Testing refinement criteria in adaptive Discontinuous Galerkin simulations of dry atmospheric convection

Andreas Müller[a,*], Jörn Behrens[b], Francis X. Giraldo[c], Volkmar Wirth[a]

[a] *Institute for Atmospheric Physics, University of Mainz (Germany), Becherweg 21, D-55127 Mainz*
[b] *KlimaCampus, University of Hamburg (Germany), Am Grindelberg 5, D-20144 Hamburg*
[c] *Department of Applied Mathematics, Naval Postgraduate School, Monterey, California (USA), Monterey, CA 93943-5216*

Abstract

Adaptive mesh refinement generally serves to increase computational efficiency without compromising the accuracy of the numerical solution. However it is an open question in which regions the spatial resolution can actually be coarsened without affecting the accuracy of the result. This question is investigated for a specific meteorological problem, namely the simulation of atmospheric convection. For this purpose a novel numerical model is developed that is tailored towards this specific meteorological problem. The compressible Euler equations are solved with a Discontinuous Galerkin method. Time integration is done with a semi-implicit approach and the dynamic grid adaptivity uses space filling curves via the AMATOS function library. So far the model is able to simulate dry flow in two-dimensional geometry without subgrid-scale modeling. The model is validated with three standard test cases.

A method is introduced which allows one to compare the accuracy between different choices of refinement regions even in a case when the exact solution is not known. Essentially this is done by comparing features of the solution that are strongly sensitive to spatial resolution. For a rising warm air bubble the average number of elements required for the adaptive simulation is about a factor three times smaller than the number required for the simulation with the uniform fine-resolution grid. Correspondingly the adaptive simulation is almost three times faster than the uniform simulation, and the advantage of adaptive mesh refinement becomes even more pronounced for larger domains. This result suggests that adaptive mesh refinement should have significant potential for future simulations of atmospheric moist convection when the refinement criterion is chosen carefully.

Keywords: Adaptive Mesh Refinement, Discontinuous Galerkin, Semi-Implicit, Meteorology, Dry Atmospheric Convection

1. Introduction

Significant progress in numerous areas of scientific computing comes from the steadily increasing capacity of computers and the advances in numerical methods. An example is the simulation of the Earth's atmosphere, which has proven to be extremely challenging owing to its multiscale and multi-process nature. Even with today's computers it is impossible to explicitly represent all scales and all processes involved. To overcome this difficulty one resorts to empirically-based closure approaches – called "parameterisations" – that try to capture the unresolved aspects of the problem. Needless to say, this introduces errors.

One possibility for reducing the computational effort of simulations is given by adaptive mesh refinement. It allows the adaption of the spatial and temporal resolution to local properties of the atmosphere. This adaption is controlled by refinement criteria. Adaptive mesh refinement has been applied successfully to

*Corresponding author. Tel.: +49 6131 39 22867, fax: +49 6131 39 23532.
Email addresses: andrmue@uni-mainz.de (Andreas Müller), joern.behrens@zmaw.de (Jörn Behrens), fxgirald@nps.edu (Francis X. Giraldo), vwirth@uni-mainz.de (Volkmar Wirth)

atmospheric sciences for over 20 years [1, 2]. Recently, this technique has found increased interest, since new grid-independent numerical methods of Galerkin type and finite volumes have emerged [3–6]. However there are still many unsolved questions [7]. A more comprehensive exposition of adaptive methods in atmospheric modeling is given in [8].

One important unsolved question is: in which regions can the spatial resolution be coarsened without affecting the accuracy of the simulation? Starting with a uniform simulation the additional error introduced by coarsening the mesh in some regions for an adaptive simulation should be much smaller compared to the inherent numerical error of the uniform simulation. However, for realistic meteorological applications the exact solution is not known and the exact error cannot be computed. Therefore we have to find some approximate measure for the error. It is difficult to solve this task in general. For this reason we focus on a specific meteorological application.

Our final goal is to develop a simulation that is able to cover a cumulus cloud as a whole and resolve smaller eddies at the cloud-environment interface simultaneously. A simulation of this problem using a uniform grid is still far beyond the capacity of today's computing power. However it appears possible when using adaptive mesh refinement. This application is important for meteorological research because the impact of evaporative cooling on the evolution of cumulus clouds is still not fully understood [9]. Furthermore this work should allow important insight into the simulation of scales between mesoscale models on larger domains and large-eddy simulations for smaller scales (sometimes called "terra incognita" [10]).

We developed a novel numerical model that is tailored towards this specific meteorological problem. The compressible Euler equations are solved with a Discontinuous Galerkin method. Time integration is done with a semi-implicit approach and the dynamic grid adaptivity uses space filling curves via the AMATOS [11] function library. So far we are able to simulate dry flow in two-dimensional geometry without subgrid-scale modeling. This model allows us to investigate the question of how to choose and test refinement criteria in a simplified test environment.

The paper is organized as follows. First, in section 2 we give an introduction into the meteorological problem that motivates our work. In section 3 we then describe the numerical methods that we are using. In section 4 we apply our code to three test cases from the literature. Section 5 presents our new method for testing refinement criteria. The paper ends with a summary and outlook in section 6.

2. Meteorological motivation

A single cumulus cloud can be considered as a prototypical basic element of atmospheric moist convection. The cloud rises through the environmental air owing to its positive buoyancy (fig. 1). Upward motion of the cloud (thick blue arrow) is associated with downward motion in a thin shell surrounding the rising cloud (thin blue arrows) [9]. The induced wind shear at the cloud-environment interface and the different densities of cloudy and environmental air lead to instabilities which eventually result in turbulence [12–14]. The ensuing mixing between moist cloudy and dry environmental air leads to evaporation of cloud droplets. This cools the parcel resulting in negative buoyancy corresponding to a downward force (red arrows). This process is called "buoyancy reversal" [15–17].

Early indications for the significance of buoyancy reversal for cloud dynamics stem from the laboratory experiments of Johari [18]. Johari found that, depending on the strength of the buoyancy reversal, the morphology of the cloud development could be vastly different. Similar results were found in highly idealized numerical two-fluid experiments by Grabowski [19].

These preliminary investigations suggest that buoyancy reversal has an important impact on cloud dynamics. However, owing to their idealized nature it is not possible to draw any firm conclusion about real clouds. On the other hand, numerical weather prediction models and even so-called "cloud resolving models" are not able to explicitly simulate the processes relevant for buoyancy reversal due to their coarse spatial resolution [20]. It is here that we want to take a step forward by developing a new numerical model that is specifically designed to deal with the mixing processes at the cloud boundary.

A Direct Numerical Simulation (DNS) would require a resolution of about 1 mm in each direction in order to properly resolve all dynamical scales [20]. In three dimensions this amounts to some 10^{24} grid points,

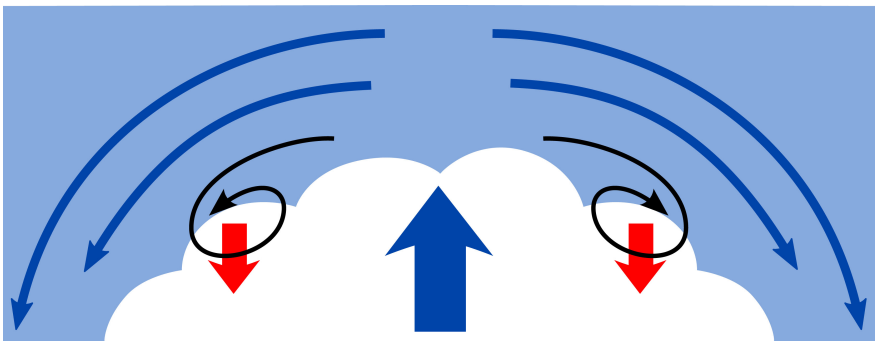

Figure 1: Illustration of buoyancy reversal. The blue arrows demonstrate the mean flow of a rising cloud, the black arrows represent turbulence produced by Kelvin-Helmholtz instability and the red arrows illustrate buoyancy reversal. For further explanation we refer to the text.

which is beyond the capacity of today's computing power. We, therefore, resort to Large Eddy Simulation (LES) in combination with a semi-implicit time integration method. The use of an adaptive technique will offer a significant reduction in numerical expense, as it allows us to focus attention to the cloud-environment interface, which is the region where mixing and buoyancy reversal takes place.

Measurements in real clouds have shown a large variety of behavior [21]: there are clouds with steep gradients in the interior (see for example the liquid water content in figure 15 of that reference [21]). On the other hand, smaller clouds often have a fairly smooth interior with discontinuities mostly at the boundary of the cloud (fig. 13 of Damiani et al. [21]). It is the latter which we intend to simulate.

As a first step we consider 2D, dry flow without a subgrid-scale modeling. This excludes the processes of buoyancy reversal but allows us to study the adaptive simulation of mixing processes in an idealized framework.

3. Numerical model

As described in the previous section our numerical model is tailored towards the simulation of cumulus clouds which have discontinuities at their boundaries. For this application a Discontinuous Galerkin (DG) discretization in combination with a semi-implicit time-integrator should be an excellent choice; the reason is the high-order accuracy and robustness in handling discontinuities of the DG method as well as the large time-steps allowed by the semi-implicit method. For avoiding the Gibbs phenomenon an artificial viscosity is used.

In the current work the fully compressible Euler equations are used (see equation set 2 in Giraldo and Restelli [22]):

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{U} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla} \cdot \left(\frac{\boldsymbol{U} \otimes \boldsymbol{U}}{\rho} + p\, \mathbf{I}_2 \right) = -\rho\, g\, \hat{\mathbf{k}}, \tag{2}$$

$$\frac{\partial \Theta}{\partial t} + \boldsymbol{\nabla} \cdot \left(\frac{\Theta\, \boldsymbol{U}}{\rho} \right) = 0, \tag{3}$$

with the variables $\left(\rho, \boldsymbol{U}^{\mathrm{T}}, \Theta\right)^{\mathrm{T}}$ where the superscript T stands for transpose, ρ is the density, $\boldsymbol{U} = (\rho\, u, \rho\, w)^{\mathrm{T}}$ is the momentum field, u is the horizontal wind speed, w is the vertical wind speed and $\Theta = \rho\, \theta$ is the density potential temperature. Furthermore we denote the gravitational constant with g, the divergence operator with $\nabla\cdot$, the tensor product by $\otimes$, the identity matrix in $\mathbb{R}^2$ by $\mathbf{I}_2$ and the unit vector in the vertical direction with $\hat{\mathbf{k}}$.

Pressure p in eq. (2) is given by the equation of state:

$$p = p_0 \left(\frac{R\,\Theta}{p_0} \right)^{\frac{c_p}{c_v}}, \tag{4}$$

with a constant reference pressure $p_0 = 10^5\,\mathrm{hPa}$, the gas constant $R = c_p - c_v$ and the specific heats for constant pressure and volume, c_p and c_v. Potential temperature θ is defined by

$$\theta = T \left(\frac{p_0}{p} \right)^{\frac{R}{c_p}} \tag{5}$$

with temperature T. The physical importance of potential temperature is due to its relation with entropy. The logarithm of potential temperature is proportional to the entropy. In our model we use potential temperature as a variable because this simplifies the extension to moist air in future research.

Atmospheric flow is often approximately in hydrostatic balance, which is defined by

$$\frac{\partial p}{\partial z} = -\rho\, g. \tag{6}$$

This balance can produce numerical instabilities because the remaining terms in the vertical component of eq. (2) are much smaller than the two terms of the hydrostatic balance (6). To avoid this instability we introduce the mean states $\bar{p}$, $\bar{\rho}$ and $\bar{\Theta}$ that are in hydrostatic balance. The mean state of pressure p is defined by $\bar{p} = p(\bar{\Theta})$. These mean states are independent of time t and horizontal position x. The deviation of the variables from the mean state is denoted by $\rho' = \rho - \bar{\rho}$, $\Theta' = \Theta - \bar{\Theta}$ and $p' = p - \bar{p}$. These deviations do not have to be small for all times and everywhere in the domain. The physical variables and also the deviations can vary in time and space. Therefore this splitting into the mean state and the deviation does not restrict the possible applications of our numerical model. But for the accuracy and stability of the simulation it is an advantage to choose the mean state in such a way that the deviation remains as small as possible. By this procedure the set of equations (1) – (3) can be written as

$$\frac{\partial \rho'}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{U} = 0, \tag{7}$$

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla} \cdot \left(\frac{\boldsymbol{U} \otimes \boldsymbol{U}}{\rho} + p'\, \mathbf{I}_2 \right) = -\,\rho'\, g\, \hat{\mathbf{k}}, \tag{8}$$

$$\frac{\partial \Theta'}{\partial t} + \boldsymbol{\nabla} \cdot \left(\frac{\Theta\, \boldsymbol{U}}{\rho} \right) = 0. \tag{9}$$

To discretize these equations in space we introduce the commonly used notation

$$\frac{\partial \boldsymbol{q}}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{F}\,(\boldsymbol{q}) = \boldsymbol{S}\,(\boldsymbol{q}), \tag{10}$$

with the vector $\boldsymbol{q} = \left(\rho, \boldsymbol{U}^{\mathrm{T}}, \Theta \right)^{\mathrm{T}}$, the source function

$$\boldsymbol{S}\,(\boldsymbol{q}) = \begin{pmatrix} 0 \\ -\rho'\, g\, \hat{\mathbf{k}} \\ 0 \end{pmatrix}, \tag{11}$$

and the flux tensor

$$\mathbf{F}\,(\boldsymbol{q}) = \begin{pmatrix} \boldsymbol{U} \\ \boldsymbol{U} \otimes \boldsymbol{U}/\rho + p'\, \mathbf{I}_2 \\ \Theta\, \boldsymbol{U}/\rho \end{pmatrix}. \tag{12}$$

Equation (10) is discretized using the discontinuous Galerkin method which we now describe in detail.

3.1. Discontinuous Galerkin Method

In our work we use a nodal discontinuous Galerkin method based on the strong formulation using the Rusanov flux at the cell interfaces. Furthermore, we consider a two dimensional triangular mesh; the extension to a full three dimensional method will remain a task for the future but we envision using tetrahedra for this task. The triangular discontinuous Galerkin method used in our work is described by Giraldo and Warburton [23] for the case of the shallow water equations. Despite a different definition of conserved variables $\boldsymbol{q}$, flux tensor $\mathbf{F}(\boldsymbol{q})$ and source function $\boldsymbol{S}(\boldsymbol{q})$, eq. (10) remains unchanged. Therefore, we repeat in this paper only the main ideas of the discretization.

We start by multiplying eq. (10) with a test function ψ, integrating over an arbitrary element Ω_e and bringing the spatial derivative in front of the test function with integration by parts. Replacing the flux in the boundary terms by a numerical flux $\mathbf{F}^*$ leads to the following equation for the numerical solution $\boldsymbol{q}_N$:

$$\int_{\Omega_e} \left(\frac{\partial \boldsymbol{q}_N}{\partial t} - \mathbf{F}_N \cdot \boldsymbol{\nabla} - \boldsymbol{S}_N \right) \psi(\boldsymbol{x}) \, \mathrm{d}\Omega = - \int_{\Gamma_e} \psi(\boldsymbol{x}) \, \hat{\mathbf{n}} \cdot \mathbf{F}_N^* \mathrm{d}\Gamma, \tag{13}$$

where Γ_e is the boundary of element Ω_e, $\hat{\mathbf{n}}$ is the outward pointing unit normal vector on Γ_e, $\mathbf{F}_N = \mathbf{F}(\boldsymbol{q}_N)$, $\boldsymbol{S}_N = \boldsymbol{S}(\boldsymbol{q}_N)$, $\mathrm{d}\Omega$ is the area element and $\mathrm{d}\Gamma$ is the line element. Applying again integration by parts gives the *strong formulation* [24]

$$\int_{\Omega_e} \left(\frac{\partial \boldsymbol{q}_N}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{F}_N - \boldsymbol{S}_N \right) \psi(\boldsymbol{x}) \, \mathrm{d}\Omega = \int_{\Gamma_e} \psi(\boldsymbol{x}) \, \hat{\mathbf{n}} \cdot (\mathbf{F}_N - \mathbf{F}_N^*) \, \mathrm{d}\Gamma. \tag{14}$$

We construct an Nth degree approximation of the solution $\boldsymbol{q}_N(\boldsymbol{x})$ as follows

$$\boldsymbol{q}_N(\boldsymbol{x}) = \sum_{j=1}^{M_N} \psi_j(\boldsymbol{x}) \, \boldsymbol{q}_j \tag{15}$$

where $\psi(\boldsymbol{x})$ are the basis functions, $\boldsymbol{q}_j$ is the solution at the $j = 1, \ldots, M_N$ gridpoints of each element where $M_N = \frac{1}{2}(N+1)(N+2)$. As in [23] we use Lagrange polynomials for the basis functions ψ with Fekete points [25] for the interpolation points and Gauss points [26] for the integration. With this combination of interpolation and quadrature points the model can use up to polynomial degree 15; however in this paper we always used polynomial degree 3. Using these basis functions we get

$$\frac{\partial \boldsymbol{q}_i}{\partial t} = - \int_{\Omega_e} \hat{\psi}_i \left(\boldsymbol{\nabla} \cdot \mathbf{F}_N - \boldsymbol{S}_N \right) \mathrm{d}\Omega + \int_{\Gamma_e} \hat{\psi}_i \hat{\mathbf{n}} \cdot (\mathbf{F}_N - \mathbf{F}_N^*) \, \mathrm{d}\Gamma, \tag{16}$$

where $\hat{\psi}_i = \sum_{k=1}^{M_N} M_{ik}^{-1} \psi_k$ with the mass matrix $M_{ik} = \int_{\Omega_e} \psi_i \psi_k \mathrm{d}\Omega$; for the sake of simplicity, we did not write the dependence on $\boldsymbol{x}$ of the basis functions although it should be understood that the basis functions depend on the spatial coordinates defined at both the interpolation and integration points. Note that the construction of $\hat{\psi}$ is constructed as a matrix-vector product between the inverse of the mass matrix M^{-1} and the column-vector of basis functions ψ. The inverse of the $M_N \times M_N$ matrix M is constructed via Gauss-Jordan and only needs to be done once (at the beginning of the simulation). Furthermore, if we maintain the same polynomial order N throughout all the elements Ω_e in the domain $\Omega = \bigcup_{e=1}^{N_e} \Omega_e$ and if we insist that the elements have straight edges, then the matrix M^{-1} needs to be calculated for only one canonical element (in the computational space) and then scaled by the Jacobian of the element Ω_e. This allows for a very simple and efficient construction of the key matrices required in an adaptive DG simulation.

The numerical flux function F^* is approximated by the Rusanov flux, given by:

$$\mathbf{F}_N^* = \frac{1}{2} \left[\mathbf{F}(\boldsymbol{q}_N^L) + \mathbf{F}(\boldsymbol{q}_N^R) - \lambda \, \hat{\mathbf{n}} \left(\boldsymbol{q}_N^R - \boldsymbol{q}_N^L \right) \right] \tag{17}$$

with the maximum wave speed $\lambda = ||\boldsymbol{u}||_2 + a$ where $||\boldsymbol{u}||_2 = \sqrt{u^2 + w^2}$ and a is the speed of sound. The superscripts L and R stand for the left and right limiting values at the boundary of the element. If the

normal vector $\hat{\mathbf{n}}$ of element Ω_e is pointing to the right, $\boldsymbol{q}_N^L$ is the left limiting value of $\boldsymbol{q}_N$ and $\boldsymbol{q}_N^R$ is the right limiting value.

So far we have derived a discontinuous Galerkin discretization for our set of equations (10). At this point, the right hand side of eq. (16) is known and we can integrate the equation in time. This can be done either by an explicit or implicit method. For an explicit method we implement a third order Runge-Kutta method of Cockburn and Shu [27]. Because of the fast sound and gravity waves this explicit time-integration is restricted to a very short time-step. As explained before we are not interested in simulating these fast waves accurately; therefore, we also use a semi-implicit time-integrator as presented in the next subsection.

3.2. Semi-Implicit Time Integration

The semi-implicit time integration is implemented in a similar fashion to the approach of Restelli and Giraldo [28, 29]. The main difference is that we use potential temperature instead of total energy as the fourth variable.

The full nonlinear operator $\mathcal{N}(\boldsymbol{q})$ is given in our notation by

$$\mathcal{N}(\boldsymbol{q}) = -\boldsymbol{\nabla} \cdot \mathbf{F}(\boldsymbol{q}) + \boldsymbol{S}(\boldsymbol{q}). \tag{18}$$

For the semi-implicit approach we define an operator $\mathcal{L}$ by

$$\mathcal{L}\boldsymbol{q} = -\begin{pmatrix} \boldsymbol{\nabla} \cdot \boldsymbol{U} \\ \partial p'/\partial x \\ \partial p'/\partial z + g\,\rho' \\ \boldsymbol{\nabla} \cdot \left(\bar{\Theta}\,\boldsymbol{U}/\bar{\rho}\right) \end{pmatrix}, \tag{19}$$

where $p' = p - \bar{p}$ with $p = p(\Theta)$ and $\bar{p} = p(\bar{\Theta})$. The operator $\mathcal{L}$ and the term p' are linear with respect to $\left(\rho', \boldsymbol{U}^{\mathrm{T}}, \Theta'\right)$. As explained by Restelli [30] the operator $\mathcal{L}$ is responsible for the fast moving sound and gravity waves and, therefore, must be integrated implicitly. This splitting is done by writing

$$\frac{\partial \boldsymbol{q}}{\partial t} = \{\mathcal{N}(\boldsymbol{q}) - \mathcal{L}\,\boldsymbol{q}\} + \mathcal{L}\,\boldsymbol{q}. \tag{20}$$

For discretizing (20) in time, we use a backward difference formula of order 2, that leads to

$$\frac{1}{\gamma\,\Delta t}\sum_{m=-1}^{1} \alpha_m \boldsymbol{q}^{n-m} = \sum_{m=0}^{1} \beta_m \left[\mathcal{N}\left(\boldsymbol{q}^{n-m}\right) - \mathcal{L}\,\boldsymbol{q}^{n-m}\right] + \mathcal{L}\,\boldsymbol{q}^{n+1} \tag{21}$$

with $\alpha_{-1} = 1$, $\alpha_0 = 4/3$, $\alpha_1 = -1/3$, $\gamma = 2/3$, $\beta_0 = 2$, $\beta_1 = -1$ and Δt is the time step. We rewrite this equation collecting all terms with $\boldsymbol{q}^{n+1}$ and get

$$\left[\mathbf{I} - \gamma\,\Delta t\,\mathcal{L}\right]\,\boldsymbol{q}^{n+1} = \tilde{\boldsymbol{q}}^{\mathrm{ex}} - \gamma\,\Delta t \sum_{m=0}^{1} \beta_m \mathcal{L}\,\boldsymbol{q}^{n-m}, \tag{22}$$

where

$$\tilde{\boldsymbol{q}}^{\mathrm{ex}} = -\sum_{m=0}^{1} \alpha_m \boldsymbol{q}^{n-m} + \gamma\,\Delta t \sum_{m=0}^{1} \beta_m \mathcal{N}\left(\boldsymbol{q}^{n-m}\right) \tag{23}$$

is an explicit predictor that has to be calculated first. The identity matrix $\mathbf{I}$ on the left hand side of eq. (22) comes from the coefficient α_{-1}. Solving the linear system of equations (22) (e.g., with GMRES) gives the implicit corrector. For details on this solution strategy the reader is referred to [29] and [31].

The process of solving the linear system of equations can be improved by using preconditioners. Currently we just use the simple Jacobi preconditioner. For this preconditioner we found, that the simulation using semi-implicit time-integration is still most efficient when using a time step which is about twice as large as for the explicit simulation. The time step could be much larger but with this simple preconditioner a larger time step takes more CPU time. We are working on developing better preconditioners [32]. We expect to be able to reduce the CPU time based on future research on preconditioners.

3.3. Mesh Refinement with Space Filling Curve Approach

As explained in section 2 we expect steep gradients at the boundary of the cumulus clouds for which our numerical model is designed. For being able to increase the spatial resolution in these regions we use h-adaptive mesh refinement. H-adaptive mesh refinement means that spatial resolution is adapted by dividing elements into smaller elements (refinement) or collecting elements into larger elements (coarsening). In our code this is managed with the function library AMATOS [11]. The main advantage of this function library is that it handles the entire h-adaptive mesh refinement. Furthermore it leads to an order of unknowns that preserves data adjacency and linearizes data access in computer memory by using a so-called space filling curve approach. For further information we refer to the publication of Behrens et al. [11].

The only modification that was necessary for our work was the calculation of the new values at the grid points when elements are refined or coarsened. This is quite straight forward. For refinement we simply evaluate the old polynomials at the positions of the new degrees of freedom. For coarsening we use a different approach because we want to conserve mass. Therefore we make a coordinate transformation to modal basis functions and use the average values of the modal coefficients on the small child elements for the new values of the mother element.

The refinement criterion used for the results presented in this paper is given by:

$$|\theta'(\boldsymbol{x},t)| \geq \sigma \max_{\boldsymbol{x}} \left(|\theta'(\boldsymbol{x},t)|\right), \tag{24}$$

with the deviation of the potential temperature from the background state denoted by $\theta' = \theta - \bar{\Theta}/\bar{\rho}$ and a user-specified and problem dependent constant σ. For the density current of Straka et al. [33] we use $\sigma = 0.05$. In all the other results shown in this paper we used $\sigma = 0.1$. Wherever this condition (24) is fulfilled the mesh is refined until it reaches a specified finest resolution. In the rest of the domain the grid is coarsened until it reaches a specified coarsest resolution without modifying the resolution in the refinement region. The transition between fine and coarse meshes is given by the conformity of the grid. After each time-step we calculate the number of elements that have to be changed for grid refinement. If more than 1% of all elements have to be changed, the grid is adapted. Otherwise the grid remains unchanged.

To avoid small scale structures moving into a region with a coarse mesh we add a few rows of fine elements to the refinement region. We do not know a priori which size of the refinement region is best. For the validation in section 4 we choose the size of this buffer zone as we expect it to be the best. For getting a more objective instruction on how to choose the size of the refinement region we develop a new method for testing refinement criteria in section 5. In that section we will also compare the results for different sizes of this buffer zone.

The time-step is determined by the smallest grid spacing. So far we do not have any sub-cycling implemented, because we expect that for the simulation of cumulus clouds the majority of elements will be very small. For this reason we do not expect much benefit by using sub-cycling in these applications.

The refinement criterion (24) should not be used for every application. For the simple test cases shown in the rest of this paper we will see that it works very well. However, for more realistic simulations, we intend to use better refinement criteria, based on physical parameters, gradients in fluid components and mathematical error indicators.

3.4. Artificial viscosity

For simulating the density current test case of Straka et al. [33] we implemented the artificial viscosity that is used in [33]. A diffusion term $\boldsymbol{\nabla} \cdot (\mu\, \rho\, \boldsymbol{\nabla} \boldsymbol{u})$ is added to the right hand side of eq. (2) and the term $\boldsymbol{\nabla} \cdot (\mu\, \rho\, \boldsymbol{\nabla} \theta')$ is added to the right hand side of eq. (3), with μ the viscosity parameter. We use the deviation of potential temperature θ' in the viscosity term because we do not want viscosity to modify the hydrostatic basic state.

In addition to the use of artificial viscosity for simulating the test case of Straka et al. we use the diffusion terms with a non-constant viscosity parameter μ_{lim} as a slope limiter for avoiding the Gibbs phenomenon. The viscosity parameter μ_{lim} is still constant within each element of our grid but we allow μ_{lim} to change between different elements. The total viscosity parameter for each element e is given by

$$\mu_e = \max\left(\mu_{\text{tc}}, \mu_{\text{lim},e}\right), \tag{25}$$

where μ_{tc} is the viscosity parameter given by the test case (e.g. $\mu_{\text{tc}} = 75\text{m}^2/\text{s}$ for the density current of Straka et al. [33]). For the slope limiting viscosity parameter $\mu_{\text{lim},e}$ we introduced the following formula (see Appendix A):

$$\mu_{\text{lim},e} = \mu_{\text{ref}} \left(\frac{\Delta\theta'_e}{\alpha\, \Delta\theta'_0} \right)^{\kappa} \frac{\Delta x_{\text{eff}}}{\Delta x_{\text{ref}}} \frac{v_{\text{max}}}{v_{\text{ref}}}, \tag{26}$$

where $\Delta\theta'_e$ is the difference between the maximum and minimum of θ' in element e, $\Delta\theta'_0$ is the difference between the maximum and minimum of θ' at time $t = 0\text{s}$ taken over the whole domain, Δx_{eff} is the spatial resolution, v_{max} is the approximate maximum wind speed throughout the whole simulation, and μ_{ref}, α, κ, Δx_{ref} and v_{ref} are fixed parameters. We tested different choices of parameters and found that $\mu_{\text{ref}} = 0.1\,\text{m}^2/\text{s}$, $\alpha = 0.4$, $\kappa = 1$, $\Delta x_{\text{ref}} = 3.12\,\text{m}$ and $v_{\text{ref}} = 3\,\text{m/s}$ give us the best compromise between damping of the Gibbs phenomenon and preserving the amplitude of the flow for all test cases considered in this paper. The maximum wind speed v_{max} was estimated by making high-resolution simulations of the test cases. For the density current test case from Straka et al. [33] we found $v_{\text{max}} \approx 40\,\text{m/s}$. The other test cases presented in this paper show a maximum wind speed of about $v_{\text{max}} \approx 3\,\text{m/s}$. In order to avoid steep gradients moving into elements with a low viscosity coefficient we compute μ_{lim} according to (26) and take the maximum value of all neighboring elements. For the high-resolution run for the test case of Giraldo and Restelli in figure 6d we found that the maximum should be taken over two rows of neighboring elements.

This approach does not remove the Gibbs phenomenon completely. For dry inviscid atmospheric convection potential temperature is conserved. This allows us to filter most of the Gibbs phenomenon by using the following cutoff-filter $\mathcal{F}$ at any of our degrees of freedom i:

$$\mathcal{F}(\theta_i) = \begin{cases} \theta_{\text{max},0}, & \text{if } \theta_i > \theta_{\text{max},0}, \\ \theta_i, & \text{if } \theta_{\text{max},0} \geq \theta_i \geq \theta_{\text{min},0}, \\ \theta_{\text{min},0}, & \text{if } \theta_i < \theta_{\text{min},0}, \end{cases} \tag{27}$$

where $\theta_{\text{min},0}$ and $\theta_{\text{max},0}$ are the global minimum and maximum of potential temperature θ at time $t = 0\text{s}$. For the test cases shown in this paper we will see that this simple filter works very well in combination with artificial viscosity. Nevertheless we are working on implementing and testing other limiting techniques.

The second order derivatives produced by the diffusion terms are discretized by using a local discontinuous Galerkin method [34]. That means that the order of the derivatives is reduced by introducing the following new variables

$$\boldsymbol{\alpha} = \boldsymbol{\nabla} u, \tag{28}$$
$$\boldsymbol{\beta} = \boldsymbol{\nabla} w, \tag{29}$$
$$\boldsymbol{\gamma} = \boldsymbol{\nabla} \theta'. \tag{30}$$

Similar to eq. (16) we get

$$\boldsymbol{\alpha}_i = -\int_{\Omega_e} \hat{\psi}_i\, u_j \boldsymbol{\nabla} \psi_j \mathrm{d}\Omega + \int_{\Gamma_e} \hat{\psi}_i \psi_j \hat{\mathbf{n}} \left(u_j - u_j^* \right) \mathrm{d}\Gamma, \tag{31}$$

$$\boldsymbol{\beta}_i = -\int_{\Omega_e} \hat{\psi}_i\, w_j \boldsymbol{\nabla} \psi_j \mathrm{d}\Omega + \int_{\Gamma_e} \hat{\psi}_i \psi_j \hat{\mathbf{n}} \left(w_j - w_j^* \right) \mathrm{d}\Gamma, \tag{32}$$

$$\boldsymbol{\gamma}_i = -\int_{\Omega_e} \hat{\psi}_i\, \theta'_j \boldsymbol{\nabla} \psi_j \mathrm{d}\Omega + \int_{\Gamma_e} \hat{\psi}_i \psi_j \hat{\mathbf{n}} \left(\theta'_j - \theta'^*_j \right) \mathrm{d}\Gamma. \tag{33}$$

As these viscosity terms do not describe a flow in a certain direction (as in the case of the advection terms) we use the following numerical flux for the viscosity terms in $\mathbf{F}_N^*$:

$$Q^* = \frac{1}{2} \left(Q^R + Q^L \right) \tag{34}$$

where Q represents either u, w or θ'. Note that this is not the only possibility for discretizing the second order operators; for other choices see Shahbazi et al. [35].

	adaptive	uniform
Δt	0.020 s	0.020 s
Δx_{eff}	4.42 m	4.42 m
L	15.62 m	15.62 m
#elements	2833.59	8192
total time	10857.04 s	32804.87 s
grid time	721.71 s	0.16 s
$\max(\theta')$	0.50 K	0.50 K
$\min(\theta')$	−0.05 K	−0.04 K
figure	2	—

Table 1: Details for our simulations of a bubble test case by Robert (1993) [36]. The variables shown in this table are: the time step Δt, the finest effective resolution Δx_{eff}, the length of the shortest element edge L, the average number of elements for the whole simulation, the total CPU time, the CPU time spent for initializing and adapting the grid ("grid time"), the maximum and minimum of θ' at the end of the simulation ($t = 600$s) and the figure where the simulation is shown. The number of elements is an average value from time $t = 0$s to end time $t = 600$s. The CPU times give the time until the end of the simulation at $t = 600$s is reached. All simulations use semi-implicit time-integration and are computed on the same single Linux CPU.

4. Validation

For the validation of our numerical model we considered three test cases that are relevant for atmospheric convection. These test cases are a small cold air bubble on top of a large warm air bubble from Robert [36], a density current from Straka et al. [33], and a smooth warm air bubble from Giraldo and Restelli [22]. In all of these cases there is no exact solution, but we can compare our results with those from the literature.

The results of adaptive simulations of the different test cases are shown in figures 2–6. Some more details of the simulations are given in tables 1–3. As a measure of the spatial resolution we use the average distance between neighboring Fekete points. We call this value the effective resolution Δx_{eff}. It should be a good measure for the smallest scale that can be present in our numerical model. However this does not imply that other numerical methods use exactly the same number of unknowns. As discontinuities between elements are allowed for DG multiple values occur at the interfaces between elements and increase the number of unknowns.

The tables show also the details of uniform simulations using the finest spatial resolution of the corresponding adaptive computation. The values suggest that adaptive simulations are much faster compared to the uniform simulations. However at this point it is not completely clear how large the refinement region should be. For a more reliable comparison between adaptive and uniform simulations we will develop an objective criterion for choosing the size of the refinement region in section 5.

4.1. Small Cold Air Bubble on Top of Large Warm Air Bubble

The first test case is a small cold air bubble on top of a large warm air bubble in a domain of 1km×1km. This test case was introduced by Robert [36] in 1993. The background state has a constant potential temperature of $\bar{\theta} = 303.15\,\text{K}$. Both bubbles have a Gaussian profile in θ'. The warm air bubble has an amplitude of 0.5 K, the amplitude of the cold air bubble is $-0.15\,\text{K}$. The initial conditions are chosen identically to those of Robert [36]. However, we use a slightly different resolution. This is necessary because in our case the domain has to be divided into a hierarchy of triangles for adaptive grid refinement. Therefore the resolution can only be changed by a factor of $\sqrt{2}$. In this test case we use a resolution of $\Delta x_{\text{eff}} = 4.4$m (table 1) which is slightly smaller than 5m of Robert [36].

Figure 2 shows our result for this test case. The mesh is continuously adapted to the position of the temperature anomaly. Correspondingly the fine mesh follows the bubbles very nicely. As mentioned before we use polynomials of degree three for all results shown in this paper.

By comparing our result with the corresponding figure of Robert [36] one can see that the results agree very well. After 600s the position and shape of the warm air is still almost identical to the corresponding

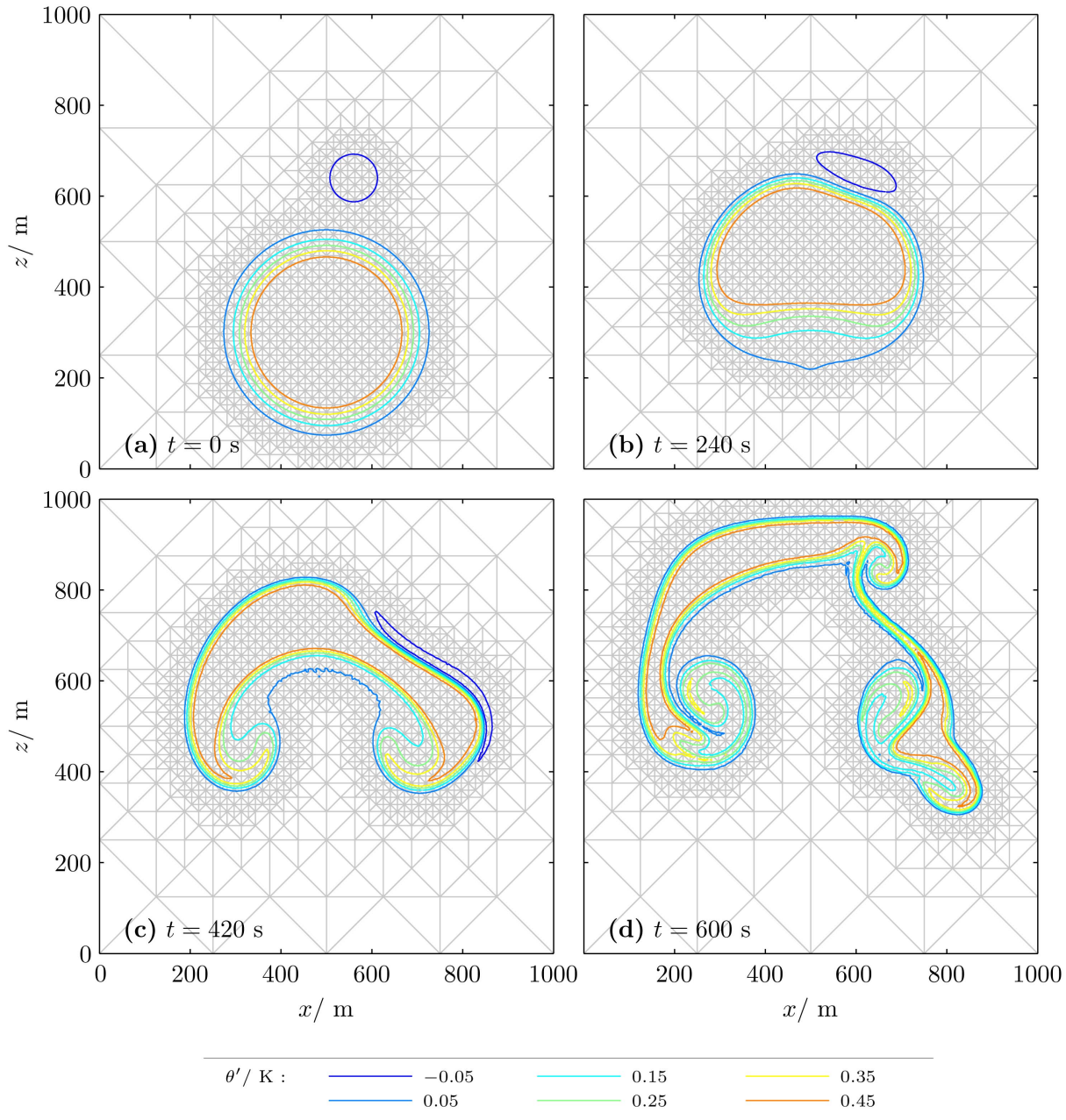

Figure 2: Small cold air bubble on top of a large warm air bubble as introduced by Robert [36]. The contour lines show the deviation of the potential temperature from the background state and the gray lines show the adaptively refined triangular mesh used in our simulation. We used the same contour values as in [36] for making it easier to compare our figure with the one of Robert. The contour values are from -0.05 K to 0.45 K with an interval of 0.1 K. For the time-integration we use the semi-implicit method for all simulations shown in this paper. We also tested explicit time-integration which produces the same results.

	Straka 1	Straka 2	Straka 3	Straka 4	Straka 5	Straka 6
adaptive	yes	no	yes	no	yes	no
Δt	0.100 s	0.100 s	1.000 s	1.000 s	2.000 s	2.000 s
Δx	28.26 m	28.26 m	226.08 m	226.08 m	452.16 m	452.16 m
L	100.00 m	100.00 m	800.00 m	800.00 m	1600.00 m	1600.00 m
#elements	3176.62	32768	151.22	512	64.78	128
total time	4366.20 s	26914.68 s	23.36 s	61.77 s	4.87 s	8.17 s
grid time	325.45 s	0.44 s	1.00 s	0.01 s	0.20 s	0.00 s
$\max(\theta')$	0.00 K	0.00 K	0.00 K	0.00 K	0.00 K	0.00 K
$\min(\theta')$	−9.84 K	−9.81 K	−9.14 K	−9.37 K	−7.25 K	−7.15 K
front position:	15477.90 m	15467.63 m	15106.56 m	15126.06 m	14525.47 m	14547.48 m
figure	3	—	4	—	—	—

Table 2: Details for our simulations of the density current test case by Straka et al. (1993) [33]. The different columns represent different setups of spatial resolution and adaptivity. The variables shown in this table are described in the caption of table 1. The end time of the simulation is $t = 900$s (instead of $t = 600$s in table 1). Additionally the horizontal position of the density current front at time $t = 900$s (given by the −1K contour) is denoted.

plot of Robert [36]. Even the smaller vortices on the right hand side of the domain are very similar to the result of Robert.

4.2. Density Current

A second test case is a density current initialized by a cold air bubble with a cosine profile and an amplitude of 16.624 K in θ' (figure 3). This test case was introduced by Straka et al. [33]. The viscosity of $\mu_{\text{tc}} = 75\text{m}^2/\text{s}$ is identical to the setup of Straka et al. [33].

As described for the previous test case we are not able to use exactly the same resolution as in the literature. For the high-resolution run shown in figure 3 we use $\Delta x_{\text{eff}} = 28.26$m which is slightly larger than the reference resolution of Straka et al. with 25m. Again we see no differences between our result and the result in the literature. The position of the density current and the shape of the Kelvin-Helmholtz rotors agrees very well with the result of Straka et al. [33] throughout the whole simulation. As given in table 2 the horizontal location of the front at time $t = 900$s is $x_{\text{front}} = 15477.90$m while the reference simulation of Straka et al. gives a position of $x_{\text{front}}^{\text{Straka}} = 15537.44$m.

Figure 4 shows the corresponding result for a much coarser resolution of 226.08m. The position of the front $x_{\text{front}} = 15106.56$m and the amplitude of the density current of -9.14 K agree fairly well with the highly-resolved simulations. Compared to the numerical models presented in [33] we think this result reaches at least the average quality of the different results presented in that publication.

When decreasing our slope limiter parameter μ_{ref} the front position and amplitude of the density current get even closer to the high-resolution result in figure 3. If we set $\mu_{\text{ref}} = 0.05\,\text{m}^2/\text{s}$ instead of $\mu_{\text{ref}} = 0.1\,\text{m}^2/\text{s}$ we get the result shown in figure 5. In this result the position of the front $x_{\text{front}} = 15423.5$m and the amplitude of the density current 10.49 K are very close to the high-resolution result. The noise in figure 5 is slightly more pronounced than in figure 4, but it is still very reasonable. This demonstrates that small changes in μ_{ref} could be used for adapting the simulation to the degree of grid noise that can be allowed in a certain application. Nevertheless our results demonstrate that even a fixed value of μ_{ref} produces satisfying results in all test cases.

4.3. Smooth Warm Air Bubble

As a third test case we computed the rising thermal bubble introduced by Giraldo and Restelli [22] (test case 2). It is a single warm air bubble with a cosine profile in θ'. As in the test case in section 4.1 the domain has an extent of 1km in each direction and the bubble has an amplitude of 0.5 K. We use the same parameters as in the publication of Giraldo and Restelli [22] except for the slightly different resolutions as given in table 3. Furthermore we replaced the Boyd-Vandeven type filter of Giraldo and Restelli [22] with our artificial viscosity based slope limiter.

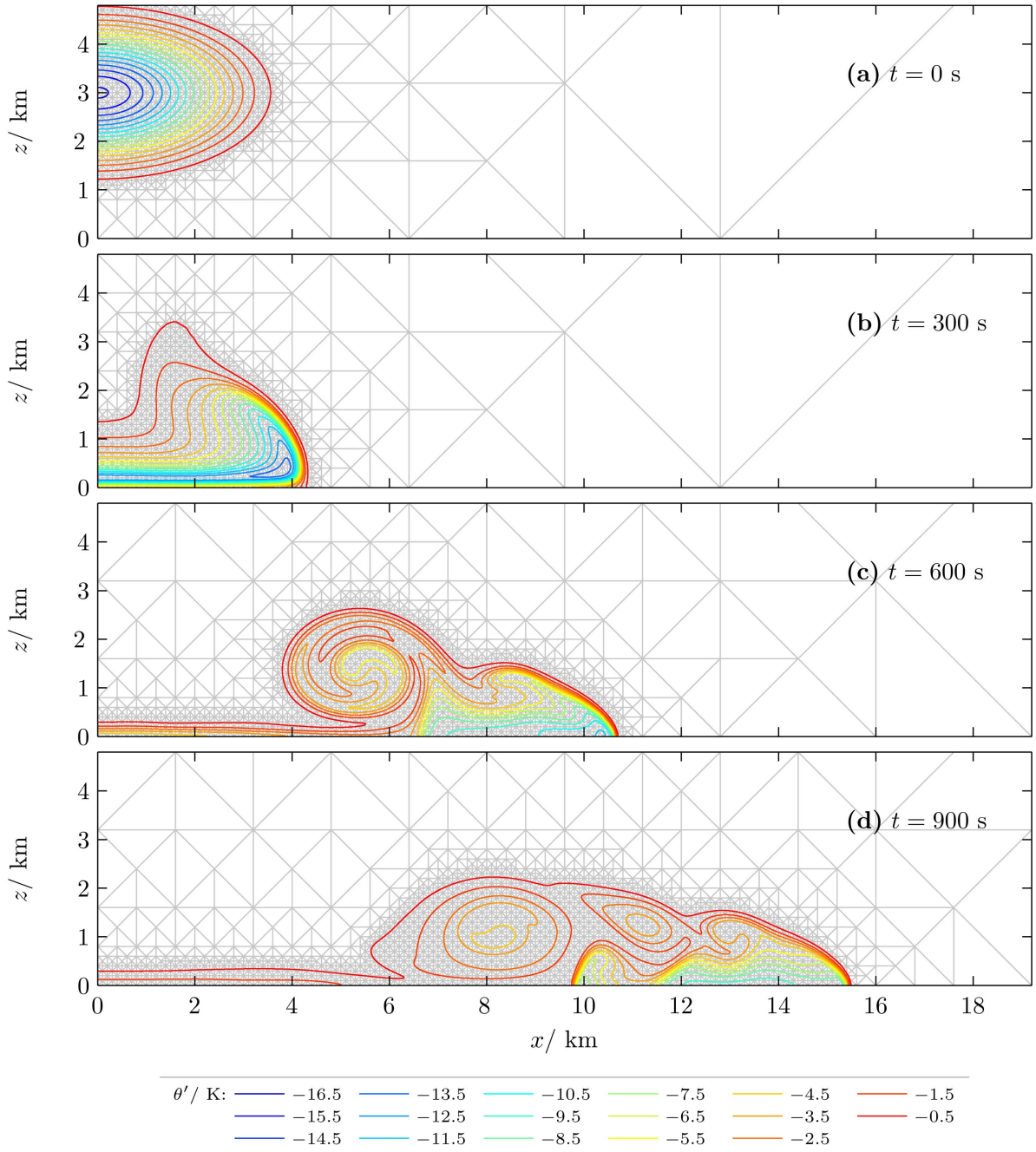

Figure 3: Density current as introduced by Straka et al. [33]. As in figure 2 the contour lines show the deviation θ' of the potential temperature θ from the background state $\bar{\theta}$ and gray lines show the adaptively refined triangular mesh. For making it easier to compare our results with those in [33] we use the same contour values, which are from -16.5 K to -0.5 K with an interval of 1 K.

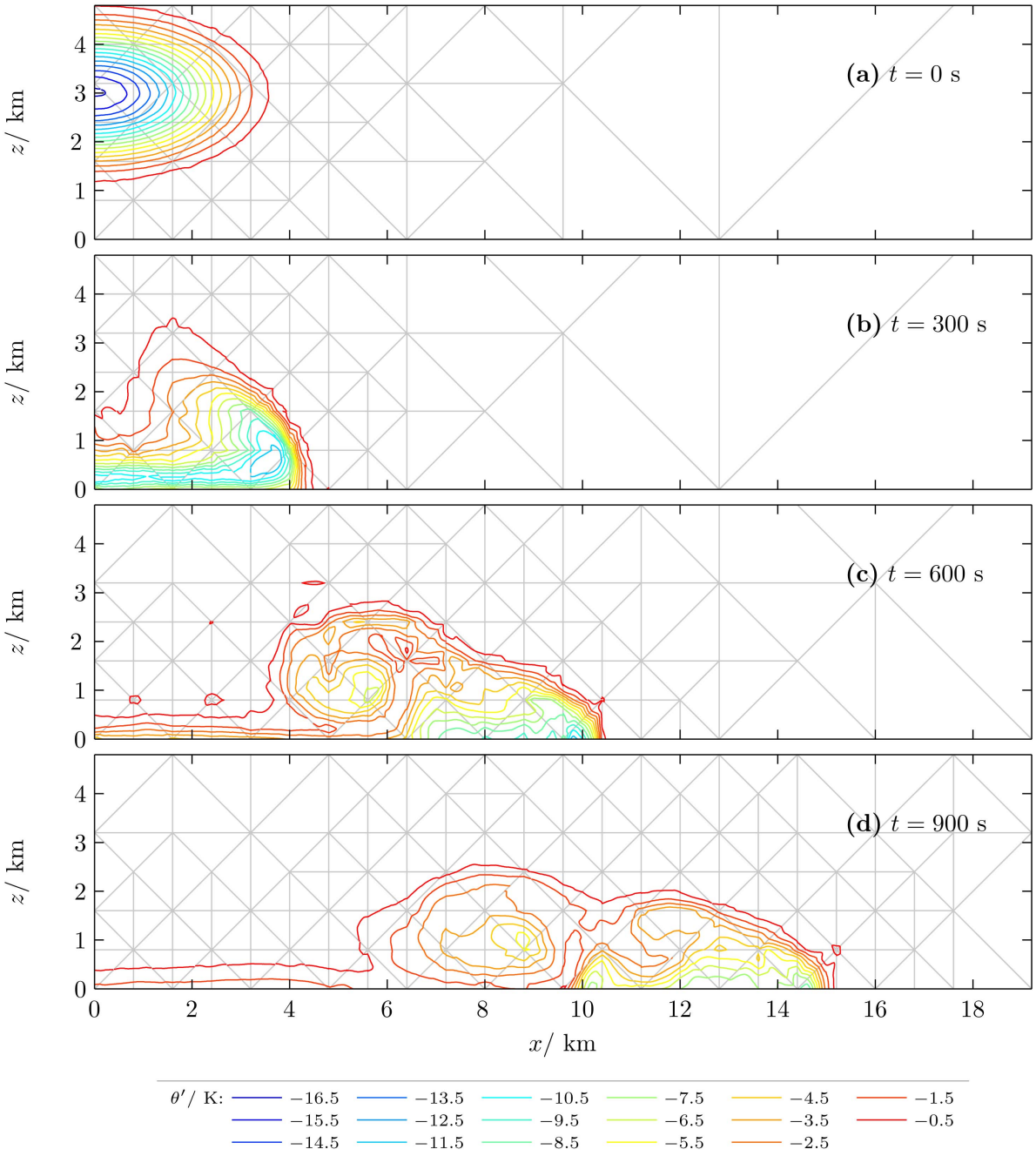

Figure 4: Density current as in figure 3 for an effective resolution of 226.08m.

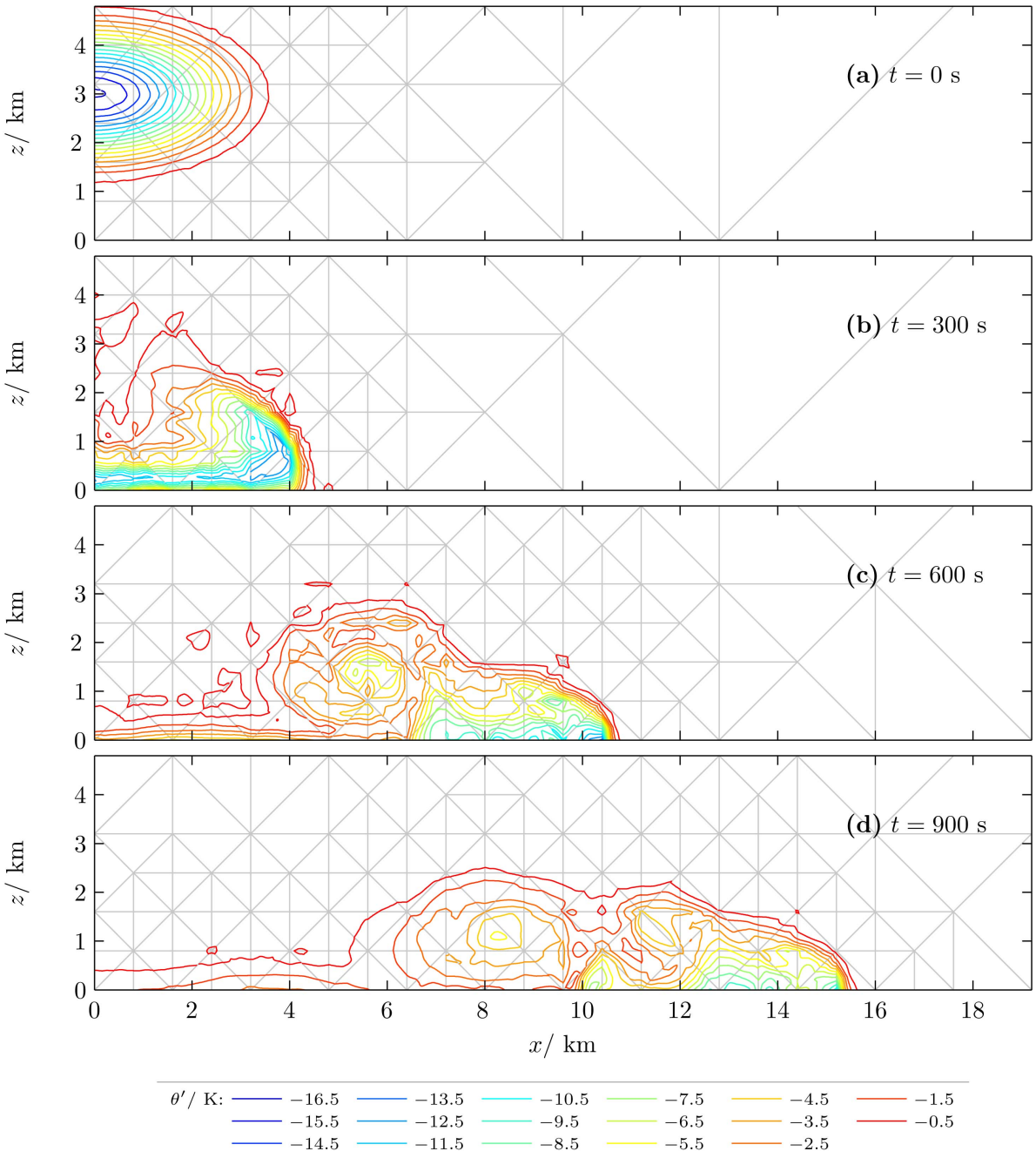

Figure 5: Density current as in figure 4 for a modified slope limiter parameter of $\mu_{\text{ref}} = 0.05\,\text{m}^2/\text{s}$.

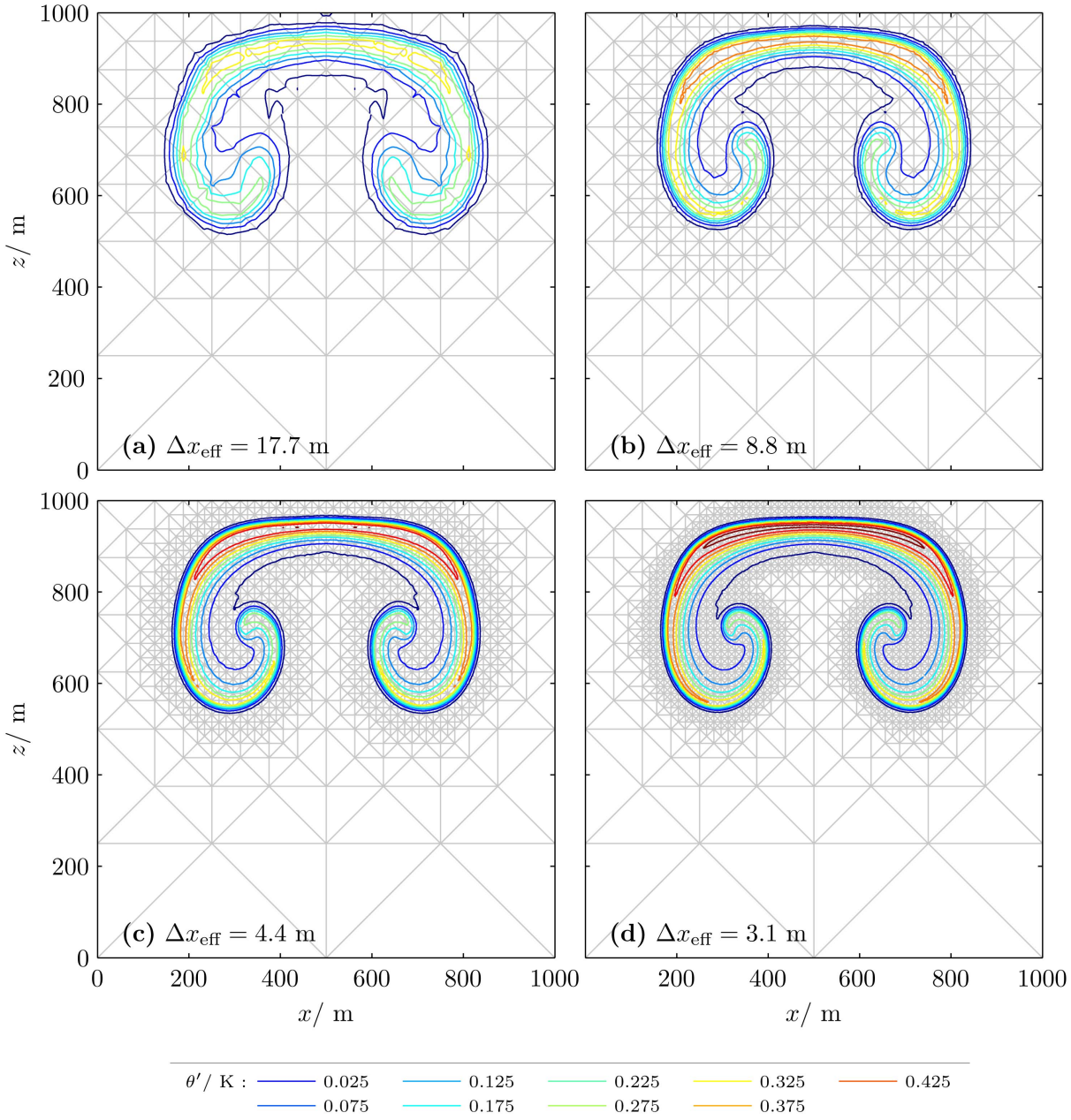

Figure 6: Rising thermal bubble introduced by Giraldo and Restelli [22] at time $t = 700$s. As in figure 2 the contour lines show the deviation θ' of the potential temperature θ from the background state $\bar{\theta}$ and gray lines show the adaptively refined triangular mesh. Contour values are from 0.025 K to 0.425 K with an interval of 0.05 K.

	Giraldo 1	Giraldo 2	Giraldo 3	Giraldo 4	Giraldo 5	Giraldo 6	Giraldo 7	Giraldo 8
adaptive	yes	no	yes	no	yes	no	yes	no
Δt	0.050 s	0.050 s	0.030 s	0.030 s	0.020 s	0.020 s	0.010 s	0.010 s
Δx_{eff}	17.66 m	17.66 m	8.83 m	8.83 m	4.42 m	4.42 m	3.12 m	3.12 m
L	62.50 m	62.50 m	31.25 m	31.25 m	15.62 m	15.62 m	11.05 m	11.05 m
#elements	269.17	512	843.42	2048	2157.05	8192	3234.37	16384
total time	300.69 s	579.18 s	1804.15 s	4209.46 s	9134.92 s	33010.70 s	22378.92 s	106093.58 s
grid time	21.19 s	0.02 s	150.53 s	0.06 s	618.30 s	0.13 s	1869.66 s	0.31 s
$\max(\theta')$	0.32 K	0.32 K	0.36 K	0.36 K	0.42 K	0.42 K	0.45 K	0.45 K
$\min(\theta')$	0.00 K	0.00 K	0.00 K	0.00 K	0.00 K	0.00 K	0.00 K	0.00 K
figure	6a	—	6b	—	6c	—	6d	—

Table 3: Details for our simulations of the warm air bubble test case by Giraldo and Restelli (2008) [22]. The different columns represent different setups of spatial resolution and adaptivity. The variables shown in this table are described in the caption of table 1. The end time of the simulation is $t = 700$s (instead of $t = 600$s in table 1).

As in the previous test cases there are no obvious differences between our results (figure 6) and those from the literature. The position and shape of the bubble seems to be identical to the result of Giraldo and Restelli [22]. This gives us confidence that our code is free of fundamental errors.

5. Testing Refinement Criteria

So far we used adaptive mesh refinement without exactly knowing how large the refinement region should be. In this section we will develop a new method for testing refinement criteria. Furthermore we apply this approach for choosing the size of the refinement region of the warm air bubble test case from section 4.3 and use it for an objective comparison of the CPU time between adaptive and uniform simulations.

5.1. The Qualitative Criterion

Our goal is to make the adaptive simulation as efficient as possible while still producing approximately the same accuracy as a uniform simulation that uses the finest resolution of the adaptive computation. For this purpose we need some kind of error measure for deciding how accurate the different results are. Furthermore we want to test the refinement criterion in a situation that is similar to the application it is designed for. For modeling cumulus clouds the warm air bubble from section 4.3 appears to be a well suited test case. As explained before no exact solution for the warm air bubble is known. Hence it is impossible to calculate the error by comparing numerical and exact solution. Furthermore the exact solution cannot be estimated by using an extremely high spatial resolution: for increasing numerical resolution the results will keep changing as more and more turbulence is resolved. Instead we look for a qualitative measure of the numerical error. For illustration we consider the rising thermal bubble of Giraldo and Restelli [22] as in section 4.3. But this time we continue the simulation much longer until numerical errors become clearly visible. For keeping the result as simple as possible a constant artificial viscosity of $\mu_{\text{tc}} = 0.1\text{m}^2/\text{s}$ avoids the occurrence of more small scale vortices with increasing resolution. As for the density current of Straka et al. [33] we consider this artificial viscosity to be part of our test problem. After 1000 seconds we get the results shown in figure 7 for three different resolutions. Apparently the results have not yet converged: the vortices at the bottom left edge of the bubble depend strongly on the numerical resolution.

This is not surprising. A similar situation can be seen in the case of shear-instability of a horizontal shear-flow. The exact solution of a perfectly horizontal shear-flow is just stationary [37]. A perfectly horizontal shear-flow will remain perfectly horizontal as long as there is no perturbation. Some kind of perturbation is necessary to make the instability of the flow appear and grow. In a similar way a perfectly circular warm air bubble in an infinitely large domain should never develop the instabilities which eventually lead to the small vortices visible in figure 8. But even a tiny perturbation will grow in time and turn into a vortex. In our numerical simulations the perturbations are produced by numerical errors and by the boundaries of

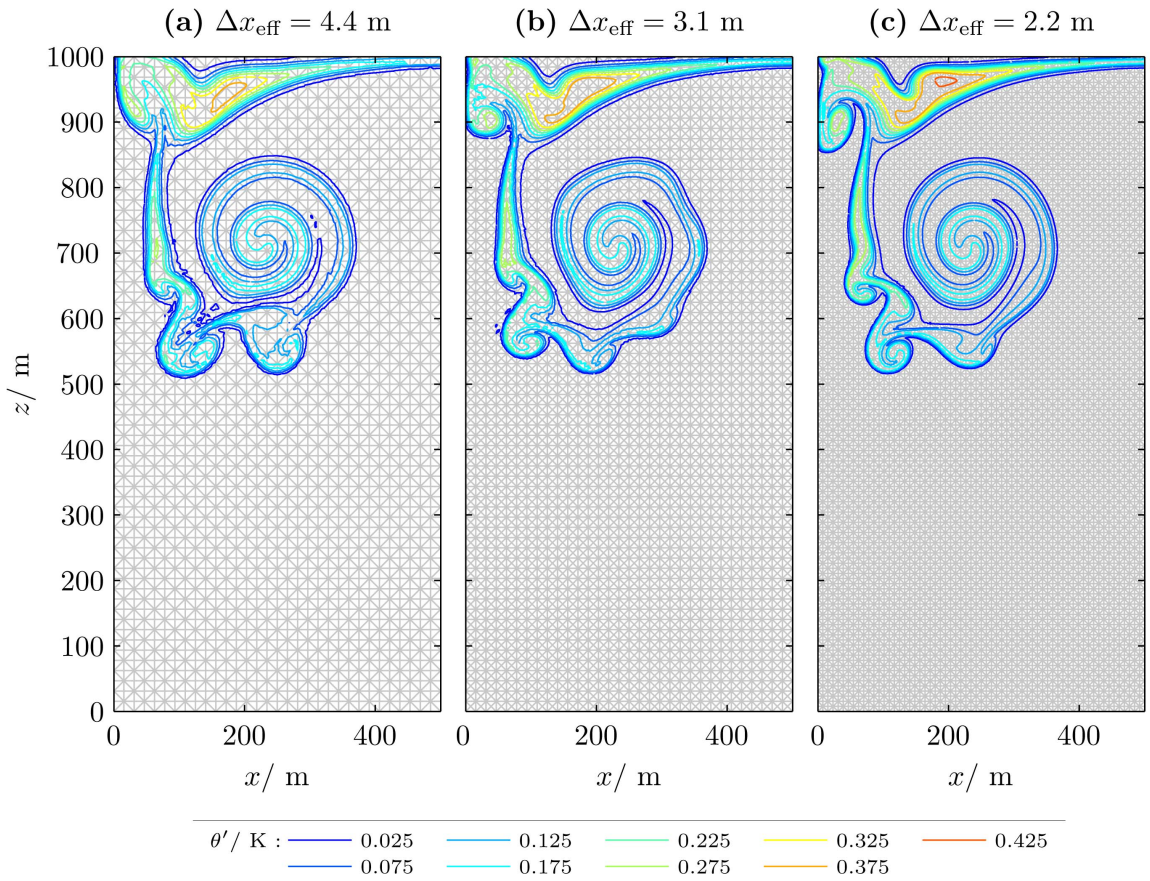

Figure 7: Rising thermal bubble as in figure 6 for three different resolutions with a constant artificial viscosity of $\mu_{\text{tc}} = 0.1\text{m}^2/\text{s}$ at time $t = 1000$s. Simulated is only the left half of the bubble. As in the previous figure the contours indicate the potential temperature deviation from the background state. The adaptively refined triangular mesh is shown by the gray lines. The numerical resolution is given by the average distance between neighboring degrees of freedom. The time step of the semi-implicit time-integration is $\Delta t = 0.01$ s for the two coarser resolutions (a,b) and $\Delta t = 0.005$ s for the highest resolution (c). The remaining parameters of the rising bubble are identical to those in figure 6.

the domain. Some vortices are mainly produced by the boundaries. An example for such a vortex is the one at the top left corner of the domain. This vortex shows little sensitivity to numerical resolution and hence to the accuracy of the solution. Other vortices are dominated by numerical errors as the ones at the bottom left edge of the bubble. We focus our attention on these vortices that are strongly sensitive to the numerical resolution. With the help of these vortices we can compare the accuracy of different simulations. Our method for testing refinement criteria is the following: we consider a refinement criterion to be good if the small scale vortices at the bottom left edge of the rising bubble are similar to those occurring in a simulation using the finest resolution in a uniform grid even when these vortices are fully developed. So far this method seems to be restricted to testing refinement criteria in a qualitative way. However we will see in the next subsection that for a fixed spatial resolution and a certain model setup we can use this method even in a quantitative way.

Note that this approach is not suitable for comparing the accuracy of different numerical models. This is because the onset of instabilities is, amongst others, sensitive to implicit numerical diffusion. The larger the numerical diffusion the later the small scale vortices occur. A numerical model with strong implicit numerical diffusion shows less small scale vortices but is no more accurate. Therefore the explicit artificial viscosity should be large compared to possible differences in the implicit numerical diffusion of the simulations that are compared.

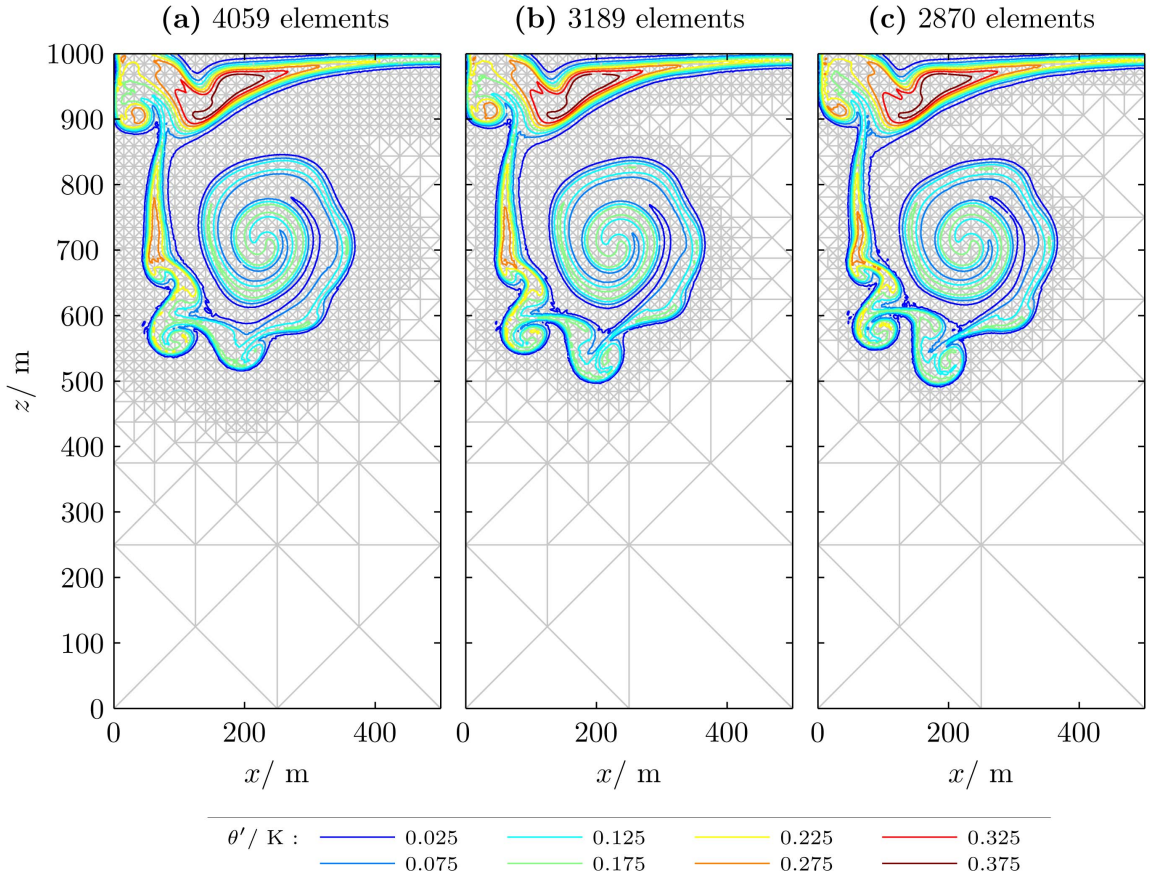

Figure 8: Adaptive simulation for three different refinement regions after 1000 seconds with a finest resolution of $\Delta x_{\text{eff}} = 3.1$ m which is identical to the uniform resolution in figure 7b. The refinement criterion is always given by (24), but the number of additional fine elements surrounding this region is varied. The time step of the semi-implicit time-integration is $\Delta t = 0.01$ s. The number of elements given in the captions of the subfigures is taken at time $t = 1000$s. This is different from the average number of elements given in table 1–4.

Some readers might think that we should initialize vortices with predefined noise as in [12]. However we do not want to study the evolution of those perturbations. We want to detect the degree of numerical noise present in the simulation. Initial perturbations produce vortices which make it more difficult to distinguish between these physically correct vortices and those produced by numerical errors. For testing refinement criteria the simulation result should be as simple as possible while still being sensitive to numerical errors. For this reason we do not use initial perturbations and use a constant amount of artificial viscosity.

5.2. Size of Refinement Region

With the method introduced in the previous subsection we can now ask the following question: what size refinement region will produce the same accuracy as a uniform simulation using the finest spatial resolution of the adaptive computation? To answer this question we compared the results at time $t = 1000$s between different size refinement regions. All simulations share the same finest effective resolution of $\Delta x_{\text{eff}} = 3.12$m. Three results are shown in figure 8. At first glance all results seem to be very similar. However the size of the lowest vortex at $x = 200$m and $z \approx 500$m does change between the different simulations. As explained in the previous subsection we expect that the (unknown) exact solution does not feature these small scale vortices. This allows us to use the following quantity as a measure for the additional errors introduced by

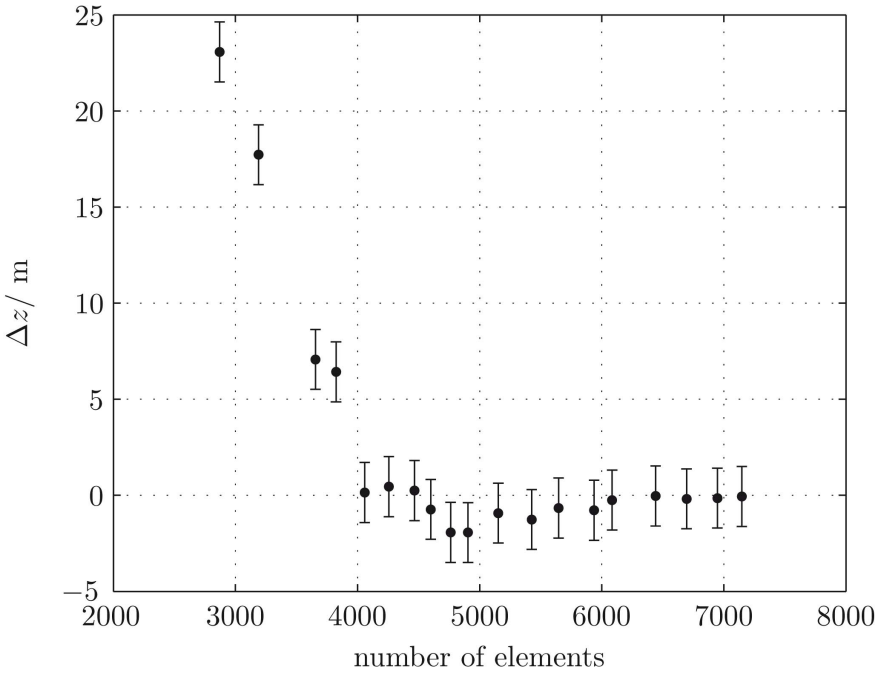

Figure 9: Difference Δz between the minimum height of the 0.025K contour in a uniform simulation z_{uniform} and the corresponding height z_{min} in adaptive simulations that use the resolution of the uniform simulation for their finest resolution. As in figure 8 the different simulations differ in the number of fine elements that are added to the refinement region. The number of elements at time $t = 1000$s is given on the horizontal axis. The uniform simulation uses 8192 elements. We do not expect that differences in Δz that are smaller than the effective resolution of the simulations are reliable. For this reason we added error bars with a length equal to the effective resolution of 3.12m. This uncertainty indicates that the negative values of Δz are probably not significant.

coarsening the mesh:

$$\Delta z := z_{\text{uniform}} - z_{\text{min}} \tag{35}$$

where z_{min} is the minimum height of the 0.025K contour in the adaptive simulation and z_{uniform} is the corresponding height in a uniform simulation that uses the finest resolution of the adaptive simulations. This measure is plotted in figure 9 as a function of the number of elements. Error bars are used in this figure for indicating that we do not expect differences in Δz smaller than the effective resolution of 3.12m to be reliable. This figure shows that for this model setup about 4000 elements are necessary for producing approximately the same results as in the corresponding uniform simulation with 8192 elements. For this reason we expect the simulation in figure 8a to be the fastest adaptive simulation in which the additional errors introduced by coarsening the mesh can be neglected.

As described before we do not expect the (unknown) exact solution to show small scale vortices. For this reason we estimated the correct minimum height of the 0.025K contour z_{ref} to be at about 545m. Our uniform simulation gives a height of $z_{\text{uniform}} = 513.13$m. This suggests that the relative additional error $\Delta z/(z_{\text{ref}} - z_{\text{uniform}})$ is in the order of a few percent when more than 4000 elements are used.

The results of this subsection are not completely surprising: in the simulations with more than 4000 elements at $t = 1000$s the finest resolution exceeds the initial temperature anomaly by at least one row of fine elements. This explains why the influence of the coarse mesh might grow significantly when the number of elements is reduced below a value of 4000. Nevertheless it is surprising that changes in the outer periphery of the bubble do already produce detectable errors. Given this sensitivity it is also surprising that the poorly resolved environmental wind field does not produce significant errors.

	adaptive	uniform
Δt	0.010 s	0.010 s
Δx_{eff}	3.12 m	3.12 m
L	11.05 m	11.05 m
#elements	3025.67	8192
total time	27616.75 s	74804.76 s
grid time	2604.47 s	0.27 s
$\max(\theta')$	0.42 K	0.42 K
$\min(\theta')$	0.00 K	0.00 K

Table 4: Details as in table 1–3 for the two simulations shown in figure 10 with semi-implicit time-integration. The time values are the time used for reaching $t = 1000$s of the warm air bubble test case and the number of elements is an average value over the whole time. The finest resolution in the adaptive simulation covers 33.8% of the whole domain averaged over the whole simulation time. Compared to figure 6d (table 3) the simulations in figure 10 were faster because in this section 5 we simulated only the left half of the bubble. However here the simulations were done until time $t = 1000$s whereas in table 3 an end time of $t = 700$s was used.

5.3. CPU Time: Adaptive vs. Uniform

In the previous subsection we derived a refinement region that produces approximately the same accuracy as a simulation using a uniform grid when both share the same finest resolution. This allows us to address the question: how much faster is the adaptive computation compared to the uniform simulation when both produce approximately the same accuracy and share the same finest spatial resolution? Table 4 shows that our simulation of the warm air bubble on a locally refined mesh is almost three times faster than the simulation using a uniform mesh. Only about one third of the elements is used, i.e. about one third memory requirements compared to the uniform grid. Figure 10 shows again the corresponding simulation results at time $t = 1000$s.

The CPU times in table 1 – 3 suggest that CPU time is correlated to the average number of elements used for the simulation. For this reason the advantage of local grid refinement is strongly problem dependent. For applications where boundary effects are especially significant, local grid refinement should provide a large advantage as the domain can easily be enlarged.

6. Summary and outlook

In this paper we presented a novel numerical model for solving the 2D compressible Euler equations in Cartesian geometry. It uses a high-order discontinuous Galerkin method based on triangular elements [23] in combination with a semi-implicit time-integrator [28]. This avoids the severe time-step restriction of explicit schemes. For the h-adaptivity we use the function library AMATOS [11] that uses a very efficient space filling curve approach. The choice of these numerical methods is motivated by the future application which the numerical model is designed for, namely the simulation of single cumulus clouds.

For testing our numerical model we simulated three standard test cases that are relevant for atmospheric convection. Our results agree well with the results from the literature. The adaptive mesh refinement uses a very simple refinement criterion: wherever the absolute value of the deviation of potential temperature from the background state exceeds a certain threshold a given finest resolution is used. However we do not know a priori how large the refinement region (defined by the value of the threshold) should be. We think that adaptive computations should be able to produce approximately the same results as uniform simulations when both use the same finest spatial resolution. This should be true even for setups that are strongly sensitive to numerical accuracy.

As the exact solution of our test cases is not known we looked for an approximate way to measure the accuracy of the simulation. We found that small vortices in the simulation of warm air bubbles are strongly sensitive to numerical accuracy. For identifying these vortices as clearly as possible we did not use any initial perturbations and we added artificial viscosity with a constant viscosity parameter. Our criterion for testing

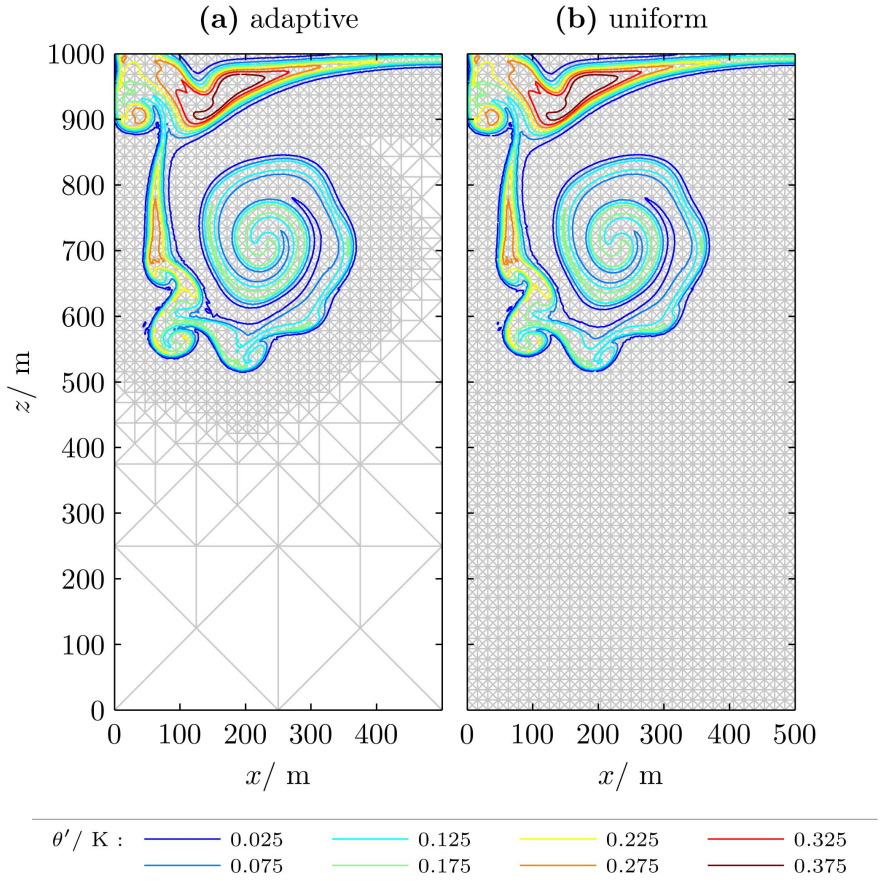

Figure 10: Direct comparison between a simulation using an adaptive mesh and a simulation using a uniform mesh after 1000 seconds. The adaptive simulation is identical to the one shown in figure 8a and the uniform result is the same as in figure 7b. The resolution of the uniform mesh and the finest resolution of the adaptive mesh is $\Delta x_{\text{eff}} = 3.1$ m.

refinement criteria is that these small vortices should be almost identical when being simulated by either adaptive or uniform computations when both share the same finest spatial resolution. The additional error introduced by using adaptivity should then be negligible compared to the numerical errors of simulations using a uniform grid.

This criterion allowed us to compare different sizes of the refinement region (figure 8). For a fixed spatial resolution we were even able to quantify the differences between the simulation results of different size refinement regions. We found that for a rising warm air bubble the average number of elements required for the adaptive simulation is about a factor three times smaller than the number required for the simulation with the uniform fine-resolution grid. Correspondingly the adaptive simulation is almost three times faster than the uniform simulation, and the advantage of adaptive mesh refinement becomes even more pronounced for larger domains. There is, however, one restriction: adaptive grid refinement is only efficient when an environment exists which can be resolved with a fairly coarse resolution.

For a simulation of cumulus clouds we expect that the domain has to be very large for avoiding boundary effects. However it is not completely clear which resolution is necessary in the environment of the cloud. Probably it should be possible to use a fairly coarse resolution in the environment when modeling non-resolved turbulence with a sub-grid scale model (e.g., a Smagorinsky-model [38]). If this turns out to be true adaptive mesh refinement should have a big potential for future cloud simulations when the refinement criterion is carefully chosen.

To reach the ultimate goal of our project, that is to simulate a single cloud, requires the following further

developments: first, we must include moisture into our model. This makes it necessary to add variables for the water vapor content and the liquid water content of the air. As water vapor can condensate and liquid water can evaporate, both accompanied with a change of potential temperature, an additional heat source has to be added to eq. (3). We are working on solving the equations with these additional terms by implementing the approach used by Klemp and Wilhelmson [39]. Finally, we plan to extend this work to three-dimensions where we expect to modify our models to handle either tetrahedral or triangular prismatic elements.

We will also continue our work on testing refinement criteria. Questions that should be considered include: how does the size of a carefully chosen refinement region depend on the spatial resolution and on the degree of the polynomials; and does our new method for testing refinement criteria produce similar efficiency results for moist air bubbles and for the extension to three dimensions?

7. Acknowledgements

We are grateful to the reviewers for giving many excellent suggestions for improving this paper. Financial support for this work was provided by the priority program MetStröm (SPP 1276) of the german research foundation (Deutsche Forschungsgemeinschaft). FXG gratefully acknowledges the support of the Office of Naval Research through program element PE-0602435N.

Appendix A. Motivation for our choice of μ_{lim}

In this section we give some additional motivation for our choice of the slope limiting viscosity parameter μ_{lim} in eq. (26). Let us start with a simple 1D diffusion equation for a function ϕ:

$$\frac{\partial \phi}{\partial t} = \mu \frac{\partial^2 \phi}{\partial x^2}, \tag{A.1}$$

with time t and spatial coordinate x. Scale analysis of this equation gives

$$M = \frac{L^2}{T}, \tag{A.2}$$

with the time scale T and the spatial scale L at which the viscosity acts and the scale of the viscosity coefficient M. The time scale T is given by the scale of the wind speed $V = L/T$. This gives:

$$M = V\, L. \tag{A.3}$$

In our simulation μ_{lim} is intended to dampen steep slopes. The spatial scale of these slopes is given by the spatial resolution of the numerical model Δx_{eff}. For the scale of the wind speed we use the maximum wind speed v_{max} that is expected throughout the whole simulation.

Our viscosity coefficient μ_{lim} should only dampen steep gradients. For this reason we assume the viscosity coefficient μ_{lim} to be element dependent and proportional to some exponent κ of the temperature gradient $\Delta\theta'_e$ of a considered element e. This yields

$$\mu_{\text{lim},e} \propto \left(\Delta\theta'_e\right)^{\kappa}\, v_{\text{max}}\, \Delta x_{\text{eff}}. \tag{A.4}$$

For the proportionality constant we use $\mu_{\text{ref}} / \left(\left(\alpha\, \Delta\theta'_0\right)^{\kappa}\, \Delta x_{\text{ref}}\, v_{\text{ref}} \right)$. The parameter μ_{ref} gives the viscosity that should be used for a temperature gradient of $\alpha\, \Delta\theta'_0$ with a spatial resolution Δx_{ref} and maximum wind speed v_{ref}.

References

[1] W. Skamarock, J. Oliger, R. L. Street, Adaptive grid refinement for numerical weather prediction, J. Comput. Phys. 80 (1989) 27–60.

[2] G. S. Dietachmayer, K. K. Droegemeier, Application of continuous dynamic grid adaption techniques to meteorological modeling .1. basic formulation and accuracy, Mon. Weather Rev. 120 (1992) 1675–1706.

[3] D. P. Bacon, N. N. Ahmad, Z. Boybeyi, T. J. Dunn, M. S. Hall, P. C. S. Lee, R. A. Sarma, M. D. Turner, K. T. Waight, S. H. Young, J. W. Zack, A dynamically adapting weather and dispersion model: The operational multiscale environment model with grid adaptivity (omega), Mon. Weather Rev. 128 (2000) 2044–2076.

[4] N. N. Ahmad, D. P. Bacon, M. S. Hall, A. Sarma, Application of the multidimensional positive definite advection transport algorithm (mpdata) to environmental modelling on adaptive unstructured grids, Int. J. Numer. Meth. Fl. 50 (2006) 1247–1268.

[5] R. Ford, C. C. Pain, M. D. Piggott, A. J. H. Goddard, C. R. E. de Oliveira, A. P. Umpleby, A nonhydrostatic finite-element model for three-dimensional stratified oceanic flows. part i: Model formulation, Mon. Weather Rev. 132 (2004) 2816–2831.

[6] C. Jablonowski, R. C. Oehmke, Q. F. Stout, Block-structured adaptive meshes and reduced grids for atmospheric general circulation models, Philos. T. R. Soc. A 367 (2009) 4497–4522.

[7] H. Weller, T. Ringler, M. Piggott, N. Wood, Challenges facing adaptive mesh modeling of the atmosphere and ocean, B. Am. Meteorol. Soc. 91 (2010) 105–108.

[8] J. Behrens, Adaptive atmospheric modeling: key techniques in grid generation, data structures, and numerical operations with applications, Springer Verlag, 2006.

[9] T. Heus, H. J. J. Jonker, Subsiding shells around shallow cumulus clouds, J. Atmos. Sci. 65 (2008) 1003–1018.

[10] J. C. Wyngaard, Toward numerical modeling in the "terra incognita", J. Atmos. Sci. 61 (2004) 1816–1826.

[11] J. Behrens, N. Rakowsky, W. Hiller, D. Handorf, M. Lauter, J. Papke, K. Dethloff, amatos: Parallel adaptive mesh generator for atmospheric and oceanic simulation, Ocean Model. 10 (2005) 171–183.

[12] W. W. Grabowski, T. L. Clark, Cloud environment interface instability - rising thermal calculations in 2 spatial dimensions, J. Atmos. Sci. 48 (1991) 527–546.

[13] W. W. Grabowski, T. L. Clark, Cloud environment interface instability. 2. extension to 3 spatial dimensions, J. Atmos. Sci. 50 (1993) 555–573.

[14] W. W. Grabowski, T. L. Clark, Cloud environment interface instability. 3. direct influence of environmental shear, J. Atmos. Sci. 50 (1993) 3821–3828.

[15] D. A. Randall, Conditional instability of the 1st kind upside-down, J. Atmos. Sci. 37 (1980) 125–130.

[16] W. W. Grabowski, Cumulus entrainment, fine-scale mixing, and buoyancy reversal, Q. J. R. Meteor. Soc. 119 (1993) 935–956.

[17] J. P. Mellado, B. Stevens, H. Schmidt, N. Peters, Buoyancy reversal in cloud-top mixing layers, Q. J. R. Meteor. Soc. 135 (2009) 963–978.

[18] H. Johari, Mixing in thermals with and without buoyancy reversal, J. Atmos. Sci. 49 (1992) 1412–1426.

[19] W. W. Grabowski, Entrainment and mixing in buoyancy-reversing convection with applications to cloud-top entrainment instability, Q. J. R. Meteor. Soc. 121 (1995) 231–253.

[20] G. H. Bryan, J. C. Wyngaard, J. M. Fritsch, Resolution requirements for the simulation of deep moist convection, Mon. Weather Rev. 131 (2003) 2394–2416.

[21] R. Damiani, G. Vali, S. Haimov, The structure of thermals in cumulus from airborne dual-doppler radar observations, J. Atmos. Sci. 63 (2006) 1432–1450.

[22] 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.

[23] F. X. Giraldo, T. Warburton, A high-order triangular discontinuous galerkin oceanic shallow water model, Int. J. Numer. Meth. Fl. 56 (2008) 899–925.

[24] F. X. Giraldo, J. S. Hesthaven, T. Warburton, Nodal high-order discontinuous galerkin methods for the spherical shallow water equations, J. Comput. Phys. 181 (2002) 499–525.

[25] M. A. Taylor, B. A. Wingate, R. E. Vincent, An algorithm for computing fekete points in the triangle, SIAM J. Numer. Anal. 38 (2000) 1707–1720.

[26] S. Wandzura, H. Xiao, Symmetric quadrature rules on a triangle, Comput. Math. Appl. 45 (2003) 1829–1840.

[27] B. Cockburn, C. W. Shu, Runge–kutta discontinuous galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (2001) 173–261.

[28] M. Restelli, F. X. Giraldo, A conservative discontinuous galerkin semi-implicit formulation for the navier-stokes equations in nonhydrostatic mesoscale modeling, SIAM J. Sci. Comput. 31 (2009) 2231–2257.

[29] F. X. Giraldo, M. Restelli, High-order semi-implicit time-integrators for a triangular discontinuous galerkin oceanic shallow water model, Int. J. Numer. Meth. Fl. 63 (2010) 1077–1102.

[30] M. Restelli, R. Sacco, L. Bonaventura, Semi–Lagrangian and semi–implicit discontinuous Galerkin methods for atmospheric modeling applications, Ph.D. thesis, Politecnico di Milano, 2007.

[31] F. X. Giraldo, M. Restelli, M. Lauter, Semi-implicit formulations of the navier-stokes equations: Application to nonhydrostatic atmospheric modeling, SIAM J. Sci. Comput. 32 (2010) 3394–3425.

[32] L. E. Carr, C. A. Borges, F. X. Giraldo, Spectrally-optimized preconditioners for element-based galerkin methods for the euler equations, SIAM J. Sci. Comput. in review (2011) 25pp.

[33] J. M. Straka, R. B. Wilhelmson, L. J. Wicker, J. R. Anderson, K. K. Droegemeier, Numerical-solutions of a nonlinear density-current - a benchmark solution and comparisons, Int. J. Numer. Meth. Fl. 17 (1993) 1–22.

[34] B. Cockburn, C. W. Shu, The local discontinuous galerkin method for time-dependent convection-diffusion systems, SIAM J. Numer. Anal. 35 (1998) 2440–2463.
[35] K. Shahbazi, P. F. Fischer, C. R. Ethier, A high-order discontinuous galerkin method for the unsteady incompressible navier-stokes equations, J. Comput. Phys. 222 (2007) 391–407.
[36] A. Robert, Bubble convection experiments with a semiimplicit formulation of the euler equations, J. Atmos. Sci. 50 (1993) 1865–1873.
[37] P. K. Kundu, Fluid Mechanics, Academic Press, 1990.
[38] J. Smagorinsky, General circulation experiments with the primitive equations, Mon. Weather Rev. 91 (1963) 99–164.
[39] J. B. Klemp, R. B. Wilhelmson, Simulation of 3-dimensional convective storm dynamics, J. Atmos. Sci. 35 (1978) 1070–1096.