Development and test of a multiple grids option in a mesoscale model

Dissertation zur Erlangung
des Doktorgrades der Naturwissenschaften
im Department Geowissenschaften der
Universität Hamburg

Vorgelegt von

Guido Schröder

aus

Diepholz

Hamburg
2007

Als Dissertation angenommen vom Department Geowissenschaften der Universität Hamburg

Aufgrund der Gutachten von Prof. Dr. K. Heinke Schlünzen
und Prof. Dr. Maria Lukáčová-Medviďová

Hamburg, den 14. Dezember 2007

Prof. Dr. Kay-Christian Emeis
Leiter des Department Geowissenschaften

Abstract

In this thesis the non-hydrostatic atmospheric mesoscale model METRAS is extended for employing multiple grids. By this the model domain can be decomposed into several fully coupled grids with different resolutions. The multiple grids allow to directly simulate some scale interactions.

An essential part of multiple grids models are robust numerical schemes when used on several coupled grids of different resolutions. To achieve this the advection schemes originally used in METRAS have been replaced by second order essentially non-oscillatory advection (ENO) methods, which are shown to cope with grid refinement. For momentum a plain ENO scheme is used, for scalars a newly developed flux integrated ENO scheme is applied.

Gravity waves are numerically dispersive, which results in wave reflection at the inner boundary to coarser grids. A theoretic dispersion relation is derived that confirms that gravity waves are the slower the coarser the grid as shown by numerical experiments. Simulations with different grid refinement factors reveal that coarsening factors larger than three are not recommendable due to wave reflection.

METRAS is an anelastic model, therefore, pressure is solved diagnostically from an elliptic equation. For this purpose two solvers, MULTIGRID and preconditioned Bi-CGSTAB, are modified for the needs of the multiple grids model. When multiple grids are applied, MULTIGRID converges faster than Bi-CGSTAB. Only when a single grid is used and $\Delta z/\Delta x << 1$, Bi-CGSTAB is superior to MULTIGRID.

Using the new second order ENO advection schemes and 3-level MULTIGRID, the multiple grids METRAS has been validated for three test cases. For comparison, each test case has been performed with a coarse grid, a fine grid and a refined grid (refinement factor of 3). The test cases for a rising bubble and a polynia show good agreement of the results on the fine grid and the refined grid, while the length of the simulation on the refined grid is three to four times shorter than on the fine grid. The improvement to the coarse grid and the CPU times justify grid refinement for these test cases. In the third test case the wind conditions at the America's Cup 2007 sailing site are simulated. For this case the results in the refined area, which is located over water, is not as good as with the homogeneous fine grid. Relocating the fine grid might improve the simulation with two grids.

Zusammenfassung

In dieser Dissertation wird das nicht-hydrostatische mesoskalige Modell METRAS um eine Option erweitert, mit der das Modellgebiet in mehrere Gitter verschiedener Auflösung zerlegt werden kann. Dadurch kann die Wechselwirkung von Prozessen verschiedener Skalen direkt simuliert werden.

Ein wichtiger Aspekt für das Mehrfachgittermodell sind robuste numerische Schemata, damit Gitter verschiedener Auflösung verwendet werden können. Dafür werden die in METRAS vorhandenen Advektionsschemata durch wesentlich nicht-oszillatorische (ENO) Schemata zweiter Ordnung ersetzt, die sich für Gitterverfeinerung eignen. Für die Impulsadvektion kommt dabei ein einfaches ENO-Schema zum Einsatz, für die skalaren Größen wurde ein flussintegriertes ENO-Schema entwickelt.

Schwerewellen sind numerisch dispersiv. Dies führt dazu, dass sie reflektiert werden können, wenn sie sich vom feinen ins grobe Gitter bewegen. Eine theoretische Dispersionsbeziehung wird abgeleitet, die bestätigt, dass Schwerewellen auf einem gröberen Gitter langsamer sind. Dies wird in numerischen Simulationen bestätigt. Simulationen mit verschiedenen Verfeinerungsfaktoren zeigen, dass kein Faktor größer als drei verwendet werden sollte.

METRAS ist ein anelastisches Modell, so dass der Druck diagnostisch durch eine elliptische Gleichung ermittelt wird. Vorhandene iterative Löser, Bi-CGSTAB und MULTIGRID, werden für Mehrfachgitter angepasst, wobei sich MULTIGRID für die Verwendung von Mehrfachgittern als der schnellere Löser herausstellt. Lediglich bei Verwendung von nur einem Gitter und $\Delta z/\Delta x << 1$ ist Bi-CGSTAB zu empfehlen.

Unter Verwendung der ENO Schemata für die Advektion und Level-3 MULTIGRID wird die Mehrfachgitterversion von METRAS für drei Testfälle validiert. Jeder Test wird auf einem groben, einem feinen und einem verfeinertem Gitter (Faktor 3) durchgeführt. Ein Warmluftblasen- und ein Polyniatestfall zeigen eine gute Übereinstimmung der Ergebnisse auf verfeinertem Gitter und feinem Gitter bei nur einem Drittel bis Viertel der Simulationszeit. Im dritten Testfall werden die Windverhältnisse für das Segelgebiet des America's Cup 2007 vor Valencia simuliert. Hier zeigt sich, dass die Simulation mit verfeinertem Gitter weniger gute Ergebnisse liefert als die Simulation mit feinem Gitter. Dies liegt vermutlich an der nicht idealen Lage des verfeinerten Gitters.

Contents

1 Introduction

The atmospheric flow is characterised by phenomena of different temporal and spatial scales, ranging from turbulent eddies being the smallest and most transient to planetary waves being the largest and most persistent phenomena. The different scales are linked by nonlinear processes leading to complex circulation systems. Modelling the atmosphere needs to take this into account in order to resemble nature as closely as possible. Despite continuously growing computer power it is still not feasible to explicitly simulate all atmospheric processes. Simulations are restricted by the resolution of the grid, and processes smaller than the grid width must be parameterised as subgrid-scale processes. Generally, it must be assumed that parameterisations are less accurate than the explicit simulation of the respective processes. Therefore, it is desirable to explicitly resolve as many scales as possible. On the other hand, increasing resolution is accompanied by increasing computational costs which makes it desirable to resolve not more scales than necessary. A solution for an efficient simulation is having different resolutions within the domain and resolving only those areas and processes that are relevant for the solution.

There are two philosophies on how the resolution of the domain should be distributed. One is that the ideal model should have a grid that has a high resolution only in regions where relevant small scale phenomena (e.g. clouds, frontal processes) are present that have significant influence on the large-scale flow. By this the overall accuracy of the simulation can be maintained. This philosophy leads to adaptive modelling. The second philosophy focuses on having a higher resolution in areas of interest. On the one hand these can be areas, where local effects are of great importance (e.g. precipitation or wind fields over mountainous terrain). On the other hand, e.g. in weather forecasts or for regional climate predictions, detailed information is relevant where people are affected, but it is not necessarily important

over the oceans or in sparsely populated areas. Different approaches are used to receive a high resolution within a coarse model domain.

Nesting methods

A standard approach to refine areas of interest are nesting methods. A model with a fine grid is nested into a model with a coarse grid. Both coarse and fine grid can be uniform (Fig. 1.1a), but this is not mandatory. Basically, there are two different methods, one-way and two-way nesting (Phillips and Shukla, 1973). For one-way nesting the simulation on the coarse grid is independent of the fine grid simulation and is used to prescribe the boundary conditions of the fine grid (e.g. Davies, 1976). Usually, this is done by pulling (nudging) the fine grid simulation to the coarse grid results with strong nudging at the boundaries and vanishing nudging in the model interior (Bungert, 2007). A disadvantage of this method is that for longer integration times the fine grid simulation and the coarse forcing may diverge. This can be overcome by two-way nesting (e.g. Zhang et al., 1986). Here, coarse and fine grid simulations run simultaneously with mutual influence. This can be achieved, e.g., by exchanging fluxes between coarse and fine grids. Two-way nesting has also been applied to regional climate modelling by Lorenz and Jacob (2005). Skamarock and Klemp (1993) have presented a non-hydrostatic model that can cope with several internal (two-way) nests used for adaptive refinement. They apply the adaptive method of Berger and Oliger (1984). Density current experiments and a supercell storm simulation show the suitability of their set-up for atmospheric simulations.

Nonuniform grids

A seemingly straightforward way to refine a certain area are nonuniform grids. They are usually based on a coordinate transformation that projects a grid that is deformed in physical space to a regular computational grid. In areas that need higher resolutions the coordinate lines accumulate. This can be done in each horizontal direction separately (Fig. 1.1b) or in both directions at once (Fig. 1.1c). In literature these configurations are referred to by both terms, nonuniform grid and stretched grid. In the vertical nonuniform grid spacing is standard, since the boundary layer needs a higher resolution than the free atmosphere. One of the first using stretched grids was Anthes (1970) who applied them to the shallow water equations. He also faced the problem of instabilities and accumulation of energy in the small scales

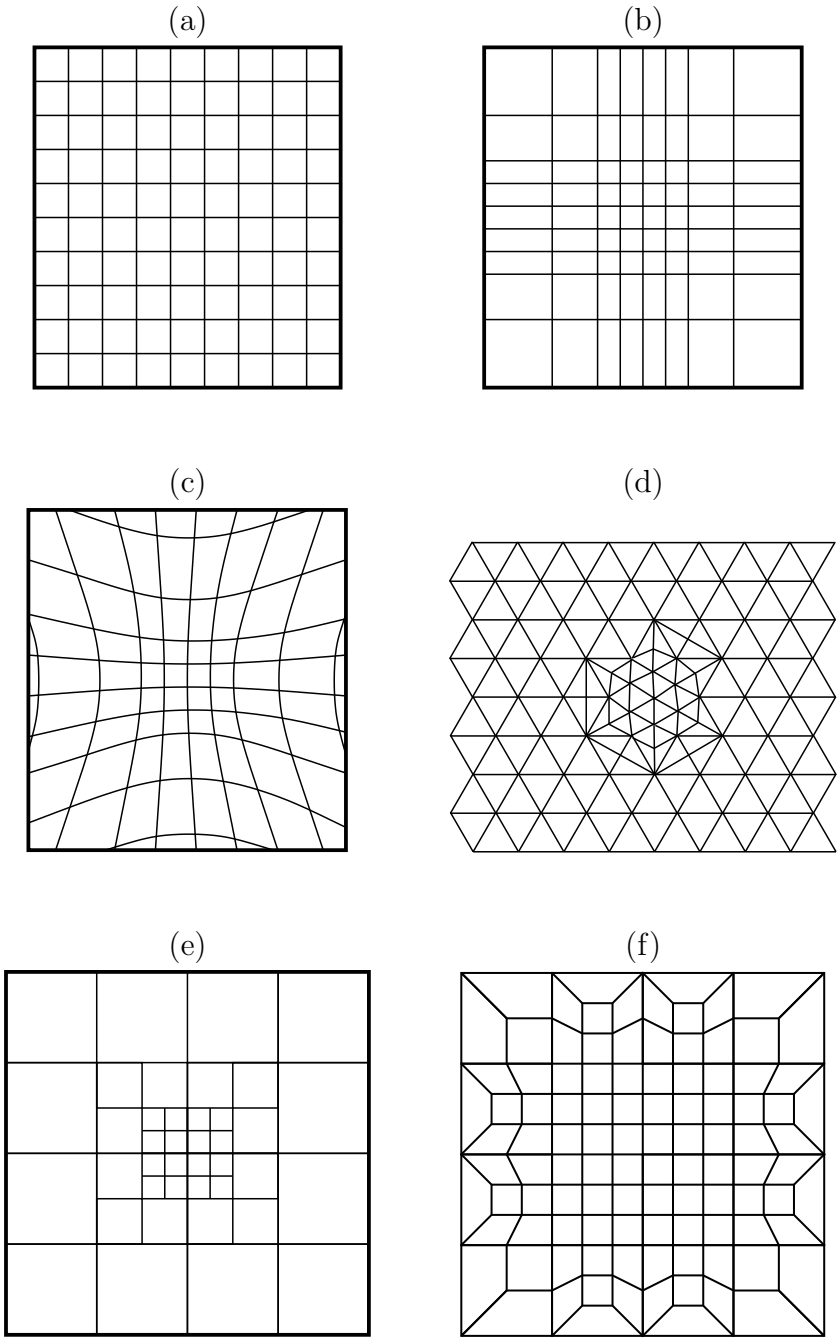

Figure 1.1: Horizontal sketches of different grid types. (a) uniform grid, (b,c) nonuniform (stretched) grid, (d) unstructured grid, (e) block-structured grid and (f) spectral element refinement by factor 3 (after Baer et al., 2006).

caused by the nonuniform grid.

To avoid numerical noise caused by the grid in conjunction with standard numerical schemes it is necessary to smoothly vary the mesh size. Schlünzen (1990) al-

lowed refinement in each spatial direction separately by a refinement factor of 1.2. With this configuration sea breezes over tidally flooded mudflats could be successfully simulated with high resolution over the coastline and coarser resolution at the boundaries.

Sündermann (1990) successfully applied the Brackbill-Saltzmann method (Brackbill and Saltzman, 1982) to calculate optimally stretched grids for the simulation of roll convection. This method is based on minimising a weighting function. The weighting function becomes small when the grid lines are close to being orthogonal and large gradients are resolved with many grid points. This guarantees stability of the numerical schemes and high accuracy in regions with large gradients. The changing grid is accounted for by additional fluxes between the computational grid cells. The disadvantage of the Brackbill-Saltzmann method is that the minimisation of the weighing function involves solving a large linear system of equations at each time step which is computationally expensive.

Dietachmayer and Droegemeier (1992) applied continuous dynamic grid adaption (CDGA) to meteorological modelling. For CDGA the grid is changed continuously which leads to additional terms in the equations instead of interpolating fields from the old to the new grid. This method is also based on the grid optimisation scheme of Brackbill and Saltzman (1982). The scheme was successfully tested with a simple two-dimensional passive scalar advection test as well as a rising warm bubble test.

Prusa and Smolarkiewicz (2003) used continuously changing stretched grids in a multi-scale anelastic model. Again this leads to additional terms in the governing equations due to the moving grid. The mesh size is advected by a special velocity with the monotone positive definite advection algorithm (MPDATA, Smolarkiewicz and Margolin, 1998). The advective velocity for the mesh size can be defined by the velocity of the phenomenon to be followed. The performance of the model presented by Prusa and Smolarkiewicz (2003) was analysed with a mesoscale test case (gravity waves) as well as a global scale test (Held-Suarez test). The advantage in comparison to CDGA is that it is not necessary to solve an optimisation problem which is computationally expensive. The disadvantage is that simply advecting the mesh size is not truly adaptive. Furthermore, physical parameterisations are not included in the model, and it is only a research model.

Stretched grids have also been used in global models for weather and climate forecasts (Fox-Rabinovitz et al., 2006). Fox-Rabinovitz et al. (1997), e.g., used a finite difference general circulation model in conjunction with stretched grids. They ap-

plied local stretching factors of less than 1.1, resulting in total stretching factors of 4 to 32. Their method needed the application of a polar filter and a Shapiro (1971) filter to stabilise simulations with nonuniform grids. With this configuration they compared the results on a fine uniform grid to results on nonuniform stretched grids for two different types of simulation. The first, a 10 day simulation, showed good agreement between the refined grid result and the uniformly fine grid result. For the second, a three year climate simulation, the means of the meteorological fields were very similar for the refined grid results and the uniformly fine grid result.

Spectral global models need the application of transformed spherical harmonics for grid refinement (Schmidt, 1977). Cote et al. (1998b) and Cote et al. (1998a) describe the Global Environmental Multiscale (GEM) model, which is applying this method and is used by the Canadian Meteorological Centre (CMC). Latitude and longitude grid lines can be concentrated at the area of interest which keeps them orthogonal. Forecast experiments yielded results with good agreement of fine grid and refined grid simulations in the refined area.

Another possibility of grid refinement in a global model is a combination of grid stretching and moving the pole of a rotated longitude-latitude coordinate system to the area of interest. When the longitude and latitude grid is uniform, at the pole the zonal distance between grid points is smaller than at the equator, leading to a higher zonal resolution at the pole. The method of moving the pole has been successfully applied, e.g. by Hardiker (1997) for the simulation of a hurricane.

Unstructured grids

Stretched grids lack flexibility when refining the grid. Unstructured grids circumvent this problem. While unstructured grids have commonly been used in aerodynamics, they have only recently become popular in atmospheric modelling (Fig. 1.1d).

Läuter et al. (2007) presented a shallow water model that is discretised on a triangular grid in the horizontal and applies the grid generator amatos (Behrens et al., 2005). Among other tests they validated their model with the simulation of Rossby-Haurwitz waves and the flow over a mountain. The results on a uniform resolution grid and the adaptive grid correspond well to each other and an efficient grid adaption to the flow was achieved.

Bacon et al. (2000) introduced the non-hydrostatic model OMEGA (Operational Multiscale Environment Model with Grid Adaptivity). It is based on triangular

grids as well and can adapt to orography and to flow features with large gradients. Parameterisation of cumulus convection is scale dependent which is achieved by a weighting factor. OMEGA is run operationally for hurricane forecasting (Gopalakrishnan et al., 2002; Bacon et al., 2007).

The German Weather Service (DWD) and the Max Planck Institute for Meteorology are currently developing a non-hydrostatic global model on an icosahedral grid (ICON, Bonaventura et al., 2005; Gassmann et al., 2007). It is supposed to be used for weather forecasts as well as climate simulation with the option of triangular grid refinement.

Block-structured grids

One of the disadvantages of unstructured grids is the difficult data handling and less efficient algorithms compared to structured grids (especially on vector computers). A good compromise between structured grids and unstructured grids are block-structured grids (Fig. 1.1e). These decompose the domain into blocks where the grid is uniform. Each block can be treated like a regular grid.

Schimmel (2002) used this approach to develop a fully compressible dynamical core that is based on the essentially non-oscillatory interpolation method (ENO, Harten et al., 1987). He showed in a density current experiment that despite a refinement factor of four explicit diffusion for stabilisation is not necessary due to the non-linearity of the implicit diffusion of the ENO scheme.

Fournier et al. (2004) presented a global model based on the spectral element method (SEAM). With this method the domain is decomposed into spectral elements (Fig. 1.1f). Each element is subdivided into a Gauss-Lobatto grid (e.g. 8x8 points) and Legendre cardinal functions are used as basis functions for each element. The advantage of this method is that recursive refinement is easily done. A disadvantage is that due to the small amount of basis functions in each element this method lacks efficiency on vector computers compared to traditionally gridded data. Baer et al. (2006) compared SEAM to the spectral NCAR (National Center for Atmospheric Research) model applying the same physical parameterisations and found good agreement between both model versions. Experiments with grid refinement did not change zonal mean fields in comparison to experiments without refinement, but the solution in the refined region was more detailed. However, the authors do not validate the refined solution against a uniform fine grid solution.

Jablonowski (2004) and Jablonowski et al. (2006) tested a block-structured hydrostatic global model with two-dimensional shallow water tests and a baroclinic wave test. There, a local refinement factor of two was applied and the model has an adaptive option. Again, good agreement between refined and uniform fine grid results was found.

The different models and methods described above all have advantages and disadvantages. Most of them are hydrostatic, others do not include physical parameterisation schemes for subgrid-scale processes. Some need explicit numerical diffusion to control the change in mesh size. Furthermore, there seems to be a general agreement not to apply refinement factors larger than four to avoid wave reflection (Zhang et al., 1986), even though this has not been analysed for waves more complex than advection waves.

It is the purpose of this work to analyse computational methods suitable for grid refinement as well as to estimate the restriction the numerical dispersion of gravity waves imposes on grid refinement. The knowledge gained from these studies is used to add a multiple grids option with grid refinement to the non-hydrostatic model METRAS. METRAS includes parameterisation schemes necessary to adequately simulate the atmosphere. In Chapter 2 METRAS is described and the changes necessary for a multiple grids option are stated. Chapters 3 to 5 are direct copies of published (Chapter 3) papers or papers that are intended for publication (Chapters 4 and 5). In Chapters 3 and 4 new advection schemes are introduced and applied to METRAS. These schemes preserve large gradients and can handle grid refinement without adding explicit diffusion. In Chapter 5 the numerical dispersion of gravity waves is analysed and a recommendation is given, which refinement factors are justifiable in the context of gravity wave dispersion. Chapter 6 focuses on the iterative elliptic solver for pressure. The problem that grid refinement easily slows down iterative solvers is outlined and a fast solution based on MULTIGRID is presented. In Chapter 7 the new METRAS version with multiple grids is tested. A warm bubble test for the dynamical core and two realistic tests with the full model are presented. Chapter 8 gives conclusions and an outlook to future work.

2 Concept of the multiple grids METRAS

The Mesoscale Transport- and Stream model METRAS has been developed at the University of Hamburg and is described in Schlünzen (1990) and Schlünzen et al. (1996). It is able to simulate air flow over steep terrain (Niemeier and Schlünzen, 1993) and was validated by Schlünzen (1996) and Dierer (1997). It has been used for various applications. METRAS was coupled to a sea-ice model to calculate the cyclone sea-ice interaction in the Fram Strait (Dierer et al., 2005; Dierer and Schlünzen, 2005). Furthermore, it was used to calculate pollen transport (Schueler et al., 2005; Schueler and Schlünzen, 2006) and pollutant deposition in conjunction with a chemistry model (Trukenmüller et al., 2004; Schlünzen and Meyer, 2007). In all these applications one resolution or one-way nesting was used. The corresponding single grid METRAS version is described in this chapter in Sections 2.1 to 2.3. These sections explain the main parts of the model and their state prior to this work.

For this thesis METRAS is transformed into a model with multiple grids that are fully coupled. Section 2.4 describes the changes in the grid configuration and in the numerical integration that were made for this work. For using multiple grids with different resolution, robust and computationally inexpensive numerical schemes are necessary. These are not described in this chapter but in Chapters 3, 4 and 6.

2.1 Model equations

The METRAS model equations are based on the Reynolds-averaged Navier-Stokes equations in flux form. Before they are Reynolds-averaged, they are converted from Cartesian (x, y, z) into surface-following coordinates $(\dot{x}, \dot{y}, \dot{z})$:

$$\begin{aligned} \dot{x} &= \dot{x}(x) \\ \dot{y} &= \dot{y}(y) \\ \eta(z) = z_t \tfrac{z - z_s(x,y)}{z_t - z_s(x,y)} & \\ \dot{z}(x,y,z) &= z(x,y,\eta) \end{aligned} \tag{2.1}$$

where $z_t = \text{const}$ is the height of the upper model boundary, and $z_s(x, y)$ the height of the lower boundary corresponding to the topography height. Given the Cartesian velocity components (u, v, w) the contravariant velocity components $(\dot{u}, \dot{v}, \dot{w})$ are created in the new coordinate system. These are normal to the coordinate surfaces:

$$\begin{aligned} \dot{u} &= \frac{\partial \dot{x}}{\partial x} u \\ \dot{v} &= \frac{\partial \dot{y}}{\partial y} v \\ \dot{w} &= \frac{\partial \dot{z}}{\partial x} u + \frac{\partial \dot{z}}{\partial y} v + \frac{\partial \dot{z}}{\partial z} w. \end{aligned}$$

These are the velocities that advect through the surfaces of a grid cell after discretisation along the new coordinates $(\dot{x}, \dot{y}, \dot{z})$. An important transformation constant is

$$\alpha^* = \frac{\partial x}{\partial \dot{x}} \frac{\partial y}{\partial \dot{y}} \frac{\partial z}{\partial \dot{z}} \tag{2.2}$$

which can be interpreted as the volume of a grid cell.

The METRAS equations are derived from the compressible equations which are simplified by the anelastic and Boussinesq approximation (Ogura and Phillips, 1962; Dutton and Fichtl, 1969) to eliminate time step restricting sound waves. This leads to replacing the density $\overline{\rho}(x, y, z) = \tilde{\rho}(x, y, z) + \rho_0(z)$ by the background density $\rho_0(z)$ in most terms. Only in the buoyancy term the whole density $\overline{\rho}(x, y, z)$ is needed including the mesoscale part $\tilde{\rho}(x, y, z)$. Furthermore, the continuity equation is no longer dependent on time and becomes a diagnostic equation.

The resulting system of equations is Reynolds-averaged and contains three prognostic equations for momentum, prognostic equations for potential temperature $\overline{\theta}$, specific humidity $\overline{q_v}$, cloud liquid water content $\overline{q_l}$ and rain water content $\overline{q_r}$, and diagnostic equations for the mesoscale density $\tilde{\rho}$, and pressure p. Here $\overline{\cdot}$ indicates Reynolds-averaging.

The momentum equations using the Reynolds-averaged velocity components $\overline{u}$,$\overline{v}$ and $\overline{w}$ can be written as

$$\frac{\partial \alpha^* \vec{m}}{\partial t} = -\mathcal{A}(\alpha^* \vec{m}) - \vec{\mathcal{P}}(p) - \vec{\mathcal{B}}(\overline{\rho}) + \vec{\mathcal{C}}(\overline{u}, \overline{v}, \overline{w}) - \vec{F}_{\vec{m}} \tag{2.3}$$

where $\vec{m}$ is the momentum vector

$$\vec{m} \;=\; (\overline{u}\rho_0, \overline{v}\rho_0, \overline{w}\rho_0)^\tau$$

with $^\tau$ indicating the transpose of the vector. The following operators and definitions have been used:

$$\begin{aligned}
\text{Advection} \;:\; & \mathcal{A}(\chi) = \frac{\partial}{\partial \dot{x}}(\overline{\dot{u}}\chi) + \frac{\partial}{\partial \dot{y}}(\overline{\dot{v}}\chi) + \frac{\partial}{\partial \dot{z}}(\overline{\dot{w}}\chi) \\
\text{Pressure gradient} \;:\; & \vec{\mathcal{P}}(p) = \begin{pmatrix} \alpha^* \dfrac{\partial \dot{x}}{\partial x}\dfrac{\partial p}{\partial \dot{x}} + \alpha^* \dfrac{\partial \dot{z}}{\partial x}\dfrac{\partial p}{\partial \dot{z}} \\ \alpha^* \dfrac{\partial \dot{y}}{\partial y}\dfrac{\partial p}{\partial \dot{y}} + \alpha^* \dfrac{\partial \dot{z}}{\partial y}\dfrac{\partial p}{\partial \dot{z}} \\ \alpha^* \dfrac{\partial \dot{z}}{\partial z}\dfrac{\partial p}{\partial \dot{z}} \end{pmatrix} \\
\text{Coriolis force} \;:\; & \vec{\mathcal{C}}(\overline{u}, \overline{v}, \overline{w}) = \begin{pmatrix} f\rho_0\alpha^*\overline{v} - f'\rho_0\alpha^*\overline{w} \\ -f\rho_0\alpha^*\overline{u} + f'\rho_0\alpha^*\overline{w} \\ \rho_0\alpha^* f'(\overline{u} - \overline{v}) \end{pmatrix} \\
\text{Buoyancy force} \;:\; & \vec{\mathcal{B}}(\rho) = (0, 0, \rho\alpha^* g)^\tau \\
\text{Turbulent diffusion} \;:\; & \vec{F}_{\vec{m}} = (F_{\overline{u}}, F_{\overline{v}}, F_{\overline{w}})^\tau
\end{aligned} \tag{2.4}$$

Here, $f = 2\Omega\sin(\varphi)$ and $f' = 2\Omega\cos(\varphi)$ are the Coriolis parameters with earth angular momentum Ω and the latitude φ. Note that an operator applied to a vector (e.g. $\mathcal{A}(\vec{m})$) means applying the operator to each component separately, which results in a new vector. While the third component of the pressure gradient vector above is given for the real vertical momentum $\rho_0 w$, the pressure gradient is actually calculated for the transformed vertical velocity in METRAS:

$$\begin{aligned}
\dot{\mathcal{P}}_3 = \alpha^* & \left[\left(\frac{\partial \dot{z}}{\partial x}\right)^2 + \left(\frac{\partial \dot{z}}{\partial y}\right)^2 + \left(\frac{\partial \dot{z}}{\partial z}\right)^2 \right] \frac{\partial p}{\partial \dot{z}} \\
& + \alpha^* \left[\frac{\partial \dot{x}}{\partial x}\frac{\partial \dot{z}}{\partial x}\frac{\partial p}{\partial \dot{x}} + \frac{\partial \dot{x}}{\partial x}\frac{\partial \dot{z}}{\partial x}\frac{\partial p}{\partial \dot{x}} \right]
\end{aligned} \tag{2.5}$$

After the calculation of the pressure gradient the transformed vertical velocity is transformed to the real vertical velocity.

Additionally, like density, pressure and potential temperature are also separated into a large-scale and mesoscale part to use typical physical balances in the equations:

$$p(x, y, z) \;=\; p_0(z) + p_g(x, y, z) + p_1(x, y, z) + p_2(x, y, z) \tag{2.6}$$

$$\theta(x, y, z) \;=\; \theta_0(z) + \tilde{\theta}(x, y, z) \tag{2.7}$$

The pressure terms p_0 and p_1 result from hydrostatic balance with the density terms ρ_0 and $\tilde{\rho}$:

$$\frac{\partial p_0}{\partial \dot{z}} = -g\rho_0 \frac{\partial z}{\partial \dot{z}} \tag{2.8}$$

$$\frac{\partial p_1}{\partial \dot{z}} = -g\tilde{\rho} \frac{\partial z}{\partial \dot{z}} \tag{2.9}$$

Therefore, the background pressure p_0 contains the major pressure portion. p_2 is a dynamic pressure that acts to keep the momentum free of divergence. The geostrophic pressure p_g which contains the large scale horizontal pressure gradient is not directly calculated. It is assumed to be in balance with the geostrophic wind:

$$u_g = -\frac{1}{\rho_0 f}\left(\frac{\partial \dot{y}}{\partial y}\frac{\partial p_g}{\partial \dot{y}} + \frac{\partial \dot{z}}{\partial y}\frac{\partial p_g}{\partial \dot{z}}\right) \tag{2.10}$$

$$v_g = +\frac{1}{\rho_0 f}\left(\frac{\partial \dot{x}}{\partial x}\frac{\partial p_g}{\partial \dot{x}} + \frac{\partial \dot{z}}{\partial x}\frac{\partial p_g}{\partial \dot{z}}\right) \tag{2.11}$$

The background pressure is not needed in the equations for the geostrophic wind, because p_0 only depends on z. The geostrophic wind is derived from observations or reanalysis data for case studies, or it is prescribed according to the settings of idealised experiments.

Using hydrostatic and geostrophic balance, the momentum equations result into:

$$\frac{\partial \alpha^* \vec{m}}{\partial t} = -\mathcal{A}(\alpha^* \vec{m}) - \vec{\mathcal{P}}(p_1 + p_2) - \vec{\mathcal{B}}(\tilde{\rho}) + \mathcal{C}(\overline{u} - u_g, \overline{v} - v_g, \overline{w}) - \vec{F}_{\vec{m}} \tag{2.12}$$

Note that the term $\vec{\mathcal{B}}(\tilde{\rho})$ is only kept for completeness. Actually, it cancels out with the pressure term $\vec{\mathcal{P}}_3(p_1)$ in the equation for vertical momentum according to equation (2.9). Consequently, neither term is calculated in the model.

The conservation of a scalar quantity χ is given by

$$\frac{\partial \rho_0 \alpha^* \chi}{\partial t} = -\mathcal{A}(\rho_0 \alpha^* \chi) - F_\chi - S_\chi \tag{2.13}$$

Again, F_χ denotes turbulent diffusion of the respective quantity. Sources and sinks (S_χ) for the different quantities may be:

quantity χ	sources and sinks
$\overline{\theta}$	divergence of radiative and surface fluxes, cloud effects
$\overline{q_v}, \overline{q_c}$	surface fluxes, cloud effects
$\overline{q_r}$	sedimentation, cloud effects

The continuity equation in anelastic form is:

$$0 = \frac{\partial}{\partial \dot{x}}(\overline{\dot{u}}\rho_0\alpha^*) + \frac{\partial}{\partial \dot{y}}(\overline{\dot{v}}\rho_0\alpha^*) + \frac{\partial}{\partial \dot{z}}(\overline{\dot{w}}\rho_0\alpha^*) = \mathcal{A}(\rho_0\alpha^*) = \vec{\nabla} \bullet \vec{m} \tag{2.14}$$

For simplicity, the operator $\vec{\nabla} \bullet \vec{m}$ is evaluated in Cartesian coordinates.

The perfect gas law for a gas of temperature T with the gas constant R is

$$\rho = \frac{p}{RT}$$

The gas law is linearised about the background state under inclusion of the specific humidity, cloud and rain water content (Schlünzen et al., 1996)

$$\frac{\tilde{\rho}}{\rho_0} = -\frac{\tilde{\theta}}{\theta_0} + \frac{c_v}{c_p}\frac{p_1 + p_2}{p_0} - \left(\frac{R_v}{R_d} - 1\right)\overline{q_v} + \overline{q_c} + \overline{q_r} \tag{2.15}$$

with the gas constants for dry air (R_d) and water vapour (R_v). c_v is the specific heat of air at constant volume, c_p is the specific heat of air at constant pressure.

Equations (2.12) to (2.15) form the mathematical model of METRAS.

2.2 Model physics - parameterisations

In this section a short overview on the applied parameterisation schemes is given; they have not been changed for this work. Details can be found in Schlünzen et al. (1996). It should be noted that most parameterisations are only suitable for a limited range of scales. Using the same parameterisations on coupled grids strongly limits the refinement factors that can be used. Currently, scale dependent parameterisations for METRAS are investigated by Bohnenstengel (2007) and Ries (2009).

Turbulent flux divergences (F_X) and the sources and sinks (S_X) mentioned in the previous section are calculated by parameterisations. The physical processes are:

- turbulence

  Turbulent fluxes are calculated with a first order closure. For the exchange coefficients a mixing length approach is used for stable stratifications with a mixing length after Blackadar (1962) and stability functions after Dyer (1974). For unstable stratifications the counter-gradient scheme of Lüpkes and Schlünzen (1996) is applied. At the surface ($z < 10m$) the Monin-Obhukov similarity theory (e.g. Stull, 1988) is assumed.

- radiation

  Without liquid water in the atmosphere only the surface energy balance is calculated, and an empirical cooling rate for the atmosphere is assumed. With

the presence of cloud and rain droplets a two-stream approximation for short- and longwave radiation is applied, where absorption depends on the droplet size (Schlünzen et al., 1996).

- cloud microphysics

  Cloud microphysics is parameterised after Kessler (1969). The scheme does not contain the ice phase. The main processes included are condensation, evaporation, autoconversion (colliding cloud droplets form rain drops), acreation (rains drops collect cloud droplets and thereby grow) and sedimentation.

- surface representation

  To account for subgrid-scale surface heterogeneities, a blending height concept (von Salzen et al., 1996; Schlünzen and Katzfey, 2003) is available. With this scheme, surface fluxes are calculated for each subgrid surface class separately and then aggregated per grid cell at the blending height.

2.3 Grid structure and numerical integration for a single grid

In this section the grid structure is described and the numerical integration is outlined for the original single grid version of METRAS. Besides the numerical schemes these are parts of the model that need to be changed for the new multiple grids version. These changes are described in Section 2.4.

The model equations are spatially discretised on an Arakawa-C-Grid (Mesinger and Arakawa, 1976). The wind vector components $\overline{u}$,$\overline{v}$,$\overline{w}$ are shifted eastward, northward, upward, respectively, against the scalar quantities. $\dot{x}$ is discretised on vector levels by $\dot{x}_{i+1/2}$, $\dot{y}$ by $\dot{y}_{j+1/2}$ and $\dot{z}$ by $\dot{z}_{k+1/2}$ (see Eq. (2.1) for the transformation). The vector levels mark the boundaries of the scalar boxes. The location of the scalar levels is given by

$$\begin{aligned} \dot{x}_i &= \frac{\dot{x}_{i-1/2} + \dot{x}_{i+1/2}}{2} \\ \dot{y}_j &= \frac{\dot{y}_{j-1/2} + \dot{y}_{j+1/2}}{2} \\ \dot{z}_k &= \frac{\dot{z}_{k-1/2} + \dot{z}_{k+1/2}}{2}. \end{aligned}$$

The scalar levels mark the boundaries for u in x-direction, v in y-direction and w in z-direction. Fig. 2.1 and Fig. 2.2 show the staggered grid in the xy-plane. Grid staggering has the advantage that the accuracy is increased. For calculating the scalar advection fluxes, the cell boundary normal wind is automatically given by the three wind components. Due to the staggered grid the pressure is located at the cell boundaries of wind cells with no need for further interpolation when the pressure gradient is calculated. Again this increases the accuracy.

Time integration is also staggered. Scalar quantities are integrated to time levels t^n, the velocity components to $t^{n+1/2}$ (Fig. 2.3).

The operators are discretised in flux form with a finite volume approach to conserve momentum and the thermodynamic quantities. Advection and turbulent diffusion ($\vec{F}_{\vec{m}}$) are solved centred in space and with a second order Adams-Bashforth scheme in time for momentum. For numerical stability and to remove small scale disturbances caused by nonlinear interaction and limited model resolution, a Shapiro (1971) seven-point filter is used for the momentum components in the horizontal. For scalar quantities advection is solved with a first order upstream scheme and forward in time. The advection of background fields like θ_0 is calculated centred in space. Scalar diffusion and pressure gradient are solved centred in space and forward in time. Coriolis force is solved forward in time. When vertical diffusion imposes a too strong constraint on the time step, it is not solved as described above but with the

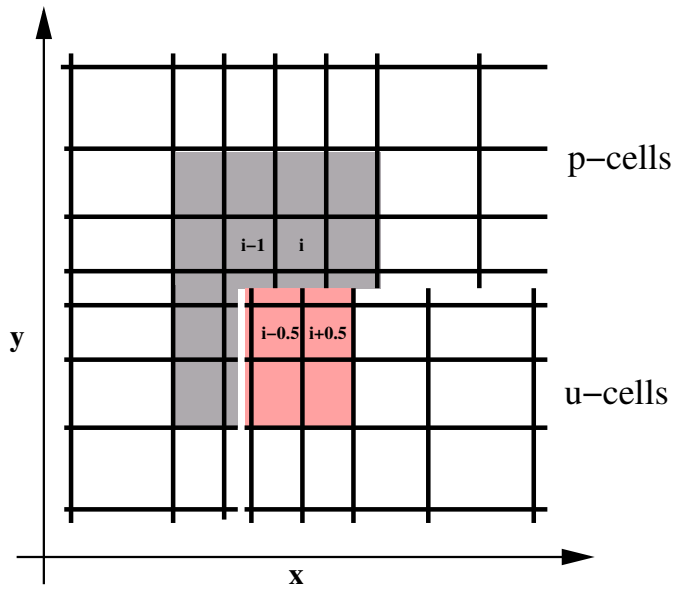

Figure 2.1: Staggered Arakawa-C-grid in the xy-plane. Here the cells of u and p are shown (staggered in west-east direction, index i).

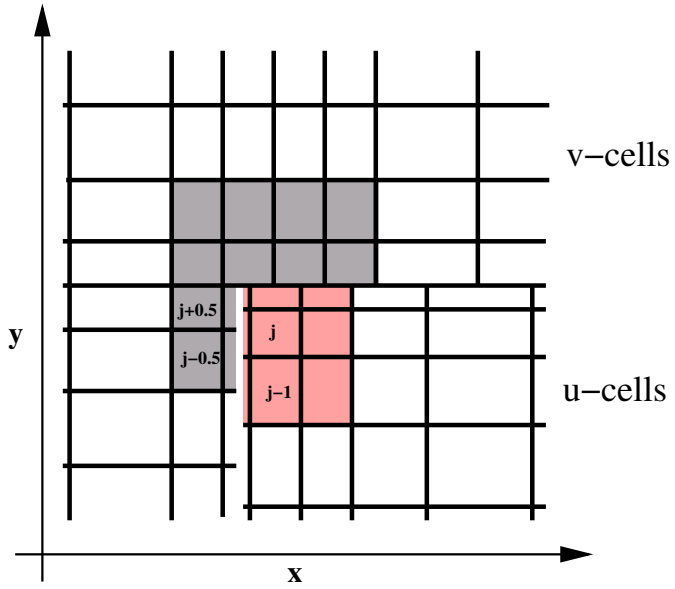

Figure 2.2: Staggered Arakawa-C-grid in the xy-plane. Here the cells of u and v are shown (staggered in west-east and south-north direction, indices i and j, respectively; only index j is shown).

implicit Crank-Nicholson scheme.

In the following, one integration is outlined at a point where the scalar quantities are known at t^n and the velocity components at $t^{n-0.5}$. The only exception is p_2 which is known at t^{n-1}. Note that the advection operator is time dependent with

$$\mathcal{A}^{n+1/2}(\chi) = \frac{\partial}{\partial \dot{x}}(\overline{\dot{u}}^{n+1/2}\chi) + \frac{\partial}{\partial \dot{y}}(\overline{\dot{v}}^{n+1/2}\chi) + \frac{\partial}{\partial \dot{z}}(\overline{\dot{w}}^{n+1/2}\chi).$$

The following steps are taken:

- Calculate the preliminary momentum fields (denoted by $\hat{\cdot}$; preliminary, because p_2^n is not known yet):

$$\begin{aligned}\rho_0\alpha^*\hat{\overline{u}}^{n+0.5} = \rho_0\alpha^*\overline{u}^{n-0.5} + \big[& -1.5\mathcal{A}^{n-0.5}(\rho_0\alpha^*\overline{u}^{n-0.5}) \\ & +0.5\mathcal{A}^{n-1.5}(\rho_0\alpha^*\overline{u}^{n-1.5}) \\ & -\mathcal{P}_1(p_1^n + p_2^{n-1}) \\ & +\mathcal{C}_1(\overline{u}^{n-0.5} - u_g, \overline{v}^{n-0.5} - v_g, \overline{w}^{n-0.5}) \\ & -1.5F_{\overline{u}}^{n-0.5} + 0.5F_{\overline{u}}^{n-1.5}\big]\Delta t\end{aligned} \tag{2.16}$$

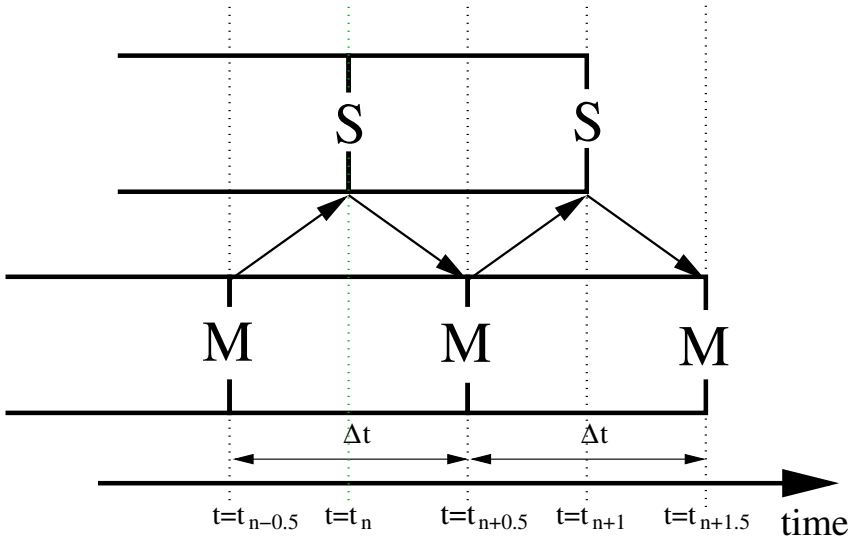

Figure 2.3: Staggering of time integration of scalar (S) and momentum quantities (M). The open boxes mark the period of one time step. At the end of the time step the respective quantity (momentum or scalar) is updated.

$$\begin{aligned}\rho_0\alpha^*\hat{\overline{v}}^{n+0.5} = \rho_0\alpha^*\overline{v}^{n-0.5} + \big[& -1.5\mathcal{A}^{n-0.5}(\rho_0\alpha^*\overline{v}^{n-0.5}) \\ & +0.5\mathcal{A}^{n-1.5}(\rho_0\alpha^*\overline{v}^{n-1.5}) \\ & -\mathcal{P}_2(p_1^n + p_2^{n-1}) \\ & +\mathcal{C}_2(\overline{u}^{n-0.5} - u_g, \overline{v}^{n-0.5} - v_g, \overline{w}^{n-0.5}) \\ & -1.5F_{\overline{v}}^{n-0.5} + 0.5F_{\overline{v}}^{n-1.5}\big]\Delta t\end{aligned} \tag{2.17}$$

$$\begin{aligned}\rho_0\alpha^*\hat{\overline{w}}^{n+0.5} = \rho_0\alpha^*\overline{w}^{n-0.5} + \big[& -1.5\mathcal{A}^{n-0.5}(\rho_0\alpha^*\overline{w}^{n-0.5}) \\ & +0.5\mathcal{A}^{n-1.5}(\rho_0\alpha^*\overline{w}^{n-1.5}) \\ & -\mathcal{P}_3(p_2^{n-1}) \\ & +\mathcal{C}_3(\overline{u}^{n-0.5}, \overline{v}^{n-0.5}, \overline{w}^{n-0.5}) \\ & -1.5F_{\overline{w}}^{n-0.5} + 0.5F_{\overline{w}}^{n-1.5}\big]\Delta t\end{aligned} \tag{2.18}$$

- Calculate the correction of dynamic pressure $\hat{p}$ with the preliminary momentum vector $\vec{\hat{m}} = (\rho_0\hat{\overline{u}}, \rho_0\hat{\overline{v}}, \rho_0\hat{\overline{w}})^\tau$ from the continuity equation. This involves solving a Poisson equation:

$$\vec{\nabla} \bullet (\vec{\mathcal{P}}(\hat{p})) = \vec{\nabla} \bullet \vec{\hat{m}}^{n+0.5}/(\Delta t) \tag{2.19}$$

  Note that for simplicity Eq. (2.19) is written in Cartesian coordinates. Different methods for the solution of $\hat{p}$ are compared in Chapter 6.

- Correct the dynamic pressure with $p_2^n = p_2^{n-1} + \hat{p}$. Note that the pressure p_2 is defined at $t = t^n$, because p_2 acts on momentum in the time interval

from $t^{n-0.5}$ to $t^{n+0.5}$. The new momentum field is calculated by applying the gradient of the perturbation pressure $\hat{p}$:

$$\vec{m}^{n+0.5} = \vec{\hat{m}}^{n+0.5} - \vec{\mathcal{P}}(\hat{p})\Delta t. \tag{2.20}$$

At this stage the new momentum field is known at $t^{n+0.5}$ and can be used for advecting the scalar quantities:

- Calculate the prognostic scalar (thermodynamic quantities) at the new time step:

$$\rho_0\alpha^*\chi^{n+1} \quad = \rho_0\alpha^*\chi^n + \left[\quad -\mathcal{A}^{n+0.5}(\rho_0\alpha^*\chi^n) - S_\chi^n \quad \right]\Delta t.$$

- Calculate the mesoscale density $\tilde{\rho}^{n+1}$ and the hydrostatic pressure p_1^{n+1} with (2.15) and (2.9). Actually, these two equations cannot be solved independently from each other, and an iteration of (2.15) and (2.9) would be required. However, for p_1 and $\tilde{\rho}$ the changes per time step are small, so that the iteration is not necessary.

Now the velocities can be forwarded again to $t^{n+1.5}$ by repeating the process for momentum as described above.

2.4 Grid structure and numerical integration for multiple grids

The new METRAS version developed in this thesis can operate with several nested (refined) grids of different resolution. Besides the necessity to use robust and computationally inexpensive numerical schemes (Chapters 3, 4 and 6) the numerical integration as well as the grid structure need to be changed as well. This is described in this section.

When multiple grids are used, it is necessary that each subgrid is surrounded by only one higher ranking (coarser) grid (Fig. 2.4). The vertical grid needs not to be changed, since it is generally of much higher resolution in a mesoscale model. Horizontal refinement factors must be odd numbers to be able to conserve grid staggering (Fig. 2.5). Fig. 2.6 and Fig. 2.7 show an example grid with the refinement factors in x- and y-direction $N_x = N_y = 3$. It is important to note that

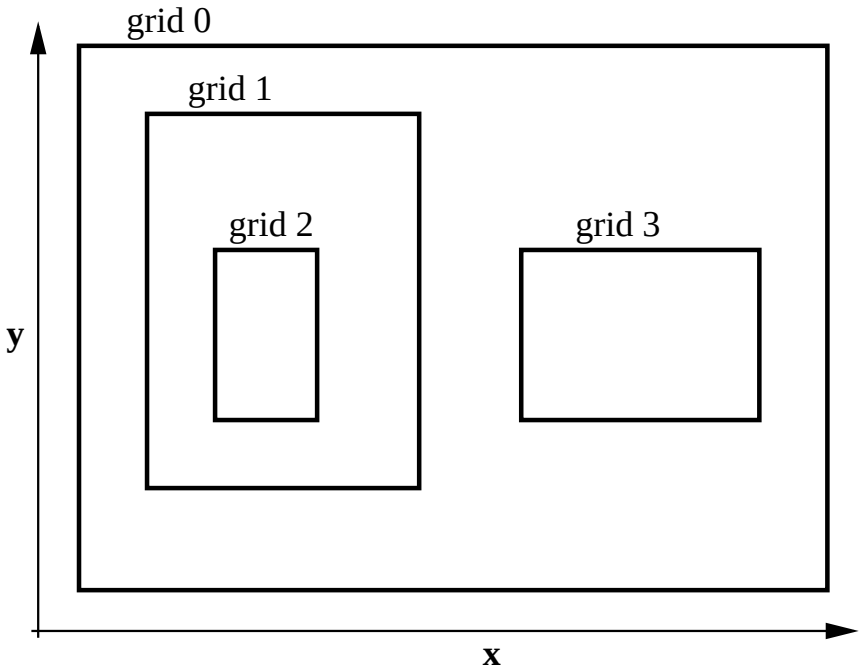

Figure 2.4: Grid hierarchy. Main grid 0 with three refined subgrids. Refinement is only possible in the horizontal, a refined grid can only be adjacent to either a coarser or a finer grid, not both at once.

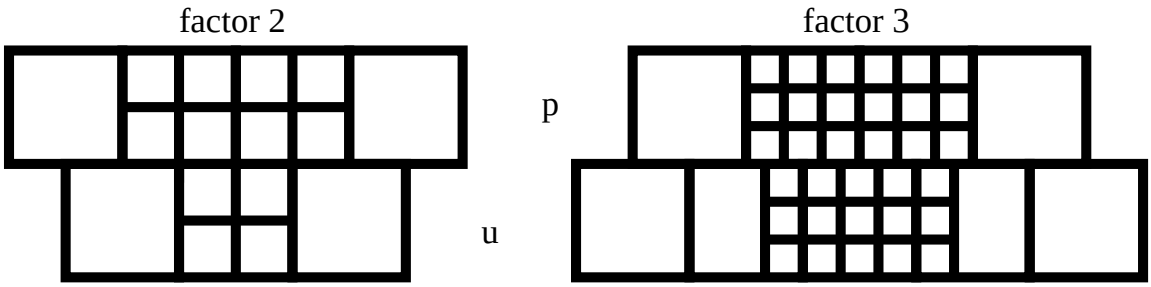

Figure 2.5: Schematic clarification of necessity of odd refinement factors for a staggered grid. With even factors the fine grid is not staggered anymore.

$(n_x + 1)$ u-vector grid cells are refined due to staggering if n_x scalar coarse grid cells are refined to a fine grid in x-direction. On the fine grid the outermost u cell, i.e. the cell of the boundary normal velocity, has an extension in x-direction of $\frac{N_x+1}{2}$ times the extension of the remaining fine cells. For a refinement factor of 3 the outermost u cell (in $\dot{x}$-direction) is twice as large as the inner u-cells (Fig. 2.6).

For practical applications refinement factors larger than 3 are not recommended due to wave dispersion (Chapter 5, Schroeder and Schlünzen, 2007b). For larger refinement factors too much energy of gravity waves with scales close to the coarse grid resolution is reflected when the waves are travelling from the fine to the coarse

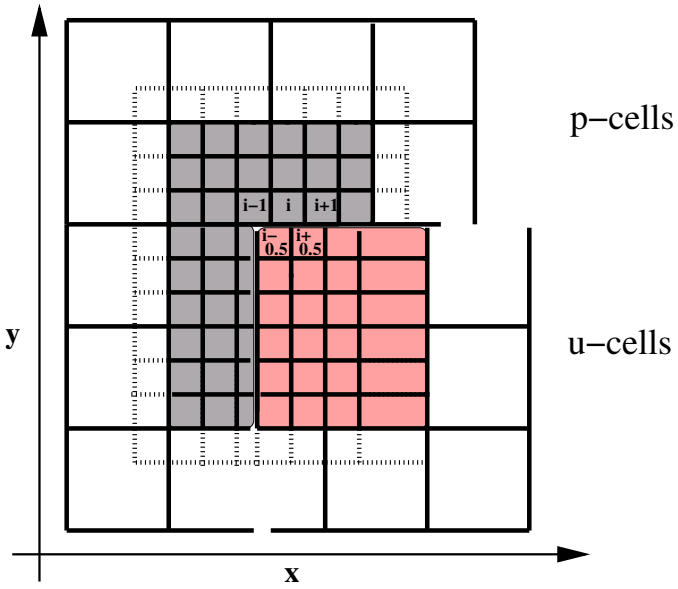

Figure 2.6: Staggered Arakawa-C-grid in the xy-plane with a refined subgrid. Here the cells of u and p are shown (staggered in west-east direction in index i).

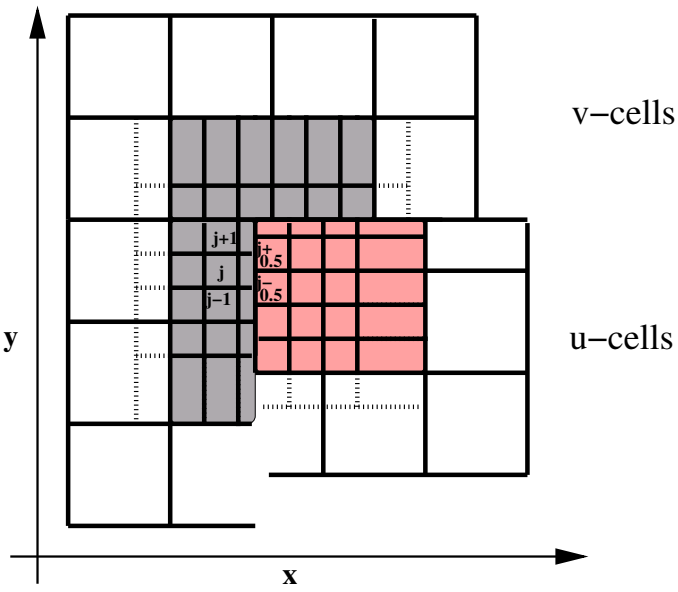

Figure 2.7: Staggered Arakawa-C-grid in the xy-plane with a refined subgrid. Here the cells of u and v are shown (staggered in west-east and south-north-direction in indices i and j, respectively; only index j is shown).

grid. The waves may be trapped within the fine grid. Furthermore, the current parameterisation schemes cannot be used for too large ranges of resolution.

Additional numerical problems occur even with a grid refinement factor of 3, if the numerical schemes are not suitable for stepwise changes in grid spacing. For

example, central differences tend to produce spurious oscillations. Therefore, new advection schemes based on the essentially non-oscillatory method are implemented and tested. Chapters 3 and 4 explain the new advection schemes for momentum and scalar quantities. These schemes are able to cope with grid refinements.

Another important topic when using stepwise changes in grid spacing is the speed of the solvers on multiple grids. This is crucial for iterative schemes that need rapid convergence. For METRAS the most time consuming iterative scheme is the solution of the elliptic equation (2.19) that arises from demanding a divergence free momentum field. A perturbation pressure is calculated to correct the divergences caused by the processes acting on momentum. In Chapter 6 different methods to solve the perturbation pressure $\hat{p}$ for the dynamic pressure p_2 are compared.

The time integration on multiple grids consists of the same steps as on a single grid (Section 2.3). The difference is that on the fine grid there may be a smaller time step.

Fixed time step refinement factor

The time integration is outlined here for a fixed time step refinement factor 3 (Fig. 2.8). This factor was chosen in consistency with the grid refinement factor of 3, but other refinement factors are possible. The time steps on the coarse grid are counted as $t^n, t^{n+1}, ...$ for scalars and as $t^{n-3/6}, t^{n+3/6}, ...$ for momentum. For the fine grid the time steps are counted as $t^{n-2/6}, t^n, t^{n+2/6}, ..$ for scalars and as $t^{n-3/6}, t^{n-1/6}, t^{n+1/6}, ...$ for momentum.

- ...
- $t = t^{n-1/6}$

  At this time one momentum step has been performed on the fine grid from $t = t^{n-3/6}$ to $t = t^{n-1/6}$ (small red box to the right of $t = t^{n-3/6}$, Fig. 2.8). During this step the coarse momentum field was held constant at $t = t^{n-3/6}$ and the (fine grid) momentum fluxes $\vec{F}_m^{n-1/6}$ at the internal boundaries are saved. The momentum field is made divergence free by having solved the Poisson equation on the fine grid with Dirichlet boundary conditions at the internal boundaries (i.e. the coarse grid values are held constant).
- $t = t^n$

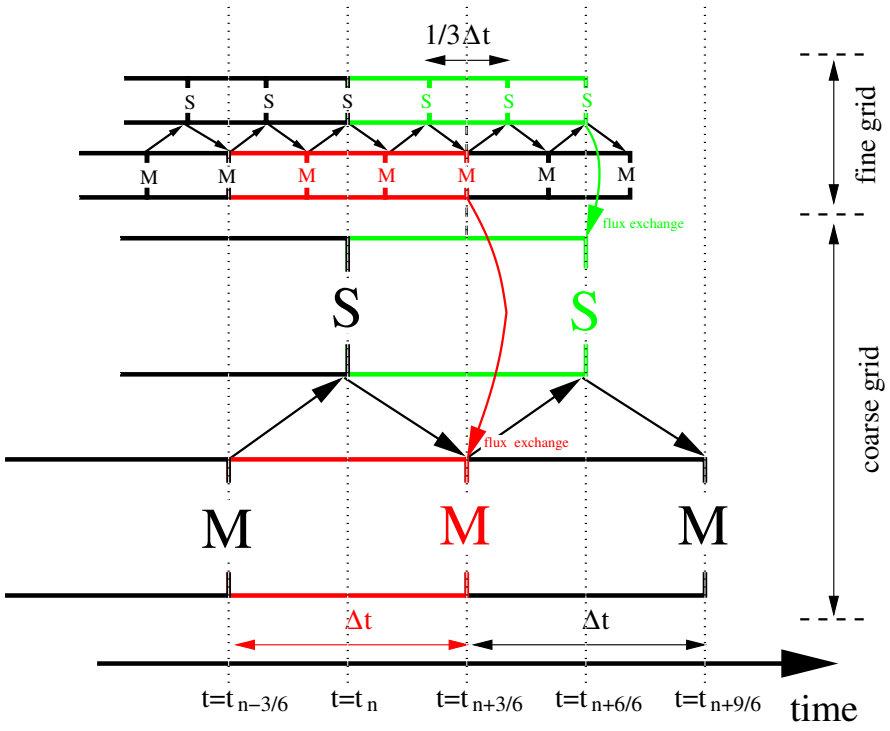

Figure 2.8: Staggering of time integration of scalar (S) and momentum quantities (M) for fine and coarse grid. The colours mark which time steps of coarse and fine grid belong to each other (with exchanging fluxes). The red arrow marks the exchange of the momentum fluxes at $t = t^{n+3/6}$ (the fluxes $\vec{F}_m^{n-1/6} + \vec{F}_m^{n+1/6} + \vec{F}_m^{n+3/6}$ are given from the fine to the coarse grid). The green arrow marks the exchange of the scalar fluxes at $t = t^{n+6/6}$ (the fluxes $F_s^{n+2/6} + F_s^{n+4/6} + F_s^{n+6/6}$ are given from the fine to the coarse grid).

The scalar quantities are known on all grids, and scalar fluxes at the internal boundaries have just been given from the fine to the coarse grid.

- $t = t^{n+1/6}$

  Another momentum step has been performed on the fine grid with storing the fluxes $\vec{F}_m^{n+1/6}$ at the internal boundaries and again solving the Poisson equation on the fine grid.

- $t = t^{n+2/6}$

  A scalar step has been performed on the fine grid with storing the fluxes $F_s^{n+2/6}$ at the internal boundaries.

- $t = t^{n+3/6}$

  A momentum step has been performed on the fine grid with storing the fluxes $\vec{F}_m^{n+3/6}$ at the internal boundaries. The fluxes $\vec{F}_m^{n-1/6} + \vec{F}_m^{n+1/6} + \vec{F}_m^{n+3/6}$ are

used when forwarding the coarse grid momentum from $t = t^{n-3/6}$ to $t = t^{n+3/6}$ along with all other preliminary processes. The Poisson equation is solved on both grids at once. Note that the divergences need to be divided by each grid's own time step in order to get a correct perturbation pressure with Eq. (2.20). For the coarse grid this would be three times the fine grid time step in this particular case.

- $t = t^{n+4/6}$

  Another scalar step has been performed on the fine grid with storing the fluxes $F_s^{n+4/6}$ at the internal boundaries.

- $t = t^{n+5/6}$

  Another momentum step has been performed on the fine grid with storing the fluxes $\vec{F}_m^{n+5/6}$ at the internal boundaries and again solving the Poisson equation on the fine grid.

- $t = t^{n+6/6}$

  Another scalar step has been performed on the fine grid with storing the fluxes $F_s^{n+6/6}$ at the internal boundaries. The fluxes $F_s^{n+2/6} + F_s^{n+4/6} + F_s^{n+6/6}$ are taken to forward the coarse grid scalars from $t = t^n$ to $t = t^{n+6/6}$.

- ...

Transition from uniform time step to time step refinement factor 3

There are situations when the time step does not need to be different on grids with different resolution. This is the case when vertical diffusion is restricting the time step, because the vertical resolution is the same on all grids. Consequently, a transition from uniform time stepping to nonuniform time stepping is necessary.

Fig. 2.9 shows a schematic on how such a transition is performed. The text description is starting from a uniform time step:

- ...

- $t = t^n$

  The scalar quantities are known on all grids. The momentum field is divergence free by having solved the Poisson equation on all grids at $t = t^{n-1/6}$. Up to now on the coarse and fine grid the same time stepping was used.

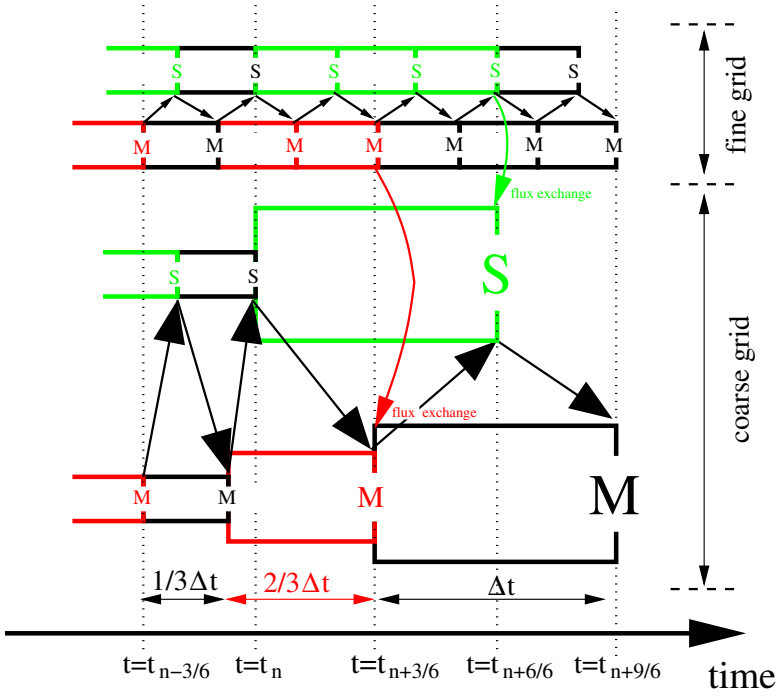

Figure 2.9: Staggering of time integration of scalar (S) and momentum quantities (M). Transition from initially synchronous time integration of fine and coarse grid to greater time steps for the coarse grid. The colours mark which time steps of coarse and fine grid belong to each other. The red arrow marks the exchange of the momentum fluxes at $t = t^{n+3/6}$ (the fluxes $\vec{F}_m^{n+1/6} + \vec{F}_m^{n+3/6}$ are given from the fine to the coarse grid). The green arrow marks the exchange of the scalar fluxes $t = t^{n+6/6}$ (the fluxes $F_s^{n+2/6} + F_s^{n+4/6} + F_s^{n+6/6}$ are given from the fine to the coarse grid).

- $t = t^{n+1/6}$

  A momentum step has been performed on the fine grid storing the fluxes $\vec{F}_m^{n+1/6}$ at the internal boundaries and solving the Poisson equation on the fine grid applying Dirichlet boundary conditions (the coarse grid momentum is held constant).

- $t = t^{n+2/6}$

  A scalar step has been performed on the fine grid with storing the fluxes $F_s^{n+2/6}$ at the internal boundaries. The coarse grid scalars are held constant.

- $t = t^{n+3/6}$

  A momentum step has been performed on the fine grid with storing the fluxes $\vec{F}_m^{n+3/6}$ at the internal boundaries. The fluxes $\vec{F}_m^{n+1/6} + \vec{F}_m^{n+3/6}$ are taken to forward the coarse grid momentum from $t = t^{n-1/6}$ to $t = t^{n+3/6}$ along with all other preliminary processes. The Poisson equation is solved on both grids.

Note that the divergences need to be divided by each grid's own time step in order to get a correct perturbation pressure. For the coarse grid this would be twice the fine grid time step. Now the transition to different time steps is complete.

- $t = t^{n+4/6}$

  Another scalar step has been performed on the fine grid with storing the fluxes $F_s^{n+4/6}$ at the internal boundaries.

- to be continued as in the fully time staggered algorithm outlined above ...

3 Use of (weighted) essentially non-oscillatory advection schemes in a mesoscale model

This chapter was published as:

Schroeder, G., Schlünzen, K. H. and Schimmel, F. (2006): Use of (weighted) essentially non-oscillatory advection schemes in a mesoscale model. *Q. J. R. Meteorol. Soc.*, **132**, 1509-1526.

Since this chapter has been published, the notation may sometimes deviate from the one used in the other chapters of this thesis.

Summary

Plain and weighted essentially non-oscillatory advection schemes (ENO and WENO) are implemented in the atmospheric model METRAS to solve the advective terms in the momentum equations. They are an alternative to the currently used centered differences that are stabilised by a Shapiro filter. For the (W)ENO schemes two approaches are tested for reconstructing the advective fluxes on an Arakawa-C-grid. The resulting four second order (one filtered and three unfiltered) and two (unfiltered) third order numerical schemes are tested for three different types of test cases, i.e. one type consisting of three idealised density current cases, one case with stationary mountain waves and one case with land-sea breeze combined with mountain-valley wind circulations (LSBM). The results are compared to each other and to results of METRAS with centered differences (CDF). When using uniform grids, one of the (W)ENO schemes is ruled out as it turns out to be unstable. The (W)ENO results as well as the CDF result are altogether plausible. They are very similar for stationary and non-convective situations. For convective situations (case LSBM) the results are spread leading to a multi-numeric ensemble. By comparison of the results of the filtered and unfiltered second order ENO scheme it is shown that one reason for the spread is the Shapiro filter. As the filter is only meant to stabilise the solution it must be assumed that the changes caused by the filter are artificial. A grid with step-wise changes in mesh size tested for the density current identifies some shortcomings in the numerical schemes. On this grid the CDF solution contains a high amount of non-physical oscillations while the (W)ENO schemes show reasonable results. When using the second order (W)ENO schemes the total CPU time is not much larger than for the CDF scheme. Therefore, two second order (W)ENO schemes are recommended for step-wise changes in mesh size. The use of the third order (W)ENO schemes is not justifiable as they do not provide considerably better results but may need more total CPU time.

3.1 Introduction

A huge amount of literature deals with scalar advection schemes. This can be explained with the need to simulate the advective transport of tracers and pollutants at a high accuracy and with little numerical diffusion. Several schemes have been compared by Sokol (1999) with the conclusion that the modified Bott scheme (Bott, 1989; Easter, 1993) provides the best results. Semi-Lagrangian schemes (e.g.

Staniforth and Cote, 1991) have been shown to be very accurate, as well as the multidimensional flux integrated UTOPIA scheme by Leonard et al. (1995). The development of sophisticated flux limiters has ensured shape preservation and positive definiteness for multidimensional applications (e.g. Thuburn, 1996).

For calculating the advection of momentum fewer efforts were made to improve the numerical schemes. More sophisticated advection schemes are also desirable for the momentum equations, since the flow field is of relevance for all transports in the atmosphere. Many models apply simple second order central advection schemes (e.g. Schlünzen, 1990; Grell et al., 1994; Xue et al., 2000; Steppeler et al., 2003). ARPS (Xue et al., 2000) alternatively employs a fourth-order central scheme. Central schemes are easy to implement and relatively cheap to compute. Nevertheless, they lack stability and tend to produce artificial oscillations making it necessary to introduce numerical diffusion. This can be done by filtering or by using minimum diffusion coefficients. Furthermore, central schemes can lead to oscillations if the computational grid contains sudden changes in mesh size. Those large changes in mesh size are helpful to avoid nesting or to allow for adaptivity.

With models in development that use self-refining grids, sudden changes in mesh size will become standard in future numerical models. Numerical schemes need to be employed that are capable of grid changes. The class of essentially non-oscillatory (ENO) schemes and weighted essentially non-oscillatory (WENO) schemes seems to be very suitable for this purpose. ENO schemes have first been presented by Harten et al. (1987) and WENO schemes are a further improvement of them (e.g. Jiang and Shu, 1996). These schemes are shock-capturing high resolution schemes and have been successfully used when simulating shock turbulence interactions (e.g. Garnier et al., 2002). A good overview of ENO and WENO schemes is given by Shu (1999).

The objective of this work is to present and compare numerical schemes for momentum advection with focus on (W)ENO methods. They are implemented in the mesoscale model METRAS and results are compared to the standard centered differences for several test cases. Special interest is given to the stability of the numerical schemes and to the ability of handling sudden changes in mesh size.

Section 2 presents the model METRAS that has been used for the test of the advection schemes. Section 3 describes the (W)ENO advection schemes and different numerical flux functions. Results of test simulations are given and compared in Section 4. The conclusions are drawn in Section 5.

3.2 Model METRAS

The atmospheric mesoscale transport and stream model METRAS is used to test the advection schemes. Here only a short description of the numerics is given. Physical parameterisations are described in Schlünzen (1990) and Lüpkes and Schlünzen (1996).

METRAS is based on the equations for conservation of momentum, mass and energy, resulting in prognostic equations for the three wind components, potential temperature, humidity, cloud and rain water content as well as pollutants. To avoid sound waves, the anelastic and Boussinesq approximations are applied. The equations are volume integrated and solved in flux form. A first order closure is employed using a counter gradient scheme for convective situations (Lüpkes and Schlünzen, 1996) To account for orography the model is written in terrain-following coordinates. The equations are discretised on an Arakawa-C-grid (Mesinger and Arakawa, 1976).

In the momentum equations advection and diffusion are simultaneously solved with the second order Adams-Bashforth scheme in time and centered differences in space. This scheme was shown to be stable for Courant numbers (CFL) up to 1 if a sufficient amount of turbulent diffusion exists in the atmosphere (Schumann et al., 1987). Coriolis force and buoyancy are solved forward in time. Pressure gradient force is calculated iteratively from a Poisson equation using the Bi-CGSTAB algorithm (Van der Vorst, 1992) with ILU preconditioning. This solver is named pressure solver in the following. To avoid nonlinear instability a seven point filter is applied to the wind components every time step (Shapiro, 1971).

Temperature, humidity, cloud and rain water content as well as pollutant concentrations equations are solved forward in time. For these a first order upstream scheme is used for advection. Vertical diffusion is solved semi-implicitly with the Crank-Nicholson scheme if the otherwise used explicit approach leads to a too strong constraint on the time step.

3.3 (W)ENO interpolation

The essentially non-oscillatory (ENO) schemes go back to Harten et al. (1987). In contrast to central schemes they do not require any artificial diffusion as they largely avoid generating spurious oscillations and have some scheme inherent diffusivity. The backbone of the ENO scheme is the ENO interpolation used for the

reconstruction of values at the cell boundaries. These are essential for calculating flux divergences. For example, for a quantity $q(x)$ discretised by $q_i = q|_{x_i}$ at the cell centre x_i the boundary value $q_{i+1/2}$ is obtained by calculating an auxiliary polynomial Q_i^x for each cell so that $q_{i+1/2} = Q_i^x(x_{i+1/2})$ (ENO interpolation in x-direction, indicated by superscript x). To obtain a scheme of n-th order the polynomial must be based on a stencil $(q_k, \dots q_{k+n-1})$ with $Q_i^x(x_l) = q_l$ and $i, l \in \{k, \dots, k+n-1\}$. So n possibilities for choosing a stencil exist. The idea behind ENO interpolation is to choose the stencil on which the smoothest interpolation polynomial Q_i^x is obtained. This automatically precludes interpolation across discontinuities and, thus, avoids artificial oscillations. For calculating the interpolants divided differences are helpful. They are commonly defined recursively. Here we only state divided differences with order less than three:

$$q[x_i] = q_i \tag{3.1}$$

$$q[x_i, x_{i+1}] = (q_{i+1} - q_i)/(x_{i+1} - x_i) \tag{3.2}$$

$$q[x_i, x_{i+1}, x_{i+2}] = (q[x_{i+1}, x_{i+2}] - q[x_i, x_{i+1}])/(x_{i+2} - x_i) \tag{3.3}$$

For a second order reconstruction the polynomial can be written as

$$Q_i^x = q_i + (x - x_i)(\alpha_i q[x_{i-1}, x_i] + \beta_i q[x_i, x_{i+1}]) \tag{3.4}$$

$$\alpha_i = \begin{cases} 1 & : \quad |q[x_{i-1}, x_i]| < |q[x_i, x_{i+1}]| \\ 0 & : \quad \text{else} \end{cases} \tag{3.5}$$

$$\beta_i = 1 - \alpha_i, \tag{3.6}$$

the third order polynomial is given by

$$\begin{aligned} Q_i^x = q_i \quad &+ \quad \alpha_i\{(x - x_i)q[x_{i-1}, x_i] + (x - x_i)(x - x_{i-1})q[x_{i-2}, x_{i-1}, x_i]\} \\ &+ \quad \beta_i\{(x - x_i)q[x_{i-1}, x_i] + (x - x_i)(x - x_{i-1})q[x_{i-1}, x_i, x_{i+1}]\} \\ &+ \quad \gamma_i\{(x - x_i)q[x_i, x_{i+1}] + (x - x_i)(x - x_{i+1})q[x_i, x_{i+1}, x_{i+2}]\} \end{aligned} \tag{3.7}$$

$$\alpha_i = \begin{cases} 1 & : \quad |q[x_{i-1}, x_i]| < |q[x_i, x_{i+1}]| \wedge |q[x_{i-2}, x_{i-1}, x_i]| < |q[x_{i-1}, x_i, x_{i+1}]| \\ 0 & : \quad \text{else} \end{cases} \tag{3.8}$$

$$\gamma_i = \begin{cases} 1 & : \quad |q[x_i, x_{i+1}]| < |q[x_{i-1}, x_i]| \wedge |q[x_i, x_{i+1}, x_{i+2}]| < |q[x_{i-1}, x_i, x_{i+1}]| \\ 0 & : \quad \text{else} \end{cases} \tag{3.9}$$

$$\beta_i = 1 - \alpha_i - \gamma_i. \tag{3.10}$$

For an exact finite volume formulation the interpolation needs to conserve the cell averages. This can be achieved by setting $\tilde{Q}_i^x(x) = q_i + Q_i^x(x) - \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} Q_i^x(x')\,dx'$.

This becomes only necessary for interpolants with orders larger than one (i.e. schemes of orders larger than two). The third order results in this paper are produced without a cell average conserving interpolation as the cell average conserving scheme tested did not show a noticeable improvement. Unlike the schemes presented here, cell average conservation is especially important for flux integrated schemes like UTOPIA (Leonard et al., 1995).

Weighted essentially non-oscillatory (WENO) schemes (e.g. Liu et al., 1994; Jiang and Shu, 1996) are a further development of ENO schemes. Instead of choosing the smoothest interpolation polynomial a weighted convex combination of the interpolation polynomials is used for the reconstruction. The weights α_i, β_i and γ_i are chosen as in Shu (1999). The smoother the polynomial the larger is its weight. Note that while each single polynomial is interpolating at $n+1$ points, the convex combination for the WENO scheme is only interpolating at x_i. Figure 3.1 shows an example of a third order ENO interpolated quantity. In the general case the interpolant in each cell can be based on different stencils. At the cell boundaries discontinuities are apparent for which numerical flux functions have to be found.

The (W)ENO interpolation is applied to the advection terms of the momentum equations:

$$\frac{\partial \rho_0 u}{\partial t} = \quad -\frac{\partial}{\partial x}(u\rho_0 u) - \frac{\partial}{\partial y}(v\rho_0 u) - \frac{\partial}{\partial z}(w\rho_0 u) \quad + R_x$$

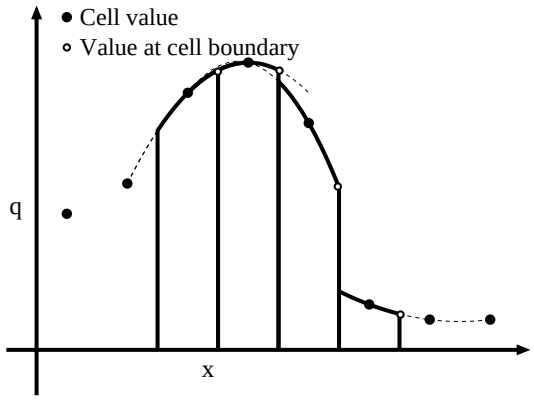

Figure 3.1: Example of a third order ENO interpolated quantity. Solid line: reconstruction of the discrete quantity q. Dashed line: Interpolant through the neighbouring values.

$$\begin{aligned}\frac{\partial \rho_0 v}{\partial t} &= -\frac{\partial}{\partial x}(u\rho_0 v) - \frac{\partial}{\partial y}(v\rho_0 v) - \frac{\partial}{\partial z}(w\rho_0 v) \quad + R_y \\ \frac{\partial \rho_0 w}{\partial t} &= -\frac{\partial}{\partial x}(u\rho_0 w) - \frac{\partial}{\partial y}(v\rho_0 w) - \frac{\partial}{\partial z}(w\rho_0 w) \quad + R_z\end{aligned}$$

where (u, v, w) is the velocity vector, ρ_0 the large scale density and (R_x, R_y, R_z) the vector of the remaining terms of the momentum equations (e.g. pressure gradient force). In case a coordinate transformation is used (e.g. terrain-following coordinates as used in METRAS) only the advected quantity itself (i.e. u, v and w) is ENO interpolated. The metric terms must not be ENO interpolated as an interpolation of a product of quantity and the metric term would lead to distortion.

u, v, w and ρ_0 are staggered on the Arakawa-C-grid (Mesinger and Arakawa, 1976). The scalar points in x- and y-direction are given by x_i and y_j, the vector points by $x_{i+1/2}$ and $y_{j+1/2}$. u is defined at $(x_{i+1/2}, y_j)$ by $u_{i+1/2,j}$, v is defined at $(x_i, y_{j+1/2})$ by $v_{i,j+1/2}$, ρ_0 at (x_i, y_j) by $\rho_{0i,j}$. Figure 3.2 explains how the advection flux is calculated for the term $\frac{\partial}{\partial x}(u\rho_0 v)$. The advecting velocity component is averaged to the cell centre, the advected velocity component is ENO interpolated. As shown in Fig. 3.1 the ENO interpolated function may be discontinuous if the polynomials of two adjacent cells are based on different stencils. In this case the upstream value is taken (UP) or the simple average of both values (AV). With these flux functions the flux through the cell faces is calculated, and Euler forward is used for time integration.

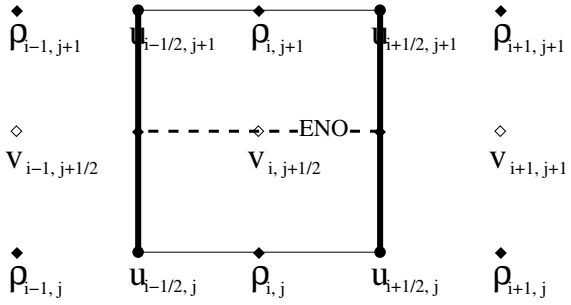

Figure 3.2: Horizontal cross-section of the Arakawa-C-grid. The $(i,j+1/2)$ v-cell is outlined. For the advection flux through the left and right cell face (bold lines), the advecting wind u is simply averaged to the centre of the cell face, the advected wind v is ENO interpolated along the dashed line.

3.4 Testing the schemes

Three different types of test cases are performed. These are selected to detect shortcomings in the numerical schemes and numerical flux functions. Results from six different schemes are compared with results from the centered differences (CDF). Table 3.1 gives an overview of the schemes, their order and the applied numerical flux functions. CDF uses a CFL number of 0.8, the four second order (W)ENO schemes apply a CFL number of 0.63 and the two third order schemes 0.3. The CFL number has been determined from experiments with Burgers' equation. A seven point filter (Shapiro, 1971) is used for stabilisation of the centered difference scheme (CDF). It is also used for comparison in conjunction with a second order ENO scheme (E2UPF).

Table 3.1: Overview of schemes tested with different test cases.

Label	Class	Order	flux function	filter	CFL
CDF	central scheme	2	-	seven point filter	0.8
E2AV	ENO	2	AV	-	0.63
E2UPF	ENO	2	UP	seven point filter	0.63
E2UP	ENO	2	UP	-	0.63
W2UP	WENO	2	UP	-	0.63
E3UP	ENO	3	UP	-	0.3
W3UP	WENO	3	UP	-	0.3

3.4.1 Idealised density current

An idealised density current is used to test the robustness of the schemes for inviscid, time dependent solutions. The test case is based on papers by Skamarock and Klemp (1993), Carpenter et al. (1990) and Schimmel (2002), the same experimental set-up is used. A two-dimensional y-z domain of 40 km length and 10 km height is prescribed. In contrast to open boundary conditions common for limited area models a solid wall is prescribed at the southern boundary for these experiments. Free-slip conditions are applied at all solid boundaries (i.e. the ground and the south). At model initialisation the atmosphere is at rest, and the potential temperature is given

by

$$\theta = \begin{cases} 290\ \mathrm{K} + \dfrac{10\ \mathrm{K}}{5000\ \mathrm{m}} z & \text{for} \quad z \leq 5000\ \mathrm{m} \quad \text{and} \quad y \leq 0\ \mathrm{m}; \\ 300\ \mathrm{K} & \text{elsewhere.} \end{cases} \tag{3.11}$$

The initial atmosphere is in hydrostatic equilibrium with a vanishing horizontal pressure gradient for $z > 5000$ m. The model simulations are performed without considering the influence of turbulence. Thus, only advective processes are accounted for in this inviscid flow. This is the basic configuration of three tests. Case one has a homogeneous horizontal and vertical mesh size of 125 m (DC125), case two a step-wise change in mesh size (factors 2 and 4, DCstep), and case three includes a density current over a mountain (DCMOUNT).

(i) 125 m mesh size (DC125)

Figure 3.3 shows the results for the experiment with 125 m mesh size as vertical cross-section for all schemes except for E2AV after 15 minutes integration time. The E2AV run does not remain stable and crashes after less than 13 minutes. Therefore, in Fig. 3.3 the E2AV result is shown after 12 minutes simulation time. Oscillations caused by the advection of momentum are clearly visible even in the field of potential temperature.

All other schemes show a clear front at about $y = 15$ km after 15 minutes, but differ in the intensity and wave length of the Kelvin-Helmholtz billows behind the frontal head. CDF shows the shortest waves (wave length of about 5 km), the second order (W)ENO schemes are more diffusive. Only one clear billow is visible. The third order schemes are somewhat less diffusive, the development of a small billow close to the frontal region is apparent (Fig. 3.3(e,f)). The results compare well to those by Carpenter et al. (1990) and Skamarock and Klemp (1993) who used different mesh sizes. The Piecewise Parabolic Method (PPM) used by Carpenter et al. (1990) is third order like E3UP and W3UP. With 83 m mesh size it shows five billows and the solution seems to be a lot less smooth than the results presented here. One reason for this is the lower mesh size. Additionally, the third order (W)ENO schemes are more diffusive by choosing the smoothest stencil. However, the main reason is probably that temperature is advected with first order upstream. The very fine scale simulation of Skamarock and Klemp (1993) with $\Delta x \approx 29m$ shows a lot more small scale billows as well. Due to the nonlinear behaviour of the simulation an aggregated fine scale solution is not a meaningful reference for the large scale simulation (e.g.

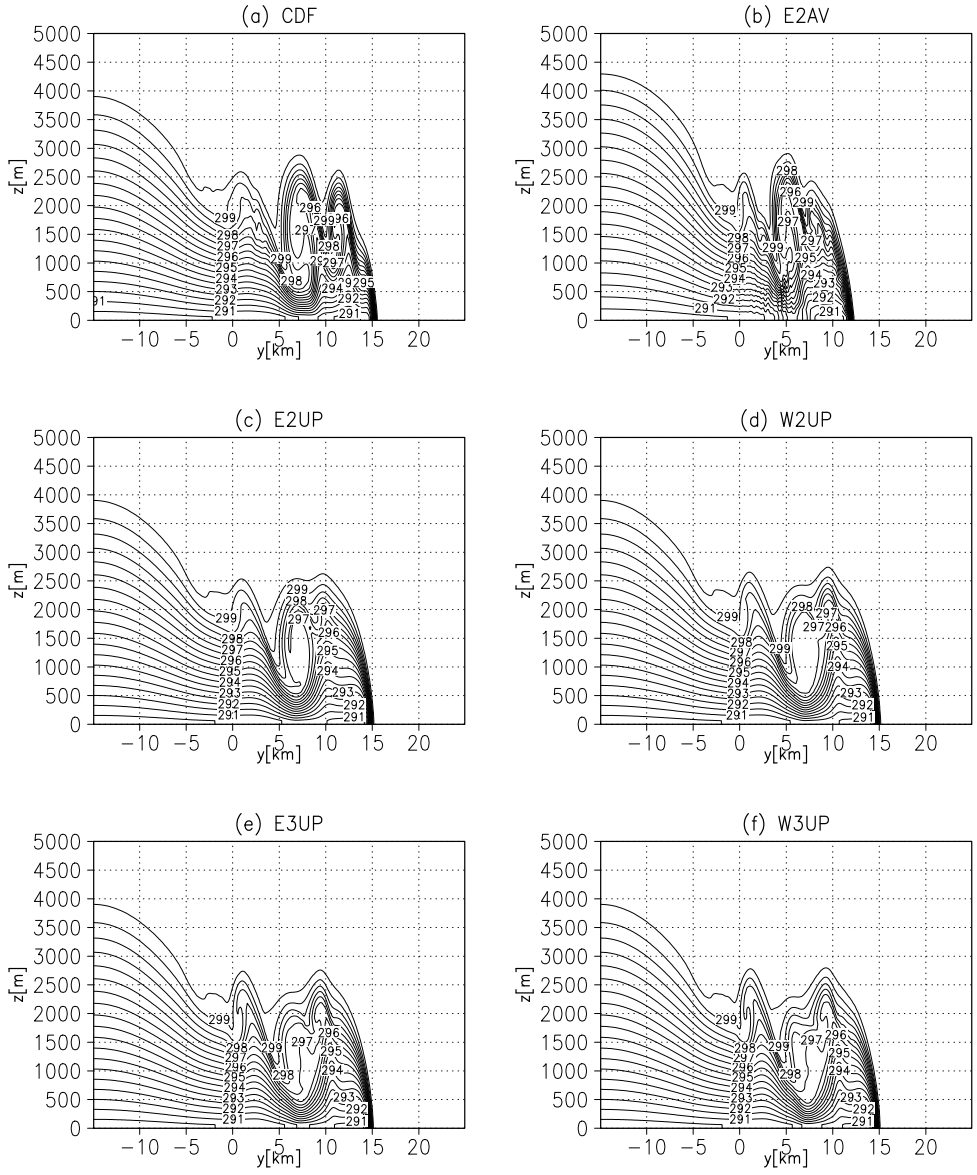

Figure 3.3: Results of DC125: Potential temperature from 290 K to 300 K, increment 0.5 K, for (a) the reference case (CDF), (b) E2AV, (c) E2UP, (d) W2UP, (e) E3UP and (f) W3UP. Except for (b) all plots are given after 15 minutes integration time. For (b) the result after 12 minutes integration time is shown (shortly before the model run aborted).

for the locations of billows).

CDF shows also some numerically caused very short waves of wave length 1 km and less. This is a disadvantage of the centered differences which becomes visible when the physical diffusion is small.

The second order (W)ENO results are very similar. The same is true for the third order results (Fig. 3.3(e, f)). All four stable (W)ENO schemes show no noticeable numerical noise. Due to a lack of validation data it can not be decided which of these results is most realistic, but all three classes of numerical schemes are believed to show a reliable density current.

To get an estimate of the numerical stability a horizontal Fourier analysis of the temperature and wind fields is performed. With the vertical grid index k and $\psi_{j,k}$ being the discretised field ψ at the grid point (j,k), the vertically maximal spectral coefficient is defined by

$$\hat{\psi}_l = \max_k \left| \sum_{j=0}^{N-1} \psi_{j,k} e^{-i2\pi lj/N} \right|, \tag{3.12}$$

where N is the number of horizontal grid levels taken into account. Note that l is the wave number. It is assumed that the variability associated with waves with wave lengths fitting into 2 to 8 grid cells is not supposed to grow in time for a stable numerical scheme. To evaluate the numerical schemes, the average variability per wave number is calculated for waves within 2 to 8 grid cells:

$$\overset{\circ}{\psi} = \frac{1}{N/2 - N/8 + 1} \sum_{l=N/8}^{N/2} \hat{\psi}_l. \tag{3.13}$$

Figure 3.4a shows the time dependence of $\overset{\circ}{\psi}$ for the horizontal wind v. For the unstable E2AV scheme a large increase of $\overset{\circ}{v}$ is visible after 10 minutes for case DC125. For the other schemes $\overset{\circ}{v}$ remains bounded. However, the CDF scheme produces a little more variability in the short waves than the other stable methods. The Fourier analysis can clearly identify the unstable scheme, but similarly stable experiments cannot be distinguished.

(ii) 125 m to 500 m step-wise change in mesh size (DCstep)

To test the robustness of the numerical schemes when dealing with a sudden change in mesh size, a grid with refinement factors of two and four is tested for the same

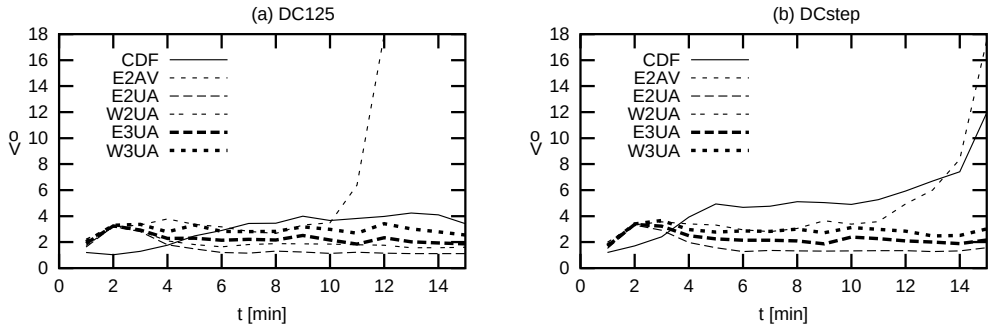

Figure 3.4: The average variability of the horizontal wind within 2-8 grid cells (Eq. (3.13)) for different numerical methods as function of integration time for cases (a) DC125 and (b) DCstep.

experimental set-up as before. It is well known that abrupt changes in mesh size are accompanied by a loss of order and by refraction and reflection of waves. Therefore, the change of mesh size should not be too large. Factors up to four are still acceptable and commonly used for two-way nesting (e.g. Zhang et al., 1986). The grid in this simulation has a mesh size of 500 m in y and z direction except for an area in the centre. For -2.5 km $< y <$ 15 km the horizontal mesh size is 125 m. Below 1000 m the vertical mesh size is 250 m, for 1000 m $< z <$ 3000 m it is 125 m. Thus it is ensured that the region of strong Kelvin-Helmholtz instability lies within the refined area. Therefore, results similar to those of case DC125 are expected if the advection scheme can handle the refinement. Figure 3.5 shows the results for CDF and E2UP. The CDF solution visibly differs from the one on the uniform grid and is influenced by spurious oscillations. The flow pattern simulated on the uniform grid is more or less destroyed. The (W)ENO results with exception of E2AV remain free of numerical noise as exemplarily shown for E2UP. The results are very similar to the results with a constant mesh size of 125 m. The main difference is that the front is somewhat slower for case DCstep than in the constant mesh size case DC125. This is due to the smaller gradients on the coarser grid.

For DCstep the Fourier analysis is done in the same way as for DC125 because computational waves are expected to appear in grid point space. Therefore, the change in mesh size does not need to be accounted for in the calculation. The Fourier analysis results are similar to DC125 except for CDF and E2AV (Fig. 3.4(b)). For

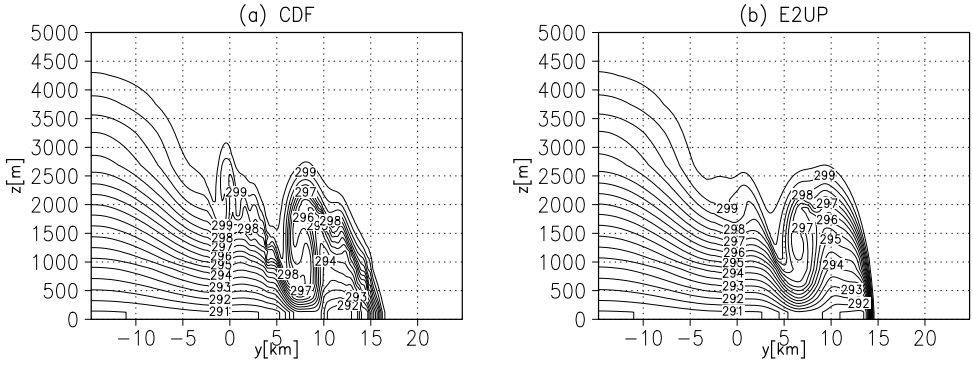

Figure 3.5: Results of DCstep. As in Fig. 3.3 but with a mesh size of 125 m - 500 m. (a) CDF and (b) E2UP.

CDF the variability is continuously larger in DCstep than in DC125 until it starts growing. After 15 minutes it is more than three times larger. E2AV remains stable for a longer time in the DCstep simulation than in DC125, but in the end it leads to a large increase in short wave variability as well.

(iii) Density current over a mountain (DCMOUNT)

DCMOUNT has the same experimental set-up as DC125 except that it includes a mountain of height $H = 1500$ m. The surface elevation z_s is given by

$$z_s = H \frac{L_h^2}{L_h^2 + (y - y_s)^2} \tag{3.14}$$

with the half-width $L_h = 3000$ m, and the peak is located at $y_s = 5000$ m. Figure 3.6 shows the result for CDF and E2UP. The air ahead of the front is moved over the ridge without great resistance, because it is neutrally stratified. As soon as the cold air reaches the top it overshoots and the reverse current aloft causes the development of a billow. At this time the front slows down due to the bifurcation of the current. The CDF solution (Fig. 3.6(a)) shows oscillations, because the diffusion of the seven point filter is not strong enough. The (W)ENO results except for E2AV are free of numerical noise. However, one best scheme cannot be identified. When looking at the Fourier analysis (not shown) only E2AV shows a strong increase of short wave

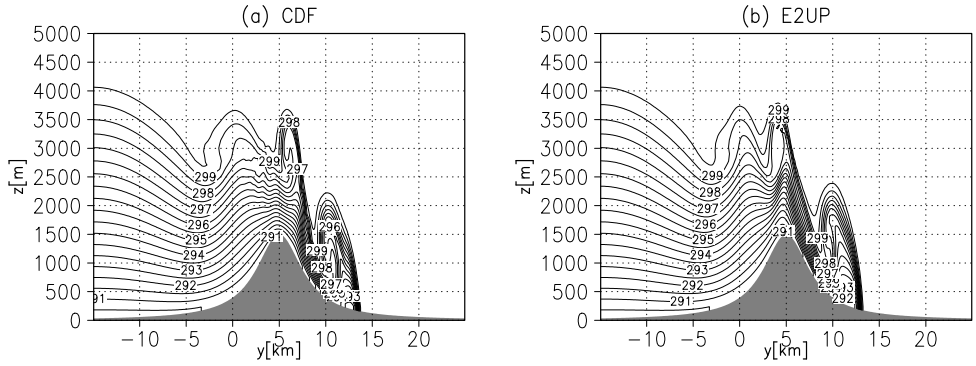

Figure 3.6: Results of DCMOUNT. As in Fig. 3.3 but with a mountain included. (a) CDF and (b) E2UP.

variability as for DC125 and DCstep.

3.4.2 Mountain waves (MWAVE)

In this experiment a two-dimensional inviscid flow over a mountain ridge is simulated within a domain of 1200 km in horizontal and 15 km in vertical direction. The mesh size is 250 m in the vertical and 10 km in the horizontal. The surface height is given by Eq. (3.14) with y substituted by x, $L_h = 30$ km, $H = 10$ m and x_s=0 km. The simulation is performed without Coriolis force effects. The approach flow is 5 ms^{-1} and the static stability of the atmosphere is $N = \sqrt{\frac{g}{\theta_0}\Gamma} \approx 0.013$ s^{-1} assuming $\theta_0 = 290\ K$. The analytical solution after Long (1953) yields a vertical wave length of 2500 m. The configuration is chosen to get a flow similar to Schumann et al. (1987) so that the numerical solution can match the analytic solution. Consequently, the simulated quantities do not deviate much from the large scale flow except for E2AV which became unstable. For the vertical velocity component the maximum differences between analytical solution and simulation is about 0.00052 ms^{-1} for CDF and ranging from 0.00053 for W3UP to 0.00062 for E2UP for the (W)ENO schemes. CDF seems to perform better and to be a bit less diffusive than all other schemes (Fig. 3.7). However, in this simulation the flow is quasilinear. The role of the nonlinear advection of momentum is of secondary nature. Therefore, the

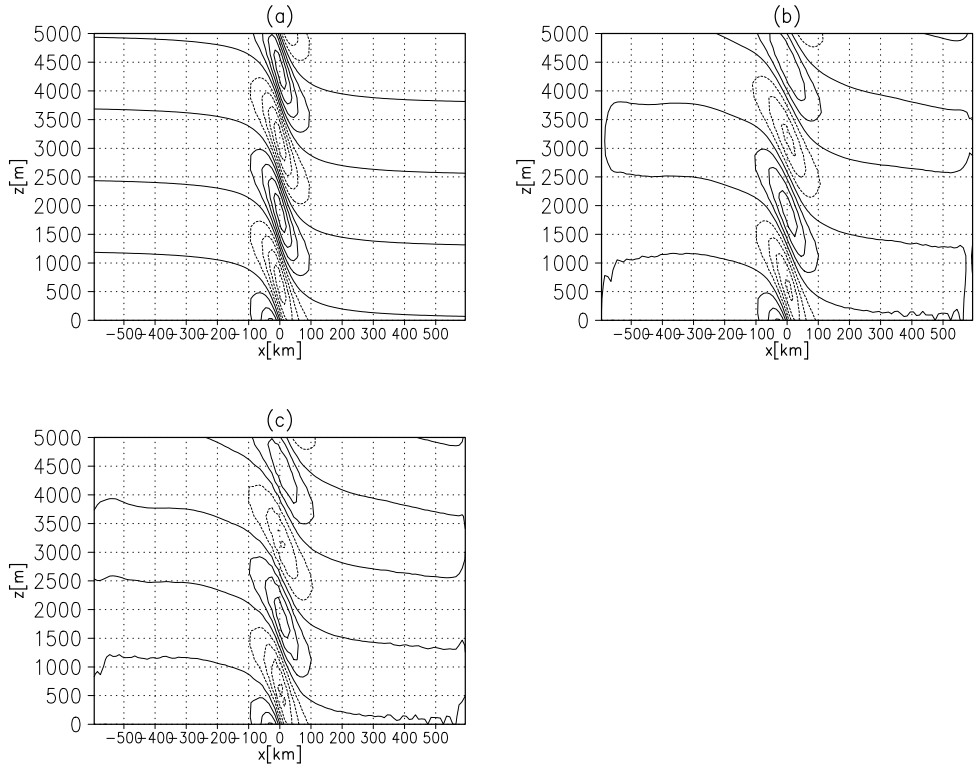

Figure 3.7: Results of MWAVE for (a) the analytical solution, (b) CDF and (c) E2UP. Shown is the vertical velocity component w. Positive contours are solid, negative contours dashed with an increment of 0.0002 ms^{-1}.

differences between the model results should not be overemphasised.

3.4.3 Land-sea breeze with an idealised mountain (LSBM)

To test the different methods using the full model dynamics and physics a land-sea breeze experiment is set up including a bell-shaped west-east oriented mountain ridge. The model area extends from y_s=-70 km to y_n=130 km in the horizontal and over 10 km in the vertical direction. It is homogeneous in x-direction. The mountain shape is given by Eq. (3.14) with H=1000 m, L_h=3000 m and y_s=20 km. South of y=0 km the surface is defined as water. Between 0 km $\leq y \leq$ 10 km the surface has the characteristics of an urban area while the rest of the surface is characterised

as sand desert. The initial surface sea-level temperature is chosen to be 288 K. A geostrophic wind of 0.5 ms^{-1} from North (off-shore) reflects the influence of a small large-scale pressure gradient. The model simulation is performed for 53^o N starting at midnight at summer solstice. The initial atmosphere is stably stratified with a temperature gradient of $\gamma = 0.0035$ K m^{-1}. Relative humidity is linearly decreasing from 80% at the ground to 5% at the tropopause.

The model simulation includes effects of cloud formation and precipitation and different stratifications including convective ones. The system is highly nonlinear. The simulations employing different advection schemes are expected to produce a range of solutions, if they differ in their performance. The simulation is performed with CDF, E2AV, E2UP, W2UP, E3UP, W3UP and E2UP with seven point filter (E2UPF). Again, E2AV became unstable.

Table 3.2: Maxima ψ^+ and minima ψ^- in defined boxes (Fig. 3.8) for the second day. AVE is the average of the CDF result and the average of the (W)ENO schemes' results. P is the total precipitation. All other values are relative to the average and given in per cent. Bold values are differences exceeding 20 per cent.

time	box	$\psi^{\pm}$	AVE	CDF	E2UPF	E2UP	W2UP	E3UP	W3UP
10h	1	v^+	3.1	10	6	-16	-10	-19	-13
10h	1	v^-	-3.1	-4	-9	2	5	10	11
10h	3	w^+	0.79	-15	-20	10	19	19	**45**
10h	2	v^+	2.3	1	-6	4	-6	4	1
12h	4	v^+	3.5	-2	-3	0	9	-3	7
12h	4	v^-	-3.7	11	-4	-7	**-21**	-4	-21
12h	5	v^+	1.8	0	4	**22**	-14	-10	0
12h	5	v^-	-2.5	12	10	-13	-19	-20	**-20**
21h	6	v^+	5.7	11	-14	**-35**	**34**	**-52**	11
$\Sigma_{24\ \mathrm{h},j}$		P	38	**-26**	**63**	2	**23**	**21**	**22**

For the evaluation of the differences between the results the development of significant features is compared for the second day of the simulation. The features are specified in the following text, they are marked with boxes in Fig. 3.8 with the results of E2UP. The evaluation is given in Tab. 3.2. At 10 LST (Local Standard Time) a sea-breeze (box 2) and an anabatic wind circulation (boxes 1, 3) are visible (Fig. 3.8(a, b)). At 12 LST (Fig. 3.8(c)) a pronounced zone of horizontal convergence accompanied by updrafts develops above the ridge (box 4). Over the sand area thermals are visible (box 5). The updrafts above the ridge trigger the development

of clouds, which are fed by humid air advected from the sea. At 21 LST the sea breeze front has crossed the ridge (Fig. 3.8(d)). The zone of horizontal convergence is now located at about 45 km inland (box 6). The mainly orographically caused precipitation P is quite local and differs considerably in the whole model area (Fig. 3.9c for E2UP). For E2UP the sum over each column in y-direction and over 24 h, $\Sigma_{24\text{ h},j} P_j$, is 39 mm for the second day. The distribution of the overall rainfall of the second day is shown in Fig. 3.9 for all simulations.

The described features are found for all numerical schemes, but their intensity and distribution varies. To get an estimate on model result similarities and differences, maximum and minimum values of the wind components in the defined boxes around

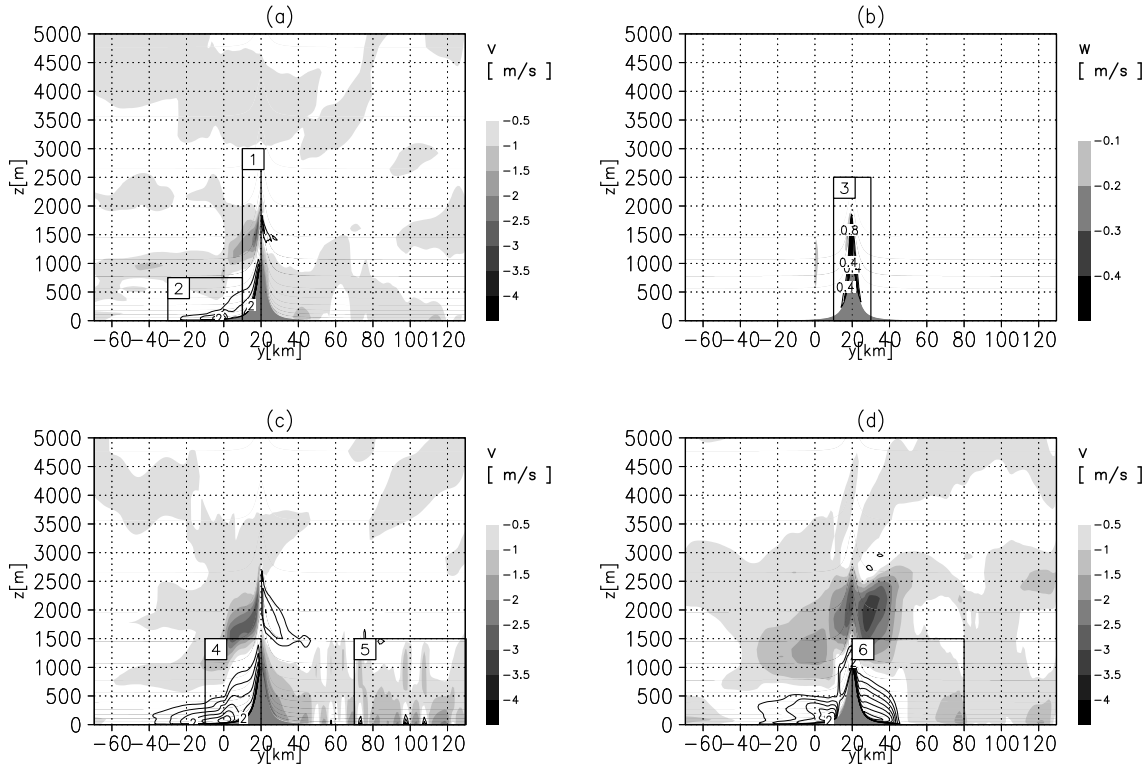

Figure 3.8: Results of LSBM at 10 LST (a, b), 12 LST (c) and 21 LST (d) on the second day as simulated with E2UP. Shown are two components of the wind, v (a, c, d) and w (b). Positive values are contoured, negative values are shaded. The increment is 0.5 ms^{-1} for v and 0.1 ms^{-1} for w.

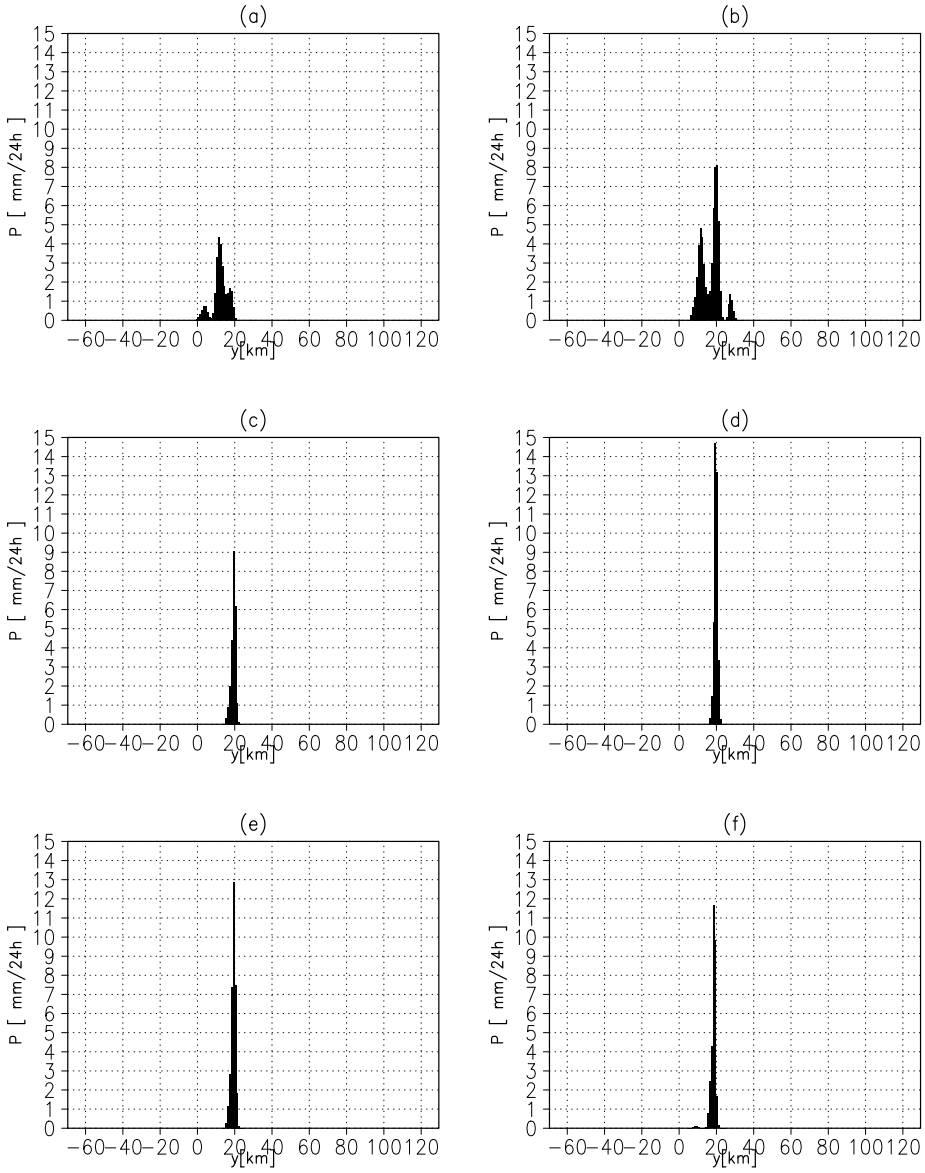

Figure 3.9: Results of LSBM at 24 LST. Shown is the 24 h integral of precipitation P of (a) CDF, (b) E2UPF, (c) E2UP, (d) W2UP, (e) E3UP and (f) W3UP.

the respective feature (Fig. 3.8) are compared (Tab. 3.2). The deviations are listed relative to the corresponding average $\mathrm{AVE}_\psi = 0.5(\psi_{\mathrm{CDF}} + \frac{1}{5}\sum_{i\in M}\psi_i)$ where M is the set of the five (W)ENO methods tested (Tab. 3.1, E2AV is excluded). By this

formulation the CDF result and the averaged (W)ENO results get the same weight. AVE_ψ can be interpreted as result of a multi-numerics ensemble.

Most values do not deviate more than 20% from the average. Largest deviations are found for the onshore flow within the sea breeze (box 6), the vertical velocity component w and the precipitation P. For the w and P this is not surprising as their intensity strongly depends on nonlinear interactions. While some numerical schemes produce quite similar model results for the previous test cases, results for case LSBM differ considerably. In LSBM, where adiabatic and superadiabatic conditions become relevant, small differences in the advection can lead to solutions with strong variations in the intensity of some features, especially those associated with vertical motion and precipitation. When comparing the results of E2UP and E2UPF, a dramatic difference is visible in the distribution of rainfall (Fig. 3.9(b,c)). The simulation with filter yields precipitation resulting in three peaks whereas the simulation without filter only gives one peak. Furthermore, the total precipitation is more than 60% higher with E2UPF. The CDF simulation which needs the application of the seven point filter does also shows the spread (Fig. 3.9(a)), but the precipitation is reduced by 26% compared to the average. The (W)ENO methods without filter show one distinct (Fig. 3.9(d-f)) maximum. As the filter is only applied for stabilisation without physical meaning it must be assumed that the spread distribution of precipitation is less accurate.

The results of the simulations show that the different numerical schemes can finally lead to different solutions albeit the physical boundary conditions provoke similar physical features as described above. Despite being different the results of all schemes are physically plausible. Results for precipitation suggest that a filter might cause artificial structures. Except for the filter effect it cannot be decided which result is better and which is worse. Note that for application in atmospheric models the numerical scheme can indeed influence model results considerably in the case of relevant nonlinear physical effects.

3.4.4 Computational costs

Simulations applying more complex advection schemes are assumed to need more CPU time than centered differences. Figure 3.10 shows the increase of CPU time for each advection scheme relative to centered differences. The values range from +26% for E2AV to +116% for W3UP. Obviously, the less complicated numerical flux function AV needs less CPU time than UP. WENO schemes are more costly

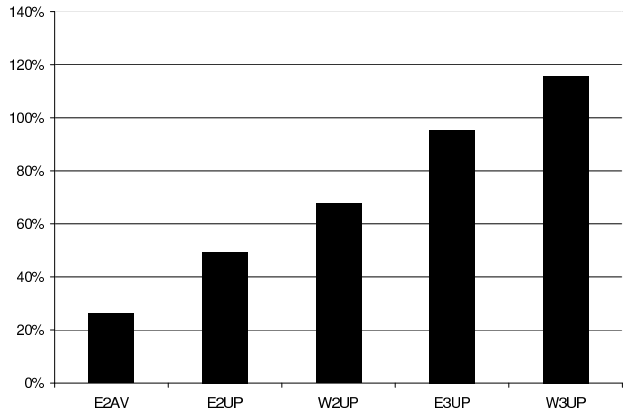

Figure 3.10: CPU time of the algorithm of different advection schemes. All values are given as surplus relative to CDF ($\equiv 0\%$) as derived from the DC125 simulation.

than the corresponding ENO scheme due to the calculation of the extra weighting terms. Not surprisingly the third order schemes (E3UP, W3UP) need considerably more CPU time than the corresponding second order schemes (E2UP, W2UP).

However, when the CPU times of the advection schemes are seen in the context of total CPU time the situation is different. Figure 3.11 shows the CPU times for the experiments DCstep, DCMOUNT, LSBM and a case similar to MWAVE (MWAVE(2)). For DCMOUNT and DCstep the time step is fixed. Therefore, the CFL criterion has no influence on CPU time in these cases. Furthermore, the time consumption of the pressure solver (ca. 80% for DCstep and DCMOUNT relative to the total time) is dominating the whole CPU time making the time needed by the advection scheme negligible. This time can be reduced when allowing a greater residual and thus larger remaining divergences. The maximum residual has been adjusted for this test case as in Smolarkiewicz et al. (1997) so that a smaller residual does only lead to an unneeded extra accuracy. The strong pressure gradients within the frontal region cause large divergence and convergence patterns leading to very large CPU times needed by the pressure solver to determine the balancing pressure field. However, the advection scheme itself indirectly influences the pressure solver. If the advection scheme produces a smooth divergence field the pressure solver needs less time to converge. This seems mostly to be the case for CDF where the filter

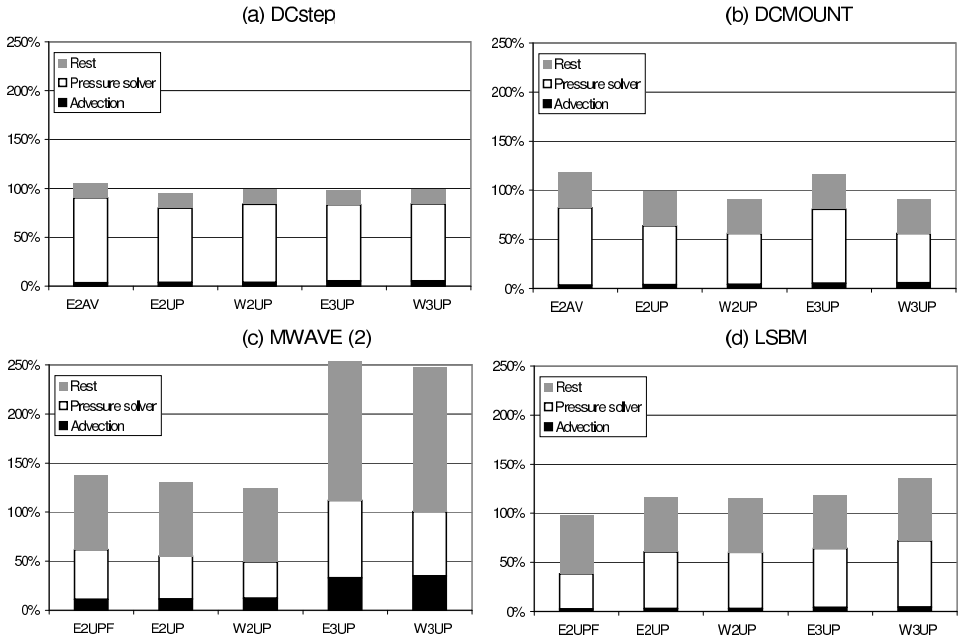

Figure 3.11: CPU time of the (a) DCstep, (b) DCMOUNT, (c) MWAVE(2), a case similar to MWAVE, and (d) LSBM experiment for all (W)ENO schemes. Times are given relative to CDF ($\equiv 100\%$) and separated into the CPU time needed for pressure solver, advection routine and the rest.

results in smoothing, but E2UP and W2UP are not much more time consuming, in some cases even less. For LSBM the effect of the filter is also visible for E2UPF in reduced times for the pressure solver compared to E2UP. The filtered wind field can be made convergence free by the pressure solver with less iterations. For atmospheric problems solvers designed for the Helmholtz equation like the ADI preconditioned conjugate-residual solver proposed by Skamarock et al. (1997) may be more efficient then the Bi-CGSTAB algorithm used in this work. This has not been investigated by the authors.

For MWAVE(2) the CFL criterion is important leading to much longer CPU times for the third order (W)ENO schemes. For the most complex simulation, LSBM, the time step is dominated by vertical diffusion. Therefore, the CPU times of all schemes are almost the same.

These results show that for an anelastic model it is impossible to predict the CPU time on the basis of the advection scheme alone. The nonlinearity of the model and the slow convergence of the pressure solver for complex wind fields can lead to CPU times nearly independent of the advection scheme. For realistic experiments with full model physics (W)ENO schemes may be applied without having to fear much larger CPU times than needed for the CDF scheme. Time splitting (Strang, 1968) can be used to avoid longer time steps needed for high order ENO schemes.

It should be pointed out that the conclusions above are only true for the advection of momentum. When more sophisticated methods are also used for scalar quantities, e.g. a huge amount of chemical constituents, the computational cost of the respective advection scheme will be very important.

3.5 Conclusions

The aim of the work was to apply and test (W)ENO methods for the advection of momentum in an anelastic mesoscale model. Their performance was compared to a commonly used central scheme with a stabilising seven point filter (CDF).

Associated with ENO and WENO schemes is an interpolation polynomial in each cell, where discontinuities might occur at the cell boundaries. Two numerical flux functions, AV and UP, have been presented to compute fluxes at the cell boundaries. In combination with second and third order (W)ENO interpolation six schemes (E2AV, E2UP, E2UPF, W2UP, E3UP and W3UP) and centered differences (CDF) have been examined in different test cases.

The first set of test cases is based on an idealised density current. The stability of the schemes has been tested with and without orography and with changes in mesh size by a factor of two and four. E2AV is generally unstable and leads to unphysical short wave oscillations. This is probably caused by the downstream part of the flux which is not physically meaningful. CDF is unable to cope with large step-wise changes in mesh size and tends to produce some oscillations in the experiment with orography. The other schemes (E2UP, E2UPF, W2UP, E3UP and W3UP) yield better results and are stable. They are also accurate in the refined area where the solution does not differ much from the solution with an equally spaced grid. With respect to total CPU time the (W)ENO schemes do not differ much from CDF.

The second test case is a two-dimensional flow over a mountain ridge, for which

an analytical solution is known. Comparing the fields themselves and with the analytical solution reveals that all methods perform similarly well. Experiments with the third order schemes are very CPU time demanding due to their stronger constraint on the time step. The schemes E2UP and W2UP also need some but only little more CPU time than CDF.

The third test case includes the full model dynamics and physics. It contains a land-sea breeze and mountain-valley wind circulation. The model results differ as a result of the relevance of nonlinear interactions. Especially the amount and the distribution of precipitation differs remarkably. The distribution seems to be strongly influenced by the seven point filter. For flow over mountains Zängl (2002) showed that horizontal diffusion that is not calculated truly horizontally in terrain-following coordinates can distort the solution. The filter applied for CDF and E2UPF acts as unphysical terrain-following diffusion as well and introduces errors.

The differences in the simulations, however, cannot be used to judge on the quality of the model results, but they do show the important influence the advection scheme and employed filter can have. This leads to the conclusion that for ensemble forecasts it should not only be thought of perturbing the initial fields but also of using different numerical methods e.g. for advection. For LSBM the difference in CPU time was especially influenced by the pressure solver. Schemes (like CDF and E2UPF) which produce smoother wind fields lead to faster convergence, but the CPU time increase was below 20% for the two stable second order (W)ENO schemes.

When uniform grids are used a general recommendation cannot be given and (W)ENO schemes and CDF scheme give similar results. However, if step-wise changes of mesh size are needed the CDF scheme is not suitable and the E2UP or W2UP scheme should be used instead. They are the only schemes that use about the same CPU time as scheme CDF and are thus most cost effective (better results for the same price) whereas the two third order schemes might use more computing time.

4 Flux integrated (weighted) essentially non-oscillatory advection schemes

This chapter will be submitted to *Q. J. R. Meteorol. Soc.* as:

Schroeder, G. and Schlünzen, K. H.(2007): Flux integrated (weighted) essentially non-oscillatory advection schemes.

Since the chapter is intended for publication, the notation may sometimes deviate from the one used in the other chapters of this thesis.

Summary

(Weighted) essentially non-oscillatory ((W)ENO) advection schemes have been shown to be suitable for momentum advection when nonuniform grids associated with grid refinement are involved. One drawback of these schemes is that the third order version needs a Courant number of 0.3 and for scalar advection even less. Furthermore, these schemes are not multidimensional. Therefore, in this work two new multidimensional flux integrated advection schemes for momentum and scalar quantities are presented. These have a Courant number of one and are based on the flux integrated advection scheme of Leonard et al. (1995) UTOPIA and on (W)ENO interpolation. The schemes are compared to the original UTOPIA scheme that applies centered interpolation. Simple two-dimensional advection tests as well as a rising bubble simulation show that the new schemes are superior to UTOPIA, since the third order flux integrated ENO (FIENO3) scheme produces no spurious oscillations. With the third order flux integrated WENO (FIWENO3) scheme oscillations of much smaller amplitude than simulated with UTOPIA are visible. For momentum advection it is shown that the one-dimensional FIENO3 scheme performs similarly as the ENO scheme proposed by Schroeder et al. (2006) with the advantage of a CFL number 1.

4.1 Introduction

The accurate simulation of advective transport of quantities like temperature or tracers were of interest since the first development of numerical models for the atmosphere. Various approaches to numerically solve the advection problem have been developed. One of the first schemes, the first order upstream scheme, is stable for Courant numbers smaller than or equal to one. It is conservative and positive definite but not very accurate for small Courant numbers. In these cases upstream is very diffusive and smoothes out extrema. The second order Lax-Wendroff scheme (Lax and Wendroff, 1960) adds antidiffusion to the upstream advection to increase accuracy. It has the same stability properties as the upstream scheme but lacks positive definiteness which can lead to spurious oscillations. The Smolarkiewicz scheme (Smolarkiewicz, 1983) is of similar type as Lax-Wendroff. Its antidiffusive step is governed by an antidiffusive velocity which advects the field in an upstream manner. The resulting scheme preserves positivity for positive fields. It has been extended and generalised by multiple options (e.g. non-oscillatory) resulting in the class of MPDATA schemes (Smolarkiewicz and Margolin, 1998). By including cross

derivatives it has also been made multidimensional. Unlike one-dimensional schemes a multidimensional scheme has the advantage of not distorting the field in case of diagonal advection.

A different approach was introduced by Crowley (1968). He used interpolating polynomials to reconstruct the field and favoured the advective form over the conservation form (better known as flux form), since the conservation form based on integrated fluxes lacks accuracy for higher orders of the numerical scheme. This problem has been removed by Tremback et al. (1987) by making the interpolation cell-average conserving. They named the resulting scheme a constant grid flux scheme. However, their scheme was by no means positive definite which has been achieved by Bott (1989) by applying flux limiters. Chlond (1994) has presented a modified version of this scheme that switches locally to a more diffusive exponential upwind scheme wherever the Bott scheme would violate monotonicity.

A method similar to Crowley's conservation form is the QUICKEST scheme (Leonard, 1979). In addition to the Crowley scheme it also handles diffusion. It is cell-average conserving so that its advection part is identical to the constant grid flux form of Tremback et al. (1987). The improved multidimensional form of QUICKEST is UTOPIA (Leonard et al., 1995) to which a multidimensional flux limiter has been applied by Thuburn (1996). UTOPIA is also called a flux integrated scheme.

Another approach to the advection equation are semi-Lagrangian schemes (Staniforth and Cote, 1991) which are similar to Crowley's advective form. One of their advantages are Courant numbers larger than one. One of their disadvantages is that they are usually not conservative in the general form though efforts have been made to overcome this drawback. For example, Lin and Rood (1996) and similarly Leonard et al. (1996) combine advective and flux form to get a multidimensional, flux limited semi-Lagrangian-like scheme based on time splitting with unrestricted time steps. Skamarock (2006) presents an improved form of limiters used for unrestricted-time-step transport schemes.

More recently, Finite Volume Evolution Galerkin (FVEG) methods have been proposed (Lukáčŏvá-Medviďová et al., 2004; Lukáčŏvá-Medviďová and Saibertova-Zatocilová, 2006; Lukáčŏvá-Medviďová et al., 2007). These schemes are designed for complex hyperbolic systems of equations and are based on the calculation of characteristics along which the system is evolved. The finite volume property is achieved by calculating the integrated finite volume fluxes at an intermediate update time step with the evolution Galerkin method, and forwarding the prognostic variables with

theses fluxes. Multidimensionality is achieved by using bicharacteristics. FVEG methods were developed for complex systems like the Euler equations, but when applied to a simple advection equation they are similar to semi-Lagrangian advection schemes with the advective velocity as the characteristic.

Most of the schemes described above are based on the reconstruction of the field by interpolation which often leads to oscillatory under- and overshoots for higher order schemes. This can cause problems since most scalars are needed to be positive for physical reasons. Furthermore, overshoots in potential temperature can force artificial lifting, overshoots in humidity can cause artificial condensation. Therefore, to reach monotonicity or positive-definiteness it is necessary to use flux-limiters. These are diffusive and can become very complex for multidimensional schemes. It is the motivation for this paper to construct a multidimensional scheme that can be used for advecting scalars without applying flux limiters. A promising method seems to be the application of (weighted) essentially non-oscillatory schemes ((W)ENO Harten et al., 1987; Liu et al., 1994; Jiang and Shu, 1996; Shu, 1999). These have been shown to be (essentially) free of artificial oscillations even when simulating flow with shocks. Schroeder et al. (2006) have also described (W)ENO schemes that are suitable for momentum advection when discontinuous grid spacing is used in atmospheric models. The drawback of these schemes is that they require a Courant number of smaller than one, e.g. for the third order schemes 0.3. Therefore, in this work the (W)ENO interpolation is applied in conjunction with the flux-integration methods that was also used for UTOPIA. This leads to two new combined schemes.

In Section 4.2 the combined schemes are described. Their phase accuracy and amplification factors for one dimension are analysed in Section 4.3, followed by simple two-dimensional tests in Section 4.4. A rising bubble experiment outlines the properties of the schemes within an anelastic model framework in Section 4.5. The suitability of FIENO for momentum advection is demonstrated in Section 4.6. Conclusions are drawn in Section 4.7.

4.2 Flux integrated (W)ENO scheme

Under the assumption of a divergence free momentum field $(\rho u, \rho v)$ the two-dimensional advection equation for a scalar quantity q is given by:

$$\frac{\partial \rho q}{\partial t} + \frac{\partial u \rho q}{\partial x} + \frac{\partial v \rho q}{\partial y} = 0 \tag{4.1}$$

Here ρ is the density of air and (u, v) the two-dimensional wind vector. The two-dimensional field q is discretised with $q^n_{i,j}$ at grid point (x_i, y_j) and time step $t_n = \sum_{l=1}^{n} \Delta t_l$. (i, j) are the grid indices in x- and y-direction. Applying the two-dimensional flux integration (Leonard et al., 1995) leads to the equation

$$(\rho q)^{n+1}_{i,j} = (\rho q)^n_{i,j} - \alpha_x (F^x_{i+1/2,j} - F^x_{i-1/2,j}) - \alpha_y (F^y_{i,j+1/2} - F^y_{i,j-1/2}). \tag{4.2}$$

where $F^x_{i+1/2,j}$ and $F^y_{i,j+1/2}$ are the fluxes through each cell face in x- and y-direction and $\alpha_x = \Delta t/\Delta x$ and $\alpha_y = \Delta t/\Delta y$. Δx and Δy are the horizontal grid widths. It is assumed that during one time step Δt air from a defined upstream area is advected through each cell face by a constant wind (u,v). Figure 4.1 shows examples for two wind directions. The average flux through each cell face is given by the respective wind component and the average of the scalar quantity in the area. This can be calculated if the discretised field $q_{i,j}$ is reconstructed to get a continuous field $\tilde{Q}(x, y)$. $\overline{q}^x_{i,j}$ is defined to be the average of $\tilde{Q}$ over the area that is advected through the cell face in x-direction:

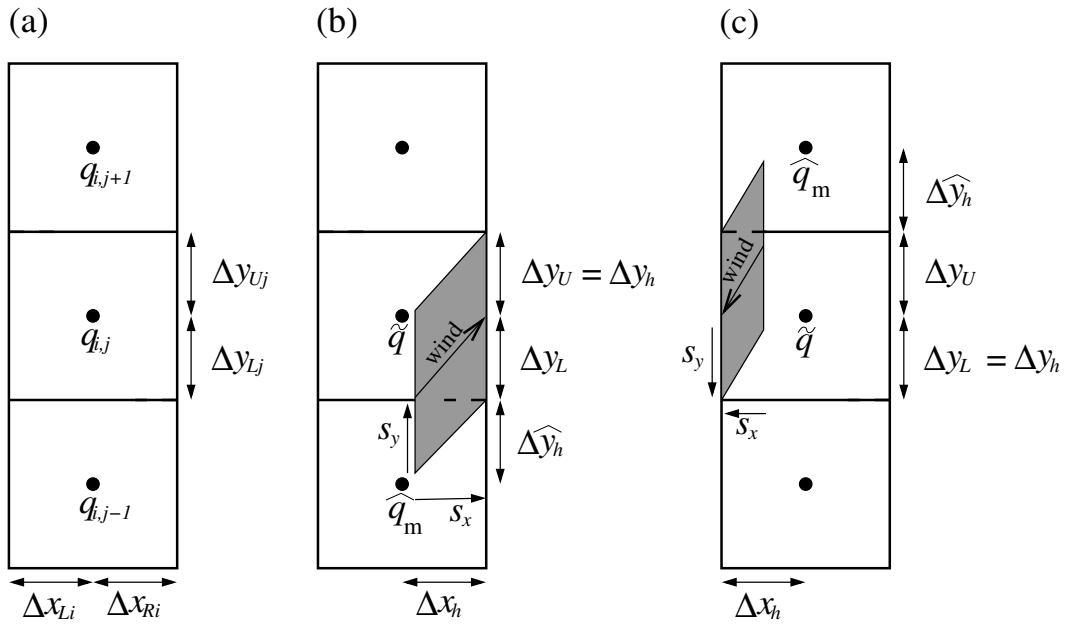

Figure 4.1: Schematic view of flux integration in x-direction. (a) scalar grid; integration over grey area $a = s_x\,(y_{j+1/2} - y_{j-1/2}) = s_x(\Delta y_U + \Delta y_L)$ when wind comes from (b) south-west or (c) north-east.

$$\overline{q}^x_{i,j} = \frac{1}{a}\int_{x_{i+1/2}-s_x}^{x_{i+1/2}}\int_{y_{j-1/2}-\delta y}^{y_{j+1/2}-\delta y}\tilde{Q}(x',y')d\,y'd\,x' \tag{4.3}$$

with the distance $s_x = u\Delta t$ the air moves in x-direction, $\delta y = (v/u)x'$ and the area $a = s_x\,(y_{j+1/2} - y_{j-1/2})$. Then, for example, the flux in x-direction is given by

$$F^x_{i+1/2,j} = \frac{u}{a}\int_{x_{i+1/2}-s_x}^{x_{i+1/2}}\int_{y_{j-1/2}-\delta y}^{y_{j+1/2}-\delta y}\tilde{Q}(x',y')d\,y'd\,x' = u \quad \overline{q}^x_{i,j}. \tag{4.4}$$

It is important to demand that the reconstruction is cell-average conserving, i.e.

$$\frac{1}{V_{i,j}}\int_{x_{i-1/2}}^{x_{i+1/2}}\int_{y_{i-1/2}}^{y_{i+1/2}}Q(x',y')d\,y'd\,x' = q_{i,j} \tag{4.5}$$

with the cell-volume $V_{i,j} = (x_{i+1/2} - x_{i-1/2})(y_{i+1/2} - y_{i-1/2})$. Only by this it is guaranteed that e.g. for $v = 0$ and $u = \Delta x/\Delta t$ the flux through the cell face is $uq_{i,j}$, which leads to a displacement of the field by one grid cell in x-direction.

The reconstructed field $\tilde{Q}(x', y')$ can be created by defining separate polynomials for each cell. The polynomials are only valid within the associated cell which can lead to discontinuities at the cell boundaries. For the third order scheme the polynomial $Q_{i,j}$ of cell (i, j) has the form:

$$\begin{aligned} Q_{i,j}(x,y) = q_{i,j} \quad &+ \quad A_{i,j}(x - x_i) + B_{i,j}(x - x_i)^2 \\ &+ \quad C_{i,j}(y - y_j) + D_{i,j}(y - y_j)^2 \end{aligned} \tag{4.6}$$

Here the missing $\tilde{\cdot}$ above Q indicates that $Q_{i,j}$ is not cell-average conserving.

For UTOPIA the polynomial reconstruction is based on centred interpolation so that the following conditions hold:

$$\begin{aligned} Q_{i,j}(x_k, y_j) &= q_{k,j} \quad \text{for} \quad k \in \{i-1, i, i+1\} \quad \text{and} \\ Q_{i,j}(x_i, y_l) &= q_{i,l} \quad \text{for} \quad l \in \{j-1, j, j+1\}. \end{aligned}$$

$Q_{i,j}$ interpolates through the cell value (i, j) and the values of its adjacent cells. A cell-averaging reconstruction $\tilde{Q}_{i,j}$ can be achieved by setting

$$\tilde{Q}_{i,j}(x,y) = Q_{i,j}(x,y) + q_{i,j} - \frac{1}{V_{i,j}}\int_{x_{i-1/2}}^{x_{i+1/2}}\int_{y_{i-1/2}}^{y_{i+1/2}}Q_{i,j}(x',y')d\,y'd\,x' \tag{4.7}$$

with the cell volume $V_{i,j}$. The resulting scheme is UTOPIA.

Since centred interpolation leads to under- and overshoots in the vicinity of discontinuities or sharp gradients (e.g. Thuburn, 1996), in this work the essentially

non-oscillatory method (ENO, Harten et al., 1987) as well as the weighted ENO method (WENO, Liu et al., 1994; Jiang and Shu, 1996; Shu, 1999) are used instead. They are applied in each dimension separately. With the ENO interpolation, e.g. in x-direction, for each cell (i,j) three parabolas are possible based on the stencils $\{i-2, i-1, i\}$, $\{i-1, i, i+1\}$ and $\{i, i+1, i+2\}$. The smoothest parabola is chosen so that interpolation across discontinuities is avoided. For WENO the three parabolas are weighted according to their smoothness. The resultant is based on a five-point stencil. The smoother the parabola the larger is its weight. Here the weights of Shu (1999) are applied.

The (W)ENO interpolation in each direction yields the coefficients $A_{i,j}, B_{i,j}, C_{i,j}$ and $D_{i,j}$ of (4.6). As mentioned before, the interpolants $Q_{i,j}$ are not cell-average conserving.

The schemes derived here are named FIENO3 (third order flux integrated ENO scheme) and FIWENO3 (third order flux integrated WENO scheme). The necessary calculations are given in the following:

Inserting the (W)ENO interpolation polynomial of cell (i,j) (4.6) into (4.7) yields the cell-average conserving reconstruction:

$$\tilde{Q}_{i,j}(x,y) = \tilde{q}_{i,j} + A_{i,j}(x-x_i) + B_{i,j}(x-x_i)^2 + C_{i,j}(y-y_j) + D_{i,j}(y-y_j)^2$$

where the cell-average conserving term is added to $q_{i,j}$ to get $\tilde{q}_{i,j}$:

$$\begin{aligned}
\tilde{q}_{i,j} = q_{i,j} &- \frac{1}{V_{i,j}} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{i-1/2}}^{y_{i+1/2}} Q_{i,j}(x',y')\,d\,y'\,d\,x' \\
= q_{i,j} &- 1/2 A_{i,j}(\Delta x_{Ri} - \Delta x_{Li}) \\
&- 1/2 C_{i,j}(\Delta y_{Uj} - \Delta y_{Lj}) \\
&- 1/3 B_{i,j}({\Delta x_{Ri}}^3 + {\Delta x_{Li}}^3)/(\Delta x_{Ri} + \Delta x_{Li}) \\
&- 1/3 D_{i,j}({\Delta y_{Uj}}^3 + {\Delta y_{Lj}}^3)/(\Delta y_{Uj} + \Delta y_{Lj}).
\end{aligned}$$

The meaning of Δx_{Ri} (R=right), Δx_{Li} (L=left), Δy_{Uj} (U=upper) and Δy_{Lj} (L=lower) is demonstrated in Fig. 4.1. Note that a nonuniformly spaced Arakawa C-grid is assumed so that it is possible that for e.g. the x-direction $\Delta x_{Li} \neq \Delta x_{Ri} \neq \Delta x_{Ri+1}$. This is the reason for introducing an upper/lower/right/left grid spacing within one cell (Fig. 4.1).

The final result of the scheme for the flux in x-direction is given using some abbreviations that are explained in the following. To be able to use the formula for all wind

directions, the neighbouring cell (in y-direction) that is involved gets the index m:

$$m = \begin{cases} j-1 & : \quad v > 0 \\ j+1 & : \quad \text{else} \end{cases} . \tag{4.8}$$

Furthermore, in certain terms either the upper y-half-width (Δy_{Uj}) or the lower half-width (Δy_{Lj}) of cells (i,j) and (i,m) is needed depending on the sign of v:

$$\Delta y_h = \begin{cases} \Delta y_{Uj} & : \quad v > 0 \\ \Delta y_{Lj} & : \quad \text{else} \end{cases} \qquad \Delta \widehat{y_h} = \begin{cases} \Delta y_{Um} & : \quad v > 0 \\ \Delta y_{Lm} & : \quad \text{else} \end{cases} . \tag{4.9}$$

Here and in the following the $\hat{\cdot}$ indicates that the value belongs to cell (i,m). In x-direction the sign of u decides, if the flux through the left or right face of cell (i,j) is calculated. Consequently, either Δx_{Ri} or Δx_{Li} is needed:

$$\Delta x_h = \begin{cases} \Delta x_{Ri} & : \quad u > 0 \\ \Delta x_{Li} & : \quad \text{else} \end{cases} . \tag{4.10}$$

s_x and s_y are the distance in x- and y-direction, respectively, the air moves during one time step. For both the sign indicates the direction of the flow:

$$s_x = u\Delta t \quad s_y = v\Delta t. \tag{4.11}$$

r_y is the fraction of the cell width in y-direction the air moves during one time step:

$$r_y = s_y/(\Delta y_U + \Delta y_L). \tag{4.12}$$

The remainder of the abbreviations serves the purpose to reduce the number of indices in the formula:

$$\begin{aligned} \Delta y_U &= \Delta y_{Uj} & \Delta y_L &= \Delta y_{Lj} \\ \tilde{q} &= \tilde{q}_{i,j} & \widehat{q} &= \tilde{q}_{i,m} \\ A &= A_{i,j} & \widehat{A} &= A_{i,m} \\ B &= B_{i,j} & \widehat{B} &= B_{i,m} \\ C &= C_{i,j} & \widehat{C} &= C_{i,m} \\ D &= D_{i,j} & \widehat{D} &= D_{i,m}. \end{aligned} \tag{4.13}$$

Using the above abreviations the average of $\tilde{Q}$ over the area advected through a cell

face in x-direction is given by:

$$\begin{aligned}
\overline{q}^x_{i,j} &= \frac{1}{a}\int_{x_{i+1/2}-s_x}^{x_{i+1/2}}\int_{y_{j-1/2}-\delta y}^{y_{j+1/2}-\delta y}\tilde{Q}(x',y')d\,y'd\,x' \\
&= \tilde{q}(1-1/2|r_y|) \\
&+ A[\mathsf{SIGN}(u)\Delta x_h - 1/2 s_x - \mathsf{SIGN}(u)|r_y|(1/2\Delta x_h - 1/3|s_x|)] \\
&+ C[1/2(\Delta y_U - \Delta y_L) - r_y(1/2\Delta y_h - 1/6|s_y|)] \\
&+ B[1/3 s_x^2 + \Delta x_h(\Delta x_h - |s_x|) - |r_y|(1/4 s_x^2 - 2/3\Delta x_h|s_x| + 1/2{\Delta x_h}^2)] \\
&+ D[1/3({\Delta y_U}^3 + {\Delta y_L}^3)/(\Delta y_U + \Delta y_L) - |r_y|(1/12 s_y^2 - 1/3\Delta y_h|s_y| + 1/2{\Delta y_h}^2)] \\
&+ 1/2\widehat{q}|r_y| \\
&+ \widehat{A}\mathsf{SIGN}(u)|r_y|(1/2\Delta x_h - 1/3|s_x|) \\
&+ \widehat{C} r_y(1/2\Delta\widehat{y_h} - 1/6|s_y|) \\
&+ \widehat{B}|r_y|(1/4 s_x^2 - 2/3\Delta x_h|s_x| + 1/2{\Delta x_h}^2) \\
&+ \widehat{D}|r_y|(1/12 s_y^2 - 1/3\Delta\widehat{y_h}|s_y| + 1/2{\Delta\widehat{y_h}}^2).
\end{aligned}$$

The area a does not explicitly appear in the final formula, because parts of it (either s_x or $\Delta y_U + \Delta y_L$ or both) cancel out or are contained in the fractional value r_y.

Fluxes in y-direction are calculated in the same way by exchanging x- and y-direction.

4.3 One-dimensional results

For the one-dimensional scheme the phase accuracy and wave amplification of FIENO3 and FIWENO3 are calculated and compared to the constant grid flux form (Tremback et al., 1987) which is identical to the QUICKEST scheme (Leonard, 1979). Table 4.1 summarises the compared schemes. The phase accuracy is an indicator for numerical dispersion, while the wave amplification factor is an indicator for stability and the scale dependent diffusifity of the scheme. The von Neumann stability analysis, necessary for the calculation of accuracy and wave amplification, cannot be done analytically, since FIENO3 and FIWENO3 are nonlinear advection operators. Moreover, phase accuracy and wave amplification are not constant but depending on the current phase shift of the wave. Therefore, on a domain with 40 grid cells and cyclic boundary conditions waves with $\lambda = 4\Delta x$ and $\lambda = 8\Delta x$ are prescribed for a scalar quantity. They are advected using different Courant numbers (CFL) for 60 time steps. Their relative phase speed and amplification factors are calculated

Table 4.1: Overview of different advection schemes applied in this section to scalar quantities.

Scheme	order	description
QUICKEST	3	Leonard (1979)
FIENO3	3	flux integrated ENO scheme
FIWENO3	3	flux integrated WENO scheme
FIENO2	2	flux integrated ENO scheme
E2UP	2	ENO scheme (Schroeder et al., 2006)
E3UP	3	WENO scheme (Schroeder et al., 2006)
UP1	1	upstream scheme

by Fourier analysis at each time step. The average amplification factor and average relative phase speed are given in Tables 4.2 and 4.3, respectively. For comparison, the results for the first order upstream scheme (UP1), the second order FIENO scheme (FIENO2) and the second (E2UP) and third (E3UP) order ENO schemes of Schroeder et al. (2006) are also given.

For the QUICKEST scheme the results are very similar to those given by Tremback et al. (1987) for the constant grid flux scheme. Differences are in the third digit and can be attributed to roundoff errors. For $8\Delta x$ waves the amplification factors (Table 4.2) and relative phase speeds (Table 4.3) are nearly identical for the three third order schemes. This is similarly valid for the amplification factors of $\lambda 4 = \Delta x$, though the QUICKEST scheme is slightly better than the others. For small Courant numbers the phase speed is most accurate for QUICKEST. For Courant numbers of 0.1 and larger the differences between the schemes are small and below 5%. The first order upstream scheme (UP1) is much less accurate than the other schemes. Especially $4\Delta x$ waves are damped much more for CFL numbers close to 0.5, whereas the phase speed becomes the more inaccurate the smaller the CFL number. Not surprisingly, concerning the amplification factor FIENO2 is somewhere between UP1 and FIENO3. For the CFL numbers 0.01 and 0.3, however, the phase speed is closer to 1 for FIENO2 than for FIENO3. E2UP and E3UP are not stable for all CFL numbers smaller than 1. Especially for E3UP this is a dramatic drawback as even using a the CFL number of 0.3 fails. For momentum advection it was shown that a CFL number of 0.3 is sufficient (Schroeder et al., 2006). Consequently, for scalar advection these schemes are not recommended.

Table 4.2: Amplification factor of $4\Delta x$ and $8\Delta x$ waves for different Courant numbers and advection schemes as determined numerically in a one-dimensional advection test (averaged over 60 time steps).

	CFL						
Scheme	0.01	0.10	0.30	0.50	0.70	0.90	1.00
				$4\Delta x$			
QUICKEST	0.997	0.966	0.908	0.884	0.908	0.966	1.000
FIENO3	0.998	0.958	0.890	0.866	0.893	0.958	1.000
FIWENO3	0.998	0.966	0.906	0.888	0.892	0.957	1.000
FIENO2	0.993	0.929	0.828	0.791	0.828	0.929	1.000
E2UP	0.993	0.932	0.853	0.866	0.965	unstable	unstable
E3UP	0.998	0.966	0.942	0.974	unstable	unstable	unstable
UP1	0.990	0.906	0.762	0.707	0.762	0.906	1.000
				$8\Delta x$			
QUICKEST	1.000	0.997	0.993	0.992	0.993	0.997	1.000
FIENO3	1.000	0.997	0.993	0.991	0.993	0.997	1.000
FIWENO3	1.000	0.999	0.997	0.996	0.996	0.998	1.000
FIENO2	0.999	0.988	0.972	0.966	0.972	0.988	1.000
E2UP	0.999	0.990	0.983	0.991	unstable	unstable	unstable
E3UP	1.000	1.000	unstable	unstable	unstable	unstable	unstable
UP1	0.997	0.973	0.936	0.924	0.936	0.973	1.000

As expected, the third order schemes are less diffusive than the lower order schemes. For simple wave advection it can be concluded that FIENO3 and FIWENO3 are as appropriate as the QUICKEST scheme. Their advantage will be visible when advecting square waves which can be decomposed into the sum of an infinite number of sine waves of different frequencies. For square waves the nonlinearity of the (W)ENO interpolation helps avoiding under- and overshoots. This is shown in the following section with two-dimensional simulations.

4.4 Two-dimensional advection test cases

Three two-dimensional test cases are presented in this section. These are based on solving Eq. (4.1) with $\rho = 1$. The first two cases have also been used by

Table 4.3: Relative phase speed of waves with $\lambda = 4\Delta x$ and $\lambda = 8\Delta x$ for different Courant numbers and advection schemes as determined numerically in a one-dimensional advection test (averaged over 60 time steps).

	CFL						
Scheme	0.01	0.10	0.30	0.50	0.70	0.90	1.00
				$4\Delta x$			
QUICKEST	0.852	0.879	0.945	1.000	1.023	1.013	1.000
FIENO3	0.696	0.864	0.958	1.000	1.019	1.015	1.000
FIWENO3	0.736	0.849	0.906	0.998	0.992	1.006	1.000
FIENO2	0.759	0.828	0.924	1.000	1.033	1.019	1.000
E2UP	0.759	0.842	0.948	1.000	0.989	unstable	unstable
E3UP	0.691	0.866	0.952	0.954	unstable	unstable	unstable
UP1	0.643	0.704	0.859	1.000	1.060	1.033	1.000
				$8\Delta x$			
QUICKEST	0.988	0.991	0.996	1.000	1.002	1.001	1.000
FIENO3	0.981	0.995	0.999	1.000	1.000	1.001	1.000
FIWENO3	0.995	0.992	0.990	0.991	0.994	0.998	1.000
FIENO2	0.990	0.984	0.993	1.000	1.003	1.002	1.000
E2UP	0.991	0.989	0.999	1.000	unstable	unstable	unstable
E3UP	0.980	0.993	unstable	unstable	unstable	unstable	unstable
UP1	0.903	0.926	0.970	1.000	1.013	1.008	1.000

Thuburn (1996). They are based on diagonal advection in a domain with 31x31 grid points and cyclic boundary conditions. A Gaussian cone of $3\Delta x$ standard deviation (GAUSSIAN) and a square step profile of width $15\Delta x$ (SQRSTEP) are advected with a Courant number of 0.25 in x- and y-direction. Results are evaluated after a full advection cycle, corresponding to 124 time steps (Fig. 4.2).

For UTOPIA and FIWENO3 spurious oscillations are visible within the domain. Note that Thuburn (1996) has calculated slightly different results for UTOPIA. The reason is that he left out some higher order terms (the last two terms in Eq. (39) in Leonard et al., 1995), while in this work these terms are kept. Table 4.4 lists minimum and maximum field values as well as common error norms for the three schemes showing that they are similarly accurate, but undershoots are three orders of magnitude larger for UTOPIA than for FIENO3 and FIWENO3, respectively. Here, $||\cdot||_\infty$ is the maximum norm, $||\cdot||_1$ the L1-norm and $||\cdot||_2$ the L2-norm.

Table 4.4: Minimum and maximum values as well as common error norms for different two-dimensional advection tests. GAUSSIAN: diagonal advection of the Gaussian cone after 124 time steps. SQRSTEP: diagonal advection of the square step signal after 124 time steps. ROT: advection of the Gaussian cone with maximum 1 in a rotational flow field after 600 time steps (one revolution). In all cases the initial maximum value of q is 1.

test case	Scheme	min	max	$\|\|\cdot\|\|_\infty$	$\|\|\cdot\|\|_1$	$\|\|\cdot\|\|_2$
GAUSSIAN	UTOPIA	-3.53E-03	0.884	0.120	7.554E-03	1.871E-02
GAUSSIAN	FIENO3	-6.45E-06	0.887	0.126	7.226E-03	1.859E-02
GAUSSIAN	FIWENO3	-2.56E-05	0.954	0.092	5.958E-03	1.606E-02
SQRSTEP	UTOPIA	-5.97E-02	1.115	0.672	8.163E-02	0.140
SQRSTEP	FIENO3	-3.22E-05	1.000476	0.693	8.491E-02	0.158
SQRSTEP	FIWENO3	-1.33E-04	1.000686	0.697	7.510E-02	0.146
ROT	UTOPIA	-8.685E-03	0.807	0.1892	2.622E-03	1.254E-02
ROT	FIENO3	8.254E-25	0.807	0.1896	2.349E-03	1.220E-02
ROT	FIWENO3	-2.170E-06	0.936	0.0772	9.977E-04	5.445E-03

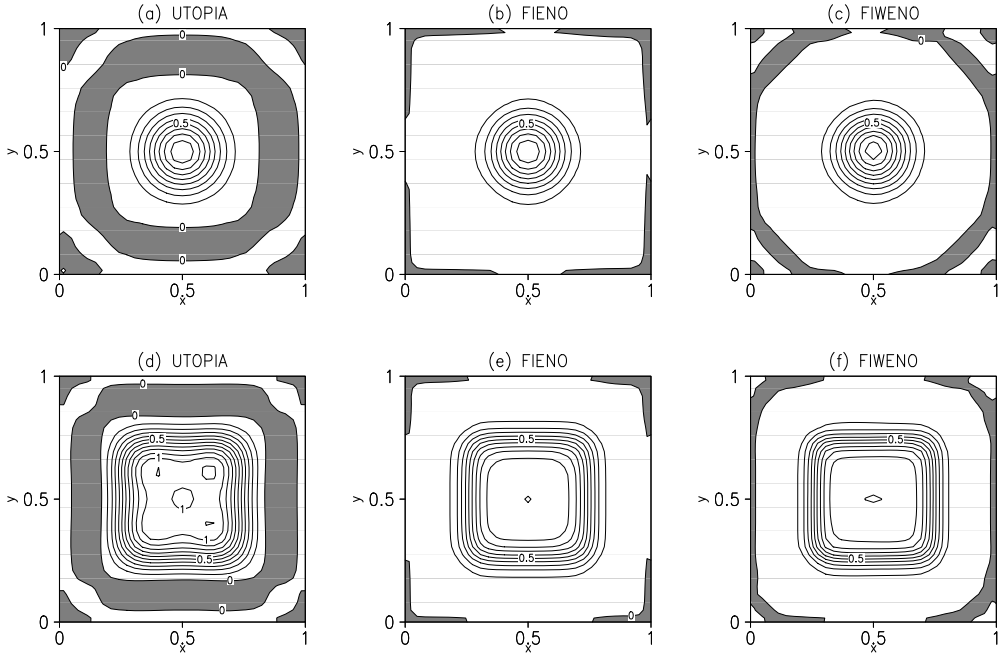

Figure 4.2: A Gaussian cone (a, b, c) and a square step profile (d, e, f) are diagonally advected with a Courant number of 0.25 using cyclic boundary conditions. Results are given with contour increment 0.1 after 124 time steps for UTOPIA (a,d), FIENO3 (b,e) and FIWENO3 (c,f). Areas with negative values are shaded.

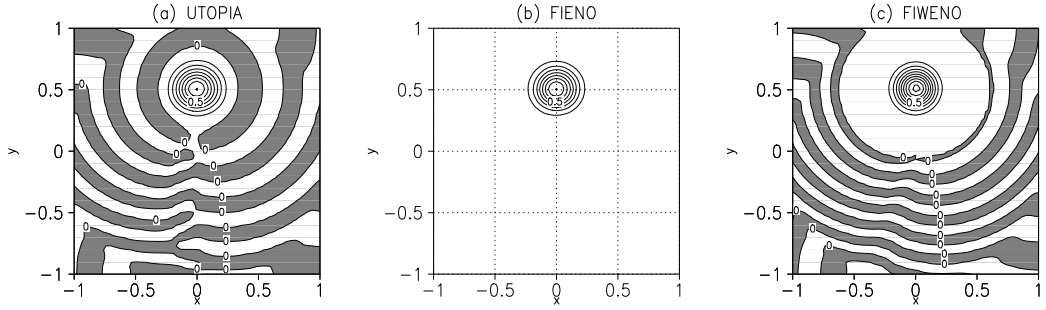

Figure 4.3: A Gaussian cone is advected in a rotational flow field. Shown is the field after one revolution (600 time steps) for (a) UTOPIA, (b) FIENO3 and (c) FIWENO3 (contour increment 0.1). Areas with negative values are shaded.

The third test has a rotational flow field with a constant angular velocity (ROT). On a domain with 61x61 grid cells a Gaussian cone initially centred at grid point (31, 16) is advected. The results after 600 time steps (one revolution) are shown in Fig. 4.3 with error norms in Tab. 4.4. Again, UTOPIA and FIWENO3 show spurious oscillations. FIENO3 remains positive during the integration. The undershoots of FIWENO3 are, however, three orders of magnitude smaller than for UTOPIA. Compared to UTOPIA and FIENO3 the $||\cdot||_\infty$ error is reduced by a factor of two. This accuracy is gained since FIWENO3 is based on a real five-point stencil.

4.5 Rising bubble experiment

Another common test for numerical schemes is the rising bubble experiment (e.g. Smolarkiewicz and Pudykiewicz, 1992; Robert, 1993; Laprise and Plante, 1995). For these tests the scalar advection schemes are implemented in the anelastic model METRAS (Schlünzen, 1990; Dierer et al., 2005). The METRAS model equations consist of three equations for momentum conservation with the Boussinesq approximation, a potential temperature conservation equation, the perfect gas law and a mass conservation equation in anelastic form. All equations are solved in flux form. The anelastic approximation results in a divergence free momentum field which is guaranteed by a pressure correction in the model. The pressure correction is calculated

iteratively with Bi-CGSTAB (Van der Vorst, 1992), therefore small residual divergences cannot be avoided. For this two-dimensional experiment Coriolis force and all physical parameterisations (e.g. turbulence parameterisation, cloud processes, radiation) are omitted. Density effects are included. Momentum advection is solved with E3UP (Schroeder et al., 2006). The domain extends over 2000 m in horizontal as well as in vertical direction. The grid spacing is 10 m in both directions. The time step is 0.5 s. The potential temperature θ is separated by $\theta(x,z) = \theta_0(z) + \tilde{\theta}(x,z)$ into a background part θ_0 and a mesoscale deviation $\tilde{\theta}(x,z)$. The background potential temperature is 300 K throughout the atmosphere. The centre of the bubble (diameter 500 m) is located 290 m above the ground at $x = 0$ m with a temperature excess of 0.5 K. The advection operators are only applied to the mesoscale potential temperature deviation.

The results (Fig. 4.4) are very similar to what can be found in literature. Due to the strong wind shear billows develop at the temperature discontinuity. However, all three schemes produce potential temperatures below 300K and above 300.5K. This is physically not plausible. For UTOPIA and FIWENO3 this is associated with spurious oscillations. After 500 s overshoots for UTOPIA (approximately 13.6% of the initial temperature excess of 0.5 K) are larger than for FIWENO3 (2.1%) and FIENO3 (0.2%). These overshoots are not only caused by the advection scheme but

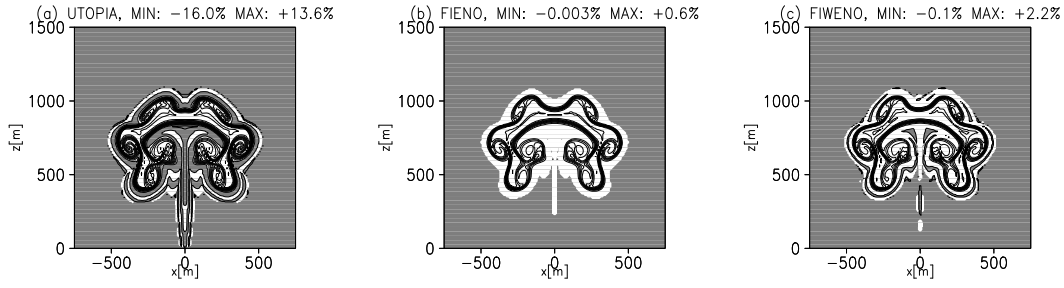

Figure 4.4: Potential temperature of the rising bubble experiment after 500 s for (a) UTOPIA, (b) FIENO3 and (c) FIWENO3 (contour increment 0.05K). Minimum (maximum) values are given on top of the figure as negative (positive) deviations from 300K (300.5K) relative to the initial temperature excess of the bubble (0.5K). Areas with potential temperatures below 300K are shaded.

also by the momentum field. The latter is not completely divergence free as required for (4.1), as the divergences are corrected by an iterative pressure solver that gives only an approximate solution. However, for FIWENO and UTOPIA, this effect is negligible, as the total overshoots are of greater magnitude than for FIENO3. Therefore, for FIWENO3 and UTOPIA the overshoots must mainly be caused by the advection schemes themselves. The undershoots occur in areas with a smoother wind field where rest divergences are less problematic. Here, they are much smaller for FIENO3 (-0.003%) and FIWENO3 (-0.1%) while for UTOPIA (-16.0%) they are approximately as large as the overshoots.

When positive quantities like cloud water content or chemical concentrations are simulated, negative values caused by undershoots are a problem. To avoid negative values a correction option is included in the schemes. A quantity q is corrected as

$$q_i^c = \max(q_i, L_q) \frac{\sum_i q_i m_i}{\sum_i \max(q_i, L_q) m_i} \tag{4.14}$$

where m_i is the mass of air in cell i. L_q is the lower limit of the quantity q. Fig. 4.5 shows the same result as Fig. 4.4 but with correction (4.14) after each execution of the advection operator. Here the mesoscale deviation of potential temperature $\tilde{\theta}$ from 300K is corrected to be positive ($L_{\tilde{\theta}} = 0$). For FIWENO3 and FIENO3 the results for corrected and uncorrected simulations are almost the same with very small damping of the positive values. The damping of the extrema is less than 0.004 % for FIWENO3 and about three orders of magnitude smaller for FIENO.

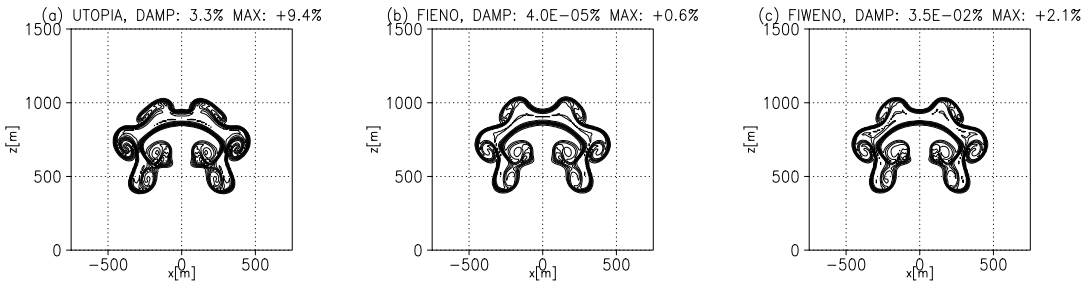

Figure 4.5: As Fig. 4.4, but with temperatures below 300K corrected by using Eq. (4.14). The effective damping of positive values due to the correction is given on top of the figures as well as maximum deviations from 300.5K relative to the initial temperature excess of the bubble (0.5K).

However, for UTOPIA the damping reaches over 3% after 500s, and the overshoots still have a magnitude of more than 9% of the initial temperature excess. This shows that Eq. (4.14) is an effective solution for preserving positivity only if the negative values produced by the advection schemes are very small.

4.6 Application for momentum advection

The described schemes can not only be applied to scalar quantities but also to momentum advection. Schroeder et al. (2006) have shown that schemes based on the (weighted) essentially non-oscillatory (ENO) interpolation are suitable for momentum advection with uniform and nonuniform grids. These schemes are not based on flux integration, but interpolate the advected wind to the cell face and use this value for flux calculation. For very small time steps the schemes nearly coincide, i.e. FIENO3≈ E3UP, as the average of advected quantity over the area a close to the cell face (Fig. 4.1) consists roughly of the cell face value only. However, when it comes to larger CFL numbers, the schemes differ. The non-flux integrated schemes have the main drawback that they restrict the time step with a CFL number smaller than 1 (0.63 for E2UP and 0.3 for E3UP, Schroeder et al., 2006). In contrast to that FIENO2 and FIENO3 have a CFL number of 1 in one dimension.

The suitability of FIENO for momentum advection is tested with a density current experiment (e.g. Skamarock and Klemp, 1993; Carpenter et al., 1990; Schroeder et al., 2006). The two-dimensional atmosphere extends over 40km in the horizontal (y-direction) and 10km in the vertical. The vertical and horizontal grid spacing is 125m. In the beginning the air is neutrally stratified with a background potential temperature of θ_0 =300K and zero wind. A block of cold air located at $y < 0$km is the driving force. Its potential temperature is decreasing linearly from 300K in 5km height to 290K at the surface:

$$\tilde{\theta} = \begin{cases} 290\ \mathrm{K} + \dfrac{10\ \mathrm{K}}{5000\ \mathrm{m}} z & \text{for} \quad z \leq 5000\ \mathrm{m} \quad \text{and} \quad y \leq 0\ \mathrm{m}; \\ 300\ \mathrm{K} & \text{elsewhere.} \end{cases}$$

Figure 4.6 shows the potential temperature after 15min simulation time for different advection schemes with time step Δt = 1s. When first order upstream is used for temperature and momentum (Fig. 4.6a), a smooth cold front has propagated 15km northward. The result with second order FIENO advection for scalar and momentum quantities shows two distinct billows (4.6b) caused by Kelvin-Helmholtz instability.

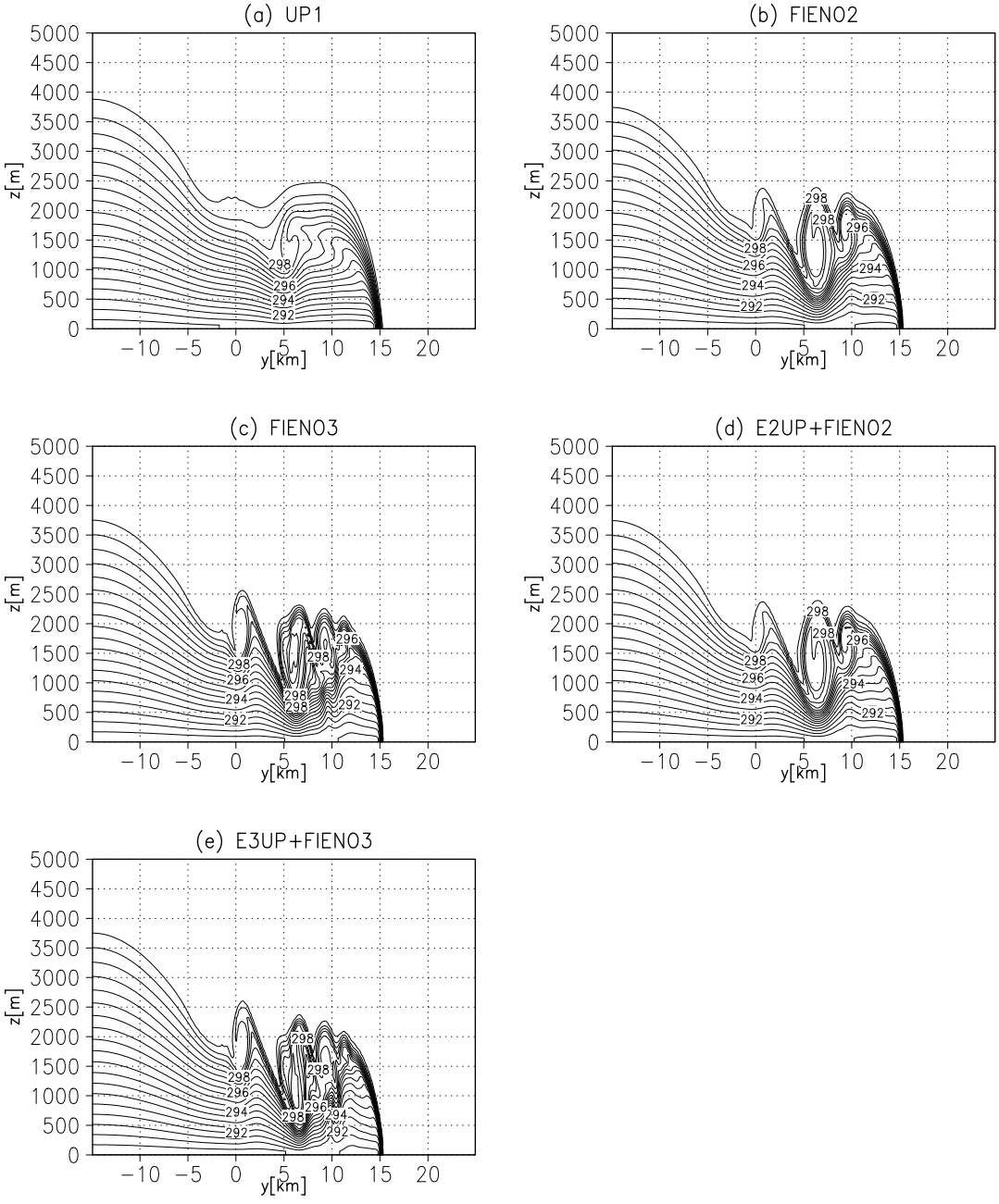

Figure 4.6: Potential temperature of the density current experiment after 15 min for (a) first order upstream, (b) FIENO2, (c) FIENO3, (d) E2UP for momentum and FIENO2 for temperature, (e) E3UP for momentum and FIENO3 for temperature (increment 0.5K). For all experiments $\Delta t = 1$s which corresponds to a maximum CFL number of 0.25.

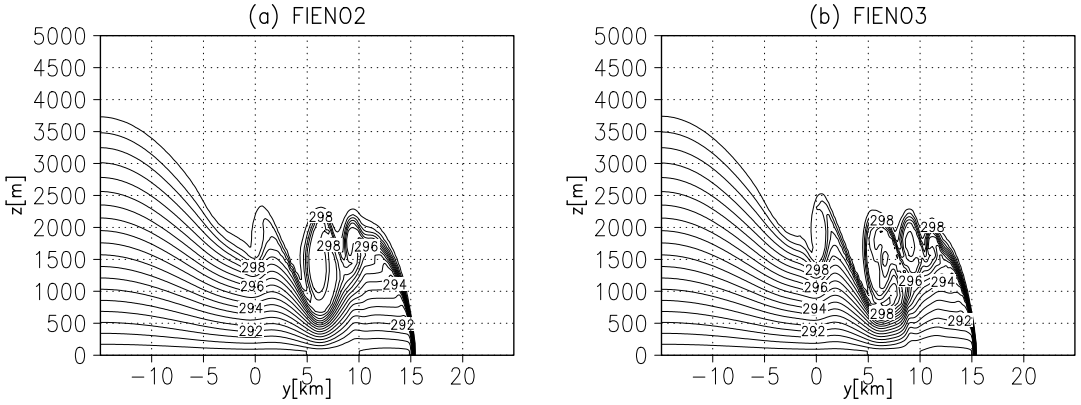

Figure 4.7: As Fig. 4.6 but with $\Delta t = 3$s which corresponds to a maximum CFL number of 0.75. (a) FIENO2, (b) FIENO3.

When FIENO2 is only applied to potential temperature and E2UP to momentum advection (Fig. 4.6d), the potential temperature is nearly indistinguishable from the results where FIENO2 is used for scalar and momentum advection. The same is true for the third order scheme. The result with FIENO3 for momentum and scalar advection (Fig. 4.6c) looks very similar to the result with FIENO3 for momentum and E3UP for scalar advection (Fig. 4.6e). The higher order schemes produce three to four Kelvin-Helmholtz billows. The main billow has a slightly different structure in both versions, but the main flow properties are the same. The four billows have developed at the same locations for both schemes.

When the time step is increased to $\Delta t = 3$s, the simulations applying the ENO schemes for momentum advection fail as high velocities develop ($v_{\max} > 30\text{ms}^{-1}$ <=> CFL > 0.7). The FIENO results are shown in Fig. 4.7. They differ only slightly from the $\Delta t = 1$s simulations. Only the main billow for the third order scheme has a different structure being more distinct from its northern neighbour in the $\Delta t = 3$s simulation. Consequently, it is recommended to use FIENO instead of ENO for advection when the time step is constrained by advection. Tests have also indicated that FIENO is similarly able to cope with changes in grid spacing as ENO (not shown here) which makes FIENO also suitable for simulations with grid refinement.

4.7 Summary and conclusions

In this work the flux integrated method for advection has been combined with the essentially non-oscillatory (ENO) interpolation and the weighted ENO interpolation (WENO). (W)ENO schemes have been shown to be very suitable for momentum advection especially in conjunction with discontinuous grid spacing (Schroeder et al., 2006). However, they require a CFL number smaller than one and are not suitable for scalar advection. This drawback is overcome with the flux integration method that was originally used by Leonard et al. (1995) for the derivation of UTOPIA. The combination of (W)ENO interpolation and flux integration that has been developed in this work leads to two new schemes (FIENO and FIWENO). The new schemes have been analysed in terms of phase speed and amplification factor for one-dimensional wave propagation and are compared to other schemes like first order upstream and QUICKEST. The latter is the one-dimensional version of UTOPIA. The new schemes are shown to be similarly accurate as the QUICKEST scheme of Leonard (1979).

Simple two-dimensional advection tests show that over- and undershoots produced by the new multidimensional schemes are between two and three orders of magnitude smaller than for the comparable UTOPIA scheme. The third order FIENO3 scheme produces no spurious oscillations. A rising bubble experiment in an anelastic model shows that for UTOPIA the intensity of oscillations reaches about 16% of the initial temperature excess of the bubble. For the third order FIWENO3 scheme overshoots are about 6 times smaller, while FIENO3 remains free of oscillations, even though temperatures below the initial 300K occur (0.003% of the initial temperature excess). None of the schemes conserves positivity.

For the discretisation of scalar advection FIENO3 is the most recommended of all three schemes. It is successfully used for temperature advection. Even though simple tests show that FIWENO3 is slightly more accurate its usage is not recommended, because it has a too oscillatory nature. For advection of positive quantities FIENO3 should be used in conjunction with a correction scheme like Eq. (4.14) to avoid undershoots. This adjustment is justifiable as in the presented test case the correction scheme had almost no effect on positive values.

It should be pointed out that Jiang and Shu (1996) have shown that when used with Runge-Kutta time integration, formally, WENO schemes have a higher accuracy than ENO schemes (eg. Jiang and Shu, 1996). This is not a contradiction to the

results found here, since in this work a complete different methodology is used for time integration and only the spatial interpolation is based on (W)ENO. Time is treated continuously by the integration of fluxes in contrast to Runge-Kutta time integration. For classical WENO schemes the weights are calculated by optimisation under consideration of the Runge-Kutta time integration (e.g. Jiang and Shu, 1996), but they are not necessarily optimal for flux integration.

For momentum advection FIENO was successfully applied in a density current experiment producing results that are nearly identical to the non-flux integrated ENO scheme of Schroeder et al. (2006) for small CFL numbers. However, with FIENO larger time steps are possible (up to CFL number one) and therefore it is more recommended than the comparable ENO scheme if the time step is dominated by advection.

5 Numerical dispersion of gravity waves

This chapter will be submitted to *Mon. Wea. Rev.* as:

Schroeder, G. and Schlünzen, K. H. (2007): Numerical dispersion of gravity waves.

Since the chapter is intended for publication, the METRAS equations are written in a simplified way. Only the dynamical core of METRAS is needed here so that Reynolds averaging is not applied (indicated by the missing $\overline{\cdot}$). Furthermore, the equations are not transformed into the surface following coordinate system, and all geometric coefficients are omitted (e.g. α^*).

Summary

When atmospheric gravity waves are simulated in numerical models, they are not only dispersive for physical but also for numerical reasons. Their wave properties (e.g. damping or propagation speed and direction) can depend on grid spacing as well as on the numerical schemes.

In this work numerical dispersion relations for atmospheric gravity waves are theoretically derived as well as experimentally measured using the anelastic mesoscale model METRAS. Both, the theoretical solution and the numerical model show a retardation of the gravity waves with decreasing grid spacing. Furthermore, the influence of a Shapiro seven-point filter is analysed. The Shapiro seven-point filter causes damping of the shorter waves. Therefore, shorter waves can only be simulated without the seven-point filter. The influence of different advection schemes is analysed by prescribing a background wind. A second order flux integrated essentially non-oscillatory (FIENO) scheme is superior to a corresponding third and first order scheme concerning the group velocity. The damping is the smaller the higher the order of the scheme, as expected.

The numerical dispersion has severe consequences, when nonuniform grid spacing is used. Waves moving from the fine grid to the coarse are reflected due to numerical dispersion if they are only poorly resolved on the coarse grid. In tests with different refinement factors and wave lengths the reflection is found to be the larger the greater the refinement factor. The results show that refinement factors larger than three should not be used with nonuniform grid spacing or two-way nested grids.

5.1 Introduction

Solutions of the Navier-Stokes equations can include several types of waves. For the atmospheric flow internal gravity waves are especially important. These waves can be found on different scales throughout the atmosphere wherever it is stably stratified. They can be excited by e.g. mountains, convection, shear instability and many more processes (Holton, 1992; Zhang, 2004). In atmospheric models, however, gravity waves can also be generated by dynamic and thermodynamic imbalances (e.g. caused by numerical noise or unbalanced initial conditions). Their group velocity depends on their wave length and the buoyancy frequency. When gravity waves propagate upward, their amplitude increases with decreasing density (e.g. Zhang

and Yi, 2004). In the troposphere the wave energy density is highest close to the tropopause (Allen and Vincent, 1995). They play also an important role in the middle atmosphere for energy and momentum transport (e.g. in the quasi-biennial oscillation, Dunkerton, 1997), and they are a dominant source for turbulence due to gravity wave breaking and dissipation (cf. Fritts and Alexander, 2003, for a review of middle atmospheric gravity waves).

As gravity waves influence the mesoscale flow (e.g. Zhang, 2004), their accurate simulation is of great interest. One of the problems associated with the simulation of gravity waves is their dispersive property. The physical dispersion is mainly influenced by the static stability of the air. However, in numerical models the type of grid spacing as well as grid size and the numerical methods used additionally influence the properties of the waves. If different grid sizes are used within one domain, gravity waves that cannot be resolved on the coarse grid cannot leave the fine grid and are reflected at the internal boundary. In the most extreme case where the wave cannot be resolved on the coarse grid it is trapped within the fine grid. Consequently, the stretching factor between fine and coarse grid is usually kept smaller than four (Zhang et al., 1986) which still leaves some reflection. While in many publications factors of four or smaller are used for grid refinement, to our knowledge an actual thorough analysis of numerical gravity wave dispersion and reflection and the influence of the refinement factor thereupon has never been published. Wave reflection due to grid refinement has only been analysed for simple wave problems like the advection of a wave structure (e.g. Vichnevetsky, 1987) which is relevant for atmospheric models. In the field of general relativity theory this effect has also been investigated for gravitational waves by Choi et al. (2004).

Since high resolution atmospheric simulations on large domains are still not feasible in terms of computational cost, two-way nesting and adaptive grid methods are frequently used accompanied by nonuniform grid spacing (e.g. Lorenz and Jacob, 2005; Bacon et al., 2007). Consequently, a justification of the acceptable refinement factors is needed, which can be achieved by quantifying gravity wave reflection. This is the motivation for this work, in which the numerical dispersion properties of gravity waves are analysed. Numerical simulations of gravity waves on different grids are presented for different numerical methods. Section 2 describes the non-hydrostatic model METRAS used for the simulation. Section 3 shortly describes analytical properties of gravity waves as found in text books and then presents a new theoretical derivation of the numerical dispersion related to grid refinement. Section 4 analyses the numerical dispersion with numerical simulations in METRAS.

Furthermore, the influence of different numerical advection schemes is investigated. The consequences of grid refinement, namely reflection, are shown in Section 5. The intensities of reflected waves are measured in different simulations. Conclusions are drawn in Section 6.

5.2 Model METRAS

The mesoscale model METRAS (Schlünzen, 1990; Dierer et al., 2005) is used for testing the properties of gravity waves under different numerical conditions. METRAS is a three-dimensional, non-hydrostatic model in which the three wind components u, v, w and the potential temperature $\theta = T(\frac{100000Pa}{p})^{R/c_p}$ (with the temperature T, the specific gas constant R for dry air and the specific heat capacity of dry air at constant pressure c_p) as well humidity, cloud liquid water content and rain water content are calculated from prognostic equations. The equations are solved in flux form. The pressure p is calculated diagnostically. A perfect gas law relates density ρ to the potential temperature θ and pressure. Turbulence, radiation and cloud processes are parameterised.

Since for this work only the dry, dynamical core of the model is applied without frictional effects, Coriolis force and the parameterisation of turbulence, in the following only the simplified model equations are stated without the overbar commonly used to denote Reynolds-averaged quantities. Furthermore, all simulations presented here are two-dimensional, so that the equation for v and dependencies on y are omitted. In the equations of motion the Boussinesq approximation and the anelastic approximation are applied. It is assumed that $\rho \approx \rho_0$ except for the buoyancy term which nearly cancels out with the vertical pressure gradient. The anelastic approximation assumes that density changes slowly with time. To make use of major balances in the atmosphere, the thermodynamic quantities are separated into a background part (subscript 0, only depending on z) and a deviation part (denoted by $\tilde{\cdot}$):

$$\begin{aligned} p(x,z) &= p_0(z) + \tilde{p}(x,z) \\ \theta(x,z) &= \theta_0(z) + \tilde{\theta}(x,z) \\ \rho(x,z) &= \rho_0(z) + \tilde{\rho}(x,z) \end{aligned}$$

with $\tilde{p}(x,z) = p_1(x,z) + p_2(x,z)$. p_0 and ρ_0 as well as p_1 and $\tilde{\rho}$ satisfy the hydrostatic equation

$$\frac{\partial p_0}{\partial z} = -g\rho_0; \; \frac{\partial p_1}{\partial z} = -g\tilde{\rho}$$

with the gravity acceleration g. p_2 is the dynamic pressure calculated diagnostically from a Poisson equation to keep the momentum field divergence free.

This leads to two momentum equations

$$\begin{aligned}\frac{\partial \rho_0 u}{\partial t} &= -\frac{\partial}{\partial x}(u\rho_0 u) - \frac{\partial}{\partial z}(w\rho_0 u) - \frac{\partial p_1 + p_2}{\partial x} \\ \frac{\partial \rho_0 w}{\partial t} &= -\frac{\partial}{\partial x}(u\rho_0 w) - \frac{\partial}{\partial z}(w\rho_0 w) - \frac{\partial p_2}{\partial z} \ ,\end{aligned}$$

a thermodynamic energy equation

$$\frac{\partial \rho_0 \theta}{\partial t} = -\frac{\partial}{\partial x}(u\rho_0 \theta) - \frac{\partial}{\partial z}(w\rho_0 \theta) \ ,$$

the continuity equation

$$0 = \frac{\partial}{\partial x}(u\rho_0) + \frac{\partial}{\partial z}(w\rho_0)$$

and a linearised perfect gas law

$$\frac{\tilde{\rho}}{\rho_0} = -\frac{\tilde{\theta}}{\theta_0} + \frac{c_v}{c_p}\frac{p_1 + p_2}{p_0}.$$

Here, c_v is the specific heat capacity of dry air at constant volume.

For the results presented here advection of potential temperature and momentum is solved with first order upstream (UP1) or a second (FIENO2) or a third (FIENO3) order flux integrated essentially non-oscillatory scheme (Schroeder and Schlünzen, 2007a). Pressure p_2 is solved from a hyperbolic Poisson equation by an ILU(1) preconditioned Bi-CGSTAB algorithm (Van der Vorst, 1992).

5.3 Theoretical analysis - properties of gravity waves on grids of different resolution

The linearised model equations yield gravity waves as solution. The derivation of the physical dispersion relation of linear gravity waves can be found in many text books (e.g. Holton, 1992). In this section, the results are briefly stated. The linearisation of the velocities $u = \overline{u} + u'$ and $w = w'$ leads to a wave equation for the vertical deviation velocity w'

$$\frac{\partial^2}{\partial t^2}\left(\frac{\partial^2 w'}{\partial x^2} + \frac{\partial^2 w'}{\partial z^2}\right) + N^2 \frac{\partial^2 w'}{\partial x^2} = 0 \tag{5.1}$$

Note that in this case w' is not associated with turbulence, but with the deviation part that arises from the linearisation. This is a common notation in dynamic meteorology. It can be shown that the group velocity depends on the vertical (λ_z) and horizontal (λ_x) wave lengths as well as on the static stability (buoyancy frequency) $N = \sqrt{g\partial \ln\theta/\partial z}$. The horizontal group velocity for a solution to this equation is given by

$$c_a = N\frac{m^2}{(k^2+m^2)^{1.5}} \tag{5.2}$$

with the horizontal wave number $k = 2\pi\lambda_x^{-1}$ and the vertical wave number $m = 2\pi\lambda_z^{-1}$.

While the derivations above can be found in standard text books, to our knowledge a theoretical derivation of the numerical dispersion relation of gravity waves has never been published. The numerical dispersion relation can be calculated in two steps. Firstly, Eq. (5.1) is discretised spatially with the horizontal and vertical grid spacing Δx and Δz, respectively,

$$\left(\frac{\partial^2}{\partial t^2}+N^2\right)\left(\frac{w'_{j+1,l}-2w'_{j,l}+w'_{j-1,l}}{\Delta x^2}\right)+\frac{\partial^2}{\partial t^2}\left(\frac{w'_{j,l+1}-2w'_{j,l}+w'_{j,l-1}}{\Delta z^2}\right)=0 \tag{5.3}$$

Secondly, a wave function $w'_{j,l}(t)$ is inserted in Eq. (5.3) (similarly to a von Neumann stability analysis):

$$w'_{j,l}(t) = \hat{w}\exp(i[kj\Delta x + ml\Delta z - \nu t])$$

with the frequency ν and the amplitude $\hat{w}$. This yields

$$(-\nu^2+N^2)\frac{\exp(ik\Delta x)-2+\exp(-ik\Delta x)}{\Delta x^2}-\nu^2\frac{\exp(im\Delta z)-2+\exp(-im\Delta z)}{\Delta z^2}=0.$$

Solving this for ν leads to

$$\nu = N\left(\frac{c_k}{c_k+c_m}\right)^{0.5}$$

with the abbreviations

$$c_k = \frac{1-\cos(k\Delta x)}{\Delta x^2}$$
$$c_m = \frac{1-\cos(m\Delta z)}{\Delta z^2}.$$

The numerical horizontal group velocity in x-direction is

$$c_{xn} = \frac{\partial\nu}{\partial k} = 0.5Ns_k\left\{[c_k(c_k+c_m)]^{-0.5}-c_k^{0.5}(c_k+c_m)^{-1.5}\right\} \tag{5.4}$$

where the abbreviation

$$s_k = \frac{\sin(k\Delta x)}{\Delta x}$$

has been used. From (5.4) it is clear that for $\Delta x \approx \lambda_x/2$ the numerical horizontal group velocity tends against zero. In this case the waves are stationary.

5.4 Model analysis - properties of gravity waves

Zhang and Yi (2004) analysed the propagation of gravity wave packets in a global atmospheric model. Their method to initialise gravity waves is also used in this work to investigate the numerical dispersion of gravity waves with METRAS. The hydrostatic background atmosphere is assumed to be isothermal:

$$T_0 = T_c;\ \rho_0 = \rho_c \exp\left(-\frac{gz}{RT_c}\right).$$

This leads to a stable stratification where θ_0 increases with height. The horizontal wind field perturbation is initialised by

$$u'(x,z,t=0) = u_c \exp\left[-\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(z-z_0)^2}{2\sigma_z^2}\right] \cos\left[(x-x_c)\frac{2\pi}{\lambda_x} + (z-z_0)\frac{2\pi}{\lambda_z}\right].$$

where (x_c, z_c) =(-30km,3km) is the centre of the wave packet. For the analysis the following values are chosen: $T_c = 250$K, $u_c = 0.8\text{ms}^{-1}$, $\sigma_x = \lambda_x = 15$km and $\sigma_z = \sigma_z = 1$km. In this case the static stability is $N = 0.02s^{-1}$ and the analytical horizontal group velocity is $c_a = 3.09\text{ms}^{-1}$. Figure 5.1 shows the initial conditions for u. The amplitudes of the initial waves are kept small in order to

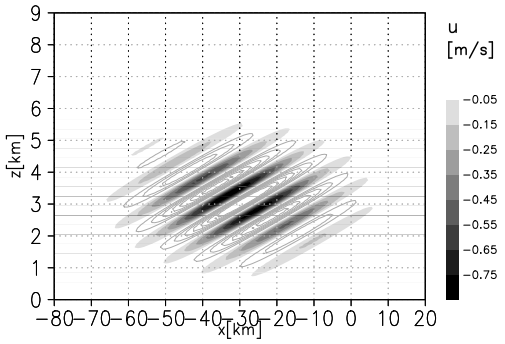

Figure 5.1: Initial condition for the horizontal wind u. Contours are 0.1 ms^{-1}. Negative contours are shaded.

avoid non-linear effects like wave breaking. The initial vertical wind, deviation pressure, deviation density and deviation temperature are calculated with phase shifts to match the linear wave solution with a group velocity pointing upward and in positive x-direction.

The model results are analysed in terms of the wave energy density which is given by Zhang and Yi (2004)

$$I = \frac{1}{2}\rho_0(u'^2 + w'^2) + \frac{(p_1 + p_2)^2}{2\rho_0 c_a^2} + \frac{(p_1 + p_2 - \rho' c_a^2)^2}{2\rho_0(1 - \frac{c_p}{c_v})c_a^2} \tag{5.5}$$

In this work the wave energy density is averaged over one wave length in each spatial direction to get a smoother field:

$$I'(x,z) = \int_{x'=x-\lambda_x/2}^{x'=x+\lambda_x/2} \int_{z'=z-\lambda_z/2}^{z'=z+\lambda_z/2} I(x',z')d\,x'd\,z' \tag{5.6}$$

At each time step the maximum of the smoothed wave energy density, I', is located, and its location and intensity stored. With a regression method the group velocity and the e-folding time (damping coefficient) of the wave intensity are estimated. It should be noted that the measured group velocity and damping can have an error of about 10% for highly damped waves. This estimation is based on a linear regression.

For all simulations the vertical mesh size is $\Delta z = \lambda_z/10$, the horizontal mesh size is varied from $\Delta x = \lambda_x/3$ to $\Delta x = \lambda_x/20$. An additional test simulation with $\Delta x = \lambda_x/2$ yields gravity waves that dissipate too rapidly to be analysed.

5.4.1 Background wind $\overline{u} = 0$ ms^{-1}

The simulations with zero background wind ($\overline{u} = 0$) are performed with a third order flux integrated essentially non-oscillatory advection scheme (FIENO3, Schroeder and Schlünzen, 2007a). Two types of simulation are made - one where the wind fields are filtered after each time step (FIENO3F) and one where the filter is omitted (FIENO3). The filter is a Shapiro (1971) seven-point filter written in flux form (comparable to 6th order hyperdiffusion) with a flux limiter (Xue, 2000). This kind of filtering is commonly applied to unstable schemes like centred differences. It is also used to avoid energy accumulation at the small scales due to nonlinear interaction.

Fig. 5.2 shows the horizontal group velocity for both the filtered and unfiltered runs and the theoretical solution c_{xn} of Eq. (5.4) in Section 5.3. All values are normalised

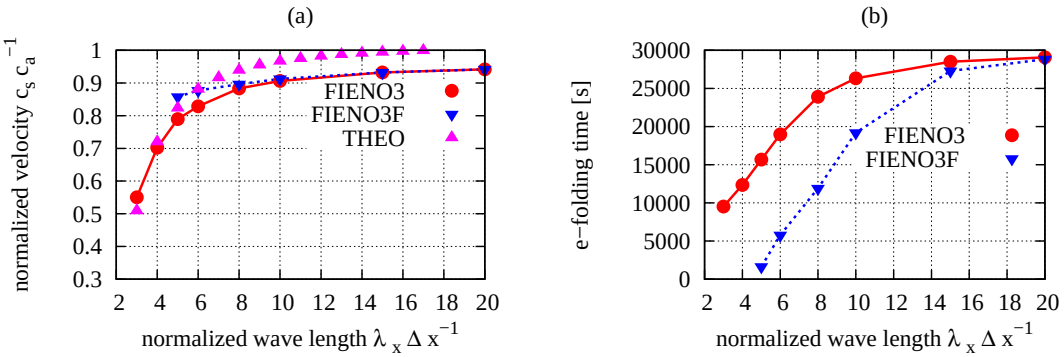

Figure 5.2: Results of gravity wave experiments with zero background wind. (a) Simulated gravity wave group velocity c_s normalised by the analytical group velocity c_a (Eq. 5.2) for different horizontal grid configurations with FIENO3 with and without filter compared to the theoretical solution of Eq. (5.4). (b) Equivalent to left but for simulated e-folding time of the wave intensity.

with the analytic solution for the group velocity (Eq. 5.2). The e-folding time is shown for the two types of model runs (Fig. 5.2b).

The coarser the resolution the lower is the group velocity (Fig. 5.2a). This behaviour is similar for both, model simulations and theoretical solution. Waves with $\lambda_x \leq 4\Delta x$ cannot be resolved with the filtered model, whereas the unfiltered model can resolve waves with $\lambda_x \geq 3\Delta x$. For $\lambda_x = 3\Delta x$ the group velocity is only about 55% of the analytical solution c_a. The theoretical solution c_{xn} predicts 50% for this resolution. For higher resolutions, the theoretical and simulated group velocities are similar with the theoretical group velocity being somewhat higher. The differences can be explained with the different discretisation. The analytical solution is derived by discretising the wave equation (5.1) for w' derived from the linearised equations which is in contrast to the METRAS discretisation on an Arakawa-C-grid.

As expected, the damping of FIENO3 and FIENO3F is quite different. The effect of the seven-point filter is clearly visible for the filtered version with the strong decrease of the e-folding time for decreasing resolution (Fig. 5.2b). $20\Delta x$ waves remain nearly unchanged by the filter. However, when smaller time steps are used, the filter will be applied more often and will therefore add more diffusion to the solution.

With the amplitudes chosen for this test the simulated wave is nearly linear. Consequently, the applied advection scheme is almost negligible (except for vertical advection of the background potential temperature).

5.4.2 Background wind $\overline{u} = \pm 10 ms^{-1}$

The effect of advection schemes on gravity wave dispersion can be measured when using a background wind. Fig. 5.3 and Fig. 5.4 show the gravity wave group velocity and damping for first order upstream advection (UP1), second (FIENO2) and third (FIENO3) order flux integrated ENO schemes for a background wind of $\pm 10 \mathrm{ms}^{-1}$ and different resolutions. The Shapiro seven-point filter is not used. The simulated wave is the same as presented before. The time step for each grid configuration is chosen to get a CFL number of 0.1 with respect to the background wind.

It is clearly visible that the damping is dominated by the order of the advection scheme (Fig. 5.3b and Fig. 5.4b). Furthermore, when the intrinsic group velocity has the same direction as the background wind (here $+10 \mathrm{ms}^{-1}$, Fig. 5.3b) the damping is less than when the intrinsic group velocity is opposed to the background wind (Fig. 5.4b). Shorter waves are damped more than longer waves as seen before. With a background wind this effect is intensified due to the diffusion of the advection schemes.

The group velocity should be approximately $13 \mathrm{ms}^{-1}$ and $-7 \mathrm{ms}^{-1}$, respectively. As observed for zero wind background conditions coarser resolutions give smaller (absolute) group velocities. This effect is intensified by the advection schemes used that also tend to be dispersive (Section 4.3, Schroeder and Schlünzen, 2007a). The lowest order scheme, UP1, has the strongest negative effect on the group velocity. However, for waves resolved with 6 or more grid cells the second order FIENO2

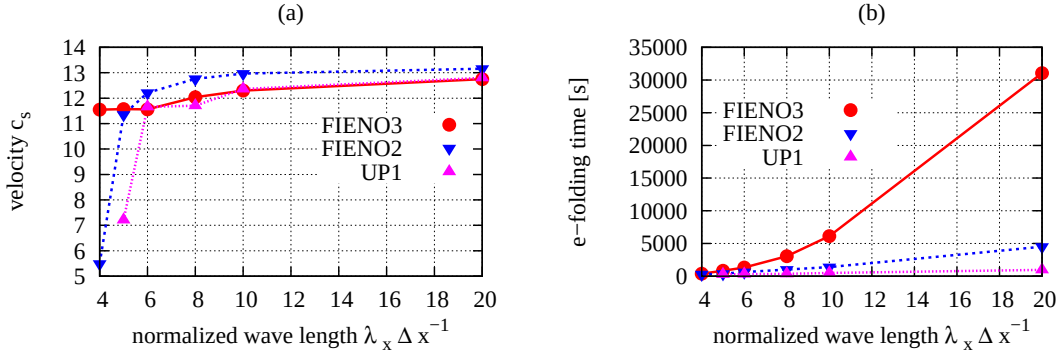

Figure 5.3: Results of gravity wave experiments with $10 \mathrm{ms}^{-1}$ background wind. (a) Simulated gravity wave group velocity c_s for different horizontal grid configurations and numerical advection schemes. (b) Equivalent to (a) but for the simulated e-folding time of the wave intensity.

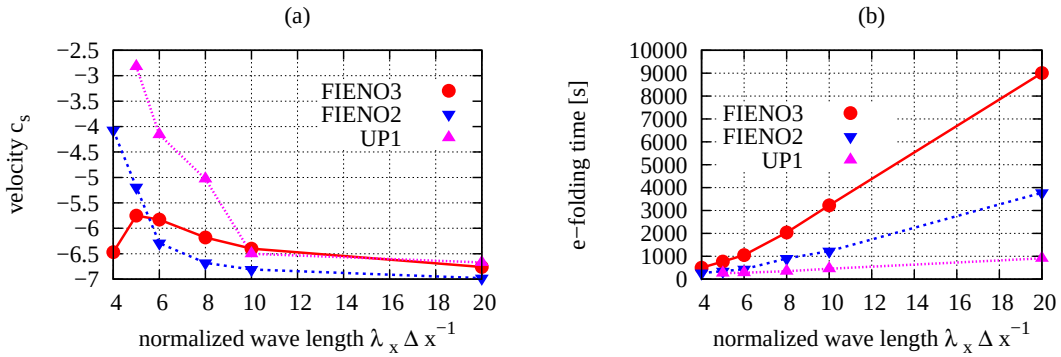

Figure 5.4: As Fig. 5.3 but for -10ms^{-1} background wind. Note the different scalings.

performs better than FIENO3 concerning the group velocity. Consequently, it can be concluded that FIENO2 has the best overall performance.

5.5 Reflection of gravity waves

The same wave as in Section 5.4 is simulated for horizontally nonuniform grid configurations. For comparison, Fig. 5.5a and Fig. 5.5b show the results on uniform grids with $\Delta x = \lambda_x/20$ and $\Delta x = \lambda_x/5$ horizontal resolution, respectively. As seen before, the wave damping is higher and the group velocity smaller on the coarse grid (Fig. 5.5a). If the grid spacing changes, e.g. with a factor of four (Fig. 5.5c,d,f), reflections take place. The reflection is total, if the wave cannot be resolved on the coarse grid (5.5c). Even if the wave can be resolved on both, the fine and the coarse grid, reflections occur, but they have a smaller amplitude (Fig. 5.5d). This is due to the dependence of group velocities on grid spacing as seen in Sections 5.3 and 5.4. When the wave packet moves from the fine to the coarse grid the energy cannot enter the coarse grid fast enough and some energy is reflected. For the same grid configuration ($\Delta x = \lambda_x/20$ for $x < 0$, $\Delta x = \lambda_x/5$ for $x > 0$) the reflection is slightly stronger for the filtered than for the unfiltered simulation (Fig. 5.5d,f). The reason is that the additional damping due to the filter is much higher in the coarse grid region than in the fine grid region. Thereby the property of the gravity wave is changed differently in both regions compared to the unfiltered case.

The intensity of the reflected waves can be measured and related to the intensity of the unreflected wave of the simulation on a uniform grid. For this purpose the results of several simulations are evaluated. They have been performed with a grid

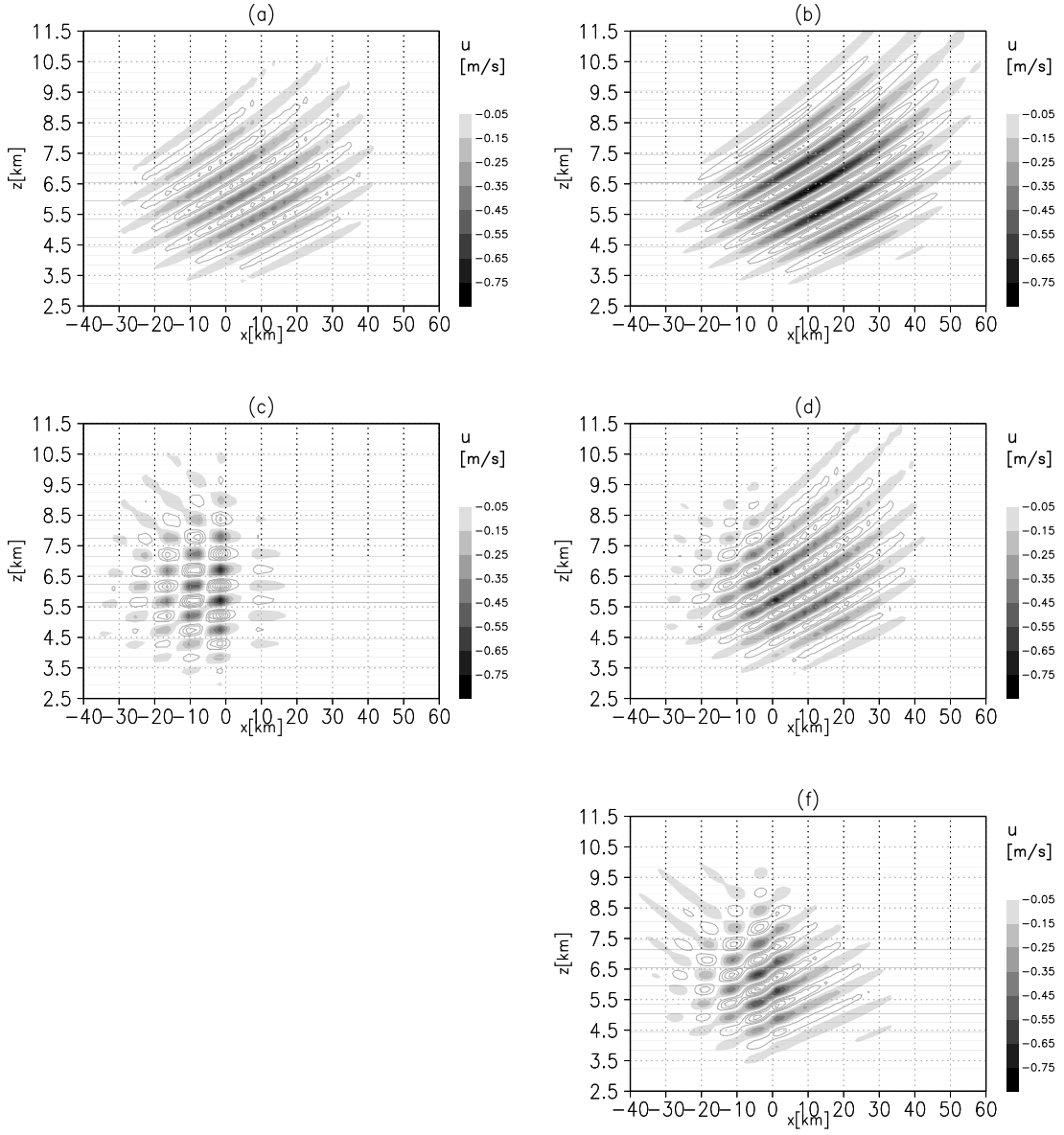

Figure 5.5: Gravity wave simulation after 3h for different horizontal grid configurations. Shown is u with scalings as in Fig. 5.1. The horizontal grid spacing is (a) $\Delta x = \lambda_x/5$, (b) $\Delta x = \lambda_x/20$, (c) $\Delta x = \lambda_x/5$ for $x < 0$ and $\Delta x = \lambda_x/1.25$ for $x > 0$, (d,f) $\Delta x = \lambda_x/20$ for $x < 0$ and $\Delta x = \lambda_x/5$ for $x > 0$. (a-d) simulation without filter , (f) simulation with filter.

spacing of $\Delta x = \lambda_x/r$ for $x < 0$. The resolution indicator r is an integer that describes how many grid cells resolve one wave length. For $x > 0$ the grid spacing is $\Delta x = \lambda_x c/r$, where the coarsening factor c is an integer. In this region r/c cells resolve one wave length. The magnitude of the reflected wave is estimated as follows: For an experiment with a fine resolution indicator r and a coarsening factor $c > 1$,

the maximum averaged wave energy density I' (Eq. 5.6) is determined in the fine region ($x < 0$). This is related to the maximum averaged wave energy density in the whole domain of the corresponding run with the resolution indicator r on a uniform grid ($c = 1$) at the same time step. The result is the relative intensity RI

$$\mathrm{RI} = \frac{\max_{x<0,c=c'}(I')}{\max_{c=1}(I')}. \tag{5.7}$$

The denominator always corresponds to an unreflected wave. Strictly speaking, RI does not indicate the relative intensity of a reflected wave, since it the maximum averaged wave energy density on the coarse grid and can still be related to parts of the unreflected wave. However, after a sufficient time the unreflected wave should have left the coarse grid, and RI should reach a saturation value that gives a hint on the magnitude of the reflection. If no reflection takes place, RI should approach zero.

Fig. 5.6 shows the relative intensities for $r = 5, 8, 10, 20$ and coarsening factors of $c = 2, 3, ..., 7$. There are two types of curves. This will be described for the example of $r = 5$. For a reflected wave (e.g. Fig. 5.6a with $c > 2$) RI is 1 as long as the wave has not reached the internal boundary with the grid discontinuity ($t < 1.5$h). When it reaches this point, reflection takes place. The already reflected part and the unreflected part are superimposed which can lead to values larger than 1 (overshoot, $t =$2-3h). The reflected wave is damped, and with increasing time the reflected and unreflected ways diverge in different directions so that they cannot overlap anymore. Consequently, RI decreases. When the total reflection is finished and parts of the wave that may not have been reflected have left the coarsely resolved region ($x < 0$), the RI curve has reached its saturation value (after 5-7h simulation time, depending on the configuration).

The second type of curve corresponds to waves that are not reflected. This will be described for $r = 20$ and $c = 2$ (Fig. 5.6d). Here the overshoot does not happen, as there is no reflected wave to overlap the unreflected wave. The RI value is decreasing with time until it has reached zero when no parts of the wave are left in the fine grid ($x < 0$). It is interesting to note that e.g. for $c = 2$ the decrease of the RI value starts the later the larger r. The reason for this is probably the damping caused by the internal boundary. The damping is weaker when the wave is better resolved on the coarse grid (larger r/c). This means for larger c, RI decreases not only due to the wave leaving the fine grid but also due to damping.

After 8h of simulation time in all simulations RI has reached saturation and the relative intensities can be compared and treated as the intensity of the reflected

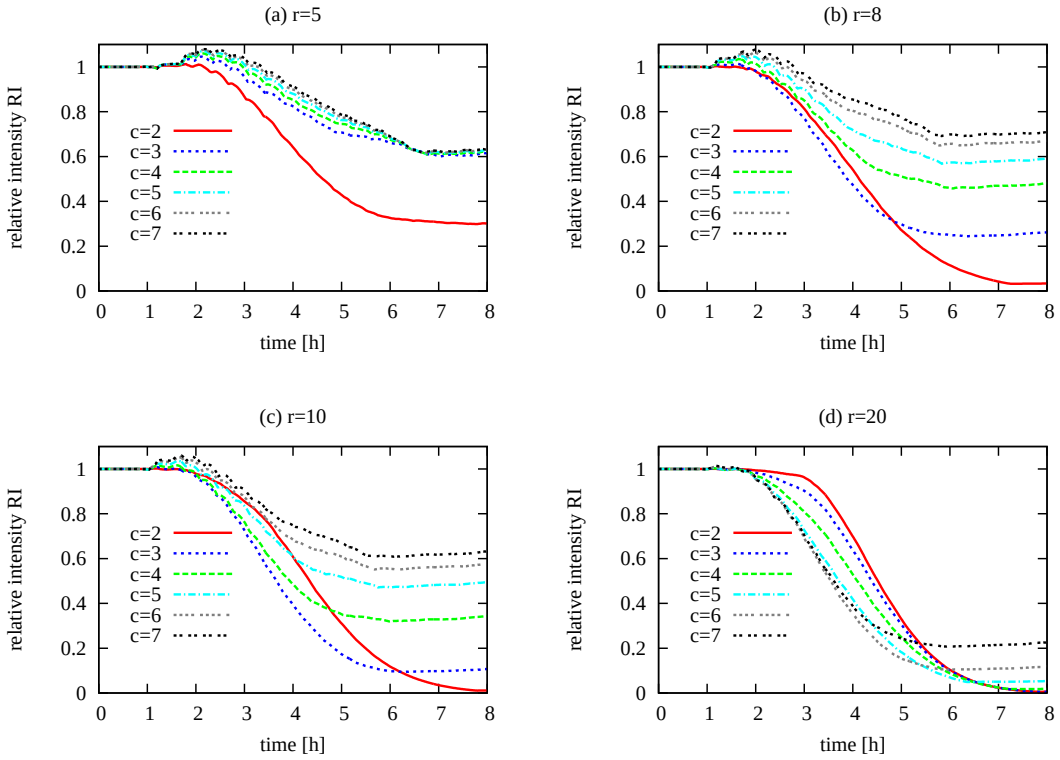

Figure 5.6: The relative intensity RI according to Eq. (5.7). The results are given for different resolutions. The fine grid resolution ($x < 0$) is (a) $r = 5$: $\Delta x = \lambda_x/5$, (b) $r = 8$: $\Delta x = \lambda_x/8$, (c) $r = 10$: $\Delta x = \lambda_x/10$ and (d) $r = 20$: $\Delta x = \lambda_x/20$ in the horizontal. For $x > 0$ the resolution is $\Delta x = \lambda_x r/c$, i.e. c is the coarsening factor.

wave. As expected the intensity of the reflected wave is lower for a smaller coarsening factor c. When $r = 20$ only coarsening factors of 5, 6 and 7 show a reflected wave of about 5%, 12% and 23%, respectively. For $c < 5$ there is almost no reflection.

For $r = 5$ the result is completely different. For $c > 2$ the reflected wave keeps about 63% of the unreflected wave intensity. This value does not vary much for different factors of c, as in neither grid configuration the wave can be resolved on the coarse grid ($x > 0$). Only for $c = 2$ the reflected wave has a lower intensity, but it still has about 30% of the unreflected wave intensity. This change in grid width seems to absorb more of the reflected wave than with larger coarsening factors, which means that the coarse grid still reacts to some extent as would be demanded by the wave to remain undisturbed.

For $r = 8$ and 10 only with $c = 2$ reflection is nearly avoided. For $c = 3$ the reflected wave intensity is less than 26%, for $c = 4$ it is less than 50%. However, for the remaining coarsening factors the reflected wave intensity reaches up to 71% for

$c = 7$ and $r = 8$. Again, the reason for the different reflected wave intensities is that the coarse grid can react more quickly and in a more physical way if the coarsening factor is smaller and the resolution for $x > 0$ is higher.

From these results it can be concluded that coarsening factors of 3 are still acceptable. For longer waves (with respect to the grid) the reflection is hardly more than a fourth of the original intensity. Smaller waves, e.g. $r = 5$ have a strong damping even on the fine grid (Section 5.4) which means they can be neglected. For coarsening factors larger than 3 the reflection can be significant even for well resolved waves, i.e. waves for which the damping or e-folding time is large on the fine grid (see Fig. 5.2 for the e-folding times). This is typically the $8\Delta x$ wave (r=8), if a Shapiro seven-point filter is applied (e.g. Chapter 3, Schroeder et al., 2006). To avoid reflected waves distorting the solution, either coarsening factors larger than 3 should not be used, or measures need to be taken to avoid reflection.

5.6 Conclusions and outlook

The dispersion of gravity waves and their reflection, when a sudden coarsening of a grid is given, were investigated in this paper. The influence of a background wind and the applied advection scheme were tested for a first order upstream scheme and a second and third order flux integrated essentially non-oscillatory scheme (Schroeder and Schlünzen, 2007a) for a CFL number of 0.1. Not surprisingly, the third order scheme FIENO3 is much less damping than the first order upstream scheme and less damping than the second order scheme FIENO2. However, the second order scheme simulated the group velocity for waves longer than $3\Delta x$ more accurately. The numerical damping is stronger when the background wind is opposed to the horizontal group velocity in comparison to the case where the background wind is blowing in the same direction.

It was shown that even for grid coarsening/refinement factors of 4 a significant reflection of gravity waves can take place. It amounts to almost 50% for waves that are well resolved on the fine grid (e.g. by 8 grid cells) but have too few grid points on the coarse grid. A reason for the reflection of waves that can be resolved on the coarse grid are the different group velocities on grids with different resolutions. A numerical dispersion relation was derived theoretically that confirms that the coarser the grid the more the gravity wave is retarded. Results from numerical experiments of gravity waves on grids with different resolution agree well with the theoretical

numerical dispersion relation.

Consequently, when two-way nesting or grid refinement is applied, the refinement factor should not be larger than 3 to keep the reflected gravity wave energy small. The results from this paper can be transferred to other spatial and time scales as long as the gravity wave intensity is weak enough to be accurately described by the linearised equations. Simulations in which the relation of horizontal grid width to wave length are the same as in this work should yield similar group velocities (normalised by the analytical solution) as presented here.

6 Solving the pressure equation on a set of grids

This chapter describes how the perturbation pressure $\hat{p}$ is calculated from Eq. (2.19) and discusses the complications arising from a solution on multiple grids. Prior to this work three pressure solvers were available for METRAS: the idealised generalised conjugate gradient method (IGCG, Kapitza and Eppel, 1987), Bi-CGSTAB (Van der Vorst, 1992) and MULTIGRID (e.g. Hackbusch, 1985). The latter was originally only available for simulations without topography. Preliminary tests showed that IGCG is not suitable when the number of grid points is large or the asymmetry of the pressure matrix is increased. Therefore, this method was discarded for the multiple grids version since the introduction of multiple grids leads to an additional asymmetry of the pressure matrix. However, MULTIGRID and Bi-CGSTAB also need adjustment for the multiple grids version to make them converge properly. The basic schemes are described in Appendix B. They are tested in this chapter to investigate which solver is most efficient for solving the pressure equation on multiple grids.

6.1 Derivation of the pressure equation

The anelastic approximation leads to the requirement of a divergence free momentum field (Eq. (2.14)). However, when the preliminary momentum vector field $\vec{\tilde{m}}^{n+0.5}$ has been calculated according to Eq. (2.16)-(2.18), this vector field is generally not divergence free, since the dynamic pressure p_2^{n-1} of the previous time step was used for the preliminary step. A pressure perturbation $\hat{p}$ needs to be calculated so that the updated momentum vector $\vec{m}^{n+0.5}$ can be calculated with Eq. (2.20). The requirement of the divergence free momentum $\vec{\nabla} \cdot \vec{m}^{n+0.5} = 0$ leads to Eq. (2.19).

When the pressure perturbation equation that ensures non-divergent momentum fields is discretised in space, this leads to a system of equations (one equation for each grid point), that can be written as

$$Ax = b. \tag{6.1}$$

The matrix A is the composition of divergence operator and pressure operator, $\vec{\nabla} \bullet \vec{\mathcal{P}}$. The solution vector x (not to be confused with the x-coordinate) corresponds to $\hat{p}$. b on the right-hand-side of Eq. (6.1) is the preliminary momentum divergence normalised by the time step, $\vec{\nabla} \bullet \tilde{\vec{m}}^{n+0.5}/(\Delta t)$. Equations of this type can be solved iteratively with Krylov subspace or MULTIGRID methods (Appendix B).

It is very important that not the analytical composition of divergence and pressure gradient operator in Eq. (2.19) are discretised to calculate $\hat{p}$, but the operators are discretised separately thereby considering the discretisations used in the momentum equations. This is the only way to make sure that the resulting discretised momentum field is divergence free according to the discretised divergence operator.

A divergence free momentum field is not only important to solve the momentum fields correctly, but also influences the conservation of scalar quantities. The discretised divergence operator indirectly appears in the advection of scalar quantities, since $\mathcal{A}(\rho_0 \alpha^* \chi) = \nabla \bullet (\vec{m}\chi)$ for scalar quantities. To clarify the impact one could assume that χ is constant with respect to space. In this case advection must not change this property. This is only guaranteed if the discretised divergence of momentum is zero.

The remainder of this chapter is structured as follows: Section 6.2 describes how the matrix A is set up for a single grid in the original METRAS version. In addition, the calculation of the new matrix elements are derived for multiple grids. In Section 6.3 the stopping criterion for the iterative solvers of Appendix B is described. In Section 6.4 the performance of MULTIGRID and Bi-CGSTAB is compared for different grid configurations, i.e. single grid versus multiple grids and different aspect ratios of horizontal and vertical grid width.

6.2 Setting up the matrix

In this section the pressure gradient and divergence operators are given in discretised form for a single grid and are derived for coupled multiple grids. This is a prerequisite

for deriving the coefficients of matrix A belonging to the discretisation of

$$\vec{\nabla} \bullet \vec{\mathcal{P}} \tag{6.2}$$

The actual composition of both discretised operators leads to the coefficients and is presented in Appendix A.

For defining the operators the following grid transformation coefficients are used

$$A = \frac{\partial x}{\partial \dot{x}} \qquad \text{scalar grid width in } x\text{-direction}$$

$$B = \frac{\partial y}{\partial \dot{y}} \qquad \text{scalar grid width in } y\text{-direction}$$

$$G = \frac{z_t - z_s}{z_t} \qquad \text{vertical compression of the grid}$$

$$\Delta z G = \frac{\partial z}{\partial \dot{z}} \qquad \text{vertical grid width for vectors}$$

$$CG \qquad \text{vertical grid width for scalars}$$

$$D = \frac{\partial z_s}{\partial x} \qquad \text{surface slope in } x\text{-direction}$$

$$E = \frac{\partial z_s}{\partial y} \qquad \text{surface slope in } y\text{-direction}$$

$$F = \left.\frac{\partial z}{\partial z_s}\right|_{\dot{z}=\text{const}} = 1 - \frac{z - z_s}{z_t - z_s} \qquad z\text{-dependent intensity of grid deformation}$$

Note that here and in the following A should not be confused with the matrix A in Eq. (6.1). While A and D are defined at u grid points, B and E at v grid points, C and F at w grid points, and G is defined at scalar grid points. Further coefficients of the coordinate transformation can be expressed by these factors:

$$\frac{\partial \dot{z}}{\partial x} = \frac{\partial \dot{z}}{\partial z} \left.\frac{\partial z}{\partial z_s}\right|_{\dot{z}=\text{const}} \frac{\partial z_s}{\partial x} = \frac{FD}{CG}$$

$$\frac{\partial \dot{z}}{\partial y} = \frac{\partial \dot{z}}{\partial z} \left.\frac{\partial z}{\partial z_s}\right|_{\dot{z}=\text{const}} \frac{\partial z_s}{\partial y} = \frac{FE}{CG}$$

Details on the coordinate transformation and their use in the transformed equations can be found in Schlünzen et al. (1996).

6.2.1 Using a single grid

The momentum divergence in transformed coordinates is used in the continuity equation (2.14). On a single grid the discretised divergence of the preliminary momentum can be written written as

$$\begin{aligned}\left[\vec{\nabla}\cdot(\rho_0\alpha^*\vec{\hat{v}})\right]_{ijk} &= \Delta y_j\Delta z_k\left[\left(\overline{\rho_0}^{\dot{x}}\overline{G}^{\bullet\dot{x}}\hat{u}\right)_{i+0.5jk}-\left(\overline{\rho_0}^{\dot{x}}\overline{G}^{\bullet\dot{x}}\hat{u}\right)_{i-0.5jk}\right] \quad (a)\\ &+ \Delta x_i\Delta z_k\left[\left(\overline{\rho_0}^{\dot{y}}\overline{G}^{\bullet\dot{y}}\hat{v}\right)_{ij+0.5k}-\left(\overline{\rho_0}^{\dot{y}}\overline{G}^{\bullet\dot{y}}\hat{v}\right)_{ij-0.5k}\right] \quad (b) \quad (6.3)\\ &+ \Delta x_i\Delta y_j\left[\left(\overline{\rho_0}^{\dot{z}}CG\hat{\dot{w}}\right)_{ijk+0.5}-\left(\overline{\rho_0}^{\dot{z}}CG\hat{\dot{w}}\right)_{ijk-0.5}\right] \quad (c).\end{aligned}$$

While in the above equation the non-transformed preliminary horizontal velocities ($\hat{u}$ and $\hat{v}$) appear, the transformed preliminary vertical velocity $\hat{\dot{w}}$ is used to simplify setting the Neumann boundary conditions for the Poisson equation.

The pressure gradient in transformed coordinates for the momentum components ($\overline{u}\rho_0, \overline{v}\rho_0, \overline{\dot{w}}\rho_0$) is given in the list of operators (2.4) and by Eq. (2.5). When the pressure gradient operator is discretised this leads to

$$\mathcal{P}_{i+0.5jk} = \left(\frac{1}{A}\delta_{\dot{x}}\hat{p}-\frac{\overline{F}^{\dot{z}}D}{\overline{C}^{\dot{z}}\overline{G}^{\bullet\dot{x}}}\delta_{\dot{z}}\overline{\hat{p}}^{\dot{x},\dot{z}}\right)_{i+0.5jk} \quad (6.4)$$

$$\mathcal{P}_{ij+0.5k} = \left(\frac{1}{B}\delta_{\dot{y}}\hat{p}-\frac{\overline{F}^{\dot{z}}E}{\overline{C}^{\dot{z}}\overline{G}^{\bullet\dot{y}}}\delta_{\dot{z}}\overline{\hat{p}}^{\dot{y},\dot{z}}\right)_{ij+0.5k} \quad (6.5)$$

$$\begin{aligned}\dot{\mathcal{P}}_{ijk+0.5} = \Bigg\{&\frac{1}{(CG)^2}\left[1+F^2\left(\overline{D}^{\dot{x}^2}+\overline{E}^{\dot{y}^2}\right)\right]\delta_{\dot{z}}\hat{p}\\ &-\frac{F}{CG}\left[\frac{\overline{D}^{\dot{x}}}{\overline{A}^{\dot{x}}}\delta_{\dot{x}}\overline{\hat{p}}^{\dot{x},\dot{z}}+\frac{\overline{E}^{\dot{y}}}{\overline{B}^{\dot{y}}}\delta_{\dot{y}}\overline{\hat{p}}^{\dot{y},\dot{z}}\right]\Bigg\}_{ijk+0.5}.\end{aligned} \quad (6.6)$$

Here δ is a simple difference operator in one spatial direction, e.g. $(\delta_{\dot{x}}\hat{p})_{i+0.5jk} = \hat{p}_{i+1\,jk}-\hat{p}_{ijk}$. Again, for the horizontal components the pressure gradient is calculated in the Cartesian x- and y-directions (for momentum associated with the real velocities u and v), whereas in the vertical it is calculated for the transformed vertical momentum $\dot{w}$. In this way the Neumann boundary conditions at the surface can be set by requiring $\dot{\mathcal{P}}_{ijk+0.5}=0$ at the surface. Note that the equations (6.4)-(6.6) describe the pressure gradient acting on $\rho_0\overline{u}$, $\rho_0\overline{v}$ and $\rho_0\overline{\dot{w}}$, therefore α^* does not appear in them.

With the definitions above the pressure matrix can be set up by replacing momentum in the discrete divergence equation (6.3) by the discrete pressure gradients (6.4)-(6.6)

(i.e. $(\overline{\rho_0}^{\dot{x}}\hat{u})_{ijk}$ by $\mathcal{P}_{ijk}$, $\overline{\rho_0}^{\dot{y}}\hat{v}$ by $\mathcal{P}_{ij+0.5k}$ and $\overline{\rho_0}^{\dot{z}}\hat{w}$ by $\dot{\mathcal{P}}_{ijk+0.5}$, see Appendix A). Figure 6.1 shows the structure of the matrix for an exemplary grid. Here the grid points are ordered lexicographically (Appendix B.3).

It should be noted that Neumann boundary conditions are used with zero gradient. To obtain a unique solution it is additionally demanded that the average pressure perturbation $\hat{p}$ is zero at the model top.

6.2.2 Using multiple grids

When the domain is decomposed into two or more grids of different resolution, the internal boundaries must be taken care of. There, the momentum field must be divergence free as in the rest of the model area. It it very important to define the discrete pressure gradient operator and divergence operator properly. Otherwise numerical noise is likely to be introduced, and the iteration to solve the perturbation pressure may converge only slowly. The pressure gradient operator also needs to conserve momentum.

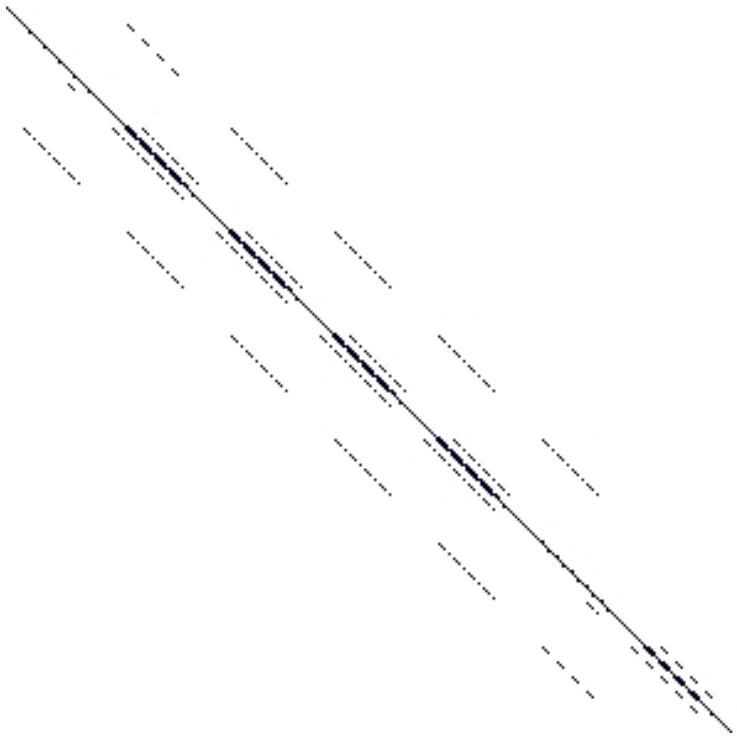

Figure 6.1: Schematic of a matrix for a single grid (only the upper part is shown). Each point represents a non-zero matrix element.

Actually, there are four different cases of boundaries: fine grid to the west, fine grid to the east, fine grid to the north and fine grid to the south. Here, only the case with the fine grid to the east is described (Fig. 6.2), as the different cases are analogous.

In the following capital letters are used for coarse grid horizontal indices, lower case letters for the fine grid and vertical indices. The scalar boundary of the coarse grid is located at $i = r$, while the scalar boundary of the coarse grid is located at $I = R$ (Fig. 6.2). Additionally, the following abbreviations are used (Fig. 6.3):

N_x		refinement factor between two grids in $\dot{x}$-direction
N_y		refinement factor between two grids in $\dot{y}$-direction
ψ_c	$= \frac{4}{3+N_x}$	weighting factor for mass flux from cell $r - 0.5$
ψ_f	$= \frac{N_x-1}{3+N_x}$	weighting factor for mass flux from cell $r + 0.5$
ψ_u	$= \frac{2}{N_x+1}$	reduction of x-gradient through boundary cell $u_r - 0.5jk$
ψ_w	$= \frac{1}{N_x+3}$	reduction of x-gradient of cell $w_{rjk+0.5}$
ψ_W	$= \frac{N_x}{3N_x+1}$	reduction of x-gradient of cell $W_{RJk+0.5}$

The divergence on the coarse grid caused by mass fluxes in x-direction is then given

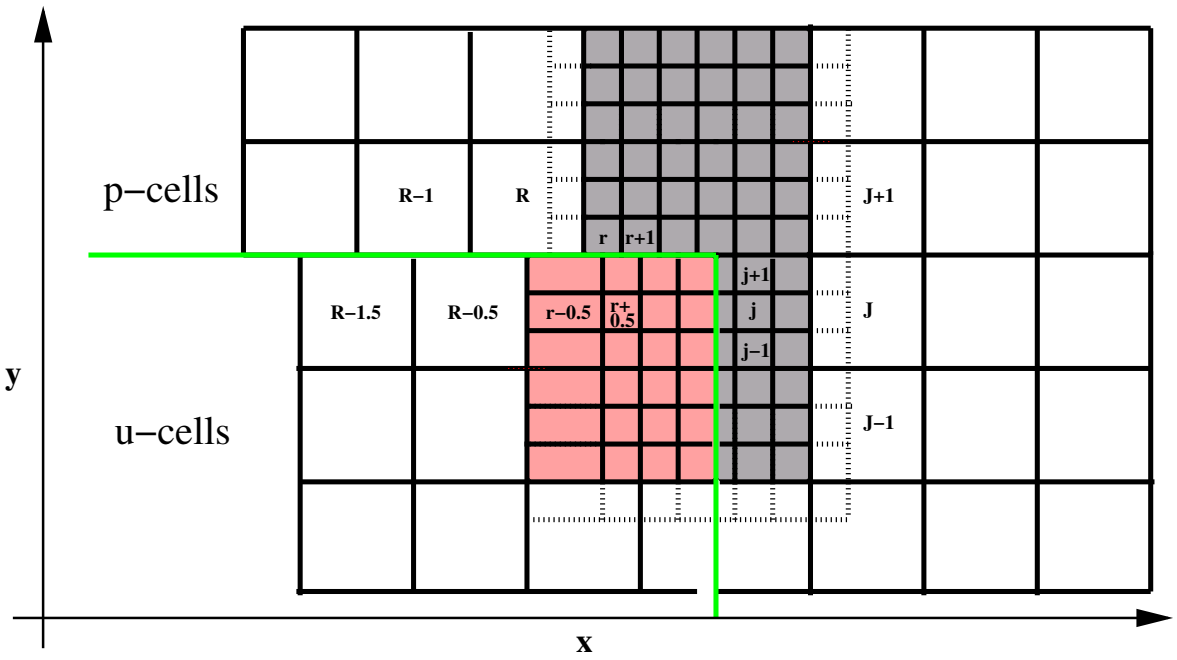

Figure 6.2: Horizontal schematic of scalar (p) and u grid with grid refinement. J and j are indices in y-direction, R and r denote indices in x-direction. R and r are the grid points at the boundary of coarse and fine grid.

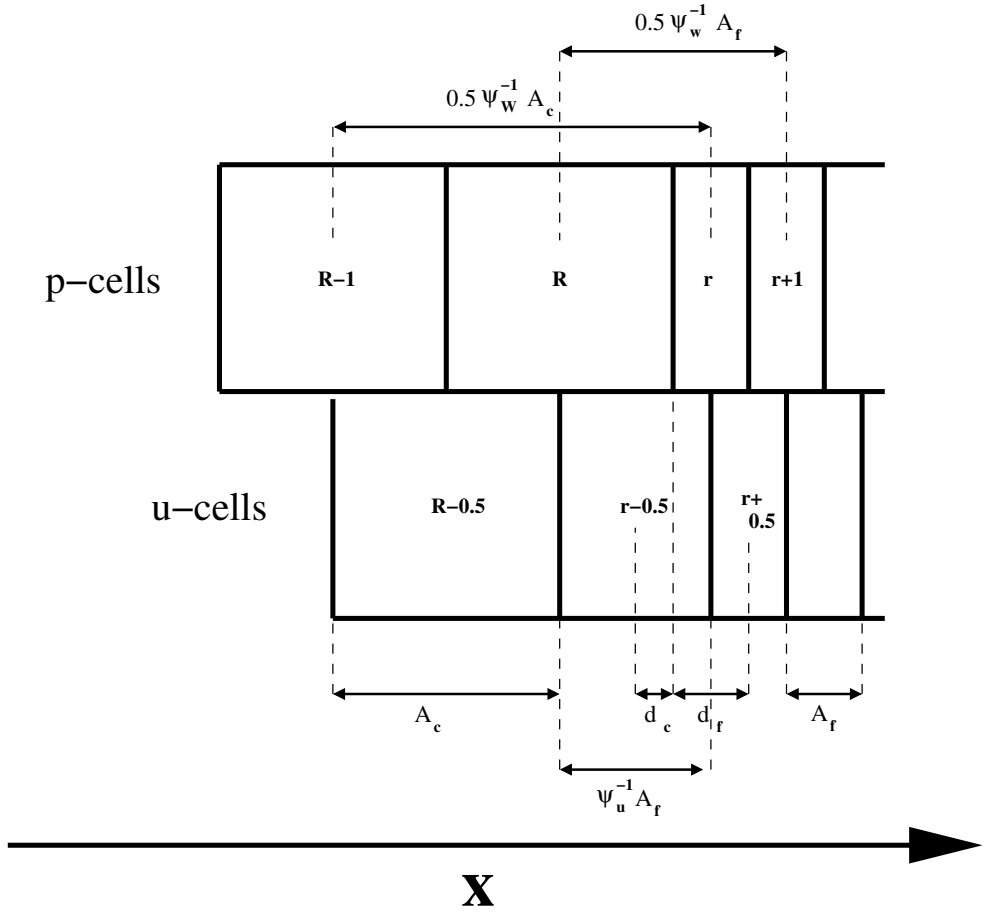

Figure 6.3: Schematic of the grid with refinement in x-direction. The meaning of the factor ψ_u, ψ_w and ψ_W used in the formulas for divergence and pressure gradient are demonstrated. The weighting factors ψ_c and ψ_f can be calculated from the distances d_c and d_f: $\psi_c = d_f/(d_f + d_c)$ and $\psi_f = d_c/(d_f + d_c)$.

by:

$$
\begin{aligned}
(\frac{\partial}{\partial x}\rho_0\alpha^*\hat{U})_{RJk} \;=\; & \Delta y_J \Delta z_k \\
& \Big\{\Big[\psi_c \frac{1}{N_y}\sum_{j=j_1}^{j_2}\Big(\overline{\rho_0}^{\dot{x}}\overline{G}^{\bullet\dot{x}}\hat{u}\Big)_{r-0.5\,jk} \\
& +\psi_f \frac{1}{N_y}\sum_{j=j_1}^{j_2}\Big(\overline{\rho_0}^{\dot{x}}\overline{G}^{\bullet\dot{x}}\hat{u}\Big)_{r+0.5\,jk}\Big] \\
& -\Big(\overline{\rho_0}^{\dot{x}}\overline{G}^{\bullet\dot{x}}\hat{U}\Big)_{R-0.5\,Jk}\Big\}
\end{aligned}
\tag{6.7}
$$

The eastern mass flux at the internal boundary ($I = R + 0.5$) is changed in comparison to the corresponding term in Eq. (6.3a). The total mass flux from the N_y fine grid cells covering the coarse grid cell R is taken. Because the most western

u-cell $(r-0.5)$ on the fine grid has a longer extension $(A_{r-0.5}/\psi_u)$ than the inner u-cells (A_i), the flux taken is a weighted average of cell $r-0.5$ (factor ψ_c) and cell $r+0.5$ (factor ψ_f). This is equivalent to a linear interpolation of the flux to $\dot{X}_{R+0.5}$. Furthermore, due to the possible refinement in y-direction the fluxes from N_y cells must be averaged in y-direction. At non-corner boundary points the remaining divergence in y- and $\dot{z}$-direction is unchanged in comparison to Eq. (6.3). At corner points the divergence in y-direction is changed in the same way as in x-direction.

For the fine grid the divergence in cell r in x-direction is almost unchanged compared to term (a) of Eq. (6.3). Since the mass flux at $I = R + 0.5$ is needed, again the weighted average of the flux at $i = r+0.5$ and $i = r-0.5$ is needed. However, since $1-\psi_f = \psi_c$, this can also be interpreted as the divergence acting over the longer distance of the extended u boundary cell at $i = r-0.5$. Therefore, this leads to a correction by factor ψ_c in comparison to term (a) of Eq. (6.3):

$$\begin{aligned}\left(\frac{\partial}{\partial x}\rho_0\alpha^*\hat{u}\right)_{rjk} = \Delta y_j\Delta z_k\Big\{ & \left(\overline{\rho_0}^{\dot{x}}\overline{G}^{\bullet\dot{x}}\hat{u}\right)_{r+0.5\,jk} \\ & - \psi_f\left(\overline{\rho_0}^{\dot{x}}\overline{G}^{\bullet\dot{x}}\hat{u}\right)_{r+0.5\,jk} \\ & - \psi_c\left(\overline{\rho_0}^{\dot{x}}\overline{G}^{\bullet\dot{x}}\hat{u}\right)_{r-0.5\,jk}\Big\} \\ = \Delta y_j\Delta z_k\psi_c\Big\{ & \left(\overline{\rho_0}^{\dot{x}}\overline{G}^{\bullet\dot{x}}\hat{u}\right)_{r+0.5\,jk} \\ & - \left(\overline{\rho_0}^{\dot{x}}\overline{G}^{\bullet\dot{x}}\hat{u}\right)_{r-0.5\,jk}\Big\}\end{aligned} \tag{6.8}$$

The pressure gradient force does not need to be changed for the coarse grid cell $R-0.5$ in comparison to Eq. (6.4). For the fine grid the pressure gradient force in x-direction is changed in two ways for cell $\hat{u}_{r-0.5}$:

$$\begin{aligned}\mathcal{P}_{r-0.5jk} = \Bigg\{ & \psi_u\frac{1}{A_{r-0.5}}\left(p_{rjk} - \overline{\overline{P_{Rjk}}^{\dot{y}}}\right) \\ - & \frac{1}{2}\left(\frac{\overline{F}^{\dot{z}}D}{\overline{C}^{\dot{z}}\overline{G}^{\bullet\dot{x}}}\right)_{r-0.5\,jk} \\ & \left[\delta_{\dot{z}}\overline{\left(p_{rj} + \overline{\overline{P_{Rj}}^{\dot{y}}}\right)}^{\dot{z}}\right]_k\Bigg\}\end{aligned} \tag{6.9}$$

In the first term the contribution of the gradient in x-direction is corrected by the factor ψ_u due to the special extension of the boundary normal u cell at $r = i-0.5$. Furthermore, for the western pressure not simply the neighbouring coarse grid value

P_{RJk} is taken, but a value parabolically interpolated in y-direction from P_{RJ-1k}, P_{RJk} and P_{RJ+1k}:

$$\overline{\overline{P_{Rjk}}}^{\dot{y}} = a_{1j}P_{RJ-1k} + a_{2j}P_{RJk} + a_{3j}P_{RJ+1k} \tag{6.10}$$

$$a_{1j} = \frac{1}{2}(\delta_j^2 - \delta_j^2) \tag{6.11}$$

$$a_{2j} = 1 - \delta_j^2 \tag{6.12}$$

$$a_{3j} = \frac{1}{2}(\delta_j^2 + \delta_j^2) \tag{6.13}$$

$$\delta_j = (\dot{y}_j - \dot{Y}_J)/(\dot{Y}_{J+0.5} - \dot{Y}_{J-0.5}) \tag{6.14}$$

with the coarse grid levels $\dot{Y}_J$ and the fine grid levels $\dot{y}_j$ in y-direction (denoted by their indices J and j, respectively, in Fig. 6.2). The parabolic interpolation is necessary, because for solving the dynamic pressure p_2 the Laplacian of pressure is calculated which is a second order operator. Consequently, the coarse pressure should also be reconstructed to the fine grid by a second or higher order interpolation to avoid artefacts. The interpolated values $\overline{\overline{P_{Rjk}}}^{\dot{y}}$ are also taken for the contribution of the $\dot{z}$-direction, for which horizontal averaging is necessary.

The pressure gradients in $\dot{z}$-direction for the transformed vertical velocity need to be corrected for the coarse grid ($\hat{\dot{W}}_{RJk+0.5}$) and the fine grid ($\hat{\dot{w}}_{rjk+0.5}$). In both cases only the horizontal contribution is affected by the factors ψ_W for the coarse grid and ψ_w for the fine grid to correct the distance between the two grid cells due to the change of the grid spacing. For the coarse grid the vertical pressure gradient is then:

$$\begin{aligned}
\dot{\mathcal{P}}_{RJk+0.5} \;=\; & \left\{ \frac{1}{(CG)^2}\left[1 + F^2\left(\overline{D}^{\dot{x}\,2} + \overline{E}^{\dot{y}\,2}\right)\right]\delta_{\dot{z}}p_2 \right. \\
& \left. -\frac{F}{CG}\left[\frac{\overline{E}^{\dot{y}}}{\overline{B}^{\dot{y}}}\delta_{\dot{y}}\overline{p_2}^{\dot{y},\dot{z}}\right]\right\}_{RJk+0.5} \\
& + \left(\frac{F}{CG}\frac{\overline{D}^{\dot{x}}}{\overline{A}^{\dot{x}}}\right)_{RJk+0.5} \psi_W \\
& \left[\frac{1}{N_y}\left(\sum_{j=j_1}^{j_2} p_{rjk} + \sum_{j=j_1}^{j_2} p_{rjk+1}\right) - P_{R-1\,Jk} - P_{R-1\,Jk+1}\right]
\end{aligned} \tag{6.15}$$

For the fine grid the vertical pressure gradient is:

$$\begin{aligned}\dot{\mathcal{P}}_{rjk+0.5} \;=\; & \left\{ \frac{1}{(CG)^2}\left[1+F^2\left(\overline{D}^{\dot{x}^2}+\overline{E}^{\dot{y}^2}\right)\right]\delta_{\dot{z}}p_2 \right. \\ & \left. -\frac{F}{CG}\left[\frac{\overline{E}^{\dot{y}}}{\overline{B}^{\dot{y}}}\delta_{\dot{y}}\overline{p_2}^{\dot{y},\dot{z}}\right]\right\}_{rjk+0.5} \\ & +\frac{1}{\overline{\rho_{0\,rjk+0.5}}^{\dot{z}}}\left(\frac{F}{CG}\frac{\overline{D}^{\dot{x}}}{\overline{A}^{\dot{x}}}\right)_{rjk+0.5}\psi_w \\ & \left[p_{r+1\,jk}+p_{r+1\,jk+1}-\overline{\overline{P_{Rjk}}^{\dot{y}}}^{\dot{y}}-\overline{\overline{P_{Rjk+1}}^{\dot{y}}}^{\dot{y}}\right]\end{aligned} \tag{6.16}$$

The composition of divergence and pressure operator leads to the coefficients of the system of equations for the divergence caused by pressure. The derivations are given in Appendix A.

Figure 6.4 shows an example matrix for a domain on two grids. The elements linking the coarse and the fine grid are clearly visible.

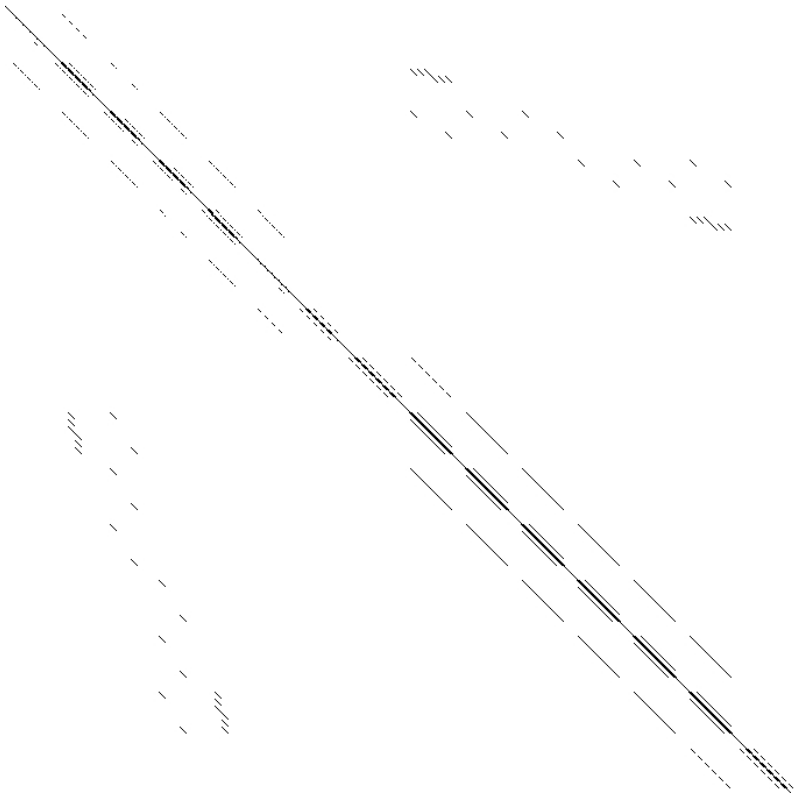

Figure 6.4: Schematic of a matrix for a main grid and a subgrid (only the upper part is shown). Each point represents a non-zero matrix element.

6.3 Stopping criteria

For the iterative solution of the dynamic pressure perturbation $\hat{p}$ a stopping criterion is necessary. With each iteration step the solution becomes more and more accurate. As soon as a sufficient accuracy has been reached, any further iteration would lead to an unnecessary waste of CPU time. The accuracy can be measured using the remaining divergence. The latter causes an error in the advection step due to the flux form. In this section a stopping criterion is derived by defining a maximum error that a remaining momentum divergence is allowed to cause in the solution of scalar quantities.

Without coordinate transformation the momentum divergence can be approximated by

$$\begin{aligned} D_{i,j,k} &= \frac{(\rho_0 u)_{i+0.5jk} - (\rho_0 u)_{i-0.5jk}}{\Delta x} \\ &+ \frac{(\rho_0 v)_{ij+0.5k} - (\rho_0 v)_{ij-0.5k}}{\Delta y} \\ &+ \frac{(\rho_0 w)_{ijk+0.5} - (\rho_0 w)_{ijk-0.5}}{\Delta z} \end{aligned} \tag{6.17}$$

and corresponds to the residual $Ax_k - b$ of Eq. (6.1) with x_k being the approximate solution after k iterations. With the error norm e a stopping criterion can be defined by demanding $e(Ax_k - b) < r_{\max}$, where $r_{\max}$ is the maximum allowed residual. $r_{\max}$ is derived in the following.

The flux form of the advection is derived by assuming a divergence free momentum vector. The advection can be approximated by

$$\begin{aligned} \mathcal{A}(\chi_{i,j,k}) &= \frac{(\rho_0 u\chi)_{i+0.5jk} - (\rho_0 uX)_{i-0.5jk}}{\Delta x} \\ &+ \frac{(\rho_0 v\chi)_{ij+0.5k} - (\rho_0 vX)_{ij-0.5k}}{\Delta y} \\ &+ \frac{(\rho_0 w\chi)_{ijk+0.5} - (\rho_0 wX)_{ijk-0.5}}{\Delta z} \end{aligned} \tag{6.18}$$

where the quantity χ is advected. Note that this is only an approximate advection operator which should not be confused with the advection schemes described in Chapters 3 and 4. For the derivations here only the approximate magnitude of the advection is important. To estimate the error in the tendency of quantity χ resulting from a divergent momentum field, χ is assumed to be constant (denoted by $\bar{\cdot}$). This

leads to an absolute error tendency of

$$\begin{aligned}\mathcal{E}_a(\mathcal{A}(\chi_{i,j,k})) &= \overline{\chi}\Big(\frac{(\rho_0 u)_{i+0.5} - (\rho_0 u)_{i-0.5}}{\Delta x} \\ &+ \frac{(\rho_0 v)_{j+0.5} - (\rho_0 v)_{j-0.5}}{\Delta y} \\ &+ \frac{(\rho_0 z)_{k+0.5} - (\rho_0 w)_{k-0.5}}{\Delta z}\Big) \\ &= \overline{\chi} D_{i,j,k}\end{aligned} \tag{6.19}$$

The error depends on the momentum divergence $D_{i,j,k}$. For a constant value of $D_{i,j,k}$ the error in χ grows with increasing time integration. After Δt_s seconds of simulation time the error relative to χ will have grown to approximately

$$\mathcal{E}_r(\mathcal{A}(X_{i,j,k})) = D_{i,j,k}\Delta t_s \tag{6.20}$$

if non-linear effects are neglected. This simple calculation shows that the specification of the accuracy is application dependent, since for micro-scale simulations the total integration time Δt_s is much smaller than for mesoscale simulations. A plausible request is $|D_{i,j,k}| < 0.01\rho_0/\Delta t_s = r_{\max}$. This means the error in quantity χ caused by a divergent momentum field is less than 1% during the simulation time. The error measure of the residual r_k for this particular stopping criterion is the maximum norm: $e(r_k) = \max|r_k|$. Note that for practical applications ρ_0 can be approximated by 1kg m^{-3} in the formula for the stopping criterion, leading to $e(r_k) < 0.01/\Delta t_s$kg m^{-3}. This leads to a somewhat higher accuracy in lower altitudes compared to higher altitudes.

6.4 Pressure solver experiments

Finding a solution to the systems of equations $Ax = b$ is not a simple task, since the system contains as many equations as there are grid points in the model. The direct (explicit) solution of the equations or inverting the matrix A is computationally too expensive. Iterative solvers as presented in Appendix B are needed. When more than one grid is present or the grid spacing is nonuniform, the matrix condition

$$\kappa(A) = \frac{\lambda_{\max}}{\lambda_{\min}} \tag{6.21}$$

becomes very large. Here, $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum Eigenvalues of the matrix A. A large matrix condition leads to slow convergence of iterative

solvers. Therefore, in this section the performance of different solvers is investigated in the context of grid refinement. The ILU(0) preconditioned Bi-CGSTAB scheme (Appendix B.1.3, B.2) and MULTIGRID (with 1, 2 and 3 levels of refinement, denoted as MULTIGRID1L, MULTIGRID2L and MULTIGRID3L, Appendix B.4) are tested for different divergence fields on two types of grids. One has a grid aspect ratio $\Delta z/\Delta x$ close to 1 (e.g. micro-scale grid) and the other a grid aspect ratio of much less than one (e.g. mesoscale grid). The CPU time is measured for an AMD Opteron 2.6 GHz CPU.

6.4.1 Aspect ratio $\frac{\Delta z}{\Delta x} \approx 1$

The aspect ratio of vertical and horizontal grid cells is usually close to one for micro-scale grids at lower altitudes and for some mesoscale grids at the model top. This leads to a matrix with the coefficients for $p_{i,j,k\pm1}$, $p_{i,j\pm1,k}$ and $p_{i\pm1,j,k}$ and the main diagonal ($p_{i,j,k}$) being of similar magnitude in the equation for grid point (ijk).

The test domain ranges from $x_{\min} = y_{\min} = -630$m to $x_{\max} = y_{\max} = +630$m in the horizontal, and from $z_{\min} = 0$m to $z_{\min} = 2000$m in the vertical. The surface elevation is given by a bell-shaped curve

$$z_s(x,y) = 100\text{m}\frac{1000^2\text{m}^2}{1000^2\text{m}^2 + x^2 + y^2} \tag{6.22}$$

The divergence fields are created from the wind fields, setting $\hat{v} = 0, \hat{w} = 0$ and

$$\hat{u}(x,y,z) = \begin{cases} \frac{10}{k}\sin\left(\frac{2\pi k(x-x_0)}{\Delta x_d}\right) * \sin\left(\frac{2\pi k(y-y_0)}{\Delta y_d}\right) & \text{for } x_0 < x < x_0 + \Delta x_d \\ 0 & \text{else} \end{cases} \tag{6.23}$$

where $k/\Delta x_d$ and $k/\Delta y_d$ are the horizontal wavenumbers, $x_0 = y_0 = -600$m, and the domain width Δx_d and length Δy_d is 1200m. Note that the area where $\hat{u} \neq 0$ is smaller than the total area to make sure that there is no mass flux out of the domain. The density is set to $\rho_0 = 1\text{kg m}^{-3}$. The resulting momentum field is divergent, and the horizontal extension of the divergence structures depends on k.

The experiment is performed for different grid configurations and values of k. As stopping criterion it is demanded that the divergence of the momentum field is $D_{i,j,k} < 0.0001\text{d}^{-1}$ kg m^{-3} (d is one day or 86400 s).

The domain is resolved with 42x42 grid points in the horizontal ($\Delta x = \Delta y =$30m) and 39 grid points in the vertical (Δz varying from 4m at the surface to 100m at the

top) for the single grid case. In the simulation with two grids the main background grid is the same as in the single grid simulation. Around the center of the model area the grid is refined with a factor three ($\Delta x = \Delta y$ =10m). The fine grid has 42x42 grid points in the horizontal. For both simulations the wavenumber k is varied from 0.5 to 20.

Figure 6.5 shows the resulting relative CPU times. For larger wavenumbers (higher frequencies) Bi-CGSTAB converges faster than for smaller wavenumbers. This is a well known phenomenon for the CG methods as per each iteration step information cannot be spread very far over the domain. This would be necessary to reduce divergences of small wavenumbers. For the current test case this effect is not very strong, possibly due to the small total number of grid points. The MULTIGRID methods also show an increase at very small wave numbers, even though they are designed to decrease low frequency disturbances faster. The reason may be that more refinement levels are necessary to see this effect for wave numbers like 0.5. However, MULTIGRID2L and MULTIGRID3L perform similarly well and need about 40% less time than Bi-CGSTAB. MULTIGRID1L needs a similar amount of time as Bi-CGSTAB. The reason is that the direct solve on MULTIGRID level 1 is still very complex and takes more time than adding coarser levels with a faster direct solve.

For the single grid case it is visible that the CPU time is zero for wavenumber $k = 20$. This is due to the staggered grid on which the $2\Delta y$ wave is zero, since the u grid point always coincides with points where the sine wave is zero. In these cases the initial divergence is zero.

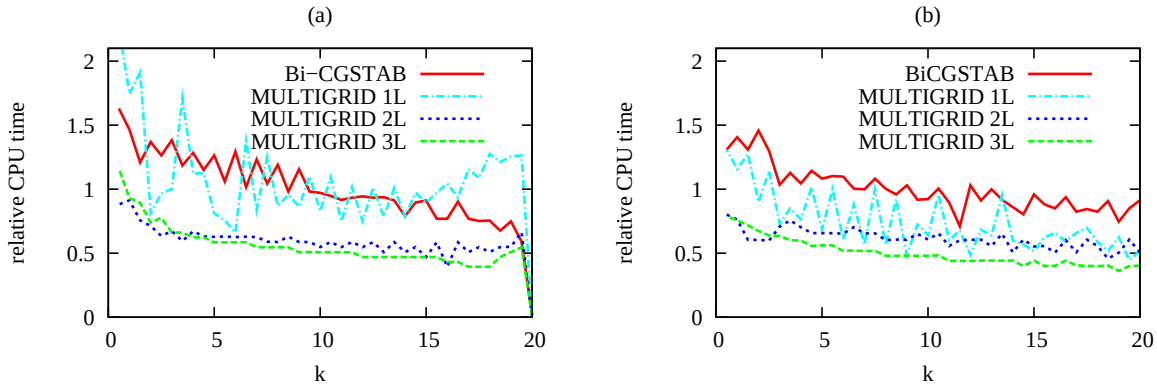

Figure 6.5: CPU times for different schemes and divergence disturbances with different wavenumbers k for grid aspect ratios $\Delta z/\Delta x \approx 1$. CPU times are normalised with the Bi-CGSTAB average over all simulations. The grid aspect ratio is typical for micro-scale type models. (a) Single grid result. (b) Result on two grids with a refinement factor of three.

When the refined grid is added (Fig. 6.5b) the relation of the relative CPU times is similar, though MULTIGRID1L performs somewhat better (relative to the other schemes).

Generally, MULTIGRID with more than 1 coarsening level can be recommended for a grid with aspects ratios between ≈ 1 and larger than one.

6.4.2 Aspect ratio $\frac{\Delta z}{\Delta x} << 1$

In the mesoscale, aspect ratios of vertical and horizontal grid cells are typically different to the micro-scale. The very small aspect ratios at the surface result from a vertical grid spacing that can be two to three orders of magnitude smaller than the horizontal grid spacing. For the equation for grid point (ijk) this leads to large coefficients for $p_{ijk\pm1}$ and p_{ijk} (main diagonal) in comparison to the remaining diagonals. At the model top this effect can be much smaller with a vertical grid spacing that is usually not more than one order of magnitude smaller than the horizontal grid spacing (aspect ratios typically between 0.1 and 1).

The topography of the chosen test case is given in Fig. 6.6. It is located at central Europe with a refined area centered over eastern Germany and western Poland. The vertical grid spacing varies from 20m at the surface to 1km at 12km height with 33 levels. For the horizontal there are three configurations: a uniform grid with $\Delta x = \Delta y = 6$km (334 horizontal levels each, EURO6), a uniform grid with

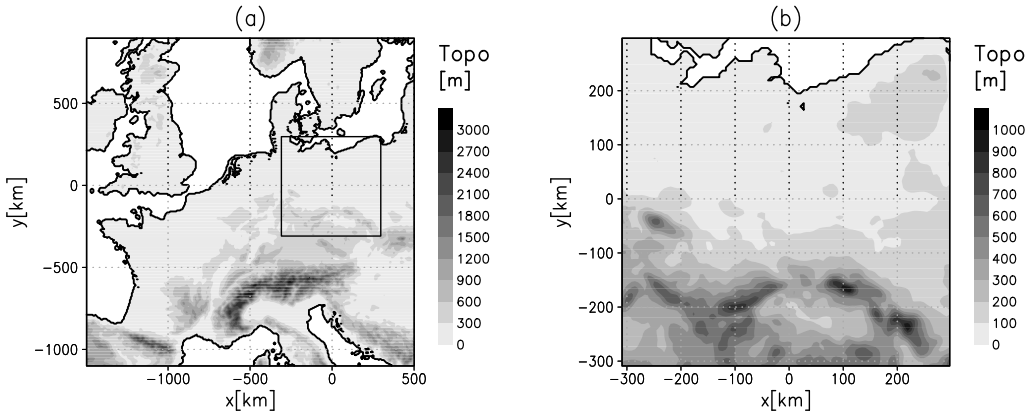

Figure 6.6: Surface height for a typical model area over Europe. (a) Single grid model domain and background grid for domain with two grids. (b) refined area (central east Europe) for simulation with two grids (corresponding to the box indicated in (a)).

$\Delta x = \Delta y = 18$km (111 horizontal levels each, EURO18) and a domain discretised by a uniform background grid with $\Delta x = \Delta y = 18$km and a refined grid as shown in Fig. 6.6b with $\Delta x = \Delta y = 6$km (102 levels in x- and in y-direction, EURO18+6).

The divergence fields are created from the wind fields similarly to the test case in Section 6.4.1, setting $\hat{v} = 0, \hat{\dot{w}} = 0$ and $\hat{u}$ according to Eq. (6.23) with $\Delta x_d = 2000$ km, $\Delta y_d = 1200$ km, $x_0 = -1500\text{km}+\Delta x$ and $y_0 = -1100\text{km}+\Delta y$.

The EURO6 simulation shows the advantage of MULTIGRID with at least 2 levels over Bi-CGSTAB and MULTIGRID1L for low wavenumber disturbances (Fig. 6.7). For larger wavenumbers (small disturbances) the difference between the schemes is smaller, as expected. For the large domain Bi-CGSTAB can remove the small disturbances with a few iteration steps, large disturbances take many more iteration steps due to the high number of grid points. This problem is overcome by MULTIGRID by transforming the small disturbances into larger ones relative to the grid spacing on the coarser levels and thereby making it possible to propagate information much faster than on the finest grid. The positive influence of the grid coarsening is more relevant when two or three refinement levels are used. In comparison to the micro-scale test case the effect of the MULTIGRID is much larger, possibly due to the larger number of grid cells. Again the MULTIGRID1L is not recommended, because the direct solve on level 1 is computationally too expensive.

When the horizontal grid size is increased by a factor of three and very small aspect ratios occur (order 10^{-2}), MULTIGRID is not the best option (EURO18, Fig. 6.8a). Here Bi-CGSTAB performs best. The reason is possibly the low number of grid points and the very small vertical mesh size relative to the horizontal grid

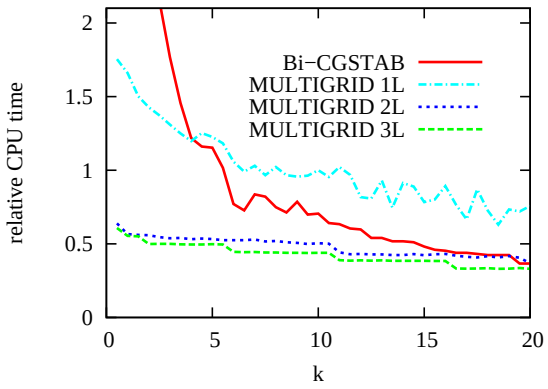

Figure 6.7: As Fig. 6.5 but for the mesoscale 6km resolution domain of Europe (EURO6, Fig. 6.6a).

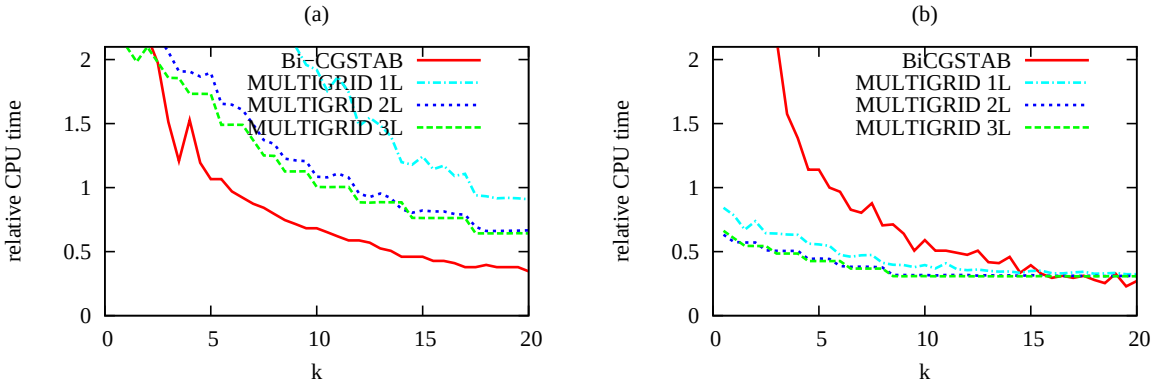

Figure 6.8: As Fig. 6.7 but for (a) 18km resolution (EURO18), and (b) with a refined grid (EURO18+6, Fig. 6.6).

spacing. This leads to a matrix main diagonal with much larger entries than the side diagonals, and the matrix has a better condition (Eq. 6.21). The resulting system of equations can be solved by Bi-CGSTAB with a few iterations, while the MULTIGRID schemes produce an unjustifiable overhead due to the repeated V-cycle and the permanent coarsening and refinement. However, when a refined subgrid is added (EURO18+6, Fig. 6.8b), the MULTIGRID schemes perform better than Bi-CGSTAB. This is likely caused by the bad condition of the matrix resulting from the varying horizontal mesh size and non-symmetric coupling terms in the matrix (Fig. 6.4). On the coarser MULTILEVEL grids the matrices are based only on single, uniform grids (Appendix B.4, Fig. B.5). Therefore, these matrices are better conditioned.

Generally, MULTIGRID can be recommended if a subgrid is used, the total number of grid points is large ($\approx 10^6$) or grid aspect ratios are close to one (accompanied by main and side matrix diagonals of similar magnitude). For domains with very small aspect ratios or few grid points ($\approx 10^5$) Bi-CGSTAB works similarly well or even better.

7 Use of the multiple grids model in different test cases

The results of the previous chapters lead to the optimal set-up that is used for the METRAS version on two grids which is tested in this chapter. In Chapter 5 it was shown that for grid refinement a factor of three is adequate (Schroeder and Schlünzen, 2007b). Larger refinement factors lead to stronger gravity wave dispersion. The pressure solver is also likely to need more time for convergence as the matrix condition gets worse for larger refinement factors due to the increasing asymmetry of the matrix, albeit this was not tested for this work. Consequently, a factor of three is used in this work. MULTIGRID with three coarsening levels is applied with an ILU(0) factorisation of the pressure matrix as Gauss-Seidel smoother (Chapter 6). For advection second order essentially non-oscillatory schemes are used (Chapters 3 and 4, Schroeder et al., 2006; Schroeder and Schlünzen, 2007a), E2UP for momentum advection and FIENO2 for scalar advection. The hitherto implemented centered differences are not stable and cannot be used. Higher order schemes are also not justifiable as they need more computation time and do not significantly improve the solution when turbulent diffusion is present.

The set-up described above is used for three test cases. A rising bubble experiment is used to test the dry dynamical core (Section 7.1), while two test cases apply the full model version including the parameterisation of sub-grid scale physics. These are a cyclogenesis experiment with an idealised polynia (Section 7.2) and a hindcast of the wind conditions at the America's Cup 2007 sailing site (Section 7.3). The test cases are performed with fine and coarse resolution single grids as well as on a set of two grids (a coarse background grid with a coupled fine subgrid).

This test bed gives the opportunity to see the effect of grid refinement for simulations of different complexity. If there are any failures, these are expected to be visible for the least complex test case (rising bubble), while the America's Cup simulation

allows for comparison with measurements.

7.1 Rising bubble experiment

The rising bubble experiment is set up in a similar way as in Section 4.5 (Schroeder and Schlünzen, 2007a). The main difference is that in this section it is three-dimensional. The domain is 2x2x2km^3. A bell shaped mountain is prescribed by

$$z_s(x,y) = H\frac{L_h^2}{L_h^2 + (x-x_c)^2 + (y-y_c)^2}$$

with the peak at $x_c = 0$m and $y_c = 200$m, the height $H = 300$m and the half width $L_h = 500$m (Fig. 7.1). The bubble has a diameter of 500 m and its center is located at $x = y = 0$ and $z = 590$m. The initial potential temperature is 300K everywhere except for the region where the bubble is located. Within the bubble the potential temperature is 300.5K. Neither the bubble nor the orography is symmetric to the refined grid in order to have surface slopes at the internal boundaries and to see the effect of a refined grid in different regions of the bubble. Only the dry dynamical core without Coriolis force is used for the simulations, all parameterisations are omitted.

Four simulations are performed, BUB30, BUB90, BUB30-90 and BUB30-90TS (Tab. 7.1). The vertical grid spacing is 20m in all simulations, the horizontal grid spacing

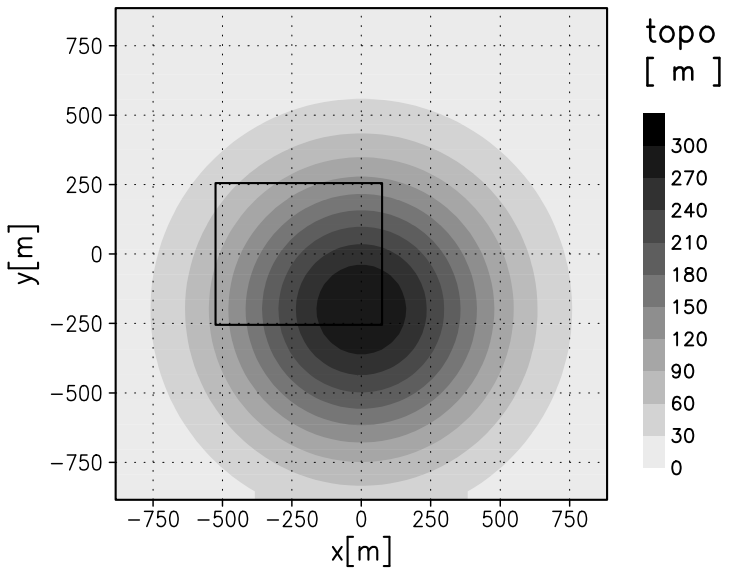

Figure 7.1: Topography for the rising bubble experiment. The rectangle marks the refined area for BUB30-90 and BUN30-90TS.

is 30m everywhere for BUB30 and 90m everywhere for BUB90. For BUB30-90 and BUB30-90TS the grid spacing is 90m except for the refined region which has a grid spacing of 30m and is located at -525m< x <75m and -255< y <255m. The time step Δt is 1s for BUB30, BUB90 and BUB30-90. For BUB30-90TS Δt is 3s on the coarse grid and 1s on the fine grid applying time splitting described in Section 2.4. Table 7.1 summarises the differences of the test cases.

Figures 7.2-7.5 show different cross-sections of the experiment after 10min simulation time. BUB30 acts as reference. In comparison to the two-dimensional result of Section 4.5 (Fig. 4.4), no clear billows develop. This is mainly due to the coarser resolution of the three-dimensional test. Furthermore, in this section only second order advection schemes were applied in contrast to the third order schemes used in Section 4.5. However, due to the strong vertical motion at the bubble's center and the weaker vertical motion further away from the center a mushroom shaped temperature distribution develops. For BUB30 and BUB90 the shape is symmetric to the yz-plane at $x = 0$ (Fig. 7.2a,b) as the initial conditions and the hill are symmetric to $x = 0$. However, as the mountain peak is shifted by 200m in y direction against the center of the bubble, there is no symmetry to the xz-plane at $y = 0$ (Fig. 7.3a,b). The general difference between coarse and fine grid solution is that the solution on the coarse grid is much smoother, as expected.

The solutions look different on two coupled grids. The symmetry to the yz-plane at $x = 0$ is lost, as the location of the fine grid is not symmetric to $x = 0$ (Fig. 7.2c,d, the results shown are aggregated to the grid of BUB90). In the western finely resolved area (-525m $< x <$ 75m) gradients are stronger than in the eastern coarsely resolved area. The difference in the isolines in the finely resolved area of Fig. 7.2c,d and Fig. 7.2a is an artefact resulting from aggregation to the coarse resolution that is used for the plot. This can be verified by comparing the non-aggregated results (Fig. 7.4c,d). Here, only the area is shown that is actually refined. In the centre

Table 7.1: Overview of the different rising bubble test cases.

case	$\Delta x = \Delta y$	time step
BUB30	30m	1s
BUB90	90m	1s
BUB30-90	30m/90m	1s
BUB30-90TS	30m/90m	1s/3s

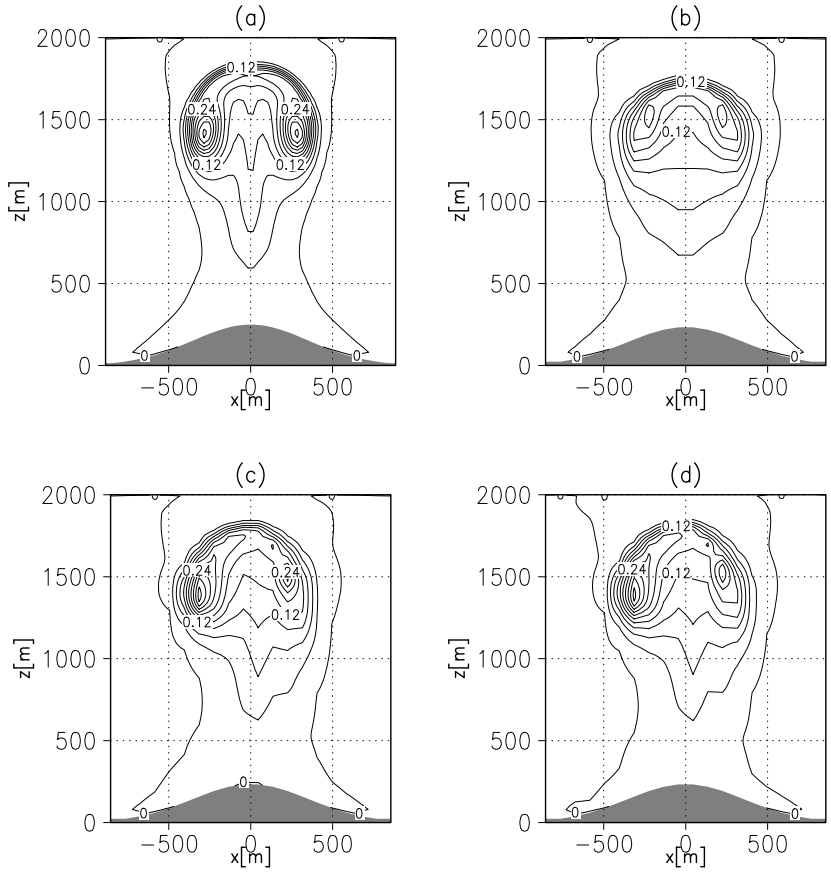

Figure 7.2: Rising bubble experiment after 10min at $y = 0m$ (xz-cross-section). Shown is the potential temperature deviation (increment 0.03K). Configuration: (a) BUB30, (b) BUB90, (c) BUB30-90 and (d) BUB30-90TS (see Tab 7.1 for the configurations). For plotting the results of (c,d) are aggregated to the grid of (b).

of the area the results on the refined grids (Fig. 7.4c,d) are nearly indistinguishable from the fine grid result BUB30 (Fig. 7.4a). At the eastern boundary the influence of the coarse region can be noticed. This is also visible at the yz-cross-section at $x = 0$ (Fig. 7.5). The BUB30-90 and BUB30-90TS solutions (Figs. 7.3c,d and 7.5c,d) are smoother than the BUB30 solution.

The effect off different time stepping and the resulting time splitting of the two grids is hardly noticeable. The results of BUB30-90 and BUB30-90TS look virtually the same in all figures.

The rising bubble experiments have all been performed without the parametrisation of turbulent diffusion. Consequently, the difference in diffusion visible in the results is caused only by the difference in resolution. On grids with different resolution

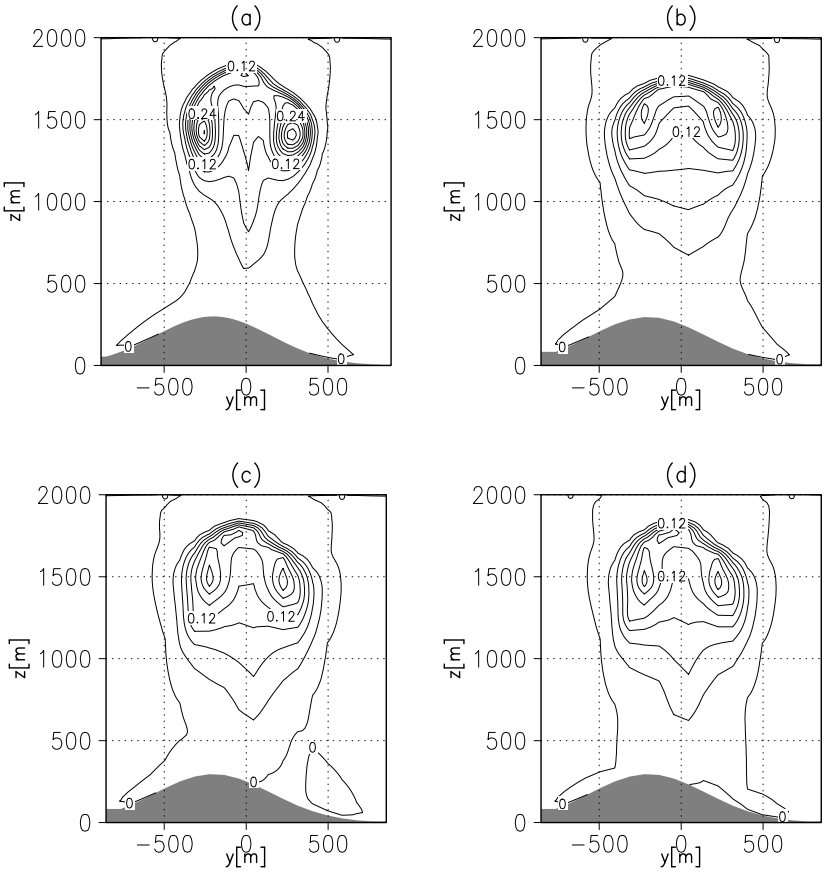

Figure 7.3: As Fig. 7.2 but yz-cross-section at $x = 0m$.

the solutions evolve in a different way, even though the physical conditions are the same. This reveals the importance of resolution dependent parameterisation schemes to achieve consistent results for different grid configurations. However, an important result of the test is that the new dynamical core is able to simulate the bubble on two different grids without unphysical disturbances.

7.2 Cyclogenesis over an idealised polynia

This test case is similar to the ones presented in Bungert (2002) and Hebbinghaus et al. (2007). The full model is used containing turbulent diffusion, Coriolis force, cloud processes and radiative processes. The surface properties are that of sea ice ($z_0 = 1$ mm) with an initial temperature of 237.15K except for the area of an open water surface. Here the surface properties of water are simulated with an initial temperature of 271.35 K. A geostrophic north wind of 3.0ms^{-1} is prescribed. The

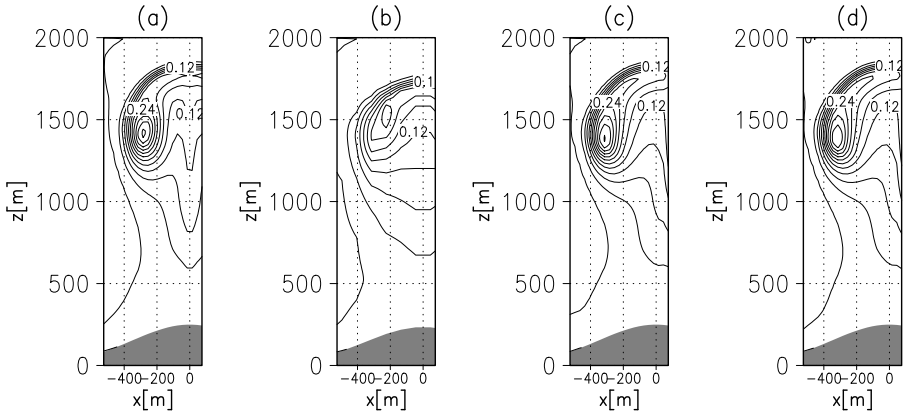

Figure 7.4: As Fig. 7.2 but only the area is shown that is refined in (c) and (d). The results of (c) and (d) are not aggregated.

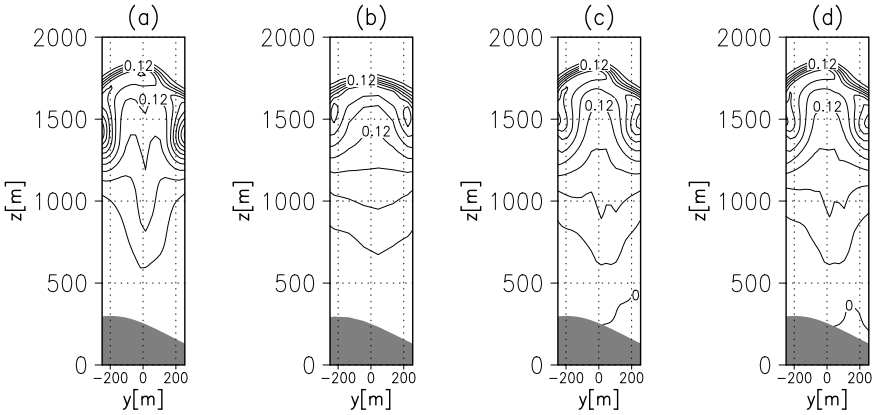

Figure 7.5: As Fig. 7.4 but yz-cross-section at $x = 0m$.

simulation is started at 18 h local time for geographic conditions at 75°N. Due to the strong differential heating which is a common condition in the Arctic sea, cyclogenesis is likely to happen (Hebbinghaus et al., 2007).

Three simulations are performed. A fine grid simulation with 2km resolution (POLY2), a coarse grid simulation with 6km resolution (POLY6) and a simulation on two grids (POLY2-6) with a background resolution of 6km and a refined area of 2km resolution (marked by the black boxes in Fig. 7.6).

Fig. 7.6 shows the surface pressure and wind vectors at 10 m height for the second day at 16h local time (after 22h of simulation). A low pressure area can be found over the polynia and a small vortex has formed and moved southward. The small vortex is clearly visible in the pressure field (relative minimum of 997hPa) and in the

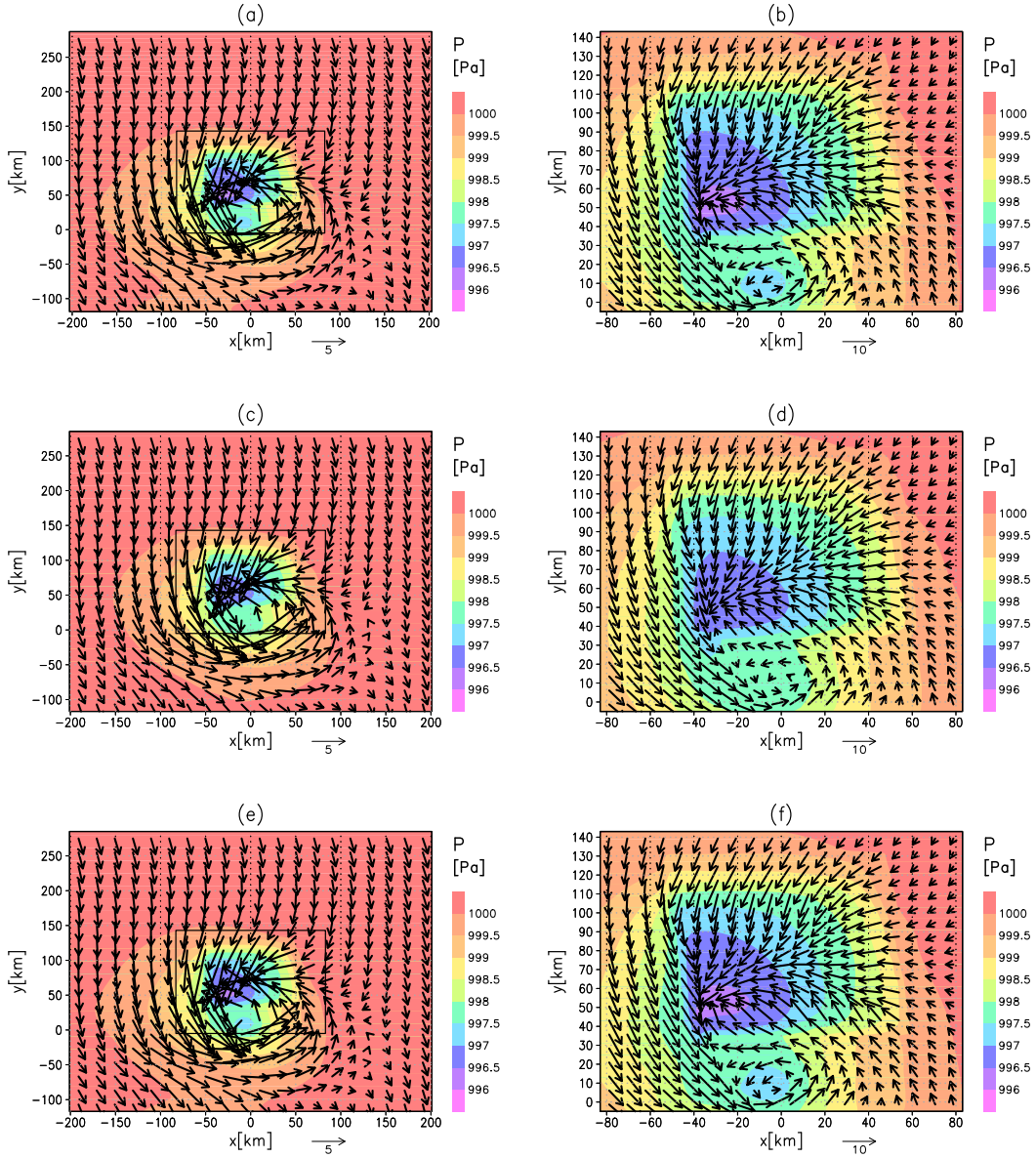

Figure 7.6: Results of the polynia test case at 16 h on the second day. Shown are surface pressure and wind vectors at 10m height for (a),(b) the 2km resolution run POLY2, (c),(d) the 6km resolution run POLY6 and (e),(f) the 2km-6km run POLY2-6. (a),(c),(e) show the entire model area with a box marking the refined area of POLY2-6. Here the POLY2-6 result of (e) is aggregated to the 6km grid of POLY6. (b),(d),(f) show the area that is refined in POLY2-6.

wind field (Fig. 7.6a,b). On the coarse grid (POLY6) the vortex is weaker with the minimum being 0.5 hPa higher (Fig. 7.6c,d). As in the section before the simulation with a refined grid within the coarse grid (POLY2-6, Fig. 7.6e,f) yields better results than the coarse grid simulation. The POLY2-6 results are nearly indistinguishable

from the POLY2 results.

This can also be found when looking at the time series of the minimum pressure (Fig. 7.7). This is located over the polynia due to thermal effects. The curves for POLY2 and POLY2-6 are nearly identical, while the pressure minimum for the POLY6 simulation is about 0.5hPa higher at the end of the second day.

It can be argued that the results of all three simulations are quite similar and the small deviation of the coarse grid results from the other two simulations is negligible for practical applications. However, it is a striking result that the simulation POLY2-6 with a refined grid gives the same results as the simulation on a homogeneous grid (POLY2). The applied numerical methods do not introduce any unphysical disturbances.

7.3 Valencia test case

The original single grid version of METRAS was used to forecast wind fields for the South African sailing team Shosholoza at the America's Cup 2007 (Augustin et al., 2007; Schlünzen et al., 2007). The modelled domain is shown in Fig. 7.8a

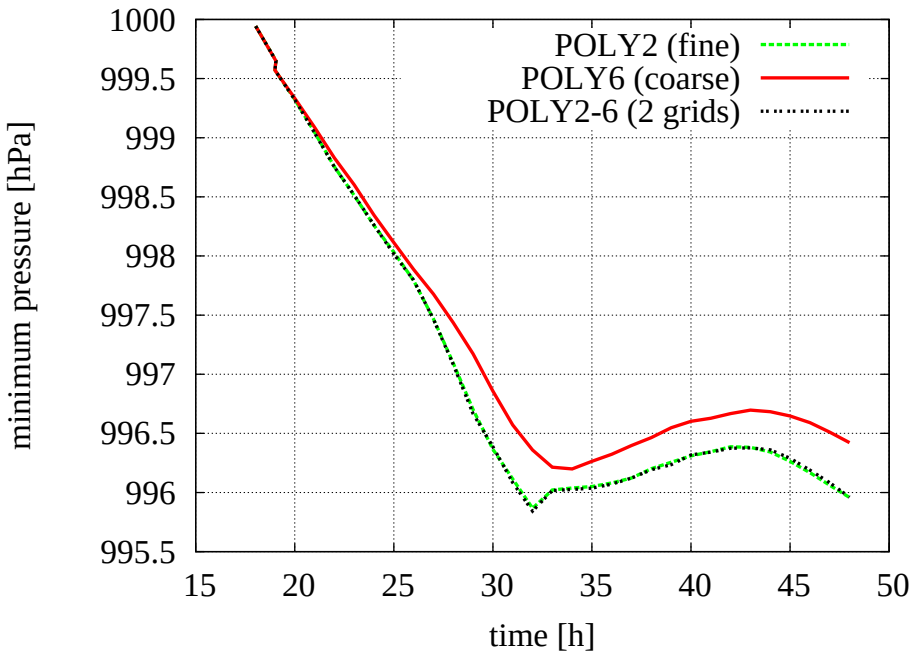

Figure 7.7: Time series of minimum pressure for the polynia test case. Note that the curves for POLY2 and POLY2-6 almost coincide.

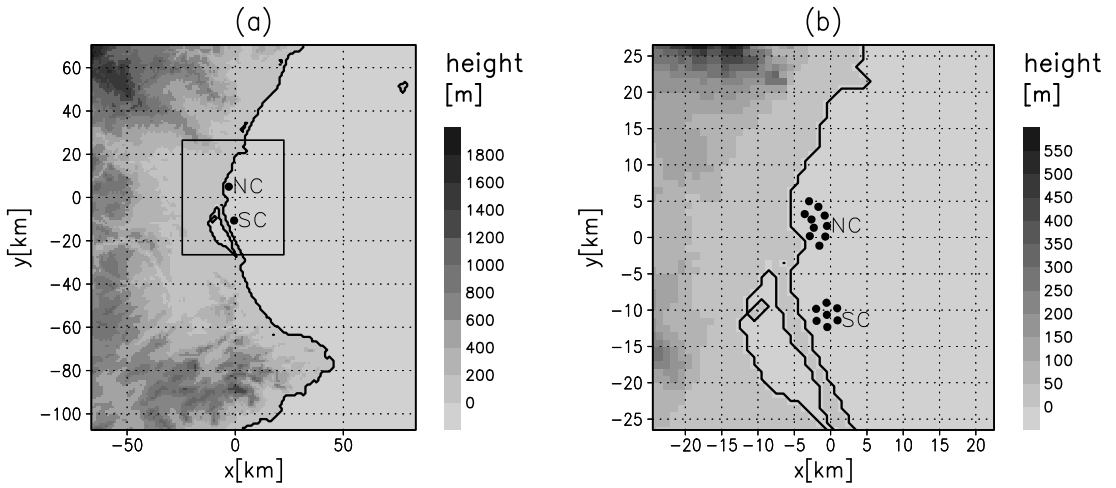

Figure 7.8: Model area and orography height of the Valencia test case. (a) Complete model area with a box marking the refined area of the simulation with two grids. NC marks the north course and SC the south course. (b) Refined area for the test case with two grids. Dots mark the position of buoy measurements of north and south course.

along with the north sailing course (NC) and the south course (SC). The Bay of Valencia is in the centre of the model area. The forecast simulations were forced by the 0 UTC HIRLAM forecasts of the Spanish Weather Service (INM). These were available on 9 vertical levels (surface, 975hPa, 950hPa, 900hPa, 850hPa, 775hPa, 700hPa, 500hPa, 200hPa) with a horizontal resolution of 0.05° (about 5km). The model fields temperature, humidity and the horizontal components of wind were nudged each time step by

$$\chi(x,y,z) = \hat{\chi}(x,y,z) + \nu(x,y,z)(\chi_f(x,y,z) - \hat{\chi}(x,y,z))\Delta t$$

$\hat{\chi}$ denotes the nudged quantity before nudging, χ is the same quantity after nudging, χ_f the forcing field and n is the nudging coefficient that is strongest at the model boundaries and vanishes in the model interior. The METRAS results were compared to buoy measurements available at the two courses (Fig. 7.8b). At spring time, when the America's Cup was held, land-sea-breeze circulations are typically observed during the afternoon. The simulations were started at 3 UTC model time (5 h local summer time) and lasted until 18 UTC (20 h local time).

For this work experiments with different model configurations were performed (Tab. 7.2). Four days were simulated, May 3rd to May 6th, 2007. On May 3rd a cut-off low pressure system, which has separated from a North Atlantic trough, was situated over Spain. During the following three days the low pressure system weakened and

Table 7.2: Overview of different model configurations for the Valencia test case.

	$\Delta x = \Delta y$	momentum advection	scalar advection
MCD	1km	centered differences + filter	upstream
MEF	1km	E2UP	FIENO2
MEF3	3km	E2UP	FIENO2
MEF1-3	1-3km	E2UP	FIENO2
MFF	1km	FIENO2	FIENO2
MFFF	1km	FIENO2 + filter	FIENO2

moved eastward to the west of Italy. Associated with this low is an occlusion that has just passed the Valencia area on May 3rd. On this day the formation of a sea-breeze is inhibited by the frontal clouds (due to a lack of solar insolation). On the subsequent days the Valencia area is more and more influenced by the high pressure ridge over the Azores. Small large-scale pressure gradients and only partly cloudy skies allow the development of a sea breeze. This is the reason why these days were chosen to be simulated, since a simulation is more likely to be improved by local refinement, if the circulation is mainly locally driven. Furthermore, the operational forecast performance of METRAS on May 4th was excellent, and it is of interest if similar results can be achieved with grid refinement.

Fig. 7.9 shows the wind fields simulated with METRAS in the configuration used for the Valencia forecasts with centered differences (MCD) at four different times for 04/05/07. The land breeze that has developed over night (Fig. 7.9a) changes to a sea breeze as the land is heated up during daytime (Fig. 7.9c). The small large-scale pressure gradients are beneficial to this development.

The model results agree well with buoy measurements. The wind speed and direction simulated by MCD for the location of the buoys of north (Fig. 7.10a,b) and south course (Fig. 7.12a,b) develop in a similar way as observed.

To test the effect of different advection schemes three further simulations are performed on the same 1km grid: MEF (second order ENO advection for wind and second order FIENO advection for scalars), MFF (second order FIENO advection for all scalars and momentum) and MFFF (second order FIENO advection for all scalars and momentum plus the additional application of the Shapiro filter for the wind fields at each time step). All three set-ups perform (Fig. 7.11 and Fig. 7.13) slightly worse than MCD (Figs. 7.10a and 7.12a). Among the three ENO set-ups

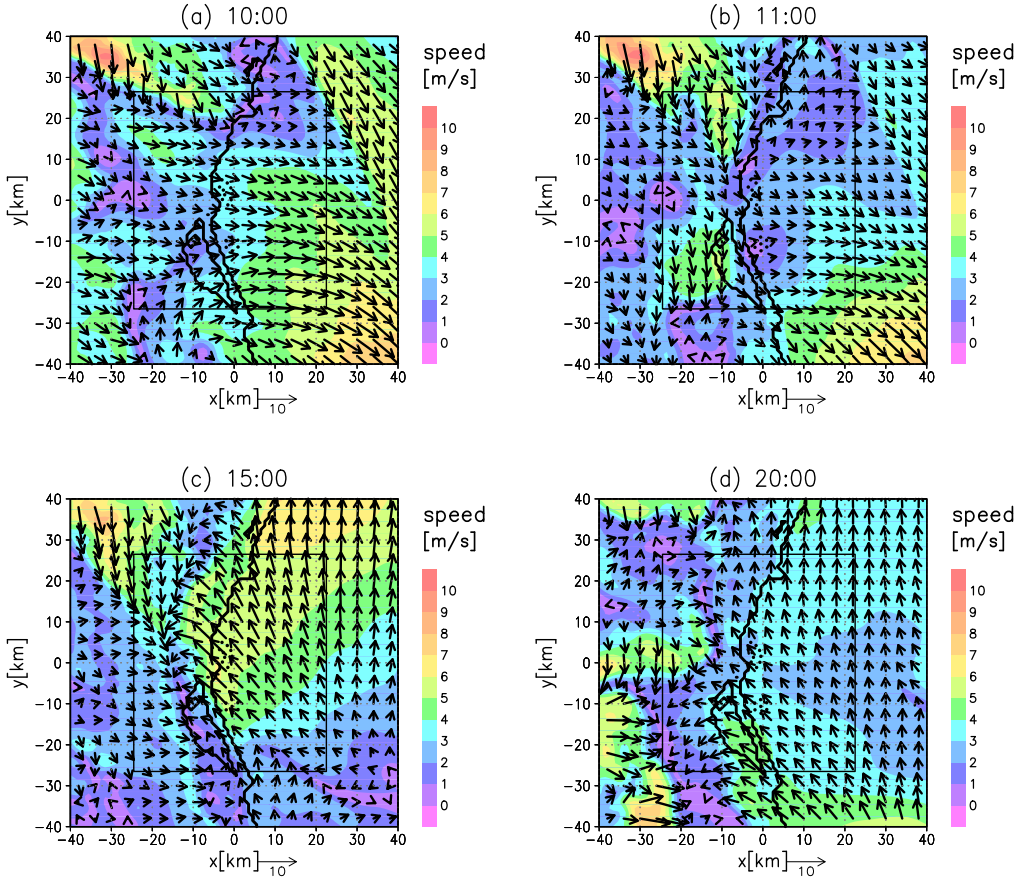

Figure 7.9: Wind speed and wind vector at 10 m height at (a) 10 LT, (b) 11 LT, (c) 15 LT and (d) 20 LT for 04/05/07. The 1km resolution simulation with standard METRAS (MCD) is shown.

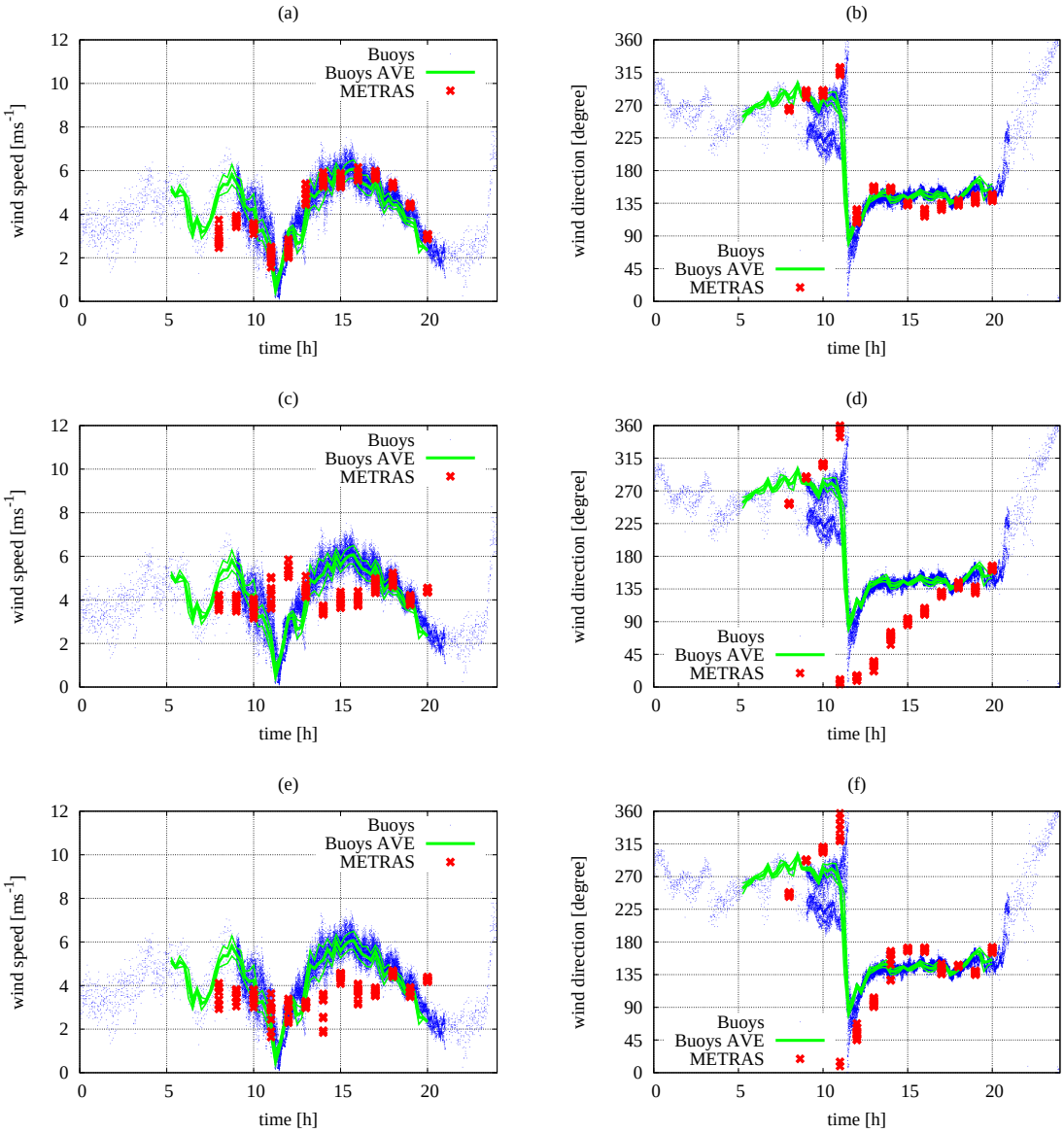

Figure 7.10: North course buoy results (blue dots = single measurements, green lines = 15min average ± standard deviation) and model results (red crosses) on 04/05/2007. Shown is the horizontal wind speed (a, c, e) and the wind direction (b, d, f) for three different simulations, MCD (a,b) MEF3 (c,d) and MEF1-3 (e,f).

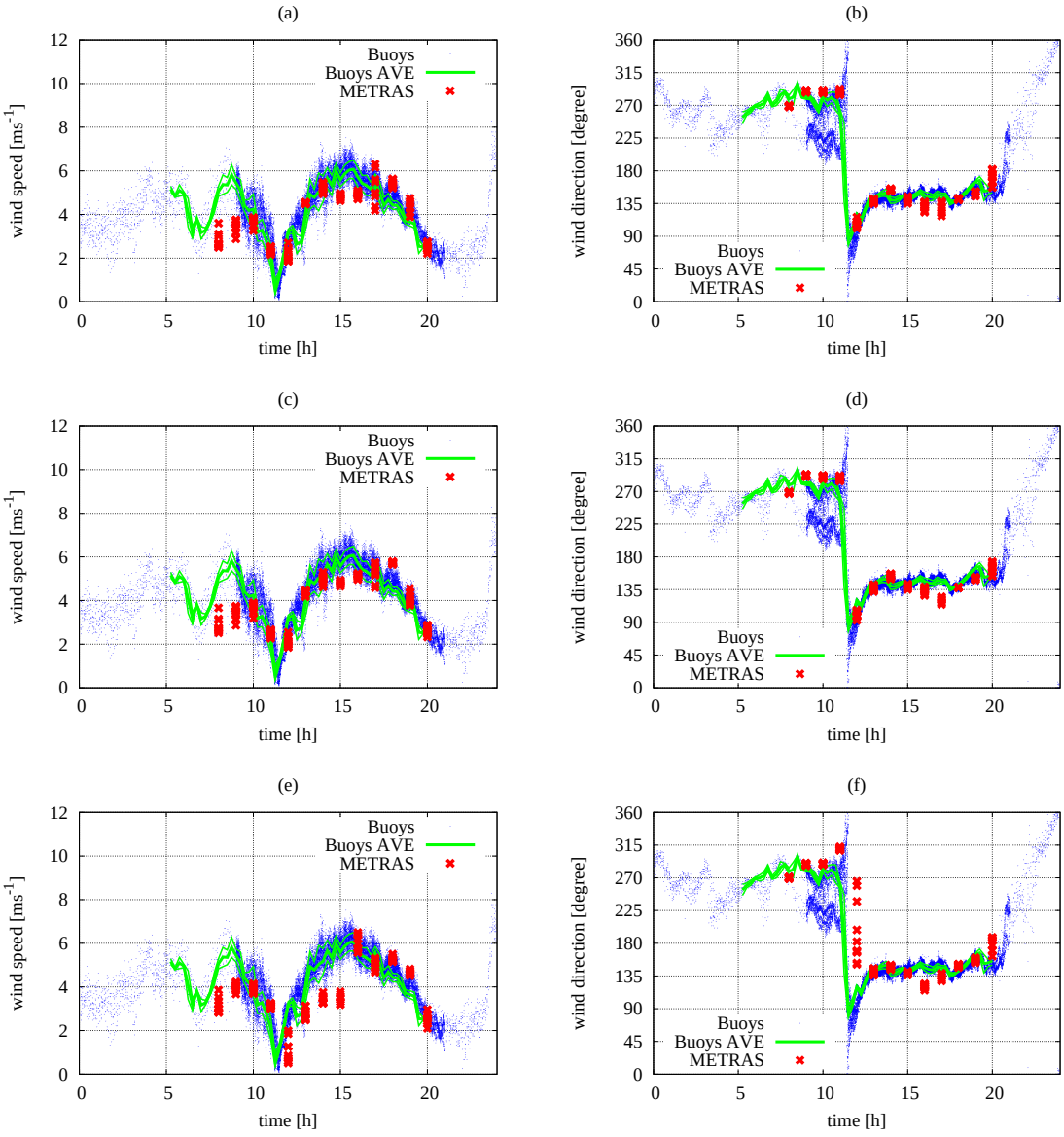

Figure 7.11: As Fig. 7.10 but shown are the 1km METRAS runs with different numerical schemes: (a,b) MEF, (c,d) MFF and (e,f) MFFF.

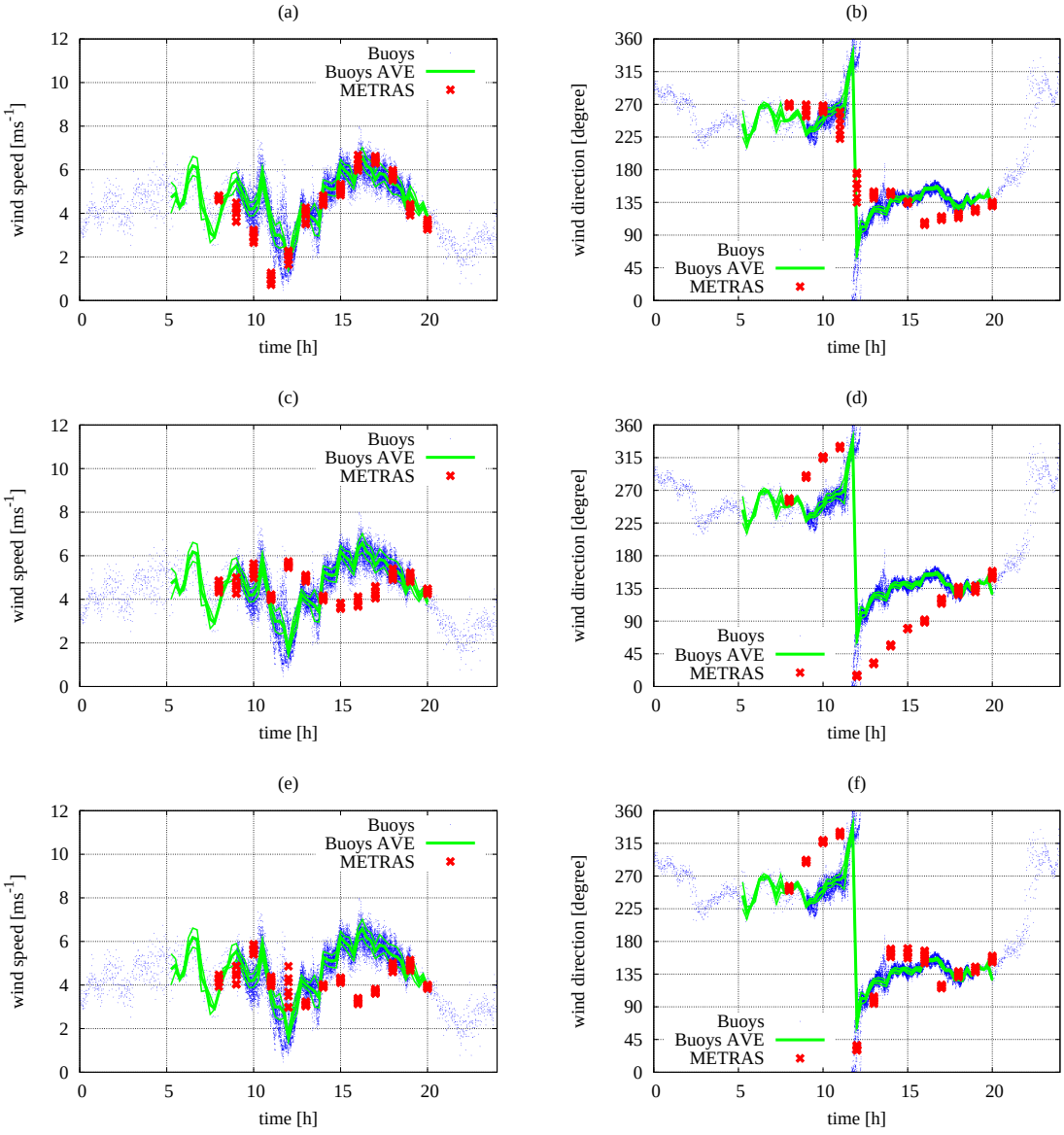

Figure 7.12: As Fig. 7.10 but for the south course.

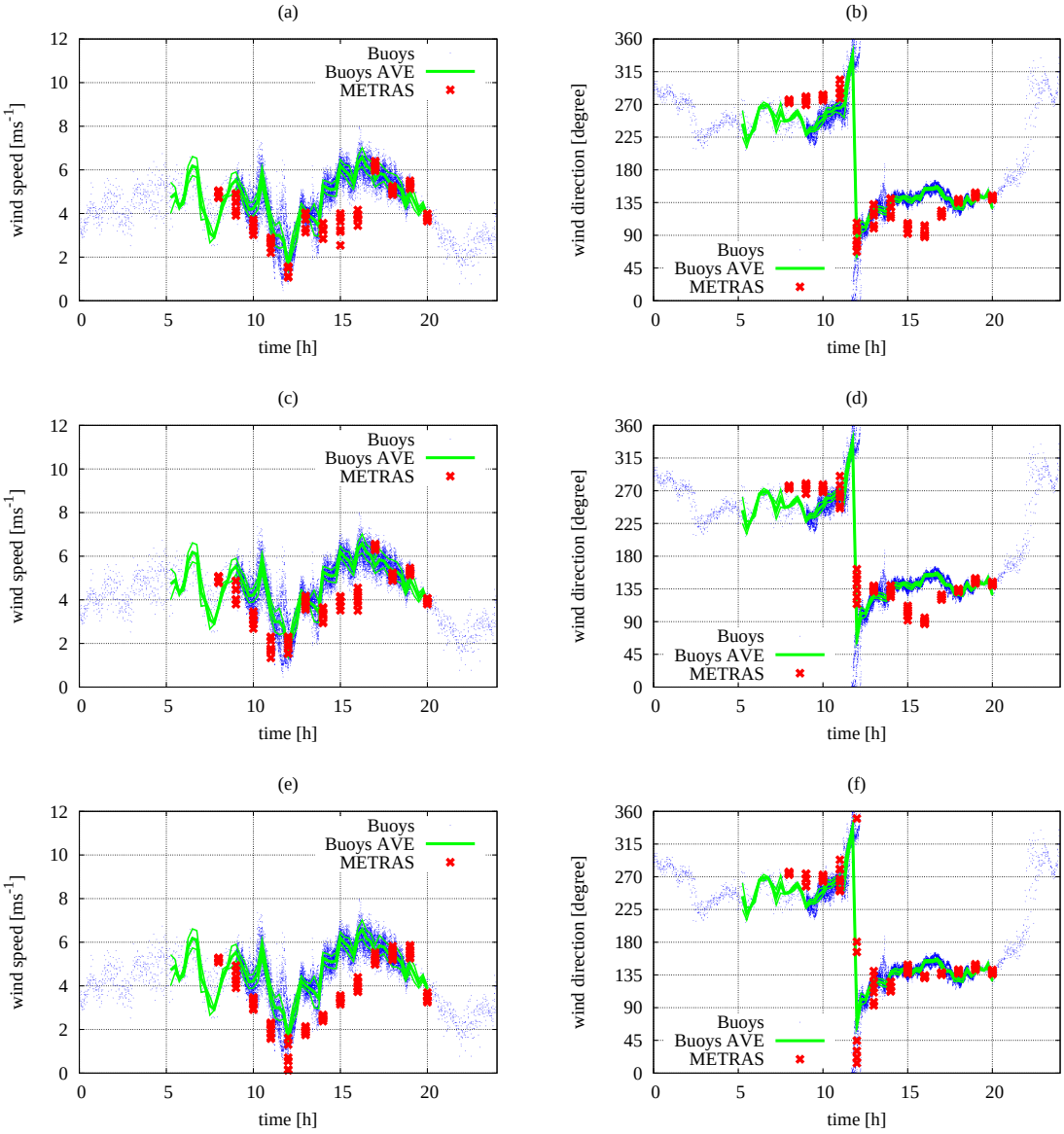

Figure 7.13: As Fig. 7.11 but for the south course.

the filtered one (MFFF) performs worst. Overall, a clear similarity between all four 1km results is visible.

The METRAS version that can contain several grids is comparable to MEF. Not surprisingly, its 3km coarse resolution simulation (MEF3) performs worse than MCD and MEF. Especially, predicting the turning of the land-breeze to a sea-breeze is predicted worse in the coarse simulation (Figs. 7.10c,d and 7.12c,d). This is possibly due to the influence of small-scale orography that is not properly resolved by the 3km grid. When a refined grid is added (MEF1-3, Fig. 7.8b) the improvement is very limited (Figs. 7.10e,f and 7.12e,f). While wind direction is predicted almost as accurate as for MCD and MEF and certainly better than for MEF3, the wind speed is underestimated as in run MEF3. In comparison to MEF it is slightly worse (Figs. 7.11a,b and 7.13a,b).

When looking at the wind fields it is visible that the refined area (7.8b) does not have a great influence (compare Figs. 7.14 and 7.15). Only at 15h local time the simulation with refinement (Fig. 7.15c) shows a different wind pattern than the simulation without (Fig. 7.14c). This is reflected by the more accurate wind direction for the refined simulation. The reason for the only small improvement is possibly the high importance of the small-scale orography, which is not covered by the refined area. A second reason may be the difference in the effect of the forcing fields. The strength of the forcing is decreasing with a greater distance to the boundaries; the distance is measured in grid points. This means that the area in which the forcing has an effect is larger in a coarse grid simulation which has less grid points than in a fine grid simulation.

For comparison and to estimate the significance of the results discussed for May 4th, hit rates and root mean square (RMS) errors are calculated for all four simulated days. The buoy measurements are averaged over 15 minutes to make the data more comparable to the Reynolds averages of the model results. The latter are interpolated to the buoy locations. For each simulation between 12 to 13 hourly values of up to 21 buoys are available leading to 240 to 273 comparison values per simulation for both, wind speed and wind direction. Wind speed is treated as hit if the difference is less than $1 \mathrm{ms}^{-1}$, for wind direction the difference must be less than 30° (Schlünzen and Katzfey, 2003). Tab. 7.3-7.6 summarise the hit rates and RMS errors for the different simulations. The significance of the values is estimated by a simple test: The hit rates and RMS errors for May 4th are calculated twice, once with all data and once by using only 50% of the buoys. According to this test the hit rates can be assumed to have an uncertainty of less than 4%, the RMS error of

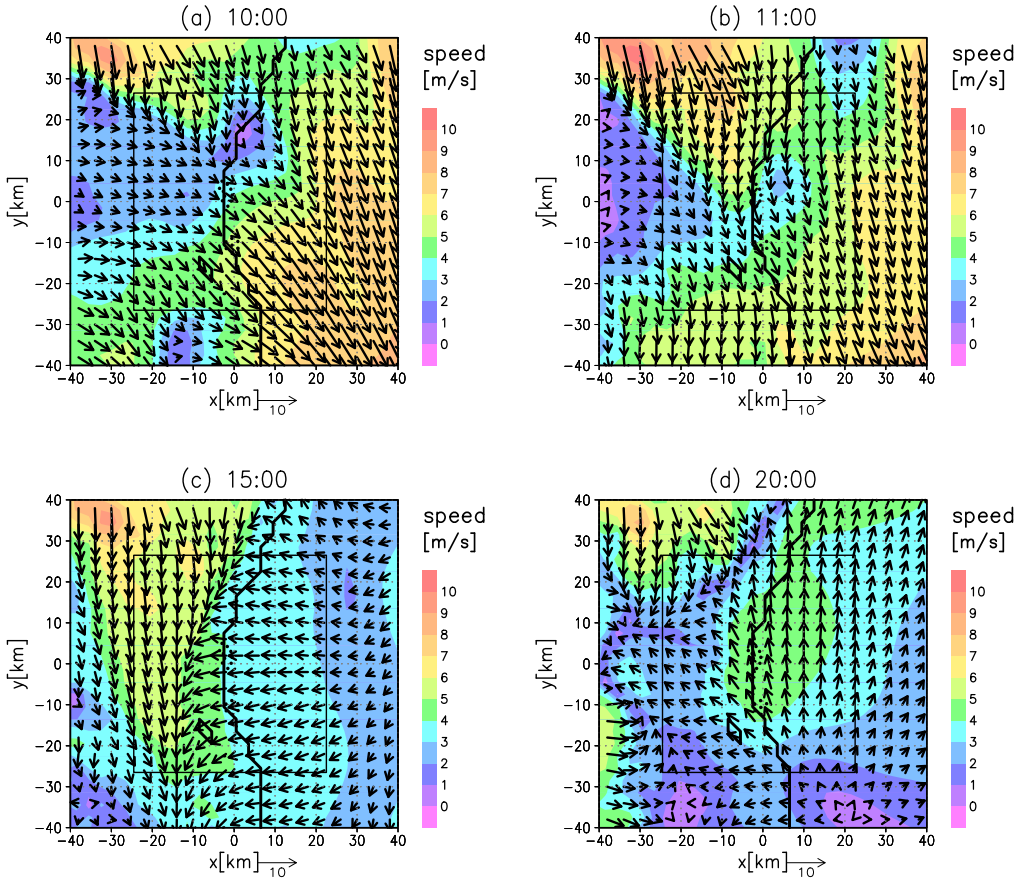

Figure 7.14: As Fig. 7.9 but for the 3km resolution MEF3 simulation.

the wind speed of less than 0.1^{-1} and the RMS error of the wind direction of less than 3°.

As seen above in the time series for 04/05/07 the advection schemes have little influence on the performance (Tabs. 7.3-7.6) While the wind direction is very similar for the four model configurations (hit rates of about 77%), MCD outperforms the other three for the wind speed (hit rates ranging from 53% for MFFF to 77% for MCD). In contrast to that the coarse run MEF3 performs much worse, and the refined run MEF1-3 shows only small improvement over MEF3. For the wind direction the hit rate is about 60% for MEF1-3 compared to 40% for MEF3. For wind speed the hit rate on the coarse grid (47%) is similar to the one on the refined grid (42%), the difference is not significant. To test the influence of the resolution itself the MEF1-3 results are aggregated to the MEF3 grid and the hit rates and RMS errors are calculated based on the interpolated aggregated fields. It is interesting to observe that neither hit rate nor RMS error change significantly compared to the

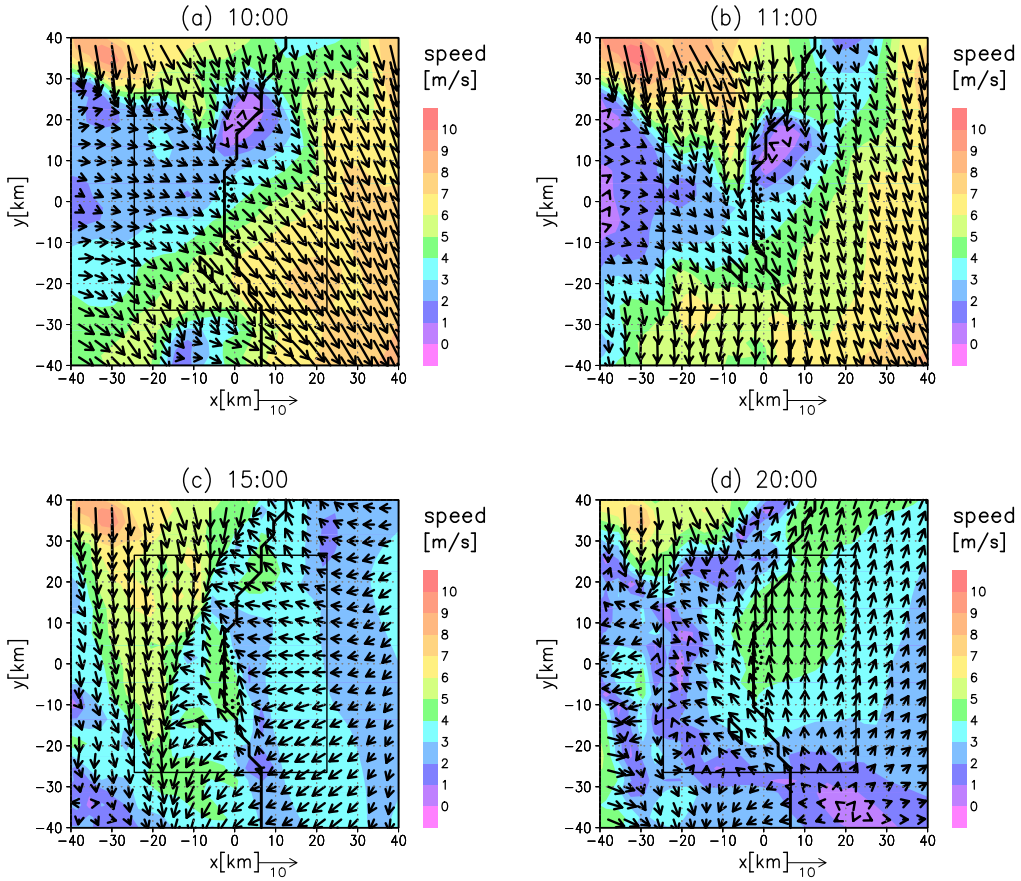

Figure 7.15: As Fig. 7.9 but for the 1km/3km resolution MEF1-3 simulation.

high resolution values of MEF1-3. Some hit rates are even slightly better when they are derived from the interpolated aggregated field, but this is within the range of uncertainty. These findings suggest that (under the current meteorological situation) an improvement of the simulation is not gained due to higher resolved data, but only due to resolving important phenomena more accurately.

On May 3rd the situation is different. Due to the influence of the occluded front and the weak solar forcing the wind is much more variable in space and time. Not surprisingly, all simulations of the day have comparably low hit rates of 22% to 38%. No significant superiority of the fine or refined grid simulation over MEF3 can be found.

On May 5th the hit rates for wind speed are very low again (less than 30%). The sea breeze intensity is underestimated by all simulations. In contrast to that the wind direction is simulated well and it is improved by a higher resolution (MEF1-3). The hit rates reach 78% for MCD, 54% for MEF3 and with the refinement 61% for

MEF1-3.

May 6th is interesting in the sense that MCD performs worst regarding wind speed (48%) compared to 73% for MEF1-3. The wind direction is simulated slightly better with MCD (78%) than with MEF3 and MEF1-3 (70-71%).

In summary, the results of the four simulated days show that there is a clear tendency that the 1km resolution METRAS (MCD) results are better than the 3km resolution METRAS results especially in terms of wind direction. This effect is stronger on days when the model performance is generally good. With the refined grid placed as shown in Fig. 7.8 the simulation of locally driven meteorological situations can be improved in comparison to the coarse grid simulation (Tab. 7.3-7.6).

In the simulations of this section the area of interest was refined, in this case with focus on the area over the water where the sailing races took place. However, refining areas with significant influence on the mesoscale meteorological situation may be more important for a good forecast. In the present simulations this could mean that refining the area over the mountains to the west of Valencia would improve the simulations since the small-scale topography may have an influence on the solution. Generally, it is also likely that the quality of the simulations can be enhanced by relocating the refined area depending on the meteorological situation. This indicates that more research needs to be done on where to place the refined grid under which conditions (adaptivity).

Table 7.3: RMS error of wind direction in ° for different model configurations.

day	MCD	MEF3	MEF1-3	MEF1-3**	MEF	MFF	MFFF
03/05/07	79	90	74	75			
04/05/07	34	60	42	42	30	33	34
04/05/07*	34	58	41	41	31	33	31
05/05/07	36	51	44	43			
06/05/07	32	62	50	51			
average	45	66	53	53			

* results are compared to every second buoy only, not included in the average

** the results are aggregated to the grid of MEF3 and then compared to the buoys

Table 7.4: Hit rates of wind direction in % for different model configurations.

day	MCD	MEF3	MEF1-3	MEF1-3**	MEF	MFF	MFFF
03/05/07	37	35	38	37			
04/05/07	77	40	59	60	76	77	79
04/05/07*	77	41	61	61	76	76	79
05/05/07	78	54	59	61			
06/05/07	78	70	71	71			
average	68	50	57	57			

* results are compared to every second buoy only, not included in the average

** the results are aggregated to the grid of MEF3 and then compared to the buoys

Table 7.5: RMS error of wind speed in ms^{-1} for different model configurations.

day	MCD	MEF3	MEF1-3	MEF1-3**	MEF	MFF	MFFF
03/05/07	3.2	2.9	2.9	2.9			
04/05/07	1.0	1.6	1.5	1.6	1.2	1.1	1.4
04/05/07*	1.0	1.6	1.5	1.5	1.2	1.2	1.4
05/05/07	1.7	2.0	1.9	1.9			
06/05/07	1.4	1.2	1.0	1.0			
average	1.8	1.9	1.8	1.8			

* results are compared to every second buoy only, not included in the average

** the results are aggregated to the grid of MEF3 and then compared to the buoys

7.4 Efficiency of grid refinement

The reason for grid refinement is saving CPU time in comparison to simulations with a uniform fine grid spacing. In addition, the amount of data storage can be reduced and, compared to nested runs, the complexity of the model set-up is low. Here the focus is on CPU time since this is most important when considering weather forecast systems to use multiple grids. For the three test cases of this chapter the CPU time is compared for the different model configurations (Tab. 7.7). The times are measured on an AMD Opteron 2.6 GHz CPU for the polynia test case, and on an Intel Xeon 3.20 GHz CPU for the rising bubble and Valencia test case. Only one CPU was used for all tests. Generally, the fine grid needs approximately 10 times as much CPU time as the coarse grid, because the fine grid has 9 times as many grid points as the coarse grid. In addition, due to the CFL criterion the time step can be smaller if restricted by advection. This is usually the case when the turbulent diffusion is small, e.g. during the night time. However, the time stepping for the

Table 7.6: Hit rates of wind speed in % for different model configurations.

day	MCD	MEF3	MEF1-3	MEF1-3**	MEF	MFF	MFFF
03/05/07	30	22	24	25			
04/05/07	77	47	41	42	70	67	53
04/05/07*	76	45	43	43	66	65	52
05/05/07	26	27	27	27			
06/05/07	48	60	73	72			
average	45	39	41	42			

* results are compared to every second buoy only, not included in the average

** the results are aggregated to the grid of MEF3 and then compared to the buoys

polynia and Valencia test case was mainly dominated by vertical diffusion, so the same time step on coarse and fine grid was used. For the rising bubble test case the time step was set to 1s for BUB30 and BUB90, even though it could have been set to 3s for the coarse grid.

The refined simulations need between two and three times as much CPU time as the coarse simulation although they have only about twice the number of grid points. This can be explained by the overhead caused by handling the internal boundaries between the grids and the reduced convergence of the pressure solver due to the poor matrix condition when two grids are present. This is especially visible for the polynia test case. The single grid simulation has a matrix that is only slightly non-symmetric due to the vertically varying grid spacing, but no orography is present. When a subgrid grid is added the matrix becomes much more non-symmetric compared to the single grid case. For the Valencia this effect cannot be observed, since the single grid matrix is non-symmetric due to the orography and the grid transformation. Here the CPU time is nearly proportional to the number of grid points, which is a sign for the good efficiency of the MULTIGRID pressure solver. When time splitting is applied for the rising bubble experiment (BUB30-90TS, Tab. 7.7) the CPU time is more than 15% smaller than without time splitting (BUB30-90). This is less than half of the expected value 33% (two thirds of the time on the coarse grid, i.e. two thirds of 50%, is saved). This can partly be attributed to the pressure solver which still needs to be calculated every time step.

When the CPU times are related to the actual performance it can be concluded that for the Valencia test case the refinement (MEF1-3) is not justified, as the refined simulation only leads to a small forecast improvement compared to MEF3. However, it is likely that positioning of the refined grid above the mountains could improve the

Table 7.7: CPU time of different test cases and model configurations.

test case	fine grid	coarse grid	refined grid	refined grid + time splitting
rising bubble	56 min	6 min	17 min	14 min
polynia	199.4 h	18.7 h	53.2 h	
Valencia	70.9 h	6.5 h	13.3 h	

refined simulation for the area of interest which lies over the bay. For the polynia case the improvement due to grid refinement is significant and the refinement is justifiable as the refined POLY2-6 results is virtually identical to the fine grid result POLY2. For this test case it could be easily estimated in advance that large gradients would occur at the polynia, where the grid then was refined. The rising bubble simulation is improved in the refined area at a highly reduced CPU time, but this has no meaning for practical applications since the test case is highly idealised.

It should be noted at this point that the refinement does not lead to results that are worse than on the coarse grid, i.e. no unphysical disturbances are caused by the nonuniform grid. Under suitable conditions with the fine grid located at the right place the results are improved compared to the coarse grid with the CPU time being much smaller than with a uniform fine grid. Therefore, the multiple grids version of METRAS can be recommended whenever the location of the refined grid is clear, e.g. for downscaling areas of interest.

8 Conclusions and outlook

In this work methods and schemes necessary to transform a non-hydrostatic model into a model with fully coupled multiple grids and grid refinement are investigated. They are implemented in the non-hydrostatic model METRAS and tested in different applications. The possibility to use multiple grids in METRAS is the key prerequisite for the future development of an adaptive model. The overall goal is to save CPU time, to reduce the necessary storage of model results and minimise the handling time for a model forecast with METRAS while retaining a simulation with high accuracy.

Robust numerical schemes are crucial for a successful grid refinement. Many standard advection schemes like centered differences tend to create spurious oscillations when the grid spacing changes abruptly. This is due to numerical dispersion of advected wave structures. Short waves that are created and resolvable on the fine grid might not be resolved on a coarse grid. When these waves propagate from the fine grid to the coarse grid, they are likely to be reflected at the interface between the two grids. It is shown in Chapter 3 that advection schemes based on essentially non-oscillatory (ENO, Harten et al., 1987) reconstruction and upwind calculation of fluxes (E2UP) do not create spurious oscillations when the grid is refined. The advection schemes themselves are non-linear and thereby avoid exciting the growth of extrema. The schemes solely based on the ENO method are suitable for momentum advection, but for scalar advection they are likely to become unstable and require small CFL numbers. Therefore, for scalar advection a flux-integrated ENO scheme is developed (FIENO, Chapter 4) which is applied in different test cases and proves to be stable for multiple grids. For scalars the FIENO scheme is more recommendable than the ENO scheme.

The numerical dispersion of gravity waves is more difficult to handle on multiple

grids. Disturbances and imbalances of both numerical and physical origin are likely to excite gravity waves. These waves are much more complex than simple advection waves and depend on wind, temperature and pressure. While the physical dispersion mainly depends on static stability, the numerical dispersion in an atmospheric model depends on the grid widths. The group velocity of simulated gravity waves is smaller for coarser grids as the discrete gradients become more inaccurate. The retardation is theoretically derived in Chapter 5 and verified in model simulations. As a result of the dispersion the waves are at least partially reflected when leaving a fine grid and entering a coarse grid. In the worst case, when the wave cannot be resolved on the coarse grid, the wave is trapped within the fine grid region. In Chapter 5 the intensity of reflected waves with different wave length is measured depending on the refinement factor. For waves that are well resolved on the fine grid ($8\Delta x$) with a refinement factor of 3 not more than 26 % of the wave intensity is reflected. For larger refinement factors the reflection becomes much stronger. Consequently, a refinement factor of 3 is used in this work when applying multiple grids.

A main idea of the application of multiple grids is saving CPU time, data storage and handling time by not resolving the entire domain with a fine grid or applying a nesting approach. The used numerical schemes should therefore not be too computationally expensive, when the grid is refined. For explicit schemes the use of grid refinement does not change the CPU time per grid point except for the small overhead caused by the exchange of values at the boundaries. However, when iterative schemes like the solvers for the elliptic pressure equation are used, the convergence rate may decrease when grid refinement is involved. This is mainly due to a bad matrix condition resulting from the increased asymmetry of the matrix. In METRAS pressure is solved iteratively to keep the momentum field free of divergence. For this work two options in METRAS to solve the elliptic pressure equation are extended for the use of multiple grids. In Chapter 6 the methods are compared. The first is an ILU(0) preconditioned Bi-CGSTAB algorithm, the second is a 2- and 3-level MULTIGRID scheme applying an ILU(0) decomposition of the pressure matrix as Gauss-Seidel smoother. The efficiency of the schemes depends on the number of grid points and the grid aspect ratio. If the domain is decomposed into more than one grid, generally MULTIGRID is recommended. Especially for low frequency disturbances it is superior to Bi-CGSTAB. The same is true for a single grid where the grid aspect ratio is $\Delta z/\Delta x \approx 1$ (micro-scale type grid). In this case the matrix has several diagonals of similar magnitude which reduces the efficiency of the preconditioner for Bi-CGSTAB. When the aspect ratio is $\Delta z/\Delta x << 1$ (mesoscale type grid)

the main diagonal and two additional diagonals are much larger than the remaining diagonals. In this case with a small number of grid points ($\approx 10^5$) Bi-CGSTAB can outperform MULTIGRID even for low frequency residuals. This is reversed if the number of grid points is increased ($\approx 10^6$).

The multiple grids version of METRAS is based on the findings above: for momentum advection the E2UP scheme of Schroeder et al. (2006) (Chapter 3) is used, for scalar advection the flux-integrated ENO scheme (FIENO, Chapter 4, Schroeder and Schlünzen, 2007a) is selected. To avoid the problem of gravity wave reflection (Chapter 5, Schroeder and Schlünzen, 2007b) the refinement factor is restricted to 3. The MULTIGRID scheme is applied for solving pressure, when multiple grids are present. Three test cases are performed with three grid configurations (coarse, fine, refined). The first, an idealised rising bubble experiment, mainly serves as a test of the model robustness. The second, an idealised polynia experiment, is included to have a test, where the major physical processes are concentrated in an area that is known a priori. The third test, a hindcast of the America's Cup 2007 wind conditions at the Bay of Valencia, is used to investigate the suitability of the multiple grids model for real applications. The most important result is that spurious disturbances due to the internal boundary between grids with different resolution are not visible in any test case. The CPU time of the refined grid simulation is always much less than in the uniform grid simulation (factor 3-5). The effect of time splitting, i.e. using a smaller time step in the refined grid, is investigated for the rising bubble experiment and shows no significant difference to the uniform time step simulation. In two cases (rising bubble and polynia) the result in the refined grid region is as good as in the uniform grid simulation. For the Valencia test case, however, there is only some improvement of the refined solution over the uniform coarse grid solution. Here, the refined grid was placed where a higher resolution of the wind field was needed. Instead, the refined grid should be placed where physical processes take place that have major influence on the overall solution. In the polynia and rising bubble test case the region of refinement is obvious - it is the region of the polynia itself where large gradients can be expected. For the simulation of the wind conditions at the America's Cup 2007 sailing site this is different. It cannot be easily estimated in advance where the refined grid(s) should be placed, since this is also likely to depend on the meteorological situation.

This shows that adaptive methods are necessary to gain maximum benefit from grid refinement. Most dynamic adaptive methods work with gradients (e.g. Brackbill and Saltzman, 1982; Berger and Oliger, 1984). If sharp gradients develop in an

area (e.g. frontogenesis) refinement can improve the solution as for the polynia and rising bubble test cases (Chapter 7). Bacon et al. (2000) point out that gradients in the atmosphere are generally weak and difficult to detect due to almost ubiquitous turbulence. Therefore, more sophisticated physical indicators must be developed. For example, they can be based on pressure minima (e.g. for tornados or hurricanes) or temperature and moisture stratification (for convection). Static adaption can also be derived from experience. Local flow patterns that develop due to orographic or land-sea effects may only be simulated if the respective area is resolved with a fine grid throughout the entire simulation and entire domain.

For steady flows Feistauer et al. (2003) describe indicators based on residual errors and on shocks. While shocks are not of any relevance for mesoscale or micro-scale flow, the residual error methods may be useful for stationary atmospheric simulations.

Noelle and Steiner (2007) have a different approach for adaptivity. They define a target functional that is calculated from the model variables. The value of the target functional is supposed to be simulated as accurate as possible thereby assuming the whole simulation will improve. The simulation is run forward on a coarse grid. Then the adjoint problem for the target functional is solved to determine the (space-time split) error source and thereby the time step and area to be refined. Noelle and Steiner (2007) test this approach successfully for Burger's equation and time adaption. Ideal target functionals for meteorological applications would be accumulated rain or the amount of deposited pollutants.

A further problem that arises from using multiple grids is to account for the scale dependence of parameterisations. In this work only one level of refinement was applied within a range of scales (1km/3km, 2km/6km) where using the same parameterisation schemes for coarse and fine grid is still acceptable. However, when several refinement levels are used and grid spacing of less than 1km are used the parameterisation schemes must be adjusted accordingly. The finer the grid is the more processes are explicitly simulated by the model and its effects must not be included in any parameterisation scheme. For example, Bacon et al. (2000) report that they apply a continuous weighting factor to their convection scheme. Where the resolutions is 3km or higher the scheme is switched off, while it is assumed that the effect of cumulus convection must be fully parameterised for resolutions coarser than 50km. For METRAS a similar method will be necessary if a grid width of smaller than 1km is used in order account for turbulence which is more and more explicitly resolved the higher the resolution. A possible method is to use the large

eddy simulations (LES) approach, where the mixing length depends on grid width. Thereby the amount of parameterised turbulence is reduced for smaller grid spacing. Currently, Fock (2007) investigates this LES approach with METRAS where the mixing length is proportional to the grid width.

Bohnenstengel (2007) is currently investigating the effect of resolution and subgrid-scale surface parameterisations on the model performance. When subgrid-scale surface information is available the surface fluxes can be determined in two alternative ways. The fluxes can be calculated for each surface class separately and are then aggregated (flux aggregation method with blending height concept). The second method is to create an artificial surface class whose surface parameters (e.g. roughness length z_0) are given by a weighted average of the respective parameter of each contributing surface class (parameter averaging). The artificial parameters are used to calculate the surface fluxes. Generally, the flux aggregation method is superior to parameter averaging, but also computationally more expensive. It was shown by Schlünzen and Katzfey (2003) that the flux aggregation is nearly independent of the resolution which makes the method very suitable for grid refinement. Recent studies of Bohnenstengel (2007) have shown that at resolutions of $\approx$ 1km results for parameter averaging are already close to results using flux aggregation.

When the grid is dynamically adaptive, the values of the meteorological fields on the newly created refined grid need to be defined. Simple interpolation from the formerly coarse grid is likely to create unbalanced fields. The problem becomes significant if the orography on the fine grid contains small scale features that are not resolved on the coarse grid. In many cases the spurious generation of gravity waves will occur. Additionally, it is questionable how fine surface grid cells should be treated that are below the surface of the coarse grid. Extrapolation will certainly cause disturbances since in the surface layer the distance to the surface is determining the flow rather than the height above sea level. Bacon et al. (2000) try to avoid this extrapolation problem by simply not allowing the surface on the refined grid to have a lower elevation than the surface on the coarse grid. The drawback of this is that fine scale topographic features are lost. These features may have important influence on the flow and the grid refinement will be less successful when they are omitted.

An adaptive multiscale model can be built, when all these problems are solved, i.e. finding suitable adaptive refinement methods, scale dependent parameterisations and interpolation methods for defining the refined grid values. The results presented in this thesis are an important step towards this goal by proving a robust dynamical core.

Acknowledgements

This work was supported by the Deutsche Forschungsgemeinschaft within the Graduiertenkolleg "Erhaltungsprinzipien in der Modellierung und Simulation mariner, atmosphärischer und technischer Syteme" (Conservation principles in modelling and simulation of marine, atmospheric and technical systems) and the DFG project "SCHL 499/2-1". Furthermore, part of the work was funded by T-Systems in the framework of supporting Team Shosholoza for the 32nd America's Cup in Valencia in 2007. In this context special thanks go to Dipl.-Met. Günter Meschkat of T-Systems for not only making this collaboration possible, but also for giving me the exciting opportunity to work at the Shosholoza base in Valencia.

Many, many thanks are to my supervisor Prof. Dr. K. Heinke Schlünzen for her great and persistent support in this work, especially when things were not going so well. I would also like to express my gratitude to Prof. Dr. Maria Lukáčová-Medviďová for useful discussions and help provided.

Thanks are also to all members of the METRAS/MITRAS working group for the great team work and work climate. Here special thanks go to Dipl.-Met. Sylvia Bohnenstengel for sharing the office in a nice atmosphere and for fruitful discussions, Volker Reinhardt for answering patiently any question on UNIX and Fortran issues, Dipl.-Met. Ursula Bungert for the good collaboration in the DFG project and Dipl.-Met. Hinnerk Ries for helping me handing in the thesis from abroad. I would also like to thank Dipl.-Met. Torben Kunz for discussions on my work.

Thanks are to the University of Hamburg for providing computing time on the NEC SX-6 vector computers of the Deutsches Klimarechenzentrum (DKRZ) where some calculations have been carried out. Thanks go also to Dr. Otto Böhringer for useful tips on parallel programming and compiler issues and to Dipl.-Ing. Rainer Senke from CIS for IT support.

HIRLAM data have been kindly provided by the Spanish Weather Service INM for 32nd America's Cup and could be used for this thesis. Thanks are to the AC Management for providing the MDS data.

Finally, I would like to thank my family and friends for support and having a great time.

List of important symbols

$\mathcal{A}$	advection operator
$\mathcal{B}$	buoyancy operator
$\mathcal{C}$	Coriolis force operator
$\mathcal{D}$	divergence operator
$\mathcal{P}$	pressure operator
α^*	grid cell volume
$\delta_{\dot{x}}$	difference operator in x-direction
$\delta_{\dot{y}}$	difference operator in y-direction
$\delta_{\dot{z}}$	difference operator in the vertical direction
δ_{ij}	Kronecker symbol
Δt	time step
Δx	horizontal extension of a scalar grid cell in x-direction
Δy	horizontal extension of a scalar grid cell in y-direction
Δz	vertical extension of a scalar grid cell
η	transformed vertical coordinate
ρ_0	background density, only dependent on z
$\tilde{\rho_0}$	mesoscale deviation density
ν	nudging coefficient
Ω	angular momentum of the earth
φ	latitude

ψ_c weighting factor for momentum flux for cell $r - 0.5$

ψ_f weighting factor for momentum flux for cell $r + 0.5$

ψ_u reduction of x-gradient in boundary cell u

ψ_w reduction of x-gradient of boundary cell w

ψ_W reduction of x-gradient of boundary cell W

$\hat{\chi}$ variable before forcing

χ any variable, e.g. as parameter for operators, could also be a vector

χ_f forcing variable taken from HIRLAM

θ_0 background potential temperature

$\tilde{\theta}$ mesoscale deviation potential temperature

$\overline{\cdot}^{\bullet\dot{x}}$ weighted average in x-direction

$\overline{\cdot}^{\bullet\dot{y}}$ weighted average in y-direction

$\overline{\cdot}^{\dot{x}}$ average in x-direction

$\overline{\cdot}^{\dot{y}}$ average in y-direction

$\overline{\cdot}^{\dot{z}}$ average in z-direction

$\overline{\overline{\cdot}}^{\dot{y}}$ parabolic interpolation in y-direction

$\overline{\cdot}$ Reynolds average

A^τ, x^τ transpose of matrix A and vector x

A horizontal extension of a vector grid cell in x-direction;
also used as the matrix in the equation $Ax = b$

B horizontal extension of a vector grid cell in y-direction

C vertical extension of a vector grid cell

c_p specific heat for air at constant pressure

c_v specific heat for air at constant volume

D $\partial z_s/\partial x$

E $\partial z_s/\partial y$

f Coriolis parameter

f'	Coriolis parameter
FD	$\partial z/\partial x\vert_{\dot{z}=\text{const}}$
FE	$\partial z/\partial y\vert_{\dot{z}=\text{const}}$
$\vec{F}_{\vec{m}}$	turbulent diffusion for momentum vector
$\vec{F}_{\chi}$	turbulent diffusion for quantity χ
$F_{\overline{u}}$	turbulent diffusion for $\overline{u}$-momentum
$F_{\overline{v}}$	turbulent diffusion for $\overline{v}$-momentum
$F_{\overline{w}}$	turbulent diffusion for $\overline{w}$-momentum
$\vec{F}_{\vec{m}}^{n-3/6}$	momentum boundary flux vector from the fine grid to the coarse grid at $t = t^{n-3/6}$ for all three momentum components
F_s^n	scalar boundary fluxes from the fine grid at $t = t^n$
$F_{i+1/2j}^x$	advection flux in x-direction
$F_{ij+1/2}^y$	advection flux in y-direction
G	vertical grid transformation coefficient
g	acceleration of gravity
i	index in x-direction on fine or single grid
I	index in x-direction on coarse grid
j	index in y-direction on fine or single grid
J	index in y-direction on coarse grid
k	index in z-direction
$\vec{m}$	momentum field
$\hat{\vec{m}}$	preliminary non-divergence free momentum field
N_x	refinement factor (x-direction) between coarse and fine grid
N_y	refinement factor (y-direction) between coarse and fine grid
p	pressure
P	pressure coarse grid
p_0	background pressure only dependent on z
p_1	mesoscale hydrostatic part of pressure
p_2	pressure that makes the momentum field divergence free

$\hat{p}$	pressure perturbation to correct p_2 at the new time step
p_g	large scale pressure (used for geostrophic wind)
q_v	water vapour content
q_l	cloud liquid water content
q_r	rain water content
R_d	gas constant for dry air
R_v	gas constant for water vapour
S_χ	sources and sinks of quantity χ
$\hat{u}$	preliminary velocity in x-direction
$\hat{U}$	preliminary velocity in x-direction (coarse grid)
u	velocity in x-direction
u_g	large scale geostrophic wind in x-direction
$\hat{\dot{u}}$	preliminary transformed vertical velocity
v	velocity in y-direction
v_g	large scale geostrophic wind in y-direction
$\hat{v}$	preliminary velocity in y-direction
$\vec{\hat{v}}$	preliminary velocity vector
$\hat{\dot{W}}$	preliminary transformed vertical velocity (coarse grid)
$\dot{w}$	$= u\frac{\partial \dot{z}}{\partial x} + v\frac{\partial \dot{z}}{\partial y} + w\frac{\partial \dot{z}}{\partial z}$, transformed vertical velocity
x	west-east coordinate
$\dot{x}$	transformed west-east coordinate
y	south-north coordinate
$\dot{y}$	transformed south-north coordinate
z	vertical coordinate
η	transformed vertical coordinate
$\dot{z}$	transformed vertical coordinate, computational grid
z_s	surface elevation above sea level
z_0	roughness length

A Pressure matrix

The definitions of pressure gradient and divergence operator in Chapter 6 directly lead to the pressure matrix. In this appendix the original matrix on a single grid is derived and the matrix elements for the internal boundaries between a coarse and a fine grid. It should be noted that in the following the coarse grid pressure perturbation values are denoted by capital $\hat{P}$, the fine grid values by lower case $\hat{p}$.

A.1 Matrix coefficients - single grid

The composition of the divergence operator (6.3) and the pressure gradient operator (6.4)-(6.6) leads to the divergence caused by the pressure gradients.

The divergence balanced by the pressure gradient in x-direction can be calculated by taking term (6.3a) and replacing $(\overline{\rho_0}^{\dot{x}}\hat{u})_{i+0.5jk}$ by $\mathcal{P}_{i+0.5jk}$ from Eq. (6.4):

$$\begin{aligned}
\Delta y_j \Delta z_k \quad & \left[\delta_{\dot{x}}\left(\overline{\rho_0}^{\dot{x}}\overline{G}^{\bullet\dot{x}}\hat{u}\right)\right]_{ijk} \\
= \quad & \Delta y_j \Delta z_k \\
& \Big[\frac{\overline{G_{i+1/2\,j}}^{\bullet\dot{x}}}{A_{i+1/2}}(\hat{p}_{i+1\,jk}-\hat{p}_{ijk}) - \frac{\overline{G_{i-1/2\,j}}^{\bullet\dot{x}}}{A_{i-1/2}}(\hat{p}_{ijk}-\hat{p}_{i-1\,jk}) \\
& -\frac{\overline{F_k}^{\dot{z}} D_{i+1/2\,j}}{\overline{C_k}^{\dot{z}}} 0.25(\hat{p}_{i+1\,jk+1}+\hat{p}_{ijk+1}-\hat{p}_{i+1\,jk-1}-\hat{p}_{ijk-1}) \\
& +\frac{\overline{F_k}^{\dot{z}} D_{i-1/2\,j}}{\overline{C_k}^{\dot{z}}} 0.25(\hat{p}_{i-1\,jk+1}+\hat{p}_{ijk+1}-\hat{p}_{i-1\,jk-1}-\hat{p}_{ijk-1})\Big]
\end{aligned} \tag{A.1}$$

The divergence balanced by the pressure gradient in y-direction is calculated by

taking term (6.3b) and replacing $(\overline{\rho_0}^{\dot{y}}\hat{v})_{ij+0.5k}$ by $\mathcal{P}_{ij+0.5k}$ from Eq. (6.5):

$$\begin{aligned}
\Delta x_i \Delta z_k \quad & \left[\delta_{\dot{y}}\left(\overline{\rho_0}^{\dot{y}}\overline{G}^{\bullet\dot{y}}\hat{v}\right)\right]_{ijk} \\
= \quad & (\Delta x \Delta z)_{ijk} \\
& \Big[\frac{\overline{G_{ij+1/2}}^{\bullet\dot{y}}}{B_{j+1/2}}(\hat{p}_{ij+1\,k}-\hat{p}_{ijk}) - \frac{\overline{G_{ij-1/2}}^{\bullet\dot{y}}}{B_{j-1/2}}(\hat{p}_{ijk}-\hat{p}_{ij-1\,k}) \\
& -\frac{\overline{F_k}^{\dot{z}}E_{ij+1/2}}{\overline{C_k}^{\dot{z}}}0.25(\hat{p}_{ij+1\,k+1}+\hat{p}_{ijk+1}-\hat{p}_{ij+1\,k-1}-\hat{p}_{ijk-1}) \\
& +\frac{\overline{F_k}^{\dot{z}}E_{ij-1/2}}{\overline{C_k}^{\dot{z}}}0.25(\hat{p}_{ij-1\,k+1}+\hat{p}_{ijk+1}-\hat{p}_{ij-1\,k-1}-\hat{p}_{ijk-1})\Big]
\end{aligned} \tag{A.2}$$

The divergence balanced by the pressure gradient in z-direction is calculated by taking term (6.3c) and replacing $(\overline{\rho_0}^{\dot{z}}\hat{w})_{ijk+0.5}$ by $\dot{\mathcal{P}}_{ijk+0.5}$ from Eq. (6.6)):

$$\begin{aligned}
\Delta x_i \Delta y_j \quad & \left[\delta_{\dot{z}}\left(\overline{\rho_0}^{\dot{z}}CG\hat{w}\right)\right]_{ijk} = (\Delta x \Delta y)_{ijk} \\
& \Big[\frac{1+F_{k+1/2}^2(\overline{D_{ij}}^{\dot{x}^2}+\overline{E_{ij}}^{\dot{y}^2})}{C_{k+1/2}G_{ij}}(\hat{p}_{ijk+1}-\hat{p}_{ijk}) \\
& -\frac{1+F_{k-1/2}^2(\overline{D_{ij}}^{\dot{x}^2}+\overline{E_{ij}}^{\dot{y}^2})}{C_{k-1/2}G_{ij}}(\hat{p}_{ijk}-\hat{p}_{ijk-1}) \\
& -F_{k+1/2}\frac{\overline{D_{ij}}^{\dot{x}}}{\overline{A_i}^{\dot{x}}}0.25(\hat{p}_{i+1\,jk+1}+\hat{p}_{i+1\,jk}-\hat{p}_{i-1\,jk+1}-\hat{p}_{i-1\,jk}) \\
& -F_{k+1/2}\frac{\overline{E_{ij}}^{\dot{y}}}{\overline{B_i}^{\dot{y}}}0.25(\hat{p}_{ij+1\,k+1}+\hat{p}_{ij+1\,k}-\hat{p}_{ij-1\,k+1}-\hat{p}_{ij-1\,k}) \\
& +F_{k-1/2}\frac{\overline{D_{ij}}^{\dot{x}}}{\overline{A_i}^{\dot{x}}}0.25(\hat{p}_{i+1\,jk-1}+\hat{p}_{i+1\,jk}-\hat{p}_{i-1\,jk-1}-\hat{p}_{i-1\,jk}) \\
& +F_{k-1/2}\frac{\overline{E_{ij}}^{\dot{y}}}{\overline{B_i}^{\dot{y}}}0.25(\hat{p}_{ij+1\,k-1}+\hat{p}_{ij+1\,k}-\hat{p}_{ij-1\,k-1}-\hat{p}_{ij-1\,k})
\end{aligned} \tag{A.3}$$

The matrix coefficients can be taken directly from the equations (A.1)-(A.3). With the abbreviation

$$s_k = \frac{\Delta z_k}{\overline{C_k}^{\dot{z}}}$$

this leads to the following coefficients for grid cell (ijk):

variable	coefficient
$\hat{p}_{ijk}$	$-\overline{G_{i+1/2\,j}}^{\bullet\dot{x}}\Delta y_j\Delta z_k \,/\, A_{i+1/2}$
	$-\overline{G_{i-1/2\,j}}^{\bullet\dot{x}}\Delta y_j\Delta z_k \,/\, A_{i-1/2}$
	$-\overline{G_{ij+1/2}}^{\bullet\dot{y}}\Delta x_i\Delta z_k \,/\, B_{j+1/2}$
	$-\overline{G_{ij-1/2}}^{\bullet\dot{y}}\Delta x_i\Delta z_k \,/\, B_{j-1/2}$
	$-\Delta x_i\Delta y_j(1+F_{k+1/2}^2(\overline{D_{ij}}^{\dot{x}^2}+\overline{E_{ij}}^{\dot{y}^2}))\,/\,(C_{k+1/2}G_{ij})$
	$-\Delta x_i\Delta y_j(1+F_{k-1/2}^2(\overline{D_{ij}}^{\dot{x}^2}+\overline{E_{ij}}^{\dot{y}^2}))\,/\,(C_{k-1/2}G_{ij})$
$\hat{p}_{i+1jk}$	$\Delta y_j\Delta z_k\overline{G_{i+1/2\,j}}^{\bullet\dot{x}} \,/\, A_{i+1/2}$
	$-\,0.25\,\Delta y_j\overline{D_{ij}}^{\dot{x}}(F_{k+1/2}-F_{k-1/2})$
$\hat{p}_{i-1jk}$	$\Delta y_j\Delta z_k\overline{G_{i-1/2\,j}}^{\bullet\dot{x}} \,/\, A_{i-1/2}$
	$+\,0.25\,\Delta y_j\overline{D_{ij}}^{\dot{x}}(F_{k+1/2}-F_{k-1/2})$
$\hat{p}_{ij+1k}$	$\Delta x_i\Delta z_k\overline{G_{ij+1/2}}^{\bullet\dot{y}} \,/\, B_{j+1/2}$
	$-0.25\,\Delta x_i\overline{E_{ij}}^{\dot{y}}(F_{k+1/2}-F_{k-1/2})$
$\hat{p}_{ij-1k}$	$\Delta x_i\Delta z_k\overline{G_{ij-1/2}}^{\bullet\dot{y}} \,/\, B_{j-1/2}$
	$+\,0.25\,\Delta x_i\overline{E_{ij}}^{\dot{y}}(F_{k+1/2}-F_{k-1/2})$
$\hat{p}_{ijk+1}$	$\Delta x_i\Delta y_j(1+{F_{k+1/2}}^2(\overline{D_{ij}}^{\dot{x}^2}+\overline{E_{ij}}^{\dot{y}^2}))\,/\,(C_{k+1/2}G_{ij})$
	$-\,0.25\,\Delta y_j s_k\overline{F_k}^{\dot{z}}(D_{i+1/2\,j}-D_{i-1/2\,j})$
	$-\,0.25\,\Delta x_i s_k\overline{F_k}^{\dot{z}}(E_{ij+1/2}-E_{ij-1/2})$
$\hat{p}_{ijk-1}$	$\Delta x_i\Delta y_j(1+F_{k-1/2}^2(\overline{D_{ij}}^{\dot{x}^2}+\overline{E_{ij}}^{\dot{y}^2}))\,/\,(C_{k-1/2}G_{ij})$
	$+\,0.25\,\Delta y_j s_k\overline{F_k}^{\dot{z}}(D_{i+1/2\,j}-D_{i-1/2\,j})$
	$+\,0.25\,\Delta x_i s_k\overline{F_k}^{\dot{z}}(E_{ij+1/2}-E_{ij-1/2})$
$\hat{p}_{i+1jk+1}$	$-0.25\,\Delta y_j(s_k\overline{F_k}^{\dot{z}}D_{i+1/2\,j}+F_{k+1/2}\overline{D_{ij}}^{\dot{x}})$
$\hat{p}_{i-1jk+1}$	$+0.25\,\Delta y_j(s_k\overline{F_k}^{\dot{z}}D_{i-1/2\,j}+F_{k+1/2}\overline{D_{ij}}^{\dot{x}})$
$\hat{p}_{i+1jk-1}$	$+0.25\,\Delta y_j(s_k\overline{F_k}^{\dot{z}}D_{i+1/2\,j}+F_{k-1/2}\overline{D_{ij}}^{\dot{x}})$
$\hat{p}_{i-1jk-1}$	$-0.25\,\Delta y_j(s_k\overline{F_k}^{\dot{z}}D_{i-1/2\,j}+F_{k-1/2}\overline{D_{ij}}^{\dot{x}})$
$\hat{p}_{ij+1k+1}$	$-0.25\,\Delta x_i(s_k\overline{F_k}^{\dot{z}}E_{ij+1/2}+F_{k+1/2}\overline{E_{ij}}^{\dot{y}})$
$\hat{p}_{ij-1k+1}$	$+0.25\,\Delta x_i(s_k\overline{F_k}^{\dot{z}}E_{ij-1/2}+F_{k+1/2}\overline{E_{ij}}^{\dot{y}})$
$\hat{p}_{ij+1k-1}$	$+0.25\,\Delta x_i(s_k\overline{F_k}^{\dot{z}}E_{ij+1/2}+F_{k-1/2}\overline{E_{ij}}^{\dot{y}})$
$\hat{p}_{ij-1k-1}$	$-0.25\,\Delta x_i(s_k\overline{F_k}^{\dot{z}}E_{ij-1/2}+F_{k-1/2}\overline{E_{ij}}^{\dot{y}})$

A.2 Matrix coefficients - boundary between two grids

When multiple grids are used, the coefficients change for the elements at the internal boundaries. In this section capital indices I and J are just for the coarse grid x- and y-direction, lower case indices are used for the vertical and the fine grid. Furthermore, the coarse grid pressure perturbation is denoted by capital $\hat{P}$, the fine grid pressure perturbation by lower case $\hat{p}$.

The calculation of the divergence in x-direction in the coarse grid cell $I = R$ caused by the pressure gradient starts from Eq. (6.7). Here, $(\overline{\rho_0}^{\dot{x}}\hat{u})_{r-0.5\,jk}$ is replaced by $\mathcal{P}_{r-0.5\,jk}$ from Eq. (6.9), $(\overline{\rho_0}^{\dot{x}}\hat{u})_{r+0.5\,jk}$ by $\mathcal{P}_{i+0.5jk}$ from Eq. (6.4) and $(\overline{\rho_0}^{\dot{x}}\hat{U})_{R-0.5\,Jk}$ by $\mathcal{P}_{R-0.5\,Jk}$ from Eq. (6.4). This yields:

$$
\begin{aligned}
\left(\frac{\partial}{\partial x}[\rho_0\alpha^*\hat{U}]\right)_{RJk} &= \Delta y_J \Delta z_k \\
\Big\{ &\psi_c\psi_u \frac{1}{A_{r-0.5}}\frac{1}{N_y}\sum_{j=j_1}^{j_2}\overline{G_{r-0.5\,j}}^{\dot{x}}\left(\hat{p}_{rjk} - \overline{\overline{\hat{P}_{Rjk}}}^{\dot{y}}\right) \\
&-\psi_c\frac{\overline{F_k}^{\dot{z}}}{\overline{C_k}^{\dot{z}}}0.25\frac{1}{N_y}\sum_{j=j_1}^{j_2} D_{r-0.5\,j}\,(\hat{p}_{rjk+1} - \hat{p}_{rjk-1} \\
&\qquad + \overline{\overline{\hat{P}_{RJk+1}}}^{\dot{y}} - \overline{\overline{\hat{P}_{RJk-1}}}^{\dot{y}}\Big) \\
&+\frac{\psi_f}{A_{r+0.5}}\frac{1}{N_y}\sum_{j=j_1}^{j_2}\left(\overline{G_{r+0.5\,j}}^{\dot{x}}(\hat{p}_{r+1\,jk} - \hat{p}_{rjk})\right) \\
&-\psi_f 0.25\frac{\overline{F_k}^{\dot{z}}}{\overline{C_k}^{\dot{z}}}\frac{1}{N_y}\sum_{j=j_1}^{j_2}[D_{r+0.5\,j}(\hat{p}_{r+1\,jk+1} + \hat{p}_{rjk+1} \\
&\qquad - \hat{p}_{r+1\,jk-1} - \hat{p}_{rjk-1})] \\
&-\frac{\overline{G_{R-0.5\,J}}^{\dot{x}}}{A_{R-0.5}}(\hat{P}_{RJk} - \hat{P}_{R-1\,Jk}) \\
&+\frac{\overline{F_k}^{\dot{z}} D_{R-0.5\,J}}{\overline{C_k}^{\dot{z}}}0.25(\hat{P}_{R-1\,Jk+1} + \hat{P}_{RJk+1} \\
&\qquad - \hat{P}_{R-1\,Jk-1} - \hat{P}_{RJk-1})\Big\}
\end{aligned}
$$

In the equation above the blue colours denote pressure values with additional coefficients in comparison to Eq. (A.1) which is still valid for non-boundary grid points.

Red colours denote dependencies on the neighbouring fine grid.

For the fine grid at the boundary (cell $i = r$) the procedure is similar to the one described for the coarse grid. The derivation starts from Eq. (6.8). $\mathcal{P}_{r-0.5jk}$ from Eq. (6.9) is inserted for $(\overline{\rho_0}^{\dot{x}}\hat{u})_{r-0.5}$, and replacing $(\overline{\rho_0}^{\dot{x}}\hat{u})_{r+0.5ij}$ by $\mathcal{P}_{r+0.5ij}$ from Eq. (6.4) yields:

$$
\begin{aligned}
\left(\frac{\partial}{\partial x}[\rho_0\alpha^*\hat{u}]\right)_{rjk} = \quad & \Delta y_J \Delta z_k \psi_c \\
\Bigg\{ \quad & + \frac{\overline{G_{r+0.5\,j}}^{\dot{x}\bullet}}{A_{r+0.5}}(\hat{p}_{r+1jk} - \hat{p}_{r\,jk}) \\
& - \frac{\overline{F_k}^{\dot{z}} D_{r+0.5\,j}}{\overline{C_k}^{\dot{z}}} 0.25(\hat{p}_{r+1\,jk+1} + \hat{p}_{rjk+1} - \hat{p}_{r+1\,jk-1} - \hat{p}_{rjk-1}) \\
& - \psi_u \frac{\overline{G_{r-0.5\,j}}^{\dot{x}\bullet}}{A_{r-0.5}} \left(\hat{p}_{rjk} - \overline{\overline{\hat{P}_{Rjk}}}^{\dot{y}}\right) \\
& + \frac{\overline{F_k}^{\dot{z}} D_{r-0.5\,j}}{\overline{C_k}^{\dot{z}}} 0.25 \left(\hat{p}_{rjk+1} - \hat{p}_{rjk-1} + \overline{\overline{\hat{P}_{Rjk+1}}}^{\dot{y}} - \overline{\overline{\hat{P}_{Rjk-1}}}^{\dot{y}}\right) \\
\Bigg\} &
\end{aligned}
$$

Again, blue colours denote pressure values with additional coefficients in comparison to Eq. (A.1) and red colours dependencies on the neighbouring coarse grid.

For the divergence in the coarse grid cell R caused by the pressure gradient in z-direction term (6.3c) is taken and $\dot{\mathcal{P}}_{RJk\pm0.5}$ from Eq. (6.15) is inserted for $(\overline{\rho_0}^{\dot{z}}\dot{\hat{W}})_{RJk\pm0.5}$:

$$
\begin{aligned}
\left(\frac{\partial}{\partial \dot{z}} \quad [\rho_0 \alpha^* \hat{\dot{W}}]\right)_{RJk} &= \Delta x_R \Delta y_J \\
&\left\{ \frac{1 + F_{k+0.5}^2 (\overline{D_{RJ}}^{\dot{x}2} + \overline{E_{RJ}}^{\dot{y}2})}{C_{k+0.5} G_{RJ}} (\hat{P}_{RJk+1} - \hat{P}_{RJk}) \right. \\
&- \frac{1 + F_{k-0.5}^2 (\overline{D_{RJ}}^{\dot{x}2} + \overline{E_{RJ}}^{\dot{y}2})}{C_{k-0.5} G_{RJ}} (\hat{P}_{RJk} - \hat{P}_{RJk-1}) \\
&- F_{k+0.5} \frac{\overline{E_{RJ}}^{\dot{y}}}{\overline{B_R}^{\dot{y}}} 0.25 (\hat{P}_{RJ+1\,k+1} + \hat{P}_{RJ+1\,k} - \hat{P}_{RJ-1\,k+1} - \hat{P}_{RJ-1\,k}) \\
&+ F_{k-0.5} \frac{\overline{E_{RJ}}^{\dot{y}}}{\overline{B_R}^{\dot{y}}} 0.25 (\hat{P}_{RJ+1\,k-1} + \hat{P}_{RJ+1\,k} - \hat{P}_{RJ-1\,k-1} - \hat{P}_{RJ-1\,k}) \\
&- F_{k+0.5} \frac{\overline{D_{RJ}}^{\dot{x}}}{\overline{A_R}^{\dot{x}}} \psi_W \left[\frac{1}{N_y} \sum_{j=j_1}^{j_2} (\hat{p}_{rjk} + \hat{p}_{rjk+1}) - \hat{P}_{R-1\,Jk} - \hat{P}_{R-1\,Jk+1} \right] \\
&\left. + F_{k-0.5} \frac{\overline{D_{RJ}}^{\dot{x}}}{\overline{A_R}^{\dot{x}}} \psi_W \left[\frac{1}{N_y} \sum_{j=j_1}^{j_2} (\hat{p}_{rjk} + \hat{p}_{rjk-1}) - \hat{P}_{R-1\,Jk} - \hat{P}_{R-1\,Jk-1} \right] \right\}
\end{aligned}
$$

Blue colours denote pressure values with additional coefficients in comparison to Eq. (A.3) and red colours dependencies on the neighbouring fine grid.

For the divergence in the fine grid cell r caused by the pressure gradient in z-direction term (6.3c) is taken and $\dot{\mathcal{P}}_{rjk\pm0.5}$ from Eq. (6.16) is inserted for $(\overline{\rho_0}^{\dot{z}} \hat{\dot{W}}_{RJk\pm0.5})$:

$$
\begin{aligned}
\left(\frac{\partial}{\partial \dot{z}} \quad [\rho_0 \alpha^* \hat{\dot{w}}]\right)_{rjk} &= \Delta x_r \Delta y_k \\
&\left\{ \frac{1 + F_{k+0.5}^2 (\overline{D_{rj}}^{\dot{x}2} + \overline{E_{rj}}^{\dot{y}2})}{C_{k+0.5} G_{rj}} (\hat{p}_{rjk+1} - \hat{p}_{rjk}) \right. \\
&- \frac{1 + F_{k-0.5}^2 (\overline{D_{rj}}^{\dot{x}2} + \overline{E_{rj}}^{\dot{y}2})}{C_{k-0.5} G_{rj}} (\hat{p}_{rjk} - \hat{p}_{rjk-1}) \\
&- F_{k+0.5} \frac{\overline{E_{rj}}^{\dot{y}}}{\overline{B_r}^{\dot{y}}} 0.25 (\hat{p}_{rj+1\,k+1} + \hat{p}_{rj+1\,k} - \hat{p}_{rj-1\,k+1} - \hat{p}_{rj-1\,k}) \\
&+ F_{k-0.5} \frac{\overline{E_{rj}}^{\dot{y}}}{\overline{B_r}^{\dot{y}}} 0.25 (\hat{p}_{rj+1\,k-1} + \hat{p}_{rj+1\,k} - \hat{p}_{rj-1\,k-1} - \hat{p}_{rj-1\,k}) \\
&- F_{k+0.5} \frac{\overline{D_{rj}}^{\dot{x}}}{\overline{A_r}^{\dot{x}}} \psi_w \left(\hat{p}_{r+1\,jk} + \hat{p}_{r+1\,jk+1} - \overline{\overline{\hat{P}_{Rjk}}}^{\dot{y}} - \overline{\overline{\hat{P}_{Rjk+1}}}^{\dot{y}} \right) \\
&\left. + F_{k-0.5} \frac{\overline{D_{rj}}^{\dot{x}}}{\overline{A_r}^{\dot{x}}} \psi_w \left(\hat{p}_{r+1\,jk} + \hat{p}_{r+1\,jk-1} - \overline{\overline{\hat{P}_{Rjk}}}^{\dot{y}} - \overline{\overline{\hat{P}_{Rjk-1}}}^{\dot{y}} \right) \right\}
\end{aligned}
$$

Again, blue colours denote pressure values with additional coefficients in comparison to Eq. (A.3) and red colours dependencies on the neighbouring coarse grid.

The matrix coefficients can be taken directly from the equations for the divergence caused by the pressure gradients. Changes in comparison to matrix elements belonging to the inner grid are denoted with blue colour. The red colour indicates, that there is no dependency anymore (0) due to the replacement by the neighbouring grid. Note that factors a_{1j}, a_{2j} and a_{3j} have their origin from the parabolic interpolation in y-direction ($\overline{\overline{\cdot}}^{y}$). The factors are calculated as indicated in Eq. (6.11)-(6.13).

A.2.1 Elements of coarse grid

The coefficients of the coarse grid internal boundary cell (RJk) (Fig. 6.2) for variables belonging to the coarse grid pressure field variables are

variable	coefficient
$\hat{P}_{RJk}$	$-\psi_c\psi_u\frac{1}{N_y}\sum_{j=j_1}^{j_2} a_{2j}\overline{G_{r-1/2\,j}}^{\dot{x}}\Delta y_J\Delta z_k \,/\, A_{r-1/2}$
	$-\overline{G_{R-1/2\,J}}^{\dot{x}}\Delta y_J\Delta z_k \,/\, A_{R-1/2}$
	$-\overline{G_{RJ+1/2}}^{\dot{y}}\Delta x_R\Delta z_k \,/\, B_{J+1/2} \; -\overline{G_{RJ-1/2}}^{\dot{y}}\Delta x_R\Delta z_k \,/\, B_{J-1/2}$
	$-\Delta x_R\Delta y_J(1+F_{k+1/2}^2(\overline{D_{RJ}}^{\dot{x}^2}+\overline{E_{RJ}}^{\dot{y}^2}))\,/\,(C_{k+1/2}G_{RJ})$
	$-\Delta x_R\Delta y_J(1+F_{k-1/2}^2(\overline{D_{RJ}}^{\dot{x}^2}+\overline{E_{RJ}}^{\dot{y}^2}))\,/\,(C_{k-1/2}G_{RJ})$
$\hat{P}_{R+1Jk}$	0
$\hat{P}_{R-1Jk}$	$\Delta y_J\Delta z_k\overline{G_{R-1/2\,J}}^{\dot{x}} \,/\, A_{R-1/2}$
	$+\,\Delta y_J\overline{D_{RJ}}^{\dot{x}}\psi_W(F_{k+1/2}-F_{k-1/2})$
$\hat{P}_{RJ+1k}$	$\Delta x_R\Delta z_k\overline{G_{RJ+1/2}}^{\dot{y}} \,/\, B_{J+1/2}$
	$-0.25\;\Delta x_R\overline{E_{RJ}}^{\dot{y}}(F_{k+1/2}-F_{k-1/2})$
	$-\psi_c\psi_u\frac{1}{N_y}\sum_{j=j_1}^{j_2} a_{3j}\overline{G_{r-1/2\,j}}^{\dot{x}}\Delta y_J\Delta z_k \,/\, A_{r-1/2}$
$\hat{P}_{RJ-1k}$	$\Delta x_R\Delta z_k\overline{G_{RJ-1/2}}^{\dot{y}} \,/\, B_{J-1/2}$
	$+\,0.25\;\Delta x_R\overline{E_{RJ}}^{\dot{y}}(F_{k+1/2}-F_{k-1/2})$
	$-\psi_c\psi_u\frac{1}{N_y}\sum_{j=j_1}^{j_2} a_{1j}\overline{G_{r-1/2\,j}}^{\dot{x}}\Delta y_J\Delta z_k \,/\, A_{r-1/2}$
$\hat{P}_{RJk+1}$	$\Delta x_R\Delta y_J(1+F_{k+1/2}{}^2(\overline{D_{RJ}}^{\dot{x}^2}+\overline{E_{RJ}}^{\dot{y}^2}))\,/\,(C_{k+1/2}G_{RJ})$
	$-\,0.25\;\Delta y_J s_k\overline{F_k}^{\dot{z}}(\frac{\psi_c}{N_y}\sum_{j=j_1}^{j_2} a_{2j}D_{r-1/2\,j}-D_{R-1/2\,J})$
	$-\,0.25\;\Delta x_R s_k\overline{F_k}^{\dot{z}}(E_{RJ+1/2}-E_{RJ-1/2})$
$\hat{P}_{RJk-1}$	$\Delta x_R\Delta y_J(1+F_{k-1/2}^2(\overline{D_{RJ}}^{\dot{x}^2}+\overline{E_{RJ}}^{\dot{y}^2}))\,/\,(C_{k-1/2}G_{RJ})$
	$+\,0.25\;\Delta y_J s_k\overline{F_k}^{\dot{z}}(\frac{\psi_c}{N_y}\sum_{j=j_1}^{j_2} a_{2j}D_{r-1/2\,j}-D_{R-1/2\,J})$
	$+\,0.25\;\Delta x_R s_k\overline{F_k}^{\dot{z}}(E_{RJ+1/2}-E_{RJ-1/2})$
$\hat{P}_{R+1Jk+1}$	0
$\hat{P}_{R-1Jk+1}$	$+0.25\;\Delta y_J(s_k\overline{F_k}^{\dot{z}}D_{R-1/2\,J})+\Psi_W\Delta y_J F_{k+1/2}\overline{D_{RJ}}^{\dot{x}}$
$\hat{P}_{R+1Jk-1}$	0
$\hat{P}_{R-1Jk-1}$	$-0.25\;\Delta y_J(s_k\overline{F_k}^{\dot{z}}D_{R-1/2\,J})-\Psi_W\Delta y_J F_{k-1/2}\overline{D_{RJ}}^{\dot{x}}$

variable	coefficient
$\hat{P}_{RJ+1k+1}$	$-0.25\ \Delta x_R(s_k\overline{F_k}^{\dot{z}}E_{RJ+1/2} + F_{k+1/2}\overline{E_{RJ}}^{\dot{y}})$
	$-\ 0.25\ \Delta y_J s_k\overline{F_k}^{\dot{z}}\frac{\psi_c}{N_y}\sum_{j=j_1}^{j_2} a_{3j}D_{r-1/2\,j}$
$\hat{P}_{RJ-1k+1}$	$+0.25\ \Delta x_R(s_k\overline{F_k}^{\dot{z}}E_{RJ-1/2} + F_{k+1/2}\overline{E_{RJ}}^{\dot{y}})$
	$-\ 0.25\ \Delta y_J s_k\overline{F_k}^{\dot{z}}\frac{\psi_c}{N_y}\sum_{j=j_1}^{j_2} a_{1j}D_{r-1/2\,j}$
$\hat{P}_{RJ+1k-1}$	$+0.25\ \Delta x_R(s_k\overline{F_k}^{\dot{z}}E_{RJ+1/2} + F_{k-1/2}\overline{E_{RJ}}^{\dot{y}})$
	$+\ 0.25\ \Delta y_J s_k\overline{F_k}^{\dot{z}}\frac{\psi_c}{N_y}\sum_{j=j_1}^{j_2} a_{3j}D_{r-1/2\,j}$
$\hat{P}_{RJ-1k-1}$	$-0.25\ \Delta x_R(s_k\overline{F_k}^{\dot{z}}E_{RJ-1/2} + F_{k-1/2}\overline{E_{RJ}}^{\dot{y}})$
	$+\ 0.25\ \Delta y_J s_k\overline{F_k}^{\dot{z}}\frac{\psi_c}{N_y}\sum_{j=j_1}^{j_2} a_{1j}D_{r-1/2\,j}$

N_y fine grid indices $j = j_1, ..., j_2$ correspond to the coarse y-index J. The associated fine grid pressure variables are part of the equation for grid cell (IJk). The coefficients the these fine grid pressure variables are:

variable	coefficient
$\hat{p}_{rjk}$	$+\psi_c\psi_u\frac{1}{N_y}\overline{G_{r-1/2\,j}}^{\bullet\dot{x}}\Delta y_J\Delta z_k\ /\ A_{r-1/2}$
	$-\ \psi_f\frac{1}{N_y}\overline{G_{r+1/2\,j}}^{\bullet\dot{x}}\Delta y_J\Delta z_k\ /\ A_{r+1/2}$
	$-\ \frac{1}{N_y}\Delta y_J\overline{D_{RJ}}^{\dot{x}}\psi_W(F_{k+1/2} - F_{k-1/2})$
$\hat{p}_{r+1jk}$	$+\psi_f\frac{1}{N_y}\overline{G_{r+1/2\,j}}^{\bullet\dot{x}}\Delta y_J\Delta z_k\ /\ A_{r+1/2}$
$\hat{p}_{rjk+1}$	$-0.25\ \Delta y_J s_k\overline{F_k}^{\dot{z}}\frac{\psi_c}{N_y}D_{r-1/2\,j}$
	$-0.25\ \Delta y_J s_k\overline{F_k}^{\dot{z}}\frac{\psi_f}{N_y}D_{r+1/2\,j}$
	$-\ \frac{1}{N_y}\Delta y_J\overline{D_{RJ}}^{\dot{x}}\psi_W F_{k+1/2}$
$\hat{p}_{rjk-1}$	$+0.25\ \Delta y_J s_k\overline{F_k}^{\dot{z}}\frac{\psi_c}{N_y}D_{r-1/2\,j}$
	$+0.25\ \Delta y_J s_k\overline{F_k}^{\dot{z}}\frac{\psi_f}{N_y}D_{r+1/2\,j}$
	$+\ \frac{1}{N_y}\Delta y_J\overline{D_{RJ}}^{\dot{x}}\psi_W F_{k-1/2}$
$\hat{p}_{r+1k+1}$	$-0.25\ \Delta y_J s_k\overline{F_k}^{\dot{z}}\frac{\psi_f}{N_y}D_{r+1/2\,j}$
$\hat{p}_{r+1jk-1}$	$+0.25\ \Delta y_J s_k\overline{F_k}^{\dot{z}}\frac{\psi_f}{N_y}D_{r+1/2\,j}$

A.2.2 Elements of fine grid

The coefficients of the fine grid internal boundary cell (rjk) (Fig. 6.2) belonging to the fine grid pressure field variable are

variable	coefficient
$\hat{p}_{rjk}$	$-\psi_c \overline{G_{r+1/2\,j}}^{\dot{x}} \Delta y_j \Delta z_k \,/\, A_{r+1/2} \; - \psi_c \psi_u \overline{G_{r-1/2\,j}}^{\dot{x}} \Delta y_j \Delta z_k \,/\, A_{r-1/2}$
	$-\overline{G_{rj+1/2}}^{\dot{y}} \Delta x_r \Delta z_k \,/\, B_{j+1/2} \; - \overline{G_{rj-1/2}}^{\dot{y}} \Delta x_r \Delta z_k \,/\, B_{j-1/2}$
	$-\Delta x_r \Delta y_j (1 + F_{k+1/2}^2 (\overline{D_{rj}}^{\dot{x}^2} + \overline{E_{rj}}^{\dot{y}^2})) \,/\, (C_{k+1/2} G_{rj})$
	$-\Delta x_r \Delta y_j (1 + F_{k-1/2}^2 (\overline{D_{rj}}^{\dot{x}^2} + \overline{E_{rj}}^{\dot{y}^2})) \,/\, (C_{k-1/2} G_{rj})$
$\hat{p}_{r+1jk}$	$\psi_c \Delta y_j \Delta z_k \overline{G_{r+1/2\,j}}^{\dot{x}} \,/\, A_{r+1/2}$
	$-\psi_w \; \Delta y_j \overline{D_{rj}}^{\dot{x}} (F_{k+1/2} - F_{k-1/2})$
$\hat{p}_{r-1jk}$	0
$\hat{p}_{rj+1k}$	$\Delta x_r \Delta z_k \overline{G_{rj+1/2}}^{\dot{y}} \,/\, B_{j+1/2}$
	$-0.25 \; \Delta x_r \overline{E_{rj}}^{\dot{y}} (F_{k+1/2} - F_{k-1/2})$
$\hat{p}_{rj-1k}$	$\Delta x_r \Delta z_k \overline{G_{rj-1/2}}^{\dot{y}} \,/\, B_{j-1/2}$
	$+ 0.25 \; \Delta x_r \overline{E_{rj}}^{\dot{y}} (F_{k+1/2} - F_{k-1/2})$
$\hat{p}_{rjk+1}$	$\Delta x_r \Delta y_j (1 + {F_{k+1/2}}^2 (\overline{D_{rj}}^{\dot{x}^2} + \overline{E_{rj}}^{\dot{y}^2})) \,/\, (C_{k+1/2} G_{rj})$
	$- \; 0.25 \; \Delta y_j \psi_c s_k \overline{F_k}^{\dot{z}} (D_{r+1/2\,j} - D_{r-1/2\,j})$
	$- \; 0.25 \; \Delta x_r s_k \overline{F_k}^{\dot{z}} (E_{rj+1/2} - E_{rj-1/2})$
$\hat{p}_{rjk-1}$	$\Delta x_r \Delta y_j (1 + F_{k-1/2}^2 (\overline{D_{rj}}^{\dot{x}^2} + \overline{E_{rj}}^{\dot{y}^2})) \,/\, (C_{k-1/2} G_{rj})$
	$+ \; 0.25 \; \Delta y_j \psi_c s_k \overline{F_k}^{\dot{z}} (D_{r+1/2\,j} - D_{r-1/2\,j})$
	$+ \; 0.25 \; \Delta x_r s_k \overline{F_k}^{\dot{z}} (E_{rj+1/2} - E_{rj-1/2})$
$\hat{p}_{r+1jk+1}$	$-0.25 \; \Delta y_j (\psi_c s_k \overline{F_k}^{\dot{z}} D_{r+1/2\,j}) - \psi_w \Delta y_j F_{k+1/2} \overline{D_{rj}}^{\dot{x}}$
$\hat{p}_{r-1jk+1}$	0
$\hat{p}_{r+1jk-1}$	$+0.25 \; \Delta y_j (\psi_c s_k \overline{F_k}^{\dot{z}} D_{r+1/2\,j}) + \psi_w \Delta y_j F_{k-1/2} \overline{D_{rj}}^{\dot{x}}$
$\hat{p}_{r-1jk-1}$	0
$\hat{p}_{rj+1k+1}$	$-0.25 \; \Delta x_r (s_k \overline{F_k}^{\dot{z}} E_{rj+1/2} + F_{k+1/2} \overline{E_{rj}}^{\dot{y}})$
$\hat{p}_{rj-1k+1}$	$+0.25 \; \Delta x_r (s_k \overline{F_k}^{\dot{z}} E_{rj-1/2} + F_{k+1/2} \overline{E_{rj}}^{\dot{y}})$
$\hat{p}_{rj+1k-1}$	$+0.25 \; \Delta x_r (s_k \overline{F_k}^{\dot{z}} E_{rj+1/2} + F_{k-1/2} \overline{E_{rj}}^{\dot{y}})$
$\hat{p}_{rj-1k-1}$	$-0.25 \; \Delta x_r (s_k \overline{F_k}^{\dot{z}} E_{rj-1/2} + F_{k-1/2} \overline{E_{rj}}^{\dot{y}})$

The coefficients for the coarse grid pressure field variables in the pressure equation

for the fine grid cell (ijk) are:

variable	coefficient	
$\hat{P}_{RJk}$	$+a_{2j}\big[$	$\psi_u\psi_c\overline{G_{r-1/2\,j}}^{\bullet\dot{x}}\Delta y_j\Delta z_k \,/\, A_{r-1/2}$
		$+\psi_w\ \Delta y_j\overline{D_{rj}}^{\dot{x}}(F_{k+1/2}-F_{k-1/2})\big]$
$\hat{P}_{RJk+1}$	$+a_{2j}\big[$	$\Delta y_j 0.25\psi_c s_k\overline{F_k}^{\dot{z}}D_{r-1/2\,j}$
		$+\psi_w\ \Delta y_j\overline{D_{rj}}^{\dot{x}}F_{k+1/2}\big]$
$\hat{P}_{RJk+1}$	$-a_{2j}\big[$	$\Delta y_j 0.25\psi_c s_k\overline{F_k}^{\dot{z}}D_{r-1/2\,j}$
		$+\psi_w\ \Delta y_j\overline{D_{rj}}^{\dot{x}}F_{k-1/2}\big]$
$\hat{P}_{RJ-1k}$	$+a_{1j}\big[$	$\psi_u\psi_c\overline{G_{r-1/2\,j}}^{\bullet\dot{x}}\Delta y_j\Delta z_k \,/\, A_{r-1/2}$
		$+\psi_w\ \Delta y_j\overline{D_{rj}}^{\dot{x}}(F_{k+1/2}-F_{k-1/2})\big]$
$\hat{P}_{RJ-1k+1}$	$+a_{1j}\big[$	$\Delta y_j 0.25\psi_c s_k\overline{F_k}^{\dot{z}}D_{r-1/2\,j}$
		$+\psi_w\ \Delta y_j\overline{D_{rj}}^{\dot{x}}F_{k+1/2}\big]$
$\hat{P}_{RJ-1k+1}$	$-a_{1j}\big[$	$\Delta y_j 0.25\psi_c s_k\overline{F_k}^{\dot{z}}D_{r-1/2\,j}$
		$+\psi_w\ \Delta y_j\overline{D_{rj}}^{\dot{x}}F_{k-1/2}\big]$
$\hat{P}_{RJ+1k}$	$+a_{3j}\big[$	$\psi_u\psi_c\overline{G_{r-1/2\,j}}^{\bullet\dot{x}}\Delta y_j\Delta z_k \,/\, A_{r-1/2}$
		$+\psi_w\ \Delta y_j\overline{D_{rj}}^{\dot{x}}(F_{k+1/2}-F_{k-1/2})\big]$
$\hat{P}_{RJ+1k+1}$	$+a_{3j}\big[$	$\Delta y_j 0.25\psi_c s_k\overline{F_k}^{\dot{z}}D_{r-1/2\,j}$
		$+\psi_w\ \Delta y_j\overline{D_{rj}}^{\dot{x}}F_{k+1/2}\big]$
$\hat{P}_{RJ+1k+1}$	$-a_{3j}\big[$	$\Delta y_j 0.25\psi_c s_k\overline{F_k}^{\dot{z}}D_{r-1/2\,j}$
		$+\psi_w\ \Delta y_j\overline{D_{rj}}^{\dot{x}}F_{k-1/2}\big]$

B Solvers for the elliptic equation

This appendix describes iterative solvers for elliptic equations like Eq. (2.19). When equations of this type are discretised this generally leads to a systems of equations of the form

$$Ax = b \tag{B.1}$$

with the coefficient matrix $A \in \mathbb{R}^{n \times n}$, the right-hand-side $b \in \mathbb{R}^n$ and the solution vector $x \in \mathbb{R}^n$. Note that x should not be confused with the Cartesian coordinate. x corresponds to the pressure vector $\hat{p}$, A to the discretised operator $\vec{\nabla} \bullet \vec{\mathcal{P}}$ and b to the momentum divergence. Since for the pressure equation the system of equations has as many unknowns and equations as there are grid points ($= n$ in the example above), its direct solution is not feasible. Therefore, approximate iterative solvers are needed. This appendix describes two types of methods, namely Krylov subspace methods (Section B.1) and MULTIGRID methods (Section B.4). Additionally, the preconditioning of matrix A is described. Preconditioning is a standard method to speed-up the convergence (Section B.2). The triangular solve of preconditioners on vector computers is explained in Section B.3.

B.1 Krylov subspace methods

Krylov subspace methods were developed to calculate iteratively the solution $x \in \mathbb{R}^n$ of Eq. (B.1). For a symmetric and positive definite matrix A the method of the conjugate gradients (CG) is very efficient. They were developed by Hestenes and Stiefel (1952). It is based on defining an appropriate (Krylov) subspace of $\mathbb{R}^n$ to which a projection of x is calculated. In each iteration step the subspace is extended by one dimension by demanding the residual of each iteration step to be perpendicular to the Krylov subspace.

As the pressure matrix A of METRAS is not symmetric, CG are expected to fail in many cases even for a single grid. The asymmetry of A is caused by the surface following coordinate transformation and by the nonuniform grid. The latter is commonly applied at least in the vertical. Furthermore, if more than one grid is used, coupling terms lead to additional non-symmetry.

For non-symmetric matrices A methods based on biorthogonalisation were derived. These methods can also be interpreted as projections of the system of equations to a lower dimensional subspace, which is extended with each iteration step. In METRAS Bi-CGSTAB (Van der Vorst, 1992) is implemented.

The CG methods are described briefly in the following (Section B.1.1). Bi-CG (Fletcher, 1975, Section B.1.2) is also explained as a step towards Bi-CGSTAB (Section B.1.3). Details can be found in various text books (e.g. Hackbusch, 1993; Deuflhard and Hohmann, 2003; Van der Vorst, 2003; Saad, 2003). A collection of algorithms is given by Barret et al. (1994).

B.1.1 Conjugate gradients

For the system (B.1) it is the aim of the conjugate gradients methods to find an appropriate subspace of $\mathbb{R}^n$ out of which a best approximation to x is calculated relative to a previously defined error norm. If A is symmetric and positive definite it can be used to define the inner product

$$< x, y >_A =< x, Ay > \tag{B.2}$$

with the scalar product $< x, y >= x^\tau y$. Again, x and $y \in \mathbb{R}^n$ are not Cartesian coordinates.

Let U_k be a k-dimensional vector subspace of $\mathbb{R}^n$, and V_k the parallel vector space $V_k = x_0 + U_k$, then $< x, y >_A$ can be used to calculate an A-orthogonal basis (p_i) of U_k (i.e. $< p_i, p_j >_A = \delta_{ij}$ where δ_{ij} is the Kronecker delta). With p_i a projection x_k (with respect to $< \cdot, \cdot >_A$) of the solution x to V_k can be calculated so that

$$< A(x - x_k), v - x_0 >=< r_k, v - x_0 > 0 \text{ for all } v \in V_k. \tag{B.3}$$

The residual r_k is orthogonal to the subspace U_k which means that $| < (x - x_k), A(x - x_k) > |$ cannot be decreased be adding any vector $y \in U_k$ to x_k.

To find a solution means to create a reasonable U_k. An efficient way is to expand

U_k by the orthogonal vector r_k to get U_{k+1}, so we can demand

$$U_k = \text{span}\{r_0, Ar_0, ..., A^{k-1}r_0\} \tag{B.4}$$

This is the Krylov subspace. This way we make sure to continuously expand the space of approximate solutions and that at least after n iterations we will end up with the exact solution. However, usually only a certain accuracy is needed, and the iteration stops after the error norm of the residual is less than a preciously defined maximum residual $r_{\max}$ (Section 6.3).

This leads immediately to the conjugate gradient (CG) iteration algorithm: If we start with an approximate solution x_0, the first A-basis vector is $p_1 = r_0 = A(x - x_0) = b - Ax_0$ (1., Fig. B.1). In line 2.1 and 2.2 the new projection of x is calculated, along with the new residual in 2.3. In 2.4 the residual is tested by an error norm e. This can be defined e.g. as the $||\cdot||_2$ norm. If the result is lower than the maximum allowed residual $r_{\max}$, the iteration is finished. Section 6.3 gives a more sophisticated solution how to define e and $r_{\max}$. If the residual is not small enough, a new vector p_{k+1} is calculated to expand the Krylov subspace (line 2.5 and 2.6).

B.1.2 Bi-CG

The conjugate gradients can also be derived by projecting the systems of equations. As seen in Section B.1.1 the residual vectors r_k of each iteration step are perpendicular to each other. They can be used to form the columns of the matrix $R_k = (r_1, r_2, ..., r_k)$. With $y = (R_k)^\tau x$ ($y \in \mathbb{R}^k$) the systems of equations can be

1. $p_1 = r_0 = A(x - x_0) = b - Ax_0$
2. for k=1, 2, ...
   - 2.1 $\alpha_k = \frac{<r_{k-1}, r_{k-1}>}{<p_k, p_k>_A}$
   - 2.2 $x_k = x_{k-1} + \alpha_k p_k$
   - 2.3 $r_k = r_{k-1} - \alpha_k A p_k$
   - 2.4 if ($e(r_k) < r_{\max}$) exit
   - 2.5 $\beta_{k+1} = \frac{<r_k, r_k>}{<r_k - 1, r_k - 1>}$
   - 2.6 $p_{k+1} = r_k + \beta_{k+1} p_k$

end

Figure B.1: Algorithm for the iterative solution of $Ax = b$ with conjugate gradients.

written as

$$R_k^\tau A R_k y = R_k^\tau b \tag{B.5}$$

It can be shown that $R_k^\tau A R_k$ is a tridiagonal matrix, which is the projection of the systems of equations to the Krylov subspace. The projected system can be used to derive the recurrence formula of the Section B.1.1.

Conjugate gradients are likely to fail for a non-symmetric A. In this case $R_k^\tau A R_k$ is not tridiagonal, and r_k is not perpendicular to the Krylov subspace. However, it can be shown that the vectors $\tilde{r}_k = P_k(A^\tau) r_0$ are perpendicular to the vectors $r_0, ..., r_{k-1}$ ($r_k = Q_k(A) r_0$), where P_k and Q_k are polynomials of degree k. This also means that the vectors r_k are perpendicular to the vectors $\tilde{r}_0, ..., \tilde{r}_{k-1}$ (biorthogonality). The corresponding matrices R_k and $\tilde{R}_k = (\tilde{r}_1, \tilde{r}_2, ..., \tilde{r}_k)$ can be used to calculate a projection of the systems of equations with

$$\tilde{R}_k^\tau A R_k y = \tilde{R}_k^\tau b. \tag{B.6}$$

Again, $\tilde{R}_k^\tau A R_k$ is tridiagonal and can be used to calculate a recurrence formula similar to that of CG. If for the r_k and $\tilde{r}_k$ the residuals of $r_k = Ax_k - b$ and $\tilde{r}_k = A^\tau \tilde{x}_k - \tilde{b}$ (with arbitrary $\tilde{b}$ or $\tilde{r}_0$) are taken, the resulting scheme is Bi-CG and can also be used for solving systems of equations with non-symmetric A. The Bi-CG algorithm is similar to the CG-algorithm (Fig. B.2). Note that the $\tilde{x}_k$ are not needed and therefore not calculated.

$1.a\ p_1 = r_0 = A(x - x_0) = b - Ax_0$
$1.b\ \tilde{p}_1 = \tilde{r}_0 = r_0$
2.for k=1, 2, ...
2.1 $\alpha_k = \frac{<\tilde{r}_{k-1}, r_{k-1}>}{<\tilde{p}_k, p_k>_A}$
2.2 $x_k = x_{k-1} + \alpha_k p_k$
2.3.a $r_k = r_{k-1} - \alpha_k A p_k$
2.3.b $\tilde{r}_k = \tilde{r}_{k-1} - \alpha_k A^\tau \tilde{p}_k$
2.4. if $(e(r_k) < r_{\max})$ exit
2.5. $\beta_{k+1} = \frac{<\tilde{r}_k, r_k>}{<\tilde{r}_k - 1, r_k - 1>}$
2.6.a $p_{k+1} = r_k + \beta_{k+1} p_k$
2.6.b $\tilde{p}_{k+1} = \tilde{r}_k + \beta_{k+1} \tilde{p}_k$
end

Figure B.2: Algorithm for the iterative solution of $Ax = b$ with Bi-CG.

B.1.3 Bi-CGSTAB

Bi-CG is disadvantageous as basically two systems of equations are solved at once, but only the solution of one system is needed. Furthermore, the convergence can be quite irregular if $\alpha_k \approx 0$ in step 2.1 of the algorithm (Fig. B.2). Bi-CGSTAB uses a different approach to circumvent this problem. In Bi-CG

$$< \tilde{r}_i, r_k >=< P_i(A^\tau)\tilde{r}_0, P_j(A)r_0 >= 0 \tag{B.7}$$

which can be rewritten to

$$< \tilde{r}_0, P_i(A)P_j(A)r_0 >= 0. \tag{B.8}$$

This shows that instead of creating $\tilde{r}_k$ and r_k it is also possible to create r_k in a different way based on

$$r_i = Q_i(A)P_i(A)r_0 \tag{B.9}$$

with the P_i as in Bi-CG and $Q_i(A) = (1 - \omega_i A)(1 - \omega_{i-1}A)\cdots(1 - \omega_1 A)$. It is possible to exploit the recurrences of Bi-CG for the calculation of the residual and the basis vectors of the Krylov subspace p_i and then minimise r_i with respect to w_i. The resulting scheme is Bi-CGSTAB (Fig. B.3).

B.2 Preconditioning

The Krylov subspace methods introduced in Section B.1 converge only slowly if the matrix is not well-conditioned, i.e. if

$$\kappa(A) = \frac{\lambda_{\max}}{\lambda_{\min}}$$

is very large where $\lambda_{\max}$ denotes the largest eigenvalue and $\lambda_{\min}$ the smallest eigenvalue. For this reason preconditioning is necessary, which means instead of solving $Ax = b$, $AK^{-1}y = b$ is solved with a matrix K that is easily invertible. As soon as a solution for y has been calculated, x is calculated by $x = K^{-1}y$. Details on different preconditioning techniques can be found in Van der Vorst (2003) and Saad (2003).

K is usually taken to be an approximate of A. In METRAS this is achieved by an incomplete LU factorisation of A. The idea is to create a lower triangular matrix L and an upper triangular matrix U so that $[(L + I)(U + I)D]_{i,j} = a_{i,j}$ for the non-zero elements $(i, j) \in \mathrm{NZ}(A) = \{(i, j)|a_{i,j} \neq 0\}$. Here I is the identity matrix.

The algorithm for calculating the elements of L and U can be interpreted as an incomplete Gaussian elimination (Fig. B.4).

This leads to $K = (L + I)(U + I)D \approx A$. In the algorithms for Krylov subspace methods where Ax is evaluated (Section B.1), $AK^{-1}y$ is evaluated instead. The inversion of K only demands inverting triangular matrices, which is can easily be achieved by a serial algorithm.

B.3 Triangular solve on vector computers

The direct solve of a general triangular system cannot be easily vectorised or parallelised. However, for sparse triangular matrices like the matrices used in METRAS there is a way to reorder the equations and create groups of equations that can be calculated independently. This is the so called wavefront ordering (e.g. Van der Vorst, 1989). If the basic matrix is ordered lexicographically, i.e. with i, j, k as

1.a $p_1 = r_0 = A(x - x_0) = b - Ax_0$
1.b $\tilde{r}_0 = r_0$
2.for k=1, 2, ...
2.1 $\rho_{k-1} =< \tilde{r}_0, r_{k-1} >$
2.2 if $k = 1$
$p_k = r_{k-1}$
else
$\beta_{k-1} = \frac{<\tilde{r}_0, r_{k-1}>}{<\tilde{r}_0, r_{k-2}>} \frac{\alpha_{k-1}}{\omega_{k-1}}$
$p_k = r_{k-1} + \beta_{k-1}(p_{k-1} - \omega A p_{k-1}$
endif
2.3 $\alpha_k = \frac{<\tilde{r}_0, r_{k-1}>}{<\tilde{r}_0, p_k>_A}$
2.4 $s = r_{k-1} - \alpha_k A p_k$
2.5 $\omega_k = \frac{<As, s>}{<As, As>}$
2.6 $x_k = x_{k-1} + \alpha_k p_k + \omega_k s$
2.7 $r_k = s - \omega_k As$
2.8 if $(e(r_k) < r_{\max})$ exit
end

Figure B.3: Algorithm for the iterative solution of $Ax = b$ with Bi-CGSTAB.

indices in (x, y, z)-direction and (N_{x1}, N_{x2}, N_{x3}) as number of grid points in (x, y, z)-direction, the $<$ ind $>th$ equation in lexicographical ordering is related to (i, j, k) by

$$< \text{ind} >= k + 1 + jN_{x3} + iN_{x3}N_{x2}$$

```
for i = 1, ..., n
    for j = 1, ..., n and (i,k) ∈ NZ(A)
        d_ij = a_ij for i = j
        d_ij = 0 for i ≠ j
        l_ij = a_ij for i > j
        l_ij = 0 for i <= j
        u_ij = a_ij for i < j
        u_ij = 0 for i >= j
    end
end
for i = 1, ..., n
    d_ii = 1/d_ii
    for k = 1, ..., i-1 and (i,k) ∈ NZ(A)
        l_ik = l_ik d_kk
        for j = k+1, ..., i-1 and (i,k) ∈ NZ(A)
            l_ij = l_ij - l_ik u_kj
        end
        If (i,k) ∈ NZ(A) then d_ii = d_ii - l_ik u_ki
        for j = i+1, ..., n and (i,k) ∈ NZ(A)
            u_ij = u_ij - l_ik u_kj
        end
    end
end
for i = 1, ..., n
    for j = i+1, ..., n and (i,k) ∈ NZ(A)
        u_ij = u_ij d_ii
    end
end
```

Figure B.4: Algorithm for the incomplete LU decomposition of the matrix A.

with $k = 0, ..., N_{x3} - 1$, $j = 0, ..., N_{x2} - 1$ and $i = 0, ..., N_{x1} - 1$. Each equation of the full matrix has the stencil

	$(i, j+1, k+1)$ $(i, j+1, k)$ $(i, j+1, k-1)$	
$(i-1, j, k+1)$ $(i-1, j, k)$ $(i-1, j, k-1)$	$(i, j, k+1)$ (i, j, k) $(i, j, k-1)$	$(i+1, j, k+1)$ $(i+1, j, k)$ $(i+1, j, k-1)$
	$(i, j-1, k+1)$ $(i, j-1, k)$ $(i, j-1, k-1)$	

. (B.10)

Equations of the lower triangular matrix have the stencil

	$-$	
$(i-1, j, k+1)$		
$(i-1, j, k)$	(i, j, k)	
$(i-1, j, k-1)$	$(i, j, k-1)$	
	$(i, j-1, k+1)$ $(i, j-1, k)$ $(i, j-1, k-1)$	

. (B.11)

For the upper triangular matrix the stencil is reversed with (i, j, k) being part of both stencils. This shows that equations belonging to grid points with $i + j + 2k = c = \text{const}$ can be solved simultaneously, as they do not depend on each other. It is, however, necessary that all equations with $i + j + 2k < c$ have already been solved for the lower triangular matrix and all equations with $i + j + 2k > c$ for the upper triangular matrix. The outlined independence of the equations can be exploited for vectorisation.

B.4 MULTIGRID

Basic iteration schemes like Gauß-Seidel or successive overrelaxation (SOR) as well as the ILU preconditioning step described in Section B.2 reduce high frequency residuals with a few iteration steps while low frequency residuals need many iteration steps. The information is spread by the solver only over short grid distances. For this reason MULTIGRID methods have been developed (e.g. Hackbusch, 1985; Briggs

et al., 2000; Trottenberg et al., 2001). The basic idea is that the problem is also solved on a coarser grid where the residual has a higher frequency with respect to the grid. While on the high frequencies are removed on the initial fine grid, the low frequencies are removed on the coarse grid. A basic MULTIGRID scheme consists of the following elements which will be explained in this section: a restriction operator, a smoother, a prolongation operator and a direct solver on the coarsest grid. Let G_i be $n+1$ grid levels for $i = 0, ..., n$ where G_0 is the main grid. Higher i indicate a coarser grid (Fig. B.5). For each grid level the problem to solve is $A^i x^i = b^i$. Note that in this section the superscript does not denote potentiation but that the quantity is on grid level i. The restriction operator R_i^{i+1} acts on the residual r^i on level i to yield the residual $r^{i+1} = R_i^{i+1} r^i$ on level $i+1$. This means that the residual r^i is coarsened to grid G_{i+1}. The prolongation operator P_{i+1}^i does the reverse - it interpolates the solution x^{i+1} on level $i+1$ to the solution $x^i = P_{i+1}^i x^{i+1}$. The smoother S^i is an operator that removes the high frequencies on level i that have not been removed by the coarse grid solution and are also caused by the prolongation.

The basic algorithm used in METRAS is based on a V-cycle (Fig. B.6). In the coarsening stage the residual on level i is smoothed and thereby a correction to the solution on this level is found. The new residual is restricted to level $i+1$ where the steps as on level i are taken. If the coarsest level has been reached, the residual on this level is not smoothed, but a direct solution to remove it completely (in terms of the requested accuracy) is sought. Now the refining stage is started. The solution on level $i+1$ is prolongated to level i, where it is used to correct the hitherto solution on this level. This new residual is smoothed again and the improved solution is prolongated again unless the finest level has been reached.

Basically, there are two methods to set up the matrices A^i. Algebraic MULTIGRID uses the prolongation and restriction operators to transform the matrix on level i

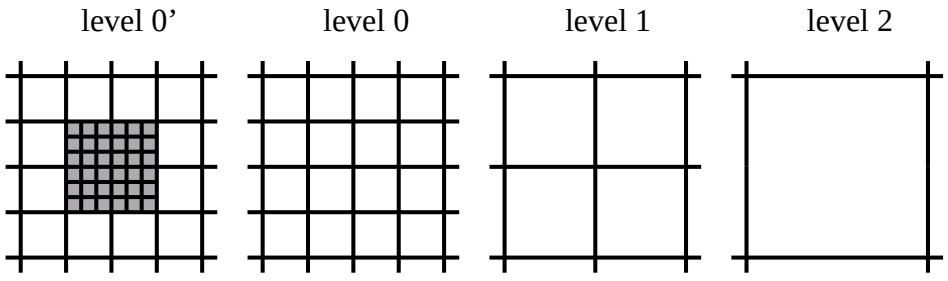

Figure B.5: Different grid levels for multigrid (horizontal cross-section). Level 0' is only used if a refined grid is present.

to get the matrix on level $i+1$:

$$A^{i+1} = R_i^{i+1} A^i P_{i+1}^i$$

In this case prolongation operator and restriction operator are usually chosen to satisfy $R_i^{i+1} = c{P_{i+1}^i}^\tau$. This means they are adjoint operators except for the constant c.

The alternative is geometric MULTIGRID. This method can be chosen when the matrix A is derived by the discretisation of a differential equation. In this case the discretisation can be performed on a coarser grid to yield the higher level matrices. In the context of multiple grids geometric MULTIGRID is also easier to implement.

Tests for this work have shown that geometric MULTIGRID is superior to algebraic MULTIGRID methods for the METRAS matrix on a single grid (not presented here). Therefore, the experiments presented in this thesis are always performed with geometric MULTIGRID (or Bi-CGSTAB).

For METRAS the grid is coarsened in each spatial direction by a factor of two from one level to the next (Fig. B.5). If a refined grid is present, an extra level 0' is added. The coarsening from level 0' to level 0 takes place in the area of the refined grid and only in the horizontal - the fine grid is removed (Fig. B.5). This means that from level 0' to level 0 the grid is not coarsened by a factor of two but by the actual refinement factors N_x and N_y of the subgrid, i.e. three in Fig. B.5.

For this work the restriction operation is based on aggregation, while the prolongation operator distributes the solution from the coarse cell to the fine cell. The latter can be interpreted as 0th order interpolation. Higher order interpolation is

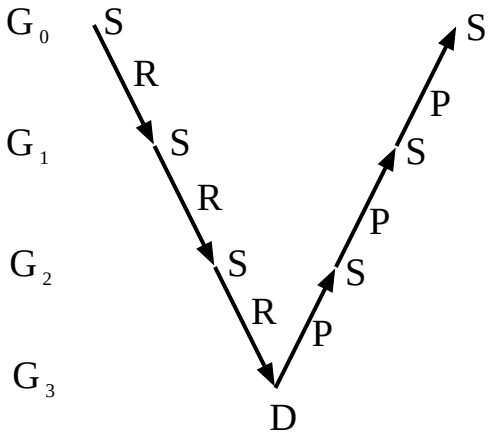

Figure B.6: V-cycle for MULTIGRID. G_i is the grid level, S = smoothing, P = prolongation, R = restriction, D = direct solve.

not necessary, since the coarse grid solution is only an approximation where higher order interpolation does not create additional information. Furthermore, a smoothing operator quickly eliminates the high frequency disturbances caused by the 0th order interpolation. As smoothing operator the inverse of the ILU(0) factorisation of A^i is taken ($S^i = (U^i + I)^{-1}(L^i + I)^{-1}$, Section B.2). Here it is assumed that A^i has already been scaled with D^i. As outlined in the algorithm in Fig. B.7 within each smoothing step the smoothing operator is applied twice.

$x = 0$
$r^0 = b$
for k=1, 2, ...
  for i=0, ..., n-1
    $x'^i = S^i r^i$
    $r' = r^i - A^i x'^i$
    $x'^i = x'^i + S^i r'$
    $r' = r^i - A^i x'^i$ } smoothing
    $r^{i+1} = R_i^{i+1} r'$ restriction
  end
  solve of $A^n x^n = r^n$ with Bi-CGSTAB
  for i=n-1, ..., 0
    $x'^i = x'^i + P_{i+1}^i x^{i+1}$ prolongation
    $r' = r^i - A^i x'^i$
    $x'^i = x'^i + S^i r'$
    $r' = r^i - A^i x'^i$
    $x'^i = x'^i + S^i r'$ } smoothing
  end
  $r^0 = A^0 x'^0 - r^0$
  $x = x + x'^0$
end

Figure B.7: Algorithm for the iterative solution of $Ax = b$ with MULTIGRID.

Bibliography

Allen, S. J. and **R. A. Vincent** (**1995**): Gravity wave activity in the lower atmosphere: Seasonal and latitudinal variations. *J. Geophys. Res.*, **100**, 1327–1350.

Anthes, R. A. (**1970**): Numerical experiments with a 2-dimensional horizontal variable grid. *Mon. Wea. Rev.*, **98**, 810–822.

Augustin, W., **V. Heuveline**, **G. Meschkat**, **K. H. Schlünzen** and **G. Schroeder** (**2007**): OpenMP Parallelization of the METRAS meteorology model: Application to the America's Cup. *High Performance Computing in Science and Enginering*. Springer.

Bacon, D. P., **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** and **J. W. Zack** (**2000**): A dynamically adapting weather and dispersion model: The operational multiscale environmental model with grid adaptivity (OMEGA). *Mon. Wea. Rev.*, **128**, 2044–2076.

Bacon, D. P., **N. N. Ahmad**, **T. J. Dunn**, **S. G. Gopalakrishnan**, **M. S. Hall** and **A. Sarma** (**2007**): Hurricane track forecasting with OMEGA. *Nat. Hazards*, **41**, 457–470.

Baer, F., **H. J. Wang**, **J. J. Tribbia** and **A. Fournier** (**2006**): Climate modeling with spectral elements. *Mon. Wea. Rev.*, **134**, 3610–3624.

Barret, R., **M. Berry**, **T. F. Chan**, **J. Demmel**, **J. M. Donato**, **J. Dongarra**, **V. Eijkhout**, **R. Pozo**, **C. Romine** and **H. Van der Vorst** (**1994**): *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, 124 pp.

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

Berger, M. J. and **J. Oliger** (**1984**): Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comp. Phys.*, **53**, 484–512.

Blackadar, A. K. (**1962**): The vertical distribution of wind and turbulent exchange in a neutral atmosphere. *J. Geophys. Res.*, **67**, 3095–3102.

Bohnenstengel, S. I. (**2007**): *Applicability range of subgrid-scale surface flux parameterizations*. PhD thesis, in preparation, Meteorological Institute, ZMAW, University of Hamburg, Hamburg.

Bonaventura, L., **L. Kornblueh**, **T. Heinze** and **P. Ripodas** (**2005**): A semi-implicit method conserving mass and potential vorticity for the shallow water equations on the sphere. *Int. J. Numer. Meth. Fluids*, **47**, 863–869.

Bott, A. (**1989**): A positive definite advection scheme obtained by nonlinear renormalization of the advective fluxes. *Mon. Wea. Rev.*, **117**, 1006–1015.

Brackbill, J. U. and **J. S. Saltzman** (**1982**): Adaptive zoning for singular problems in 2 dimensions. *J. Comp. Phys.*, **46**, 342–368.

Briggs, W. L., **H. Van Emden** and **S. F. McCormick** (**2000**): *A Multigrid Tutorial*. SIAM, Philadelphia, 193 pp.

Bungert, U. (**2002**): *Bedeutung von Neueisbildung für die Entwicklung eines zyklonalen Wirbels an einer Polynja*. Diploma thesis, Fachbereich Geowissenschaften, Universität Hamburg, 91 pp.

Bungert, U. (**2007**): *Einfluss der Nestung auf die Ergebnisse meteorologischer Modelle*. PhD thesis, in preparation, Meteorological Institute, ZMAW, University of Hamburg, Hamburg.

Carpenter, R. L., **K. K. Droegemeier**, **P. R. Woodward** and **C. E. Hane** (**1990**): Application of the piecewise parabolic method (PPM) to meteorological modeling. *Mon. Wea. Rev.*, **118**, 586–612.

Chlond, A. (**1994**): Locally modified version of Botts advection scheme. *Mon. Wea. Rev.*, **122**, 111–125.

Choi, D.-I., **J. D. Brown**, **B. Imbiriba**, **J. Centrella** and **P. MacNeice** (**2004**): Interface conditions for wave propagation through mesh refinement boundaries. *J. Comp. Phys.*, **193**, 398–425.

Cote, J., **J. G. Desmarais**, **S. Gravel**, **A. Methot**, **A. Patoine**, **M. Roch** and **A. Staniforth** (**1998**a): The operational CMC-MRB global environmental multiscale (GEM) model. Part II: Results. *Mon. Wea. Rev.*, **126**, 1397–1418.

Cote, J., **S. Gravel**, **A. Methot**, **A. Patoine**, **M. Roch** and **A. Staniforth** (**1998**b): The operational CMC-MRB global environmental multiscale (GEM) model. Part I: Design considerations and formulation. *Mon. Wea. Rev.*, **126**, 1373–1395.

Crowley, W. P. (**1968**): Numerical advection experiments. *Mon. Wea. Rev.*, **96**, 1–11.

Davies, H. C. (**1976**): Lateral boundary formulation for multilevel prediction models. *Q. J. R. Meteorol. Soc.*, **102**, 405–418.

Deuflhard, P. and **A. Hohmann** (**2003**): *Numerische Mathematik I. Eine algorithmisch orientierte Einführung.* de Gruyter Lehrbuch, Berlin, 370 pp.

Dierer, S. (**1997**): *Validierung des mesoskaligen Transport- und Strömungsmodells METRAS*. Diploma thesis, Fachbereich Geowissenschaften, Universität Hamburg, 120 pp.

Dierer, S. and **K. Schlünzen** (**2005**): Influence parameters for a polar mesocyclone development. *Meteorol. Z.*, **14**, 781–792.

Dierer, S., **K. Schlünzen**, **G. Birnbaum**, **B. Brümmer** and **G. Müller** (**2005**): Atmosphere-sea ice interactions during cyclone passage investigated by using model simulations and measurements. *Mon. Wea. Rev.*, **133**, 3678–3692.

Dietachmayer, G. S. and **K. K. Droegemeier** (**1992**): Application of continuous dynamic grid adaption techniques to meteorological modeling. 1. Basic formulation and accuracy. *Mon. Wea. Rev.*, **120**, 1675–1706.

Dunkerton, T. J. (**1997**): The role of gravity waves in the quasi-biennial oscillation. *J. Geophys. Res.*, **102**, 26 053–26 076.

Dutton, J. A. and **G. H. Fichtl** (**1969**): Approximate equations of motion for gases and liquids. *J. Atmos. Sci.*, **26**, 241–264.

Dyer, A. J. (**1974**): A review of flux-profile relationships. *Boundary-Layer Meteorol.*, **7**, 363–372.

Easter, R. C. (**1993**): Two modified versions of Bott's positive-definite numerical advection scheme. *Mon. Wea. Rev.*, **121**, 297–304.

Feistauer, M., **J. Felcman** and **I. Straskraba** (**2003**): *Mathematical and Computational Methods for Compressible Flow.* Oxford University Press, 552 pp.

Fletcher, R. (**1975**): Conjugate gradient methods for indefinite systems. G. A. Watson (Hg.), *Numerical Analysis*. Springer, Berlin, 73–89.

Fock, B. H. (**2007**): *METRAS als Grobstrukturs imulationsmodell*. Diploma thesis, in preparation, Fachbereich Geowissenschaften, Universität Hamburg.

Fournier, A., **M. A. Taylor** and **J. J. Tribbia** (**2004**): The spectral element atmosphere model (SEAM): High-resolution parallel computation and localized resolution of regional dynamics. *Mon. Wea. Rev.*, **132**, 726–748.

Fox-Rabinovitz, M., **J. Cote**, **B. Dugas**, **M. Deque** and **J. L. McGregor** (**2006**): Variable resolution general circulation models: Stretched-grid model intercomparison project (SGMIP). *J. Geophys. Res.*, **111**, D16 104.

Fox-Rabinovitz, M. S., **G. L. Stenchikov**, **S. M. J.** and **L. L. Takacs** (**1997**): A finite-difference GCM dynamical core with a variable-resolution stretched grid. *Mon. Wea. Rev.*, **125**, 2943–2968.

Fritts, D. C. and **M. J. Alexander** (**2003**): Gravity wave dynamics and effects in the middle atmosphere. *Rev. Geophys.*, **41**, 1003.

Garnier, E., **P. Sagaut** and **M. Deville** (**2002**): Large eddy simulations of shock/homogene turbulence interaction. *Comput. Fluids*, **31**, 245–268.

Gassmann, A., **M. Baldauf**, **L. Bonaventura**, **T. Heinze**, **H.-J. Herzog**, **P. Korn**, **L. Kornblueh**, **D. Majewski**, **M.-P. Ripodas** and **H. Wan** (**2007**): ICON - ICOsahedral Nonhydrostatic general circulation model: Die Zukunft der Numerischen Wettervorhersage und Klimasimulation. *Abstract, DACH 2007.* Hamburg, Germany.

Gopalakrishnan, S. G., **D. P. Bacon**, **N. N. Ahmad**, **Z. Boybeyi**, **T. J. Dunn**, **M. S. Hall**, **Y. Jin**, **P. C. S. Lee**, **D. E. Mays**, **R. V. Madala**,

A. Sarma, **M. D. Turner** and **T. R. Wait** (**2002**): An operational multiscale hurricane forecasting system. *Mon. Wea. Rev.*, **130**, 1830–1847.

Grell, G., **J. Dudhia** and **D. R. Stauffer** (**1994**): A Description of the Fifth-Generation Penn State/NCAR Mesoscale Model (MM5) . Techn. Rep. NCAR/TN-398+STR, National Center for Atmospheric Research, Boulder, Colorado.

Hackbusch, W. (**1985**): *Multi-grid Methods and Applications.* Springer, Berlin, 377 pp.

Hackbusch, W. (**1993**): *Iterative Lösung großer schwachbesetzter Gleichungssysteme.* Teubner Verlag, Stuttgart, 404 pp.

Hardiker, V. (**1997**): A global numerical weather prediction model with variable resolution. *Mon. Wea. Rev.*, **125**, 59–73.

Harten, A., **B. Engquist**, **S. Osher** and **S. R. Chakravarthy** (**1987**): Uniformly high order accurate essentially non-oscillatory schemes, III. *J. Comput. Phys.*, **71**, 231–303.

Hebbinghaus, H., **S. Dierer** and **K. H. Schlünzen** (**2007**): Sensitivity studies on vortex development over a polynya. *Theoretical and Applied Climatology*, **88**, 1–16.

Hestenes, M. R. and **E. Stiefel** (**1952**): Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.*, **49**, 409–436.

Holton, J. R. (**1992**): *An Introduction to Dynamic Meteorology.* Academic Press, San Diego, 507 pp.

Jablonowski, C., **M. Herzog**, **J. E. Penner**, , **R. C. Oehmke**, **Q. F. Stout**, **B. van Leer** and **K. G. Powell** (**2006**): Block-structured adaptive grids on the sphere: Advection experiments. *Mon. Wea. Rev.*, **134**, 3691–3713.

Jablonowski, J. (**2004**): *Adaptive grids in weather and climate modeling.* PhD thesis, The University of Michigan, Michigan, 272 pp.

Jiang, G.-S. and **C.-W. Shu** (**1996**): Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, **126**, 202–228.

Kapitza, H. and **D. Eppel** (**1987**): A 3-D Poisson solver based on conjugate gradients compared to standard iterative methods and its performance on vector computers. *J. Comput. Phys.*, **68**, 474–484.

Kessler, E. (**1969**): On distribution and continuity of water substance in atmospheric circulations. Mereorol. Monogr., American Meteorological Society, 84 pp.

Laprise, J. P. R. and **A. Plante** (**1995**): A class of semi-Lagrangian integrated-mass (SLIM) numerical transport algorithms. *Mon. Wea. Rev.*, **123**, 553–565.

Läuter, M., **D. Handorf**, **N. Rakowsky**, **J. Behrens**, **S. Frickenhaus**, **M. Best**, **K. Dethloff** and **W. Hiller** (**2007**): A parallel adaptive barotropic model of the atmosphere. *J. Comp. Phys.*, **223**, 609–628.

Lax, P. D. and **B. Wendroff** (**1960**): Systems of conservation laws. *Comm. Pure Appl. Math.*, **13**, 217–237.

Leonard, B. P. (**1979**): Stable and accurate convective modeling procedure based on quadratic upstream interpolation. *Comput. Meth. Appl. Mech. Eng.*, **19**, 59–98.

Leonard, B. P., **A. P. Lock** and **M. K. MacVean** (**1996**): Conservative explicit unrestricted-time-step multidimensional constancy-preserving advection schemes. *Mon. Wea. Rev.*, **124**, 2588–2606.

Leonard, B. P., **M. K. MacVean** and **A. P. Lock** (**1995**): The flux integral method for multidimensional convection and diffusion. *Appl. Math. Modelling*, **19**, 333–342.

Lin, S. J. and **R. B. Rood** (**1996**): Multidimensional flux-form semi-Lagrangian transport schemes. *Mon. Wea. Rev.*, **124**, 2046–2070.

Liu, X.-D., **S. Osher** and **T. Chan** (**1994**): Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, **115**, 200–212.

Long, R. R. (**1953**): Some aspects of the flow of stratified fluids. I: A theoretical investigation. *Tellus*, **5**, 42–58.

Lorenz, P. and **D. Jacob** (**2005**): Influence of regional scale information on the global circulation: A two-way nesting climate simulation. *Geophys. Res. Let.*, **32**, L18 706.

Lukáčová-Medviďová, M., **K. W. Morton** and **G. Warnecke** (**2004**): Finite Volume Evolution Galerkin (FVEG) methods for hyperbolic systems. *SIAM J. Sci. Comp.*, **26(1)**, 1–30.

Lukáčová-Medviďová, M., **S. Noelle** and **M. Kraft** (**2007**): Well-balanced Finite Volume Evolution Galerkin methods for the shallow water equations. *J. Comp. Phys.*, **221**, 122–147.

Lukáčová-Medviďová, M. and **J. Saibertova-Zatocilová** (**2006**): Finite Volume schemes for multi-dimensional hyperbolic systems based on the use of bicharacteristics. *Appl. Math.*, **51**, 205–228.

Lüpkes, C. and **K. H. Schlünzen** (**1996**): Modelling the Arctic convective boundary-layer with different turbulence parameterizations. *Boundary-Layer Meteorol.*, **79**, 107–130.

Mesinger, F. and **A. Arakawa** (**1976**): Numerical methods used in atmospheric models. GARP Publication Series No. 14, WMO/ICSU Joint Organizing Committee, 64 pp.

Niemeier, U. and **K. H. Schlünzen** (**1993**): Modelling steep terrain influences on flow patterns at the isle of Helgoland. *Beitr. Phys. Atmosph.*, **66**, 45–62.

Noelle, S. and **C. Steiner** (**2007**): Space-time splitting of adjoint error representations for hyperbolic conservation laws. *Submitted to Numerical Methods for Partial Differential Equations.*

Ogura, Y. and **N. A. Phillips** (**1962**): Scale analysis of deep and shallow convection in the atmosphere. *J. Atmos. Sci.*, **19**, 173–179.

Phillips, N. A. and **J. Shukla** (**1973**): On the strategy of combining coarse and fine grid meshes in numerical weather prediction. *J. Appl. Met.*, **12**, 763–770.

Prusa, J. M. and **P. K. Smolarkiewicz** (**2003**): An all-scale anelastic model for geophysical flows: dynamic grid deformation. *J. Comput. Phys.*, **190**, 601–622.

Ries, H. (**2009**): *Influence of sea ice on polar mesocyclones.* PhD thesis, in preparation, Meteorological Institute, ZMAW, University of Hamburg, Hamburg.

Robert, A. (**1993**): Bubble convection experiments with a semi-implicit formulation of the Euler equations. *J. Atmos. Sci.*, **50**, 1865–1873.

Saad, Y. (**2003**): *Iterative Methods for Sparse Linear Systems.* SIAM, Philadelphia, 528 pp.

von Salzen, K., **M. Claussen** and **K. H. Schlünzen** (**1996**): Application of the concept of blending height to the calculation of surface fluxes in a mesoscale model. *Meteorol. Z.*, **5**, 60–66.

Schimmel, F. (**2002**): *Development and Test of an Atmospheric Flow Model Employing Adaptive Numerical Methods.* PhD thesis, Berichte aus dem Zentrum für Meeres- und Klimaforschung, A 36, Zentrum für Meeres- und Klimaforschung, Universität Hamburg, Hamburg, 122 pp.

Schlünzen, K. H. (**1990**): Numerical studies on the inland penetration of sea breeze fronts at a coastline with tidally flooded mudflats. *Beitr. Phys. Atmosph.*, **63**, 243–256.

Schlünzen, K. H. (**1996**): Validierung hochauflösender Regionalmodelle. Habilitationsschrift, Berichte aus dem Zentrum für Meeres- und Klimaforschung A 23, Zentrum für Meeres- und Klimaforschung, Universität Hamburg, Hamburg, 184 pp.

Schlünzen, K. H., **K. Bigalke**, **C.-J. Lenz**, **C. Lüpkes**, **U. Niemeier** and **K. von Salzen** (**1996**): Concept and realization of the mesoscale transport and fluid model 'METRAS'. METRAS Technical Report 5, Meteorologisches Institut der Universität Hamburg, 131 pp.

Schlünzen, K. H. and **J. J. Katzfey** (**2003**): Relevance of sub-grid-scale land-use effects for mesoscale models. *Tellus*, **55A**, 232–246.

Schlünzen, K. H. and **E. M. I. Meyer** (**2007**): Impacts of meteorological situations and chemical reactions on daily dry deposition of nitrogen into the Southern North Sea. *Atmos. Environ.*, **41-2**, 289–302.

Schlünzen, K. H., **G. Schröder**, **G. Meschkat** and **V. Augustin, W. Heuveline** (**2007**): High resolution model simulations for the region of Valencia with the non-hydrostatic model METRAS. *In preparation.*

Schmidt, F. (**1977**): Variable fine mesh in spectral global models. *Beitr. Phys. Atmos.*, **50**, 211–217.

Schroeder, G. and **K. H. Schlünzen** (**2007**a): Flux integrated (weighted) essentially non-oscillatory advection schemes. *To be submitted to Q. J. R. Meteorol. Soc.*, pp. 20, Chapter 4 of this thesis.

Schroeder, G. and **K. H. Schlünzen** (**2007**b): Numerical dispersion of gravity waves. *To be submitted to Mon. Wea. Rev.*, pp. 14, Chapter 5 of this thesis.

Schroeder, G., **K. H. Schlünzen** and **F. Schimmel** (**2006**): Use of (weighted) essentially non-oscillatory advection schemes in a mesoscale model. *Q. J. R. Meteorol. Soc.*, **132**, 1509–1526, Chapter 3 of this thesis.

Schueler, S. and **K. H. Schlünzen** (**2006**): Modeling of oak pollen dispersal on the landscape level with a mesoscale atmospheric model. *Environ. Mod. Assess.*, **11**, 179–194.

Schueler, S., **K. H. Schlünzen** and **F. Scholz** (**2005**): Viability and sunlight sensitivity of oak pollen and its implications for pollen-mediated gene flow. *Trees*, **19**, 154–161.

Schumann, U., **T. Hauf**, **H. Schmidt** and **H. Volkert** (**1987**): A mesoscale model for the simulation of turbulence, clouds and flow over mountains: Formulation and validation examples. *Beitr. Phys. Atmosph.*, **60**, 413–446.

Shapiro, R. (**1971**): The use of linear filtering as a parameterization of atmospheric motion. *J. Atmos. Sci.*, **28**, 523–531.

Shu, C.-W. (**1999**): High order ENO and WENO schemes for computational fluid dynamics. T. J. Barth and H. Deconinck (Hg.), *High-Order Methods for Computational Physics*. Springer, New York, 439–582.

Skamarock, W. C. (**2006**): Positive-definite and monotonic limiters for unrestricted-time-step transport schemes. *Mon. Wea. Rev.*, **134**, 2241–2250.

Skamarock, W. C. and **J. B. Klemp** (**1993**): Adaptive grid refinement for two-dimensional and three-dimensional nonhydrostatic atmospheric flows. *Mon. Wea. Rev.*, **121**, 788–804.

Skamarock, W. C., **P. K. Smolarkiewicz** and **J. B. Klemp** (**1997**): Preconditioned conjugate-residual solvers for Helmholtz equations in nonhydrostatic models. *Mon. Wea. Rev.*, **125**, 587–599.

Smolarkiewicz, P., **V. Grubisic** and **L. Margolin** (**1997**): On forward-in-time differencing for fluids: Stopping criteria for iterative solutions of anelastic pressure equations. *Mon. Wea. Rev.*, **125**, 647–654.

Smolarkiewicz, P. K. (**1983**): A simple positive definite advection scheme with small implicit diffusion. *J. Comput. Phys.*, **111**, 479–486.

Smolarkiewicz, P. K. and **L. G. Margolin** (**1998**): MPDATA: A finite-difference solver for geophysical flows. *J. Comput. Phys.*, **140**, 459–480.

Smolarkiewicz, P. K. and **J. A. Pudykiewicz** (**1992**): A class of semi-Lagrangian approximations for fluids. *J. Atmos. Sci.*, **49**, 2082–2096.

Sokol, Z. (**1999**): Comparison of several numerical schemes applied to advection equations. *Q. J. R. Meteorol. Soc.*, **125**, 213–224.

Staniforth, A. and **J. Cote** (**1991**): Semi-Lagrangian integration schemes for atmospheric models - a review. *Mon. Wea. Rev.*, **119**, 2206–2223.

Steppeler, J., **G. Doms**, **U. Schättler**, **H. W. Bitzer**, **A. Gassmann**, **U. Damrath** and **G. Gregoric** (**2003**): Meso-gamma scale forecasts using the nonhydrostatic model LM. *Meteorol. Atmos. Phys.*, **82**, 75–96.

Strang, G. (**1968**): On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, **5**, 506–517.

Stull, B. S. (**1988**): *An Introduction to Boundary Layer Meteorology.* Kluwer Academic Publishers, Dordrecht, 666 pp.

Sündermann, A. (**1990**): *Die Anwendung von adaptiven Gittern in zwei einfachen Modellen für eine atmosphärische Rollenzirkulation.* PhD thesis, Berichte aus dem Zentrum für Meeres- und Klimaforschung, A 5, Zentrum für Meeres- und Klimaforschung, Universität Hamburg, Hamburg, 118 pp.

Thuburn, J. (**1996**): Multidimensional flux-limited advection schemes. *J. Comput. Phys.*, **123**, 74–83.

Tremback, C. J., **J. Powell**, **W. R. Cotton** and **R. A. Pielke** (**1987**): The forward-in-time upstream advection scheme: extension to higher orders. *Mon. Wea. Rev.*, **115**, 540–555.

Trottenberg, U., **C. W. Oosterlee** and **A. Schüller** (**2001**): *Multigrid*. Academic Press, San Diego, 631 pp.

Trukenmüller, A., **D. Grawe** and **K. H. Schlünzen** (**2004**): A model system for the assessment of ambient air quality conforming to EC directives. *Meteorol. Z.*, **13**, 387–394.

Vichnevetsky, R. (**1987**): Wave propagation and reflection in irregular grids for hyperbolic equations. *Appl. Numer. Math.*, **3**, 133–166.

Van der Vorst, H. A. (**1989**): High-performance preconditioning. *SIAM J. Sci. Statist. Comput.*, **10**, 1174–1185.

Van der Vorst, H. A. (**1992**): Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, **13**, 631–644.

Van der Vorst, H. A. (**2003**): *Iterative Krylov Methods for Large Linear Systems*. Cambridge University Press, Cambridge, 221 pp.

Xue, M. (**2000**): High-order monotonic numerical diffusion and smoothing. *Mon. Wea. Rev.*, **128**, 2853–2864.

Xue, M., **K. K. Droegemeier** and **V. Wong** (**2000**): The Advanced Regional Prediction System (ARPS) - A multi-scale nonhydrostatic atmospheric simulation and prediction tool. Part I: Model dynamics and verifications. *Meteorol. Atmos. Phys.*, **75**, 161–193.

Zängl, G. (**2002**): An improved method for computing horizontal diffusion in a sigma-coordinate model and its application to simulations over mountainous topography. *Mon. Wea. Rev.*, **130**, 1423–1432.

Zhang, D. L., **H. R. Chang**, **N. L. Seaman**, **T. T. Warner** and **J. M. Fritsch** (**1986**): A two-way interactive nesting procedure with variable terrain resolution. *Mon. Wea. Rev.*, **114**, 1330–1339.

Zhang, P. (**2004**): Generation of mesoscale gravity waves in upper-tropospheric jet-front systems. *J. Atmos. Sci.*, **61**, 440–457.

Zhang, S. and **F. Yi** (**2004**): A numerical study on global propagations and amplitude growths of large-scale gravity wave packets. *J. Geophys. Res.*, **109**, D07 106.