Large-eddy simulations with ClimateMachine v0.2.0: a new open-source code for atmospheric simulations on GPUs and CPUs

Akshay Sridhar[1], Yassine Tissaoui[2], Simone Marras[2], Zhaoyi Shen[1], Charles Kawczynski[1], Simon Byrne[1], Kiran Pamnany[1], Maciej Waruszewski[3], Thomas H. Gibson[4], Jeremy E. Kozdon[3], Valentin Churavy[5], Lucas C. Wilcox[3], Francis X. Giraldo[3], and Tapio Schneider[1,6]

[1]California Institute of Technology, Pasadena, California, USA
[2]New Jersey Institute of Technology, Newark, New Jersey, USA
[3]Naval Postgraduate School, Monterey, California, USA
[4]University of Illinois Urbana-Champaign, Urbana-Champaign, Illinois, USA
[5]Massachusetts Institute of Technology, Cambridge, Massachussetts, USA
[6]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, USA

Correspondence: Akshay Sridhar (asridhar@caltech.edu)

Abstract. We introduce ClimateMachine, a new open-source atmosphere modeling framework using the Julia language, designed to be scalable on central processing units (CPUs) and graphics processing units (GPUs). ClimateMachine uses a common framework both for coarser-resolution global simulations and for high-resolution, limited-area large-eddy simulations (LES). Here, we demonstrate the LES configuration of the atmosphere model in canonical benchmark cases and atmospheric flows, using a total energy-conserving nodal discontinuous-Galerkin (DG) discretization of the governing equations. Resolution dependence, conservation characteristics, and scaling metrics are examined in comparison with existing LES codes. They demonstrate the utility of ClimateMachine as a modelling tool for limited-area LES flow configurations.

1 Introduction

Hybrid computer architectures and the need to exploit the power of graphics processing units (GPUs) are increasingly driving developments in atmosphere and climate modeling (e.g., Schalkwijk et al., 2012; Palmer, 2014; Schalkwijk et al., 2015; Marras et al., 2015; Abdi et al., 2017b, a; Fuhrer et al., 2018; Schär et al., 2020). The sheer computing power available on modern hardware architectures presents opportunities to accelerate atmosphere and climate modeling. However, exploiting this computing power requires re-coding atmosphere and climate models to an extent not seen in decades, and portable performance and scaling across different platforms remain difficult to achieve (Fuhrer et al., 2014; Balaji, 2021).

In this paper, we introduce ClimateMachine, a new open-source atmosphere model written in the Julia programming language (Bezanson et al., 2017), using the Julia Message Passing Interface (MPI), to provide a computational framework that is portable across CPU and GPU architectures. The use of Julia aims to increase accessibility and utility of ClimateMachine as a simulation tool. Julia, developed with the aim of bridging the gap between productivity (e.g., Python) and performance (e.g., Fortran) languages (Bezanson et al., 2018), is a strong candidate for an application such as atmospheric LES. Features such as package composability, GPU programming tools, and interactive programming interfaces contribute to its usability in

scientific computing. Compilation time, incompatibilities between Julia MPI programs and interactive read-eval-print-loops (REPL), and the discovery of code issues during runtime due to Julia's dynamic typing are general drawbacks for new users to this language. Despite these drawbacks, a recent assessment by Lin and McIntosh-Smith (2021) further supports the suitability of Julia as a high-performance computing language. The atmospheric model presented here is designed to be usable across a range of physical process scales, from large-eddy simulations (LES) with meter-scale resolution to global circulation models (GCM) with horizontal resolutions of tens of kilometers, as in a few other recent models (Dipankar et al., 2015). We focus on the LES configuration of ClimateMachine in this paper.

Since the pioneering work on turbulence in stratified flows by Lilly (1962), on horizontal subgrid-scale dissipation in GCMs by Smagorinsky (1963), and the subsequent extension of such models to LES (Lilly, 1966), several models have been developed to improve the ability of LES to model atmospheric turbulence: from the extensive work by Deardorff in the 1970s and 1980s (Deardorff, 1970, 1974, 1976, 1980), by Moeng in the 1980s and beyond (Moeng, 1984; Moeng and Wyngaard, 1988; Sullivan et al., 1994; Moeng et al., 2003), to Stevens, Teixeira, Mellado, and others in the last two decades (Stevens et al., 2003, 2005; Savic-Jovcic and Stevens, 2008; Matheou et al., 2011; Pressel et al., 2015; Matheou, 2016; Matheou and Teixeira, 2019; Mellado, 2017; Mellado et al., 2018). LES results in canonical flows are sensitive to the fine details of the equations used to represent the flow dynamics, the viscous dissipation, the thermodynamics, and the numerical methods used to solve them (Ghosal, 1996; Chow and Moin, 2003; Kurowski et al., 2014), especially in the case of cloud simulations (Stevens et al., 2005; Siebesma et al., 2003; Schalkwijk et al., 2012, 2015; Schneider et al., 2019; Pressel et al., 2015, 2017).

One distinguishing aspect of the ClimateMachine LES is that it uses a nodal discontinuous Galerkin (DG) formulation to approximate the Navier-Stokes equations for compressible flow (Giraldo et al., 2002; Hesthaven and Warburton, 2008a; Giraldo and Restelli, 2008; Kopriva, 2009; Kelly and Giraldo, 2012; Giraldo, 2020). The DG method is a spectral-element generalization of finite-volume methods. It lends itself well to modern high-performance computing architectures because its communication overhead is low, enabling scaling on manycore processors including GPUs (Abdi et al., 2017b). Another important consideration within ClimateMachine is the use of total energy of moist air as a prognostic variable, ensuring energetic consistency of the simulations. We demonstrate that the ClimateMachine LES can be successfully used to simulate canonical LES benchmarks, including simulations of flows over mountains and different cloud and boundary-layer regimes (e.g., Straka et al., 1993; Schär et al., 2002; Stevens et al., 2005).

In what follows, we describe the conceptual and numerical foundations and governing equations of ClimateMachine and demonstrate the model in a set of standard two- and three-dimensional benchmark simulations. Section 2 begins by highlighting the governing equations. Their numerical approximation through the DG representation is described in Section 3. Section 4 presents sub-grid scale models used in the LES to represent under-resolved flow physics, with results from key benchmarks presented in Section 5. Conservation properties are examined in Section 6, and performance on CPU and GPU hardware is described in Sections 7 and 8, respectively. Section 9 contains closing remarks. Additional details about the model, boundary conditions, statistical definitions, and computer hardware are summarized in the appendices.

2 Governing Equations

2.1 Working fluid

The working fluid of the atmosphere model is moist, potentially cloudy air, considered to be an ideal mixture of dry air, water vapor, and condensed water (liquid and ice) in clouds. Dry air and water vapor are taken to be ideal gases. The specific volume of the cloud condensate is neglected relative to that of the gas phases (it is a factor 10^3 less than that of the gas phases). All gas phases are assumed to have the same temperature, and are advected with the same velocity $\boldsymbol{u} = (u, v, w)^T$. Cloud condensate is assumed to sediment relative to gaseous phases slowly enough to be in thermal equilibrium with the surrounding fluid.

The density of the moist air is denoted by ρ. We use the following notation for the mass fractions of the moist air mixture (mass of a constituent divided by the total mass of the working fluid):

- q_d: dry air mass fraction,

- q_v: water vapor specific humidity,

- q_l: liquid water specific humidity,

- q_i: ice specific humidity,

- $q_c = q_l + q_i$: condensate specific humidity,

- $q_t = q_v + q_c$: total specific humidity.

Because this enumerates all constituents of the working fluid, we have $q_t + q_d = 1$. In Earth's atmosphere, the water vapor specific humidity q_v dominates the total specific humidity q_t and is usually $\mathcal{O}(10^{-2})$ or smaller; the condensate specific humidity is typically $\mathcal{O}(10^{-4})$. Hence, water is a trace constituent of the atmosphere, and only a small fraction of atmospheric water is in condensed phases. The working fluid pressure is the sum of the partial pressures of dry air and water vapor such that $p = \rho(R_d q_d + R_v q_v)T$, where R_d is the specific gas constant of dry air, and R_v is the specific gas constant of water vapor.

2.2 Mass balance

Moist air mass satisfies the conservation equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = \rho \hat{\mathcal{S}}_{q_t}. \tag{1}$$

Moist air mass is not exactly conserved where precipitation forms, sublimates, or evaporates, where water diffuses, or where condensate sediments relative to the gas phases (Bott, 2008; Romps, 2008). The right-hand side involves the local source/sink of water mass $\hat{\mathcal{S}}_{q_t}$ owing to such non-conservative processes, which we take into account although it is small because water is a trace constituent of the atmosphere.

2.3 Total water balance

Total water satisfies the balance equation

$$\frac{\partial(\rho q_t)}{\partial t} + \nabla \cdot (\rho q_t \boldsymbol{u}) = \rho \mathcal{S}_{q_t} - \nabla \cdot (\rho \boldsymbol{d}_{q_t}) + \nabla \cdot \left(\rho q_c w_c \hat{\boldsymbol{k}}\right) \equiv \rho \hat{\mathcal{S}}_{q_t}. \tag{2}$$

Here, the source/sink $\mathcal{S}_{q_t}$ arises from evaporation or sublimation of precipitation and formation of precipitation. Diffusive fluxes of moisture are captured by $\boldsymbol{d}_{q_t}$. The effective sedimentation velocity of cloud condensate w_c is defined such that

$$q_c w_c = q_l w_l + q_i w_i \tag{3}$$

with w_l and w_i defined to be positive downward ($\hat{\boldsymbol{k}}$ being the upward pointing unit vector). The right-hand side $\rho \hat{\mathcal{S}}_{q_t}$ of the total water balance equation is the same as the right-hand side of the mass balance equation (1).

2.4 Momentum balance

The coordinate independent form of the conservation law for momentum is

$$\frac{\partial(\rho \boldsymbol{u})}{\partial t} + \nabla \cdot [\rho \boldsymbol{u} \otimes \boldsymbol{u} + (p - p_r)\boldsymbol{I}_3] = -(\rho - \rho_r)\boldsymbol{\nabla}\Phi - 2\boldsymbol{\Omega} \times \rho \boldsymbol{u}$$
$$- \nabla \cdot (\rho \boldsymbol{\tau}) - \nabla \cdot (\boldsymbol{d}_{q_t} \otimes \rho \boldsymbol{u}) + \nabla \cdot \left(q_c w_c \hat{\boldsymbol{k}} \otimes \rho \boldsymbol{u}\right) + \rho \boldsymbol{F_u}, \quad (4)$$

where $\boldsymbol{I}_3$ is the rank-3 identity matrix, Φ is the effective gravitational potential including centrifugal accelerations, $\boldsymbol{\tau}$ is a viscous and/or subgrid-scale (SGS) momentum flux tensor; and $\boldsymbol{F_u}$ (typically with $\boldsymbol{F_u} \cdot \boldsymbol{u} < 0$, so that $\boldsymbol{F_u}$ represents a momentum sink) is any other drag force per unit mass that may be applied, for example, at the lower boundary. The term involving the planetary angular velocity $\boldsymbol{\Omega}$ accounts for Coriolis forces. To improve numerical stability, we have factored out a reference state with a pressure $p_r(z)$ and density $\rho_r(z)$ that depend only on altitude z and are in hydrostatic balance, so that they satisfy

$$\boldsymbol{\nabla}p_r = -\rho_r \boldsymbol{\nabla}\Phi.$$

The tensor involving the diffusive flux $\boldsymbol{d}_{q_t}$ of water on the right-hand side of (4) represents the momentum flux carried by water that is diffusing; this term is usually very small, but we take it into account.

2.5 Energy balance

The specification of a thermodynamic or energy conservation equation closes the equations of motion for the working fluid. We use the total specific energy, e^{tot}, as the prognostic variable. Total energy is conserved in reversible moist processes such as phase transitions of water.

Total energy satisfies the conservation law (Romps, 2008; Bott, 2008)

$$\frac{\partial(\rho e^{\text{tot}})}{\partial t} + \nabla \cdot ((\rho e^{\text{tot}} + p)\boldsymbol{u}) = -\nabla \cdot (\rho \boldsymbol{F}_R) - \nabla \cdot [\rho(\boldsymbol{J} + \boldsymbol{D})] + \rho Q + \nabla \cdot \left(\rho W_c \hat{\boldsymbol{k}}\right) - \nabla \cdot (\boldsymbol{u} \cdot \rho \boldsymbol{\tau}) - \sum_{j \in \{v, l, i\}} (I_j + \Phi)\rho C(q_j \to q_p) - M,$$
$$\tag{5}$$

where the total specific energy e^{tot} is defined by

$$e^{\mathrm{tot}} = \frac{1}{2}\|\boldsymbol{u}\|^2 + \Phi + I. \tag{6}$$

The constituents (dry air and moisture components) here are assumed to be moving with the same velocity $\boldsymbol{u}$ (that is, we neglect, as is common, the diffusive and sedimentation fluxes of water in the kinetic energy). The constituents are also assumed to be in thermal equilibrium at the same temperature T, so that the specific internal energy of moist air is the weighted sum of the specific energies of the constituents: dry air (I_d), water vapor (I_v), liquid water (I_l), and ice (I_i):

$$I(T,q) = (1 - q_t)I_d(T) + q_v I_v(T) + q_l I_l(T) + q_i I_i(T), \tag{7}$$

with

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

Here, c_{vk} for $k \in \{d, v, l, i\}$ are isochoric specific heat capacities for the appropriate species denoted by k; they are taken to be constant. The reference specific internal energy $I_{v,0}$ is the difference in specific internal energy between vapor and liquid at the arbitrary reference temperature T_0; $I_{i,0}$ is the difference in specific internal energy between ice and liquid at T_0 (Romps, 2008). The reference internal energies are related to specific latent heats of vaporization and fusion, $L_{v,0}$ and $L_{f,0}$, at the reference temperature T_0 through

$$I_{v,0} = L_{v,0} - R_v T_0, \tag{9}$$
$$I_{i,0} = L_{f,0}. \tag{10}$$

The values of the thermodynamic constants we use are listed in Table 1.

Furthermore, the flux $\boldsymbol{F}_R$ is the radiative energy flux per unit mass; $\boldsymbol{J}$ is the conductive energy flux per unit mass, and $\boldsymbol{D}$ is the specific enthalpy flux associated with the diffusive flux of water

$$\boldsymbol{D} = (e_v^{\mathrm{tot}} + R_v T)\boldsymbol{d}_{q_v} + e_l^{\mathrm{tot}}\boldsymbol{d}_{q_l} + e_i^{\mathrm{tot}}\boldsymbol{d}_{q_i}. \tag{11}$$

The flux $\boldsymbol{u} \cdot \rho\boldsymbol{\tau}$ is the energy flux associated with the viscous and/or SGS turbulent momentum flux; and Q is any internal energy source (e.g., external diabatic heating). The flux

$$W_c = q_l e_l^{\mathrm{tot}} w_l + q_i e_i^{\mathrm{tot}} w_i \tag{12}$$

represents the downward energy flux due to sedimenting condensate.

Table 1. Thermodynamic constants in CLIMAParameters

R_d	287 J (kg K)$^{-1}$
R_v	462 J (kg K)$^{-1}$
c_{vd}	717.5 J (kg K)$^{-1}$
c_{vv}	1397.5 J (kg K)$^{-1}$
c_{vl}	4181 J (kg K)$^{-1}$
c_{vi}	2100 J (kg K)$^{-1}$
T_0	273.16 K
$L_{v,0}$	2.5008×10^6 J kg^{-1}
$L_{f,0}$	0.3336×10^6 J kg^{-1}

The terms involving $\rho C(q_j \to q_p)$ $(j \in \{v, l, i\})$ represent the loss of internal and potential energy of moist air masses owing to precipitation formation; the kinetic energy loss is neglected, consistent with the neglect of the source/sink associated with precipitation formation in the momentum balance (4). Additional energy sinks involve the energy loss owing to heat transfer from the working fluid to precipitation as it falls through air and possibly melts at the freezing level (Raymond, 2013); the associated energy sources/sinks are generally provided by a microphysics parameterization and are subsumed in the term M.

2.6 Equation of state

Pressure p is calculated from the ideal-gas law

$$p = \rho R_m T, \tag{13}$$

where R_m is the gas "constant" of moist air,

$$R_m(q) = R_d(1 - q_t) + R_v q_v$$

$$= R_d \left[1 + (\varepsilon_{dv} - 1)q_t - \varepsilon_{dv} q_c \right], \tag{14}$$

with the ratio of the gas constants of water vapor and of dry air $\varepsilon_{dv} = R_v/R_d$.

2.7 Saturation adjustment

Gibbs' phase rule states that in thermodynamic equilibrium, the temperature T and liquid and ice specific humidities q_l and q_i can be obtained from the three thermodynamic state variables density ρ, total water specific humidity q_t, and internal energy I. Thus, the above equations suffice to completely specify the thermodynamic state of the working fluid, given ρ, q_t, and I, the latter obtained from the total energy via its definition (6).

Obtaining the temperature and condensate specific humidities from the state variables ρ, q_t, and I is the problem of finding the root T of

$$I^*(T; \rho, q_t) - I = 0, \tag{15}$$

where $I^*(T; \rho, q_t)$ is the internal energy at equilibrium, when the air is either unsaturated and there is no condensate ($q_v = q_t$), or water vapor is in saturation and the saturation excess $q_t - q_v$ is apportioned, according to temperature T, among the condensed phases q_l and q_i. We solve this nonlinear "saturation adjustment" problem by Newton iterations with analytical gradients (cf. Tao et al., 1989; Pressel et al., 2015). To obtain the saturation vapor pressure and derived functions needed in this calculation, we assume all isochoric heat capacities to be constant (i.e., we assume the gases to be calorically perfect); with this assumption, the Clausius-Clapeyron can be integrated analytically, resulting in a closed-form expression for the saturation vapor pressure (Romps, 2008).

This procedure allows the use of total moisture q_t as the sole prognostic variable, but confines the system to the assumption of equilibrium thermodynamics. Alternatively, using explicit tracers for the condensate specific humidities q_l and q_i allows non-equilibrium thermodynamics to be considered and mixed-phase processes to be explicitly modeled.

3 Discretization of the governing equations

3.1 Space discretization

The governing equations are discretized in space via a nodal DG approximation. To describe the DG procedure, we recast the equations (1)–(5) in divergence form as

$$\frac{\partial \boldsymbol{Y}}{\partial t} = -\nabla \cdot (\boldsymbol{F}_1 + \boldsymbol{F}_2) + \boldsymbol{S}(\boldsymbol{Y}), \tag{16}$$

where $\boldsymbol{Y} = [\rho, \rho \boldsymbol{u}, \rho e^{\text{tot}}, \rho q_t]^T$ is an abstract vector of state variables; $\boldsymbol{F}_1$ contains the fluxes not involving gradients of state variables and functions thereof; $\boldsymbol{F}_2$ contains the fluxes involving gradients of state variables (e.g., diffusive fluxes); and $\boldsymbol{S}(\boldsymbol{Y})$ contains the sources.

The DG solution of (16) is approximated on the finite-dimensional counterpart Ω^h of the flow domain Ω, which consists of N_{Ω_e} non-overlapping hexahedral elements Ω_e such that

$$\Omega^h = \bigcup_{e=1}^{N_{\Omega_e}} \Omega_e,$$

where a superscript h indicates the discrete analog of a continuous quantity. By virtue of tensor-product operations allowed on hexahedral elements and the ability to rely on inexact quadrature when elements of order greater than 3 are utilized, high-order Galerkin methods are particularly attractive for operation intensive solutions (Kelly and Giraldo, 2012). Within each element, the finite dimensional approximation of $\boldsymbol{Y}(\mathbf{x}, t)$ is given by the expansion

$$\boldsymbol{Y}_e^h(\mathbf{x}, t) = \sum_{\alpha=1}^{(N+1)^3} \psi_\alpha^e(\mathbf{x}) \boldsymbol{Y}_\alpha^e(t), \tag{17}$$

where $(N+1)^3$ is the number of collocation points within the three-dimensional element of order N, and ψ_α^e are the interpolation polynomials evaluated at local point α inside element e.

From now on, the subscript/superscript e is omitted with the understanding that all operations are executed element-wise unless otherwise stated. Furthermore, the physical elements in the $\mathbf{x} = (x, y, z)$ space are mapped to a reference element $\boldsymbol{\xi} = (\xi, \eta, \zeta)$. The three-dimensional basis functions ψ_α result from the one-dimensional functions $L_i(\xi)$, $L_j(\eta)$, and $L_k(\zeta)$ as the tensor product:

$$\psi_\alpha(\boldsymbol{\xi}) = L_i(\xi) \otimes L_j(\eta) \otimes L_k(\zeta), \qquad \forall i, j, k = 1, ..., N+1.$$

Each function L is a one-dimensional (1D) Lagrange polynomial defined on the 1D reference element $[-1, 1]$. The Lagrange function evaluated at points i along the ξ direction within the element is

$$L_i(\xi) = \prod_{l=1, l \neq i}^{N+1} \frac{\xi - \xi_l}{\xi_i - \xi_l},$$

where ξ_i are the $N+1$ co-located interpolation points along ξ. The polynomials L_j and L_k in the two other directions η and ζ are built in the same way. The $N+1$ interpolation points may be chosen in variety of ways (Deville et al., 2002; Karniadakis and Sherwin, 1999); here we choose Legendre-Gauss-Lobatto (LGL) points (Giraldo and Restelli, 2008). The Kronecker δ property of the Lagrange polynomials is such that

$$\psi_i(\xi_l) = \delta_{il}$$

in 1D which, in three-dimensions (3D), translates to

$$\Psi_\alpha(\xi_a, \eta_b, \zeta_c) = \delta_{ia} \otimes \delta_{jb} \otimes \delta_{kc}. \tag{18}$$

This allows us to reduce the operation count, as follows.

We construct the space and time derivatives as

$$\frac{\partial \mathbf{Y}^h(\mathbf{x}, t)}{\partial \mathbf{x}} = \sum_{\alpha=1}^{(N+1)^3} \frac{\partial \psi_\alpha(\mathbf{x})}{\partial \mathbf{x}} \mathbf{Y}_\alpha(t), \tag{19}$$

$$\frac{\partial \mathbf{Y}^h(\mathbf{x}, t)}{\partial t} = \sum_{\alpha=1}^{(N+1)^3} \psi_\alpha(\mathbf{x}) \frac{\partial \mathbf{Y}_\alpha(t)}{\partial t}. \tag{20}$$

By virtue of the 3D Kronecker δ property, the spatial derivatives of the basis functions appearing here are given by

$$\frac{\partial \psi_\alpha}{\partial \xi}(\xi_a, \eta_b, \zeta_c) = \frac{\partial \psi_i(\xi_a)}{\partial \xi} \otimes \delta_{jb} \otimes \delta_{ck}, \tag{21}$$

$$\frac{\partial \psi_\alpha}{\partial \eta}(\xi_a, \eta_b, \zeta_c) = \delta_{ia} \otimes \frac{\partial \psi_j(\eta_b)}{\partial \eta} \otimes \delta_{ck}, \tag{22}$$

$$\frac{\partial \psi_\alpha}{\partial \zeta}(\xi_a, \eta_b, \zeta_c) = \delta_{ia} \otimes \delta_{jb} \otimes \frac{\partial \psi_k(\zeta_c)}{\partial \zeta}. \tag{23}$$

Using this property reduces the operation count significantly since we only require $3N$ operations instead of the N^3 operations otherwise needed to compute the derivatives at a given node (Abdi et al., 2017b).

The operators defined on the reference elements are mapped onto the physical space by means of the transformation

$$\nabla\psi = \mathbf{J}^{-1}\widehat{\nabla}\psi \tag{24}$$

where $\nabla = (\partial_x, \partial_y, \partial_z)^T$, $\widehat{\nabla} = (\partial_\xi, \partial_\eta, \partial_\zeta)^T$ and

$$\mathbf{J}^{-1} = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix}$$

is the inverse Jacobian of the transformation from physical space to the reference element.

The DG approximation of the differential equations (16) is constructed by multiplying, within each element, the equation by the test function ψ_α and then integrating over the element volume Ω_e, such that

$$\int_{\Omega_e} \psi_\alpha \left(\partial_t \mathbf{Y} + \nabla \cdot \mathbf{F}_1(\mathbf{Y}) + \nabla \cdot \mathbf{F}_2(\mathbf{Y}, \nabla \mathbf{Y}) \right) d\Omega_e$$

$$= \int_{\Omega_e} \psi_\alpha \boldsymbol{S}(\mathbf{Y}) d\Omega_e, \tag{25}$$

where ψ_α within each element belongs to the function space of square integrable piecewise polynomials of order N (i.e., $\psi \in L^2$). By definition, these functions are discontinuous across element boundaries; differentiability is not globally required but only within each element (Hesthaven and Warburton, 2008b). Integrating the divergence term by parts yields

$$\int_{\Omega_e} \Psi_\alpha \partial_t \mathbf{Y}\, d\Omega_e + \oint_{\Gamma_e} \Psi_\alpha \mathbf{n} \cdot \mathbf{F}_1^*(\mathbf{Y})\, d\Gamma_e$$

$$- \int_{\Omega_e} \nabla\Psi_\alpha \cdot \mathbf{F}_1(\mathbf{Y})\, d\Omega_e - \int_{\Omega_e} \Psi_\alpha \nabla \cdot \mathbf{F}_2(\mathbf{Y}, \boldsymbol{\nabla}\mathbf{Y})\, d\Omega_e$$

$$= \int_{\Omega_e} \Psi_\alpha \boldsymbol{S}(\mathbf{Y})\, d\Omega_e, \tag{26}$$

where Ω_e and Γ_e are, respectively, the volume and boundary of each element, $\mathbf{n}$ is the outward facing unit vector orthogonal to each element face, and $\boldsymbol{F}_1^*$ is a numerical flux. The imposition of the numerical fluxes across element boundaries is the numerical mechanism that promotes continuity of the discontinuous solution across the elements. The numerical fluxes are calculated as the approximate solution to a Riemann problem across two neighboring elements. ClimateMachine currently implements the Rusanov (1961), Roe (1981), and Harten-Lax-van Leer-Contact (HLLC) (E. F. Toro et al., 1994; Harten, 1983) numerical fluxes. The Rusanov flux, for instance, is constructed as

$$\mathbf{n} \cdot \boldsymbol{F}_1^*(\boldsymbol{Y}) = \frac{\mathbf{n}}{2} \cdot \left[\boldsymbol{F}_1(\boldsymbol{Y}^-) + \boldsymbol{F}_1(\boldsymbol{Y}^+) \right] + \mathbf{n}\lambda_{\Gamma_e} \left(\boldsymbol{Y}^- - \boldsymbol{Y}^+ \right), \tag{27}$$

where $\boldsymbol{Y}^-$ is the state at the internal interface of element e, $\boldsymbol{Y}^+$ is the state at the external interface of e, and $\lambda_{\Gamma_e}(\boldsymbol{Y}^-, \boldsymbol{Y}^+)$ is an estimate of the maximum flow speed (e.g., the maximum eigenvalue of the Jacobian of the flux $\boldsymbol{F}_1$ with respect to the state variables, which is the speed of sound).

235   Because the second-order derivatives in $\nabla \cdot \boldsymbol{F}_2$ cannot be directly built with the weak variational formulation if a discontinuous function space is used (Bassi and Rebay, 1997), an auxiliary variable $\widetilde{\mathbf{Y}}$ is introduced such that

$$\nabla \mathbf{Y} = \widetilde{\mathbf{Y}} \tag{28}$$

$$\nabla \cdot (\mu \nabla \mathbf{Y}) = \nabla \cdot (\mu \widetilde{\mathbf{Y}}), \tag{29}$$

which can then be discretized via DG as

$$240 \quad \int_{\Omega_e} \Psi_\alpha \nabla \cdot \nabla \boldsymbol{Y} \, d\Omega_e \approx \oint_{\Gamma_e} \Psi_\alpha \mathbf{n} \cdot \left(\mu \widetilde{\boldsymbol{Y}}^* - \mu \widetilde{\boldsymbol{Y}} \right) d\Gamma_e + \int_{\Omega_e} \Psi_\alpha \nabla \cdot \left(\mu \widetilde{\boldsymbol{Y}} \right) d\Omega_e. \tag{30}$$

Here, $\widetilde{\boldsymbol{Y}}^*$ is approximated via centered flux like in Bassi and Rebay (1997). We also refer to Abdi et al. (2017b) for more details.

     For algorithmic efficiency, inexact quadrature is used to calculate the integrals above. By virtue of inexact integration and of equations (17), (19), and (24), the variational DG equations yield the semi-discrete matrix problem

$$245 \quad \frac{d\boldsymbol{Y}_i^e}{dt} = - \left(\nabla_{j,i}^T \right) \boldsymbol{F}_j^e + \boldsymbol{\mathcal{S}}_i^e + \frac{w_i{}^s |\mathbf{J}|_i^s}{w_i{}^e |\mathbf{J}|_i^e} \boldsymbol{n}_i^s \left(\boldsymbol{F}^e - \boldsymbol{F}^* \right)_i, \tag{31}$$

where w_i^s are interpolation weights. The algebraic details to obtain this expression can be found in Giraldo and Restelli (2008), where the s superscript indicates a value that is defined on the element boundary surface. The system (31) is integrated on each element with respect to time.

     In order to achieve good parallel scaling it is necessary to overlap communication and computation to the fullest extent

250 possible. With DG (and all element-based Galerkin methods) this can be naturally achieved by splitting Eq. (31) into terms that arise from the approximation of volume integrals and surface integrals. All volume contributions can be calculated independently of element-to-element communication regardless of the order of the spatial approximation, as can surface integrals that are not on elements which share boundaries across MPI ranks. Thus, in the code, we start with message passing interface (MPI) communication, do all volume calculations and surface calculations for elements not on boundaries shared across ranks, and

255 then apply surface calculations for elements on the rank boundaries after communication operations have been completed. This approach makes DG naturally effective with respect to parallel computing as previously shown by, e.g., Müller et al. (2018) on CPUs, Abdi et al. (2017b) on GPUs. At high order, element-based Galerkin methods such as DG require fewer neighboring degrees of freedoms than high-order finite difference and finite volume discretizations.

3.2   Time discretization

260 ClimateMachine provides a suite of time-integrators consisting of explicit Runge-Kunge methods, low-storage (Carpenter and Kennedy, 1994; Niegemann et al., 2012), strong stability-preserving (Shu and Osher, 1988) and additive Runge-Kutta (ARK) implicit-explicit (IMEX) methods (Giraldo et al., 2013; Kennedy and Carpenter, 2019).

The benchmarks presented in this paper with isotropic grid spacing are run using the 4th-order 14-stage method of Niegemann et al. (2012), which has a large explicit time-stepping stability region. One of the benchmarks, however, uses a highly anisotropic grid, which benefits from the use of a 1-D IMEX approximation; there we use a variant of the horizontally explicit, vertically implicit (HEVI) schemes by Bao et al. (2015). While 3-D IMEX is also an option, its performance in terms of time-to-solution is ultimately limited by the availability of scalable 3-D implicit solver algorithms.

4 Sub-grid scale models

The governing equations are resolved with the discretizations presented in Section 3. This leaves unresolved but dynamically significant scales on the computational grid that must be modeled with three-dimensional SGS models. We note that, while filtering over the grid volume is typically a formal requirement for LES variables, no explicit filter kernel is applied to the equations here. The resolution of the prognostic variables on a discretized grid constitutes an implicit filtering operation, with an equivalent lengthscale given by the grid resolution. In general, SGS fluxes are modeled as diffusive fluxes, which capture down-gradient transport of conservable scalar quantities assuming that mixing lengths are small compared with the scales over which the gradients of the scalars vary. We address the physical form of the diffusive flux components in equations (1)–(5), following which we describe standard models of subgrid-scale turbulence available for use in ClimateMachine.

The diffusive turbulent stress tensor $\boldsymbol{\tau}$ is represented in terms of the symmetric rate of strain tensor $\boldsymbol{S}$ such that

$$\boldsymbol{S}(\boldsymbol{u}) = \frac{1}{2}\left(\boldsymbol{\nabla}\boldsymbol{u} + (\boldsymbol{\nabla}\boldsymbol{u})^T\right),\tag{32}$$

with

$$\boldsymbol{\tau} = -(2\boldsymbol{\nu}_t\boldsymbol{S}).\tag{33}$$

Here, $\boldsymbol{\nu}_t$ is a turbulent viscosity tensor whose components are typically orders of magnitude larger than the molecular viscosity and are a function of the velocity gradient tensor.

The diffusive flux $\boldsymbol{d}_{q_t}$ of total water specific humidity in equation 2 is modeled as

$$\boldsymbol{d}_{q_t} = -(\boldsymbol{\mathcal{D}}_t\boldsymbol{\nabla}q_t),\tag{34}$$

where $\boldsymbol{\mathcal{D}}_t$ is a turbulent diffusivity vector. The turbulent diffusivity $\boldsymbol{\mathcal{D}}_t$ is related to the turbulent viscosity tensor $\boldsymbol{\nu}_t$ via the turbulent Prandtl number such that

$$\boldsymbol{\mathcal{D}}_t = \frac{\text{diag}(\boldsymbol{\nu}_t)}{\text{Pr}_t},\tag{35}$$

where Pr_t takes a typical value of $\text{Pr}_t = 1/3$.

The unresolved flux of total enthalpy h^{tot} results in a diffusive subgrid flux term of the form

$$\boldsymbol{J} + \boldsymbol{D} = -(\boldsymbol{\mathcal{D}_t}\boldsymbol{\nabla}h^{\text{tot}}),\tag{36}$$

where $\boldsymbol{J}$ is the thermal diffusion flux analogous to the molecular conductive heat flux, and $\boldsymbol{D}$ is the energy flux carried by water vapor, defined in equation 11. For energetic consistency, we use the same turbulent diffusivity $\mathcal{D}_t$ for moist enthalpy and water.

4.1 Smagorinsky-Lilly model

The turbulent eddy viscosity ν_t in the model by Lilly (1962) and Smagorinsky (1963) (SL henceforth) is defined by means of the magnitude of the rate of the strain tensor $\boldsymbol{S}$, whose components are $S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$, according to

$$\nu_t = (C_s \Delta)^2 \sqrt{2 S_{ij} S_{ij}}, \tag{37}$$

for $i, j = 1, 2, 3$; C_s is a constant Smagorinsky coefficient usually within the range $0.12 < C_s < 0.21$; and Δ is the LES filter-width. Inside each hexahedral element of order N and side lengths $L_{x,y,z}$ along the x, y, z directions, the effective grid resolution is $\Delta(x, y, z) = L_{x,y,z}/N$, which is the average distance between two consecutive nodal points. We use an isotropic eddy viscosity tensor $\boldsymbol{\nu_t}$ in LES, with its components defined by equation (37). Similar eddy viscosity models can be found in other recent models of compressible atmospheric flows (e.g., Jähn et al. (2015); Shi et al. (2018)).

4.2 Vreman eddy viscosity model

The SGS model developed by Vreman (2004) is of interest because of its robustness across flow regimes and because it has low dissipation near wall boundaries and in transitional flows. Its computational complexity is similar to the classical SL model. While the Vreman model is extensively used in engineering LES, it is uncommon in atmospheric flows, where a constant coefficient SL or the 1-equation TKE model by Deardorff (1970, 1980) are the most common choices (see, e.g., Stevens et al. (2005)).

The turbulent eddy viscosity of this model depends on first-order derivatives of velocities and is given by

$$\nu_t = 2.5 C_s^2 \sqrt{\frac{B_\beta}{u_{i,j} u_{i,j}}}, \tag{38a}$$

where

$$B_\beta = \beta_{11}\beta_{22} + \beta_{11}\beta_{33} + \beta_{22}\beta_{33} - (\beta_{13}^2 + \beta_{12}^2 + \beta_{23}^2), \tag{38b}$$

$$\beta_{ij} = \Delta_m^2 u_{i,m} u_{j,m}, \tag{38c}$$

$$u_{i,j} = \frac{\partial u_i}{\partial x_j}. \tag{38d}$$

Here, summation over repeated indices $i, j \in \{1, 2, 3\}$ is implied, C_s is the constant Smagorinsky coefficient, and u_i represent the components of the resolved-scale velocity vector, so that $u_{i,j}$ is the velocity gradient tensor. The mixing lengths Δ_m can be determined as the grid spacing in the direction implied by subscript m; in this paper $\Delta_m \equiv L_{x,y,z}/N$. Near-surface corrections to the characteristic length (e.g. Mason and Callen (1986)) have not been applied to either subgrid-scale model in the present work.

Richardson correction in stable regions of the atmosphere To account for the atmospheric stability and, effectively, reduce turbulence generation to zero in stably stratified atmospheres, we multiply the eddy viscosity by the correction factor

$$
f_b = \begin{cases} 1 & \mathrm{Ri} \leq 0, \\ \max(0, 1 - \mathrm{Ri}/\mathrm{Pr}_t)^{1/4} & \mathrm{Ri} > 0 \end{cases} \tag{39}
$$

where

$$
\mathrm{Ri} = \frac{\left(\frac{g}{\theta_v} \frac{\partial \theta_v}{\partial z} \right)}{|\mathbf{S}|^2}.
$$

Here, $\mathrm{Pr}_t = 1/3$ is a constant turbulent Prandtl number, and θ_v is the virtual potential temperature for a given specific humidity q_t and specific liquid-water content q_l (see, e.g., Deardorff (1980)). This temperature is computed consistently with the moist phase partitioning (saturation adjustment) in the system, and therefore is a reasonable measure of the effect of buoyancy on flow stratification. This is applied to both turbulence closures discussed in this manuscript.

4.3 Numerical stability

When high-order Galerkin methods are used to solve non-linear advection dominated problems, spurious Gibbs oscillations may affect the solution and need to be addressed. ClimateMachine provides a set of spectral filters, cut-off filters, and artificial diffusion methods to remove these oscillations. While filters may be effective, we found that stabilizing the LES solution by means of the SGS eddy viscosity alone is effective and robust; this is in agreement with results shown by Marras et al. (2015); Marras and Giraldo (2015) and Reddy et al. (2021) in the case of continuous Galerkin methods. Other methods of stabilizing numerical solutions are discussed by Light and Durran (2016), Yu et al. (2015). This approach stems from the idea that under-resolved scales are responsible for the numerical oscillations observed in solutions through the introduction of unphysical artifacts in properties of the solution variables. Detailed analyses of the interactions of subgrid-scale models and filtering techniques with DG numerics in the context of atmospheric flows will be presented in a forthcoming paper.

5 Numerical experiments and discussions

We first demonstrate the convergence of the numerical solution to the Euler equations with an isentropic vortex advection problem. Following this, we demonstrate results from the ClimateMachine using standard benchmark problems including: (1) dry rising thermal bubble in a neutrally stratified atmosphere; (2) dry density current; (2) hydrostatic and non-hydrostatic mountain-triggered linear gravity waves; (4) the Barbados Oceanographic and Meteorological Experiment (BOMEX); and (5) decaying Taylor-Green vortex in a triply periodic domain.

5.1 Advection of an isentropic vortex

To demonstrate convergence of the numerical solutions, we consider the two-dimensional dry isentropic vortex advection problem with fourth order polynomials, consistent with the benchmark cases shown in the sections that follow. We consider

the pure advection of a vortex in a domain with edge lengths $L = 0.1$ m, with a vortex radius $R = 0.005$ m. The initial velocity profile is given by

$$\boldsymbol{u} = (u_0 \cos\alpha + \delta_u, u_0 \sin\alpha + \delta_v), \tag{40}$$

with a prescribed translational velocity $u_0 = 150$ ms^{-1}, translation angle $\alpha = \pi/4$, and a vortex speed $v_s = 50$ ms^{-1} so that the velocity perturbations are given by

$$\delta_u = -v_s x \frac{y'}{R} \exp\left(\frac{r}{2R}\right)^2, \tag{41}$$

$$\delta_v = v_s x \frac{x'}{R} \exp\left(\frac{r}{2R}\right)^2. \tag{42}$$

The perturbations are computed with an offset coordinate given by

$$x' = x - 2L \, \text{floor}(\frac{x+L}{2L}), \tag{43}$$

$$y' = y - 2L \, \text{floor}(\frac{y+L}{2L}), \tag{44}$$

where x and y represent Cartesian coordinates, to ensure that the resulting functions are periodic. The initial temperature profile is given by

$$T = T_\infty \left(1 - \kappa_d \frac{v_s^2}{2} \frac{\rho_\infty}{p_\infty} \exp\left(-\frac{r}{R}\right)^2\right), \tag{45}$$

with a reference temperature of $T_\infty = 300$ K, reference pressure $p_\infty = 10^5$ Pa, and the corresponding density ρ_∞ computed from the ideal gas law. The simulation time is given by $t_{\text{end}} = L/(10u_0)$, We consider fourth order polynomials, with increasing refinement given by the addition of elements in the domain discretisation, to generate the Euclidean distance between the initial state, and the time-stepped solution state at $t_{\text{end}} = 10^{-4}$ s, as shown in Figure 1. The results demonstrate consistent convergence across three numerical flux methods, for element numbers ranging from $5 - 320$. The convergence rate approaches 5 for the fourth order polynomials considered in this study.

Tests presented in sections 5.2 – 5.5 are executed in a 3D domain even when the problem is two dimensional, in which case we have effectively zero tendencies in the third dimension; this setup is identified as 2.5D in what follows.

5.2 2.5D Rising thermal bubble in a neutrally stratified atmosphere

A neutrally stratified atmosphere with uniform background potential temperature $\theta_0 = 300$ K is perturbed by a circular bubble of warmer air. The hydrostatic background pressure decreases with z as

$$p = p_0 \left(1 - \frac{g}{c_{pd}\theta_0} z\right)^{c_{pd}/R_d} \tag{46}$$

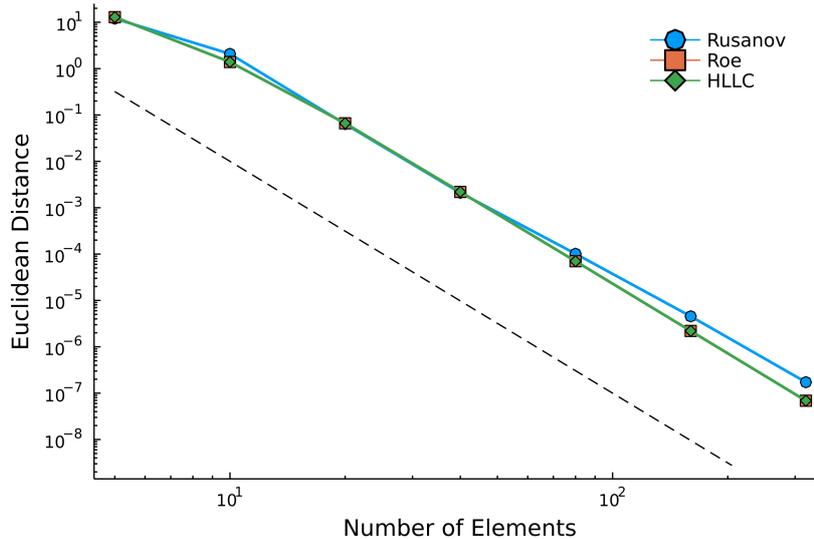



Figure 1. Euclidean distance between initial and final solution vectors of the prognostic variables, generated using Rusanov (octagons), Roe (squares) and Harten-Lax-van Leer-Contact (HLLC) numerical fluxes. Final solutions are evaluated at $t_{\mathrm{end}} = 10^{-4}$ s, based on a translational velocity of 150 m s^{-1}. For each numerical flux method, we consider $N_e = [5, 10, 20, 40, 80, 160, 320]$ elements with fourth-order polynomials. The dashed line represents N_e^{-5} convergence.

in a domain $\Omega = [0, 10000] \times [-\infty, \infty] \times [0, 10000]$ m^3. The perturbation is as defined in Ahmad and Lindeman (2007),

$$\Delta\theta = \theta_c \left[1.0 - \frac{r}{r_0} \right] \qquad \text{if } r \leq r_0 = 2000 \text{ m}, \tag{47}$$

where $r = \sqrt{(x - x_c)^2 + (z - z_c)^2}$, $(x_c, z_c) = (5000, 2000)$ m, and $\theta_c = 2$ K. The initial velocity field is zero everywhere. Periodic boundary conditions are used along y, and solid walls with impenetrable, free-slip boundary conditions are used in the x and z directions. Detailed information on boundary conditions for all test cases is provided in Appendix A. Five runs are performed at effective uniform resolutions $\Delta x = \Delta z = 250$ m, 125 m, 62.5 m, and 31.25 m, and 15.625 m, with polynomial order $N = 4$. Potential temperature θ and the two velocity components u and w are plotted at $t = 1000$ s in Figure 2 for the grid resolution of 15.625 m, which represents a reference solution for comparison with solutions at coarser resolutions. The value of the maximum potential temperature perturbation $\Delta\theta_{\max}$ and of the horizontal and vertical velocity components agree with the 125 m resolution results shown by Ahmad and Lindeman (2007). The grid dependence of the solution is shown in Figure 3, where potential temperature is plotted for $\Delta x = \Delta z = 31.25$ m, 62.5 m, 125 m, and 250 m. While the solution is visibly more dissipative at coarser resolutions, the bubble's leading edge position (and hence propagation speed) is not sensitive to the grid resolution.

The SL and Vreman closures are used to model diffusive fluxes in this problem. The solutions show no discernible differences, and only the SL solution is shown. A visual comparison of the two becomes more meaningful when shear triggers mixing, which is shown for the density current test in Section 5.3. Although DG inherits an implicit numerical diffusion as an

effect of the numerical flux calculation across elements, under-resolved advection-dominated problems still require a dissipation or filtering mechanism to preserve the solution's stability. In the case of Marras et al. (2015), a dynamically adaptive SGS model was used whereas a Boyd-Vandeven filter (Boyd, 1996; Vandeven, 1991) was used by Giraldo and Restelli (2008).

(a) Potential temperature θ (b) Horizontal velocity u (c) Vertical velocity w

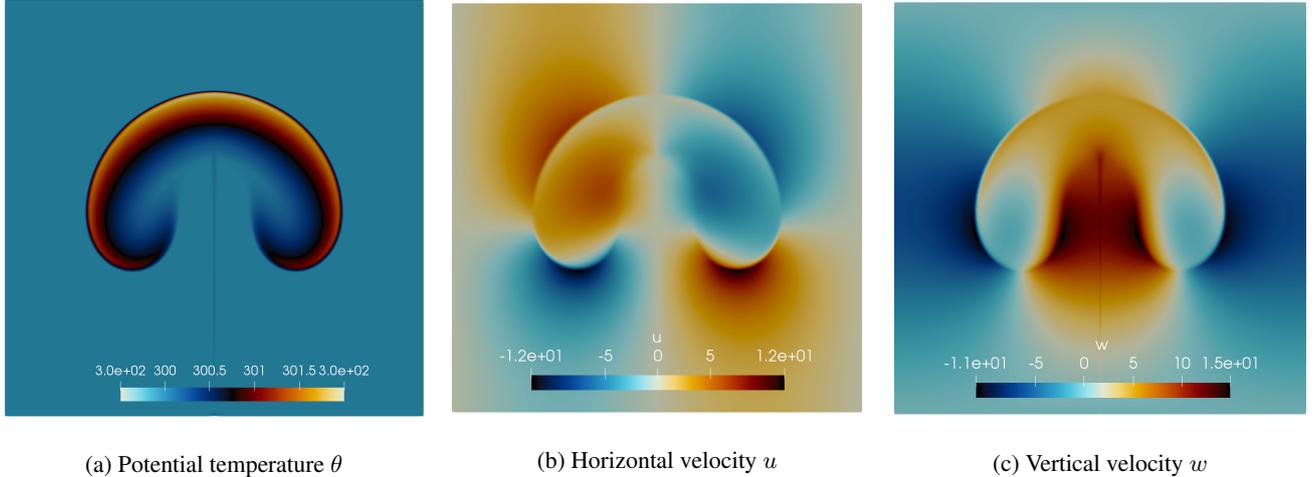



Figure 2. 2.5D rising thermal bubble with effective resolution $\Delta x = \Delta z = 15.625$ m and $N = 4$. Panels depict (a) potential temperature θ, (b) horizontal velocity u, and (c) vertical velocity w at t=1000 s in $\Omega = 10 \times 10$ km^2.

5.3 2.5D Density current in a neutrally stratified atmosphere

The density current problem by Straka et al. (1993) is used to test the LES framework in a flow with Kelvin-Helmholtz instabilities. As for the rising thermal bubble, the background initial state is in hydrostatic equilibrium at uniform potential temperature $\theta_0 = 300$ K. A perturbation of θ centered on $(x_c, z_c) = (0, 3000)$ m and with radii $(r_x, r_z) = (4000, 2000)$ m is given by the function

$$\Delta\theta = \frac{\theta_c}{2}\left[1 + \cos(\pi_c r)\right] \qquad \text{if } r \leq 1 \tag{48}$$

where $\theta_c = -15$ K and $r = \sqrt{(x - x_c)/r_x^2 + (z - z_c)/r_z^2}$ in the domain $\Omega = [0, 25600] \times [-\infty, \infty] \times [0, 6400]$ m^3. Periodic boundary conditions are used along y; impenetrable free-slip conditions are imposed in x and z. The flow is initially stationary.

To reach solution grid convergence, this test is classically executed with a constant kinematic viscosity $\nu = 75$ m^2 s^{-1}. Increasingly finer structures are resolved when the resolution increases (Marras et al., 2012, 2015). A measure of solution fidelity is the front position, which we compare against other models in Appendix D for different resolutions. The ClimateMachine results show quantitative agreement with respect to the frontal location from a range of models with varying spatial discretizations at resolutions ranging from 12.5 m to 100 m. This demonstrates the scope for capturing small-scale flow features with the numerics described in Section 3.

The structure of potential temperature at the final time $t = 900$ s is shown in Figure 4 for the Vreman and SL solutions. When Rusanov is the chosen numerical flux (Figures 4a and 4b), the solutions are very similar although Vreman is visibly

(a) 31.25 m resolution   (b) 62.5 m resolution

(c) 125 m resolution   (d) 250 m resolution

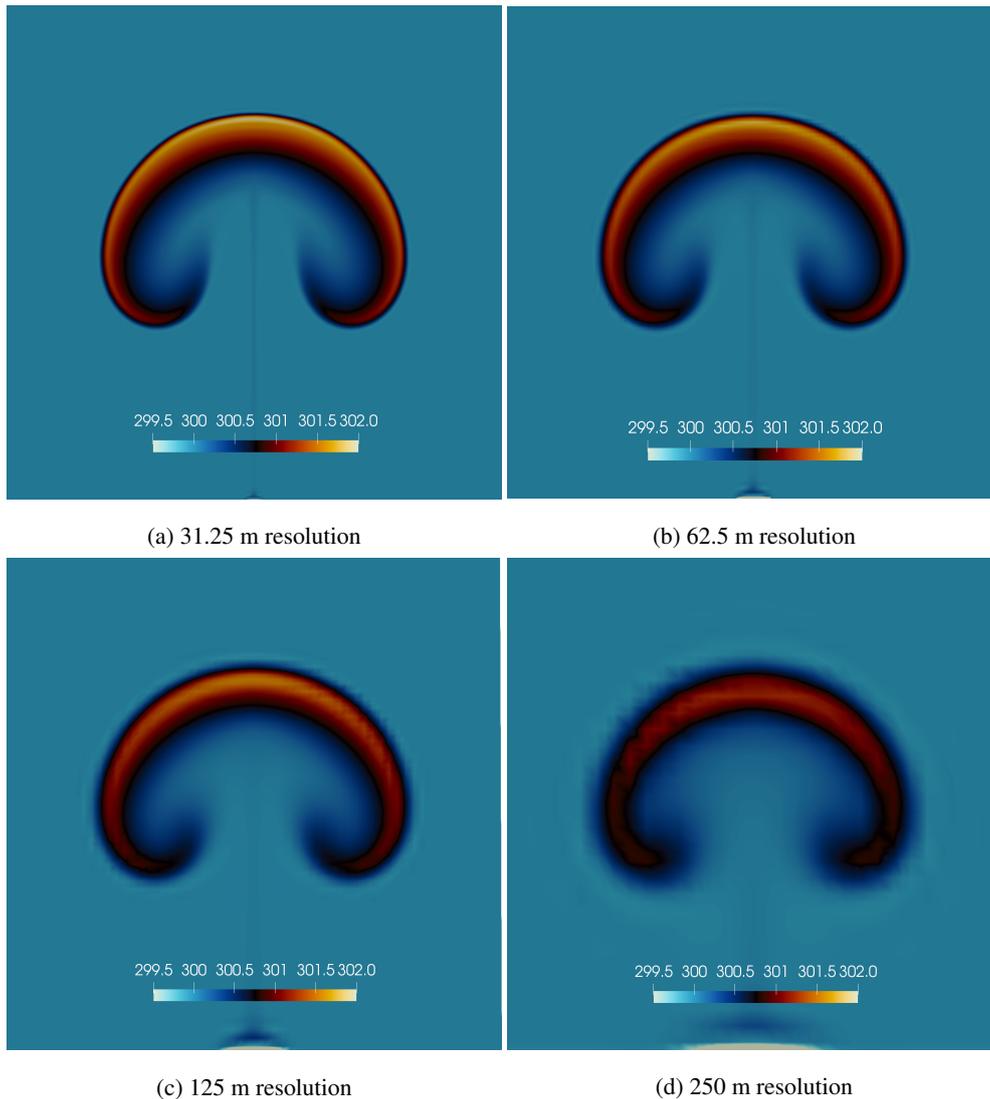



Figure 3. 2.5D rising thermal bubble solution with $N=4$ at decreasing effective resolution. Grid convergence of potential temperature θ at four different resolutions to be compared against the 15.625 m resolution results shown in Figure 2. From left to right: $\Delta x = \Delta z = 31.25$ m, 62.5 m, 125 m, and 250 m. Although the solution is visibly more dissipated at coarser resolution, the bubble's leading edge (and hence propagation speed) is not affected.

less dissipative for a prescribed value of the Smagorinsky coefficient $C_s = 0.18$, with finer scales of motion apparent in the contours of potential temperature. Further quantitative analysis of the dissipative properties of the SGS models is reported in Section 5.6. Despite Vreman being less dissipative than SL, the Roe (Figure 4c) and HLLC (Figure 4d) fluxes contribute to additional numerical diffusion when compared to Rusanov fluxes. Detailed analysis of the interaction between numerical fluxes and subgrid-scale models will be presented in future articles.

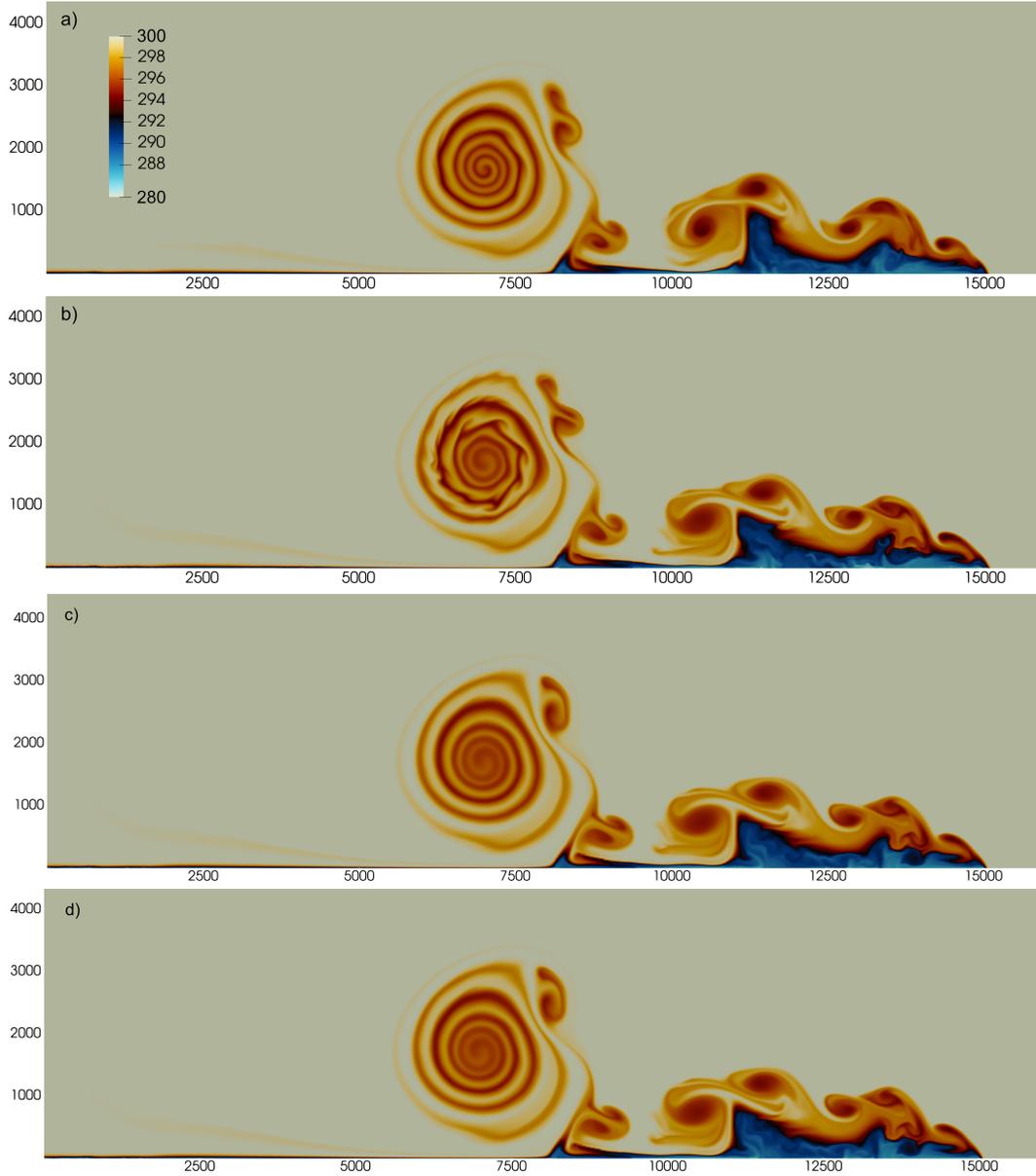



Figure 4. 2.5D density current. Potential temperature θ (K) at $t = 900$ s computed with an effective DG resolution of $\Delta x = \Delta z = 12.5$ m with domain extents shown in meters. (a) Rusanov numerical flux with SL SGS. (b) Rusanov numerical flux with Vreman SGS. (c) Roe numerical flux with Vreman SGS. (d) HLLC numerical flux with Vreman SGS. The color scale ranges from $\theta = 285$ to 300 K. Shared colorbar for all plots shown in panel (a) for clarity.

5.4 Passive transport over warped grids

To verify the correct behavior of the DG implementation in the presence of topographic features, the simple passive advection test described by Schär et al. (2002) is used. The conservation law for the diffusive transport of a passive tracer χ is

$$\frac{\partial(\rho\chi)}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}\chi) = -\nabla \cdot (\rho \boldsymbol{d}_\chi), \tag{49}$$

which is approximated via DG in the same way as Eq. (16). For a scalar tracer variable χ, we model diffusive fluxes $\boldsymbol{d}_\chi$ such that

$$\boldsymbol{d}_\chi = -(\delta_\chi \boldsymbol{\mathcal{D}}_t \nabla \chi), \tag{50}$$

where δ_χ relates the ratio of turbulent diffusivity of the tracer to that of the energy and moisture variables. For this test case, however, tracer diffusivity is set to zero to assess the stability and transport properties when using a warped grid. The volume grid in ClimateMachine is built by stacking elements above the surface and warping them around the terrain profile. To reduce the element distortion across the domain, a linear grid damping function is used such that a topography conforming surface of nodal points near the domain's bottom surface decays to a horizontal plane at higher altitudes (Gal-Chen and Somerville, 1975).

The initial scalar field χ is described by an elliptical perturbation centered on $(x_c, z_c) = (25, 9)$ km, with radii $(r_x, r_z) = (25, 3)$ km, such that

$$\chi = \begin{cases} \chi_0 \cos^2\left(\frac{\pi r}{2}\right), & \text{for } r \leq 1, \\ 0 \text{ otherwise} \end{cases} \tag{51}$$

where $\chi_0 = 1$ and $r = \sqrt{(x - x_c)/r_x^2 + (z - z_c)/r_z^2}$ in the domain $\Omega = [0, 150000] \times [-\infty, \infty] \times [0, 30000] \ \mathrm{m}^3$. An effective uniform grid resolution $\Delta x = \Delta z = 500$ m is used for this test. The initial velocity profile is given by

$$u(z) = u_0 \begin{cases} 1, & \text{for } z \geq z_2 \\ \sin\left(\frac{\pi}{2}\frac{z - z_1}{z_2 - z_1}\right), & \text{for } z_1 \leq z < z_2, \\ 0, & \text{for } z < z_1, \end{cases} \tag{52}$$

where $u_0 = 10 \ \mathrm{m \ s^{-1}}$, $z_1 = 4$ km, and $z_2 = 5$ km.

The topography is defined by the function

$$z_{\mathrm{sfc}}(x) =$$

$$\begin{cases} h_0 \cos^2\left(\frac{\pi(x - x_0)}{2a}\right) \cos^2\left(\frac{\pi(x - x_0)}{\lambda}\right) & \text{for } |x - x_0| \leq a \\ 0, & \text{for } |x - x_0| > a, \end{cases} \tag{53}$$

where $h_0 = 3$ km, $a = 25$ km, $\lambda = 8$ km, and $x_0 = 75$ km.

The contours of χ in Figure 5 show minimal distortion in spite of the warped elements directly above the topographical feature, indicating that the DG transformation metrics from physical to logical space in the presence of topography do not adversely affect the solution. Deviation from the initial profile of tracer magnitudes lie between -4% and +2% when the tracer is above the topographical feature, with maximum deviation amplitudes of -5% and +3% at the end of the test, showing a favorable comparison with the hybrid and SLEVE coordinate results presented in Schär et al. (2002).

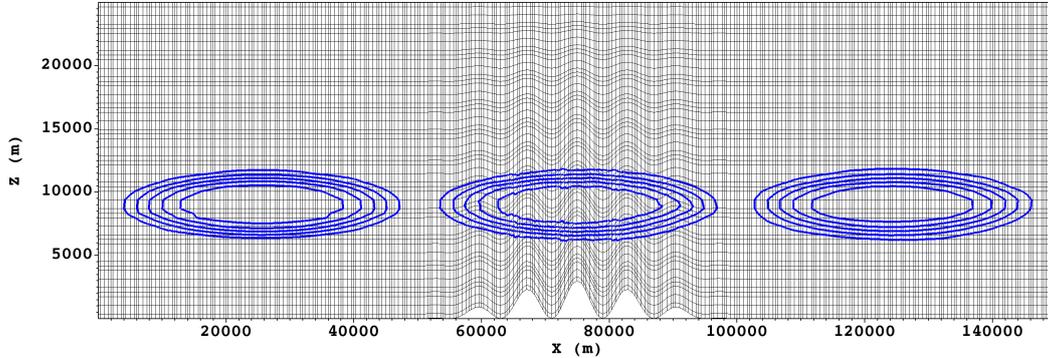



Figure 5. Solution of the passive transport of scalar χ at three different time instances, with contours of tracer quantity χ (from 0, outermost contour, to 1, innermost contour) overlaid on a representation of nodal points on the underlying mesh. Tracer advection is driven by a prescribed velocity profile from left to right. As it crosses the deformed grid above the mountain ridge, only a minimal distortion of tracer contours is observed, which is completely recovered back to a smooth solution downwind of the ridge.

5.5 Mountain-triggered gravity waves

To assess the correct implementation of a Rayleigh sponge layer to attenuate fast, upward propagating gravity waves before they reach the top of the domain, two steady-state mountain-triggered gravity wave problems suggested by Smith (1980) are solved. The sponge layer is described in Appendix A2. These problems consist of a flow that moves eastward with uniform horizontal velocity $\boldsymbol{u} = (u,0,0)$ m s^{-1} in a doubly periodic domain. The flow impinges against a mountain of height h_m and base length a centered at x_c as

$$z_{\mathrm{sfc}}(x) = \frac{h_m a^2}{(x - x_c)^2 + a^2}. \tag{54}$$

The background state is in hydrostatic balance with Brunt-Väisälä frequency $\mathcal{N}$, such that

$$\theta = \theta_{\mathrm{sfc}} \ \exp\left(\frac{\mathcal{N}^2}{g}z\right)$$

for a given surface potential temperature $\theta_{\mathrm{sfc}} = T_{\mathrm{sfc}}$. The hydrostatically balanced pressure is

$$p = p_{\mathrm{sfc}}\left[1 + \frac{g^2}{c_{pd}\theta_{\mathrm{sfc}}\mathcal{N}^2}\left(\exp\left(\frac{-z\mathcal{N}^2}{g}\right) - 1\right)\right]^{c_{pd}/R_d} \tag{55}$$

which yields, by means of the ideal gas law, the background density

$$\rho = \frac{p_{\mathrm{sfc}}}{\theta R_d \left(\frac{p}{p_{\mathrm{sfc}}} \right)^{c_{vd}/c_{pd}}}. \tag{56}$$

These tests are affected by spurious oscillations that appear at approximately 5000 s into the simulation. In the absence of shear, because of the free-slip bottom and top boundaries, the SGS models are unable to introduce sufficient diffusion to remove the Gibbs modes so that an exponential filter (Hesthaven and Warburton, 2008b) of order 64 was applied on the velocity field to remove spurious modes. The filter assumes the form of

$$\sigma(\eta) = e^{-\alpha \eta^s}, \tag{57}$$

where s is the filter order, η is a function of the polynomial order, and $\alpha = -\log(\varepsilon_M)$ is a parameter that controls the smallest value of the filter function for machine precision ε_M. In double precision, $\alpha \approx 36$. The filter in this form is applied to perturbations of the prognostic variables from the balanced background state.

5.5.1 Linear hydrostatic

The linear hydrostatic case proposed by Smith (1979) consists of a neutrally stratified isothermal atmosphere with $\theta = \theta_{\mathrm{sfc}} = 250$ K. The background atmosphere is isothermal with temperature T_0, resulting in a Brunt-Väisälä frequency of

$$\mathcal{N} = \frac{g}{\sqrt{c_{pd}T_0}}.$$

The flow moves in a periodic channel along the x-direction with velocity $\boldsymbol{u} = (20 \text{ m s}^{-1}, 0, 0)$ over a mountain with $h_m = 1$ m and $a = 10000$ m. A Rayleigh absorbing layer is added at $z_{\mathrm{s}} = 25$ km with relaxation coefficient $\alpha = 0.5$ s^{-1}, power $\gamma = 2$ and domain top $z_{\mathrm{top}} = 30$ km (see Appendix A2 for details). The domain extends from 0 to 240 km in the horizontal direction.

The steady-state solution at $t = 15000$ s is shown in Figure 6a. It is consistent with the DG results shown by Giraldo and Restelli (2008).

5.5.2 Linear non-hydrostatic

The linear non-hydrostatic mountain waves are forced by a flow of uniform horizontal velocity $\boldsymbol{u} = (10 \text{ m s}^{-1}, 0, 0)$ over a mountain with $h_m = 1$ m and $a = 1000$ m. The domain extends from 0 to 144 km in the horizontal direction and is 30 km high.

The steady-state solution at $t = 18000$ s is shown in Figure 6b. It is consistent with results shown by Giraldo and Restelli (2008).

5.6 Decaying Taylor-Green Vortex

The decaying Taylor-Green vortex (TGV) is a classical test to estimate the dissipative properties of turbulence models in the absence of solid boundaries. The gravity-free flow is initialized in a triply periodic cube of dimensions $[-\pi, \pi]^3$. The solenoidal

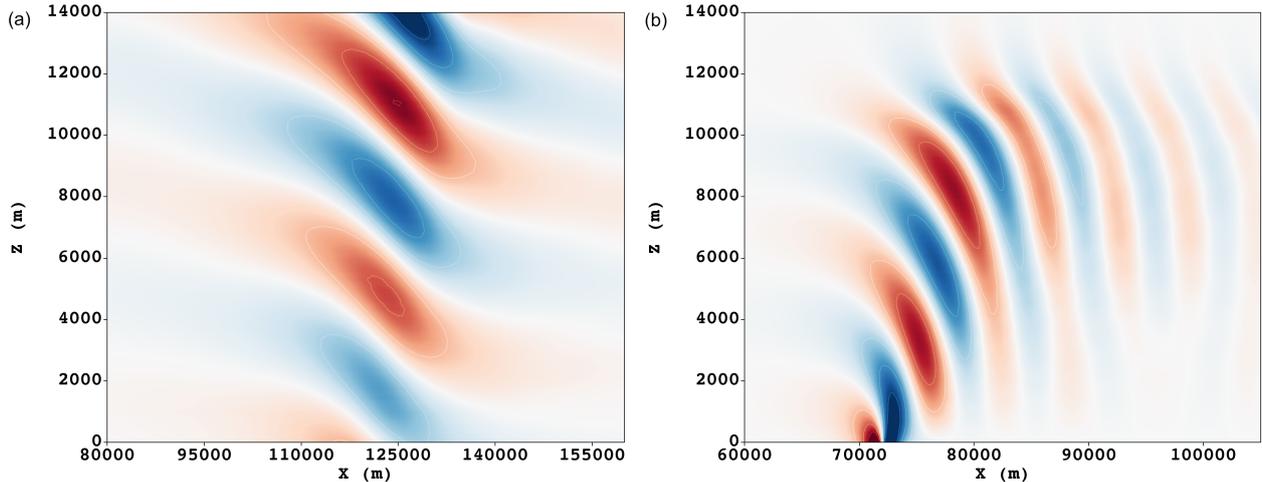



Figure 6. Vertical velocity w for two linear mountain wave tests. Velocity contours are shown in the range from -4×10^{-3} m s^{-1} (blue) to 4×10^{-3} m s^{-1} (red). (a) Hydrostatic test at $t = 15000$ s, with an effective horizontal resolution $\Delta x = 500$ m and effective vertical resolution $\Delta z = 240$ m with $N = 4$. (b) Non-hydrostatic test at $t = 18000$ s, with an effective horizontal resolution $\Delta x = 200$ m and effective vertical resolution $\Delta z = 160$ m with $N = 4$.

initial velocity field $\boldsymbol{u_0} = (u_0, v_0, w_0)$ is defined as

$$u_0 = U_0 \sin(kx) \cos(ky) \cos(kz), \tag{58}$$

$$v_0 = U_0 \cos(kx) \sin(ky) \cos(kz), \tag{59}$$

$$w_0 = 0, \tag{60}$$

with initial pressure

$$p_0 = p_\infty + \frac{\rho_0 U_0^2}{16}(2 + \cos(kz))(\cos(2kx) + \cos(2ky)), \tag{61}$$

where k is the wavenumber, $U_0 = 100$ m s^{-1}, $\rho_0 = 1.178$ kg m^{-3}, and $p_\infty = 101325$ Pa. Fourth-order polynomials are used for all simulations considered in this section.

We first consider the volume-averaged kinetic energy, which provides insight into the dissipation characteristics of the flow with respect to non-dimensionalized time $t^* = kU_0 t$. In integral form, the kinetic energy can be written as:

$$E_k = \frac{1}{2}\langle|\boldsymbol{u}|^2\rangle = \frac{1}{2\Omega^h}\int_{\Omega^h} \boldsymbol{u} \cdot \boldsymbol{u}\, d\Omega^h, \tag{62}$$

where $\langle\cdot\rangle$ denotes a volumetric average over the volume Ω^h. If the flow is inviscid, the kinetic energy should be conserved. This is only valid if the numerics or SGS models do not introduce numerical dissipation, or if all flow scales are well resolved. As such, the time series of kinetic energy is a metric that shows the point along the simulation at which the solution becomes

under-resolved. The kinetic energy dissipation rate is the second quantity of interest, which allows us to quantify the rate of decay of kinetic energy over time. This is defined as

$$\epsilon = -\frac{dE_k}{dt}. \tag{63}$$

A third quantity of interest for this analysis is enstrophy, which is defined as the square of the vorticity norm:

$$\langle \omega^2 \rangle = \langle ||\nabla \times \boldsymbol{u}||^2 \rangle. \tag{64}$$

The enstrophy of a fully resolved flow should go to infinity if the flow is inviscid. Therefore, enstrophy can be used as a criterion to estimate the effect of numerical dissipation.

By means of a three-dimensional fast fourier transform (FFT) of the velocity field, the kinetic energy spectrum is calculated as:

$$E(k) = \int\limits_0^{2\pi} \int\limits_0^{\pi} \int\limits_0^{K} A(k_x, \upsilon, \zeta) k^2 \sin(\zeta) \, dk \, d\upsilon \, d\zeta, \tag{65}$$

where $K = 2\pi/L$, L is the characteristic length, A is a three dimensional array of Fourier mode amplitudes, $\upsilon = k_z/k$ and $\zeta = \tan^{-1}(k_y/k_x)$ and $k = \sqrt{k_x^2 + k_y^2 + k_z^2}$. The TGV flow is simulated using both the SL and Vreman models on grids with 32^3, 64^3, 128^3, and 192^3 points. Figure 7 shows a 3D visualization of the flow at two different non-dimensional times using zero Q-criterion isosurfaces, which identify balance between rotation and shear in the flow (Hunt et al., 1988). As the flow evolves, the flow generates smaller and smaller-scale vortices. Eventually, the flow becomes under-resolved, making it impossible to conserve kinetic energy. As the flow continues to evolve, an instability occurs, which causes the disintegration of the vortex sheet. After this point, the TGV's dynamics are controlled by the interaction of small-scale vortical structures formed by vortex stretching.

Results for the coarse-resolution simulations are presented in Figure 8, and those for the fine-resolution simulations are shown in Figure 9. Figure 8a shows that the kinetic energy changes with time for the 32^3 resolution simulations are distinctly different from their higher-resolution counterparts in Figure 9a. The severe under-resolution of the flow seems to generate much larger amounts of dissipation early on. This is further demonstrated when comparing Figures 8b and 9b. Beyond 64^3 resolution, the peaks in dissipation are larger as the resolution increases, since smaller vortices can be resolved before the instability eventually happens. The 32^3 simulations (Figure 8b) have larger peaks than even the 192^3 (Figure 9b) simulations, demonstrating that the under-resolution of the vortex structures leads to different, more dissipative early-flow behavior.

Figure 9b shows that the 128^3 and 192^3 simulations using both SGS closures are characterized by a peak in the kinetic energy dissipation at $t^* = 9$. Brachet et al. (1983) and Brachet (1991) demonstrate this result for their direct numerical simulations (DNS) of the Taylor-Green vortex for a Reynolds number $R_e = U_0/k\nu \geq 3000$. Examining Figure 8b, the 64^3 simulations suggest a dissipation peak time of $t^* = 8$, and the 32^3 simulations suggest a dissipation peak time of $t^* = 6$. The under-prediction of the time at which the peak occurs is due to an inability to resolve the vortices that appear early on in the flow's evolution at extremely coarse resolutions. This leads to the early appearance of the instability which causes the dissipation

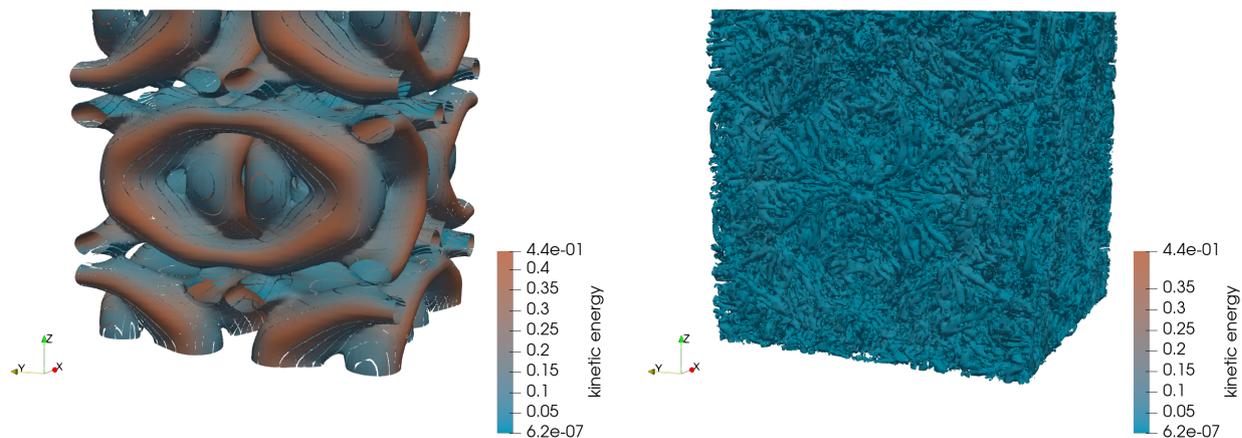



Figure 7. Taylor-Green Vortex. Isosurfaces of zero Q-criterion (these idenfity surfaces where the vorticity norm is identical to the strain-rate magnitude) on a 192^3 grid, corresponding to 48 elements of order 4 at $t^* = 4$ (left) and $t^* = 60$ (right). Plots are shown for the Vreman solution. The isosurfaces of the Q-criterion are colored by dimensionless kinetic energy.

peak. Furthermore, we see that the kinetic energy decay occurs sooner for the lower-resolution simulations, as a result of increased dissipation from the SGS models.

530   Figure 9c also shows that the flow's enstrophy behaves as expected, with peak values at $t^* = 9$ coinciding with peaks in kinetic energy. The higher-resolution simulations are able to reach a higher enstrophy than the lower-resolution simulations as they are naturally able to resolve more vortical motion. On the other hand, the choice of SL or Vreman models seems to have very little impact on the ability to resolve more small-scale eddies for low resolutions. However, the SL SGS scheme leads to higher enstrophy than the Vreman SGS scheme, increasingly so as resolution increases.

535   Figure 10 shows the kinetic energy spectra obtained for the higher-resolution simulations used for this test. All simulations present peaks in their respective spectra at $k = 2$ to 4, which persist even as the flow evolves over time. These peaks are explained by Drikakis et al. (2007) as being imprinted on the spectra by the initial velocity field. Furthermore, as the flow evolves, all spectra show a slope close to the theoretical $k^{-5/3}$ of homogeneous turbulence and eventually decay towards a k^{-3} slope at higher wavenumbers. This behavior is consistent with the DNS results of Brachet et al. (1983) who showed, using

540   DNS, this transition occurs around $k=60$.

5.7 Barbados Oceanographic and Meteorological Experiment (BOMEX)

BOMEX features a shallow cumulus topped boundary layer as described in Holland and Rasmusson (1973). The setup of this test follows Siebesma et al. (2003). The initial profiles are characterized by a well-mixed sub-cloud layer below 500 m, a cumulus layer between 500 m and 1500 m, an inversion layer up to 2000 m, and a free troposphere above. Large-scale

(a) Kinetic energy

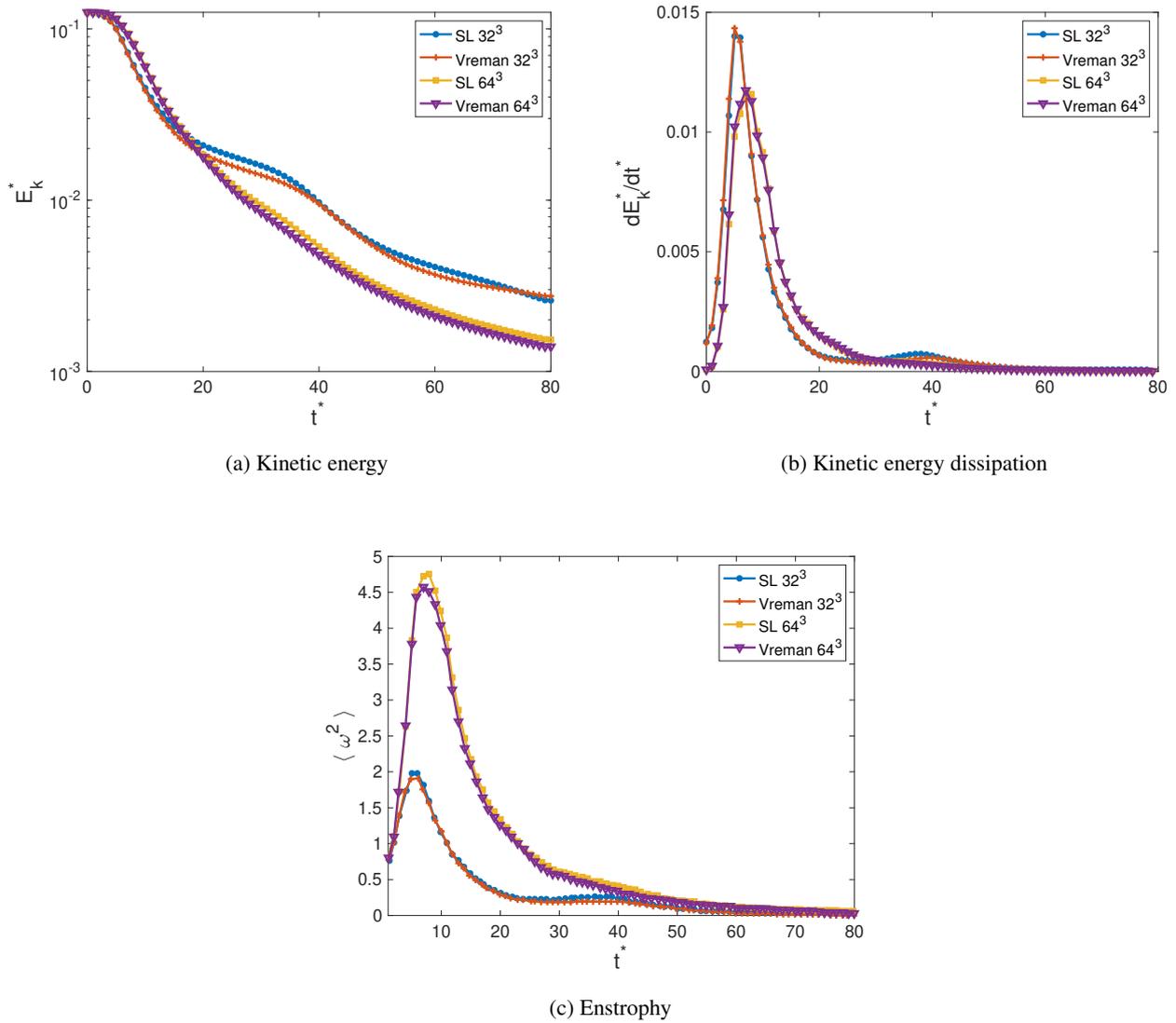
(b) Kinetic energy dissipation

(c) Enstrophy

Figure 8. Evolution of volumetrically averaged (a) kinetic energy represented on a semi-logarithmic scale, (b) kinetic energy dissipation, and (c) enstrophy, each computed on 32^3 and 64^3 grids with the Vreman and SL SGS models.

545 forcing includes prescribed large-scale subsidence, horizontal advective drying, radiative cooling, and pressure gradient effects. Sensible and latent heat fluxes at the surface are prescribed to $\text{SHF} = \boldsymbol{n} \cdot (\rho \boldsymbol{J}_{\text{sfc}}) = 9.5 \text{ W m}^{-2}$ and $\text{LHF} = \boldsymbol{n} \cdot (\rho \boldsymbol{D}_{\text{sfc}}) = 147.2 \text{ W m}^{-2}$. Additional detail on the application of boundary conditions is presented in Appendix A. The domain, $\Omega = 6400 \times 6400 \times 3000 \text{ m}^3$, is doubly periodic in the x and y directions. A Rayleigh sponge layer (see Appendix A2 for details) is applied along the z direction to damp upward propagating gravity waves. On the bottom surface, a momentum drag forcing

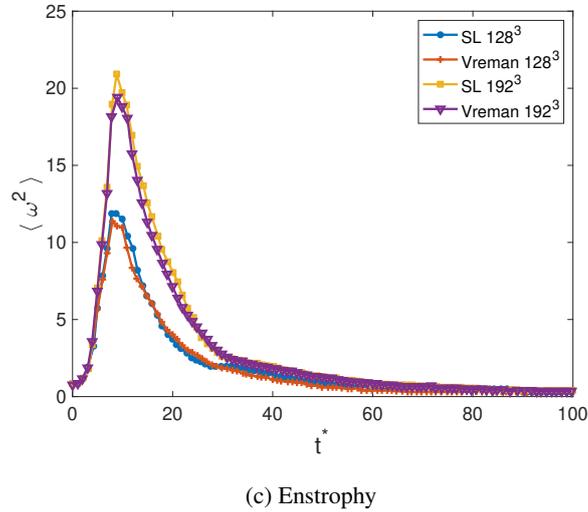
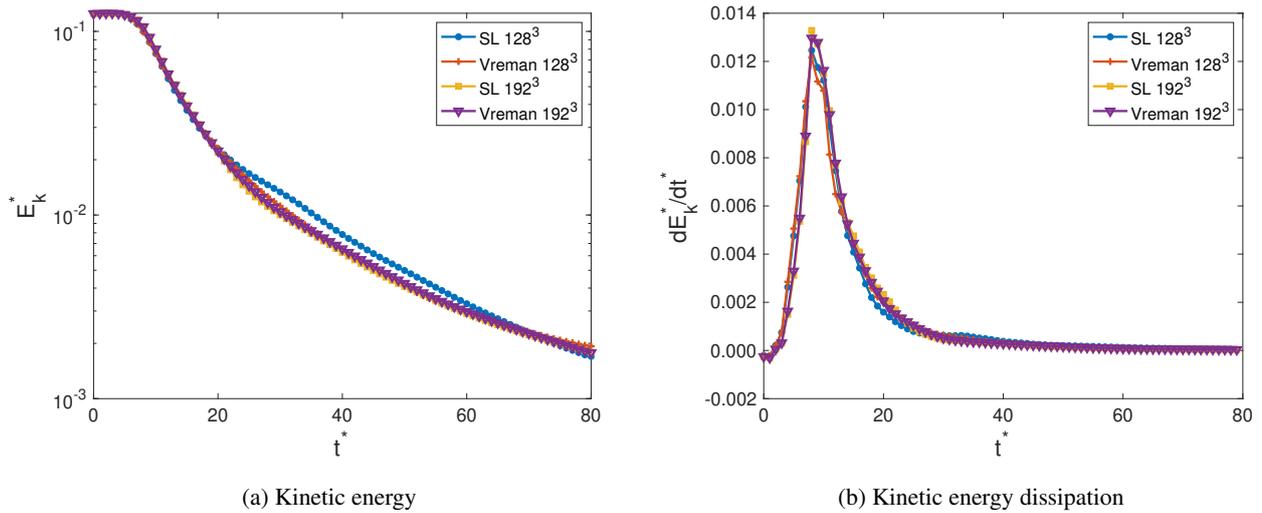



Figure 9. Evolution of volumetrically averaged quantities in the numerical solution to the Taylor-Green vortex problem, computed on 128^3 and 192^3 grids with the Vreman and SL SGS models.

is applied (see Appendix A1). The effective horizontal and vertical resolutions are, respectively, $\Delta x = \Delta y = 100$ m and $\Delta z = 40$ m. The simulation time is 6 hours.

Figure 11 shows the vertical profiles of the domain-mean thermodynamic and turbulence properties over the last hour of the simulations. Figure 12 shows the time series of liquid water paths (LWP), cloud cover, and turbulence kinetic energy. The time averaged results of the vertical profiles of θ_l, q_l, cloud fraction, and $q_v = q_t - q_l$ during the last hour are in good agreement with the same quantities presented by Siebesma et al. (2003). The SGS model does not have much effect on the simulation

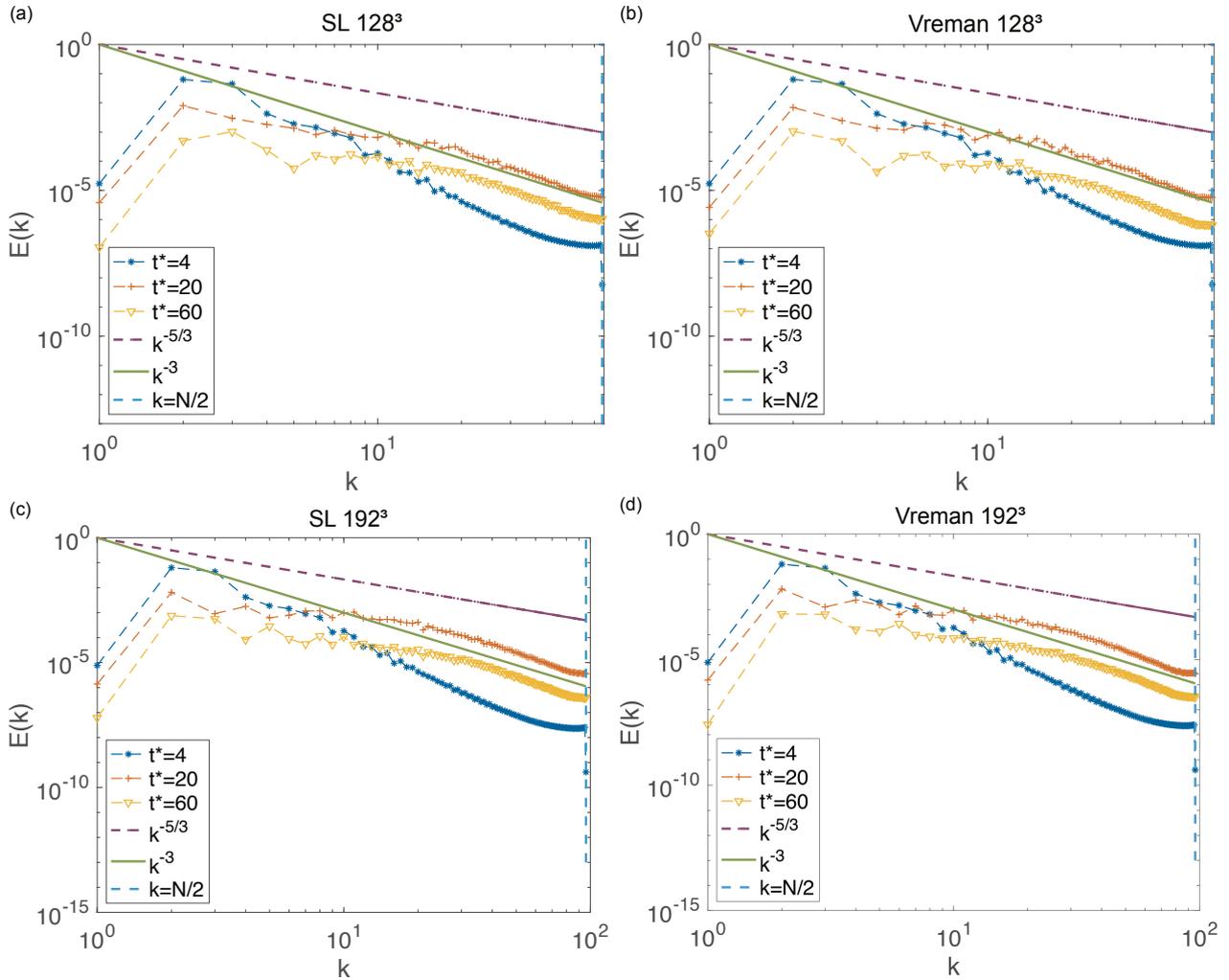



Figure 10. Kinetic energy spectra obtained using (a) 128^3 points and SL, (b) 128^3 points and Vreman, (c) 192^3 points and SL, and (d) 192^3 points and Vreman.

characteristics, except that the Vreman SGS model produces a stronger peak in the variance of vertical velocity, $\overline{w'w'}$, near the cloud top. Although the difference is mild, it is possibly due to the low dissipation nature of the Vreman's model. Excluding the first hour of flow spin up, the results compare well with PyCLES (Pressel et al., 2015) and fall within the ensemble range shown in Figure 2 of Siebesma et al. (2003). PyCLES solves the anelastic conservation equations with entropy as the prognostic
560   thermodynamic variable, using weighted essentially non-oscillatory (WENO) numerics. Differences in the initial conditions (as seen in the initial LWP in Figure 12) are likely contributing to the initial mismatch in horizontally averaged properties of moisture and turbulence. Details on the computation of horizontally-averaged profiles can be found in Appendix B.

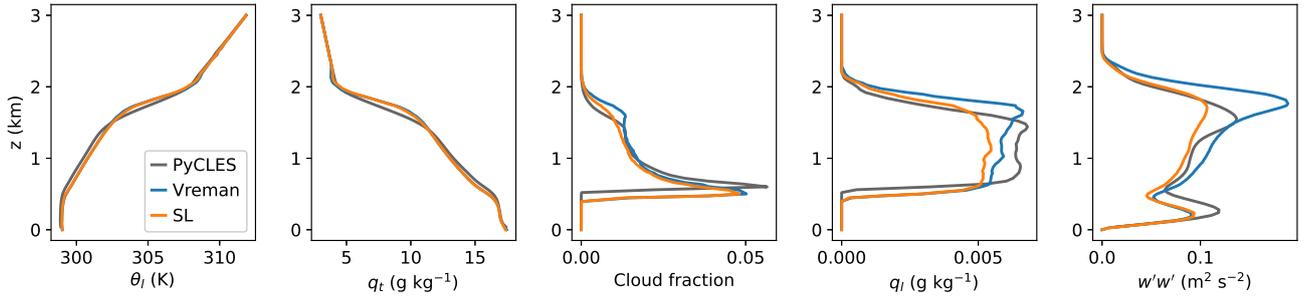



Figure 11. BOMEX. Profile of the mean state of liquid potential temperature, total specific humidity, cloud fraction, liquid water specific humidity, and variance of the vertical velocity fluctuations averaged along the last hour of the simulation. The solutions with PyCLES and ClimateMachine were calculated with effective grid resolution $\Delta x = \Delta y = 100$ m and $\Delta z = 40$ m. Results calculated with Vreman (blue lines) and SL SGS (orange lines) models in ClimateMachine are compared against results of PyCLES (grey lines) in its Paired-SGS modality using the SL model.

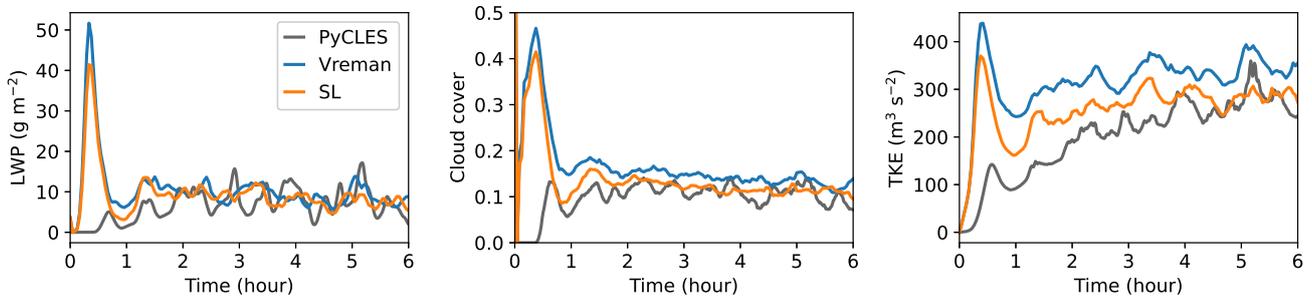



Figure 12. BOMEX. From left to right, time series of horizontally averaged LWP, cloud cover, and turbulence kinetic energy diagnosed from ClimateMachine using Vreman and SL. These results are consistent with ensemble results presented in Figure 2 of the intercomparison study by Siebesma et al. (2003). Line colors as in Figure 11.

A large domain simulation of BOMEX with effective horizontal resolution $\Delta x = \Delta y = 50$ m and vertical resolution $\Delta z = 20$ m in a $30 \times 30 \times 3.6$ km^3 domain was executed using 16 GPUs on the Google Cloud Platform. The simulation was executed using 1D IMEX time integration (1D implicit in the vertical direction and 2D explicit in the horizontal direction) at maximum horizontal advective Courant number C = 0.9. A visualization of instantaneous shallow cumulus structures is shown in Figure 13.

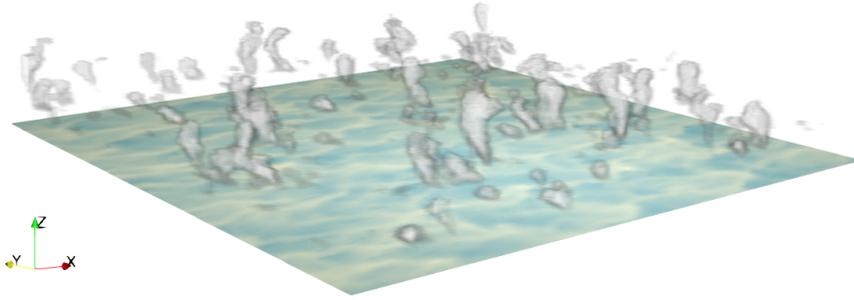



Figure 13. BOMEX. Instantaneous visualization of the shallow cumulus structures on a $30\,\mathrm{km}\times30\,\mathrm{km}\times3.6\,\mathrm{km}$ domain. The white shading of the volume rendering of the cloud corresponds to a maximum value of $q_l = 2.5\times10^{-4}\,\mathrm{kg\,kg^{-1}}$; a maximum $q_t = 1.8\times10^{-2}\,\mathrm{kg\,kg^{-1}}$ is shown on the bottom surface in light yellow shading. The simulation was executed using 16 GPUs on the Google Cloud Platform.

6 Verification of conservation property of prognostic variables

We define the time dependent normalized total mass and energy changes as, respectively,

$$\Delta M(t) = \frac{\int_{\Omega}\left[\rho(t) - \rho(t_0)\right]d\Omega}{\int_{\Omega}\rho(t_0)d\Omega}, \tag{66}$$

and

$$\Delta\rho e^{tot}(t) = \frac{\int_{\Omega}\left[\rho e^{tot}(t) - \rho e^{tot}(t_0)\right]d\Omega}{\int_{\Omega}\rho e^{tot}(t_0)d\Omega}, \tag{67}$$

where t_0 indicates the initial time and Ω is the full domain. Figure 14 shows $\Delta M(t)$ and $\Delta\rho e^{tot}(t)$ for a 1 hour simulation of a moist rising thermal bubble. The SL eddy viscosity model was used to represent under-resolved diffusive fluxes. This simulation was run using free-slip boundary conditions with adiabatic walls. We note that the loss of energy and mass in the system is contained to $\mathcal{O}(10^{-15})$, that is, numerical roundoff-error. This result highlights a key benefit of the general formulation of the prognostic conservation equations in flux form, which guarantees conservation properties up to source or sink contributions.

7 CPU strong-scaling

Demonstration of favorable scaling capabilities across multiple hardware types is critical to the utility of ClimateMachine as a competitive tool for large-eddy simulations. Toward this, we first examine strong scaling on CPU architectures. The rising thermal bubble problem described in Section 5.2 is used as the test problem, with its domain extents modified to form an $8.192\,\mathrm{km}^3$ cube, with an effective nodal resolution of 32 m to ensure that CPU memory on a single rank is maximally loaded. For tests with multiple MPI-ranks, each rank resides on a unique node, to ensure communication overhead is appropriately represented; in practice, one would expect to use multiple ranks per node. Scaling across multiple threads is not assessed

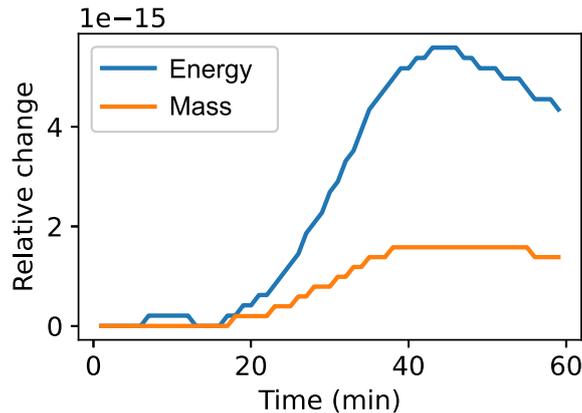



Figure 14. Time evolution of the mass and total energy loss (relative change when compared against initial conditions) for a moist thermal bubble simulation. Blue line: relative change in energy; Orange line: relative change in mass.

in the present work. Figure 15 shows the speedup in time-to-solution for 10 time-integration steps of the test problem with $N_{\text{ranks}} \in \{1, 2, 4, 8, 16, 32\}$ for both dry and moist simulations. We exclude checkpoint, diagnostic and periodic run-time output steps from time-to-solution measurements. In both dry and moist simulations, we see a speedup of approximately 19.7 when using 32 ranks compared with the corresponding single-rank simulation.

590   A single rank GPU run of the test problem on a $6.144\,\text{km}^3$ domain with 32 m effective resolution (restricted by GPU memory capacity) has a wall-clock time for ten integration steps of 314 s. The wall-clock time for a 32-rank CPU run was 449 s for a 2.37 times larger problem.

This provides an estimate for a comparison between CPU and GPU hardware performance. However, the balance between memory bandwidth limits and compute operation limits guides the maximum scaling possible on the GPU hardware relative

595   to its CPU counterpart, so this cannot be interpreted as a direct comparison across hardware types. Based on the present results, we conclude that it is more feasible to pursue strong-scaling improvements on CPU hardware than on GPU hardware. Further optimization and exploration of scaling in ClimateMachine is ongoing work. Additional details on the hardware used for scaling tests can be found in Appendix C.

8   GPU weak-scaling

600   To test the multi-GPU scalability of ClimateMachine, we first execute a BOMEX setup that is sufficiently large to saturate one GPU. The single-GPU execution represents the baseline from which we calculate the average time per time-step denoted by t_1. We then expand the domain size to match an increase in the number of GPUs and measure the average time per time step. Our scaling is then obtained as the ratio t_1/t_n, with t_n being the average time per time-step obtained with n GPUs. The results are obtained using up to 16 NVIDIA Tesla V100 GPUs running Julia version 1.4.2, CUDA 10.0, and CUDA-aware

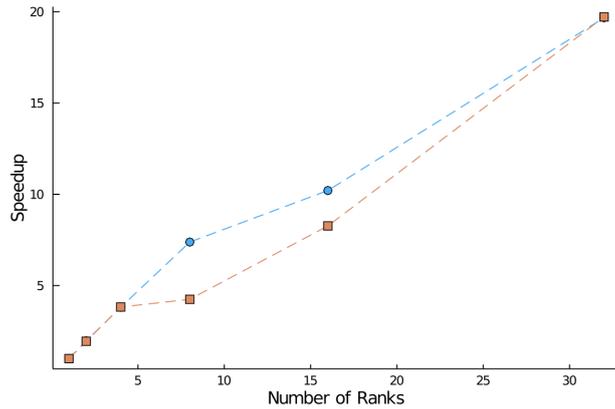



Figure 15. Speedup of time-integration (solver) step relative to time to solution for single-rank simulation of the rising thermal bubble problem in an 8.192 km^3 domain with uniform effective resolution of 32 m (with fourth-order polynomials). Blue circles: dry simulation; Orange squares: moist simulation.

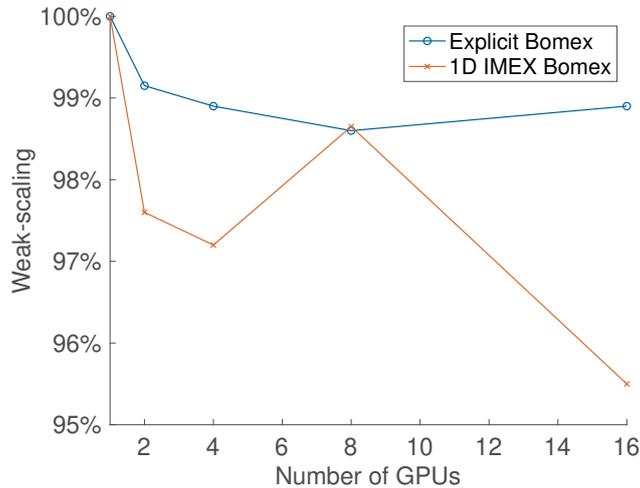



Figure 16. BOMEX weak scaling using 1D IMEX versus fully explicit time integration.

605   OpenMPI 4.0.3. Figure 16 shows excellent weak scaling for up to 16 GPUs on Google Cloud Platform resources. Over 95% weak scaling was achieved with 1D-IMEX time integration, and over 98% for the simulation with explicit timestepping. This is an encouraging result and supports the ability to prototype smaller problem setups and deploy larger simulations in the ClimateMachine limited area configuration at identical resolutions without significantly compromising the time to solution.

9 Conclusions

This paper introduced and assessed the LES configuration of ClimateMachine, a new Julia language simulation framework designed for parallel CPU and GPU architectures. Notable features of this LES framework are:

- Conservative flux form model equations for mass, momentum, total energy and total moisture to ensure global conservation of dynamical variables of interest (up to non-conservative source or sink processes)

- Discontinuous-Galerkin discretization with element-wise evaluation of the approximations to volume and interface integrals resulting in reduced time-to-solution due to MPI operations

- Application of model equations to the solution of benchmark problems in typical LES codes, including atmospheric flows in the shallow cumulus regime (BOMEX)

- Demonstration of strong-scaling on CPUs with up to 32 MPI-ranks (speed-up of 19.7 in time-to-solution), and weak scaling up to 16 GPUs (95-98%), in both dry and moist simulation configurations.

Code availability. ClimateMachine is an open source framework, and maintained on Github: https://github.com/CliMA/ClimateMachine.jl. Documentation for installing and running ClimateMachine is available at https://clima.github.io/ClimateMachine.jl/latest/. The version used in this paper is v0.2.0, which can be downloaded from https://doi.org/10.5281/zenodo.5542395, or https://github.com/CliMA/ClimateMachine.jl/releases/tag/v0.2.0. The parameters included in Table 1 are available in the accompanying CLIMAParameters.jl Julia package. The code is released under the Apache License, Version 2.0

Data availability. The test cases presented in the paper are available in the following directory locations in the source code:

- Section 5.1: /test/Numerics/DGMethods/Euler/isentropicvortex.jl

- Section 5.2: /tutorials/Atmos/risingbubble.jl

- Section 5.3: /tutorials/Atmos/densitycurrent.jl

- Section 5.4: /experiments/AtmosLES/schar_scalar_advection.jl

- Section 5.5: /experiments/AtmosLES/agnesi_hs_lin.jl and /experiments/AtmosLES/agnesi_nh_lin.jl

- Section 5.6: /experiments/AtmosLES/taylor_green.jl

- Section 5.7: /experiments/AtmosLES/bomex_model.jl and /experiments/AtmosLES/bomex_les.jl

Configuration instructions and methods for resolution, numerical flux schemes, boundary conditions and turbulence model specification can be found in the code documentation at https://clima.github.io/ClimateMachine.jl/v0.2/. Instructions for installing and launching simulations can be accessed at https://clima.github.io/ClimateMachine.jl/v0.2/GettingStarted/RunningClimateMachine/.

Appendix A: Boundary Conditions

A1 Solid walls and wall fluxes

Momentum Rigid surfaces are considered impenetrable such that the wall-normal component of the velocity vanishes at rigid boundaries by imposing $\boldsymbol{u} \cdot \boldsymbol{n} = 0$. The viscous sublayer is not explicitly resolved, and a momentum sink is applied to model the effect of wall-shear stresses. While the wall-normal advective momentum flux vanishes, the wall-normal viscous or SGS momentum flux, also known as the bulk surface stress (units of Pa),

$$\boldsymbol{n} \cdot (\rho \boldsymbol{\tau}) = -\boldsymbol{n} \cdot [2 \rho \boldsymbol{\nu}_t \boldsymbol{S}(\boldsymbol{u}_p)], \tag{A1}$$

is not necessarily negligible. Here, $\boldsymbol{S}(\boldsymbol{u}_p) = (\nabla \boldsymbol{u}_p + \nabla \boldsymbol{u}_p^T)/2$ is the strain rate tensor of the near-surface wall-parallel velocity, $\boldsymbol{u_p}$. We note that, throughout appendix A, $\boldsymbol{n}$ refers to the inward-pointing normal vector at domain boundaries, distinct from the prior definition of the element-interface normal vector in Section 3.

In the case of free-slip conditions at a solid surface (indicated by subscript "sfc"), there is no viscous or SGS momentum transfer between the atmosphere and the surface, such that

$$\boldsymbol{n} \cdot (\rho \boldsymbol{\tau}) \big|_{\text{sfc}} = 0.$$

Because the momentum flux tensor depends linearly on velocity derivatives, this amounts to homogeneous Neumann boundary conditions on velocity components parallel to the surface. On the other hand, viscous drag is imposed by the classical aerodynamic drag law

$$\boldsymbol{n} \cdot (\rho \boldsymbol{\tau}) = -\rho C_{d,\text{int}} \| \boldsymbol{u}_{p,\text{int}} \| \boldsymbol{u}_{p,\text{int}}, \tag{A2}$$

where the quantities with subscript int are evaluated at an interior point $\boldsymbol{x}_{\text{int}}$ adjacent to the surface. The drag coefficient

$$C_{d,\text{int}} = C_d(\boldsymbol{Y}; \boldsymbol{x}_{\text{int}})$$

can depend parameterically on state variables $\boldsymbol{Y}(\boldsymbol{x}, t)$ and on the position $\boldsymbol{x}_{\text{int}}$ of the interior point relative to the surface. In the present implementation, the plane of interior points relevant to boundary flux evaluation is interpreted as the first layer of interior nodes in the surface-adjacent elements. The drag law boundary condition amounts to inhomogeneous Neumann boundary conditions on velocity components parallel to the surface.

Specific humidity As for momentum, the advective specific humidity fluxes normal to a rigid surface vanish, but the diffusive or SGS specific humidity fluxes normal to the surface may not vanish. Normal components of SGS fluxes of condensate, q_l, are generally set to zero at boundaries

$$\boldsymbol{n} \cdot (\rho \boldsymbol{d}_{q_l}) \big|_{\text{sfc}} = 0, \tag{A3}$$

implying homogeneous Neumann boundary conditions ($\boldsymbol{n} \cdot \nabla q_l = 0$) on the condensate specific humidities. The total SGS specific humidity flux then reduces to the vapor flux at the surface. SGS turbulent deposition of condensate (fog) on the surface

665 can in principle occur; representing this would require nonzero condensate fluxes at the surface. With the assumption of zero condensate boundary fluxes, we have

$$\boldsymbol{n} \cdot (\rho \boldsymbol{d}_{q_t})\big|_{\text{sfc}} = \boldsymbol{n} \cdot (\rho \boldsymbol{d}_{q_v})\big|_{\text{sfc}},$$

where evaporation (measured in $\text{kg m}^{-2}\,\text{s}^{-1}$), or condensation if negative, is given by

$$E = \boldsymbol{n} \cdot (\rho \boldsymbol{d}_{q_v})\big|_{\text{sfc}} = -\boldsymbol{n} \cdot (\rho \boldsymbol{\mathcal{D}}_t \boldsymbol{\nabla} q_t)\big|_{\text{sfc}}.$$

670 Evaporation can be zero (water impermeable) or it can be given as a function of $\boldsymbol{Y}$ at the surface according to

$$E = \boldsymbol{n} \cdot (\rho \boldsymbol{d}_{q_v})\big|_{\text{sfc}} = E(\boldsymbol{Y}; \boldsymbol{x}_{\text{sfc}}, t),$$

which, numerically, translates into an inhomogeneous Neumann boundary condition on the vapor specific humidity.

Energy As for momentum and humidity, the advective energy fluxes normal to a rigid surface vanish, but the diffusive or SGS flux of total enthalpy, h^{tot}, normal to the surface (units of W m^{-2})

675 $$\boldsymbol{n} \cdot \rho(\boldsymbol{J} + \boldsymbol{D}) = -\boldsymbol{n} \cdot (\rho \boldsymbol{\mathcal{D}_t} \boldsymbol{\nabla} h^{\text{tot}}),$$

may not vanish. Because the kinetic energy contribution to the total enthalpy flux near a surface is usually 3–4 orders of magnitude smaller than the thermal and potential energy components, it is generally neglected, so that the total enthalpy flux $\boldsymbol{J} + \boldsymbol{D}$ reduces to a flux of moist static energy $\text{MSE} = h + \Phi$. The surface can be insulating, in which case the SGS transfer of total enthalpy between the atmosphere and the surface is zero:

680 $$\boldsymbol{n} \cdot \rho(\boldsymbol{J} + \boldsymbol{D})_{\text{sfc}} = 0,$$

which, from a numerical point of view, translates to a homogeneous Neumann condition on the total enthalpy ($\boldsymbol{n} \cdot \boldsymbol{\nabla} h^{\text{tot}} = 0$) or, by neglecting kinetic energy, on MSE such that ($\boldsymbol{n} \cdot \boldsymbol{\nabla} \text{MSE} = 0$). If MSE is a known function, the total energy flux is given by

$$\boldsymbol{n} \cdot \rho(\boldsymbol{J} + \boldsymbol{D})_{\text{sfc}} = \text{MSE}(\boldsymbol{Y}; \boldsymbol{x}_{\text{sfc}}, t).$$

685 The value of $\rho \boldsymbol{n} \cdot (\boldsymbol{J} + \boldsymbol{D})$ can also be assigned by the summation of given LHF and SHF, as done in the case of BOMEX described in Section 5.

A2 Non-reflecting top boundary

To prevent the reflection of fast, upward propagating gravity waves at the top boundary, a Rayleigh-damping sponge is added to the right-hand side of the momentum equation (see Section 5). The damping in the momentum equations takes the form:

690 $$\frac{\partial \rho \boldsymbol{u}}{\partial t} = \cdots - \tau_s \rho(\boldsymbol{u} - \boldsymbol{u}_{\text{relax}}) \tag{A4}$$

where $\boldsymbol{u}_{\mathrm{relax}}$ is a specified background velocity to which the flow is relaxed within the absorbing layer, with a characteristic relaxation coefficient τ_s. Of the many alternative options known for τ_s (e.g., Durran and Klemp (1983)), the default in ClimateMachine is

$$\tau_s = \alpha \sin^\gamma \left(\frac{1}{2} \frac{z - z_{\mathrm{s}}}{z_{\mathrm{top}} - z_{\mathrm{s}}} \right) \qquad \mathrm{for}\, z > z_{\mathrm{s}} \tag{A5}$$

where the absorbing sponge layer starts at $z = z_{\mathrm{s}}$, γ is a positive even power, typically set to 2, and $\alpha > 0$ is a relaxation coefficient, typically of order $\mathcal{O}(1\,\mathrm{s}^{-1})$.

A3   Numerical implementation

For the boundary velocity corresponding to the impenetrable wall condition, we use the following reflecting condition

$$\boldsymbol{u}_{bc} = \boldsymbol{u}^- - (\boldsymbol{n} \cdot \boldsymbol{u}^- \boldsymbol{n}), \tag{A6}$$

$$\boldsymbol{u}^+ = 2\boldsymbol{u}_{bc} - \boldsymbol{u}^-. \tag{A7}$$

The no-slip condition follows from (A6) by setting all components of $\boldsymbol{u}_{bc} = 0$. Boundary conditions on a scalar χ are similarly specified as follows

$$\chi^+ = 2\chi_{bc} - \chi^-. \tag{A8}$$

Non-zero mass-flux boundary conditions can be imposed at penetrable or free surfaces by applying the transmissive boundary condition

$$\boldsymbol{u}^+ = \boldsymbol{u}^-. \tag{A9}$$

Diffusive fluxes are applied by a direct specification of the wall-normal fluxes, and over-specified boundary conditions are avoided by using only the interior $(^-)$ gradients and first-order fluxes.

Appendix B: Statistics

Since the flow is compressible, we use density-weighed Favre averages following Canuto (1997) when computing horizontally-averaged statistics. For a scalar ϕ, the density-weighed average $\bar{\phi}$ at a given height-level z is defined by

$$\bar{\phi} = \frac{\langle \rho\phi \rangle}{\langle \rho \rangle}, \tag{B1}$$

where $\langle \cdot \rangle$ denotes a horizontal mean. All calculations of horizontal statistics are done on the DG nodal mesh to avoid introducing interpolation errors (Yamaguchi, 2012), with metric terms accounted for in the descriptions of diagnostic variables. The density-weighted vertical eddy flux for a variable ϕ is defined by

$$\overline{w'\phi'} = \frac{\langle \rho\, w'\phi' \rangle}{\langle \rho \rangle}, \tag{B2}$$

where

$$\phi' = \phi - \bar{\phi} \tag{B3}$$

denotes the deviation from the density-weighted mean. The variance can then be defined as

$$\overline{\phi'^2} = \frac{\langle \rho \, \phi'^2 \rangle}{\langle \rho \rangle}. \tag{B4}$$

Appendix C: Hardware

This section summarises the hardware characteristics for the primary compute resources used in tests throughout this paper. This is particularly relevant to the data presented in Section 7. Compute nodes for the CPU tests were 14-core Intel Xeon (2.4 GHz), with a maximum memory capacity of 1.5 TB. GPU nodes on this cluster were of 14-core Intel Broadwell (2.4 GHz) type with 28 cores per node and 256GB memory per node. Compute nodes on the Google Cloud Platform leverage Tesla V100 GPUs available for general-purpose use.

Appendix D: Supplementary Results

This section provides additional information on the comparison of the density current benchmark in ClimateMachine with existing literature references in Table A1.

Table A1. Summary of frontal locations for the density current test case from existing literature. Results tabulated are of the front location at $t = 900$ s. The results are reported for the following models: Climate Machine with SL and Vreman, Finite Element Model (FEM) Variational Multiscale Stabilization (VMS), f-wave, filtered Spectral Elements (SE), filtered Discontinuous Galerkin (DG), and the Piecewise Parabolic Method (PPM). NUMA is the Nonhydrostatic Unified Model of the Atmosphere.

Model	Space discr.	Resolution	Order	$\mu = 75 \, \mathrm{m^2 \, s^{-1}}$	Front Location [m]
ClimateMachine, SL	DG	12.5 m	4th	No	15090
"	"	25 m	4th	No	14990
"	"	50 m	4th	No	14770
"	"	100 m	4th	No	14669
ClimateMachine, Vreman	"	12.5 m	4th	No	15091
"	"	25 m	4th	No	14950
"	"	50 m	4th	No	14739
"	"	100 m	4th	No	14606
Giraldo-Restelli (Giraldo and Restelli, 2008)	DG	50 m	4th	Yes	14767
Giraldo-Restelli (Giraldo and Restelli, 2008)	SEM	50 m	4th	Yes	14767
NUMA Dyn-SGS (Marras et al., 2015)	SEM	12.5 m	4th	No	15056
"	"	25 m	4th	No	14992
"	"	50 m	4th	No	14535
"	"	100 m	4th	No	14325
NUMA SL (Marras et al., 2015)	SEM	25 m	4th	No	14918
"	"	50 m	4th	No	14726
"	"	100 m	4th	No	14551
VMS (Marras et al., 2013)	FEM	25 m	1st	No	14890
"	"	50 m	1st	No	14629
"	"	75 m	1st	No	14487
"	"	100 m	1st	No	14355
f-wave (Ahmad and Lindeman, 2007)	FV	50 m	2nd	Yes	14975
PPM (Straka et al., 1993)	FD	50 m		Yes	15027

Author contributions. **Akshay Sridhar**: analysis; methodology; software; writing–review and editing. **Yassine Tissaoui**: analysis; visualization; software; writing–review and editing. **Simone Marras**: conceptualization; methodology; software; writing–original draft preparation, review and editing. **Zhaoyi Shen**: software; analysis; visualization; writing–review and editing. **Charlie Kawczynski**: software. **Simon Byrne**: software. **Kiran Pamnany**: software; analysis. **Maciej Waruszewski**: methodology; software. **Thomas H. Gibson**: methodology; software; writing–review and editing. **Jeremy E. Kozdon**: conceptualization; methodology; software. **Valentin Churavy**: software. **Lucas C. Wilcox**: conceptualization; methodology; software. **Francis X. Giraldo**: conceptualization; methodology; software; writing–review and editing. **Tapio Schneider**: conceptualization; methodology; software; project administration; writing–original draft preparation, review and editing.

Competing interests. Simone Marras is a member of the editorial board of Geoscientific Model Development. The peer-review process was guided by an independent editor. The authors have no other competing interests to declare.

Acknowledgements. This research was made possible by the generosity of Eric and Wendy Schmidt by recommendation of the Schmidt Futures program, by the Paul G. Allen Family Foundation, Charles Trimble, Audi Environmental Foundation, Heising-Simons Foundation, and the National Science Foundation (grants AGS-1835860 and AGS-1835881). Additionally, V.C. was supported by the Defense Advanced Research Projects Agency (DARPA, agreement HR0011-20-9-0016) and by NSF (grant OAC-1835443). The computations presented here were conducted on the Resnick High Performance Computing Center, a facility supported by Resnick Sustainability Institute at the California Institute of Technology (formerly known as the Central HPC Cluster, with partial support by a grant from the Gordon and Betty Moore Foundation), and on the Google Cloud Platform, with in-kind support by Google. We thank the Google team for their assistance with operations on the Google Cloud Platform. Part of this research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

750 Abdi, D. S., Giraldo, F. X., Constantinescu, E., Lester III, C., Wilcox, L., and Warburton, T.: Acceleration of the Implicit-Explicit Non-Hydrostatic Unified Model of the Atmosphere (NUMA) on Manycore Processors, International Journal of High Performance Computing Applications, 33, 242–267, https://doi.org/10.1177/1094342017732395, 2017a.

Abdi, D. S., Wilcox, L. C., Warburton, T. C., and Giraldo, F. X.: A GPU-accelerated continuous and discontinuous Galerkin non-hydrostatic atmospheric model, International Journal of High Performance Computing Applications, 33, 81–109,
755 https://doi.org/10.1177/1094342017694427, 2017b.

Ahmad, N. and Lindeman, J.: Euler solutions using flux-based wave decomposition, International Journal of Numerical Methods in Fluids, 54, 47–72, https://doi.org/10.1002/fld.1392, 2007.

Balaji, V.: Climbing down Charney's ladder: machine learning and the post-Dennard era of computational climate science, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 379, 20200 085,
760 https://doi.org/10.1098/rsta.2020.0085, 2021.

Bao, L., Klöfkorn, R., and Nair, R. D.: Horizontally Explicit and Vertically Implicit (HEVI) Time Discretization Scheme for a Discontinuous Galerkin Nonhydrostatic Model, Monthly Weather Review, 143, 972–990, https://doi.org/https://doi.org/10.1175/MWR-D-14-00083.1, 2015.

Bassi, F. and Rebay, S.: A high-order discontinuous Galerkin finite element method solution of the 2d Euler equations, Journal of Computa-
765 tional Physics, 138, 251–285, https://doi.org/https://doi.org/10.1006/jcph.1997.5454, 1997.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B.: Julia: A Fresh Approach to Numerical Computing, SIAM Review, 59, 65–98, https://doi.org/10.1137/141000671, 2017.

Bezanson, J., Chen, J., Chung, B., Karpinski, S., Shah, V. B., Vitek, J., and Zoubritzky, L.: Julia: Dynamism and Performance Reconciled by Design, Proc. ACM Program. Lang., 2, https://doi.org/10.1145/3276490, 2018.

770 Bott, A.: Theoretical considerations on the mass and energy consistent treatment of precipitation in cloudy atmospheres, Atmos. Res., 89, 252–269, 2008.

Boyd, J. P.: The erfc-log filter and the asymptotics of the Euler and Vandeven sequence accelerations, A.V. Ilin, L.R. Scott (Eds.), Proceedings of the Third International Conference on Spectral and High Order Methods, Houston Journal of Mathematics, pp. 267–276, 1996.

Brachet, M. E.: Direct simulation of three-dimensional turbulence in the Taylor-Green vortex., Fluid Dyn. Res., 8, 1–8,
775 https://doi.org/10.1016/0169-5983(91)90026-f, 1991.

Brachet, M. E., Meiron, D. I., Orszag, A., Nickel, B. G., Morf, R. H., and Frisch, U.: Small-scale structure of the Taylor-Green vortex., Journal of Fluid Mechanics, 130, 411–452, https://doi.org/10.1017/S0022112083001159, 1983.

Canuto, V. M.: Compressible turbulence, Astrophys. J., 482, 827–851, https://doi.org/10.1086/304175, 1997.

Carpenter, M. H. and Kennedy, C. A.: Fourth-order 2N-storage Runge-Kutta schemes, Tech. Rep. NASA TM-109112, National Aeronautics
780 and Space Administration, Langley Research Center, Hampton, VA, 1994.

Chow, F. K. and Moin, P.: A further study of numerical errors in large-eddy simulations, Journal of Computational Physics, 184, 366–380, https://doi.org/https://doi.org/10.1016/S0021-9991(02)00020-7, 2003.

Deardorff, J. W.: A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers, Journal of Fluid Mechanics, 41, 452–480, https://doi.org/10.1017/S0022112070000691, 1970.

785 Deardorff, J. W.: Three-dimensional numerical study of the height and mean structure of a heated planetary boundary layer., Boundary Layer Meteorology, 7, 81–106, https://doi.org/10.1007/BF00224974, 1974.

Deardorff, J. W.: Usefulness of liquid-water potential temperature in a shallow-cloud model., Journal of Applied Meteorology, 15, 98–102, https://doi.org/10.1175/1520-0450(1976)015<0098:UOLWPT>2.0.CO;2, 1976.

Deardorff, J. W.: Stratocumulus-capped mixed layers derived from a three-dimensional model, Boundary Layer Meteorology, 18, 495–527,
790 https://doi.org/10.1007/BF00119502, 1980.

Deville, M. O., Fischer, P. F., and Mund, E. H.: High-order methods for incompressible fluid flow, Cambridge University Press, https://doi.org/10.1017/CBO9780511546792, 2002.

Dipankar, A., Stevens, B., Heinze, R., Moseley, C., Zängl, G., Giorgetta, M., and Brdar, S.: Large eddy simulation using the general circulation model ICON, J. Adv. Model. Earth Sys., 7, 963–986, https://doi.org/https://doi.org/10.1002/2015MS000431, 2015.

795 Drikakis, D., Fureby, C., and Youngs, F.: Simulation of transition and turbulence decay in the Taylor–Green vortex, Journal of Turbulence, 8, 1–12, https://doi.org/10.1080/14685240701250289, 2007.

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

E. F. Toro, E. F., Spruce, M., and Speares, W.: Restoration of the Contact Surface in the HLL–Riemann Solver, Shock Waves, 4, 25–34,
800 https://doi.org/10.1007/BF01414629, 1994.

Fuhrer, O., Osuna, C., Lapillonne, X., Gysi, T., Cumming, B., Bianco, M., Arteaga, A., and Schulthess, T. C.: Towards a performance portable, architecture agnostic implementation strategy for weather and climate models, Supercomputing Frontiers and Innovations, 1, 45–62, https://doi.org/10.14529/jsfi140103, 2014.

Fuhrer, O., Chadha, T., Hoefler, T., Kwasniewski, G., Lapillonne, X., Leutwyler, D., Lüthi, D., Osuna, C., Schär, C., Schulthess, T. C., and
805 Vogt, H.: Near-global climate simulation at 1 km resolution: establishing a performance baseline on 4888 GPUs with COSMO 5.0, Geosc. Model Dev., 11, 1665–1681, https://doi.org/10.5194/gmd-11-1665-2018, 2018.

Gal-Chen, T. and Somerville, R.: Numerical solution of the Navier-Stokes equations with topography, J. Comput. Phys., 17, 276–310, https://doi.org/10.1016/0021-9991(75)90054-6, 1975.

Ghosal, S.: An Analysis of Numerical Errors in Large-Eddy Simulations of Turbulence, Journal of Computational Physics, 125, 187–206,
810 https://doi.org/10.1006/jcph.1996.0088, 1996.

Giraldo, F. X.: An Introduction to Element-based Galerkin Methods on Tensor-Product Bases: Analysis, Algorithms, and Applications, Springer, https://doi.org/10.1007/978-3-030-55069-1, 2020.

Giraldo, F. X. and Restelli, M.: A study of spectral element and discontinuous Galerkin methods for the Navier-Stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases, Journal of Computational Physics, 227, 3849–3877,
815 https://doi.org/10.1016/j.jcp.2007.12.009, 2008.

Giraldo, F. X., Hesthaven, J. S., and Warburton, T.: Nodal high-order discontinuous Galerkin methods for spherical shallow water equations, J. Comput. Phys., 181, 499–525, https://doi.org/10.1006/jcph.2002.7139, 2002.

Giraldo, F. X., Kelly, J. F., and Constantinescu, E. M.: Implicit-explicit formulations of a three-dimensional nonhydrostatic unified model of the atmosphere (NUMA), SIAM Journal on Scientific Computing, 35, B1162–B1194, https://doi.org/10.1137/120876034, 2013.

820 Harten, A.: High resolution schemes for hyperbolic conservation laws, J. Comput. Phys., 49, 357–393, https://doi.org/10.1016/0021-9991(83)90136-5, 1983.

Hesthaven, J. and Warburton, T.: Nodal discontinuous Galerkin method. Algorithms, analysis and applications., Springer, https://doi.org/10.1007/978-0-387-72067-8, 2008a.

Hesthaven, J. S. and Warburton, T.: Nodal discontinuous Galerkin methods: algorithms, analysis, and applications, vol. 54, Springer-Verlag New York Inc, https://doi.org/10.1007/978-0-387-72067-8, 2008b.

Holland, J. Z. and Rasmusson, E. M.: Measurements of the atmospheric mass, energy, and momentum budgets over a 500-kilometer square of tropical ocean, Mon. Wea. Rev, 101, 44–57, https://doi.org/10.1175/1520-0493(1973)101<0044:MOTAME>2.3.CO;2, 1973.

Hunt, J. C. R., Wray, A., and Moin, P.: Eddies, stream, and convergence zones in turbulent flows, Tech. Rep. CTR-S88, Center for Turbulence Research Report CTR-S88, Stanford University, 1988.

Jähn, M., Knoth, O., König, M., and Vogelsberg, U.: ASAM v2.7: a compressible atmospheric model with a Cartesian cut cell approach, Geoscientific Model Development, 8, 317–340, https://doi.org/10.5194/gmd-8-317-2015, 2015.

Karniadakis, G. and Sherwin, S.: Spectral/hp element methods for CFD, Oxford University Press, https://doi.org/10.1093/acprof:oso/9780198528692.001.0001, 1999.

Kelly, J. F. and Giraldo, F. X.: Continuous and discontinuous Galerkin methods for a scalable three-dimensional nonhydrostatic atmospheric model: limited-area mode, J. Comput. Phys., 231, 7988–8008, https://doi.org/10.1016/j.jcp.2012.04.042, 2012.

Kennedy, C. A. and Carpenter, M. H.: Higher-order additive Runge–Kutta schemes for ordinary differential equations, Applied Numerical Mathematics, 136, 183–205, https://doi.org/10.1016/j.apnum.2018.10.007, 2019.

Kopriva, D. A.: Implementing spectral methods for partial differential equations: Algorithms for scientists and engineers, Springer Science & Business Media, https://doi.org/10.1007/978-90-481-2261-5, 2009.

Kurowski, M., Grabowski, W. W., and Smolarkiewicz, P. K.: Anelastic and compressible simulation of moist deep convection, Journal of Atmospheric Sciences, https://doi.org/10.1175/JAS-D-14–0017.1, 2014.

Light, D. and Durran, D.: Preserving Nonnegativity in Discontinuous Galerkin Approximations to Scalar Transport via Truncation and Mass Aware Rescaling (TMAR), Monthly Weather Review, 144, 4771–4786, https://doi.org/10.1175/MWR-D-16-0220.1, 2016.

Lilly, D. K.: On the numerical simulation of buoyant convection, Tellus, 14, 148–172, https://doi.org/10.3402/tellusa.v14i2.9537, 1962.

Lilly, D. K.: On the application of the eddy viscosity concept in the inertial sub-range of turbulence, NCAR manuscript, 123, https://doi.org/10.5065/D67H1GGQ, 1966.

Lin, W.-C. and McIntosh-Smith, S.: Comparing Julia to Performance Portable Parallel Programming Models for HPC, in: 2021 International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS), pp. 94–105, https://doi.org/10.1109/PMBS54543.2021.00016, 2021.

Marras, S. and Giraldo, F. X.: A parameter-free dynamic alternative to hyper-viscosity for coupled transport equations: application to the simulation of 3D squall lines using spectral elements, J. Comput. Phys., 283, 360–373, https://doi.org/10.1016/j.jcp.2014.11.046, 2015.

Marras, S., Kelly, J. F., Giraldo, F. X., and Vázquez, M.: Variational multiscale stabilization of high-order spectral elements for the advection-diffusion equation, J. Comput. Phys., 231, 7187–7213, 2012.

Marras, S., Moragues, M., Vázquez, M., Jorba, O., and Houzeaux, G.: A Variational Multiscale Stabilized Finite Element Method for the Solution of the Euler Equations of Nonhydrostatic Stratified Flows, J. Comput. Phys., 236, 380–407, 2013.

Marras, S., Kelly, J. F., Moragues, M., Müller, A., Kopera, M. A., Vázquez, M., Giraldo, F. X., Houzeaux, G., and Jorba, O.: A Variational Multiscale Stabilized Finite Element Method for the Solution of the Euler Equations of Nonhydrostatic Stratified Flows, Arch. Comput. Methods Eng., 23, 673–722, https://doi.org/https://doi.org/10.1016/j.jcp.2012.10.056, 2015.

Marras, S., Nazarov, M., and Giraldo, F. X.: Stabilized high-order Galerkin methods based on a parameter-free dynamic SGS model for LES, J. Comput. Phys., 301, 77–101, https://doi.org/10.1016/j.jcp.2015.07.034, 2015.

Mason, P. J. and Callen, N. S.: On the magnitude of the subgrid-scale eddy coefficient in large-eddy simulations of turbulent channel flow, Journal of Fluid Mechanics, 162, 439–462, https://doi.org/10.1017/S0022112086002112, 1986.

Matheou, G.: Numerical discretization and subgrid-scale model effects on large-eddy simulations of a stable boundary layer, Q. J. Roy. Meteor. Soc., 142, 3050–3062, 2016.

Matheou, G. and Teixeira, J.: Sensitivity to Physical and Numerical Aspects of Large-Eddy Simulation of Stratocumulus, Monthly Weather Review, 147, 2621–2639, 2019.

Matheou, G., Chung, D., Nuijens, L., Stevens, B., and Teixeira, J.: On the fidelity of large-eddy simulation of shallow precipitating cumulus convection, Mon. Weather. Rev., 139, 2918–2939, https://doi.org/10.1175/2011MWR3599.1, 2011.

Mellado, J.: Cloud-Top Entrainment in Stratocumulus Clouds, Annual Review of Fluid Mechanics, 49, 145–169, 2017.

Mellado, J. P., Bretherton, C. S., Stevens, B., and Wyant, M. C.: DNS and LES for Simulating Stratocumulus: Better Together, Journal of Advances in Modeling Earth Systems, 10, 1421–1438, https://doi.org/10.1029/2018MS001312, 2018.

Moeng, C., McWilliams, J., Rotunno, R., Sullivan, P., and Weil, J.: Investigating 2D modelling of atmospheric convection in the PBL, J. Atmos. Sci., 61, 889–903, 2003.

Moeng, C. H.: A Large-Eddy simulation model for the study of planetary boundary-layer turbulence, J. Atmos. Sci., 41, 2052–2062, 1984.

Moeng, C. H. and Wyngaard, J. C.: Spectral analysis of large-eddy simulations of the convective boundary layer., Journal of Atmospheric Science, 45, 3573–3587, 1988.

Müller, A., Kopera, M., Marras, S., Wilcox, L., Isaac, T., and Giraldo, F.: Strong scaling for numerical weather prediction at petascale with the atmospheric model NUMA, International Journal of High Performance Computing Applications, 33, 411–426, https://doi.org/10.1177/1094342018763966, 2018.

Niegemann, J., Diehl, R., and Busch, K.: Efficient low-storage Runge–Kutta schemes with optimized stability regions, Journal of Computational Physics, 231, 364–372, https://doi.org/10.1016/j.jcp.2011.09.003, 2012.

Palmer, T.: Climate forecasting: build high-resolution global climate models, Nature, 515, 338–339, https://doi.org/10.1038/515338a, 2014.

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

Pressel, K. G., Mishra, S., Schneider, T., Kaul, C. M., and Tan, Z.: Numerics and Subgrid-Scale Modeling in Large Eddy Simulations of Stratocumulus Clouds, Journal of Advances in Modeling Earth Systems, 9, 1342–1365, https://doi.org/10.1002/2016MS000778, 2017.

Raymond, D. J.: Sources and sinks of entropy in the atmosphere, Journal of Advances in Modeling Earth Systems, 5, 755–763, 2013.

Reddy, S., Tissaoui, Y., De Bragan ça Alves, F., Marras, S., and Giraldo, F.: Comparison of Sub-Grid Scale Models for Large-Eddy Simulation Using a High-Order Spectral Element Approximation of the Compressible Navier-Stokes Equations at Low Mach Number, Journal of Computational Physics (In review), https://doi.org/10.13140/RG.2.2.17576.90885, 2021.

Roe, P.: Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes, Journal of Computational Physics, 43, 357–372, https://doi.org/10.1016/0021-9991(81)90128-5, 1981.

Romps, D. M.: The dry-entropy budget of a moist atmosphere, J. Atmos. Sci., 65, 3779–3799, https://doi.org/10.1175/2008JAS2679.1, 2008.

Rusanov, V.: Calculation of Interaction of Non–Steady Shock Waves with obstacles, USSR Computational Mathematics and Mathematical Physics, 1, 267–279, https://doi.org/10.1016/0041-5553(62)90062-9, 1961.

Savic-Jovcic, V. and Stevens, B.: The structure and mesoscale organization of precipitating stratocumulus, Journal of Atmospheric Science, 65, 1587–1605, https://doi.org/10.1175/2007JAS2456.1, 2008.

Schalkwijk, J., Griffith, E., Post, H., and Jonker, H. J. J.: High performance simulations of turbulent clouds on a desktop PC: Exploiting the GPU, Bulletin of the American Meteorological Society, 93, 307–314, https://doi.org/10.1175/BAMS-D-11-00059.1, 2012.

900 Schalkwijk, J., Jonker, H., Siebesma, A., and Bosveld, F.: A year-long Large-Eddy Simulation of the weather over Cabauw: an overview, Monthly Weather Review, 143, 828–844, https://doi.org/10.1175/MWR-D-14-00293.1, 2015.

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

905 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.

Schneider, T., Kaul, C., and Pressel, K.: Possible climate transitions from breakup of stratocumulus decks under greenhouse warming, Nature 910 Geoscience, 12, https://doi.org/"10.1038/s41561-019-0310-1", 2019.

Shi, X., Chow, F. K., Street, R. L., and Bryan, G. H.: An Evaluation of LES Turbulence Models for Scalar Mixing in the Stratocumulus-Capped Boundary Layer, Journal of the Atmospheric Sciences, 75, 1499 – 1507, https://doi.org/10.1175/JAS-D-17-0392.1, 2018.

Shu, C.-W. and Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes, Journal of Computational Physics, 77, 439–471, https://doi.org/https://doi.org/10.1016/0021-9991(88)90177-5, 1988.

915 Siebesma, A. P., Bretherton, C. S., Brown, A., Chlond, A., Cuxart, J., Duynkerke, P. G., Jiang, H., Khairoutdinov, M., Lewellen, D., Moeng, C.-H., et al.: A large eddy simulation intercomparison study of shallow cumulus convection, Journal of the Atmospheric Sciences, 60, 1201–1219, https://doi.org/10.1175/1520-0469(2003)60<1201:ALESIS>2.0.CO;2, 2003.

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

920 Smith, R.: Linear theory of stratified hydrostatic flow past an isolated mountain, Tellus, 32, 348–364, https://doi.org/10.3402/tellusa.v32i4.10590, 1980.

Smith, R. B.: The influence of mountains on the atmosphere, Advances in Geophysics, 21, 87–230, https://doi.org/https://doi.org/10.1016/S0065-2687(08)60262-9, 1979.

Stevens, B., Lenschow, D. H., Vali, G., Gerber, H., Bandy, A., Blomquist, B., Brenguier, J.-L., Bretherton, C. S., Burnet, F., Campos, T., 925 Chai, S., Faloona, I., et al.: Dynamics and chemistry of marine stratocumulus–DYCOMS-II, Bull. Amer. Meteor. Soc., 84, 579–593, https://doi.org/"10.1175/BAMS-84-5-579",, 2003.

Stevens, B., Moeng, C.-H., Ackerman, A. S., Bretherton, C. S., Chlond, A., de Roode, S., Edwards, J., Golaz, J.-C., Jiang, H., Khairoutdinov, M., Kirkpatrick, M. O., Lewellen, D. C., Lock, A., Müller, F., Stevens, D. E., Whelan, E., and Zhu, P.: Evaluation of Large-Eddy Simulations via Observations of Nocturnal Marine Stratocumulus, Monthly Weather Review, 133, 1443–1462, 930 https://doi.org/10.1175/MWR2930.1, 2005.

Straka, J., Wilhelmson, R., Wicker, L., Anderson, J., and Droegemeier, K.: Numerical solution of a nonlinear density current: a benchmark solution and comparisons, International Journal for Numerical Methods in Fluids, 17, 1–22, https://doi.org/https://doi.org/10.1002/fld.1650170103, 1993.

Sullivan, P., McWilliams, J., and Moeng, C.: A subgrid-scale model for large-eddy simulation of planetary boundary-layer flows, Boundary-Layer Meteorology, 71, 247–276, https://doi.org/10.1007/BF00713741, 1994.

Tao, W.-K., Simpson, J., and McCumber, M.: An Ice-Water Saturation Adjustment, Monthly Weather Review, 117, 231–235, 1989.

Vandeven, H.: Family of spectral filters for discontinuous problems, Journal of Scientific Computing, 6, 159–192, https://doi.org/10.1007/BF01062118, 1991.

Vreman, A.: An eddy-viscosity subgrid-scale model for turbulent shear flow: algebraic theory and applications, Physics of Fluids, 16, 3670–3681, https://doi.org/10.1063/1.1785131, 2004.

Yamaguchi, T. Feingold, G.: Technical note: Large-eddy simulation of cloudy boundary layer with the Advanced Research WRF model, J. Adv. Model. Earth Systems, 4, M09 003, https://doi.org/10.1029/2012MS000164, 2012.

Yu, M. L., Giraldo, F. X., Peng, M., and Wang, Z. J.: Localized Artificial Viscosity Stabilization of Discontinuous Galerkin Methods for Nonhydrostatic Mesoscale Atmospheric Modeling, Monthly Weather Review, 143, 4823 – 4845, https://doi.org/10.1175/MWR-D-15-0134.1, 2015.