Large-eddy simulation in an anelastic framework with closed water and entropy balances

AGU PUBLICATIONS

Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1002/2015MS000496

Large-eddy simulation in an anelastic framework with closed water and entropy balances

Kyle G. Pressel[1], Colleen M. Kaul[1], Tapio Schneider[1,2], Zhihong Tan[1,2], and Siddhartha Mishra[3]

[1]Department of Earth Sciences, ETH Zürich, Zürich, Switzerland, [2]California Institute of Technology, Pasadena, California, USA, [3]Seminar for Applied Mathematics, ETH Zürich, Zürich, Switzerland

Key Points:
- An anelastic framework for LES with closed water and entropy balances is developed
- State-of-the-art numerical techniques yield superior results to traditional techniques
- Implementation in Python and Cython makes the framework easy to use

Correspondence to:
T. Schneider,
tapio@ethz.ch

Citation:
Pressel, K. G., C. M. Kaul, T. Schneider, Z. Tan, and S. Mishra (2015), Large-eddy simulation in an anelastic framework with closed water and entropy balances, *J. Adv. Model. Earth Syst.*, *7*, 1425–1456, doi:10.1002/2015MS000496.

Received 9 JUN 2015
Accepted 20 AUG 2015
Accepted article online 22 AUG 2015
Published online 26 SEP 2015

Abstract A large-eddy simulation (LES) framework is developed for simulating the dynamics of clouds and boundary layers with closed water and entropy balances. The framework is based on the anelastic equations in a formulation that remains accurate for deep convection. As prognostic variables, it uses total water and entropy, which are conserved in adiabatic and reversible processes, including reversible phase changes of water. This has numerical advantages for modeling clouds, in which reversible phase changes of water occur frequently. The equations of motion are discretized using higher-order weighted essentially nonoscillatory (WENO) discretization schemes with strong stability preserving time stepping. Numerical tests demonstrate that the WENO schemes yield simulations superior to centered schemes, even when the WENO schemes are used at coarser resolution. The framework is implemented in a new LES code written in Python and Cython, which makes the code transparent and easy to use for a wide user group.

1. Introduction

Large-eddy simulation (LES) has a long history in the study of three-dimensional atmospheric turbulence in boundary layers and in convective clouds [e.g., *Deardorff*, 1974a,b, 1980]. LES resolves the most energetic, larger-scale turbulent motions and represents smaller-scale motions through subgrid-scale (SGS) models. Numerous codes to do so are available [e.g., *Raasch and Schröter*, 2001; *Savic-Jovcic and Stevens*, 2008; *Heus et al.*, 2010; *Matheou et al.*, 2011; *Savre et al.*, 2014]. They have been used, for example, to study dry convective boundary layers [e.g., *Schmidt and Schumann*, 1989; *Sullivan and Patton*, 2011], stably stratified boundary layers [e.g., *Beare et al.*, 2006], shallow cumulus convection [e.g., *Jiang and Cotton*, 2000; *Siebesma et al.*, 2003], and stratocumulus topped boundary layers [e.g., *Stevens et al.*, 2005]. Increasing computing power has recently made it possible to conduct LES of deep moist convection [e.g., *Khairoutdinov and Randall*, 2006; *Khairoutdinov et al.*, 2009], and to conduct LES for sufficiently long times to reach statistically steady states [e.g., *Chung et al.*, 2012; *Blossey et al.*, 2013; *Bretherton et al.*, 2013]. At the same time, it has allowed LES domains to increase in size from 10^4 grid points in the 1970s to $\gtrsim 10^9$ grid points today.

It is clear that LES has the potential to help resolve some of the most important and pressing problems in climate dynamics. In particular, questions about the climate feedback of low clouds continue to be the largest source of uncertainty in climate projections [e.g., *Cess et al.*, 1990, 1996; *Bony and Dufresne*, 2005; *Webb et al.*, 2006; *Dufresne and Bony*, 2008; *Vial et al.*, 2013]. To facilitate addressing such questions, here we present a new LES framework, which exploits recent developments in atmosphere dynamics and in numerical methods and aims to make LES both more reliable and more accessible to a wider user group.

The equations of motion on which the LES framework is based are a variant of the anelastic equations of atmospheric motion, suitable for a range of applications from boundary layer dynamics to deep moist convection [*Batchelor*, 1953; *Ogura and Phillips*, 1962; *Dutton and Fichtl*, 1969; *Lipps and Hemler*, 1982; *Bannon*, 1996; *Pauluis*, 2008]. The anelastic equations eliminate sound waves from the equations of motion, which are generally not important for atmospheric dynamics as the Mach number is small. Eliminating sound waves is desirable to reduce the computational burden that otherwise arises in explicit time integration of the fully compressible equations of motion, in which resolving sound waves requires short time steps [e.g., *Satoh et al.*, 2008; *Duarte et al.*, 2014]. The anelastic equations eliminate sound waves by permitting only small deviations of the density from a reference state, which can depend on height and traditionally has been taken to be hydrostatic with constant or at most with a weakly varying dry entropy [*Ogura and Phillips*, 1962; *Lipps and Hemler*, 1982; *Bannon*, 1996]. A problem of traditional formulations of the anelastic

equations is that they generally do not conserve energy when the reference state entropy varies substantially with height [*Nance and Durran*, 1994]. They become inaccurate, for example, for deep convection, when deviations from a dry-isentropic reference state can be substantial [*Bannon*, 1996]. *Pauluis* [2008] proposed an anelastic approximation that is thermodynamically consistent and remains accurate for deep convection. It uses a reference state that is hydrostatic with constant moist entropy and is most naturally formulated with moist entropy as a prognostic variable. The LES framework we develop here is based on this variant of the anelastic approximation and uses total water and moist entropy as prognostic variables.

Using total water and moist entropy as prognostic variables has numerous numerical advantages because these quantities are conserved in adiabatic and reversible processes. For example, reversible phase changes of water are common in clouds and can occur on time scales of seconds—within a computational time step of typical LES. Because total water and moist entropy are conserved in such phase changes, using them as prognostic variables can allow larger computational time steps [e.g., *Deardorff*, 1976; *Ooyama*, 1990, 2001; *Zeng et al*., 2005, 2008]. Their conservative nature also makes common SGS models more applicable than they would be, for example, for temperature: SGS models represent the effects of unresolved turbulent eddies, typically at scales of tens of centimeters and larger. At those scales, turbulent dynamics can still be nearly adiabatic and reversible, and conservative variables such as total water and entropy, but generally not nonconservative variables such as temperature, can mix linearly, as is assumed in many common SGS models. Additionally, both total water and entropy are extensive variables, making it possible with suitable discretization to construct a discrete LES code that has closed water and entropy balances.

We use state-of-the-art numerical methods to discretize the equations of motion. LES studies of atmospheric and engineering flows have demonstrated how important the discretization methods are for the quality of LES solutions [e.g., *Ghosal*, 1996; *Kravchenko and Moin*, 1997; *Chow and Moin*, 2003]. They are more important for the quality of LES solutions than, for example, the choice of anelastic or fully compressible equations of motion [*Kurowski et al*., 2014]. We focus on higher-order weighted essentially nonoscillatory (WENO) schemes, based on WENO reconstructions from third through 11th order for the transport of scalars and momentum [*Liu et al*., 1994; *Jiang and Shu*, 1996; *Balsara and Shu*, 2000]. The advantage of WENO schemes, for example, over central difference schemes is that they do not generate spurious numerical oscillations, particularly around sharp gradients, and can also be made monotonicity preserving [*Balsara and Shu*, 2000]. They also conserve integrals of the density-weighted prognostic variables, up to explicit sources and sinks.

We have implemented the LES framework in a mixture of Python and Cython [*Behnel et al*., 2011], high-level languages that are routinely and successfully used in high-performance computing (HPC) applications [e.g., *Turk et al*., 2009]. Python maximizes readability of code, for example, by determining variable types dynamically at runtime. Cython is a superset for the Python programming language that supports static variable types and integration of C functions, both features that are necessary to achieve high computing performance on parallel architectures. Yet Python and Cython lend themselves to scripting applications, for example, for postprocessing of simulation output. Our new Python Cloud Large Eddy Simulation code (PyCLES) allows users to employ a seamless end-to-end Python workflow, instead of the common dichotomy of using simulation code written in a low-level language with static variable types, such as Fortran or C++, but performing postprocessing in a high-level language with dynamic variable types, such as Matlab or Python [*Rossum*, 1995; *Lin*, 2012]. The resulting code has slight performance loss compared, for example, with native Fortran or C++ code (at most approximately 20% [*Wilbers et al*., 2009]). However, the portability across diverse computing platforms, the readability and ease of use of the resulting code, and the seamlessness of simulation and postprocessing, in our view, more than make up for this slight performance disadvantage.

This paper describes the LES framework implemented in PyCLES, from the formulation of the equations of motion to the numerical solution. Section 2 reviews the equations of motion on which PyCLES is based, including boundary conditions and SGS models. Section 3 develops the underlying thermodynamics of moist air and discusses situations in which an entropy-based thermodynamic formulation poses special challenges (e.g., mixed-phase clouds). Section 4 discusses numerical methods, with a focus on how to implement higher-order WENO schemes for the anelastic equations. Section 5 provides simulation results for idealized test cases that illustrate numerical properties of WENO schemes and compares their accuracy and computational cost with traditional centered schemes. Section 6 summarizes the main advances of this

paper. Application of PyCLES to atmospheric boundary layers and convective clouds will be presented in a forthcoming paper. The PyCLES code is available at climate-dynamics.org/software.

2. Equations of Motion

The anelastic equations of *Pauluis* [2008] are based on a linearization of pressure and specific volume fluctuations around a reference state of constant moist entropy. The pressure $p_0(x_3)$ and specific volume $\alpha_0(x_3)=\rho_0^{-1}(x_3)$ in the reference state depend on the height x_3 and are in hydrostatic balance,

$$\alpha_0 \frac{\partial p_0}{\partial x_3} = -g. \tag{1}$$

The specific volume $\alpha_0=\alpha_0\left[s_0, q_{t,0}, p_0(x_3)\right]$ is a function of the thermodynamic state variables specific entropy s_0, total water specific humidity $q_{t,0}$, and pressure $p_0(x_3)$ in the reference state, indicated by the subscript 0. (Appendix B contains a list of the symbols used repeatedly in this paper.)

The fluid is moist air, considered to be an ideal mixture of dry air, water vapor, and water in condensed phases. As is common in hydrodynamics, local thermodynamic equilibrium is assumed to hold on the microscales, the smallest scales at which the continuum equations of motion are valid, so that local temperatures, entropies etc. can be defined. Water that is not in equilibrium with the surrounding air is not part of the fluid and will be treated separately. In particular, hydrometeors that are falling so rapidly that they do not equilibrate with the surrounding air will be treated separately from the water for which a local thermodynamic equilibrium assumption can be made [cf. *Zeng et al*., 2008]. Thus, the total water specific humidity of the fluid is the sum $q_t=q_v+q_l+q_i$ of the water vapor (q_v), liquid (q_l), and ice (q_i) specific humidities of the water phases in local thermodynamic equilibrium. By Gibbs' phase rule, the local thermodynamic equilibrium assumption for moist air implies that the specific entropy s, total water specific humidity q_t, and pressure $p=p_0+p'$ suffice as state variables from which the partitioning of water into its phases (q_v, q_l, and q_i) can be uniquely determined. Thus, all thermodynamic-equilibrium variables needed can be calculated from s, q_t, and $p=p_0+p'$. As discussed by *Pauluis*, thermodynamic consistency of the anelastic equations in this formulation requires that the pressure p in such calculations is approximated by the reference pressure p_0, neglecting the dynamic pressure perturbation $p\prime$. The thermodynamic errors resulting from this approximation are small for typical situations in Earth's atmosphere [*Kurowski et al*., 2014].

What follows begins with a brief review of the anelastic equations of motion of *Pauluis* [2008] and extends his equations by a discussion of the source and sink terms, SGS closures, and boundary conditions that complement them.

2.1. Momentum

The anelastic momentum equations in Cartesian geometry are given by

$$\begin{aligned}\frac{\partial u_i}{\partial t} &+ \frac{1}{\rho_0}\frac{\partial(\rho_0 u_i u_j)}{\partial x_j} = -\frac{\partial \alpha_0 p'}{\partial x_i} + b\delta_{i3} \\ &-\epsilon_{ijk}\delta_{j3} f\left(u_k - u_{g,k}\right) - \frac{1}{\rho_0}\frac{\partial(\rho_0 \tau_{ij})}{\partial x_j} + \Sigma_i\end{aligned} \tag{2}$$

where summation over repeated indices is implied, δ_{ij} is the Kronecker delta, and ϵ_{ijk} is the antisymmetric Levi-Civita symbol. The velocity u_i is the i-th component of the three-dimensional velocity, and $u_{g,i}$ is a corresponding large-scale geostrophic velocity component, in balance with large-scale horizontal pressure gradients, which may be imposed externally. The buoyancy

$$b = g\frac{\alpha[s, q_t, p_0(x_3)] - \alpha_0(x_3)}{\alpha_0(x_3)} \tag{3}$$

depends on the specific volume $\alpha=\alpha[s, q_t, p_0(x_3)]$, which, like other thermodynamic variables, is a function of the state variables specific entropy s, total water specific humidity q_t, and reference pressure p_0 [*Pauluis*, 2008]. The tensor τ_{ij} represents SGS stresses, which include surface stresses at the lower boundary. The term

Σ_i is a momentum source that may represent, for example, externally imposed large-scale turbulent stresses.

2.2. Continuity

The anelastic continuity equation takes the form

$$\frac{\partial \rho_0 u_i}{\partial x_i} = 0. \tag{4}$$

This continuity equation neglects fluctuations $\rho\prime$ of the density about the reference value $\rho_0(x_3)$. It also neglects changes in the mass of fluid elements resulting from the loss or gain of water, for example, by formation or evaporation of precipitation. This is consistent as long as the material rate of change of total water specific humidity q_t is of similar magnitude or smaller than the material rate of change of the neglected relative density fluctuations ρ'/ρ_0. The relative density fluctuations ρ'/ρ_0 are of order 10^{-2} (corresponding to temperature or virtual temperature fluctuations of several kelvin). By contrast, evaporation or precipitation lead to material changes of q_t through changes in the specific humidity q_c of water in condensed phases, which is at most of order 10^{-3} in Earth's atmosphere. Therefore, neglecting mass changes owing to evaporation or precipitation of condensate is usually an unproblematic approximation. Neglecting the gain or loss of water in the continuity equation then remains generally justified provided any externally imposed sources or sinks of water mass also remain small.

By neglecting the density fluctuations, the continuity equation (4) becomes a nondivergence constraint. Time derivatives of density no longer appear in it, which is how sound waves are eliminated from the equations of motion: the speed of sound is effectively infinite. As a result, pressure variations that would usually be communicated by sound waves are communicated instantaneously across the domain. The dynamic pressure perturbation p' therefore satisfies an elliptic diagnostic equation, which can be obtained by multiplying the momentum equation (2) by ρ_0, taking the divergence, and using the anelastic continuity equation (4). This yields

$$\frac{\partial}{\partial x_i}\rho_0 \frac{\partial \alpha_0 p'}{\partial x_i} = \frac{\partial}{\partial x_i}\rho_0 \Gamma_i, \tag{5}$$

where Γ_i is the time tendency of velocity component u_i owing to all factors other than pressure gradients,

$$\begin{aligned}\Gamma_i \quad &= -\frac{1}{\rho_0}\frac{\partial(\rho_0 u_i u_j)}{\partial x_j} + b\delta_{i3} - \epsilon_{ijk}\delta_{j3} f\left(u_k - u_{g,k}\right) \\ &\quad - \frac{1}{\rho_0}\frac{\partial(\rho_0 \tau_{ij})}{\partial x_j} + \Sigma_i.\end{aligned} \tag{6}$$

The advantage of using the anelastic equations is that sound waves are eliminated, making longer time steps possible. The price to pay is that a nonlocal Poisson equation (5) must be solved at each time step to obtain the dynamic pressure perturbation p'.

2.3. Entropy

The balance equation for the specific entropy s is

$$\frac{\partial s}{\partial t} + \frac{1}{\rho_0}\frac{\partial(\rho_0 u_i s)}{\partial x_i} = \frac{Q}{T} + (s_v - s_d)\left(\frac{dq_v}{dt}\right)_e - \frac{1}{\rho_0}\frac{\partial(\rho_0 \gamma_{s,i})}{\partial x_i} + \dot{S}. \tag{7}$$

Here Q is a diabatic heating rate, for example, owing to radiative heating or to externally imposed sources (e.g., convergence of large-scale dry enthalpy fluxes). Externally imposed water vapor sources or sinks (e.g., convergence or divergence of large-scale water vapor fluxes) also modify the entropy. The corresponding source term is the product of the externally imposed tendency $(dq_v/dt)_e$ of the water vapor specific humidity q_v and the difference $s_v - s_d$ between the specific entropies of water vapor and dry air; the apparent entropy sink $-s_d(dq_v/dt)_e$ arises because any imposed tendency $(dq_v/dt)_e$ of the water vapor specific humidity is compensated by a corresponding tendency of the mass fraction $1-q_t$ of dry air: $[d(1-q_t)/dt]_e = -(dq_v/dt)_e$. The flux $\gamma_{s,i}$ is the i-th component of the SGS flux of specific entropy, which at the lower boundary includes surface fluxes. The term $\dot{S}$ subsumes irreversible entropy sources, such as dissipation of kinetic energy and diffusion of water by the SGS closure, evaporation or sublimation of

precipitation, or autoconversion of suspended cloud condensate to precipitation. How such irreversible processes contribute to the entropy source will be discussed in section 3.

In the absence of entropy sources and sinks, the anelastic equations conserve the integral $\int_V \rho_0 s\, dV$ over the fluid volume V. Because the reference density ρ_0 appears here in place of the density ρ as a result of the approximation of the continuity equation (4), this is an approximation—the anelastic approximation—of the second law of thermodynamics for the fluid as a whole. The entropy sources entering the second law for the fluid as a whole are given by the ρ_0-weighted integral of the right-hand side of the entropy equation (7).

The continuous entropy equation (7) could equivalently be expressed in terms of an entropy temperature, $\eta \propto \exp s$, which, like the specific entropy s, is materially conserved in adiabatic and reversible processes [e.g., *Hauf and Höller*, 1987; *Marquet*, 2011]. This would have some advantages for calculating other thermodynamic variables; for example, temperature in the case of dry air becomes a linear function of the entropy temperature (potential temperature) at fixed pressure. However, a disadvantage of using such an entropy temperature is that it does not become a standard extensive thermodynamic variable when multiplied by density. It would be difficult to ensure that a discretized form of the entropy equation (7) with an entropy temperature as a prognostic variable satisfies the second law of thermodynamics for the fluid as a whole when entropy sources are nonzero. The same is true for approximations of the entropy temperature such as the liquid water potential temperature or ice-liquid potential temperature, which have been used in LES codes because, under certain approximations, they are also materially conserved in adiabatic and reversible processes [e.g., *Betts*, 1973; *Deardorff*, 1976; *Tripoli and Cotton*, 1981; *Raymond and Zeng*, 2005; *Savic-Jovcic and Stevens*, 2008; *Heus et al*., 2010; *Matheou et al*., 2011; *Marquet*, 2011; *Savre et al*., 2014]. Not only does using specific entropy itself make the approximations typically made in defining the potential temperatures unnecessary. Additionally, when multiplied by density it becomes the standard entropy, an extensive variable, which facilitates satisfying integral conservation laws when the equations are discretized.

2.4. Water

The anelastic equations for the moist air fluid (including the condensed phases of water in equilibrium) are completed by the balance equation for the total water specific humidity q_t,

$$\frac{\partial q_t}{\partial t} + \frac{1}{\rho_0}\frac{\partial(\rho_0 u_i q_t)}{\partial x_i} = \left(\frac{dq_v}{dt}\right)_e - \frac{1}{\rho_0}\frac{\partial(\rho_0 \gamma_{q,i})}{\partial x_i} + E - P. \tag{8}$$

Here $(dq_v/dt)_e$ again is the tendency of the water vapor specific humidity q_v owing to externally imposed sources, and $\gamma_{q,i}$ is the i-th component of the SGS flux of total water specific humidity, which includes surface fluxes at the lower boundary. The term E represents all microphysical sources of total water specific humidity q_t owing to evaporation or sublimation of precipitation, and P represents the corresponding sinks owing to the formation of precipitation.

Like for entropy and other tracers, the anelastic equations conserve the integral $\int_V \rho_0 q_t\, dV$ in the absence of water sources and sinks, which is an approximation of the conservation law for total water mass, in which the density ρ would appear in place of the reference density ρ_0.

If water out of local thermodynamic equilibrium is present in moist air (e.g., supersaturated water vapor in the upper troposphere or supercooled liquid water in mixed-phase clouds), the partitioning of water into its phases can no longer be uniquely determined from the specific entropy s, total water specific humidity q_t, and pressure $p=p_0+p'$. If such out-of-equilibrium water phases in the moist air (as opposed to in precipitation) are to be modeled explicitly, the balance equation for total water can be replaced by separate balance equations for specific humidities of the phases (e.g., q_v, q_l, and q_i), with explicit conversion terms between the phases. The thermodynamic formulation discussed in section 3 would have to be adjusted correspondingly: the entropy functional would have to be modified, and a determination of the phase partitioning would no longer be necessary. While this would allow a detailed treatment of nonequilibrium processes, what would be lost are some of the advantages accruing from using a conservative variable such as total water as a prognostic variable. Hence, here we focus on using total water as a prognostic variable, with heuristic generalizations for mixed-phase clouds to be discussed in section 3. Extensions of PyCLES to explicit modeling of nonequilibrium processes in moist air are left to future work.

2.5. Precipitation

Precipitation generally is not in equilibrium with the surrounding air and so is not considered part of the moist-air fluid whose dynamics are represented by the momentum, continuity, water, and thermodynamic balance equations. It is treated separately from the equilibrium phases of water, which include the condensate suspended in clouds. Precipitation encompasses all hydrometeors, be they rain, snow, or other species such as graupel or hail. What species need to be considered depends on how the microphysical processes involved in their formation are parameterized.

Each species of precipitation satisfies its own balance equation, which we write in the generic form

$$\frac{\partial q_p}{\partial t}+\frac{1}{\rho_0}\frac{\partial\left[\rho_0(u_i-w_p\delta_{i3})q_p\right]}{\partial x_i}=-\frac{1}{\rho_0}\frac{\partial(\rho_0\gamma_{p,i})}{\partial x_i}+H_p, \tag{9}$$

where q_p is the specific humidity of the precipitation species p, and w_p is its fall velocity (terminal velocity). The flux $\gamma_{p,i}$ is the i-th component of the SGS flux of the specific humidity q_p. The sources and sinks H_p of precipitation species p include conversion from and to other species, and formation of precipitation, for example, by autoconversion of suspended equilibrium condensate to precipitating condensate.

All precipitation ultimately originates with the cloud water that is in equilibrium with the surrounding air. Therefore, conservation of water demands that the sum of all net sources of precipitation equals the net sink of equilibrium water associated with the formation and the evaporation or sublimation of precipitation:

$$\sum_p H_p=P-E. \tag{10}$$

The precise forms of the source terms H_p and of the fall velocities w_p depend on the microphysical parameterizations employed, which we leave open here. For example, the microphysical parameterizations can model conversions between species of precipitation in a way that goes beyond the thermodynamics of the moist-air fluid and the effects (discussed in section 3) that formation and evaporation or sublimation of precipitation has on it.

2.6. Scalar Transport Equation

For arbitrary other scalars ϕ (e.g., chemical tracers), the transport equation takes the form

$$\frac{\partial\phi}{\partial t}+\frac{1}{\rho_0}\frac{\partial(\rho_0 u_i\phi)}{\partial x_i}=-\frac{1}{\rho_0}\frac{\partial(\rho_0\gamma_{\phi,i})}{\partial x_i}+\dot{\Phi} \tag{11}$$

where $\gamma_{\phi,i}$ is the i-th component of the SGS scalar flux of ϕ, and $\dot{\Phi}$ is a source of ϕ.

2.7. Subgrid-Scale Closures

LES is based on the partitioning of the flow into the most energetic larger scales, which are resolved explicitly on the computational grid, and less energetic smaller scales, which are modeled through SGS closures [e.g., *Pope*, 2004]. The resolved scales are more strongly affected by environmental conditions and are usually less isotropic than the unresolved subgrid scales, whose effect on the resolved scales one may expect to be able to represent by universal closures.

Many closures for SGS fluxes take the form of turbulent diffusion down the gradients of the resolved variables. For example, the SGS stress tensor is often written as

$$\tau_{ij}=-2\nu_t S_{ij}, \tag{12}$$

where ν_t is an eddy viscosity and

$$S_{ij}=\frac{1}{2}\left(\frac{\partial u_j}{\partial x_i}+\frac{\partial u_i}{\partial x_j}\right) \tag{13}$$

is the strain rate tensor of the resolved velocities. Analogous formulations hold for scalars such as entropy and total water specific humidity, with eddy diffusivities D_t in place of the eddy viscosity ν_t and resolved tracer gradients in place of the strain-rate term $2S_{ij}$.

As for all diffusive closures, this approach is justifiable if the variable to be diffused is approximately materially conserved, and if the SGS mixing lengths are small compared with the scale over which gradients of the variable in question vary [e.g., *Corrsin*, 1974]. Because specific entropy and total water specific humidity are materially conserved in adiabatic and reversible processes, even in the presence of phase changes of water, they lend themselves better to universal SGS closures than, for example, thermodynamic variables such as temperature or virtual potential temperature: Latent heat release in phase changes of water modifies the temperature and virtual potential temperature, and phase changes can be triggered by SGS mixing, especially near cloud boundaries. Such latent heat release should be taken into account in SGS closures in models that use temperature or virtual potential temperature as prognostic variables. By contrast, specific entropy and total water are conserved even in phase changes of water, as long as these remain reversible; thus, they lend themselves more easily to SGS closures. SGS closures in LES that use prognostic variables such as liquid water potential temperature or ice-liquid potential temperature have similar advantages [e.g., *Betts*, 1973; *Deardorff*, 1976; *Tripoli and Cotton*, 1981; *Raymond and Zeng*, 2005; *Savic-Jovcic and Stevens*, 2008; *Heus et al*., 2010; *Matheou et al*., 2011; *Savre et al*., 2014]. Using specific entropy and total water as prognostic variables avoids making the approximations typically made in defining the potential temperatures.

Several ways of specifying the eddy viscosity ν_t and the corresponding eddy diffusivities D_t for scalars are implemented in PyCLES.

2.7.1. Smagorinsky-Lilly Closure

Smagorinsky [1958, 1963] and *Lilly* [1962] proposed an eddy viscosity ν_t for momentum of the form

$$\nu_t = (c_S \Delta)^2 f_B |S| \tag{14}$$

where c_S is the nondimensional Smagorinsky coefficient (typical value $c_S = 0.17$) and $|S| = (2S_{ij}S_{ij})^{1/2}$ is the magnitude of strain-rate tensor of the resolved velocities. The length scale $\Delta = (\Delta x_1 \Delta x_2 \Delta x_3)^{1/3}$ is the geometric mean of the grid spacings Δx_i. In the case of stable stratification, the eddy viscosity is reduced by the buoyancy factor [*Lilly*, 1962]

$$f_B = \begin{cases} 1 & \text{for} \quad N^2 \leq 0, \\ \max\left[0,\ 1 - N^2/(\mathrm{Pr}_t |S|^2)\right]^{1/2} & \text{for} \quad N^2 > 0. \end{cases} \tag{15}$$

Here Pr_t is the turbulent Prandtl number (typical value $\mathrm{Pr}_t = 1/3$), and N is a moist buoyancy frequency [*Durran and Klemp*, 1982], which takes the reduction of the buoyancy frequency in saturated air into account and which is calculated with the density temperature described in section 3.4.

The corresponding eddy diffusivities D_t for other quantities are then found by invoking their relationship implied by the assumed turbulent Prandtl number, $D_t = \nu_t / \mathrm{Pr}_t$. This gives SGS scalar fluxes of the form

$$\gamma_{\phi,i} = -D_t \frac{\partial \phi}{\partial x_i}, \tag{16}$$

where ϕ stands for any scalar quantity, such as entropy or total water specific humidity.

2.7.2. Turbulence Kinetic Energy Closure

In the turbulence kinetic energy (TKE) closure implemented in PyCLES, the SGS fluxes are likewise modeled diffusively, but the eddy viscosity is determined by

$$\nu_t = c_k \ell e^{1/2}, \tag{17}$$

where c_k is an empirical coefficient (typical value $c_k = 0.1$) [*Moeng and Wyngaard*, 1988]. The mixing length $\ell \leq \Delta$ depends on the local stability [*Deardorff*, 1980],

$$\ell = \begin{cases} \Delta & \text{for} \quad N^2 \leq 0, \\ \min\left[c_N (e/N^2)^{1/2}, \Delta\right] & \text{for} \quad N^2 > 0, \end{cases} \tag{18}$$

where c_N is an empirical constant (typical value $c_N = 0.76$). The SGS TKE e is found from a prognostic second-order equation in which third-order terms and dissipation are modeled semiempirically [e.g., *Cotton et al*., 2011, chap. 3],

$$\frac{\partial e}{\partial t}+\frac{1}{\rho_0}\frac{\partial(\rho_0 u_i e)}{\partial x_i}=\frac{1}{\rho_0}\frac{\partial}{\partial x_i}\left(2\nu_t\rho_0\frac{\partial e}{\partial x_i}\right)+\nu_t|S|^2-D_t N^2-c_\varepsilon\frac{e^{3/2}}{\ell} \tag{19}$$

The terms on the right-hand side represent, respectively, the redistribution of TKE by SGS velocity and pressure fluctuations, shear production of TKE, buoyancy production or destruction of TKE, and dissipation of TKE. The empirical dissipation coefficient c_ε is given by

$$c_\varepsilon=1.9c_k+(0.93-1.9c_k)\frac{\ell}{\Delta}. \tag{20}$$

It takes the value $c_\varepsilon=0.93$ for the typical choice $c_k=0.1$ in conditions of neutral or unstable stratification ($\ell=\Delta$); it is lower in stable stratification ($\ell<\Delta$).

The eddy diffusivity D_t is computed from the eddy viscosity ν_t using a stability-dependent turbulent Prandtl number, as described by *Deardorff* [1980].

2.8. Boundary Conditions

The lower boundary is taken to be flat for now. Vertical momentum fluxes (surface stresses) at it are modeled with a drag law

$$\begin{aligned}\tau_{13}&=-C_m U_b u_{b,1},\\ \tau_{23}&=-C_m U_b u_{b,2},\\ \tau_{33}&=0,\end{aligned} \tag{21}$$

where $U_b=\left(u_{b,1}^2+u_{b,2}^2\right)^{1/2}$ is the horizontal wind speed at the lowest model level (indicated by a subscript b). The drag coefficient C_m is determined from Monin-Obukhov similarity theory, as described in *Byun* [1990]. It depends on the roughness length z_0 for momentum exchange and on the local stability, measured by the bulk Richardson number

$$\mathrm{Ri}=\frac{N_b^2 z_b^2}{U_b^2}, \tag{22}$$

where z_b is the height of the lowest model level, and N_b is the buoyancy frequency obtained from the density potential temperature difference between the lowest model level and the ground. The surface stress (21) provides the lower boundary condition for the SGS stress tensor τ_{ij}. At the upper boundary, the surface stresses are taken to vanish. However, a sponge layer may be included near the top of the domain, in which, for example, velocity fluctuations about the domain mean are linearly damped toward the domain mean through an appropriate momentum source Σ_i.

Similarly, for thermodynamic variables and tracers, the boundary conditions at the top are insulating and impenetrable. If the thermodynamic fluxes at the lower boundary are obtained through coupling with a water surface, they can likewise be modeled through exchange laws,

$$\gamma_{q,3}=-C_q U_b\left(q_{t,b}-q_{v,g}^*\right), \tag{23}$$

$$\gamma_{s,3}=-C_s U_b\left(s_b-s_g^*\right), \tag{24}$$

where $\gamma_{q,3}$ is the surface flux of specific humidity owing to evaporation, and $\gamma_{s,3}$ is the surface flux of specific entropy. We take the exchange coefficients C_q and C_s to be equal, again determined from Monin-Obukhov similarity theory, as described in *Byun* [1990]. The surface fluxes depend on the surface values of the saturation specific humidity $q_{v,g}^*$ and the saturation specific entropy s_g^*, where the subscript g indicates surface (ground) quantities and the asterisk saturation values. The saturation specific entropy at the ground is the sum of the dry-air and the water vapor components and can be expressed as

$$s_g^*=(1-q_{v,g}^*)s_{d,g}^*+q_{v,g}^* s_{v,g}^*, \tag{25}$$

where $s_{d,g}^*=s_{d,g}^*(T_g)$ is the specific entropy of the dry component of air that is saturated with water vapor at the surface with temperature T_g, and $s_{v,g}^*=s_{v,g}^*(T_g)$ is the corresponding specific entropy of the water vapor component, both to be discussed in greater detail in section 3. That is, a microscopic water vapor layer is assumed to cover the surface and to be in thermodynamic equilibrium with it, so that $q_{t,g}=q_{v,g}^*$. This layer

acts as a source of water vapor to the above-lying atmosphere if the atmospheric specific humidity is lower; if the atmospheric specific humidity is higher, condensation onto the ground (dew or frost) occurs. The entropy flux at the surface includes contributions from a dry entropy flux and a water vapor flux. The resulting surface fluxes (23) and (24) provide the lower boundary conditions for the SGS fluxes $\gamma_{q,i}$ and $\gamma_{s,i}$ of total water specific humidity and specific entropy. Over land surfaces, suitable generalizations of these exchange laws, for example, with limited evaporation over dry surfaces, may be employed. (Note that the entropy source resulting from formulating a surface exchange law that is linear in an entropy difference is not identical to the implied entropy source when exchange laws are formulated for dry enthalpy (sensible heat) and water vapor separately, even if all exchange coefficients are equal. In the latter case, an irreversible entropy source arises from the diffusion of sensible heat from the surface into the atmosphere [*Raymond*, 2013]. The magnitude of this irreversible entropy source is typically $\lesssim$1% of the magnitude of the surface entropy source. Similar comments apply to the SGS flux of specific entropy, which differs from the entropy source associated with separate diffusion of dry enthalpy and water by an irreversible entropy source term. The irreversible source terms ultimately arise because specific entropy is a nonlinear function of temperature.)

In LES, the surface fluxes of water and entropy (or equivalent thermodynamic variables) are often prescribed, instead of being determined from exchange laws that couple the atmosphere to an underlying surface reservoir. For example, the surface flux of specific entropy $\gamma_{s,3}$ can be related to the commonly prescribed surface sensible heat flux F_T and latent heat flux F_q per unit area through [cf. *Raymond*, 2013]

$$\gamma_{s,3} = \frac{1}{\rho_{0,b}} \left[\frac{F_T}{T_b} + \frac{F_q}{L_v} \left(s_{v,b} - s_{d,b} \right) \right]. \tag{26}$$

The first term represents the entropy source owing to deposition of the sensible heat flux F_T in the lowest model level with temperature T_b, and the second the specific entropy source owing to the addition of water vapor with specific entropy $s_{v,b}$ to the lowest model level. As for the externally imposed water vapor source in the entropy equation (7), an apparent dry entropy sink proportional to $s_{d,b}$ arises because any increase in specific humidity q_t is compensated by a corresponding decrease in dry air mass fraction $1-q_t$.

3. Moist Thermodynamics

Moist thermodynamics with total water and entropy as prognostic variables poses special challenges. For example, temperature and the phase partitioning of water need to be calculated diagnostically to determine the buoyancy or to model microphysical processes in clouds, and entropy sources and sinks need to be specified. Moreover, situations when local thermodynamic equilibrium may not apply, such as may occur in mixed-phase clouds, need to be considered. The groundwork for such a moist thermodynamics has been laid, among others, by *Ooyama* [1990, 2001] and *Raymond* [2013]. Here we develop it further into a coherent framework consistent with the anelastic equations of motion we use, with heuristic generalizations for mixed-phase clouds.

3.1. Specific Humidities and Entropies

The moist-air fluid is an ideal mixture of dry air, water vapor, liquid water, and ice, all assumed to be in local thermodynamic equilibrium. That is, the total water specific humidity q_t is the sum of the specific humidities of water vapor (q_v), liquid water (q_l), and ice (q_i), so that

$$q_t = q_v + q_l + q_i. \tag{27}$$

We can define a mass fraction q_d of the dry-air component analogously to the specific humidities by

$$q_d = 1 - q_t. \tag{28}$$

The composition of a moist air mass is uniquely characterized by the mass fractions q_d, q_v, q_l, and q_i of its mixture components.

The specific entropy of moist air is the weighted sum of the entropies of the mixture components,

Table 1. Thermodynamic Constants Used in PyCLES[a]

Parameter	Value
Gas constant, dry air	$R_d = 287.1\ \mathrm{J\,kg^{-1}\,K^{-1}}$
Gas constant, water vapor	$R_v = 461.5\ \mathrm{J\,kg^{-1}\,K^{-1}}$
Specific heat, dry air	$c_{pd} = 1004\ \mathrm{J\,kg^{-1}\,K^{-1}}$
Specific heat, water vapor	$c_{pv} = 1859\ \mathrm{J\,kg^{-1}\,K^{-1}}$
Temperature freezing point	$T_f = 273.15\ \mathrm{K}$
Temperature triple point	$T_t = 273.16\ \mathrm{K}$
Standard temperature	$\tilde{T} = 298.15\ \mathrm{K}$
Standard pressure	$\tilde{p} = 10^5\ \mathrm{Pa}$
Vapor pressure triple point	$p_v^{*,t} = 611.7\ \mathrm{Pa}$
Standard entropy, dry air	$\tilde{s}_d = 6864.8\ \mathrm{J\,kg^{-1}\,K^{-1}}$
Standard entropy, water vapor	$\tilde{s}_v = 10513.6\ \mathrm{J\,kg^{-1}\,K^{-1}}$

[a]The standard entropy value for dry air is computed based on the reference data given in *Lemmon et al*. [2000], and the standard entropy value for water vapor is based on the reference data given in *Chase* [1998].

$$\begin{aligned} s &= q_d s_d + q_v s_v + q_l s_l + q_i s_i \\ &= (1-q_t) s_d + q_t s_v - q_l (s_v - s_l) - q_i (s_v - s_i), \end{aligned} \tag{29}$$

where s_d, s_v, s_l, and s_i are the specific entropies of dry air, water vapor, liquid water, and ice. We take the gaseous components (dry air and water vapor) to be ideal gases, so that their specific entropies can be written as

$$s_d = \tilde{s}_d + c_{pd} \log\left(\frac{T}{\tilde{T}}\right) - R_d \log\left(\frac{p_d}{\tilde{p}}\right), \tag{30a}$$

$$s_v = \tilde{s}_v + c_{pv} \log\left(\frac{T}{\tilde{T}}\right) - R_v \log\left(\frac{p_v}{\tilde{p}}\right), \tag{30b}$$

where p_d and p_v are the partial pressures of dry air and water vapor, c_{pd} and c_{pv} are the specific heat capacities at constant pressure, and R_d and R_v are the specific gas constants. Tildes indicate constants under thermodynamic standard conditions, which we choose to be the standard conditions conventional in chemistry, with $\tilde{T} = 298.15\ \mathrm{K}$ and $\tilde{p} = 1000\ \mathrm{hPa}$. The specific heat capacities c_{pd} and c_{pv} vary by less than 1% over typical atmospheric conditions, and we take them to be constant, as is common practice. The thermodynamic constants used in PyCLES are listed in Table 1.

Using the ideal-gas law, we can express the partial pressures p_d and p_v in the gas-phase specific entropies (30) in terms of given specific humidities q_t and q_v and pressure p_0 as

$$p_d = p_0 \frac{1-q_t}{1-q_t+\varepsilon_v^{-1} q_v}, \tag{31a}$$

$$p_v = p_0 \frac{\varepsilon_v^{-1} q_v}{1-q_t+\varepsilon_v^{-1} q_v}, \tag{31b}$$

where $\varepsilon_v = R_d/R_v$ is the ratio of the specific gas constants for dry air and water vapor, or the inverse ratio of their mean molecular weights [*Emanuel*, 1994, chap. 4]. Here we have used $q_d = 1-q_t$ and have neglected the specific volume (but not the mass) of the condensed phases relative to the gas phases; we have also replaced the pressure p by the reference pressure $p_0(x_3)$ for consistency with the thermodynamics of the anelastic approximation [*Pauluis*, 2008]. Given the thermodynamic state variables q_t and p_0 and once the water vapor specific humidity q_v and temperature T are known, the partial pressures (31) and with them the gas-phase specific entropies (30) can be calculated.

Various choices are possible for defining the specific entropies of the condensed phases. They may be approximated as explicit functions of temperature. For example, $s_l \propto c_l \log(T/\tilde{T})$ (specific heat of liquid water c_l) would be an approximation for the entropy of a liquid that ignores effects of surface tension [e.g., *Raymond*, 2013]. Or they may be defined relative to the gas phase specific entropies, using the relations between specific entropy differences and the specific latent heats of phase changes. For example, at thermodynamic equilibrium, when the temperatures and chemical potentials of water vapor and any condensed water are equal, we have

$$s_v - s_l = L_v/T,$$
$$s_v - s_i = L_s/T, \tag{32}$$

where L_v and L_s are the specific latent heats of vaporization and sublimation, respectively. We choose this latter route, as it allows us to use empirical approximations for the temperature dependence of the latent heats L_v and L_s in a thermodynamically consistent manner. Therefore, we take $L_v = L_v(T)$ and $L_s = L_s(T)$ as given, leaving their functional forms unspecified for now, and define the specific entropy differences between water vapor and the condensed phases through the relations (32). The total specific entropy (29) of moist air in thermodynamic equilibrium then becomes [e.g., *Marquet*, 2011; *Raymond*, 2013]

$$s = (1-q_t)s_d + q_t s_v - \frac{q_l L_v + q_i L_s}{T}. \tag{33}$$

It is to be kept in mind that this expression for the specific entropy holds only in local thermodynamic equilibrium, that is, either when the air is unsaturated and no condensed phase is present, or when water vapor is at saturation relative to any condensed phase that is present.

3.2. Temperature and Water Phase Partitioning

From the thermodynamic state variables s, q_t, and p_0, all other thermodynamic variables, such as the temperature and the phase partitioning of water, can be computed.

To calculate the temperature T and the partial specific humidities q_v, q_l, and q_i, one has to invert the expression for the specific entropy (33) subject to the local thermodynamic equilibrium constraint, which fixes the partitioning of water into its phases. This proceeds in two steps [cf. *Ooyama*, 1990, 2001]. First, the temperature $T = T_1$ that corresponds to unsaturated conditions, without any condensate, is calculated. Second, if that temperature T_1 implies supersaturation, it and the water phase partitioning are iteratively adjusted until expression (33) is satisfied for the given specific entropy s and total specific humidity $q_t = q_v + q_l + q_i$.

The first step exploits that in unsaturated conditions, local thermodynamic equilibrium demands $q_l = q_i = 0$, so that $q_v = q_t$. Therefore, the terms involving q_l and q_i in the specific entropy (33) drop out, the temperature T appears only in the logarithms in the gas-phase entropies (30), and the temperature T_1 can be directly calculated from the given specific entropy s and total water specific humidity $q_t = q_v$ by exponentiating the expression (33) for the specific entropy, using expressions (30) for the gas phase specific entropies and the relations (31) between the partial pressures and the given total specific humidity $q_t = q_v$.

The second step is only needed if the specific humidity q_t exceeds the saturation specific humidity $q_v^*(T_1)$. For now, we assume that heterogenous nucleation of ice crystals occurs at temperatures below the freezing point T_f and no supercooled liquid water is present, so that the saturation specific humidity $q_v^*(T)$ is the saturation specific humidity $q_v^{*,l}$ over liquid water above the freezing point, and the saturation specific humidity $q_v^{*,i}$ over ice below it,

$$q_v^*(T) = \begin{cases} q_v^{*,l} & \text{for} \quad T > T_f, \\ q_v^{*,i} & \text{for} \quad T \le T_f. \end{cases} \tag{34}$$

If there is supersaturation, the temperature T is iteratively adjusted while keeping the specific humidity at saturation, $q_v = q_v^*(T)$. Any saturation excess,

$$\sigma = (q_t - q_v^*)\mathcal{H}(q_t - q_v^*), \tag{35}$$

is apportioned to the relevant specific humidity q_l (for $T > T_f$) or q_i (for $T \le T_f$), until expression (33) is satisfied. Here $\mathcal{H}$ is the Heaviside step function, which is included to ensure that the saturation excess σ is positive semidefinite.

The needed saturation specific humidities over liquid and ice are obtained from the Clausius-Clapeyron relations that are consistent with the latent heats of vaporization and sublimation,

$$\frac{d\log\left(p_v^{*,l}\right)}{dT} = \frac{L_v}{R_v T^2}, \qquad \frac{d\log\left(p_v^{*,i}\right)}{dT} = \frac{L_s}{R_v T^2}. \tag{36}$$

Here $p_v^{*,l}$ is the saturation vapor pressure over liquid water, and $p_v^{*,i}$ is that over ice. These forms of the Clausius-Clapeyron relation neglect the specific volume of the condensed phases relative to that of the vapor, as is common; they make no other approximations beyond those we have already made in the thermodynamics. Given the latent heats $L_v(T)$ and $L_s(T)$, the saturation vapor pressures can be computed by integrating the Clausius-Clapeyron relations, with the vapor pressure $p_v^{*,l}(T_t)=p_v^{*,i}(T_t)=p_v^{*,t}$ at the triple point temperature T_t of water as the integration constant (see Table 1). The corresponding saturation specific humidities q_v^* are obtained using the inverse of the relation (31b) between vapor pressure p_v and specific humidities,

$$q_v^* = \frac{\varepsilon_v(1-q_t)p_v^*}{p_0 - p_v^*}, \tag{37}$$

where p_v^* is the saturation vapor pressure over liquid water for $T > T_f$, or that over ice for $T \leq T_f$. Standard relations between vapor pressure and specific humidity equate q_t and q_v^* and therefore differ slightly from (37) [cf. *Emanuel*, 1994, chap. 4]. Consistent with using the total specific humidity q_t as a prognostic variable, relation (37) takes q_t and by implication the dry air fraction $1-q_t$ as given, which is justifiable as long as the air is saturated, so that the contribution of the water vapor to the total specific humidity q_t is the saturation specific humidity q_v^*. Equation (37) is needed only in that case, for the computation of the saturation excess σ.

3.3. Generalizations for Mixed-Phase Clouds

In mixed-phase clouds, metastable supercooled liquid water can coexist with ice at temperatures below the freezing point T_f. The supercooled liquid water freezes spontaneously below the homogeneous nucleation temperature $T_i \approx 233$ K. At temperatures between T_i and T_f, water vapor and the mixture of supercooled liquid and ice are not in local thermodynamic equilibrium. For example, the ice-liquid partitioning in a fluid element does not depend on thermodynamic state variables alone, but also on the history of the fluid element (e.g., supercooled liquid can freeze spontaneously when a fluid element cools, but it does not form spontaneously when it warms). Therefore, the local thermodynamic equilibrium assumption strictly cannot be made for supercooled liquid in mixed-phase clouds, and it would be more accurate, for example, to introduce separate balance equations for the individual phases of water, with explicit transitions between them (see section 2.4).

Nonetheless, one can represent supercooled liquid water at temperatures between T_i and T_f in an equilibrium framework heuristically, by introducing an empirical phase partitioning function $\lambda(T)$ that increases monotonically with temperature and indicates what proportion of the condensed water is in the liquid phase. That is, any saturation excess $\sigma > 0$ is apportioned into ice and liquid according to [e.g., *Squires and Turner*, 1962; *Lord et al.*, 1984; *Tao et al.*, 1989]

$$q_l = \lambda(T)\sigma, \qquad q_i = [1-\lambda(T)]\sigma, \tag{38}$$

where

$$\lambda(T) = \begin{cases} 0 & \text{for} \quad T < T_i, \\ \lambda(T), \quad 0 \leq \lambda(T) \leq 1 & \text{for} \quad T_i \leq T \leq T_f, \\ 1 & \text{for} \quad T_f < T \end{cases} \tag{39}$$

is the liquid fraction. Such an empirical phase partitioning that only depends on temperature ignores, e.g., the history of fluid elements. It effectively treats freezing of supercooled liquid as a reversible phase transformation, which in reality it is not.

To make a temperature-dependent phase partitioning of the condensed phases thermodynamically consistent, a saturation specific humidity q_v^* over the ice-liquid mixture in the intermediate temperature range T_i

$\leq T \leq T_f$ needs to be defined, so that the saturation excess (35) is thermodynamically consistent with the partitioning. With the phase partitioning (38), the expression (33) for the specific entropy can be rewritten as

$$s=(1-q_t)s_d+q_t s_v-\frac{\sigma L}{T}, \tag{40}$$

where

$$L=\lambda L_v+(1-\lambda)L_s \tag{41}$$

is an effective specific latent heat, which can be related to the difference between the water vapor specific entropy s_v and an effective condensate specific entropy $\lambda s_l+(1-\lambda)s_i$ analogously to equation (32):

$$L/T=s_v-[\lambda s_l+(1-\lambda)s_i]. \tag{42}$$

For thermodynamic consistency, the same entropy difference or latent heat needs to appear in the Clausius-Clapeyron relation, which therefore becomes

$$\frac{d\log(p_v^*)}{dT}=\frac{L}{R_v T^2}. \tag{43}$$

Integration of this Clausius-Clapeyron relation gives the saturation vapor pressure p_v^*, from which the saturation specific humidity q_v^* can be computed via equation (37). In the special case of $T_i = T_f$, the phase partitioning function $\lambda(T)$ becomes a step function, and the relations (38)–(43) reduce to the case of pure liquid or pure ice clouds.

For general phase partitioning functions $\lambda(T)$, neither the saturation vapor pressure p_v^* nor the saturation specific humidity q_v^* are simple weighted means of the corresponding quantities over liquid water and ice, as they are in some other ways of treating mixed-phase clouds [e.g., *Grabowski*, 1998; *Khairoutdinov and Randall*, 2003]. Rather, the effective latent heat L is such a weighted mean, and the saturation vapor pressure and saturation specific humidity need to be computed consistently with it, to account for the implied entropy difference between water vapor and the effective condensate. This implies in particular that even at temperatures $T < T_i$, the saturation vapor pressure over ice is modified because it needs to be computed by an integration of the Clausius-Clapeyron relation (43) across the intermediate temperature range $T_i \leq T \leq T_f$ with effective latent heat L [cf. *Ooyama*, 1990]. For example, for a phase partitioning function with all liquid ($\lambda = 1$) above $T_f = -5^\circ$C, all ice ($\lambda = 0$) below $T_i = -20^\circ$C, and a linearly varying λ at temperatures in between [cf. *Grabowski*, 1998], the saturation vapor pressure p_v^* at low temperatures $T < T_i$ is about 13% higher than the actual saturation vapor pressure over ice. This bias of the saturation vapor pressure at low temperatures is an unrealistic aspect of this formulation. It arises from heuristically forcing nonequilibrium processes into a thermodynamic equilibrium framework.

Appendix A outlines a general saturation adjustment scheme that uses the secant method to determine the temperature T and the specific humidities q_v, q_l, and q_i that are consistent with a given specific entropy s and total water specific humidity q_t and with given functions $\lambda(T)$, $L_v(T)$, and $L_s(T)$. This scheme, along with several common choices for the temperature dependence of the latent heats of vaporization $L_v(T)$ and sublimation $L_s(T)$ and for the liquid fraction $\lambda(T)$, are implemented in PyCLES. PyCLES precomputes the saturation vapor pressure p_v^* by integration of the Clausius-Clapeyron relations (43) for the given functions $\lambda(T)$, $L_v(T)$, and $L_s(T)$, and uses linear interpolation in look-up tables to determine the saturation vapor pressure at LES runtime.

3.4. Specific Volume and Density Temperature

Once the temperature and the water phase partitioning are known, the specific volume on which the buoyancy depends can be calculated. We again neglect the specific volume of the condensed phases relative to that of the gas phases, as we already did in the relations (31) between partial pressures and specific humidities and in the Clausius-Clapeyron relations (36) and (43); however, we do take the effect of condensate mass on the specific volume into account [cf. *Emanuel*, 1994, chap. 4]. Using the ideal-gas law, the specific volume then becomes

$$\alpha = \frac{(q_d R_d + q_v R_v)T}{p_0} = \frac{R_d T}{p_0}(1 - q_t + \varepsilon_v^{-1} q_v), \tag{44}$$

where, again, the reference pressure $p_0(x_3)$ was substituted for the pressure p for consistency with the anelastic approximation. That is, as is well known, the presence of water vapor increases the specific volume and hence the buoyancy of fluid elements (because $1 - q_t + \varepsilon_v^{-1} q_v = (\varepsilon_v^{-1} - 1) q_v - q_l - q_i$ and $\varepsilon_v^{-1} - 1 \approx 0.61 > 0$); by contrast, the presence of condensate ($q_l > 0$ or $q_i > 0$) reduces the specific volume and buoyancy of fluid elements. Since the total water specific humidity q_t includes only water phases in thermodynamic equilibrium, the effect of precipitation on the buoyancy is not included. However, the thermodynamic effects of precipitation formation and evaporation/sublimation on the buoyancy, which are generally more significant [e.g., *Cotton et al*., 2011, chap. 7], are accounted for through the entropy sources described in section 3.6.1.

The density temperature and potential temperature that take the density effects of water vapor and condensate into account can be defined in the usual way [*Emanuel*, 1994], as

$$T_\rho = T(1 - q_t + \varepsilon_v^{-1} q_v) \tag{45}$$

and

$$\theta_\rho = T_\rho \left(\frac{\tilde{p}}{p_0(x_3)} \right)^\kappa, \tag{46}$$

with adiabatic exponent $\kappa = R_d / c_{pd}$ (i.e., ignoring the small effect of water loading on the adiabatic exponent of moist air). From these, the buoyancy frequencies required by the SGS closures can be calculated [*Durran and Klemp*, 1982].

Because the temperature, phase partitioning, and derived thermodynamic variables such as the specific volume depend nonlinearly on the prognostic state variables specific entropy s and total water specific humidity q_t, SGS fluctuations in the prognostic state variables can have a nonzero mean effect on the grid-scale temperature, phase partitioning, etc. That is, SGS fluctuations can lead to deviations from local thermodynamic equilibrium on the grid scale of the LES. Therefore, just like the nonlinear terms in the prognostic equations, the nonlinear aspects of the thermodynamics require a SGS closure. This can be accomplished by assuming a probability density function (e.g., Gaussian) for the SGS fluctuations of the prognostic variables, and using the assumed probability density function to calculate the mean effect of the SGS fluctuations on grid-scale variables [e.g., *Sommeria and Deardorff*, 1977; *Larson et al*., 2001; *Tompkins*, 2002; *Cotton et al*., 2011]. Adaptation of such a procedure to our context will be described in a forthcoming publication.

3.5. Entropy Temperature

Because it is common in the atmospheric sciences to express entropies in units of temperature, one may define an entropy temperature that reduces to the standard dry potential temperature in the absence of water and is materially conserved to the extent specific entropy and total water specific humidity are simultaneously conserved. Various choices of such entropy temperatures are available [e.g., *Hauf and Höller*, 1987; *Romps and Kuang*, 2010; *Marquet*, 2011]. A simple choice, in its essence proposed by *Marquet* [1993], is

$$\theta_s = \tilde{T} \exp\left(\frac{s - (1 - q_t)\tilde{s}_d - q_t \tilde{s}_v}{c_{pm}} \right), \tag{47}$$

where $c_{pm} = (1 - q_t) c_{pd} + q_t c_{pv}$ is the specific heat of moist air at constant pressure. It is clear that this entropy temperature is materially conserved if s and q_t are: all other quantities in (47) are constants. From the definition of specific entropy (29)–(33), it can be seen that the entropy temperature θ_s is related to the temperature T and other thermodynamic variables through

$$\theta_s = T \left(\frac{\tilde{p}}{p_d} \right)^{(1 - q_t) R_d / c_{pm}} \left(\frac{\tilde{p}}{p_v} \right)^{q_t R_v / c_{pm}} \exp\left(- \frac{L\sigma}{c_{pm} T} \right). \tag{48}$$

For dry air ($q_t = 0$, $c_{pm} = c_{pd}$, $p_d = p_0(x_3)$), this reduces to the standard dry potential temperature with reference pressure $\tilde{p}$. For small total water specific humidity q_t, it has the same conservation properties as the

ice-liquid water potential temperature of *Tripoli and Cotton* [1981], in the sense that both are materially conserved in reversible processes if q_t is conserved. We use this entropy temperature to illustrate the results of the numerical test cases in section 5.

3.6. Irreversible Entropy Sources

The irreversible entropy source $\dot{S}$ consists of several parts, associated with precipitation processes and SGS dynamics. *Raymond* [2013] discusses these entropy sources in detail, for equations of motion that use dry air as the fluid, with mixing ratios as moisture variables. We draw on his work in what follows and adapt it to our setting, which uses moist air as the fluid, with specific humidities as moisture variables. This leads to some differences in the entropy source $\dot{S}$, principally because of the relation $q_d=1-q_t$ between dry air mass fraction q_d and total water specific humidity q_t: Changes in q_t imply compensating changes in q_d, which lead to apparent sinks of dry entropy s_d that do not occur when dry air is taken to be the fluid.

3.6.1. Precipitation Processes

Precipitation processes give rise to entropy sources and sinks for the moist-air fluid, for example, when precipitation forms, when hydrometeors evaporate or sublimate, or when heat is transferred from the air to precipitating hydrometeors.

1. When precipitation forms, the air loses water at the precipitation formation rate $P=-(dq_t/dt)_P>0$. We assume that the associated entropy changes are only due to the mass transfer out of the fluid of condensed water with specific entropy $s_c=s_v^*-L(T)/T$, i.e., we assume local thermodynamic equilibrium for the condensate from which the precipitation forms. Then, differentiation of the specific humidities in the specific entropy (40) and using $P=-(dq_t/dt)_P=-(dq_c/dt)_P=(dq_d/dt)_P$, where q_c is the specific humidity of the condensate that becomes precipitation, yields the entropy source

$$\dot{S}_P=\left(s_d(T)-s_v^*(T)+\frac{L(T)}{T}\right)P. \tag{49}$$

   The first term is the apparent entropy source owing to the changes in q_d that compensate the changes in q_t. The second and third term can be interpreted as the entropy sink owing to the loss of water vapor at saturation from the fluid ($-s_v^*P<0$), and the entropy source owing to latent heat release during condensation or deposition of that water vapor ($LP/T>0$).

2. Conversely, when precipitating hydrometeors evaporate or sublimate, the air gains water at the evaporation/sublimation rate $E=(dq_t/dt)_E>0$. The precipitating condensate is not in equilibrium with the surrounding air at temperature T; instead, its temperature can be approximated by the wet-bulb temperature $T_w\le T$ (with equality at saturation). The specific entropy of the precipitation can therefore be approximated as $s_p=s_v^*(T_w)-L(T_w)/T_w$, which assumes that the hydrometeors are in equilibrium with an adjacent microscopic layer that is saturated with water vapor at the wet-bulb temperature. Thus, the entropy source owing to the mass transfer into the fluid associated with the evaporation/sublimation of hydrometeors is

$$S_E=\left(s_v^*(T_w)-\frac{L(T_w)}{T_w}-s_d(T)\right)E, \tag{50}$$

   where the term $-s_d(T)E$ represents the apparent entropy sink arising because $(dq_d/dt)_E=-E$.

3. From the saturated microscopic layer adjacent to evaporating/sublimating hydrometeors, water vapor diffuses into the surrounding air, and heat is conducted from the surrounding air toward the hydrometeors to balance their cooling by evaporation/sublimation. *Raymond* [2013] showed that both processes combined give rise to a diffusive entropy source

$$S_D=E\left[-R_v\log\left(\frac{p_v}{p_v^*(T)}\right)+c_{pv}\log\left(\frac{T}{T_w}\right)\right], \tag{51}$$

   which is positive semidefinite because $p_v\le p_v^*(T)$ and $T_w\le T$.

4. As hydrometeors fall into warmer air, heat is transferred from the surrounding air to the hydrometeors to keep them at the wet-bulb temperature T_w and, if falling ice melts at or near the freezing level, to supply the latent heat of fusion (assumed to be transferred instantaneously). This results in a (usually negative) heat source for the moist-air fluid,

$$Q_p = q_p(u_3 - w_p)\left[-c_c\frac{\partial T_w}{\partial x_3} + L_f\delta(x_3 - z_f)\right] \tag{52}$$

where c_c is the specific heat of the falling precipitation species, L_f is the specific latent heat of fusion, $\delta(\cdot)$ is the Dirac delta function, and z_f is the freezing level where $T_w = T_f$. This assumes that the freezing level is well defined, that is, ice and liquid water do not coexist over a wider temperature range. The heat source acts on the surrounding air at temperature T, so that the associated entropy source is

$$\dot{S}_Q = \frac{Q_p}{T}. \tag{53}$$

(*Raymond* [2013] uses the wet-bulb temperature T_w instead of the local air temperature T in the source term (53), which corresponds to assuming that the heat is transferred at the wet bulb temperature to a microscopic vapor layer surrounding and in equilibrium with a hydrometeor. A separate diffusive entropy generation term owing to diffusion from this microscopic layer to the surrounding air would then have to be included.)

5. The gravitational acceleration of hydrometeors that fall at their terminal velocity is balanced by the aerodynamic drag on them, and the drag implies frictional dissipation of kinetic energy in the microscopic shear zones surrounding the hydrometeors [*Pauluis et al.*, 2000; *Pauluis and Held*, 2002a,b]. This implies a heat and thus an entropy source for the surrounding air, which can be written as

$$\dot{S}_W = \frac{g q_p w_p}{T}. \tag{54}$$

3.6.2. SGS Dynamics

The representation of SGS dynamics in LES also gives rise to entropy sources:

1. The SGS stresses imply dissipation of kinetic energy, which is converted into heat and implies an entropy source

$$\dot{S}_\tau = \frac{\nu_t |S|^2}{T} \tag{55}$$

for eddy viscosity closures of the form (12).

2. The SGS diffusion of water implies a material tendency dq_t/dt of total water specific humidity and a compensating material tendency $-dq_d/dt$ of dry-air mass fraction, associated with an entropy source that is analogous to other entropy sources associated with specific humidity changes:

$$\dot{S}_\gamma = -\frac{s_w - s_d}{\rho_0}\frac{\partial(\rho_0 \gamma_{q,i})}{\partial x_i}. \tag{56}$$

Here $s_w = s_v - (\sigma/q_t)L/T$ is the specific entropy of the water component of the moist air, including vapor and condensate representing a fraction σ/q_t of the total water mass.

Each of these entropy sources is discussed in greater detail in *Raymond* [2013]. The total irreversible entropy source is the sum of the sources (49)–(56) and is taken into account in PyCLES. In models with SGS diffusion of dry enthalpy, an additional irreversible entropy source would appear (see note after equation (25)). But it does not appear here because dry enthalpy does not diffuse separately from entropy, and we ignore molecular diffusion and conduction.

4. Numerical Approximations

We seek a discrete approximation of the prognostic equations that conserves first moments (e.g., total water, entropy) to the same degree the continuous anelastic equations do and is accurate at high order. We consider the domain $\mathbf{x} = (x_1, x_2, x_3) \in [X_{1,l}, X_{1,r}] \times [X_{2,l}, X_{2,r}] \times [X_{3,l}, X_{3,r}]$. Defining superscript indices i, j, k such that $0 \le i \le N_{x_1}$, $0 \le j \le N_{x_2}$, $0 \le k \le N_{x_3}$, we construct uniformly spaced coordinate grid points $x_1^i = i\Delta x_1$, $x_2^j = j\Delta x_2$, and $x_3^k = k\Delta x_3$, as well as coordinate-grid half points $x_1^{i+\frac{1}{2}} = i\Delta x_1 + 0.5\Delta x_1$, $x_2^{j+\frac{1}{2}} = j\Delta x_2 + 0.5\Delta x_2$, and $x_3^{k+\frac{1}{2}} = k\Delta x_3 + 0.5\Delta x_3$. The domain is partitioned into a collection of grid cells $C^{i,j,k} = \left[x_1^i, x_1^{i+1}\right) \times \left[x_2^j, x_2^{j+1}\right) \times \left[x_3^k, x_3^{k+1}\right)$ that are centered on the half points $\left(x_1^{i+1/2}, x_2^{j+1/2}, x_3^{k+1/2}\right)$.

The discrete equations are solved on a staggered grid to avoid the pressure-velocity decoupling that arises when one discretizes the anelastic continuity equation on a collocated grid. In the staggered-grid arrangement, the prognostic scalar variables are located at the centers of the grid cells $C^{i,j,k}$, and the velocity components are located at the centers of the grid cell faces. In particular, the u_1 velocity component is located at $\left(x_1^i, x_2^{j+1/2}, x_3^{k+1/2}\right)$, the u_2 velocity component at $\left(x_1^{i+1/2}, x_2^j, x_3^{k+1/2}\right)$, and the u_3 velocity component at $\left(x_1^{i+1/2}, x_2^{j+1/2}, x_3^k\right)$. This grid configuration is known as the Arakawa-C grid [*Arakawa and Lamb*, 1977].

The prognostic equations for momentum and scalar variables can be written in semidiscrete form with respect to each grid cell $C^{i,j,k}$ in terms of the state vector

$$\mathbf{T}^{i,j,k} = \begin{bmatrix} u_1^{i,j+\frac{1}{2},k+\frac{1}{2}} \\ u_2^{i+\frac{1}{2},j,k+\frac{1}{2}} \\ u_3^{i+\frac{1}{2},j+\frac{1}{2},k} \\ \phi^{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \end{bmatrix}, \tag{57}$$

with scalar prognostic variables represented by $\phi \in \{s, q_p, q_t\}$, a reference density vector

$$\boldsymbol{\rho_0}^{i,j,k} = \begin{bmatrix} \rho_0^{k+\frac{1}{2}} \\ \rho_0^{k+\frac{1}{2}} \\ \rho_0^{k} \\ \rho_0^{k+\frac{1}{2}} \end{bmatrix}, \tag{58}$$

and a source vector containing all nonpressure and nontransport terms

$$\dot{\mathbf{T}}^{i,j,k} = \begin{bmatrix} \dot{u}_1^{i,j+\frac{1}{2},k+\frac{1}{2}} \\ \dot{u}_2^{i+\frac{1}{2},j,k+\frac{1}{2}} \\ \dot{u}_3^{i+\frac{1}{2},j+\frac{1}{2},k} \\ \dot{\phi}^{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \end{bmatrix}. \tag{59}$$

In terms of these quantities, the prognostic equations become

$$\begin{aligned} \frac{d\mathbf{T}}{dt^{i,j,k}} = &- \frac{1}{\boldsymbol{\rho_0}^{i,j,k}} \frac{\mathbf{U}^{i,j,k} - \mathbf{U}^{i-1,j,k}}{\Delta x_1} - \frac{1}{\boldsymbol{\rho_0}^{i,j,k}} \frac{\mathbf{V}^{i,j,k} - \mathbf{V}^{i,j-1,k}}{\Delta x_2} \\ &- \frac{1}{\boldsymbol{\rho_0}^{i,j,k}} \frac{\mathbf{W}^{i,j,k} - \mathbf{W}^{i,j,k-1}}{\Delta x_3} + \dot{\mathbf{T}}^{i,j,k}. \end{aligned} \tag{60}$$

The first three terms on the right-hand side denote the divergence of the combined advective and SGS fluxes in the x_1, x_2, and x_3 directions. They are given by

$$\mathbf{U}^{i,j,k} = \begin{bmatrix} (\rho_0 \tilde{u}_1 \tilde{u}_1 + \rho_0 \tilde{\tau}_{11})^{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \\ (\rho_0 \tilde{u}_1 \tilde{u}_2 + \rho_0 \tilde{\tau}_{21})^{i+1,j,k+\frac{1}{2}} \\ (\rho_0 \tilde{u}_1 \tilde{u}_3 + \rho_0 \tilde{\tau}_{31})^{i+1,j+\frac{1}{2},k} \\ (\rho_0 u_1 \tilde{\phi} + \rho_0 \tilde{\gamma}_{\phi,1})^{i+1,j+\frac{1}{2},k+\frac{1}{2}} \end{bmatrix}, \tag{61}$$

$$\mathbf{V}^{i,j,k} = \begin{bmatrix} (\rho_0 \tilde{u}_2 \tilde{u}_1 + \rho_0 \tilde{\tau}_{12})^{i,j+1,k+\frac{1}{2}} \\ (\rho_0 \tilde{u}_2 \tilde{u}_2 + \rho_0 \tilde{\tau}_{22})^{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \\ (\rho_0 \tilde{u}_2 \tilde{u}_3 + \rho_0 \tilde{\tau}_{32})^{i+\frac{1}{2},j+1,k} \\ (\rho_0 u_2 \tilde{\phi} + \rho_0 \tilde{\gamma}_{\phi,2})^{i+\frac{1}{2},j+1,k+\frac{1}{2}} \end{bmatrix}, \tag{62}$$

$$\mathbf{w}^{i,j,k} = \begin{bmatrix} (\rho_0 \tilde{u}_3 \tilde{u}_1 + \rho_0 \tilde{\tau}_{13})^{i,j+\frac{1}{2},k+1} \\ (\rho_0 \tilde{u}_3 \tilde{u}_2 + \rho_0 \tilde{\tau}_{23})^{i+\frac{1}{2},j,k+1} \\ (\rho_0 \tilde{u}_3 \tilde{u}_3 + \rho_0 \tilde{\tau}_{33})^{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \\ (\rho_0 u_3 \tilde{\phi} + \rho_0 \tilde{\gamma}_{\phi,3})^{i+\frac{1}{2},j+\frac{1}{2},k+1} \end{bmatrix}. \tag{63}$$

The staggered variable arrangement prevents direct computation of the advective fluxes, which instead must be reconstructed by interpolation. The form of the reconstruction required depends on whether the advected variable is a scalar quantity or a momentum component. Scalar fluxes are computed at the location of the advecting velocities, so that no reconstruction of the velocity field is necessary. However, for the momentum fluxes, both the momentum component and the advecting velocity need to be reconstructed. The terms requiring reconstruction are marked by tildes $(\tilde{\cdot})$ in the fluxes (61)–(63). The reference density ρ_0 can be determined to arbitrary precision at initialization; it is horizontally homogeneous and fixed in time. Therefore, the value of the reference density is assumed to be known everywhere; no reconstruction is needed.

4.1. Transport Schemes

Numerical properties of the transport schemes can be varied by modifying the way in which reconstructions are performed. To simplify the discussion, we will explicitly consider the approximation of only two of the fluxes appearing in (61)–(63), namely,

$$(\rho_0 u_1 \tilde{\phi})^{i+1} = (\rho_0 u_1 \tilde{\phi})^{i+1,j+\frac{1}{2},k+\frac{1}{2}} \tag{64}$$

and

$$(\rho_0 \tilde{u}_1 \tilde{u}_1)^{i+\frac{1}{2}} = (\rho_0 \tilde{u}_1 \tilde{u}_1)^{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}, \tag{65}$$

where we suppressed the j and k grid indices, which remain unchanged throughout the following discussion. The computation of all other fluxes proceeds analogously to these two.

Because ϕ is located at the $x_1^{i+\frac{1}{2}}$ grid half point, reconstruction is required to compute its flux at x_1^{i+1}. Likewise, since u_1 is located at the x_1^{i} grid point, reconstruction must be performed to obtain its flux at $x_1^{i+\frac{1}{2}}$. The numerical schemes we consider are based on polynomial interpolations to reconstruct these values.

An interpolation of ϕ at x_1^{i+1} to r-th order accuracy requires knowledge of ϕ at r other spatial locations. For $r \geq 2$, there are r candidate interpolation stencils that contain the point x_1^{i+1}. We define two collections of such stencils, for an interpolation of r-th order at a spatial location x_1^{i+1}. One is upwind biased for positive advecting velocities u_1,

$$S_{k^*}^{r,+}(x_1^{i+1}) = \left\{ x_1^{i+\frac{3}{2}-r+k^*}, \ldots, x_1^{i+\frac{1}{2}+k^*} \right\}. \tag{66}$$

The other is upwind biased for negative advecting velocities u_1,

$$S_{k^*}^{r,-}(x_1^{i+1}) = \left\{ x_1^{i+\frac{3}{2}-k^*}, \ldots, x_1^{i+\frac{1}{2}+r-k^*} \right\}. \tag{67}$$

Here $k^* = 0, 1, \ldots, r-1$ denotes the shift of the stencil relative to the grid index under consideration (not to be confused with the grid index k in the x_3 direction), and superscripts + and – are used to denote upwind biasing for positive and negative advecting velocities, respectively. Analogous positively and negatively biased collections of stencils can be defined for the interpolation of u_1 to $x_1^{i+\frac{1}{2}}$. The upwind biased stencil for positive velocities u_1 is

$$S_{k^*}^{r,+}(x_1^{i+\frac{1}{2}}) = \{ x_1^{i+1-r+k^*}, \ldots, x_1^{i+k^*} \}, \tag{68}$$

and the upwind biased stencil for negative velocities u_1 is

$$S_{k^*}^{r,-}(x_1^{i+\frac{1}{2}}) = \{ x_1^{i+1-k^*}, \ldots, x_1^{i+r-k^*} \}. \tag{69}$$

For conciseness, we focus on the reconstruction from the upwind biased stencils for positive velocities. The reconstruction from the upwind biased stencils for negative velocities follows analogously, given that the reconstruction is symmetric about the reconstruction point.

For each candidate stencil, polynomial reconstructions of order r for scalars and velocities can be expressed as

$$\tilde{\phi}^{i+1,+} = q^{r,+}_{k^*}\left(\phi^{i+\frac{3}{2}-r+k^*}, \ldots, \phi^{i+\frac{1}{2}+k^*}\right) \tag{70}$$

and

$$\tilde{u}_1^{i+\frac{1}{2},+} = q^{r,+}_{k^*}\left(u_1^{i+1-r+k^*}, \ldots, u_1^{i+k^*}\right), \tag{71}$$

where

$$q^{r,+}_{k^*}(g_0, \ldots, g_{r-1}) = \sum_{l=0}^{r-1} g_l a^r_{k^*,l} \tag{72}$$

is the k^*-th Lagrange interpolating polynomial with

$$a^r_{k^*,l} = \sum_{m=l+1}^{r} \frac{\sum_{j^*=0, j^*\neq m}^{r} \prod_{q=0, q\neq m, j^*}^{r} (r-k^*-q)}{\prod_{j^*=0, j^*\neq m}^{r} (m-j^*)}, \tag{73}$$

as described in *Shu* [1998]. (The Lagrange interpolating polynomial $q^{r,+}_{k^*}$ should not be confused with specific humidities such as q_t.)

Various numerical approximations can be formed based on these interpolations. For example, using an even-order accuracy interpolation with stencils centered on the reconstruction location provides a centered approximation to the flux divergences (60). Convex combinations of interpolating polynomials serve as the basis for WENO schemes.

4.1.1. Centered Schemes

4.1.1.1. Second-Order Schemes

A second-order centered approximation to the scalar advective flux is obtained by using the reconstruction (70) with $r = 2$ and $k^* = 1$, leading to the scalar

$$\tilde{\phi}^{i+1} = \frac{1}{2}\left(\phi^{i+\frac{1}{2}} + \phi^{i+\frac{3}{2}}\right). \tag{74}$$

With the advecting velocity u_1^{i+1}, the advective flux is given by

$$(\rho_0 u_1 \tilde{\phi})^{i+1} = \frac{\rho_0 u_1^{i+1}}{2}\left(\phi^{i+\frac{1}{2}} + \phi^{i+\frac{3}{2}}\right). \tag{75}$$

The corresponding second-order centered approximation to the momentum flux is obtained by using the reconstruction (71) with $r = 2$ and $k^* = 1$, leading to the velocity

$$\tilde{u}_1^{i+\frac{1}{2}} = \frac{1}{2}\left(u_1^i + u_1^{i+1}\right) \tag{76}$$

and advective momentum flux

$$(\rho_0 \tilde{u}_1 \tilde{u}_1)^{i+\frac{1}{2}} = \frac{\rho_0}{4}\left(u_1^i + u_1^{i+1}\right)\left(u_1^i + u_1^{i+1}\right). \tag{77}$$

This second-order centered scheme is commonly referred to as the *Harlow and Welch* [1965] scheme and has been widely used in LES. Because it is based on a discretization of fluxes (61)–(63), it conserves first moments of momentum and scalars such as total water and entropy. It also conserves kinetic energy (a second moment).

4.1.1.2. Fourth and Sixth-Order Schemes

Higher-order centered difference schemes can be constructed in a similar way. For example, for a fourth-order scheme, one chooses $r = 4$ and $k^* = 2$ in the polynomial reconstructions (70) and (71). For a sixth-order scheme, one chooses $r = 6$ and $k^* = 3$. The resulting schemes are the fourth and sixth-order centered schemes described in *Wicker and Skamarock* [2002]. They are used in the Weather Research and Forecasting Model (WRF) [*Skamarock and Klemp*, 2008] and in the Dutch Atmospheric Large Eddy Simulation (DALES) code [*Heus et al.*, 2010]. They are also implemented in PyCLES. Schemes of yet higher order can be constructed using other values of r and k^*. Such schemes still conserve first moments of momentum and scalars

such as total water and entropy. However, like all centered schemes of higher than second order, they generally do not conserve second moments such as kinetic energy [e.g., *Morinishi et al.*, 1998].

4.1.1.3. Shortcomings

Centered schemes have even-order accuracy and symmetric stencil weights. Consequently, the leading-order term in the resulting discrete equations is dispersive, leading to short-wavelength oscillations in the numerical solutions [e.g., *Wicker and Skamarock*, 2002; *Ghosh and Baeder*, 2012]. As a means of controlling the short-wavelength oscillations, *Wicker and Skamarock* [2002] propose third- and fifth-order schemes formed by adding higher-order dissipative terms to their fourth- and sixth-order schemes. While *Wicker and Skamarock* [2002] sought to reduce dispersive numerical error by incorporating dissipation in the numerical scheme itself, others have relied upon artificial viscosity/diffusion or the SGS closures to supply additional dissipation. However, in the LES context, relying on the SGS closures to control dispersive numerical error obscures the physical motivation behind many of the commonly used SGS closures.

4.1.2. Weighted Essentially Nonoscillatory Schemes

WENO schemes were proposed to remedy the shortcomings of centered schemes. They achieve arbitrarily high order of accuracy while maintaining nearly oscillation-free solutions, even in inviscid flows, by combining multiple interpolation stencils [*Jiang and Shu*, 1996; *Balsara and Shu*, 2000]. WENO reconstructions are computed as convex combinations of the interpolation polynomials from all possible interpolation stencils, with the largest weights placed on the stencils over which the advected field varies most smoothly. In smooth flow regions, the optimally weighted convex combination of all candidate interpolating polynomials of order r achieves $(2r+1)$-th order accurate interpolation. Hence, WENO schemes combine a high order of accuracy with (essentially) oscillation-free stability.

To be more specific, the convex combinations of interpolating polynomials for the positively biased reconstructions of $\tilde{\phi}^{i+1,+}$ and $\tilde{u}_1^{i+\frac{1}{2},+}$ are given by

$$\tilde{\phi}^{i+1,+} = \sum_{k^*=0}^{r-1} w_{k^*}^r q_{k^*}^{r,+} \left(\phi^{i+\frac{3}{2}-r+k^*}, \ldots, \phi^{i+\frac{1}{2}+k^*} \right) \tag{78}$$

and

$$\tilde{u}_1^{i+\frac{1}{2},+} = \sum_{k^*=0}^{r-1} w_{k^*}^r q_{k^*}^{r,+} \left(u_1^{i+1-r+k^*}, \ldots, u_1^{i+k^*} \right). \tag{79}$$

The WENO weights $w_{k^*}^r$ are given by

$$w_{k^*}^r = \frac{\alpha_{k^*}^r}{\alpha_0^r + \ldots + \alpha_{r-1}^r}, \tag{80}$$

where

$$\alpha_{k^*}^r = \frac{C_{k^*}^r}{\left(\varepsilon + \mathcal{S}_{k^*}^r\right)^p}, \tag{81}$$

with optimal weights $C_{k^*}^r$, smoothness measures $\mathcal{S}_{k^*}^r$, regularization parameter $\varepsilon = 10^{-10}$ to prevent division by zero, and exponent $p = 2$ as in *Balsara and Shu* [2000]. The values of $C_{k^*}^r$ and smoothness measures $\mathcal{S}_{k^*}^r$ for third and fifth-order WENO reconstructions are given in *Jiang and Shu* [1996], and those for 7th through 11th-order reconstructions in *Balsara and Shu* [2000].

From the reconstructed scalars $\tilde{\phi}^{i+1,+}$ and $\tilde{\phi}^{i+1,-}$, the upwinded flux is computed by

$$(\rho_0 u_1 \tilde{\phi})^{i+1} = \frac{\rho_0}{2}\left(u_1^{i+1} + |u_1^{i+1}|\right)\tilde{\phi}^{i+1,-} + \frac{\rho_0}{2}\left(u_1^{i+1} - |u_1^{i+1}|\right)\tilde{\phi}^{i+1,+}. \tag{82}$$

To compute the upwinded momentum flux, the advecting velocity must be reconstructed at the flux location. We follow the method described in *Ghosh and Baeder* [2012] and use an $(m-1)$-th order symmetric polynomial interpolation (71) to approximate the advecting velocity $u_1^{i+\frac{1}{2}}$ for a WENO scheme of order m. The upwinded momentum flux is then computed by

$$(\rho_0 \tilde{u}_1 \tilde{u}_1)^{i+\frac{1}{2}} = \frac{\rho_0}{2}\left(\tilde{u}_1^{i+\frac{1}{2}} + |\tilde{u}_1^{i+\frac{1}{2}}|\right)\tilde{u}_1^{i+\frac{1}{2},-} + \frac{\rho_0}{2}\left(\tilde{u}_1^{i+\frac{1}{2}} - |\tilde{u}_1^{i+\frac{1}{2}}|\right)\tilde{u}_1^{i+\frac{1}{2},+}. \tag{83}$$

Although it is unclear whether the resulting WENO schemes are indeed of m-th order, we will still refer to them as m-th order WENO schemes.

Like higher-order centered schemes that discretize the fluxes (61)–(63), WENO schemes conserve first moments of momentum and scalars such as total water and entropy; however, they do not conserve kinetic energy.

4.1.3. SGS Fluxes

The SGS fluxes in the combined fluxes (61)–(63) also require reconstruction to the correct flux locations. As the SGS closures are generally functions of gradients of the velocity and scalar fields, it is natural to compute the eddy viscosity and diffusivity at the grid cell centers. Second-order interpolation is used to interpolate the eddy viscosity and diffusivity when their values are needed on grid cell faces.

4.2. Pressure Solver

PyCLES does not directly solve the Poisson equation (5) for the dynamic pressure perturbation p' that enforces the continuity equation. Rather, it uses a projection method that offers advantages when solving the Poisson equation discretely and that can be derived as follows [e.g., *Chorin*, 1968; *Kim and Moin*, 1985].

Multiplying the momentum equation (2) by ρ_0 and taking the divergence, but not using the continuity equation (4) to eliminate the time derivative, gives

$$\frac{\partial}{\partial t}\frac{\partial}{\partial x_i}\rho_0 u_i = \frac{\partial}{\partial x_i}\rho_0 \dot{\Gamma}_i - \frac{\partial}{\partial x_i}\rho_0 \frac{\partial \alpha_0 p'}{\partial x_i}. \tag{84}$$

Writing the above equation in semidiscrete form and integrating the time derivative from time n to $n+1$ using a forward Euler time step δt^n, we obtain

$$\left(\frac{\partial}{\partial x_i}\rho_0 u_i\right)^{n+1} - \left(\frac{\partial}{\partial x_i}\rho_0 u_i\right)^{n} = \delta t^n \left(\frac{\partial}{\partial x_i}\rho_0 \dot{\Gamma}_i^n - \frac{\partial}{\partial x_i}\rho_0 \frac{\partial \alpha_0 p'}{\partial x_i}\right). \tag{85}$$

Setting the first term on the left-hand side to zero and rearranging gives

$$\left(\frac{\partial}{\partial x_i}\rho_0 \frac{\partial p''}{\partial x_i}\right)^{n} = \frac{\partial}{\partial x_i}\rho_0 \left[(u_i)^n + \dot{\Gamma}_i^n \delta t^n\right], \tag{86}$$

where $p'' = \alpha_0 p' \delta t^n$. This is a Poisson equation for p''. Determining the dynamic pressure perturbation in this way removes any divergent component of the anelastic mass flux $\rho_0 u_i$, which may arise through initial conditions or numerical error. By contrast, using the Poisson equation (5) directly may not remove such divergent components.

In PyCLES, the Poisson equation (86) is solved subject to the boundary conditions of periodicity in the horizontal directions and zero normal pressure gradients at the upper and lower boundaries. The zero normal pressure gradient boundary condition is implicitly applied by enforcing a no-penetration boundary condition on the velocity field at the upper and lower boundaries [e.g., *Kim and Moin*, 1985]. The right-hand side of (86) is discretized using second-order central differences, and the horizontal periodicity of the domain is exploited to take the two-dimensional horizontal Fourier transform of the equation with discretized right-hand side, as is often done in atmospheric LES codes [e.g., *Heus et al.*, 2010; *Sullivan and Patton*, 2011]. The horizontal derivatives of the pressure p'' are evaluated spectrally, using the modified wavenumber representation of a second-order central difference scheme [e.g., *Moin*, 2010]. The vertical derivatives are approximated using a second-order central difference scheme. The resulting tridiagonal system of linear equations for the horizontally Fourier transformed pressure variable $\hat{p}''$ is solved with standard methods [*Press et al.*, 1992]. Inverse Fourier transformation of $\hat{p}''$ yields the desired solution p''. This approach gives an exact solution to the discretized Poisson equation. It requires a fixed, known number of floating point operations, which has advantages over iterative solution methods, whose operation count is problem-dependent and hence a priori unknown.

4.3. Time Stepping

PyCLES uses strong stability preserving Runge-Kutta Schemes (SSP-RK) schemes to integrate the prognostic equations forward in time [*Gottlieb*, 2005]. A predictor-corrector method uses p'' as determined by the Poisson solver to ensure that the mass flux $\rho_0 u_i$ following each Runge-Kutta stage remains approximately divergence free. To simplify the notation in what follows, we will drop inessential grid indices *i*, *j*, and *k*; superscripts *n* again denote the time step.

4.3.1. Second-Order, Two-Stage SSP-RK Scheme

The second-order, two-stage SSP-RK scheme given in *Shu and Osher* [1988] is implemented in PyCLES as

$$\begin{aligned}
\mathcal{T}^* &= \mathbf{T}^n + \delta t^n \frac{d\mathbf{T}^n}{dt} \\
\mathbf{T}^* &= \mathcal{T}^* + \mathbf{P}^* \\
\mathcal{T}^{**} &= \mathbf{T}^* + \delta t^n \frac{d\mathbf{T}^*}{dt} \\
\mathbf{T}^{**} &= \mathcal{T}^{**} + \mathbf{P}^{**} \\
\mathbf{T}^{n+1} &= \frac{1}{2}(\mathbf{T}^n + \mathbf{T}^{**})
\end{aligned} \tag{87}$$

where $\mathcal{T}^*$ and $\mathcal{T}^{**}$ are predicted state vectors in the first and second Runge-Kutta stages, $\mathbf{T}^*$ and $\mathbf{T}^{**}$ are corrected state vectors with divergence-free mass fluxes, and δt^n is the time step at time *n*. The corrector vectors $\mathbf{P}^*$ and $\mathbf{P}^{**}$ are obtained from

$$\mathbf{P}^{i,j,k} = \begin{bmatrix} -\left(\frac{\partial p''}{\partial x_1}\right)^{i,j+\frac{1}{2},k+\frac{1}{2}} \\ -\left(\frac{\partial p''}{\partial x_2}\right)^{i+\frac{1}{2},j,k+\frac{1}{2}} \\ -\left(\frac{\partial p''}{\partial x_3}\right)^{i+\frac{1}{2},j+\frac{1}{2},k} \\ 0 \end{bmatrix}, \tag{88}$$

where p'' is taken at the first Runge-Kutta stage for $\mathbf{P}^*$ and at the second Runge-Kutta stage for $\mathbf{P}^{**}$. The partial derivatives in the corrector vector (88) are computed using second-order central differences.

4.3.2. Third and Fourth-Order SSP-RK Schemes

The predictor-corrector method along with higher-order time stepping schemes can be implemented similarly to the second-order scheme described above. In PyCLES, included as options for higher-order time stepping are the widely used three-stage, third-order SSP-RK scheme known as the *Shu and Osher* [1988] method, and the fourth-order, five-stage SSP-RK scheme described in *Ruuth* [2005].

5. Numerical Tests and Discussion

To illustrate the numerical properties of the transport schemes discussed above, we consider two test cases, which track the evolution of two-dimensional thermodynamic perturbations ("bubble"). They are relevant to the buoyancy-driven flows typically encountered in the atmosphere. For each test case, we perform simulations at a range of resolutions using the following four schemes for transport of momentum and scalars:

1. fourth-order centered scheme
2. sixth-order centered scheme
3. fifth-order WENO scheme
4. 11th-order WENO scheme

All simulations use the third-order SSP-RK scheme for time integration. The time step is dynamically adjusted to ensure that the Courant number remains near 0.3.

5.1. Negatively Buoyant Bubble

The first test case is the benchmark problem proposed by *Straka et al.* [1993]. It involves a two-dimensional negatively buoyant dry bubble, which has been widely used as a test of atmospheric LES codes [e.g.,

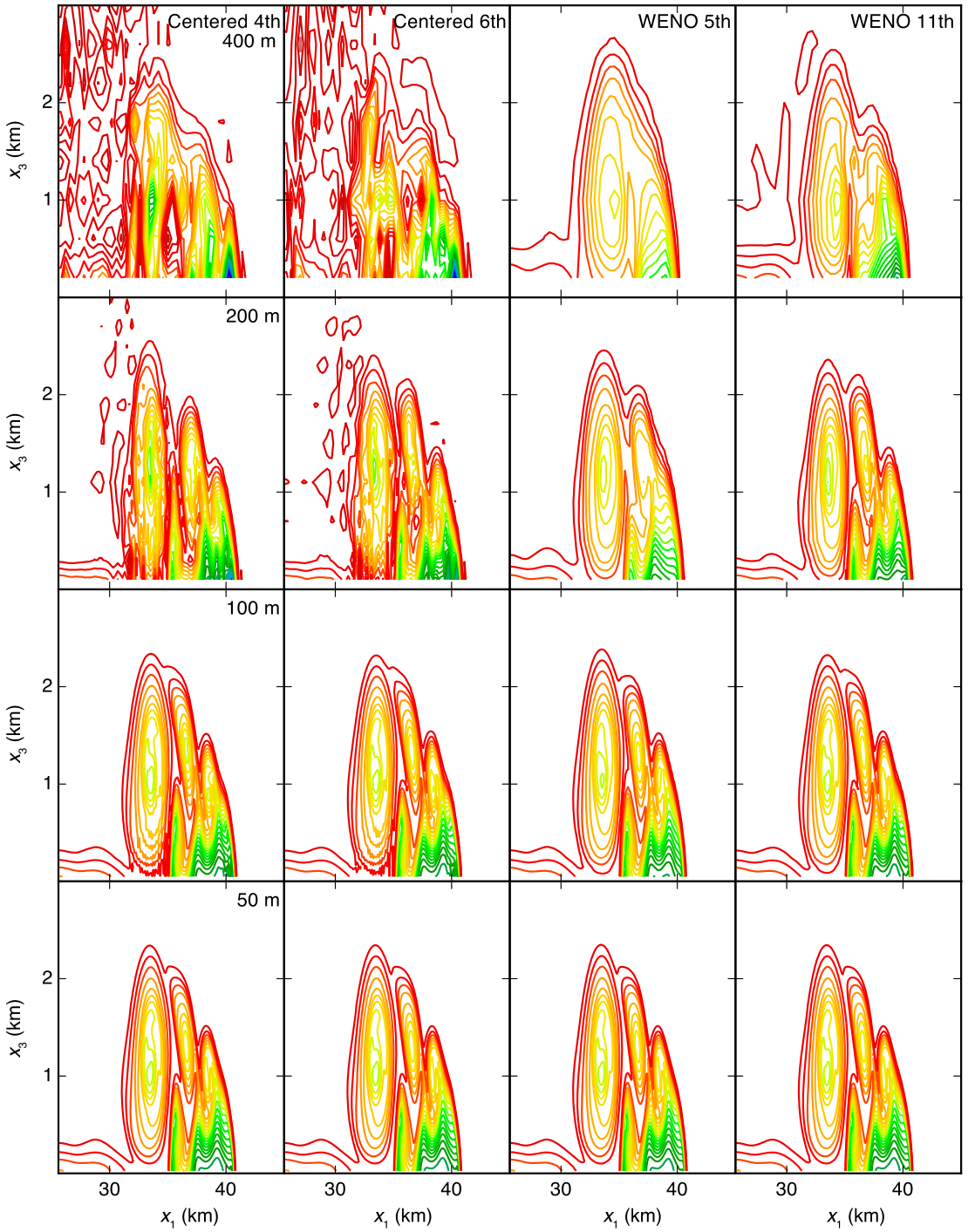

Figure 1. Entropy temperature (47) in simulations of a two-dimensional negatively buoyant dry bubble in a viscous fluid at 900 s. The simulations are performed at four resolutions (rows, resolution increasing from top to bottom) using four different schemes for the transport of scalars and momentum (columns). (The same schemes are used for both scalars and momentum.) The contour interval is 0.5 K.

Skamarock and Klemp, 1993; *Wicker and Skamarock*, 2002; *Ooyama*, 2001; *Schroeder et al.*, 2006]. The flow exhibits dynamic Kelvin-Helmholtz instabilities, which quickly generate small scales. Uniform viscosities and diffusivities of 75 $m^2\ s^{-1}$ are added to ensure that simulations converge numerically at moderate resolution, so that lower-resolution simulations can be compared to a converged high-resolution simulation.

We use the benchmark problem described in *Straka et al.* [1993] with periodic horizontal boundary conditions, rather than the no-flux boundary conditions originally proposed. Nonetheless, the symmetry of the

Table 2. Wall-Clock Times for Integration From 100 s to 200 s of Simulation Time for the Two-Dimensional Negatively Buoyant Dry Bubble Test Case[a]

Resolution	Centered Fourth	Centered Sixth	WENO Fifth	WENO 11th
400 m	0.04	0.05	0.06	0.21
200 m	0.21	0.28	0.37	1.0
100 m	1.59	1.92	2.60	7.22
50 m	10.64	14.91	20.43	54.67

[a]The LES are performed on 16 cores within one compute node, using third-order SSP-RK time stepping with a dynamical time step that maintains a Courant number of approximately 0.3. Times are normalized relative to the simulation with 200 m resolution using the 11th-order WENO scheme.

initial condition and subsequent flow evolution mean that the periodic boundary conditions very well approximate no-flux boundary conditions. The domain is 51.2 km wide (x_1) and 6.4 km high (x_3).

Figure 1 shows the entropy temperature (or potential temperature in this dry case) at $t = 900$ s for resolutions ranging from $(400\,\mathrm{m})^2$ (128×16 grid points) to $(50\,\mathrm{m})^2$ (1024×128 grid points). Because the numerical solutions are nearly horizontally symmetric about the center of the domain, only the right half of the bubble is shown. Based on the figure, we make the following observations:

1. The numerical solutions are essentially converged for all schemes at 50 m resolution.
2. At resolutions coarser than 100 m, the centered schemes produce small-scale oscillations associated with dispersive numerical error. The oscillations adversely affect even large-scale flow features. The dispersive error attenuates when the resolution reaches 100 m or finer. By contrast, the WENO schemes do not exhibit any spurious oscillations, even at coarse resolutions.
3. At resolutions coarser than 100 m, the fifth-order WENO scheme is visibly more dissipative than the 11th-order WENO scheme. Nevertheless, both schemes capture the large-scale flow features well.

In this example, using WENO schemes only for the thermodynamic scalars, but not for momentum, yields similar results (not shown), as the 2D advecting velocity fields are smoother than the scalar fields.

The advantage of using higher-order WENO schemes becomes evident when considering the computational cost of each scheme. The wall-clock times for integration of the test case from 100 s to 200 s are given in Table 2 for each of the configurations shown in Figure 1. The timings shown are for simulations using the third-order Runge-Kutta scheme with time step dynamically adjusted so that the simulations maintain Courant numbers of approximately 0.3, rather than the theoretical maximum Courant number for stability for each of the transport schemes. In practice, we have found that the maximum stable Courant number is relatively insensitive to the order of the transport scheme. The times reported in the table are normalized by the time for the 11th-order WENO scheme at 200 m resolution. The computational cost of higher-order WENO schemes is greater than that of the centered schemes by an $O(1)$ factor that is approximately resolution-independent (e.g., the 11th-order WENO scheme requires about a factor 3.8 more compute time than the sixth-order centered scheme). The parallelization overhead of higher-order WENO schemes is also comparable to that of centered schemes of a similar order. By contrast, the computational cost of all schemes increases roughly like Δ^{-3} in 2D (or like Δ^{-4} in 3-D) as the grid spacing Δ decreases. (There are deviations from this scaling for coarse resolutions, probably because we do not exactly use a 2D domain but our standard code with a 3-D domain with one homogeneous horizontal direction with 6 grid points, corresponding to the maximum number of ghost points required by an 11th-order WENO scheme.) Hence, a WENO scheme at a coarser resolution can be considerably faster than a centered scheme at finer resolution, yet produces solutions of comparable or better quality. For example, for the negatively buoyant bubble in Figure 1, the large-scale flow features are well captured and are free of spurious oscillations at 200 m resolution using the 11th-order WENO scheme. By contrast, the centered schemes remain oscillatory at 200 m resolution, and even still to some degree at 100 m resolution. Yet the 11th-order WENO scheme at 200 m resolution is about a factor 2 faster than a sixth-order centered scheme at 100 m resolution, which produces solutions of comparable or still slightly inferior quality.

5.2. Saturated Buoyant Bubble

The second test case is similar to the benchmark problem proposed by *Bryan and Fritsch* [2002]. It involves the vertical ascent of a two-dimensional buoyant bubble in a saturated moist environment ($q_t \geq q_v^*$), with reversible equilibrium thermodynamics (i.e., without precipitation). *Bryan and Fritsch* [2002] initialize the simulation as a perturbation to a neutrally stratified background state characterized by uniform equivalent

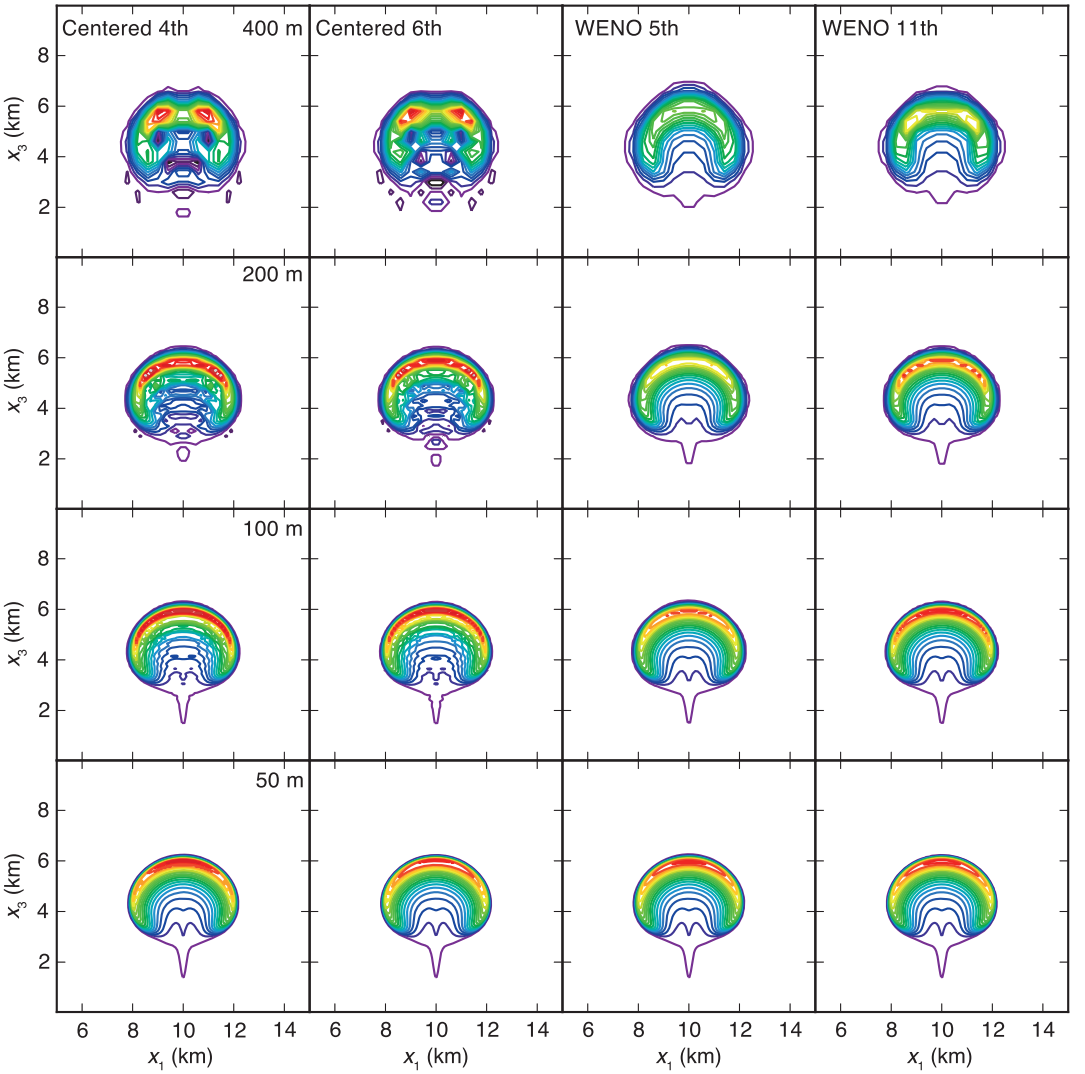

Figure 2. Entropy potential temperature (47) in simulations of a two-dimensional positively buoyant saturated bubble at 1000 s. As in Figure 1, the simulations are performed at four resolutions (rows, resolution increasing from top to bottom) using four different schemes for the transport of scalars and momentum (columns). The contour interval is 0.1 K.

potential temperature. We take a slightly different approach [cf. *Kurowski et al.*, 2014], seeking to ensure that the background has uniform moist specific entropy consistent with the thermodynamic formulation in PyCLES. This is achieved by initializing the bubble as a perturbation to a uniform entropy temperature field, $\theta_s = 320$ K, with uniform total water specific humidity $q_t = 0.0196$ kg/kg. The initial perturbation follows the same functional form of that given in *Bryan and Fritsch* [2002] for their dry case, with a peak amplitude of 2 K. Aside from the use of θ_s in specifying the initial conditions, the simulations are configured to follow *Bryan and Fritsch* [2002], except that we set the horizontal boundary conditions to be periodic. The domain is 20 km wide (x_1) and 10 km high (x_3).

Figures 2 and 3 show the entropy temperature θ_s and fluctuations of liquid-water specific humidity q_l around the horizontal mean for resolutions ranging from $(400\,\mathrm{m})^2$ (50×25 grid points) to $(50\,\mathrm{m})^2$ (400×200 grid points). Several important features are apparent:

1. Simulations using centered schemes are adversely affected by dispersive numerical error at all resolutions.
2. Simulations using 5th and 11th-order WENO schemes are essentially converged at 50 m resolution.
3. Spurious fluctuations in liquid-water specific humidity q_l seen in Figure 3 arise solely from the effects of dispersive numerical errors on the specific entropy s field, because the total-water specific humidity q_t remains constant everywhere.

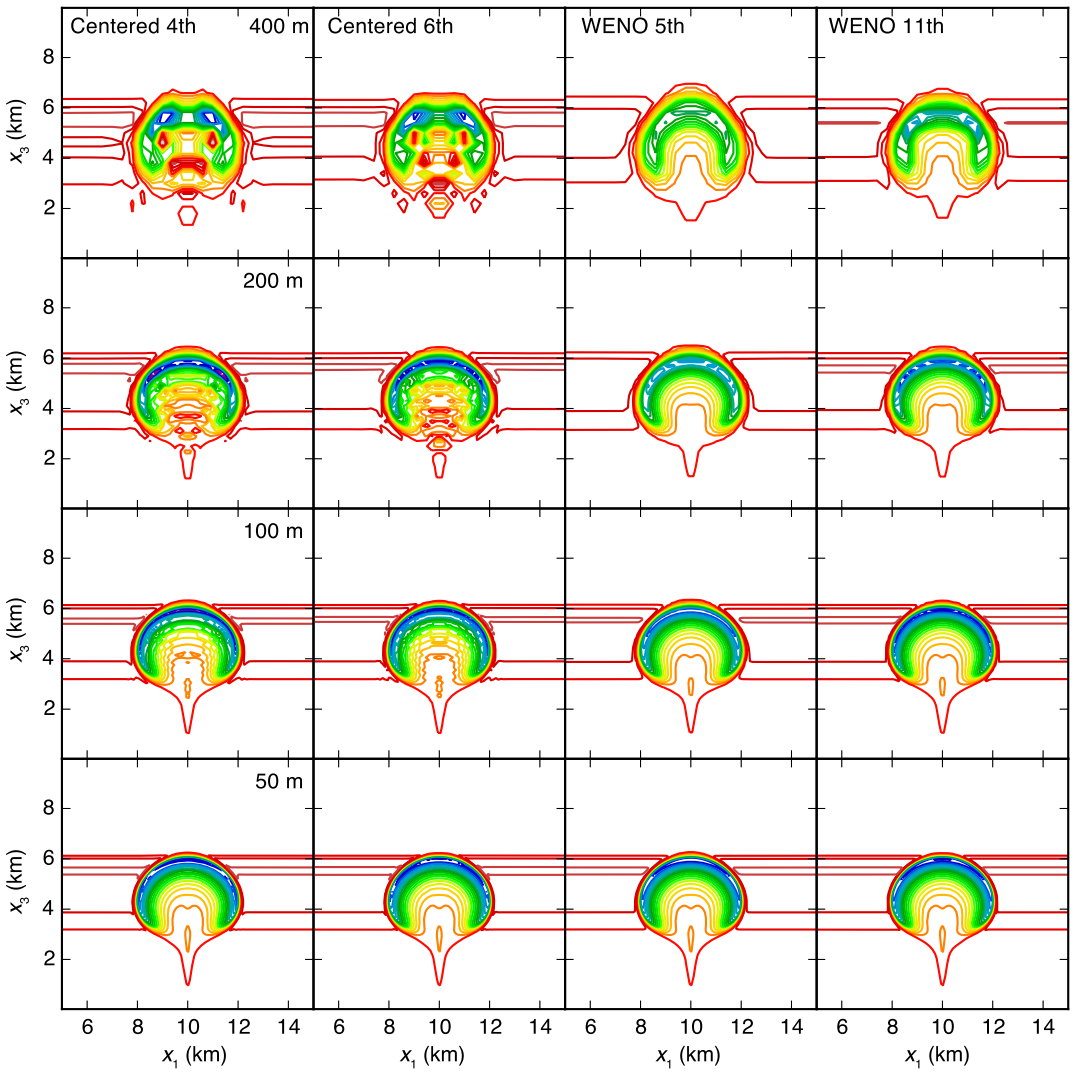

Figure 3. Liquid-water specific humidity q_l in the simulations of a two-dimensional positively buoyant saturated bubble at 1000 s shown in Figure 2. The liquid water specific humidity fluctuations about the mean at each vertical level are plotted, with a contour interval of 2×10^{-5} kg/kg.

As for the dry bubble, the moist bubble test case strongly suggests that WENO schemes can produce simulations at relatively coarse resolution that are comparable with or superior to solutions obtained with centered schemes at finer resolution.

6. Conclusion

We have described a new framework for LES of atmospheric flows. It is the first to use the anelastic equations with total water and moist entropy as prognostic variables, which has numerical advantages especially in the simulation of clouds, because both are conserved in the adiabatic and reversible phase changes of water that frequently occur in clouds. Additionally, both are extensive variables, which makes it possible to obtain closed water and entropy balances for discrete forms of the equations.

The developments of this paper provide several advances:

1. We have extended existing formulations of entropy-based thermodynamics of moist air [e.g., *Ooyama*, 1990, 2001; *Raymond*, 2013] into a coherent framework appropriate for the anelastic equations, with heuristic generalizations that allow a thermodynamically consistent representation of mixed-phase clouds.
2. We have shown how to solve the resulting dynamic and thermodynamic equations numerically, in a way that preserves closed water and entropy balances of the discretized equations and that allows the use of

higher-order WENO schemes on the staggered grids that are commonly used to discretize the anelastic equations.

3. Numerical tests have demonstrated that higher-order WENO schemes yield numerically converged solutions at significantly lower resolution than centered schemes, making WENO schemes attractive even from the point of view of computational cost.

We have implemented the methods described in this paper in code (PyCLES) written in Python and Cython, which is publicly available and makes the LES framework accessible to a wide user group.

Some numerical questions have remained open. For example, it remains to be shown what the exact order of accuracy of a WENO scheme for the momentum equations with m-th order reconstruction of variables on a staggered grid is. This is subject of ongoing research. Tests of PyCLES with realistic atmospheric boundary layers and convective clouds will appear in a forthcoming paper.

Appendix A: Saturation Adjustment Scheme

Given the thermodynamic state variables specific entropy s, total water specific humidity q_t, and reference pressure $p_0(x_3)$, the saturation adjustment scheme calculates the temperature T and specific humidities q_v, q_l, and q_i (with $q_v+q_l+q_i=q_t$) that are consistent with the local thermodynamic equilibrium assumption. When iteration is necessary, it uses the secant method to calculate the temperature and phase partitioning.

Data: (s, q_t, p_0)
Result: (T, q_v, q_l, q_i)
$q_l = 0$;
$q_i = 0$;
$q_v = q_t$;
Compute T_1 by exponentiating (33), given s and $q_t = q_v$;
Look up saturation vapor pressure $p_v^*(T_1)$;
Compute saturation specific humidity $q_{v,1}^* = q_v^*[p_v^*(T_1), q_t, p_0]$ from (37);
if $q_t \leq q_{v,1}^*$ **then**
  $T = T_1$;
  return (T, q_v, q_l, q_i)
else
  $\sigma_1 = q_t - q_{v,1}^*$;
  $q_{l,1} = \lambda(T_1)\sigma_1$;
  $q_{i,1} = [1 - \lambda(T_1)]\sigma_1$;
  Compute s_1 using (33), given T_1, q_t, $q_{v,1}^*$, $q_{l,1}$, $q_{i,1}$;
  $f_1 = s - s_1$;
  $\Delta T = L(T_1)\sigma_1/c_p$ with $c_p = (1 - q_t)c_{pd} + q_{v,1}^* c_{pv}$;
  $T_2 = T_1 + \Delta T$;
  repeat
    Look up saturation vapor pressure $p_v^*(T_2)$;
    Compute saturation specific humidity $q_{v,2}^* = q_v^*[p_v^*(T_2), q_t, p_0]$ from (37);
    $\sigma_2 = q_t - q_{v,2}^*$;
    $q_{l,2} = \lambda(T_2)\sigma_2$;
    $q_{i,2} = [1 - \lambda(T_2)]\sigma_2$;
    Compute s_2 using (33), given T_2, q_t, $q_{v,2}^*$, $q_{l,2}$, $q_{i,2}$;
    $f_2 = s - s_2$;
    $T_n = T_2 - f_2(T_2 - T_1)/(f_2 - f_1)$;
    $T_1 = T_2$;
    $T_2 = T_n$;
    $f_1 = f_2$;
  until $|T_1 - T_2| \leq \delta T$;
  return $\left(T_2, q_{v,2}^*, q_{l,2}, q_{i,2}\right)$
end

Here ΔT is the trial temperature increment that corresponds to warming of the moist air by the release of the latent heat when the initial saturation excess σ_1 condenses/deposits. Given the accuracy of the thermodynamic approximations made, a convergence tolerance of $\delta T \approx 10^{-3}$ K is reasonable. The scheme typically converges within three iterations.

Appendix B: List of Symbols

Unless otherwise noted, subscripts 0 indicate reference state variables, subscripts g indicate surface (ground) values, subscripts b values at the lowest model level, superscript asterisks $(\cdot)^*$ saturation values, and superscript tildes $(\tilde{\cdot})$ constants at thermodynamic standard conditions.

Notation

α	Specific volume.
α_0	Reference state specific volume.
$\gamma_{\phi,i}$	SGS flux of scalar ϕ.
$\gamma_{p,i}$	SGS flux of precipitation specific humidity q_p.
$\gamma_{q,i}$	SGS flux of total water specific humidity q_t.
$\gamma_{s,i}$	SGS flux of specific entropy s.
Γ_i	Velocity tendency without pressure gradients.
δ_{ij}	Kronecker delta.
$\delta(\cdot)$	Dirac delta function.
Δx_i	Grid spacing in x_i direction.
Δ	Grid scale $\Delta=(\Delta x_1 \Delta x_2 \Delta x_3)^{1/3}$.
ε_v	Ratio of gas constants $\varepsilon_v = R_d/R_v$.
ϵ_{ijk}	Antisymmetric Levi-Civita symbol.
θ	Potential temperature.
θ_s	Entropy temperature.
θ_ρ	Density potential temperature.
κ	Adiabatic exponent.
λ	Liquid fraction.
ν_t	Eddy viscosity.
ϕ	Arbitrary scalar field.
$\dot{\Phi}$	Source of scalar ϕ.
ρ	Density.
ρ_0	Reference density.
σ	Saturation excess $\sigma = q_t - q_v^*$.
Σ_i	Specific momentum source.
τ_{ij}	SGS stress.
b	Buoyancy.
c_c	Specific heat of condensate (liquid or ice).
c_k, c_N, c_ε	Coefficients in TKE SGS closure.
c_{pd}	Specific heat of dry air at constant pressure.
c_{pm}	Specific heat of moist air at constant pressure.
c_{pv}	Specific heat of water vapor at constant pressure.
$C^{i,j,k}$	Grid cell centered at $x_1^{i+\frac{1}{2}}$, $x_2^{j+\frac{1}{2}}$, $x_1^{i+\frac{1}{2}}$.
c_S	Smagorinsky coefficient.
C_m	Surface drag coefficient for momentum.
C_q, C_s	Surface exchange coefficients for q_t and s.
D_t	Eddy diffusivity.
e	SGS turbulence kinetic energy.
E	Precipitation evaporation/sublimation rate.
f	Coriolis parameter.
f_B	Buoyancy factor in Smagorinsky-Lilly model.

F_q, F_T	Surface fluxes of latent heat and sensible heat.
g	Gravitational acceleration.
H_p	Source of precipitation q_p.
ℓ	Mixing length in TKE SGS closure.
L	Effective specific latent heat.
L_f, L_s, L_v	Specific latent heats (fusion, sublimation, vaporization).
N	Buoyancy frequency in moist air.
N_{x_i}	Number of grid cells in x_i direction.
p	Pressure.
p'	Dynamic pressure perturbation around p_0.
p_d	Partial pressure of dry air.
p_v	Partial pressure of water vapor.
p_0	Hydrostatic reference pressure.
P	Precipitation formation rate.
Pr_t	Turbulent Prandtl number.
q_c	Specific humidity of condensate (liquid or ice).
q_d	Dry-air mass fraction.
q_l, q_i, q_v	Specific humidities (liquid, ice, vapor).
q_p	Specific humidity of precipitation.
q_t	Total water specific humidity.
Q	Diabatic heating rate.
R_d	Specific gas constant for dry air.
R_v	Specific gas constant for water vapor.
s	Specific entropy of moist air.
s_c	Specific entropy of condensate (liquid or ice).
s_d	Specific entropy of dry air.
s_l, s_i, s_v	Specific entropies (liquid, ice, vapor).
s_p	Specific entropy of precipitation.
s_0	Reference state specific entropy.
S_{ij}	Strain rate of resolved velocities.
$\lvert S\rvert$	Magnitude of strain rate.
$\dot{S}$	Irreversible specific entropy source.
T	Temperature.
T_f	Freezing point temperature.
T_i	Homogeneous ice nucleation temperature.
T_t	Triple point temperature of water.
T_ρ	Density temperature.
T_w	Wet-bulb temperature.
τ_{ij}	SGS stress.
u_i	i-th component of velocity.
$u_{g,i}$	i-th component of geostrophic velocity.
U_b	Wind speed at lowest model level.
w_p	Fall velocity of precipitation.
x_i	Cartesian coordinates.
$X_{i,l}$	Domain lower spatial bound in coordinate x_i.
$X_{i,r}$	Domain upper spatial bound in coordinate x_i.
z_0	Roughness length.
z_f	Freezing level.

Acknowledgments
This work was supported by the U.S. National Science Foundation (grants ARC-1107795 and CCF-1048575), by Caltech's Terrestrial Hazard Observation and Reporting (THOR) Center, and by the Swiss National Science Foundation. We thank João Teixeira for stimulating discussions and Cheikh Mbengue, Bettina Meyer, David Raymond, and Sally Zhang for their help and suggestions during the development of PyCLES. The PyCLES code is freely available at climate-dynamics.org/software.

References

Arakawa, A., and V. R. Lamb (1977), Computational design of the basic dynamical processes of the UCLA general circulation model, *Methods Comput. Phys.*, *17*, 173–265, doi:10.1016/B978-0-12–460817-7.50009-4.

Balsara, D. S., and C.-W. Shu (2000), Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy, *J. Comput. Phys.*, *160*, 405–452, doi:10.1006/jcph.2000.6443.

Bannon, P. R. (1996), On the anelastic approximation for a compressible atmosphere, *J. Atmos. Sci.*, *53*, 3618–3628, doi:10.1175/1520-0469(1996)053<3618:OTAAFA>2.0.CO;2.
Batchelor, G. K. (1953), The conditions for dynamical similarity of motions of a frictionless perfect-gas atmosphere, *Q. J. R. Meteorol. Soc.*, *79*, 224–235, doi:10.1002/qj.49707934004.
Beare, R. J., et al. (2006), An intercomparison of large-eddy simulations of the stable boundary layer, *Boundary Layer Meteorol.*, *118*, 247–272, doi:10.1007/s10546-004-2820-6.
Behnel, S., R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith (2011), Cython: The best of both worlds, *Comput. Sci. Eng.*, *13*, 31–39, doi:10.1109/MCSE.2010.118.
Betts, A. K. (1973), Non-precipitating cumulus convection and its parameterization, *Q. J. R. Meteorol. Soc.*, *99*, 178–196, doi:10.1002/qj.49709941915.
Blossey, P. N., et al. (2013), Marine low cloud sensitivity to an idealized climate change: The CGILS LES intercomparison, *J. Adv. Model. Earth Syst.*, *5*, 234–258, doi:10.1002/jame.20025.
Bony, S., and J.-L. Dufresne (2005), Marine boundary layer clouds at the heart of tropical cloud feedback uncertainties in climate models, *Geophys. Res. Lett.*, *32*, L20806, doi:10.1029/2005GL023851.
Bretherton, C. S., P. N. Blossey, and C. R. Jones (2013), Mechanisms of marine low cloud sensitivity to idealized climate perturbations: A single-LES exploration extending the CGILS cases, *J. Adv. Model. Earth Syst.*, *5*, 316–337, doi:10.1002/jame.20019.
Bryan, G. H., and J. M. Fritsch (2002), A benchmark simulation for moist nonhydrostatic numerical models, *Mon. Weather Rev.*, *130*(12), 2917–2928, doi:10.1175/1520-0493(2002)130<2917:ABSFMN>2.0.CO;2.
Byun, D. W. (1990), On the analytical solutions of flux-profile relationships for the atmospheric surface layer, *J. Appl. Meteorol.*, *29*, 652–657, doi:10.1175/1520-0450(1990)029<0652:OTASOF>2.0.CO;2.
Cess, R. D., et al. (1990), Intercomparison and interpretation of climate feedback processes in 19 atmospheric general circulation models, *J. Geophys. Res.*, *95*, 16,601–16,615, doi:10.1029/JD095iD10p16601.
Cess, R. D., et al. (1996), Cloud feedback in atmospheric general circulation models: An update, *J. Geophys. Res.*, *101*, 12,791–12,794, doi:10.1029/96JD00822.
Chase, M. W., Jr. (1998), NIST-JANAF thermochemical tables, fourth edition, *J. Phys. Chem. Ref. Data, Monogr.*, *9*, 1–1951.
Chorin, A. C. (1968), Numerical solution of the Navier-Stokes equations, *Math. Comput.*, *22*, 745–762.
Chow, F. K., and P. Moin (2003), A further study of numerical errors in large-eddy simulations, *J. Comput. Phys.*, *184*, 366–380, doi:10.1016/S0021–9991(02)00020-7.
Chung, D., G. Matheou, and J. Teixeira (2012), Steady-state large-eddy simulations to study the stratocumulus to shallow cumulus cloud transition, *J. Atmos. Sci.*, *69*, 3264–3276, doi:10.1175/JAS-D-11–0256.1.
Corrsin, S. (1974), Limitations of gradient transport models in random walks and in turbulence, in *Advances in Geophysics*, vol. 18A, pp. 25–60, edited by F. N. Frenkiel and R. E. Munn, Academic, doi:10.1016/S0065-2687(08)60451-3.
Cotton, W. R., G. Bryan, and S. van den Heever (2011), *Storm and Cloud Dynamics*, 2nd ed., Academic, Burlington, Mass.
Deardorff, J. W. (1974a), Three-dimensional numerical study of the height and mean structure of a heated planetary boundary layer, *Boundary Layer Meteorol.*, *7*(1), 81–106, doi:10.1007/BF00224974.
Deardorff, J. W. (1974b), Three-dimensional numerical study of turbulence in an entraining mixed layer, *Boundary Layer Meteorol.*, *7*(2), 199–226, doi:10.1007/BF00227913.
Deardorff, J. W. (1976), Usefulness of liquid-water potential temperature in a shallow-cloud model, *J. Appl. Meteorol.*, *15*, 98–102, doi:10.1175/1520-0450(1976)015<0098:UOLWPT>2.0.CO;2.
Deardorff, J. W. (1980), Stratocumulus-capped mixed layers derived from a three-dimensional model, *Boundary Layer Meteorol.*, *18*(4), 495–527, doi:10.1007/BF00119502.
Duarte, M., A. S. Almgren, K. Balakrishnan, J. B. Bell, and D. M. Romps (2014), A numerical study of methods for moist atmospheric flows: Compressible equations, *Mon. Weather Rev.*, *142*, 4269–4283, doi:10.1175/MWR-D-13–00368.1.
Dufresne, J.-L., and S. Bony (2008), An assessment of the primary sources of spread of global warming estimates from coupled atmosphere–ocean models, *J. Clim.*, *21*, 5135–5144, doi:10.1175/2008JCLI2239.1.
Durran, D. R., and J. B. Klemp (1982), On the effects of moisture on the Brunt-Väisälä frequency, *J. Atmos. Sci.*, *39*, 2152–2158, doi:10.1175/1520-0469(1982)039<2152:OTEOMO>2.0.CO;2.
Dutton, J. A., and G. H. Fichtl (1969), Approximate equations of motion for gases and liquids, *J. Atmos. Sci.*, *26*, 241–254, doi:10.1175/1520-0469(1969)026<0241:AEOMFG>2.0.CO;2.
Emanuel, K. A. (1994), *Atmospheric Convection*, 580 pp., Oxford Univ. Press, N. Y.
Ghosal, S. (1996), An analysis of numerical errors in large-eddy simulations of turbulence, *J. Comput. Phys.*, *125*, 187–206, doi:10.1006/jcph.1996.0088.
Ghosh, D., and J. D. Baeder (2012), High-order accurate incompressible Navier-Stokes algorithm for vortex-ring interactions with solid wall, *AIAA J.*, *50*(11), 2408–2422, doi:10.2514/1.J051537.
Gottlieb, S. (2005), On high order strong stability preserving Runge-Kutta and multi step time discretizations, *J. Sci. Comput.*, *25*(1), 105–128, doi:10.1007/BF02728985.
Grabowski, W. W. (1998), Toward cloud resolving modeling of large-scale tropical circulations: A simple cloud microphysics parameterization, *J. Atmos. Sci.*, *55*(21), 3283–3298, doi:10.1175/1520-0469(1998)055<3283:TCRMOL>2.0.CO;2.
Harlow, F. H., and J. E. Welch (1965), Numerical calculation of timedependent viscous incompressible flow of fluid with free surface, *Phys. Fluids*, *8*(12), 2182–2189, doi:10.1063/1.1761178.
Hauf, T., and H. Höller (1987), Entropy and potential temperature, *J. Atmos. Sci.*, *44*, 2887–2901, doi:10.1175/1520-0469(1987)044<2887:EAPT>2.0.CO;2.
Heus, T., et al. (2010), Formulation of the Dutch Atmospheric Large-Eddy Simulation (DALES) and overview of its applications, *Geosci. Model Dev.*, *3*, 415–444, doi:10.5194/gmd-3–415-2010.
Jiang, G.-S., and C.-W. Shu (1996), Efficient implementation of weighted ENO schemes, *J. Comput. Phys.*, *126*(1), 202–228, doi:10.1006/jcph.1996.0130.
Jiang, H., and W. R. Cotton (2000), Large eddy simulation of shallow cumulus convection during BOMEX: Sensitivity to microphysics and radiation, *J. Atmos. Sci.*, *57*(4), 582–594, doi:10.1175/1520-0469(2000)057<0582:LESOSC>2.0.CO;2.
Khairoutdinov, M., and D. Randall (2006), High-resolution simulation of shallow-to-deep convection transition over land, *J. Atmos. Sci.*, *63*(12), 3421–3436, doi:10.1175/JAS3810.1.
Khairoutdinov, M. F., and D. A. Randall (2003), Cloud resolving modeling of the ARM summer 1997 IOP: Model formulation, results, uncertainties, and sensitivities, *J. Atmos. Sci.*, *60*(4), 607–625, doi:10.1175/1520-0469(2003)060<0607:CRMOTA>2.0.CO;2.

Khairoutdinov, M. F., S. K. Krueger, C.-H. Moeng, P. A. Bogenschutz, and D. A. Randall (2009), Large-eddy simulation of maritime deep tropical convection, *J. Adv. Model. Earth Syst.*, *1*, doi:10.3894/JAMES.2009.1.15.

Kim, J., and P. Moin (1985), Application of a fractional-step method to incompressible Navier-Stokes equations, *J. Comput. Phys.*, *59*(2), 308–323, doi:10.1016/0021–9991(85)90148-2.

Kravchenko, A., and P. Moin (1997), On the effect of numerical errors in large eddy simulations of turbulent flows, *J. Comput. Phys.*, *131*(2), 310–322, doi:10.1006/jcph.1996.5597.

Kurowski, M. J., W. W. Grabowski, and P. K. Smolarkiewicz (2014), Anelastic and compressible simulation of moist deep convection, *J. Atmos. Sci.*, *71*(10), 3767–3787, doi:10.1175/JAS-D-14–0017.1.

Larson, V. E., R. Wood, P. R. Field, J.-C. Golaz, T. H. Vonder Haar, and W. R. Cotton (2001), Small-scale and mesoscale variability of scalars in cloudy boundary layers: One-dimensional probability density functions, *J. Atmos. Sci.*, *58*, 1978–1994, doi:10.1175/1520-0469(2001)058<1978:SSAMVO>2.0.CO;2.

Lemmon, E. W., R. T. Jacobsen, S. G. Penoncello, and D. G. Friend (2000), Thermodynamic properties of air and mixtures of nitrogen, argon, and oxygen from 60 to 2000 K at pressures to 2000 MPa, *J. Phys. Chem. Ref. Data*, *29*, 331–385.

Lilly, D. K. (1962), On the numerical simulation of buoyant convection, *Tellus*, *14*, 148–172, doi:10.1111/j.2153–3490.1962.tb00128.x.

Lin, J. W.-B. (2012), Why Python is the next wave in earth sciences computing, *Bull. Am. Meteorol. Soc.*, *93*, 1823–1824, doi:10.1175/BAMS-D-12-00148.1.

Lipps, F. B., and R. S. Hemler (1982), A scale analysis of deep moist convection and some related numerical-calculations, *J. Atmos. Sci.*, *39*(10), 2192–2210, doi:10.1175/1520-0469(1982)039<2192:ASAODM>2.0.CO;2.

Liu, X.-D., S. Osher, and T. Chan (1994), Weighted eessentially nonoscillatory schemes, *J. Comput. Phys.*, *115*(1), 200–212, doi:10.1006/jcph.1994.1187.

Lord, S. J., H. E. Willoughby, and J. M. Piotrowicz (1984), Role of a parameterized ice-phase microphysics in an axisymmetric, nonhydrostatic tropical cyclone model, *J. Atmos. Sci.*, *41*(19), 2836–2848, doi:10.1175/1520-0469(1984)041<2836:ROAPIP>2.0.CO;2.

Marquet, P. (1993), Exergy in meteorology: Definition and properties of moist available enthalpy, *Q. J. R. Meteorol. Soc.*, *119*, 567–590, doi:10.1002/qj.49711951112.

Marquet, P. (2011), Definition of a moist entropy potential temperature: Application to FIRE-I data flights, *Q. J. R. Meteorol. Soc.*, *137*, 768–791, doi:10.1002/qj.787.

Matheou, G., D. Chung, L. Nuijens, B. Stevens, and J. Teixeira (2011), On the fidelity of large-eddy simulation of shallow precipitating cumulus convection, *Mon. Weather Rev.*, *139*(9), 2918–2939, doi:10.1175/2011MWR3599.1.

Moeng, C.-H., and J. C. Wyngaard (1988), Spectral analysis of large-eddy simulations of the convective boundary layer, *J. Atmos. Sci.*, *45*(23), 3573–3587, doi:10.1175/1520-0469(1988)045<3573:SAOLES>2.0.CO;2.

Moin, P. (2010), *Fundamentals of Engineering Numerical Analysis*, Cambridge Univ. Press, N. Y.

Morinishi, Y., T. S. Lund, O. V. Vasilyev, and P. Moin (1998), Fully conservative higher order finite difference schemes for incompressible flow, *J. Comput. Phys.*, *143*(1), 90–124, doi:10.1006/jcph.1998.5962.

Nance, L. B., and D. R. Durran (1994), A comparison of the accuracy of three anelastic systems and the pseudo-incompressible system, *J. Atmos. Sci.*, *51*, 3549–3565, doi:10.1175/1520-0469(1994)051<3549:ACOTAO>2.0.CO;2.

Ogura, Y., and N. A. Phillips (1962), Scale analysis of deep and shallow convection in the atmosphere, *J. Atmos. Sci.*, *19*, 173–179, doi:10.1175/1520-0469(1962)019<0173:SAODAS>2.0.CO;2.

Ooyama, K. V. (1990), A thermodynamic foundation for modeling the moist atmosphere, *J. Atmos. Sci.*, *47*(21), 2580–2593, doi:10.1175/1520-0469(1990)047<2580:ATFFMT>2.0.CO;2.

Ooyama, K. V. (2001), A dynamic and thermodynamic foundation for modeling the moist atmosphere with parameterized microphysics, *J. Atmos. Sci.*, *58*(15), 2073–2102, doi:10.1175/1520-0469(2001)058<2073:ADATFF>2.0.CO;2.

Pauluis, O. (2008), Thermodynamic consistency of the anelastic approximation for a moist atmosphere, *J. Atmos. Sci.*, *65*(8), 2719–2729, doi:10.1175/2007JAS2475.1.

Pauluis, O., and I. M. Held (2002a), Entropy budget of an atmosphere in radiative-convective equilibrium. Part I: Maximum work and frictional dissipation, *J. Atmos. Sci.*, *59*, 125–139, doi:10.1175/1520-0469(2002)059<0125:EBOAAI>2.0.CO;2.

Pauluis, O., and I. M. Held (2002b), Entropy budget of an atmosphere in radiative-convective equilibrium. Part II: Latent heat transport and moist processes, *J. Atmos. Sci.*, *59*, 140–149, doi:10.1175/1520-0469(2002)059<0125:EBOAAI>2.0.CO;2.

Pauluis, O., V. Balaji, and I. M. Held (2000), Frictional dissipation in a precipitating atmosphere, *J. Atmos. Sci.*, *57*, 989–994, doi:10.1175/1520-0469(2000)057<0989:FDIAPA>2.0.CO;2.

Pope, S. B. (2004), Ten questions concerning the large-eddy simulation of turbulent flows, *New J. Phys.*, *6*(1), 35, doi:10.1088/1367–2630/6/1/035.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (1992), *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed., Cambridge Univ. Press, N. Y.

Raasch, S., and M. Schröter (2001), PALM–A large-eddy simulation model performing on massively parallel computers, *Meteorol. Z.*, *10*(5), 363–372, doi:10.1127/0941–2948/2001/0010–0363.

Raymond, D., and X. Zeng (2005), Modelling tropical atmospheric convection in the context of the weak temperature gradient approximation, *Q. J. R. Meteorol. Soc.*, *131*, 1301–1320, doi:10.1256/qj.03.97.

Raymond, D. J. (2013), Sources and sinks of entropy in the atmosphere, *J. Adv. Model. Earth Syst.*, *5*, 755–763, doi:10.1002/jame.20050.

Romps, D. M., and Z. Kuang (2010), Do undiluted convective plumes exist in the upper tropical troposphere?, *J. Atmos. Sci.*, *67*, 468–484, doi:10.1175/2009JAS3184.1.

Rossum, G. (1995), Python reference manual, technical report, Cent. for Math. and Comput. Sci., Amsterdam. Available at: http://oai.cwi.nl/oai/asset/5008/05008D.pdf.

Ruuth, S. J. (2005), Global optimization of explicit strong-stability-preserving Runge-Kutta methods, *Math. Comput.*, *75*(253), 183–207.

Satoh, M., T. Matsuno, H. Tomita, H. Miura, T. Nasuno, and S. Iga (2008), Nonhydrostatic icosahedral atmospheric model (NICAM) for global cloud resolving simulations, *J. Comput. Phys.*, *227*, 3486–3514, doi:10.1016/j.jcp.2007.02.006.

Savic-Jovcic, V., and B. Stevens (2008), The structure and mesoscale organization of precipitating stratocumulus, *J. Atmos. Sci.*, *65*(5), 1587–1605, doi:10.1175/2007JAS2456.1.

Savre, J., A. M. L. Ekman, and G. Svensson (2014), Technical note: Introduction to MIMICA, a large-eddy simulation solver for cloudy planetary boundary layers, *J. Adv. Model. Earth Syst.*, *6*, 630–649, doi:10.1002/2013MS000292.

Schmidt, H., and U. Schumann (1989), Coherent structure of the convective boundary-layer derived from large-eddy simulations, *J. Fluid Mech.*, *200*, 511–562, doi:10.1017/S0022112089000753.

Schroeder, G., K. H. Schlünzen, and F. Schimmel (2006), Use of (weighted) essentially non-oscillatory advection schemes in a mesoscale model, *Q. J. R. Meteorol. Soc.*, *132*, 1509–1526, doi:10.1256/qj.04.191.

Shu, C.-W. (1998), Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes Math.*, vol. 1697, edited by A. Quarteroni, pp. 325–432, Springer, Berlin, doi:10.1007/BFb0096355.

Shu, C.-W., and S. Osher (1988), Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.*, *77*(2), 439–471, doi:10.1016/0021–9991(88)90177-5.

Siebesma, A. P., et al. (2003), A large eddy simulation intercomparison study of shallow cumulus convection, *J. Atmos. Sci.*, *60*(10), 1201–1219, doi:10.1175/1520-0469(2003)60<201:ALESIS>2.0.CO;2.

Skamarock, W. C., and J. B. Klemp (1993), Adaptive grid refinement for two-dimensional and three-dimensional nonhydrostatic atmospheric flow, *Mon. Weather Rev.*, *121*(3), 788–804, doi:10.1175/1520-0493(1993)121<0788:AGRFTD>2.0.CO;2.

Skamarock, W. C., and J. B. Klemp (2008), A time-split nonhydrostatic atmospheric model for weather research and forecasting applications, *J. Comput. Phys.*, *227*(7), 3465–3485, doi:10.1016/j.jcp.2007.01.037.

Smagorinsky, J. (1958), On the numerical integration of the primitive equations of motion for baroclinic flow in a closed region, *Mon. Weather Rev.*, *86*(12), 457–466, doi:10.1175/1520-0493(1958)086<0457:OTNIOT>2.0.CO;2.

Smagorinsky, J. (1963), General circulation experiments with the primitive equations, *Mon. Weather Rev.*, *91*(3), 99–164, doi:10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2.

Sommeria, G., and J. W. Deardorff (1977), Subgrid-scale condensation in models of nonprecipitating clouds, *J. Atmos. Sci.*, *34*(2), 344–355, doi:10.1175/1520-0469(1977)034<0344:SSCIMO>2.0.CO;2.

Squires, P., and J. S. Turner (1962), An entraining jet model for cumulo-nimbus updraughts, *Tellus*, *14*(4), 422–434, doi:10.1111/j.2153–3490.1962.tb01355.x.

Stevens, B., et al. (2005), Evaluation of large-eddy simulations via observations of nocturnal marine stratocumulus, *Mon. Weather Rev.*, *133*, 1443–1462, doi:10.1175/MWR2930.1.

Straka, J. M., R. B. Wilhelmson, L. J. Wicker, J. R. Anderson, and K. K. Droegemeier (1993), Numerical solutions of a non-linear density current: A benchmark solution and comparisons, *Int. J. Numer. Methods Fluids*, *17*(1), 1–22, doi:10.1002/fld.1650170103.

Sullivan, P. P., and E. G. Patton (2011), The effect of mesh resolution on convective boundary layer statistics and structures generated by large-eddy simulation, *J. Atmos. Sci.*, *68*(10), 2395–2415, doi:10.1175/JAS-D-10–05010.1.

Tao, W.-K., J. Simpson, and M. McCumber (1989), An ice-water saturation adjustment, *Mon. Weather Rev.*, *117*(1), 231–235, doi:10.1175/1520-0493(1989)117<0231:AIWSA>2.0.CO;2.

Tompkins, A. M. (2002), A prognostic parameterization for the subgrid-scale variability of water vapor and clouds in large-scale models and its use to diagnose cloud cover, *J. Atmos. Sci.*, *59*, 1917–1942, doi:10.1175/1520-0469(2002)059<1917:APPFTS>2.0.CO;2.

Turk, M. J., B. D. Smith, J. S. Oishi, S. Skory, S. W. Skillman, T. Abel, and M. L. Norman. (2011), yt: A Multi-code analysis toolkit for astrophysical simulation data, *Astrophys. J. Suppl. Ser.*, *192*(1), 1–16, doi:10.1088/0067-0049/192/1/9.

Tripoli, G. J., and W. R. Cotton (1981), The use of ice-liquid water potential temperature as a thermodynamic variable in deep atmospheric models, *Mon. Weather Rev.*, *109*(5), 1094–1102, doi:10.1175/1520-0493(1981)109<1094:TUOLLW>2.0.CO;2.

Vial, J., J.-L. Dufresne, and S. Bony (2013), On the interpretation of inter-model spread in CMIP5 climate sensitivity estimates, *Clim. Dyn.*, *41*(11), 3339–3362, doi:10.1007/s00382-013-1725-9.

Webb, M. J., et al. (2006), On the contribution of local feedback mechanisms to the range of climate sensitivity in two GCM ensembles, *Clim. Dyn.*, *27*(1), 17–38, doi:10.1007/s00382-006-0111-2.

Wicker, L. J., and W. C. Skamarock (2002), Time-splitting methods for elastic models using forward time schemes, *Mon. Weather Rev.*, *130*(8), 2088–2097, doi:10.1175/1520-0493(2002)130<2088:TSMFEM>2.0.CO;2.

Wilbers, I. M., H. P. Langtangen, and Å. Ødegård (2009), Using Cython to speed up numerical Python programs, *Proc. MekIT 09*, 495–512.

Zeng, X., W.-K. Tao, and J. Simpson (2005), An equation for moist entropy in a precipitating and icy atmosphere, *J. Atmos. Sci.*, *62*(12), 4293–4309, doi:10.1175/JAS3570.1.

Zeng, X., W.-K. Tao, and J. Simpson (2008), A set of prognostic variables for long-term cloud- resolving model simulations, *J. Meteorol. Soc. Jpn.*, *86*, 839–856.