A non-column based, fully unstructured implementation of Kessler's microphysics with warm rain using continuous and discontinuous spectral elements

Yassine Tissaoui[1], Simone Marras[2], Annalisa Quaini[3], Felipe De Braganca Alves[4], and Francis X Giraldo[5]

[1]New Jersey Institute of Technology
[2]Unknown
[3]University of Houston
[4]Naval Postgraduate School
[5]Naval Pastgraduate School

November 25, 2022

Abstract

Numerical weather prediction is pushing the envelope of grid resolution at local and global scales alike. Aiming to model topography with higher precision, a handful of articles introduced unstructured vertical grids and tested them for dry atmospheres. The next step towards effective high-resolution unstructured grids for atmospheric modeling requires that also microphysics is independent of any vertical columns, in contrast to what is ubiquitous across operational and research models. In this paper, we present a non-column based continuous and discontinuous spectral element implementation of Kessler's microphysics with warm rain as a first step towards fully unstructured atmospheric models.

We test the proposed algorithm against standard three-dimensional benchmarks for precipitating clouds and show that the results are comparable with those presented in the literature across all of the tested effective resolutions. While presented for both continuous and discontinuous spectral elements in this paper, the method that we propose can very easily be adapted to any numerical method utilized in other research and legacy codes.

A non-column based, fully unstructured implementation of Kessler's microphysics with warm rain using continuous and discontinuous spectral elements

Yassine Tissaoui[1], Simone Marras[1], Annalisa Quaini[2], Felipe A. V. de Brangaca Alves[3], Francis X. Giraldo[3]

[1]Department of Mechanical Engineering, New Jersey Institute of Technology, Newark, New Jersey
[2]Department of Mathematics, University of Houston, Houston, Texas
[3]Department of Applied Mathematics, Naval Post Graduate School, Monterey, California

Key Points:

- A non-column based spectral element approach to model precipitating clouds is presented.
- The approach can be easily applied to any microphysics scheme with and without precipitation.
- This approach will make full physics simulations of the atmosphere on unstructured grids possible.

Corresponding author: Yassine Tissaoui, yt277@njit.edu

Abstract

Numerical weather prediction is pushing the envelope of grid resolution at local and global scales alike. Aiming to model topography with higher precision, a handful of articles introduced unstructured vertical grids and tested them for dry atmospheres. The next step towards effective high-resolution unstructured grids for atmospheric modeling requires that also microphysics is independent of any vertical columns, in contrast to what is ubiquitous across operational and research models. In this paper, we present a non-column based continuous and discontinuous spectral element implementation of Kessler's microphysics with warm rain as a first step towards fully unstructured atmospheric models. We test the proposed algorithm against standard three-dimensional benchmarks for precipitating clouds and show that the results are comparable with those presented in the literature across all of the tested effective resolutions. While presented for both continuous and discontinuous spectral elements in this paper, the method that we propose can very easily be adapted to any numerical method utilized in other research and legacy codes.

Plain Language Summary

The earth climate is warming faster than ever. While climate models are the tool available to scientists to forecast its future evolution, they are biased by uncertainties that are, arguably, mostly embedded in the modeling of clouds. Thanks to the advent of exascale computing, a reduction of cloud modeling uncertainties can be expected by simulating clouds at higher and higher resolutions. While uniform high resolution across the whole domain is ideal, for computational efficiency reasons scientist are likely to increase the model resolution in some regions more than others not only in the horizontal direction —which is a standard approach— but also along the vertical direction. Grid refinement in the vertical direction, however, may lead to the loss of the vertical structure of the grid columns, affecting the usability of column-based physics packages that are used to model clouds and precipitation. To overcome this problem, we present an algorithm to solve the equations that model precipitating clouds along arbitrarily shaped grids in any spatial direction. This approach is advantageous from a modeling perspective as well as from a computational one because it allows full flexibility of the domain partitioning algorithms when hundreds of thousands of parallel processors are used.

1 Introduction

Exascale computing on hybrid architectures is expected to become available by the start of 2023. Massive parallelism will enable the use of very fine grids for computational simulations. This is especially attractive for climate and weather simulations as more physical processes will be resolved instead of parameterized. For example, the use of sufficiently refined meshes makes it possible for atmospheric models to resolve extreme precipitation events more precisely than usually aimed for nowadays (Iorio et al., 2004; Terai et al., 2018; Wehner et al., 2014; Atlas et al., 2005; Caldwell et al., 2019; Bacmeister et al., 2014). If highly refined meshes for climate and weather simulations are also unstructured, it is possible to heighten the resolution of topographical features, including those that have been classically smoothed for the purpose of stabilizing global climate models (Lauritzen et al., 2015). Poor topography resolution makes precise weather forecast challenging (Giorgi & Marinucci, 1996), especially in the vicinity of steep mountain ranges (Yamazaki et al., 2022) such as, for example, the Himalayan region. Better resolved topography and coastal boundaries have been shown to improve the accuracy of simulations involving orographic precipitation and sea breeze effects (Caldwell et al., 2019; Delworth et al., 2012; Duffy et al., 2003; Pope & Stratton, 2002; Love et al., 2011). This paper presents the first implementation of a method capable of solving the fully compressible Euler equations with moisture, cloud formation, and warm rain on three-dimensional fully unstructured grids. It aims to show that it is possible to effectively implement a traditionally column-reliant parameterization on vertically unstructured meshes.

Despite the fact that there has been interest in using unstructured grids since the 1960s (Nikiforakis, 2009), most of the operational and research weather forecast models are constrained by vertically structured and column-based grids, even in the cases when non-structured discretizations are used in the horizontal direction. While horizontally unstructured meshes are often utilized (e.g. Danabasoglu et al. (2020); Dennis et al. (2012)), vertically unstructured grids are not. This is due to the column constraints imposed by the microphysics packages that have been historically used.

The first two atmospheric research models to adopt unstructured grids in the vertical direction were presented by Aubry et al. (2010) and Smolarkiewicz et al. (2013), with Szmelter et al. (2015) extending the latter to unstructured tetrahedral grids in 2015. At the time of writing this article, the latest in this series of efforts was published by Li et al. (2021). All of them demonstrate that the use of unstructured grids combined with adaptive mesh refinement reduces the numerical errors for dry mountain waves problems with steep orography, even at high resolutions. Large numerical errors when using structured grids to represent steep topography are a well known problem summarized by, e.g., Baldauf (2021), which shows that simulations run with the COSMO model (COSMO, 1998) break down with slopes larger than approximately 30 degrees. The choice of structured grids is motivated by the fact that the inclusion of microphysical processes has typically relied on a column-based, vertically structured implementation. Ever since the 1960s and 1970s when some of the first simulations of clouds and precipitation were performed utilizing microphysical parametrizations (Klemp & Wilhelmson, 1978; Kessler, 1969; Soong & Ogura, 1973; Weisman & Klemp, 1982), the implementation of these parameterizations has always relied on column-based grids. Although interpolation from the native grid to a physics grid is usually required, the native grid in all of the operational and research models depends on a column-based structure.

This paper presents a fully unstructured discretization of the compressible Euler equations with moisture to model clouds and precipitation. To support non-column based precipitation, we approximated the transport equation governing precipitation by means of the same approximation of the underlying dynamics model (i.e., the compressible Euler equations). To achieve this, we modified the Kessler's microphysics implementation in the Nonhydrostatic Unified Model of the Atmosphere (NUMA) (Kelly & Giraldo, 2012). In this way, we leverage the natural unstructured nature of the element-based Galerkin discretization (Giraldo, 2020) on which NUMA relies. We test the new implementation for both continuous and discontinuous elements (e.g., see (D. B. Abdi & Giraldo, 2017) for how this can be achieved in the same source code). Other models that use either continuous or discontinuous spectral elements for atmospheric flows are, e.g., CESM2 (Danabasoglu et al., 2020), E3SM(Caldwell et al., 2019), both via the CAM-SE dycore (Dennis et al., 2012), and ClimateMachine (Sridhar et al., 2022).

We show that with a simple modification of the Kessler precipitation routine, the spectral element method is capable of simulating rain precipitation through sedimentation on fully unstructured grids that do not rely on the vertical columns of a Cartesian grid. This is done in the typical spectral/finite element fashion of solving the local equations of motion on a reference element before projecting the local solution back to the physical space. This makes it possible to solve the equations of motion without any regard for the type of grid (structured or unstructured). The only constraint is that the solution quality will depend on the accuracy of the metric terms used to map the physical elements to the reference element (Giraldo, 2020; Nelson et al., 2016). We test this method in 3D by performing several squall lines (Rotunno et al., 1988; Weisman et al., 1988) and supercell (Skamarock et al., 2012) simulations. We show that this method is able to produce results comparable to those available in the literature. This work will help lead the way towards moist-air simulations of flow over steep orography using unstructured grids, and possibly both horizontal and vertical adaptive mesh refinement.

Finally, this approach has important consequences on the parallel efficiency for very high resolution atmospheric simulations because Message Passing Interface (MPI) is no longer limited to a column based subdivision of the domain, but will allow for a parallel load balancing decomposition in any direction.

The remainder of the paper is organized as follows. The governing equations are presented in § 2. The numerical approximation of the governing equations, including the details of the discretization of the rain equation and the algorithm for non-column-based rain sedimentation are presented in § 3. The numerical results are described in § 4. The conclusions are drawn in § 5.

2 Problem definition

Moist air is a mixture of dry air with density ρ, water vapor with density ρ_v, and suspended cloud condensate with density ρ_c. The mass fractions of water vapor and cloud water are defined as $q_v = \rho_v/\rho$ and $q_c = \rho_c/\rho$, respectively. In addition, let ρ_r be the rain density and $q_r = \rho_r/\rho$ the rain mass fraction. Warm rain is assumed (No ice formation or precipitation takes place). We denote by c_p and c_v the specific heat capacities at constant pressure and volume for dry air. The specific gas constants of dry air and vapor are denoted by R_d and R_v and set $\epsilon = \frac{R_d}{R_v}$. Let:

$$\theta = (1 + \epsilon q_v)\frac{T}{\pi}, \quad \text{with } \pi = \left(\frac{p}{p_s}\right)^{\frac{R_d}{c_p}}, \tag{1}$$

be the virtual potential temperature, where T is the absolute temperature and $p_s = 10^5$ Pa is the ground surface pressure. Finally, let $\mathbf{u}$ be the wind velocity.

We consider a fixed spatial domain Ω and a time interval of interest $(0, t_f]$. Balance of mass, momentum, and potential temperature for moist air in terms of *prognostic variables* ρ, $\mathbf{u}$, and θ in conservative form are given by:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \qquad \text{in } \Omega \times (0, t_f], \tag{2}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \odot \mathbf{u}) = -\nabla p + \rho \mathbf{b} \qquad \text{in } \Omega \times (0, t_f], \tag{3}$$

$$\frac{\partial (\rho \theta)}{\partial t} + \nabla \cdot (\rho \theta \mathbf{u}) = \rho \mathcal{S}_\theta \qquad \text{in } \Omega \times (0, t_f]. \tag{4}$$

where $\mathbf{b}$ is the total buoyancy. We have $\mathbf{b} = -(1+\epsilon q_v - q_c - q_r) g \widehat{\mathbf{k}}$, where $g = 9.81$ m/s^2 is the magnitude of the acceleration of gravity, and $\widehat{\mathbf{k}}$ is the unit vector aligned with the vertical axis z. Finally, the source/sink term $\mathcal{S}_\theta$ in (4) describes latent heat release–uptake during phase changes of moisture variables and is detailed in Sec. 2.1. Eq. (3) and (4) can be rewritten in non-conservative form as follows:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \mathbf{b} \qquad \text{in } \Omega \times (0, t_f], \tag{5}$$

$$\frac{\partial \theta}{\partial t} + \mathbf{u} \cdot \nabla \theta = \mathcal{S}_\theta \qquad \text{in } \Omega \times (0, t_f]. \tag{6}$$

A thermodynamics equation of state for the pressure of moist air p is needed for closure. We assume that p is the sum of the partial pressures of dry air and vapor (p_d and p_v, respectively), both taken to be ideal gases. Thus, neglecting the volume of the condensed phase, the equation of state relating p to ρ and T is given by:

$$p = p_d + p_v = \rho R_d T + \rho q_v R_v T = \rho R_d T (1 + \epsilon q_v). \tag{7}$$

To facilitate the numerical solution of system (2)-(4) or (2), (5)-(6), we write density, pressure, and potential temperature as the sum of their mean hydrostatic values and fluctuations:

$$\rho(x,y,z,t) = \rho_0(z) + \rho'(x,y,z,t), \tag{8}$$
$$\theta(x,y,z,t) = \theta_0(z) + \theta'(x,y,z,t), \tag{9}$$
$$p(x,y,z,t) = p_0(z) + p'(x,y,z,t). \tag{10}$$

Note that the hydrostatic reference states are functions of the vertical coordinate z only. Hydrostatic balance relates p_0 to ρ_0 as follows:

$$\frac{dp_0}{dz} = -\rho_0 g. \tag{11}$$

Plugging (8)-(10) into (2)-(4) and accounting for (11) leads to:

$$\frac{\partial \rho'}{\partial t} + \nabla \cdot ((\rho_0 + \rho')\mathbf{u}) = 0, \tag{12}$$
$$\frac{\partial((\rho_0+\rho')\mathbf{u})}{\partial t} + \nabla \cdot ((\rho_0+\rho')\mathbf{u}\otimes\mathbf{u}) + \rho' g\widehat{\mathbf{k}} = -\nabla p' + (\rho_0+\rho')\widetilde{\mathbf{b}}, \tag{13}$$
$$\frac{\partial((\rho_0+\rho')(\theta_0+\theta'))}{\partial t} + \nabla \cdot ((\rho_0+\rho')\theta'\mathbf{u}) + \nabla \cdot ((\rho_0+\rho')\theta_0\mathbf{u}) = (\rho_0+\rho')\mathcal{S}_\theta, \tag{14}$$

where $\widetilde{\mathbf{b}} = -\left(\frac{\rho'}{\rho_0+\rho'} + \epsilon q_v - q_c - q_r\right) g\widehat{\mathbf{k}}$ is a modified total buoyancy. Following a similar procedure for Eq. (5)-(6), we obtain

$$\frac{\partial \rho'}{\partial t} + \nabla \cdot ((\rho_0 + \rho')\mathbf{u}) = 0, \tag{15}$$
$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\frac{1}{\rho_0+\rho'}\nabla p' + \widetilde{\mathbf{b}}, \tag{16}$$
$$\frac{\partial \theta'}{\partial t} + \mathbf{u}\cdot\nabla\theta_0 + \mathbf{u}\cdot\nabla\theta' = \mathcal{S}_\theta. \tag{17}$$

Remark 2.1 *To preserve numerical stability of the solution, we add an artificial diffusion term with a constant diffusivity coefficient β to equation sets* (12)-(14) *and* (15)-(17)*; the units of β are given consistently with the equations at hand. The term $\beta\nabla^2\mathbf{u}$ is added to the right-hand side of the momentum equation, while the term $\beta\nabla^2\theta'$ is added to the right-hand side of the equation of the potential temperature.*

Remark 2.2 *While we usually stabilize NUMA simulations by leveraging the eddy viscosity from an LES model (see (Marras et al., 2015; Reddy et al., 2021)), in this paper we consider artificial viscosity with constant β as it is done in (Gaberšek et al., 2012; Skamarock et al., 2012) whose results we are testing against.*

Next, we write the balance equations for q_v and q_c in conservative form:

$$\frac{\partial(\rho q_v)}{\partial t} + \nabla\cdot(\rho q_v \mathbf{u}) = \rho\mathcal{S}_v \qquad \text{in } \Omega\times(0,t_f], \tag{18}$$
$$\frac{\partial(\rho q_c)}{\partial t} + \nabla\cdot(\rho q_c \mathbf{u}) = \rho\mathcal{S}_c \qquad \text{in } \Omega\times(0,t_f], \tag{19}$$

and non-conservative form:

$$\frac{\partial q_v}{\partial t} + \mathbf{u}\cdot\nabla q_v = \mathcal{S}_v \qquad \text{in } \Omega\times(0,t_f], \tag{20}$$
$$\frac{\partial q_c}{\partial t} + \mathbf{u}\cdot\nabla q_c = \mathcal{S}_c \qquad \text{in } \Omega\times(0,t_f]. \tag{21}$$

The source/sink terms on the right-hand side in the equations above are related to conversion rates. In particular, we have:

$$\mathcal{S}_v = C(q_c \to q_v) + C(q_r \to q_v), \quad \mathcal{S}_c = C(q_v \to q_c) + C(q_r \to q_c), \quad \mathcal{S}_t = \mathcal{S}_v + \mathcal{S}_c, \tag{22}$$

where the terms $C(q_\phi \to q_\psi) = -C(q_\psi \to q_\phi)$ represent the conversion of species ϕ to species ψ. All of these terms, which account for processes such as evaporation of cloud condensate, are provided by the microphysics equations reported in Sec. 2.1.

Precipitating water (rain) is treated in the same manner. Letting w_r be the fall speed of rain (provided by the microphysics equations), we can write the conservation law for rain in conservative form:

$$\frac{\partial(\rho q_r)}{\partial t} + \nabla \cdot (\rho q_r(\mathbf{u} - w_r\widehat{\mathbf{k}})) = \rho \mathcal{S}_r \quad \text{in } \Omega \times (0, t_f], \tag{23}$$

and non-conservative form:

$$\frac{\partial q_r}{\partial t} + \mathbf{u} \cdot \nabla q_r = \mathcal{S}_r + \frac{1}{\rho}\frac{\partial}{\partial z}(\rho q_r w_r) \quad \text{in } \Omega \times (0, t_f], \tag{24}$$

with

$$\mathcal{S}_r = C(q_v \to q_r) + C(q_c \to q_r). \tag{25}$$

In summary, the conservative form of the atmospheric model considered in this paper is given by (12)-(14), (18)-(19), (23) and (7), while its non-conservative form is given by (15)-(17), (20)-(21), (24) and (7). In both cases, the problem has to be supplemented with proper initial and boundary conditions that will be specified in Sec. 4.

2.1 Microphysical parameterization

The terms on the right-hand sides of Eq. (14), (18), (19), and (23), and their respective non-conservative counterparts are defined according to (Klemp & Wilhelmson, 1978). Let q_{vs} be the saturation water vapor fraction. To determine q_{vs} we use Teten's formula following (Klemp & Wilhelmson, 1978). The evaporation of cloud water is given by:

$$C(q_c \to q_v) = -C(q_v \to q_c) = \frac{\partial q_{vs}}{\partial t}. \tag{26}$$

This is computed with the saturation adjustment approach of Soong and Ogura (Soong & Ogura, 1973). The evaporation of rain, i.e. conversion rate $C(q_r \to q_v) = -C(q_v \to q_r)$, is taken directly from Klemp and Wilhelmson (1978), which uses an approach similar to Ogura and Takahashi (1971). We have

$$C(q_c \to q_r) = -C(q_r \to q_c) = A_r + C_r, \tag{27}$$

where A_r and C_r represent rain auto-conversion and rain accretion (Kessler, 1969), respectively. Finally, the source/sink term in Eq. (17) is given by:

$$S_\theta = -\gamma \left(\frac{\partial q_{vs}}{\partial t} + C(q_r \to q_v) \right), \quad \gamma = \frac{L}{c_p \pi}, \tag{28}$$

where L is the latent heat of vaporization and π is the Exner pressure defined in (1).

Finally, we define the terminal velocity of rain following (Soong & Ogura, 1973; Kessler, 1969; Klemp & Wilhelmson, 1978):

$$w_r = 3634(\rho q_r^{0.1346}) \left(\frac{\rho}{\rho_g} \right)^{-\frac{1}{2}}, \tag{29}$$

where ρ_g is the reference density at the surface.

3 Numerical method

3.1 The Galerkin spectral element method

In time, the equations are advanced using an implicit-explicit order 3 additive Runge-Kutta (ARK3) scheme (Kennedy & Carpenter, 2003) whereby the non-linear terms of the governing equations are treated explicitly and the linear terms are treated implicitly (see (Giraldo et al., 2013)). As for the space discretization, we use spectral elements and show results for both continuous and discontinuous approximations. This section focuses on the space discretization alone.

To make the description of the numerical method easy to follow, we consider a generic equation of the form:

$$\frac{\partial f}{\partial t} + G(f) = 0, \tag{30}$$

where f is the unknown variable and G is a linear functional that may contain first and second derivatives of f. If the equations to be solved are written in conservation form, then G is the divergence of a flux. Notice that all the equations in Sec. 2 can be rewritten as (30).

We subdivide the domain Ω into a set of conforming [1] N_e hexahedral elements Ω_e of arbitrary orientation to create the discrete domain Ω^h as

$$\Omega \approx \Omega^h = \bigcup_{e=1}^{N_e} \Omega_e. \tag{31}$$

Fig. 1 shows examples of a structured and unstructured grid in 2D. Using a fully unstructured grid means that structures such as the rows or columns that are seen on the left side of Fig. 1 are no longer present. Let Ω_{ref} be reference element: $(\xi, \eta) \in [-1, 1]^2$ in 2D and $(\xi, \eta, \zeta) \in [-1, 1]^3$ in 3D. Regardless of whether the mesh is structured or unstructured, we introduce a mapping from a generic element in the global system of coordinates, i.e. (x, y) in 2D and (x, y, z) in 3D, to the reference element. Let $\mathbf{J}$ be the Jacobian matrix of this mapping.

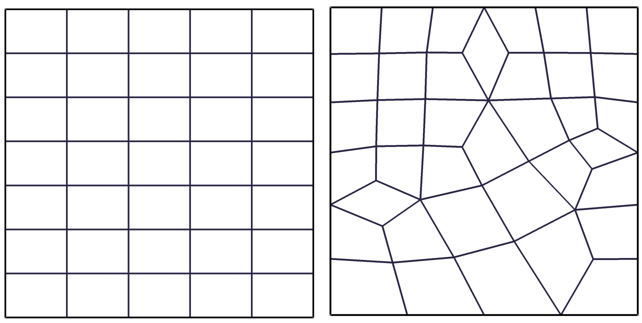

Figure 1: Examples of a structured (left) and an unstructured grid (right) made of quadrilateral elements.

Let h_i, $i = 1, \ldots, N+1$, be the Lagrange polynomials of degree N:

$$h_i(\xi) = \frac{1}{N(N+1)} \frac{(1-\xi^2)P_N'(\xi)}{(\xi - \xi_i)P_N(\xi)},$$

[1] The condition of conformity is not strictly necessary, although it simplifies the discussion of the method. For results with non-conforming grids, the reader is referred to, e.g., (Kopera & Giraldo, 2014).

where P_N is the Legendre polynomial of order N, and $P_N^{'}$ its derivative evaluated at the point ξ. The polynomials in multiple dimensions are built via a tensor product of the 1D bases, as shown below. The remainder of this section is written for a 3D case.

For every element, we seek an approximation f^h of variable f of the form:

$$f^h(\boldsymbol{\xi}, t) = \sum_{l=1}^{(N+1)^3} \psi_l(\boldsymbol{\xi}) \hat{f}_l(t), \tag{32}$$

where $\boldsymbol{\xi} = (\xi, \eta, \zeta)$, $\hat{f}_l$ are the expansion coefficients, and ψ_l are nodal basis functions defined as tensor products of the Lagrange polynomials

$$\psi_l = h_i[\xi(\mathbf{x})] \odot h_j[\eta(\mathbf{x})] \odot h_k[\zeta(\mathbf{x})], \quad l = i + 1 + j(N+1) + k(N+1)(N+1), \tag{33}$$

where $\mathbf{x} = (x, y, z)$. The Legendre-Gauss-Lobatto (LGL) points are not equidistant and represent the solutions of the following equation:

$$(1 - \xi^2)P_N^{'}(\xi) = 0.$$

The LGL points are associated with the following quadrature weights:

$$\omega(\xi_i) = \frac{2}{N(N+1)} \left[\frac{1}{P_N(\xi_i)}\right]^2$$

used to approximate the integrals with a Gauss quadrature rule of accuracy $\mathcal{O}(2N-1)$. Over a generic element Ω_e, this is done as follows:

$$\int_{\Omega_e} f(\mathbf{x}) d\mathbf{x} = \int_{\Omega_{ref}} f(\boldsymbol{\xi}) |\mathbf{J}(\boldsymbol{\xi})| d\boldsymbol{\xi} \approx \sum_{i,j,k=1}^{N+1} \omega(\xi_i)\omega(\eta_j)\omega(\zeta_k) f(\xi_i, \eta_j, \zeta_k) |\mathbf{J}(\xi_i, \eta_j, \zeta_k)|, \tag{34}$$

where $|\mathbf{J}|$ is the determinant of the Jacobian matrix.

To approximate the solution of Eq. (30), let $(\cdot, \cdot)$ be the Legendre inner product on a given element Ω_e:

$$(f, g)_e = \int_{\Omega_e} f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x}.$$

If in (30) we replace f with f^h as defined in (32), we will obtain the following residual:

$$R = \frac{\partial f^h}{\partial t} + G(f^h), \tag{35}$$

which is orthogonal to the expansion functions in Galerkin methods, i.e.:

$$(R, \psi_k)_e = 0, \;\; k = 1, \ldots, (N+1)^3. \tag{36}$$

Taking (36) into account, we can now write an approximation of Eq. (30) on each element Ω_e as follows :

$$\int_{\Omega_e} \psi_i(\mathbf{x}) \frac{\partial f^h(\mathbf{x}, t)}{\partial t} d\mathbf{x} = -\int_{\Omega_e} \psi_i(\mathbf{x}) G(f^h(\mathbf{x}, t)) d\mathbf{x}, \;\; i = 1, \ldots, (N+1)^3. \tag{37}$$

Let us first consider the case where $G(f) = \boldsymbol{\nabla} \cdot \mathbf{f}$, where $\boldsymbol{\nabla} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$ and $\mathbf{f} = (f, f, f)$.

We can use the polynomial expansion to write (37) as follows:

$$\int_{\Omega_e} \psi_i(\mathbf{x}) \sum_{j=1}^{(N+1)^3} \psi_j(\mathbf{x}) \frac{\partial \hat{f}_j^e(t)}{\partial t} d\mathbf{x} = -\int_{\Omega_e} \psi_i(\mathbf{x}) \sum_{j=1}^{(N+1)^3} \boldsymbol{\nabla}\psi_j(\mathbf{x}) \cdot \hat{\mathbf{f}}^{\mathbf{e}}_{\mathbf{j}}(\mathbf{t}) d\mathbf{x}, \;\; , i = 1, \ldots, (N+1)^3, \tag{38}$$

where the superscript e is used to denote that the expansion is defined on an element basis and $\hat{\mathbf{f}}^{\mathbf{e}}_{\mathbf{j}}(\mathbf{t}) = (\hat{f}^e_j(t), \hat{f}^e_j(t), \hat{f}^e_j(t))$. We can now write the mass matrix $\mathbf{M}^e_{ij}$ and the differentiation matrix $\mathbf{D}^e_{ij}$ on each element:

$$\mathbf{M}^e_{ij} = \int_{\Omega_e} \psi_i(\mathbf{x})\psi_j(\mathbf{x})d\mathbf{x} = \int_{\Omega_{ref}} \psi_i(\boldsymbol{\xi})\psi_j(\boldsymbol{\xi})|\mathbf{J}(\boldsymbol{\xi})|d\boldsymbol{\xi}, \tag{39}$$

$$\mathbf{D}^e_{ij} = \int_{\Omega_e} \psi_i(\mathbf{x})\boldsymbol{\nabla}\psi_j(\mathbf{x})d\mathbf{x} = \int_{\Omega_{ref}} \psi_i(\boldsymbol{\xi})\left(\boldsymbol{\nabla}_\xi\psi_j(\boldsymbol{\xi})\mathbf{J}^{-1}(\boldsymbol{\xi})\right)|\mathbf{J}(\boldsymbol{\xi})|d\boldsymbol{\xi}, \tag{40}$$

with $i, j = 1, \ldots, (N+1)^3$ and $\boldsymbol{\nabla}_\xi = \left(\frac{\partial}{\partial\xi}, \frac{\partial}{\partial\eta}, \frac{\partial}{\partial\zeta}\right)$. By approximating the integrals with a quadrature rule, we obtain:

$$\mathbf{M}^e_{ij} = \sum_{k=1}^{N+1}\sum_{m=1}^{N+1}\sum_{n=1}^{N+1} \omega(\xi_k,\eta_m,\zeta_n)\psi_i(\xi_k,\eta_m,\zeta_n)\psi_j(\xi_k,\eta_m,\zeta_n)|\mathbf{J}(\xi_k,\eta_m,\zeta_n)|, \tag{41}$$

$$\mathbf{D}^e_{ij} = \sum_{k=1}^{N+1}\sum_{m=1}^{N+1}\sum_{n=1}^{N+1} \omega(\xi_k,\eta_m,\zeta_n)\psi_i(\xi_k,\eta_m,\zeta_n)\boldsymbol{\nabla}\psi_j(\xi_k,\eta_m,\zeta_n)|\mathbf{J}(\xi_k,\eta_m,\zeta_n)|. \tag{42}$$

Note that $\boldsymbol{\nabla}\psi_j(\xi_k,\eta_m,\zeta_n) = \boldsymbol{\nabla}_\xi\psi_j(\xi_k,\eta_m,\zeta_n)\mathbf{J}^{-1}(\xi_k,\eta_m,\zeta_n)$. Then, the matrix form of Eq. (38) is:

$$\mathbf{M}^e_{ij}\frac{\partial \hat{f}^e_j(t)}{\partial t} = -\mathbf{D}_{ij}\hat{f}^e_j(t), \;\; i, j = 1, \ldots, (N+1)^3. \tag{43}$$

Let us now consider $G(f) = \boldsymbol{\nabla}\cdot\mathbf{f} - \boldsymbol{\nabla}^2 f$ in Eq. (30), where $\boldsymbol{\nabla}^2 = \boldsymbol{\nabla}\cdot\boldsymbol{\nabla}$. In this case, Eq. (37) becomes:

$$\begin{aligned}\int_{\Omega_e}\psi_i(\mathbf{x})\sum_{j=1}^{(N+1)^3}\psi_j(\mathbf{x})\frac{\partial\hat{f}^e_j(t)}{\partial t}d\mathbf{x} = &-\int_{\Omega e}\psi_i(\mathbf{x})\sum_{j=1}^{(N+1)^3}\boldsymbol{\nabla}\psi_j(\mathbf{x})\cdot\hat{\mathbf{f}}^{\mathbf{e}}_{\mathbf{j}}(\mathbf{t})d\mathbf{x}\\ &+\int_{\Omega_e}\psi_i\boldsymbol{\nabla}\cdot\left[\sum_{j=1}^{(N+1)^3}\boldsymbol{\nabla}\psi_j(\mathbf{x})\hat{f}^e_j(t)\right]d\mathbf{x},\end{aligned} \tag{44}$$

where $i, j = 1, \ldots, (N+1)^3$. After integrating by parts the second term on the right-hand side, we can rewrite (44) as:

$$\begin{aligned}\mathbf{M}^e_{ij}\frac{\partial\hat{f}^e_j(t)}{\partial t} = &-\mathbf{D}^e_{ij}\hat{f}^e_j(t) + \left[\psi_i(\mathbf{x})\sum_{j=1}^{N+1}\boldsymbol{\nabla}\cdot\psi_j(\mathbf{x})\hat{f}^e_j(t)\right]_{\Gamma_e}\\ &-\int_{\Omega_e}\boldsymbol{\nabla}\psi_i(\mathbf{x})\cdot\sum_{j=1}^{N+1}\boldsymbol{\nabla}\psi_j(\mathbf{x})\hat{f}^e_j(t)d\Omega_e \;\; i, j = 1, \ldots, (N+1)^3,\end{aligned} \tag{45}$$

where Γ_e represents the element boundary. For the sake of brevity, we assume that the boundary term, i.e., the second term on the right-hand side in (45), vanishes at all element boundaries. We refer the reader to, e.g., (Giraldo, 2020; Kelly & Giraldo, 2012) for a detailed explanation of how this term is handled when it is not zero, as is the case for DG. Under the assumption of vanishing boundary terms, Eq. (45) becomes:

$$\mathbf{M}^e_{ij}\frac{\partial\hat{f}^e_j(t)}{\partial t} = -\mathbf{D}^e_{ij}\hat{f}^e_j(t) - \int_{\Omega_e}\boldsymbol{\nabla}\psi_i(\mathbf{x})\cdot\boldsymbol{\nabla}\psi_j(\mathbf{x})d\mathbf{x}\hat{f}^e_j, \;\; i, j = 1, \ldots, (N+1)^3. \tag{46}$$

We define the Laplacian matrix as follows:

$$\mathbf{L}^e_{ij} = \int_{\Omega_e}\boldsymbol{\nabla}\psi_i(\mathbf{x})\cdot\boldsymbol{\nabla}\psi_j(\mathbf{x})d\mathbf{x} = \int_{\Omega_{ref}}(\boldsymbol{\nabla}_\xi\psi_i(\boldsymbol{\xi})\mathbf{J}^{-1}(\boldsymbol{\xi}))\cdot(\boldsymbol{\nabla}_\xi\psi_j(\boldsymbol{\xi})\mathbf{J}^{-1}(\boldsymbol{\xi}))|\mathbf{J}(\boldsymbol{\xi})|d\boldsymbol{\xi}, \tag{47}$$

where $i, j = 1, \ldots, N+1$. By approximating the integral in (47) with a quadrature rule, we obtain:

$$\mathbf{L}^e_{ij} = \sum_{k=1}^{N+1} \sum_{m=1}^{N+1} \sum_{n=1}^{N+1} \omega(\xi_k, \eta_m, \zeta_n) \boldsymbol{\nabla} \psi_i(\xi_k, \eta_m, \zeta_n) \cdot \boldsymbol{\nabla} \psi_j(\xi_k, \eta_m, \zeta_n) |\mathbf{J}(\xi_k, \eta_m, \zeta_n)|, \tag{48}$$

where $i, j = 1, \ldots, (N+1)^3$. Then, we write (46) as:

$$\mathbf{M}^e_{ij} \frac{\partial \hat{f}^e_j(t)}{\partial t} = -\mathbf{D}^e_{ij} \hat{f}^e_j(t) - \mathbf{L}^e_{ij} \hat{f}^e_j(t) \ \ , i, j = 1, \ldots, (N+1)^3. \tag{49}$$

Next, we present briefly how the global solution is calculated depending on the choice of continuous Galerkin (CG) or discontinuous Galerkin (DG) spectral elements. The reader interested in more details on Galerkin spectral element methods is referred to, e.g., (Giraldo, 2020; Hesthaven & Warburton, 2008; Kopriva, 2008; Sherwin & Karniadakis, 2005).

3.1.0.1 CG approximation: Let $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{L}$ be the global mass matrix, global differentiation matrix, and global Laplacian matrix. These matrices are, in principle, assembled using Direct Stiffness Summation (DSS):

$$\mathbf{M} = \sum_{e=1}^{N_e} \mathbf{M}^e, \ \ \mathbf{D} = \sum_{e=1}^{N_e} \mathbf{D}^e, \ \ \mathbf{L} = \sum_{e=1}^{N_e} \mathbf{L}^e$$

where $\mathbf{M}^e$ is the element mass matrix (41), $\mathbf{D}^e$ is the element differentiation matrix (42), and $\mathbf{L}^e$ is the element weak Laplacian matrix (48). Since the same set of LGL points are used for both interpolation and integration, the global mass matrix $\mathbf{M}$ is diagonal and thus easy to invert. This is only the case if we integrate using $N+1$ LGL points as shown in (34). This type is known as inexact numerical integration, since the number of LGL quadrature points necessary to integrate a polynomial of order $2N$ (such as is the case for the mass matrix) up to machine precision is $N+2$. We choose to sacrifice accuracy in favor of obtaining an easily invertible mass matrix, which allows us to save considerable computational time. Additionally, it has been shown that when using polynomials of order $N \geq 4$ this type of integration has a minimal impact on accuracy, with the impact decreasing as the polynomial order is increased (Giraldo, 2020). For the results in Sec. 4, we use $N = 4$. It should be noted, however, that no global matrix is actually constructed (except for the diagonal mass matrix); the differentiation and Laplacian global matrices are never stored, only the action of these matrices on the solution vector is computed (see, e.g., (Giraldo, 2020)).

The global form associated with Eq. (30) for $G(f) = \boldsymbol{\nabla} f + \boldsymbol{\nabla}^2 f$ can be written as:

$$\frac{\partial \mathbf{f}^h}{\partial t} + \mathbf{M}^{-1}(\mathbf{D}\mathbf{f}^h + \mathbf{L}\mathbf{f}^h) = 0, \tag{50}$$

where $\mathbf{f}^h$ is the vector containing the nodal values of f^h.

3.1.0.2 DG approximation: For this kind of approximation, the global matrices are not constructed since an element communicates only with the neighboring elements through inter-element numerical fluxes. Thus, we write a local approximation of Eq. (30), instead of a global one as in (50).

Let us apply integration by parts to the entries of the differentiation matrix:

$$\mathbf{D}^e_{ij} = \int_{\Omega_e} \psi_i(\mathbf{x}) \boldsymbol{\nabla} \psi_j(\mathbf{x}) d\mathbf{x} = \int_{\Gamma_e} \psi_i(\mathbf{x}) \psi_j(\mathbf{x}) \mathbf{n}^{(F,e)} d\Omega_e - \int_{\Omega_e} \boldsymbol{\nabla} \cdot \psi_i(\mathbf{x}) \psi_j(\mathbf{x}) d\mathbf{x}, \tag{51}$$

where $i, j = 1, \ldots, (N+1)^3$, $\mathbf{n}^{(F,e)}$ is the outwards facing normal of inter-element face F of the element e. The first term of the right-hand side in (51) represents an inter-element flux or a boundary flux if the element is a boundary element and it enforces the continuity of the global solution. Notice that in a CG discretization this term vanishes as continuity is enforced via DSS. We define the corresponding matrix as follows:

$$\mathbf{F}^e_{ij} = \int_{\Gamma_e} \psi_i(\mathbf{x})\psi_j(\mathbf{x})\mathbf{n}^{(F,e)} d\mathbf{x} \approx \sum_{F=1}^{N_F}\sum_{k=1}^{N+1}\sum_{m=1}^{N+1} \omega(\boldsymbol{\xi}_{F,km})\psi_i(\boldsymbol{\xi}_{F,km})\psi_j(\boldsymbol{\xi}_{F,km})|\mathbf{J}(\boldsymbol{\xi}_{F,km})|\mathbf{n}^{(F,e)}, \tag{52}$$

where $i, j = 1, \ldots, (N+1)^3$, N_F is the number of faces for element e and $\boldsymbol{\xi}_{F,km}$ denotes an integration point on the face F of the element. The second term on the right-hand side in (51) is called the weak differentiation matrix and is approximated as follows:

$$\hat{\mathbf{D}}^e_{ij} = \int_{\Omega_e} \nabla\psi_i(\mathbf{x})\psi_j(\mathbf{x})d\mathbf{x} \quad \approx \sum_{k=1}^{N+1}\sum_{m=1}^{N+1}\sum_{n=1}^{N+1} \omega(\xi_k,\eta_m,\zeta_n)|\mathbf{J}(\xi_k,\eta_m,\zeta_n)|\nabla\psi_i(\xi_k,\eta_m,\zeta_n)\psi_j(\xi_k,\eta_m,\zeta_n), \tag{53}$$

204 where $i, j = 1, \ldots, (N+1)^3$.

We can now rewrite (49) for a DG discretization taking $G(f) = \nabla \cdot f + \nabla^2 f$, which holds on each element as follows:

$$\mathbf{M}^e_{ij}\frac{\partial \hat{f}^e_j(t)}{\partial t} = -\hat{\mathbf{D}}^e_{ij}\hat{f}^e_j(t) + \mathbf{F}^e_{ij}\mathbf{f}^*_j(t) - \mathbf{L}^e_{ij}\hat{f}^e_j(t) = 0, \quad i,j = 1,\ldots,(N+1)^3,$$

where $\mathbf{f}^*$ represents the inter-element interface values of $\hat{f}^e_j$. We define $\mathbf{f}^*$ as follows:

$$\mathbf{f}^*_j = \mathbf{C}(\hat{f}^e_j) - \mathbf{P}(\hat{f}^e_j)$$

where $\mathbf{P}$ is a penalty term and the central term $\mathbf{C}$ is defined as follows:

$$\mathbf{C}(\hat{f}^e_j) = (g(\hat{f}^{e,R}_j) + g(\hat{f}^{e,L}_j))/2,$$

where L and R refer to the left and right sides of a given inter-element interface. The function g is dependent on the first derivative component of G in (30) where, in this case, $G(f) = \nabla \cdot \mathbf{f} + \nabla^2 f$ and $g(f) = \mathbf{f}$. The definition of P depends on the choice of numerical flux. The simplest and most commonly used flux for DG is the Rusanov flux (Giraldo, 2020), which gives:

$$\mathbf{P}(\hat{f}^e_j) = \mathbf{n}^{(F,e)} w_s(\hat{f}^{e,R}_j - \hat{f}^{e,L}_j)/2,$$

where w_s is the wave speed across the interface, which depends on the specific equation to be solved. This gives the following equation for $\mathbf{f}^*$:

$$\mathbf{f}^*_j = \frac{1}{2}\left(\hat{\mathbf{f}}^{e,R}_j + \hat{\mathbf{f}}^{e,L}_j - \mathbf{n}^{F,e} w_s(\hat{f}^{e,R}_j - \hat{f}^{e,L}_j)\right), \quad j = 1,\ldots,(N+1)^3, \tag{54}$$

205 where $\hat{\mathbf{f}}^e_j = (\hat{f}^e_j, \hat{f}^e_j, \hat{f}^e_j)$. We note that in the DG formulation for $G(f) = \nabla \cdot f + \nabla^2 f$
206 the boundary term in (45) does not vanish and needs to be evaluated. Such term is treated
207 in a similar fashion as the boundary term in (51). For the details, we refer the interested
208 reader to (Giraldo, 2020; Hesthaven & Warburton, 2008).

209 ### 3.2 Non-column based rain sedimentation

210 The main novelty of this work lies in the computation of the sedimentation term
211 for the rain equation (i.e., the last term on the right-hand side in Eq. (24)) which dif-
212 fers from the methods in, e.g., (Kessler, 1969; Klemp & Wilhelmson, 1978; Soong & Ogura,
213 1973; Ogura & Takahashi, 1971; Houze, 1993). The typical column-based approach to
214 handle the sedimentation term is by computing the spatial derivative along each indi-
215 vidual column starting from the top of the domain and descending. See, e.g., (Gaberšek
216 et al., 2012; Marras et al., 2013a) for a spectral element implementation of this approach.

Although widely used, the traditional column-based implementation has a main drawback: it requires the availability of column-aware data structures that may not serve other purposes in the numerical method, thereby forcing the use of structured grids. Unstructured grids are highly advantageous around topography. By forgoing the use of columns, our approach to compute sedimentation could help yield more accurate predictions for storm behavior in mountainous regions.

Computing the sedimentation term is done separately from the other microphysics calculations, and is done after solving the compressible Euler and moisture advection equations. This term is included by solving the following equation:

$$\frac{\partial q_r}{\partial t} = \frac{1}{\rho}\frac{\partial}{\partial z}(\rho q_r w_r) \tag{55}$$

in non-conservative form and

$$\frac{\partial(\rho q_r)}{\partial t} = \boldsymbol{\nabla}\cdot(\rho q_r w_r \hat{\mathbf{k}}) \tag{56}$$

in conservation form. Given that $\hat{\mathbf{k}} = (0,0,-1)^T$ for the domains we consider, (56) can be written as follows:

$$\frac{\partial(\rho q_r)}{\partial t} = \frac{\partial}{\partial z}(\rho q_r w_r).$$

This makes it so that for either the conservative or non-conservative form, solving the sedimentation equation essentially amounts to calculating the term $\frac{\partial}{\partial z}(\rho q_r w_r)$.

We can rewrite the sedimentation equation in the form of (30) by taking $G(f) = -c\frac{\partial F_{\text{sed}}}{\partial z}$, where $F_{\text{sed}} = (\rho q_r w_r)$, $c = 1$ and $f = \rho q_r$ in conservation form, while $c = \frac{1}{\rho}$ and $f = q_r$ in non-conservative form. By multiplying by the expansion functions and integrating, we get:

$$\int_{\Omega_e} \psi_i(\mathbf{x})\frac{\partial f^h(\mathbf{x},t)}{\partial t}d\mathbf{x} = \int_{\Omega_e} \psi_i(\mathbf{x})\sum_{j=1}^{(N+1)^3}\frac{\partial \psi_j(\mathbf{x})}{\partial z}c\hat{F}^e_{j,sed}(t)(\mathbf{x})d\mathbf{x}, \quad i = 1,\ldots,(N+1)^3,$$

where $\hat{F}_{j,sed}$ are the expansion coefficients of $F_{j,sed}$. Moving to the reference element and identifying the mass matrix yields

$$\mathbf{M}^e_{ij}\frac{\partial \hat{f}^e_j(t)}{\partial t} = \int_{\Omega_{ref}} \psi(\boldsymbol{\xi})\left[\boldsymbol{\nabla}_\xi\psi_j(\boldsymbol{\xi})\cdot\left(\frac{\partial\xi}{\partial z},\frac{\partial\eta}{\partial z},\frac{\partial\zeta}{\partial z}\right)(\boldsymbol{\xi})\right]c\hat{F}^e_{j,sed}(t)|\mathbf{J}(\boldsymbol{\xi})|d\boldsymbol{\xi}, \tag{57}$$

where $i,j = 1,\ldots,(N+1)^3$. Let us call $\mathbf{D}^e_{\text{sed}}$ the element-wise differentiation matrix for (55) and write Eq. (57) in matrix form:

$$\mathbf{M}^e_{ij}\frac{\partial \hat{f}^e_j(t)}{\partial t} = \mathbf{D}^e_{ij,\text{sed}}c\hat{F}^e_{j,sed}(t), \quad i,j = 1,\ldots,(N+1)^3. \tag{58}$$

We can write $\mathbf{D}^e_{\text{sed}}$ discretely as follows:

$$\mathbf{D}^e_{ij,\text{sed}} = \sum_{k=1}^{N+1}\sum_{m=1}^{N+1}\sum_{n=1}^{N+1}\omega(\xi_k,\eta_m,\zeta_n)\psi_i(\xi_k,\eta_m,\zeta_n)\boldsymbol{\nabla}_\xi\psi_j(\xi_k,\eta_m,\zeta_n)\cdot\left(\frac{\partial\xi}{\partial z},\frac{\partial\eta}{\partial z},\frac{\partial\zeta}{\partial z}\right)(\xi_k,\eta_m,\zeta_n)|\mathbf{J}(\xi_k,\eta_m,\zeta_n)|, \tag{59}$$

where $i,j = 1,\ldots,(N+1)^3$. From this point, if CG is used the global equation can be solved using DSS as follows:

$$\frac{\partial \mathbf{f}^h}{\partial t} - \mathbf{M}^{-1}\mathbf{D}_{\text{sed}}(\mathbf{c}\odot\mathbf{F}^h_{\text{sed}}) = 0, \tag{60}$$

where $\mathbf{D}_{\text{sed}} = \sum_{e=1}^{N_e}\mathbf{D}^e_{\text{sed}}$, $\mathbf{c}$ is the vector containing the nodal values of c, $\mathbf{F}_{\text{sed}}$ is the vector containing the nodal values of F_{sed}, and $\odot$ denotes a component-wise multiplication. The local DG problem is given by:

$$\frac{\partial \mathbf{f}^h}{\partial t} - \mathbf{M}^{-1(e)}(\hat{\mathbf{D}}^e_{\text{sed}}(\mathbf{c}\odot\mathbf{F}^h_{\text{sed}}) - \mathbf{F}^{\mathbf{e}}(\mathbf{c}^*\odot\mathbf{F}^*_{\text{sed}})), \tag{61}$$

where $\hat{\mathbf{D}}^e_{\text{sed}}$ is the weak form of $\mathbf{D}^e_{\text{sed}}$, $\mathbf{F}^e$ is the flux matrix at each element, $\mathbf{F}^*_{\text{sed}}$ is the interface value of $\mathbf{F}_{\text{sed}}$, and $\mathbf{c}^*$ is the interface value of $\mathbf{c}$.

In what follows, we present the procedure we use to solve the fully compressible Euler equations with moisture, including rain. Algorithm 1 summarizes the entire procedure. The algorithm makes use of the following quantities: $N_{LGL} = N+1$ is the number LGL points in each element, Δt is the time step, $f^{h,n}$ is the approximation of f^h at the time $t^n = n\Delta t$, N_{points} the total number of points the domain has been discretized into including repeating nodes at element edges and faces, t_n the current discrete time, and $t_{n+1} = t_n + \Delta t$. We also define the sedimentation Courant number, which we use to determine the time sub-step for the sedimentation problem:

$$Cr = w_r \frac{\Delta t}{\Delta z}. \tag{62}$$

This number is used to determine the appropriate sedimentation time step as follows:

$$\Delta t_{\text{sed}} = \frac{\Delta t}{\max(1, 0.5 + Cr_{\max}/Cr_{\text{limit}})}, \tag{63}$$

where:

$$Cr_{\max} = \max([Cr_i]_{i=1}^{N_{points}}), \tag{64}$$

is the maximum sedimentation Courant number among all points in the domain and Cr_{limit} is the maximum allowable Courant number for the sedimentation problem. The rest of the notation is defined in Sec. 3.

Next, we report on the results obtained with this algorithm and fully unstructured grids.

4 Results

We assess the method presented in Sec. 3.2 with an idealized squall line test from (Gaberšek et al., 2012) and a fully 3D supercell problem from (Skamarock et al., 2012). All the simulations are run with the Nonhydrostatic Unified Model of the Atmosphere (NUMA) (Kelly & Giraldo, 2012), which is designed to solve the dry Euler equations, with the addition of artificial viscosity as described in Sec. 3, on unstructured grids of hexahedra with arbitrary orientation. NUMA enables the use of both CG and DG spectral elements and has been shown to scale exceptionally well on CPUs and GPUs in (D. Abdi et al., 2017; Müller et al., 2018).

4.1 2.5D Squall line

The first benchmark we consider is an idealized test presented in (Gaberšek et al., 2012). While the computational domain in (Gaberšek et al., 2012) is two-dimensional, we run the same test in a 2.5 D domain $\Omega = [150 \times 12 \times 24]$ km^3. The domain is discretized with a single element in the y direction and a resolution dependent number of elements in the x and z directions. Periodic boundary conditions are applied to the lateral boundaries, a free-slip type boundary condition is applied at the domain bottom and the domain top utilizes a Rayleigh sponge for gravity wave damping. In this domain, a squall line forms in a weakly stable atmosphere with Brunt-Väisälä frequency $N = 0.01$ s^{-1} below the tropopause and a more stable atmosphere with $N = 0.02$ s^{-1} above 12 km. The cloud begins to form around $t \approx 500$ s, while rain starts to form and fall at approximately $t \approx 900$ s. The initial condition consists of a saturated boundary layer typical of mid-latitude storms that has been used in several numerical studies (see, e.g, (Rotunno et al., 1988; Weisman et al., 1988)). A low altitude wind shear in the x direction is imposed to break the cloud symmetry and allow for a continuous storm evolution. The initial background sounding is tabulated in the Appendix.

Algorithm 1 Simulation of moist-air and rain sedimentation with unstructured grids.

for $time = 0, \Delta t, \ldots, t_f$ **do**
    for $e = 1, 2, \ldots, N_e$ **do**
        for $node = 1, 2, \ldots, N_{LGL}$ **do**
            Calculate contributions to element-wise derivatives from each LGL point along
            the reference element.
        end for
        Compute these local derivatives in physical space.
    end for
    Perform DSS for CG or calculate numerical fluxes for DG.
    Solve the discrete version of the Euler equations: (2), (3) and (4) if using conservation
    form, and (2), (5) and (6) if using non-conservative form.
    Solve the advection equations for q_v, q_c and q_r by the flow velocity u: (18), (19) and
    (23) if using conservation form, and (20), (21) and (24) if using non-conservative form.
    for $i = 1, 2, \ldots, N_{points}$ **do**
        Determine w_r using Eq. (29)
        Determine $Cr_{\max}$ using (64)
        Determine Δt_{sed} using (63)
    end for
    for ts **do** $= t_n, t_n + \Delta t_{\text{sed}}, \ldots, t_{n+1}$
        for $e = 1, 2, \ldots, Ne$ **do**
            if space method == CG
                Compute $\mathbf{D}^e_{\text{sed}}$
            else if space method == DG
                Compute $\hat{\mathbf{D}}^e_{\text{sed}}$
            end if
        end for
        if space method == CG
            Perform DSS.
        else if space method == DG
            Apply inter-element fluxes for the sedimentation equation using w_r as the wave
            speed.
        end if
        Solve (55)
    end for
    for $e = 1, 2, \ldots, Ne$ **do**
        Update moisture variables and potential temperature to account for phase changes
        following equations (28)-(27)
    end for
end for

The storm is triggered by a thermal perturbation of the background state (Rotunno et al., 1988) centered at $(x_c, z_c) = (75000, 2000)$ m and defined by:

$$\Delta\theta = \begin{cases} \theta_c \cos\left(\frac{\pi r}{2}\right) & \text{if } r \leq r_c, \\ 0 & \text{if } r \geq r_c, \end{cases} \tag{65}$$

where

$$r = \sqrt{\frac{(x - x_c)^2}{r_x^2} + \frac{(z - z_c)^2}{r_z^2}}, \quad \theta_c = 3 \text{ K}, \quad r_c = 1, \quad r_x = 10000 \text{ m}, \quad r_z = 1500 \text{ m}.$$

We generated seven grids using GMSH (Geuzaine & Remacle, 2009). Table 1 lists the total number of hexahedral elements and the effective resolution Δx for each mesh. We choose to report the effective resolution because the LGL points for an element are not equidistant (Giraldo, 2020; Hesthaven & Warburton, 2008; Kopriva, 2008). NUMA relies on P4est (Burstedde et al., 2011) to read unstructured meshes and perform the graph partitioning for the parallel application.

Fig. 2 shows an example of clouds and precipitation calculated on a fully unstructured grid of hexahedra for an effective resolution of 150 m in both spatial directions.

# elements	473	1078	3181	4134	6485	11447	25863
Δx	750 m	500 m	290 m	250 m	200 m	150 m	100 m

Table 1: Total number of hexahedral elements and effective resolution for all the meshes used for the squall line simulations.

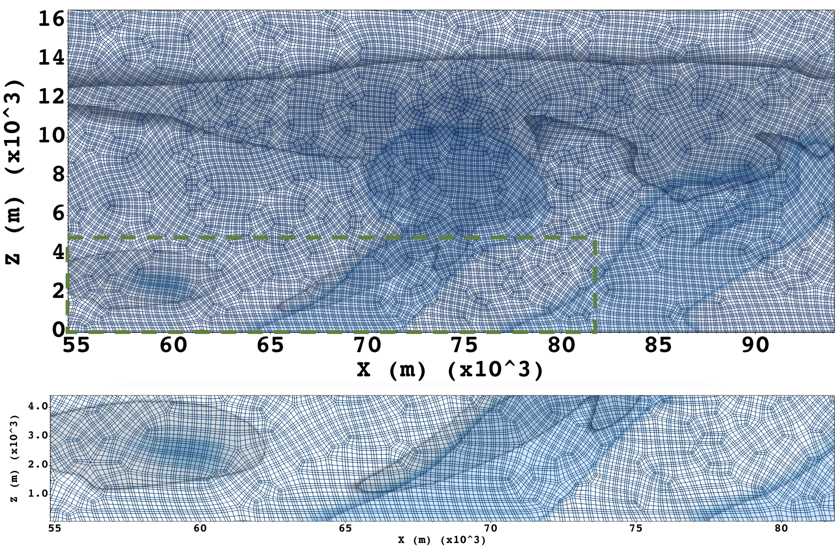

Figure 2: Top: q_c and q_r over unstructured grid $\Delta x = 150$ m. Cloud water is shaded in grey for values of $q_c > 1 \times 10^{-5}$ kg/kg whereas rain is shaded in blue for values of $q_r > 1 \times 10^{-4}$ kg/kg. Bottom: close-up view corresponding to the dashed rectangle in the top figure.

For all the simulations, we use an Additive Runge Kutta third order (ARK3) semi-implicit time integrator and elements of polynomial order 4. We maintain the acoustic Courant number $C \leq 1$ for all the simulations. While the ARK3 time integrator allows for larger acoustic Courant numbers, we limit the time step for the purposes of obtaining a greater deal of accuracy for the higher-resolution simulations. We run this test using both the CG approach with the governing equations in non-conservation form and the DG approach with the governing equations in conservation form. Consistently with (Gaberšek et al., 2012), a constant artificial viscosity of $\beta = 200$ (for the units see Remark 2.1) is used to stabilize the simulations.

Let us examine the results obtained with the finest mesh, i.e. the one with $\Delta x =$ 100 m. Figs. 3 and 4 show the stages of the storm evolution given by the CG and DG simulations, respectively. Both simulations yield very similar plots at $t = 1500$ s. Additionally, in both cases we observe a downwind tilt of the convective tower, which is caused by the horizontal wind-shear, and the eventual development of the anvil cloud near the tropopause where the atmosphere presents higher stability. For the sake of brevity, we do not report the plots associated with other meshes, but a similar early storm evolution is observed in all the simulations at all resolutions with both CG and DG approaches. The differences between the CG and DG simulations remain minimal even up to about $t = 6000$ s. This is a rather long period of time since by then the storm has fully developed. Starting from $t = 6000$ s till the end of the time interval of interest, some differences in the CG and DG simulations arise, as can been seen by comparing Figs. 3 and 4. At $t = 9000$ s, when additional convective towers are observed, the DG simulation generates multiple convective towers, some of which are significantly downwind. This is not as pronounced in the CG simulation. Compare the bottom right panels in Figs. 3 and 4.

Figs. 3 and 4 reports also the rain accumulated on the ground. At $t = 1500$ s, no rain has accumulated yet in either the DG or CG simulations. This is confirmed by the rain contours plots, where we see that the contour lines have yet to reach the ground. See top left panel in Figs. 3 and 4. At $t = 3000$ s, the accumulated rain is primarily near the center of the domain for both methods. Indeed, from the top right panel in Figs. 3 and 4 we see that rain accumulates at the location of the convective tower, with a slight asymmetry that follows the asymmetry of the convective tower seen at $t = 1500$ s. As time progresses, the convective tower tilts. An early stage of this is visible at $t = 3000$ s, but the tilting becomes more pronounced at $t = 6000$ s when the effect of the wind shear is more noticeable. The rain accumulation reflects the tilting and location of the convective tower in both the CG and DG simulations, as shown in the bottom left panel of Figs. 3 and 4. By $t = 9000$ s, we observe once again some differences in the results given by the two methods. For the CG simulation, in the bottom right panel of Fig. 3 we see a much wider distribution of accumulated rain with a secondary peak below the new location of the convective tower and a third peak appearing below the location of the secondary convective tower. As for the DG simulation, in the bottom right panel of Fig. 4 we notice that the rain accumulation matches the downwind shifting of the main column and small peaks appear where secondary convective towers are present.

Regardless of the space discretization method, we see that once rain appears within the convective tower it is correctly transported downward without the need for a vertically structured grid. This hold true also when multiple, possibly disconnected, sources of rain are present in the domain. In both sets of simulations, the rain falls to the ground following the location of the convective towers and the effects of the wind-shear. This gives us confidence that our algorithm is able to correctly transport rain despite the lack of a vertically structured grid and regardless of the space discretization method.

The results obtained with the $\Delta x = 250, 200, 150, 100$ m meshes at $t = 9000$ s are compared in Fig. 5 for the CG approximation and in Fig. 6 for the DG approximation. In Fig. 5, we observe the same cloud structure (anvil extent, downwind tilt of the

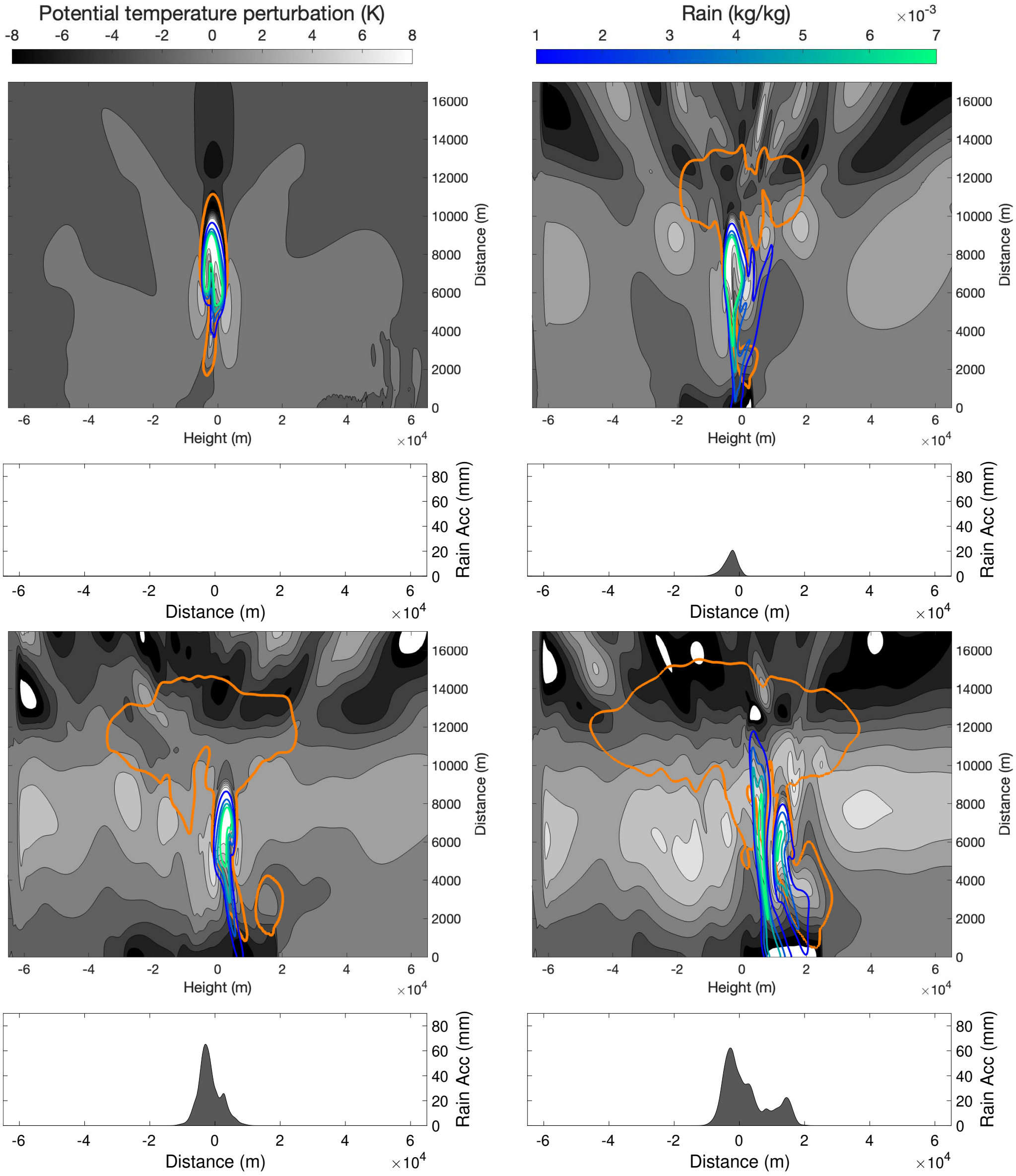

Figure 3: Storm evolution obtained with a CG approximation and mesh with resolution $\Delta x = 100$ m at t =1500 s (top-left), 3000 s (top-right), 6000 s (bottom-left) and 9000 s (bottom-right). In the top portion of each panel, the thick orange contour line ($q_c = 10^{-5}$ kgkg^{-1}) represents the outline of the cloud. The white and gray contours represent the perturbation potential temperature, and the blue and green contours represent q_r. The bottom portion of each panel shows the rain accumulated at the surface for each time as a function of horizontal distance from the point $x = 0$ m.

convective tower) and similar profiles of perturbation potential temperature for all the meshes under consideration. However, the spatial distributions of the rainfall accumulated at the ground show some differences: the simulations with resolutions $\Delta x = 250$ m and $\Delta x = 200$ m have smaller peaks of rain accumulation near the domain center than the simulations with $\Delta x = 150$ m and $\Delta x = 100$ m. The simulations with the $\Delta x = 290, 500, 750$ m meshes (not shown for brevity) give even more intense rainfall than the

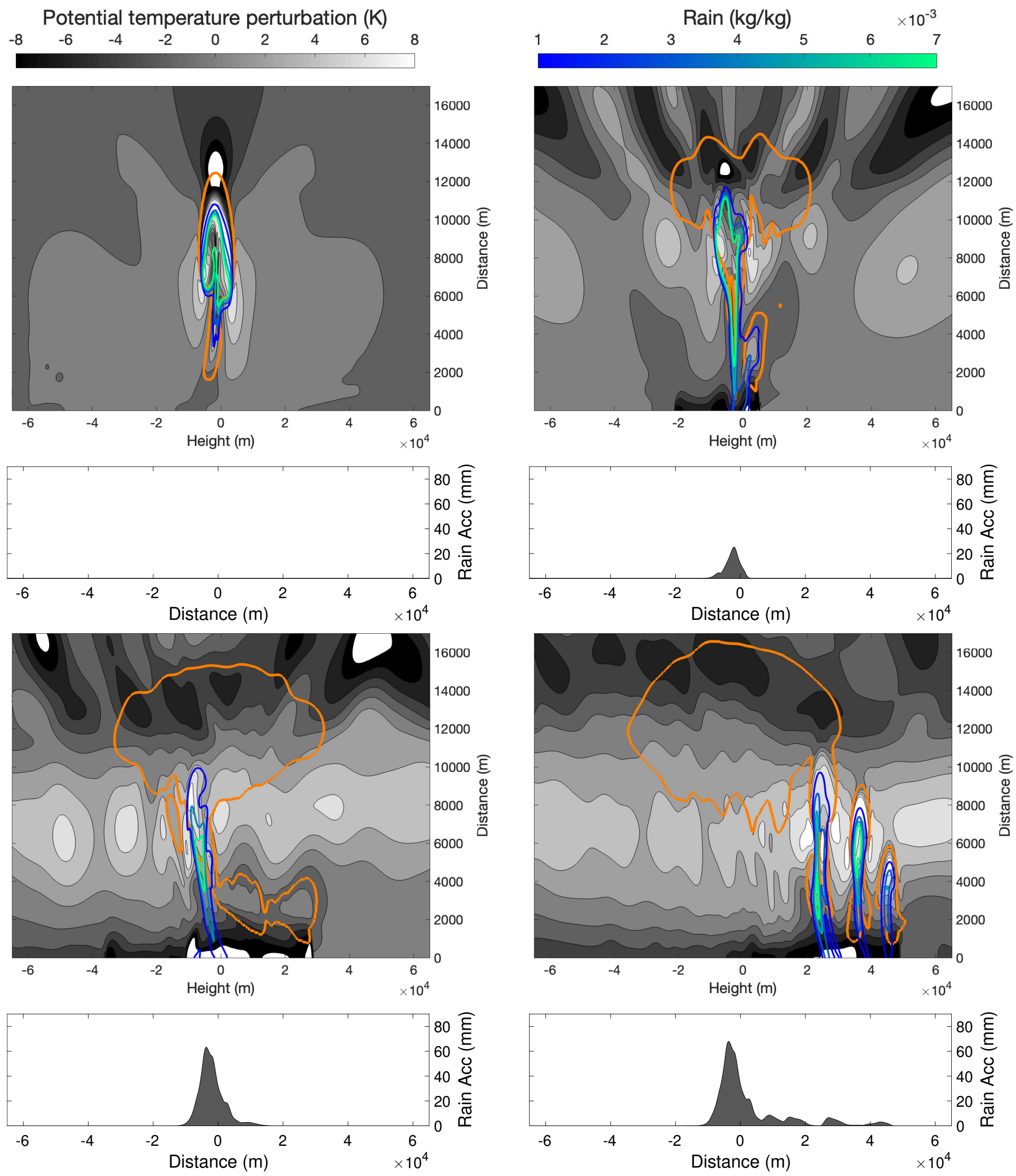

Figure 4: Storm evolution obtained with a DG approximation and mesh with resolution $\Delta x = 100$ m at t =1500 s (top-left), 3000 s (top-right), 6000 s (bottom-left) and 9000 s (bottom-right). In the top portion of each panel, the thick orange contour line ($q_c = 10^{-5}$ kgkg^{-1}) represents the outline of the cloud. The white and gray contours represent the perturbation potential temperature and the blue and green contours represent q_r. The bottom portion of each panel shows the rain accumulated at the surface for each time as a function of horizontal distance from the point $x = 0$ m.

$\Delta x = 250$ m and $\Delta x = 200$ m simulations. A similar observation on rain accumulation and mesh resolution for this benchmark can be found in (Weisman et al., 1997; Gaberšek et al., 2012), where it is shown that higher resolutions are correlated with faster storm development, weaker storm circulation and less overall precipitation over the length of the simulation. The DG simulations also show similar tilt in the convective tower, similar anvil extents and similar profiles of perturbation potential temperature at $t = 9000$

s for all the meshes; see Fig. 6. Concerning the rain accumulation, the DG simulation with the $\Delta x = 250$ m mesh gives a very large primary and secondary peak near the center of the domain. The amount of rain falling at the domain center decreases with increasing resolution. Indeed, the $\Delta x = 200, 150$ m simulations give a smaller amount of accumulated rain in the domain center and slightly larger peaks downwind and away from the center, reflecting the availability of more moisture for the secondary convective tower. Once again, we observe a decrease in precipitation with increasing resolution as expected (Gaberšek et al., 2012; Weisman et al., 1997; Marras et al., 2013b; Marras & Giraldo, 2015).

We conclude by reporting the maximum vertical velocity obtained over the course of the CG and DG simulation as a function of the resolution in Fig. 7. We see that for $\Delta x \geq 290$ m the maximum vertical velocity for both DG and CG simulations lies between 20 ms^{-1} and 30 ms^{-1}, as in (Bryan et al., 2006; Weisman & Rotunno, 2004; Gaberšek et al., 2012). Increasing the resolution yields an increase in the maximum velocity, as shown in (Gaberšek et al., 2012). We note that the CG and DG simulations give similar values of the maximum vertical velocity for a given mesh, with the values getting closer as the resolution increases.

The results in this section demonstrate that our algorithm successfully transports the rain downwards along the convective towers without the need for a vertically structured grid.

4.2 3D supercell

In this section we test our algorithm for a fully three-dimensional supercell. The convective cell develops within a domain $\Omega = [150 \times 100 \times 24]$ km^3. The storm is initiated by a thermal perturbation of the background state defined by (65), with center $(x_c, y_c, z_c) = (75000, 50000, 2000)$ m and

$$r = \sqrt{\frac{(x - x_c)^2}{r_x^2} + \frac{(y - y_c)}{r_y^2} + \frac{(z - z_c)^2}{r_z^2}}, \quad \theta_c = 3 \text{ K}, \quad r_c = 1,$$

where:

$$r_x = r_y = 10000 \text{ m}, \quad r_z = 2000 \text{ m}.$$

The domain is discretized using a grid of unstructured hexahedra of order 4 in all directions for an approximate effective resolution $\mathbf{\Delta x} \approx 250$ m. The grid is partially shown in Fig. 8.

We use periodic boundary conditions for the lateral boundaries, a free-slip boundary at the domain bottom and a Rayleigh sponge at the domain top. Like for the squall line test described above, we use the ARK3 3D semi-implicit time integrator to advance the simulation in time and keep the acoustic Courant number $C \leq 1$. An artificial viscosity $\beta = 200$ (see Remark 2.2 for the units) is used to provide stabilization. The wind shear in the x direction is the same as the one used for the squall-line. The cloud begins to form at $t \approx 500$ s while rain forms and starts to precipitate at $t \approx 900$ s.

A 3D view of the fully developed storm at $t = 7200$ s is shown in Fig. 8, along with a partial view of the three-dimensional grid. The semi-transparent blue shading is the iso-surface $q_r = 1e{-}4$ kg/kg. The blue shading is the perturbation potential temperature (blue is negative) showing the cold pools due to rain evaporation. All of the convective towers exhibit tilting due to wind-shear, with the parts closer to the ground experiencing a greater wind-shear and thus trailing the rest of the convective tower. An anvil cloud is also observed near the top of the troposphere.

Fig. 9 shows the state of the storm at $t = 7200$ s. The right side of the figure shows the existence of 3 distinct convective towers in the supercell. One in the center of the

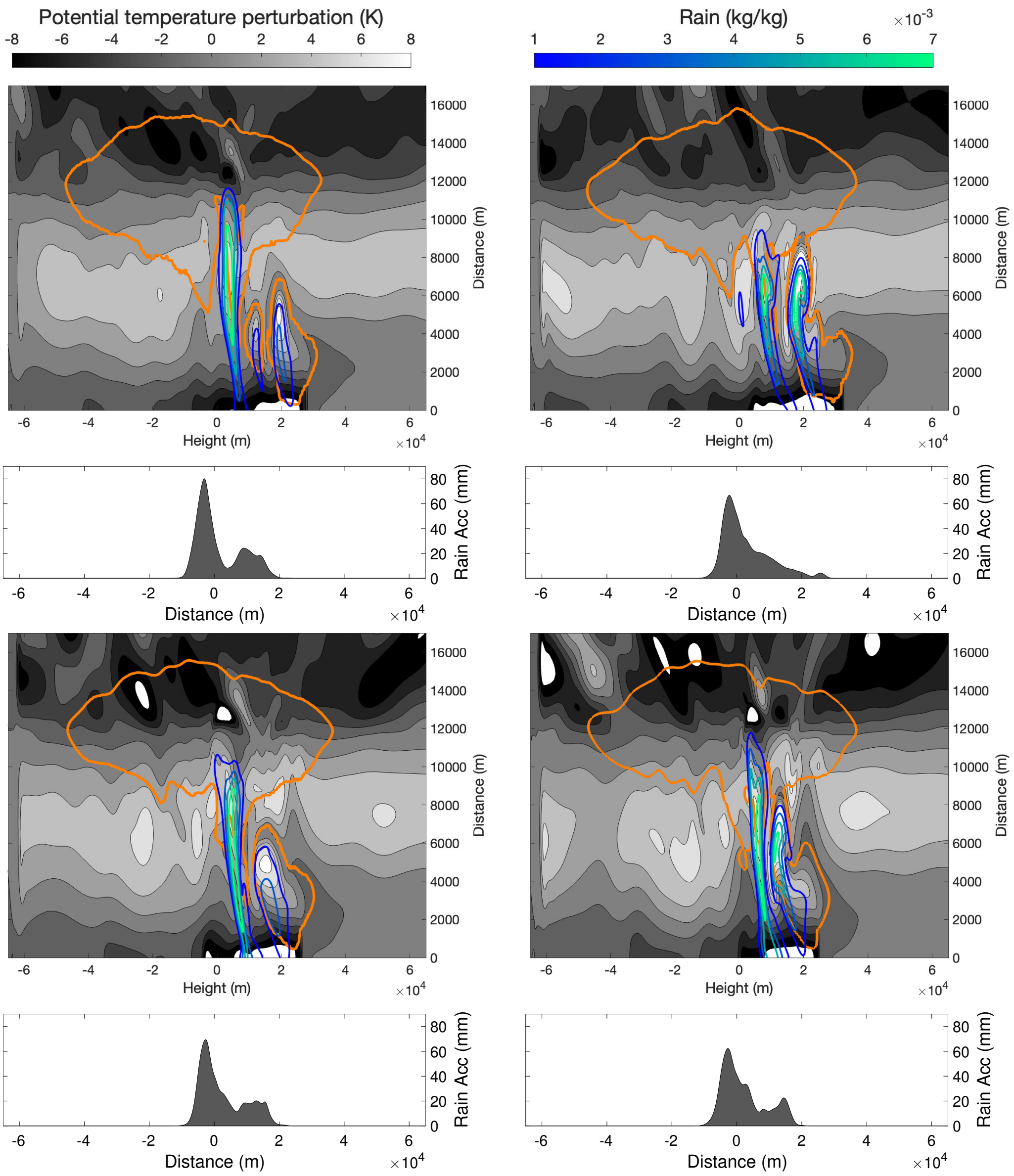

Figure 5: Storm at $t = 9000$ s computed with the CG method and meshes $\Delta x = 250$ m (top-left), $\Delta x = 200$ m (top-right), $\Delta x = 150$ m (bottom-left), and $\Delta x = 100$ m (bottom-right). The thick orange contour line ($q_c = 10^{-5}$ kgkg^{-1}) represents the outline of the cloud. The white and gray contours represent the perturbation potential temperature and the blue and green contours represent q_r. The bottom portions of each panel show the rain accumulated at the surface as a function of horizontal distance from the point $x = 0$ m.

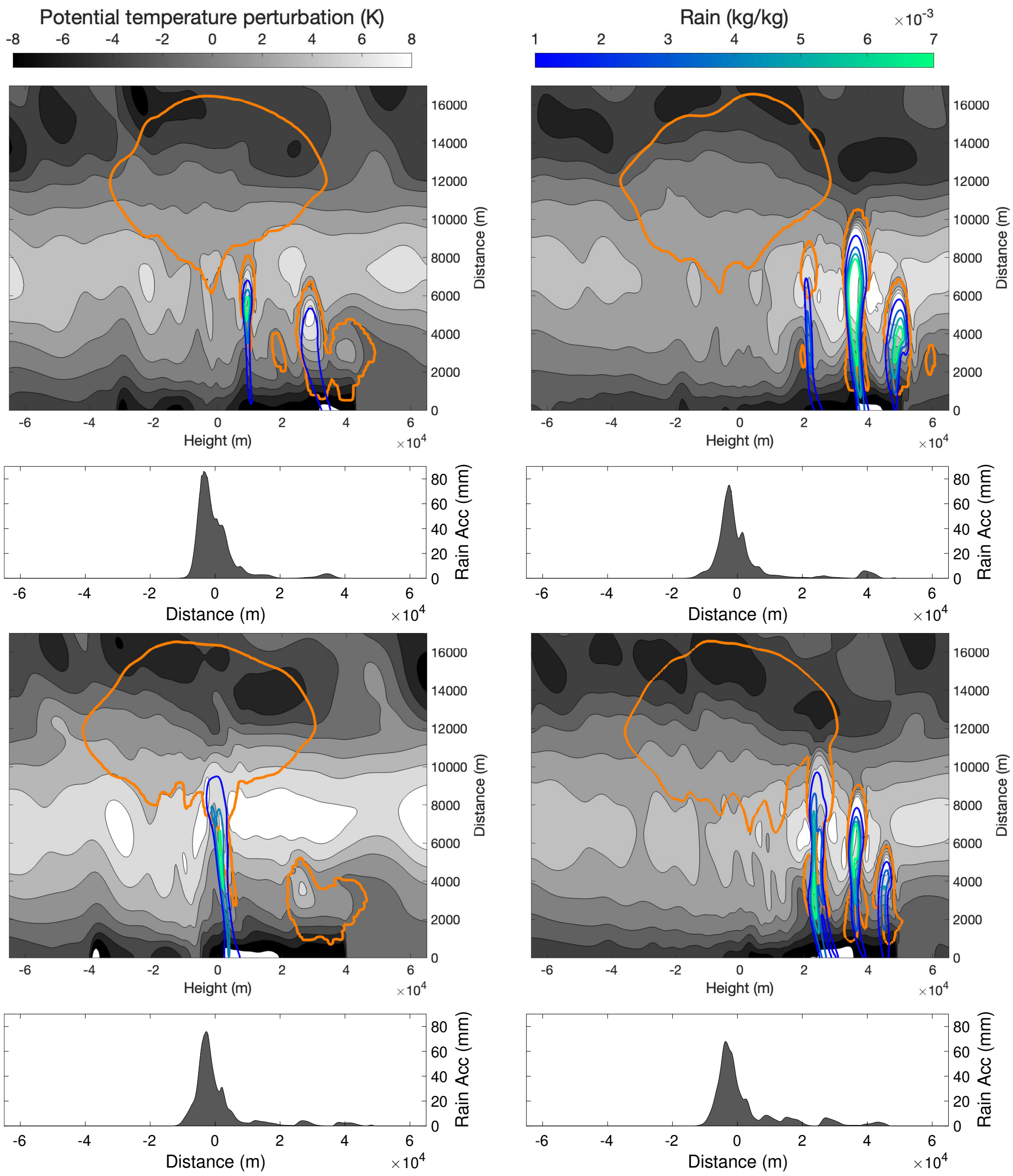

Figure 6: Storm at $t = 9000$ s computed with the DG method and meshes $\Delta x = 250$ m (top-left), $\Delta x = 200$ m (top-right), $\Delta x = 150$ m (bottom-left), and $\Delta x = 100$ m (bottom-right). The thick orange contour line ($q_c = 10^{-5}$ kgkg^{-1}) represents the outline of the cloud. The white and gray contours represent the perturbation potential temperature and the blue and green contours represent q_r. The bottom portions of each panel show the rain accumulated at the surface as a function of horizontal distance from the point $x = 0$ m.

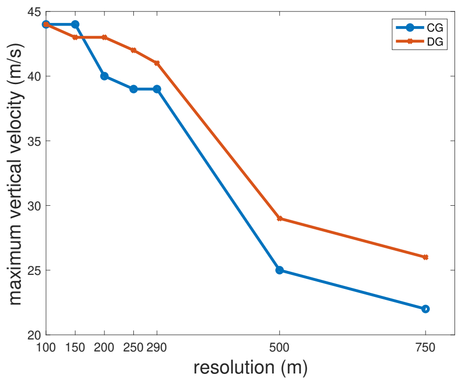

Figure 7: Maximum vertical velocity obtained over the course of the CG and DG simulations as a function of the resolution.

Y axis at $y = 50000$ m and two columns symmetric about $y = 50000$ m plane. The three towers merge into the anvil cloud near the tropopause. Fig. 9 (left) shows the rain distribution at the ground at $t = 7200$ s. The position of the rain concentration follows the location of the convective towers, falling below them. The largest amount of rain is present below the larger central tower as indicated by the maximum over $y = 50000$ m. Additionally we can see the presence of some rain slightly separated from the main rain distribution which corresponds to the small low clouds that are shown symmetric to the $y = 50000$ m plane in the right side of the figure.

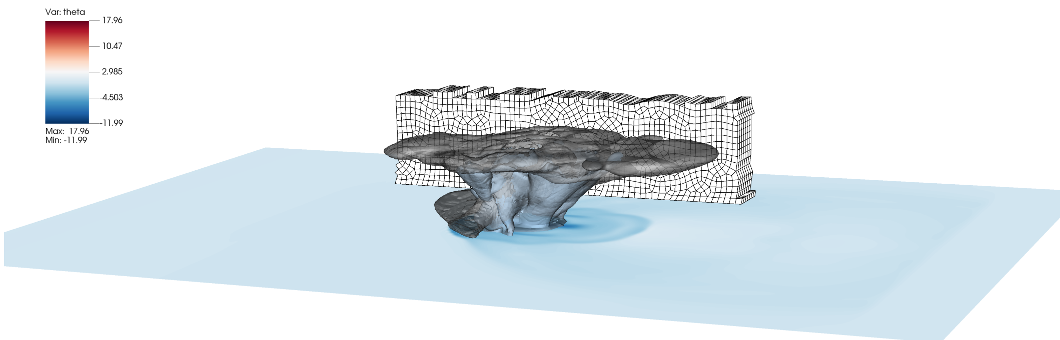

Figure 8: 3D mature supercell at $t = 7200$ s. The grey shading is the iso-surface $q_c = 1e - 5$ kg/kg. The semi-transparent blue shading is the iso-surface $q_r = 1e - 4$ kg/kg. The blue shading is the perturbation potential temperature (blue is negative) showing the cold pools due to rain evaporation. A small sample of the three-dimensional unstructured grid is shown in the background.

The results presented in this section show that the storm develops in a symmetrical manner and the rain falls correctly following the location of the convective towers, as is expected. This is accomplished without a column based grid. This demonstrates that our algorithm successfully transports the rain downward along the convective towers without the need for a vertically structured grid also in three dimensions.

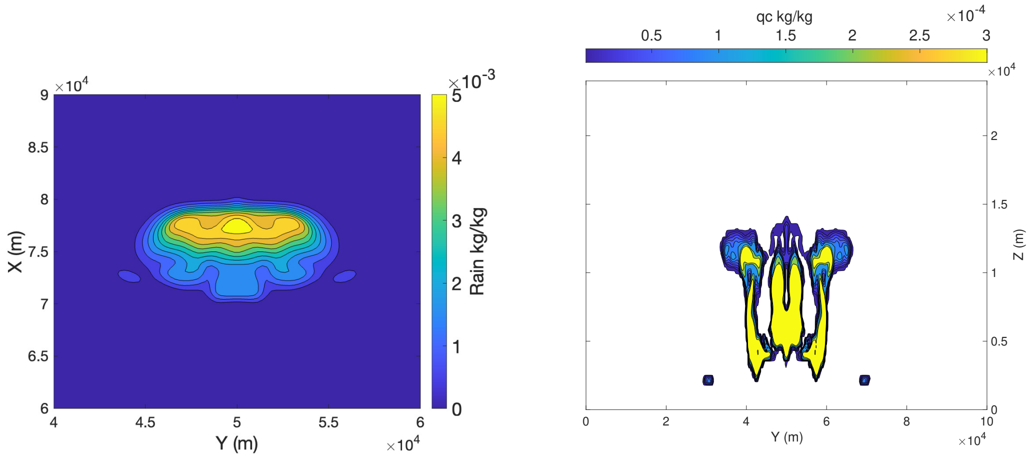

Figure 9: State of the storm at $t = 7200$ s. Left: Horizontal cross-section of the instantaneous distribution of rain along the surface ($z = 0$ m) at $t = 7200$ s. Right: Vertical cross section taken at $x = 75000$ m of the cloud fraction $t = 7200$ s

5 Conclusions

We presented an algorithm to solve the transport equation of precipitating clouds and Kessler's microphysical processes on fully unstructured grids. The Euler equations of moist atmospheric flows (embedded with artificial diffusion for stabilization purposes) were discretized by 4^{th}-order continuous and discontinuous spectral elements in space and advanced in time by a 3^{rd}-order additive Runge-Kutta semi-implicit time integrator. The results of these simulations are in very good agreement with results in the literature obtained using vertically structured meshes and column-based microphysics. This shows that the algorithm, while simple, does succeed in handling moisture with unstructured grids.

Coupled with the flexibility of the spectral element method, we believe that our algorithm could successfully resolve storms over steep terrain (Marisco & Stechmann, 2020) using unstructured meshes with and without adaptive mesh refinement, without the need for a special physics grid on which to handle moisture. Work in this direction is recommended. While we presented results only for warm rain, extension to other moist precipitation processes is natural. Probably the greatest advantage of fully unstructured atmospheric simulations is the fact that parallel load balancing decomposition can be done in any direction, which is of fundamental importance for efficient exascale simulations of high-resolution weather and climate modeling.

6 Data Availability Statement

All data presented in the paper and the source code with the unstructured algorithm are available on a public github repository through Zeonodo via this DOI `https://doi.org/10.5281/zenodo.6787870`, with the GNU General Public License v3.0. (Tissaoui et al., 2022)

Author contributions

Yassine Tissaoui: Methodology, Software, Validation, Formal Analysis, Investigation, Visualization, Writing. **Simone Marras, PI**: Conceptualization, Methodology, Software, Writing, Review, Editing, Supervision. **Annalisa Quaini**: Writing, Review, Editing. **Felipe A. V. de Braganca**: Software. **Francis X. Giraldo**. Software, Writing, Review, Editing.

Acknowledgments

The authors are grateful to Dr. James F. Kelly from the U.S. Naval Research Laboratory (Washington, DC) for his feedback on the manuscript. Yassine Tissaoui and Simone Marras acknowledge the partial support by the National Science Foundation through grant PD-2121367. Annalisa Quaini acknowledges partial support by the National Science Foundation through grant DMS-1953535 and support from the Radcliffe Institute for Advanced Study at Harvard University where she has been the 2021-2022 William and Flora Hewlett Foundation Fellow. Francis Giraldo gratefully acknowledges the support of ONR under grant # N0001419WX00721. Francis Giraldo and Felipe Alves gratefully acknowledge support from the Defense Sciences Office of the Defense Applied Research and Projects Agency (DARPA DSO) through the Space Environment Exploitation (SEE) program. This work was performed when Felipe Alves held a National Academy of Sciences' National Research Council (NRC) Fellowship at the Naval Postgraduate School. Yassine Tissaoui and Simone Marras gratefully acknowledge the Extreme Science and Engineering Discovery Environment (XSEDE) for providing core hours on Bridges-2 with allocation TG-EES210027.

References

Abdi, D., Giraldo, F. X., Constantinescu, C., E. Lester III, Wilcox, L., & Warburton, T. C. (2017). Acceleration of the Implicit-Explicit Non-Hydrostatic Unified Model of the Atmosphere (NUMA) on Manycore Processors. *Int. J. High Perform. Comput.*. doi: 10.1177/1094342017732395

Abdi, D. B., & Giraldo, F. X. (2017). Efficient construction of unified continuous and discontinuous Galerkin formulations for the 3D Euler equations. *J. Comput. Phys.*, *320*, 46-68.

Atlas, R., Reale, O., Shen, B.-W., Lin, S.-J., Chern, J.-D., Putman, W., & et al. (2005). Hurricane forecasting with the high-resolution NASA finite volume general circulation model. *Geophysical Research Letters*, *32(3)*.

Aubry, R., Vázquez, M., Houzeaux, G., Cela, J. M., & Marras, S. (2010). An unstructured CFD approach to numerical weather prediction. In *Proceedings: 48th aiaa aerospace sciences meeting, 4-7 january 2010, orlando, florida. aiaa paper 691-783.*

Bacmeister, J., Wehner, M., Neale, R., Gettelman, A., Hannay, C., Lauritzen, P., & et al. (2014). Exploratory high-resolution climate simulations using the Community Atmosphere Model (CAM). *Journal of Climate*, *27(9)*, 3073–3099.

Baldauf, M. (2021). A horizontally explicit, vertically implicit (HEVI) discontinuous Galerkin scheme for the 2-dimensional Euler and Navier-Stokes equations using terrain-following coordinates. *J. Comput. Phys.*, *446*, 110635. doi: https://doi.org/10.1016/j.jcp.2021.110635

Bryan, G. H., Knievel, J. C., & Parker, M. D. (2006). A multimodel as- sessment of RKW theory's relevance to squall-line charac- teristics. *Mon. Wea. Rev.*, *134*, 2772-2792.

Burstedde, C., Wilcox, L. C., & Ghattas, O. (2011). `p4est`: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing*, *33*(3), 1103-1133. doi: 10.1137/100791634

Caldwell, P. M., Mametjanov, A., Tang, Q., Van Roekel, L. P., Golaz, J.-C., Lin, W., & et al. (2019). The DOE E3SM coupled model version 1: Description and results at high resolution. *Journal of Advances in Modeling Earth Systems*, *11*, 4095–4146.

COSMO, P. (1998). Consortium for Small-scale Modeling. cosmo project. *www.cosmo-model.org/content/model/documentation.*

Danabasoglu, G., Lamarque, J.-F., Bacmeister, J., Bailey, D. A., DuVivier, A. K., Edwards, J., ... Strand, W. G. (2020). The community earth system model version 2 (cesm2). *Journal of Advances in Modeling Earth Systems*, *12*(2), e2019MS001916.

Delworth, T., Rosati, A., Anderson, W., Adcroft, A. J., Balaji, V., Benson, R., & et al. (2012). Simulated climate and climate change in the GFDL CM2.5 high-resolution coupled climate model. *Journal of Climate*, *25*, 2755–2781.

Dennis, J., Edwards, J., Evans, K. J., Guba, O., Lauritzen, P. H., Mirin, A. A., ... Worley, P. H. (2012). CAM-SE: A scalable spectral element dynamical core for the Community Atmosphere Model. *Int. J. High Perf. Comput. Appl.*, *26*, 74-89.

Duffy, P., Govindasamy, B., Iorio, J., Milanovich, J., Sperber, K., Taylor, K., & et al. (2003). High-resolution simulations of global climate, Part 1: Present climate. *Climate Dynamics*, *21*, 371–390.

Gaberšek, S., Giraldo, F. X., & Doyle, J. (2012). Dry and moist idealized experiments with a two-dimensional spectral element model. *Mon. Wea. Rev.*, *140*, 3163-3182.

Geuzaine, C., & Remacle, J. (2009). Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *Internation Journal for Numerical Methods in Engineering*, *0*, 1-24.

Giorgi, F., & Marinucci, M. R. (1996). A investigation of the sensitivity of simulated precipitation to model resolution and its implications for climate studies. *Mon. Wea. Rev.*, *124*, 148-166.

Giraldo, F. X. (2020). *An introduction to element-based galerkin methods on tensor-product bases - analysis, algorithms, and applications* (Vol. 24). Springer.

Giraldo, F. X., Kelly, J. F., & Constantinescu, E. (2013). Implicit-explicit formulations of a three-dimensional Nonhydrostatic Unified Model of the Atmosphere (NUMA). *SIAM J. Sci. Comput.*, *35*, 1162-1194.

Hesthaven, J. S., & Warburton, T. (2008). *Nodal discontinuous galerkin methods: algorithms, analysis, and applications* (Vol. 54). Springer-Verlag New York Inc.

Houze, R. A. (1993). *Cloud dynamics.* Academic Press.

Iorio, J., Duffy, P., Govindasamy, B., Thompson, S., Khairoutdinov, M., & Randall, D. (2004). Effects of model resolution and subgrid-scale physics on the simulation of precipitation in the continental United States. *Climate Dynamics*, *23*, 243–258.

Kelly, J. F., & Giraldo, F. X. (2012). Continuous and discontinuous Galerkin methods for a scalable three-dimensional nonhydrostatic atmospheric model: limited-area mode. *J. Comput. Phys.*, *231*, 7988-8008.

Kennedy, C., & Carpenter, M. (2003). Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Applied Numerical Mathematics*, *44*(3), 139-181.

Kessler, E. (1969). On the distribution and continuity of water substance in atmospheric circulation. *Meteorol. Monogr.*, *10*, 32.

Klemp, J., & Wilhelmson, R. (1978). The simulation of three-dimensional convective storm dynamics. *J. Atmos. Sci.*, *35*, 1070-1096.

Kopera, M. A., & Giraldo, F. X. (2014). Analysis of adaptive mesh refinement for IMEX discontinuous Galerkin solutions of the compressible Euler equations with application to atmospheric simulations. *J. Comput. Phys.*, *275*, 92-117.

Kopriva, D. (2008). *Implementing Spectral Methods for Partial Differential Equations.* Springer.

Lauritzen, B., Bacmeister, J., Callaghan, P., & Taylor, M. (2015). NCAR_Topo (v1.0): NCAR global model topography generation software for unstructured grids. *Geophysical Model Development*, *8*, 3975–3986.

Li, J., Fang, F., Steppeler, J., Zhu, J., Cheng, Y., & Wu, X. (2021). Demonstration of a three-dimensional dynamically adaptive atmospheric dynamic framework for the simulation of mountain waves. *Meteorology and Atmospheric Physics*.

Love, B., Matthews, A., & Lister, G. (2011). The diurnal cycle of precipitation over the maritime continent in a high-resolution atmospheric model. *Quaterly Journal of the Royal Meteorological Society*, *137*, 934–947.

Marisco, D., & Stechmann, S. (2020). Expanding grids for efficient cloud dynamics simulations across scales. *Math. Clim. Weather Forecast.*, *6*, 38–49.

Marras, S., & Giraldo, F. X. (2015). A parameter-free dynamic alternative to hyper-viscosity for coupled transport equations: application to the simulation of 3D squall lines using spectral elements. *J. Comput. Phys.*, *283*, 360-373.

Marras, S., Moragues, M., Vázquez, M., Jorba, O., & Houzeaux, G. (2013b). A Variational Multiscale Stabilized finite element method for the solution of the Euler equations of nonhydrostatic stratified flows. *J. Comput. Phys.*, *236*, 380-407.

Marras, S., Moragues, M., Vázquez, M. R., Jorba, O., & Houzeaux, G. (2013a). Simulations of moist convection by a variational multiscale stabilized finite element method. *J. Comput. Phys.*, *252*, 195-218.

Marras, S., Nazarov, M., & Giraldo, F. X. (2015). Stabilized high-order Galerkin methods based on a parameter-free dynamic SGS model for LES. *J. Comput. Phys.*, *301*, 77-101.

Müller, A., Kopera, M., Marras, S., Wilcox, L. C., Isaac, T., & Giraldo, F. X. (2018). Strong scaling for numerical weather prediction at petascale with the atmospheric model NUMA. *The International Journal of High Performance Computing Applications*, *33*, 411-426.

Nelson, D. A., Jacobs, G. B., & Kopriva, D. A. (2016). Effect of boundary representation on viscous, separated flows in a discontinuous-Galerkin Navier-Stokes solver. *Theor. Comput. Fluid Dyn.*, 363-385.

Nikiforakis, N. (2009). Mesh generation and mesh adaptation for large-scale Earth-system modelling, Introduction. *Phil. Trans. R. Soc.*, *367*, 4473-4481.

Ogura, Y., & Takahashi, T. (1971). Numerical simulation of the lifecycle of a thunderstorm cell. *Mon. Wea. Rev.*, *99*, 895-911.

Pope, V., & Stratton, R. (2002). The processes governing horizontal resolution sensitivity in a climate model. *Climate Dynamics*, *19*, 211–236.

Reddy, S., Tissaoui, Y., Alves, F. A. V. D. B. a., Marras, S., & Giraldo, F. X. (2021). Comparison of Sub-Grid Scale Models for Large-Eddy Simulation using a High-Order Spectral Element Approximation of the Compressible Navier-Stokes Equations at Low Mach Number. *arXiv:2204.03160 [physics.flu-dyn]*. doi: https://doi.org/10.48550/arXiv.2204.03160

Rotunno, R., Klemp, J. B., & Weisman, M. L. (1988). A theory for strong, long-lived squall lines. *J. Atmos. Sci.*, *45*, 463-485.

Sherwin, S., & Karniadakis, G. (2005). *Spectral/hp Element Methods for Computational Fluid Dynamics.* Oxford University Press.

Skamarock, W., Klemp, J., Duda, M., Fowler, L., Park, S.-H., & Ringler, T. (2012). A multiscale nonhydrostatic atmospheric model using centroidal Voronoi tesselations and C-grid staggering. *Mon. Wea. Rev.*, *140*(9), 3090-3105.

Smolarkiewicz, P. K., Szmelter, J., & Wyszogrodzki, A. A. (2013). An unstructured-mesh atmospheric model for nonhydrostatic dynamics. *J. Comput. Phys.*, *254*, 184–199.

Soong, S., & Ogura, Y. (1973). A comparison between axisymmetric and slab-symmetric cumulus cloud models. *J. Atmos. Sci.*, *30*, 879-893.

Sridhar, A., Tissaoui, Y., Marras, S., Shen, Z., Kawczynski, C., Byrne, s., ... Schneider, T. (2022). Large-eddy simulations with ClimateMachine v0.2.0: a new open-source code for atmospheric simulations on GPUs and CPUs. *Geoscient. Model Develop.*, *gmd-2021-335*.

Szmelter, J., Zhang, Z., & Smolarkiewicz, P. K. (2015). An unstructured-mesh atmospheric model for nonhydrostatic dynamics: towards optimal mesh resolution. *J. Comput. Phys.*, *294*, 363-381.

Terai, C., Caldwell, P., Klein, S., Tang, Q., & Branstetter, M. (2018). The atmospheric hydrologic cycle in the ACME v0.3 model. *ClimateDynamics*, *50*, 3251–3279.

Tissaoui, Y., Marras, S., Quaini, A., De Braganca Alves, F. A. V., & Giraldo, F. X. (2022). *A non-column based, fully unstructured implementation of Kessler's microphysics with warm rain using continuous and discontinuous spectral elements* [dataset]. Retrieved from `https://doi.org/10.5281/zenodo.6787870` doi: 10.5281/zenodo.6787270

Wehner, M., Reed, K., Li, F., Prabhat, Bacmeister, J., Chen, C.-T., & et al. (2014). The effect of horizontal resolution on simulation quality in the Community Atmospheric Model, CAM5.1. *Journal of Advances in Modeling Earth Systems*, *6*, 980–997.

Weisman, M. L., & Klemp, J. B. (1982). The dependence of numerically simulated convective storms on vertical wind shear and buoyancy. *Mon. Wea. Rev.*, *110*, 504-520.

Weisman, M. L., Klemp, J. B., & Rotunno, R. (1988). Structure and evolution of numerically simulated squall lines. *J. Atmo. Sci.*, *45*, 1990-2013.

Weisman, M. L., & Rotunno, R. (2004). A theory for strong long- lived squall lines revisited. *J. of Atmos. Sci.*, *61*, 361.

Weisman, M. L., Skamarock, W. C., & Klemp, J. B. (1997). The resolution dependence of explicitly modeled convective systems. *Mon. Wea. Rev.*, *125*, 527.

Yamazaki, H., Weller, H., Cotter, C. J., & Browne, P. A. (2022). Conservation with moving meshes over orography. *J. Comp. Phys.*, *461*, 111217.

597 # Appendix

Table 2: Squall line sounding

z (m)	θ (K)	q_v (g/kg)	u (m/s)	v (m/s)	p (Pa)
0.0	303.025079	14.000	12.0	0.0	100000.0
480.0	303.337272	14.000	9.696000	0.0	94697.28
960.0	304.402985	14.000	7.392000	0.0	89609.81
1440.0	305.397187	12.796	5.088000	0.0	84736.79
1920.0	306.306214	10.556	2.784000	0.0	80070.30
2400.0	307.365269	8.678	0.540000	0.0	75604.36
2880.0	308.550318	7.104	0.0	0.0	71334.51
3360.0	309.845257	5.788	0.0	0.0	67255.79
3840.0	311.235047	4.691	0.0	0.0	63362.95
4320.0	312.708238	3.777	0.0	0.0	59650.49
4800.0	314.255743	3.020	0.0	0.0	56112.80
5280.0	315.869985	2.396	0.0	0.0	52744.15
5760.0	317.544512	1.885	0.0	0.0	49538.82
6240.0	319.273784	1.469	0.0	0.0	46491.09
6720.0	321.052868	1.134	0.0	0.0	43595.27
7200.0	322.877588	0.866	0.0	0.0	40845.73
7680.0	324.744235	0.653	0.0	0.0	38236.93
8160.0	326.649534	0.487	0.0	0.0	35763.41
8640.0	328.590559	0.357	0.0	0.0	33419.84
9120.0	330.565013	0.259	0.0	0.0	31200.99
9600.0	332.571020	0.184	0.0	0.0	29101.75
10080.0	334.606102	0.129	0.0	0.0	27117.17
10560.0	336.668475	0.088	0.0	0.0	25242.39
11520.0	340.869535	0.038	0.0	0.0	21803.59
12000.0	343.712008	0.026	0.0	0.0	20232.15
12480.0	350.647306	0.026	0.0	0.0	18763.71
12960.0	358.453724	0.029	0.0	0.0	17401.15
13440.0	366.433620	0.031	0.0	0.0	16138.11
13920.0	374.591035	0.034	0.0	0.0	14967.29
14400.0	382.929618	0.037	0.0	0.0	13881.93
15360.0	400.170355	0.044	0.0	0.0	11942.99
15840.0	409.081924	0.049	0.0	0.0	11078.24
16320.0	418.191751	0.053	0.0	0.0	10276.53
16800.0	427.504224	0.058	0.0	0.0	9533.23
17280.0	437.023716	0.063	0.0	0.0	8844.07
17760.0	446.755038	0.069	0.0	0.0	8205.09
18720.0	466.871821	0.083	0.0	0.0	7063.24
19200.0	477.267160	0.091	0.0	0.0	6553.82
19680.0	487.891998	0.094	0.0	0.0	6081.42
20160.0	498.742611	0.094	0.0	0.0	5643.35
20640.0	509.643457	0.094	0.0	0.0	5237.00
21120.0	520.544304	0.094	0.0	0.0	4859.92
21600.0	531.445151	0.094	0.0	0.0	4509.85
22560.0	553.246845	0.094	0.0	0.0	3882.66
23040.0	564.147692	0.094	0.0	0.0	3601.93
23520.0	575.048539	0.094	0.0	0.0	3340.96
24000.0	585.949386	0.094	0.0	0.0	3098.30