



A discontinuous Galerkin approach for atmospheric flows with implicit condensation

Sabine Doppler^a, Philip L. Lederer^b, Joachim Schöberl^c, Henry von Wahl^{a,d,*}

^a University of Vienna, Faculty of Mathematics, Austria

^b University of Twente, Department of Applied Mathematics, Netherlands

^c TU Wien, Institute for Analysis and Scientific Computing, Austria

^d Friedrich Schiller University Jena, Institute for Mathematics, Germany

ARTICLE INFO

Keywords:

Atmospheric flow
Discontinuous Galerkin
Matrix-free
High-order
Compressible Euler equations with source terms
Moisture
Implicit condensation
Hyperbolic conservation laws

ABSTRACT

We present a discontinuous Galerkin method for moist atmospheric dynamics, with and without warm rain. By considering a combined density for water vapour and cloud water, we avoid the need to model and compute a source term for condensation. We recover the vapour and cloud densities by solving a pointwise non-linear problem each time step. Consequently, we enforce the requirement for the water vapour not to be supersaturated implicitly. Together with an explicit time-stepping scheme, the method is highly parallelisable and can utilise high-performance computing hardware. Furthermore, the discretisation works on structured and unstructured meshes in two and three spatial dimensions. We illustrate the performance of our approach using several test cases in two and three spatial dimensions. In the case of a smooth, exact solution, we illustrate the optimal higher-order convergence rates of the method.

1. Introduction

Precipitation still causes one of the largest uncertainties in weather forecasts and climate models. This is evident, as the modelling has to build a bridge across many scales to arrive from the processes in and around precipitation particles at the evolutionary dynamics of clouds and cloud systems on the mesoscales. Thus, cloud models exist with very different ranges of complexity. To model the moisture balances, we implement a bulk microphysical warm cloud model corresponding to the Kessler scheme, which is widely used in meteorology. Then, compared to the dry dynamics, the thermodynamic equation gets much more complex, as it provides a strong coupling to the moisture balances. This additional complexity is also due to the different heat capacities for vapour and liquid water and the dependence of latent heat on the temperature, which are often neglected. In this work, we retain all these thermodynamic details, as they have been demonstrated to be essential, e.g. in the case of deep convective clouds via asymptotic analysis in [1].

While the equations of dry air are generally accepted, the equations governing cloudy air are still actively debated. Therefore, it is necessary to develop numerical methods to simulate these complex equations accurately and efficiently. Typically, numerical schemes are established for individual model reductions obtained by scale analysis. For an overview of numerical methods for numerical weather prediction, see, for example, [2] and the references therein.

* Corresponding author at: Friedrich Schiller University Jena, Institute for Mathematics, Germany.

E-mail addresses: sabine.doppler@univie.ac.at (S. Doppler), p.l.lederer@utwente.nl (P.L. Lederer), joachim.schoeberl@tuwien.ac.at (J. Schöberl), henry.von.wahl@uni-jena.de (H. von Wahl).

<https://doi.org/10.1016/j.jcp.2023.112713>

Received 23 May 2023; Received in revised form 12 December 2023; Accepted 13 December 2023

Available online 18 December 2023

0021-9991/© 2023 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

On the other hand, increasing, massively-parallel computing power allows for ever more fine-scale computations for weather and climate models. However, numerical methods have for the most part been based on structured grids and discretisations exploiting the structure of the computational grids. While the orthogonality of the grid can lead to a number of desirable properties, see, for example, [3], there are drawbacks in the setting of massively parallel high-performance computing hardware. In particular, load balancing can be an issue when the discretisation is limited to column-based subdivisions of the mesh. Furthermore, for global discretisations, structured grids on the globe lead to strong clustering of grid points at the poles, creating an additional bottleneck. To avoid these issues, it is desirable to consider discretisation methods that work on unstructured meshes. This brings the additional benefit that steep topographical changes can be modelled by unstructured meshes easily and accurately, which would cause severe distortion of structured meshes. In the context of dry atmospheric flows, methods based on unstructured meshes were first considered in [4,5].

While the equations modelling moist air are usually given in a non-conservative form, they can be written in the form of a non-linear hyperbolic conservation law system with source terms. This form of the equation lends itself to a discontinuous Galerkin (DG) discretisation, which we consider in this work. Additionally, DG methods work both on structured and unstructured meshes. The system of equations is essentially the compressible Euler equations with multiple densities and some additional constraints and source terms modelling the phase changes of water. The different densities model the different water phases. While discontinuous Galerkin methods for the compressible Euler equations have been studied very well [6–8], only a few approaches consider the system with moisture and even fewer include rain dynamics. Among the (discontinuous) element-based approaches for atmospheric motion, we mention [9] in the context of the Nonhydrostatic Unified Model of the Atmosphere (NUMA), where both continuous and discontinuous elements are used on unstructured grids and rain dynamics are included, and [10] in the context of the COSMO (Consortium for Small-scale Modelling) model. In the latter work, rain dynamics were not included. Furthermore, we mention [11], where a compatible finite element discretisation (partially discontinuous) was considered, based on the ENDGame model by the UK Met Office. While this work included rain dynamics, the discretisation relied on vertically structured grids. Finally, we mention ClimateMachine [12], where discontinuous elements using vertical columns on Cartesian grids were considered and implemented both on CPUs and GPUs.

A particular challenge for discretisations is the realisation of the source term, which models the phase change between water vapour and cloud water, i.e., *condensation*. The main cause of problems is that the source term is only non-zero when the atmosphere is fully saturated by water vapour. However, once the saturation threshold is reached, the condensation source term is not given as an explicit function of the other variables, but rather as the right-hand side term balancing another transport equation. To avoid this, we take an approach used, for example, in [12–14], and consider a single density for cloud water and water vapour, such that the phase change between these two does not have to be modelled. We then reconstruct the individual densities after every time-step by solving an algebraic non-linear problem.

While individual aspects of our approach have been used in the literature, our combination and realisation of these techniques is novel and goes beyond what is considered in most cases in related works. For example, while [12] essentially consider the same set of equations and use a similar approach to deal with condensation, their work is based on structured grids, does not consider rain and only presents a convergence study for fourth-order polynomials and in a dry test-case over a very short time. While [9] also considered a DG approach, including unstructured meshes and rain, the equations used are different and simplified. In particular, the momentum and potential temperature only take dry mass into account. Furthermore, a constant artificial viscosity term is added, which modifies the problem being solved, condensation is implemented through a source term and correction scheme, and no convergence results for the scheme is presented in [9]. In [15], the same set of equations with an artificial viscosity is used together with spectral elements. While higher-order polynomials up to order ten are used, this is only applied on structured quadrilateral meshes, and a diagonal mass matrix is only achieved through inexact integration. While the temporal convergence of the utilised IMEX scheme is studied, the spatial convergence of the scheme is not presented. A similar set of equations to those in [9] is also used in [10]. A convergence study for linear and quadratic elements is shown in [10]. However, rain was not included in that work. Finally, we mention [11], where again, a different set of equations, compared to our work, is used. Convergence for the moist case is also shown here for the piecewise constant and linear cases, and rain is included in further examples. We also note that [11] require structured meshes and only use discontinuous elements for parts of the discretisation. However, this then leads to a compatible finite element method.

The remainder of this work is structured as follows. In Section 2, we present the equations under consideration in this paper. This includes the microphysics parametrisation used and the derivation of the implicit condensation equations. Our discontinuous Galerkin approach is presented in Section 3, and we present a number of numerical examples taken from the literature and realised with this approach in Section 4. This includes a high-order convergence study for multiple polynomial orders in the moist case without rain. The results in Section 4 are based on the open-source finite element library NGSolve and are fully reproducible through openly accessible python scripts implementing the presented examples. Details of the thermodynamic constants are stated in Appendix A, and additional details on the numerical set-ups are given in Appendix B.

2. Governing equations

The equations of motion for cloudy air with warm rain are given in a conservative form as

$$\partial_t \rho_d + \nabla \cdot (\rho_d \mathbf{u}) = 0, \quad (1a)$$

$$\partial_t \rho_v + \nabla \cdot (\rho_v \mathbf{u}) = (S_{ev} - S_{cd}), \quad (1b)$$

$$\partial_t \rho_c + \nabla \cdot (\rho_c \mathbf{u}) = (S_{cd} - S_{au} - S_{ac}), \quad (1c)$$

$$\partial_t \rho_r + \nabla \cdot (\rho_r \mathbf{u} - \rho_r v_r \mathbf{e}_z) = (S_{au} + S_{ac} - S_{ev}), \quad (1d)$$

$$\partial_t (\rho \mathbf{u}) + 2\mathbf{\Omega} \times \rho \mathbf{u} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} - \rho_r v_r \mathbf{u} \otimes \mathbf{e}_z + \text{Id } p) = -\rho g \mathbf{e}_z, \quad (1e)$$

$$\partial_t E + \nabla \cdot ((E + p)\mathbf{u} - (c_l(T - T_{\text{ref}}) + \mathbf{u}^2/2)\rho_r v_r \mathbf{e}_z) = -\rho g \mathbf{e}_z \cdot \mathbf{u}, \quad (1f)$$

see, e.g. [16,13,17]. The variables ρ_d, ρ_v, ρ_c , and ρ_r are the densities of dry air, water vapour, cloud water and rain, respectively. While \mathbf{u} is related to the velocity of air, v_r is the terminal rain speed, T is the temperature, T_{ref} is the reference temperature and E is the sum of the internal and kinetic energy densities. The total density ρ is given as the sum of the component densities

$$\rho = \rho_d + \rho_v + \rho_c + \rho_r. \quad (2)$$

The vector $\mathbf{\Omega}$ is the Coriolis vector due to the Earth's rotation. As our simulations are on a time-scale where the effects from this are negligible, we will not consider this term further below. To close the system we have the pressure given by the equation of state

$$p = (\rho_d R_d + \rho_v R_v) T, \quad (3)$$

with the gas constants R_d, R_v for dry air and water vapour, respectively. The temperature can be recovered from the equation for the sum of the internal and kinetic energy densities

$$E = (c_{vd} \rho_d + c_{vv} \rho_v + c_l(\rho_c + \rho_r))(T - T_{\text{ref}}) + \rho_v (L_{\text{ref}} - R_v T_{\text{ref}}) + \rho \frac{1}{2} \mathbf{u} \cdot \mathbf{u}, \quad (4)$$

where c_{vd}, c_{vv} are the specific heats of dry air and water vapour, both at constant volume, respectively, while c_l is the specific heat of liquid water at the reference temperature. The constant L_{ref} is the latent heat of vaporization at the reference temperature. The values used for these quantities are summarized in the appendix in Table A.1. Finally, we note that while this form of the equations is rarely used for atmospheric modelling, it is the form that best lends itself to a discontinuous Galerkin discretisation. We refer to [18] for an overview of other common forms of the equations used in mesoscale atmospheric modelling.

2.1. Microphysics parametrisation

The source terms modelling the phase-changes of water are $S_{ev}, S_{cd}, S_{au}, S_{ac}$ which describe evaporation, condensation, auto-conversion of cloud water into rain droplets and the collection of cloud droplets by raindrops (accretion), respectively. The closure of the moist dynamics is based on the microphysics closure due to Kessler [19].

We take the specific form of the microphysics parametrisation from the COSMO model, as described in [20, Section 5.4]. The evaporation, auto-conversion and collecting-rain source terms are defined as

$$S_{ev} := (3.86 \times 10^{-3} - 9.41 \times 10^{-5}(T - T_{\text{ref}})) \left(1 + 9.1 \rho_r^{3/16}\right) (\rho_{vs} - \rho_v) \rho_r^{1/2}, \quad (5a)$$

$$S_{au} := 0.001 \max\{\rho_c - q_{au} \rho, 0\}, \quad (5b)$$

$$S_{ac} := 1.72 \rho_c \rho_r^{7/8}, \quad (5c)$$

with the auto-conversion threshold q_{au} , which is chosen as $q_{au} = 0$.

This leaves the condensation source term open. Let us now denote by $q_i = \rho_i / \rho_d$, the mixing ratios of the different water-phase densities with respect to the dry air density. The condensation source term is defined implicitly by (1b) via two assumptions:

(a) Water vapour is saturated in the presence of cloud water, i.e.,

$$\text{if } q_c > 0, \text{ then } q_v = q_{vs} \text{ and } S_{ev} = 0.$$

(b) Cloud water evaporates instantaneously in undersaturated regions, i.e.,

$$\text{if } q_v < q_{vs}, \text{ then } q_c = 0 \text{ and } S_{cd} = 0.$$

In oversaturated regions, the source term due to condensation S_{cd} is then defined through the relationship

$$\partial_t q_{vs} + \mathbf{u} \cdot \nabla q_{vs} = -S_{cd},$$

where the saturation vapour mixing ratio is approximated in a saturated atmosphere in terms of the saturation vapour pressure e_s by

$$q_{vs} = \frac{e e_s(T)}{p - e_s(T)}. \quad (6)$$

The saturation vapour pressure is recovered from the Clausius–Clapeyron relation [16], and by using a linear approximation of the latent heat of condensation under the assumption of constant specific heats. This leads to the saturation vapour pressure

$$e_s(T) = e_{\text{ref}} \left(\frac{T}{T_{\text{ref}}} \right)^{\frac{c_{pv}-c_l}{R_v}} \exp \left[\frac{L_{\text{ref}} - (c_{pv} - c_l)T_{\text{ref}}}{R_v} \left(\frac{1}{T_{\text{ref}}} - \frac{1}{T} \right) \right], \quad (7)$$

where $e_{\text{ref}} := e_s(T_{\text{ref}})$ and c_{pv} is the specific heat of water vapour at constant pressure, see also [21]. The specific values used for these constants are again shown in Table A.1.

Finally, again following the warm rain scheme of the COSMO model, the mean terminal velocity of rain is defined as

$$v_r = (\pi \rho_w N_0^r)^{-1/8} \frac{v_0^r \Gamma(4.5)}{6} \rho_r^{1/8},$$

with the distribution parameter set as $N_0^r = 8 \times 10^6 \text{ m}^{-4}$, $v_0^r = 130 \text{ m}^{1/2} \text{ s}^{-1}$ and the water density $\rho_w = \rho_v + \rho_c + \rho_r$, [20].

2.2. Implicit condensation

For the numerical method, we want all the source terms to be given explicitly. To avoid the modelling of the condensation source term, we follow [13] and remove the condensation source term by adding equations (1b) and (1c). This defines a new *moist* density $\rho_m = \rho_v + \rho_c$ of water which is only transported by the velocity of dry air. This leads to the equation

$$\partial_t \rho_m + \nabla \cdot (\rho_m \mathbf{u}) = S_{ev} - S_{au} - S_{ac}. \quad (8)$$

To recover ρ_c, ρ_v , we then need to compute the vapour density at saturation. This can be obtained from the saturation pressure (7) and the equation of state for the vapour pressure $p_v = \rho_v R_v T$. Since this depends on the temperature, we need to recover the temperature from the internal energy contribution in the energy equation (4), which in turn depends on all densities separately. We assume that all cloud water evaporates instantaneously in undersaturated regions. Using the moist density, the internal energy density $\rho e = E - \rho \mathbf{u} \cdot \mathbf{u} / 2$, where e denotes the internal energy, and the saturation requirement, we then arrive at the non-linear system

$$\rho e = (c_{vd} \rho_d + c_{vv} \rho_v + c_l (\rho_c + \rho_r)) (T - T_{\text{ref}}) + \rho_v (L_{\text{ref}} - R_v T_{\text{ref}}), \quad (9a)$$

$$\rho_v = \min \left(\frac{e_s(T)}{R_v T}, \rho_m \right), \quad (9b)$$

$$\rho_c = \rho_m - \rho_v, \quad (9c)$$

which we can use to solve for the temperature, vapour density and cloud density. The saturation vapour pressure $e_s(T)$ is defined in (7). We note that while [13] also considered a single moist density, rain dynamics were not included. For a detailed discussion of our solution strategy of system 9, see Section 3.4 below.

2.3. Perturbation formulation

To facilitate the numerical approximation of the system (1a), (8), (1d), (1e) and (1f), we split the densities, pressure and energy density into the hydrostatic part (time-independent and only dependent on the spatial z -direction) plus a perturbation from the hydrostatic state:

$$\rho_i(x, y, z, t) = \bar{\rho}_i(z) + \rho'_i(x, y, z, t), \quad i \in \{d, v, c, m, r\}, \quad (10a)$$

$$E(x, y, z, t) = \bar{E}(z) + E'(x, y, z, t), \quad (10b)$$

$$p(x, y, z, t) = \bar{p}(z) + p'(x, y, z, t), \quad (10c)$$

for which we have the relation

$$\nabla \bar{p} = -\bar{\rho} g e_z. \quad (11)$$

This then leads to the set of equations we use in our numerical method

$$\partial_t \rho'_d + \nabla \cdot ((\bar{\rho}_d + \rho'_d) \mathbf{u}) = 0, \quad (12a)$$

$$\partial_t \rho'_m + \nabla \cdot ((\bar{\rho}_m + \rho'_m) \mathbf{u}) = S_{ev} - S_{au} - S_{ac}, \quad (12b)$$

$$\partial_t \rho'_r + \nabla \cdot ((\bar{\rho}_r + \rho'_r) (\mathbf{u} - v_r e_z)) = S_{au} + S_{ac} - S_{ev}, \quad (12c)$$

$$\partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} - (\bar{\rho}_r + \rho'_r) v_r \mathbf{u} \otimes e_z + \text{Id } p') = -\rho' g e_z, \quad (12d)$$

$$\partial_t E' + \nabla \cdot ((\bar{E} + E' + \bar{p} + p') \mathbf{u} - (c_l (T - T_{\text{ref}}) + \mathbf{u}^2 / 2) (\bar{\rho}_r + \rho'_r) v_r e_z) = -(\bar{\rho} + \rho') g e_z \cdot \mathbf{u}, \quad (12e)$$

where the vapour, cloud densities and temperature (perturbations) are again recovered via system (9).

3. Discretisation

Our system of equations is a hyperbolic balance law, i.e., we may write it in short as

$$\partial_t U + \nabla \cdot F(U) = G(U), \tag{13}$$

with $U = (\rho'_d, \rho'_m, \rho'_r, \mathbf{m}, E') \in [L^2(\Omega)]^{4+d}$, $d \in \{2, 3\}$ the spatial dimension, and $\mathbf{m} = \rho \mathbf{u}$, which we aim to solve using a discontinuous Galerkin scheme.

3.1. Preliminaries and notation

For the discretisation in space, let \mathcal{T}_h be a quasi-uniform and shape-regular simplicial or hexahedral mesh of Ω . In the case of a simplicial mesh, we define our discrete space as

$$W_h = [\mathbb{P}^k(\mathcal{T}_h)]^{4+d} := \{V_h \in L^2(\Omega) : V_h|_K \in \mathbb{P}^k(K), \text{ for all } K \in \mathcal{T}_h\}^{4+d},$$

and in the case of a hexahedral mesh, we set

$$W_h = [\mathbb{Q}^k(\mathcal{T}_h)]^{4+d} := \{V_h \in L^2(\Omega) : V_h|_K \in \mathbb{Q}^k(K), \text{ for all } K \in \mathcal{T}_h\}^{4+d}.$$

The dimension $4 + d$ is due to the presence of four scalar variables in (12) plus the number of momentum variables given by d . The polynomial and tensor product spaces are

$$\mathbb{P}^k = \text{span} \left\{ \prod_{i=1}^d x_i^{\alpha_i} : \alpha_i \in \mathbb{N}_0, \sum_{i=1}^d \alpha_i \leq k \right\}, \quad \text{and} \quad \mathbb{Q}^k = \text{span} \left\{ \prod_{i=1}^d x_i^{\alpha_i} : \alpha_i \in \mathbb{N}_0, 0 \leq \alpha_i \leq k \right\},$$

for a given order $k \geq 1$. We note that in `NGSolve`, we use a basis of Legendre polynomials [22] leading to a diagonal mass-matrix. Furthermore, let $\mathcal{F}_h := \{F = \partial K_1 \cap \partial K_2 \mid \forall K_1, K_2 \in \mathcal{T}_h\}$ be the set of facets of the mesh. We divide this into the set of boundary facets $\mathcal{F}_h^b := \{F \in \mathcal{F}_h \mid F \subset \partial \Omega\}$ and interior facets $\mathcal{F}_h^i := \mathcal{F}_h \setminus \mathcal{F}_h^b$. On a facet $F \in \mathcal{F}_h$, we define a fixed unit normal vector as

$$\mathbf{n} = \begin{cases} \mathbf{n}_{K_1} & \text{the unit normal vector to } F \text{ at } x \text{ pointing from } K_1 \text{ to } K_2 \text{ if } F \in \mathcal{F}_h^i \text{ with } F = \partial K_1 \cap \partial K_2; \\ & \text{the orientation is arbitrary but fixed in what follows.} \\ \mathbf{n} & \text{the unit outward pointing normal to } \Omega \text{ at } x \text{ if } x \in \mathcal{F}_h^b. \end{cases}$$

With the fixed unit normal vector and K_1, K_2 numbered accordingly, we define

$$U^- = \begin{cases} U|_{K_1} & \text{if } F \in \mathcal{F}_h^i, \\ U|_K & \text{if } F \in \mathcal{F}_h^b, \end{cases} \quad U^+ = \begin{cases} U|_{K_2} & \text{if } F \in \mathcal{F}_h^i, \\ g & \text{if } F \in \mathcal{F}_h^b, \end{cases}$$

for boundary data g to be defined for a given example. We then define the jump and average operators on a facet F as

$$[[U]] = U^- - U^+, \quad \text{and} \quad \{\{U\}\} = \frac{1}{2} (U^- + U^+).$$

3.2. Spatial semi-discretisation

The derivation of a DG formulation for (13) can be found in many textbooks, e.g., [23–26]. The idea is to multiply (13) with an arbitrary test function $V_h \in W_h$, integrate over Ω , divide the integral into element contributions, integrate by parts and choose a numerical flux on element boundaries (since U_h and V_h are discontinuous). This gives the spatially semi-discrete form on every element $K \in \mathcal{T}_h$

$$(\partial_t U_h, V_h)_K - (F(U_h), \nabla V_h)_K + (F_n(U_h), V_h)_{\partial K} = (G(U_h), V_h)_K, \tag{14}$$

where $(\cdot, \cdot)_\omega$ denotes the L^2 -inner-product on $\omega \subset \Omega$. For the numerical flux, we choose the Lax-Friedrich flux, which is given by

$$F_n(U_h) = \{\{F(U_h)\}\} \mathbf{n} + \frac{\Lambda}{2} [[U_h]],$$

where \mathbf{n} is the outward-pointing unit normal vector on a given element boundary and Λ corresponds to the largest Eigenvalue of $F'(U)$. We take

$$\Lambda = \max\{|U_h^+ \cdot \mathbf{n}| + |v_r e_z \cdot \mathbf{n}| + c_m, |U_h^- \cdot \mathbf{n}| + |v_r e_z \cdot \mathbf{n}| + c_m\}, \tag{15}$$

where c_m is the speed of sound in moist air

$$c_m = \sqrt{\frac{\gamma_m p}{\rho}}, \quad \text{with} \quad \gamma_m = \frac{q_d c_{vd} + q_v c_{vv} + (q_c + q_r) c_l + q_d R_d + q_v R_v}{q_d c_{vd} + q_v c_{vv} + (q_c + q_r) c_l}.$$

The ratio γ_m is the isentropic expansion factor for moist air [13], extended to include the specific heat and mass ratio for liquid water. In the case of $q_v = q_c = q_r = 0$, this reduces to the isentropic expansion factor for dry air, and (15) reduces to the Lax-Friedrichs Flux for the dry Euler equations. Consequently, this can be viewed as the natural extension to the case with moisture and rain. Note that (15) might not give the exact eigenvalue, however it is an upper bound for the fastest wave and therefore sufficient, since Λ only needs to be large enough, [23]. On domain boundaries, we will either consider periodic or slip-wall (reflective) boundary conditions. Solid walls are characterised by $\mathbf{u} \cdot \mathbf{n} = 0$. To implement this, we set $\mathbf{m}_h^+ = (\mathbf{m}^- - 2(\mathbf{m}_h^- \cdot \mathbf{n})\mathbf{n})$ on slip-wall boundaries and $(U_h^+)_i = (U_h^-)_i$ for the remaining components.

We then arrive at the spatially semi-discrete form by summing over all elements

$$(\partial_t U_h, V_h)_{\Omega} + F_h(U_h)(V_h) = G_h(U_h)(V_h), \quad (16)$$

with

$$F_h(U_h)(V_h) := \sum_{K \in \mathcal{T}_h} -(F(U_h), \nabla V_h)_K + \sum_{F \in \mathcal{F}_h} (F_n(U_h), \llbracket V_h \rrbracket)_F,$$

$$G_h(U_h)(V_h) := \sum_{K \in \mathcal{T}_h} (G(U_h), V_h)_K,$$

where we define the jump $\llbracket V_h \rrbracket = V_h$ on domain boundary facets $F \in \mathcal{F}_h^b$.

3.3. Time-integration

Let $[0, T_{\text{end}}]$ be the time-interval of interest and consider the constant time-step $\Delta t = T_{\text{end}}/N$ for some given $N \in \mathbb{N}$. We will use an explicit time-stepping scheme to advance the equations in time, since we know this results in a very efficient scheme, in the sense that it is highly parallelizable, thus able to utilize HPC hardware. As usual with explicit schemes, the time-step needs to be chosen sufficiently small, dependant on h and k , to obtain stability. We note that IMEX schemes [27,28] are able to improve on the constant in the time-step restriction, by treating the terms leading to the fastest waves implicitly.

To this end, let $\{\varphi_i\}_{i=1}^{N_W}$ be the basis of W_h and \mathbf{u} be the coefficient vector of U_h in this basis. Furthermore, let $\mathbf{M}, \mathbf{F}, \mathbf{G} : \mathbb{R}^{N_W} \rightarrow \mathbb{R}^{N_W}$ denote the mass matrix flux and forcing operators, i.e.,

$$\mathbf{M}_{ij} = (\varphi_i, \varphi_j)_{\Omega}, \quad \mathbf{F}(\mathbf{u})_i = F_h(U_h)(\varphi_i), \quad \text{and} \quad \mathbf{G}(\mathbf{u})_i = G_h(U_h)(\varphi_i). \quad (17)$$

Then the explicit Euler scheme for (16) reads as: For all $n = 1, \dots, N$, find \mathbf{u} such that

$$\mathbf{M}\mathbf{u}^n = \mathbf{M}\mathbf{u}^{n-1} - \Delta t \mathbf{F}(\mathbf{u}^{n-1}) + \Delta t \mathbf{G}(\mathbf{u}^{n-1}). \quad (18)$$

Note that solving the above system can be implemented very efficiently since (beside assembling several source vectors) it only involves inverting the mass-matrix. As there is no coupling between different elements, the latter is block diagonal (or even diagonal with the correct choice of basis) and thus solving the system can be implemented in a parallel manner and/or even matrix-free to better utilise modern high-performance computing architectures. This is also illustrated in Section 4.1.2 below.

Remark 1. In practice, we will use the four stage, third order strong stability preserving Runge-Kutta scheme (SPPRK(4,3)), [29,30]. To ease the readability, we use the explicit Euler scheme in the presentation of the method.

3.4. Dependant variable reconstruction

To solve system (18), we need to evaluate the discrete flux and source terms F_h and G_h . Unfortunately, they depend on the variables ρ_v, ρ_c, T which are not present in the (conservative) variables U_h , but derived from the non-linear relationships (9).

After every time-step, we construct $\rho_{v,h}^n, \rho_{c,h}^n, T_h^n$ from U_h^n by solving system (9) in every quadrature point¹ by Newtons method, taking the state at the last time-step as the initial guess. The resulting function is then only well defined in the quadrature points. The polynomial solution in $\mathbb{P}^k(\mathcal{T}_h)$ is then obtained via an L^2 -projection. As a result, we implicitly enforce the saturation requirement which allows for both condensation of water vapour and (instantaneous) evaporation of cloud water in undersaturated regions. With these secondary, dependent variables reconstructed, we can then assemble F_h and G_h for the next time-step or stage of the Runge-Kutta scheme. Our approach is similar to that of the one-step-scheme presented in [13], where system (9) was solved globally by an iterative and decoupled Newton scheme. In contrast, we solve (9) pointwise and fully coupled using a standard Newton approach, and then reconstruct the piecewise polynomial solution to evaluate the numerical flux. This has the advantage of converging quadratically, and being easily computable in parallel.

¹ NGSolve uses a product of Gauss-Legendre rules, which in the case of simplicial elements are mapped onto the simplex using a Duffy transformation.

3.5. Stabilisation through artificial diffusion

Phase changes due to rain dynamics can lead to spurious oscillations which eventually could have an effect on the stability of our method. To dampen these oscillations and assure stability, we apply a proper scaled artificial diffusion term to our system. In the discontinuous Galerkin setting, this is done through the symmetric interior penalty bilinear form

$$A_h(U_h, V_h) = \sum_{K \in \mathcal{T}_h} (\alpha \nabla U_h, \nabla V_h)_K + \sum_{F \in \mathcal{F}_h} \left[-(\alpha \{\{\nabla U_h\}\} n, \llbracket V_h \rrbracket)_F - (\alpha \{\{\nabla V_h\}\} n, \llbracket U_h \rrbracket)_F + \sigma h^2/k (\alpha \llbracket U_h \rrbracket, \llbracket V_h \rrbracket)_F \right], \quad (19)$$

with the penalty parameter $\sigma > 0$ chosen sufficiently large and a vector-valued diffusion coefficient $\alpha > 0$.

Convection dominated convection-diffusion problems can become unstable in the case of a mesh Péclet number (Pe) larger than one. The mesh Péclet number (for \mathbf{u} -transported quantities) is given by

$$Pe = \frac{\|\mathbf{u}\|_{\infty, K} h}{2\alpha}.$$

For pure convection problems (i.e. $\alpha = 0$), this can be seen as infinitely large. Motivated by this, we choose α in (19) element-wise and component-wise for the components of U_h as

$$\alpha_K = \gamma 0.5 h^{1-d/2} \|\mathbf{u}\|_{0, K} \quad \text{and} \quad \alpha_K = \gamma 0.5 h^{1-d/2} \|\mathbf{u} - v_r e_z\|_{0, K},$$

where the latter is only chosen for the rain density ρ_r , since this is the velocity-field by which it is transported. Note that we use the element wise L^2 -norm, omit the scaling by the mesh size and include a scaling parameter γ . This is because we cannot easily compute $\|\mathbf{u}\|_{\infty, K}$ and because $\|\mathbf{u}\|_{0, K} \simeq c h^{d/2} \|\mathbf{u}\|_{\infty, K}$, with a constant $c > 0$ independent of h . For our numerical experiments, we choose γ as small as possible, without getting an unstable method, and project the piecewise constant α_K into a piecewise-linear, continuous function, such that α_K is well defined on every facet. This is different to [9,15], where the added artificial viscosity is scaled with a single constant coefficient.

Finally, we note that for the case of the moist equations without rain, i.e., without any explicit phase changes, no artificial diffusion was found to be necessary for stability of the method. We therefore only add diffusion for the examples with rain.

Remark 2 (*Loss of higher-order convergence*). By adding the artificial diffusion term (19) to our system, we are solving the original problem including a perturbation of order $\mathcal{O}(h)$. As a result, we cannot expect high order convergence of the DG method, even if higher-order elements are used. For a fixed artificial viscosity (independent of h), we expect convergence to the exact solution of the perturbed system with the given fixed viscosity, but not towards the solution of the Euler equations (no viscosity) under consideration here.

3.6. Explicit sponge layer

In some problems, the top of the domain should represent an open domain. That is, waves should exit the domain and not reflect off the top boundary. Consequently, a slip-boundary condition, representing a solid wall, is unphysical. However, mathematical modelling of a non-reflective boundary is a non-trivial task. In order to prevent waves from reflecting off the top solid wall boundary, we implement a Rayleigh sponge layer to damp gravity waves, as is common in the atmospheric flow literature [9,10,15,31,32]. The idea is essentially to relax the solution towards a known far-field condition (usually the hydrostatic background state).

In our code, we relax the moist mass-fraction perturbations, the velocity and energy perturbation towards the known velocity and zero in the case of scalar perturbations. In all the cases below, the velocity is also zero, so the damping of the mass-fraction perturbations, velocity and energy perturbation is equivalent to damping the density perturbations, momentum and energy density perturbation. That is, at the end of each time step, we compute

$$U_h^n = (1 - \delta_R(z)) \tilde{U}_h^{n, \text{pred}} + \delta_R(z) \tilde{U}_h^{\text{far}}, \quad (20)$$

where $\tilde{U}_h^{n, \text{pred}} = (\rho'_m, \rho'_r, \rho \mathbf{u}, E')$ is the solution resulting from the time-stepping scheme, and $\tilde{U}_h^{\text{far}} = (0, 0, \rho \bar{\mathbf{u}}, 0)$ is the far-field state for these variables. This is in contrast to [10,31], where the damping is applied to the entire state vector (velocity and scalars). Since the mass-fraction damping can lead to mass loss in the sponge layer, we add a correction step and locally add the moist mass lost through damping to the dry density. Consequently, the total mass is still preserved globally by our scheme. We note that mass fixing in numerical weather forecasting is done in the literature [33,34] with both local corrections and by spreading the mass correction in the entire domain.

In (20), we use the specific blending function

$$\delta_R(z) = \begin{cases} \frac{\alpha}{2} \left(1 - \cos \left(\pi \frac{z - z_D}{z_T - z_D} \right) \right) & \text{for } z \geq z_D, \\ 0 & \text{for } z < z_D. \end{cases}$$

Here z_D is the height of the bottom of the sponge layer, z_T is the height of the top boundary and $0 < \alpha \leq 1$ is a parameter to tune the intensity of the sponge layer.

We implement (20) through an L^2 -projection. The sponge layer can also be implemented by adding a corresponding L^2 -term to the right-hand side, however, the choice of α is more difficult then.

4. Numerical examples

All numerical examples are implemented using the finite element library Netgen/NGSolve [35,36] and www.ngsolve.org. We run all examples either in a distributed memory (MPI) parallel or shared memory parallel fashion (similar to openMP). The python scripts implementing our specific examples and the results presented here are freely available on github and archived on zenodo [37].

We consider the following computation examples:

Example 1: Inertia gravity waves in a saturated atmosphere without rain dynamics to test our method with respect to optimal, high-order convergence, and parallel scalability.

Example 2: The widely used benchmark by Bryan and Fritsch 2002 of a rising thermal in a saturated atmosphere, again without rain dynamics. We investigate the method with respect to large gradients due to the rising thermal.

Example 3: We revisit the inertia gravity wave problem, but with an initial condition of a saturated vapour density but no clouds, in order to investigate cloud formation in our method.

Example 4: Conservation of a hydrostatic state in a domain with a large steep mountain. Unstructured meshes are of particular interest in this example due the steep terrain under consideration. This example also includes the possibility of rain forming, if the solution deviates from the hydrostatic state.

Example 5: A two-dimensional rising thermal leading to cloud formation and rain dynamics as formulated by Grabowski and Clark 1991.

Example 6: The three-dimensional extension of Example 5 as described in [40].

Example 7: Two dimensional squall line storm simulation based on [15,9].

4.1. Example 1: inertia gravity waves in a saturated atmosphere

As our first example, we consider the case of a moist version of the non-hydrostatics gravity waves adapted from [41] and presented in [11]. This example results in a smooth solution, such that we can use it to test high-order convergence of our method.

4.1.1. Set-up

The domain is $\Omega = (0, 300 \text{ km}) \times (0, 10 \text{ km})$. Periodic boundary conditions are applied on the left and right boundaries, while we have solid walls (slip) boundary conditions on the bottom and top boundaries. The time interval under consideration is $[0, 3600 \text{ s}]$. The hydrostatic base state is defined via the wet equivalent potential temperature

$$\bar{\theta}_e = \Theta_0 \exp(N^2 z/g), \quad (21)$$

with $\Theta_0 = 300 \text{ K}$ and $N^2 = 10^{-1} \text{ s}^{-2}$. The total water fraction is $q_w = 0.02$ and the pressure boundary condition is $p(0) = p_{\text{ref}}$. The perturbation applied to the hydrostatic base state is

$$\theta'_e = \frac{\Delta\Theta}{1 + a^{-2}(x - L/2)^2} \sin\left(\frac{\pi z}{H}\right),$$

where H is the domain height, L the domain length, $a = 5 \times 10^3 \text{ m}$ and $\Delta\Theta = 0.01 \text{ K}$. The perturbation is applied under the requirements that q_w and the pressure remain unchanged, and the air is saturated everywhere. The initial velocity is prescribed as $u = (20, 0)^T \text{ m s}^{-1}$. Additional details of the computation of the hydrostatic base state and initial perturbation are provided in Appendix B.1.

4.1.2. Results

We consider this problem on a structured quadrilateral mesh, starting with $h = 1000 \text{ m}$ and a series of three mesh refinements. On each of these meshes, we consider the polynomial order $k = 1, 2, 3, 4$. The time-step is chosen to be within the time-step restriction of the explicit time-stepping scheme, that is $\Delta t = 1 \text{ s}$ for $h = 1000 \text{ m}$ with $k = 1$, and $\Delta t = 0.025 \text{ s}$ for $h = 125 \text{ m}$ with $k = 4$. To compute convergence rates, we consider the results on the finest mesh ($h = 125 \text{ m}$) with order $k = 4$ as the reference solution. We then use the $\mathcal{L}^2(L^2)$ -type (space-time) norm

$$\|x_h - x_{h,\text{ref}}\|^2 = \sum_{i=1}^m \|x_h(t_i) - x_{h,\text{ref}}(t_i)\|_{L^2(\Omega)}^2,$$

with a total of $m = 30$ equidistant points in time. The convergence results are shown in Fig. 1. We observe the expected optimal order of convergence of $k + 1$. There is some less than optimal convergence between the finest two meshes considered with $k = 4$. We attribute this to the numerical reference solution not being sufficiently accurate.

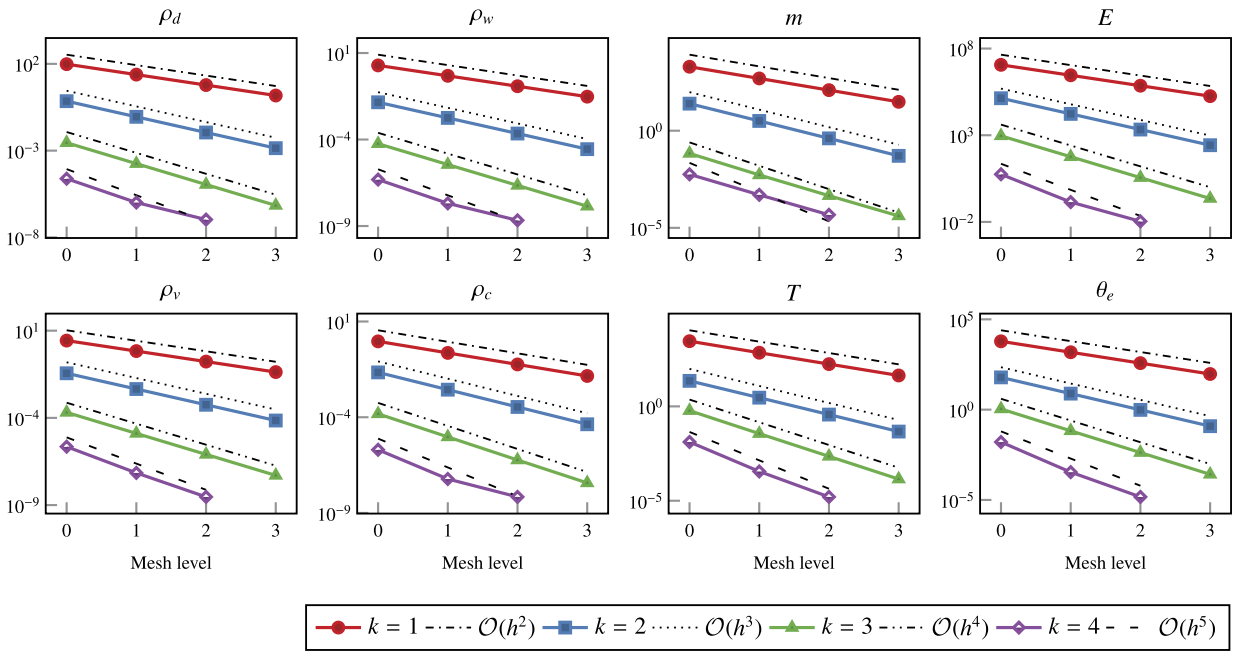


Fig. 1. Example 1: Convergence results of the primary and secondary variables, and wet equivalent potential temperature on a series of structured quadrilateral meshes for different polynomial orders.

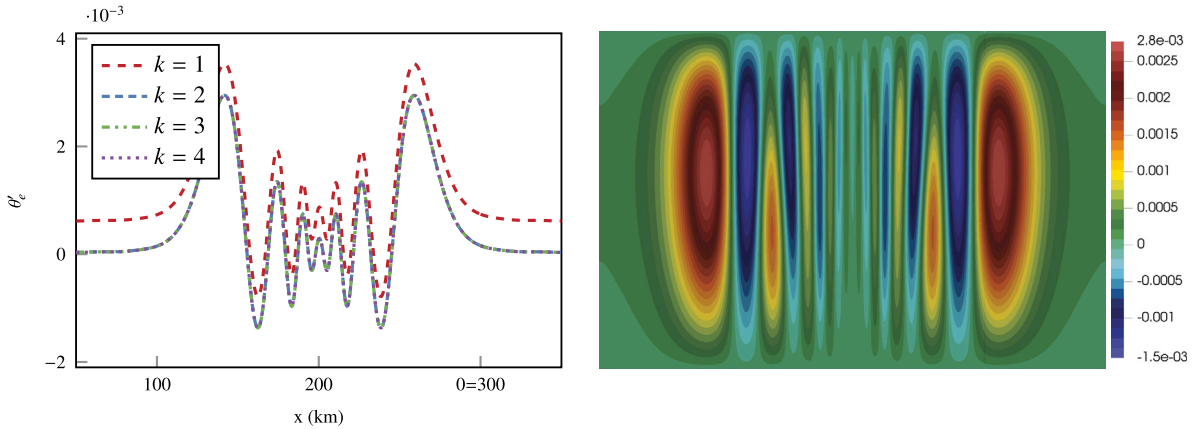


Fig. 2. Example 1: Inertia gravity waves in a saturated atmosphere. Perturbation of the wet equivalent potential temperature from the hydrostatic base state. Left: Profile along the line $z = 5$ km at $t = 2520$ s for different polynomial orders on a structured quadrilateral mesh with $h = 125$ m, centred around $x = 200$ km. Right: Solution at $t = 3600$ s computed using Q^3 elements with $h = 500$ m, centred around $x = 222$ km.

The perturbation of the wet equivalent potential temperature along the line $z = 5$ km at $t = 2520$ s can be seen on the left of Fig. 2. Here we see the expected symmetry around the $x \approx 200$ km, similar to [18], even though the latter only considered dry dynamics. Due to our initial perturbation being located at $x = 150$ km as in [11] rather than at $x = 100$ km in [18], the axis of symmetry is also shifted in our results. Furthermore, we chose the time $t = 2520$ s, rather than $t = 2500$ s as in [18] as not all our time-steps hit this point in time. We observe that the lowest order results with $k = 1$ are visibly different from the higher order results, while for $k = 2, 3, 4$, the results are indistinguishable. On the right of Fig. 2, we see the perturbation of the wet equivalent potential temperature at $t = 3600$ s in the bulk of the domain computed using Q^3 elements on the mesh with $h = 500$ m. This matches the results presented in [11] very well. However, we emphasize that this is not a variable in our system of equations, but a quantity obtained by post-processing.

Finally, we consider the parallel scalability of our method. As discussed in Section 3.3, every time step requires vector assemblies, and the only large systems that have to be solved are mass-matrix problems. Due to the choice of basis, the matrices are diagonal and solving the system is implemented matrix-free, which is known to be a scalable approach. Our scheme's second major computationally expensive part is the dependant variable reconstruction, discussed in Section 3.4. However, since this is done completely locally, we expect this to also scale well in parallel. A strong scaling plot for one hundred time steps is shown in Fig. 3 using either NGSolve's

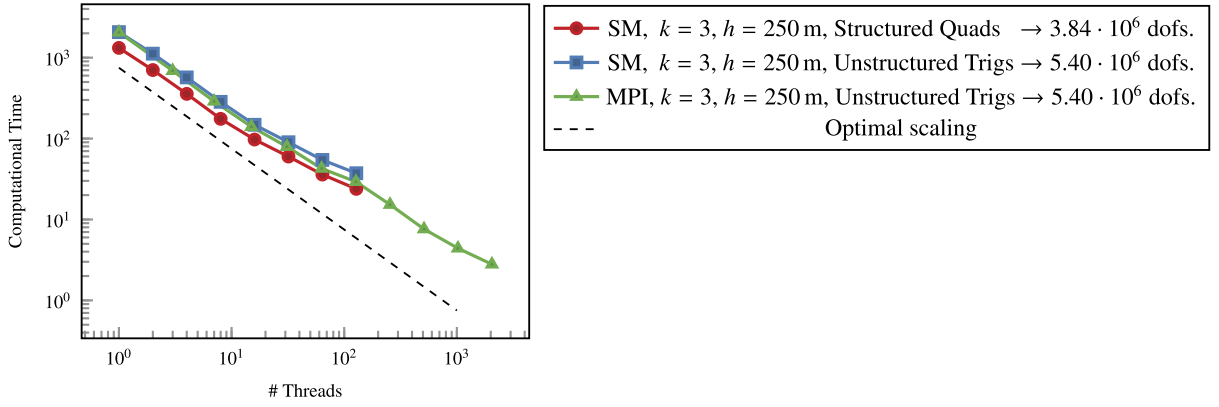


Fig. 3. Example 1: Strong scaling results for the time-stepping loop using NGSolve’s shared memory (SM) and distributed memory (MPI) parallel implementations. The number of degrees of freedom (dofs) are counted for the finite element space for the eight primary variables.

own shared memory parallel loops or an MPI-distributed memory parallel version. As we can see, both versions scale well. We have nearly perfect scaling up to 16 parallel threads and good scaling even up to 2048 MPI ranks across 16 nodes.

4.2. Example 2: Bryan-Fritsch moist benchmark

Having established the optimal order error convergence of our method, we consider a benchmark problem proposed in [38] as our second example. This problem includes water vapour and clouds but still no rain.

4.2.1. Set-up

The domain is given by $\Omega = (0, 20 \text{ km}) \times (0, 10 \text{ km})$. Periodic boundary conditions are applied on the left and right boundaries and rigid wall (slip) boundary conditions are applied on the top and bottom boundaries, i.e., $\mathbf{u} \cdot \mathbf{n} = 0$. The time interval under consideration is $[0, 1000 \text{ s}]$. The hydrostatic base state is defined by assuming a saturated atmosphere, $q_w = 0.02$, a constant wet equivalent potential temperature of $\bar{\theta}_e = 320 \text{ K}$ and the boundary condition for the pressure of $\bar{p} = p_{\text{ref}} = 10^5 \text{ Pa}$. The initial perturbation is computed by perturbing the density potential temperature in a circular region, such that there is positive buoyancy in this region while the pressure remains unchanged. Based on this, we compute the temperature perturbation, from which we then reconstruct the vapour and cloud densities using the saturation vapour pressure. This leads to the initial condition $\rho'_d = 0$, $\rho'_m = 0$, $\rho u = (0, 0)$, E' , ρ'_v , ρ'_c and T' . Details on the computation of the hydrostatic state and perturbation are given in Appendix B.2

4.2.2. Results

The quantity of interest in this benchmark is the perturbation of the density potential temperature θ'_ρ , the definition of which is given below in (B.2). The resulting perturbation θ'_ρ over a series of meshes and polynomial orders can be seen in Fig. 4. The time-step is chosen to be within the time-step restriction of the explicit time-stepping scheme. The exact values are given in Fig. 4. We emphasize that the density potential temperature quantity is not a variable we solve for in our system, but a quantity which has to be post-processed from the available data.

The final states are comparable to that presented in [38]. There are some oscillations inside the temperature bubble visible throughout, although we note that they become smaller under mesh refinement and increasing polynomial order. Unsurprisingly, the symmetry of the solution is maintained on the structured quadrilateral mesh, and there are some instabilities visible at the boundary of the temperature bubble for higher-order elements. However, these also appear to become smaller under mesh refinement and increasing polynomial order.

Finally, we note that for all polynomial orders considered, the temperature bubble is consistent. This contrasts with the results in [11], where additional plumes appeared for their higher-order case $k = 1$. The authors of the latter paper attributed this to a physical instability which is damped in the lowest order case $k = 0$, due to higher numerical diffusing.

4.3. Example 3: inertia gravity waves in a saturated atmosphere without initial clouds

In the previous examples, the vapour density is saturated throughout. To investigate cloud formation, we consider the inertia gravity waves set-up, but define the initial condition such that vapour density is fully saturated, but without any cloud water present.

4.3.1. Set-up

The spatial and temporal set-up is identical to that in Section 4.1.1. The hydrostatic base state is computed using the same $\bar{\theta}_e$ given in (21), but now with $\bar{p}_v = \bar{p}_{vs}$ and $\bar{p}_c = 0$. The same perturbation is added to the wet equivalent temperature and the initial state is again given by assuming that the pressure is not changed by the perturbation and the presence of a saturated atmosphere without clouds. Additional details are provided in Appendix B.3.

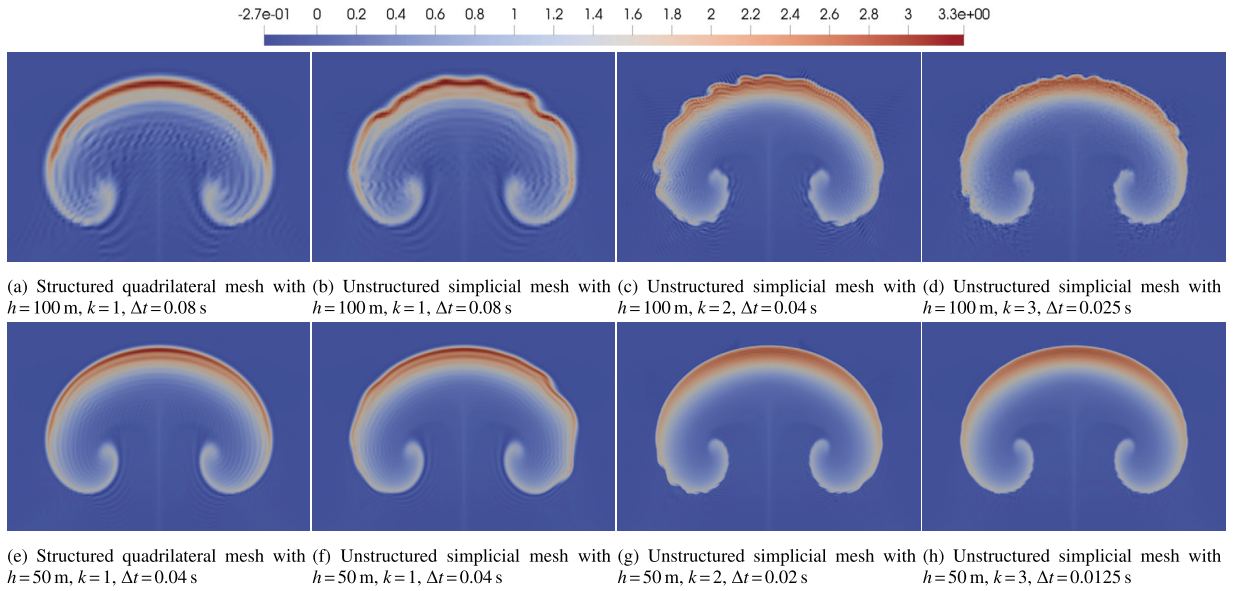


Fig. 4. Example 2: Density potential temperature perturbation at $t = 1000$ s in the region $(6 \text{ km}, 14 \text{ km}) \times (3 \text{ km}, 8.5 \text{ km})$ on a series of different triangular and quadrilateral meshes using different order finite elements.

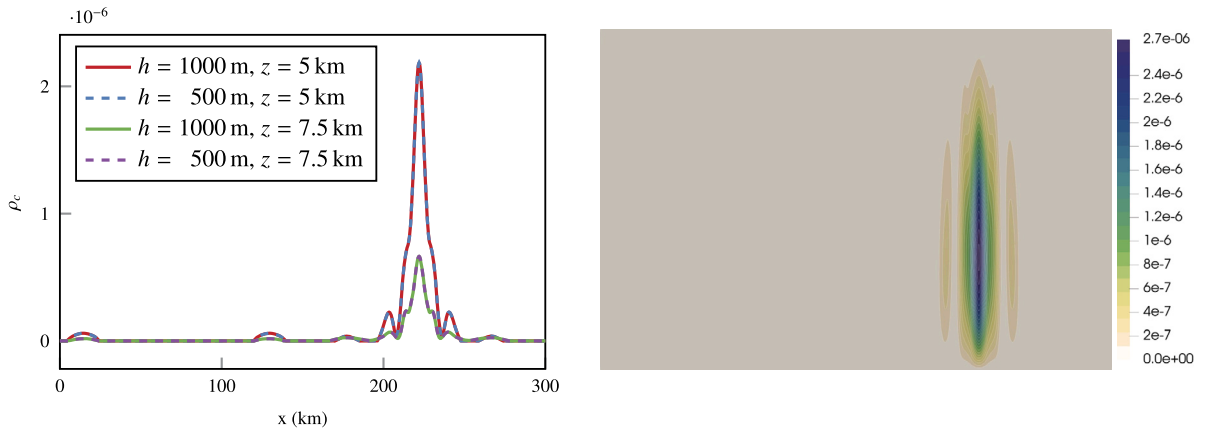


Fig. 5. Example 3: Cloud density (kg/m^3) and contours at $t = 3600$ s resulting from inertia gravity waves in a saturated atmosphere without initial clouds. Computed using \mathbb{Q}^3 elements on a structured quadrilateral mesh. Left: Profiles along the lines $z = 5 \text{ km}$ and 7.5 km . Right: Density in the whole domain, computed with $h = 500 \text{ m}$, $\Delta t = 0.15 \text{ s}$.

4.3.2. Results

We consider elements of order three on the two coarsest meshes used in the previous example, since the errors in Section 4.1 were already very small for this choice. The mesh sizes and time steps are $(h, \Delta t) = (1000 \text{ m}, 0.3 \text{ s})$ and $(h, \Delta t) = (500 \text{ m}, 0.15 \text{ s})$, resulting in 2.4×10^5 and 9.6×10^5 degrees of freedom for the finite element space of the primal variables, respectively.² Profiles of the cloud density along two horizontal lines, and the density in the entire volume are shown in Fig. 5 for $t = 3600$ s. Looking at the results in Fig. 5, we see that the cloud formation is consistent between the two meshes, and in fact there is no visible difference between the resulting cloud profiles.

4.4. Example 4: atmosphere at rest with a steep mountain

Approaches based on coordinate following discretisations can struggle in cases of severe slopes, even with the atmosphere at rest [10,42]. As an example where unstructured meshes are particularly attractive, we consider a domain with a single Gaussian mountain, based on the dry example described in [42], but consider the full moist system (12).

² The compute wall times on a single node with two AMD EPYC 7713 64-Core Processors with hyper-threading enabled using 256 shared memory parallel threads was 279 s ($h = 1000, \Delta t = 0.3$) and 1390 s ($h = 500, \Delta t = 0.15$), respectively.

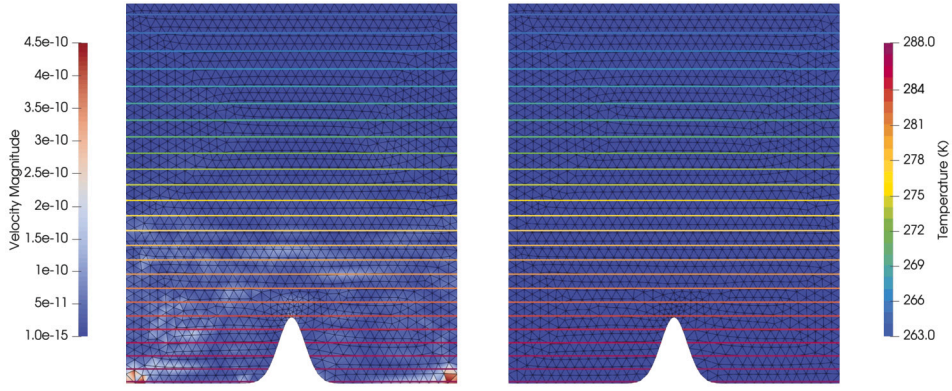


Fig. 6. Example 4: Results for the atmosphere-at-rest experiment with a steep mountain after 6 h on unstructured triangular meshes. Solid lines indicate temperature contour lines (1 K intervals) and the shading indicates spurious velocity. Left: $h = 1000$ m, $k = 1$, $\Delta t = 0.4$ s, Right: $h = 1000$ m, $k = 2$, $\Delta t = 0.2$ s.

4.4.1. Set-up

The domain has a height of 40 km and a width of 35 km. The terrain profile is given by

$$z_m(x) = h_0 \exp\left(-\frac{(x - x_c)^2}{a^2}\right),$$

where x_c is the centre of the domain and we consider the height $h = 7000$ m and half-width $a = 2000$ m, as in [42]. This results in a maximal slope of about 3. On the vertical boundaries, we consider periodic boundary conditions and solid wall boundaries at the top and bottom.

For this, we need to include the sponge layer discussed in Section 3.6. Specifically, we choose $z_D = 15$ km and $\alpha = 0.1$.

We define the hydrostatic base state through the temperature profile

$$\bar{T}(z) = T_{\text{str}} + (T_{\text{sl}} - T_{\text{str}}) \exp\left(-\frac{z}{H_{\text{scal}}}\right), \quad (22)$$

where $T_{\text{sl}} = 288.15$ K, $T_{\text{str}} = 213.15$ K and $H_{\text{scal}} = 10000$ m. The water densities are then defined by requiring vapour saturation throughout and setting the cloud and rain densities to zero. Further details are provided in Appendix B.4 This setting is challenging, since deviations from the hydrostatic state lead to dynamics in all variables. In particular, deviations will cause cloud and rain formation, c.f., Section 2.1.

4.4.2. Results

We consider an unstructured triangular mesh, with linear elements and mesh size $h = 1000$ m, of the domain together with elements of order $k = 1, 2$. The time step is chosen according to the stability limit, resulting in $\Delta t = 0.4$ s and 0.2 s, respectively. The resulting spurious velocity, temperature contours and the mesh can be seen in Fig. 6. Here we see that the solution has remained stable, with the largest spurious velocities visible at the coarsest elements and the bottom boundary. Furthermore, there are no oscillations visible in the temperature contours, even close to the mountain profile. In fact, it appears that for the case $k = 2$, no spurious velocities are present. A closer inspection of the data shows that the velocity is non-zero up to approximately 10^{-12} , which is not visible on the colour scale. Furthermore, we note that while the simulation allowed for the formation of rain, which occurs as soon as the cloud density is non-zero, see (5), the rain density remained stable with a perturbation of 7×10^{-13} and 3×10^{-15} for $k = 1$ and $k = 2$, respectively. Finally, while our sponge layer allows for mass exchange between the moist and dry densities in the sponge layer, the dry density perturbation observed is of order 10^{-11} and 10^{-13} for $k = 1$ and $k = 2$, respectively, and with the largest perturbations visible in the area below the sponge layer.

4.5. Example 5: rising thermal with rain in two spatial dimensions

This test problem is based on the one described in [39] and again involves a rising thermal in two dimensions, but this time in an undersaturated atmosphere and allowing for the development of rain.

4.5.1. Set-up

The domain is $\Omega = (0, 3.6 \text{ km}) \times (0, 2.4 \text{ km})$, periodic boundary conditions are set on the left and right boundaries and solid wall (slip) boundary conditions at the top and bottom boundaries. Note that while consequently dry air, water vapour and cloud water cannot be transported out of the domain, rain water can leave the domain since $v_r \cdot \mathbf{n}_z$ is not necessarily zero at the bottom boundary. The time interval under consideration is $[0, 600 \text{ s}]$.

The hydrostatic base state is defined by specifying a relative humidity and the dry potential temperature with

$$\bar{H} = 0.2, \quad \text{and} \quad \bar{\theta}_d = \Theta e^{S z},$$

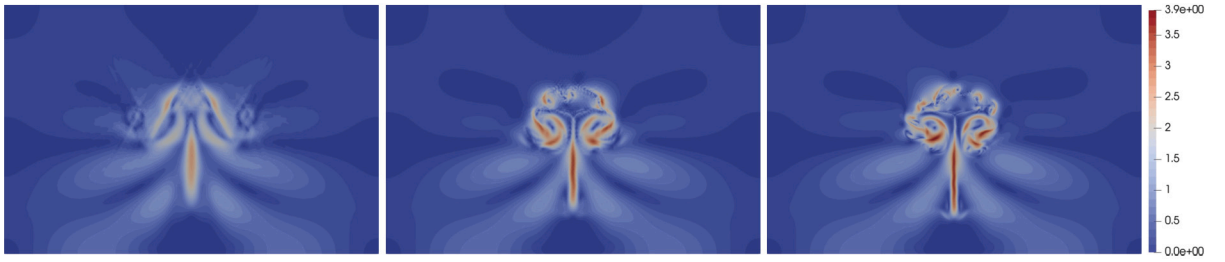


Fig. 7. Example 5: Dry air velocity (m s^{-1}) solution at $t = 600$ s for a two-dimensional rising thermal leading to precipitation. Discretisation parameters from left to right: $(h, k, \Delta t) = (25 \text{ m}, 1, 0.02 \text{ s}), (25 \text{ m}, 2, 0.01 \text{ s}), (12.5 \text{ m}, 2, 0.005 \text{ s})$. Computed on unstructured simplicial meshes.

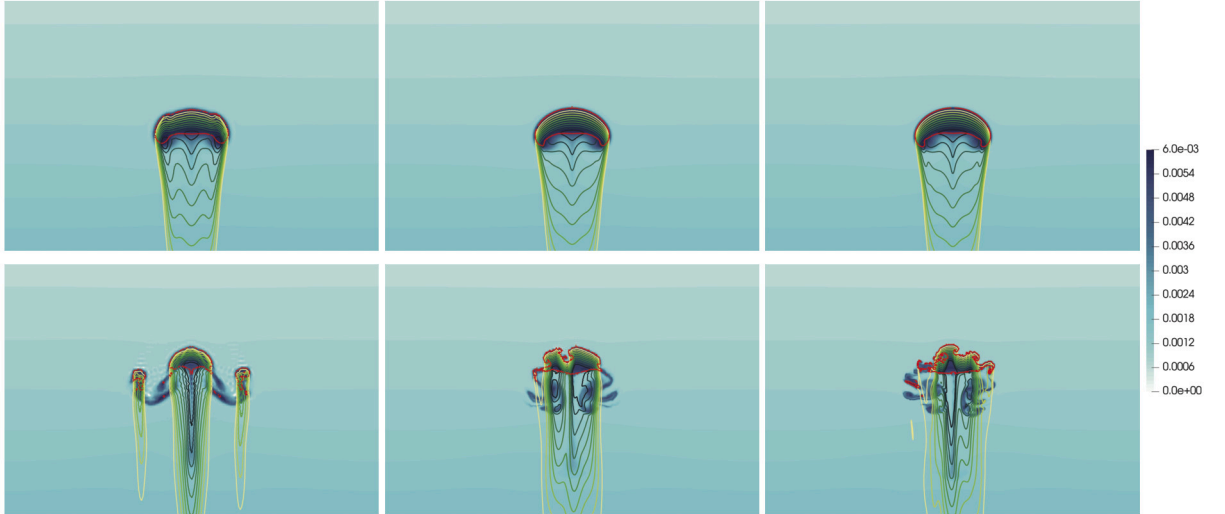


Fig. 8. Example 5: Water vapour density (kg/m^3) solution (background), cloud density (kg/m^3) edge contour at 0.001 (red) and rain water density (kg/m^3) contours at $\{10^{-6}, 2 \times 10^{-6}, 3 \times 10^{-6}, 4 \times 10^{-6}, 5 \times 10^{-6}, 6 \times 10^{-6}, 7 \times 10^{-6}, 8 \times 10^{-6}, 9 \times 10^{-6}, 10^{-5}\}$ (greens), for a two-dimensional rising thermal leading to precipitation. Top row: $t = 300$ s, bottom row: $t = 600$ s. Discretisation parameters as in Fig. 7.

where Θ is the dry potential temperature corresponding to $T_{\text{surf}} = 283 \text{ K}$ and $p = 8.5 \times 10^4 \text{ Pa}$. This pressure is also used for the pressure boundary condition at $z = 0$. The stratification is given by $S = 1.3 \times 10^{-5} \text{ m}^{-1}$. There are no clouds and no rain. Details on the computation of the hydrostatic base state are given in Appendix B.5.

The initial perturbation of the initial condition is then given by a circular bubble, where the air is saturated but no clouds are present, without changing the pressure and dry potential temperature. Further details on the initial condition are again provided in Appendix B.5.

4.5.2. Results

We consider an unstructured simplicial mesh of the domain with $h = 25 \text{ m}$ and orders $k = 1, 2$, as well as a second mesh with $h = 12.5 \text{ m}$ and $k = 2$. The time step is chosen as large as the time-step restriction allows, resulting in $\Delta t = 0.02 \text{ s}, 0.01 \text{ s}$ and 0.005 s , respectively. For stability, we include artificial diffusion here and choose the parameter to be $\gamma = 0.06$, which we found to be the lower limit to preserve stability in this case.

The resulting velocity field at $t = 600$ s can be seen in Fig. 7, and the water vapour density, cloud boundary contour and rain density contours at $t = 300$ s, 600 s in Fig. 8. We also show the total rain transported out of the bottom of the domain in Fig. 9.

Looking at the velocity solution, we see that the results are consistent, and higher order results in faster velocities and more details in the velocity field. Looking at the water vapour, cloud and rain density solutions at $t = 300$ s in Fig. 8, we have similar results for all three discretisations. In particular, the results for $k = 2$ are consistent with the literature [39]. At $t = 600$ s, the densities have more pronounced differences; notably, in the case $k = 1$, the cloud has split into three separate clouds, which is mirrored in the rain contours and in the more spread out total rainfall seen in Fig. 9. We note that the single rain column is consistent with results in the literature [43]. With regard to the water vapour, we see that the $k = 2$ results show significantly more small-scale features than the $k = 1$ solution. Finally, we note that we do not preserve the symmetry of the initial condition due to the use of unstructured meshes. This lack of symmetry can be observed consistently in Fig. 7, Fig. 8 and Fig. 9.

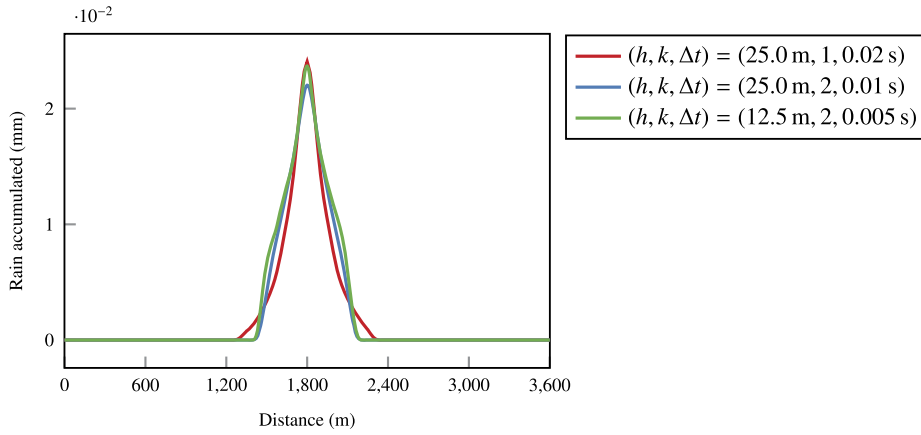


Fig. 9. Example 5: Total rain fallout at the bottom of the domain for a two-dimensional rising thermal leading to precipitation after $t = 600$ s.

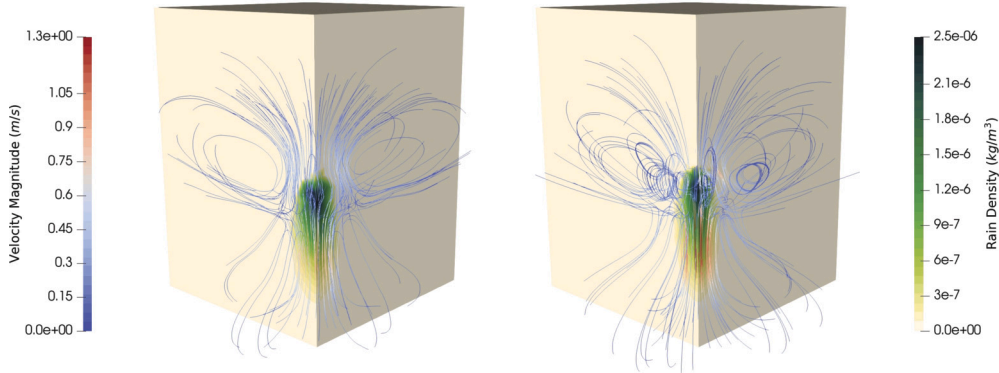


Fig. 10. Example 6: Velocity and rain water density results for a three-dimensional rising thermal leading to precipitation on an unstructured tetrahedral mesh at $t = 360$ s. Left: $h = 100$ m, $k = 1$, $\Delta t = 0.04$, Right: $h = 100$ m, $k = 2$, $\Delta t = 0.02$.

4.6. Example 6: rising thermal with rain in three spatial dimensions

We take the extension of the previous example as presented in [40].

4.6.1. Set-up

The domain is $\Omega = (0, 3.6 \text{ km}) \times (0, 3.6 \text{ km}) \times (0, 2.4 \text{ km})$, periodic boundary conditions are set on the horizontal boundaries and solid wall (slip) boundary conditions at the top and bottom boundaries. The hydrostatic state and the initial conditions are defined analogously to those in Section 4.5, however, the bubble is now three-dimensional.

4.6.2. Results

We consider an unstructured tetrahedral mesh of the domain with $h = 100$ m. We consider the polynomial orders $k = 1, 2$, resulting in approximately 4×10^6 and 10^7 unknowns for the DG space for the primary variables. The time-step is chosen according to the time-step restriction, resulting in $\Delta t = 0.04$ s and $\Delta t = 0.02$ s, respectively. The artificial diffusion parameter is again chosen as small as possible, resulting in $\gamma = 0.6$.

The rain density for a quarter of the domain, together with velocity streamlines in the remaining domain, can be seen in Fig. 10, and a vertical slice through the centre of the domain ($y = 1.8 \text{ km}$) with the water vapour density, cloud outline contour and rain contours can be seen in Fig. 11. Here, we again see faster and more dynamics in the velocity solution for the case $k = 2$ and that the rain density is concentrated more towards the centre of the domain, consistent with our two-dimensional results. The higher velocity is also consistent with the observation that the rain falls from a higher height in the case $k = 2$. Furthermore, we observe that the higher order simulation preserves the symmetry of significantly better than the case for $k = 1$.

4.7. Example 7: squall line

As a final example, we consider an idealised test-case presented by [15]. The initial condition for this consists of a synthetic vertical profile, based on a typical environment typical for mid-latitude squall lines [15,44,45]. While we remain in the original

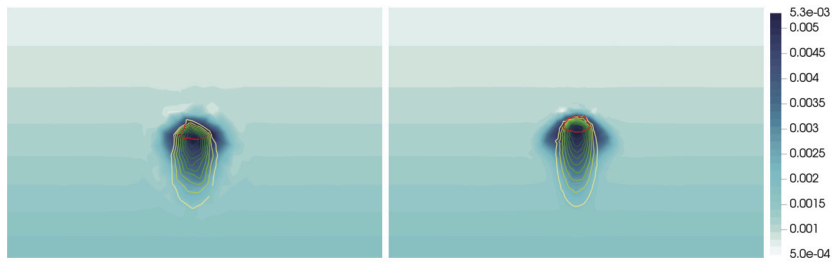


Fig. 11. Example 6: Water vapour density (background), cloud density contour at 0.0001 (red) and rain density contours at $\{2.4 \times 10^{-7}, 4.8 \times 10^{-7}, 7.2 \times 10^{-7}, 9.6 \times 10^{-7}, 1.2 \times 10^{-6}, 1.44 \times 10^{-6}, 1.68 \times 10^{-6}, 1.92 \times 10^{-6}, 2.16 \times 10^{-6}, 2.4 \times 10^{-6}\}$ (greens) for a three-dimensional rising thermal leading to precipitation on an unstructured tetrahedral mesh in the plane $y = 1.8$ km at $t = 360$ s.

two-dimensional setting from [15], we take the precise values from [9]. In particular, the set-up in [9] considers a smaller domain in the horizontal direction, and a shear wind in the opposite direction.

4.7.1. Set-up

The domain is $\Omega = (0, 150 \text{ km}) \times (0, 24 \text{ km})$. Periodic boundary conditions are applied on the lateral boundaries and a free-slip on the top and bottom boundary conditions. To avoid the reflection of waves from the non-physical top boundary condition, we apply an explicit sponge layer in the top 6 km of the domain. The time interval under consideration is $[0, 9000 \text{ s}]$.

The hydrostatic background state is computed based on a given vertical profile for the potential temperature θ and the water vapour mass fraction q_v . The specific values for this are taken from [9, Table A1]. Further details are given in Appendix B.6.

The initial condition is then given by a temperature perturbation bubble, which we apply to the temperature under the assumption that the pressure is unchanged as above. Details are again given in Appendix B.6. The densities remain unchanged. However, the velocity is initialized with a horizontal shear flow with $u_x = 12 \text{ m s}^{-1}$ at $z = 0$ and decreases linearly to zero at $z = 2.5$ km.

4.7.2. Results

We consider unstructured simplicial meshes of the domain with $h = 500, 250$, and 125 m, respectively. On these meshes, we consider $k = 1$ and time steps as large as the time-step restriction allows. We present the resulting perturbation of the potential temperature and velocity field, together with the cloud outline and rain contours at $t = 1500$ s and $t = 3000$ s in Fig. 12. We further show the storm's evolution in the same quantities resulting from our finest mesh at $t = 1500, 3000, 6000$, and 9000 s in Fig. 13. Finally, we plot the total rain fallout at the bottom of the domain in Fig. 14.

The main cloud forms at around $t = 300$ s, and the precipitating water reaches the ground at around $t = 900$ s (not shown). We note that both these times are significantly earlier than those reported in [46]. However, the latter uses a different equation for the temperature/energy, slightly different source terms governing phase changes and a different formula for the saturation vapour pressure. Nevertheless, this occurs in the convective tower as expected, and a pool of colder and denser air forms downwind of the tower, see Fig. 12, consistent with the literature [15,46,9]. The main cloud then spreads out into the expected anvil shape, and the profile of the potential temperature perturbation is consistent with the literature. We attribute the visual differences to the fact that our energy equation takes the effects due to moisture into account, and there are differences in the source terms governing phase changes.

The total accumulated rain is larger than that reported in the literature. However, it reduces with mesh refinement, consistent with faster storm development and less total accumulated rain reported in [15,9]. The accumulated rain on the lowest mesh is indeed consistent with the results in the literature. We also observe multiple peaks in the continuous Galerkin discretisation in [9]. However, in that work, the secondary peaks were observed in the downwind direction rather than the upwind direction. However, more accumulated rain in the upwind direction is consistent with the literature, e.g., [46], where the two-dimensional case was also considered. This contrasts [9], where a fully three-dimensional simulation was run with a single element with polynomials of order four in the y -direction.

5. Conclusions

We presented a discontinuous Galerkin method for the equations governing moist atmospheric flows in their conservative form. In particular, we retained thermodynamic details in these equations, such as the different specific heats and heat capacities, which are often neglected in the literature. The source terms governing phase changes were based on Kessler's microphysics closure as taken from the COSMO model. To avoid the difficult modelling of a source term governing condensation, we considered a single moist density for both water vapour and cloud water in our hyperbolic balance-law equation set. To recover the individual vapour and cloud densities from the moist density, we had to solve an additional non-linear problem. This reconstruction was necessary in order to evaluate the (numerical) flux and in particular the source terms governing water phase changes. Since the problem of reconstructing the densities is algebraic, we presented a novel approach by solving this non-linear problem in each quadrature point, which is highly parallelisable. For time-stepping, we used an explicit scheme such that only mass-matrix problems had to be solved to advance the solution in time. As the Legendre polynomial basis is orthogonal, the mass matrix is diagonal and the

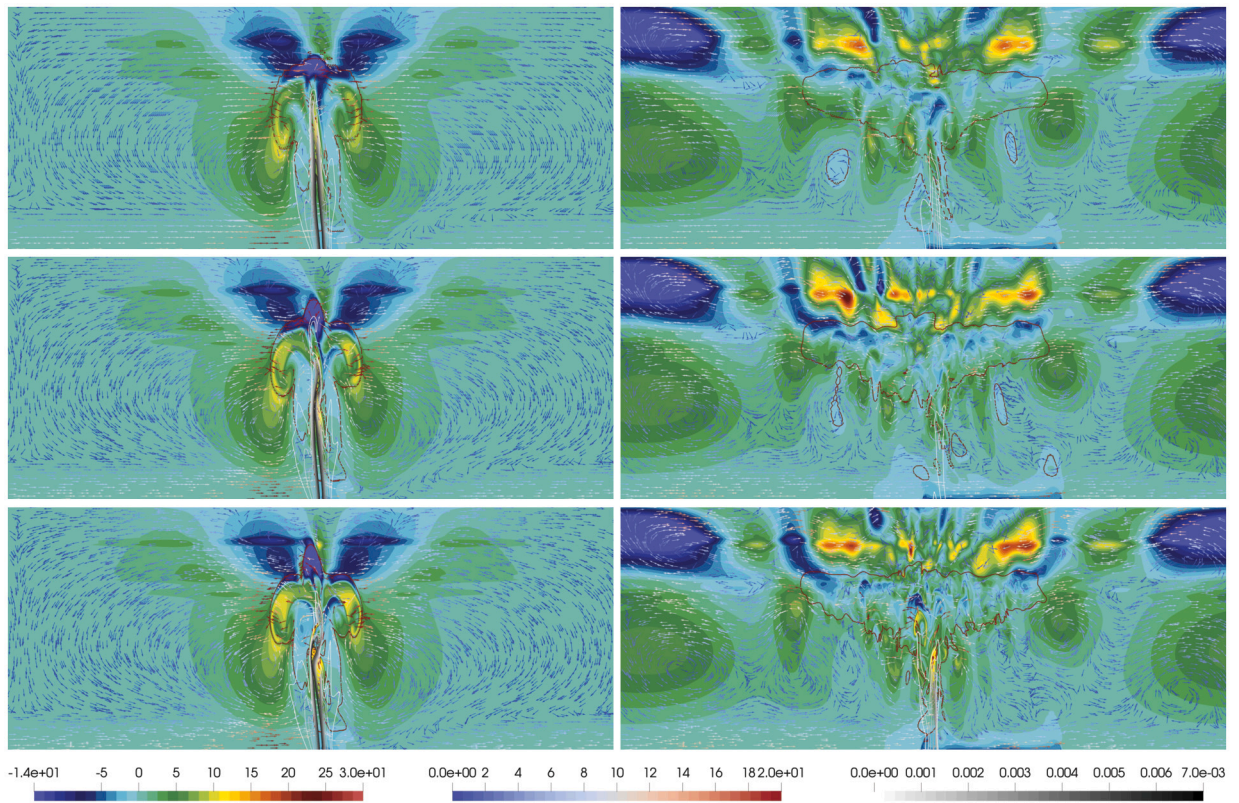


Fig. 12. Example 7: Potential temperature perturbation (background), cloud density contour at 10^{-5} (red), rain density contours (gray), and velocity field (blue to red arrows) for a two dimensional squall-line test below $z = 18$ km. Left: $t = 1500$ s, Right: $t = 1500$ s. Computed on unstructured simplicial meshes with $k = 1$. Top: $h = 500$ m, $\Delta t = 0.2$ s, centre: $h = 250$ m, $\Delta t = 0.1$ s, bottom: $h = 125$ m, $\Delta t = 0.05$ s.

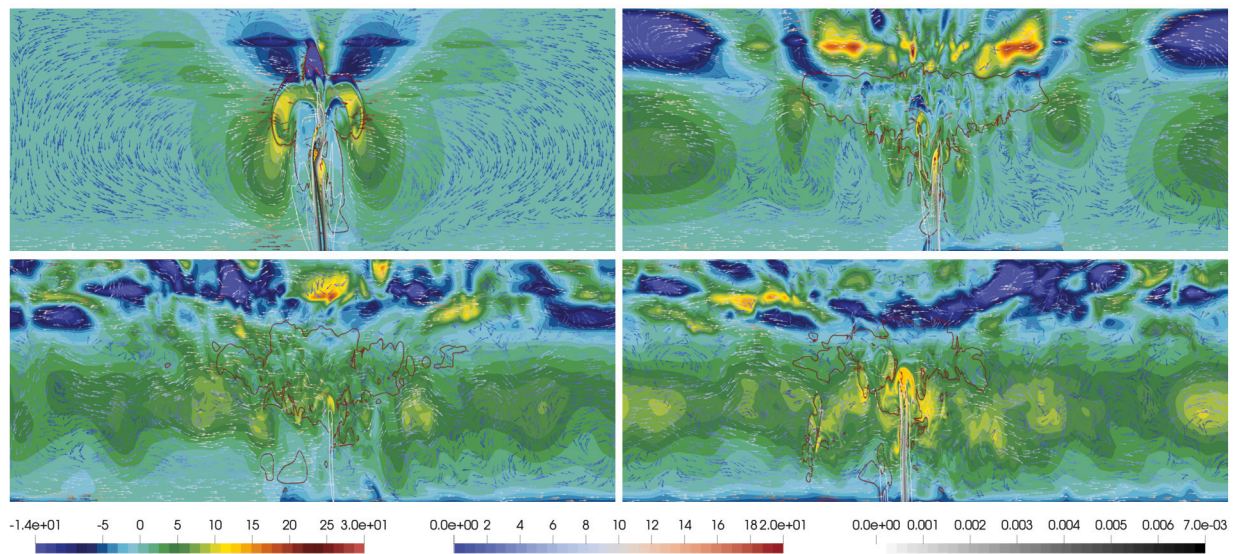


Fig. 13. Example 7: Potential temperature perturbation (background), cloud density contour at 10^{-5} (red), rain density contours (gray), and velocity field (blue to red arrows) for a two dimensional squall-line test below $z = 18$ km. Computed on unstructured simplicial meshes with $h = 125$ m, $k = 1$, and $\Delta t = 0.05$ s. From left to right and top to bottom: $t = 1500, 3000, 6000, 9000$ s.

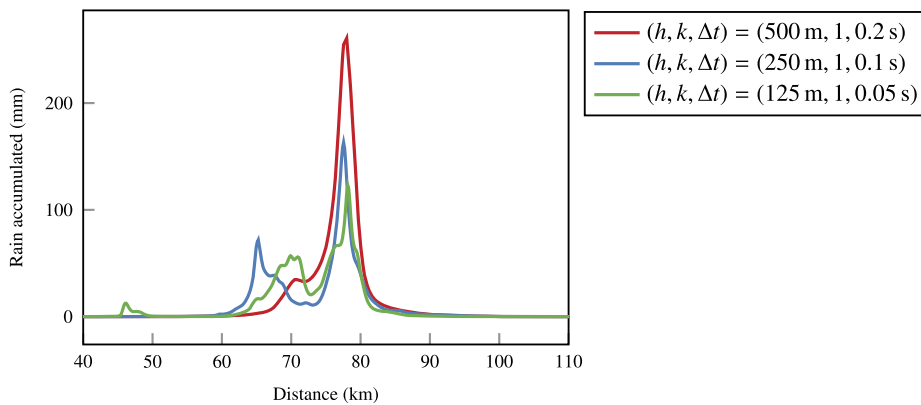


Fig. 14. Example 7: Total rain fallout at the bottom of the domain for a two dimensional squall-line test after $t = 9000$ s.

mass-matrix problems could be solved matrix-free. The combination of the local density and temperature reconstruction with the explicit time-stepping scheme led to a highly parallelisable scheme. To stabilise the method, we added an artificial viscosity term with local, dynamic, and asymptotically vanishing viscosity parameter. This is in contrast with the artificial viscosity widely used in the literature on DG for atmospheric flows which uses a constant and non-vanishing scaling, which in turn changes the problem at hand.

We validated our scheme on a number of numerical examples taken from the literature. We illustrated the optimal high-order convergence of the method for polynomial orders from one to four in the case of moist dynamics without rain. We also illustrated that the method works on both structured tensor-product meshes and unstructured simplicial meshes. This flexibility with respect to the mesh and the higher-order convergence are particular advantages of discontinuous Galerkin approaches. The mesh flexibility also allows for good parallel load balancing in every direction. We also noted that even on unstructured meshes, higher-order schemes performed well in retaining structures (symmetry) of the initial data and geometry. Overall, we obtained results in good agreement with the literature.

Our examples, which included rain, were also stable, again with a higher-order choice for the polynomial order, resulting in finer details. Our results for the storm resulting from a squall-line set-up were consistent with the available literature. However, it remains an open and interesting question to study the exact effects of retaining the thermodynamic details in the energy equation, which are usually neglected.

CRediT authorship contribution statement

Sabine Doppler: Conceptualization, Funding acquisition, Project administration, Writing – review & editing. **Philip L. Lederer:** Investigation, Software, Writing – review & editing. **Joachim Schöberl:** Funding acquisition, Methodology, Software, Writing – review & editing. **Henry von Wahl:** Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code used to realise the results presented in this paper is available online in the github repository <https://github.com/hvonwah/cloud-models-code>, and is archived on zenodo [37].

Acknowledgements

SH, JS and HvW acknowledge funding by the Austrian Science Fund (FWF) through the research program “Taming complexity in partial differential systems” project [10.55776/F65](https://www.fwf.ac.at/foyer/foyer.do?method=showProject&id=10.55776/F65). The computational results presented have been achieved in part using the Vienna Scientific Cluster (VSC). We would also like to thank Rupert Klein for many fruitful discussions.

Appendix A. Thermodynamic parameters

The values of the thermodynamic variables used are given in Table [A.1](#).

Table A.1

Thermodynamic equation of state parameters for moist air at reference temperature. Source: Cotton et al. 2011.

Parameter	Value	Units	Description
c_l	4.218×10^3	$\text{J kg}^{-1} \text{K}^{-1}$	Specific heat of liquid water (at T_{ref}).
c_{pd}	1.005×10^3	$\text{J kg}^{-1} \text{K}^{-1}$	Specific heat of dry air at constant pressure.
c_{pv}	1.850×10^3	$\text{J kg}^{-1} \text{K}^{-1}$	Specific heat of water vapour at constant pressure.
c_{vd}	7.18×10^2	$\text{J kg}^{-1} \text{K}^{-1}$	Specific heat of dry air at constant volume.
c_{vv}	1.390×10^3	$\text{J kg}^{-1} \text{K}^{-1}$	Specific heat of water vapour at constant volume.
e_{ref}	6.107×10^2	Pa	Saturation vapour pressure with respect to water (at T_{ref}).
L_{ref}	2.835×10^6	J kg^{-1}	Latent heat of vaporization (at T_{ref}).
R_d	2.8705×10^2	$\text{J kg}^{-1} \text{K}^{-1}$	Gas constant for dry air.
R_v	4.6151×10^2	$\text{J kg}^{-1} \text{K}^{-1}$	Gas constant for water vapour.
T_{ref}	2.7315×10^2	K	Reference temperature.
p_{ref}	1.0×10^5	Pa	Reference pressure.
ϵ	6.22×10^{-1}		$\epsilon \equiv R_d/R_v$.

Appendix B. Computation of initial conditions

B.1. Inertia gravity waves in a saturated atmosphere

The hydrostatic state is determined by solving the (1-dimensional) problem: Find \bar{p} , $\bar{\rho}_d$, $\bar{\rho}_{vs}$ and \bar{T} such that

$$\begin{aligned} \partial_z \bar{p} &= -(1 + q_w) \bar{\rho}_d g, \\ \bar{p} &= (\bar{\rho}_d R_d + \bar{\rho}_{vs} R_v) \bar{T}, \\ \bar{\rho}_{vs} R_v \bar{T} &= e_{\text{ref}} \left(\frac{\bar{T}}{T_{\text{ref}}} \right)^{\frac{c_{pv} - c_l}{R_v}} \exp \left(\frac{L_{\text{ref}} - (c_{pv} - c_l) T_{\text{ref}}}{R_v} \left(\frac{1}{T_{\text{ref}}} - \frac{1}{\bar{T}} \right) \right), \\ \theta_e &= \bar{T} \left(\frac{\bar{\rho}_d R_d \bar{T}}{p_{\text{ref}}} \right)^{-R_d/(c_{pd} + c_l q_w)} \exp \left(\frac{(L_{\text{ref}} + (c_{pv} - c_l)(\bar{T} - T_{\text{ref}})) \bar{\rho}_{vs}}{\bar{\rho}_d (c_{pd} + c_l q_w) \bar{T}} \right), \end{aligned}$$

and the pressure boundary condition is $\bar{p}(0) = p_{\text{ref}}$. The definition of θ_e is taken from [47]. The initial perturbation is computed by solving the non-linear problem: Find (ρ_d, ρ_{vs}, T) such that

$$\bar{\theta}_e + \theta'_e = T \left(\frac{\rho_d R_d T}{p_{\text{ref}}} \right)^{-R_d/(c_{pd} + c_l q_w)} \exp \left(\frac{(L_{\text{ref}} + (c_{pv} - c_l)(T - T_{\text{ref}})) \rho_{vs}}{\rho_d (c_{pd} + c_l q_w) T} \right), \quad (\text{B.1a})$$

$$\bar{p} = (\rho_d R_d + \rho_{vs} R_v) T, \quad (\text{B.1b})$$

$$\rho_{vs} R_v T = e_{\text{ref}} \left(\frac{T}{T_{\text{ref}}} \right)^{\frac{c_{pv} - c_l}{R_v}} \exp \left(\frac{L_{\text{ref}} - (c_{pv} - c_l) T_{\text{ref}}}{R_v} \left(\frac{1}{T_{\text{ref}}} - \frac{1}{T} \right) \right), \quad (\text{B.1c})$$

and set $\rho'_d = \rho_d - \bar{\rho}_d$, $\rho'_v = \rho_{vs} - \bar{\rho}_{vs}$, $\rho'_m = \rho'_d - \rho'_v$ and $T' = T - \bar{T}$. The total initial energy (perturbation) can be computed from (4).

B.2. Bryan-Fritsch moist benchmark

The hydrostatic base state is computed as in Appendix B.1. The initial perturbations are then defined by perturbing the *density potential temperature*

$$\theta_\rho = T \left(\frac{p_{\text{ref}}}{p} \right)^{R_d/c_{pd}} (1 + q_v/\epsilon) = \theta_d (1 + q_v/\epsilon), \quad (\text{B.2})$$

where $\theta_d(p, T)$ is the *dry potential temperature*. This perturbation is defined as

$$\theta' = 2 \cos^2 \left(\frac{\pi L}{2} \right), \quad \text{with} \quad L = \min \left\{ \left\| \left(\frac{x - x_c}{x_r}, \frac{z - z_c}{z_r} \right) \right\|_2, 1 \right\},$$

where $x_c = 10 \text{ km}$, $z_c = 2 \text{ km}$ and $x_r, z_r = 2 \text{ km}$. The perturbation to the potential temperature is then introduced by assuming that the resulting buoyancy is the same as the buoyancy in a similar the dry test case with $\theta_d = 300 \text{ K}$. This leads to the equation

$$\theta_d(p, T) \left[1 + \frac{q_{vs}(p, T)}{\epsilon} \right] = \theta_{\rho 0} (1 + q_w) \left(\frac{\theta'_d}{300} + 1 \right),$$

which we solve for T in every point, using the hydrostatic pressure $p = \bar{p}$ and (6) for the saturation vapour mixing ratio. With this temperature, we can then reconstruct the vapour and cloud densities using the saturation vapour pressure.

B.3. Inertia gravity waves in a saturated atmosphere without initial clouds

The hydrostatic state is determined by solving the (1-dimensional) problem: Find $\bar{p}, \bar{\rho}_d, \bar{\rho}_{vs}$ and \bar{T} such that

$$\begin{aligned}\partial_z \bar{p} &= -(\bar{\rho}_d + \bar{\rho}_{vs})g, \\ \bar{p} &= (\bar{\rho}_d R_d + R_v \bar{\rho}_{vs})\bar{T}, \\ \bar{\rho}_{vs} R_v \bar{T} &= e_{\text{ref}} \left(\frac{\bar{T}}{T_{\text{ref}}} \right)^{\frac{c_{pv}-c_l}{R_v}} \exp \left(\frac{L_{\text{ref}} - (c_{pv} - c_l)T_{\text{ref}}}{R_v} \left(\frac{1}{T_{\text{ref}}} - \frac{1}{\bar{T}} \right) \right), \\ \theta_e &= \bar{T} \left(\frac{\bar{\rho}_d R_d \bar{T}}{p_{\text{ref}}} \right)^{-R_d/(c_{pd}+c_l(\bar{\rho}_{vs}/\bar{\rho}_d))} \exp \left(\frac{(L_{\text{ref}} + (c_{pv} - c_l)(\bar{T} - T_{\text{ref}}))\bar{\rho}_{vs}}{(c_{pd}\bar{\rho}_d + c_l\bar{\rho}_{vs})\bar{T}} \right),\end{aligned}$$

and the pressure boundary condition $\bar{p}(0) = p_{\text{ref}}$. The perturbation of the hydrostatic state can then be computed analogously to (B.1) by replacing q_w with $q_{vs} = \rho_{vs}/\rho_d$.

B.4. Atmosphere at rest with a steep mountain

With the temperature profile $\bar{T}(z)$ given in (22) water vapour is at saturation and there is no cloud water present. The hydrostatic state can then be computed by solving the following (one-dimensional) non-linear problem: Find $\bar{p}, \bar{\rho}_d$, such that

$$\partial_z \bar{p} = -(\bar{\rho}_d + \bar{\rho}_{vs})g, \quad \bar{p} = (\bar{\rho}_d R_d + \bar{\rho}_{vs} R_v)\bar{T}, \quad \text{with} \quad \bar{\rho}_{vs}(\bar{T}) = \frac{e_s(\bar{T})}{R_v \bar{T}},$$

the pressure boundary condition $\bar{p}(0) = p_{\text{ref}}$, and where the saturation-vapour pressure is computed from the Clausius–Clapeyron relation (7).

B.5. Rising thermal with rain

The relative humidity is related to our variables by

$$\mathcal{H} = \frac{q_v}{q_{vs}} \left(\frac{1 + q_{vs}/\epsilon}{1 + q_v/\epsilon} \right).$$

The hydrostatic base state is computed by solving the (one-dimensional) problem: Find $\bar{p}, \bar{\rho}_d, \bar{\rho}_v, \bar{T}$, such that

$$\begin{aligned}\partial_z \bar{p} &= -(\bar{\rho}_d + \bar{\rho}_v)g, \\ \bar{p} &= (\bar{\rho}_d R_d + \bar{\rho}_v R_v)\bar{T}, \\ \frac{\bar{q}_v}{\bar{q}_{vs}} \left(\frac{1 + \bar{q}_{vs}/\epsilon}{1 + \bar{q}_v/\epsilon} \right) &= 0.2, \\ \bar{T} \left(\frac{p_{\text{ref}}}{\bar{p}} \right)^{R_d/c_{pd}} &= T_{\text{surf}} \left(\frac{p_{\text{ref}}}{8.5 \times 10^4} \right)^{R_d/c_{pd}} \exp(1.3 \times 10^{-5} z),\end{aligned}$$

the pressure boundary condition $\bar{p}(0) = 8.5 \times 10^4$, and with $\bar{q}_v = \bar{\rho}_v/\bar{\rho}_d, \bar{q}_{vs} = \bar{\rho}_{vs}/\bar{\rho}_d$.

The initial relative humidity is given by

$$\tilde{\mathcal{H}} = \begin{cases} \bar{\mathcal{H}} & r > r_1, \\ \bar{\mathcal{H}} + (1 - \bar{\mathcal{H}}) \cos^2 \left(\frac{\pi(r-r_1)}{2(r_1-r_2)} \right) & r_2 \leq r < r_1, \\ 1 & r < r_2, \end{cases}$$

with the radius $r = \sqrt{(x - c_x)^2 + (z - c_z)^2}$, $c_x = L/2$, $c_z = 800$ m, $r_1 = 300$ m and $r_2 = 200$ m. The initial perturbations are then obtained by solving the problem

$$\frac{q_v}{q_{vs}(T)} \left(\frac{1 + q_{vs}(T)/\epsilon}{1 + q_v/\epsilon} \right) = \tilde{\mathcal{H}}, \quad T \left(\frac{p_{\text{ref}}}{\bar{p}} \right)^{R_d/c_{pd}} = \bar{\theta}_d, \quad \rho_d R_d T + \rho_v R_v T = \bar{p},$$

for ρ_v, ρ_d and T . The initial cloud and rain densities are zero, and the initial velocity is at rest, i.e., $\rho_v = \rho_r = 0, u = (0, 0)^T$ and the initial energy density can be computed from these variables.

B.6. Squall line

The hydrostatic base state is computed by solving the one-dimensional problem: Given $\bar{\theta}, \bar{q}_v$, find $(\bar{p}, \bar{\rho}_d, \bar{\rho}_v, \bar{T})$ such that

$$\partial_z \bar{p} = -(\bar{\rho}_d + \bar{\rho}_v)g, \quad \bar{p} = (\bar{\rho}_d R_d + \bar{\rho}_v R_v) \bar{T}, \quad \bar{\rho}_v = \bar{\rho}_d \bar{q}_v, \quad \bar{\theta} = T \left(\frac{p_{\text{ref}}}{p} \right)^{R_d/c_{pd}},$$

with pressure boundary condition $p(0) = p_{\text{ref}}$. The initial temperature perturbation defined by

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

with

$$r = \sqrt{\frac{(x - x_c)^2}{r_x^2} + \frac{(z - z_c)^2}{r_z^2}}, \quad x_c = 75 \text{ km}, \quad r_x = 10 \text{ km}, \quad z_c = 2 \text{ km}, \quad r_x = 1.5 \text{ km}, \quad r_c = 1, \quad \theta_c = 3 \text{ K}.$$

We obtain the temperature perturbation by solving the constitutive equation for the potential temperature

$$\theta = T \left(\frac{p_{\text{ref}}}{p} \right)^{R_d/c_{pd}} = T \left(\frac{p_{\text{ref}}}{(\rho_d R_d + \rho_v R_v) T} \right)^{R_d/c_{pd}}.$$

References

- [1] S. Hittmeir, R. Klein, Asymptotics for moist deep convection I: Refined scalings and self-sustaining updrafts, *Theor. Comput. Fluid Dyn.* 32 (2017) 137–164, <https://doi.org/10.1007/s00162-017-0443-z>.
- [2] S. Marras, J.F. Kelly, M. Moragues, A. Müller, M.A. Kopera, M. Vázquez, F.X. Giraldo, G. Houzeaux, O. Jorba, A review of element-based Galerkin methods for numerical weather prediction: Finite elements, spectral elements, and discontinuous Galerkin, *Arch. Comput. Methods Eng.* 23 (2015) 673–722, <https://doi.org/10.1007/s11831-015-9152-1>.
- [3] A. Staniforth, J. Thuburn, Horizontal grids for global weather and climate prediction models: a review, *Q. J. R. Meteorol. Soc.* 138 (2011) 1–26, <https://doi.org/10.1002/qj.958>.
- [4] R. Aubry, M. Vazquez, G. Houzeaux, J.M. Cela, S. Marras, An unstructured CFD approach for numerical weather prediction, in: *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, American Institute of Aeronautics and Astronautics*, 2010.
- [5] P.K. Smolarkiewicz, J. Szmelter, A.A. Wyszogrodzki, An unstructured-mesh atmospheric model for nonhydrostatic dynamics, *J. Comput. Phys.* 254 (2013) 184–199, <https://doi.org/10.1016/j.jcp.2013.07.027>.
- [6] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution of the 2d Euler equations, *J. Comput. Phys.* 138 (1997) 251–285, <https://doi.org/10.1006/jcph.1997.5454>.
- [7] C.E. Baumann, J.T. Oden, A discontinuous finite element method for the Euler and Navier-Stokes equations, *Int. J. Numer. Methods Fluids* 31 (1999) 79–95, [https://doi.org/10.1002/\(sici\)1097-0363\(19990915\)31:1<79::aid-fld956>3.0.co;2-c](https://doi.org/10.1002/(sici)1097-0363(19990915)31:1<79::aid-fld956>3.0.co;2-c).
- [8] R. Hartmann, P. Houston, Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations, *J. Comput. Phys.* 183 (2002) 508–532, <https://doi.org/10.1006/jcph.2002.7206>.
- [9] Y. Tissaoui, S. Marras, A. Quaini, F.A.V. De Braganca Alves, F.X. Giraldo, A non-column based, fully unstructured implementation of Kessler's microphysics with warm rain using continuous and discontinuous spectral elements, *J. Adv. Model. Earth Syst.* 15 (2023), <https://doi.org/10.1029/2022ms003283>.
- [10] D. Schuster, S. Brdar, M. Baldauf, A. Dedner, R. Klöforn, D. Kröner, On discontinuous Galerkin approach for atmospheric flow in the mesoscale with and without moisture, *Meteorol. Z.* 23 (2014) 449–464, <https://doi.org/10.1127/0941-2948/2014/0565>.
- [11] T.M. Bendall, T.H. Gibson, J. Shipton, C.J. Cotter, B. Shipway, A compatible finite-element discretisation for the moist compressible Euler equations, *Q. J. R. Meteorol. Soc.* 146 (2020) 3187–3205, <https://doi.org/10.1002/qj.3841>.
- [12] A. Sridhar, Y. Tissaoui, S. Marras, Z. Shen, C. Kawczynski, S. Byrne, K. Pamnany, M. Waruszewski, T.H. Gibson, J.E. Kozdon, V. Churavy, L.C. Wilcox, F.X. Giraldo, T. Schneider, Large-eddy simulations with ClimateMachine v0.2.0: a new open-source code for atmospheric simulations on GPUs and CPUs, *Geosci. Model Dev.* 15 (2022) 6259–6284, <https://doi.org/10.5194/gmd-15-6259-2022>.
- [13] M. Duarte, A.S. Almgren, K. Balakrishnan, J.B. Bell, D.M. Roms, A numerical study of methods for moist atmospheric flows: Compressible equations, *Mon. Weather Rev.* 142 (2014) 4269–4283, <https://doi.org/10.1175/mwr-d-13-00368.1>.
- [14] K.V. Ooyama, A thermodynamic foundation for modeling the moist atmosphere, *J. Atmos. Sci.* 47 (1990) 2580–2593, [https://doi.org/10.1175/1520-0469\(1990\)047<2580:affmt>2.0.co;2](https://doi.org/10.1175/1520-0469(1990)047<2580:affmt>2.0.co;2).
- [15] S. Gaberšek, F.X. Giraldo, J.D. Doyle, Dry and moist idealized experiments with a two-dimensional spectral element model, *Mon. Weather Rev.* 140 (2012) 3163–3182, <https://doi.org/10.1175/mwr-d-11-00144.1>.
- [16] W. Cotton, G. Bryan, S. van den Heever (Eds.), *Storm and Cloud Dynamics*, 2nd ed., *International Geophysics*, vol. 99, 2011.
- [17] K.V. Ooyama, A dynamic and thermodynamic foundation for modeling the moist atmosphere with parameterized microphysics, *J. Atmos. Sci.* 58 (2001) 2073–2102, [https://doi.org/10.1175/1520-0469\(2001\)058<2073:adatff>2.0.co;2](https://doi.org/10.1175/1520-0469(2001)058<2073:adatff>2.0.co;2).
- [18] F.X. Giraldo, M. Restelli, A study of spectral element and discontinuous Galerkin methods for the Navier–Stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases, *J. Comput. Phys.* 227 (2008) 3849–3877, <https://doi.org/10.1016/j.jcp.2007.12.009>.
- [19] E. Kessler, On the distribution and continuity of water substance in atmospheric circulations, in: *On the Distribution and Continuity of Water Substance in Atmospheric Circulations*, American Meteorological Society, Boston, MA, 1969, pp. 1–84.
- [20] G. Doms, J. Förstner, E. Heise, H.-J. Herzog, D. Mironov, M. Raschendorfer, T. Reinhardt, B. Ritter, R. Schrodin, J.-P. Schulz, G. Vogel, COSMO-Model Version 6.00: A Description of the Nonhydrostatic Regional COSMO-Model - Part II: Physical Parametrizations, Technical Report, COSMO Consortium for Small-Scale Modelling, 2021.
- [21] D.M. Roms, The dry-entropy budget of a moist atmosphere, *J. Atmos. Sci.* 65 (2008) 3779–3799, <https://doi.org/10.1175/2008jas2679.1>.
- [22] S. Zaglmayr, High Order Finite Element Methods for Electromagnetic Field Computation, Ph.D. thesis, Johannes Kepler Universität Linz, 2006, [numa.jku.at/media/finder_public/e6/98/e6988974-e1ce-4cb2-ac0e-bcfd20350e7/phd-zaglmayr.pdf](https://media/finder_public/e6/98/e6988974-e1ce-4cb2-ac0e-bcfd20350e7/phd-zaglmayr.pdf).
- [23] J.S. Hesthaven, T. Warburton, *Nodal Discontinuous Galerkin Methods*, Springer, New York, 2008.
- [24] D.A.D. Pietro, A. Ern, *Mathematical aspects of discontinuous Galerkin methods*, Springer, Berlin, Heidelberg, 2012.

- [25] D.A. Kopriva, *Implementing Spectral Methods for Partial Differential Equations*, Springer, Dordrecht, 2009.
- [26] F.X. Giraldo, *An Introduction to Element-Based Galerkin Methods on Tensor-Product Bases*, Springer, Cham, 2020.
- [27] F.X. Giraldo, J.F. Kelly, E.M. Constantinescu, Implicit-explicit formulations of a three-dimensional nonhydrostatic unified model of the atmosphere (NUMA), *SIAM J. Sci. Comput.* 35 (2013) B1162–B1194, <https://doi.org/10.1137/120876034>.
- [28] D.S. Abdi, F.X. Giraldo, E.M. Constantinescu, L.E. Carr, L.C. Wilcox, T.C. Warburton, Acceleration of the IMplicit-EXplicit nonhydrostatic unified model of the atmosphere on manycore processors, *Int. J. High Perform. Comput. Appl.* 33 (2017) 242–267, <https://doi.org/10.1177/1094342017732395>.
- [29] J.F.B.M. Kraaijevanger, Contractivity of Runge-Kutta methods, *BIT* 31 (1991) 482–528, <https://doi.org/10.1007/bf01933264>.
- [30] S. Gottlieb, D. Ketcheson, C.-W. Shu, *Strong Stability Preserving Runge-Kutta and Multistep Time Discretizations*, World Scientific, Singapore, 2011.
- [31] J.F. Kelly, F.X. Giraldo, Continuous and discontinuous Galerkin methods for a scalable three-dimensional nonhydrostatic atmospheric model: Limited-area mode, *J. Comput. Phys.* 231 (2012) 7988–8008, <https://doi.org/10.1016/j.jcp.2012.04.042>.
- [32] J. Sachsperger, S. Serafin, V. Grubišić, Dynamics of rotor formation in uniformly stratified two-dimensional flow over a mountain, *Q. J. R. Meteorol. Soc.* 142 (2016) 1201–1212, <https://doi.org/10.1002/qj.2746>.
- [33] M. Diamantakis, J. Flemming, Global mass fixer algorithms for conservative tracer transport in the ECMWF model, in: *Technical Memorandum*, vol. 713, European Centre for Medium-Range Weather Forecasts, 2013, <https://www.ecmwf.int/sites/default/files/elibrary/2013/9055-global-mass-fixer-algorithms-conservative-tracer-transport-ecmwf-model.pdf>.
- [34] S. Malardel, M. Diamantakis, A. Augusti-Panareda, J. Flemming, Dry mass versus total mass conservation in the IFS, in: *Technical Memorandum*, vol. 849, European Centre for Medium-Range Weather Forecasts, 2019, www.ecmwf.int/sites/default/files/elibrary/2019/19114-dry-mass-versus-total-mass-conservation-ifs.pdf.
- [35] J. Schöberl, NETGEN an advancing front 2D/3D-mesh generator based on abstract rules, *Comput. Vis. Sci.* 1 (1997) 41–52, <https://doi.org/10.1007/s007910050004>.
- [36] J. Schöberl, C++11 implementation of finite elements in NGSolve, Technical Report, Institute for Analysis and Scientific Computing, TU Wien, 2014, www.asc.tuwien.ac.at/~schoeberl/wiki/publications/ngs-cpp11.pdf.
- [37] S. Doppler, P.L. Lederer, J. Schöberl, H. von Wahl, A discontinuous Galerkin approach for atmospheric flows with implicit condensation - Reproduction scripts, <https://doi.org/10.5281/zenodo.8422378>, 2023.
- [38] G.H. Bryan, J.M. Fritsch, A benchmark simulation for moist nonhydrostatic numerical models, *Mon. Weather Rev.* 130 (2002) 2917–2928, [https://doi.org/10.1175/1520-0493\(2002\)130<2917:absfmm>2.0.co;2](https://doi.org/10.1175/1520-0493(2002)130<2917:absfmm>2.0.co;2).
- [39] W.W. Grabowski, T.L. Clark, Cloud–environment interface instability: Rising thermal calculations in two spatial dimensions, *J. Atmos. Sci.* 48 (1991) 527–546, [https://doi.org/10.1175/1520-0469\(1991\)048<0527:ciirtc>2.0.co;2](https://doi.org/10.1175/1520-0469(1991)048<0527:ciirtc>2.0.co;2).
- [40] W.W. Grabowski, T.L. Clark, Cloud–environment interface instability: Part II: Extension to three spatial dimensions, *J. Atmos. Sci.* 50 (1993) 555–573, [https://doi.org/10.1175/1520-0469\(1993\)050<0555:ceiipi>2.0.co;2](https://doi.org/10.1175/1520-0469(1993)050<0555:ceiipi>2.0.co;2).
- [41] W.C. Skamarock, J.B. Klemp, Efficiency and accuracy of the Klemp-Wilhelmson time-splitting technique, *Mon. Weather Rev.* 122 (1994) 2623–2630, [https://doi.org/10.1175/1520-0493\(1994\)122<2623:eeaotk>2.0.co;2](https://doi.org/10.1175/1520-0493(1994)122<2623:eeaotk>2.0.co;2).
- [42] G. Zängl, Extending the numerical stability limit of terrain-following coordinate models over steep slopes, *Mon. Weather Rev.* 140 (2012) 3722–3733, <https://doi.org/10.1175/mwr-d-12-00049.1>.
- [43] W.W. Grabowski, P.K. Smolarkiewicz, Two-time-level semi-Lagrangian modeling of precipitating clouds, *Mon. Weather Rev.* 124 (1996) 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).
- [44] M.L. Weisman, J.B. Klemp, R. Rotunno, Structure and evolution of numerically simulated squall lines, *J. Atmos. Sci.* 45 (1988) 1990–2013, [https://doi.org/10.1175/1520-0469\(1988\)045<1990:saeons>2.0.co;2](https://doi.org/10.1175/1520-0469(1988)045<1990:saeons>2.0.co;2).
- [45] R. Rotunno, J.B. Klemp, M.L. Weisman, A theory for strong, long-lived squall lines, *J. Atmos. Sci.* 45 (1988) 463–485, [https://doi.org/10.1175/1520-0469\(1988\)045<0463:atfsll>2.0.co;2](https://doi.org/10.1175/1520-0469(1988)045<0463:atfsll>2.0.co;2).
- [46] S. Marras, M. Moragues, M. Vázquez, O. Jorba, G. Houzeaux, Simulations of moist convection by a variational multiscale stabilized finite element method, *J. Comput. Phys.* 252 (2013) 195–218, <https://doi.org/10.1016/j.jcp.2013.06.006>.
- [47] K.A. Emanuel, *Atmospheric Convection*, Oxford University Press, New York, 1994.