi$. This model-dependent form of PV differs from Ertel potential vorticity and allows the horizontal discretization to follow the shallow-water formulation. The resulting nonlinear Coriolis term accounts for both planetary and relative vorticity and contributes to the edge-normal velocity tendency as

$$\left.\frac{\partial u_{n,e}}{\partial t}\right|_{\mathrm{rot}} = -(\zeta_p \delta\pi u_t)_e = -\frac{1}{d_e} \sum_{e' \in ECP(e)} w_{e,e'}\, l_{e'}\, M_{e'}\, \zeta_{p,ee'}, \qquad \zeta_{p,ee'} = \frac{\zeta_{p,e} + \zeta_{p,e'}}{2}. \tag{24}$$

The set $ECP(e)$ contains the edges of the two primal cells sharing e. The dimensionless tangential reconstruction weights $w_{e,e'}$ depend on mesh geometry and edge orientation (Thuburn et al., 2009). With $\boldsymbol{t}_e = \boldsymbol{n}_e \times \boldsymbol{k}$, the tangential reconstruction takes the form $u_{t,e} = d_e^{-1} \sum_{e' \in ECP(e)} w_{e,e'} l_{e'} u_{n,e'}$. To obtain the edge PV values in this flux sum, ζ_p is first diagnosed at dual-cell centers from absolute vorticity and the corresponding layer mass thickness, then reconstructed at edges as $\zeta_{p,e}$ using the transport scheme on the dual mesh. This discretization ensures that the nonlinear Coriolis term does no net work in the semi-discrete system (Ringler et al., 2010).

The accompanying kinetic-energy gradient in Eq. (15) requires a primal-cell estimate of horizontal kinetic energy. To alleviate the Hollingsworth instability associated with the discrete nonlinear momentum terms, a local estimate is blended with one

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

remapped from dual cells as described in Gassmann (2013), thereby enlarging the reconstruction stencil:

$$KE_c = \Gamma KE_c^{\mathrm{R10}} + (1-\Gamma) KE_c^{\mathrm{dc}\to\mathrm{pc}}. \tag{25}$$

Here $KE_c^{\mathrm{R10}} = (4A_c)^{-1} \sum_{e\in E(c)} l_e d_e u_{n,e}^2$ is the primal-cell estimate of Ringler et al. (2010). The second estimate is obtained on dual cells and remapped to primal cells using the areas of their intersections as weights. The dimensionless coefficient Γ sets the relative weights of the two estimates and has a default value of 0.9 in TerraFlow. The resulting cell-centered values are used in Eq. (22) to evaluate the edge-normal kinetic-energy gradient.

In terrain-following coordinates, the pressure-gradient contribution in Eq. (15) consists of a pressure gradient along coordinate surfaces and a correction for their slope, each of which may be large relative to the net force. To evaluate this residual accurately, TerraFlow's default scheme separates a hydrostatic reference profile from departures:

$$p' = p - p^{\mathrm{ref}}, \qquad \phi' = \phi - \phi^{\mathrm{ref}}, \qquad \alpha_d' = \alpha_d - \alpha_d^{\mathrm{ref}}, \qquad \pi' = \pi - \pi^{\mathrm{ref}}, \tag{26}$$

where the reference profile is fixed in time and satisfies $p^{\mathrm{ref}} = \pi^{\mathrm{ref}}$ and dry hydrostatic balance, $\alpha_d^{\mathrm{ref}} \partial p^{\mathrm{ref}}/\partial n + \partial \phi^{\mathrm{ref}}/\partial n = 0$. Combining the departure definitions in Eq. (26) with $p^{\mathrm{ref}} = \pi^{\mathrm{ref}}$ gives $\delta p/\delta\pi = 1 + (\delta p' - \delta\pi')/\delta\pi$. Substitution into Eq. (15) and cancellation of the reference hydrostatic terms give the pressure-gradient force (PGF)

$$\left.\frac{\partial u_n}{\partial t}\right|_{\mathrm{PGF}} = -\mathcal{L}\left[\alpha_d' \frac{\partial p^{\mathrm{ref}}}{\partial n} + \alpha_d \frac{\partial p'}{\partial n} + \frac{\partial \phi'}{\partial n} + \left(\frac{\delta p' - \delta\pi'}{\delta\pi}\right)\frac{\partial \phi}{\partial n}\right]. \tag{27}$$

The reference hydrostatic terms are thus canceled analytically before discretization, while the nonlinear products are retained without assuming infinitesimal departures. To evaluate Eq. (27) on the C-grid, normal gradients are computed using Eq. (22), and scalar coefficients multiplying these gradients are arithmetic means of the adjacent cell values. The moisture factor is evaluated at the same edge as $\mathcal{L}_e = [1 + (q_{m,c_1} + q_{m,c_2})/2]^{-1}$, where q_m is the sum of the water-species mixing ratios.

The vertical-advection term in Eq. (15) uses the energy-conserving centered discretization of Simmons and Burridge (1981). The mass flux F_η, diagnosed at primal-cell centers on half levels, is averaged horizontally to primal-edge locations at the same half levels. Layer thickness $\delta\pi$ is likewise averaged to edges at full levels. With the horizontal edge index omitted, the discrete operator can be written in two equivalent forms:

$$\begin{aligned}\left(\dot\eta \frac{\partial u_n}{\partial \eta}\right)_k &\approx \frac{\delta(F_\eta u_n)_k - u_{n,k}\delta F_{\eta,k}}{\delta\pi_k} \\ &= \frac{F_{\eta,k+1/2}(u_{n,k+1} - u_{n,k}) + F_{\eta,k-1/2}(u_{n,k} - u_{n,k-1})}{2\delta\pi_k}.\end{aligned} \tag{28}$$

The two forms are equivalent when the interface velocity is defined as the arithmetic mean of the adjacent full-level velocities, $u_{n,k+1/2} = (u_{n,k} + u_{n,k+1})/2$. This centered discretization allows vertical transport to redistribute horizontal kinetic energy among layers without introducing artificial sources or sinks, provided that the momentum and layer continuity equations use the same vertical mass flux. With impermeable top and bottom boundaries, the column-integrated, dry-mass-weighted horizontal kinetic energy is therefore conserved by vertical transport in the semi-discrete system. This semi-discrete property of vertical transport does not imply exact total-energy conservation by the fully time-discrete model.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

For the tracer transport equation, Eq. (20), spatial fluxes are evaluated using conservative finite-volume transport with first-, third-, and fifth-order upwind reconstruction options (Zhang, 2018; Skamarock and Gassmann, 2011). The higher-order horizontal reconstructions combine centered approximations of edge mixing ratios with velocity-dependent upwind dissipation. Both higher-order schemes use second derivatives estimated from quadratic least-squares fits to neighboring cell values on local tangent planes, while the fifth-order scheme additionally uses fourth derivatives from quartic fits over an extended stencil.

These high-order tracer fluxes can be limited using optional flux-corrected transport (FCT) to suppress nonphysical oscillations near sharp gradients (Zalesak, 1979; Skamarock, 2006; Wang et al., 2009). For three-stage Runge–Kutta (RK3) transport, the limiter is applied only in the final stage, with the first two stages retaining the unmodified high-order fluxes. The final-stage flux is then constructed by adding a limited antidiffusive correction to a first-order upwind flux:

$$\mathcal{F}_{i,e}^{\mathrm{FCT}} = \mathcal{F}_{i,e}^{\mathrm{L}} + a_{i,e}\left(\mathcal{F}_{i,e}^{\mathrm{H}} - \mathcal{F}_{i,e}^{\mathrm{L}}\right), \qquad 0 \le a_{i,e} \le 1. \tag{29}$$

Here $\mathcal{F}_{i,e} = M_e q_{i,e}$ denotes the tracer flux across edge e. The antidiffusive correction is the difference between the high-order flux $\mathcal{F}_{i,e}^{\mathrm{H}}$, evaluated from the second RK stage, and the low-order flux $\mathcal{F}_{i,e}^{\mathrm{L}}$, evaluated from the state at the beginning of the transport substep. Starting from the low-order solution, the limiter determines how much tracer each cell can gain or lose without violating its local mixing-ratio bounds. For the antidiffusive transfer across edge e, the coefficient $a_{i,e}$ is chosen using the more restrictive of the receiving cell's inflow limit and the donating cell's outflow limit, ensuring that the correction respects the bounds in both cells. This requires the low-order update to remain within the local bounds under its Courant constraint, with positive layer masses. An analogous limiting procedure is applied to the vertical tracer flux $F_\eta q_i$.

The tracer scheme is coupled to the dynamical core through mass fluxes averaged over each tracer time step. A tracer time step $\Delta t_{\mathrm{tr}} = N_d \Delta t$ comprises N_d dynamical steps of length Δt, where N_d is a positive integer. The dynamical core advances through these steps before tracer transport, providing horizontal and vertical mass fluxes from the final Runge–Kutta stage of each step. The averages supplied to the tracer scheme are

$$\overline{M}_e = \frac{1}{N_d}\sum_{j=1}^{N_d} M_e^{[j]}, \qquad \overline{F}_\eta = \frac{1}{N_d}\sum_{j=1}^{N_d} F_\eta^{[j]}, \tag{30}$$

where $[j]$ labels dynamical steps within the tracer interval. At each dynamical step, the vertical mass fluxes are diagnosed from dry-air continuity with impermeable top and bottom boundaries. For fixed hybrid-coordinate coefficients, this diagnosis is linear in the horizontal mass-flux divergence. The averaged vertical flux $\overline{F}_\eta$ can therefore be obtained either by averaging the diagnosed fluxes or by diagnosing it from $\overline{M}_e$.

Using these averaged mass fluxes, three-dimensional tracer transport is split into horizontal and vertical updates, with horizontal transport applied first by default. The dynamical core also supplies the layer masses at the beginning and end of the tracer interval, $\delta\pi^n$ and $\delta\pi^{n+N_d}$. With $Q_i = \delta\pi q_i$, the two directional updates take the conservative form

$$Q_i^h = Q_i^n - \Delta t_{\mathrm{tr}} \nabla_\eta \cdot \mathcal{F}_{i,h}, \qquad q_i^h = \frac{Q_i^h}{\delta\pi^{n+N_d}}, \tag{31}$$

$$Q_i^{n+N_d} = Q_i^h - \Delta t_{\mathrm{tr}} \delta\mathcal{F}_{i,\eta}, \qquad q_i^{n+N_d} = \frac{Q_i^{n+N_d}}{\delta\pi^{n+N_d}}. \tag{32}$$

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

Here h denotes the intermediate state after horizontal transport. For RK3 transport, $\mathcal{F}_{i,h}$ and $\mathcal{F}_{i,\eta}$ are the final-stage tracer fluxes in the respective directional integrations. These fluxes are constructed from $\overline{M}_e$ and $\overline{F}_\eta$, respectively, using the reconstruction and optional FCT limiter described above. Both the intermediate and final mixing ratios are recovered from the updated tracer content using the end-of-interval layer mass.

3.2 Time integration and vertical implicit solver

TerraFlow supports three-stage and two-stage Runge–Kutta schemes, as well as Euler integration, to advance the spatially discrete equations. The calculations reported here mainly used the three-stage scheme of Wicker and Skamarock (2002). For the state vector $\boldsymbol{Y}$ and a purely explicit tendency operator $\mathcal{T}$, its stage updates take the form

$$\boldsymbol{Y}^{(1)} = \boldsymbol{Y}^n + \frac{\Delta t}{3}\mathcal{T}(\boldsymbol{Y}^n), \tag{33}$$

$$\boldsymbol{Y}^{(2)} = \boldsymbol{Y}^n + \frac{\Delta t}{2}\mathcal{T}(\boldsymbol{Y}^{(1)}), \tag{34}$$

$$\boldsymbol{Y}^{n+1} = \boldsymbol{Y}^n + \Delta t\,\mathcal{T}(\boldsymbol{Y}^{(2)}). \tag{35}$$

Each stage advances from the beginning-of-step state $\boldsymbol{Y}^n$, with the tendency evaluated at the preceding stage state. For a purely explicit problem, this scheme is third-order accurate for a time-independent linear operator and second-order accurate for a general nonlinear operator. The temporal accuracy of the coupled model also depends on explicit–implicit coupling and splitting. In the dynamical core, each stage is implemented as a forward–backward update (Mesinger and Arakawa, 1976). Tendencies associated with mass continuity, thermodynamics, rotation, and momentum advection are first evaluated from the known stage state. After diagnosing the vertical mass flux F_η, the core updates π_s, $\delta\pi$, and $\delta\pi\theta_m$. The thermodynamic variable θ_m is then recovered from the updated mass-weighted quantity using the updated layer mass. The HDC next diagnoses pressure and geopotential hydrostatically, while the NDC solves the coupled vertical-momentum and geopotential equations and updates pressure. Finally, both configurations use the updated fields to evaluate the pressure-gradient force and complete the u_n update together with the explicit momentum tendencies.

The NDC treats vertically propagating acoustic-wave terms with a semi-implicit solver, while horizontal and vertical advection remain explicit (Zhang et al., 2019). The vertical-momentum and geopotential equations form a subsystem closed by the equation of state. For stages $s = 1,2,3$, let $\boldsymbol{Y}^{(0)} = \boldsymbol{Y}^n$ and $\boldsymbol{Y}^{(3)} = \boldsymbol{Y}^{n+1}$, with stage increments $\Delta\tau_s = \Delta t/3$, $\Delta t/2$, and Δt, respectively. The overbarred terms in Eqs. (16) and (17) combine known and updated contributions according to

$$\overline{\chi}^{(s)} = (1-\xi)\chi^{(s-1)} + \xi\chi^{(s)}, \qquad s = 1,2,3, \quad \frac{1}{2} \le \xi \le 1, \tag{36}$$

where χ denotes $g\mathcal{L}D_\pi p - g$ or gw, and ξ is the semi-implicit weighting coefficient. The moisture factor $\mathcal{L}$ is held fixed during the implicit solve. The choice $\xi = 1/2$ assigns equal weights to the known and updated contributions, whereas $\xi > 1/2$ introduces additional damping of high-frequency vertical acoustic modes (Durran and Klemp, 1983).

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

To obtain a linear system for the implicit update, the equation of state, Eq. (19), is linearized about the beginning-of-step state $\boldsymbol{Y}^n$. With $\Theta = \delta\pi\theta_m$ and $\Theta^{(s)}$ known from the thermodynamic update, the first-order pressure approximation is

$$\widetilde{p}^{(s)} = p^n + \gamma p^n \left(\frac{\Theta^{(s)}}{\Theta^n} - \frac{\delta\phi^{(s)}}{\delta\phi^n} \right). \tag{37}$$

Using this pressure approximation, the coupled stage updates can be written as

$$\phi^{(s)} = \widehat{\phi}^{(s)} + \xi g \Delta\tau_s \, w^{(s)}, \tag{38}$$

$$w^{(s)} = \widehat{w}^{(s)} + \xi g \Delta\tau_s \, \mathcal{L} D_\pi^{(s)} \widetilde{p}^{(s)}. \tag{39}$$

Here $\widehat{\phi}^{(s)}$ and $\widehat{w}^{(s)}$ collect the step-initial values, explicit advection increments, and known parts of the semi-implicit averages. The latter contribute $(1-\xi) g \Delta\tau_s w^{(s-1)}$ to $\widehat{\phi}^{(s)}$ and $(1-\xi) g \Delta\tau_s \mathcal{L}(D_\pi p)^{(s-1)}$ to $\widehat{w}^{(s)}$. The gravity increment $-g\Delta\tau_s$ is also included in $\widehat{w}^{(s)}$, while the operator $D_\pi^{(s)}$ uses the updated mass coordinate. Substituting the geopotential update, Eq. (38), into the linearized equation of state, Eq. (37), expresses pressure in terms of $w^{(s)}$. Inserting this expression into Eq. (39) yields a system for the interface velocities alone,

$$w^{(s)} + (\xi g \Delta\tau_s)^2 \mathcal{L} D_\pi^{(s)} \left(\frac{\gamma p^n}{\delta\phi^n} \, \delta w^{(s)} \right) = \widehat{w}^{(s)} + \xi g \Delta\tau_s \, \mathcal{L} D_\pi^{(s)} \widehat{p}^{(s)}, \tag{40}$$

where the known pressure contribution is

$$\widehat{p}^{(s)} = p^n + \gamma p^n \left(\frac{\Theta^{(s)}}{\Theta^n} - \frac{\delta\widehat{\phi}^{(s)}}{\delta\phi^n} \right). \tag{41}$$

On the vertically staggered grid, Eq. (40) couples each interior interface velocity to its two vertical neighbors and therefore defines a tridiagonal linear system in each column. These column systems are solved independently. The resulting $w^{(s)}$ is used to recover geopotential from Eq. (38), followed by pressure from Eq. (37). The recovered pressure is then available for the horizontal pressure-gradient update, with the optional pressure filter applied beforehand as described in Sect. 3.3.

3.3 Damping and diffusion

TerraFlow supplements dissipation from upwind fluxes and off-centered semi-implicit integration with optional Smagorinsky horizontal diffusion and horizontal divergence damping. The NDC also includes pressure filtering and an upper-level absorbing layer for controlling acoustic modes and reducing gravity-wave reflection, respectively.

For horizontal mixing, Smagorinsky diffusion determines the horizontal eddy viscosity K_h, with units of $\mathrm{m^2\,s^{-1}}$, from the local horizontal deformation rate (Smagorinsky, 1963). Its value at edge e, denoted by $K_{h,e}$, is

$$K_{h,e} = \ell_{s,e}^2 \sqrt{ \left(\frac{\partial u_x}{\partial x} - \frac{\partial u_y}{\partial y} \right)_e^2 + \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right)_e^2 }, \qquad \ell_{s,e}^2 = C_s^2 d_e l_e. \tag{42}$$

Here (u_x, u_y) are reconstructed velocities in local orthogonal horizontal coordinates (x, y), distinct from zonal and meridional (u, v). The quantity $\ell_{s,e}$ is the Smagorinsky mixing length at edge e, measured in meters, and C_s is the dimensionless

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

Smagorinsky constant. The eddy viscosity enters the momentum and thermodynamic diffusion tendencies as

$$\left.\frac{\partial u_{n,e}}{\partial t}\right|_{\text{smag}} = K_{h,e}\left(\Delta_h \boldsymbol{V}\right)_{n,e}, \tag{43}$$

$$\left.\frac{\partial(\delta\pi\theta_m)}{\partial t}\right|_{\text{smag}} = \delta\pi\,\nabla_\eta\cdot(K_h\nabla_\eta\theta_m). \tag{44}$$

For each primal edge e, c_1 and c_2 denote the two adjacent primal cells, while v_1 and v_2 denote the dual cells centered at its endpoints. The momentum Laplacian uses velocities reconstructed at these four cell centers and projected onto the normal at edge e. Denoting the primal- and dual-cell velocity projections by $(u_{n,e}^{c_1}, u_{n,e}^{c_2})$ and $(u_{n,e}^{v_1}, u_{n,e}^{v_2})$, respectively, gives

$$\left(\Delta_h \boldsymbol{V}\right)_{n,e} = 4\left[\frac{u_{n,e}^{c_1}+u_{n,e}^{c_2}-2u_{n,e}}{d_e^2}+\frac{u_{n,e}^{v_1}+u_{n,e}^{v_2}-2u_{n,e}}{l_e^2}\right]. \tag{45}$$

The factor of four accounts for half-edge distances and is included in the definition of Δ_h here. In Zängl et al. (2015, Eqs. (35)–(37)), it is applied outside the Laplacian. The accuracy of this stencil depends on mesh geometry. For the thermodynamic tendency in Eq. (44), the spatially varying diffusivity is retained inside the divergence, so the operator reduces to $K_h\nabla_\eta^2\theta_m$ only when K_h is spatially uniform.

The divergent component of the horizontal wind can also be damped directly through

$$\left.\frac{\partial u_n}{\partial t}\right|_{\text{div}} = (-1)^{r+1}\nu_{2r}\frac{\partial}{\partial n}(\nabla_\eta^2)^{r-1}D, \qquad D=\nabla_\eta\cdot\boldsymbol{V}, \quad r=1,2. \tag{46}$$

Here D is horizontal divergence (in inverse seconds), and $\nu_{2r}\geq 0$ has units of $\mathrm{m}^{2r}\,\mathrm{s}^{-1}$. The choices $r=1$ and $r=2$ give second- and fourth-order divergence damping, respectively. The coefficients may vary with mesh scale and height but are kept outside the derivative. For constant coefficients, the sign convention ensures decay of divergent Fourier modes. The fourth-order form follows Zängl et al. (2015, Eq. (32)), with horizontal divergence used in place of three-dimensional divergence.

In the NDC, pressure filtering is applied after the vertical implicit solve and before evaluation of the horizontal pressure-gradient force. It uses the pressure increment between the beginning-of-step value p^n and the linearized pressure $\widetilde{p}^{(s)}$ from Eq. (37), with the filtered pressure written as

$$p^{(s)} = \widetilde{p}^{(s)} + \beta_p\left(\widetilde{p}^{(s)} - p^n\right), \tag{47}$$

where β_p is a dimensionless coefficient. For positive β_p, the form shown in Eq. (47) extrapolates the pressure increment forward in time. Through the relation between pressure tendency and three-dimensional divergence, this extrapolation adds an acoustic-damping contribution to the horizontal pressure-gradient force (Klemp et al., 2007, Eq. (45)).

To reduce artificial reflection of upward-propagating gravity waves at the model top, the NDC provides an optional absorbing layer that applies implicit Rayleigh damping to w (Klemp et al., 2008). The damping coefficient $\mu_{\mathrm{R}}(z)$, with units of s^{-1}, is zero below the absorbing layer and increases smoothly within it from zero at its base to a prescribed maximum at the model top, following a sine-squared profile. At RK stage s, the vertical velocity w^* obtained from the vertically implicit solve is damped according to

$$w^{(s)} = \frac{w^*}{1+\mu_{\mathrm{R}}(z)\Delta\tau_s} \tag{48}$$

before being used in the geopotential update described in Sect. 3.2.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

3.4 Software design

TerraFlow is organized as a Python library with an application programming interface (API) for constructing hydrostatic, non-hydrostatic, shallow-water, and advection models (Fig. 2). Users specify the planetary parameters, horizontal mesh, vertical coordinate, initial conditions, and numerical settings through this interface. Simulation variables are held in model objects rather than module-level global variables, allowing users to construct, integrate, and inspect models within interactive environments such as Jupyter notebooks.

A device object selects JAX or PyTorch, the target hardware, and numerical precision throughout model construction. Numerical operators access backend-specific array operations and state updates through a shared interface, so the numerical formulation can be used with either JAX or PyTorch. JAX uses functional updates because its arrays are immutable. TerraFlow's PyTorch backend provides functional updates for differentiation and in-place updates for forward integration through the same state-update interface. Functional updates preserve values that may be needed during differentiation, but retaining these values can increase memory requirements. In-place updates overwrite existing tensors, allowing their storage to be reused during forward integration.

Parallel communication is handled through the Message Passing Interface (MPI) in a layer separate from the local numerical operators. During mesh construction, METIS (Karypis and Kumar, 1998) partitions the horizontal domain, and the API identifies interior, boundary, and halo regions (Fig. 3). Operators use local connectivity and geometry, while the communication layer exchanges halo data through `mpi4jax` (Häfner and Vicentini, 2021) for JAX or `mpi4py` (Dalcin and Fang, 2021) for PyTorch. In serial execution, the local mesh spans the full domain. This separation allows the same local operators to be reused in serial and distributed forward integrations.

To initialize a model, users can select an idealized test-case object or supply background and topography fields through data readers. The model builder prepares these fields on the target mesh. Background fields, including the fifth-generation reanalysis from the European Centre for Medium-Range Weather Forecasts (ERA5; Hersbach et al., 2020), can be remapped using built-in bilinear interpolation or the Earth System Modeling Framework (ESMF; Hill et al., 2004). Terrain inputs include the ETOPO1 1 Arc-Minute Global Relief Model dataset (Amante and Eakins, 2009) and Global Multi-resolution Terrain Elevation Data 2010 (GMTED2010; Danielson and Gesch, 2011). These fields can be transferred to the mesh using k-dimensional tree (K-D tree) searches and Laplacian smoothing.

After initialization, users can choose from three output modes according to their analysis needs. The `integrate` mode returns the model at the final time, while `trajectory` additionally returns arrays formed by stacking the values of each model variable at successive output times along a new time dimension. The `IO_callback` mode writes output to Network Common Data Form (netCDF) files at specified times for subsequent analysis. The final states or trajectories can also be used to define loss functions and compute gradients through the integration. All three modes allow users to enable JAX just-in-time (JIT) compilation or the PyTorch compiler to accelerate execution.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

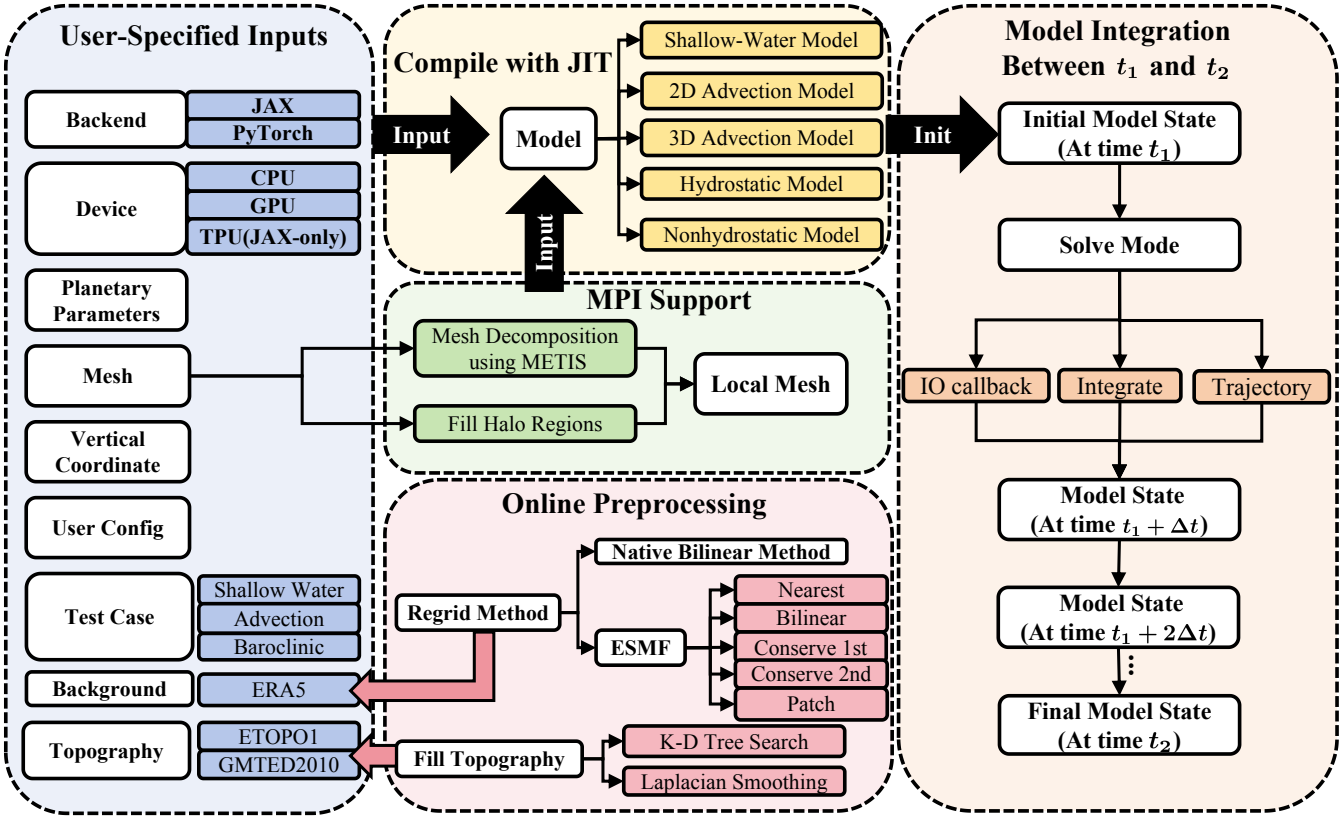

Figure 2. TerraFlow architecture and model-construction workflow. "Solve Mode" branches select final-state, trajectory, or history-file output. The JAX interface includes a tensor processing unit (TPU) option, although only central processing units (CPUs) and GPUs were assessed here. JIT denotes just-in-time compilation. The integration window extends from t_1 to t_2, with Δt denoting a full time step.

4 Dynamical tests

We evaluated TerraFlow's representation of tracer transport, large-scale rotational flow, and nonhydrostatic waves using five idealized tests. All tests were restricted to dry dynamics and passive transport, with $\mathcal{L} = 1$ and $\theta_m = \theta$. Table 1 lists meshes, vertical levels, time steps, and principal dissipation settings. DCMIP denotes the Dynamical Core Model Intercomparison Project. DYN denotes the dynamical time step, and ADV denotes the tracer advection time step. The semi-implicit coefficient ξ applies only to the NDC. "Div damp" denotes horizontal divergence damping, "P damp" denotes acoustic-mode damping by the pressure filter, and "Rayleigh (w)" denotes upper-level Rayleigh damping of vertical velocity. "None" indicates that no dissipation or damping is applied beyond that inherent in upwind fluxes and time discretization. The label Gn denotes icosahedral refinement level n, with $10 \times 4^n + 2$ horizontal cells. The G4, G5, G6, and G7 meshes therefore contain 2562, 10242, 40962, and 163842 cells, respectively. At Earth's radius, G6 and G7 have nominal horizontal spacings of approximately

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

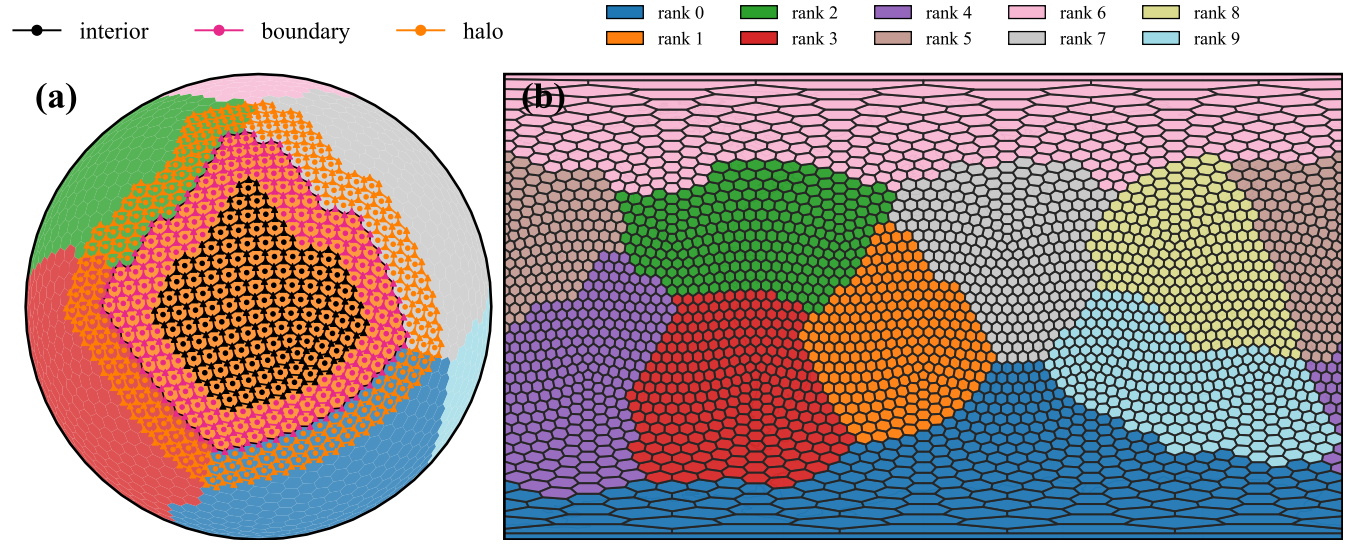

Figure 3. Partitioning a G4 mesh among 10 processes. The legend uses zero-based MPI process identifiers (ranks). (a) Local mesh for process 1: black, magenta, and orange denote the interior, boundary, and halo regions, respectively. (b) Planar projection of the global partition, with colors indicating the regions assigned to different processes.

120 and 60 km. With reduced radius, physical spacing is divided by the scaling factor X. Thus, although both final tests used G6, their physical spacings were approximately 0.96 and 0.72 km.

4.1 Three-dimensional advection

To isolate transport errors from dynamical feedback, we used DCMIP 1-2 with analytically prescribed zonal wind and a Hadley-like meridional–vertical circulation (Hall et al., 2016, Eqs. (25)–(28)). The background temperature is 300 K and density varies with height. Initially, the tracer is horizontally uniform and confined to $z = 2$–5 km, with a cosine profile reaching 1 at $z = 3.5$ km and zero outside the layer. Meridional and vertical velocities reverse after 12 h, giving a 24 h return period. The initial tracer field therefore provides an exact reference for the continuous advection problem at 24 h. Since the tracer field is zonally uniform, the prescribed zonal wind does not affect its return to the initial distribution.

We compared G5, G6, and G7 meshes, with nominal horizontal spacings of 240, 120, and 60 km and 30, 60, and 120 vertical layers, respectively. The layers were uniformly spaced in η. The DYN and ADV time steps were equal within each configuration: 600, 300, and 150 s for G5, G6, and G7, respectively (Table 1). All three configurations used third-order upwind reconstruction in the horizontal and vertical, with FCT enabled. We evaluated normalized errors at 24 h relative to the initial tracer field. The passive-tracer mixing ratio q corresponds to one component q_i in Eq. (8). For each configuration, let q_j^0 denote the initial mixing ratio, $e_j = q_j(24\,\mathrm{h}) - q_j^0$, and V_j the corresponding volume weight. Following the DCMIP definitions (Ullrich

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

Table 1. Model configurations for the idealized tests. Horizontal spacings are approximate physical distances on the simulated sphere. DCMIP 3-1 and the quasi-two-dimensional ridge use radius-reduction factors $X = 125$ and 166.7, respectively. The semi-implicit coefficient ξ applies only to the NDC. Dashes indicate an inapplicable or unconfigured option.

Test case	Grid / spacing	Levels	Time step (s)	ξ (NDC)	Damping
DCMIP 1-2 advection	G5 / 240 km	30	DYN: 600 ADV: 600	–	–
	G6 / 120 km	60	DYN: 300 ADV: 300	–	–
	G7 / 60 km	120	DYN: 150 ADV: 150	–	–
Mountain-induced Rossby waves	G6 / 120 km	26	HDC: 300 NDC: 150	0.55	HDC: None NDC: Div damp
3D Rossby–Haurwitz wave	G6 / 120 km	26	HDC: 300 NDC: 150	0.55	HDC: None NDC: Div damp
DCMIP 3-1 gravity waves	G6 / 0.96 km	10	NDC: 2	0.55	NDC: None
Quasi-2D mountain ridge waves	G6 / 0.72 km	40	NDC: 1	0.55	NDC: P damp + Rayleigh (w)

et al., 2012), the global error norms are

$$L_1 = \frac{\sum_j V_j |e_j|}{\sum_j V_j |q_j^0|}, \qquad L_2 = \left(\frac{\sum_j V_j e_j^2}{\sum_j V_j (q_j^0)^2} \right)^{1/2}, \qquad L_\infty = \frac{\max_j |e_j|}{\max_j |q_j^0|}. \tag{49}$$

The volume weights combine spherical cell areas with vertical layer thicknesses. Latitude-band errors use the same L_2 definition, with sums restricted to longitude and height within each band.

All three global error norms decreased under joint horizontal, vertical, and temporal refinement (Fig. 4a). From G5 to G7, (L_1, L_2, L_∞) decreased from $(0.338, 0.301, 0.515)$ to $(0.0341, 0.0366, 0.0989)$. With horizontal spacing halved between configurations, the observed rate for each error norm E was $p_{\mathrm{obs}} = \log(E_{\mathrm{coarse}}/E_{\mathrm{fine}})/\log 2$. The rates were $(1.61, 1.40, 1.04)$ from G5 to G6 and $(1.70, 1.64, 1.34)$ from G6 to G7, indicating slower convergence in L_∞ than in L_1 and L_2. Figure 4b shows the latitude dependence of the normalized L_2 error. Peak values were 0.520 on G5 and 0.229 on G6, both near $\pm 33.5°$, and 0.0828 on G7 near the Equator.

The G7 latitude–height sections illustrate tracer deformation and recovery (Fig. 4c–d). At 12 h, the layer reached approximately 10–11 km near the Equator, with secondary maxima in layer height near $\pm 60°$ and minima near $\pm 30°$. By 24 h, the layer had largely recovered its initial 2–5 km distribution, with weak undulations and latitudinal variations in mixing ratio remaining.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

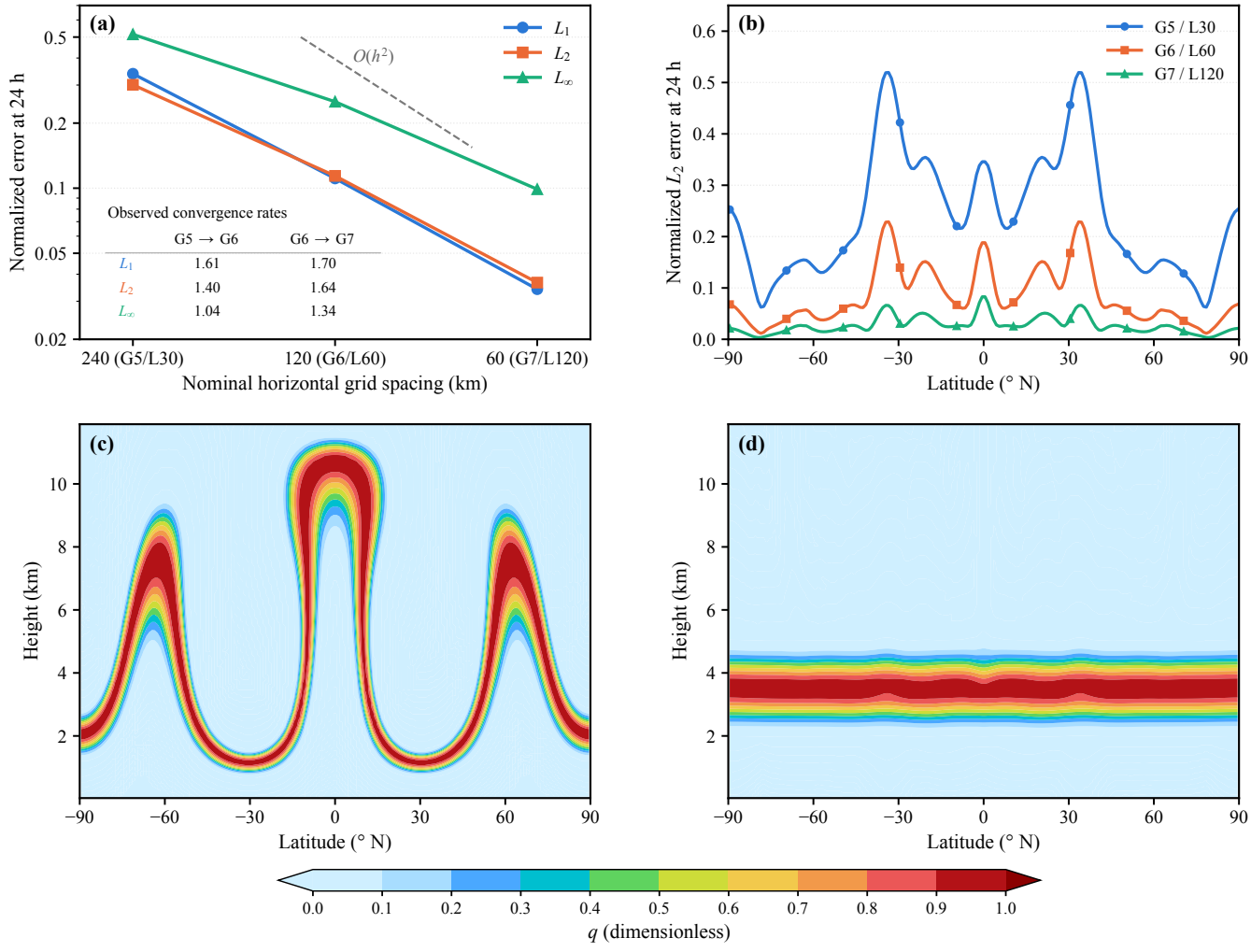

Figure 4. DCMIP 1-2 advection. Normalized errors at 24 h for G5/L30, G6/L60, and G7/L120: (a) global L_1, L_2, and L_∞ norms and (b) latitude-band L_2 errors. L30, L60, and L120 denote vertical layer counts. The inset in panel (a) reports observed convergence rates under joint spatial and temporal refinement. The dashed $O(h^2)$ line provides a reference slope, where h denotes the horizontal grid spacing. (c, d) Latitude–height sections of dimensionless tracer mixing ratio q for G7/L120 at $\lambda = 180.5°$ E: (c) 12 h and (d) 24 h.

4.2 Mountain-induced Rossby waves

A localized mountain in a planetary-scale zonal flow can generate a Rossby wave train affecting remote circulation. We used the test case described in Sect. 1.5 of Jablonowski et al. (2008): the background temperature is 288 K, initial zonal wind is $u = 20 \cos\varphi \, \mathrm{m\,s^{-1}}$, and initial meridional and vertical winds are zero. A Gaussian mountain centered at $(90° \mathrm{E}, 30° \mathrm{N})$ has
440 a peak height of 2 km and a horizontal width parameter of 1500 km. Initial pressure follows the balance relation in the test

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

specification. The horizontal scale produces a predominantly hydrostatic response, while finite mountain height introduces appreciable nonlinearity.

Figure 5 shows the $700\,\mathrm{hPa}$ fields on day 25. Pronounced troughs and ridges developed around and downstream of the Northern Hemisphere mountain. A cold region near the mountain lay adjacent to a downstream warm region, accompanied by horizontal-wind acceleration and alternating meridional-wind bands. The response extended beyond the terrain: broad disturbances in Southern Hemisphere height and wind fields reflect global adjustment to the forcing. Both the HDC and NDC reproduced the main spatial patterns of geopotential height, temperature, and horizontal wind shown in Figs. 19 and 20 of Jablonowski et al. (2008). The NDC temperature and wind fields were smoother. Divergence damping, enabled only in the NDC configuration (Table 1), may have contributed to this difference.

4.3 Three-dimensional Rossby–Haurwitz waves

The three-dimensional Rossby–Haurwitz wave evolves from a prescribed global rotational flow without topographic forcing. We used the wavenumber-4 initial state in Sect. 1.4 of Jablonowski et al. (2008): horizontal wind is nondivergent and independent of height, initial vertical wind is zero, and temperature and surface pressure follow the corresponding balance relations. A Rossby–Haurwitz wave is an analytical solution of the barotropic vorticity equation, but not of the three-dimensional nonlinear governing equations. A rigidly translated initial field is therefore not an exact reference for this test. The test instead examines preservation of large-scale rotational modes, wave symmetry, and the effects of discretization errors on the wavenumber-4 pattern.

Figure 6 shows the day-15 fields from both the HDC and NDC. The four-wave pattern in $500\,\mathrm{hPa}$ geopotential height and the alternating wind maxima and minima at $850\,\mathrm{hPa}$ were consistent with the reference solution in Fig. 17 of Jablonowski et al. (2008). Both simulations also reproduced the reference solution's equatorial symmetry in geopotential height and zonal wind and antisymmetry in meridional wind. Temperature variations at $850\,\mathrm{hPa}$ remained small, with alternating warm and cold regions broadly matching the reference pattern, although finer structures differed.

4.4 Nonhydrostatic gravity waves

DCMIP 3-1 reduces the spherical radius to $1/125$ of Earth's radius and switches off planetary rotation, bringing horizontal scales into the nonhydrostatic regime. A localized potential-temperature perturbation excites gravity waves over flat terrain (Ullrich et al., 2012). The background has constant Brunt–Väisälä frequency $N = 0.01\,\mathrm{s}^{-1}$ and zonal flow $u_b = 20\cos\varphi\,\mathrm{m\,s}^{-1}$. The perturbation is centered at $(120^\circ\,\mathrm{E}, 0^\circ)$, with a bell-shaped horizontal profile of width parameter $5\,\mathrm{km}$, a sinusoidal vertical profile of wavelength $20\,\mathrm{km}$, and maximum amplitude $1\,\mathrm{K}$. Full expressions are given in Eqs. (102)–(107) of Ullrich et al. (2012). Half the vertical wavelength corresponds to the approximately $10\,\mathrm{km}$ model top. We used the NDC on G6 with 10 vertical layers and reduced-radius spacing of approximately $0.96\,\mathrm{km}$.

Panels (a) and (b) of Fig. 7 show longitude–η sections at $1\,\mathrm{h}$, sampled at $0.5^\circ\,\mathrm{N}$. The perturbations $\theta' = \theta - \theta_b$ and $u' = u - u_b$ are departures from the unperturbed background, denoted by subscript b. The initially localized thermal disturbance developed into a wave train spanning a broad longitude range. Broad temperature perturbations occurred on either side of the wave

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

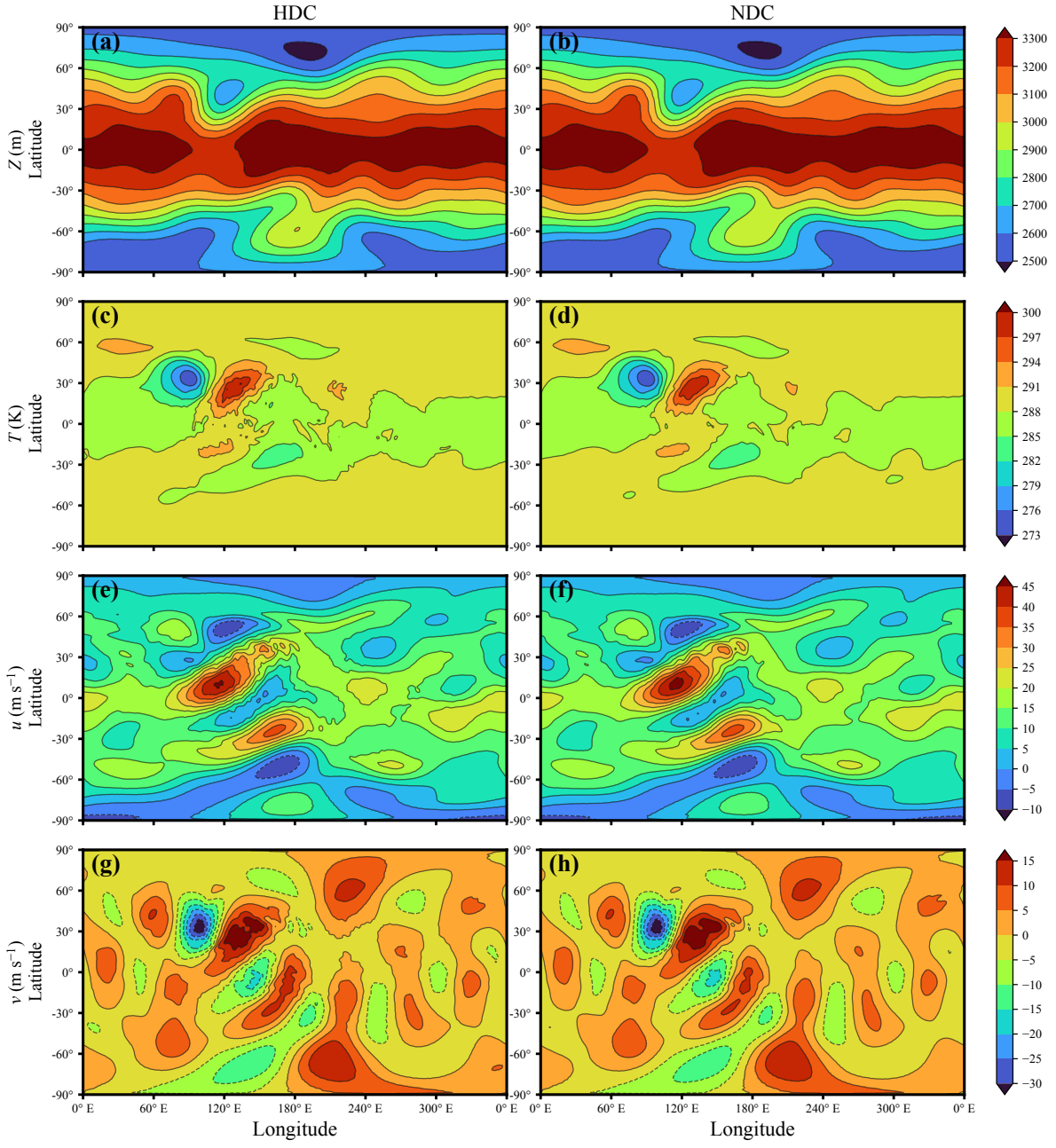

Figure 5. Mountain-induced Rossby waves at 700 hPa on day 25, with HDC in the left column and NDC in the right. Panels (a, b) show geopotential height $Z = \phi/g$ (m), (c, d) temperature T (K), and (e, f) and (g, h) zonal wind u and meridional wind v ($\mathrm{m\,s^{-1}}$), respectively. Both columns use the same color scale within each row.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

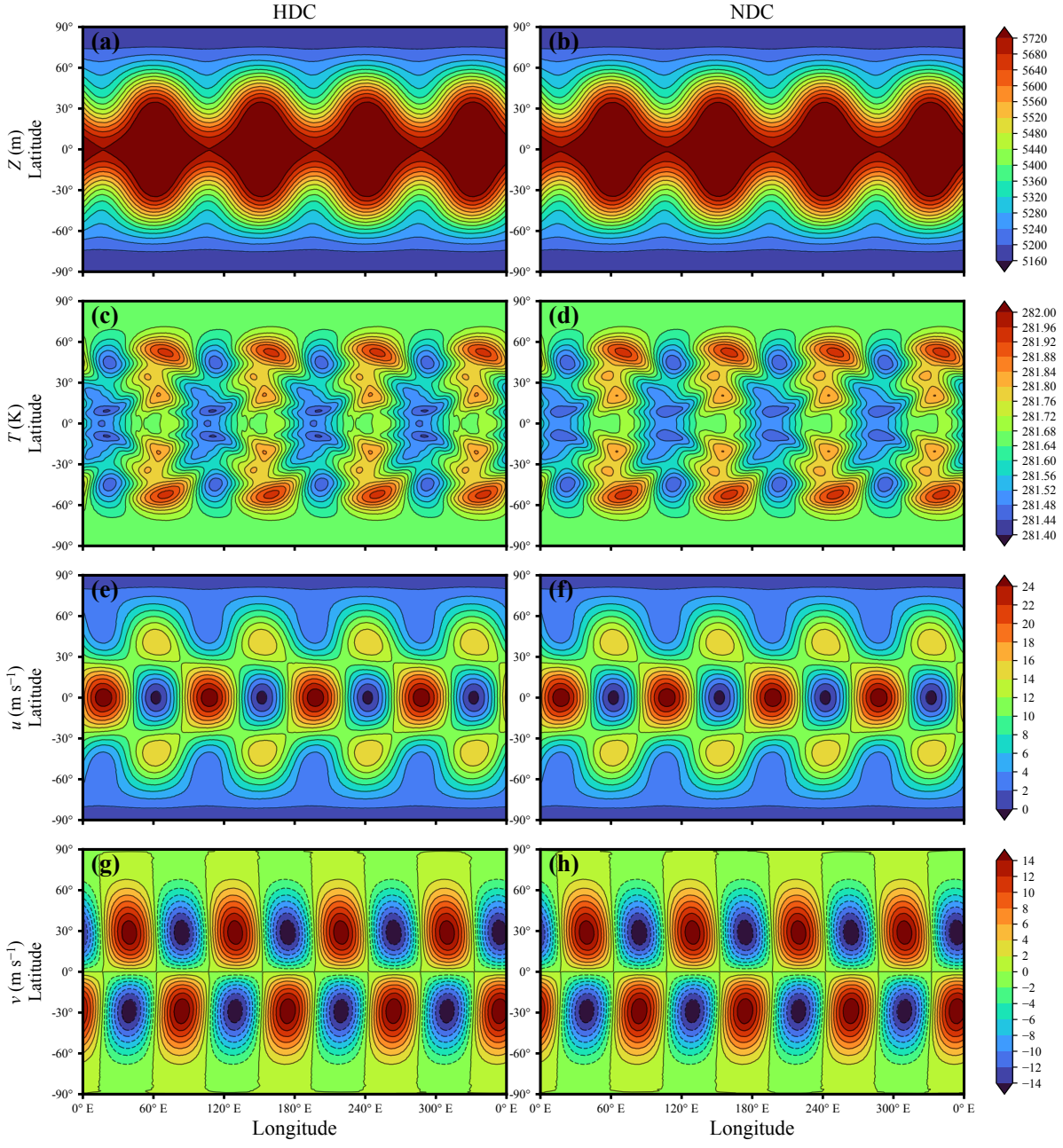

Figure 6. Three-dimensional Rossby–Haurwitz wave on day 15, with HDC in the left column and NDC in the right. Panels (a, b) show 500 hPa geopotential height $Z = \phi/g$ (m), (c, d) 850 hPa temperature T (K), and (e, f) and (g, h) 850 hPa zonal wind u and meridional wind v ($\mathrm{m\,s^{-1}}$), respectively. Color scales are shared within each row.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

train, with shorter-wavelength structures near its center. Zonal-wind perturbations alternated at corresponding longitudes, with approximately opposite signs in upper and lower layers. Advection by the background westerlies gave eastward- and westward-propagating components different ground-relative phase speeds and displaced the wave train eastward, breaking mirror symmetry about the initial longitude. The simulated gravity-wave structure was qualitatively consistent with the comparable-resolution results in Fig. 11 of Zhang et al. (2019).

4.5 Mountain waves forced by a quasi-two-dimensional ridge

The final test moves the wave source to the lower boundary, using the quasi-two-dimensional ridge in Sect. 2 of Klemp et al. (2015). The radius-reduction factor is $X = 166.7$, and planetary rotation is zero. The initial atmosphere is isothermal at $300\,\mathrm{K}$, with vertically uniform zonal wind $u = 20\cos\varphi\,\mathrm{m\,s^{-1}}$ and pressure satisfying the corresponding hydrostatic and gradient-wind balances. The ridge extends along longitude $180°$. Its equatorial profile is a squared cosine with wavelength parameter $\ell_0 = 4\,\mathrm{km}$, modulated by a Gaussian envelope with width parameter $d_0 = 5\,\mathrm{km}$, as in Eq. (9) of that study. Ridge height and physical horizontal scales decrease with latitude as $\cos\varphi$, making equatorial flow approximately two dimensional. We used the small-amplitude configuration with a peak ridge height of $h_0 = 25\,\mathrm{m}$, G6, and 40 vertical layers, giving physical horizontal spacing of approximately $0.72\,\mathrm{km}$.

This test used the pressure filter with dimensionless coefficient $\beta_p = 0.12$ and implicit Rayleigh damping of vertical velocity within $10\,\mathrm{km}$ of the local model top. The damping rate increases upward from zero at the base of the absorbing layer with a sine-squared profile, reaching $0.2\,\mathrm{s^{-1}}$. It reduces reflection of upward-propagating gravity waves at the top. Horizontal divergence damping and Smagorinsky diffusion were disabled. Pressure filtering and upper-level wave absorption are separate damping treatments in this test.

The vertical-velocity section at $0.5°\,\mathrm{N}$ after $2\,\mathrm{h}$ (Fig. 7c) reproduces the main wave structures in the small-amplitude simulation obtained with the Model for Prediction Across Scales (MPAS), shown in Fig. 2a of Klemp et al. (2015). Short waves forced by the narrow ridges were confined to the lower levels and decayed with height, while longer waves forced by the broad terrain envelope propagated vertically. As in the reference simulation, the main ascent and descent regions tilted upstream with height, and weaker perturbations occurred downstream of the main wave train. The agreement in wave structure and spatial distribution over the plotted 0–$10\,\mathrm{km}$ range indicates that the NDC captures the principal nonhydrostatic response to the multiscale terrain forcing.

5 Gradient verification

We examined the local consistency of AD derivatives in a two-parameter experiment based on DCMIP 3-1 using centered finite differences and Taylor remainder tests. We then used the loss gradients to recover prescribed initial potential-temperature amplitudes from synthetic observations.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

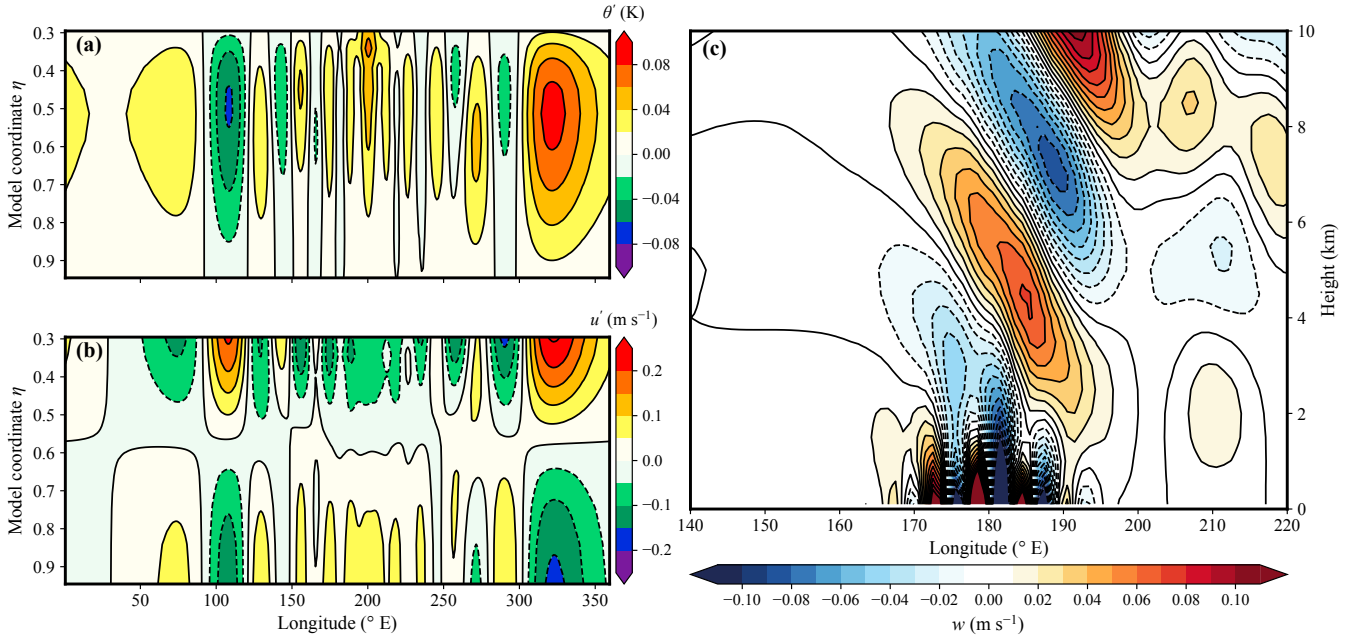

Figure 7. NDC results for the nonhydrostatic gravity-wave and mountain-wave tests. Panels (a, b) show DCMIP 3-1 at 1 h: (a) potential-temperature perturbation θ' (K) and (b) zonal-wind perturbation u' ($\mathrm{m\,s^{-1}}$), both relative to the unperturbed background. Their vertical axes show the dimensionless model coordinate $\eta = \eta_A + \eta_B$, increasing downward. Panel (c) shows vertical velocity w ($\mathrm{m\,s^{-1}}$) for the quasi-two-dimensional ridge test at 2 h, with geometric height on the right-hand axis over 0–10 km. The ridge is centered at 180° with a peak height of 25 m. In panel (c), warm and cool colors indicate ascent and descent, respectively. All sections are at 0.5° N, the available latitude nearest the Equator, with longitude in degrees east on the horizontal axes. Each panel has its own color scale.

5.1 Experimental setup and loss function

For gradient verification, we adapted the DCMIP 3-1 nonhydrostatic gravity-wave test described in Sect. 4.4 by adding two controlled potential-temperature perturbations and a passive tracer. The reference initialization followed DCMIP 3-1 for all dynamical fields, including wind and pressure.

All integrations used dry dynamics, with $\theta_m = \theta$ and $\mathcal{L} = 1$, and retained the horizontal mesh, vertical levels, planetary radius, rotation, and terrain of the DCMIP 3-1 configuration in Sect. 4.4. We integrated for 60 s using 2 s dynamical and advection time steps, with output every 10 s and all additional dissipation and damping disabled. We tested derivatives in four configurations, combining JAX or PyTorch with FCT enabled or disabled. All derivative tests used double precision on a single CPU process, with checkpointing for reverse-mode AD in both backends.

To parameterize the additional thermal perturbations, we introduced two control amplitudes $\boldsymbol{a} = (a_1, a_2)^{\mathrm{T}}$, in kelvin, such that

$$\theta_0(\boldsymbol{a}) = \theta_{\mathrm{ref}} + a_1\psi_1 + a_2\psi_2, \tag{50}$$

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

where θ_{ref} includes the original DCMIP 3-1 thermal perturbation and differs from the unperturbed background θ_b. The dimensionless basis functions ψ_i have fixed spatial shapes. When the controls change, temperature, hydrostatically balanced initial geopotential height, and dry-air density are recomputed from the modified potential-temperature field. These initialization steps are included in the AD calculation. To examine how the perturbed flow affects transport, we included a positive, spatially varying passive tracer in the derivative tests with and without FCT. Its initial mixing ratio is independent of the controls, and it contributes neither water loading nor thermodynamic feedback. Its sensitivity therefore arises only through the perturbed flow and subsequent transport. Appendix B specifies the perturbation basis functions and tracer initialization.

To assess how these initial perturbations affect the simulated fields, we formed the output vector $F(\boldsymbol{a})$ by concatenating the full fields of θ, u_n, w, and q_{dye} at the six output times. We compared outputs at fixed mesh and vertical indices, without interpolating to common geometric heights as the controls changed.

For derivative verification, we used $\boldsymbol{a}_0 = (-0.4, 0.5)^{\mathrm{T}}\,\mathrm{K}$ as the common evaluation point and applied directional increments $\Delta\boldsymbol{a} = \epsilon\boldsymbol{d}$, where $\boldsymbol{d} = (0.6, -0.8)^{\mathrm{T}}$ is a dimensionless unit vector and ϵ is measured in kelvin. Field-output derivatives were checked directly against finite-difference estimates without an observation target. For the loss-derivative and Taylor tests, we additionally generated noise-free observations at the prescribed synthetic truth $\boldsymbol{a}^* = (0.8, -0.6)^{\mathrm{T}}\,\mathrm{K}$, providing a fixed loss target for each configuration. The subsequent two-parameter inversion used the same target to recover $\boldsymbol{a}^*$, starting from $\boldsymbol{a}_0$ as the initial guess. Figure 8 shows the initial potential-temperature perturbations relative to θ_{ref} for the synthetic truth and the derivative-test point, together with the field increment produced by a 1 K perturbation along $\boldsymbol{d}$.

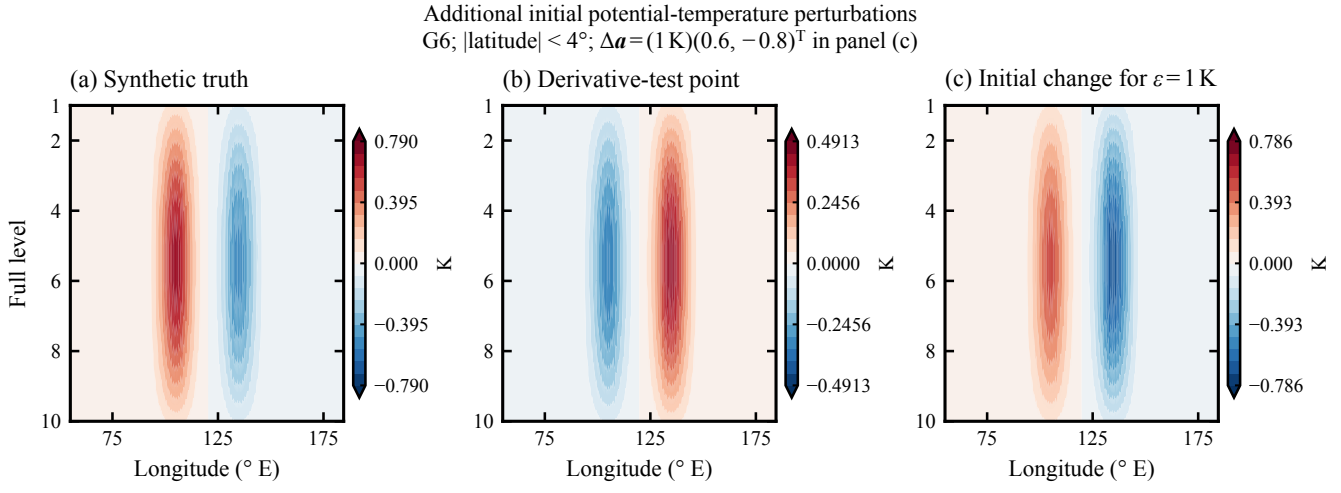

Figure 8. Additional initial potential-temperature perturbations on G6. Panels (a) and (b) show $\theta_0 - \theta_{\mathrm{ref}}$ for the synthetic truth $\boldsymbol{a}^* = (0.8, -0.6)^{\mathrm{T}}\,\mathrm{K}$ and derivative-test point $\boldsymbol{a}_0 = (-0.4, 0.5)^{\mathrm{T}}\,\mathrm{K}$, respectively. Panel (c) shows $\Delta\theta_0 = (1\,\mathrm{K})(0.6\psi_1 - 0.8\psi_2)$ for a control increment $\Delta\boldsymbol{a} = (1\,\mathrm{K})\boldsymbol{d}$, with $\boldsymbol{d} = (0.6, -0.8)^{\mathrm{T}}$. Mesh data within $|\varphi| < 4^\circ$ are displayed by triangular interpolation against longitude and full-level index. Indices 1–10 increase downward from the model top toward the surface. Each panel uses an independent color range, in kelvin.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

For each configuration, the loss compares the model output $F(\boldsymbol{a})$ with the fixed synthetic observations $F(\boldsymbol{a}^*)$:

$$L(\boldsymbol{a}) = \frac{1}{2}\boldsymbol{r}(\boldsymbol{a})^{\mathrm{T}}\boldsymbol{W}\boldsymbol{r}(\boldsymbol{a}), \qquad \boldsymbol{r}(\boldsymbol{a}) = F(\boldsymbol{a}) - F(\boldsymbol{a}^*), \tag{51}$$

where the diagonal matrix $\boldsymbol{W}$ defines a dimensionless least-squares loss with spatial weighting based on dry-air mass in the reference initial state. Residuals are normalized by prescribed variable scales, and the loss is averaged equally over variables and output times. The synthetic observations $F(\boldsymbol{a}^*)$ and the weight matrix $\boldsymbol{W}$ are held fixed when differentiating L with respect to $\boldsymbol{a}$. Details of the weight construction are provided in Appendix B.

5.2 Finite-difference and Taylor remainder tests

At $\boldsymbol{a}_0$, we compared AD directional derivatives of the model outputs and the loss with centered finite-difference estimates in all four configurations. Let $\boldsymbol{J} = F'(\boldsymbol{a}_0)$ and $\boldsymbol{g} = \nabla_{\boldsymbol{a}} L(\boldsymbol{a}_0)$. Forward-mode AD computes the Jacobian–vector product (JVP) $\boldsymbol{J}\boldsymbol{d}$ for the model outputs, while reverse-mode AD computes the loss gradient $\boldsymbol{g}$. The corresponding loss directional derivative is $\boldsymbol{g}^{\mathrm{T}}\boldsymbol{d}$. With the target and weights fixed, $\boldsymbol{g} = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{r}(\boldsymbol{a}_0)$. Since L is dimensionless and the control amplitudes are measured in kelvin, the loss-gradient components and the directional derivative $\boldsymbol{g}^{\mathrm{T}}\boldsymbol{d}$ have units of inverse kelvin.

For either the vector mapping F or scalar loss L, define the centered finite-difference (FD) operator

$$D_\epsilon H = \frac{H(\boldsymbol{a}_0 + \epsilon\boldsymbol{d}) - H(\boldsymbol{a}_0 - \epsilon\boldsymbol{d})}{2\epsilon}, \qquad H = F \text{ or } L, \tag{52}$$

where ϵ ranged from $10^{-7}\,\mathrm{K}$ to $1\,\mathrm{K}$. For each variable χ, let F_χ collect its outputs at all spatial points and six output times, and define $\boldsymbol{J}_\chi = F'_\chi(\boldsymbol{a}_0)$. The corresponding diagonal block $\boldsymbol{W}_\chi$ of $\boldsymbol{W}$ supplies the weights for the field-derivative comparison. The diagonal matrix $\boldsymbol{W}_\chi^{1/2}$ contains the positive square roots of the weights, giving $\|\boldsymbol{W}_\chi^{1/2}\boldsymbol{y}\|_2 = (\boldsymbol{y}^{\mathrm{T}}\boldsymbol{W}_\chi\boldsymbol{y})^{1/2}$, where $\|\cdot\|_2$ denotes the Euclidean norm. The relative errors in the field and loss directional derivatives are

$$E_\chi(\epsilon) = \frac{\left\|\boldsymbol{W}_\chi^{1/2}\left(D_\epsilon F_\chi - \boldsymbol{J}_\chi\boldsymbol{d}\right)\right\|_2}{\left\|\boldsymbol{W}_\chi^{1/2}\boldsymbol{J}_\chi\boldsymbol{d}\right\|_2}, \qquad E_L(\epsilon) = \frac{|D_\epsilon L - \boldsymbol{g}^{\mathrm{T}}\boldsymbol{d}|}{|\boldsymbol{g}^{\mathrm{T}}\boldsymbol{d}|}. \tag{53}$$

We used $E_\chi < 10^{-4}$ as the criterion for agreement between AD and finite-difference field derivatives.

Figure 9 shows how the AD–FD relative errors vary with perturbation amplitude ϵ. Where the model outputs or loss vary sufficiently smoothly with the controls, the centered finite-difference truncation error scales as ϵ^2. At very small amplitudes, subtracting nearly equal values makes the finite-difference estimates sensitive to round-off, so further reductions in ϵ can increase the relative error. The balance between truncation and round-off errors can therefore produce minima at intermediate perturbation amplitudes.

Across the four configurations, the minimum relative error was approximately 2.9×10^{-9} for the potential-temperature JVP and of order 10^{-8} for both wind variables. The minimum loss directional-derivative error ranged from 1.23×10^{-11} to 1.70×10^{-11}. Enabling FCT altered the tracer evolution, producing a maximum absolute mixing-ratio difference of approximately 2.02×10^{-5} relative to the run without FCT. Tracer JVPs satisfied the 10^{-4} criterion at suitable intermediate perturbation amplitudes, with minimum relative errors of approximately 9.5×10^{-7}–9.6×10^{-7} without FCT and 6.8×10^{-6}–7.0×10^{-6}

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

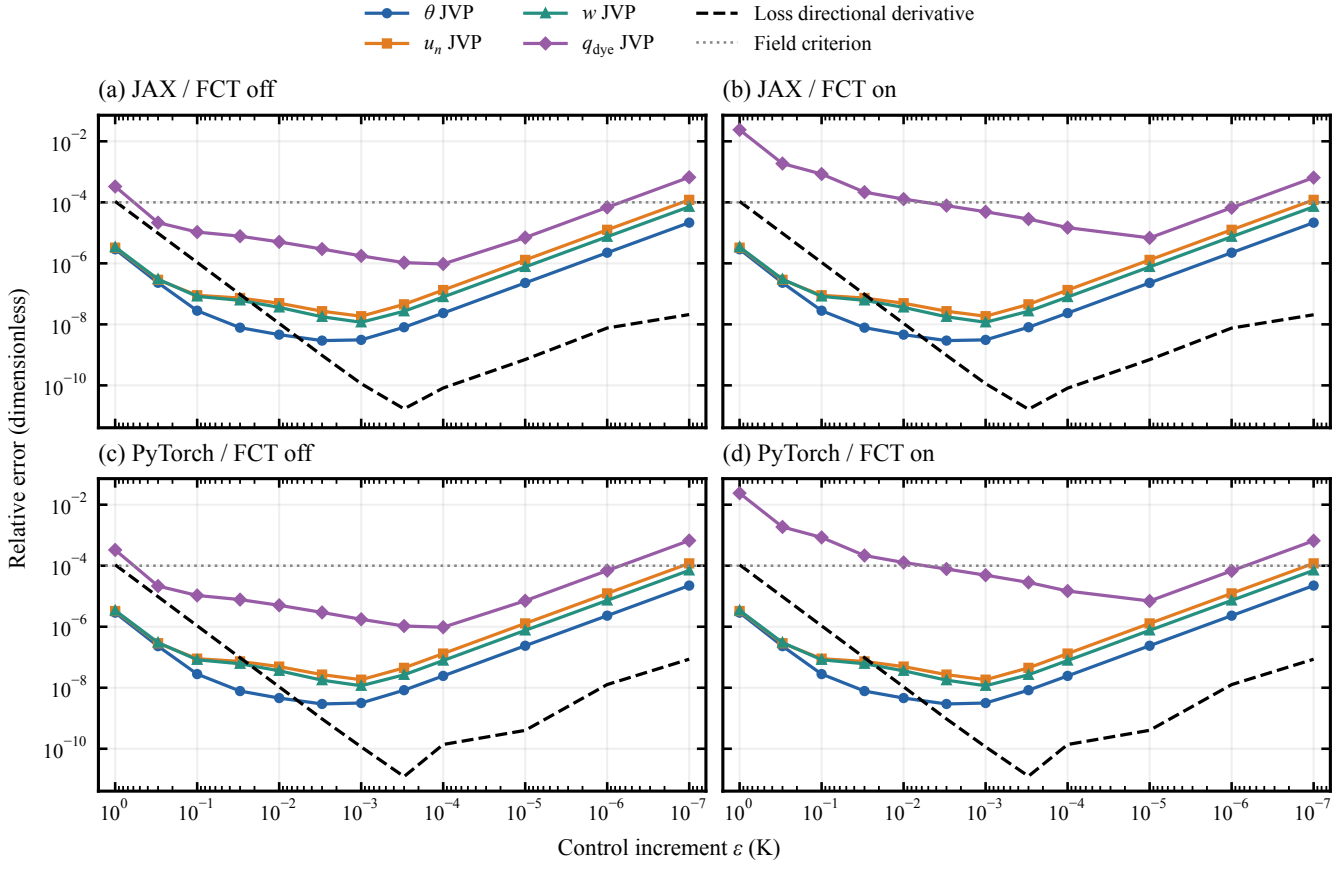

Figure 9. AD–FD relative errors at $\boldsymbol{a}_0$ along $\boldsymbol{d} = (0.6, -0.8)^{\mathrm{T}}$. Panels (a)–(d): JAX/FCT off, JAX/FCT on, PyTorch/FCT off, and PyTorch/FCT on. Colored curves and black dashed curves show the relative errors for the field and loss directional derivatives, E_χ and E_L, respectively (Eq. (53)). The field errors are evaluated for θ, u_n, w, and q_{dye} over all spatial points and six output times. Gray dotted lines mark the 10^{-4} field criterion. Dimensionless errors are plotted against perturbation amplitude ϵ (K) on logarithmic axes, with ϵ decreasing from left to right.

with FCT. With FCT enabled, the minimum occurred at a smaller perturbation amplitude, while the error at $\epsilon = 10^{-2}$ K was approximately 1.28×10^{-4}, slightly above the criterion. These results support local consistency between AD and finite-difference directional derivatives at $\boldsymbol{a}_0$ over suitable perturbation ranges in all four configurations.

To complement the finite-difference comparisons, we used a Taylor remainder test to assess whether subtracting the first-order loss change predicted by AD leaves a second-order remainder. We defined R_0 as the magnitude of the loss change and R_1 as the magnitude of the remainder after subtracting this linear prediction:

$$R_0(\epsilon) = |L(\boldsymbol{a}_0 + \epsilon \boldsymbol{d}) - L(\boldsymbol{a}_0)|, \quad R_1(\epsilon) = \left|L(\boldsymbol{a}_0 + \epsilon \boldsymbol{d}) - L(\boldsymbol{a}_0) - \epsilon \boldsymbol{g}^{\mathrm{T}} \boldsymbol{d}\right|. \tag{54}$$

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

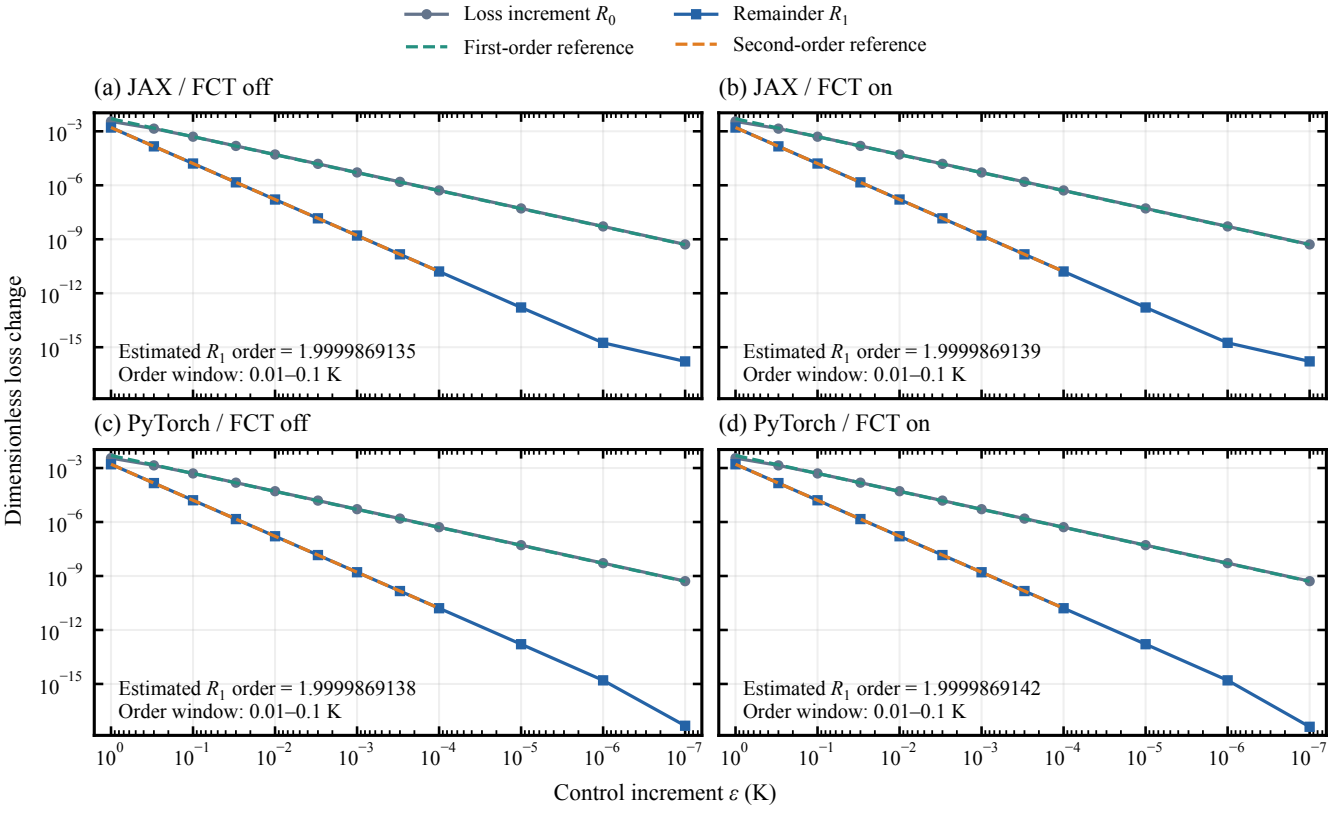

Figure 10. Loss Taylor test at $\boldsymbol{a}_0$ along $\boldsymbol{d} = (0.6, -0.8)^{\mathrm{T}}$. Panel configurations are the same as in Fig. 9. Gray circles show the absolute loss change R_0, and blue squares show the magnitude of the remainder R_1 after subtracting the AD linear term (Eq. (54)). Green and orange dashed reference lines scale as ϵ and ϵ^2, anchored to the respective curves at 0.1 K. The dimensionless quantities R_0 and R_1 are plotted against perturbation amplitude ϵ (K) on logarithmic axes, with ϵ decreasing from left to right. For each configuration, the annotated convergence order is the median of the logarithmic slopes of R_1 between successive points at $\epsilon = 0.1$, 0.03, and 0.01 K, rounded to 10 decimal places.

For a loss with a bounded second derivative along $\boldsymbol{d}$ near $\boldsymbol{a}_0$, subtracting the correct linear term gives $R_1 = O((\epsilon/(1\,\mathrm{K}))^2)$. Since $\boldsymbol{g}^{\mathrm{T}}\boldsymbol{d} \neq 0$, R_0 has a nonzero first-order term. Consistent with these expectations, R_0 approached first-order scaling as ϵ decreased (Fig. 10). Across the three sampled amplitudes $\epsilon = 0.1$, 0.03, and 0.01 K, R_1 exhibited second-order scaling in all four configurations. We estimated the convergence order from these three amplitudes and excluded the smallest perturbations affected by round-off.

As a complementary check of consistency between AD modes, we applied a dot-product test relating the forward-mode JVP to the reverse-mode vector–Jacobian product (VJP). At $\boldsymbol{a}_0$, relative errors ranged from 2.56×10^{-15} to 1.62×10^{-14} across all four configurations, indicating agreement within floating-point round-off for the tested direction. The test formulation and detailed results are given in Appendix C.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

5.3 Two-parameter inversion

Following derivative verification, we used the JAX backend with FCT enabled to recover the two control amplitudes from the noise-free observations generated at $\boldsymbol{a}^*$. Starting from $\boldsymbol{a}_0$ and keeping all other model settings fixed, we minimized the loss in Eq. (51) by gradient descent using reverse-mode AD gradients and an Armijo backtracking line search:

$$\boldsymbol{a}^{(m+1)} = \boldsymbol{a}^{(m)} - \alpha_m \nabla_{\boldsymbol{a}} L(\boldsymbol{a}^{(m)}), \qquad \boldsymbol{a}^{(0)} = \boldsymbol{a}_0, \tag{55}$$

where $m = 0, 1, 2, \ldots$ denotes the iteration index. Each line search was initialized with the same trial step size, $\alpha_{\mathrm{trial}} = 1/\lambda_{\max}(\boldsymbol{B}_0)$, where $\boldsymbol{B}_0 = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{J}$ is the Gauss–Newton matrix evaluated at $\boldsymbol{a}_0$ and $\lambda_{\max}(\boldsymbol{B}_0)$ denotes its largest eigenvalue. Backtracking reduces the trial step size, if necessary, until the Armijo condition is satisfied. The accepted step size α_m has units of K^2. The gradient-based stopping criterion was $\|\nabla_{\boldsymbol{a}} L\|_2 < 10^{-10}\,\mathrm{K}^{-1}$. We monitored convergence using the total loss $L = \sum_\chi L_\chi$ and the individual variable contributions $L_\chi = \frac{1}{2}\boldsymbol{r}_\chi^{\mathrm{T}}\boldsymbol{W}_\chi\boldsymbol{r}_\chi$, where $\boldsymbol{r}_\chi$ contains the residuals for variable χ at all spatial points and output times.

Most of the adjustment toward the synthetic truth occurred within the first few updates (Fig. 11a, b). After six updates, the inversion satisfied the gradient-based stopping criterion, with a Euclidean parameter error of approximately $1.83 \times 10^{-9}\,\mathrm{K}$. During the inversion, the total loss and all four variable contributions decreased monotonically, with L reaching approximately 5.43×10^{-21} (Fig. 11c). Under the prescribed weighting, the vertical-velocity contribution L_w accounted for approximately $89.9\,\%$ of the total loss throughout the inversion, explaining the near overlap of the L and L_w curves. Although the tracer contribution accounted for only approximately 2.7×10^{-6} of the initial total loss, it decreased by approximately 18 orders of magnitude, from 1.16×10^{-8} to 1.47×10^{-26}.

6 Performance

We benchmarked forward integration using JAX with functional updates and PyTorch with either in-place or functional updates, as described in Sect. 3.4. The benchmarks used the hydrostatic mountain-induced Rossby-wave case with 26 vertical layers on successively refined G4–G7 meshes. All simulations ran in double precision with compilation enabled and dynamical and advection time steps of $50\,\mathrm{s}$. Each implementation was evaluated on a single NVIDIA RTX 4090 GPU and on 10 CPU nodes, with one MPI process and 256 CPU cores per node (2560 cores in total). We measured wall-clock time per step, excluding compilation, model construction, and history-file output. The GPU timer was stopped only after device synchronization, and CPU timings included MPI communication. For each mesh, backend, and update mode, the GPU speedup is defined as $S = T_{\mathrm{CPU}}/T_{\mathrm{GPU}}$, where T_{CPU} and T_{GPU} denote the corresponding times per step.

Figure 12 shows that JAX achieved the shortest time per step on both hardware configurations across all tested meshes, followed by PyTorch with in-place updates and then PyTorch with functional updates. As the mesh was refined from G4 to G7, the JAX time per step increased from 9.313 to $108.928\,\mathrm{ms}$ on the GPU and from 28.961 to $331.720\,\mathrm{ms}$ on the CPU configuration. Within the PyTorch backend, functional updates required approximately $25\,\%$–$54\,\%$ more time per step on

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

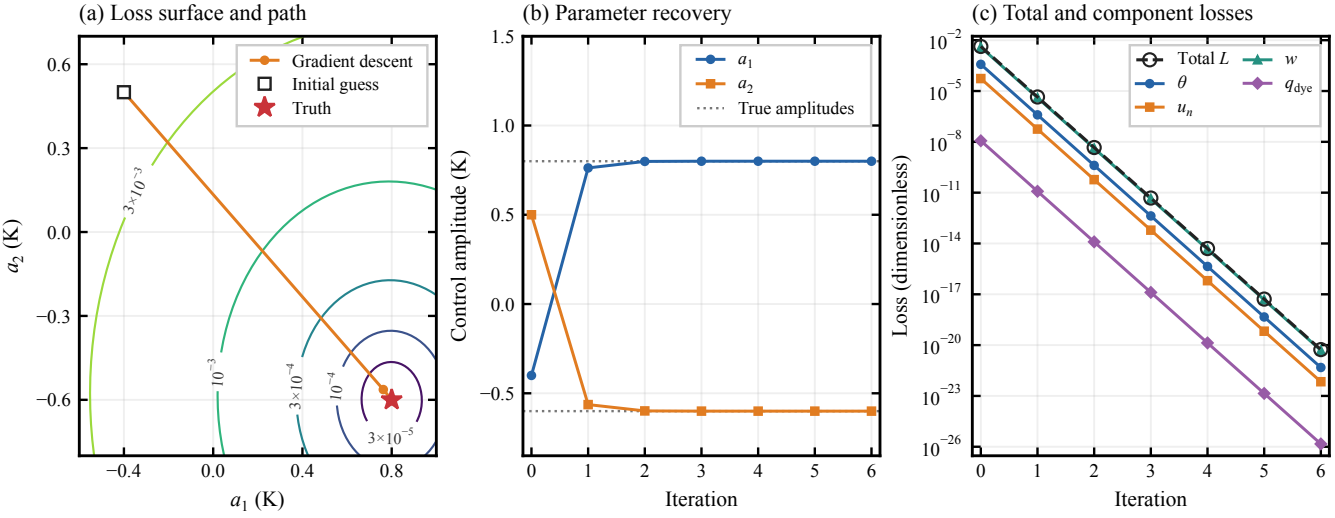

Figure 11. Two-parameter inversion with JAX and FCT. (a) The orange gradient-descent trajectory is overlaid on contours of the dimensionless loss L, obtained by bicubic interpolation of values sampled on a parameter grid. The open square and red star mark $\boldsymbol{a}_0$ and $\boldsymbol{a}^*$, respectively. Both control axes are in kelvin. (b) Amplitudes a_1 (blue circles) and a_2 (orange squares) in kelvin as functions of iteration, with dotted lines marking their synthetic-truth values. (c) Total loss L (black dashed line, open circles) and four contributions L_χ (colored lines), with loss shown on a logarithmic scale. Iteration 0 corresponds to the initial guess $\boldsymbol{a}_0$.

the GPU and 9 %–23 % more on the CPU configuration than in-place updates. Thus, although functional updates support differentiation, they incurred an additional cost for forward integration in the present PyTorch implementation.

Relative to the corresponding CPU runs, GPU speedups ranged from approximately 2.3 to 7.0, as shown by the bars in Fig. 12. For all three implementations, speedup increased from G4 to G6, reaching approximately 5.7, 7.0, and 5.5 for JAX, PyTorch with in-place updates, and PyTorch with functional updates, respectively. PyTorch with in-place updates achieved the largest speedup on G6, whereas JAX retained the shortest absolute GPU time per step. This difference in ranking reflects the use of each implementation's own CPU runtime as the speedup baseline. The increase in speedup from G4 to G6 is consistent with improved GPU utilization as the problem size grows. From G6 to G7, however, GPU time per step increased by factors of 3.9–4.5, compared with 1.8–2.4 on the CPU configuration, reducing the speedup for all three implementations. This decline may indicate that GPU throughput is approaching saturation.

Differences in how state management and communication interact with compilation may help explain the shorter JAX runtimes. In the current PyTorch implementation, PyTree state management occurs outside the compiled regions, as does MPI communication during distributed runs. This separation may limit optimization across region boundaries. In JAX, native PyTree support allows model states to be traced during JIT compilation, while `mpi4jax` allows MPI communication to be included in the compiled time-stepping function. The present benchmarks do not isolate the contributions of these implementation

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

differences. Future work will use kernel-level profiling to identify the bottlenecks underlying the observed performance trends and guide further optimization.

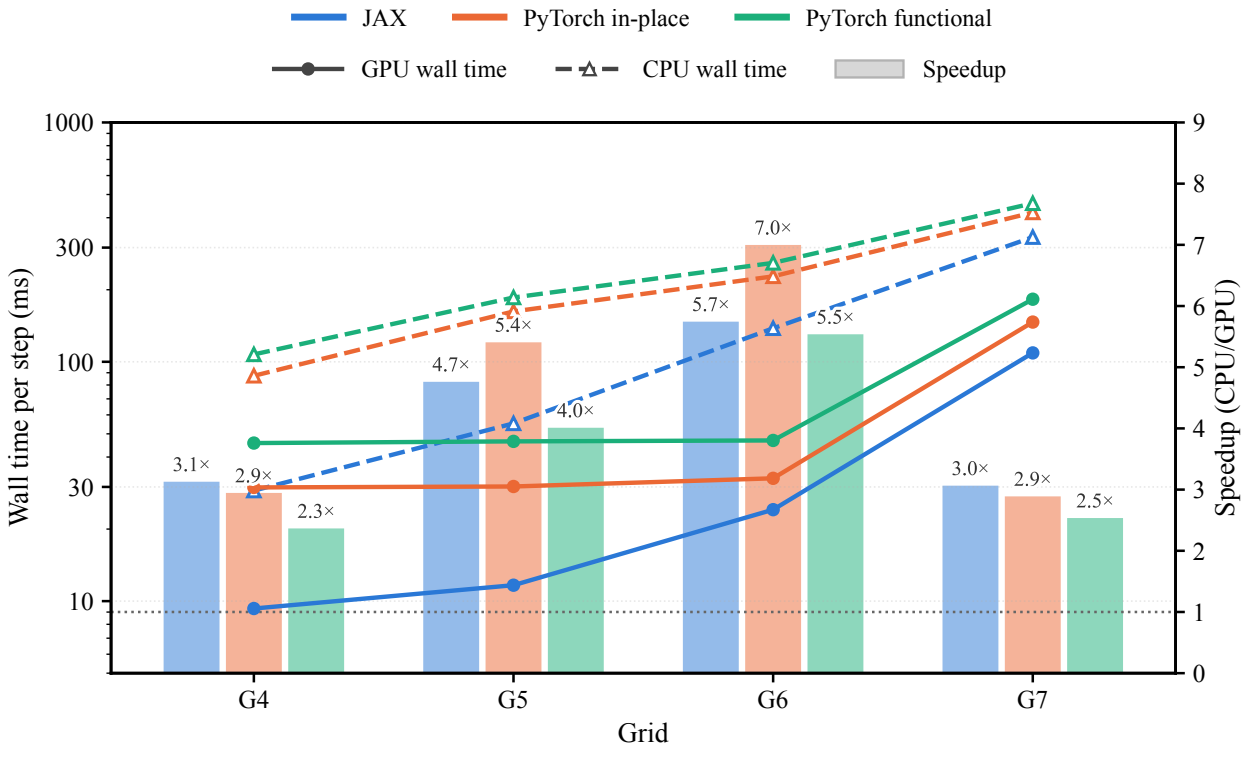

Figure 12. Double-precision hydrostatic forward-integration performance on G4–G7 (26 layers; 50 s dynamical and advection steps). Solid lines with filled circles and dashed lines with open triangles show time per step (ms; logarithmic left axis) on one NVIDIA RTX 4090 and 10 CPU nodes, respectively, with one MPI process and 256 cores per CPU node. Bars show GPU speedup $T_{\mathrm{CPU}}/T_{\mathrm{GPU}}$ on the linear right axis, with a dotted reference line at $T_{\mathrm{CPU}}/T_{\mathrm{GPU}} = 1$. Colors identify implementations. Compilation was enabled for all implementations. Reported timings exclude compilation, model construction, and output. CPU timings included MPI communication, and GPU timers were stopped only after device synchronization.

7 Conclusions

TerraFlow (version 1.0) provides a differentiable, multi-backend nonhydrostatic atmospheric dynamical core on unstructured spherical meshes. By combining a common array and state-update interface with a separate MPI communication layer, TerraFlow supports automatic differentiation, just-in-time compilation, GPU execution, and distributed forward integration within a shared implementation for JAX and PyTorch.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

In the idealized forward tests, global L_1, L_2, and L_∞ tracer errors decreased under joint horizontal, vertical, and temporal refinement in DCMIP 1-2, and the tracer layer largely returned to its initial height range after a full transport cycle. Both the HDC and NDC reproduced the principal geopotential-height, temperature, and wind patterns of the mountain-induced Rossby-wave reference solution, as well as the four-wave structure and equatorial symmetry properties of the three-dimensional Rossby–Haurwitz wave. At nonhydrostatic scales, the NDC captured the dispersive gravity-wave response to a localized thermal perturbation and its eastward displacement by the background flow. Over the multiscale ridge, it reproduced short waves that decayed with height near the surface and longer waves that propagated vertically, with the main ascent and descent regions tilting upstream as in the reference simulation.

Complementary derivative tests in short integrations showed agreement between AD and centered finite-difference directional derivatives in both backends, with relative errors below 10^{-4} for all output fields over suitable perturbation ranges. The loss Taylor remainders exhibited second-order convergence. Dot-product tests and direct backend comparisons also supported consistency between differentiation modes and backend implementations for the tested inputs and direction. Using these verified derivatives, we performed a two-parameter inversion with noise-free synthetic observations generated by the same model configuration. Gradient descent recovered the prescribed initial potential-temperature amplitudes in six updates, with a Euclidean parameter error of approximately $1.83 \times 10^{-9}\,\mathrm{K}$.

In double-precision hydrostatic benchmarks on G4–G7, a single NVIDIA RTX 4090 achieved forward-integration speedups of approximately 2.3–7.0 relative to the corresponding 10-node CPU runs (2560 cores), excluding compilation, model construction, and output. JAX achieved the shortest time per step on both hardware configurations across all tested meshes. Within PyTorch, functional updates incurred additional forward-integration cost relative to in-place updates.

Taken together, these results demonstrate TerraFlow's ability to support accurate and efficient atmospheric simulation, consistent local gradient calculations, and parameter estimation in a controlled inversion. Future work will extend gradient verification to longer integrations and higher-dimensional control spaces, refine checkpointing strategies to balance memory use and recomputation costs during reverse-mode differentiation, and explore coupling to physical parameterizations and machine-learning components.

Appendix A: Symbols and units

Tables A1 and A2 collect the symbols and units used in the model equations and variable arrangement.

Appendix B: Details of the gradient verification experiments

This appendix supplements the experimental setup in Sect. 5.1 by providing explicit definitions of the potential-temperature perturbation basis functions, the initial passive-tracer field, and the loss weights.

All experiments used a G6 mesh with 40962 cells and 122880 edges. The fixed basis functions in Eq. (50) are

$$\psi_i(\lambda, \varphi, z_{\mathrm{ref}}) = \exp\left(-\frac{r_i^2}{2\sigma_h^2}\right) \sin\left(\frac{2\pi z_{\mathrm{ref}}}{L_z}\right), \qquad i = 1, 2, \tag{B1}$$

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

Table A1. Principal symbols used in the governing equations and variable arrangement. A unit of 1 denotes a dimensionless quantity (including mass ratios written as $\mathrm{kg\,kg^{-1}}$), and $[\chi]$ denotes the units of χ.

Symbol	Definition	Unit
t	Time	s
η, η_t, η_s	Generalized vertical coordinate and its top and surface values	1
η_A, η_B	Hybrid-coordinate coefficients	1
$k,\ k \pm \frac{1}{2}$	Full- and half-level indices, increasing downward	1
π, π_t, π_s	Dry-mass coordinate variable and its top and surface values	Pa
$\mu, \delta\pi$	$\partial\pi/\partial\eta$; layer dry-mass thickness	Pa
$\dot{\eta}$	Generalized vertical velocity $D\eta/Dt$	$\mathrm{s^{-1}}$
F_η	Vertical dry-air mass flux $\mu\dot{\eta}$, in pressure units	$\mathrm{Pa\,s^{-1}}$
p, p_s	Actual pressure and its surface value	Pa
p_0	Reference pressure, 10^5 Pa	Pa
ρ_d, ρ_m	Dry-air density; total moist-air density including water species	$\mathrm{kg\,m^{-3}}$
α_d	Dry-air specific volume $1/\rho_d$	$\mathrm{m^3\,kg^{-1}}$
q_i	Dry-air mixing ratio of species i; also a passive scalar transported with dry-air mass weighting	1
q_v, q_c, q_r	Dry-air mixing ratios of water vapor, cloud water, and rainwater	$\mathrm{kg\,kg^{-1}}$
$\mathcal{L}$	Dry-air to total moist-air density ratio ρ_d/ρ_m	1
T, T_m, T_v	Temperature, modified temperature, and virtual temperature	K
θ, θ_m	Potential temperature and modified potential temperature	K
R_d, R_v	Gas constants of dry air and water vapor	$\mathrm{J\,kg^{-1}\,K^{-1}}$

Table A2. Principal symbols (continued from Table A1).

Symbol	Definition	Unit
c_{pd}, c_{vd}	Dry-air specific heats at constant pressure and volume	$\mathrm{J\,kg^{-1}\,K^{-1}}$
$\epsilon_v, \kappa, \gamma$	R_v/R_d, R_d/c_{pd}, and c_{pd}/c_{vd}	1
$\boldsymbol{V}, u_n, u_t$	Horizontal velocity and its edge-normal and tangential components	$\mathrm{m\,s^{-1}}$
u, v	Zonal and meridional wind	$\mathrm{m\,s^{-1}}$
$\boldsymbol{n}_e, \boldsymbol{t}_e, \boldsymbol{k}$	Edge-normal, edge-tangential, and local upward unit vectors	1
w	Geometric vertical velocity Dz/Dt, positive upward	$\mathrm{m\,s^{-1}}$
z, z_s	Geometric height and terrain height	m
ϕ, ϕ_s	Geopotential gz and surface geopotential gz_s	$\mathrm{m^2\,s^{-2}}$
Z	Geopotential height ϕ/g	m
KE	Horizontal kinetic energy per unit mass, $\lvert\boldsymbol{V}\rvert^2/2$	$\mathrm{m^2\,s^{-2}}$
ζ_r, ζ_a, f	Vertical relative and absolute vorticity; Coriolis parameter	$\mathrm{s^{-1}}$
ζ_p	Potential vorticity in the model, $\zeta_a/\delta\pi$	$\mathrm{Pa^{-1}\,s^{-1}}$
Ω	Planetary rotation rate	$\mathrm{s^{-1}}$
λ, φ	Longitude and latitude (degrees in geographic plots)	rad
g	Magnitude of gravitational acceleration	$\mathrm{m\,s^{-2}}$
$\nabla_\eta, \partial/\partial n$	Horizontal derivative along constant-η surfaces; edge-normal directional derivative	$\mathrm{m^{-1}}$
$\delta\chi$	Difference across a layer, lower minus upper interface	$[\chi]$
$S(\chi)$	Non-advective source or sink tendency of χ	$[\chi]\,\mathrm{s^{-1}}$

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

where r_i is the chord distance from (λ, φ) to perturbation center i on the reduced-radius sphere. The two perturbations are centered on the equator at $105^\circ\,\mathrm{E}$ and $135^\circ\,\mathrm{E}$, respectively, and share the horizontal width $\sigma_h = 5\,\mathrm{km}$ and vertical wavelength $L_z = 20\,\mathrm{km}$. The full-level height z_{ref} is diagnosed from the reference initial geopotential and held fixed as the controls change.

The passive tracer introduced in Sect. 5.1 is initialized with the dimensionless mixing ratio

$$q_{\mathrm{dye},0} = 0.3 + 0.2\cos\varphi\cos(\lambda - \lambda_d)\left[0.7 + 0.3\cos\left(\frac{\pi z_{\mathrm{ref}}}{L_z}\right)\right], \tag{B2}$$

where $\lambda_d = 2.0\,\mathrm{rad}$ and trigonometric arguments are in radians. This strictly positive field varies horizontally and vertically and is independent of the control amplitudes.

To construct the weight matrix $\boldsymbol{W}$ in Eq. (51), we first assigned spatial weights using the dry-air mass per unit area $\delta\pi/g$ in the reference initial state and the mesh geometry at each variable's location. Cell-centered variables use layer mass per unit area multiplied by cell area. Normal-wind weights use the mean layer mass per unit area of the two adjacent cells multiplied by the edge-diamond area. For w, the mass per unit area assigned to an interior interface is the mean of the adjacent layer values, while each boundary interface receives half the adjacent layer value. These values are multiplied by cell area to obtain the spatial weights. Let $\omega_{\chi,\ell}$ denote these spatial weights normalized to sum to one for each variable $\chi \in \{\theta, u_n, w, q_{\mathrm{dye}}\}$, with ℓ indexing its horizontal and vertical locations. These normalized spatial weights are combined with the variable scales and equal averaging over the four variables and $N_t = 6$ output times $t_j = 10j\,\mathrm{s}$ $(j = 1, \ldots, N_t)$ to form $\boldsymbol{W}$. For variable χ at output time t_j and spatial location ℓ, the corresponding diagonal entry of $\boldsymbol{W}$ is

$$W_{\chi j\ell} = \frac{\omega_{\chi,\ell}}{4N_t s_\chi^2}, \qquad \sum_\ell \omega_{\chi,\ell} = 1, \tag{B3}$$

where $s_\theta = 1\,\mathrm{K}$, $s_{u_n} = 1\,\mathrm{m\,s^{-1}}$, $s_w = 0.1\,\mathrm{m\,s^{-1}}$, and $s_{q_{\mathrm{dye}}} = 0.01$. Normalization by s_χ^2 makes each variable's contribution to L dimensionless. The resulting matrix $\boldsymbol{W}$ is held fixed as the control amplitudes vary. For each variable χ, the entries $W_{\chi j\ell}$ form the diagonal block $\boldsymbol{W}_\chi$ used in the weighted derivative-error norms in Eq. (53).

Appendix C: Forward–reverse and cross-backend derivative consistency

This appendix provides the formulation and detailed results of the forward–reverse dot-product test summarized in Sect. 5.2, together with an additional comparison of JAX and PyTorch derivatives under matching FCT settings. Both checks used the experimental setup and evaluation point $\boldsymbol{a}_0$ defined in Sect. 5.1, with FCT enabled and disabled. The JVPs were evaluated along the prescribed direction $\boldsymbol{d}$.

C1 Forward–reverse consistency

Let $\boldsymbol{J} = F'(\boldsymbol{a}_0)$. For the dot-product test, $\widehat{\boldsymbol{J}}$ and $\widehat{\boldsymbol{W}}$ denote the numerical representations of the Jacobian $\boldsymbol{J}$ and the loss-weight matrix $\boldsymbol{W}$, respectively. The control amplitudes and potential temperature are expressed in kelvin, and the velocity components are expressed in meters per second. The tracer mixing ratio is dimensionless. Using this numerical representation, we compared

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

Table C1. Relative errors in the JVP–VJP dot-product test and relative differences between JAX and PyTorch derivatives at $\boldsymbol{a}_0$. The dot-product test is performed separately for each backend, and cross-backend differences use JAX as the reference under matching FCT settings. Field-JVP comparisons include all spatial points and six output times, while the loss-gradient comparison includes both control components. All reported errors and relative differences are dimensionless and rounded to three significant figures.

Diagnostic	FCT off	FCT on
JVP–VJP dot-product error, E_{dot}		
JAX	2.56×10^{-15}	4.12×10^{-15}
PyTorch	1.62×10^{-14}	1.55×10^{-14}
JAX–PyTorch relative difference, E_{back}		
Potential-temperature JVP	7.36×10^{-15}	7.36×10^{-15}
Normal-wind JVP	8.87×10^{-14}	8.87×10^{-14}
Vertical-velocity JVP	2.47×10^{-13}	2.47×10^{-13}
Tracer JVP	3.00×10^{-13}	4.12×10^{-13}
Loss gradient	3.65×10^{-14}	3.65×10^{-14}

forward- and reverse-mode AD through the Euclidean identity

$$\langle \widehat{\boldsymbol{J}}\boldsymbol{d}, \boldsymbol{v} \rangle = \langle \boldsymbol{d}, \widehat{\boldsymbol{J}}^{\mathrm{T}} \boldsymbol{v} \rangle. \tag{C1}$$

Forward-mode AD supplies the Jacobian–vector product (JVP), $\widehat{\boldsymbol{J}}\boldsymbol{d}$, while reverse-mode AD supplies the vector–Jacobian product (VJP), written in column-vector form as $\widehat{\boldsymbol{J}}^{\mathrm{T}}\boldsymbol{v}$.

We constructed a test vector $\boldsymbol{v}$ with the same dimension as the model output $F(\boldsymbol{a}_0)$ from a vector $\boldsymbol{z}$ of independent standard-normal samples generated with a fixed random seed:

$$\boldsymbol{v} = \frac{\widehat{\boldsymbol{W}}^{1/2}\boldsymbol{z}}{\|\widehat{\boldsymbol{W}}^{1/2}\boldsymbol{z}\|_2}. \tag{C2}$$

The resulting test vector was held fixed during differentiation. The inner products in Eq. (C1) are evaluated on numerical arrays without additional weighting, with $\widehat{\boldsymbol{W}}$ entering only through the construction of $\boldsymbol{v}$ in Eq. (C2).

The two scalar products, $s_{\mathrm{f}} = \langle \widehat{\boldsymbol{J}}\boldsymbol{d}, \boldsymbol{v} \rangle$ and $s_{\mathrm{r}} = \langle \boldsymbol{d}, \widehat{\boldsymbol{J}}^{\mathrm{T}}\boldsymbol{v} \rangle$, then give the relative dot-product error

$$E_{\mathrm{dot}} = \frac{|s_{\mathrm{f}} - s_{\mathrm{r}}|}{\max(|s_{\mathrm{f}}|, |s_{\mathrm{r}}|, \epsilon_{\mathrm{floor}})}. \tag{C3}$$

Here $\epsilon_{\mathrm{floor}} > 0$ is a small denominator floor. Across the four configurations, the dot-product errors ranged from 2.56×10^{-15} to 1.62×10^{-14} (Table C1), indicating agreement within floating-point round-off for the tested inputs and direction.

C2 Cross-backend consistency

We next compared the JAX and PyTorch derivatives under matching FCT settings. Each comparison used either a field JVP concatenated over all spatial points and six output times or the two-component loss gradient. For the corresponding numerical

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

vector $\widehat{\boldsymbol{x}}$, the relative difference is

$$E_{\mathrm{back}}(\widehat{\boldsymbol{x}}) = \frac{\|\widehat{\boldsymbol{x}}_{\mathrm{PyTorch}} - \widehat{\boldsymbol{x}}_{\mathrm{JAX}}\|_2}{\max(\|\widehat{\boldsymbol{x}}_{\mathrm{JAX}}\|_2, \epsilon_{\mathrm{floor}})}. \tag{C4}$$

This measure uses an unweighted Euclidean norm with JAX as the reference. In all reported tests, the denominators exceeded $\epsilon_{\mathrm{floor}}$, so the floor did not affect the relative errors.

The resulting backend differences were at most 4.12×10^{-13} for the field JVPs and approximately 3.65×10^{-14} for the loss gradient (Table C1). The equal entries for the dynamical-field JVPs in Table C1 reflect the passive role of the tracer. In this dry experiment, FCT affects tracer transport without altering the dynamics, leaving the θ, u_n, and w JVPs unchanged within each backend. Although FCT changes the tracer contribution to the loss gradient, the cross-backend relative differences for the two FCT settings round to the same value at three significant figures. These implementation checks complement the finite-difference and Taylor remainder tests in Sect. 5.2, which assess whether the derivatives agree with local changes in the model outputs and loss.

Code and data availability. The TerraFlow source code is publicly available on GitHub (https://github.com/TiangeHuang-Cedric/terraflow.git). The code is distributed under the MIT License. The model version used in this study, together with the computational meshes, experiment configurations, and the scripts and notebooks needed to run the experiments, analyze the results, and generate the figures, is archived on Zenodo at https://doi.org/10.5281/zenodo.24191174 (Huang et al., 2026).

Author contributions. TH has developed TerraFlow's framework, implemented the model code, and conducted the numerical experiments. LD and ZX have provided advice on model development. ZX, BP and GL have provided resources and financial support. All authors have contributed to the writing and revision of the manuscript.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgments. This work was supported by Yunnan Fundamental Research Program (Grant No. 202501BC070001) and CAS Project for Young Scientists in Basic Research (Grant No. YSBR-137). We also thank for the technical support of the National Large Scientific and Technological Infrastructure "Earth System Numerical Simulation Facility"(https://cstr.cn/31134.02.EL).

Financial support. This research has been supported by Yunnan Fundamental Research Program (Grant No. 202501BC070001) and CAS Project for Young Scientists in Basic Research (Grant No. YSBR-137).

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

References

Adcroft, A. and Campin, J.-M.: Rescaled Height Coordinates for Accurate Representation of Free-Surface Flows in Ocean Circulation Models, Ocean Modelling, 7, 269–284, https://doi.org/10.1016/j.ocemod.2003.09.003, 2004.

Amante, C. and Eakins, B. W.: ETOPO1 1 Arc-Minute Global Relief Model: Procedures, Data Sources and Analysis, NOAA Technical Memorandum NESDIS NGDC-24, National Geophysical Data Center, NOAA, Boulder, Colorado, https://doi.org/10.7289/V5C8276M, 2009.

Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., Hirsh, B., Huang, S., Kalambarkar, K., Kirsch, L., Lazos, M., Lezcano, M., Liang, Y., Liang, J., Lu, Y., Luk, C. K., Maher, B., Pan, Y., Puhrsch, C., Reso, M., Saroufim, M., Siraichi, M. Y., Suk, H., Zhang, S., Suo, M., Tillet, P., Zhao, X., Wang, E., Zhou, K., Zou, R., Wang, X., Mathews, A., Wen, W., Chanan, G., Wu, P., and Chintala, S.: PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation, in: Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, pp. 929–947, ACM, https://doi.org/10.1145/3620665.3640366, 2024.

Bannister, R. N.: A Review of Operational Methods of Variational and Ensemble-variational Data Assimilation, Quarterly Journal of the Royal Meteorological Society, 143, 607–633, https://doi.org/10.1002/qj.2982, 2017.

Bauer, P., Thorpe, A., and Brunet, G.: The Quiet Revolution of Numerical Weather Prediction, Nature, 525, 47–55, https://doi.org/10.1038/nature14956, 2015.

Bauer, P., Dueben, P. D., Hoefler, T., Quintino, T., Schulthess, T. C., and Wedi, N. P.: The Digital Revolution of Earth-System Science, Nature Computational Science, 1, 104–113, https://doi.org/10.1038/s43588-021-00023-0, 2021.

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M.: Automatic Differentiation in Machine Learning: A Survey, Journal of Machine Learning Research, 18, 1–43, https://jmlr.org/papers/v18/17-468.html, 2018.

Bleck, R.: Ocean Modeling in Isopycnic Coordinates, in: Ocean Modeling and Parameterization, edited by Chassignet, E. P. and Verron, J., vol. 516 of *NATO Science Series*, pp. 423–448, Springer, Dordrecht, https://doi.org/10.1007/978-94-011-5096-5_18, 1998.

Bleck, R., Bao, J.-W., Benjamin, S. G., Brown, J. M., Fiorino, M., Henderson, T. B., Lee, J.-L., MacDonald, A. E., Madden, P., Middlecoff, J., Rosinski, J., Smirnova, T. G., Sun, S., and Wang, N.: A Vertically Flow-Following Icosahedral Grid Model for Medium-Range and Seasonal Prediction. Part I: Model Description, Monthly Weather Review, 143, 2386–2403, https://doi.org/10.1175/MWR-D-14-00300.1, 2015.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Katariya, Y., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q.: JAX: Composable Transformations of Python+NumPy Programs, http://github.com/jax-ml/jax, 2018.

Chen, X., Andronova, N., Van Leer, B., Penner, J. E., Boyd, J. P., Jablonowski, C., and Lin, S.-J.: A Control-Volume Model of the Compressible Euler Equations with a Vertical Lagrangian Coordinate, Monthly Weather Review, 141, 2526–2544, https://doi.org/10.1175/MWR-D-12-00129.1, 2013.

Dalcin, L. and Fang, Y.-L. L.: Mpi4py: Status Update after 12 Years of Development, Computing in Science & Engineering, 23, 47–54, https://doi.org/10.1109/MCSE.2021.3083216, 2021.

Danielson, J. J. and Gesch, D. B.: Global Multi-Resolution Terrain Elevation Data 2010 (GMTED2010), Tech. Rep. 2011-1073, U.S. Geological Survey, ISSN 2331-1258, https://doi.org/10.3133/ofr20111073, 2011.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

Davenport, E. H., Madan, J. V., Gjini, R., Brzenski, J., Crawford, B., Ho, N., Hsu, T.-Y., Liang, Y., Liu, Z., Manivannan, V., Pham, E., Vutukuru, R., Williams, A. I. L., Yang, Z., Yu, R., Lutsko, N. J., Hoyer, S., and Watson-Parris, D.: JCM v1.1: A Differentiable, Intermediate-Complexity Atmospheric Model, Geoscientific Model Development, 19, 6451–6466, https://doi.org/10.5194/gmd-19-6451-2026, 2026.

Durran, D. R. and Klemp, J. B.: A Compressible Model for the Simulation of Moist Mountain Waves, Monthly Weather Review, 111, 2341–2361, https://doi.org/10.1175/1520-0493(1983)111<2341:ACMFTS>2.0.CO;2, 1983.

Errico, R. M.: What Is an Adjoint Model?, Bulletin of the American Meteorological Society, 78, 2577–2591, https://doi.org/10.1175/1520-0477(1997)078<2577:WIAAM>2.0.CO;2, 1997.

Fuhrer, O., Chadha, T., Hoefler, T., Kwasniewski, G., Lapillonne, X., Leutwyler, D., Lüthi, D., Osuna, C., Schär, C., Schulthess, T. C., and Vogt, H.: Near-Global Climate Simulation at 1 Km Resolution: Establishing a Performance Baseline on 4888 GPUs with COSMO 5.0, Geoscientific Model Development, 11, 1665–1681, https://doi.org/10.5194/gmd-11-1665-2018, 2018.

Gassmann, A.: A Global Hexagonal C-Grid Non-Hydrostatic Dynamical Core (ICON-IAP) Designed for Energetic Consistency, Quarterly Journal of the Royal Meteorological Society, 139, 152–175, https://doi.org/10.1002/qj.1960, 2013.

Gelbrecht, M., White, A., Bathiany, S., and Boers, N.: Differentiable Programming for Earth System Modeling, Geoscientific Model Development, 16, 3123–3135, https://doi.org/10.5194/gmd-16-3123-2023, 2023.

Giering, R. and Kaminski, T.: Recipes for Adjoint Code Construction, ACM Transactions on Mathematical Software, 24, 437–474, https://doi.org/10.1145/293686.293695, 1998.

Griewank, A. and Walther, A.: Algorithm 799: Revolve: An Implementation of Checkpointing for the Reverse or Adjoint Mode of Computational Differentiation, ACM Transactions on Mathematical Software, 26, 19–45, https://doi.org/10.1145/347837.347846, 2000.

Häfner, D. and Vicentini, F.: Mpi4jax: Zero-Copy MPI Communication of JAX Arrays, Journal of Open Source Software, 6, 3419, https://doi.org/10.21105/joss.03419, 2021.

Hall, D. M., Ullrich, P. A., Reed, K. A., Jablonowski, C., Nair, R. D., and Tufo, H. M.: Dynamical Core Model Intercomparison Project (DCMIP) Tracer Transport Test Results for CAM-SE, Quarterly Journal of the Royal Meteorological Society, 142, 1672–1684, https://doi.org/10.1002/qj.2761, 2016.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., Chiara, G. D., Dahlgren, P., Dee, D., Diamantakis, M., Dragani, R., Flemming, J., Forbes, R., Fuentes, M., Geer, A., Haimberger, L., Healy, S., Hogan, R. J., Hólm, E., Janisková, M., Keeley, S., Laloyaux, P., Lopez, P., Lupu, C., Radnoti, G., de Rosnay, P., Rozum, I., Vamborg, F., Villaume, S., and Thépaut, J.-N.: The ERA5 Global Reanalysis, Quarterly Journal of the Royal Meteorological Society, 146, 1999–2049, https://doi.org/10.1002/qj.3803, 2020.

Hill, C., DeLuca, C., Balaji, Suarez, M., and Da Silva, A.: The Architecture of the Earth System Modeling Framework, Computing in Science & Engineering, 6, 18–28, https://doi.org/10.1109/MCISE.2004.1255817, 2004.

Huang, T., Dong, L., Xiao, Z., Pan, B., and Lu, G.: TerraFlow v1.0: Source Code and Supporting Materials for a Differentiable Multi-Backend Nonhydrostatic Atmospheric Dynamical Core on Unstructured Spherical Meshes, https://doi.org/10.5281/zenodo.24191174, 2026.

Jablonowski, C., Lauritzen, P. H., Nair, R. D., and Taylor, M.: Idealized Test Cases for the Dynamical Cores of Atmospheric General Circulation Models: A Proposal for the NCAR ASP 2008 Summer Colloquium, https://public.websites.umich.edu/cjablono/NCAR_ASP_2008_idealized_testcases_29May08.pdf, 2008.

Karypis, G. and Kumar, V.: A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs, SIAM Journal on Scientific Computing, 20, 359–392, https://doi.org/10.1137/S1064827595287997, 1998.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.



Klemp, J. B., Skamarock, W. C., and Dudhia, J.: Conservative Split-Explicit Time Integration Methods for the Compressible Nonhydrostatic Equations, Monthly Weather Review, 135, 2897–2913, https://doi.org/10.1175/MWR3440.1, 2007.

Klemp, J. B., Dudhia, J., and Hassiotis, A. D.: An Upper Gravity-Wave Absorbing Layer for NWP Applications, Monthly Weather Review, 136, 3987–4004, https://doi.org/10.1175/2008MWR2596.1, 2008.

Klemp, J. B., Skamarock, W. C., and Park, S.-H.: Idealized Global Nonhydrostatic Atmospheric Test Cases on a Reduced-radius Sphere, Journal of Advances in Modeling Earth Systems, 7, 1155–1177, https://doi.org/10.1002/2015MS000435, 2015.

Kochkov, D., Yuval, J., Langmore, I., Norgaard, P., Smith, J., Mooers, G., Klöwer, M., Lottes, J., Rasp, S., Düben, P., Hatfield, S., Battaglia, P., Sanchez-Gonzalez, A., Willson, M., Brenner, M. P., and Hoyer, S.: Neural General Circulation Models for Weather and Climate, Nature, 632, 1060–1066, https://doi.org/10.1038/s41586-024-07744-y, 2024.

Laprise, R.: The Euler Equations of Motion with Hydrostatic Pressure as an Independent Variable, Monthly Weather Review, 120, 197–207, https://doi.org/10.1175/1520-0493(1992)120<0197:TEEOMW>2.0.CO;2, 1992.

Lin, S.-J.: A "Vertically Lagrangian" Finite-Volume Dynamical Core for Global Models, Monthly Weather Review, 132, 2293–2307, https://doi.org/10.1175/1520-0493(2004)132<2293:AVLFDC>2.0.CO;2, 2004.

MacDonald, A. E., Middlecoff, J., Henderson, T., and Lee, J.-L.: A General Method for Modeling on Irregular Grids, The International Journal of High Performance Computing Applications, 25, 392–403, https://doi.org/10.1177/1094342010385019, 2011.

Mesinger, F. and Arakawa, A.: Numerical Methods Used in Atmospheric Models. Volume I, GARP Publications Series 17, World Meteorological Organization, Geneva, 1976.

Moses, W. S., Cheng, G., Churavy, V., Gelbrecht, M., Klöwer, M., Kump, J., Morlighem, M., Williamson, S., Apte, D., Berg, P., Giordano, M., Hill, C., Loose, N., Montoison, A., Narayanan, S. H. K., Pal, A., Schanen, M., Silvestri, S., Wagner, G., and Heimbach, P.: DJ4Earth: Differentiable, and Performance-Portable Earth System Modeling via Program Transformations, Journal of Advances in Modeling Earth Systems, 18, e2025MS005 615, https://doi.org/10.1029/2025MS005615, 2026.

Oberhuber, J. M.: Simulation of the Atlantic Circulation with a Coupled Sea Ice-Mixed Layer-Isopycnal General Circulation Model. Part I: Model Description, Journal of Physical Oceanography, 23, 808–829, https://doi.org/10.1175/1520-0485(1993)023<0808:SOTACW>2.0.CO;2, 1993.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library, in: Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., https://papers.nips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html, 2019.

Rasp, S.: Coupled Online Learning as a Way to Tackle Instabilities and Biases in Neural Network Parameterizations: General Algorithms and Lorenz 96 Case Study (v1.0), Geoscientific Model Development, 13, 2185–2196, https://doi.org/10.5194/gmd-13-2185-2020, 2020.

Ringler, T., Ju, L., and Gunzburger, M.: A Multiresolution Method for Climate System Modeling: Application of Spherical Centroidal Voronoi Tessellations, Ocean Dynamics, 58, 475–498, https://doi.org/10.1007/s10236-008-0157-2, 2008.

Ringler, T., Thuburn, J., Klemp, J., and Skamarock, W.: A Unified Approach to Energy Conservation and Potential Vorticity Dynamics for Arbitrarily-Structured C-Grids, Journal of Computational Physics, 229, 3065–3090, https://doi.org/10.1016/j.jcp.2009.12.007, 2010.

Ringler, T., Petersen, M., Higdon, R. L., Jacobsen, D., Jones, P. W., and Maltrud, M.: A Multi-Resolution Approach to Global Ocean Modeling, Ocean Modelling, 69, 211–232, https://doi.org/10.1016/j.ocemod.2013.04.010, 2013.

Satoh, M., Stevens, B., Judt, F., Khairoutdinov, M., Lin, S.-J., Putman, W. M., and Düben, P.: Global Cloud-Resolving Models, Current Climate Change Reports, 5, 172–184, https://doi.org/10.1007/s40641-019-00131-0, 2019.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.

Schär, C., Fuhrer, O., Arteaga, A., Ban, N., Charpilloz, C., Di Girolamo, S., Hentgen, L., Hoefler, T., Lapillonne, X., Leutwyler, D., Osterried, K., Panosetti, D., Rüdisühli, S., Schlemmer, L., Schulthess, T. C., Sprenger, M., Ubbiali, S., and Wernli, H.: Kilometer-Scale Climate Models: Prospects and Challenges, Bulletin of the American Meteorological Society, 101, E567–E587, https://doi.org/10.1175/BAMS-D-18-0167.1, 2020.

Simmons, A. J. and Burridge, D. M.: An Energy and Angular-Momentum Conserving Vertical Finite-Difference Scheme and Hybrid Vertical Coordinates, Monthly Weather Review, 109, 758–766, https://doi.org/10.1175/1520-0493(1981)109<0758:AEAAMC>2.0.CO;2, 1981.

Skamarock, W. C.: Positive-Definite and Monotonic Limiters for Unrestricted-Time-Step Transport Schemes, Monthly Weather Review, 134, 2241–2250, https://doi.org/10.1175/MWR3170.1, 2006.

Skamarock, W. C. and Gassmann, A.: Conservative Transport Schemes for Spherical Geodesic Grids: High-Order Flux Operators for ODE-Based Time Integration, Monthly Weather Review, 139, 2962–2975, https://doi.org/10.1175/MWR-D-10-05056.1, 2011.

Skamarock, W. C., Klemp, J. B., Duda, M. G., Fowler, L. D., Park, S.-H., and Ringler, T. D.: A Multiscale Nonhydrostatic Atmospheric Model Using Centroidal Voronoi Tesselations and C-Grid Staggering, Monthly Weather Review, 140, 3090–3105, https://doi.org/10.1175/MWR-D-11-00215.1, 2012.

Smagorinsky, J.: General Circulation Experiments with the Primitive Equations: I. The Basic Experiment, Monthly Weather Review, 91, 99–164, https://doi.org/10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2, 1963.

Staniforth, A. and Thuburn, J.: Horizontal Grids for Global Weather and Climate Prediction Models: A Review, Quarterly Journal of the Royal Meteorological Society, 138, 1–26, https://doi.org/10.1002/qj.958, 2012.

Thuburn, J., Ringler, T., Skamarock, W., and Klemp, J.: Numerical Representation of Geostrophic Modes on Arbitrarily Structured C-Grids, Journal of Computational Physics, 228, 8321–8335, https://doi.org/10.1016/j.jcp.2009.08.006, 2009.

Ullrich, P. A., Jablonowski, C., Kent, J., Lauritzen, P. H., Nair, R. D., and Taylor, M. A.: Dynamical Core Model Intercomparison Project (DCMIP) Test Case Document, https://public.websites.umich.edu/~cjablono/DCMIP-2012_TestCaseDocument_v1.7.pdf, 2012.

Wang, H., Skamarock, W. C., and Feingold, G.: Evaluation of Scalar Advection Schemes in the Advanced Research WRF Model Using Large-Eddy Simulations of Aerosol–Cloud Interactions, Monthly Weather Review, 137, 2547–2558, https://doi.org/10.1175/2009MWR2820.1, 2009.

Wicker, L. J. and Skamarock, W. C.: Time-Splitting Methods for Elastic Models Using Forward Time Schemes, Monthly Weather Review, 130, 2088–2097, https://doi.org/10.1175/1520-0493(2002)130<2088:TSMFEM>2.0.CO;2, 2002.

Zalesak, S. T.: Fully Multidimensional Flux-Corrected Transport Algorithms for Fluids, Journal of Computational Physics, 31, 335–362, https://doi.org/10.1016/0021-9991(79)90051-2, 1979.

Zängl, G., Reinert, D., Rípodas, P., and Baldauf, M.: The ICON (ICOsahedral Non-hydrostatic) Modelling Framework of DWD and MPI-M : Description of the Non-hydrostatic Dynamical Core, Quarterly Journal of the Royal Meteorological Society, 141, 563–579, https://doi.org/10.1002/qj.2378, 2015.

Zhang, W. and Chen, X.: LMARSpy: A GPU-Ready Nonhydrostatic Dynamical Core With a Sharpness-Preserving Monotonicity Limiter and a Conservative Vertical Implicit Solver, Journal of Advances in Modeling Earth Systems, 17, e2025MS005 056, https://doi.org/10.1029/2025MS005056, 2025.

Zhang, Y.: Extending High-Order Flux Operators on Spherical Icosahedral Grids and Their Applications in the Framework of a Shallow Water Model, Journal of Advances in Modeling Earth Systems, 10, 145–164, https://doi.org/10.1002/2017MS001088, 2018.

https://doi.org/10.5194/egusphere-2026-7116
Preprint. Discussion started: 24 November 2026
© Author(s) 2026. CC BY 4.0 License.



Zhang, Y., Li, J., Yu, R., Zhang, S., Liu, Z., Huang, J., and Zhou, Y.: A Layer-Averaged Nonhydrostatic Dynamical Framework on an Unstructured Mesh for Global and Regional Atmospheric Modeling: Model Description, Baseline Evaluation, and Sensitivity Exploration, Journal of Advances in Modeling Earth Systems, 11, 1685–1714, https://doi.org/10.1029/2018MS001539, 2019.

Zhang, Y., Li, J., Yu, R., Liu, Z., Zhou, Y., Li, X., and Huang, X.: A Multiscale Dynamical Model in a Dry-Mass Coordinate for Weather and Climate Modeling: Moist Dynamics and Its Coupling to Physics, Monthly Weather Review, 148, 2671–2699, https://doi.org/10.1175/MWR-D-19-0305.1, 2020.