V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

AN ADAPTIVE DISCONTINUOUS GALERKIN METHOD FOR MODELING CUMULUS CLOUDS

Andreas Müller*, Jörn Behrens†, Francis X. Giraldo†† and Volkmar Wirth*

*Institute for Atmospheric Physics,
University of Mainz (Germany)
e-mail: andrmue@uni-mainz.de

†KlimaCampus
University of Hamburg (Germany)
e-mail: joern.behrens@zmaw.de

††Department of Applied Mathematics
Naval Postgraduate School
Monterey, California (USA)
e-mail: fxgirald@nps.edu

Key words: Discontinuous Galerkin, Large Eddy Simulation, Adaptive, Meteorology, Clouds, Semi-Implicit

Abstract. *Theoretical understanding and numerical modeling of atmospheric moist convection still pose great challenges to meteorological research. The present work addresses the following question: How important is mixing between cloudy and environmental air for the development of a cumulus cloud? A Direct Numerical Simulation of a single cloud is way beyond the capacity of today's computing power. The use of a Large Eddy Simulation in combination with semi-implicit time-integration and adaptive techniques offers a significant reduction of complexity.*

So far this work is restricted to dry flow in two-dimensional geometry. The compressible Navier-Stokes equations are discretized using a discontinuous Galerkin method introduced by Giraldo and Warburton in 2008. Time integration is done by a semi-implicit backward difference. For the first time we combine these numerical methods with an h-adaptive grid refinement. This refinement of our triangular grid is implemented with the function library AMATOS and uses a space filling curve approach.

Validation through different test cases shows very good agreement between the current results and those from the literature. For comparing different adaptivity setups we developed a new qualitative error measure for the simulation of warm air bubbles. With the help of this criterion we show that the simulation of a rising warm air bubble on a locally refined grid can be more than six times faster than a similar computation on a uniform mesh with the same accuracy.

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 an extremely challenging subject owing to its multiscale and multi-process nature. Even with todays 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.

An application with high practical relevance is numerical weather prediction. Generally, its skill has improved considerably over the past few decades, and a significant portion of this improvement has been attributed to the increased computing power and refined numerical methods[1]. A notable exception to this general development is the forecast of precipitation, where the progress has been almost non-existent[2]. The reason for this state of affairs is likely based on the fact that most processes leading to precipitation are parameterized rather than explicitly simulated in today's prediction models. In this paper we shall develop and present a new numerical model that is specifically tailored to investigate one of these processes in detail.

The paper is organized as follows. First, in section 2 we motivate the new model development by explaining the meteorological problem. In section 3 we then present the numerical methods used in our work. This includes the discontinuous Galerkin method for the spatial discretization, a semi-implicit method for the time integration, and a space filling curve approach for the adaptive grid management. In section 4 we validate our code by a convergence study and three test cases from the literature. Section 5 provides some tests concerning the accuracy of the adaptive mesh refinement and sensitivity to viscosity. The paper ends with a summary and outlook in section 6.

2 METEOROLOGICAL PROBLEM

In this work we focus on a rather specific scenario, namely a single cloud that 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)[3]. This induces wind shear at the cloud-environment interface, leading to Kelvin-Helmholtz instability which eventually results in turbulence. 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 aptly called "buoyancy reversal"[5,6,7].

Early indications for the significance of buoyancy reversal for cloud dynamics stem from the laboratory experiments of Johari[8]. He introduced buoyancy reversal in his watertank experiments with the help of chemical reactions occurring in the mixing region between two fluids. Johari found that, depending on the strength of the buoyancy reversal, the

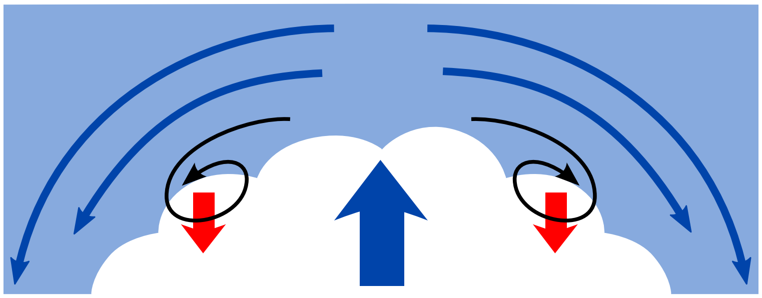

Figure 1: Illustration of buoyancy reversal. The blue arrows demonstrate the mean flow of a rising cloud, the black arrows represent turbulence produced by Kelvin-Helmholtz instability and the red arrows illustrate buoyancy reversal. For further explanation we refer to the text.

morphology of the cloud development could be vastly different.

Similar results were found in highly idealized numerical two-fluid experiments by Grabowski[4] in 1995. These simulations started with two fluid layers, one on top of the other. Convergence was imposed in the lower layer thus leading to a rising plume. The simulation was done twice: once with and once without buoyancy reversal. The buoyancy reversal was implemented as an additional downward force right at the interface between the two fluids. Initially the two simulations looked very similar, but in the further development increasing differences could be seen.

These preliminary investigations suggest that buoyancy reversal has an important impact on cloud dynamics and, hence, on the formation of precipitation. 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 explicity simulate the processes relevant for buoyancy reversal owing to their coarse spatial resolution[10]. It is here that we want to make 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[10]. In three dimensions this amounts to some 10^{24} grid points, which is way beyond the capacity of today’s computing power. We, therefore, resort to Large Eddy Simulation (LES) in combination with a semi-implicit time integration. 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.

3 NUMERICAL METHODS

In this section we present the numerical methods that are used in our work. The choice of the numerical method requires some *a priori* knowledge of what a real cloud looks like.

Measurements have shown a large variety of behavior[9]: There are clouds with numerous steep gradients in the interior (see for example the liquid water content in figure 15 of that reference[9]). 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.[9]). It is the latter which we will concentrate on. For such clouds a discontinuous Galerkin (DG) discretization in combination with a semi-implicit time-integration should be the best choice; the reason for this choice is due to 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. The work in this paper represents the first application of a semi-implicit high-order discontinuous Galerkin method with h-adaptive mesh refinement for meteorological flow problems. Before describing these methods in the following subsections we first introduce the equations that we use in our simulations.

In the current work the fully compressible Navier-Stokes equations are used. For the dry case we use the following set of equations (see equation set 2 in Giraldo and Restelli[12]):

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

$$\frac{\partial\rho\boldsymbol{u}}{\partial t} + \boldsymbol{\nabla}\cdot(\rho\,\boldsymbol{u}\otimes\boldsymbol{u} + p\,I_2) = -\,\rho\,g\,\boldsymbol{k} + \boldsymbol{\nabla}\cdot(\mu\,\rho\,\boldsymbol{\nabla}\boldsymbol{u})\,, \tag{2}$$

$$\frac{\partial\rho\theta}{\partial t} + \boldsymbol{\nabla}\cdot(\rho\,\theta\,\boldsymbol{u}) = \boldsymbol{\nabla}\cdot(\mu\,\rho\,\boldsymbol{\nabla}\theta)\,, \tag{3}$$

where the variables are $(\rho, \rho\boldsymbol{u}, \rho\theta)^{\mathrm{T}}$, ρ is the density, $\boldsymbol{u} = (u, w)^{\mathrm{T}}$ is the velocity field and θ is the potential temperature. Furthermore we denotd the gravitational constant with g, the tensor product by $\otimes$, the identity in $\mathbb{R}^2$ by I_2 and the unit vector in the vertical direction with $\boldsymbol{k}$. The last term in eq. (2) and eq. (3) describes viscosity in a simplified way with the dynamic viscosity parameter μ. Pressure p in eq. (2) is given by the equation of state:

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

with a constant reference pressure p_0, 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. This definition can be illustrated in the following way: if we consider dry air with temperature T and pressure p, potential temperature θ is given by the temperature which the air would have when being transported adiabatically to a place with pressure p_0. 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 $\bar{\theta}$ which are in hydrostatic balance. The mean state of pressure p is defined by $\bar{p} = p(\bar{\rho}, \bar{\theta})$. The deviation of the variables from the mean state is denoted by $\rho' = \rho - \bar{\rho}$, $\theta' = \theta - \bar{\theta}$ and $p' = p - \bar{p}$. By this procedure the set of equations (1) – (3) can be written as

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

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho\, \boldsymbol{u} \otimes \boldsymbol{u} + p'\, I_2) = -\, \rho'\, g\, \boldsymbol{k} + \boldsymbol{\nabla} \cdot (\mu\, \rho\, \boldsymbol{\nabla} \boldsymbol{u})\,, \tag{8}$$

$$\frac{\partial \rho \theta'}{\partial t} + \boldsymbol{\nabla} \cdot (\rho\, \theta\, \boldsymbol{u}) = \boldsymbol{\nabla} \cdot (\mu\, \rho\, \boldsymbol{\nabla} \theta)\,. \tag{9}$$

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 $\boldsymbol{q} = (\rho, \rho\boldsymbol{u}, \rho\theta)^{\mathrm{T}}$, the source function

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

and the flux tensor

$$\mathbf{F}\,(\boldsymbol{q}) = \begin{pmatrix} \rho\, \boldsymbol{u} \\ \rho\, \boldsymbol{u} \otimes \boldsymbol{u} + p'\, I_2 - \mu\, \rho\, \boldsymbol{\alpha} \\ \rho\, \theta\, \boldsymbol{u} - \mu\, \rho\, \beta \end{pmatrix}. \tag{12}$$

In the viscosity term of the flux tensor we reduced the order of the derivatives by introducing the following new variables

$$\boldsymbol{\alpha} = \boldsymbol{\nabla} \boldsymbol{u}, \tag{13}$$

$$\beta = \boldsymbol{\nabla} \theta. \tag{14}$$

These equations are solved in each timestep with a discontinuous Galerkin discretization. We come back to this at the end of the next subsection.

3.1 Discontinuous Galerkin Method

In our work we use a 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 triangular prisms for this task. The triangular discontinuous Galerkin method used in our work is described by Giraldo and Warburton[11] for the case of 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 only repeat in this paper the main ideas of the descretization.

We start with 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 $\boldsymbol{q}_N$:

$$\int_{\Omega_e} \left(\frac{\partial \boldsymbol{q}_N}{\partial t} - \mathbf{F}_N \cdot \boldsymbol{\nabla} - \boldsymbol{S}_N \right) \psi(\boldsymbol{x})\, \mathrm{d}\mathbf{x} = - \int_{\Gamma_e} \psi(\boldsymbol{x})\, \boldsymbol{n} \cdot \mathbf{F}^*_N \mathrm{d}\mathbf{x}, \tag{15}$$

where Γ_e is the boundary of element Ω_e, $\boldsymbol{n}$ is the outward pointing normal vector of Γ_e, $\mathbf{F}_N = \mathbf{F}(\boldsymbol{q}_N)$ and $\boldsymbol{S}_N = \boldsymbol{S}(\boldsymbol{q}_N)$. Applying again integration by parts gives the *strong formulation*

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

Now we introduce an expansion by the polynomial basis functions

$$\boldsymbol{q}_N(\boldsymbol{x}) = \sum_{j=1}^{M_N} \psi_j(\boldsymbol{x})\, \boldsymbol{q}_j \tag{17}$$

and assume that the test function ψ can be written as a linear combination of the basis functions. Similarly we get coefficients $\mathbf{F}_j$ and $\boldsymbol{S}_j$ by expanding $\mathbf{F}_N$ and $\boldsymbol{S}_N$. Using Einstein's sum convention we get

$$\frac{\partial \boldsymbol{q}_i}{\partial t} = - \int_{\Omega_e} \hat{\psi}_i \left(\mathbf{F}_j \boldsymbol{\nabla} \psi_j - \boldsymbol{S}_j \psi_j \right) \mathrm{d}\mathbf{x} + \int_{\Gamma_e} \hat{\psi}_i \psi_j \boldsymbol{n}\, \mathrm{d}\mathbf{x} \cdot \left(\mathbf{F}_j - \mathbf{F}^*_j \right), \tag{18}$$

where $\hat{\psi}_i = M^{-1}_{ik} \psi_k$ with the mass matrix $M_{ik} = \int_{\Omega_e} \psi_i \psi_k \mathrm{d}\mathbf{x}$. 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.

The integrals in eq. (18) are evaluated using high order cubature and quadrature rules[11]. For the numerical flux we use the Rusanov flux which is given by

$$\mathbf{F}^*_N = \frac{1}{2} \left[\mathbf{F}\left(\boldsymbol{q}^L_N\right) + \mathbf{F}\left(\boldsymbol{q}^R_N\right) - \lambda \left(\boldsymbol{q}^R_N - \boldsymbol{q}^L_N\right) \right] \tag{19}$$

with the maximum wave speed $\lambda = \max\left(|u| + a, |u| - a\right)$ where a is the speed of sound. If the normal vector $\boldsymbol{n}$ 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). By using a similar approach for the viscosity terms (13) and (14) we get

$$\boldsymbol{\alpha}_i = -\int_{\Omega_e} \hat{\psi}_i\, \boldsymbol{u}_j \boldsymbol{\nabla} \psi_j \mathrm{dx} + \int_{\Gamma_e} \hat{\psi}_i \psi_j \boldsymbol{n}\, \mathrm{dx} \left(\boldsymbol{u}_j - \boldsymbol{u}_j^*\right), \tag{20}$$

$$\beta_i = -\int_{\Omega_e} \hat{\psi}_i\, \theta_j \boldsymbol{\nabla} \psi_j \mathrm{dx} + \int_{\Gamma_e} \hat{\psi}_i \psi_j \boldsymbol{n}\, \mathrm{dx} \left(\theta_j - \theta_j^*\right). \tag{21}$$

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 $\mathbf{F}_N^*$ as well as for $\boldsymbol{u}^*$ and θ^*:

$$\mathbf{F}^*_{\text{visc}}\left(\boldsymbol{q}_N\right) = \frac{1}{2}\left(\boldsymbol{q}_N^R + \boldsymbol{q}_N^L\right). \tag{22}$$

At this point, the right hand side of eq. (18) is known and we can integrate the equation in time. This can be done either by an explicit or an implicit method. For an explicit method we implement a third order Runge-Kutta method of Cockburn and Shu[13]. 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[14,15]. The main difference is that we use potential temperature instead of total energy as fourth variable.

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

$$\mathcal{N}\left(\boldsymbol{q}\right) = -\boldsymbol{\nabla}\cdot\mathbf{F}\left(\boldsymbol{q}\right) + \boldsymbol{S}\left(\boldsymbol{q}\right). \tag{23}$$

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

$$\boldsymbol{\mathcal{L}}\boldsymbol{q} = \begin{pmatrix} \boldsymbol{\nabla}\cdot\left(\rho\,\boldsymbol{u}\right) \\ \partial p/\partial x \\ \partial p/\partial z + g\,\rho' \\ \boldsymbol{\nabla}\cdot\left(\bar{\theta}\,\rho\,\boldsymbol{u}\right) \end{pmatrix}, \tag{24}$$

where a linearized version of the pressure p is used, given by $p = \frac{c_p\,\bar{p}}{c_v\,\bar{\rho}\,\bar{\theta}}\,\rho\,\theta'$. As explained by Restelli[16] this linear operator is responsible for the fast moving sound and gravity waves and, therefore, must be integrated implicitly. This splitting is done by writing

$$\frac{\partial \boldsymbol{q}}{\partial t} = \{\mathcal{N}(\boldsymbol{q}) - \mathcal{L}\,\boldsymbol{q}\} + \mathcal{L}\,\boldsymbol{q}. \tag{25}$$

For discretizing (25) in time, we use a backward difference of order 2, that leads to

$$\frac{1}{\gamma\,\Delta t}\sum_{m=-1}^{1}\alpha_m \boldsymbol{q}^{n-m} = \sum_{m=0}^{1}\beta_m\left[\mathcal{N}\left(\boldsymbol{q}^{n-m}\right) - \mathcal{L}\,\boldsymbol{q}^{n-m}\right] + \mathcal{L}\,\boldsymbol{q}^{n+1} \tag{26}$$

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

$$\left[1 - \gamma\,\Delta t\,\mathcal{L}\right]\,\boldsymbol{q}^{n+1} = \tilde{\boldsymbol{q}}^{\mathrm{ex}} - \gamma\,\Delta t\sum_{m=0}^{1}\beta_m\mathcal{L}\,\boldsymbol{q}^{n-m}, \tag{27}$$

where

$$\tilde{\boldsymbol{q}}^{\mathrm{ex}} = \sum_{m=0}^{1}\alpha_m\boldsymbol{q}^{n-m} + \gamma\,\Delta t\sum_{m=0}^{1}\beta_m\mathcal{N}\left(\boldsymbol{q}^{n-m}\right) \tag{28}$$

is an explicit predictor that has to be calculated first. Solving the linear system of equations (27) (e.g., with a GMRES) gives the implicit corrector.

3.3 Mesh Refinement with Space Filling Curve Approach

As explained in section 2 we expect steep gradients at the boundary of the cloud. For increasing the numerical resolution in these regions we use h-adaptive mesh refinement. This is managed with the function library AMATOS[19]. The main advantage of this function library is that it handles the entire h-adaptive mesh refinement. Furthermore it orders the unknowns very efficiently by using a so-called space filling curve approach. For further information we refer to the publication of Behrens et al.[19].

The only modification which 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. We simply evaluate the old polynomials at the positions of the new degrees of freedom. Mass conservation is guaranteed by a corrective constant.

For the refinement criterion we currently use

$$|\theta'| \geq \max_{\boldsymbol{x}}\left(|\theta'(\boldsymbol{x}, t=0)|\right)/10. \tag{29}$$

Wherever this condition is fulfilled the mesh is refined until it reaches a given finest resolution. In the rest of the domain the grid is coarsened until it reaches a given coarsest resolution without modifying the resolution in the refinement region. The transition between fine mesh and coarse mesh is given by the conformity of the grid. For avoiding small scale structures moving into a region with a coarse mesh we add two to four rows of fine elements to the refinement region. In the future we will test different refinement criterions. For example the gradient of the potential temperature and the gradient of the density of water vapor are interesting candidates.

Each of the numerical methods presented in this section have already been successfully used for different applications. The novelty of our work is the combination of the discontinuous Galerkin discretization by Giraldo and Warburton[11] with this h-adaptive mesh refinement for meteorological flow problems.

4 VALIDATION

For the validation of our new numerical model we used a convergence study with the method of manufactured solutions. Furthermore we considered three test cases that are fairly similar to our cloud simulation. These test cases are a small cold air bubble on top of a large warm air bubble from Robert[17], a density current from Straka et al.[18], and a smooth warm air bubble from Giraldo and Restelli[12]. The results are presented in the following subsections.

4.1 Convergence Study

A discontinuous Galerkin method of polynomial order n_p should be able to describe polynomial solutions of this order (plus one) exactly. Therefore the order of convergence should be $n_p + 1$. This is the so-called *optimal order of convergence.* For proving that our code reaches this optimal order of convergence we have to know an exact solution. Up to now no exact solution for the two dimensional unsteady Navier-Stokes equations is known (at least not for realistic flow conditions with gravity effects included). Instead we have to consider the inhomogeneous equations which arise analytically when inserting given functions for the variables ρ, u, w and θ. These functions have to be consistent with the given boundary conditions. We consider solid wall boundary conditions. Therefore we define

$$u/\frac{\mathrm{m}}{\mathrm{s}} = A\sin^2(2\pi\, n\, x)\sin^2(2\pi\, n\, z)\sin(2\pi\, t/T)\,, \quad w = u, \tag{30}$$
$$\rho/\frac{\mathrm{kg}}{\mathrm{m}^3} = u/\frac{\mathrm{m}}{\mathrm{s}} + 2A + 1, \quad \text{and} \quad \theta/\mathrm{K} = \rho/\frac{\mathrm{kg}}{\mathrm{m}^3}$$

in a domain of 1m×1m. The amplitude A, spatial wave number $n \in \mathbb{N}$ and period T can be chosen arbitrarily. We use the parameters:

$$A = 0.1, \quad n = 3, \quad T = 10\,\mathrm{ms}, \tag{31}$$

and a timestep of $\Delta t = 0.5$ms. By inserting these given functions into eq. (10) we derive analytically a new source function. For this modified problem eq. (30) is the exact solution and the L^2-error can be computed.

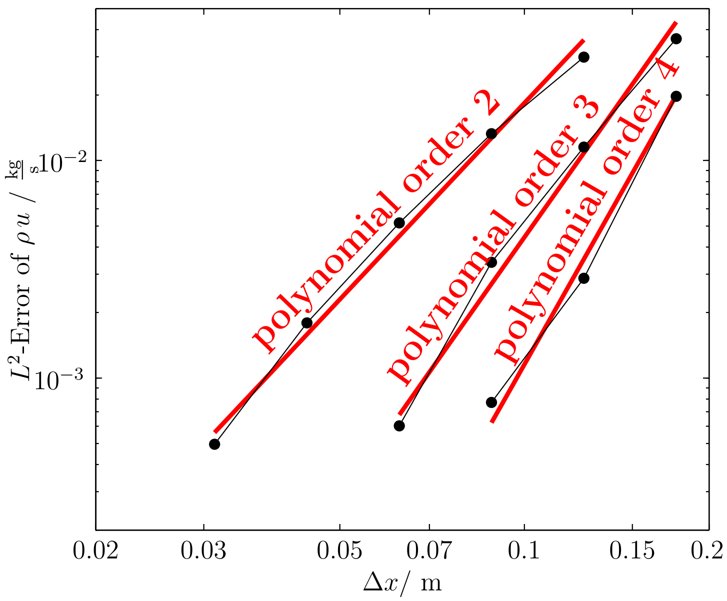

Figure 2: L^2-error of horizontal wind speed u as a function of resolution Δx, which is given by the length of the shortest element edge. The corresponding polynomial order n_p is given by the red labels. The thick red lines show the result of a linear regression with the slope set to the optimal order of convergence $n_p + 1$.

polynomial order:	**2**	**3**	**4**
experimental order of convergence:	2.9424	3.8996	4.6764

Table 1: Experimental order of convergence computed by linear regression of the error in figure 2.

Figure 2 shows the L^2-error of ρu at time $t = T$. A linear regression of the error shown in figure 2 gives the order of convergence shown in table 1. As the order of convergence is approximately one order higher than the polynomial order, this table clearly indicates that optimal order of convergence is achieved. In this case we used explicit time-integration with a third order Runge-Kutta method of Cockburn and Shu[13] and no viscosity. Similar results are obtained with viscosity and with semi-implicit time-integration (provided that a sufficiently small time-step is used to isolate the error due to the spatial discretization).

4.2 Three Test Cases

So far we have shown that our method converges for a modified set of equations. For validating our code with the original Navier-Stokes equations we use test cases that are as similar as possible to our cloud simulation. In these cases no exact solution exists but we can compare our results with those from the literature.

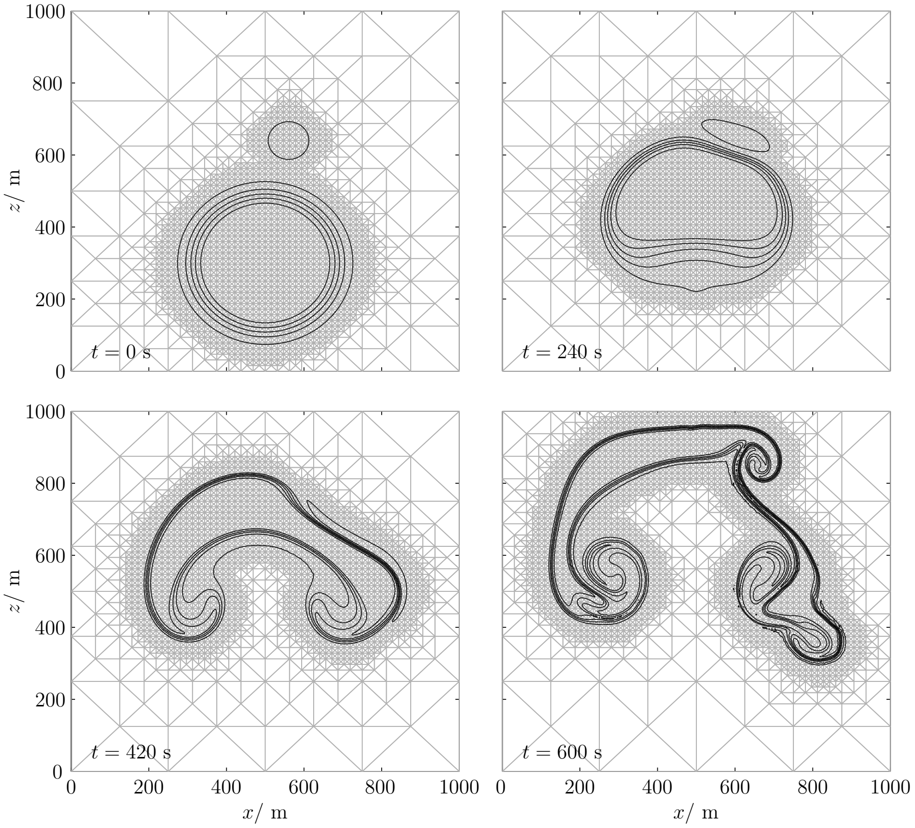

Figure 3: Small cold air bubble on top of a large warm air bubble as introduced by Robert[17]. The contour lines show the deviation of the potential temperature from the background state and the gray lines show the adaptively refined triangular mesh used in our simulation. The contour values are from -0.05 K to 0.45 K with an interval of 0.05 K. For avoiding artificial oscillations we use a constant physical viscosity of $\mu = 0.1\mathrm{m}^2/\mathrm{s}$. For the time-integration we used here the explicit Runge-Kutta method.

4.2.1 Small Cold Air Bubble on Top of Large Warm Air Bubble

The first test case that we consider 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[17] in 1993. The background state has a constant potential temperature of $\bar{\theta} = 300\,\mathrm{K}$. 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 0.17 K. All parameters are chosen identically to those in the publication of Robert[17].

There are only two differences: First we use a slightly different resolution. The shortest element edge in our simulation has a length of 11m. In combination with third order polynomials the number of degrees of freedom corresponds to a first order method with a resolution of about 4m. We call this reduced value “effective resolution”. With 4m it is slightly smaller than 5m of Robert[17]. Second we use a constant physical viscosity of $\mu = 0.1\text{m}^2/\text{s}$, because we have currently no flux limiter implemented in our code. We found that this viscosity is suitable for avoiding artificial oscillations at this resolution of 11m.

Figure 3 shows our result for this test case. By comparing our result with the corresponding figure of Robert[17] one can see that the results agree very well.

4.2.2 Density Current

A second test case is a density current initialized by a cold air bubble with a cosine profile and an amplitude of 15 K in θ' (figure 4). This test case was introduced by Straka et al.[18]. The viscosity of $\mu = 75\text{m}^2/\text{s}$ is identical to the setup of Straka et al.[18]. As in the previous test case we use third order polynomials. The shortest element edge has in our computation a length of 50m. This leads again to an effective resolution that is slightly smaller than the smallest resolution of Straka et al. with 25m. Again we see no differences between our result and the result in the literature.

4.2.3 Smooth Warm Air Bubble

As a third test case we computed the rising thermal bubble introduced by Giraldo and Restelli[12] (test case 2) in 2008. It is a single warm air bubble with a cosine profile in θ'. As in the test case of Robert[17] 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[12] except the viscosity parameter and the numerical resolution are now modified. For the viscosity parameter we use again $\mu = 0.1\text{m}^2/\text{s}$ and the shortest element edge has a length of 11m. By using third order polynomials our effective resolution is about 4m. This is significantly larger than all the effective resolutions used by Giraldo and Restelli. As they use 10th order polynomials one has to divide their resolution by a factor of three to get a comparable number of degrees of freedom as in our simulation. This explains that they get almost no artificial oscillations even without using a physical viscosity.

As in the previous test cases there are no obvious differences between our results and those from the literature. This gives us confidence that our code is error-free.

5 SENSITIVITY STUDIES

One important question for each adaptive numerical model is: how accurate is the adaptive method? For comparing different adaptivity setups we introduce a new error measure for the simulation of warm air bubbles. In this section we start with describing this criterion that is used later for some sensitivity studies. These studies include a comparison between a simulation on an adaptive mesh with a simulation on a uniform mesh, a sensitivity study concerning the size of the refinement region and a sensitivity

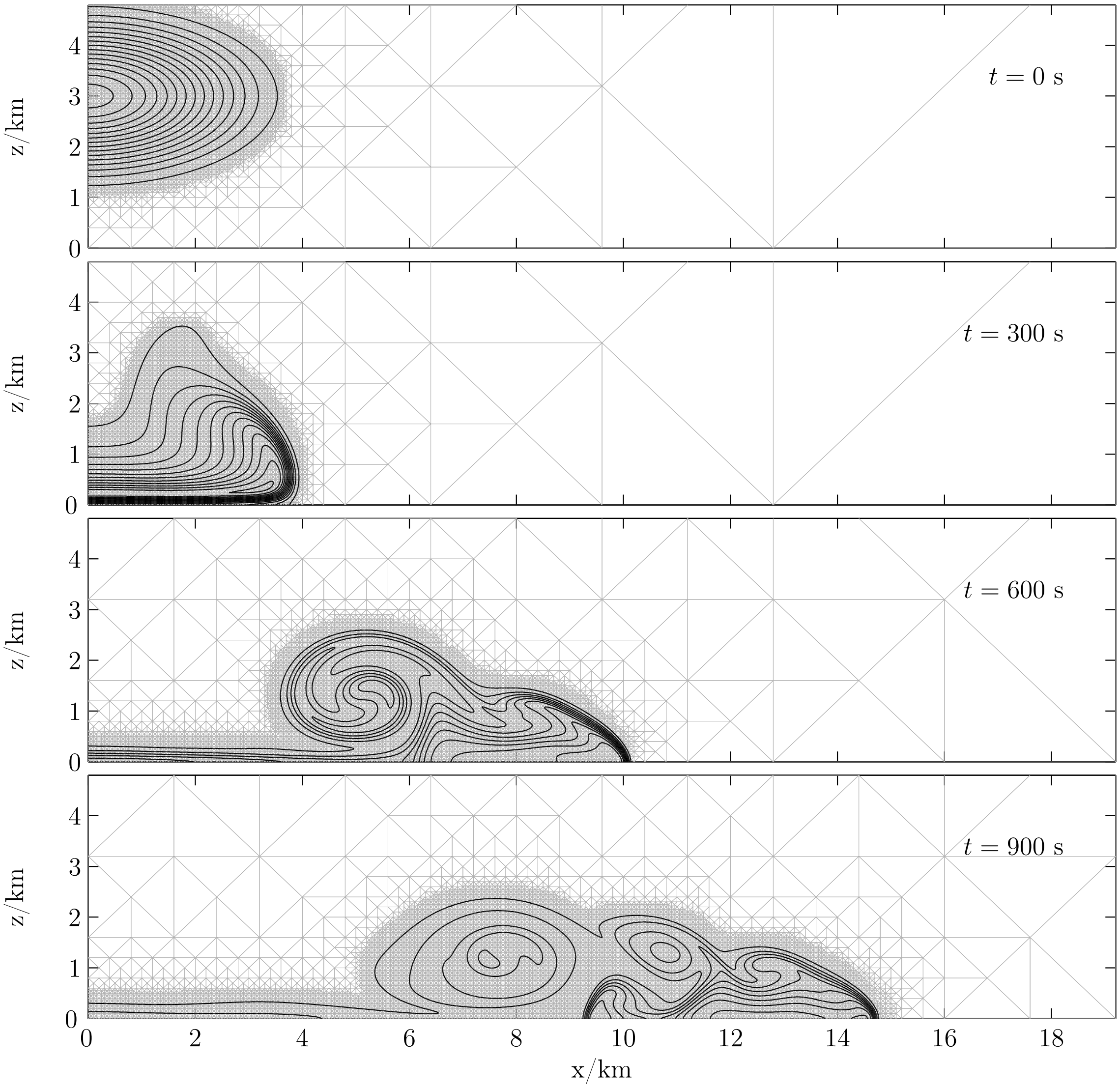

Figure 4: Density current as introduced by Straka et al.[18]. As in figure 3 the contour lines show the deviation θ' of the potential temperature θ from the background state $\bar{\theta}$ and gray lines show the adaptively refined triangular mesh. Contour values are from 0.5 K to 14.5 K with an interval of 1 K. Because of the fairly strong physical viscosity of $\mu = 75\text{m}^2/\text{s}$ no values outside this range exist.

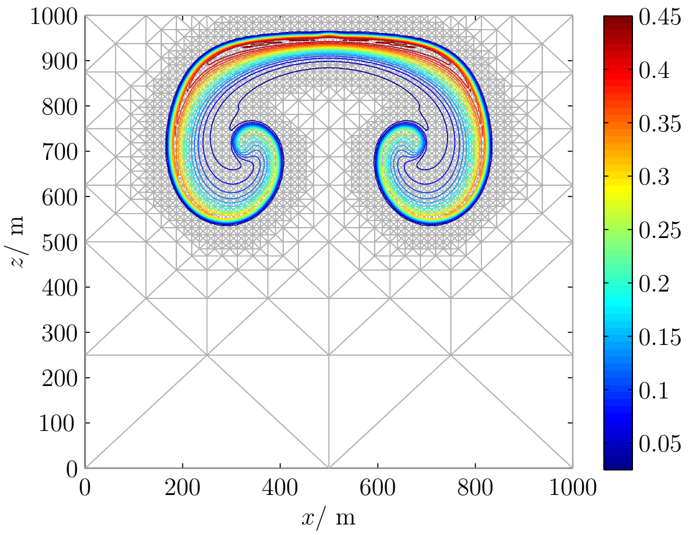

Figure 5: Rising thermal bubble introduced by Giraldo and Restelli[12]. As in figure 3 the contour lines show the deviation θ' of the potential temperature θ from the background state $\bar{\theta}$ and gray lines show the adaptively refined triangular mesh. Contour values are from 0.025 K to 0.45 K with an interval of 0.025 K. Because of the physical viscosity of $\mu = 0.1\text{m}^2/\text{s}$ there exist no negative values and no values larger than 0.45 K.

study concerning on the dependence of the solution quality to the viscosity parameter μ.

5.1 Comparison Criterion

For comparing different adaptivity setups we had to find a criterion which gives at least an approximation of the accuracy of an adaptive simulation. One possibility would be to construct a smooth function that satisfies all boundary conditions. As in section 4.1 this could be used to construct a modified set of equations that is solved analytically by the given function. In this approach the L^2-error could be calculated directly. The problem is that it is very difficult to define a smooth function that is suitable for applying a similar adaptive mesh refinement as in our cloud simulation.

As explained before no exact solution for the full Navier-Stokes equations is known. Hence it is impossible to calculate the error of a simulation using the full Navier-Stokes equations. Instead we decided to find a qualitative measure for the numerical errors to compare different simulations. Therefore we use the rising thermal bubble of Giraldo and Restelli[12] as in section 4.2.3. But this time we continue the simulation much longer until numerical errors become clearly visible. After 900 seconds we get the results shown in figure 6 for three different resolutions. As in section 4.1 this figure shows, that the results converge. Furthermore one can see that vortices which arise by Kelvin-Helmholtz instability at the left bottom part of the bubble strongly depend on numerical resolution.

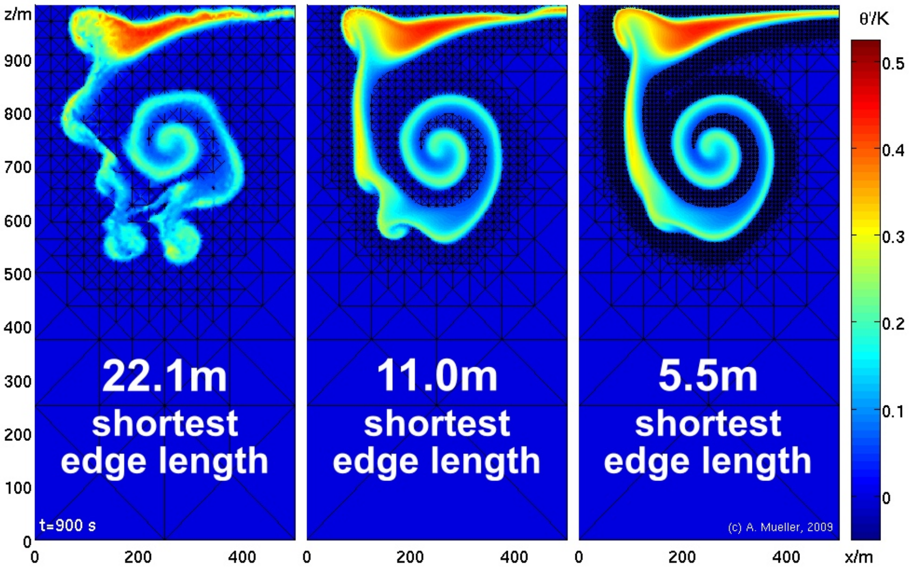

Figure 6: Rising thermal bubble as in figure 5 for three different resolutions. Shown is only the left half of the bubble. Color shading indicates the potential temperature deviation from the background state. The adaptively refined triangular mesh is shown by the black lines. The mesh is not shown where the refinement criterion (29) is fulfilled. The fine grid surrounding this region has to be continued in this refinement region. The numerical resolution is given by the length of the shortest element edge.

This is not surprising. A perfect horizontal sheer-flow will never show any sheer-instability. To see instabilities it is necessary to have an initial perturbation [20]. In a similar way we expect that a perfect circular warm air bubble in an infinitely large domain should never show any Kelvin-Helmholtz instability. But even a tiny perturbation will produce an instability. This instability becomes visible if the simulation is done long enough. In our numerical simulations this initial perturbation is given only by numerical errors. So we can compare different simulations by comparing the time when an instability at the bottom left side of the bubble becomes visible.

This criterion is not suitable for comparing different numerical models. As we will see in figure 9 the onset of Kelvin-Helmholtz instability depends also on diffusion. A numerical model with strong numerical diffusion produces less instability but is not more accurate. Nevertheless the instability is a good criterion for comparing different simulations of the same numerical model with a constant physical and numerical diffusion.

A further remark concerning figure 6. At the top left corner of the domain one can see a vortex that is almost independent of the numerical resolution. This vortex is caused by Kelvin-Helmholtz instability too, but in this case the instability is initialized by the solid wall boundaries. A similar behavior can be seen in the test case of Straka et al. in figure 4. These vortices are almost independent of the numerical resolution and cannot be used

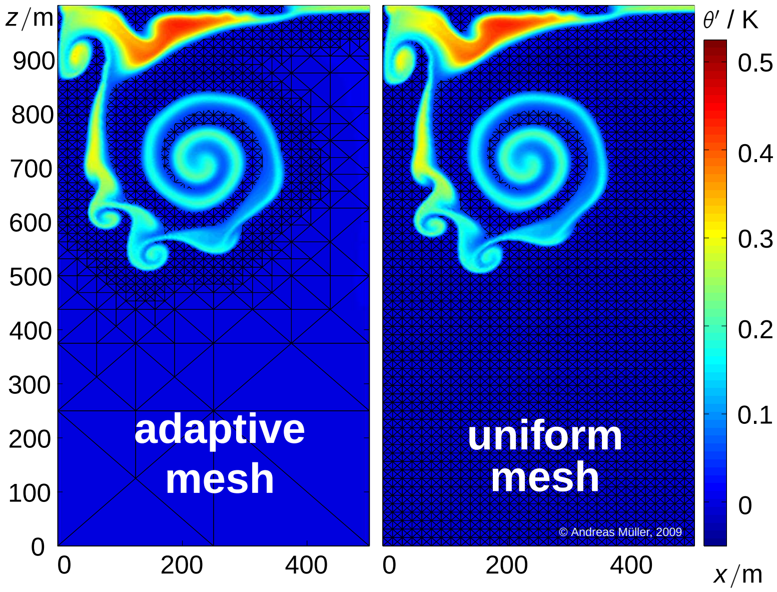

Figure 7: Comparison between a simulation using an adaptive mesh and a simulation using a uniform mesh after 1000 seconds. The resolution of the uniform mesh is equal to the finest resolution of the adaptive mesh. The length of the shortest element edges is 11m in both cases. All parameters of the rising bubble are identical to those in figure 5.

for comparing the accuracy of different simulations.

5.2 Adaptive versus Uniform

With the criterion introduced in the previous subsection we now compare a simulation done on an adaptively refined triangular mesh with a simulation on a uniform grid. The result is shown in figure 7. The onset of the Kelvin-Helmholtz instability and even the small vortices produced by the instability are almost identical in both simulations. This shows that the locally refined mesh is able to produce the same accuracy as a uniform mesh.

But there is one important difference between these two simulations: the simulation on the locally refined mesh is more than six times faster than the simulation using the uniform mesh. This demonstrates very clearly the advantage of using adaptive mesh refinement for simulating clouds.

5.3 Size of Refinement Region

As shown in the previous subsection the simulation using adaptive mesh refinement is much faster than a simulation using a uniform mesh. Nevertheless both simulations have the same accuracy. Can we improve the efficiency by using a smaller refinement region without losing accuracy? To answer this question we repeated the adaptive simulation of

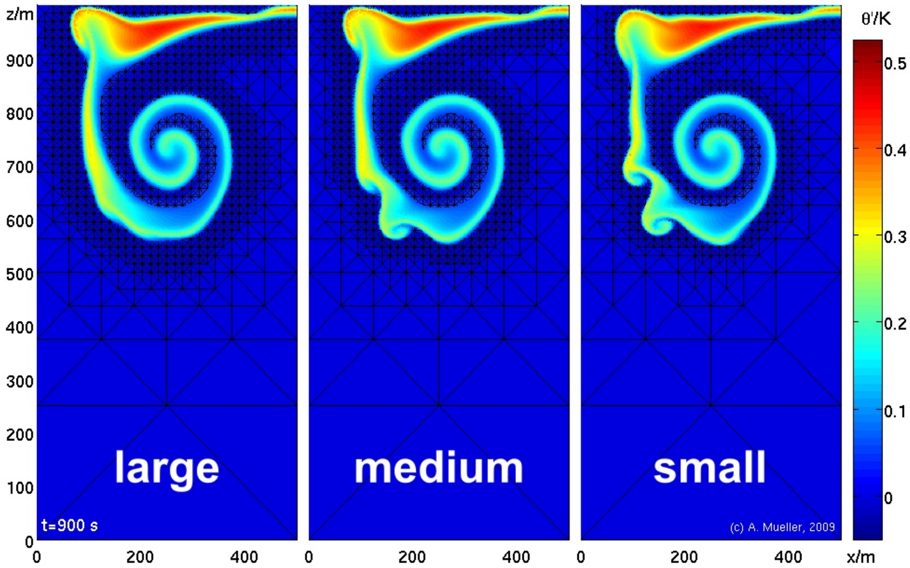

Figure 8: Adaptive simulation as in figure 7 for three different refinement regions after 900 seconds. The refinement criterion is always given by (29), but the number of additional fine elements surrounding this region is varied. From the right hand side to the left, one row of fine elements is added in each simulation. The largest refinement region shown on the left hand side is the same as in figure 7.

figure 7 with three slightly different refinement regions (figure 8).

The onset of Kelvin-Helmholtz instability is significantly earlier when reducing the size of the refinement region. This indicates that the numerical errors are increasing. In the case of this rising warm air bubble it seems to be better to use the large refinement region as in figure 7. Regardless, the computational effort is not much reduced by the smaller refinement regions. When developing new refinement criterions in future research we will have to repeat this test for making sure that our adaptive mesh gives the same accuracy as a simulation on a uniform mesh.

5.4 Viscosity Parameter

As a last sensitivity study we repeated the simulation of the rising warm air bubble for three different values of the physical viscosity parameter μ (figure 9). When decreasing viscosity smaller vortices become visible. Most of the vortices are certainly initiated by numerical errors. But we expect the same transition to smaller vortices if we start our simulations with more realistic turbulent initial data. This result is no surprise. For our goal to simulate clouds it is important to recognize that we will never be able to make a Direct Numerical Simulation of a real cloud, resolving all small scale vortices. We want to find out whether a resolution of some meters is small enough for simulating the mixing between cloudy air and environmental air.

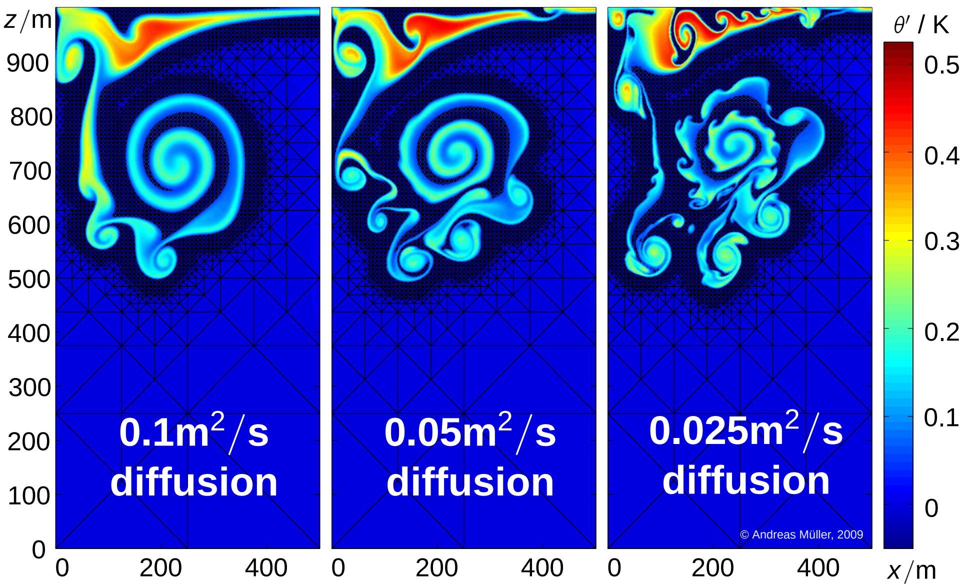

Figure 9: Rising warm air bubble as in figure 7 for three different values of the physical viscosity parameter μ after 1000 seconds. For avoiding artificial oscillations in the case of the small viscosity we use a resolution of 5.5m.

6 SUMMARY AND OUTLOOK

In this paper we presented a numerical model for solving the stratified compressible Navier-Stokes equations. It uses a high-order discontinuous Galerkin method based on triangular elements[11] in combination with a semi-implicit time-integrator[14]. This avoids the severe time-step restriction of explicit schemes. To our knowledged, this is the first time that a DG compressible Navier-Stokes model has been combined with an h-adaptive mesh refinement for stratified flows that are of interest in meteorological applications. For the h-adaptivity we use the function library AMATOS[19] which uses a very efficient space filling curve approach.

Validation of our numerical model by a convergence study with the method of manufactured solutions indicates that we reach the optimal order of convergence. As no exact solution exists for the full compressible Navier-Stokes equations we simulated three commonly used test cases. Our results agree very well with the corresponding results from the literature.

For testing whether the results are affected by using adaptive mesh refinement we developed a new qualitative error measure for the simulation of rising warm air bubbles. We found that Kelvin-Helmholtz instability strongly depends on the accuracy of the numerical simulation. Therefore the time when the instability becomes visible is a good criterion for comparing different simulations.

With this criterion we compared a simulation using an adaptive mesh with a simulation using a uniform mesh. The finest resolution is the same in both cases. Figure 7 shows almost no difference between the two simulations even after the instability has fully evolved. The simulation on the locally refined mesh is as accurate as the simulation on the uniform mesh. But the simulation using adaptive mesh refinement is more than six times faster than the simulation on the uniform mesh. This indicates that adaptive mesh refinement should be a big advantage for our future cloud simulations.

For simulating clouds we are currently working on implementing moisture in our code. Furthermore we have to develop suitable flux limiters and to test different refinement criteria; two possibilities are the gradient of the potential temperature and the gradient of the density of water vapor.

With our numerical model we want to find out which resolution is necessary for simulating mixing processes between cloudy air and environmental air. A better understanding of these processes should be an important step towards a more realistic simulation of clouds. We expect that our results will improve the daily weather forecast by allowing better parameterizations for mixing processes.

7 ACKNOWLEDGEMENT

Financial support for this work was provided by the priority program MetStröm (SPP 1276) of the german research foundation (Deutsche Forschungsgemeinschaft). Francis X. Giraldo gratefully acknowledges the support of the Office of Naval Research through program element PE-0602435N.

References

[1] E. Kalnay, S. J. Lord, and R. D. McPherson, Maturity of operational numerical weather prediction: Medium range. *Bull. Amer. Meteor. Soc*, **79**, p. 2753–2769 (1998).

[2] Deutscher Wetterdienst, Wie gut sind unsere Wettervorhersagen?, *internal report* (2002).

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

[4] W. W. Grabowski, Entrainment and mixing in buoyancy-reversing convection with applications to cloud-top entrainment instability. *Quaterly Journal of the Royal Meteorological Society* **121**, 231–253 (1995).

[5] D. A. Randall, Conditional Instability of the First Kind Upside-Down. *J. Atmos. Sci.*, **37**, no. 1, p. 125–130 (1980).

[6] W. W. Grabowski, Cumulus entrainment, fine-scale mixing, and buoyancy reversal, *Quaterly Journal of the Royal Meteorological Society*, **119**, no. 513, p. 935–956 (1993).

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

[8] H. Johari, Mixing in thermals with and without buoyancy reversal. *J. Atmos. Sci.*, vol. **49**, no. 16, p. 1412–1426 (1992).

[9] R. Damiani, G. Vali, and S. Haimov, The structure of thermals in cumulus from airborne dual-Doppler radar observations. *Journal of the Atmospheric Sciences*, vol. **63**, Issue 5, pp.1432–1450 (2006).

[10] G. H. Bryan, J. C. Wyngaard, and J. M. Fritsch, Resolution requirements for the simulation of deep moist convection. *Monthly Weather Review*, vol. **131**, no. 10, p. 2394–2416 (2003).

[11] F. Giraldo and T. Warburton, A high-order triangular discontinuous Galerkin oceanic shallow water model, *Int. J. Numer. Methods Fl.*, 56, pp. 899–925 (2008).

[12] F. Giraldo and 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, pp. 3849–3877 (2008).

[13] B. Cockburn and C-W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16:173–261 (2001).

[14] M. Restelli and F. Giraldo, A conservative discontinuous Galerkin semi-implicit formulation for the Navier-Stokes equations in nonhydrostatic mesoscale modeling, *Journal of Scientific Computing*, Vol. 31, No. 3, pp. 2231–2257 (2009).

[15] F. Giraldo and M. Restelli, High-order semi-implicit time-integrators for a triangular discontinuous Galerkin oceanic shallow water model, *Int. J. Numer. Methods Fl.*, in press, DOI: 10.1002/fld.2118 (2010).

[16] M. Restelli, Semi-Lagrangian and semi-implicit discontinuous Galerkin methods for atmospheric modeling applications, *Ph.D. Thesis*, Politecnico di Milano (2007).

[17] A. Robert, Bubble convection experiments with a semi-implicit formulation of the Euler equations, *Journal of the Atmospheric Sciences* 50, pp. 1865–1873 (1993).

[18] J. M. Straka, R. B. Wilhelmson, L. J. Wicker, J. R. Anderson, K. K. Droegemeier, Numerical solutions of a non-linear density current: a benchmark solution and comparisons, *International Journal for Numerical Methods in Fluids* 17, pp. 1–22 (1993).

[19] J. Behrens, N. Rakowsky, W. Hiller, D. Handorf, M. Läuter, J. Päpke, K. Dethloff, amatos: Parallel adaptive mesh generator for atmospheric and oceanic simulation, *Ocean Modelling* 10, pp.171–183 (2005).

[20] P. K. Kundu, *Fluid Mechanics.* Academic Press, 638 pp. (1990).