



RESEARCH ARTICLE

10.1029/2025MS005293

Key Points:

- Numerical methods are presented for moist atmospheric dynamics and accounting for non-smoothness at cloud edge
- Approximately second-order numerical accuracy is seen in root-mean-square (L^2) error, even with cloud formation
- Degraded accuracy is seen in maximum (L^∞) error, due to clouds

Correspondence to:

S. N. Stechmann,
stechmann@wisc.edu

Citation:

Marsico, D. H., Du, S., & Stechmann, S. N. (2025). Can second-order numerical accuracy be achieved for moist atmospheric dynamics with non-smoothness at cloud edge? *Journal of Advances in Modeling Earth Systems*, 17, e2025MS005293. <https://doi.org/10.1029/2025MS005293>

Received 18 JUN 2025

Accepted 15 SEP 2025

Can Second-Order Numerical Accuracy Be Achieved for Moist Atmospheric Dynamics With Non-Smoothness at Cloud Edge?

David H. Marsico^{1,2} , Shukai Du³, and Samuel N. Stechmann^{4,5} 

¹Cooperative Institute for Research in Environmental Sciences, University of Colorado Boulder, Boulder, CO, USA,

²Physical Sciences Laboratory, National Oceanic and Atmospheric Administration, Boulder, CO, USA, ³Department of Mathematics, Syracuse University, Syracuse, NY, USA, ⁴Department of Mathematics, University of Wisconsin–Madison, Madison, WI, USA, ⁵Department of Atmospheric and Oceanic Sciences, University of Wisconsin–Madison, Madison, WI, USA

Abstract Non-smoothness arises at cloud edge because, in moist thermodynamics, the thermodynamic properties of the atmosphere are different inside a cloud versus in clear air. In particular, inside a cloud, the vapor pressure of water is constrained by the saturation vapor pressure, which acts as a threshold. Due to this threshold, while the water vapor mixing ratio may vary continuously across cloud edge, its derivatives are not necessarily continuous at cloud edge. Similarly, non-smoothness also arises for buoyancy and other variables. Consequently, this non-smoothness in buoyancy and other variables can cause a degraded accuracy in computational simulations. Here we consider special treatment of numerical methods for the interface that arises from phase changes and cloud edges, in order to enhance the accuracy and potentially achieve second-order accuracy. Numerical solutions are computed for the moist non-precipitating Boussinesq equations as an idealized cloud-resolving model with phase changes of water, that is, with cloud formation. Convergence tests, both spatial and temporal, are conducted to measure the numerical error as the grid spacing and time step are refined. While approximately second-order accuracy is seen in root-mean-square (L^2) error, the accuracy is degraded in the maximum (L^∞) error. Discussion is also included on theoretical issues and potential implications for numerical simulations.

Plain Language Summary An important issue that arises from clouds is the difference in atmospheric properties inside the cloud versus outside the cloud. Inside a cloud, the vapor pressure of water is constrained by the saturation vapor pressure, which acts as a threshold. Due to this threshold, while water vapor, temperature, and many other common thermodynamic variables may vary continuously across cloud edge, their derivatives are not necessarily continuous. This lack of smoothness poses challenges for numerical approximations on computers. For instance, due to cloud edge, traditional numerical methods used for cloud-resolving models may suffer a loss of accuracy. In this paper, we study the accuracy degradation that arises from the presence of clouds, and consider whether it is possible to achieve second-order accuracy for numerical solutions in the presence of clouds and phase changes, where for instance, water vapor changes to cloud water.

1. Introduction

In the thermodynamics of a moist atmosphere, non-smoothness arises due to phase changes of water, that is, due to clouds. The fundamental source of the non-smoothness is a threshold: the water vapor mixing ratio q_v does not exceed the saturation value, q_{vs} . While q_v is continuous, its derivative is not continuous; that is, q_v is not smooth. (For discussions of supersaturation and notation for non-smoothness, see later paragraphs in Section 1.) The non-smoothness manifests itself in a variety of other ways as well. Three examples are as follows. (a) The buoyancy is continuous but its derivative is not continuous (see Equation 7 of Stevens (2005)). (b) The equation of state has a different form within clouds versus outside of clouds (see Equations 3.18 to 3.21 of Ooyama (1990)). (c) The lapse rate is different for unsaturated versus saturated parcels (see Chapter 4 of Emanuel (1994), and Equations 1 and 2 of Pauluis (2008)).

Due to non-smoothness, certain numerical issues can arise in computational simulations. In terms of accuracy, traditional second-order (or higher-order) finite difference methods used for spatial discretizations can have less accuracy than expected. This is because Taylor series, on which finite difference schemes are based, are only valid

for smooth functions, and do not directly apply to discretizations of spatial derivatives near interfaces (LeVeque & Li, 1994; Tzou & Stechmann, 2019). (Similar issues exist for other numerical schemes such as spectral methods, finite volume methods, and finite element methods; see further discussion at the end of Section 2.2.) Time discretization is also an important consideration in the presence of phase changes. Time integration schemes such as Runge-Kutta methods also depend on the continuity and differentiability of the underlying field that is being integrated forward in time. Because different regions of space may change phase over the course of a time step, that is, a region of the atmosphere changes from clear to cloudy, one may no longer expect to observe a temporal order of accuracy as high as a formal Taylor series analysis would suggest.

In this paper, we investigate numerical methods that account for the interfaces that arise due to phase changes and cloud edge. By introducing numerical corrections near the interface, it is possible that the accuracy degradation due to cloud edge could be alleviated, and it may be possible to achieve second order accuracy. To the author's knowledge, interface corrections for second-order accuracy have not previously been considered in applications to clouds in moist atmospheric dynamics.

Interfaces also arise in other scientific contexts, such as two-phase flow, and numerical methods have been investigated with special treatment at the interface (De Sousa et al., 2004; Hellrung Jr et al., 2012; LeVeque & Li, 1994; Oevermann et al., 2009; Peskin, 2002; Sussman et al., 1998, 2007; Tzou & Stechmann, 2019; Walsh, 2016). Many of these concepts may be applicable for clouds as well, although some differences in the setup exist. For instance, in two-phase fluid flow or multi-fluid flow, a common scenario is a large contrast in fluid density between an air bubble and its surrounding liquid environment, whereas no such large contrast in fluid density occurs between a cloud and its surrounding environment. In considering numerical methods for any of these interface problems, a variety of factors can be considered. The focus of the present paper is on second-order accuracy. For second-order accuracy, one approach has been the immersed interface method (Lee & LeVeque, 2003; Li & Lai, 2001; Xu & Wang, 2006a, 2006b), although the early versions of these methods involve derivations that can be complex due to the need for spatial and temporal jump conditions and derivatives of jump conditions, and can be complex to implement, and hence other approaches have been pursued that may be simpler to derive and implement. Here we will build on the approach of Tzou and Stechmann (2019), which brings properties such as symmetric discretized operators, and was designed for ease of implementation on top of existing solvers.

Second-order or higher-order accuracy is a valuable aim for a variety of reasons. The most readily apparent reason is that one would naturally desire that the numerical error would decrease rapidly as the grid spacing and time step are refined. To achieve higher order accuracy, though, one must commonly use a more complicated numerical method with a greater computational expense, and a balance between accuracy and complexity and expense should be considered. Also, one may question whether order of accuracy, as it is commonly described in terms of smooth solutions, is an important factor for turbulent fluids. Other aspects of second-order and higher-order methods are also important, though, beyond the most readily apparent considerations of errors and accuracy. For example, in terms of stability, the third-order Runge-Kutta schemes offer improved stability properties for oscillatory or wave solutions, in comparison to second-order Runge-Kutta schemes, and hence third-order schemes are sometimes chosen instead of second-order schemes on the basis of stability properties rather than accuracy as a primary consideration. Similarly, for the spatial discretization of advection, a different order of accuracy can be associated with differences in inherent numerical effects of a different nature (e.g., dissipative vs. dispersive). See, for instance, Durran (1999), Chapter 2; LeVeque (2007), Chapter 7 and Section 10.9; and Warming and Hyett (1974). Finally, we seek to understand second order accuracy because it is a precise property of a numerical method, and it provides an important test of our understanding of numerical methods for moist atmospheric dynamics.

We are motivated by cloud-resolving models (CRMs) and large eddy simulation (LES) (and their use for numerical weather prediction and forecasting and for multi-scale superparameterization versions of global climate models (GCMs) and Earth system models (ESMs)). We would like to simultaneously (a) include the essential aspects of clouds in CRMs and (b) also allow greater transparency into the non-smoothness associated with clouds. To do so, we will use an idealized version of a CRM in the form of a moist, non-precipitating Boussinesq atmosphere. This type of setup, and variations of it, have also been valuable in many past studies (Bretherton, 1987; Chien et al., 2022; Cuijpers & Duynkerke, 1993; Hernandez-Duenas et al., 2013; Kuo, 1961; Pauluis & Schumacher, 2010, 2011; Vallis et al., 2019). The results here with the idealized CRM should carry over to comprehensive CRMs since they share the same type of non-smoothness due to phase changes and clouds.

Also note that, in this paper, we are assuming that supersaturation is zero (i.e., in a cloud, the water vapor is exactly at saturation), in which case the thermodynamics has an inherent non-smoothness. It is possible that introducing supersaturation could mitigate the non-smoothness at cloud edge, if the small scales of supersaturation could be numerically resolved. However, the typical scales of supersaturation are small. Hence, at typical resolutions, the supersaturation is approximately zero, and an approximate non-smoothness is present, and numerical treatment of these issues can be needed. In other words, regardless of whether a discontinuity is inherent (i.e., present in the nondiscretized equations) or apparent (i.e., due to low resolutions in the discretization), accurate and robust numerical methods are needed.

Regarding definitions and notation, non-smoothness is often characterized by the notation C^k , which indicates that a function and its first k derivatives ($\partial/\partial x, \partial^2/\partial x^2, \dots, \partial^k/\partial x^k$) are continuous. If a function is in the function space C^∞ , with $k = \infty$, it is said to be smooth. On the other hand, if a function is in C^k but not C^{k+1} for some finite k , then it is non-smooth.

We also mention a related setting which is not the focus here but which could be investigated in the future: the setting of coarse-resolution GCMs and ESMs that employ parameterizations of moist convection and clouds (Arakawa, 2004; Lauritzen et al., 2022; Wang et al., 2021). In such a large-scale setting, individual clouds are not resolved, so there is no cloud edge per se. Nevertheless, an interface can arise as the evolving boundary between convective and non-convecting regions, as a result of thresholds or triggers in convective parameterizations. Such an interface can potentially be addressed using the concepts presented herein. We note, though, that the type of non-smoothness may differ; and even among different convective/cloud parameterizations, one may have different types of non-smoothness. For example, in idealized versions of convective adjustment schemes, the solutions could have vertical velocity that could have different types of non-smoothness, where they could be either continuous or discontinuous, depending on whether the convective adjustment time is finite or instantaneous, respectively (Frierson et al., 2004; Stechmann & Majda, 2006).

From a broad perspective, the work herein is an addition to the literature on numerical aspects of moist atmospheric dynamics. Past studies have investigated topics such as time stepping (Barrett et al., 2019; Vogl et al., 2020; Wan et al., 2020) and spurious cloud-edge supersaturation in numerical solutions (Grabowski & Morrison, 2008; Grabowski & Smolarkiewicz, 1990; Margolin et al., 1997; Stevens et al., 1996) among a variety of aspects of the numerical treatment of moist dynamical models (Bryan & Fritsch, 2002; Bendall et al., 2020; Chertock et al., 2019; Doppler et al., 2024; Grabowski & Smolarkiewicz, 1996; Gaberšek et al., 2012; Khouider & Majda, 2005a, 2005b; Norman et al., 2023; Pauluis & Garner, 2006; Tissaoui et al., 2023; Walsh, 2016, 2021; Zarzycki et al., 2019; S. Zhang et al., 2023). Herein, we add to this literature by investigating the accuracy degradation that arises from a lack of smoothness at cloud edge, and whether interface correction terms can be applied to restore second-order accuracy.

In what follows, we consider the impact that cloud edge and phase changes have on the different parts of a numerical model. Section 2 describes the model equations and a motivating discussion. In subsequent sections, we consider different components of a numerical model of the atmosphere, such as the time integration scheme (Section 3), the buoyancy, and the spatial discretization of the Poisson equation and pressure gradients (Sections 2.2 and 4). In cases where clouds cause a degradation of error for standard discretizations, we derive corrected discretizations, and demonstrate that these corrections lead to a clear increase in order of accuracy. We then consider in Section 5 the combined effect of these individual components on the full moist dynamical system, and discuss theoretical issues and potential implications for numerical simulations. A concluding discussion is in Section 6.

2. Moist Atmospheric Dynamics With Clouds and Phase Changes

In Section 2.1 we describe the evolution equations used here, and in Section 2.2 we describe an example component (the pressure Poisson equation) as motivation for a careful treatment of numerical methods near cloud edge.

2.1. Evolution Equations

As mentioned in the Introduction section, the moist Boussinesq equations with phase changes are used here as an idealized version of a CRM. They share the same type of non-smoothness (due to phase changes and clouds) as comprehensive CRMs. A valuable aspect of the idealized CRM is that the form of the non-smoothness is more transparent, since exact expressions can be written down for transformations between thermodynamic variables.

Our presentation of the thermodynamics and dynamics will follow Marsico et al. (2019) and Hernandez-Duenas et al. (2013). We assume that moisture is in the form of either cloud water or water vapor. Regions where cloud water is present are referred to as saturated, or cloudy, regions, and regions without cloud water are referred to as unsaturated, or clear, regions. For a non-precipitating atmosphere, the partial differential equations (PDEs) take the form.

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = b \hat{\mathbf{z}} + \nu_u \Delta \mathbf{u}, \quad (1)$$

$$\frac{\partial \theta_e}{\partial t} + \mathbf{u} \cdot \nabla \theta_e + \frac{d\tilde{\theta}_e}{dz} w = \nu_\theta \Delta \theta_e, \quad (2)$$

$$\frac{\partial q_t}{\partial t} + \mathbf{u} \cdot \nabla q_t + \frac{d\tilde{q}_t}{dz} w = \nu_q \Delta q_t, \quad (3)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (4)$$

where $\mathbf{u}(\mathbf{x}, t)$ is the velocity vector, p is the pressure (scaled by a constant background density, ρ_0), $\hat{\mathbf{z}}$ is the unit vector in the vertical direction, and b is the buoyancy. Here p is the deviation from a hydrostatically balanced background state. The (anomalous) equivalent potential temperature is θ_e , and the (anomalous) total water mixing ratio is q_t . Note that the subscript t in q_t stands for “total” and should not be confused with a time derivative. The equivalent potential temperature and total water have been decomposed into background and anomalous/fluctuating parts as $\tilde{\theta}_e(z) + \theta_e(\mathbf{x}, t)$ and $\tilde{q}_t(z) + q_t(\mathbf{x}, t)$, respectively. The background profiles $\tilde{\theta}_e(z)$ and $\tilde{q}_t(z)$ are assumed to be known, whereas the anomalous/fluctuating components are unknown prognostic variables. For the buoyancy we define

$$b = g \left(\frac{\theta}{\theta_0} + R_{vd} q_v - q_l \right), \quad (5)$$

which is a commonly used definition of buoyancy in cloud-resolving models (Emanuel, 1994; Grabowski & Smolarkiewicz, 1996; Khairoutdinov & Randall, 2003). Here $\theta_0 = 300$ K is a constant background potential temperature, $g = 9.8 \text{ m s}^{-2}$ is the gravitational acceleration, and $R_{vd} = (R_v/R_d) - 1 = 0.61$, where R_v and R_d are the gas constants of water vapor and dry air. In order to evaluate (5), the cloud liquid water mixing ratio q_l and water vapor mixing ratio q_v are diagnosed from q_t . The liquid water is defined as the excess water above the saturation mixing ratio q_{vs} , so that

$$q_l = \max(0, q_t - q_{vs}), \quad (6)$$

$$q_v = \min(q_t, q_{vs}). \quad (7)$$

The potential temperature, θ , is also needed to evaluate (5), and in the setup here θ is related to the equivalent potential temperature, θ_e , by the equation

$$\theta_e = \theta + \frac{L_v}{c_p} q_v. \quad (8)$$

In Equations 6 and 7, $q_{vs}(z)$ is a height dependent saturation mixing ratio. Such a simplified q_{vs} is helpful because it eliminates the need for solving implicit nonlinear algebraic equations for diagnosing (q_v, q_l, θ) from (θ_e, q_t) . Consequently, it allows for the buoyancy in Equation 5 to be expressed analytically in piecewise form as

$$b = g \begin{cases} \frac{\theta_e}{\theta_0} + \left(R_{vd} - \frac{L_v}{c_p \theta_0} \right) q_t & \text{if } q_t < q_{vs}(z), \\ \frac{\theta_e}{\theta_0} + \left(R_{vd} - \frac{L_v}{c_p \theta_0} + 1 \right) q_{vs} - q_t & \text{if } q_t \geq q_{vs}(z). \end{cases} \quad (9)$$

The piecewise buoyancy in (9), along with (1)–(4), completes the specification of the model.

Note the transparent form of the non-smoothness in buoyancy in (9). It follows from the original form of the buoyancy in (5) after changing to the conserved thermodynamic variables, θ_e and q_t . In more comprehensive versions of CRMs and LES, the same type of non-smoothness in buoyancy is present, although it is less transparent. In CRMs and LES, for diagnosing (q_v, q_l, θ) from (θ_e, q_t) or related thermodynamic transformations, one obtains implicit nonlinear algebraic equations that are commonly solved via iterative methods such as Newton's method, in a procedure that is called “saturation adjustment.” For additional information, see Appendix A, section c, of Khairoutdinov and Randall (2003) or Section 2.7 of Sridhar et al. (2022). Here, in contrast, owing to the simplified thermodynamics used here, the iterative procedure is not needed, and, instead, the buoyancy can be expressed analytically as shown in (9).

As mentioned in the introduction, this type of idealized CRM setup, and variations of it, have also been valuable in many past studies (Bretherton, 1987; Chien et al., 2022; Cuijpers & Duynkerke, 1993; Hernandez-Duenas et al., 2013; Kuo, 1961; Pauluis & Schumacher, 2010, 2011; Vallis et al., 2019). This system is valuable here because it includes clouds, phase changes, and cloud latent heating, with few additional complexities; hence the non-smoothness at cloud edge can be isolated as the main challenge. Also, this moist non-precipitating Bousinesq system forms the foundation for cases with more complexity (with precipitation, anelastic approximation, ice microphysics, etc.).

The overall numerical strategy follows the second-order projection method (Bell et al., 1989; Bell & Marcus, 1992) which builds on the original first-order projection method (Chorin, 1968; Temam, 1969). Eddy viscosity and eddy diffusion coefficients (ν_u, ν_θ, ν_q) are small and are included in (1)–(3). Special considerations for phase changes and cloud edge will be investigated here and are described below.

2.2. Implications of Phase Changes for Numerical Accuracy: A First Look

The buoyancy defined in Equation 9 is continuous, but not necessarily differentiable (Marsico et al., 2019; Stevens, 2005). This lack of smoothness has potentially important consequences for numerical methods that are based on Taylor series approximations, which assume a certain amount of smoothness. (Similar issues arise for other numerical methods such as spectral methods, finite volume methods, and finite element methods; see further discussion at the end of Section 2.2.) In this paper, a main goal is to investigate the various ways that the non-smooth buoyancy could degrade the accuracy of a numerical solution, as the buoyancy provides the most direct coupling between moisture and dynamics.

In this subsection, we describe a first motivating example of how the non-smooth buoyancy will degrade the accuracy of the solution. In particular, we consider the pressure Poisson equation. This equation is obtained by taking the divergence of the momentum equations, and takes the form:

$$\nabla^2 p = -\nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) + \nabla \cdot (\nu_u \Delta \mathbf{u}) + \frac{\partial b}{\partial z}. \quad (10)$$

As the Laplacian is a linear operator, we can focus on each component on the right hand side of Equation 10 separately. As we are primarily concerned with the impact of buoyancy, our focus will be on the equation

$$\nabla^2 p = \frac{\partial b}{\partial z}. \quad (11)$$

We can choose θ_e and q_t to be independent of x , and such that the buoyancy becomes a piecewise linear function of z alone. Equation 11 then simplifies to

$$\frac{\partial^2 p}{\partial z^2} = \frac{\partial b}{\partial z}. \quad (12)$$

In what follows, we provide a brief derivation of the standard second-order finite difference approximation of Equation 12. The purpose of this is to demonstrate where the assumptions of smoothness enter into a formal

Taylor series argument for high order accuracy, and to explain why the level of smoothness needed on the buoyancy is higher than what actually exists in cloud resolving models.

To see these effects, the derivation is as follows. Assume that solutions of Equation 12 are approximated on a uniformly spaced grid with grid spacing h and gridpoints $z_i = ih$. To derive a second order accurate approximation to (12) at z_i , we Taylor expand p about z_i to obtain expressions for z_{i+1} and z_{i-1} . Assuming the first three derivatives of p exist and are continuous on the interval $[z_{i-1}, z_{i+1}]$, and using p_i to denote the solution at z_i , we have

$$p_{i+1} = p_i + h(p_z)_i + \frac{h^2}{2}(p_{zz})_i + \frac{h^3}{6}(p_{zzz})_i + \frac{h^4}{24}p_{zzzz}(\zeta_1), \quad (13)$$

$$p_{i-1} = p_i - h(p_z)_i + \frac{h^2}{2}(p_{zz})_i - \frac{h^3}{6}(p_{zzz})_i + \frac{h^4}{24}p_{zzzz}(\zeta_2), \quad (14)$$

where $\zeta_1 \in (z_i, z_{i+1})$ and $\zeta_2 \in (z_{i-1}, z_i)$. Straightforward algebra with the previous two equations then shows that the standard centered difference approximation to the second derivative of p is ostensibly second order:

$$\frac{p_{i+1} - 2p_i + p_{i-1}}{h^2} = \frac{h^2}{24}(p_{zzzz}(\zeta_1) - p_{zzzz}(\zeta_2)). \quad (15)$$

Note, however, the relationship between the derivatives of pressure and buoyancy:

$$p_{zz} = b_z, \quad p_{zzz} = b_{zz}, \quad p_{zzzz} = b_{zzz}. \quad (16)$$

Also recall from the definition of buoyancy in (9) that the buoyancy is continuous but its derivative is not continuous. Hence, if there is a cloud edge on the interval (z_{i-1}, z_{i+1}) , the derivatives of b , and therefore of p , are not continuous, and strictly speaking, may not even exist over this entire interval. This invalidates the use of Taylor series as a way to justify the second order accurate nature of the standard centered difference approximation in Equation 15 when a cloud edge lies between two grid points.

Note that the same types of issues with non-smoothness and second-order accuracy are also present for other types of numerical methods, such as spectral methods, finite volume methods, finite element methods, and discontinuous Galerkin methods. While the weak formulation or integral-based formulation of some of these methods is valuable for certain purposes, it does not automatically yield second-order or higher-order accuracy if interfaces (e.g., cloud edges) are present. Special treatment of the interface is needed to allow second-order accuracy. Examples of special treatments include the immersed interface method for finite differences (LeVeque & Li, 1994) and the virtual node method for finite element methods (Hellrung Jr et al., 2012). We will focus on finite difference methods here because they are simple and because they are commonly used in atmospheric science. In what follows we consider the effect of interfaces on the time discretization in Section 3 and on the spatial discretization in subsequent sections.

3. Time-Discretization

In this section, we examine the impact of the buoyancy and phase changes on the temporal accuracy. The model we consider is the ordinary differential equations (ODEs) for a spatially homogeneous form of Equations 1–4.

$$\frac{d}{dt}w = \max(b_u, b_s), \quad (17a)$$

$$\frac{d}{dt}b_u = -N_u^2 w, \quad (17b)$$

$$\frac{d}{dt}b_s = -N_s^2 w, \quad (17c)$$

where w is the vertical velocity, and where a change of variables from (θ_e, q_t) to (b_u, b_s) was used, based on the form of the buoyancy b from (9). In particular, from (9), b_u is the buoyancy in unsaturated regions (where $q_t < q_{vs}(z)$) given by the expression

$$b_u = g \left[\frac{\theta_e}{\theta_0} + \left(R_{vd} - \frac{L_v}{c_p \theta_0} \right) q_t \right], \quad (18)$$

b_s is the buoyancy in saturated regions (where $q_t \geq q_{vs}(z)$) given by the expression

$$b_s = g \left[\frac{\theta_e}{\theta_0} + \left(R_{vd} - \frac{L_v}{c_p \theta_0} + 1 \right) q_{vs}(z) - q_t \right], \quad (19)$$

and N_u^2 and N_s^2 are the unsaturated and saturated squared buoyancy frequencies

$$N_u^2 = g \frac{d}{dz} \left[\frac{\tilde{\theta}_e}{\theta_0} + \left(R_{vd} - \frac{L_v}{c_p \theta_0} \right) \tilde{q}_t \right], \quad (20)$$

$$N_s^2 = g \frac{d}{dz} \left[\frac{\tilde{\theta}_e}{\theta_0} + \left(R_{vd} - \frac{L_v}{c_p \theta_0} + 1 \right) \tilde{q}_{vs}(z) - \tilde{q}_t \right]. \quad (21)$$

Recall that the background profiles $\tilde{\theta}_e(z)$, $\tilde{q}_t(z)$, and $q_{vs}(z)$ are known, making the squared buoyancy frequencies N_u^2 and N_s^2 known functions. Thus, in Equations 17a–17c, the quantities b_u , b_s , and w are unknowns and N_u^2 and N_s^2 are known, with the conversions between the variables (b_u, b_s) and (θ_e, q_t) given by (18)–(19). Further details of the conversion between the formulation in terms of (θ_e, q_t) versus (b_u, b_s) are given in Appendix A of Marsico et al. (2019). This buoyancy-based formulation has also been used in other studies (Marsico et al., 2019; Pauluis & Schumacher, 2010; Smith & Stechmann, 2017). In this formulation, the condition $q_t = q_{vs}$ is equivalent to $b_u = b_s$, and the system is saturated if $b_s > b_u$, and unsaturated if $b_u \leq b_s$. This system of ODEs shares a key feature of the governing equations in (1)–(4): the piecewise buoyancy. By neglecting spatial variations, we can isolate the impact that the piecewise nature of the buoyancy has on the temporal accuracy of numerical solutions.

To more clearly illustrate the piecewise nature of the solution, we construct an exact solution of this system of equations (see also Section 5.2 of Y. Zhang et al. (2021)). The details are given in the appendix. We use initial conditions $b_u(0) = b_s(0) = 0$, $w(0) = a > 0$, and with the assumption that $N_u^2 > N_s^2$. The exact expression for w for $t > 0$ is given by

$$w(t) = \begin{cases} a \cos(N_s(t - n\mathcal{T})) & \text{if } t \in \left[n\mathcal{T}, n\mathcal{T} + \frac{\pi}{N_s} \right], \\ -a \cos\left(N_u\left(t - n\mathcal{T} - \frac{\pi}{N_s}\right)\right) & \text{if } t \in \left[n\mathcal{T} + \frac{\pi}{N_s}, (n+1)\mathcal{T} \right], \end{cases} \quad (22)$$

where the oscillation period is

$$\mathcal{T} = \frac{\pi}{N_u} + \frac{\pi}{N_s} \quad (23)$$

and where $n = 0, 1, 2, \dots$. As shown in the appendix, the solution undergoes phase changes at times $t = n\mathcal{T}$ and $t = \pi/N_s + n\mathcal{T}$, and to illustrate the lack of smoothness of the vertical velocity at these points, we compute the derivatives with respect to time and evaluate them at the phase change at $t = \pi/N_s$ for the parameters $N_u = \sqrt{2}$, $N_s = 1$, and $a = 1$. See the appendix for details of the calculation. Routine differentiation shows that the first derivative of w is continuous because $\lim_{t \rightarrow (\pi/N_s)^+} \dot{w}(t) = \lim_{t \rightarrow (\pi/N_s)^-} \dot{w}(t) = 0$. However, for the second derivative, for the left and right limits, the results are $\lim_{t \rightarrow (\pi/N_s)^-} \ddot{w}(t) = aN_s^2$ and $\lim_{t \rightarrow (\pi/N_s)^+} \ddot{w}(t) = aN_u^2$. As we are assuming $N_u^2 \neq N_s^2$, it follows that $\ddot{w}(t)$ is not continuous. This lack of smoothness is illustrated in Figure 1. For numerical methods, recall that a correct application of Taylor series to derive a high-order time-stepping

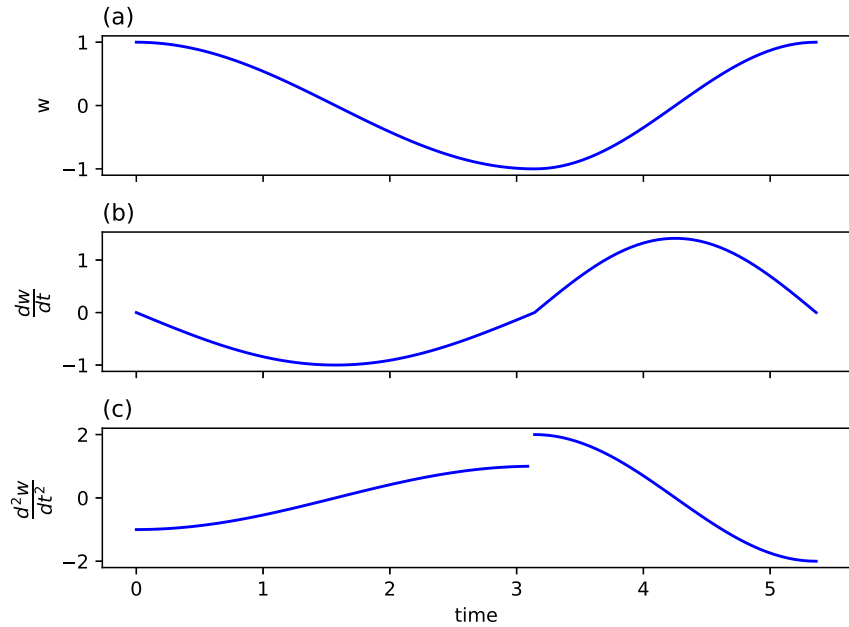


Figure 1. Lack of smoothness for exact solution (22) to the ordinary differential equation system with phase changes, (17a–17c). The parameters are $N_u = \sqrt{2}$, $N_s = 1$, and $a = 1$. Subplot (a) shows the vertical velocity, (b) shows the first derivative of the vertical velocity, and (c) shows the second derivative.

method relies on the exact solution having a certain number of continuous derivatives. Therefore, we may potentially observe a reduction in the order of accuracy of numerical solutions to Equations 17a–17c.

To test our hypothesis regarding accuracy degradation, we solve Equations 17a–17c numerically using two-stage Runge-Kutta, with the same set of parameters used in Figure 1. Figure 2 shows the numerical solution, and Figure 3 shows the convergence results. Observe from Figure 3 that all variables converge at second order, despite undergoing a phase change (see Figure 2). Although intuitively we may have expected a degradation of accuracy, we can explain the correct order of convergence as follows. Without loss of generality, assume that the solution

undergoes a phase change only once during the simulation. That is, the sign of $b_u - b_s$ changes just one time on the interval $[0, T]$, where T is the length of the simulation. The one-step error induced by this change of phase is $\mathcal{O}(\Delta t^2)$. The details of the one-step error analysis when the solution changes phase are included in an appendix. When the solution does not change phase, the one-step error of second order Runge-Kutta is $\mathcal{O}(\Delta t^3)$. The step-size, Δt , is related to T by the equation $T = M\Delta t$, where M is the number of time steps. Thus, the one-step errors consist of $M - 1$ errors that are $\mathcal{O}(\Delta t^3)$, and a single error that is $\mathcal{O}(\Delta t^2)$. To move from one-step errors to total error, one must account for the accumulation and potential growth of errors over all time steps. This can be done using standard techniques (e.g., see chapter 6 of LeVeque (2007)), although a small modification is needed to account for the $\mathcal{O}(\Delta t^2)$ error that arises at the single time step when the phase change occurs. From such an accounting, the total error is then

$$\mathcal{O}(\Delta t^3) \cdot \mathcal{O}(\Delta t^{-1}) + \mathcal{O}(\Delta t^2) \cdot \mathcal{O}(1) = \mathcal{O}(\Delta t^2), \quad (24)$$

where the relationship $M - 1 = \mathcal{O}(\Delta t^{-1})$ was used, as follows from $T = M\Delta t$. We therefore can expect to see second order convergence for the spatially homogeneous system even in the presence of a phase change.

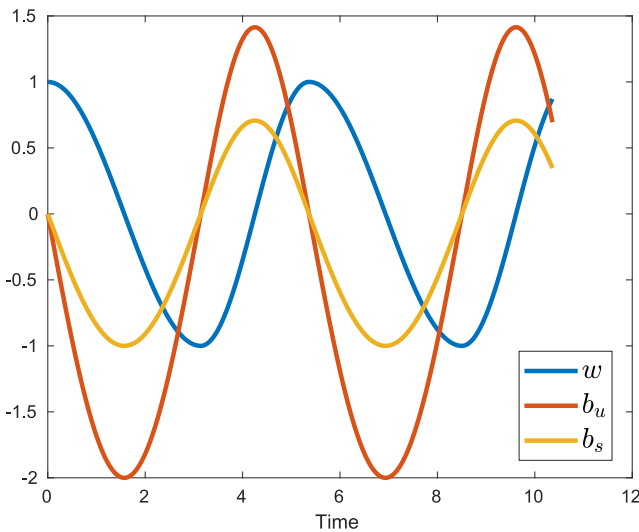


Figure 2. The numerical solution to the ordinary differential equation system with phase changes, (17a)–(17c), for the parameters $N_u = \sqrt{2}$, $N_s = 1$, and $a = 1$. The convergence test is done at $t = 5$.

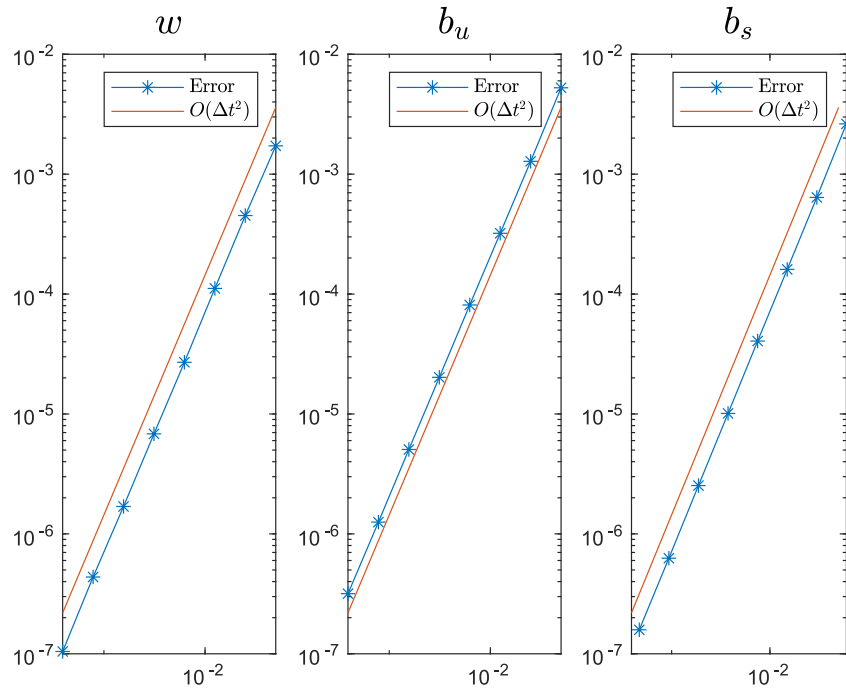


Figure 3. Convergence results for the ordinary differential equation system with phase changes, (17a–17c). The plots show error versus time step Δt , shown on a log-log scale.

For the spatially varying dynamics of the atmosphere, one can potentially expect an estimate like (24) to continue to hold, since the number of space-time points that experience a phase change is a small fraction of the total number of space-time points in the space-time grid. Numerical simulations are conducted for time-evolving dynamics with spatial variations in Section 5, after first considering the pressure Poisson equation in Section 4.

4. Spatial Discretization of Pressure Poisson Equation

In this section, we examine the role of cloud edge in the spatial discretization of the pressure Poisson equation. First, we consider the one-dimensional Poisson equation, and investigate correction equations for the pressure and its gradient in the case of a discontinuous forcing, as would be the case for a moist atmosphere with phase changes. We then demonstrate that these one-dimensional correction equations for the pressure and gradient can be used to achieve second order convergence for the pressure and pressure gradient for a two-dimensional discontinuous forcing.

4.1. Poisson Equation in One Dimension

Consider the one-dimensional Poisson equation,

$$\frac{d^2 p}{dx^2} = f(x). \quad (25)$$

In this section, we derive the second order approximations for the pressure and pressure gradient in one dimension. Let p_i denote the pressure at the grid-point $x_i = i\Delta x$, where Δx is the grid-spacing. Suppose the cloud-edge, that is, the discontinuity of $f(x)$, is located within the interval $[x_i, x_{i+1}]$, at interface location $x_l = x_i + (1 - \alpha)(x_{i+1} - x_i)$, where $\alpha \in [0, 1]$ (see Figure 4).

When the location of the cloud interface is known in the form of an analytic expression, we use the exact location of x_l . This is the case for the linear-steady state problems we consider in Equation 44. When the interface is unknown, we use linear interpolation to estimate its location. This is the case for the full nonlinear simulations, and we describe the procedure in more detail in Section 5.2. We refer to the points x_i and x_{i+1} as “cloud edge adjacent” grid-points, in order to distinguish them from “regular” grid-points that are not adjacent to a cloud-edge.

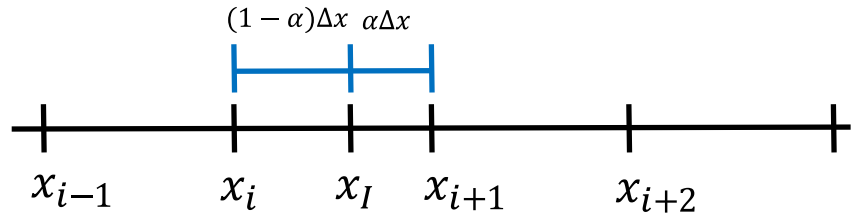


Figure 4. The location of the cloud edge, denoted by interface location x_I , on a one dimensional evenly spaced grid with grid points x_i .

At regular grid-points, we use a centered, second order finite difference approximation to Poisson's equation:

$$\frac{p_{i-1} - 2p_i + p_{i+1}}{\Delta x^2} = f_i. \quad (26)$$

At cloud-edge-adjacent grid-points, on the other hand, we use the method from Tzou and Stechmann (2019). While it was originally developed for steady-state problems, here we apply this correction method to the time-dependent equations of atmospheric flow. Furthermore, here, we also derive additional correction terms for gradients of variables that do not vary smoothly across cloud edges. In particular, additional correction terms for the gradient of pressure are considered below in the present section, and the derivative of buoyancy is considered in Section 5.1.

The idea behind the method is that if the number of cloud-edge grid-points is small, then a first order accurate discretization of p_{xx} at the cloud-edge grid-points will not degrade the global accuracy of the method. Proofs of this can be found in the literature (LeVeque, 2007; LeVeque & Li, 1994; Tzou & Stechmann, 2019). Using this idea, one can derive a first-order discretization for Poisson's equation at cloud-edge-adjacent grid-points x_i and x_{i+1} that is similar to Equation 26, but with additional correction terms on the right hand side, and take the form

$$\frac{p_{i-1} - 2p_i + p_{i+1}}{\Delta x^2} = \left(\frac{2 - \alpha}{2}\right)f_i + \frac{\alpha}{2}(\alpha f_{i+} + (1 - \alpha)f_{i-}), \quad (27)$$

$$\frac{p_i - 2p_{i+1} + p_{i+2}}{\Delta x^2} = \left(\frac{1 + \alpha}{2}\right)f_{i+1} + \frac{1 - \alpha}{2}(\alpha f_{i+} + (1 - \alpha)f_{i-}), \quad (28)$$

where subscripts “−” and “+” indicate left limit and right limit, respectively. Note that the interface correction terms on the right-hand side of (27)–(28) include the appearance of α ; that is, the correction terms include information about the sub-grid-scale location of the interface. Also note the desirable property that the standard centered discretization appears at all grid points in (26)–(28), and only the right-hand side requires modification at cloud-edge-adjacent grid points.

As it is ultimately the derivative of pressure that impacts the velocities, we also need accurate expressions for p_x at $x_{i+1/2}$, where $x_{i+1/2} = x_i + \Delta x/2$. To derive the discretization for $(p_x)_{i+1/2}$, the derivative of p at the point $x_{i+1/2}$, we first note that.

$$p(x_i) = p(x_i^-) + p'(x_i^-)(x_i - x_I) + \frac{p''(x_I^-)}{2}(x_i - x_I)^2 + \mathcal{O}(\Delta x^3), \quad (29)$$

$$p(x_{i+1}) = p(x_I^+) + p'(x_I^+)(x_{i+1} - x_I) + \frac{p''(x_I^+)}{2}(x_{i+1} - x_I)^2 + \mathcal{O}(\Delta x^3). \quad (30)$$

Since $p_{xx} = f$ and f is discontinuous at x_I , we know p and p_x are continuous at x_I . Using this fact, from (29)–(30) we find

$$p'(x_I) = \frac{p(x_{i+1}) - p(x_i)}{\Delta x} + \frac{p''(x_I^+)}{2\Delta x}(x_{i+1} - x_I)^2 - \frac{p''(x_I^-)}{2\Delta x}(x_i - x_I)^2 + \mathcal{O}(\Delta x^2). \quad (31)$$

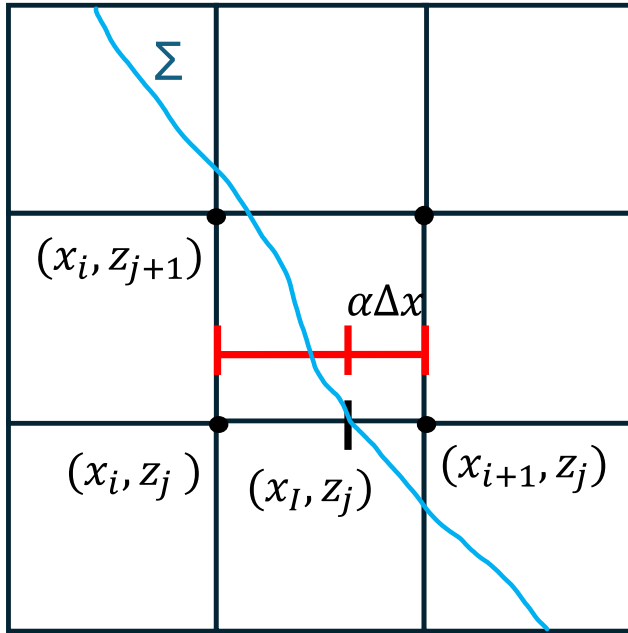


Figure 5. Illustration of the grid in 2D in the presence of an interface, that is, cloud edge. The cloud edge, Σ , intersects the line $z = z_j$ at the point (x_I, z_j) . Here, the points located at (x_i, z_j) and (x_{i+1}, z_j) are two examples of cloud-edge-adjacent grid-points.

Next, we aim to replace the term $p'(x_I)$ with a formula including the term $p'(x_{i+\frac{1}{2}})$. Note that if $x_I \leq x_{i+\frac{1}{2}}$, then

$$p'(x_{i+\frac{1}{2}}) = p'(x_I) + p''(x_I^+)(x_{i+\frac{1}{2}} - x_I) + \mathcal{O}(\Delta x^2). \quad (32)$$

Similarly, if $x_I \geq x_{i+\frac{1}{2}}$, then

$$p'(x_{i+\frac{1}{2}}) = p'(x_I) + p''(x_I^-)(x_{i+\frac{1}{2}} - x_I) + \mathcal{O}(\Delta x^2). \quad (33)$$

We can combine the above two equations in a more compact form:

$$p'(x_{i+\frac{1}{2}}) = p'(x_I) + (\chi_{x_I \leq x_{i+\frac{1}{2}}} p''(x_I^+) + \chi_{x_I > x_{i+\frac{1}{2}}} p''(x_I^-))(x_{i+\frac{1}{2}} - x_I) + \mathcal{O}(\Delta x^2), \quad (34)$$

where

$$\chi_{x_I \leq x_{i+\frac{1}{2}}} = \begin{cases} 1 & \text{if } x_I \leq x_{i+\frac{1}{2}}, \\ 0 & \text{else,} \end{cases} \quad (35)$$

and

$$\chi_{x_I > x_{i+\frac{1}{2}}} = \begin{cases} 1 & \text{if } x_I > x_{i+\frac{1}{2}}, \\ 0 & \text{else,} \end{cases} \quad (36)$$

Finally, we combine (31) and (34) and obtain

$$p'(x_{i+\frac{1}{2}}) = \frac{p(x_{i+1}) - p(x_i)}{\Delta x} + \frac{p''(x_I^+)}{2\Delta x}(x_{i+1} - x_I)^2 - \frac{p''(x_I^-)}{2\Delta x}(x_i - x_I)^2 + (\chi_{x_I \leq x_{i+\frac{1}{2}}} p''(x_I^+) + \chi_{x_I > x_{i+\frac{1}{2}}} p''(x_I^-))(x_{i+\frac{1}{2}} - x_I) + \mathcal{O}(\Delta x^2). \quad (37)$$

4.2. Poisson Equation in Two Dimensions

In this section, we extend our one-dimensional discretizations for the pressure and pressure gradient to the two-dimensional Poisson equation given by

$$\nabla^2 p(x, z) = f(x, z). \quad (38)$$

Here, we assume that the pressure is stored at the grid points (x_i, z_j) , where $x_i = i\Delta x$ and $z_j = j\Delta z$.

Assume that there is a cloud edge, denoted by Σ , that passes through the numerical mesh, as illustrated in Figure 5. We assume that f , the right-hand side in (38), is discontinuous on Σ .

We now describe the extension of the one-dimensional ideas from Section 4.1 to the two-dimensional case. Again, the cloud-edge-adjacent grid points will need special treatment, for which we follow the method from Tzou and Stechmann (2019). In the two-dimensional case, to define which grid points are the cloud-edge-adjacent grid points, consider the four line segments that are connected to each grid point in the numerical mesh. For example, in Figure 5, the grid point (x_i, z_j) is connected to one mesh-line segment that intersects the cloud interface (i.e., the mesh-line segment from (x_i, z_j) to (x_{i+1}, z_j)), so (x_i, z_j) is considered a cloud-edge-adjacent grid point. On the other hand, for the grid point (x_{i+1}, z_j) , none of its four connected mesh-line segments is crossed by the cloud interface, so it is considered a regular grid point.

We now use grid point (x_i, z_j) from Figure 5 as an example of a cloud-edge-adjacent grid point, and we describe the interface correction terms that are needed. For grid point (x_i, z_j) , note that the cloud edge passes through one of

its four mesh-line segments, and it is the horizontal line segment between (x_i, z_j) and (x_{i+1}, z_j) . We denote the point where Σ intersects the line $z = z_j$ by (x_I, z_j) . To obtain an approximation of p_{xx} at the cloud-edge grid-point (x_i, z_j) , we can replace f in Equation 27 by the expression $f - p_{zz}$. This leads to the following approximation to Equation 38 at the cloud-edge-adjacent grid-point (x_i, z_j) :

$$\begin{aligned} \frac{-2p_{i,j} + p_{i-1,j} + p_{i+1,j}}{\Delta x^2} + \frac{-2p_{i,j} + p_{i,j-1} + p_{i,j+1}}{\Delta z^2} &= \left(\frac{2-\alpha}{2}\right)f_{i,j} + \frac{\alpha}{2}(p_{zz})_{i,j} \\ &+ \frac{\alpha}{2}[\alpha(f - p_{zz})(x_{I+}, z_j) \\ &+ (1-\alpha)(f - p_{zz})(x_{I-}, z_j)]. \end{aligned} \quad (39)$$

A first order accurate approximation to the term $p_{zz}(x_{I-}, z_j)$ is simply $p_{zz}(x_i, z_j)$, which we approximate using standard centered differences with $p_{i,j}$, $p_{i,j+1}$ and $p_{i,j-1}$. Similarly, we can use $p_{zz}(x_{I+}, z_j)$ as a first order accurate approximation to $p_{zz}(x_{I+}, z_j)$. Note that as we are assuming the number of cloud-edge grid-points is small (relative to the number of grid-points on the overall grid), a first order accurate approximation will not degrade the overall second-order accuracy of the method (LeVeque, 2007).

Expressions similar to Equation 39 can be derived for other scenarios of cloud-edge-adjacent grid points as well. For instance, in Figure 5, the grid point (x_{i+1}, z_j) is a cloud-edge-adjacent grid point that has the additional feature that two of its mesh-line segments are intersecting the interface. In such a scenario, two groups of correction terms are needed on the right-hand side, one group for each mesh-line segment that is crossed by the interface. For further details, we refer the reader to Appendix A of Tzou and Stechmann (2019).

To derive an expression for the gradient of pressure, we follow the derivation from (29) to (37) to arrive at the following expression for p_x at $(x_{i+1/2}, z_j)$:

$$\begin{aligned} p_x(x_{i+1/2}, z_j) &= \frac{p(x_{i+1}, z_j) - p(x_i, z_j)}{\Delta x} + \frac{p_{xx}(x_I^+, y_j)}{2\Delta x}(x_{i+1} - x_I)^2 \\ &- \frac{p_{xx}(x_I^-, y_j)}{2\Delta x}(x_i - x_I)^2 \\ &+ (\chi_{x_I \leq x_{i+1/2}} p_{xx}(x_I^+, z_j) + \chi_{x_I > x_{i+1/2}} p_{xx}(x_I^-, z_j))(x_{i+1/2} - x_I) + \mathcal{O}(\Delta x^2). \end{aligned} \quad (40)$$

Then we approximate the terms $p_{xx}(x_{I+}, z_j)$ and $p_{xx}(x_{I-}, z_j)$ by $(f - p_{zz})(x_{i+1}, z_j)$ and $(f - p_{zz})(x_{i-1}, z_j)$, respectively, as in (39) and the discussion following (39). A similar expression can be derived for $p_z(x_i, z_{j+1/2})$ as well.

For regular grid-points, we can use the standard five-point discretization of the Laplacian, which results in the equation

$$\frac{-2p_{i,j} + p_{i-1,j} + p_{i+1,j}}{\Delta x^2} + \frac{-2p_{i,j} + p_{i,j-1} + p_{i,j+1}}{\Delta z^2} = f_{i,j}. \quad (41)$$

The standard centered difference equation can be used for the pressure gradient at regular points as well:

$$p_x(x_{i+1/2}, z_j) = \frac{p(x_{i+1}, z_j) - p(x_i, z_j)}{\Delta x}, \quad (42a)$$

$$p_z(x_i, z_{j+1/2}) = \frac{p(x_i, z_{j+1}) - p(x_i, z_j)}{\Delta z}. \quad (42b)$$

To test these corrections, we solve the equation

$$\nabla^2 p = f \quad (43)$$

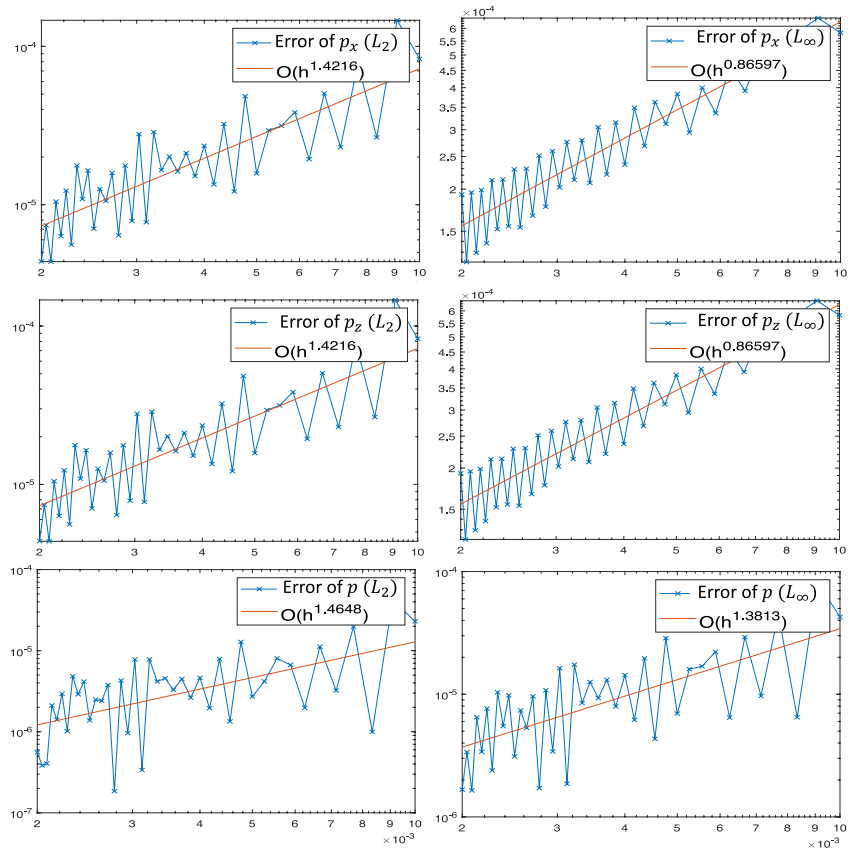


Figure 6. Convergence results for the Poisson equation, using methods without interface corrections. The errors are plotted on a log-log scale as a function of $\sqrt{\Delta x \Delta z}$. Order of convergence is approximately 1.5 in L^2 norm, that is, root-mean-square error (left column) and approximately 1 in L^∞ norm, that is, maximum error (right column) for p_x , p_z , and p .

on a two-dimensional domain with a length of L_x in the horizontal and L_z in the vertical. Here, f is essentially a stand-in for $\partial b / \partial z$ in a cloudy atmosphere (see Section 2.2). Therefore, we define it so that the solution p to (43) has the same smoothness that we would expect the pressure to have in the simulation of a cloudy atmosphere, namely, a function with a continuous first derivative, but discontinuous second derivative. To that end, we use the method of manufactured solutions and define p as

$$p(x, z) = \begin{cases} p_1(r) := Ae^{-\gamma r^2} r^4 & 0 \leq r \leq r_0, \\ p_2(r) := Ae^{-\gamma r^2} (-\lambda r_0^2 r^2 + 2\lambda r_0 r^3 + (1 - \lambda) r^4) & r > r_0, \end{cases} \quad (44)$$

where $r = \sqrt{(x - x_c)^2 + (z - z_c)^2}$, $x_c = L_x/2$, and $z_c = L_z/2$. In our numerical tests, we let $L_x = L_z = 1$, and $r_0 = L_x/4$. The remaining parameters in (44) are chosen so that $p_1(r_0) = p_2(r_0)$ and $p'_1(r_0) = p'_2(r_0)$. We then define the forcing f in (43) as simply $\nabla^2 p$. As p possesses continuous first derivative only, the term $\nabla^2 p$ will be discontinuous. Note that, in accordance with the method of manufactured solutions, this definition of the forcing f implies that p from (44) is an exact solution to (43).

First, we solve Poisson's equation without any corrections to the pressure or its gradient. That is, we use Equations 41 and 42 at every grid-point. The error is plotted as a function of the equivalent mesh width, $\sqrt{\Delta x \Delta z}$, on a log-log scale. As can be seen from Figure 6, this results in only first order convergence to the pressure and its gradient in the L^∞ norm, and approximately 1.5 order convergence in the L^2 norm.

Then we incorporate the correction to pressure only, that is, we use Equation 39 to approximate Poisson's equation at cloud-edge grid-points, but we make no correction to the pressure gradient. As can be seen from Figure 7, we

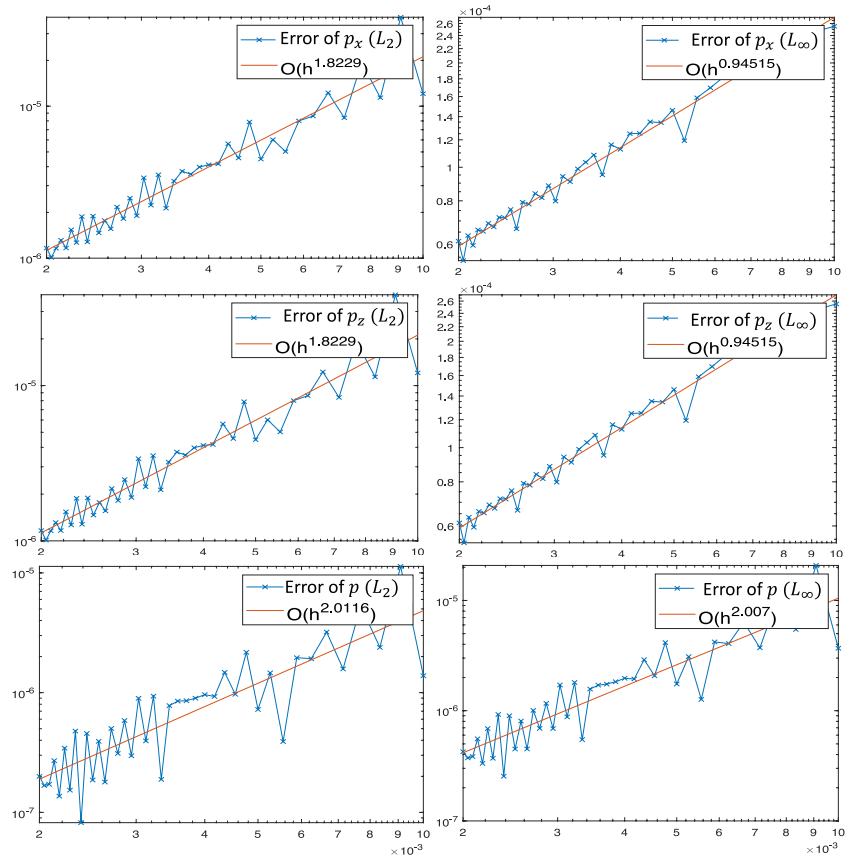


Figure 7. Convergence results for the Poisson equation, using methods with interface correction to the pressure (as in (39)) but without interface correction for pressure gradient. The errors are plotted on a log-log scale as a function of $\sqrt{\Delta x \Delta z}$. Order of convergence is approximately 2 for p in both L^2 and L^∞ norms (bottom row) and for p_x and p_z in L^2 norm (left column), but is approximately 1 for p_x and p_z in L^∞ norm (right column).

obtain second order accurate convergence for the pressure in both the L^2 norm and L^∞ norm. The gradient of pressure converges at second order in the L^2 norm, but only first order in the L^∞ norm. This indicates that it is not enough to simply correct the pressure to obtain second order convergence for p_x and p_z in the L^∞ norm.

Second order convergence of all variables in both the L^2 and L^∞ norms can only be obtained by using both Equation 39 for solving Poisson's equation, and Equation 40 for computing p_x and p_z . This is confirmed in Figure 8, where p , p_x , and p_z all converge at second order for both norms.

5. Time Dependent Moist System

In this section, we consider the full system from (1)–(4), that is, the moist Boussinesq equations with clouds and phase changes. First we consider a test problem without advection in Section 5.1 to consider almost the entirety of the complications of the full system. A further modification is included, as an interface correction for computing the piecewise-defined buoyancy. Then the full system is considered in Section 5.2.

5.1. Interface Correction for Piecewise Buoyancy, in Test Problem Without Advection

We now incorporate time dependence by first looking at the system

$$\frac{\partial \mathbf{u}}{\partial t} = -\nabla p + \mathbf{b}, \quad (45)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (46)$$

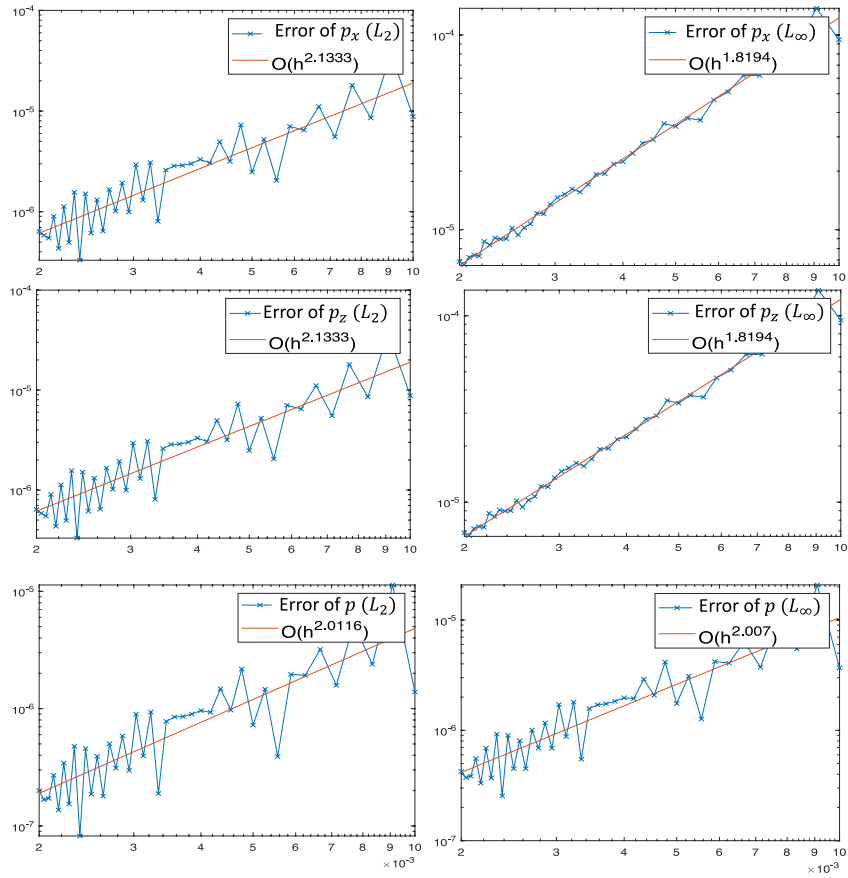


Figure 8. Convergence results for the Poisson equation, using methods with interface corrections to both the pressure (as in (39)) and pressure gradient (as in (40)). The errors are plotted on a log-log scale as a function of $\sqrt{\Delta x \Delta z}$. Order of convergence is approximately 2 for all of p , p_x , and p_z in both L^2 and L^∞ norms.

where $\mathbf{b}$ is a buoyancy-like vector, which we refer to as buoyancy for brevity. In the absence of non-linear advection in this system, the Poisson equation that enforces incompressibility is obtained by taking the divergence of the momentum equations, and using incompressibility to eliminate the time tendency of velocity. This results in the equation

$$\nabla^2 p = f, \quad (47)$$

where $f = \nabla \cdot \mathbf{b}$. We prescribe the buoyancy as

$$\mathbf{b} = \nabla^\perp \phi(\mathbf{x}) + \nabla \phi(\mathbf{x}), \quad (48)$$

where ϕ is defined in the functional form of (44). Note that (48) essentially amounts to a Helmholtz decomposition of the vector field $\mathbf{b}$ into the sum of a divergence-free component, $\nabla^\perp \phi(\mathbf{x})$, and a curl-free component $\nabla \phi(\mathbf{x})$, where the potential ϕ is used for both components for simplicity. The operator $\nabla^\perp$ is defined as $\nabla^\perp = (-\partial_z, \partial_x)$.

Now an interface correction is described for computing the derivative of a piecewise-defined buoyancy. In the test cases in Section 4, in the pressure Poisson equation, the buoyancy derivative $\partial b / \partial z$ was treated as a prescribed function, f , to emphasize the influence of non-smoothness at an interface. However, in the full system in (1)–(4), the buoyancy derivative $\partial b / \partial z$ is computed by taking a derivative of a piecewise-defined function, given in (9), and care should be taken in computing a derivative near the interface.

To avoid computing derivatives in Equation 48 across interfaces, we use the fact that the buoyancy is composed of two functions that are smooth over the entire domain, ϕ_1 and ϕ_2 (see (44) and (9)). For instance, we compute the gradient of ϕ according to

$$\nabla\phi(\mathbf{x}) = H(r - r_0)\nabla\phi_1(\mathbf{x}) + (1 - H(r - r_0))\nabla\phi_2(\mathbf{x}), \quad (49)$$

where H is the Heaviside step function. The gradient is then equal to $\nabla\phi_1$ when $r - r_0 > 0$, and equal to $\nabla\phi_2$ when $r - r_0 \leq 0$. The derivative $\nabla \cdot \mathbf{b}$ for buoyancy, for (47), is computed similarly. This is analogous to the calculation of $\partial b / \partial z$ based on (9) for the pressure Poisson equation in (11) for the full system, given prognostic variables θ_e and q_t .

Here, we work on a staggered grid (or Arakawa C-grid), so that the spatial domain consists of grid cells whose centers are located at the points (x_i, z_j) , where $x_i = i\Delta x$ and $z_j = j\Delta z$. The buoyancy and pressure are stored at cell-centers. The horizontal velocity is stored at the midpoints of vertical cell sides, which have coordinates $(x_{i+1/2}, z_j)$, where $x_{i+1/2} = (i + 1/2)\Delta x$. The vertical velocity is stored at the midpoints of horizontal cell edges, which have coordinates $(x_i, z_{j+1/2})$, where $z_{j+1/2} = (j + 1/2)\Delta z$.

For our numerical tests, we use a domain whose dimensions are $L_x = 6400$ m in the horizontal, $L_z = 5120$ m in the vertical, and let $r_0 = 1200$ m. We also allow the buoyancy to move vertically at a constant speed of $v_0 = 20$ m/s, so that we replace $\phi(\mathbf{x})$ in (48) with $\phi(\mathbf{x} - tv_0\hat{\mathbf{z}})$. Using the form of the buoyancy given in Equation 48, it is straightforward to obtain the exact solutions for velocity and pressure.

$$\mathbf{u} = \int_0^t \nabla^\perp \phi(\mathbf{x} - sv_0\hat{\mathbf{z}}) ds, \quad (50)$$

$$p = \phi(\mathbf{x} - tv_0\hat{\mathbf{z}}). \quad (51)$$

We use this exact solution as a baseline for comparison to the numerical solutions to determine the order of convergence. For our numerical tests, we use a two-stage Runge-Kutta projection method where, at each time step, Equation 47 is solved to enforce incompressibility. Figure 9 shows the results of the convergence test in the absence of any interface corrections, and Figure 10 shows results including interface corrections to the pressure, the pressure gradient, and the buoyancy derivative (similar to (49)). The errors are plotted as a function of the equivalent mesh width, $\sqrt{\Delta x \Delta z}$, on a log-log scale. Incorporating the interface corrections gives an improvement in the order of convergence for all variables in the L^2 norm. The increase in the rate of convergence for the pressure in the L^2 norm is particularly pronounced, as we observe a full order of accuracy improvement. In the case of the L^∞ norm, there is nearly one full order of accuracy improvement in the convergence rates of all the variables.

5.2. The Full Moist System With Phase Changes

We now incorporate interface corrections into the full, nonlinear moist system. Specifically, we solve Equations 1–4 on a domain that is 5 km in the vertical, and 6.4 km in the horizontal. The benchmark test case here is a variant of the commonly used rising thermal, in an initial state that is unsaturated (i.e., without clouds initially). A cloud forms after the thermal rises sufficiently high. For initial conditions, all variables are set to zero, except the potential temperature, which we set to be a Gaussian perturbation of the form

$$\theta(x, z, 0) = 3 \exp \left[-\frac{1}{a^2} ((x - x_c)^2 + (z - z_c)^2) \right], \quad (52)$$

where $a = 450$ m, $x_c = 3200$ m, and $z_c = 1800$ m. For the background moisture and potential temperature profiles, we use

$$\tilde{q}_t(z) = Az + q_0, \quad (53)$$

$$\tilde{\theta}(z) = Bz + \theta_0, \quad (54)$$

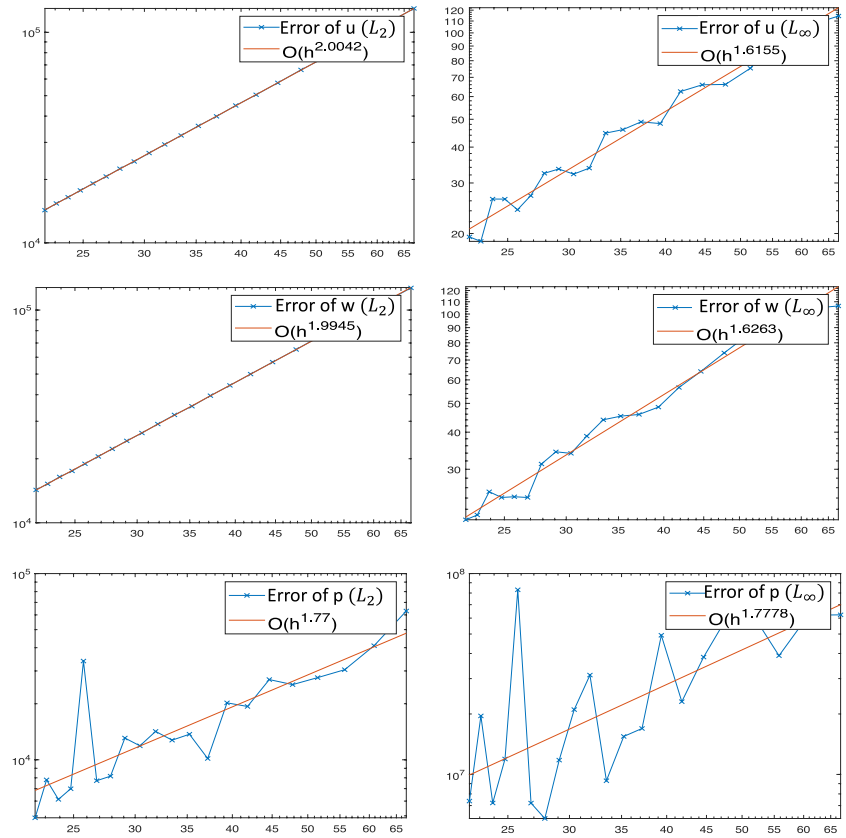


Figure 9. Numerical solution to the time dependent system with phase interface from (45)–(46) using methods without any interface corrections. The errors are plotted on a log-log scale as a function of $\sqrt{\Delta x \Delta z}$. Compare convergence rates to Figure 10, which incorporates interface corrections.

where $A = -17/5 \text{ g kg}^{-1} \text{ km}^{-1}$, $q_0 = 17 \text{ g kg}^{-1}$, $B = 3 \text{ K km}^{-1}$, and $\theta_0 = 300 \text{ K}$. The equivalent potential temperature anomaly θ_e and its background profile, $\tilde{\theta}_e(z)$, are obtained by using the linearized thermodynamic relation $\theta_e = \theta + (L_v/c_p)q_v$, where $L_v = 2.5 \times 10^6 \text{ J kg}^{-1}$ and $c_p = 1000 \text{ J kg}^{-1} \text{ K}^{-1}$ are the latent heat of vapourization and the specific heat at constant pressure, respectively.

We estimate the location of the interface using linear interpolation, since, unlike the linear test problem in Equation 44, the location of the interface is now not known exactly. In more detail, suppose there is a phase interface at the spatial gridpoint (x_l, z_l) that lies between (x_i, z_j) and (x_{i+1}, z_j) (this is shown in Figure 5). We fit a line through the points $(x_i, z_j, b_u(x_i, z_j) - b_s(x_i, z_j))$ and $(x_{i+1}, z_j, b_u(x_{i+1}, z_j) - b_s(x_{i+1}, z_j))$. We then find the zero of this line on the interval (x_i, x_{i+1}) and use it as the estimate of x_l . Note that this zero is guaranteed to exist because the sign of $b_u - b_s$ is different at (x_i, z_j) and (x_{i+1}, z_j) .

Plots of the simulation are shown in Figures 11 and 12. Figure 11 shows the fields after they have evolved for 4 min, which is approximately 2 min after a cloud has formed. The cloud can be seen in the upper left-most panel. In order to demonstrate the loss of regularity at the cloud-edge, Figure 12 shows a more detailed view of the potential temperature at the same time. Here, the contours of the potential temperature are manifestly non-differentiable at the edge of the cloud, as can be seen by the “corners” of the contours.

In Figures 13 and 14, the rates of convergence of each variable are plotted as functions of time. In other words, at each time t , a convergence test is conducted. In the absence of an exact solution, we compute the errors relative to a high-resolution numerical reference solution. To be precise, let $G_i(t)$ denote the numerical solution on grid refinement level i at time t , and let $G_r(t)$ denote the reference numerical solution. If we define the error at each time step as $E_i(t) = \|G_i(t) - G_r(t)\|$, then the rate of convergence, $R(t)$, for grid refinement level i is defined to be the ratio of $E_i(t)$ to $E_{i+1}(t)$, that is $R(t) = \log_2(E_i(t)/E_{i+1}(t))$, where a logarithm base 2 is used to correspond to the

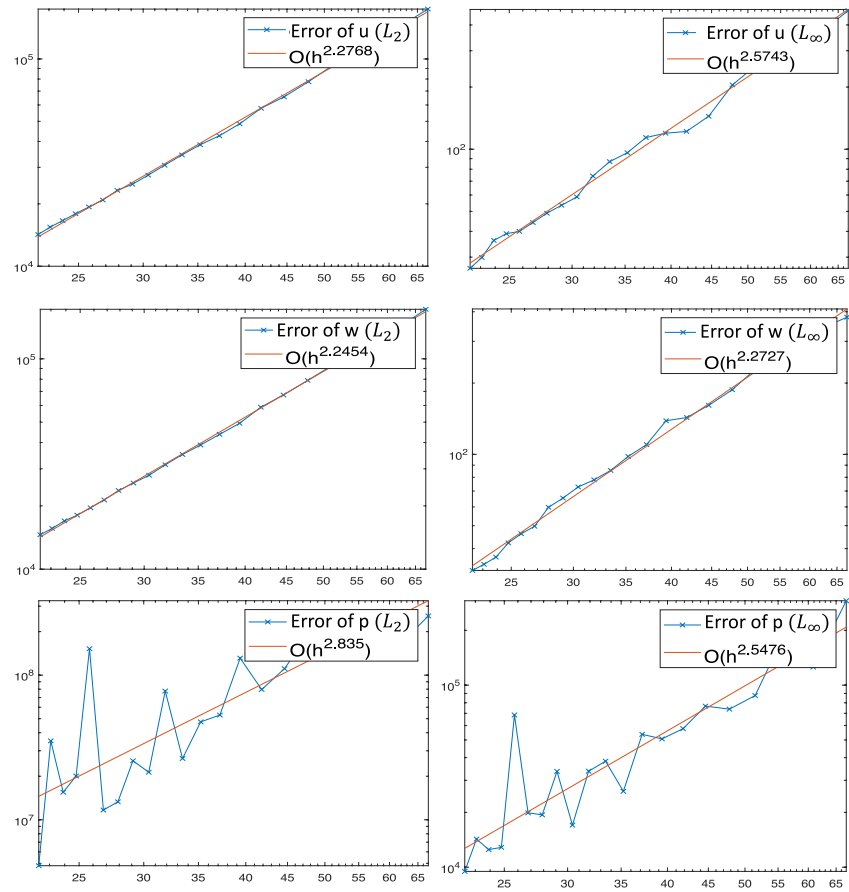


Figure 10. Numerical solution to the time dependent system with phase interface from (45)–(46), using methods with interface corrections to the pressure (Equation 39), pressure gradient (Equation 40), and buoyancy derivative (similar to (49)). The errors are plotted on a log-log scale as a function of $\sqrt{\Delta x \Delta z}$. Convergence rates in L^∞ norm are approximately one order higher here, in comparison to Figure 9 results that do not utilize interface corrections. Second-order accuracy is seen in all variables and in both L^2 and L^∞ norms.

reduction of mesh width by a factor of 2 in changing from grid refinement level i to level $i + 1$. For each grid refinement level i , we plot $R(t)$ as a function of t .

In Figure 13, we first plot the rate of convergence in the case where moisture is present, but in the absence of phase changes. That is, we solve Equations 1–4, but where the buoyancy is simply the unsaturated buoyancy b_u , defined in Equation 18. Here, we observe manifestly second order accuracy for the velocity, equivalent potential temperature, and total water in both the L^2 and L^∞ norms under grid refinement, that is as the grid refinement level number increases from one to four. We can therefore be certain that if there is a degradation in accuracy, it will be due to the presence of clouds.

In Figure 14, we plot the results in the case where cloud formation occurs. Here, the setup is the same as it was before, but now moisture is allowed to change phase from water vapor to cloud water. We observe clear second order accuracy for both the L^2 and L^∞ norms for the first 2 minutes of the simulation, before a cloud forms. However, after cloud formation at approximately 2 minutes, the accuracy picture is more complex. On the one hand, for the L^2 error, we observe approximately second-order accuracy for all variables under grid refinement, that is, as the grid refinement level increases from one to four. On the other hand, for the L^∞ error, the convergence rate is degraded and appears to be less than second order, after cloud formation at approximately $t = 2$ minutes.

Given that the interface corrections derived in the previous sections did not result in manifestly second order rates of convergence in the L^∞ norm for the full moist system with phase changes, one might wonder if the solutions

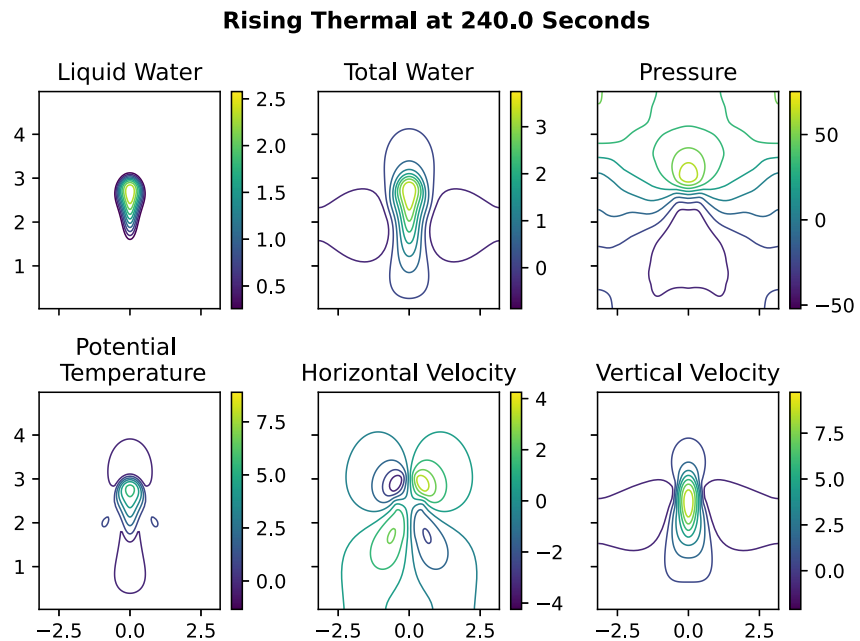


Figure 11. Numerical solution to the full moist system (1)–(4) at $t = 240$ seconds for a rising thermal test case. Quantities are plotted as functions of x (km) and z (km). Units are g kg^{-1} for liquid water q_l and total water q_t , $\text{Pa (kg m}^{-3})^{-1}$ for p (which has been scaled by a constant Boussinesq density of $\rho_0 = 1 \text{ kg m}^{-3}$), K for potential temperature θ , and m s^{-1} for u and w .

lack the necessary amount of differentiability, or regularity. One difficulty that arises when phase changes are included is that the regularity of the solutions is not as clear as it was in the absence of any phase changes, and it may be the case that the smoothness assumptions that aided in the derivation of the interface corrections in Section 4.2 may not be applicable here. Results on well-posedness and the regularity of solutions have been presented, for example, by Cao et al. (2018) and Hittmeir et al. (2020), for atmospheric dynamics with various treatments of moisture and phase changes. For instance, in Hittmeir et al. (2020), a system with moisture and phase changes was considered, and it was shown that solutions (as members of the Sobolev space H^1) are C^0 in time, that is, continuous, but not continuously differentiable, in time. However, as an example comparison, for the system in Equations 17a–17c, solutions were C^1 in time, which was critical for second order accuracy of Runge-Kutta time integration. Some differences exist between the non-hydrostatic, non-precipitating model here, on the one hand, and the hydrostatic, warm-rain microphysics of (Hittmeir et al., 2020), on the other hand. Another regularity result for cloud edge was proved by Remond-Tiedrez et al. (2024) although it is for an elliptic PDE, not dynamics, and for a quasi-geostrophic setting. For a summary of other relevant literature, see Remond-Tiedrez et al. (2024) and references therein. Without knowing the regularity of solutions, it is not clear if second order numerical accuracy can truly be achieved, and in which norms.

6. Concluding Discussion

We have examined the impact of cloud formation and cloud edges on the order of accuracy of numerical solutions. Clouds have an influence on numerical accuracy because cloud edge is an interface where non-smoothness is present. The non-smoothness arises due to cloud latent heating and the threshold effect of the saturation vapor pressure. One manifestation of the non-smoothness is in the buoyancy: buoyancy is continuous but its derivative is not continuous. Intuitively, we expect the accuracy of standard finite

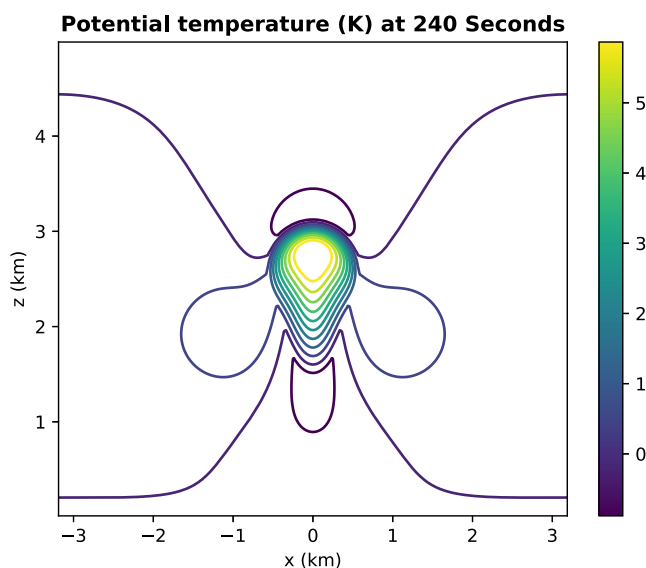


Figure 12. Zoomed-in view of potential temperature θ for the full moist system at $t = 240$ s, to illustrate the “corners” in the contours, which are indicators of non-differentiability.

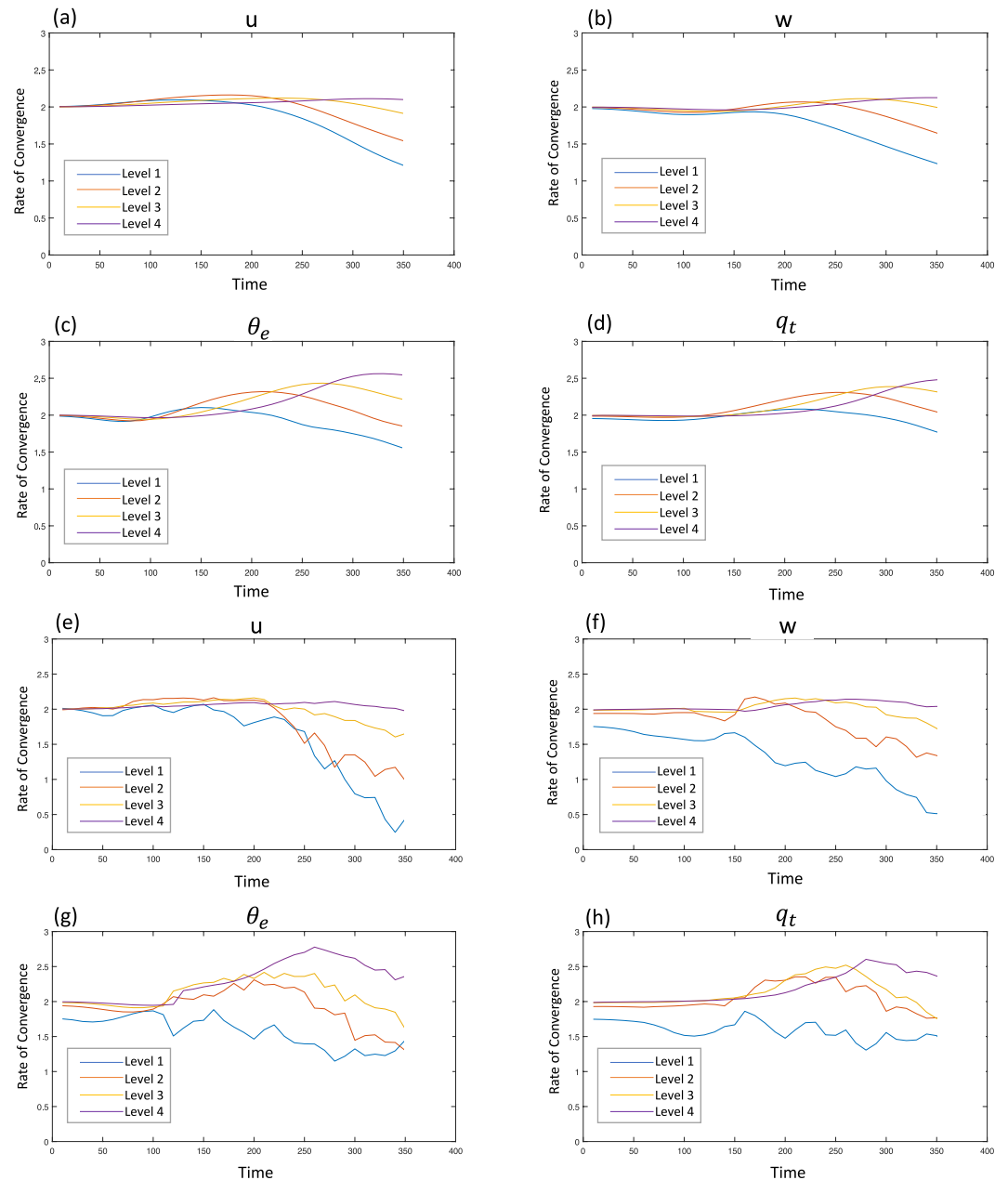


Figure 13. Convergence results for the numerical solution of the full moist system but in the absence of phase changes. The top four panels (a–d) show the L^2 errors, and the bottom four panels (e–h) show the L^∞ errors. At each time, for all variables, and in both L^2 and L^∞ norms, a second-order rate of convergence is seen as the grid is refined from grid refinement level 1 to level 4.

difference methods to be degraded in the presence of clouds and phase changes, since they are based on Taylor series approximations, which can break down due to non-smoothness at cloud edge. Consequently, to achieve higher-order accuracy, finite difference methods must be modified to include interface corrections, in the spirit of numerical methods for two-phase fluid flow (see discussion in Section 1 and the end of Section 2.2). In brief, two of the key ideas of interface corrections are that (a) care must be taken to ensure that Taylor series approximations are not applied across an interface, and (b) in ensuring this, the sub-grid-cell location of the interface needs to be taken into account.

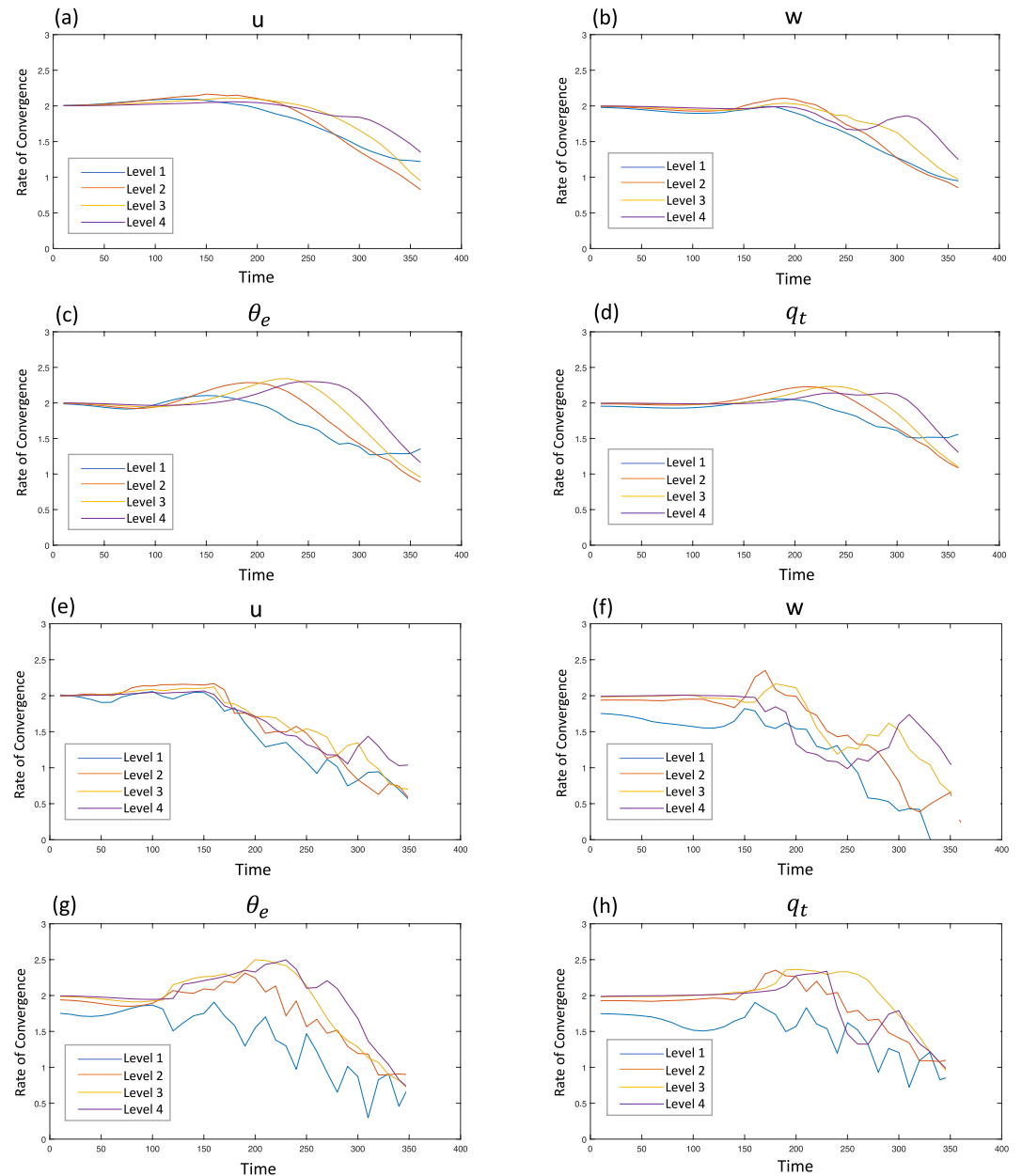


Figure 14. Convergence results for the numerical solution of the full moist system (1)–(4) with phase changes. (a–d) In the L^2 norm, the convergence rate is approximately second-order at each time t up to time $t \approx 300$ s, as the grid is refined from grid refinement level 1 to level 4. (e–h) In the L^∞ norm, approximately second-order convergence is seen before cloud formation time of $t \approx 120$ s, after which the convergence rate is degraded.

Here we considered such interface corrections for several components of the full moist system: time stepping, pressure Poisson equation, and buoyancy derivative. For each of these individual components, second-order accuracy was achieved in both the L^2 (i.e., root mean square) and L^∞ (i.e., maximum) norms. Then, when the full moist system was considered, approximately second-order convergence was seen in the L^2 norm, whereas the convergence rate was degraded in the L^∞ (i.e., maximum) norm. Hence it may in fact be possible to observe second order convergence for certain error metrics.

It is possible that rigorous mathematical analysis could help to understand the lack of smoothness in solutions and implications for numerical accuracy. Some relevant literature was discussed at the end of Section 5. Further research in this direction would be valuable.

While the investigation here was focused on finite difference methods, we note that the same issues with cloud edge arise in other numerical methods as well (e.g., spectral methods, finite volume methods, finite element methods, discontinuous Galerkin methods, etc.). Interfaces require special treatment in order to achieve second-order or higher-order accuracy. Some additional literature was discussed in Section 1 and at the end of Section 2.2.

Future work could also consider topics such as the influence of order of accuracy (e.g., 1st vs. 2nd order) on turbulent simulations of the cloudy boundary layer, and statistical quantities such as vertical moisture flux. Turbulent statistical properties provide a different way to measure the behavior of a numerical method. In contrast, here we considered convergence analysis, which can assess the errors that occur when an initial condition is well-resolved and it evolves in time for an order-one amount of time into a solution that is also well-resolved. Other scenarios would also be interesting to consider, such as longer-time simulations when turbulence develops and/or when the turbulent solutions are marginally resolved.

Also, additional cloud processes, such as precipitation and ice microphysics, would be interesting to consider in the future. In order to illustrate the non-smoothness at cloud edge more transparently, a simplified CRM under the Boussinesq approximation was investigated here. The results here with the idealized CRM should carry over to comprehensive CRMs since they share the same type of non-smoothness due to phase changes and clouds.

Appendix A

Here, we provide the details of the exact solution given in Equation 22. The idea is to find the solution in saturated and unsaturated regions separately, and then piece them together at the phase interface.

We will assume that $N_u^2 > N_s^2$ and $a > 0$. In unsaturated regions, where $b_u > b_s$, the equation for $\dot{b}_u$ is given by

$$\ddot{b}_u + N_u^2 b_u = 0, \quad (\text{A1})$$

where we use notation of $\dot{b}_u$ for db_u/dt and $\ddot{b}_u$ for d^2b_u/dt^2 . The solution of this equation is

$$b_u = c_1 \sin(N_u t) + c_2 \cos(N_u t). \quad (\text{A2})$$

In saturated regions, where $b_u < b_s$, straightforward algebra shows that the equation for b_u can be written as

$$\ddot{b}_u + N_s^2 b_u = N_u^2 N_s^2 M, \quad (\text{A3})$$

where M is defined by the equation

$$M = \frac{b_u}{N_u^2} - \frac{b_s}{N_s^2}, \quad (\text{A4})$$

and $dM/dt = 0$. If the initial conditions are $b_u(0) = b_s(0) = 0$, then $M = 0$ for all t , and Equation A3 becomes

$$\ddot{b}_u + N_s^2 b_u = 0, \quad (\text{A5})$$

the solution of which is

$$b_u = d_1 \sin(N_s t) + d_2 \cos(N_s t). \quad (\text{A6})$$

The initial conditions on the derivatives of the buoyancy variables are $\dot{b}_u(0) = -N_u^2 w(0) = -N_u^2 a$ and $\dot{b}_s(0) = -N_s^2 w(0) = -N_s^2 a$. As we are assuming that $N_u^2 > N_s^2$ and $a > 0$, it follows that $\dot{b}_u(0) < \dot{b}_s(0) < 0$, which in turn implies that $b_u - b_s < 0$ for $t > 0$ initially. The solution will therefore first enter the saturated region, and be given by Equation A6, as can be seen in the example shown in Figure 2. The solution will then enter the unsaturated phase at a later time, and be given by Equation A2. The particular exact solution for b_u that satisfies the initial conditions, and whose derivative is continuous, is given by

$$b_u(t) = \begin{cases} -\frac{N_u^2}{N_s} a \sin(N_s(t - n\mathcal{T})) & \text{if } t \in \left[n\mathcal{T}, n\mathcal{T} + \frac{\pi}{N_s}\right], \\ aN_u \sin\left(N_u\left(t - n\mathcal{T} - \frac{\pi}{N_s}\right)\right) & \text{if } t \in \left[n\mathcal{T} + \frac{\pi}{N_s}, (n+1)\mathcal{T}\right], \end{cases} \quad (\text{A7})$$

where the oscillation period is

$$\mathcal{T} = \frac{\pi}{N_u} + \frac{\pi}{N_s}, \quad (\text{A8})$$

and where $n = 0, 1, 2, \dots$. The solution oscillates between saturated and unsaturated regions, with phase changes occurring at times $t = n\mathcal{T}$ and $t = n\mathcal{T} + \pi/N_s$. The inclusion of the phase shifts in Equation A7 is necessary to ensure that the solution is valid for all times $t > 0$. We can now solve for the vertical velocity using the equation $\dot{w} = \max(b_u, b_s)$. Since $\dot{M} = 0$ and $M = 0$, we know that

$$b_s = \frac{N_s^2}{N_u^2} b_u, \quad (\text{A9})$$

so that the equation for w is either $\dot{w} = b_u$ or $\dot{w} = (N_s^2/N_u^2) b_u$, depending on the phase. The vertical velocity is then given by the expression

$$w(t) = \begin{cases} a \cos(N_s(t - n\mathcal{T})) & \text{if } t \in \left[n\mathcal{T}, n\mathcal{T} + \frac{\pi}{N_s}\right], \\ -a \cos\left(N_u\left(t - n\mathcal{T} - \frac{\pi}{N_s}\right)\right) & \text{if } t \in \left[n\mathcal{T} + \frac{\pi}{N_s}, (n+1)\mathcal{T}\right]. \end{cases} \quad (\text{A10})$$

We now verify directly that the solution for $w(t)$ is continuous and differentiable for all times $t > 0$. This amounts to checking that $w(t)$ and $\dot{w}(t)$ are continuous at the phase change points $t = n\mathcal{T}$ and $t = n\mathcal{T} + \pi/N_s$. Evaluating left and right-hand side limits at $t = n\mathcal{T}$ yields.

$$\lim_{t \rightarrow (n\mathcal{T})^+} w(t) = \lim_{t \rightarrow (n\mathcal{T})^+} a \cos(N_s(t - n\mathcal{T})) \quad (\text{A11})$$

$$= a \cos(0) \quad (\text{A12})$$

$$= a \quad (\text{A13})$$

and

$$\lim_{t \rightarrow (n\mathcal{T})^-} w(t) = \lim_{t \rightarrow (n\mathcal{T})^-} -a \cos(N_u(t - (n-1)\mathcal{T} - \pi/N_s)) \quad (\text{A14})$$

$$= -a \cos(N_u(n\mathcal{T} - (n-1)\mathcal{T} - \pi/N_s)) \quad (\text{A15})$$

$$= -a \cos(\pi) \quad (\text{A16})$$

$$= a. \quad (\text{A17})$$

Thus, $w(t)$ is continuous at $t = n\mathcal{T}$. To verify continuity at $t = n\mathcal{T} + \pi/N_s$, we evaluate left and right-hand side limits of $w(t)$ at $t = n\mathcal{T} + \pi/N_s$.

$$\lim_{t \rightarrow (n\mathcal{T} + \pi/N_s)^+} w(t) = \lim_{t \rightarrow (n\mathcal{T} + \pi/N_s)^+} -a \cos(N_u(t - n\mathcal{T} - \pi/N_s)) \quad (\text{A18})$$

$$= -a \cos(N_u(n\mathcal{T} + \pi/N_s - n\mathcal{T} - \pi/N_s)) \quad (\text{A19})$$

$$= -a \cos(0) \quad (\text{A20})$$

$$= -a \quad (\text{A21})$$

and

$$\lim_{t \rightarrow (nT + \pi/N_s)^-} w(t) = \lim_{t \rightarrow (nT + \pi/N_s)^-} a \cos(N_u(t - nT)) \quad (\text{A22})$$

$$= a \cos(N_u(nT + \pi/N_s - nT)) \quad (\text{A23})$$

$$= a \cos(\pi) \quad (\text{A24})$$

$$= -a. \quad (\text{A25})$$

Thus, $w(t)$ is continuous at $t = nT + \pi/N_s$, and therefore $w(t)$ is continuous for all $t > 0$. Carrying out similar calculations for $\dot{w}$ shows that

$$\lim_{t \rightarrow (nT)^+} \dot{w}(t) = \lim_{t \rightarrow (nT)^-} \dot{w}(t) = 0, \quad (\text{A26})$$

$$\lim_{t \rightarrow (nT + \pi/N_s)^+} \dot{w}(t) = \lim_{t \rightarrow (nT + \pi/N_s)^-} \dot{w}(t) = 0. \quad (\text{A27})$$

It follows that $\dot{w}(t)$ is continuous for all $t > 0$. Similar calculations can be done to show that b_u and b_s are continuous and have continuous first derivatives for $t > 0$.

Appendix B

In this section, we describe the details of the local error analysis for two-stage Runge-Kutta methods. In particular we show that the one-step error of two-stage Runge-Kutta methods are second order accurate even when the solution changes phase on the time interval (t_i, t_{i+1}) . We can write Equations 17a–17c in the form

$$\frac{d}{dt} \begin{pmatrix} w \\ b_u \\ b_s \end{pmatrix} = \begin{pmatrix} f_1(t, w, b_u, b_s) \\ f_2(t, w, b_u, b_s) \\ f_3(t, w, b_u, b_s) \end{pmatrix}, \quad (\text{B1})$$

where $f_1 = \max(b_u, b_s)$, $f_2 = -N_u^2 w$, and $f_3 = -N_s^2 w$. We present the details of two-stage Runge-Kutta update for w ; the other variables are treated similarly. The two-stage Runge-Kutta scheme for the vertical velocity can be written in the form of a one-step method as

$$w_{i+1} = w_i + \frac{\Delta t}{2} \max(b_{u_i}, b_{s_i}) + \frac{\Delta t}{2} \max(b_{u_i} - \Delta t N_u^2 w_i, b_{s_i} - \Delta t N_s^2 w_i). \quad (\text{B2})$$

The one-step error for w at time t_i is then given by the expression

$$w(t_{i+1}) - w(t_i) - \frac{\Delta t}{2} \max(b_u(t_i), b_s(t_i)) - \frac{\Delta t}{2} \max(b_u(t_i) - \Delta t N_u^2 w(t_i), b_s(t_i) - \Delta t N_s^2 w(t_i)), \quad (\text{B3})$$

where $w(t_i)$, $b_u(t_i)$ and $b_s(t_i)$ denote the exact solution to (82) at time t_i . Now assume that there is a phase change at time t_I , that is to say, $b_u(t_I) = b_s(t_I)$, and that $t_I = t_i + \alpha \Delta t$ for some $\alpha > 0$. Taylor expansions then yield the expressions

$$w(t_i) = w(t_I^-) - \alpha \Delta t \dot{w}(t_I^-) + \frac{\alpha^2 \Delta t^2}{2} \ddot{w}(t_I^-) + \mathcal{O}(\Delta t^3), \quad (\text{B4})$$

$$w(t_{i+1}) = w(t_i^+) + (1 - \alpha)\Delta t \dot{w}(t_i^+) + \frac{(1 - \alpha)^2 \Delta t^2}{2} \ddot{w}(t_i^+) + \mathcal{O}(\Delta t^3). \quad (\text{B5})$$

By continuity of w and $\dot{w}$, we know that $w(t_i^-) = w(t_i^+)$ and $\dot{w}(t_i^-) = \dot{w}(t_i^+)$. After subtracting Equations B4 and B5 (and sub algebra), we have

$$w(t_{i+1}) - w(t_i) = \Delta t \dot{w}(t_i^+) + \frac{(1 - \alpha)^2 \Delta t^2}{2} \ddot{w}(t_i^+) - \frac{\alpha^2 \Delta t^2}{2} \ddot{w}(t_i^-) + \mathcal{O}(\Delta t^3). \quad (\text{B6})$$

Now we focus on the other two terms in Equation B3. From Taylor expanding about t_i^- , we have

$$b_u(t_i) = b_u(t_i^-) + \alpha N_u^2 w(t_1), \quad (\text{B7})$$

$$b_s(t_i) = b_s(t_i^-) + \alpha N_s^2 w(t_2), \quad (\text{B8})$$

where $t_1, t_2 \in (x_i, x_i^-)$. Then we have

$$\max(b_u(t_i), b_s(t_i)) = \max(b_u(t_i^-) + \alpha \Delta t N_u^2 w(t_1), b_s(t_i^-) + \alpha \Delta t N_s^2 w(t_2)) \quad (\text{B9})$$

$$= \max(b_u(t_i^-) + \alpha \Delta t N_u^2 w(t_1), b_u(t_i^-) + \alpha \Delta t N_s^2 w(t_2)), \quad (\text{B10})$$

where we have used the fact that $b_u(t_i^-) = b_s(t_i^{-1})$. Continuing, we have

$$\max(b_u(t_i^-) + \alpha \Delta t N_u^2 w(t_1), b_u(t_i^-) + \alpha \Delta t N_s^2 w(t_2)) = b_u(t_i^-) + \max(\alpha \Delta t N_u^2 w(t_1), \alpha \Delta t N_s^2 w(t_2)) \quad (\text{B11})$$

$$= b_u(t_i^-) + \alpha \Delta t \max(N_u^2 w(t_1), N_s^2 w(t_2)). \quad (\text{B12})$$

Using the fact that $\dot{w}(t_i^-) = b_u(t_i^{-1})$, we then have

$$\max(b_u(t_i^-) + \alpha \Delta t N_u^2 w(t_1), b_u(t_i^-) + \alpha \Delta t N_s^2 w(t_2)) = \dot{w}(t_i^-) + \alpha \Delta t \max(N_u^2 w(t_1), N_s^2 w(t_2)). \quad (\text{B13})$$

Now we consider the last term in Equation B3:

$$\max(b_u(t_i) - \Delta t N_u^2 w(t_i), b_s(t_i) - N_s^2 w(t_i)). \quad (\text{B14})$$

Taylor expanding about t_i^- we have

$$w(t_i) = w(t_i^-) - \alpha \Delta t \dot{w}(t_3), \quad (\text{B15})$$

where $t_3 \in (t_i, t_i^-)$. We also have the expressions:

$$b_u(t_i) - \Delta t N_u^2 w(t_i) = b_u(t_i^-) - \alpha \Delta t \dot{b}_u(t_i^-) - \Delta t N_u^2 w(t_i^-) + \alpha \Delta t^2 N_u^2 \dot{w}(t_3), \quad (\text{B16})$$

$$b_s(t_i) - \Delta t N_s^2 w(t_i) = b_s(t_i^-) - \alpha \Delta t \dot{b}_s(t_i^-) - \Delta t N_s^2 w(t_i^-) + \alpha \Delta t^2 N_s^2 \dot{w}(t_3). \quad (\text{B17})$$

Therefore,

$$\begin{aligned} \max(b_u(t_i) - \Delta t N_u^2 w(t_i), b_s(t_i) - N_s^2 w(t_i)) &= \max(b_u(t_i^-) - \alpha \Delta t \dot{b}_u(t_i^-) - \Delta t N_u^2 w(t_i^-) + \alpha \Delta t^2 N_u^2 \dot{w}(t_3), \\ &\quad b_s(t_i^-) - \alpha \Delta t \dot{b}_s(t_i^-) - \Delta t N_s^2 w(t_i^-) + \alpha \Delta t^2 N_s^2 \dot{w}(t_3)) \end{aligned} \quad (\text{B18})$$

$$= \max(b_u(t_l^-) - \alpha \Delta t \dot{b}_u(t_l^-) - \Delta t N_u^2 w(t_l^-) + \alpha \Delta t^2 N_u^2 \dot{w}(t_3)) \quad (B19)$$

$$b_u(t_l^-) - \alpha \Delta t \dot{b}_u(t_l^-) - \Delta t N_s^2 w(t_l^-) + \alpha \Delta t^2 N_s^2 \dot{w}(t_3)) \\ = b_u(t_l^-) + \max(-\alpha \Delta t \dot{b}_u(t_l^-) - \Delta t N_u^2 w(t_l^-) + \alpha \Delta t^2 N_u^2 \dot{w}(t_3) \\ - \alpha \Delta t \dot{b}_s(t_l^-) - \Delta t N_s^2 w(t_l^-) + \alpha \Delta t^2 N_s^2 \dot{w}(t_3)) \quad (B20)$$

$$= b_u(t_l^-) + \Delta t \max(-\alpha \dot{b}_u(t_l^-) - N_u^2 w(t_l^-) + \alpha \Delta t N_u^2 \dot{w}(t_3) \\ - \alpha \dot{b}_s(t_l^-) - N_s^2 w(t_l^-) + \alpha \Delta t N_s^2 \dot{w}(t_3)). \quad (B21)$$

Since $b_u(t_l^-) = \dot{w}(t_l^{-1})$, the above simplifies to

$$\max(b_u(t_i) - \Delta t N_u^2 w(t_i), b_s(t_i) - N_s^2 w(t_i)) = \dot{w}(t_l^-) + \Delta t \max(-\alpha \dot{b}_u(t_l^-) - N_u^2 w(t_l^-) + \alpha \Delta t N_u^2 \dot{w}(t_3) \\ - \alpha \dot{b}_s(t_l^-) - N_s^2 w(t_l^-) + \alpha \Delta t N_s^2 \dot{w}(t_3)). \quad (B22)$$

Combining Equations B13 and B22 shows that

$$\frac{\Delta t}{2} \max(b_u(t_i), b_s(t_i)) + \frac{\Delta t}{2} \max(b_u(t_i) - \Delta t N_u^2 w(t_i), b_s(t_i) - N_s^2 w(t_i)) = \Delta t \dot{w}(t_l^-) + \mathcal{O}(\Delta t^2). \quad (B23)$$

Subtracting Equation B23 from Equation B6, and using the fact that $\dot{w}(t_l^-) = \dot{w}(t_l^+)$, shows that the one-step error is $\mathcal{O}(\Delta t^2)$

Conflict of Interest

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

Data Availability Statement

Data/software/code for the numerical experiments in this study is available at <https://doi.org/10.5281/zenodo.15624691> (Marsico et al., 2025). An overview is provided in the README.pdf file that is located there.

Acknowledgments

The authors thank three anonymous reviewers for helpful comments that improved the presentation of the paper. The research of S.N.S. and S.D. is partially supported by Grant NSF AGS 2326631.

References

- Arakawa, A. (2004). The cumulus parameterization problem: Past, present, and future. *Journal of Climate*, 17(13), 2493–2525. [https://doi.org/10.1175/1520-0442\(2004\)017<2493:ratcp>2.0.co;2](https://doi.org/10.1175/1520-0442(2004)017<2493:ratcp>2.0.co;2)
- Barrett, A. I., Wellmann, C., Seifert, A., Hoose, C., Vogel, B., & Kunz, M. (2019). One step at a time: How model time step significantly affects convection-permitting simulations. *Journal of Advances in Modeling Earth Systems*, 11(13), 641–658. <https://doi.org/10.1029/2018MS001418>
- Bell, J. B., Colella, P., & Glaz, H. M. (1989). A second-order projection method for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 85(2), 257–283. [https://doi.org/10.1016/0021-9991\(89\)90151-4](https://doi.org/10.1016/0021-9991(89)90151-4)
- Bell, J. B., & Marcus, D. L. (1992). A second-order projection method for variable-density flows. *Journal of Computational Physics*, 101(2), 334–348. [https://doi.org/10.1016/0021-9991\(92\)90011-m](https://doi.org/10.1016/0021-9991(92)90011-m)
- Bendall, T. M., Gibson, T. H., Shipton, J., Cotter, C. J., & Shipway, B. (2020). A compatible finite-element discretisation for the moist compressible Euler equations. *Quarterly Journal of the Royal Meteorological Society*, 146(732), 3187–3205. <https://doi.org/10.1002/qj.3841>
- Bretherton, C. S. (1987). A theory for nonprecipitating moist convection between two parallel plates. Part I: Thermodynamics and “Linear” solutions. *Journal of the Atmospheric Sciences*, 44(14), 1809–1827. [https://doi.org/10.1175/1520-0469\(1987\)044<1809:ATFNMC>2.0.CO;2](https://doi.org/10.1175/1520-0469(1987)044<1809:ATFNMC>2.0.CO;2)
- Bryan, G. H., & Fritsch, J. M. (2002). A benchmark simulation for moist nonhydrostatic numerical models. *Monthly Weather Review*, 130(12), 2917–2928. [https://doi.org/10.1175/1520-0493\(2002\)130<2917:ABSFMN>2.0.CO;2](https://doi.org/10.1175/1520-0493(2002)130<2917:ABSFMN>2.0.CO;2)
- Cao, Y., Hamouda, M., Temam, R., Tribbia, J., & Wang, X. (2018). The equations of the multi-phase humid atmosphere expressed as a quasi variational inequality. *Nonlinearity*, 31(10), 4692–4723. <https://doi.org/10.1088/1361-6544/aad525>
- Chertock, A., Kurganov, A., Lukáčová-Medvid'ová, M., Spichtinger, P., & Wiebe, B. (2019). Stochastic Galerkin method for cloud simulation. *Mathematics of Climate and Weather Forecasting*, 5(1), 65–106. <https://doi.org/10.1515/mcwf-2019-0005>
- Chien, M.-H., Pauluis, O. M., & Almgren, A. S. (2022). Hurricane-like vortices in conditionally unstable moist convection. *Journal of Advances in Modeling Earth Systems*, 14(7), e2021MS002846. <https://doi.org/10.1029/2021MS002846>
- Chorin, A. J. (1968). Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22(104), 745–762. <https://doi.org/10.1090/s0025-5718-1968-0242392-2>
- Cuijpers, J., & Duynkerke, P. (1993). Large eddy simulation of trade wind cumulus clouds. *Journal of the Atmospheric Sciences*, 50(23), 3894–3908. [https://doi.org/10.1175/1520-0469\(1993\)050<3894:lesotw>2.0.co;2](https://doi.org/10.1175/1520-0469(1993)050<3894:lesotw>2.0.co;2)

- De Sousa, F., Mangiacavchi, N., Nonato, L. G., Castelo, A., Tomé, M. F., Ferreira, V. G., et al. (2004). A front-tracking/front-capturing method for the simulation of 3D multi-fluid flows with free surfaces. *Journal of Computational Physics*, 198(2), 469–499. <https://doi.org/10.1016/j.jcp.2004.01.032>
- Doppler, S., Lederer, P. L., Schöberl, J., & von Wahl, H. (2024). A discontinuous Galerkin approach for atmospheric flows with implicit condensation. *Journal of Computational Physics*, 499, 112713. <https://doi.org/10.1016/j.jcp.2023.112713>
- Durran, D. R. (1999). *Numerical methods for wave equations in geophysical fluid dynamics*. Springer-Verlag.
- Emanuel, K. (1994). *Atmospheric convection*. Oxford University Press.
- Frierson, D. M. W., Majda, A. J., & Pauluis, O. M. (2004). Large scale dynamics of precipitation fronts in the tropical atmosphere: A novel relaxation limit. *Communications in Mathematical Sciences*, 2(4), 591–626. <https://doi.org/10.4310/cms.2004.v2.n4.a3>
- Gaberšek, S., Giraldo, F. X., & Doyle, J. D. (2012). Dry and moist idealized experiments with a two-dimensional spectral element model. *Monthly Weather Review*, 140(10), 3163–3182. <https://doi.org/10.1175/MWR-D-11-00144.1>
- Grabowski, W. W., & Morrison, H. (2008). Toward the mitigation of spurious cloud-edge supersaturation in cloud models. *Monthly Weather Review*, 136(3), 1224–1234. <https://doi.org/10.1175/2007mwr2283.1>
- Grabowski, W. W., & Smolarkiewicz, P. K. (1990). Monotone finite-difference approximations to the advection-condensation problem. *Monthly Weather Review*, 118(10), 2082–2098. [https://doi.org/10.1175/1520-0493\(1990\)118<2082:mfdatt>2.0.co;2](https://doi.org/10.1175/1520-0493(1990)118<2082:mfdatt>2.0.co;2)
- Grabowski, W. W., & Smolarkiewicz, P. K. (1996). Two-time-level semi-lagrangian modeling of precipitating clouds. *Monthly Weather Review*, 124(3), 487–497. [https://doi.org/10.1175/1520-0493\(1996\)124<0487:ttlslm>2.0.co;2](https://doi.org/10.1175/1520-0493(1996)124<0487:ttlslm>2.0.co;2)
- Hellrung Jr, J. L., Wang, L., Sifakis, E., & Teran, J. M. (2012). A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions. *Journal of Computational Physics*, 231(4), 2015–2048. <https://doi.org/10.1016/j.jcp.2011.11.023>
- Hernandez-Duenas, G., Majda, A. J., Smith, L. M., & Stechmann, S. N. (2013). Minimal models for precipitating turbulent convection. *Journal of Fluid Mechanics*, 717, 576–611. <https://doi.org/10.1017/jfm.2012.597>
- Hittmeir, S., Klein, R., Li, J., & Titi, E. S. (2020). Global well-posedness for the primitive equations coupled to nonlinear moisture dynamics with phase changes. *Nonlinearity*, 33(7), 3206–3236. <https://doi.org/10.1088/1361-6544/ab834f>
- Khairoutdinov, M. F., & Randall, D. A. (2003). Cloud resolving modeling of the ARM summer 1997 IOP: Model formulation, results, uncertainties, and sensitivities. *Journal of the Atmospheric Sciences*, 60(4), 607–625. [https://doi.org/10.1175/1520-0469\(2003\)060<0607:CRMOTA>2.0.CO;2](https://doi.org/10.1175/1520-0469(2003)060<0607:CRMOTA>2.0.CO;2)
- Khouider, B., & Majda, A. J. (2005). A non-oscillatory balanced scheme for an idealized tropical climate model: Part I: Algorithm and validation. *Theoretical and Computational Fluid Dynamics*, 19(5), 331–354. <https://doi.org/10.1007/s00162-005-0170-8>
- Khouider, B., & Majda, A. J. (2005). A non-oscillatory balanced scheme for an idealized tropical climate model: Part II: Nonlinear coupling and moisture effects. *Theoretical and Computational Fluid Dynamics*, 19(5), 355–375. <https://doi.org/10.1007/s00162-005-0171-7>
- Kuo, H. (1961). Convection in conditionally unstable atmosphere. *Tellus*, 13(4), 441–459. <https://doi.org/10.3402/tellusa.v13i4.9516>
- Lauritzen, P. H., Kevlahan, N. K.-R., Toniazzo, T., Eldred, C., Dubos, T., Gassmann, A., et al. (2022). Reconciling and improving formulations for thermodynamics and conservation principles in Earth System Models (ESMs). *Journal of Advances in Modeling Earth Systems*, 14(9), e2022MS003117. <https://doi.org/10.1029/2022MS003117>
- Lee, L., & LeVeque, R. J. (2003). An immersed interface method for incompressible Navier–Stokes equations. *SIAM Journal on Scientific Computing*, 25(3), 832–856. <https://doi.org/10.1137/S1064827502414060>
- LeVeque, R. J. (2007). *Society for Industrial and Applied Mathematics*. Finite difference methods for ordinary and partial differential equations. <https://doi.org/10.1137/1.9780898717839>
- LeVeque, R. J., & Li, Z. (1994). The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31(4), 1019–1044. <https://doi.org/10.1137/0731054>
- Li, Z., & Lai, M.-C. (2001). The immersed interface method for the Navier–Stokes equations with singular forces. *Journal of Computational Physics*, 171(2), 822–842. <https://doi.org/10.1006/jcph.2001.6813>
- Margolin, L., Reisner, J. M., & Smolarkiewicz, P. K. (1997). Application of the volume-of-fluid method to the advection–condensation problem. *Monthly Weather Review*, 125(9), 2265–2273. [https://doi.org/10.1175/1520-0493\(1997\)125<2265:aotvo>2.0.co;2](https://doi.org/10.1175/1520-0493(1997)125<2265:aotvo>2.0.co;2)
- Marsico, D. H., Du, S., & Stechmann, S. N. (2025). Code for manuscript, can second-order numerical accuracy be achieved for moist atmospheric dynamics with non-smoothness at cloud edge? *Zenodo*. <https://doi.org/10.5281/zenodo.13316895>
- Marsico, D. H., Smith, L. M., & Stechmann, S. N. (2019). Energy decompositions for moist Boussinesq and anelastic equations with phase changes. *Journal of the Atmospheric Sciences*, 76(11), 3569–3587. <https://doi.org/10.1175/JAS-D-19-0080.1>
- Norman, M. R., Eldred, C., & Gopalakrishnan Meena, M. (2023). Investigating inherent numerical stabilization for the moist, compressible, non-hydrostatic Euler equations on collocated grids. *Journal of Advances in Modeling Earth Systems*, 15(10), e2023MS003732. <https://doi.org/10.1029/2023MS003732>
- Oevermann, M., Scharfenberg, C., & Klein, R. (2009). A sharp interface finite volume method for elliptic equations on Cartesian grids. *Journal of Computational Physics*, 228(14), 5184–5206. <https://doi.org/10.1016/j.jcp.2009.04.018>
- Ooyama, K. V. (1990). A thermodynamic foundation for modeling the moist atmosphere. *Journal of the Atmospheric Sciences*, 47(21), 2580–2593. [https://doi.org/10.1175/1520-0469\(1990\)047<2580:ATFFMT>2.0.CO;2](https://doi.org/10.1175/1520-0469(1990)047<2580:ATFFMT>2.0.CO;2)
- Pauluis, O. (2008). Thermodynamic consistency of the anelastic approximation for a moist atmosphere. *Journal of the Atmospheric Sciences*, 65(8), 2719–2729. <https://doi.org/10.1175/2007jas2475.1>
- Pauluis, O., & Garner, S. (2006). Sensitivity of radiative–convective equilibrium simulations to horizontal resolution. *Journal of the Atmospheric Sciences*, 63(7), 1910–1923. <https://doi.org/10.1175/jas3705.1>
- Pauluis, O., & Schumacher, J. (2010). Idealized moist Rayleigh–Bénard convection with piecewise linear equation of state. *Communications in Mathematical Sciences*, 8(1), 295–319. <https://doi.org/10.4310/cms.2010.v8.n1.a15>
- Pauluis, O., & Schumacher, J. (2011). Self-aggregation of clouds in conditionally unstable moist convection. *Proceedings of the National Academy of Sciences*, 108(31), 12623–12628. <https://doi.org/10.1073/pnas.1102339108>
- Peskin, C. S. (2002). The immersed boundary method. *Acta Numerica*, 11, 479–517. <https://doi.org/10.1017/s0962492902000077>
- Remond-Tiedrez, A., Smith, L. M., & Stechmann, S. N. (2024). A nonlinear elliptic PDE from atmospheric science: Well-posedness and regularity at cloud edge. *Journal of Mathematical Fluid Mechanics*, 26(2), 30. <https://doi.org/10.1007/s00021-024-00865-4>
- Smith, L. M., & Stechmann, S. N. (2017). Precipitating quasigeostrophic equations and potential vorticity inversion with phase changes. *Journal of the Atmospheric Sciences*, 74(10), 3285–3303. <https://doi.org/10.1175/JAS-D-17-0023.1>
- Sridhar, A., Tissaoui, Y., Marras, S., Shen, Z., Kawczynski, C., Byrne, S., et al. (2022). Large-eddy simulations with ClimateMachine v0.2.0: A new open-source code for atmospheric simulations on GPUs and CPUs. *Geoscientific Model Development*, 15(15), 6259–6284. <https://doi.org/10.5194/gmd-15-6259-2022>

- Stechmann, S. N., & Majda, A. J. (2006). The structure of precipitation fronts for finite relaxation time. *Theoretical and Computational Fluid Dynamics*, 20(5–6), 377–404. <https://doi.org/10.1007/s00162-006-0014-1>
- Stevens, B. (2005). Atmospheric moist convection. *Annual Review of Earth and Planetary Sciences*, 33(33), 605–643. <https://doi.org/10.1146/annurev.earth.33.092203.122658>
- Stevens, B., Walko, R. L., Cotton, W. R., & Feingold, G. (1996). The spurious production of cloud-edge supersaturations by Eulerian models. *Monthly Weather Review*, 124(5), 1034–1041. [https://doi.org/10.1175/1520-0493\(1996\)124<1034:tspace>2.0.co;2](https://doi.org/10.1175/1520-0493(1996)124<1034:tspace>2.0.co;2)
- Sussman, M., Fatemi, E., Smereka, P., & Osher, S. (1998). An improved level set method for incompressible two-phase flows. *Computers and Fluids*, 27(5–6), 663–680. [https://doi.org/10.1016/s0045-7930\(97\)00053-4](https://doi.org/10.1016/s0045-7930(97)00053-4)
- Sussman, M., Smith, K. M., Hussaini, M. Y., Ohta, M., & Zhi-Wei, R. (2007). A sharp interface method for incompressible two-phase flows. *Journal of Computational Physics*, 221(2), 469–505. <https://doi.org/10.1016/j.jcp.2006.06.020>
- Temam, R. (1969). Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (II). *Archive for Rational Mechanics and Analysis*, 33(5), 377–385. <https://doi.org/10.1007/bf00247696>
- Tissaoui, Y., Marras, S., Quaini, A., de Brangaca Alves, F. A., & Giraldo, F. X. (2023). A non-column based, fully unstructured implementation of Kessler's microphysics with warm rain using continuous and discontinuous spectral elements. *Journal of Advances in Modeling Earth Systems*, 15(3), e2022MS003283. <https://doi.org/10.1029/2022ms003283>
- Tzou, C.-N., & Stechmann, S. N. (2019). Simple second-order finite differences for elliptic PDEs with discontinuous coefficients and interfaces. *Communications in Applied Mathematics and Computational Science*, 14(2), 121–147. <https://doi.org/10.2140/camcos.2019.14-2>
- Vallis, G. K., Parker, D. J., & Tobias, S. M. (2019). A simple system for moist convection: The rainy–Benard model. *Journal of Fluid Mechanics*, 862, 162–199. <https://doi.org/10.1017/jfm.2018.954>
- Vogl, C. J., Wan, H., Zhang, S., Woodward, C. S., & Stinis, P. (2020). Improving time step convergence in an atmosphere model with simplified physics: Using mathematical rigor to avoid nonphysical behavior in a parameterization. *Journal of Advances in Modeling Earth Systems*, 12(10), e2019MS001974. <https://doi.org/10.1029/2019MS001974>
- Walsh, R. P. (2016). Interfacial numerics for the mathematical modeling of cloud edge dynamics. Retrieved from <https://summit.sfu.ca/item/16809>
- Walsh, R. P. (2021). *Boussinesq dynamics at a cloud edge: Theory and simulation (unpublished doctoral dissertation)*. Simon Fraser University. Retrieved from <https://summit.sfu.ca/item/34739>
- Wan, H., Woodward, C. S., Zhang, S., Vogl, C. J., Stinis, P., Gardner, D. J., et al. (2020). Improving time step convergence in an atmosphere model with simplified physics: The impacts of closure assumption and process coupling. *Journal of Advances in Modeling Earth Systems*, 12(10), e2019MS001982. <https://doi.org/10.1029/2019MS001982>
- Wang, Y., Zhang, G. J., Xie, S., Lin, W., Craig, G. C., Tang, Q., & Ma, H.-Y. (2021). Effects of coupling a stochastic convective parameterization with the Zhang–McFarlane scheme on precipitation simulation in the DOE E3SMv1.0 atmosphere model. *Geoscientific Model Development*, 14(3), 1575–1593. <https://doi.org/10.5194/gmd-14-1575-2021>
- Warming, R. F., & Hyett, B. (1974). The modified equation approach to the stability and accuracy analysis of finite-difference methods. *Journal of Computational Physics*, 14(2), 159–179. [https://doi.org/10.1016/0021-9991\(74\)90011-4](https://doi.org/10.1016/0021-9991(74)90011-4)
- Xu, S., & Wang, Z. J. (2006). An immersed interface method for simulating the interaction of a fluid with moving boundaries. *Journal of Computational Physics*, 216(2), 454–493. <https://doi.org/10.1016/j.jcp.2005.12.016>
- Xu, S., & Wang, Z. J. (2006). Systematic derivation of jump conditions for the immersed interface method in three-dimensional flow simulation. *SIAM Journal on Scientific Computing*, 27(6), 1948–1980. <https://doi.org/10.1137/040604960>
- Zarzycki, C. M., Jablonowski, C., Kent, J., Lauritzen, P. H., Nair, R., Reed, K. A., et al. (2019). DCMIP2016: The splitting supercell test case. *Geoscientific Model Development*, 12(3), 879–892. <https://doi.org/10.5194/gmd-12-879-2019>
- Zhang, S., Vogl, C. J., Larson, V. E., Bui, Q. M., Wan, H., Rasch, P. J., & Woodward, C. S. (2023). Removing numerical pathologies in a turbulence parameterization through convergence testing. *Journal of Advances in Modeling Earth Systems*, 15(5), e2023MS003633. <https://doi.org/10.1029/2023MS003633>
- Zhang, Y., Smith, L. M., & Stechmann, S. N. (2021). Effects of clouds and phase changes on fast-wave averaging: A numerical assessment. *Journal of Fluid Mechanics*, 920, A49. <https://doi.org/10.1017/jfm.2021.427>