Development and Test of an Atmospheric Flow Model Employing Adaptive Numerical Methods

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

vorgelegt von

Frank Schimmel
aus
Hamburg

Hamburg
22. Mai 2002

Als Disseration angenommen vom Fachbereich Geowissenschaften
der Universität Hamburg
auf Grund der Gutachten von PD. Dr. K. Heinke Schlünzen
und Prof. Dr. Jens Struckmeier

Hamburg, den 9. Juli 2002

Prof. Dr. J.O. Backhaus
(Prodekan des Fachbereichs Geowissenschaften)

Abstract

The development and interaction of small-scale flow features embedded in larger-scale atmospheric phenomena and their numerical simulation are the focus of this work. Capturing small-scale flow features in a numerical simulation is ultimately a question of the resolution of the computational grid. The impact of grid resolution on simulation results is studied for the example of a land-sea breeze circulation using a the mesoscale model METRAS. As expected, the dependency of simulation results on resolution is strongest in the vicinity of the sea breeze front.

To accurately capture small-scale features more efficiently in simulations, the new atmospheric model a3m is developed. One of a3m's key features is local grid refinement with a high level of control of the spatial distribution of resolution. The model is based on the fully compressible, non-hydrostatic flow equations and employs only very few approximations. The equations are discretised using a Godunov-type finite volume (FV) method employing the approximate Riemann solver of Osher and Solomon as the numerical flux function. The scheme is extended to second order by using essentially non-oscillatory (ENO) variable interpolation. Local refinement of the computational grid is achieved by using block-structured grids: a grid consists of multiple blocks of different resolution and refining blocks may overlay parts of refined blocks.

In order to verify the numerical properties and to demonstrate the capability of a3m to simulate atmospheric flows, it is tested using computational grids of uniform and locally enhanced resolution. Overall, the model performs as expected and turns out to be quite robust towards abrupt changes in resolution. The model is applied to an atmospheric density current which is successfully simulated. The front of the density current is well captured with only a very small amount of artificial, non-physical oscillations. In comparison to published simulation results that used other numerical methods, a3m's FV/ENO scheme seems to be more diffusive, but it generates significantly fewer artifical oscillations. Significant improvements of the results are gained by simulating the density current on locally refined grids.

Zusammenfassung

Die Entwicklung und Wechselwirkungen von kleinskaligen Strömungsmerkmalen, die in atmosphärische Phänomene größerer Skalen eingebettet sind, und deren numerische Simulation bilden den Schwerpunkt dieser Arbeit. Die kleinskaligen Strömungsmerkmale in einer Simulation richtig wiederzugeben, ist letztendlich eine Frage der Auflösung des verwendeten Rechengitters. Der Einfluß der Auflösung auf Simulationsergebnisse wird mit Hilfe des Mesoskalamodells METRAS am Beispiel der Land-Seewind-Zirkulation untersucht. Erwartungsgemäß ist dieser in der Nähe der Seewindfront am stärksten ausgeprägt.

Um die kleinskaligen Strömungsmerkmale in einer Simulation richtig und effektiv zu erfassen, wird das neue Atmosphärenmodell a3m entwickelt. Eine der wesentlichen Eigenschaften von a3m ist die Möglichkeit, lokal verfeinerte Rechengitter mit einem hohen Maß an Kontrolle über die Verteilung der Auflösung einzusetzen. Das Modell basiert auf den nichthydrostatischen, kompressiblen Strömungs-Gleichungen und verwendet nur sehr wenige Näherungen. Die Gleichungen werden mittels eines Finite-Volumen-Verfahrens (FV) vom Godunov-Typ diskretisiert. Als numerische Flußfunktion wird der approximative Riemann-Löser von Osher und Solomon benutzt. Die Approximationsgüte des Verfahrens wird durch die Verwendung der wesentlich nichtoszillatorischen (*essentially non-oscillatory*, ENO) Interpolation auf zweite Ordnung gesteigert. Lokale Gitterverfeinerung wird durch die Verwendung block-strukturierter Gitter erreicht: Ein Gitter besteht aus mehreren Blöcken unterschiedlicher Auflösungen, wobei Blöcke höherer Auflösung Blöcke geringerer Auflösung teilweise überdecken.

Mit Tests auf Gittern von gleichförmiger und lokal erhöhter Auflösung werden die numerischen Eigenschaften von a3m überprüft und die Durchführbarkeit von Simulationen atmosphärischer Strömungen demonstriert. Insgesamt erfüllt das Modell die gestellten Erwartungen und erweist sich als recht robust gegenüber abrupten Änderungen der räumlichen Auflösung. Das Modell wird auf eine atmosphärische Dichteströmung angewandt und diese erfolgreich simuliert. Die Front der Dichteströmung wird dabei gut und mit nur einem geringen Maß künstlicher unphysikalischer Schwingungen erfaßt. Im Vergleich zu anderen veröffentlichten Simulationsergebnissen, die mit anderen numerischen Methoden erzielt wurden, erscheint das Verfahren von a3m diffusiver, aber es erzeugt auch erheblich weniger künstliche Oszillationen. Durch Verwendung lokal verfeinerter Gitter konnten die Modellergebnisse noch deutlich verbessert werden.

Contents

Chapter 1

Motivation

Atmospheric flows exhibit many distinct patterns and phenomena such as Rossby waves, mid-latitude cyclones, mountain (lee) waves, thunderstorms and other convection cells, dust devils and countless others. These atmospheric phenomena or flow patterns are usually associated with a *scale*, a characteristic range of spatial and temporal extents which these phenomena or patterns have. Figure 1.1 gives an overview on some of these typical atmospheric flow patterns and their scales (after Orlanski, 1975 and Schlünzen, 1996).

In general, atmospheric flows also have several characteristic *features*: they include structures of smaller spatial and possibly temporal extent than the flows they are embedded in. A prominent example are cyclones which include fronts as an essential part of their pattern. Both, fronts and cyclones, are shown as separate phenomena in Figure 1.1, but fronts are always associated with a larger circulation. However, this

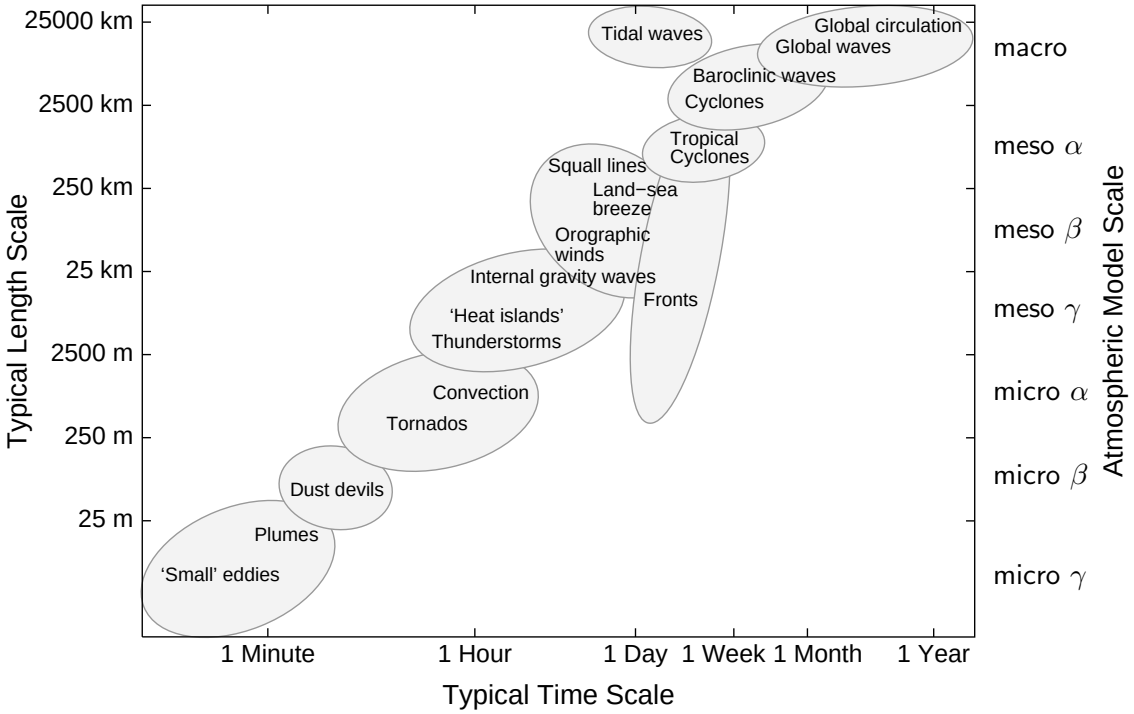

Figure 1.1: Atmospheric phenomena and atmospheric scales (after Orlanski, 1975 and Schlünzen, 1996).

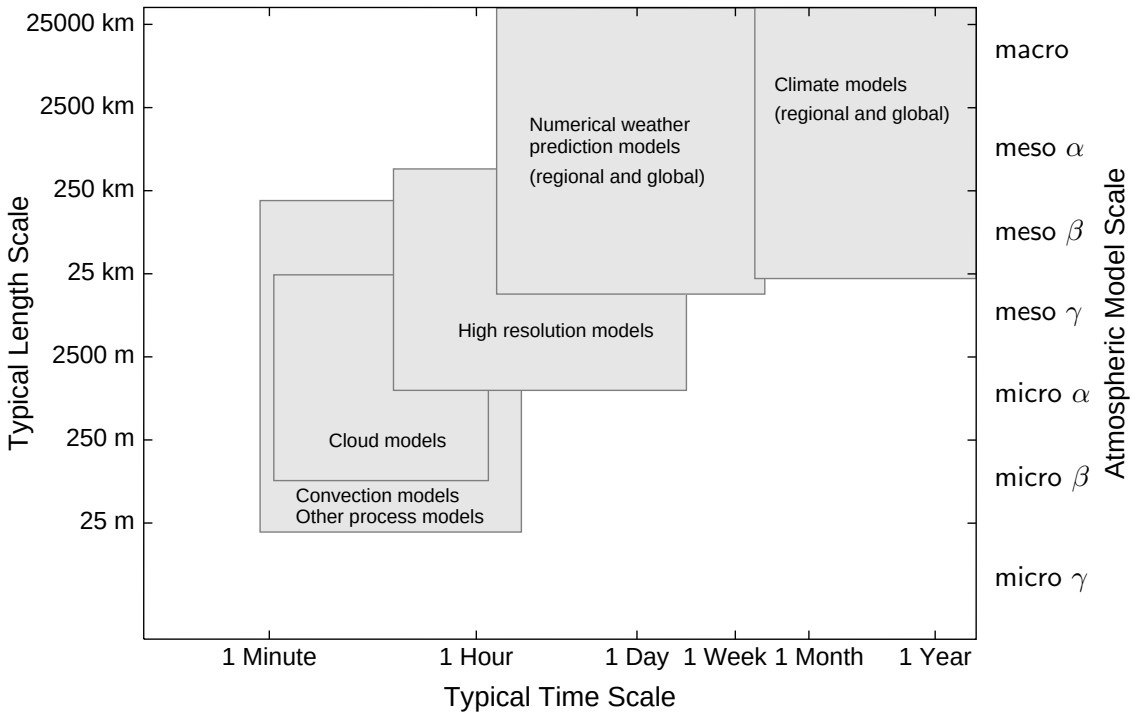

Figure 1.2: Hierarchy of atmospheric models (after Schlünzen, 1996).

is not necessarily a cyclone: land-sea breeze circulations and other density currents also have a front as one of their characteristic features (Simpson, 1987).

Studies of the atmosphere usually focus on a narrow range of spatial and temporal scales, on a single scale or even just a single phenomenon. This enables the introduction of assumptions and approximations specific for the scale under consideration. These assumptions and approximations ease the understanding of flows of that scale, their mechanism and the processes involved. In this context the notion of distinct atmospheric scales has proven very helpful because the full complexity of atmospheric processes without scale specific approximations most often precludes such understanding on an analytical level. Nevertheless, in the real atmosphere phenomena interact across scales.

The approach of atmospheric scales is also employed for numerical analysis and simulation: there are distinct 'classes' of atmospheric models focusing on certain scales and used for different purposes, e.g. models for numerical weather prediction (NWP) simulate on different scales and employ different approximations than models for process studies such as cloud micro-physics. Figure 1.2 shows some classes of atmospheric models defined by their scales and purpose.

However, the scale-wise approach has an inherent limitation: interaction between phenomena of different scales is not captured. In the real atmosphere all of the flows

and features listed in Figure 1.1 and numerous others co-exist. Flows on all spatial and temporal scales interact across scales, altering or destroying flow patterns and features and creating new ones. But assumptions and approximations hold only on specific scales. Thus, an analysis on one scale cannot be simply extended or transfered to other scales and methods developed for that scale are in general not applicable to others.

Numerical studies suffer an additional restriction: which processes, phenomena or flow features can be simulated is ultimately dependent on the spatial and temporal resolution of the model employed. Due to the multi-scale nature of atmospheric flows simulations of the larger-scale flow always have to face the problem that a resolution sufficient to resolve the flow in the large scale is too coarse to accurately capture the small-scale features. It is not feasible to run simulations on a large domain—to include large-scale structures—and at a high resolution—to capture small-scale structures—because the computer resources available for a simulation will always be limited. Consequently, the effects of small-scale processes on the resolved scale flow are usually only parameterised, and the large-scale flow is often prescribed. However, in some cases it is the accurate prediction of these smaller scale features, like the position of a front, that is considered crucial for the quality of a simulation result or forecast. So numerical simulation of phenomena on different scales and interaction between scales is highly desirable. But it faces two main problems: firstly, approximations suitable for one scale are not applicable in a different scale and, secondly, the range of spatial and temporal extents required to capture the large scale-flow patterns *and* their small-scale features is very broad.

The approach most frequently used in atmospheric modelling to (partially) work around the restrictions due to resolution is *nesting*. In this procedure, models of different scale and/or classes or the same model deployed with varying resolutions are used jointly to simulate a single scenario. Sometimes a whole suite of models is established, e.g. for NWP: a high-resolution local model is nested into a medium-resolution regional model which is in turn nested into a coarse-resolution global model. Often only one-way nesting is performed, i.e. information is transferred from the large-scale to the small-scale model, but feedback from small-scale flows on the larger ones is ignored. Nesting solves some of the problems of numerical simulations of atmospheric flows related to resolution: resolution is increased locally so that small-scale patterns within the larger-scale flow—simulated by the outer coarse-resolution model—can now be captured by the inner high-resolution model.

Nesting also has serious limitations: the focus is usually on a single region covered by the innermost highest-resolution model in a nesting arrangement. To obtain the simulation results on this rather limited region the whole suite of models has to be run giving coarser-resolution results for a much larger domain than the actual

region of interest. This procedure is costly and not very efficient, unless the coarser-resolution results are also used otherwise. An even more important aspect is the spatial distribution of resolution, i.e. where to have a fine or coarse resolution: in the nesting approach, it is determined by the region of interest and nesting arrangement alone. Which resolutions are actually *required* to capture the flow situation at some point is not taken into account. Thus, such a nested simulation system may still fail to capture multi-scale phenomena like flows with embedded fronts. Fronts can only be resolved by the inner, high-resolution model. The outer, coarse-resolution model cannot resolve fronts properly because it lacks resolution perpendicular to the front. Consequently, the boundary values fed into the inner model may contain mere bands of somewhat increased gradients but no actual fronts. The inner model has to *develop a front itself* from the front-less low-resolution values provided at its boundaries.

An additional complication in the nesting approach is that models of different scales employ different approximations, parameterisations and numerical methods. This raises some non-trivial problems at the interior boundaries between the different models: numerical solutions gained with one model, i.e. solutions of a specific set of model equations, are in general no valid solutions for a different set of model equations with different approximations. Consequently, they cannot be directly used as boundary values of the inner model: some kind of assimilation to the different model equations has to be performed.

Admittedly, flows with an embedded front are an example of atmospheric flows that are particularly difficult to simulate. A high resolution is required to capture a front, but simulating the whole flow at such a high resolution is prohibitively expensive. However, fronts are also a very localised phenomena: idealised, they are only two-dimensional while the atmosphere as the containing entity has three dimensions. The nesting approach with its weaknesses discussed above could be used to decrease the computational effort by providing a high resolution at the location of the front and a lower resolution elsewhere. But nesting provides a static spatial distribution of resolution only. Therefore, it is applicable only in cases where the required distribution of resolution is known in advance. This is hardly possible for the real atmosphere because fronts are not stationary. For the general case, this suggests the application of adaptive grids as they are used in other fields of application for flows containing discontinuities (e.g. Muzaferija and Gosman, 1997; Dolejší, 1998; Meister and Sonar, 1998; Abgrall *et al.*, 1999).

For the atmosphere, only few efforts have been made to employ grid adaption to improve simulation results. Basically, two different adaption strategies have been used: one strategy employs grids which are dynamically deformed with time (e.g. Sündermann, 1990) and the other strategy is the algorithm of Berger and Oliger

(1984). The latter has some resemblance to the method of nesting, but the grid arrangement is dynamically adapted and not static. It has been applied to the atmosphere by Skamarock and Klemp (1993) and is also implemented in the model ARPS (Xue *et al.*, 2000).

Although grid adaption generally seems to be considered useful and its overall applicability has been proven, neither of the above adaption strategies is in widespread or routine use for atmospheric simulations. Nevertheless, the commonly used approaches employing static distributions of resolutions—non-uniform and/or deformed grids and nesting—generally do not provide the means to capture the interaction of flow patterns and features across a broad range of scales. To derive these means, a new atmospheric model is developed in this work. It aims to solve the problem of resolution better than existing approaches and to enable simulations of atmospheric flows spanning multiple scales so that the interaction of phenomena of different scales can be explicitly simulated. To this end, it will employ a different adaption strategy than the two mentioned above.

The new model has been named a3m, where a3m really is a³m with a triple 'a'. In the tradition of recursive acronyms[1] a3m stands for "**a**3m is an **a**daptive **a**tmospheric **m**odel". a3m is developed as a research prototype for future atmospheric models. Consequently, the primary design goals are flexibility, modularity and extensibility. Performance is usually one of the strongest constraints for atmospheric models, but for the current development it is considered of minor importance. At this stage of development, a3m should be able to simulate flows accurately with only few artificial oscillations or other numerical artefacts. Equally important is the ability to stably simulate flows on grids with high, even abrupt spatial variations of resolution. This is required for the local refinement of the computational grid and the later adaption process. The early development of a3m focuses on the atmospheric mesoscale (Figure 1.1), but extensibility to larger and smaller scales is, of course, a requirement to meet the final development goal of a3m: simulations spanning a broad range of scales and thereby explicitly capturing the multi-scale nature of atmospheric flows.

'Model' and 'modelling' are very popular terms among scientists of all disciplines, and they are used with various different meanings. The common notion for the term 'model' is that it designates an approximate, formal description of phenomena, processes or other entities and the relations between them. It usually involves some degree of abstraction from the described entity, the *problem domain*, to the description given by the model. Both, formality and exactness, may vary depending

[1] The expansion of a recursive acronym contains that very acronym again. See also the corresponding entry from the jargon file (Jargon, 2002, `http://www.tuxedo.org/~esr/jargon/html/entry/recursive-acronym.html`).

on the type of model. Within this work, the term 'model' is used with four different meanings whose relations are shown in Figure 1.3:

'The' Model designates the final product, a3m: a piece of software. It consists of executable programs, libraries of routines and classes as well as accompanying documentation. It is also referred to as 'the model' or just 'a3m'.

The Mathematical Model is a formal description of the atmosphere by mathematical means, i.e. differential, integral and/or algebraic equations.

The Discrete Model is gained from the mathematical model by application of discretisation techniques resulting in a set of algebraic difference equations.

The Software Model has two sub-categories *conceptual model* and *design model* (Larman, 1998). Within the conceptual model the discrete model is analysed yielding concepts of the problem domain and their interrelations. From these

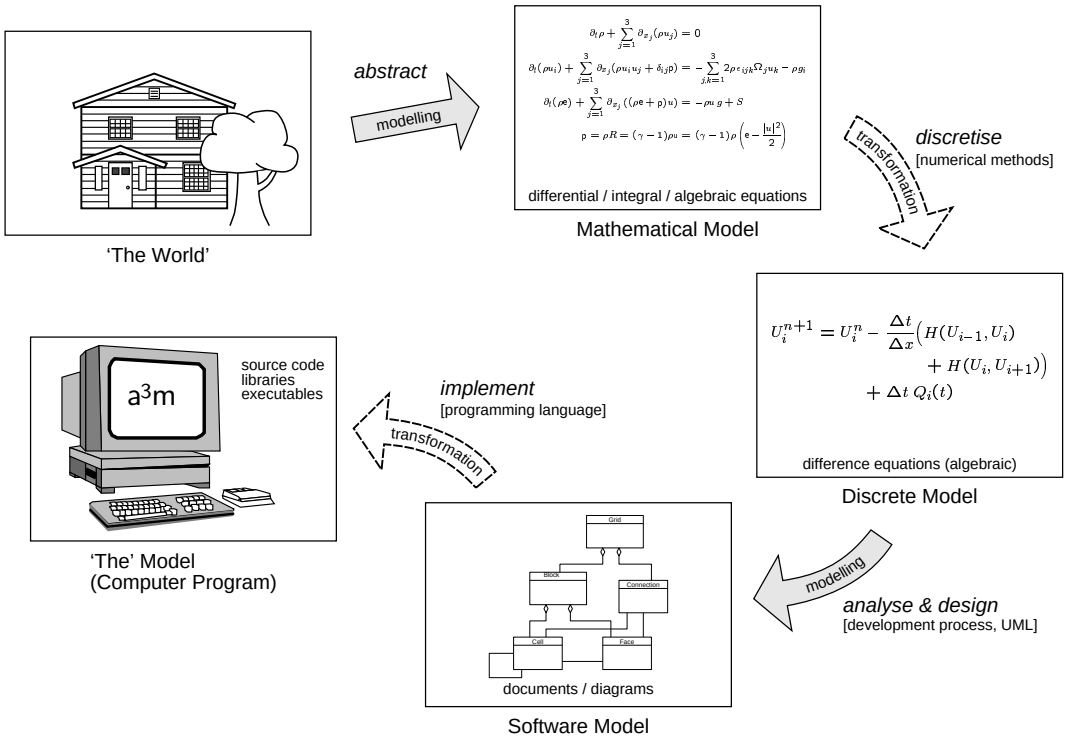

Figure 1.3: The development process of a3m.

The sketched development comprises four stages all of which are called a 'model' of some sort. Only two of the development steps between the different models are actual modelling acts, i.e. finding an abstracting, formal description (at the head of the respective arrows) of the problem to model (at the tail end). In the other two steps the models at the respective arrows head are gained from those at the tail end by some sort of transformation rules.

suitable abstractions for the implementations are derived. The design model emerging thereof describes an implementation ('the' model, a3m) of the problem (discrete model). It corresponds very closely to the actual code.

The remainder of this work is structured as follows: In Chapter 2 the difficulties encountered with conventional atmospheric models in simulations of multi-scale atmospheric flows are demonstrated using the example of the land-sea breeze circulation. In Chapter 3 the mathematical model of a3m is presented. It introduces the model equations and also includes an extensive discussion of assumptions and approximations that are frequently employed in atmospheric models of various scales. Chapter 4 presents the discretisation in space and time of the mathematical model introduced and discussed in Chapter 3. The resulting discrete model is subjected to various tests in Chapter 5. The tests are comprised of a simple mathematical model problem, the Burgers equation, to verify and demonstrate some of the numerical properties of a3m and an application of a3m to an atmospheric density current. Chapter 6 gives a summary of this work, its achievements and conclusions and an extensive outlook on the future of a3m.

The notations and symbols used in this work are summarised on page 111.

Chapter 2 Example of a Multi-scale Atmospheric Phenomenon

An example of an atmospheric flow system is the land-sea breeze circulation, a mesoscale phenomenon which exhibits features on multiple scales, most notably a front. It was chosen because it is among the most profoundly studied mesoscale flows (see Atkinson (1981) for a summary) and it has a variety of different flow features (Atkinson, 1981; Simpson, 1987, 1994).

2.1 Principal Features of the Land-Sea Breeze Circulation

Land-sea breezes are common local wind systems. They can develop on the edge of any large body of water[1] under the influence of solar heating. The contrast between the relatively quickly heating land surface and the relatively slowly heating water surface results in a gradually developing temperature difference causing a difference in density between the relatively warm air residing over land and the relatively cool air over the water. Consequently, the cooler marine air flows inland forcing the lighter air into an upward motion. Where the advancing breeze meets the continental air, a zone of high horizontal convergence is formed. The convergence increases during the development of the circulation and thereby maintains the contrast between the cool marine and warm continental air. Idealised, the temperature and density fields are discontinuous at the boundary between the two air masses, the sea breeze front. Mixing of air across the front and thereby exchange of properties between the air masses is very low which makes the sea breeze front particularly interesting with respect to e.g. concentration of pollutants. Convergence and the formation of the front is much enhanced if the breeze is opposed by a light large-scale off-shore wind. With a large-scale on-shore wind turbulent mixing is stronger relative to the convergence and the formation of the front is usually suppressed until the afternoon hours, if it occurs at all. Also, the breeze develops a bulge, the flow head, at its leading edge, if the on-shore breeze is opposed by an off-shore large-scale wind. The compensating outflow of air over sea results in a subsidence and, finally, a complete thermally driven circulation cell emerges. Figure 2.1 shows a sketch of the structure of a sea breeze.

[1] This is not necessarily an ocean: a large lake, e.g. Lake Victoria (e.g. Fraedrich and Flohn, 1966) or Lake Michigan (e.g. Laird *et al.*, 2001), is sufficient.

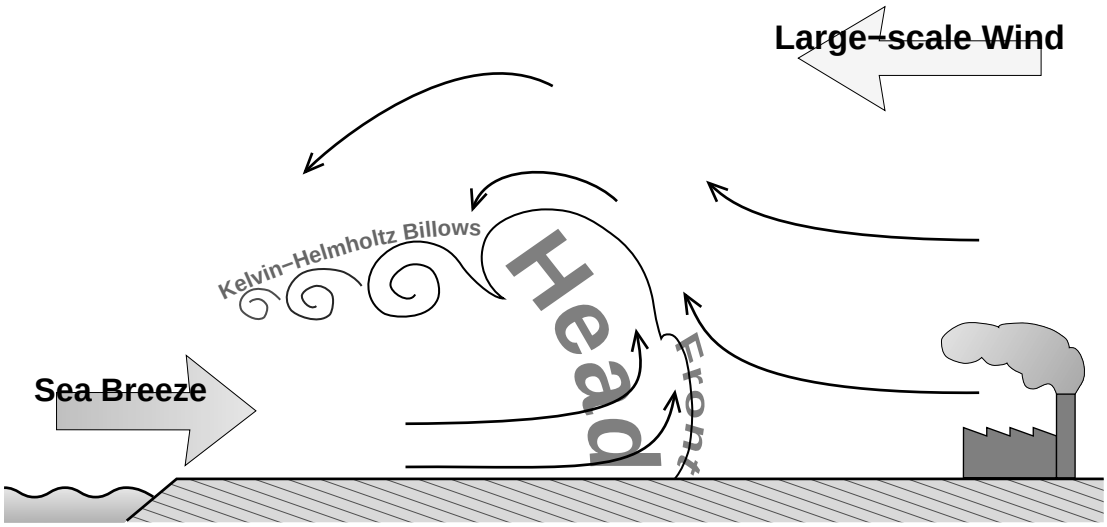

Figure 2.1: Schematic of a land-sea breeze.

Sea breezes can generate atmospheric bores or mesoscale gravity waves (Simpson, 1994, 1987). Here the still stably stratified air above the mixed boundary layer is forced to rise while flowing over the head of the advancing sea breeze. This vertical displacement triggers gravity waves travelling horizontally within the stable layers at high speed. These waves can in turn imprint patterns on the unstable and well mixed boundary layer below and the convective cells within it.

A simplified model of the sea breeze circulation already showing many of its characteristic features is a density current. These can be generated in a laboratory experiment by releasing a relatively dense fluid into a tank of less dense fluid. Indeed, land-sea breezes can be considered a special form of density current with the additional complication of the diurnal cycle, a potentially non-uniform stratification of the atmosphere, interaction with convective processes, the Coriolis force, etc.

As mentioned before, the sea breeze front separating marine and continental air is of special interest. It has received a lot of attention because of its strong impact on the costal climate—e.g. the costal temperature, humidity and wind direction—as well as on the dispersion of pollutants. Many and large urban settlements are in immediate vicinity of the sea or large lakes which has twofold consequences: firstly, urban areas are also areas of increased heat storage and emission of pollutants, and secondly, obviously a large number of people are immediately affected. From the viewpoint of numerical modelling, fronts are particularly interesting because idealised they represent discontinuities in the predicted variables, or at least small and very confined zones of strong gradients. These are generally difficult to simulate with a numerical model since a high spatial resolution is required to accurately capture the zones of strong gradients. Additionally, fronts are usually not stationary.

2.2 Dependency of Simulation Results on Model Resolution

For the following investigation of the impact of grid resolution on numerical simulation results the mesoscale transport and flow model METRAS (Schlünzen, 1988, 1990; Schlünzen *et al.*, 1996) was employed to simulate a sea breeze circulation. It is a non-hydrostatic anelastic model that solves the primitive equations in mass conserving flux form. A finite difference approximation is used for solving the equations on a non-uniform grid in terrain following coordinates.[2] Turbulent mixing is modelled by a first order closure using a counter gradient scheme for unstable and a mixing length approach for stable stratification (Lüpkes and Schlünzen, 1996). The model utilises a rigid lid at its top with damping layers to prevent reflections of waves at the upper boundary and free in/out-flow boundary conditions are employed at the lateral boundaries. At the lower boundary momentum and heat flux are calculated according to the similarity theory of Monin and Obukhov (1954). The time dependent surface temperature and humidity are calculated by application of the force restore method (Deardorff, 1978). A detailed model description is given in Schlünzen *et al.* (1996).

METRAS has been validated using an extensive set of test scenarios following an evaluation concept for mesoscale models (Schlünzen, 1996, 1997; Dierer, 1997). It has been used in a broad range of applications from examining influence factors on land-sea breezes (Schlünzen, 1990; Schlünzen and Krell, 1994), flows over steep terrain (Niemeier and Schlünzen, 1993), studies of aerosol particle development (von Salzen and Schlünzen, 1999b), interaction of atmosphere and sea ice (Birnbaum, 1998; Dierer, 2002), as well as regional case studies (Sheng *et al.*, 2000).

The scenario used here to study the effect of different model resolutions on the development of the sea breeze circulation is a regular test case for atmospheric models and part of the evaluation concept for mesoscale models mentioned above. The sensitivity of METRAS results has already been shown for two different grid resolutions for this scenario (Dierer, 1997), but has not yet been investigated systematically. To emphasise the effect of different grid resolutions on the results, the scenario is kept very simple and as many other factors as possible are excluded. The terrain is completely flat and the surface properties are uniform over sea and land, respectively. The shoreline runs straight from West to East with the sea to the South. The relative humidity is 80% near the ground and decreases linearly to 5% at the top of the model domain in 10 km height. To further reduce the complexity of the scenario, cloud formation is hindered for all simulations. The initial stratification is $\partial_z \Theta = 0.0035\,\mathrm{K/m}$, i.e. weakly stable. The large-scale geostrophic wind is 0.5 m/s

[2] In the simple cases used here this degenerates to Cartesian coordinates.

from North, i.e. off-shore. The initial surface temperatures for land and sea are 285 K and the initial sea level pressure is 1013.25 hPa.

For simplicity, the scenario is quasi-2d with a model domain four grid points in East-West direction and -80 km to +80 km in North-South direction. The shoreline runs through the origin of the domain which is situated at 51° North. The top of the model domain is at a height of 10 km. The simulated time period is one day (24 hours), so a whole cycle of land- and sea-breeze is included, starting at midnight of June 21st. Simulations are conducted on a series of grids of different but uniform horizontal resolution. In the vertical the grid spacing varies from 20 m at the surface to approximately 500 m at the upper boundary and is identical for all simulations. Resolution in East-West direction, i.e. parallel to the coast, is always 2000 m, but the resolution in North-South direction is varied from 8000 m for the coarsest grid to 500 m for the grid with the highest resolution. Note that the coarsest resolution chosen here is approximately the resolution used by todays high-resolution local weather prediction models.

Expectations towards simulations on increasingly fine grids include mainly two partially contrary effects. On the one hand, extrema are captured more accurately by a fine resolution as illustrated by Figure 2.2. This means that the magnitude of extreme values will be greater providing more feedback on the flow and thereby resulting in a more intense development. An example of this effect is shown in Figure 2.3 for the the coast-perpendicular wind obtained from two simulations of different resolution: the flow is more intense in the higher resolution simulation. On the other hand, with a higher resolution an increasing number of different processes is explicitly simulated along with the main flow. This is especially true for the

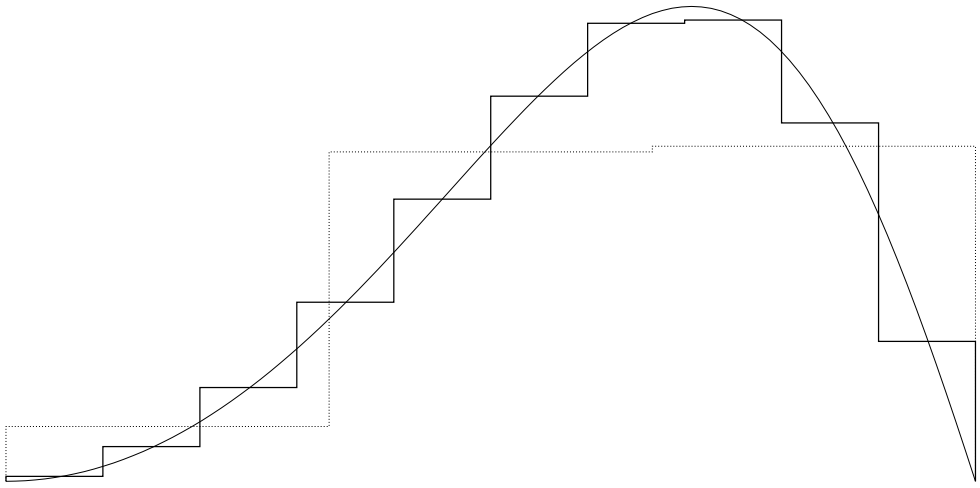

Figure 2.2: Capturing a local maximum by piecewise constant approximations of different resolution.

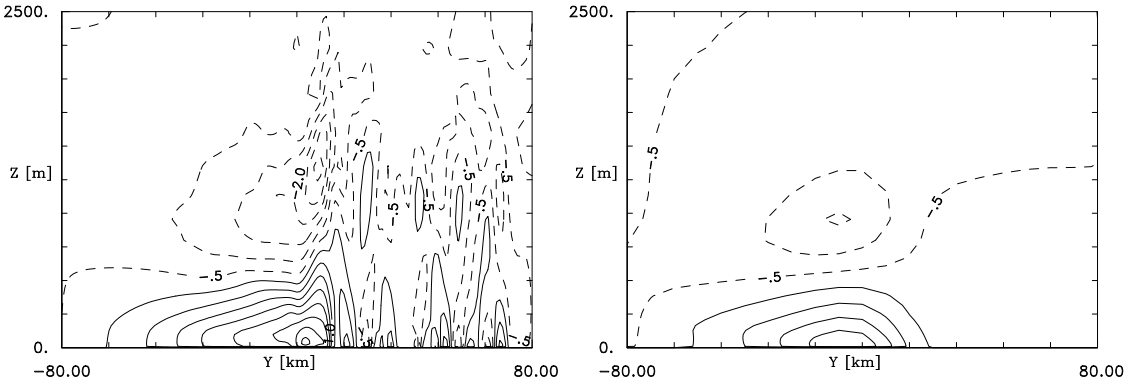

Figure 2.3: North-South velocity component of a land-sea breeze at 4 p.m. Shown are simulation results of METRAS at 1000 m (left) and 8000 m (right) horizontal resolution. On-shore winds are indicated by solid and off-shore winds by dashed contour lines.

small-scale variations that are no longer represented by the turbulence parameterisation but simulated directly. Density currents, including land-sea breezes, can e.g. develop Kelvin-Helmholtz instabilities behind the flow head with a chain of billows trailing after it (Sha *et al.*, 1991; Simpson, 1994). These small-scale structures consume energy from the main flow, such as the Kelvin-Helmholtz billows providing additional friction at the top of the current, thereby making the development *less* intense. Both effects, more intense extrema and the resolution of additional physical processes, provide non-linear feedback to the flow and may trigger instabilities. In case an instability is triggered in a simulation at one resolution but not at another, both simulations will develop substantially different solutions. It is therefore very difficult, if not impossible, to predict the effect of changing the resolution for a simulation in advance.

Of special interest in the comparison is the inland penetration of the sea breeze front shown in Figure 2.4 and the intensity of the up-draft in the frontal region shown in Figure 2.5. The up-draft is a measure of the intensity of the flow system. The sea breeze front separating marine and continental air is characterised by a rapid change in temperature, moisture and other concentrations within a small distance as well as by a strong horizontal convergence and a corresponding up-draft. Its position can be determined by any of these characteristics. Here, the location of the maximum horizontal temperature gradient, the maximum horizontal convergence and maximum up-draft are used as a measure for the position of the sea breeze front. In general, these criteria do not yield the same result and are therefore averaged. The location gained by this procedure cannot always be called the position of the front since the simulations with 8000 m (Figure 2.3, right) and 4000 m resolution did not develop a well confined zone of high gradients: their resolution is simply too

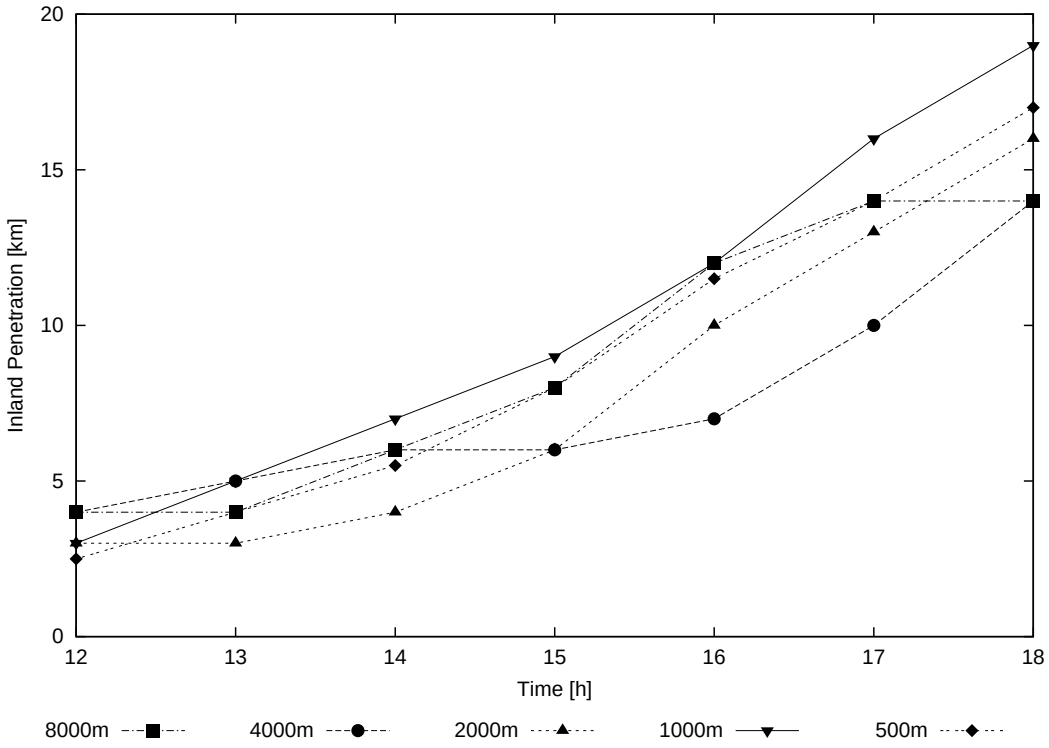

Figure 2.4: Inland penetration of the sea breeze front.

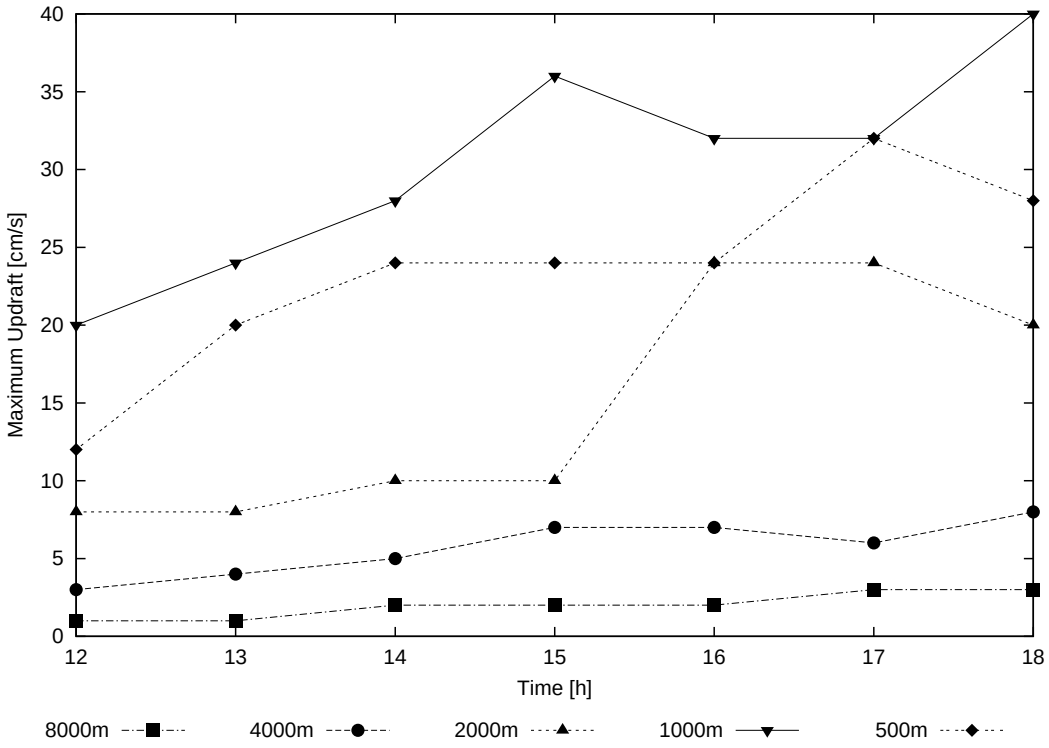

Figure 2.5: Maximum up-draft in the frontal zone.

coarse. Nevertheless, the typical acceleration in the inland penetration of the breeze during the afternoon hours (Atkinson, 1981) can be observed in all simulations, but it occurs significantly later for the 4000 m simulation (Figure 2.4). The 8000 m simulation comes to a complete halt rather early.

The differences in the advance of the front are not very large—5 km or about 25% at 6 p.m.—but the intensity of the up-draft varies from only 3 cm/s for the 8000 km simulation and 8 cm/s for the 4000 km simulation to 40 cm/s in case of the 1000 m simulation, i.e. by more than a factor of 10 (Figure 2.5). But the simulations with 8000 m and 4000 m resolution do not produce an actual front due to their lack of resolution, so a high uncertainty in the penetration should be assumed for these simulations. Also note that the highest resolution simulation with 500 m does not penetrate furthest and does not exhibit the strongest up-draft: the frictional effects mentioned above already set in.

For a statistical comparison of two simulation results obtained on differently resolving grids, the solution from the finer grid is aggregated to the coarser of the two grids by volume weighted averaging. This gives the result that the simulation on the coarse grid should have yielded if it had captured the flow as accurately as the simulation on the fine grid. Values can now be compared pointwise on the coarse grid. The results of the simulations on the 4000 m, 2000 m and 1000 m resolution grids are compared against the result of the simulation with the highest resolution grid, i.e. 500 m. The 8000 m simulation is excluded from further examinations since it is obviously very dissimilar to the other simulations' results. If a coarse resolution value does not deviate by more than a certain threshold from the corresponding aggregated reference value, it is called a *hit*. The percentage of hits in the total number of values in the comparison is the *hit rate*. The thresholds for each variable are based on the accuracy of routine measurements, i.e. if two routine measurements do not differ more than the threshold, they have to be considered equivalent. The maximum difference to count as a hit is 0.5 m/s for the horizontal velocity components and 0.1 m/s for the vertical velocity component. Note that these thresholds are significantly stricter than those of Cox *et al.* (1998). Considering that in general the frontal zone separating the sea breeze from ambient air is only approximately 200 m–400 m wide (Atkinson, 1981), a horizontal resolution of 500 m is still not high enough to resolve any details of the front structure. But it is sufficient to demonstrate the impact of grid resolution on the simulation results.

Since the sea breeze is an advectively dominated phenomenon, the wind field and its structure have a very high impact on the distribution of all other fields. It is therefore prime target for the analysis. Hit rates for the the wind field are computed in two distinct regions: one out over the sea, a quarter of the domain, and one over land 8 km to 12 km inland of the coastline. In the region over sea the hit rates for the

North-South and vertical velocity components are 100% for all runs for the whole duration of the simulation. The hit rates for the North-South velocity component in the region over land are shown in Figure 2.6. It can be seen, that in the morning hours, when the sea breeze is in its early stage of development, all simulations agree very well with the reference simulation. The 4000 m simulation starts to deviate from the reference simulation at about 2 p.m. and recovers after 7 p.m., when the driving force, the solar energy input, subsides significantly and the circulation as a whole is already in the decline. The 2000 m and 1000 m simulations show prominent drops in their hit rates at 3 p.m. and 4 p.m., respectively. In the reference simulation the front is just entering the evaluation region at 3 p.m. and has crossed it almost completely within one hour (Figure 2.4). In the 2000 m simulation the front enters and leaves the evaluation region significantly later. In the 1000 m simulation the front is slightly ahead relative to the reference simulation. So the peaks in the deviation occur at times, when the front is within the evaluation region in at least one of the compared simulations. After the front has left the evaluation region in both of the simulations compared, the agreement is again very good.

Summarising, the development of the sea breeze in the 8000 m and 4000 m simulations significantly differs from the reference simulation (and all others) already at

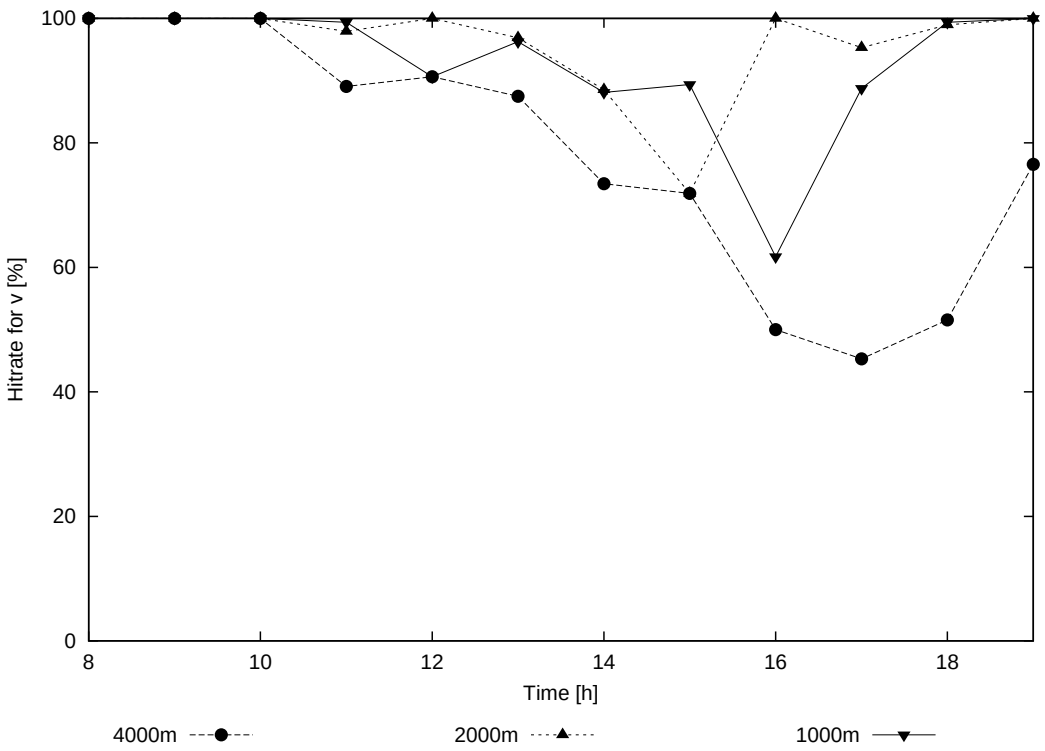

Figure 2.6: Hit rates for the North-South velocity component in a region over land in 8 km–12 km distance to the coast.

the first glance: they develop the basic thermal circulation pattern, but they do not show clear frontal areas. The characteristic flow head develops—if at all—very late. This can also be observed in the hit rate in the region over land for the 4000 m simulation: it shows significant deviations from the reference simulation during most of the intense development of the sea breeze and starts recovering, when the circulation is already decaying. In contrast to this the other simulations are in good agreement with the reference except for the frontal zone. Obviously, an increased resolution yields significantly different results near the front and the head of the flow. Therefore, a non-uniform grid would be much more suitable. The resolution could be low over sea, where no significant differences between the simulations occurred, and higher in the vicinity of the flow head and the leading front for capturing their development. But the region for which the high resolution is necessary also moves with time. In addition, for slightly different parameters, such as a change in the large-scale wind, it might be in an altogether different location. Other factors also play important roles: the intensity of the solar radiation, which in turn depends on the date and cloud cover, the surface humidity, land-use, etc. It is therefore more or less unpredictable in advance, where and when exactly a high resolution will be necessary. This problem could be overcome using adaptive grids which evolve over time with the computed solution, making land-sea-breezes—and gravity currents in general—a good test candidate for this method.

Recalling that the horizontal extent of the front is noticeably smaller than the grid spacing of the highest resolution grid used here, adaptivity could also overcome some limitations in resolution due to computing time restrictions: increasing spatial resolution also requires shortening the time step of the integration so that not only the number of grid points is increased but also this larger amount of grid points has to be re-computed more often for the same simulated period of time. With an adaptive strategy the necessity to resolve all of the computational domain as high as needed to properly capture the fronts structure is removed because it only needs to be applied in a region of similar extent than the front itself. An adaptive simulation would use much less grid points compared to a global high-resolution simulation and thus uses less memory and also computing time *despite* the decrease in the time step due to the (locally) fine computational grid.

Chapter 3

Mathematical Model of Atmospheric Motion

The previous chapter has demonstrated some difficulties of simulating atmospheric flows with an existing mesoscale model for the example of a land-sea breeze circulation. In this chapter, the mathematical model (Figure 1.3) of the new atmospheric model a3m is presented. It gives a mathematical description of the atmosphere by means of differential and algebraic equations. In order to avoid the problems and limitations experienced in Chapter 2, the new model is designed and constructed differently from existing atmospheric models. Special attention is paid to this end from the very beginning, the choice of the mathematical model.

The model, as presented here, is explicitly designed for the atmospheric mesoscale but should be extensible to the micro- and to the macroscale. In addition, the present work focuses on atmospheric dynamics. Extensions to include more parameterisations of physical processes and using a more complete set of equations are outlined at the end of the chapter.

3.1 Governing Equations and Approximations

The equations used and approximations applied to them determine the range of applicability of the implemented model—the computer program—to simulate different atmospheric flow situations. The model is limited to the scales in which the underlying equations, approximations and other assumptions hold, e.g. assuming incompressibility would restrict the model to a very small scale such as the flow around a building or even smaller (e.g. Stull, 1988) and assuming hydrostatic balance in the vertical would restrict the model to a very large scale such as the global circulation (e.g. Holton, 1992). Also, it has to be kept in mind that by solving the equations numerically, different phenomena and physical processes are captured depending on the resolution. This imposes a problem for atmospheric modelling since usually many sub-grid-scale physical processes, such as turbulence, surface layer processes or cloud microphysics, are parameterised: they are modelled in terms of the grid-scale fields to obtain an approximation of the effects of the parameterised process on the resolved scale. Obviously, parameterisations made for coarse resolutions are no longer applicable for a fine resolution if the parameterised processes are

now resolved explicitly. In the other direction, reducing the resolution makes parameterisations necessary for processes explicitly resolved before. Parameterisations and, similarly, approximations to the equations must be valid in the whole range of phenomena the model will be applied to and all grid resolutions employed. Due to the emphasis of this work on multi-scale phenomena—like flows with embedded fronts—and local grid refinement the choice of parameterisations and approximations is to be considered carefully, or they should be avoided altogether.

3.1.1 General Assumptions

From the above considerations follows that no assumptions, approximations or parameterisations can be employed which impose serious restrictions on the model with respect to simulating mesoscale phenomena and especially different resolutions of the computational grid. The only assumptions made throughout all of the following are:

1. Molecular viscous effects are negligible.
2. The flow velocity, i.e. the wind speed, is always sub-sonic.
3. Earth's angular velocity and gravitational acceleration vary only slowly compared to the resolutions considered.
4. Earth's gravitational acceleration does not change in magnitude with height within the considered domains.
5. The frame of reference is attached to the Earth, i.e. rotating.
6. Air is a homogeneous mixture of ideal gases and behaves like a single ideal gas with average properties.
7. Air is dry, i.e. there is no water vapour contained in it and, consequently, no clouds are formed.

Assumption 1 holds very well for the atmosphere except for the immediate vicinity of the ground or other solid surfaces. Usually, atmospheric models cannot resolve these viscous layers and employ parameterisations instead (Section 3.3). Restricting the flow velocity to be sub-sonic (assumption 2 above) is not an actual requirement for the considerations of this chapter. It is used by the discretisation of the advective fluxes only (Section 4.1.1.1). It is hardly a restriction anyway: except for very rare exceptions atmospheric flows are even low Mach number flows. Assumption 3

is required for the separation of resolved and sub-grid-scale parts of the solution (Section 3.2) and used only there.

Assumption 4 is employed by the introduction of the basic atmospheric state (Section 3.1.4). The error in the magnitude of the acceleration relative to the gravitational field of a spherical Earth is only about 0.3% in 10 km and 0.6% in 20 km height. This is a typical range of heights for the upper domain boundary for mesoscale simulations so that the constant-with-height assumption can be expected not to introduce a significant source of error.

In a rotating frame of reference (assumption 5) the ground is seemingly at rest which seems more intuitive. This, of course, also results in the occurrence of apparent forces, namely centrifugal and Coriolis force. The centrifugal force alters the effective gravity field by approximately 0.3% at surface level so that it is neglected in the following just as the variation of gravity with height. The Coriolis force acts always perpendicular to the flow direction and the Earth's angular velocity vector. It is thereby dependent on the relative direction of one to the other and also proportionally to the flow velocity. Thereby, its impact is dependent on the phenomena to be simulated and neglecting it may restrict the models applicability. Consequently, it has to be retained in the following.

Assumption 6 is met if the molecules in the gases involved are on average far enough apart so that electromagnetic interaction between them is negligible. So the gases have to be far from changing into the liquid or solid phase for the pressure and temperature ranges encountered in the atmosphere. This is fulfilled by all gases usually constituting air except water vapour. Nevertheless, that is usually also assumed ideal in atmospheric modelling (e.g. Doms and Herbert, 1985; Schlünzen, 1988; Pielke *et al.*, 1992; Grell *et al.*, 1995; Doms and Schättler, 1999; Xue *et al.*, 2000).

In the current stage of development of a3m water vapour is excluded explicitly by assumption 7 above. This is the most severe deviation from the real atmosphere. In fact, the latent heat released during condensation is a very efficient and important local source of energy and plays a crucial role in the dynamics of the atmosphere. It is to be considered a temporary restriction, though, and will be removed in the future. The extension of the model to the moist atmosphere is outlined in Sections 3.4 and 4.4.

No other assumptions than the 5 above are made about the frame of reference. For local atmospheric models it is common practice (e.g. Schlünzen, 1988; Pielke *et al.*, 1992; Grell *et al.*, 1995; Xue *et al.*, 2000) to write the model equations in a local coordinate system tangential to the Earth's surface with the x_1-axis pointing eastwards, the x_2-axis pointing northwards and the x_3-axis pointing up. This special

choice of a frame of reference is referred to as the (x, y, z)-system in the following—as opposed to the general Cartesian (x_1, x_2, x_3)-system. Using this frame of reference, gives $\underline{g} = (0, 0, g)^T$ for the gravitational acceleration and $\underline{\Omega} = (0, \Omega \sin\varphi, \Omega \cos\varphi)^T$ for the Earth's angular velocity, where φ is the latitude.

This kind of local coordinates are also implemented in a3m, but the program design provides a clear separation between the functionality dependent on the frame of reference and all other parts. It is thereby general enough, to use other frames of references by substituting only the relatively small portion of code dependent on the concrete choice of the frame of reference. The above choice is *not* a formal necessity for the considerations of this and the following chapters though. Accordingly, it will not be exploited and the general notation is retained, e.g. the coordinate x_3 is *not* identified with the height z, and generally the gravitational acceleration still has three non-zero components.

3.1.2 Conservation Equations

By assumptions 1 and 2 made above, atmospheric flows are assumed inviscid and sub-sonic. Otherwise, no restrictions follow from the assumptions above, e.g. the medium air is considered fully compressible. Such flows are governed by the conservation laws for mass, momentum and energy, a system of equations also known as the Euler equations. All equations are written in Cartesian coordinates and can be reviewed in many books on atmospheric dynamics or general fluid dynamics (e.g. Hirsch, 1988a; Holton, 1992; Ferziger and Perić, 1999).

The conservation of mass is expressed by the continuity equation,

$$\partial_t \rho + \sum_{j=1}^{3} \partial_{x_j}(\rho u_j) = 0, \tag{3.1}$$

where ρ is the density of air and $\underline{u} = (u_1, u_2, u_3)^T$ is the flow velocity.

The equations for conservation of momentum including gravitational and Coriolis forces are given by

$$\partial_t(\rho u_i) + \sum_{j=1}^{3} \partial_{x_j}(\rho u_i u_j + \delta_{ij}\mathsf{p}) = -\sum_{j=1}^{3}\sum_{k=1}^{3} 2\rho\, \epsilon_{ijk}\Omega_j u_k - \rho g_i \qquad i = 1, 2, 3, \tag{3.2}$$

where p is pressure, $\underline{g} = (g_1, g_2, g_3)^T$ and $\underline{\Omega} = (\Omega_1, \Omega_2, \Omega_3)^T$ are Earth's gravitational acceleration and angular velocity, respectively, δ_{ij} is the Kronecker symbol and ϵ_{ijk} is the Levi-Civita symbol giving a component-wise expression for the Coriolis force $2\rho\,\underline{\Omega} \times \underline{u}$.

The conservation of energy can be formulated for a multitude of different variables, including the internal energy u, total energy $\mathsf{e} = \mathsf{u} + \rho|\underline{u}|^2/2$ or enthalpy (again internal or total). In a3m the total specific energy e is used:

$$\partial_t(\rho\mathsf{e}) + \sum_{j=1}^{3} \partial_{x_j}\left((\rho\mathsf{e} + \mathsf{p})u_j\right) = -\sum_{j=1}^{3} \rho u_j\, g_j + S, \tag{3.3}$$

where S denotes the diabatic heating, e.g. by cloud formation or radiation if these processes were included.

3.1.3 Equation of State

The system formed by equations (3.1)–(3.3) is closed with the equation of state of an ideal gas (assumption 6 in Section 3.1.1):

$$\mathsf{p} = \rho R\mathsf{T} = (\gamma - 1)\rho\mathsf{u} = (\gamma - 1)\rho\left(\mathsf{e} - \frac{|\underline{u}|^2}{2}\right), \tag{3.4}$$

where $R = c_\mathsf{p} - c_V$ is the specific gas constant and $\gamma = c_\mathsf{p}/c_V$ the ratio of specific heat capacities of dry air at constant pressure and volume, respectively.

Note that the inner energy density $\rho\mathsf{u}$ is directly proportional to pressure and that the flow velocity needs to be in the order of 100 m/s to give an equal contribution to the total energy density $\rho\mathsf{e} = \rho\mathsf{u} + \rho|\underline{u}|^2/2$ as the inner energy, i.e. pressure. So predicting $\rho\mathsf{e}$ essentially captures the pressure field except for situations with very high wind speeds—like tornados—where dynamics also have a strong impact on the pressure field. A prediction equation for temperature could be deduced from equation (3.3) and (3.4) which is the usual approach taken by most mesoscale atmospheric models. But in the above form the analysis of the system of equations (3.1)–(3.3) utilised for the discretisation of the advective fluxes (Section 4.1.1) was found to be most convenient.

3.1.4 Basic Atmospheric State

One of the problems occurring when solving the system of equations (3.1)–(3.4) numerically is that the momentum flux contains two terms: the pure transport by the flow and the pressure gradient. These terms are of significantly different orders of magnitude which can lead to a dramatic precision loss when solving the equations numerically on a computer. Additionally, the variation of pressure with height is clearly dominated by the compression of each column of air by its own weight. But most of the net flux induced by the vertical pressure gradient is compensated by

the gravitational source term of equation (3.2). This can be taken advantage of by splitting the pressure into a basic state pressure $\mathsf{p}^{(0)}$ and a deviation from that $\mathsf{p}^{(1)}$:

$$\mathsf{p} = \mathsf{p}^{(0)}(z) + \mathsf{p}^{(1)}(\underline{x}),$$

where the basic state pressure $\mathsf{p}^{(0)}$ is a function of height z alone, while the deviation part is a fully tree-dimensional field depending on $\underline{x} = (x_1, x_2, x_3)^T$. Similarly, temperature is split into a basic state and deviation part in form of the potential temperature $\Theta := \mathsf{T}(\mathsf{p}/\mathsf{p}_\Theta)^{R/c_\mathsf{p}}$ with $\mathsf{p}_\Theta = 1000\,\mathrm{hPa}$:

$$\Theta = \Theta^{(0)}(z) + \Theta^{(1)}(\underline{x}).$$

The basic state pressure and temperature fields are defined to be in hydrostatic balance, i.e.

$$\partial_z \mathsf{p}^{(0)} = -\rho^{(0)} g = -\frac{\mathsf{p}^{(0)}}{R\mathsf{T}^{(0)}}\, g. \tag{3.5}$$

The basic state atmosphere is neutrally stratified, i.e. the basic state potential and real temperature are

$$\Theta^{(0)} = \mathsf{T}_r \left(\frac{\mathsf{p}_r}{\mathsf{p}_\Theta}\right)^{-R/c_\mathsf{p}} = const \qquad \text{and} \tag{3.6a}$$

$$\mathsf{T}^{(0)} = \mathsf{T}_r - \Gamma(z - z_r), \tag{3.6b}$$

where T_r is the real temperature of the basic state at a reference height z_r and $\Gamma := g/c_\mathsf{p}$ is the dry adiabatic lapse rate. From equations (3.5) and (3.6b) the basic state pressure follows as

$$\mathsf{p}^{(0)}(z) = \mathsf{p}_r \left(1 - \frac{\Gamma(z - z_r)}{\mathsf{T}_r}\right)^{c_\mathsf{p}/R}. \tag{3.7}$$

The basic state pressure $\mathsf{p}^{(0)}$ is a function of height z alone with respect to a reference pressure p_r and a reference temperature T_r at height z_r. By equations (3.4), (3.6) and (3.7) the basic state mass and energy density follow as

$$\rho^{(0)}(z) = \frac{\mathsf{p}^{(0)}(z)}{R\mathsf{T}^{(0)}(z)} = \frac{\mathsf{p}_r}{R\mathsf{T}_r}\left(1 - \frac{\Gamma(z - z_r)}{\mathsf{T}_r}\right)^{c_\mathsf{p}/R-1} \qquad \text{and}$$

$$(\rho\mathsf{e})^{(0)}(z) = \frac{\mathsf{p}^{(0)}(z)}{\gamma - 1} = \frac{\mathsf{p}_r}{\gamma - 1}\left(1 - \frac{\Gamma(z - z_r)}{\mathsf{T}_r}\right)^{c_\mathsf{p}/R}.$$

Note that in the adiabatic case—$S = 0$ in (3.3)—$\rho = \rho^{(0)}$, $\underline{u} = 0$ and $\rho\mathsf{e} = (\rho\mathsf{e})^{(0)}$ is already a (stationary) solution of the system of equations (3.1)–(3.4).

The basic state part can now be removed from the momentum equation (3.2) since the basic pressure gradient and gravitational force induced by the base density cancel each other. This yields

$$\begin{aligned} \partial_t(\rho u_i) + \sum_{j=1}^{3} \partial_{x_j}(\rho u_i u_j + \delta_{ij}\mathsf{p}^{(1)}) \\ = -\sum_{j=1}^{3}\sum_{k=1}^{3} 2\rho\,\epsilon_{ijk}\Omega_j u_k - \rho^{(1)} g_i \qquad i = 1,2,3. \end{aligned} \tag{3.8}$$

The major difference between the equation (3.2) and (3.8) is that only deviations from the basic state contribute to the pressure gradient and gravitational forces which are of about the same order of magnitude as the other forces, while the full pressure gradient and gravitational forces are of much greater magnitude.

3.1.5 Approximations to the Equations

Various approximations may be applied to the system of equations (3.1), (3.3), (3.4) and (3.8), depending on the scale of the phenomena to simulate. The probably most important for simulating atmospheric flows among these are—sorted from large to small scales of their applicability—the geostrophic, hydrostatic, Boussinesq, anelastic and incompressible approximations (e.g. Pielke, 1984; Doms and Herbert, 1985; Holton, 1992). From these, only the anelastic and Boussinesq approximations are frequently used in mesoscale models, e.g. METRAS (Schlünzen, 1988), MM5 (Grell *et al.*, 1995), GESIMA (Kapitza and Eppel, 1992) or RAMS (Pielke *et al.*, 1992). MM5 and RAMS also provide an option for using the hydrostatic approximation for larger-scale simulations. Other models use the compressible equations and apply the Boussinesq approximation only, e.g. ARPS (Xue *et al.*, 1995, 2000, 2001). The geostrophic and incompressible approximations are not suitable for the mesoscale at all (e.g. Pielke, 1984).

While the current stage of a3m's development is explicitly bound to the mesoscale, extension towards both, larger and smaller scales, should be possible and is planned for the future. All of the above approximations possibly impose a restriction with respect to supporting simulations on other scales than the mesoscale. Also, none of them is strictly necessary to successfully simulate atmospheric mesoscale flows. Consequently, they are not applied to the equations and the full complexity of the system (3.1), (3.3), (3.4), (3.8) is retained in a3m.

As mentioned above, using the local tangential (x, y, z) frame of reference is common practise in local atmospheric models. In this special frame of reference, Earth's angular velocity is $\underline{\Omega} = (0, \Omega\sin\varphi, \Omega\cos\varphi)^T$, where φ is the latitude. Often some

reference latitude, e.g. of the model domains centre, is used and thereby $\underline{\Omega}$ is made constant. This is a common approximation for simulations with a extent in North–South direction up to 1800 km. Following the analysis of Wippermann (1981), this approximation is even applicable for extents of up to 2500 km. For smaller-scale simulations, with a horizontal extent of 20 km or less, the Coriolis force may even be neglected entirely. For larger scales the coefficients can be linearised around a reference point resulting in the so-called β-plane approximation.

In its current stage of development, a3m uses a constant $\underline{\Omega}$ with respect to a user-specified reference latitude. Alternatively, the Coriolis force may be completely neglected as a run time option. As with all of the other terms and relations immediately dependent on the special choice of the (x, y, z) frame of reference, the Coriolis force is encapsulated in the implementation of a3m and cleanly separated from other code so that it can be exchanged for a more sophisticated treatment of the Coriolis force with changes in only a relatively small portion of the code.

3.1.6 Coordinate Transformations

Transformations to non-Cartesian coordinates are very common in atmospheric modelling. They are motivated mostly by the geometry of the model domains: the curvature of the Earth's surface becomes increasingly significant with increasing horizontal extent of the model domain and orography makes treatment of the lower boundary on a Cartesian grid difficult. Consequently, models for the mesoscale α and larger usually employ spherical coordinates. Additionally, various variations of vertical coordinates are used. Many mesoscale models, e.g. METRAS (Schlünzen, 1988) and ARPS (Xue *et al.*, 2000), employ a terrain-following vertical coordinate, the so-called η-system with

$$\eta = z_t \frac{z - z_s}{z_t - z_s},$$

where z_t is the height of the model domains upper boundary and z_s is the surface height. Note that due to $z_s = z_s(x, y)$, η depends on all three original variables x, y and z. Therefore, the differential operators are altered by transforming to a coordinate system like this, and the equations have a different and more complicated shape than in Cartesian coordinates. Other models use different vertical coordinates such as the pressure or the σ-system which is defined analogously to the η-system but for surface and domain top pressure instead of height (e.g. MM5, Grell *et al.*, 1995).

The use of curvilinear coordinates like the η-system is coupled to the discretisation methods employed by those models. All of the above models approximate their respective model equations by finite differences. Finite difference methods require a topologically regular computational grid: grid nodes are always on coordinate

lines, e.g. the grid nodes are specified by (x, y, η) with $x = i\Delta x$, $y = j\Delta y$ and $\eta = k\Delta\eta$. These coordinates simplify the treatment of the lower domain boundary which follows the topography and is given by the plane $\eta = 0$ in the curvilinear coordinates. In Cartesian coordinates finite difference methods have to approximate the terrain by steps in the grid (blocked cells) or the infinite viscosity approach.

In contrast to the above coordinate systems, the finite volume method introduced in Chapter 4 decouples the coordinates in which the *equations* are written from the construction of the computational *grid*. The grid can still match the orography for an easy treatment of the lower boundary so that nothing is to be gained from transforming the equations to more complicated coordinates like the η-system. Consequently, only Cartesian coordinates are used throughout this work. Transformations between different Cartesian frames of references are easily performed by translations and rotations. The Euler equations are invariant with respect to either, translations and rotations (e.g. Hirsch, 1988a; Sonar, 1997), so their shape is always the same as given by equations (3.1), (3.3) and (3.8).

Summarising, a3m predicts the mass density ρ, momentum density components ρu_i and the total energy density $\rho\mathsf{e}$ by means of the equations (3.1), (3.3), (3.4) and (3.8). No approximations other than the assumptions 1–7 introduced at the beginning of this section and the approximation to the Coriolis force discussed in Section 3.1.5 are made. This very un-specialised choice of a mathematical model ensures its broad validity and the potential of a3m for future extensions. But it should also be noticed that this choice of a mathematical model has a downside: it is 'too large' in the sense that it also includes processes and effects not of interest from the meteorological point of view. The most prominent of these are sound waves. This leads to some complications and restrictions which will be discussed in Section 4.3.

3.2 Resolved and Turbulent Scales

Atmospheric flows that obey the system of equations (3.1), (3.3),(3.4) and (3.8) are turbulent. This means that these flows contain motions on all spatial scales from the considered domain as a whole down to the Kolmogoroff scale, where motion is dissipated into heat. For the atmosphere this is usually in the order of a few millimetres (Stull, 1988). But the very small-scale motions are neither of meteorological interest, nor are they feasible to compute in a numerical simulation of atmospheric flows because the resolution required to resolve all motions is much too high. Following the filtering approach of Germano (1992), small-scale spatial variation can be removed from the equations by subjecting them to some sort of averaging procedure. This yields a new set of equations for the mean flow described by the mean flow variables $\overline{\rho}$, $\overline{\rho u_i}$ and $\overline{\rho\mathsf{e}}$. The equations for the mean flow contain some extra terms capturing

the effect of the removed small-scale flow perturbations on the mean variables that have no corresponding counterpart in the original un-filtered equations.

The filter separating the perturbations from the mean flow is not necessarily an averaging in physical space: it could also be conducted in phase space on the Fourier transformed equations instead. But in this work only spatial filters of the form

$$\overline{a}(\underline{x}) := \int_{\mathbb{R}^3} G(\underline{x} - \underline{x}')a(\underline{x}')\, d\underline{x}' \qquad \text{and} \tag{3.9a}$$

$$\widetilde{a}(\underline{x}) := \frac{1}{\overline{\rho}} \int_{\mathbb{R}^3} G(\underline{x} - \underline{x}')\rho(\underline{x}')a(\underline{x}')\, d\underline{x}' \tag{3.9b}$$

are used, where $a(\underline{x})$ is the field to filter and $G(\underline{x})$ is the so-called filter kernel which must satisfy

$$\int_{\mathbb{R}^3} G(\underline{x})\, d\underline{x} = 1. \tag{3.9c}$$

The integral $\int_X f(\underline{x})\, d\underline{x}$ denotes the integration of a field $f(\underline{x})$ over all points $\underline{x} \in X \subseteq \mathbb{R}^3$, so it is an integral over a region X in space. The residuals—the perturbation parts—corresponding to the filtered fields are given by

$$a' := a - \overline{a} \qquad \text{and} \qquad a'' := a - \widetilde{a}.$$

The finite volume discretisation to be introduced in Chapter 4 *is* such a spatial filter. Finite difference and finite element discretisations also implicitly filter in space (Rogallo and Monin, 1984; Germano, 1992). Any of these discretisations only loose their filtering properties with respect to turbulence if the spatial resolution is high enough to explicitly capture all the turbulent fluctuations, i.e. only within so-called direct numerical simulations. In that case the fluctuations a' and a'' vanish, of course.

Two filters are needed for the compressible equations instead of only one for incompressible or anelastic equations. Both filters, $\overline{\cdot}$ and $\widetilde{\cdot}$, are linear, i.e.

$$\overline{a+b} = \overline{a} + \overline{b} \qquad \widetilde{a+b} = \widetilde{a} + \widetilde{b} \qquad \overline{ca} = c\,\overline{a} \qquad \widetilde{ca} = c\,\widetilde{a},$$

for any fields $a(\underline{x})$ and $b(\underline{x})$ and constant c and the filter $\overline{\cdot}$ commutes with partial derivatives in space and time, i.e.

$$\overline{\partial_t a} = \partial_t \overline{a} \qquad \text{and} \qquad \overline{\partial_{x_i} a} = \partial_{x_i} \overline{a}.$$

Note that the two filters are related by $\overline{\rho a} = \overline{\rho}\,\widetilde{a}$ for some field $a(\underline{x})$ and the density ρ, e.g. $\overline{\rho u_i} = \overline{\rho}\,\widetilde{u}_i$.

The most suitable filter kernel to use in combination with finite volume methods (Chapter 4) is a simple box filter:

$$G(\underline{x}) = \begin{cases} 1/|\sigma| & \text{for } \underline{x} \in \sigma \\ 0 & \text{otherwise} \end{cases}$$

for some region σ around zero. This filter gives the moving average of a field $a(\underline{x})$ as the filtered variable $\overline{a}(\underline{x})$. Due to the close relation between filtering and discretisation (Rogallo and Monin, 1984) the width of the box σ is connected to the spacing of the grid on which the equations are to be discretised.

Unlike Reynolds averaging, the type of filtering used here gives different results for multiple applications of the filter and the filtered fluctuations do not vanish:

$$\overline{\overline{a}} \neq \overline{a}, \qquad \overline{a'} \neq 0, \qquad \widetilde{\widetilde{a}} \neq \widetilde{a}, \quad \text{and} \quad \widetilde{a''} \neq 0.$$

Density, velocity and specific energy are now decomposed as

$$\rho = \overline{\rho} + \rho', \qquad u_i = \widetilde{u_i} + v_i'', \quad \text{and} \quad \mathsf{e} = \widetilde{\mathsf{e}} + \mathsf{e}''.$$

Note that $\widetilde{u_i} = \overline{\rho u_i}/\overline{\rho}$ and $\widetilde{\mathsf{e}} = \overline{\rho \mathsf{e}}/\overline{\rho}$.

The set of equations for the mean flow variables $\overline{\rho}$, $\overline{\rho u_i}$ and $\overline{\rho \mathsf{e}}$ is gained by applying the averaging filter $\bar{\cdot}$ to equations (3.1), (3.3), (3.4) and (3.8):

$$\partial_t \overline{\rho} + \sum_{j=1}^{3} \partial_{x_j} (\overline{\rho u_j}) = 0 \tag{3.10a}$$

$$\begin{aligned} \partial_t (\overline{\rho u_i}) + \sum_{j=1}^{3} \partial_{x_j} \left(\overline{\rho u_i} \widetilde{u_j} + \delta_{ij} \overline{\mathsf{p}^{(1)}} \right) \\ = -\sum_{j=1}^{3} \sum_{k=1}^{3} 2\epsilon_{ijk} \Omega_j \overline{\rho u_k} - \overline{\rho^{(1)}} g_i - \sum_{j=1}^{3} \partial_{x_j} \tau_{ij} \end{aligned} \tag{3.10b}$$

$$\partial_t (\overline{\rho \mathsf{e}}) + \sum_{j=1}^{3} \partial_{x_j} \left((\overline{\rho \mathsf{e}} + \overline{\mathsf{p}}) \widetilde{u_j} \right) = -\sum_{j=1}^{3} \left(\overline{\rho u_j}\, g_j + \partial_{x_j} \tau_{\mathsf{e}j} \right) + \overline{S} \tag{3.10c}$$

$$\overline{\mathsf{p}} = (\gamma - 1) \left(\overline{\rho \mathsf{e}} - \frac{\overline{\rho} |\widetilde{\underline{u}}|^2}{2} - \overline{\rho} \mathsf{k} \right), \tag{3.10d}$$

where

$$\tau_{ij} := \overline{\rho u_i u_j} - \overline{\rho u_i} \widetilde{u_j} = \overline{\rho} (\widetilde{u_i u_j} - \widetilde{u_i} \widetilde{u_j}), \tag{3.11a}$$

$$\tau_{\mathsf{e}j} := \overline{\rho \mathsf{e} u_j} - \overline{\rho \mathsf{e}}\, \widetilde{u_j} + \overline{\mathsf{p} u_j} - \overline{\mathsf{p}}\, \widetilde{u_j} = \overline{\rho} (\widetilde{\mathsf{e} u_j} - \widetilde{\mathsf{e}}\, \widetilde{u_j}) + \overline{\mathsf{p} u_j} - \overline{\mathsf{p}}\, \widetilde{u_j}, \tag{3.11b}$$

and

$$\begin{aligned}\mathsf{k} &:= \frac{1}{2\overline{\rho}}\left(\overline{\rho|\underline{u}|^2} - \overline{\rho}|\widetilde{\underline{u}}|^2\right) = \frac{1}{2}\left(\widetilde{|\underline{u}^2|} - |\widetilde{\underline{u}}|^2\right) \\ &= \frac{1}{2}\sum_{i=1}^{3}(\widetilde{u_i u_i} - \widetilde{u}_i\widetilde{u}_i) = \frac{1}{2\overline{\rho}}\sum_{i=1}^{3}\tau_{ii}.\end{aligned} \tag{3.11c}$$

The system (3.10)–(3.11) is of the same shape as the un-filtered equations with some additional terms: the turbulent momentum fluxes τ_{ij}, the turbulent energy fluxes $\tau_{\mathsf{e}j}$ and the specific sub-grid-scale or turbulent kinetic energy k. According to assumption 3 introduced in Section 3.1.1, Earth's angular velocity $\underline{\Omega}$ and gravitational acceleration $\underline{g}$ do not to vary significantly within the width of the filter. Therefore they can be excluded from the averaging, i.e. set constant during integration in (3.9). The sub-grid-scale terms τ_{ij} and $\tau_{\mathsf{e}j}$ occurring in the new equations for the variables $\overline{\rho}$, $\overline{\rho u_i} = \overline{\rho}\,\widetilde{u}_i$ and $\overline{\rho\mathsf{e}} = \overline{\rho}\widetilde{\mathsf{e}}$ are modelled in Section 3.3.

Note that with filters satisfying the Reynolds rules, $\overline{\overline{a}} = \overline{a}$, $\overline{a'} = 0$, $\widetilde{\widetilde{a}} = \widetilde{a}$ and $\widetilde{a''} = 0$, system (3.10) reduces to the normal compressible Favre-averaged Euler equations to be found in many fluid dynamics books (e.g. Hirsch, 1988a).

3.3 Sub-grid-scale Parameterisations

The system of equations (3.10) is no longer closed since the turbulence terms τ_{ij} and $\tau_{\mathsf{e}j}$ cannot be expressed analytically in terms of the grid-scale variables $\overline{\rho}$, $\overline{\rho u_i}$ and $\overline{\rho\mathsf{e}}$. An additional model or parameterisation of the effect of the turbulent fluctuations on the grid-scale variables is necessary in order to gain a closed system of equations again, which then can be solved numerically.

In a3m the turbulence terms τ_{ij} and $\tau_{\mathsf{e}j}$ are parameterised by the so-called first order closure (Mellor and Yamada, 1974) which is very common in atmospheric modelling. The feedback of the sub-grid-scale turbulence on the resolved scale flow is modelled as a diffusional process with an exchange coefficient that is usually a function of height and the predicted variables, especially of the velocity shear and stratification. It is therefore potentially varying in space and time. The approach is derived from the observation that the mixing is very much enhanced in turbulent flows and resembles a highly diffusive flow. Hence, it is also referred to as eddy diffusion. With this approach the turbulence terms in equations (3.10b) and (3.10c) are set to

$$\tau_{ij} = -K_{ij}\overline{\rho}(\partial_{x_i}\widetilde{u}_j + \partial_{x_j}\widetilde{u}_i) + 2/3\,\delta_{ij}\,\overline{\rho}\mathsf{k} \tag{3.12a}$$

$$\tau_{\mathsf{e}j} = -K_{\mathsf{e}j}\overline{\rho}(\partial_{x_j}\widetilde{\mathsf{e}}) \tag{3.12b}$$

for $i, j = 1, 2, 3$. Different turbulence parameterisations can now be classified by the method used to determine the exchange coefficients K_{ij} and $K_{\mathsf{e}j}$. A classification is given by Mellor and Yamada (1974).

The atmosphere usually displays an anisotropic behaviour. There are two distinct directions to be found in the wind, temperature, pressure and concentration fields: horizontal and vertical. This has an immediate impact on turbulence which is in turn anisotropic and the effect in the vertical direction is usually considered much more important than in horizontal direction. Most turbulence parameterisations assume isentropic turbulence. They consider vertical gradients only and assume horizontal homogeneity and statistical stationarity. Therefore, it is useful to consider the turbulence parameterisation in the local tangential (x, y, z) frame of reference where the x axis points East, the y axis points North and the z axis points upwards, i.e. z is height.

The tensors of exchange coefficients (K_{ij}) and $(K_{\mathsf{e}j})$ in the (x_1, x_2, x_3) frame are obtained from the tensors $K^{(xyz)}$ and $K_{\mathsf{e}}^{(xyz)}$ in the (x, y, z) frame by

$$(K_{ij}) = T_{\text{vert}}^{-1} K^{(xyz)} T_{\text{vert}} \qquad \text{and} \tag{3.13a}$$

$$(K_{\mathsf{e}j}) = T_{\text{vert}}^{-1} K_{\mathsf{e}}^{(xyz)}, \tag{3.13b}$$

where the transformation is determined by the rotation matrix

$$T_{\text{vert}} = \begin{cases} \begin{pmatrix} -\frac{k_2}{\sqrt{k_1^2+k_2^2}} & \frac{k_1}{\sqrt{k_1^2+k_2^2}} & 0 \\ -\frac{k_1}{\sqrt{k_1^2+k_2^2}} k_3 & -\frac{k_2}{\sqrt{k_1^2+k_2^2}} k_3 & \sqrt{k_1^2+k_2^2} \\ k_1 & k_2 & k_3 \end{pmatrix} & \text{for } \sqrt{k_1^2+k_2^2} \neq 0 \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & k_3 \end{pmatrix} & \text{for } \sqrt{k_1^2+k_2^2} = 0 \\ & \Leftrightarrow k_3 = \pm 1. \end{cases}$$

The rotation by T_{vert} aligns the z-coordinate of the rotated frame of reference with the vertical unit vector $\underline{k} = (k_1, k_2, k_3)^T$ expressed in the (x_1, x_2, x_3) frame. The rotation is reversed by $T_{\text{vert}}^{-1} = T_{\text{vert}}^T$.

In the (x, y, z) frame of reference the tensor of momentum exchange coefficients can now be approximated by an exchange coefficient K_{vert} for the vertical direction and a second coefficient K_{hor} for the horizontal direction. The tensor of coefficients is then given by

$$K_{xz} = K_{zx} = K_{yz} = K_{zy} = K_{zz} = K_{\text{vert}} \quad \text{and}$$
$$K_{xx} = K_{xy} = K_{yx} = K_{yy} = K_{\text{hor}}.$$

In a3m the vertical momentum exchange coefficient is obtained from the mixing length approach. It is assumed proportional to a typical length scale l that turbulent fluctuations act on and a typical fluctuation velocity. The fluctuation velocity is in turn proportional to the absolute vertical shear of the wind field and the mixing length. This gives

$$K_{\text{vert}} = f_{\text{s}}(\text{Ri})\, l^2 \sqrt{(\partial_z u_1)^2 + (\partial_z u_2)^2 + (\partial_z u_3)^2}, \tag{3.14}$$

where $f_{\text{s}}(\text{Ri})$ is a function of the Richardson number

$$\text{Ri} = \frac{g}{\Theta} \frac{\partial_z \Theta}{(\partial_z u_1)^2 + (\partial_z u_2)^2 + (\partial_z u_3)^2}$$

to account for stability. Since turbulence is far from the focus of this work, the stability function is set to $f_{\text{s}}(\text{Ri}) \equiv 1$ which is the correct value for neutral stratification. Hence, turbulent mixing is underestimated in the unstable and overestimated in the stable case. The mixing length l is calculated following Blackadar (1962)

$$l = \frac{\kappa z}{1 + \kappa z / l_0}, \tag{3.15}$$

where κ is the von Kármán constant and l_0 is an upper bound for l. In a3m $\kappa = 0.4$ is used. Many different approaches to specify the length l_0 can be found in mesoscale atmospheric models. Often it is parameterised depending on the friction velocity, the Coriolis parameter and/or the mesh size of the computational grid (Schlünzen, 1994, and references therein). Typical values seem to be in the order of some tens of metres. For the sake of simplicity, the constant value of 30 m is used within this work. Note that for neutral stratification and horizontal homogeneity the much more complicated turbulent kinetic energy/dissipation turbulence parameterisation—also known as the k-ϵ model—reduces to the exchange coefficient given by equations (3.14) and (3.15).

Little is known about the horizontal turbulent mixing in the atmosphere. In the microscale atmospheric turbulence is isotropic so that classical approaches can be used. But for the anisotropic case in the larger scales only vertical turbulent mixing for horizontal homogenous conditions is sufficiently understood and backed up by experimental data. Consequently, no generally accepted approach for horizontal mixing by turbulence is available up to now. In a3m the horizontal exchange coefficient is set to be proportional to the vertical exchange coefficient,

$$K_{\text{hor}} = a K_{\text{vert}}, \tag{3.16}$$

where the factor a accounts for the local anisotropy of the computational grid and thereby the filters (3.9).

It is not possible to gain the specific turbulent kinetic energy k with this approach so that it has to be neglected in equations (3.10d) and (3.12a).

The exchange coefficient for energy is related to that for momentum by

$$\mathrm{Pr}_\mathsf{e} = K_\mathrm{vert}/K_\mathsf{evert}, \tag{3.17}$$

where Pr_e is an analogy for the normal Prandtl number defined for the turbulent heat flux. In the simulations presented in this work $\mathrm{Pr}_\mathsf{e} = 1$ is used. The horizontal turbulent energy flux is treated analogously to that of momentum, i.e.

$$K_\mathsf{ehor} = aK_\mathsf{evert}. \tag{3.18}$$

To compare a3m simulation results with published results obtained from different models, an option to neglect turbulence altogether is also provided:

$$K_{ij} = K_{\mathsf{e}j} = 0.$$

Within some tens of metres of a solid surface the wind speed and also scalar quantities such as temperature typically display a logarithmic profile (e.g. Stull, 1988) with strong gradients at the surface in the surface normal direction. To simulate these layers directly, a very high and anisotropically resolving grid would be necessary which requires a too high amount of computing resources. For this reason, atmospheric models usually compute turbulent fluxes near solid surfaces—most notably the ground—with another parameterisation based on the similarity theory of Monin and Obukhov (1954). However, at the stage of development of a3m presented in this work, the turbulent mixing is computed in the same manner throughout all of the computational domain including all boundaries which is more consistent with the treatment of solid boundaries as described in Section 4.1.5.

3.4 Extensibility of the Mathematical Model

The mathematical model developed so far does not include moisture, which is probably the most severe discrepancy to the real atmosphere. The water content in the air and especially phase changes of water play a crucial role in the atmospheric energy budget. The transport of other substances such as pollutants by atmospheric flows is also of great practical importance.

In the case of a simple tracer substance s with the mass mixing ratio c_s, the system of conservation laws (3.1), (3.3) and (3.8) is expanded by a (filtered) transport equation for its concentration:

$$\partial_t(\overline{\rho c_s}) + \sum_{j=1}^{3} \partial_{x_j}(\overline{\rho c_s}\widetilde{u_j}) = S_s - \sum_{j=1}^{3} \tau_{sj}$$

where S_s is a source term for the substance s and $\tau_{sj} := \overline{\rho c_s u_j} - \overline{\rho c_s}\widetilde{u_j} = \overline{\rho}(\widetilde{c_s u_j} - \widetilde{c_s}\widetilde{u_j})$ is the turbulent flux of tracer s modelled as

$$\tau_{sj} = -K_{sj}\,\overline{\rho}\,\partial_{x_j}\widetilde{c_s}, \qquad j = 1, 2, 3.$$

The exchange coefficients K_{sj} are set equal to those for energy because turbulent mixing of tracers is even less well understood than that of heat. In the case of a passive tracer, the source term S_s includes emission, deposition and, in case of a chemically active substance, also production and consumption by chemical reactions. Assuming that the tracer concentration c_s—the ratio of the partial density ρ_s and the total density of the mixture ρ—is small, the tracers fraction of the mixtures mass is negligible with respect to the physico-chemical properties of the mixture as a whole, and no further modifications of the model system are necessary.

In the case of moisture, at least two equations are needed for the mixing ratios water vapour q_v and liquid water q_l:

$$\partial_t(\overline{\rho q_v}) + \sum_{j=1}^{3} \partial_{x_j}(\overline{\rho q_v}\widetilde{u_j}) = -\overline{\rho} S_{\text{cond}} - \sum_{j=1}^{3} \partial_{x_j}\tau_{vj} \qquad \text{and}$$
$$\partial_t(\overline{\rho q_l}) + \sum_{j=1}^{3} \partial_{x_j}(\overline{\rho q_l}\widetilde{u_j}) = +\overline{\rho} S_{\text{cond}} - \sum_{j=1}^{3} \partial_{x_j}\tau_{lj},$$

where S_{cond} is the conversion rate of water vapour to liquid water by condensation and by evaporation in the opposite direction. The turbulent fluxes of vapour and liquid water, $\tau_{vj} := \overline{\rho c_v u_j} - \overline{\rho c_v}\widetilde{u_j} = \overline{\rho}(\widetilde{c_v u_j} - \widetilde{c_v}\widetilde{u_j})$ and $\tau_{lj} := \overline{\rho c_l u_j} - \overline{\rho c_l}\widetilde{u_j} = \overline{\rho}(\widetilde{c_l u_j} - \widetilde{c_l}\widetilde{u_j})$, are modelled as

$$\begin{aligned} \tau_{vj} &= -K_{vj}\,\overline{\rho}\,\partial_{x_k}\widetilde{c_v} \quad \text{and} \\ \tau_{lj} &= -K_{lj}\,\overline{\rho}\,\partial_{x_k}\widetilde{c_l}, \end{aligned} \qquad j = 1, 2, 3.$$

The exchange coefficients K_{vj} and K_{lj} are again set equal to those for energy.

The condensation process also is a source of heat. The contribution to the source term of the energy equation (3.3) is $l_v \rho S_{\text{cond}}$, where l_v is the specific latent heat of water vapour. Additionally, the equation of state (3.4) changes since now air has to be treated as a mixture of different gases, where only water vapour is separated from the dry air used in the rest of this chapter. As discussed for assumption 6 in Section 3.1.1 each of these gases—dry air and water vapour—is assumed to be ideal, i.e. the ideal gas law holds for each separately:

$$\mathsf{p} = \rho_a R_a \mathsf{T} \quad \text{and} \quad \mathsf{p} = \rho_v R_v \mathsf{T}$$

where ρ_a and ρ_v are the partial densities and R_a and R_v the specific gas constant of dry air and water vapour, respectively. The equation of state for the mixture as a whole is then

$$\mathsf{p} = (\rho_a R_a + \rho_v R_v)\mathsf{T} = \rho R_a \left(1 + \frac{R_v - R_a}{R_a} q_v\right) \mathsf{T} \tag{3.19}$$

The content of liquid water has been neglected since its influence is of minor importance for the equation of state (Doms and Herbert, 1985).

Chapter 4

Numerical Methods and Discretisation

This chapter describes the discretisation of the mathematical model introduced in Chapter 3 and the numerical methods used for this purpose. First the basic spatial discretisation of advection, turbulence and source terms using the method of finite volumes is presented and is subsequently extended to block-structured locally refined grids. Discretisation in time completes the discrete model and thereby the second step in the development process of a3m (Figure 1.3). Finally, the impact of extensions to the mathematical model (Section 3.4) on the discretisation are discussed.

4.1 Basic Spatial Discretisation

The discretisation in space is achieved by the method of finite volumes (FV). The general idea of FV methods is to substitute a partial differential equation for a function of space and time by a system of coupled *evolution equations* predicting the average values on small volumes. The evolution of the averages on the single volumes is calculated from the exchange between neighbouring volumes and source and sinks terms within each volume. A solid and in-depth introduction to FV schemes is given by Sonar (1997) and Abgrall *et al.* (1999).

To improve readability, the over-bar and tilde introduced by the scale decomposition (Section 3.2) are dropped for the rest of this work. All occurrences of ρ, $\underline{u} = (u_1, u_2, u_3)^T$, e and p refer to the corresponding filtered variables $\overline{\rho}$, $\widetilde{\underline{u}} = (\widetilde{u_1}, \widetilde{u_2}, \widetilde{u_3})^T$, $\widetilde{\mathsf{e}}$ and $\overline{\mathsf{p}}$ unless explicitly stated otherwise.

The system of equations (3.10) can be re-written in the more compact form

$$\partial_t U + \sum_{l=1}^{3} \partial_{x_l} F_l(U) = -\sum_{l=1}^{3} \partial_{x_l} G_l(U) + Q(U, \underline{x}, t) \tag{4.1}$$

with

$$U = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho \mathsf{e} \end{pmatrix}, \quad F_l(U) = \begin{pmatrix} \rho u_l \\ \rho u_1 u_l + \delta_{1l} \mathsf{p}^{(1)} \\ \rho u_2 u_l + \delta_{2l} \mathsf{p}^{(1)} \\ \rho u_3 u_l + \delta_{3l} \mathsf{p}^{(1)} \\ (\rho \mathsf{e} + \mathsf{p}) u_l \end{pmatrix}, \quad G_l(U) = \begin{pmatrix} 0 \\ \tau_{1l} \\ \tau_{2l} \\ \tau_{3l} \\ \tau_{\mathsf{e}l} \end{pmatrix},$$

and

$$Q(U, \underline{x}, t) = \begin{pmatrix} 0 \\ -(2\Omega_2 \rho u_3 - 2\Omega_3 \rho u_2) - \rho^{(1)} g_1 \\ -(2\Omega_3 \rho u_1 - 2\Omega_1 \rho u_3) - \rho^{(1)} g_2 \\ -(2\Omega_1 \rho u_2 - 2\Omega_2 \rho u_1) - \rho^{(1)} g_3 \\ -\rho \underline{u} \underline{g} + S \end{pmatrix}.$$

On a domain $D \subset \mathbb{R}^d$ for d spatial dimensions and with suitable initial and boundary conditions equation (4.1) can be re-written in an equivalent integral formulation: for any control volume $\sigma \subseteq D$ the unknown U must satisfy

$$\partial_t \int_\sigma U \, d\underline{x} + \sum_{l=1}^{3} \int_{\partial\sigma} F_l(U) n_l \, d\underline{x} = -\sum_{l=1}^{3} \int_{\partial\sigma} G_l(U) n_l \, d\underline{x} + \int_\sigma Q(U, \underline{x}, t) \, d\underline{x}, \qquad (4.2)$$

where $\partial\sigma$ designates the boundary of the volume σ and $\underline{n} = (n_1, n_2, n_3)^T$ is the outward pointing normal vector on the surface $\partial\sigma$. Note that by employing the Gauß integral theorem, all spatial derivatives of equation (4.1) are removed in favour of integrals of the fluxes F_l and G_l over the volume boundary $\partial\sigma$. This is an important property of equation (4.2) since its solutions are permitted to be discontinuous while those of equation (4.1) are not. A solution U of the integral equation (4.2) for any control volume $\sigma \subseteq D$ is a—in the mathematical sense—weak solution of the original problem (4.1). Considering that for the atmospheric case discontinuities can occur in the form of fronts, constructing a numerical method from the integral formulation and thereby approximating weak solutions seems appropriate.

To derive a numerical method, the domain D is now divided into a collection of subsets or *cells* $T_1, T_2, \ldots, T_N \subset \overline{D}$ which

1. are closed and simply connected,
2. have an actual interior: $\overset{\circ}{T}_i \neq \emptyset$,
3. cover the whole domain: $\overline{D} = \bigcup_{i=1}^{N} T_i$,
4. don't overlap except for boundaries: $\overset{\circ}{T}_i \cap \overset{\circ}{T}_j = \emptyset$ for $i \neq j$ and
5. have boundaries consisting of even polygons, its *faces*.

Requirement 2 above means that the cells have to be of the same dimensionality as the domain D itself, e.g. for three spatial dimensions cells really are finite volumes not mere surfaces.

Two cells are called *neighbours* if they share a face. The set of neighbours of a cell T_i—also called *von Neumann neighbourhood*—is thus given by

$$\mathcal{N}(i) = \{j \mid T_i \cap T_j \text{ is a face of } T_i\}.$$

For this work cells are straights in one, quadrangles in two and hexahedra in three spatial dimensions. Faces are points in one, straights in two and quadrangles in three spatial dimensions. Nevertheless, all cells are said to have a *volume* and faces a *surface area*. All computational meshes within this work are structured, i.e. each cell can be uniquely identified by an index tuple from $\mathbb{N}^d$ and each face by an index tuple from $\mathbb{N}^d \times \{1, \ldots, d\}$ for d spatial dimensions. Neighbour relations between cells and faces are computable from such indices alone. Figure 4.1 shows a possible grid configuration in two spatial dimensions.

The finite volume approach aims at constructing evolution equations for the average values U_i of the unknown U on each cell T_i. U_i is defined by

$$U_i := \frac{1}{|T_i|} \int_{T_i} U \, d\underline{x}, \tag{4.3}$$

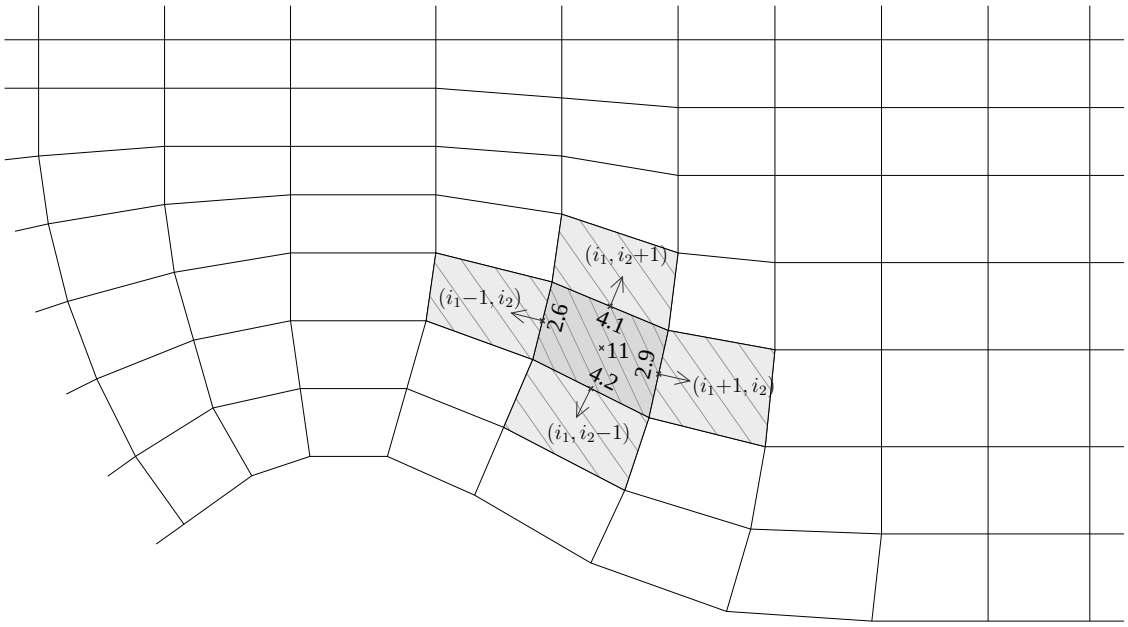

Figure 4.1: A part of a structured quadrangular grid in 2d.
Cells can be identified with an index pair (i_1, i_2) and its von Neumann neighbours are the cells $(i_1 \pm 1, i_2)$ and $(i_1, i_2 \pm 1)$. For one cell its centre of gravity and volume as well as the centres, surface areas and normal vectors of its bounding faces are shown. Its four von Neumann neighbours are shaded.

where $|T_i|$ denotes the volume of cell T_i. To this end the set of cells T_i with the above properties is used as a special choice of control volumes. Analogously to deriving the integral formulation (4.2), the averaging is applied to equation (4.1) and the averaged flux divergences on the cell are substituted by averages of the fluxes over the cell boundary themselves:

$$\begin{aligned}\frac{d}{dt}U_i(t) + \frac{1}{|T_i|}\int\limits_{\partial T_i}\sum_{l=1}^{3} F_l(U)n_l \, d\underline{x} \\ = -\frac{1}{|T_i|}\int\limits_{\partial T_i}\sum_{l=1}^{3} G_l(U)n_l \, d\underline{x} + \frac{1}{|T_i|}\int\limits_{T_i} Q(U,\underline{x},t)\, d\underline{x}.\end{aligned}$$

In contrast to its counterpart in equation (4.1), the derivative in time of the cell average values is not partial. By definition (4.3) the average U_i of U on cell T_i does not vary: it has the same value on all of the cell on which it is defined. Hence, $\partial_{x_j} U_i = 0$ and $\partial_t U_i = dU_i/dt$. So the cell averages are not a function of the spatial coordinates anymore but a function of time alone. The spatial distribution of average values is immediately bound to that of the cells.

The integral over the cell boundary ∂T_i can be split into integrals over the single faces $S_{ij} := T_i \cap T_j$ that cell T_i shares with its neighbours T_j, $j \in \mathcal{N}(i)$. This gives the desired prognostic equation

$$\begin{aligned}\frac{d}{dt}U_i(t) + \frac{1}{|T_i|}\sum_{j\in\mathcal{N}(i)}\int\limits_{S_{ij}}\sum_{l=1}^{3} F_l(U)n_{ij_l} \, d\underline{x} \\ = -\frac{1}{|T_i|}\sum_{j\in\mathcal{N}(i)}\int\limits_{S_{ij}}\sum_{l=1}^{3} G_l(U)n_{ij_l} \, d\underline{x} + \frac{1}{|T_i|}\int\limits_{T_i} Q(U,\underline{x},t)\, d\underline{x}\end{aligned} \tag{4.4}$$

for each cell average U_i on cell T_i, where $\underline{n}_{ij} = (n_{ij_1}, n_{ij_2}, n_{ij_3})^T$ is the outward pointing normal vector on the face S_{ij} as illustrated by Figure 4.1.

The set of cell averages U_i constitutes a piecewise constant approximation of the unknown U. Figure 4.2 shows a sketch of a curve in one spatial dimension and its approximation by averages on small intervals. Also note that since (4.4) is approximating weak solutions, it contains no spatial derivatives anymore: the evolution of the cell averages in time is expressed entirely by the integral advective and turbulent fluxes over the cell boundaries and sources within the cells.

The integrals over the faces S_{ij} are approximated numerically by the Gauß quadrature:

$$\int\limits_{S_{ij}} f(\underline{x})\, d\underline{x} = |S_{ij}| \sum_{k=1}^{n_q} w_k f(\underline{x}_k) + \mathcal{O}(h^{2n_q})$$

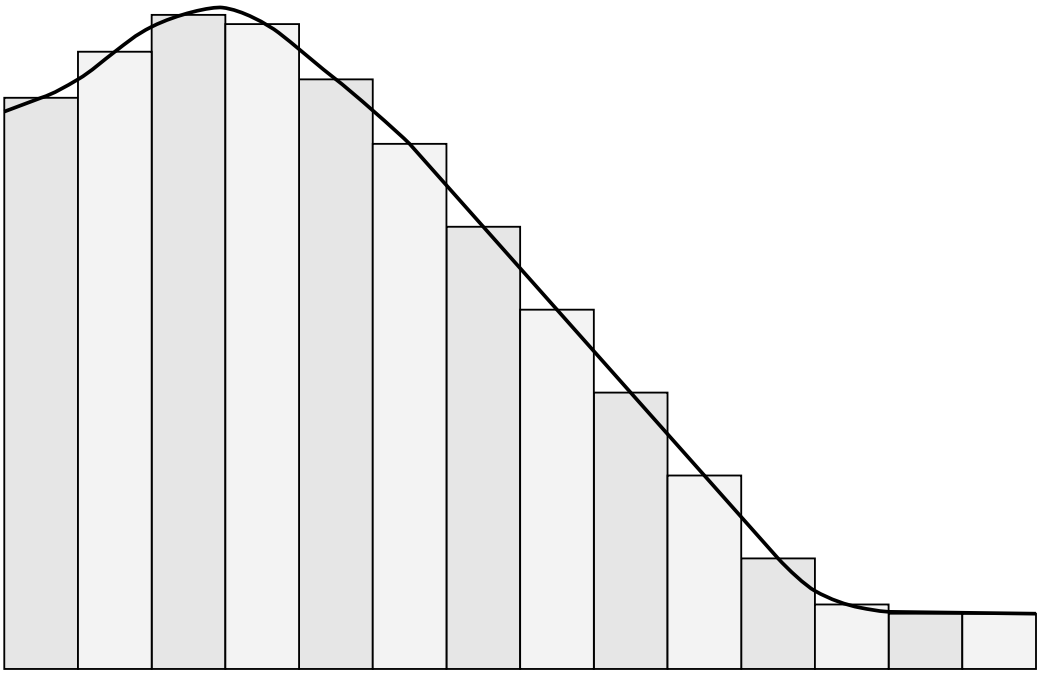

Figure 4.2: Piecewise constant approximation of a curve by cell averages.

for a smooth function $f(\underline{x})$ of space, where h is a measure of the grid resolution such as the longest edge of a cell. The weights w_k and sampling points $\underline{x}_k$ are given by e.g. Press *et al.* (1992). For the purposes of this work, an approximation of second order in space, i.e. $n_q = 1$, is sufficient. Then the weight is $w_1 = 1$ and the function has to be evaluated only at $\underline{x}_1 = \underline{c}_{ij}$, the faces centre of gravity:

$$\int_{S_{ij}} f(\underline{x})\, d\underline{x} = |S_{ij}| f(\underline{c}_{ij}) + \mathcal{O}(h^2). \tag{4.5}$$

This quadrature is also known as the midpoint rule. A similar approach is taken for approximating the average source term (Section 4.1.3).

Equation (4.4) still contains the unknown U as arguments to the flux and source functions, while only the cell averages U_i are available. So the tendency of cell averages cannot be computed directly by means of (4.4). In the following, the full discretisation of the single terms for advective and turbulent fluxes as well as sources using only cell averages is described.

4.1.1 Advective Fluxes

In the piecewise constant approximation by cell averages (Figure 4.2) two different values U_i and U_j touch discontinuously at the face S_{ij} thus forming a Riemann problem. Godunov proposed a procedure providing the solution of the Riemann problem that he then projected back to the computational grid. Exactly solving the Riemann problem is costly though. Merely to gain new cell averages, it is more efficient to use a *numerical flux function* $H(U_i, U_j; \underline{n}_{ij})$ and thereby combine the solution of the Riemann problem and the projection to the Eulerian grid into a single step. The numerical flux function has to be consistent with the fluxes F_l, i.e.

$$H(U, U; \underline{n}) = \sum_{l=1}^{3} F_l(U) n_l.$$

Inserting the numerical flux function, applying the midpoint rule (4.5) as described above to the advective flux term of (4.4) and substituting the cell averages as arguments to the numerical flux function yields

$$\begin{aligned} \int_{S_{ij}} \sum_{l=1}^{3} F_l(U(\underline{x})) n_{ij_l} \, d\underline{x} &= \int_{S_{ij}} H(U(\underline{x}), U(\underline{x}); \underline{n}_{ij}) \, d\underline{x} \\ &= |S_{ij}| H(U(\underline{c}_{ij}), U(\underline{c}_{ij}); \underline{n}_{ij}) + \mathcal{O}(h^2) \\ &= |S_{ij}| H(U_i, U_j; \underline{n}_{ij}) + \mathcal{O}(h). \end{aligned} \tag{4.6}$$

This kind of Riemann problem is inherently one-dimensional: with a transformation to a $(\hat{x}_1, \hat{x}_2, \hat{x}_3)$ frame of reference with $\hat{x}_1$ in the direction of the normal vector $\underline{n}_{ij}$, only a discontinuity in $\hat{x}_1$ direction remains while the values are constant in $\hat{x}_2$ and $\hat{x}_3$ direction. So in the numerical scheme described here, the advective flux is computed from the cell average values $\hat{U}_i$ and $\hat{U}_j$ in the rotated coordinates and subsequently rotated back to the original frame of reference. This procedure employs the rotational invariance of the Euler equations' advective fluxes F_l, i.e. the fact that they yield the same results independently of the orientation of the frame of reference or, more formally,

$$\sum_{l=1}^{3} F_l(U) n_l = T_{\underline{n}}^{-1} F_1(T_{\underline{n}} U). \tag{4.7a}$$

The transformation matrix $T_{\underline{n}}$ for the normal vector $\underline{n} = (n_1, n_2, n_3)^T = (\sin\vartheta \cos\varphi,$

$\sin\vartheta\sin\varphi, \cos\vartheta)^T$ is given by

$$T_{\underline{n}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \sin\vartheta\cos\varphi & \sin\vartheta\sin\varphi & \cos\vartheta & 0 \\ 0 & -\sin\varphi & \cos\varphi & 0 & 0 \\ 0 & -\cos\vartheta\cos\varphi & -\cos\vartheta\sin\varphi & \sin\vartheta & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{cases} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & n_1 & n_2 & n_3 & 0 \\ 0 & -\frac{n_2}{\sqrt{n_1^2+n_2^2}} & \frac{n_1}{\sqrt{n_1^2+n_2^2}} & 0 & 0 \\ 0 & -\frac{n_1}{\sqrt{n_1^2+n_2^2}}n_3 & -\frac{n_2}{\sqrt{n_1^2+n_2^2}}n_3 & \sqrt{n_1^2+n_2^2} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} & \sqrt{n_1^2+n_2^2} \neq 0 \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & n_3 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -n_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} & \sqrt{n_1^2+n_2^2} = 0 \\ & \Leftrightarrow n_3 = \pm 1 \end{cases} \tag{4.7b}$$

and its inverse by the transpose, $T^{-1} = T^T$. This procedure simplifies the numerical treatment of the advective fluxes considerably: the concrete method for calculating the flux is now independent of the spatial dimension of the problem to simulate since it is always reduced to a locally one-dimensional problem by the rotation.

4.1.1.1 Numerical Flux Function

In this work, the approximate Riemann solver of Osher and Solomon (1982) is used as the numerical flux function. It has an upwind-character that takes the whole characteristic of the system of equations (4.1) into account. The classical method usually referred to as 'upwind' or 'upstream' only considers the flow velocity to determine what is 'upstream' and what is 'downstream'. That works well for either scalar hyperbolic equations or for systems of linear hyperbolic equations. In those cases, information is transported by a single velocity only. But for non-linear systems of hyperbolic conservation laws, such as (4.1), information is propagated by several different velocities. These velocities are the phase speeds of the simple wave solutions to the equations. They are given by the eigenvalues of the Jacobian of the flux function. For the Euler equations these are $|\underline{u}| + \mathsf{c}$, $|\underline{u}|$ (occurring three times) and $|\underline{u}| - \mathsf{c}$, where c is the speed of sound, associated with the simple wave solutions

of the Euler equations: shock waves, contact discontinuities and expansion waves also known as rarefaction waves.

For strictly sub-sonic flow velocities (assumption 2 in Section 3.1.1), i.e. $|\underline{u}| < \mathsf{c} = \sqrt{\gamma \mathsf{p}/\rho}$ everywhere and to all times, the numerical flux of Osher and Solomon is determined by one of two possible states U_a or U_b:

$$H^{\mathrm{OS}}(U_i, U_j; \underline{n}_{ij}) = \begin{cases} T_{\underline{n}_{ij}}^{-1} F_1(U_a) & \text{if } u_f \geq 0 \\ T_{\underline{n}_{ij}}^{-1} F_1(U_b) & \text{if } u_f < 0, \end{cases} \tag{4.8a}$$

where

$$u_f = \frac{q\hat{u}_{i_1}/\mathsf{c}_i + \hat{u}_{j_1}/\mathsf{c}_j + 2(q-1)/(\gamma-1)}{q/\mathsf{c}_i + 1/\mathsf{c}_j}, \tag{4.8b}$$

$q := (\mathsf{p}_i/\mathsf{p}_j)^\alpha$ and $\alpha := (\gamma-1)/2\gamma = R/2c_\mathsf{p}$. Note that the velocity u_f is neither that of one of the actual states in the two cells nor the mean thereof. It depends on the complete states in both cells and not only the flow speed. That way pressure information can still travel freely in both directions, also opposing the flow.

The states U_a and U_b are constructed from the eigenvectors and eigenvalues of the Jacobian of the flux function and the Riemann invariants of the Euler equations. They are given by

$$\begin{aligned} \rho_a &= \hat{\rho}_i \left(\frac{\mathsf{p}_f}{\hat{\mathsf{p}}_i}\right)^{1/\gamma} & \rho_b &= \hat{\rho}_j \left(\frac{\mathsf{p}_f}{\hat{\mathsf{p}}_j}\right)^{1/\gamma} \\ u_{a_1} &= u_f & u_{b_1} &= u_f \\ u_{a_2} &= \hat{u}_{i_2} & u_{b_2} &= \hat{u}_{j_2} \\ u_{a_3} &= \hat{u}_{i_3} & u_{b_3} &= \hat{u}_{j_3} \\ \mathsf{e}_a &= \frac{\mathsf{p}_f}{\rho_a(\gamma-1)} + \frac{|\underline{u}_a|^2}{2} & \mathsf{e}_b &= \frac{\mathsf{p}_f}{\rho_b(\gamma-1)} + \frac{|\underline{u}_b|^2}{2} \end{aligned} \tag{4.8c}$$

with the averages in rotated coordinates $\hat{U}_i := T_{\underline{n}_{ij}} U_i$ and $\hat{U}_j := T_{\underline{n}_{ij}} U_j$, where $T_{\underline{n}_{ij}}$ is given by (4.7b). The pressure p_f of the states U_a and U_b is

$$\mathsf{p}_f = \left(\frac{\mathsf{c}_i + \mathsf{c}_j - (\hat{u}_{j_1} - \hat{u}_{i_1})(\gamma-1)/2}{\mathsf{c}_i/\mathsf{p}_i^\alpha + \mathsf{c}_j/\mathsf{p}_j^\alpha} \right)^{1/\alpha}. \tag{4.8d}$$

A brief derivation of these relations can be found in Appendix A.

The advective flux approximation (4.6)–(4.8) has many desirable properties. Most importantly, it is monotonic as long as certain stability conditions for the time integration are obeyed. Thus, approximate solutions gained with the flux function (4.8) converge to the physically relevant solution of the integral formulation (4.2) of the

Euler equations for $h \to 0$. It is monotonicity-preserving so that no artificial oscillations are introduced into the solution. Also, the method is positive definite, i.e. variables that are positive at the beginning of a simulation will always stay positive. But it also has one major drawback: it is only of first order accuracy in space. That is the consequence of directly using the cell averages U_i and U_j as arguments to the numerical flux function, i.e. of the piecewise constant approximation (Hirsch, 1988b). The unknown U is approximated with second order accuracy only in the respective cells centre of gravity but with merely first order accuracy in all other places, including the face (Sonar, 1997).

Of course, the accuracy of simulation results can be improved by increasing the resolution of the computational grid. But this is not very effective with a first order method since it converges linearly to the real solution with decreasing grid spacing. At least, convergence rates can be expected to be better with a higher-order method. In the context of locally refined grids (Section 4.2) this constitutes a good argument for a higher-order scheme.

Also, first order approximations are very diffusive and smear extrema and especially steep gradients. With the focus of this work on accurately resolving fronts and other small-scale features, such a highly diffusive scheme is not acceptable. To reduce numerical diffusion, the order of approximation must be increased.

4.1.1.2 Higher-Order Methods: Polynomial Recovery

Many approaches to reduce the numerical diffusion have been proposed and applied in atmospheric modelling. Among them the piecewise parabolic method (PPM) (Colella and Woodward, 1984; Carpenter *et al.*, 1990) and approaches like the flux-corrected transport (FCT) method of Boris and Book (1973) as well as positive definite schemes for linear advection equations such as the anti-diffusive flux correction by Smolarkiewicz (1983) or the flux limiter of Bott (1989) which were specifically designed for meteorological models. The latter schemes are limited to pure transport of scalar variables by the flow, e.g. temperature in atmospheric flow models, or other purely advective processes, e.g. modelling aerosol growth (von Salzen and Schlünzen, 1999a).

The most common method for discretising the transport term of the momentum equations used in atmospheric models, e.g. METRAS, ARPS (Xue *et al.*, 2000), MM5 (Grell *et al.*, 1995) and others, are centred finite differences. These are known to introduce artificial non-physical oscillations into the solution (e.g. Lax and Wendroff, 1960; Mesinger and Arakawa, 1976; Roache, 1982; Hirsch, 1988b). The discretisation of the pressure gradient force has received much less attention than that of the transport term: it is usually computed (implicitly in anelastic models) by a

simple centred three-point difference. In contrast to that approach, the method of Osher and Solomon (1982) considers the mathematical properties of equation (4.1) and their physical meaning in a more complete and integrated fashion: the full flux $F(U)$, i.e. transport by the flow *and* pressure term are discretised together and the strong coupling between the equations is more consistently captured.

The approach to increase the order of the advective flux approximation (4.6) taken in this work follows what is widely referred to as the MUSCL approach (**m**onotone **u**pwind-**c**entred **s**chemes for **c**onservation **l**aws), named after a computer program implementing the ideas of van Leer (1979, also re-published in 1997). Therefore, the numerical scheme of a3m has several similarities to other methods such as PPM inherited from their common predecessor MUSCL.

As mentioned above, the piecewise constant approximation of U by cell averages limits the computation of the advective flux to first order. Thus, the key issue in constructing higher-order approximations of the advective fluxes (4.6) is already identified: the cell averages themselves must not be used directly as arguments to the numerical flux function (4.8) since those are only a first order approximation of the unknown on the cell boundaries. Following the MUSCL approach of van Leer (1979, 1997), an interpolation of the flow variables is passed to the numerical flux function instead: the piecewise constant approximation is replaced by a piecewise polynomial one.

The cell averages are used to compute a *recovery polynomial* $P_i(\underline{x})$ of the solution $U(\underline{x}, t^*)$ on cell T_i at time t^* giving a better spatial approximation of U. The polynomial P_i of rank $r-1$ must preserve the cell average and be an rth order approximation of U, i.e.

$$\frac{1}{|T_i|}\int_{T_i} P_i(\underline{x})\, d\underline{x} = U_i(t^*) \qquad \text{and}$$

$$||P_i(\underline{x}) - U(\underline{x}, t^*)|| = \mathcal{O}(h^r), \quad r > 0$$

for $\underline{x} \in T_i$ and $t^* > 0$. Instead of directly computing the recovery polynomial for the conserved variables $(\rho, \rho\underline{u}, \rho\mathsf{e})$, a different representation for the state can be used. Changing the representation of the atmospheric state for the recovery procedure is useful because at least the mass density ρ and total energy density $\rho\mathsf{e}$, which is dominated in magnitude by inner energy and therefore by pressure (Section 3.1.3), will not exhibit a linear variation in space: the decay with height clearly dominates and is distinctively non-linear. So a set of variables typically showing a more (piecewise) linear pattern in the stratified atmosphere will be more suitable. The variables recovered by a3m are the deviation temperature and pressure as well as the velocity,

$$W := (\mathsf{T}^{(1)}, \underline{u}, \mathsf{p}^{(1)})^T. \tag{4.9}$$

Alternatively, the Exner function $\pi := (\mathsf{p}/\mathsf{p}_0)^{R/c_\mathsf{p}}$ and potential temperature deviation $\Theta^{(1)}$, i.e. $W = (\Theta, \underline{u}, \pi)$ have been experimented with. But performance suffered noticeably due to the additional exponentiations performed converting between the two representations U and W while no improvement in the results was obtained from that approach.

A numerical scheme of second order in space is obtained by using linear functions,

$$P_i(\underline{x}) = a_0 + a_1(x_1 - c_{i1}) + a_2(x_2 - c_{i2}) + a_3(x_3 - c_{i3}), \tag{4.10}$$

to recover W and thereby U on each cell T_i, where $\underline{c}_i = (c_{i1}, c_{i2}, c_{i3})^T$ is the centre of gravity of cell T_i. Figure 4.3 shows a sketch of a piecewise linear approximation of the same curve as in Figure 4.2. Obviously, the approximation is much better than that with the piecewise constant function.

To obtain a linear recovery (4.10), an interpolation employing the cell averages is performed. Apart from the average state on cell T_i itself, three more cells, T_j, T_k and T_l (in three spatial dimensions), are necessary for P_i being uniquely determined. The set $\{T_i; T_j; T_k; T_l\}$ is called the *interpolation stencil*. The coefficients a_1, a_2 and

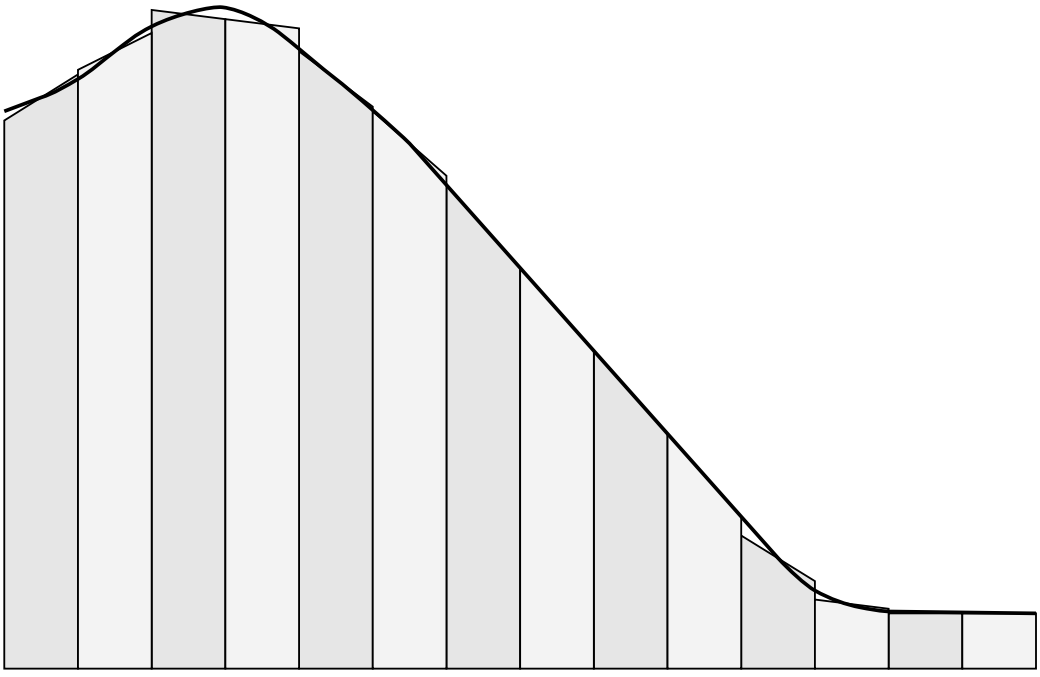

Figure 4.3: Piecewise linear approximation.
The same curve as in Figure 4.2 is now approximated by a piecewise linear function. The improvement over the piecewise constant approximation is most apparent in the regions of smooth slopes with little curvature.

a_3 (in three spatial dimensions) are then obtained by solving the linear system

$$\begin{bmatrix} c_{j1} - c_{i1} & c_{j2} - c_{i2} & c_{j3} - c_{i3} \\ c_{k1} - c_{i1} & c_{k2} - c_{i2} & c_{k3} - c_{i3} \\ c_{l1} - c_{i1} & c_{l2} - c_{i2} & c_{l3} - c_{i3} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} W_j - W_i \\ W_k - W_i \\ W_l - W_i \end{bmatrix} \tag{4.11}$$

and $a_0 = W_i$ (Sonar, 1997).

Obviously, there is more than one possible choice for the cells T_j, T_k, T_l within the vicinity of cell T_i. In a3m the cells T_j, T_k, T_l are chosen within the *Moore neighbourhood* of cell T_i:

$$j, k, l \in \{m \,|\, T_i \cap T_m \neq \emptyset\}.$$

The Moore neighbourhood contains all cells, that share a face or vertex with cell T_i. Figure 4.4 shows the Moore neighbourhood of the cell already featured in Figure 4.1. Compared to the von Neumann neighbourhood (shaded in Figure 4.1) the Moore neighbourhood is significantly larger: it contains eight instead of only four cells in two spatial dimensions and 26 instead of six in three spatial dimensions. Hence, it contains more possible interpolation stencils except for one dimension, where both are identical.

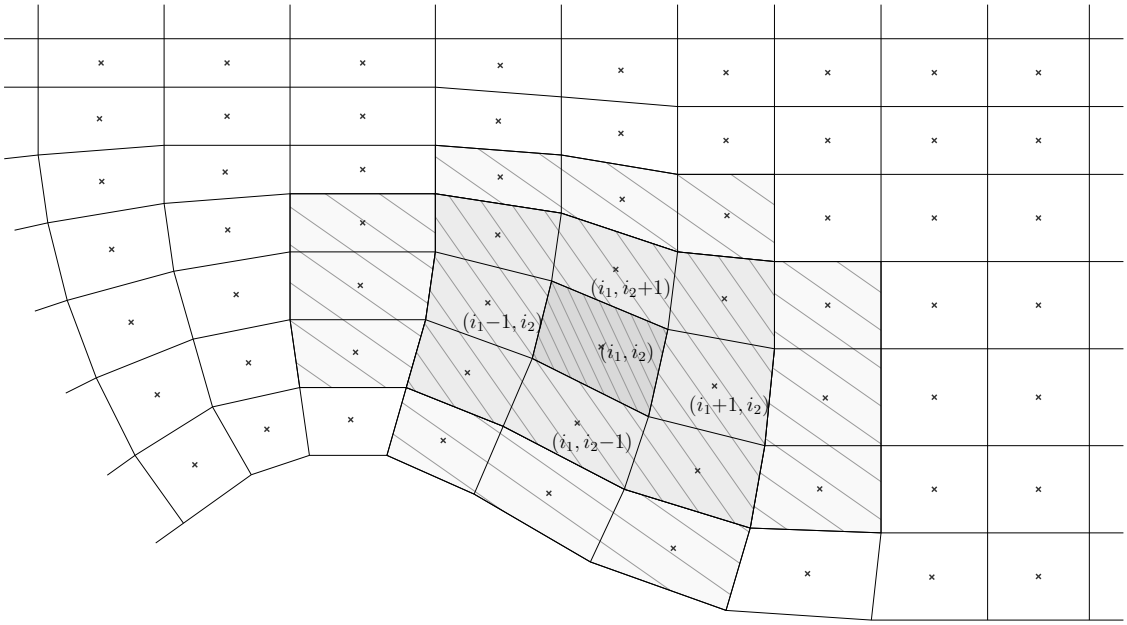

Figure 4.4: Moore neighbourhood and numerical region of influence.
The recovery on the cell marked (i_1, i_2) (darkest shade) is computed from the average on itself and from cells of its Moore neighbourhood (medium shade). All the hatched cells combined constitute the numerical region of influence: all of those cells contribute to the evolution of cell (i_1, i_2) either directly by fluxes or indirectly by influencing the recovery on the cells exchanging fluxes with cell (i_1, i_2).

Two additional rules for choosing stencils are applied in a3m:

- In two dimensions, the cells must form an 'angle' in index space, i.e. by rotation and translation in index space their index tuples can be transformed to $\{(0,0);(1,0);(0,1)\}$. All considered stencils are listed in Table 4.1.
- In three dimensions, the cells must form a 'tripod' in index space, i.e. by rotation and translation in index space their index tuples can be transformed to $\{(0,0,0);(1,0,0);(0,1,0);(0,0,1)\}$. All considered stencils are listed in Table 4.2.

In one spatial dimensional, Moore and von Neumann neighbourhoods are identical and the above rules are of no concern: there are only two possible stencils in any case.

These rules limit the number of combinations to a reasonable amount, especially in the three-dimensional case. They also reject unfavorable combinations of cells, such as cells consecutive in one index, e.g. the stencil $\{(i_1-1,i_2);(i_1,i_2);(i_1+1,i_2)\}$ in 2d, or stencils such as $\{(i_1-1,i_2-1);(i_1,i_2);(i_1+1,i_2+1)\}$ (also 2d). Most probably, such stencils do not yield a good recovery $P_i(\underline{x})$ because for grids deformed to a realistic degree, those cells' centres are close to lying on a line (2d) or plane (3d) in physical space (Figure 4.4). If the cells do lie on a line (2d) or plane (3d), equation (4.11) becomes singular, and no recovery can be defined from that stencil. If the cells are not exactly on a line but close to that, divisions by very small numbers are performed solving equation (4.11) so that even small rounding errors become significant. Therefore, these stencils are not even considered for computing recoveries in a3m. But for two or three spatial dimensions this would leave only four possible stencils in the von Neumann neighbourhood in two spatial dimensions and eight possible stencils in three spatial dimensions. Employing the Moore neighbourhood there are twelve possible stencils in two spatial dimensions

T_j	T_k	T_j	T_k
(i_1+1,i_2)	(i_1+1,i_2+1)	(i_1-1,i_2+1)	(i_1,i_2+1)
(i_1,i_2+1)	(i_1+1,i_2+1)	(i_1-1,i_2)	(i_1,i_2+1)
(i_1+1,i_2-1)	(i_1+1,i_2)	(i_1-1,i_2)	(i_1,i_2-1)
(i_1,i_2+1)	(i_1+1,i_2)	(i_1-1,i_2-1)	(i_1,i_2-1)
(i_1,i_2-1)	(i_1+1,i_2)	(i_1-1,i_2)	(i_1-1,i_2+1)
(i_1,i_2-1)	(i_1+1,i_2-1)	(i_1-1,i_2-1)	(i_1-1,i_2)

Table 4.1: Considered interpolation stencils for recoveries (2d).
Listed are index pairs of the Moore neighbours T_j and T_k of cell T_i to form an interpolation stencil $\{T_i,T_j,T_k\}$. Cell T_i has the index pair (i_1,i_2).

T_j	T_k	T_l
(i_1-1, i_2-1, i_3)	(i_1-1, i_2, i_3-1)	(i_1-1, i_2, i_3)
(i_1-1, i_2-1, i_3)	(i_1-1, i_2, i_3)	(i_1-1, i_2, i_3+1)
(i_1-1, i_2, i_3-1)	(i_1-1, i_2, i_3)	(i_1-1, i_2+1, i_3)
(i_1-1, i_2, i_3)	(i_1-1, i_2, i_3+1)	(i_1-1, i_2+1, i_3)
(i_1-1, i_2-1, i_3)	(i_1, i_2-1, i_3-1)	(i_1, i_2-1, i_3)
(i_1-1, i_2-1, i_3)	(i_1, i_2-1, i_3)	(i_1, i_2-1, i_3+1)
(i_1-1, i_2, i_3-1)	(i_1, i_2-1, i_3-1)	(i_1, i_2, i_3-1)
(i_1-1, i_2, i_3)	(i_1, i_2-1, i_3)	(i_1, i_2, i_3-1)
(i_1-1, i_2, i_3)	(i_1, i_2-1, i_3)	(i_1, i_2, i_3+1)
(i_1-1, i_2, i_3+1)	(i_1, i_2-1, i_3+1)	(i_1, i_2, i_3+1)
(i_1-1, i_2, i_3-1)	(i_1, i_2, i_3-1)	(i_1, i_2+1, i_3-1)
(i_1-1, i_2, i_3)	(i_1, i_2, i_3-1)	(i_1, i_2+1, i_3)
(i_1-1, i_2, i_3)	(i_1, i_2, i_3+1)	(i_1, i_2+1, i_3)
(i_1-1, i_2+1, i_3)	(i_1, i_2+1, i_3-1)	(i_1, i_2+1, i_3)
(i_1-1, i_2, i_3+1)	(i_1, i_2, i_3+1)	(i_1, i_2+1, i_3+1)
(i_1-1, i_2+1, i_3)	(i_1, i_2+1, i_3)	(i_1, i_2+1, i_3+1)
(i_1, i_2-1, i_3-1)	(i_1, i_2-1, i_3)	(i_1+1, i_2-1, i_3)
(i_1, i_2-1, i_3)	(i_1, i_2-1, i_3+1)	(i_1+1, i_2-1, i_3)
(i_1, i_2-1, i_3-1)	(i_1, i_2, i_3-1)	(i_1+1, i_2, i_3-1)
(i_1, i_2, i_3-1)	(i_1, i_2+1, i_3-1)	(i_1+1, i_2, i_3-1)
(i_1, i_2-1, i_3)	(i_1, i_2, i_3-1)	(i_1+1, i_2, i_3)
(i_1, i_2-1, i_3)	(i_1, i_2, i_3+1)	(i_1+1, i_2, i_3)
(i_1, i_2, i_3-1)	(i_1, i_2+1, i_3)	(i_1+1, i_2, i_3)
(i_1, i_2, i_3+1)	(i_1, i_2+1, i_3)	(i_1+1, i_2, i_3)
(i_1+1, i_2-1, i_3)	(i_1+1, i_2, i_3-1)	(i_1+1, i_2, i_3)
(i_1, i_2-1, i_3+1)	(i_1, i_2, i_3+1)	(i_1+1, i_2, i_3+1)
(i_1, i_2, i_3+1)	(i_1, i_2+1, i_3+1)	(i_1+1, i_2, i_3+1)
(i_1+1, i_2-1, i_3)	(i_1+1, i_2, i_3)	(i_1+1, i_2, i_3+1)
(i_1, i_2+1, i_3-1)	(i_1, i_2+1, i_3)	(i_1+1, i_2+1, i_3)
(i_1, i_2+1, i_3)	(i_1, i_2+1, i_3+1)	(i_1+1, i_2+1, i_3)
(i_1+1, i_2, i_3-1)	(i_1+1, i_2, i_3)	(i_1+1, i_2+1, i_3)
(i_1+1, i_2, i_3)	(i_1+1, i_2, i_3+1)	(i_1+1, i_2+1, i_3)

Table 4.2: Considered interpolation stencils for recoveries (3d).
Listed are index tripels of the Moore neighbours T_j, T_k and T_l of cell T_i to form an interpolation stencil $\{T_i, T_j, T_k, T_l\}$. Cell T_i has the index triple (i_1, i_2, i_3).

(Table 4.1) and 32 in three spatial dimensions (Table 4.2). It is more likely to find a good recovery in a larger set of possible stencils so that fewer oscillations can be expected. This has also been observed in numerical experiments (Sonar, 1997, and references therein).

At this point, one effect of introducing a recovery instead of computing the fluxes between cells only from the averages becomes apparent: the numerical region of influence of a cell T_i is enlarged significantly. While in the first order scheme the region of influence contains only those cells that exchange fluxes with cell T_i, its von Neumann neighbours (Figure 4.1), the recovery on these cells now also has an influence on the flux between the cells. So with a recovery as described above the region of influence becomes the union of the Moore neighbourhoods of all von Neumann neighbours of cell T_i (all of the hatched cells in Figure 4.4).

Since the choice of the interpolation stencil is not unique, there is also no unique recovery P_i on a cell T_i. Therefore, an additional procedure or algorithm is necessary to determine the actual recovery to use on that cell from the set of possibilities. And it is this procedure of finding the recovery that determines the oscillatory properties of the total FV scheme: if the recovered gradient on a cell is steep enough, that the recovered values on the face do not fall in the range between the averages on the cells sharing that face, non-physical oscillations will occur (e.g. Hirsch, 1988b).

Several different classes of methods with different oscillatory properties exist. Total variation diminishing (TVD) schemes (Harten, 1983, re-published in 1997) do not introduce any artificial oscillations into the solution. But they degenerate to a first order scheme in critical points, i.e. in local extrema and in the vicinity of discontinuities or steep gradients. Therefore, they still smear extrema and fronts which is not acceptable for the purposes of this work. In this respect, total variation bound (TVB) schemes (Shu, 1987) are an improvement: they avoid the degeneration to first order in critical points at the price of admitting some amount of artificial oscillations. A third class of schemes employs essentially non-oscillatory (ENO) interpolation (Harten *et al.*, 1987; Harten, 1996; Sonar, 1997; Abgrall *et al.*, 1999). ENO schemes are also of global higher order, i.e. they do not degenerate in critical points. They were specifically designed for simulations with discontinuities which is the main reason for choosing them as the recovery mechanism of a3m.

The central idea of ENO interpolation is to choose the interpolation stencil on which the solution is smoothest, i.e. there are no discontinuities in the interpolated function or its derivatives within the stencil. For the linear interpolation used here only discontinuities in the function itself and the first derivative, i.e. corners in the function, are detected. The stencil of interpolation $\{T_i, T_j, T_k, T_l\}$ (in three spatial dimensions) is chosen to minimise the coefficients a_1, a_2 and a_3 of equation (4.10). So the recovery P_i resulting from this procedure is the polynomial with the least steep gradient of all possible choices of stencils in the employed neighbourhood which can

also be seen in Figure 4.3. Thus, interpolation across discontinuities and thereby loss of accuracy is avoided. This guarantees a good resolution of steep gradients, e.g. at fronts, with only little artificial oscillations—a very desirable combination of properties in the context of this work. As the name suggests, they are not entirely free of non-physical oscillations and therefore do not ensure stability or positive definiteness in all cases. But they have been applied with great success in various fields, such as the shallow water equations (e.g. Yost and Rao, 2000) and high Mach number flows (e.g. Abgrall *et al.*, 1999).

The first order advective flux approximation (4.6) with the numerical flux function (4.8) is replaced by the second order approximation

$$\int_{S_{ij}} \sum_{l=1}^{3} F_l(U(\underline{x}))n_{ij_l}\, d\underline{x} = |S_{ij}| H^{\mathrm{OS}}\big(P_i(\underline{c}_{ij}), P_j(\underline{c}_{ij}); \underline{n}_{ij}\big) + \mathcal{O}(h^2), \tag{4.12}$$

where the recoveries P_i and P_j are linear functions in space (4.10) of the variables $\mathsf{T}^{(1)}$, $\underline{u}$ and $\mathsf{p}^{(1)}$ determined by ENO interpolation using equation (4.11) as described above.

4.1.2 Turbulent Fluxes

The separation of sub-grid-scale variations from the resolved scale in Section 3.2 introduced the turbulent fluxes G_l into the equations. Using a first order closure (Section 3.3) these are modelled as a diffusional process. So the computation of these terms requires values of the unknown itself as well as the gradients of some variables on the faces S_{ij}. For time t^* they are obtained from a finite-difference approach along the line connecting the centres $\underline{c}_i$ and $\underline{c}_j$ of two cells T_i and T_j between which to calculate the turbulent flux:

$$W(\underline{x}(s), t^*) = W_i + \frac{W_j - W_i}{|\underline{c}_j - \underline{c}_i|} s + \mathcal{O}(h^2), \tag{4.13}$$

where the same variables W given by equation (4.9) as for computing the recovery are employed. The parameter s is the distance from cell centre $\underline{c}_i$ in the direction towards the other cell centre $\underline{c}_j$ so that $\underline{x}(s) = \underline{c}_i + (\underline{c}_j - \underline{c}_i)/|\underline{c}_j - \underline{c}_i| s$. As shown by Figure 4.5, the line connecting the cell centres intersects the face S_{ij} in point $\underline{p}_{ij}$, which, in general, does not coincide with the face centre $\underline{c}_{ij}$. Evaluating (4.13) for the intersection point $\underline{p}_{ij}$ yields

$$W(\underline{p}_{ij}, t^*) = W_i + \frac{W_j - W_i}{|\underline{c}_j - \underline{c}_i|} |\underline{p}_{ij} - \underline{c}_i| + \mathcal{O}(h^2) \qquad \text{and} \tag{4.14a}$$

$$\nabla W(\underline{p}_{ij}, t^*) = \frac{W_j - W_i}{|\underline{c}_j - \underline{c}_i|} \frac{\underline{c}_j - \underline{c}_i}{|\underline{c}_j - \underline{c}_i|} + \mathcal{O}(h). \tag{4.14b}$$

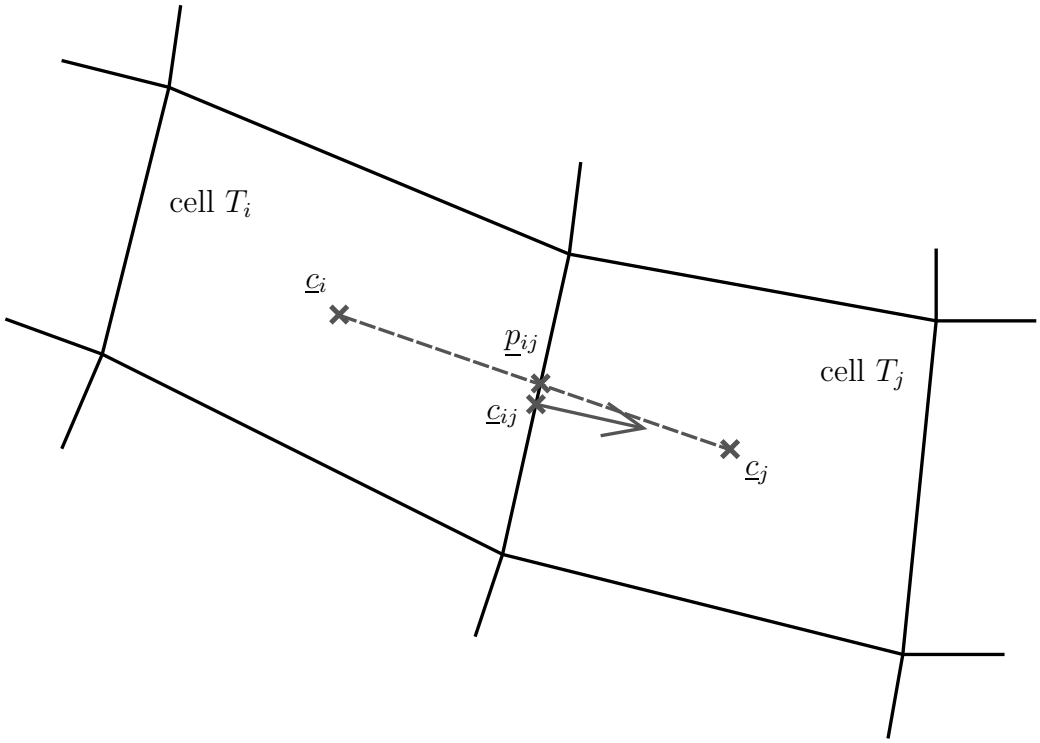

Figure 4.5: Finite-difference-like approach for the turbulence flux.

From these, the approximations of $U\big(W(\underline{p}_{ij}, t^*)\big)$ and the gradients $\partial_{x_j} u_i$ and $\partial_{x_j} \mathsf{e} = \partial_{x_j} \mathsf{u} + \partial_{x_j} |\underline{u}|^2/2 = c_V \partial_{x_j} \mathsf{T} + |\underline{u}|\, \partial_{x_j} |\underline{u}|$ required for evaluating the fluxes G_l (Section 3.3) can be readily obtained.

To complete the computation of turbulent mixing, the exchange coefficients K_{vert}, K_{hor}, $K_{\mathsf{e}\text{vert}}$ and $K_{\mathsf{e}\text{hor}}$ have to be determined using equations (3.14), (3.15), (3.17) and (3.18). For obtaining K_{vert} by equation (3.14) the vertical shear is required. But approximation (4.14b) gives gradients only in the direction from cell T_i to its neighbour T_j. Hence, the vertical shear cannot be derived from that. Consequently, a3m employs the recoveries P_i and P_j as given by equation (4.10) to this end. The required gradients are immediately given by the coefficients a_1, a_2 and a_3 yielding the exchange coefficients on each cell. For the turbulent flux between cell T_i and its neighbour T_j the exchange coefficients of both cells are averaged.

With the approximations (4.14) of the unknown and its derivative on the face the numerical flux function for turbulent mixing is defined as

$$H^{\mathrm{T}}(U_i, U_j; \underline{p}_{ij}, \underline{n}_{ij}) := \sum_{l=1}^{3} G_l\big(U(W(\underline{p}_{ij}, t^*))\big) n_{ij_l} = \sum_{l=1}^{3} G_l(U) n_{ij_l} + \mathcal{O}(h^2). \quad (4.15)$$

Approximating the integral over the face S_{ij} and inserting the above numerical flux

function into the turbulent flux term of equation (4.4) yields

$$\int_{S_{ij}} \sum_{l=1}^{3} G_l n_{ijl} \, d\underline{x} = |S_{ij}| H_{ij}^{\mathrm{T}} + \mathcal{O}(h) \tag{4.16}$$

The sampling point is $\underline{p}_{ij}$ in this case and not the centre $\underline{c}_{ij}$ of face S_{ij} (Figure 4.5) thereby giving a first order approximation of the integral. Only for rectilinear geometries of the computational grid, the faces centre of gravity and the intersection point coincide, i.e. $\underline{p}_{ij} = \underline{c}_{ij}$, thus giving second order accuracy. But the numerical flux function (4.15) is only used to compute the fluxes introduced by the turbulence model (section 3.3) which parameterises the effect of sub-grid-scale motions on the resolved fields. It therefore has—as all parameterisations—a high degree of uncertainty compared to the resolved scale processes described by the advective flux and source terms. Consequently, no further effort will be made here to increase the accuracy of the turbulent flux approximation.

4.1.3 Source Terms

The volume integral of the source function $Q(U, \underline{x}, t)$ over a cell T_i in equation (4.4) is also approximated by a quadrature:

$$\frac{1}{|T_i|} \int_{T_i} Q(U, \underline{x}, t) \, d\underline{x} = Q_i + \mathcal{O}(h^2), \tag{4.17}$$

where $Q_i := Q(U_i, \underline{c}_i, t)$ assuming that Q is smooth enough with respect to both, U and $\underline{x}$. This is of course not true for actual physical sources in the model domain, e.g. a point source like a factory stack. In this case, these sources must be accounted to the cell containing them, but within this work, no sources of this kind will occur so that equation (4.17) suffices here.

4.1.4 Finite Volume Method

Assembling the equations (4.8), (4.10) and (4.12) for the advective flux, (4.14), (4.15) and (4.16) for the turbulent mixing and (4.17) for the source terms results in the finite volume spatial semi-discretisation

$$\begin{aligned} \frac{d}{dt} U_i(t) = -\frac{1}{|T_i|} \sum_{j \in \mathcal{N}(i)} |S_{ij}| \Big(& H^{\mathrm{OS}}\big(P_i(\underline{c}_{ij}), P_j(\underline{c}_{ij}); \underline{n}_{ij}\big) \\ & + H^{\mathrm{T}}\big(U_i, U_j; \underline{p}_{ij}, \underline{n}_{ij}\big) \Big) + Q_i. \end{aligned} \tag{4.18}$$

A basic algorithm for computing the tendencies of each of the cell averages from equation (4.18) is: (1) compute the recoveries P_i on each cell T_i from all possible interpolation stencils as specified in Section 4.1.1.2, (2) compute the advective and turbulent fluxes through each face S_{ij} using the recoveries P_i and P_j and the numerical flux functions (4.8) and (4.15) and thereby the tendency induced in each of the cells T_i and T_j sharing that face and (3) compute the source terms of each cell T_i and the tendency induced by that. The time integration of a3m (Section 4.3) works in a similar way (Figure C.3).

Equation (4.18) can be combined with virtually any time discretisation to form a complete numerical method for computing approximate solutions of U. But choosing a concrete time discretisation, care has to be taken of that particular methods stability requirements and oscillatory properties. Otherwise, the advantageous essentially *non*-oscillatory property of the spatial discretisation might be lost. Section 4.3 describes the time discretisation of a3m and its stability requirements in detail.

One of the remarkable properties of the FV method is that—since the spatial derivatives are removed from the equations by the Gauß integral theorem—it does not need coordinate transformations to handle non-Cartesian geometries. In contrast to that, FD methods discretise the spatial differential operators on grids along the coordinate lines only. As mentioned in Section 3.1.6, this may result in complicated difference operators. The FV method as introduced above uses a strictly geometrical approach instead which is based on the cells and faces of the grid and their geometric properties: centre and volume content for cells and centre, surface area and normal direction for faces (Figure 4.1). All considerations as well as manipulations of the equations are done in normal physical space which is more intuitive than the sometimes rather abstract coordinates required for FD methods. Just as for FD methods, deformations of the grid must stay within reasonable bounds: extremely pointed angles in the cells will cause precision degradation and instabilities. Such cells are too close to being 'flat', i.e. having no actual content which is a violation of requirement 2 on the grids listed at the beginning of Section 4.1. In fact, $|T_i|$ vanishes in that case, and equation (4.18) becomes singular.

4.1.5 Boundary Treatment

Sections 4.1.1 and 4.1.2 describe the computation of fluxes in the interior of the computational domain. Determining fluxes across the domain boundary imposes a special problem since the state on the outside of the domain is not known. This raises the need for boundary conditions. The consequences of introducing boundary conditions are twofold. Firstly, the fluxes across the boundaries must be computed depending on the boundary conditions. This includes solving a boundary Riemann

problem in a consistent manner with the flux computation in the interior. And secondly, the boundary conditions influence the recoveries on cells at the domain boundary.

For computing the advective flux, the approach of Osher and Solomon (1982) used to construct the numerical flux function for the interior (Section 4.1.1.1) can also be applied at the domain boundaries (Spekreijse, 1988). As in Section 4.1.1, the frame of reference is rotated to obtain a locally one-dimensional problem yielding the state $\hat{U}_i$ on a cell T_i at the domain boundary. Then a state U_B outside the domain boundary is constructed from a set of boundary conditions and the state $\hat{U}_i$ by means of the eigenvectors and eigenvalues of the Jacobian of the flux function and the Riemann invariants of the Euler equations (Spekreijse, 1988). The flux across the boundary is then given by $F_1(U_B)$.

Three different cases of boundaries are considered here: *inflow* boundaries where the flow is directed into the domain, *outflow* boundaries where the flow is directed out of the domain and *solid walls* which are completely impermeable by the flow. Inflow boundaries need four boundary conditions, while outflow boundaries as well as solid walls only need one. The number of conditions is easily derived from the directions in which information is transported by the simple wave solutions of the boundary Riemann problem: positive eigenvalues of the flux functions Jacobian $\nabla_U F_1$ correspond to waves leaving the domain and negative ones to waves entering it. A wave entering the domain raises the need for a boundary condition. For sub-sonic flows as they are considered here (assumption 2 in Section 3.1.1), at least one eigenvalue, namely $\hat{u}_1 - \mathsf{c}$, is always negative. Thereby—imposing a suitable boundary condition—pressure information can still travel across the boundary, even opposing the flow.

The conditions used in a3m are $\underline{u}_B = \underline{u}_i$ and $\mathsf{p}_B = \mathsf{p}_i$ for inflow boundaries and $\mathsf{p}_B = \mathsf{p}_i$ for outflow boundaries. For solid walls the wall-normal velocity component vanishes, i.e. $u_{B_1} = 0$. From these conditions and the Riemann invariants follows

$$\begin{aligned} \rho_B &= \rho_i \\ u_{B_1} &= 0 \\ u_{B_2} &= \hat{u}_{i2} \\ u_{B_3} &= \hat{u}_{i3} \\ \mathsf{p}_B &= \mathsf{p}_i \left(\frac{(\gamma - 1)\hat{u}_{i1}}{2\mathsf{c}_i} + 1 \right)^{2c_\mathsf{p}/R} \end{aligned}$$

in the case of solid walls and

$$U_B = U_i$$

for in- and outflow boundaries.

At the upper boundary of the model domain a3m uses a rigid lid, i.e. a wall, with a damping layer underneath. Within this layer, Rayleigh damping of the momentum is introduced in the form of an artificial source term depending on the height z:

$$Q_{\mathrm{RD}}(u, \underline{x}) = -\alpha(\rho\underline{u} - (\rho\underline{u})_{\mathrm{RD}})$$

with

$$\alpha = \alpha_{\mathrm{RD}} \frac{(z_{\mathrm{top}} - \Delta z_{\mathrm{RD}} - z)^2}{\Delta z_{\mathrm{RD}}^2}$$

added to Q of equation (4.1), where $(\rho u)_{\mathrm{RD}}$ is the momentum towards which relaxation is performed, z_{top} is the height of the computational domains upper boundary, Δz_{RD} is the thickness of the damping layer and α_{RD} is the Rayleigh damping coefficient. Tests indicated $\alpha_{\mathrm{RD}} = 0.2$ to be a reasonable choice and this value is used in the simulations conducted in Chapter 5. The basic atmospheric state is used for the simulations in Chapter 5 so that $(\rho u)_{\mathrm{RD}} = 0$, but in general—depending on the simulated scenario—it has to be modified to relax the wind field towards the larger-scale geostrophic wind.

As described in Section 4.1.2, computation of turbulent fluxes requires values for the unknown itself as well as its gradient on the faces, in this case, a part of the domain boundary. The finite-difference-like approach (4.13) is obviously not applicable at the boundary. Therefore, the required quantities are directly extracted from the linear recovery (4.10) and the flux is computed by (4.15) and (4.16) as in the interior of the domain using the exchange coefficient from the cell at the boundary.

The recovery on a cell T_i at an open domain boundary also depends on the values outside the domain. At solid walls there are no values on the outside, so the recovery is computed from the domain interior cells alone. At open boundaries, i.e. inflow or outflow boundaries, a3m enforces a zero gradient in normal direction to the boundary. For testing purposes, cyclic boundaries have also been found useful. Both effects are achieved by a simple copy operation on the cell averages prior to computing the recovery on each cell which automatically yields the intended result at the cost of some extra storage.

4.2 Local Grid Refinement

The FV discretisation as described in Section 4.1 works on virtually any computational grid fulfilling the requirements 1–5 listed at the beginning of Section 4.1, which also includes unstructured grids. The algorithm just uses the concepts of cells and faces with their neighbour relationships, centres of gravity, volume, surface area and normal vectors. For reasons of computational efficiency structured grids

are used in a3m. So cells and faces can be identified by index tuples and neighbour relationships are easily determined from the indices alone. Within a framework of structured grids it is not feasible to achieve grid refinement on a cell-by-cell basis as it is done with unstructured grids (e.g. Muzaferija, 1994; Sonar, 1997; Dolejší, 1998; Abgrall *et al.*, 1999). Instead, multiple cells to refine are combined to *blocks*, rectangular regions in index space, and the whole region within the grid is overlaid by a new block of higher resolution. An example of a grid configuration resulting from this approach is shown in Figure 4.6.

The method known best employing this approach is the algorithm of Berger and Oliger (1984), which was invented to provide the classic and widely used FD methods with means for grid adaption. For this algorithm the computational domain is discretised with a hierarchy of structured rectangularly indexable blocks. Adaption is achieved by partially overlaying blocks in one level of the hierarchy with higher resolving ones from the next level. The FD solver is then applied to each block separately starting with the highest resolving level. A framework around the

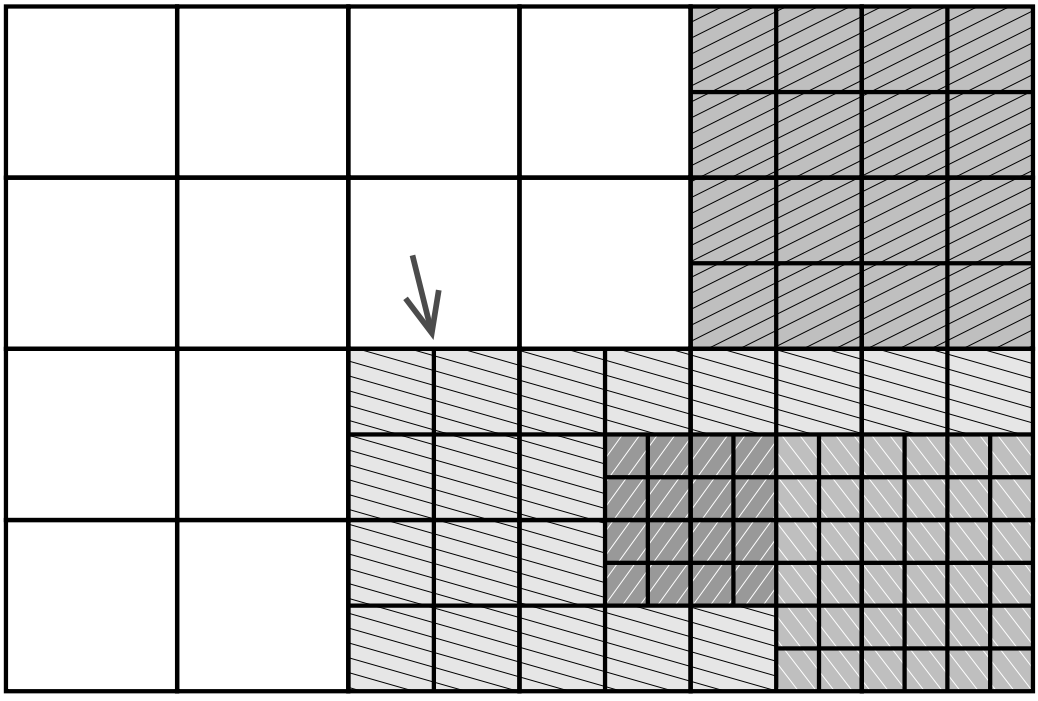

Figure 4.6: Refinement by overlaying blocks.
With the FV method introduced in Section 4.1 grid adaption can be achieved most efficiently by partially overlaying grid blocks with other blocks of higher resolution. The view of the FV solver is that of an ordinary block-structured grid. Note the presence of hanging nodes (arrow) on the connections of the different blocks, which FD methods cannot handle directly.

solver ensures the exchange of information between the different blocks by interpolation and prolongation. Additionally, the framework has to ensure the conservation properties of the scheme by correcting fluxes on the boundaries between blocks of different resolution. So the blocks are not directly connected within the FD solver but communicate indirectly via the framework and thereby *externally* to the numerical solver itself. For a more detailed description see the original article of Berger and Oliger (1984) or the comprehensive example of a framework implementing the Berger-Oliger algorithm by Neeman (1996). The Berger-Oliger algorithm has been applied to the atmosphere by Skamarock and Klemp (1993).

In principle, it is possible to use the method of Berger and Oliger also for FV schemes as that described in Section 4.1. But much of the algorithms complexity is redundant for FV schemes since they do not suffer from the same limitations as FD methods. Connecting two grid blocks of different resolution to a single grid with hanging nodes (Figure 4.6) imposes a fundamental problem for FD methods: they lack a partner cell to calculate the (finite) difference. Hence, the Berger-Oliger algorithm requires a framework to achieve the exchange between the different grid blocks using interpolation, prolongation and an additional flux correction in the Berger-Oliger framework. But FV schemes as described in Section 4.1 have no need for such a framework. They require cells, faces, their geometrical properties and neighbour relationships. Providing this information for a connection between the grid blocks, the scheme can be applied directly so that the coupling of blocks can be achieved *internally* by the numerical solver itself. Thus, a major part of the Berger-Oliger framework would be obsolete.

Consequently, grid arrangements as the one shown in Figure 4.6 are viewed from a different angle: in the Berger-Oliger algorithm the numerical solver is applied to each grid within a collection of grids and the exchange between grids is handled externally by a framework around the solver. In contrast to that constructs like the one shown in Figure 4.6 are treated as *one single grid* in a3m and not as a hierarchy. The grid is *block-structured* and consists of several *blocks* and the *connections* between them. Blocks may differ in resolution and/or partially overlay others. Overlaid cells in a block are marked as 'dead' and do not contribute to the computations in any way. Instead, the cells from the overlaying block are used and fluxes are exchanged between cells of both blocks over their connection. The only difference to a grid consisting of a single block is that the interpolation stencil for the recovery polynomial (Section 4.1.1.2) may cross from one block to another and fluxes have to be computed between cells of adjacent blocks. The neighbourhood relations between cells of different blocks are provided by the connections. The implementation in a3m requires all cells of one block within a single connection to have the same number of von Neumann neighbours, e.g. in Figure 4.6 there are always two higher resolving cells connected to one coarse cell. Note that the refinement ratio does not need to

be the same in each direction in index space. It could also be 1 : 1 in one direction in index space so that refinement is only performed in the other directions.

4.3 Time Discretisation

The simple-most method to integrate from time step n to $n+1$ is the Euler forward scheme:

$$U_i^{n+1} = U_i^n - (F_i(U^n) - Q_i^n)\Delta t + \mathcal{O}(\Delta t) \tag{4.19}$$

for all cells T_i, where

$$F_i := \sum_{j \in \mathcal{N}(i)} \frac{|S_{ij}|}{|T_i|} \Big(H^{\mathrm{OS}}(U_i, U_j; \underline{n}_{ij}) + H^{\mathrm{T}}(U_i, U_j; \underline{p}_{ij}, \underline{n}_{ij}) \Big).$$

The whole right hand side of equation (4.19)—all fluxes and sources—is evaluated at time level n and then assumed constant over the time step Δt.

For the time integration scheme (4.19) to be linearly stable, the time step Δt must not exceed a certain value. The limitation has a simple geometric interpretation sketched in Figure 4.7: the solution of the Riemann problems on the cell boundaries is a combination of the simple wave solutions of equation (4.1). These propagate with the phase speeds given by the eigenvalues of the Jacobian of the flux F from the position of the discontinuity (the faces). For the scheme to be stable, the waves from different Riemann problems, i.e. faces, must not interfere: within one time step the waves originating from face S_{ij} must not travel more than the distance $|\underline{r}_{ij}|$ from the faces, where $\underline{r}_{ij} := \underline{c}_{ij} - \underline{c}_i$. The (local) Courant number CFL_i on cell T_i must not exceed unity:

$$\mathsf{CFL}_i := \max_{j \in \mathcal{N}(i)} \frac{(\mathsf{c}_i + \underline{u}_i \underline{r}_{ij}/|\underline{r}_{ij}|)\Delta t_{\mathrm{adv}}}{|\underline{r}_{ij}|} \leq 1 \qquad i = 1, \ldots, N, \tag{4.20}$$

This is nothing but the finite volume equivalent of the classical stability criterion known as the CFL condition, named after Courant, Friedrichs and Lewy (1928). The restriction on the time step is severe because the compressible equations (4.1) also contain sound waves so that the phase speed of the fastest travelling signal is $\mathsf{c} + |\underline{u}|$ with the speed of sound $\mathsf{c} = \sqrt{\gamma \mathsf{p}/\rho} \approx 340\,\mathrm{m/s}$ at ground level. Because of the strongly varying cell size in locally refined grids, the permissible time step also varies significantly on different levels of refinement. The very small cells of strongly refined regions restrict the step used in the time integration of the grid as a whole.

The time step limitation due to sound waves is especially wasteful since sound waves themselves are not of meteorological interest. Runge-Kutta methods are very popular for higher-order time discretisations because they permit higher CFL

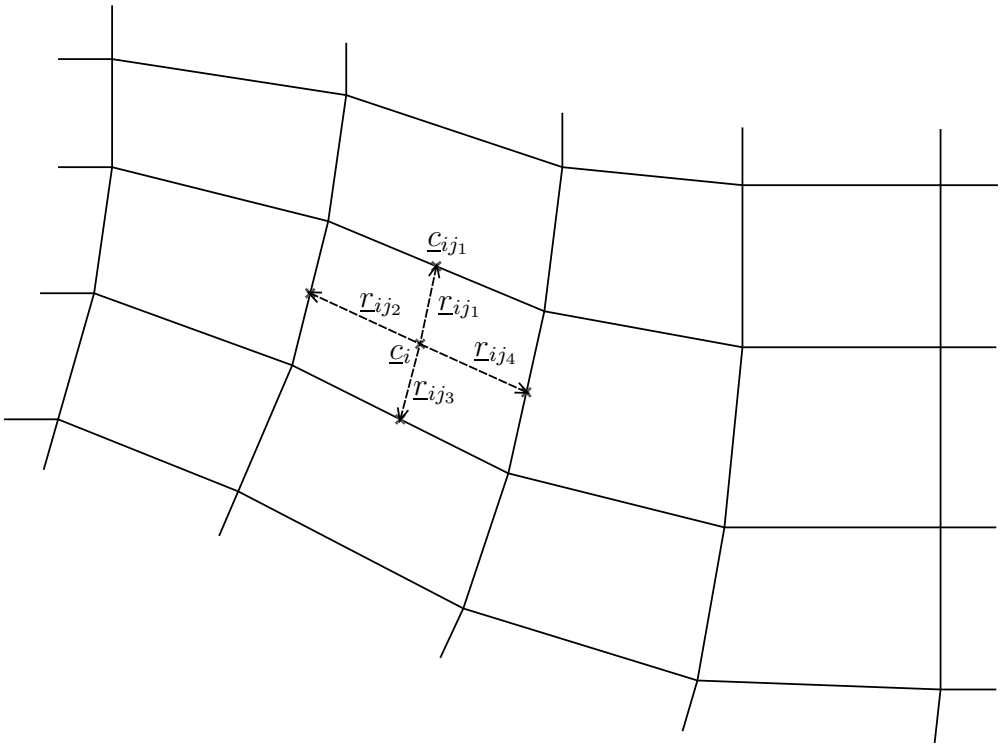

Figure 4.7: Geometric interpretation of the time step limitation.
The simple waves travelling from the discontinuity at face S_{ij} into the cell must not cover more than the distance $|\underline{r}_{ij}|$ within one time step for the time integration (4.19) to remain stable.

numbers than limit (4.20) due to their multiple stages. Unfortunately, this has the side effect of introducing oscillations into the solution so that the advantageous essentially non-oscillatory property of the space discretisation is lost. Consequently, higher-order TVD Runge-Kutta methods have been developed (Shu and Osher, 1988). But they have an *additional* restriction to the time step in order to remain TVD, namely $\mathsf{CFL} \leq 1$, so there is no gain in the time step size. Therefore, they are not implemented in a3m, but are still a possible future extension.

Other approaches for a more efficient time integration are implicit methods or time splitting schemes. The latter are very popular in atmospheric modelling (e.g. Tapp and White, 1976; Skamarock and Klemp, 1992; Xue *et al.*, 2000). Time splitting schemes integrate different processes with different time steps, making many small time steps for the fast processes, e.g. sound and gravity waves, while the remaining processes are integrated using larger time steps. It is not obvious though, how this approach would have to be applied to the schemes introduced in Section 4.1 since the strong coupling between the different processes is retained in the numerical flux function (4.8). Implicit methods require a linearisation of the numerical scheme

and solving the system of linear equations resulting thereof. Due to the high programming effort needed for their implementation, this is not done as part of this work.

The turbulent fluxes impose an additional constraint on the time step:

$$\frac{2K_i\Delta t_{\text{diff}}}{\min_{j\in\mathcal{N}(i)}|\underline{r}_{ij}|^2} \leq 1 \qquad i = 1,\ldots,N, \tag{4.21}$$

where K_i is the maximum exchange coefficient on cell T_i. This is the normal time step limitation for computing diffusion with explicit time integration methods (e.g. Roache, 1982).

For determining a suitable time step to apply the scheme (4.19), it has to be taken into account that advection and turbulence act simultaneously: advected information is spread further by the turbulence induced diffusion. So the time step used must be smaller than that gained from either one of the constraints (4.20) and (4.21). Following e.g. Roache (1982), a3m employs

$$\Delta t = \alpha_{\mathsf{CFL}} \min_{i=1,\ldots,N} \left(\frac{2K_i}{r_i^2} + \frac{\mathsf{c}_i + |\underline{u}_i|}{r_i} \right)^{-1} \tag{4.22}$$

with $r_i := \min_{j\in\mathcal{N}(i)}|\underline{r}_{ij}|$ and the user definable parameter $\alpha_{\mathsf{CFL}} \leq 1$. Note that in the time step limitation for advection (4.20) the projection of $\underline{u}$ on the vectors $\underline{r}_{ij}$ has been replaced by $|\underline{u}_i| \geq \underline{u}_i\underline{r}_{ij}/|\underline{r}_{ij}|$ as a simplification. Suitable values of α_{CFL} are in the range of 0.8 to 0.9, depending on the simulated scenario. Discontinuities—especially in the initial condition—may require a smaller time step: at the end of an integration step, after potential accelerations have taken place, the constraints (4.20) and (4.21) should still hold, so a value less than one is definitely required.

4.4 Extensions to the Physical Problem Domain

As discussed in Section 3.4, the mathematical model may be extended to include the transport of passive or reactive tracers and to the moist atmosphere with the transport of water in different phases and the effects of phase changes. This has some implications for the numerical scheme presented in this chapter so far because the structure of the equations may be changed by the inclusion of new physical processes.

For the additional prediction of tracer concentrations, the system of equations (4.1) is extended by an advection equation for each substance. For a tracer substance s the new vector of predicted variables becomes $U := (\rho, \rho u, \rho\mathsf{e}, \rho c_s)^T$ and the vector of variables used to compute the recovery (4.10) becomes $W := (\mathsf{T}^{(1)}, \underline{u}, \mathsf{p}^{(1)}, c_s)^T$,

where c_s is the mass mixing ratio of the tracer. Computation of the turbulent fluxes is straight forward. Source terms are handled analogously to the other variables, i.e. they are approximated by

$$\frac{1}{|T_i|} \int_{T_i} Q_s(U, \underline{x}, t)\, d\underline{x} = Q_{si} + \mathcal{O}(h^2).$$

Again, emission rates of actual real-world sources, e.g. point sources associated with factory stacks, have to be injected into the right cells. The advective flux is more critical because the terms (4.8b)–(4.8d) of the numerical flux function (4.8a), i.e. the states U_a and U_b, depend on the eigenvectors and eigenvalues of the Jacobian of the flux function $\nabla_U F_1(U)$ and the Riemann invariants of the equations (Appendix A). Fortunately, the addition of an advection equation to the system just gives $\hat{u}_1$ as the corresponding eigenvalue which occurs already three times. So the concentration ρc_s behaves exactly as the rotated momentum components $\rho\hat{u}_2$ and $\rho\hat{u}_3$:[1] it is passively transported by the velocity u_f.

For the transition to the moist atmosphere, the above remarks on turbulence and source terms apply as well. The expressions of the advective numerical flux function are altered though: with the introduction of moisture, and thereby the change from a homogeneous medium and a single phase to a multi-component multi-phase flow, the equation of state is now (3.19) and significantly different from before, e.g. $\partial_{q_v} \mathsf{p} \neq 0$. Consequently, the Jacobian of the flux function $\nabla_U F_1(U)$ and its eigenvalues and eigenvectors as well as the Riemann invariants are changed. However, the mechanism of the Osher-Solomon flux function is still applicable since the equations for the moist atmosphere are hyperbolic as well. Following Osher and Solomon (1982), Spekreijse (1988) and Toro (1997), the expressions for the new set of equations may be derived.

[1] Due to the inherent one-dimensional nature of the Riemann problem the momentum components $\rho\hat{u}_2$ and $\rho\hat{u}_3$ *are* nothing but passive tracers.

Chapter 5

Model Tests

In this chapter the discrete model introduced in Chapter 4 is made subject to various tests. The purpose of these tests is threefold: firstly, to test the implementation of the numerical methods for correctness, secondly, to verify some properties of the numerical methods and, thirdly, to demonstrate their suitability for simulating the atmosphere. Therefore, a3m is applied to relatively simple cases for which the outcome is analytically or at least qualitatively known, e.g. from physical experiments or simulations with other models. Additionally, two common measures to improve numerical stability, namely computational mixing (i.e. artificial smoothing) and divergence damping, are examined towards their suitability to be combined with the numerical schemes used in a3m and their effects on the simulation results.

5.1 Burgers Equation

ENO methods (Harten *et al.*, 1987) were developed as a global higher order discretisation for non-linear hyperbolic conservation laws (Sonar, 1997). As a low-complexity example of this class of mathematical problems, the Burgers equation in one spatial dimension is studied in this section to demonstrate the applicability and properties of the numerical method introduced in Chapter 4. Special emphasis is put on the robustness of the method with respect to the computational grid, especially abrupt changes in resolution.

The problem to solve numerically is

$$\partial_t u + \partial_x (u^2/2) = 0, \qquad t > 0, \tag{5.1}$$

with

$$u(x, t = 0) = u_0(x)$$

for $x \in [0; 1]$. So the advective flux function for this problem is $f(u) = u^2/2$ and there are no diffusive fluxes and source terms. Note that $\partial_x(u^2/2) = u\,\partial_x u$, so the Burgers equation can also be interpreted as a dimensionless momentum equation for a fluid of constant density. Therefore, this simple equation already captures a vital part of the non-linearity inherent to the system of equations (4.1). The value at the lower boundary, $x = 0$, is prescribed as $u(0, t) = 0$. At the upper boundary, $x = 1$, a simple outflow condition is applied extrapolating from the last cell centre to the

domain boundary. Simulations are conducted with a fixed time step of $\Delta t = 0.0008\,\mathrm{s}$ to make simulations on computational grids with different resolutions comparable. The recovered variable is the unknown u itself.

For scalar equations the numerical flux function of Osher and Solomon (1982) reduces to that proposed by Enquist and Osher (1981), which is a generalised upwind method for non-linear, scalar conservation laws. For the Burgers equation (5.1) it is given by

$$H^{\mathrm{EO}}(u_l, u_r) = H^+(u_l) + H^-(u_r),$$

where

$$H^+(u) = \begin{cases} u^2/2 & \text{if } u > 0 \\ 0 & \text{if } u \leq 0 \end{cases} \quad \text{and} \quad H^-(u) = \begin{cases} 0 & \text{if } u > 0 \\ u^2/2 & \text{if } u \leq 0. \end{cases}$$

5.1.1 Linear Initial Condition

As the first test, linearly varying values, $u_0(\underline{x}) = ax$, are used as the initial condition. The analytic solution for this initial condition is

$$u(x,t) = \frac{ax}{1+at}.$$

The solution is always varying linear in space and decaying everywhere with time. Since the numerical method presented in Chapter 4 is of second order accuracy in space, it can be expected that the model reproduces this linear solution very accurately. Additionally, a sudden change of resolution in the computational grid can be expected not to introduce any significant effects. For the simulations conducted in this section $a = 1$ was chosen, but its value is of no importance for the conclusions to be drawn from the tests.

As a measure of simulation quality the relative error,

$$e := \frac{u_h - u}{u}, \tag{5.2}$$

of the discrete solution u_h with respect to the analytical solution u is evaluated at different points along the x axis in regular time intervals. As time advances, this measure becomes increasingly sensitive to small deviations from the solution because the solution itself is monotonically decaying with time.

Simulations have been conducted on different grids. The base grid divides the domain $[0; 1]$ into 10 equally sized intervals, while the non-equidistant grids have an abrupt change in resolution at $x = 0.5$ by factors 5, 25 or 125 relative to the basic resolution of 0.1, respectively.

As an example of the simulation results the development of the relative error at $x = 0.55$ for a simulation with an increase in resolution by a factor of 125 for $x > 0.5$ is shown in Figure 5.1. Overall, the magnitude of the relative error is very low. It reaches its peak with an error still below 0.03% after about 2000 time steps. From then on, the simulation approaches the limit of $u = 0$ for $t \to \infty$ faster than the analytical solution so that the the relative error approaches zero and its magnitude decreases again. No significant oscillations can be observed in the time series. The error never changes sign and does not even overshoot as it approaches zero.

The other simulations on all of the different grids yield the same solution up to at least six decimal digits. So in the case of a solution varying at most linearly in space the numerical scheme displays the behaviour expected of a true second-order-in-space method: the computed discrete solution is completely independent of the grid resolution. Moreover, this test demonstrates that an abrupt change in resolution—even by large factors—is of no consequence for the accuracy of the solution computed if it varies only linearly in space at the point of changing resolution.

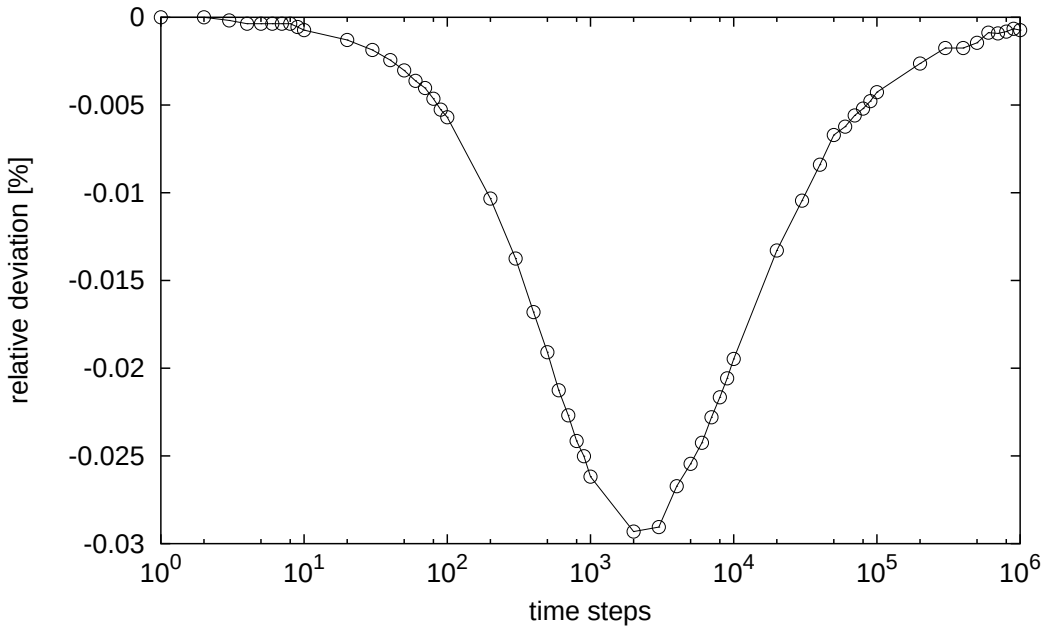

Figure 5.1: Relative error for initial condition $u_0(\underline{x}) = x$.
The time series is taken at $x = 0.55$ on a grid with a cell width of 0.1 for $x < 0.5$ and 0.0008 for $x > 0.5$, so the resolution increases abruptly by a factor 125 in the middle of the domain.

5.1.2 Non-linear Initial Condition

The obvious next step is to examine solutions varying more than linearly in space. For this purpose the initial condition $u_0(\underline{x}) = ax^2$ is used. The analytical solution is

$$u(x,t) = \frac{\left(\sqrt{4atx+1}-1\right)^2}{4at^2} \tag{5.3}$$

for $t > 0$, which is clearly not varying linearly in space. The numerical method cannot resolve this spatial variation as exactly as in the linear case. Consequently, the resolution of the computational grids becomes important and differences between the solutions on the different grids are to be expected. The constant is again chosen as $a = 1$.

The relative error e as defined in (5.2) is calculated with the analytical solution (5.3). Time series for e at $x = 0.45$ and $x = 0.55$ are shown in Figures 5.2 and 5.3, respectively. As expected, the approximate solutions for this case deviate further from the analytical solution (5.3) than in the linear case on the same computational grids (Figure 5.1). Also, there are now some obvious differences between the solutions obtained on the different grids.

- At $x = 0.45$, solutions on the uniform grid and the grid refined in the right half are identical. With the grid refined in the left half, a much more accurate solution is obtained at this location due to the much higher resolution.
- At $x = 0.55$, the error on the grid refined left of 0.5 is larger than at $x = 0.45$ but still drops much faster than for the uniform grid and the grid refined right.
- Also at $x = 0.55$, the error of the simulation on the grid refined right of 0.5 exhibits a jump from values near zero to almost -2.5% and after that it monotonically approaches the curve for the uniformly resolving grid. Before the jump occurs, the error is actually smaller than in either the uniform or refined left case due to the much smaller initial discretisation error at this location on this grid.

Probably the most prominent feature of the results shown is the jump of the error at $x = 0.55$ for the grid refined right of 0.5. It demonstrates an important property of discretisations of hyperbolic equations: not only the solution is transported in space with time but also the errors of the approximation. In this case, all information—approximate solution *and* error—is moving strictly from left to right. Therefore, the error shows a clear signal when the high initial discretisation error from the coarsely resolved left half of the domain reaches the cell at $x = 0.55$. The initial discretisation error in the left half of this grid is the same as of the coarse uniform

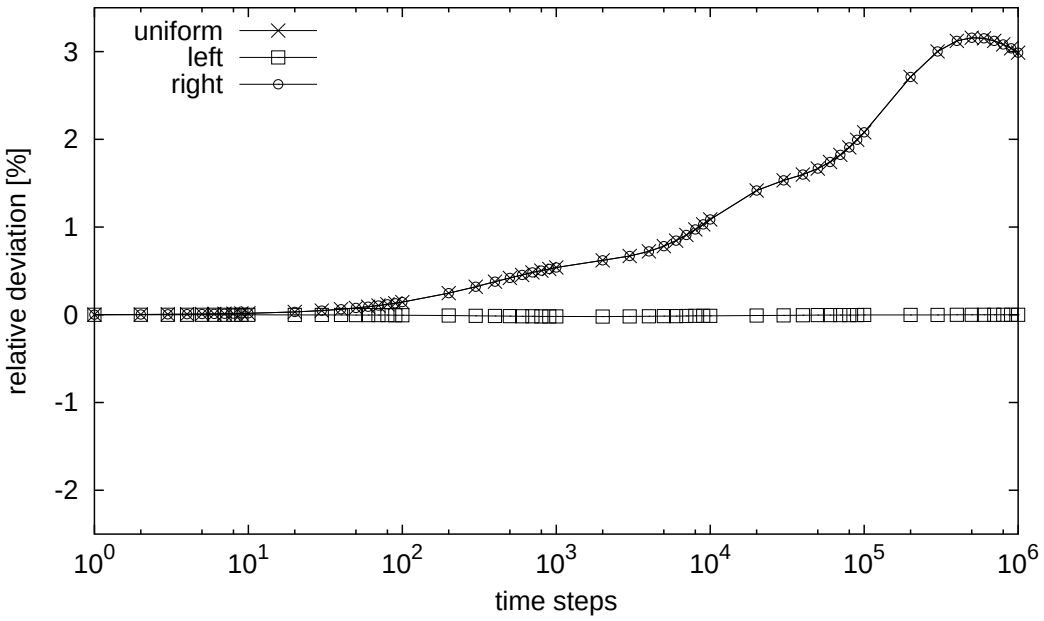

Figure 5.2: Relative error at $x = 0.45$ for initial condition $u_0 = x^2$.
Shown are time series for three grids of different resolution: a uniform resolution of 0.1 (marked with '×'), a 125 times higher resolution for $x < 0.5$ (marked with '□') and a 125 times higher resolution for $x > 0.5$ (marked with '○').

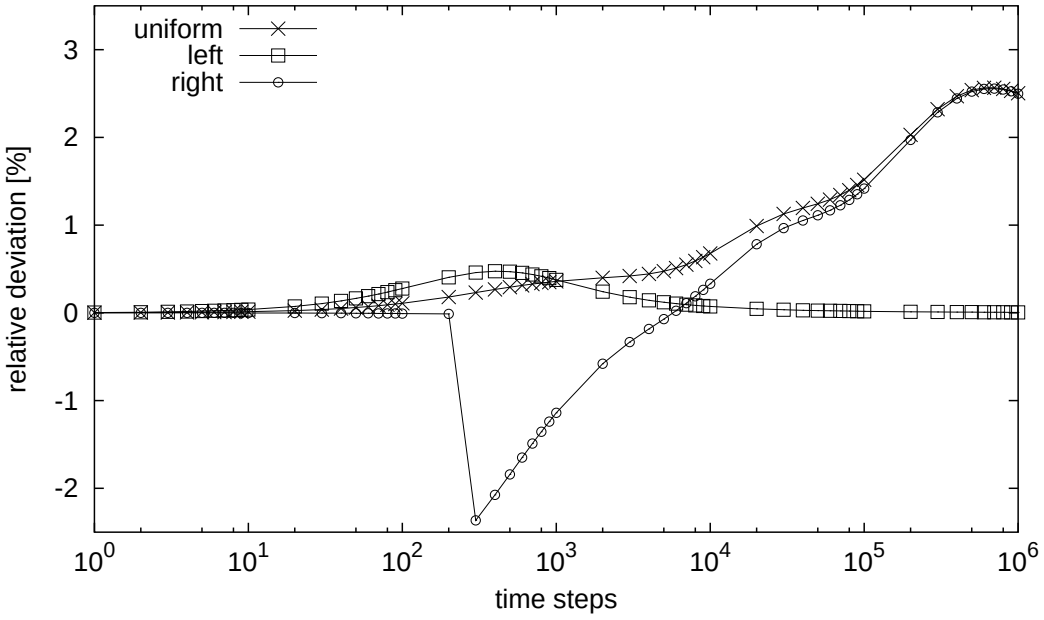

Figure 5.3: Same as Figure 5.2 at $x = 0.55$.

grid so that the series converges towards the error of the simulation on the uniform grid after the high error reaches the position $x = 0.55$.

Likewise, the error for the grid refined left of 0.5 stays low even in the right unrefined half because, as time advances, more and more of the local error is *originating* from the refined left half. Also, the refinement of the right half is of no consequence at all in the left half because all information—solution and error—is transported only rightwards. Accordingly, at $x = 0.45$ the grid refined right of 0.5 and the coarse grid produce exactly the same result.

A first conclusion from the tests with the Burgers equation is that for abrupt changes in resolution the presented method may generate artefacts, such as the jump in the relative error discussed above, for solutions varying more than linearly in space. But unlike other methods in similar tests (Schlünzen, 1988), it remained stable in all conducted simulations. Even for changes in resolution by a factor of 125, the method performed not worse than in the case of the uniform resolution—modulo artefacts, of course. The limit $u \equiv 0$ for $t \to \infty$ is reproduced in all simulations as far as machine accuracy permits, but with the fixed time step this takes very long. With a time step adjusted to ensure a maximum local CFL number (Section 4.3) steps become very large as the signal velocity u becomes smaller. Following this strategy, solutions approach zero up to machine precision within a more reasonable time as shown in Figure 5.4 for $\mathsf{CFL} \leq 0.95$.

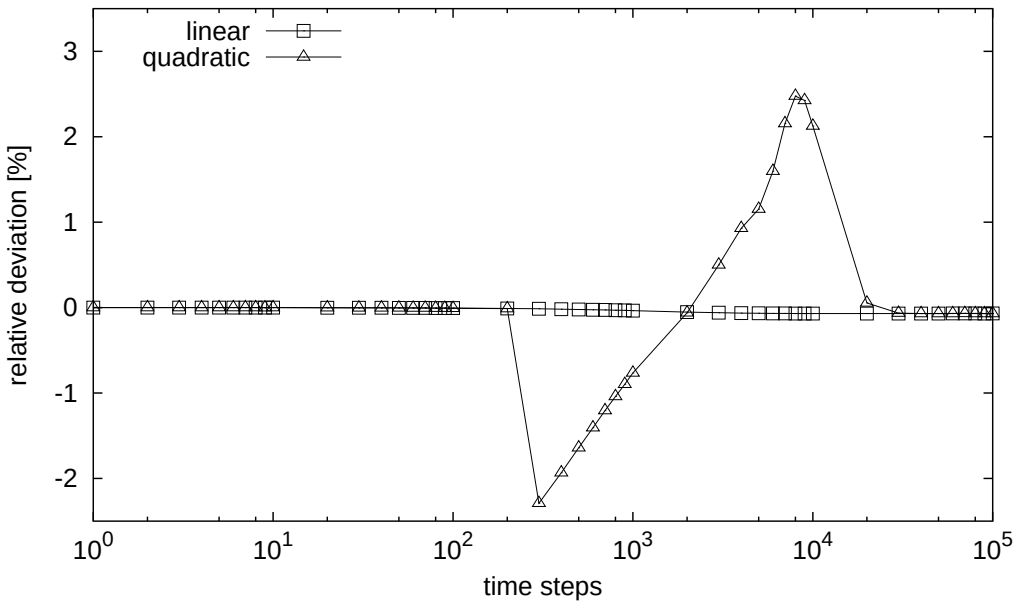

Figure 5.4: Relative error at $x = 0.55$ computed with a variable time step. Shown are the time series for simulations with linear and quadratic initial conditions on a grid with a cell width of 0.1 for $x < 0.5$ and 0.0008 for $x > 0.5$.

5.2 Density Current

A more complex test that uses the full equations (4.1) in two spatial dimensions, assuming $\partial_{x_1} \equiv 0$, is presented in this section. The density current scenario used here is one of the popular tests to be found in the literature (e.g. Carpenter *et al.*, 1990; Skamarock and Klemp, 1993). It has several similarities to the land-sea breeze which was described and simulated in Chapter 2. Indeed, land-sea breezes are a form of density currents (Simpson, 1987). Very typical features are the flow head and the front at its leading edge. Behind the head billows are formed due to Kelvin-Helmholtz instability, finally resulting in a fully turbulent wake mixing the density current and the return flow above. But in contrast to the sea breeze, the driving density difference is given in the initial conditions and does not develop with time by heating.

The frame of reference used for the simulations is the local tangential (x, y, z)-system described in Section 3.1.1. For this scale of the simulations it is very common and usually employed in local high-resolution atmospheric models, e.g METRAS (Schlünzen, 1988, 1990; Schlünzen *et al.*, 1996), ARPS (Xue *et al.*, 1995, 2000), GESIMA (Kapitza and Eppel, 1992), MM5 (Grell *et al.*, 1995) and many others. The computational domain for this scenario is a 40 km wide and 10 km high rectangle. Due to the relative smallness of the domain (Section 3.1.5) and to keep the scenario simple, the Coriolis force is neglected. The top boundary is a rigid lid with a 3000 m thick damping layer underneath, and the bottom boundary is treated as a solid wall. At the right boundary, the boundary normal gradients of all fields are forced to vanish (i.e. it is a zero-gradient open boundary), and the left boundary is another solid wall.[1] The atmosphere is initially at rest, and a 15 km wide pool of cold air resides in the lower left corner of the domain. The ambient air outside the cold pool is neutrally stratified and has an initial potential temperature of 300 K. At the bottom the air inside the cold pool is 10 K colder than the ambient air, and its potential temperature is linearly increasing to 300 K in a height of 5 km, i.e.

$$\Theta(\underline{x}, 0) = \begin{cases} 300\,\mathrm{K} - 10\,\mathrm{K}\frac{H-x_3}{H} & \text{for } x_2 < 15\,\mathrm{km} \text{ and } x_3 < H \\ 300\,\mathrm{K} & \text{otherwise} \end{cases}$$

with $H = 5000\,\mathrm{m}$. The initial pressure, temperature and density fields are in hydrostatic balance leaving a discontinuity in all three fields at $x_2 = 15\,\mathrm{km}$ and $0 \leq x_3 < H$.

To compare the simulation results gained with a3m to the results published by Carpenter *et al.* (1990) and Skamarock and Klemp (1993) turbulence is neglected. For

[1] Carpenter *et al.* (1990) use a mirroring condition and Skamarock and Klemp (1993) use a wall. Both conditions are very similar.

the density current scenario described above, advection is the dominating process anyway. This is demonstrated in Section 5.2.3, where turbulence is included, and the exchange coefficients are determined by the mixing length approach described in Section 3.3.

5.2.1 Simulation on a Grid of Uniform Resolution

The density current described above is simulated on a grid with a uniform resolution of 250 m in both directions. Figures 5.5, 5.6 and 5.7 show the simulated potential temperature and wind fields of the density current after 300 s, 600 s and 900 s integration time. The obtained results are in good overall agreement with the expectations from other numerical simulations (e.g. Carpenter *et al.*, 1990; Skamarock and Klemp, 1993) at this resolution. A qualitative comparison with laboratory experiments (e.g. Simpson, 1987) suggests that the resolution is too low for capturing all the details of a density current, e.g. the Kelvin-Helmholtz billows. But the simulation result shows a density current clearly separated from the ambient air at its leading edge, the front. The front is sharp and approximately four cells wide which matches the expectations for the numerical scheme (Spekreijse, 1988).

However, the flow head is not well developed in the simulation. The results of Skamarock and Klemp (1993) show a noticeably better developed flow head, and the head is very prominent in the results of Carpenter *et al.* (1990). But Carpenter *et al.* use the PPM, a third order method. So results should be superior to those obtained with the second order schemes of Skamarock and Klemp and a3m. Skamarock and Klemp use central three-point differences which have a very low numerical diffusion. Indeed, they have to add some amount of artificial diffusion to control "numerical noise". Nevertheless, their simulation results show a significant amount of non-physical oscillations, especially behind the front. The ENO scheme of a3m is inherently more diffusive at those spots to suppress artificial oscillations. This can be observed in the velocity components (Figures 5.5–5.7, middle and bottom panel): some of the density currents momentum 'escapes' and the lighter air in front of it is pushed rightwards instead of being displaced upwards by the current. Consequently, the development of the density current is somewhat less intense, but the results are almost free of non-physical oscillations.

The speed of the front can be roughly estimated by budgeting potential and kinetic energy as $u_{\text{front}} \approx (2 \int_0^H \Delta\rho z \, dz / \int_0^H \rho \, dz)^{1/2} \approx 22\,\text{m/s}$. In the simulation, the front advances with roughly 16.5 m/s in the interval from 300 s–600 s and 17 m/s from 600 s–900 s which is lower that the theoretical value above. But in the estimation all potential energy of the cold air is converted into uniform horizontal movement. This is clearly not the case as a part of the energy is to be found in the vertical

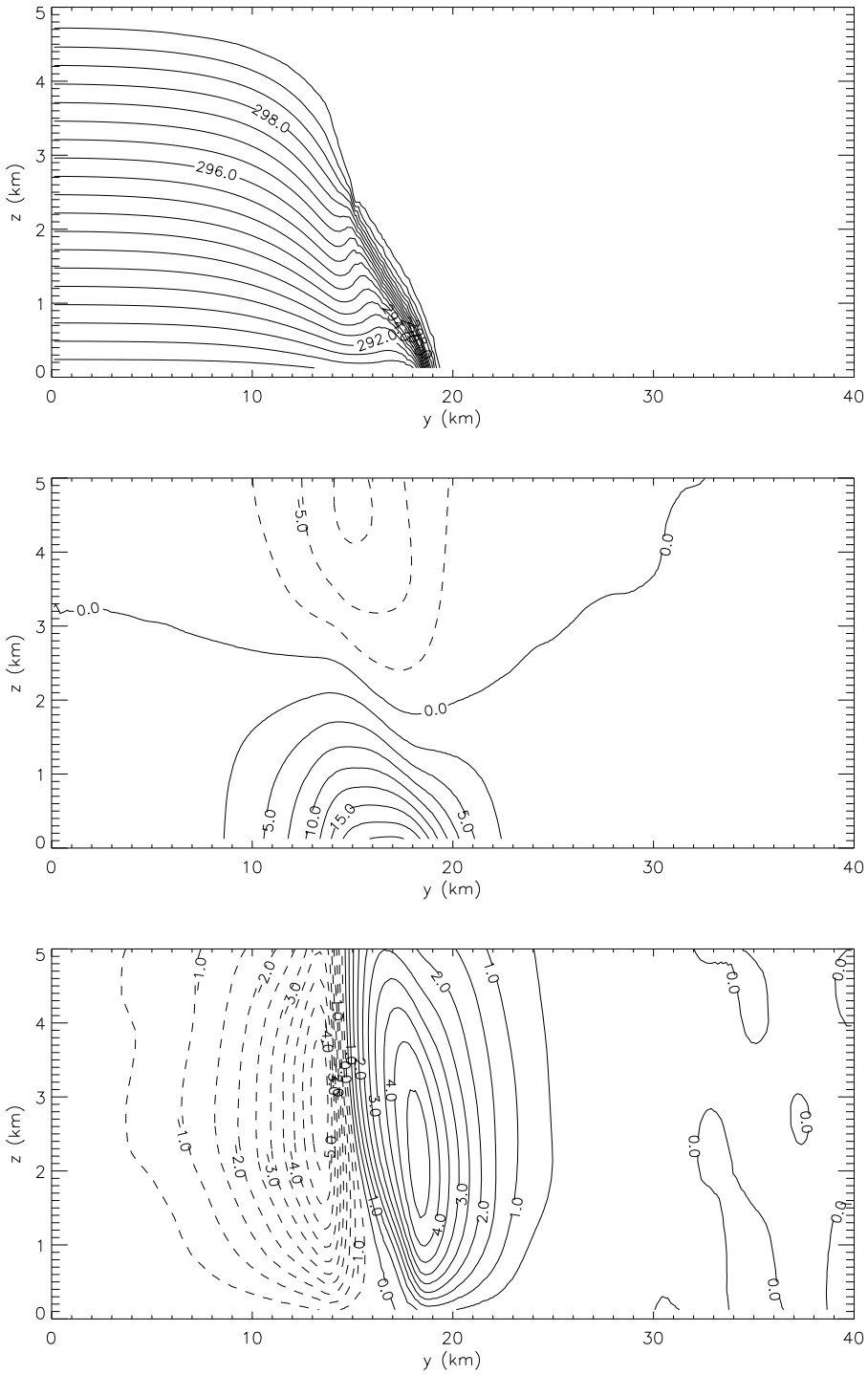

Figure 5.5: Density current after 300 s integration time.
The simulation was conducted on a grid of 250 m resolution. The top panel shows the potential temperature field with an interval of 0.5 K between contour lines. The middle and bottom panels show the horizontal and vertical wind with intervals of 2.5 m/s and 0.5 m/s between contour lines, respectively. Note that only the lower half of the domain is displayed here.

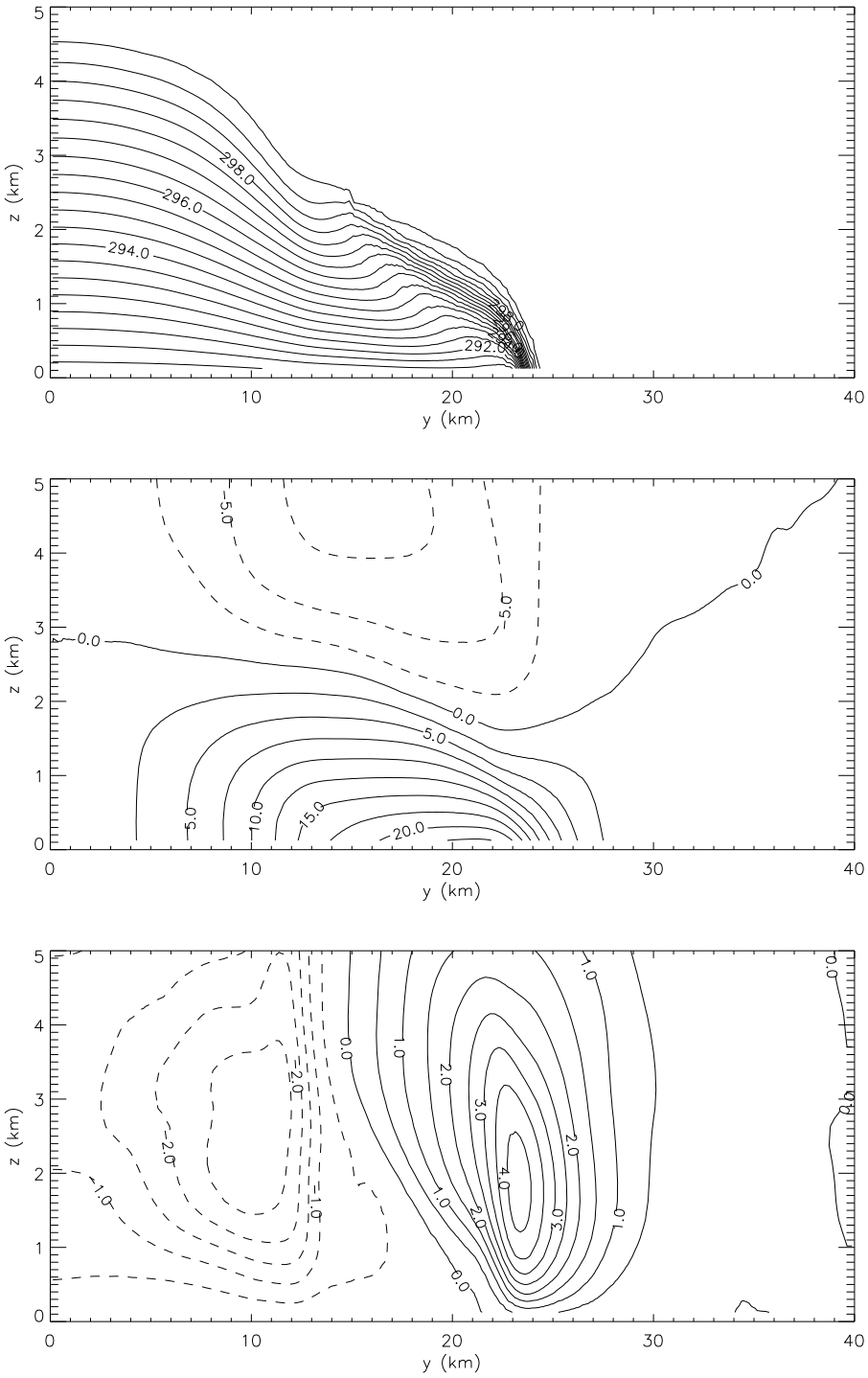

Figure 5.6: Same as Figure 5.5 after 600 s integration time.

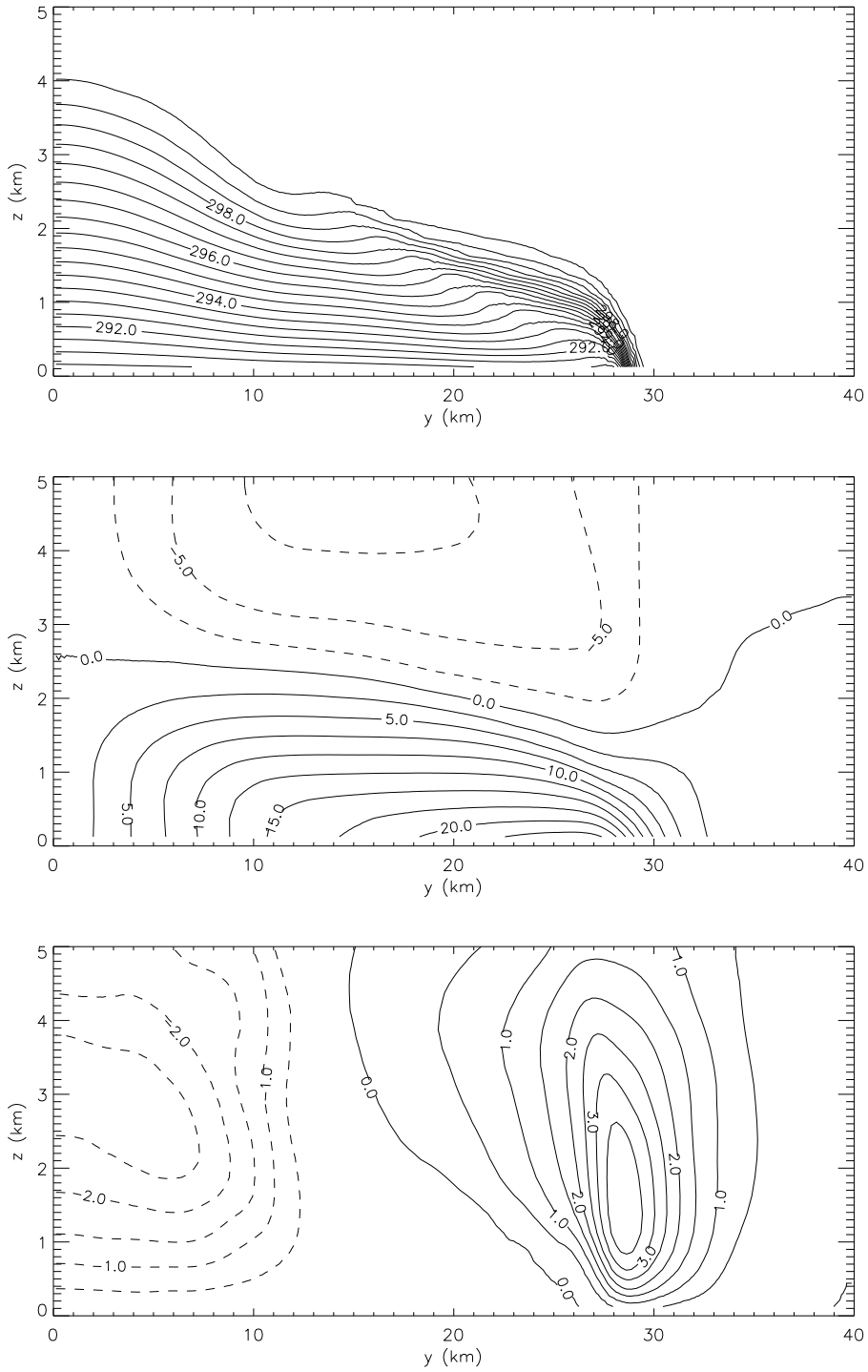

Figure 5.7: Same as Figure 5.5 after 900 s integration time.

movement as well. Also, the horizontal movement is far from homogeneous within the density current: the maximum horizontal velocities are 22.1 m/s and 23.3 m/s in the same time intervals as above, partially exceeding the estimated value of the front speed. And the flow is still accelerating at 900 s. In comparison with the numerical simulations of Skamarock and Klemp (1993) the density current advances slightly slower.

An interesting effect which cannot be reproduced with atmospheric models that eliminate sound waves from the equations by employing the anelastic approximation is a net outflow of mass from the model domain induced by the equalisation of pressure from the initial discontinuity. The process is similar to that found in a simple shock tube (e.g. Sod, 1978). On release, a shock wave moves from the initial pressure discontinuity into the region of the low pressure (rightwards in this case), and an expansion or rarefaction wave moves from the initial discontinuity into the high pressure part (leftwards in this case). Both waves move with the speed of sound relative to the flow as particles in the low pressure region are successively subjected to the effects of the higher pressure on the other side of the discontinuity and vice versa. Both waves incorporate a movement of air towards the lower pressure region, i.e. rightwards in this case. The shock wave moves out of the domain through the open right boundary, while the rarefaction wave is first reflected on the left boundary before following the shock wave out of the domain. All of this takes only approximately 160 s. Figure 5.8 shows the time history of the total mass in the model domain and the accumulated mass flow leaving the domain through the open boundary. The mass in the domain exhibits a rapid drop caused by the pressure exchange via shock and rarefaction wave occurring within the first seconds after simulation start. After that, only a small net rightwards movement remains, which causes the following slower outflow of mass. Models using anelastic equations cannot reproduce such expansions or, similarly, compressions since they do not allow divergences in the momentum field: the total mass in the domain of an anelastic model is constant. They also propagate pressure information with infinite and not the speed of sound with an elliptic equation for pressure, making pressure equalisations instantaneous.

While this effect itself is not directly of meteorological importance, these observations shed light on the different mechanisms by which pressure information spreads within different models. Thereby, a3m permits a change in e.g. the mean surface pressure (area weighted average over the whole lower domain boundary) according to the larger-scale conditions imposed at the boundary, which is more interesting from a meteorological point of view. Anelastic limited area models cannot simulate such effects. In case of the density current, the average surface level pressure drops by 2 hPa within 20 minutes, where about 1.5 hPa are due to the initial expansion, and the remaining 0.5 hPa are due to the longer term outflow from the domain.

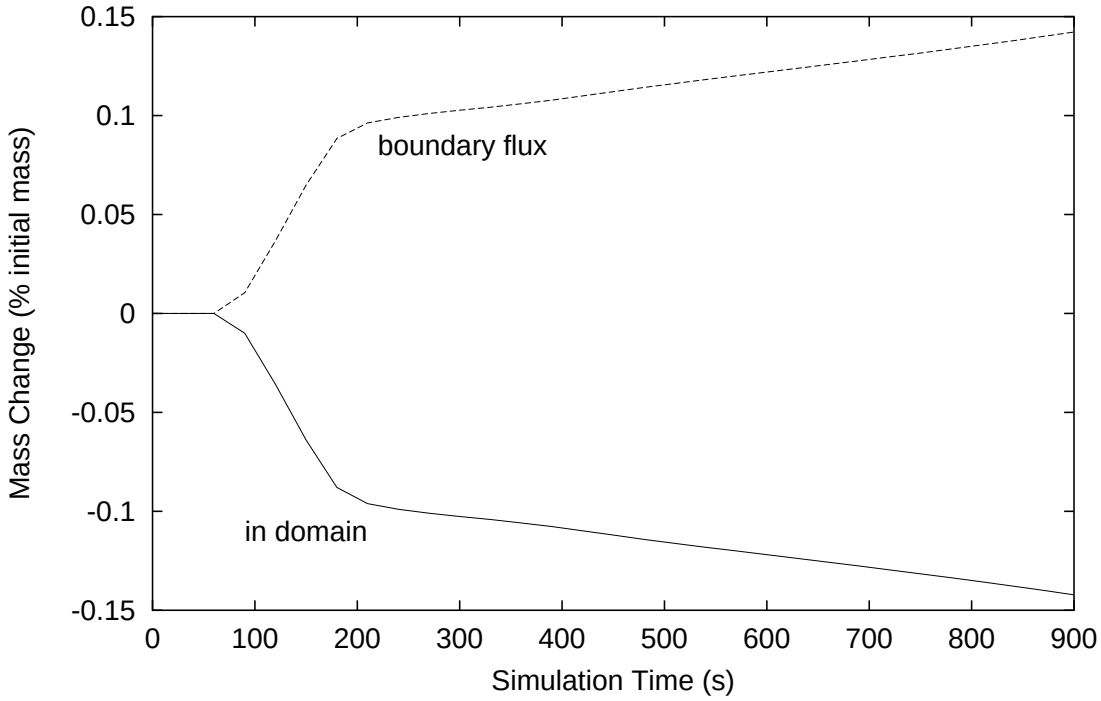

Figure 5.8: Evolution of mass in the domain and boundary flux with time.

5.2.2 Simulations with Locally Refined Grids

Simulating the same density current with a locally refined grid should yield more detail in the structure of the current, especially the flow head. The grid used for this simulation consists of four blocks with cell sizes of $250 \times 250\,\mathrm{m}^2$, $250 \times 125\,\mathrm{m}^2$ and $125 \times 125\,\mathrm{m}^2$. Figure 5.9 shows a sketch of the grid arrangement.

Figure 5.10 shows the potential temperature and wind fields for this grid, 900 s after the cold air has been released. The front—though still about four cells wide—is now substantially thinner than with the uniform 250 m grid spacing due to the enhanced resolution. Also, more of the small-scale motion is explicitly captured with the locally refined grid. Formation of billows in the flow head is setting in: the increased shear is beginning to 'roll up' the isotherms. The circulation as a whole intensifies significantly. The maximum velocity in the density current is 25.1 m/s, about 1.5 m/s higher than in the simulation with 250 m resolution. The up-draft is much stronger than in the reference simulation: the simulation with refinement yields 6.1 m/s, an increase by more than a factor 1.5 relative to the up-draft of 3.9 m/s for uniform resolution simulation. The front is advancing roughly 1 km further than in the uniformly resolved reference simulation.

Locally increasing the resolution further again amplifies the stress and thus intensifies the deformation of the isotherms. Figure 5.11 shows the results of a simulation

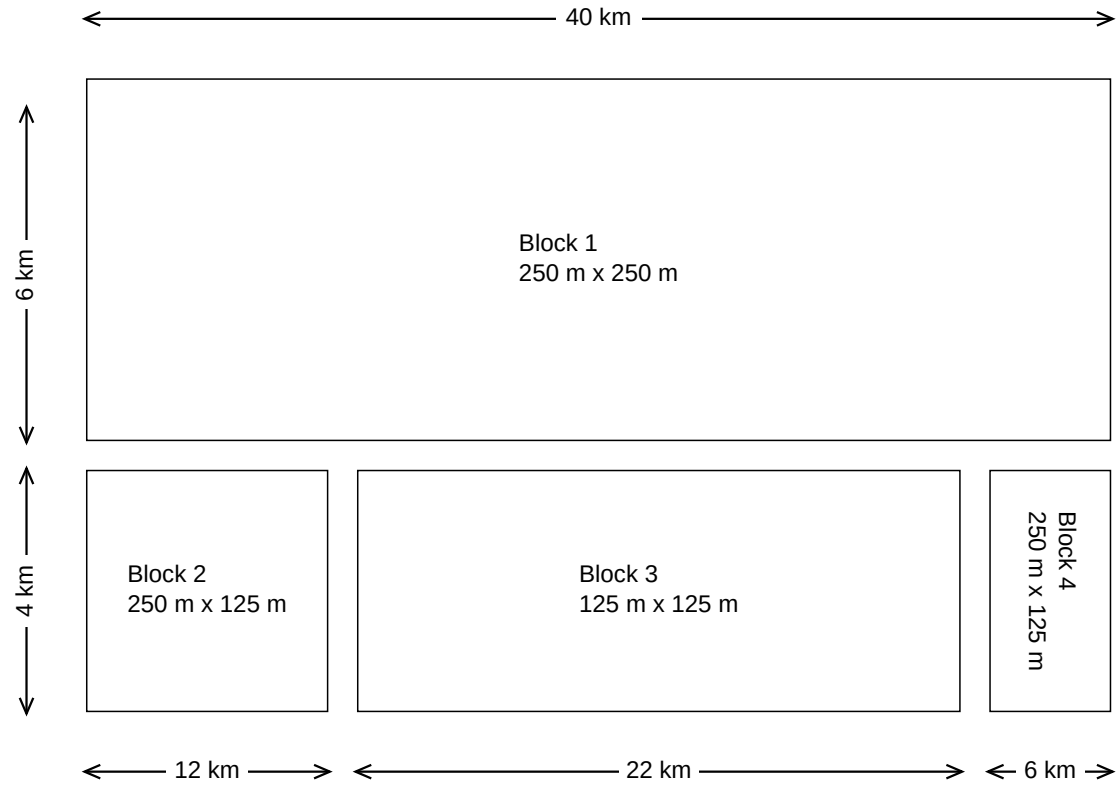

Figure 5.9: Locally refined block-structured grid.

with the same grid arrangement as sketched in Figure 5.9 but with a resolution of $62.5 \times 62.5\,\mathrm{m}^2$ in block 3 and a resolution of $250 \times 62.5\,\mathrm{m}^2$ in blocks 2 and 4. In this simulation, the maximum velocity is 24 m/s, not even 0.5 m/s higher than in the uniformly resolved simulation and lower than in the previous simulation with a locally refined grid. But the horizontal velocity is relatively high in a relatively large part of the current, and the shear at its top has increased significantly. The up-draft is now 7.2 m/s, which is an increase by more than a factor 1.8 relative to the reference simulation. Relative to the previous simulation with a locally refined grid the updraft is 1.1 m/s higher, i.e. almost 20%.

The increased stress at its top slows the density current down: the shear is almost strong enough to trigger Kelvin-Helmholtz billows. The increase in the up-draft at the front is due to the increase of horizontal convergence that can be resolved by the finer grid. Also, the velocity components exhibit wave patterns in this simulations. The vertical wind shows alternating up and down-drafts and there are regions of accelerations and decelerations in the horizontal velocity within the density current as well as the return flow above. However, no closed vortices emerge.

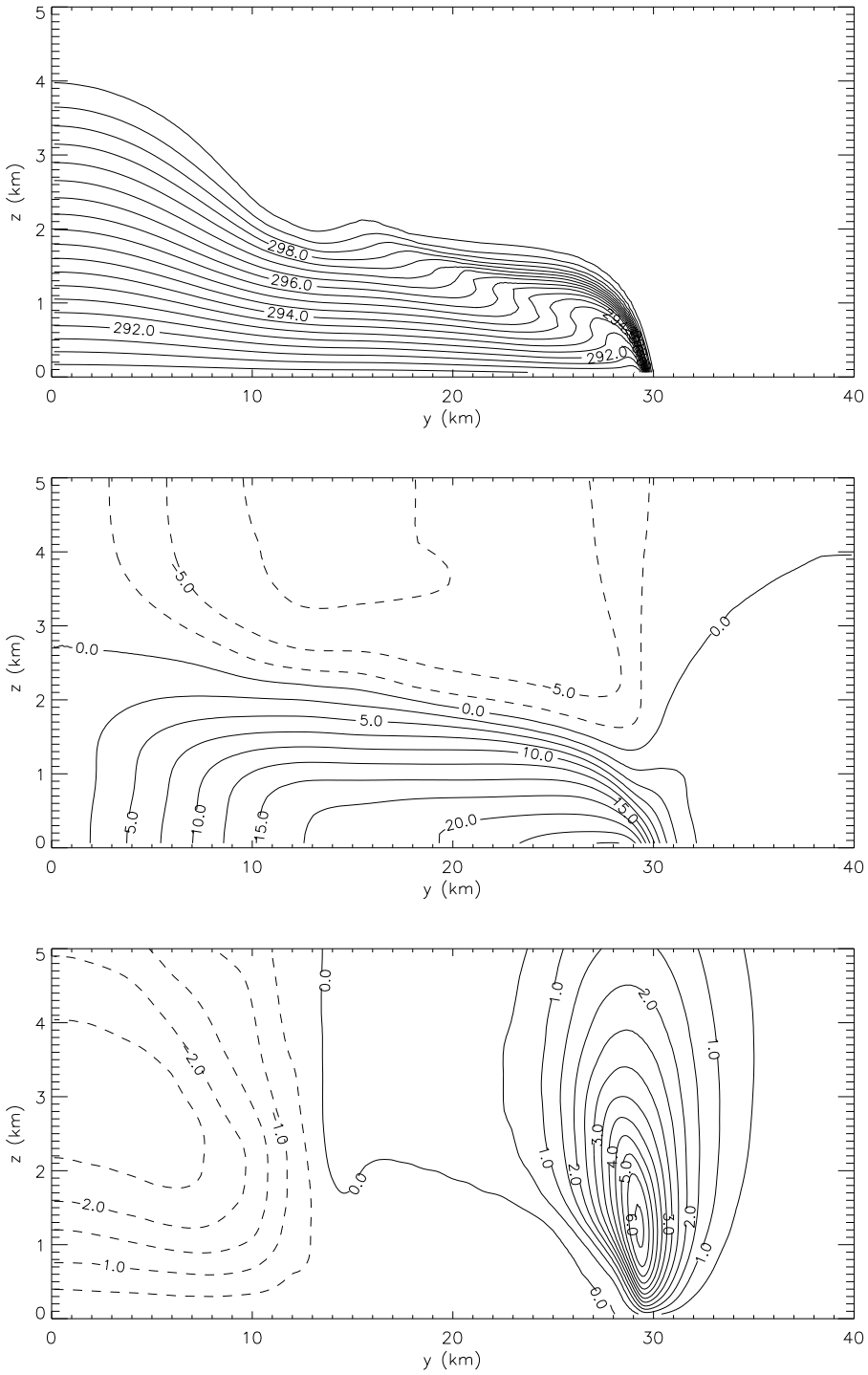

Figure 5.10: Same as Figure 5.7 simulated on a locally refined grid.

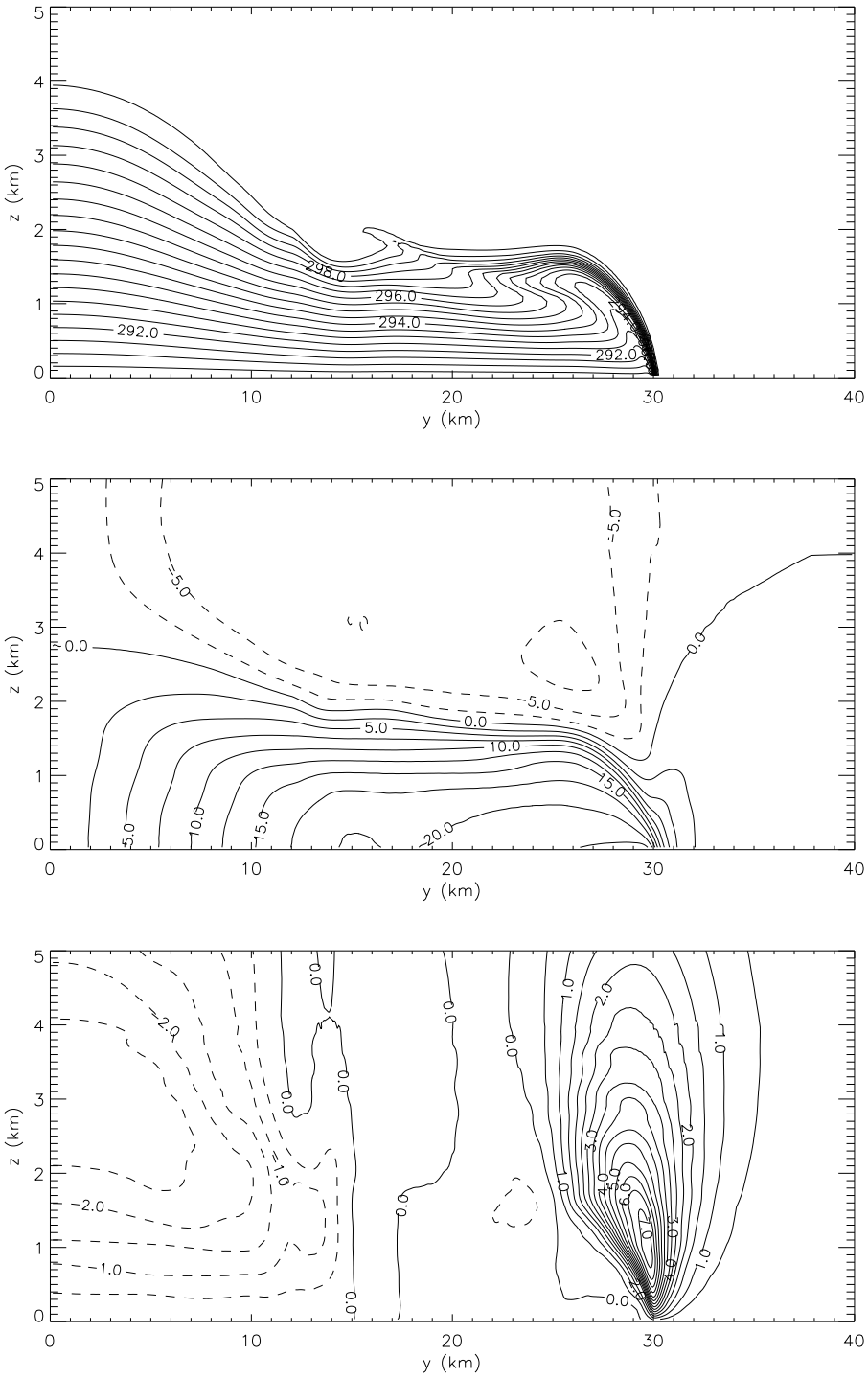

Figure 5.11: Same as Figure 5.10 with stronger refinement.

5.2.3 Simulation with Turbulence

As mentioned above, the density current is mostly dominated by advection. A simulation of the density current, where the effects of turbulence are parameterised by the mixing length approach as described in Section 3.3, yields very similar results to those discussed in Section 5.2.1. The exchange coefficients gained from the mixing length turbulence parameterisation are shown in Figure 5.12. The exchange coefficients are highest in the region behind the front due to the strong shearing between the advancing density current and the evading flow of ambient air.

The potential temperature and wind field of the simulatiuon with turbulence are shown in Figure 5.13. In comparison to Figure 5.7, no obvious differences in the simulation results can be seen. An analysis by hit rates, as already done for the land-sea breeze in Chapter 2, also reveals only very small differences: hit rates for the velocity components relative to the reference simulation—using the same thresholds as in Section 2.2—are 100% for the horizontal velocity and 98% or better for the vertical velocity over the simulated period of time. Decreasing the threshold for the horizontal velocity component from 0.5 m/s to 5 cm/s yields 99 misses out of 6400 cells after 900 s integration time, i.e. a hit rate of more than 98%.

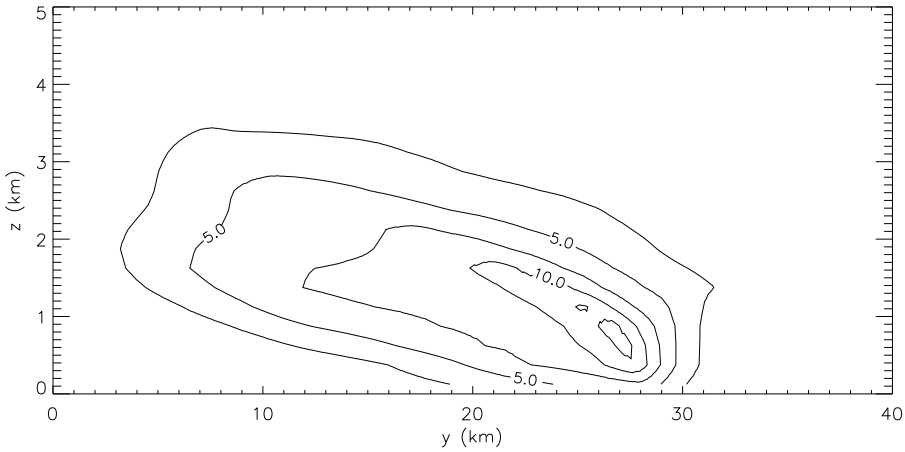

Figure 5.12: Exchange coefficient for the mixing length turbulence parameterisation.

5.3 Numerical Stabilisation Techniques

Despite strict adherence to the *linear* stability requirements like the CFL criterion (4.20) after Courant, Friedrichs and Lewy (1928) or equivalents, various numerical methods exhibit *non*-linear instabilities: they tend to accumulate energy in spurious, non-physical, very short waves (e.g. Lax and Wendroff, 1960; Hirsch,

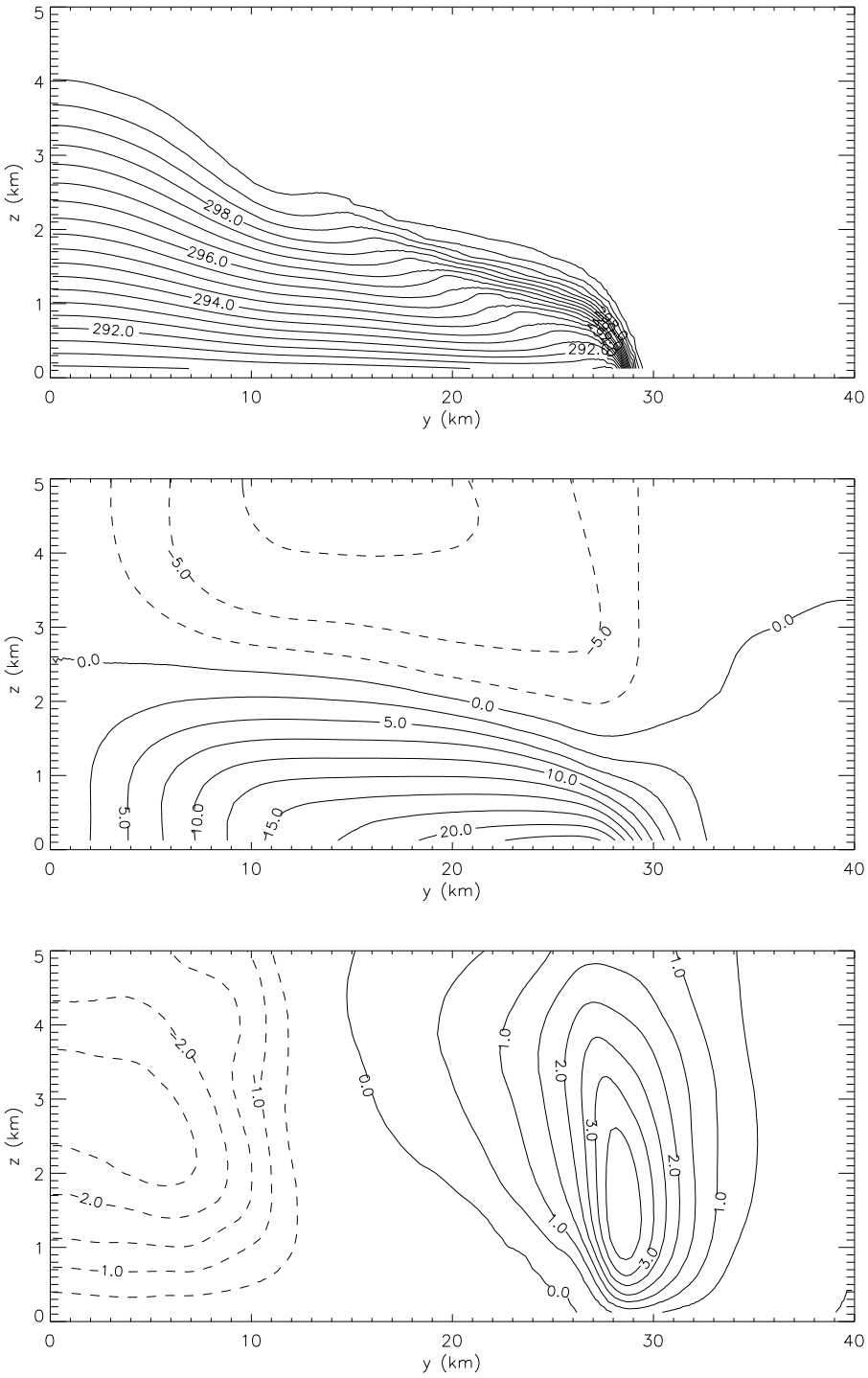

Figure 5.13: Same as Figure 5.7 with turbulence parameterisation.

1988b). Roache (1982) shows an illustrative example where the (perfectly valid) solution of the discretised difference equation exhibits a qualitatively very different behaviour from the analytic solution of the differential equation: it oscillates while the analytic solution is monotonic. For the highly non-linear numerical scheme of a3m introduced in Chapter 4, the classical methods of analysis used e.g. by Lax and Wendroff (1960), Mesinger and Arakawa (1976) and Roache (1982) are not applicable to determine its oscillatory properties. But it is known to cause some amount of artificial oscillations near discontinuities or very sharp gradients due to the ENO interpolation (e.g. Sonar, 1997). Consequently, two stabilisation techniques already in active use in different atmospheric models are examined towards their suitability to be combined with the methods of a3m in this section. The first method is known as computational mixing—as opposed to physical mixing by the turbulence parameterisation—or simply as filtering. The second is called divergence damping.

5.3.1 Computational Mixing

Numerical schemes which are not monotonicity preserving may introduce artificial oscillations into an approximate solution. These oscillations are absent in the real solution and without any physical meaning. This is especially a problem for the construction of higher-order methods by centred differences (e.g. Lax and Wendroff, 1960; Tremback *et al.*, 1987; Hirsch, 1988b; Bott, 1989, 1992) or other means (Section 4.1.1.2 and e.g. van Leer, 1979; Smolarkiewicz, 1983; Harten, 1983; Harten *et al.*, 1987; Shu, 1987; Hirsch, 1988b; Sonar, 1997). These oscillatory schemes tend to accumulate energy in the very short waves which may distort or even dominate the approximate solution with time. This problem has been extensively investigated for the case of the shallow water equations by Mesinger and Arakawa (1976). The method most frequently used in atmospheric models to compensate these shortcomings of the numerical schemes is computational mixing. It is employed e.g. by METRAS (Schlünzen, 1988; Schlünzen *et al.*, 1996), ARPS (Xue *et al.*, 1995, 2000) and LM (Doms and Schättler, 1999).

Computational mixing is achieved by a smoothing operation implemented by the application of a low-pass filter in computational (i.e. index) space. Let the grid have a uniform spacing and a_i be the value of variable a at index i. The filter of n^{th} order (for one spatial dimension) is defined by

$$\overline{a}_i^n = a_i + (-1)^{n-1}(\Delta x/2)^{2n}\,\delta_{2n}a_i, \tag{5.4}$$

where $\delta_{2n}a_i$ is the finite difference analogue of the $2n$th partial derivative in space. Filtering in more spatial dimension is achieved by consecutive application of the filter for each direction. This class of filters exactly cancels out waves of wavelength

equal to twice the grid spacing (Shapiro, 1970). Writing the filter in the above form clearly shows the connection between the filter operator and a diffusional term involving the $2n$th derivative in space. For $n = 1$ it is equivalent to the normal diffusional mixing of viscous flows with an exchange coefficient of $\Delta x^2/(4\Delta t)$ (Shapiro, 1971). A term of similar structure is employed for modelling the effects of turbulence (Section 3.3). Despite this equivalence, the above filter should not be interpreted in a physical way: its sole purpose is to eliminate the artificial short wave oscillations introduced by the numerical schemes employed in the models utilising these filters. Assuming uniform grid spacing, equation (5.4) yields the three points wide stencil $\overline{a}_i = (a_{i-1} + 2a_i + a_{i+1})/4$ for $n = 1$ and the five points wide stencil $\overline{a}_i^2 = (-a_{i-2} + 4a_{i-1} + 10a_i + 4a_{i+1} - a_{i+2})/16$ for $n = 2$. Atmospheric models usually consider computational mixing in index space and apply the above formulas also for non-uniform and non-Cartesian grids. Consequently, the correspondence to spatial derivatives is not always exact and depending on the deformation of the grid. The properties of filters (5.4) and their application to modelling atmospheric flows are discussed in detail by Shapiro (1970, 1971).

Although they are physically meaningless, the influence of these spatial filters on the solutions obtained may be significant. This is of course partially intended since they remove artificially generated short waves from the solution. On the other hand, their effect on the predicted variables may be of the same order of magnitude as all other terms which do correspond to physical processes (Hebbinghaus, 2002, pers. communication). The filters impact on longer waves, which are otherwise represented correctly within reasonable bounds by a numerical scheme, is most strongly dependent on the order of the filter: for increasing n the impact on waves longer than twice the grid spacing becomes lower. This is of special interest within this work since (near) discontinuities, such as atmospheric fronts, are—if analysed spectrally—composed of waves of all wavelengths. Accurately resolving fronts is one of the main objectives of the new model developed. Consequently, artificial reduction of gradients—smearing out the fronts—by a filter is not acceptable.

To test the inter-operability of the basic numerical scheme introduced in Section 4.1 with the low-pass filter (5.4), the density current is simulated with application of the filter and compared to the simulation without filter (Section 5.2). The density current exhibits a front with gradients which are strong enough to trigger artificial oscillations in schemes susceptible to this numerical phenomenon as can be seen in the simulation results of Skamarock and Klemp (1993). Moreover, it is initialised with a real discontinuity in the prognostic variables ρ and $\rho\mathsf{e}$. Each time step the filter was applied to the velocity components u_1, u_2 and u_3 in horizontal direction only.

Results of the simulation with a first order filter, $n = 1$ in (5.4), after 900 s integration time are shown in Figure 5.14. With filter, the temperature gradient in the front is significantly reduced compared to the results of the 'normal' simulation without filter (Figure 5.7). Although the temperature field itself is not filtered, the front is now about seven cells wide, more than a factor 1.5 compared to the un-filtered simulation. The current is advancing slower than in the reference simulation: after 900 s it is more than 2 km behind, i.e. approximately 15%. Additionally, the shape of the current has changed. The front has a much smaller angle to the ground than without filtering so that the current is more pointed or wedge-shaped and there is no flow head at all.

The influence of the filter is even more pronounced in the wind field since velocity is a variable directly effected by the filter. Without filter, the velocity exhibits clear and very condensed extrema, the flow direction is well aligned with the front, and strong shearing occurs across it. In the simulation results with 3-point filter, the extrema are much more spread out and significantly further apart resulting in a rather weak shearing. Also, the flow shows a different pattern than without filter: the expected circulation is still present, but the wind and temperature fields are aligned to a much lesser degree than in the simulation without filter. Especially the region of up-draft at the front is smeared by the filtering while it is very confined in the 'normal' simulation without filter. The flow velocities are much smaller than in the simulation without filter. The density current has a maximum velocity of 19.3 m/s, 18% lower than in the reference simulation. The up-draft is only 1.6 m/s as opposed to almost 4 m/s in the 'normal' case, a drop by 60%. Also note that the velocity in the leftmost cells (at the solid boundary) is left unchanged by the filter. As a consequence, the maximum down-daft in the region of the initial cold pool is not immediately at the boundary anymore but shifted into the domain.

From these results, the effects of filtering seem disastrous. However, with $n = 1$ the filter (5.4) constitutes an additional mixing by second derivative, i.e. like molecular diffusion or the turbulence parameterisation (3.12). But the exchange coefficient for this artificial mixing is $\Delta x^2/(4\Delta t)$ (Shapiro, 1971) which is in the order of a few 10000 m^2/s for the simulations conducted here. This value exceeds the range of turbulent exchange coefficients to be expected for the atmosphere by at least a factor of 100. For comparison, the maximum exchange coefficient occurring in the simulation with the mixing length parameterisation is only about 12.5 m^2/s. That explains the fuzzy appearance and relative featurelessness of the results obtained with the 3-point filter.

Atmospheric models usually employ filters of higher order which use five-point or even seven-point stencils instead of the three-point stencil in the example above.

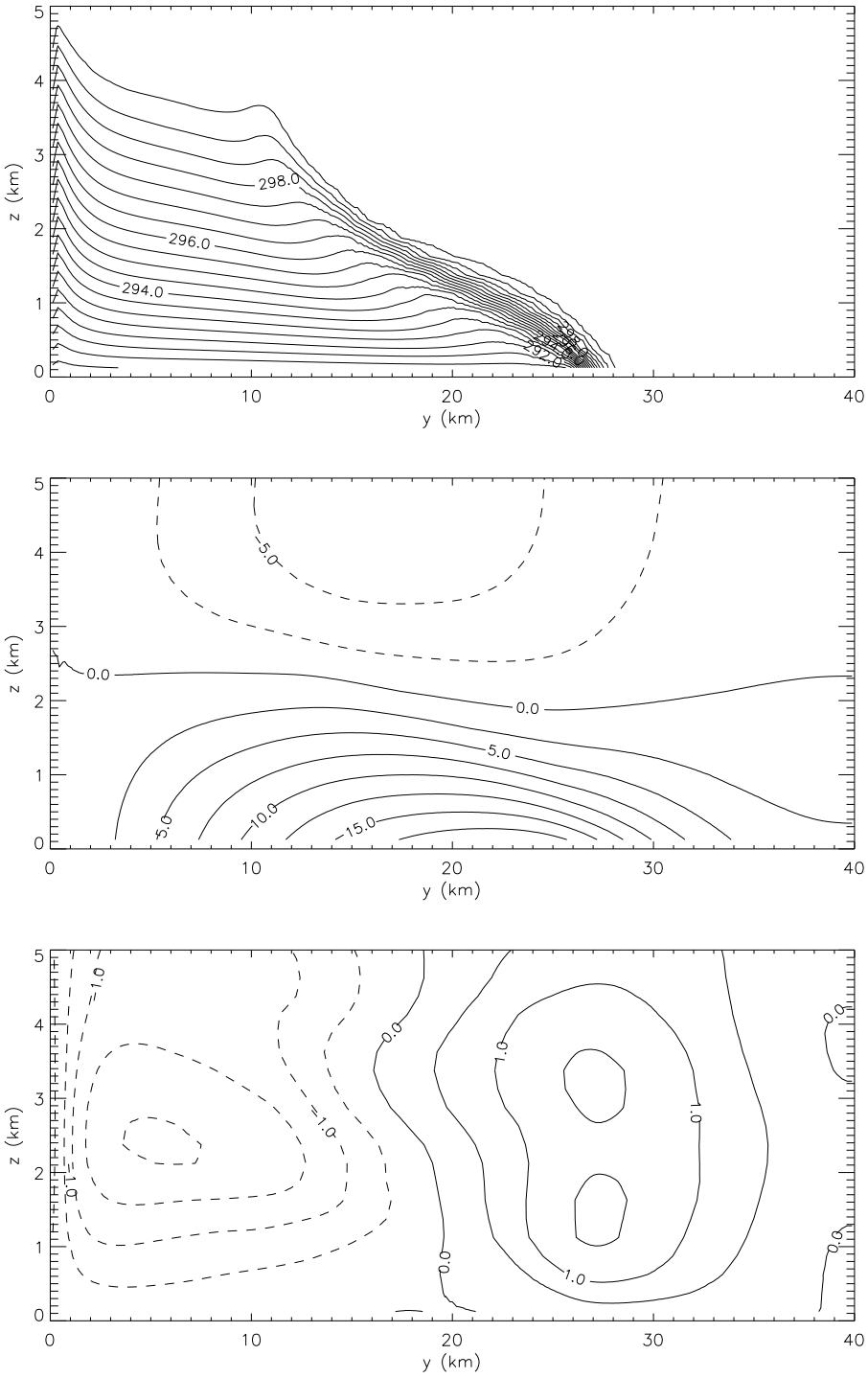

Figure 5.14: Same as Figure 5.7 with a first order filter, $n = 1$ in (5.4).

Higher-order filters are much more specific in their damping behaviour. They exactly cancel waves with a wavelength equal to twice the grid spacing leaving longer waves mostly unchanged. The higher the value of n is chosen in (5.4), i.e. the larger the filters stencil, the more specific is the filter to short waves. Figure 5.15 shows the results of a simulation with a 5-point filter, i.e. $n = 2$ in (5.4), after 900 s integration time. This corresponds to an additional fourth derivative term in the equations. Comparing with the un-filtered simulation (Figure 5.7), the most prominent difference is the filtering artefact at the left boundary already seen for $n = 1$. This affects the down-draft in the region of the initial cold pool. Otherwise, there are only small differences between the two simulations. The maximum up-draft as well as the maximum velocity in the density current differ only by a few centimetres per second.

The similarities of the simulations can also be observed in the hit rates comparing the filtered simulations to the un-filtered reference. Figure 5.16 shows the time series of the hit rates of the horizontal and vertical velocity components for simulations with first and second order filters. Using the same thresholds as in Section 2.2, the hit rate of the horizontal wind is still over 98% after 900 s, were the damping layer at the top boundary of the domain (twelve cells) as well as 20 cells (five kilometres) at the left and twelve cells (three kilometres) at the right lateral boundaries have been excluded. The hit rate of the vertical velocity is more sensitive: it drops as low as 57%. In a region around the front, $25\,\mathrm{km} \leq x_2 \leq 35\,\mathrm{km}$, the hit rate of horizontal velocity component is 100% and that for the vertical velocity component is 84% after 900 s integration time. So many of the 'misses' have to be accounted to the down-draft in the region of the initial cold pool which may still be affected by the filtering artefact at the left boundary. Also, the time series of the vertical velocity hit rate shows oscillations. This is an indication for different patterns of gravity waves in the stably stratified cold air: the diffusion of the filter provides additional damping of waves. For comparison, the hit rates for the simulation using the first order filter are 14% for the horizontal and 26% for the vertical velocity component after 900 s.

Filtering also introduces another effect into the simulation which differs from the behaviour of the 'normal' case: in the compressible equations (4.1), pressure information is propagated with the character of acoustic waves at the speed of sound. Filtering the velocity components partially flattens out the divergence propagating the pressure information through the domain. But the artificial diffusion introduced by the filter (5.4) also spreads the waves transporting the pressure information in the wind field and thereby increases their effective propagation speed. So the natural mechanism for equalising pressure is altered by the filter. This effect is observable in the different behaviour of the total mass in the domain with time. Figure 5.17 shows the time series of total mass for simulations with first and second order filters, $n = 1$

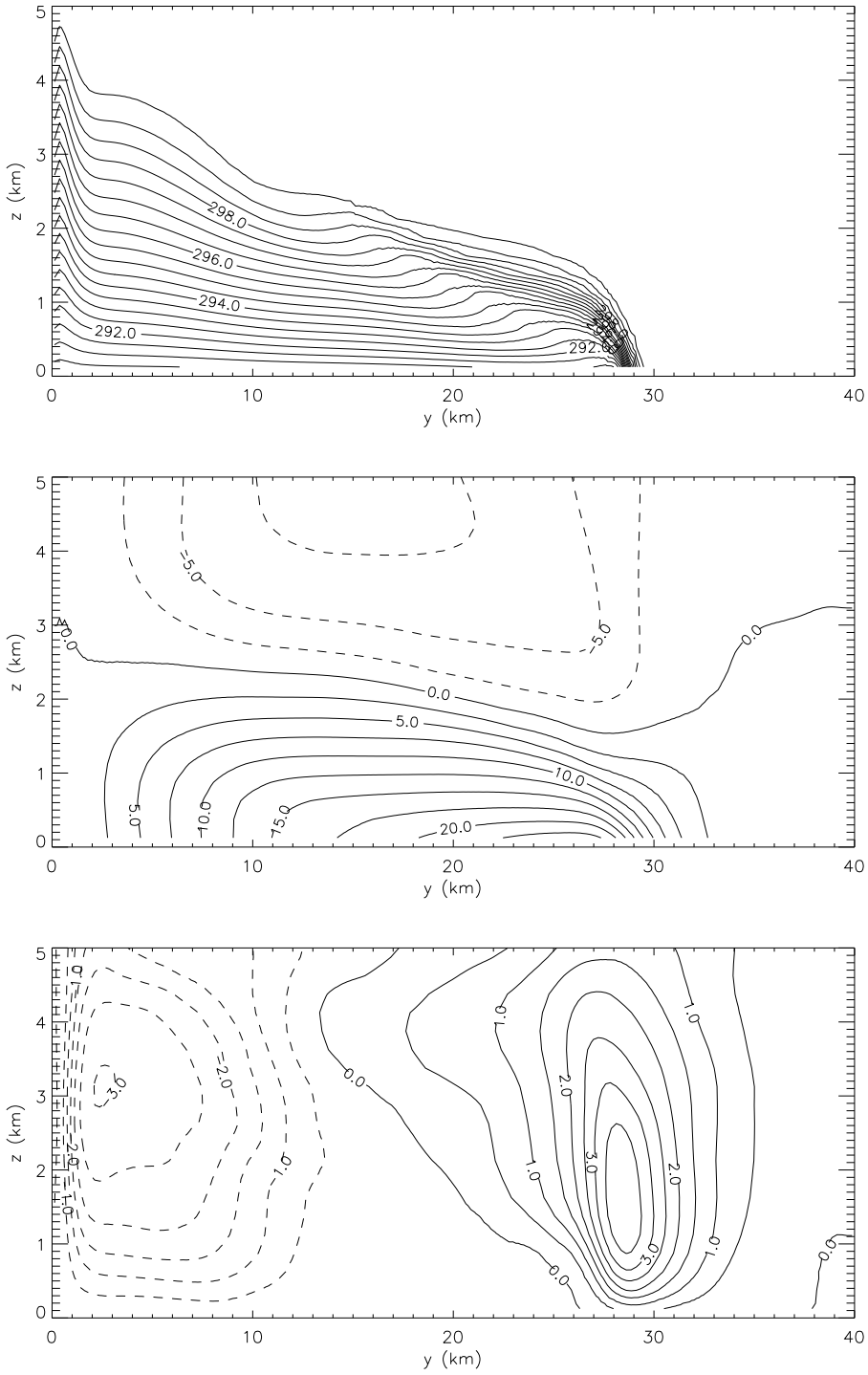

Figure 5.15: Same as Figure 5.7 with a second-order filter, $n = 2$ in (5.4).

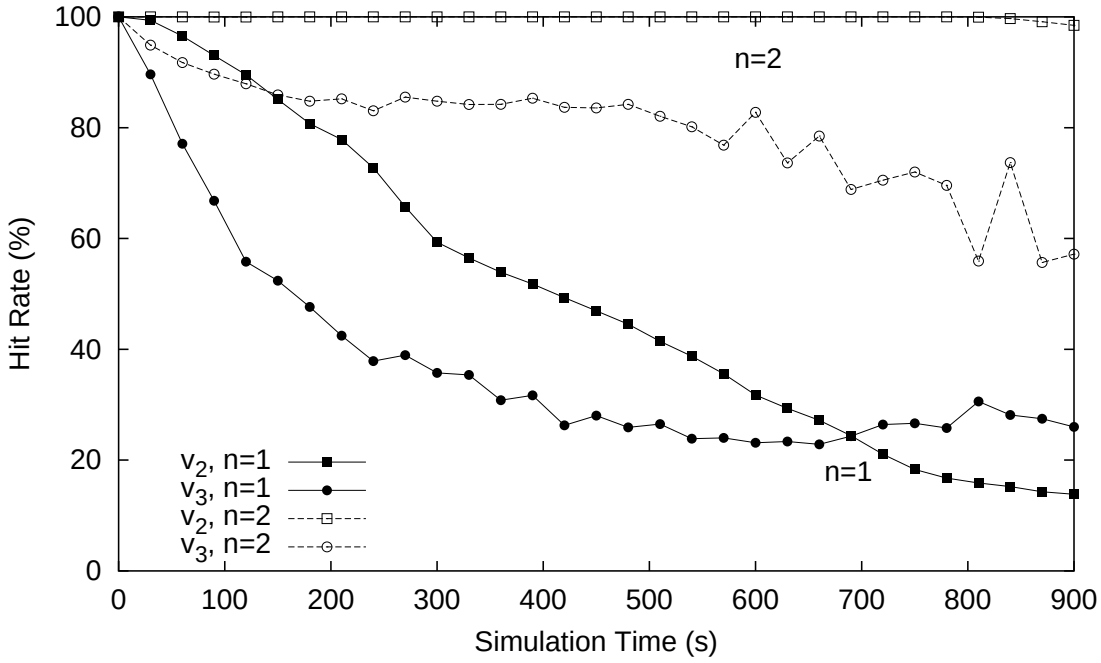

Figure 5.16: Hit rates for the simulations with filtering.

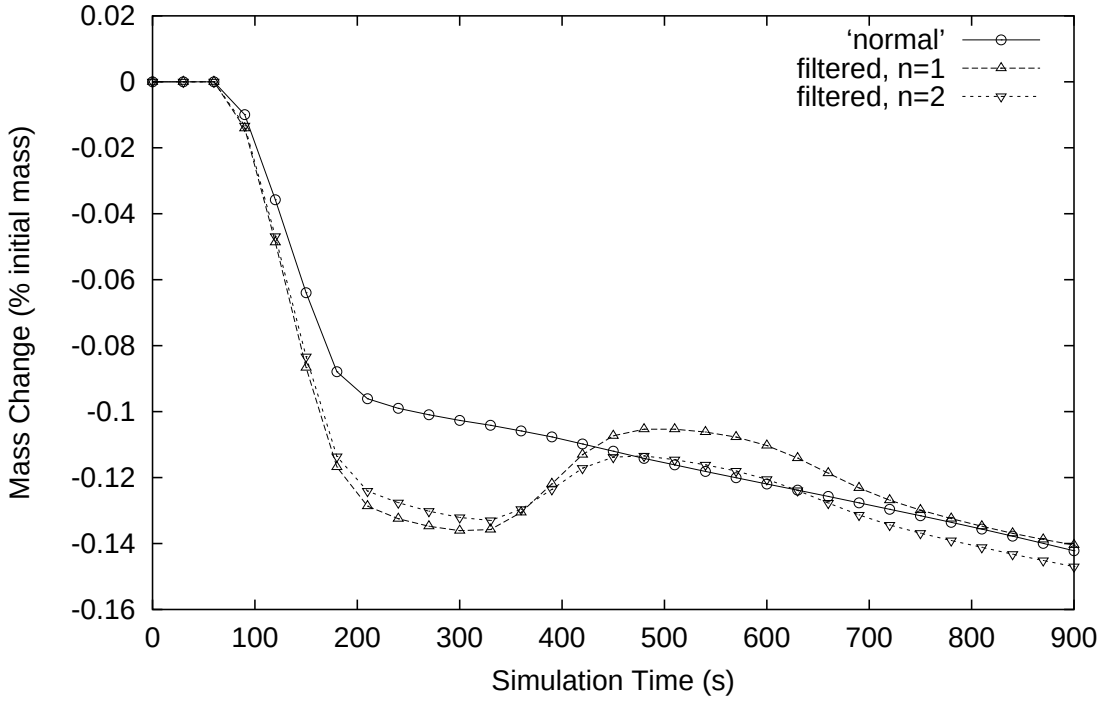

Figure 5.17: Evolution of the total mass in the filtered and un-filtered case.

and $n = 2$ in (5.4), and the un-filtered reference simulation. The overall expansion effect discussed for the un-filtered simulation in Section 5.2.1 does also occur with filtering. It can be seen that the drop in total mass due to the equalisation of the initial discontinuity is even *more* sharp with filtering because the waves propagating pressure information are subject to the additional spreading by the filtering process. Thus, more mass leaves the domain in the same period of time, and the expansion is more intense than in the un-filtered simulation. In fact, it is too intense: the time series of the total mass in the simulations with filtering 'overshoot' compared to the reference simulation and later seem to converge towards that in a decaying oscillation. Despite the fact that, compared to the un-filtered case, significantly higher diffusion is not apparent in the results of the simulation with second order filter, it behaves almost like the simulation with first order filter in this respect which is obviously very diffusive. In both filtered simulations the phase speed of waves bearing pressure information is altered the same way. For anelastic models this effect does not occur since the anelastic approximation inhibits such expansions as seen in Figures 5.8 and 5.17 in the first place. Consequently, the filters do not alter the propagation of pressure information in such models.

In any case, one definite conclusion to be drawn from this test is that the numerical method of a3m as described in Chapter 4 has no need for additional stabilisation by this technique. Also note that with the grid refinement approach introduced in Section 4.2 the filter would not be applicable at the interfaces of different grid blocks: the necessary neighbours for applying the stencil are lacking near the block boundaries—the filter stencil does not fit. So no filtering can occur across block boundaries which would generate numerical artefacts in those places.

To explain why a3m does not need any spatial filtering for stable operation, while many other atmospheric models do, a look at the similarities of the numerical methods employed in these models and the differences to the methods of a3m introduced in Chapter 4 can provide some insight. One of the obvious differences of many other atmospheric models to a3m is that the advective momentum transport term, $\partial_{x_j} \rho u_i u_j$, is discretised separately and independently of the pressure gradient force $\partial_{x_j} \mathsf{p}$. Examples for this approach are the anelastic models METRAS and GESIMA or the compressible models ARPS and LM. The pressure gradient force is discretised by simple three-point differences in all of the above models. The transport term is often discretised using simple three-point (second order) or five-point (fourth order) differences (e.g. Tremback *et al.*, 1987). These schemes are relatively easy to implement, cheap to compute and they have a very low numerical diffusion. But they also have bad oscillatory properties: they are well known to generate many non-physical, short-wave oscillations that are absent in the real solution (e.g. Mesinger and Arakawa, 1976; Roache, 1982). As sketched in Figure 5.18, waves with a wavelength of twice the grid spacing represent a special problem for these schemes: such

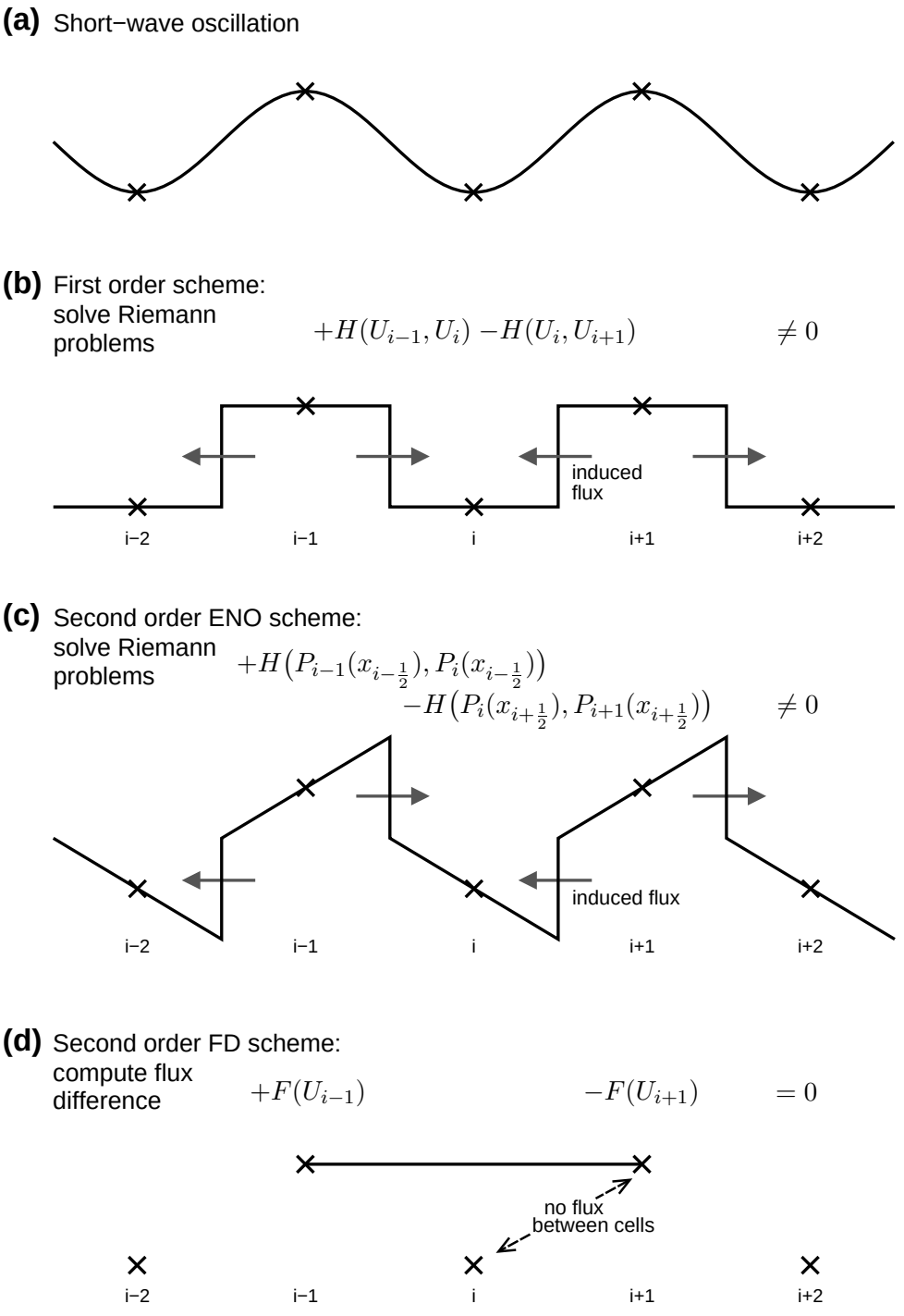

Figure 5.18: Discrete views of short wave oscillations.
Short-wave oscillation (a) with a wavelength of twice the grid spacing and the reaction of (b) the piecewise constant approximation of a first order FV scheme, (c) the piecewise linear ENO interpolating second order scheme of a3m and (d) simple centred three-point differences (second order) on this wave. In case (d) no fluxes are induced: the numerical method cannot detect the wave. TVD schemes also behave as shown in (b).

waves are not detected and stay altogether un-altered. Thus, these short waves are not propagated, and their amplitude may grow unboundedly with time. A different view of this numerical phenomenon is that two distinct and decoupled solutions are computed on alternating grid points. The classic approach to improve the oscillatory behaviour of such schemes is to add an artificial viscosity term to the scheme (von Neumann and Richtmyer, 1950; Lax and Wendroff, 1960). The diffusional term couples the two solutions again and damps the short waves. The amount of artificial diffusion to be used is not clear, though, and often has to be calibrated experimentally, as for the density current simulations of Skamarock and Klemp (1993). The short-wave oscillations are not completely removed until the amount of diffusion is increased far enough so that the artificial diffusion effectively destroys the solution. This was seen above for the first order filter, i.e. $n = 1$ in (5.4). Higher-order linear filters (5.4), i.e. $n \geq 2$, are more specific to short waves. They improve the simulation results of atmospheric models significantly compared to first order filters, but they still are essentially an artificial diffusion term.

As discussed in Section 4.1.1.2, the key issue of avoiding artificial oscillations in a3m is, how the monotonic and therefore oscillation-free first order scheme (Osher and Solomon, 1982) is extended to a higher order. In a3m this is achieved by ENO interpolation for the pressure, temperature and wind fields. The whole flux function $F(U)$ of the hyperbolic equation (4.1) is applied with the interpolated values so that all of the terms of $F(U)$ have the same essentially *non*-oscillatory property. As illustrated by Figure 5.18, first order schemes, TVD schemes, as well as a3m's second-order FV/ENO scheme detect even short waves directly and therefore propagate them just as they do with longer waves. No decoupling between alternating cells occurs, but the numerical diffusion *inherent* to the schemes is higher compared to the simple centred differences. The amount of numerical diffusion is adjusted to the current flow situation and depends *non*-linearly on the solution as opposed to linear filters (5.4) or other artificial diffusion terms.

5.3.2 Divergence Damping

The divergence damping technique involves an artificial term added to the momentum equation of (4.1). It is replaced by

$$\partial_t(\rho u_i) + \sum_{j=1}^{3} \partial_{x_j} \left(\rho u_i u_j + \delta_{ij} \left(\mathsf{p}^{(1)} - \alpha \sum_{k=1}^{3} \partial_{x_k}(\rho \underline{u}) \right) \right) = -\sum_{j=1}^{3} \sum_{k=1}^{3} 2\epsilon_{ijk}\Omega_j \rho u_k - \rho^{(1)} g_i - \sum_{j=1}^{3} \partial_{x_j} \tau_{ij}, \quad i = 1, 2, 3.$$

The term $-\alpha \sum_{k=1}^{3} \partial_{x_k}(\rho \underline{u})$ is added to pressure within the pressure gradient force with

$$\alpha = \alpha_{\mathrm{DD}} \, |\underline{r}|^2 / \Delta t,$$

where $|\underline{r}|$ is the local mesh size, Δt is the current time step and $\alpha_{\mathrm{DD}} > 0$ is the divergence damping coefficient. The value 0.05 was used for α_{DD} in the simulations, which is in the range of values used by other models employing divergence damping (e.g. ARPS, Xue *et al.*, 1995). The effect of this term becomes clear, when the divergence operator is applied to the momentum equations above. This gives

$$\partial_t(\nabla(\rho \underline{u})) = \alpha \nabla^2(\rho \underline{u}) + \ldots.$$

The divergence damping term from the momentum equation is transformed into a diffusion term acting on the momentum divergence. While the meteorological relevant atmospheric flows are all mostly divergence free, sound waves do involve divergence. By smoothing the divergence of the momentum field due to the diffusion term, the undesired sound waves are also damped. Other waves, such as gravity waves, are mostly unaffected by the artifically introduced term in the momentum equations (Skamarock and Klemp, 1992). This technique is used in different atmospheric models, e.g. ARPS, to damp acoustic waves artificially generated by the so-called mode-splitting time integration (Skamarock and Klemp, 1992). But it also helps to remove numerical noise triggered by small imbalances of the thermodynamic variables T, p and ρ, which are introduced by rounding errors in the floating point calculations on the computer.

To asses the impact of divergence damping on a3m's simulation results, the density current of Section 5.2 is simulated with divergence damping and compared to the 'normal' simulation. The results of the two simulations are very similar: the hit rate of horizontal velocity is always 100%, that of vertical velocity is almost 100% up to 500 s simulation time and still well over 80% after that. The vertical velocity hit rate shows rather strong oscillations, so there are differences in the wave patterns, which is intended in this case. The developing circulation has the same intensity in both simulations, with and without divergence damping. Extrema of velocities coincide in their location and differ only by a few centimetres per second. Figure 5.19 shows the regions where the difference of the vertical velocity exceeds 2.5 cm/s after 900 s integration time shaded gray. They clearly display the wave patterns differing in the two simulations.

Even the velocity divergence, shown in Figure 5.20, does not reveal a significant effect compared to the simulation without divergence damping. The impact of divergence damping on the velocity divergence is clearly smaller than that of the spatial filters discussed in Section 5.3.1.

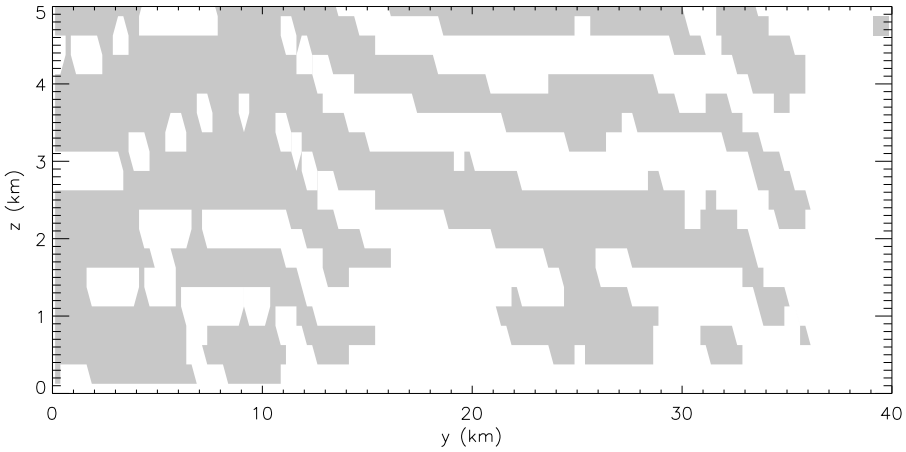

Figure 5.19: Difference in the vertical velocity in the simulations with divergence damping compared to the reference simulation. The gray areas indicate the regions where the difference exceeds 2.5 cm/s.

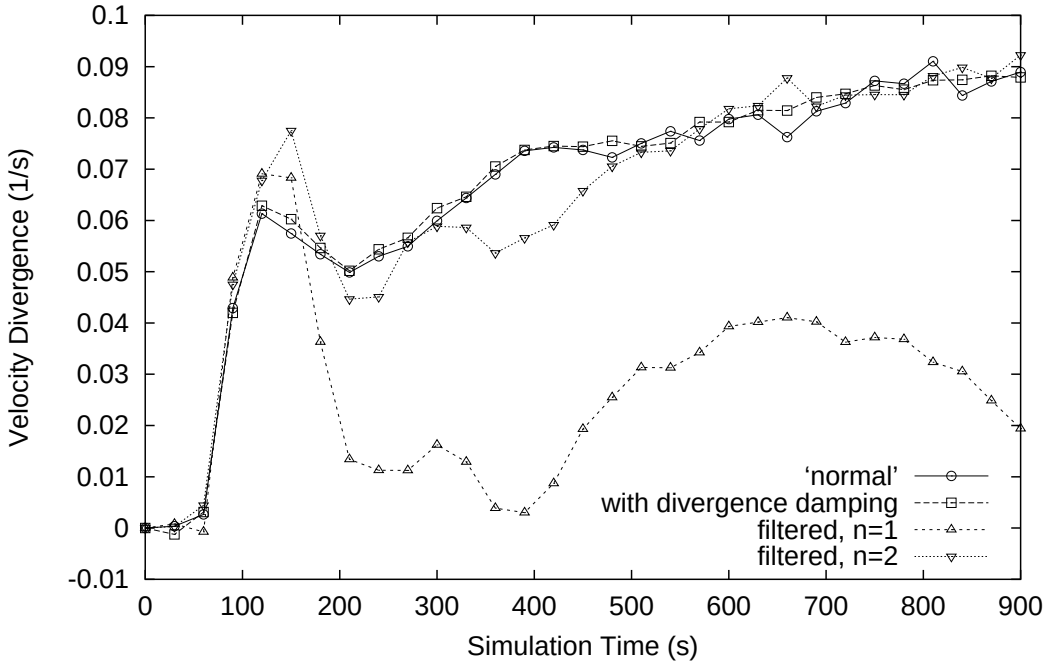

Figure 5.20: Total velocity divergence in the model domain.
Shown are the time series of the velocity divergence accumulated over all cells for the 'normal' reference simulation, the simulation with divergence damping and the simulations with spatial low-pass filters.

The lack of influence of the divergence damping on the simulation result is either due to the nature of the density current—but this seems unlikely since it does include a significant amount of acoustic waves—or the simple time integration used in this work as opposed to time splitting schemes that generate artificial acoustic waves themselves (Skamarock and Klemp, 1992) or due to a much too low damping coefficient. But the value of the damping coefficient is the same as used for other models (e.g. Xue *et al.*, 1995) and a simulation with the damping coefficient increased by a factor of ten also yields no significant change in the results. Summarising, there seems to be no need for divergence damping unless a different time integration scheme is employed. All simulations without divergence damping remained just as stable and not noticeably more noisy.

Chapter 6

Conclusions

In this work the development of a new atmospheric model is motivated and described. The newly developed model tries to solve some of the problems encountered in simulations of multi-scale atmospheric flows. It has been named a3m, where a3m really is a³m with a triple 'a'. The recursive acronym a3m stands for "**a**3m is an **a**daptive **a**tmospheric **m**odel."

In Chapter 2 the dependence of simulation results on the resolution of the computational grid was studied for the example of the land-sea breeze circulation using the mesoscale model METRAS. In agreement with the expectations, the resolution was found to be most important in regions where small-scale flow features develop. For the case of the sea breeze this is most pronounced at the sea breeze front. Such small-scale features are difficult to simulate because a broad range of resolutions is required to capture both, the large-scale flow and its embedded features. To simulate such flows with a uniform high resolution is not feasible due to the high strain on computational resources. As discussed in Chapter 1, employing non-uniform grids or the nesting approach can solve this problems only partially. An adaptive strategy, where the grid itself evolves with the computed solution, seems more reasonable.

The model a3m is based on the non-hydrostatic, fully compressible flow equations. It predicts mass density ρ, momentum density components ρu_i and the total energy density $\rho \mathsf{e}$. Only very few basic assumptions, listed in Section 3.1.1, are made. None of these is specific to a certain atmospheric scale. With the exception of neglecting moisture (assumption 7), they are non-critical for simulations of atmospheric flows and impose no serious restrictions for the applicability of a3m. Otherwise, no approximations are made.

Discretisation of the model equations is achieved by a set of up-to-date numerical methods, especially with respect to their oscillatory properties: a3m uses a Godunov-type finite volume (FV) method employing the approximate Riemann solver of Osher and Solomon (1982) as the numerical flux function. The scheme is extended to second order following the MUSCL approach (van Leer, 1979, republished 1997) by using essentially non-oscillatory (ENO) variable interpolation (Harten *et al.*, 1987; Harten, 1996; Sonar, 1997; Abgrall *et al.*, 1999). Due to their adaptive choice of stencils, ENO methods are especially well equipped for simulations of discontinuities such as fronts.

One of the key features of a3m is the ability to use locally refined computational

grids so that resolution can be enhanced in a region of the computational domain while keeping the overall consumption of resources in reasonable bounds. In a3m local grid refinement is achieved by employing block-structured grids: a grid consists of multiple blocks of different resolution and refining blocks may overlay parts of refined blocks. This approach is similar to the algorithm of Berger and Oliger (1984), but the coupling of different blocks occurs directly by the FV solver itself. This is facilitated by the very general and geometric formulation of the FV scheme.

The new model is considered a research prototype for future atmospheric models to experiment with different numerical schemes and methods for their implementation. The main development goals for a3m are modularity, flexibility and extensibility. The FV approach itself is already very modular. It enables the construction of numerical schemes from separate functional blocks: numerical flux functions, reconstructions, quadratures and recovery procedures. For the implementation of a3m an object-oriented (OO) design was developed. The OO paradigm is suited best to retain the high degree of modularity inherent to FV schemes and ensures an equally high degree of modularity in its implementation and thereby the demanded flexibility and extensibility of a3m.

The numerical methods of a3m were tested towards their stability especially with respect to abrupt changes in the resolution of the computational grid as they occur in locally refining block-structured grids. Overall, the methods performed as expected and turned out to be robust towards abrupt changes in resolution in these tests. The changes in resolution were of no consequence for solutions varying linearly in space. For solutions varying more than linearly in space, artefacts may be generated near the location of changing resolution. As a consequence, the boundaries of locally refined grid blocks should be in areas where the solution is varying only linearly in space. Within an area of linear varying solution the resolution can be strongly decreased without any impact on the simulation results.

The suitability to simulate atmospheric flows with a3m was demonstrated for the example of a density current emerging from a collapsing cold pool. The density current exhibits a front which was resolved very well by a3m in all simulations. The flow head was not well developed in most simulations. This was improved by locally increasing the resolution of the computational grid. In the region of Kelvin-Helmholtz instability behind the flow head, isotherms were deformed and beginning to 'roll up' under the stress and shearing in the flow. However, vortices and actual Kelvin-Helmholtz billows did not develop. While the FV/ENO scheme of a3m gives simulation results with a well resolved front almost free of artificial oscillations, it seems to be overly diffusive in some regions compared to other numerical methods so that e.g. development of the Kelvin-Helmholtz billows is suppressed. Turbulence was found to be of minor importance for this density current scenario.

In addition to the uniformly resolved and locally refined simulations, two methods to increase numerical stability, computational mixing and divergence damping, were examined for their inter-operability with a3m's numerical methods and their effect on the simulation results. Both methods are frequently used in atmospheric models. Computational mixing is achieved by a spatial low-pass filter which removes short waves from the approximate solution. On rectangular, uniformly spaced grids it is equivalent to an additional (higher-order) diffusional term. Using the first order filter, the diffusion introduced by the filter has a very strong impact in the simulation results, and the development of the density current is strongly hindered. A higher-order filter is much more specific to short wave lengths and has a far smaller impact on the model results. Unlike many central difference methods, the FV/ENO scheme of a3m introduces only very few artificial short waves into the solution so that a filter very specific to short waves leaves the solution mostly unaltered. Divergence damping artificially damps acoustic waves to stabilise against non-physical sound waves generated by the numerical scheme. It showed no significant impact on the results, which suggests that either the numerical scheme of a3m generates only few non-physical sound waves or the inherent numerical diffusion of the scheme sufficiently damps such waves itself.

Future developments of a3m may be divided into two categories. Firstly, a3m needs an extension to the physical problem domain and its mathematical model. Secondly, the numerical schemes described in this work offer many possibilities of extension and further improvement.

Concerning the simulated physical system, the extension to the moist atmosphere is clearly most important. It is the most severe restriction in the applicability of a3m to simulate real atmospheric cases. This includes a parameterisation of sub-grid-scale cloud physics to determine condensation and evaporation rates. Also, more sophisticated turbulence models can significantly improve simulation results for scenarios closer to reality than the density current simulated in this work. Additionally, surface layers in the vicinity of solid surfaces cannot be simulated directly in atmospheric models since that requires a prohibitively high resolution. Hence, these layers are usually parameterised using the similarity theory of Monin and Obukhov (1954).

Numerically solving equations with parameterised sub-grid-scale physics also raises a mathematical issue: for vanishing mesh size h of the computational grid, $h \to 0$, the discrete solution must converge to the analytical solution, $U_h \to U$, i.e. the numerically gained solution U_h is an approximation of the real solution U. But parameterisations themselves are an artefact of the finite resolution of the discretisation. Thus, for $h \to 0$ the parameterisations should vanish for the discretisation to be consistent. For most parameterisations this cannot be guaranteed since this

issue was not considered in their design. Skamarock and Klemp (1993) report that with parameterised turbulent mixing, the truncation error did not exhibit the rate of decrease to be expected with increasing resolution. For computational practice, it is sufficient for the parameterisations to be consistent in the range of spatial and temporal resolutions employed. But in the context of locally refined grids, and even more so for adaptive grids, this range may be significantly larger than for conventional models.

The second major issue concerning realistic simulations is the number of spatial dimensions: the real atmosphere is three-dimensional. So to obtain results comparable to reality, 3d simulations are required. As discussed in Section 4.1, the computation of fluxes in the FV scheme is independent of the number of spatial dimensions—modulo rotations and interpolations—as is the computation of source terms. This fact has been taken advantage of in the design and implementation of a3m separating the parts of the code depending on the number of spatial dimensions from other parts, especially iterations within the grid classes (Appendix C).

Considering numerics, the most important issue in the development of a3m is the dynamical adaption of the computational grid. For now, the model can simulate flows on locally refined but static grids. This is demonstrated in Section 5.2.2. For an adaptive grid, two more functional blocks are needed: a refinement indicator to find regions that need refining and regions that may be coarsened as well as a clustering procedure to group cells marked for refinement into new grid blocks. Appendix B describes an already implemented and successfully tested refinement indicator.

For a more efficient time integration the limitation on the time step by the speed of sound has to be lifted. As discussed in Section 4.3, this may be achieved by an implicit method or a time-splitting scheme. Application of the time-splitting technique requires a decomposition of the advective fluxes into processes acting on different time scales. With the tight coupling of transport and pressure terms in the numerical scheme of a3m this is not straight forward. Possibly, construction of a time-splitting scheme for a3m even has to start off at an earlier development stage (Figure 1.3), the mathematical model. A means to achieve the decomposition may be an asymptotic analysis similar to that of Botta *et al.* (1999), but this still requires much research.

Finally, the restriction of at most linearly varying solution across grid blocks of different resolution can be lifted by increasing the order of approximation of the FV/ENO scheme. This can be achieved by increasing the rank of the recovery polynomial: using parabolas instead of the linear recoveries (4.10) yields a third order scheme. But that is burdensome because of the strongly increasing number of interpolation stencils to consider for a higher polynomial rank, especially in 3d.

However, using a convex combination of the different candidates for the recovery instead of choosing one of them, the order of approximation can be increased by one. This yields a so-called weighted essentially non-oscillatory (WENO) scheme (Liu *et al.*, 1994). This also yields a third order scheme in regions where the solution is smooth enough, and in the vicinity of discontinuities it behaves like the second order scheme presented in Section 4.1. This also decreases the numerical diffusion of the scheme in smooth regions while fully retaining its desirable oscillatory properties. WENO schemes require an additional functional block: a procedure to gain weights for each of the different recovery stencils.

Acknowledgements

This work was supported by and is part of the Graduiertenkolleg *Erhaltungsprinzipien in der Modellierung und Simulation mariner, atmosphärischen und technischer Systeme* (Conservation principles in modelling and simulation of marine, atmospherical and technical systems).

First of all, I would like to thank Priv.-Doz. Dr. K.H. Schlünzen for her great support of my sometimes obscure ideas finally resulting in this project, for providing the opportunity to pursue my own goals and for constructive criticism and motivation at the right times.

Thanks also to Prof. Dr. J. Struckmeier, Dr. M. Perić and HD. Dr. A. Meister, as well as Prof. Dr. Th. Sonar, the initiator of the Graduiertenkolleg, for their inspiration and many fruitful discussions.

A special "thank you" goes to Dr. E. Fedorovich, Dr. K. Droegemeier, Dr. H. Neeman and D. Jahn for kindly hosting me at the School of Meteorology and CAPS at the University of Oklahoma, their helpful comments on my work, lunches, shopping raids and/or barbecues, to Dr. M. Breuß, Dipl.-Math. S. Günzel, Dipl.-Phys. S. Heitmann, Dipl.-Math. J. Köhnlein and Dipl.-Inf. A. Wienberg for illuminating discussions and/or proofreading and K. Richards, M.E. and J. Hamilton, M.A. M.C.D. for sharing their wisdom about the English language.

Thanks also to the active and former members of my working group and the Meteorological Institute of Hamburg University for all their feedback and support, especially Carola Kauhs, Simon, Silke, David, Amélie, Ute, Dirk and Heiko as well as the other students of the Graduiertenkolleg for their company—scientific and otherwise—on the long road to a doctors degree.

Also, I would like to thank my mother and the rest of my family as well as my friends Jacqueline, Stefan, Martina, Corinna, Flow, Jörg, Birgit, Maik and Elisabeth for still being my friends despite my ill moods and the failings on my side to keep in touch, while I was completing this thesis.

Finally, credit is due to all the countless authors of free software out there. Their efforts have greatly eased if not enabled the development `a3m`. And kudos to the maintainers and developers of GNU Compiler Collection, `gcc`, and the C++ standard library, `libstd++-v3`.

Appendix A

Numerical Flux Function of Osher and Solomon

The numerical flux function used in Chapter 4 was developed by Osher and Solomon (1982) and follows the flux vector splitting approach. It is an extension of the classic upwind discretisation for non-linear systems of hyperbolic conservation laws. In this appendix the derivation of their approximate Riemann solver is outlined. As mentioned in Section 4.1.1.1 some analysis is required to derive the numerical flux function (4.8). An in-depth review of numerical methods for hyperbolic conservation equations in general and the Euler equations as a special case of those is beyond the scope of this work but provided e.g. by Hirsch (1988b), Spekreijse (1988), Sonar (1997), Toro (1997) and also the original article of Osher and Solomon (1982).

One important property of the Euler equations (4.1) is already introduced and utilised in Chapter 4: the rotational invariance as stated by equation (4.7). Thereby, the flux computation is reduced to solving a one-dimensional Riemann problem. In the following, the x_1 coordinate is perpendicular and the x_2 and x_3 coordinates are parallel to the discontinuity of the Riemann problem. So only the flux function $F_1(U)$ needs to be considered. The Riemann problem may thus be stated as: determine $U(x_1, t)$ for $t > 0$ that solves (4.1) for the initial condition

$$U(x_1, 0) = \begin{cases} U_L & \text{for } x_1 < 0 \\ U_R & \text{for } x_1 \geq 0. \end{cases}$$

The indices L and R designate the left ($x_1 < 0$) and the right ($x_1 \geq 0$) side of the initial discontinuity.

The solution to this problem consists of the so-called simple wave solutions. These waves propagate with phase speeds equal to the eigenvalues of the Jacobian matrix $A(U) := \nabla_U F_1(U)$ of the flux F_1. The Jacobian is given by

$$A(U) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ -u_1^2 + (\gamma-1)\frac{|\underline{u}|^2}{2} & (3-\gamma)u_1 & (1-\gamma)u_2 & (1-\gamma)u_3 & (\gamma-1) \\ -u_2 u_1 & u_2 & u_1 & 0 & 0 \\ -u_3 u_1 & u_3 & 0 & u_1 & 0 \\ u_1\left((\gamma-1)|\underline{u}|^2 - \gamma \mathsf{e}\right) & \gamma\mathsf{e} - \frac{\gamma-1}{2}(|\underline{u}|^2 + 2u_1^2) & (1-\gamma)u_1 u_2 & (1-\gamma)u_1 u_3 & \gamma u_1 \end{pmatrix}$$

(e.g. Hirsch, 1988b). Since the Euler equations (4.1) are hyperbolic, the matrix $A(U)$

has five real eigenvalues,

$$\begin{aligned} \lambda_1(U) &= u_1 - \mathsf{c}, \\ \lambda_2(U) = \lambda_3(U) = \lambda_4(U) &= u_1, \qquad \text{and} \\ \lambda_5(U) &= u_1 + \mathsf{c}, \end{aligned} \tag{A.1}$$

where c is the speed of sound with $\mathsf{c}^2 = \gamma\mathsf{p}/\rho$ and five corresponding linearly independent right eigenvectors,

$$r_1(U) = \begin{pmatrix} 1 \\ u_1 - \mathsf{c} \\ u_2 \\ u_3 \\ \mathsf{e} + \mathsf{p}/\rho - \mathsf{c}u_1 \end{pmatrix}, \; r_2(U) = \begin{pmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ |\underline{u}|^2/2 \end{pmatrix},$$

$$r_3(U) = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ u_2 \end{pmatrix}, \; r_4(U) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ u_3 \end{pmatrix}, \text{ and } r_5(U) = \begin{pmatrix} 1 \\ u_1 + \mathsf{c} \\ u_2 \\ u_3 \\ \mathsf{e} + \mathsf{p}/\rho + \mathsf{c}u_1 \end{pmatrix}$$

(e.g. Hirsch, 1988b; Sonar, 1997).

As mentioned above, the eigenvalues λ_k given by equation (A.1) are the phase speeds of the simple wave solutions of the Euler equations: shock waves, contact discontinuities and expansion waves also known as rarefaction waves. This means that their direction of propagation is given by the sign of the λ_k: negative eigenvalues correspond to leftward moving waves and positive eigenvalues to rightward moving ones. This provides the means of splitting the flux $F_1(U)$ into a leftwards and rightwards acting part. To this end, the Jacobian matrix A is diagonalised to

$$\Lambda(U) := P^{-1}(U)\, A(U)\, P(U) = \begin{pmatrix} \lambda_1 & 0 & \cdots & \cdots & 0 \\ 0 & \lambda_2 & \ddots & & \vdots \\ \vdots & \ddots & \lambda_3 & \ddots & \vdots \\ \vdots & & \ddots & \lambda_4 & 0 \\ 0 & \cdots & \cdots & 0 & \lambda_5 \end{pmatrix},$$

where the diagonalisation matrix P is the concatenation of the right eigenvectors (e.g. Toro, 1997). The diagonal matrix $\Lambda(U)$ is now split into two matrices, Λ^+ with only positive or zero eigenvalues and Λ^- with only negative or zero eigenvalues:

$$\Lambda^{\pm}(U) := (\Lambda(U) \pm |\Lambda(U)|)/2.$$

Reversing the transformation via P for the two matrices $\Lambda^{\pm}$ gives a splitting of the Jacobian matrix A into the matrices $A^{\pm}$,

$$A^{\pm}(U) := P(U)\, \Lambda^{\pm}(U)\, P^{-1}(U),$$

corresponding to the rightwards and leftwards propagating parts of the flux $F_1(U)$. Using this flux splitting into $A^{\pm}$ the numerical flux function of Osher and Solomon is defined as

$$\begin{aligned} H^{OS}(U_L, U_R) &:= H^+(U_L) + H^-(U_R) \\ &= F_1(U_L) - H^-(U_L) + H^-(U_R) = F_1(U_L) + \int_{U_L}^{U_R} A^-(U)\, dU \\ &= F_1(U_R) - H^+(U_L) + H^-(U_R) = F_1(U_R) - \int_{U_L}^{U_R} A^+(U)\, dU \\ &= \frac{1}{2}\left(F_1(U_L) + F_1(U_R) - \int_{U_L}^{U_R} |A(U)|\, dU \right) \end{aligned} \tag{A.2}$$

where $|A| := A^+ - A^-$. The integrals in (A.2) are evaluated along a path Γ in the space of states $\mathbb{R}^+ \times \mathbb{R}^3 \times \mathbb{R}^+$ connecting the states U_L and U_R. Note that the integration evaluates the part of the functions $A^{\pm}(U)$ and $|A|(U)$ tangential to the integration path Γ in the space of states and is not performed component-wise.

Assuming the solution of the underlying Riemann problem is composed of simple wave solutions, Osher and Solomon proposed to connect the states U_L and U_R by a continuous path Γ consisting of five sub-paths $\Gamma_1 \ldots \Gamma_5$ tangential to the corresponding eigenvectors $r_1, \ldots, r_5$:

$$\Gamma = \bigcup_{k=1}^{5} \Gamma_k.$$

Thus, the integrations in equation (A.2) can be broken down into five integrations on the sub-paths Γ_k in direction of the eigenvectors. To evaluate the integrals, the intersection points of two successive sub-paths Γ_k and Γ_{k+1}, i.e. the start and end states of the single sub-integrations, are needed. To determine these, another construct is required: let $r_k(U)$ be the kth eigenvector of the Jacobian matrix $A(U)$. A function $R_k \in C^1(\mathbb{R}^+ \times \mathbb{R}^3 \times \mathbb{R}^+; \mathbb{R})$ with

$$\nabla_U R_k(U)\, r_k(U) = 0, \qquad U \in \mathbb{R}^+ \times \mathbb{R}^3 \times \mathbb{R}^+, \tag{A.3}$$

i.e. whose gradient is perpendicular to the eigenvector in the space of states, is called a k-Riemann invariant. The 1-Riemann invariants—corresponding to the eigenvector $r_1(U)$—are

$$R_1^1(U) = u_1 + \frac{2\mathsf{c}}{\gamma - 1}, \quad R_1^2(U) = u_2, \quad R_1^3(U) = u_3, \quad \text{and} \quad R_1^4(U) = \mathsf{s},$$

where $\mathsf{s} = \ln(\mathsf{p}/\rho^\gamma)$ is the specific entropy, the 2-, 3- and 4-Riemann invariants are

$$R^1_{2,3,4}(U) = u_1 \quad \text{and} \quad R^2_{2,3,4}(U) = \mathsf{p},$$

and the 5-Riemann invariants are

$$R^1_5(U) = u_1 - \frac{2\mathsf{c}}{\gamma - 1}, \quad R^2_5(U) = u_2, \quad R^3_5(U) = u_3, \quad \text{and} \quad R^4_5(U) = \mathsf{s}$$

(e.g. Sonar, 1997; Toro, 1997).

The intersections of the sub-paths Γ_k, i.e. the starting and ending points of the integrations on the sub-paths, can be found by using the Riemann invariants: due to (A.3) a k-Riemann invariant is constant on the corresponding sub-path Γ_k. Also note that the integration over sub-paths Γ_2, Γ_3 and Γ_4 can be collapsed into a single integral due to the linear degeneration in the eigenvalues λ_2, λ_3 and λ_4. Let now $U_a := \Gamma_1 \cap \Gamma_2$ and $U_b := \Gamma_4 \cap \Gamma_5$ be the states in the first and last intersection point of the integration sub-paths Γ_k, respectively. The first velocity component u_1 and pressure p are Riemann invariants for the partial integration paths Γ_2, Γ_3 and Γ_4 so that $u_f := u^a_1 = u^b_1$ and $\mathsf{p}_f = \mathsf{p}_a = \mathsf{p}_b$ can be used to designate their values in the intersection points U_a and U_b. With $R^1_1(U) = u_1 + \frac{2\mathsf{c}}{\gamma-1} = const$ on Γ_1 and $R^1_5(U) = u_1 - \frac{2\mathsf{c}}{\gamma-1} = const$ on Γ_5 and the isentropic law these are obtained as

$$u_f = \frac{q u_{L_1}/\mathsf{c}_L + u_{R_1}/\mathsf{c}_R + 2(q-1)/(\gamma-1)}{q/\mathsf{c}_L + 1/\mathsf{c}_R} \qquad \text{and} \tag{4.8b}$$

$$\mathsf{p}_f = \left(\frac{\mathsf{c}_L + \mathsf{c}_R - (u_{R_1} - u_{L_1})(\gamma-1)/2}{\mathsf{c}_L/\mathsf{p}_L^\alpha + \mathsf{c}_R/\mathsf{p}_R^\alpha} \right)^{1/\alpha} \tag{4.8d}$$

with $q := (\mathsf{p}_i/\mathsf{p}_j)^\alpha$ and $\alpha := (\gamma-1)/2\gamma = R/2c_\mathsf{p}$ which were already given in Section 4.1.1.1. The Riemann invariants $R^4_1(U) = \mathsf{s} = const$ on Γ_1 and $R^4_5(U) = \mathsf{s} = const$ on Γ_5 yield the densities at the intersection points:

$$\rho_a = \rho_L \left(\frac{\mathsf{p}_f}{\mathsf{p}_L} \right)^{1/\gamma} \quad \text{and} \quad \rho_b = \rho_L \left(\frac{\mathsf{p}_f}{\mathsf{p}_R} \right)^{1/\gamma}. \tag{A.4}$$

The other velocity components are Riemann invariants themselves and thus identical to those of the states U_L and U_R, respectively. The full states U_a and U_b are given in equation (4.8c).

Now e.g. the integral $\int_{U_L}^{U_R} A^-(U)\, dU$ in (A.2) may be evaluated yielding 16 cases for the numerical flux function $H^{OS}(U_L, U_R)$ listed in the Table A.1. The states U^S_1 and U^S_5 are sonic points, i.e. points in the space of states in which the corresponding eigenvalues λ_1 and λ_5 change sign on the paths Γ_1 or Γ_5, respectively. Along the eigenvectors R_2, R_3 and R_4 the corresponding eigenvalues are constant so that the integration fully contributes to the numerical flux or not at all.

$u_{L_1} < \mathsf{c}_L, u_{R_1} > -\mathsf{c}_R$	
$u_f > \mathsf{c}_a$	$F_1(U_1^S)$
$\mathsf{c}_a > u_f > 0$	$F_1(U_a)$ $\rightarrow$ first case in (4.8)
$0 > u_f > -\mathsf{c}_b$	$F_1(U_b)$ $\rightarrow$ second case in (4.8)
$-\mathsf{c}_b > u_f$	$F_1(U_5^S)$

$u_{L_1} > \mathsf{c}_L, u_{R_1} > -\mathsf{c}_R$	
$u_f > \mathsf{c}_a$	$F_1(U_L)$
$\mathsf{c}_a > u_f > 0$	$F_1(U_L) - F_1(U_1^S) + F_1(U_a)$
$0 > u_f > -\mathsf{c}_b$	$F_1(U_L) - F_1(U_1^S) + F_1(U_b)$
$-\mathsf{c}_b > u_f$	$F_1(U_L) - F_1(U_1^S) + F_1(U_5^S)$

$u_{L_1} < \mathsf{c}_L, u_{R_1} < -\mathsf{c}_R$	
$u_f > \mathsf{c}_a$	$F_1(U_1^S) - F_1(U_5^S) + F_1(U_R)$
$\mathsf{c}_a > u_f > 0$	$F_1(U_a) - F_1(U_5^S) + F_1(U_R)$
$0 > u_f > -\mathsf{c}_b$	$F_1(U_b) - F_1(U_5^S) + F_1(U_R)$
$-\mathsf{c}_b > u_f$	$F_1(U_R)$

$u_{L_1} > \mathsf{c}_L, u_{R_1} < -\mathsf{c}_R$	
$u_f > \mathsf{c}_a$	$F_1(U_L) - F_1(U_5^S) + F_1(U_R)$
$\mathsf{c}_a > u_f > 0$	$F_1(U_L) - F_1(U_1^S) + F_1(U_a) - F_1(U_5^S) + F_1(U_R)$
$0 > u_f > -\mathsf{c}_b$	$F_1(U_L) - F_1(U_1^S) + F_1(U_b) - F_1(U_5^S) + F_1(U_R)$
$-\mathsf{c}_b > u_f$	$F_1(U_L) - F_1(U_1^S) + F_1(U_R)$

Table A.1: Numerical flux function of Osher and Solomon.

Since atmospheric flows are sub-sonic (assumption 2 in Section 3.1.1), 14 of the 16 cases in Table A.1 can never occur, leaving only the two middle cases of the topmost table A.1 as possible branches. Consequently, the sonic points are of no concern here, but their values can be found elsewhere (e.g. Spekreijse, 1988; Toro, 1997). The two cases remaining are those already given by equation (4.8a).

Note that the ordering of the sub-paths $\Gamma_1 \ldots \Gamma_5$ is relevant since for different orderings the intersection points will be different. The original article of Osher and Solomon (1982) used an ordering usually referred to as the *O*-ordering (for 'original' or 'Osher' ordering). The ordering discussed above and used in a3m is the so-called *P*-ordering (for 'physical' ordering). It is preferred more in recent works since it has a physically more plausible behaviour in extreme cases, e.g. $\mathsf{p} \rightarrow 0$ (Spekreijse, 1988; Toro, 1997).

Appendix B

Dynamic Grid Adaption

In a3m's stage of development presented in this work, it can compute atmospheric flows on locally refined but static block-structured grids. To adapt the grid dynamically with time as the solution evolves regions of the computational domain have to be identified that need refinement or that may be coarsened. Basically, the resolution should be adjusted, such that the error on a grid of mesh size h

$$e_h := U_h - U,$$

i.e. the difference between the numerical solution U_h and the analytical solution U of the differential equation, should always be within certain bounds. If $||e_h||$ is too high within some region of the computational domain, the approximate solution becomes inaccurate, and resolution should be increased locally. If it is very low in some part of the domain, the effort spent on computing the numerical solution could be significantly decreased while still getting an acceptable result using a coarser local resolution. Since the error e_h cannot be computed itself without knowing the exact solution, an estimate η_h of the error has to be employed instead. A reliable error estimate controls the error, i.e.

$$||e_h|| \leq C_1 \eta_h \tag{B.1}$$

with a constant C_1. This may still result in a dramatic overestimation of the error and thereby much too fine grids. For practical purposes the efficiency of an error estimator, i.e.

$$C_2 \eta_h \leq ||e_h||$$

with a constant C_2, is a very desirable property. For the class of problems under consideration here, i.e. systems of non-linear hyperbolic equations, no such estimator is known to date.

A multitude of approaches towards constructing an error *indicator* has been investigated including extension of the residual based framework of error estimation for elliptic problems developed from the finite element approximations (e.g. Sonar and Süli, 1998), estimation of higher-order derivatives (e.g. Dolejší, 1998; Muzaferija and Gosman, 1997; Muzaferija, 1994) and estimating the source of errors by Richardson extrapolation as suggested by Berger and Oliger (1984). The latter approach was already successfully applied to the atmosphere (Skamarock and Klemp, 1993; Skamarock, 1989).

The strategy used by a3m is based on the ENO idea already used for computing the recovery polynomial. As mentioned in Section 4.1.1.2, the ENO interpolation procedure chooses the interpolation stencil such that it does not include discontinuities of the interpolated function or its derivatives (Harten, 1996). For linear recoveries (4.10) used by a3m this includes the first derivative only: the ENO method tries not to interpolate across jumps and corners in the function to interpolate. For refinement indication the recovery on a cell is tested if it can reproduce the cell averages of its neighbours: for the moment the linear recovery $P_i(\underline{x})$ is assumed to extend beyond the cell T_i and is evaluated at the centre of gravity $\underline{c}_j$ of a neighbouring cell T_j, $j \in \mathcal{N}(i)$, as is shown in Figure B.1. If the value $P_i(\underline{c}_j)$ obtained hereby differs from the cell average U_j, this cell has been rejected during the recovery step by the ENO mechanism. This means that the approximated function is more likely to exhibit a jump or bend between the cells i and j than between cell i and other neighbours—the ones that have been chosen for the recovery.

Formally, the refinement indicator on a cell i is

$$\eta_i(\underline{x}) := \max_{j \in \mathcal{N}(i)} ||P_i(\underline{c}_j) - W_j||. \tag{B.2}$$

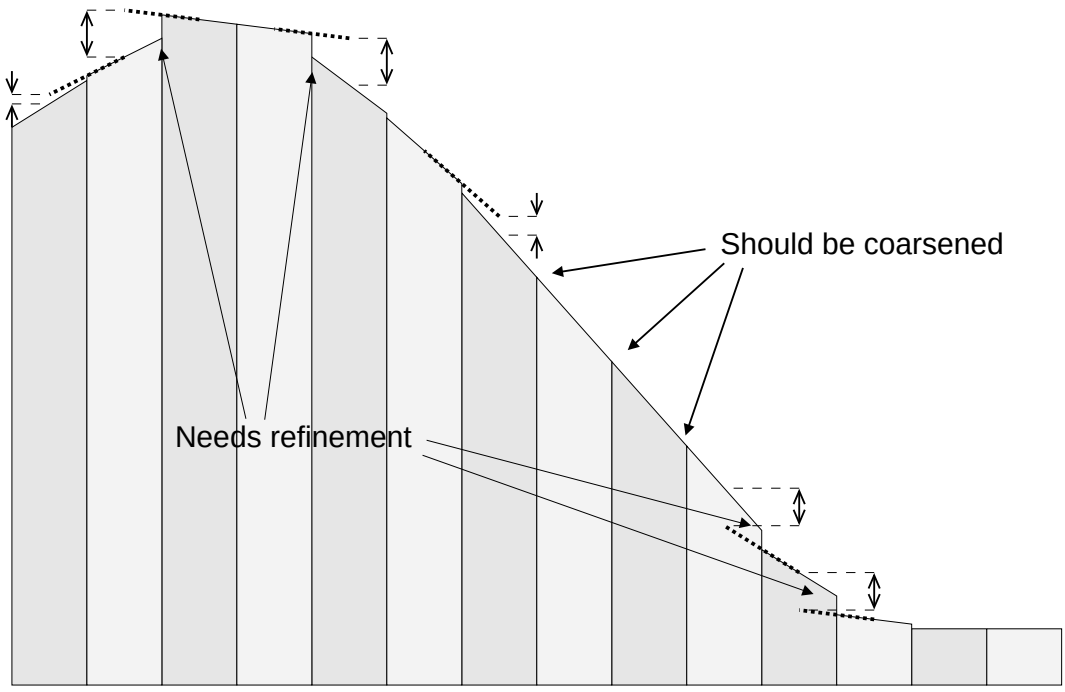

Figure B.1: Refinement indication using the ENO recoveries on each cell.
Where the approximated solution displays a high curvature in space, the indicator η_h of equation (B.2) yields high values. On the other hand, parts linear in space give $\eta_h = 0$ because they can already be represented perfectly with a second order of approximation.

This is not a true error estimator, it is merely a heuristic procedure indicating where refinement or coarsening should be performed. It is not completely unlike the estimation of higher order derivatives as proposed by e.g.Dolejší (1998) or Muzaferija and Gosman (1997): it indicates the places where the used order of approximation does not represent the data well. But due to the employment of the ENO recovery, the above procedure is more related to multi resolution representations derived from the interpolation frameworks of approximation theory (Harten, 1996). These locally increase the resolution of sampling points of the interpolated function if the order of approximation is not sufficient to accurately represent it, i.e. where the function is not smooth enough. Condition (B.1) is probably not guaranteed by the indicator (B.2) because of the hyperbolic nature of the equations: as seen in Section 5.1, not only the solution is propagated in space with time but also the error. Although the indicator (B.2) is suitable to bound the sources of errors, the error originating from different regions in the computational domain might still converge and accumulate in some place.

Appendix C

Software Model and Implementation

From a certain point of view a3m is nothing but a computer program, a piece of software. From that perspective, utilising methodology from software engineering for the construction of a3m and employing a *software development process* seems reasonable; at least as a part of a3m's development as outlined in Chapter 1 and Figure 1.3. But existing software development processes (e.g. Larman, 1998) focus mainly on business applications and not on scientific software such as an atmospheric model. They employ a use-case driven approach to partition the complexity of a project into manageable tasks. There is basically only one use-case for a3m: a scientist conducts a simulation. Consequently, the use-case oriented development processes are not applicable without modification.

C.1 Design, Complexity and Responsibility

Since there is only one use-case, the reduction of complexity by use-cases to derive single design, implementation and testing tasks is not applicable for the development of a3m. Other approaches of partitioning the complexity of the problem to implement, the discrete model (Figure 1.3 and Chapter 4), into manageable units are

- to start with fewer spatial dimensions,
- to strip off functionality and program features that are not essential for obtaining first simulation results or even
- using a simpler physical problem than the atmospheric flow equations.

As an example, the implementation of a3m was begun with a one-dimensional advection equation and later the Burgers equation (5.1).

With this relatively low-complexity system the possibilities of implementing the numerical methods described in Chapter 4 have been explored. For this purpose, a *conceptual model* (see the 'Software Model' in Chapter 1 and Larman, 1998) was developed and fixed using the unified modelling language (UML) (UML, 1997; UML RTF, 1999). UML provides fairly general methods and notations for describing

processes and structures of all kinds so that some process can be first modelled and then that model can be implemented. The model is represented mainly using various kinds of diagrams. Figure C.1 shows the overall structure of a3m in the form of a package diagram. The functionality is grouped into five different categories:

- the physical problem at hand (advection, Burgers or Euler equations),
- input and output of model parameters, logging information and results,
- the FV solver for computing recoveries, fluxes and source terms,
- the grid with cells, faces, vertices and their neighbourship relations and
- memory management.

The program parts implementing this functionality are put into the corresponding packages shown in Figure C.1. The dashed arrows in the Figure express dependencies, i.e. package 'solv' needs, uses and relies on the packages 'problem' and 'grid' to provide the functionality in its responsibility.

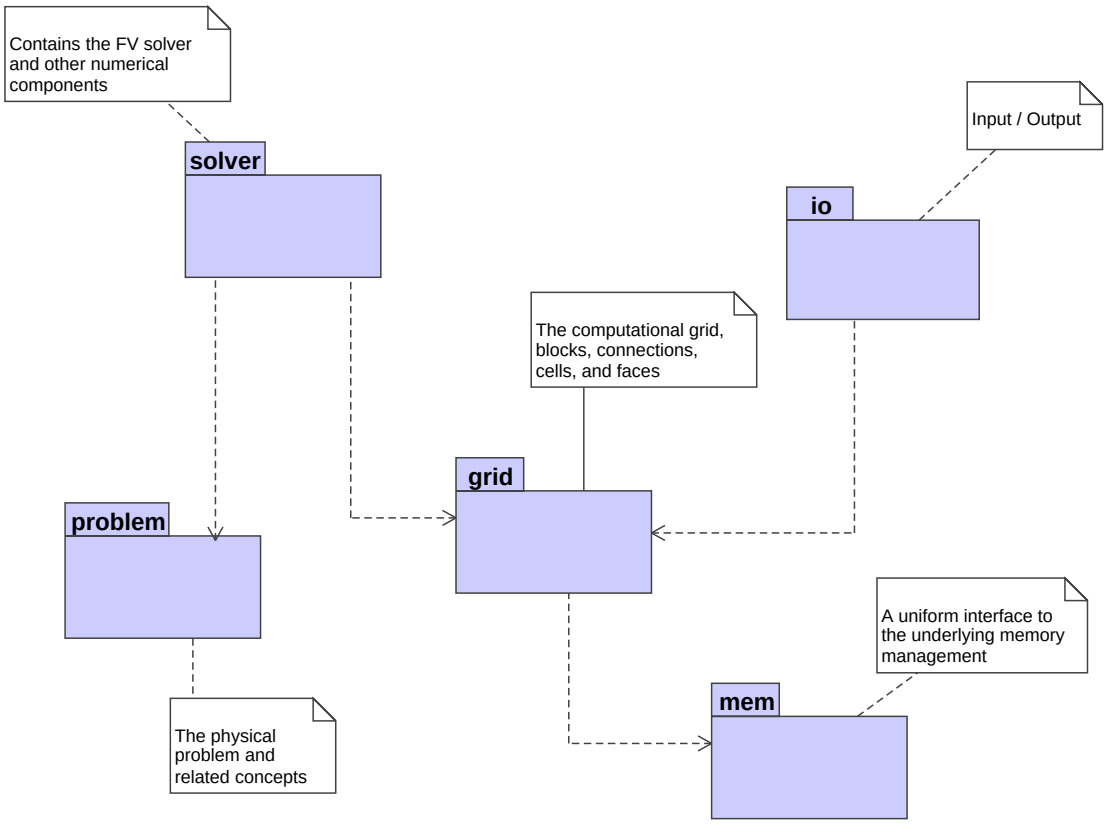

Figure C.1: Hierarchy of packages: the large-scale architecture of a3m.

Figure C.2 shows a class diagram of the 'grid' package. The diagram is on the conceptual level: it is part of the analysis of the FV scheme developed in Chapter 4. So it shows concepts and their interrelations rather than actual classes used for the implementation. Note the close correspondence between the concepts displayed in the figure and the terms used to formulate the FV method of a3m: grid, cell, face, neighbours, etc. and for locally refined grids by block-structuring: blocks and connections.

Figures C.1 and C.2 are static diagrams. They show concepts and components of a3m and their static relations but provide no information on the dynamic behaviour of the program: to accomplish a specific task the components have to act together. As an example, Figure C.3 shows *instances* of concepts (partially featured in Figure C.2) collaborating to achieve the computation of tendencies. The figure shows the sequence of actions taken for the Euler forward time integration (Section 4.3).

The different components participating in a collaboration contribute different parts of the functionality realised by the collaboration as a whole. Which component is responsible for which part of the functionality is mostly a question of which

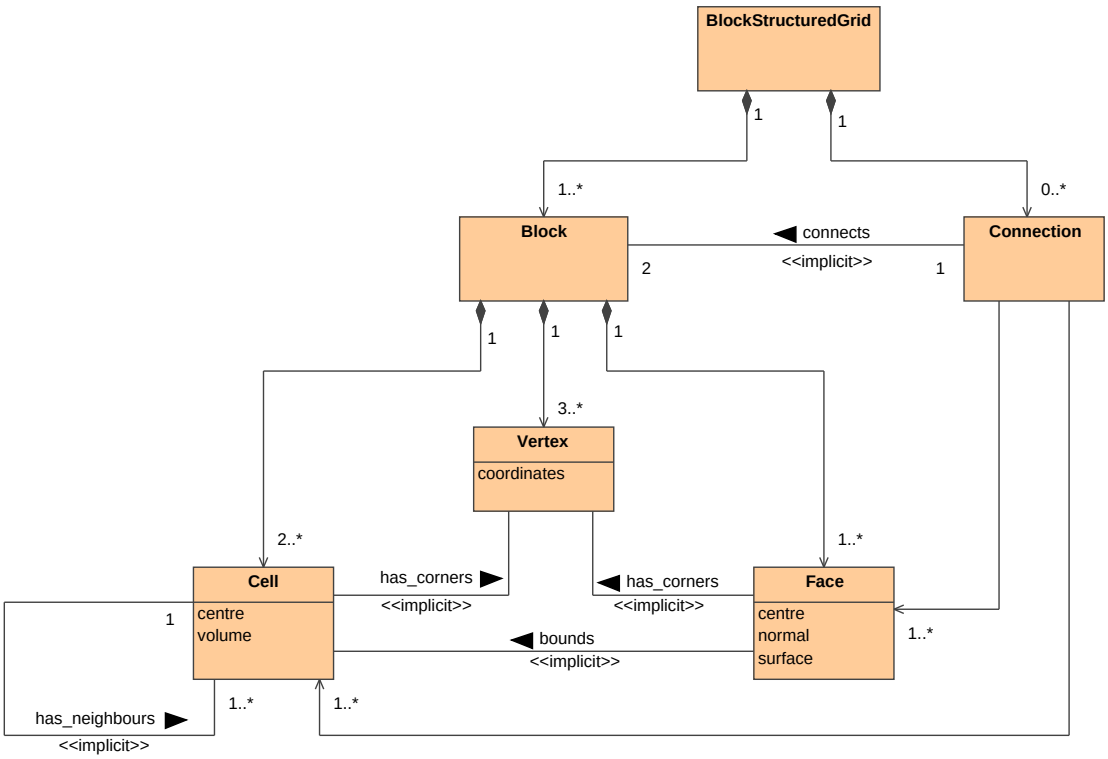

Figure C.2: Conceptual model of the FV grids of a3m.
The stereotype '≪implicit≫' means that the relations do not need to be explicitly represented in the implementation. They are implied by the structure of the grid.

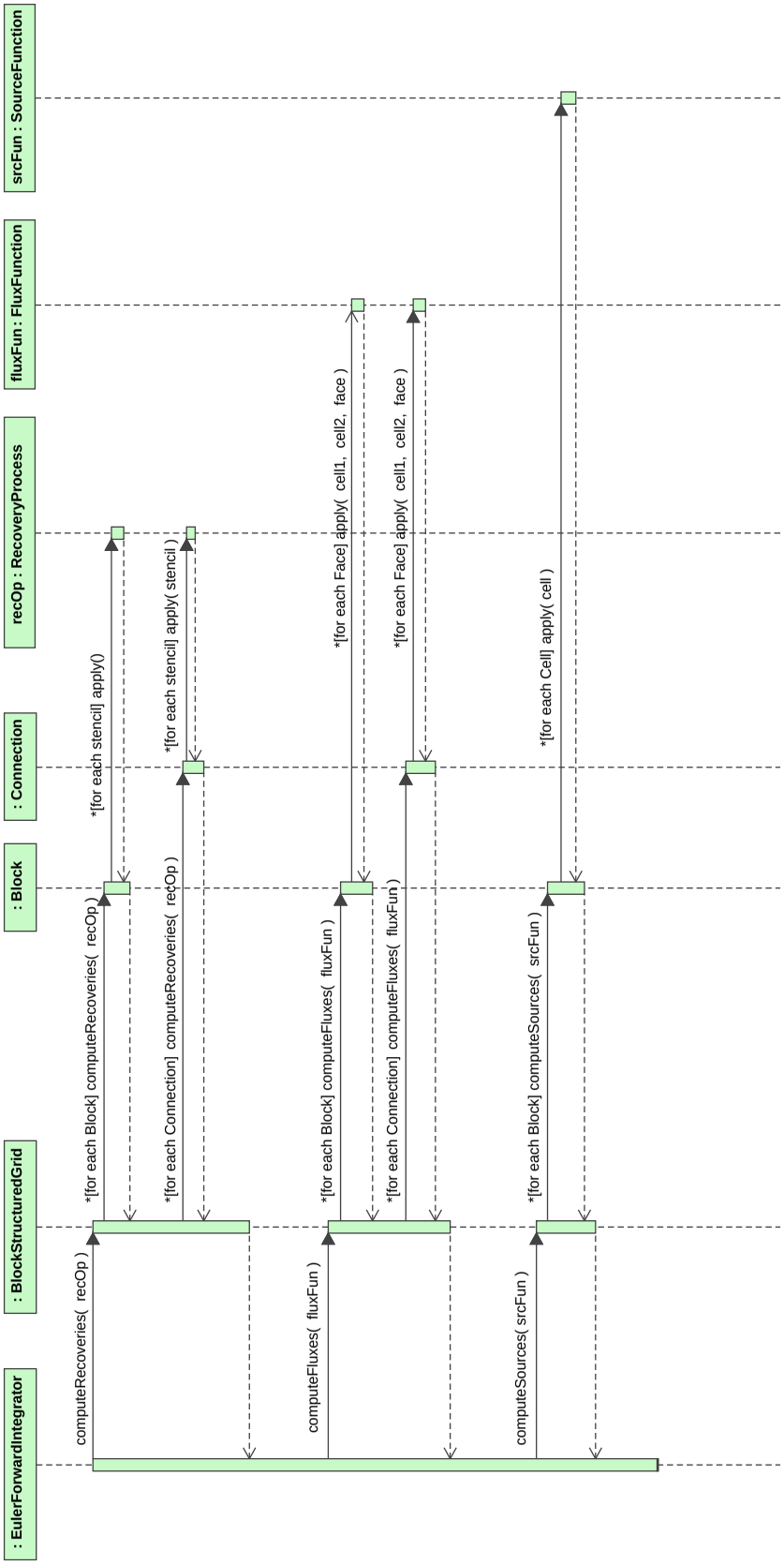

Figure C.3: Sequence diagram for the time integration process: computing the tendencies.

component has the required information to accomplish the task: if the required information is not known at some point, it has either to be queried and gathered from known sources or the task or a part thereof has to be delegated. For the example shown in Figure C.3 a `BlockStructuredGrid` is the only construct that has knowledge of `Block`s and `Connection`s. But it holds no other informations and therefore simply delegates the tasks of computing the recoveries, fluxes and source terms to its `Block`s and `Connection`s. Those in turn have information about the neighbour relationships between cells and faces. So it is their responsibility to pass the right combinations of cells and faces to the respective components of the FV solver, `RecoveryProcess`, `FluxFunction` and `SourceFunction`. As outlined in Section 4.2, the solver components do not need and therefore have no knowledge of the grids structure. Their responsibility lies in evaluating the terms of the FV scheme. To perform their tasks they operate merely on interpolation stencils, recovery polynomials, average values, cells (with a volume and centre), faces (with surface area, centre and normal vector), etc.

For details on the UML notations employed in the above diagrams see the UML notation guides UML (1997) and UML RTF (1999).

C.2 Implementation Issues

For the implementation of a3m the C++ programming language (ISO/IEC, 1998; Stroustrup, 1997; Josuttis, 1999) was chosen. It directly supports the object-oriented (OO) paradigm so that the designs outlined in Section C.1 can be realised directly in the programming language without additional constructs or conventions. An additional advantage is the support of generic and generative programming by C++ templates. Extensive use is made of this special construct in the implementation of a3m. It is one of the key features for achieving the high degree of extensibility and flexibility in the program.

The code of a3m conforms to the ISO C++ standard (ISO/IEC, 1998). This ensures high portability to different hardware platforms as long as they provide a standards compliant compiler. Additionally, coding conventions were defined and used to enable a more uniform use of language constructs and guarantee a mostly uniform appearance of the code. This enhances code readability and eases the comprehension of the program.

Notations and Symbols

The following table summarises the notations used througout this work and gives a list of the symbols designating the different variables and constants.

Notations

$\mathbb{R}, \mathbb{R}^+$	the sets of real numbers and of positive real numbers
$\underline{a} = (a_1, a_2, a_3)^T$	a vector in physical space $\mathbb{R}^3$ (a column vector)
$\underline{a}^T$	transpose of a vector: convert column to row vectors and vice versa
$\lvert\underline{a}\rvert$	length (Euclidian norm) of a vector
$A = (A_{ij})$	a matrix with the elements A_{ij}
A^{-1}	inverse of the matrix A
$A^T = (A_{ij})^T := (A_{ji})$	transpose of the matrix A
$a(\underline{x})$	a field: a function $\underline{x} \to a(\underline{x})$, $\mathbb{R}^3 \to \mathbb{R}^n$ of physical space, where the point in space is identified by the vector of coordinates $\underline{x} \in \mathbb{R}^3$
∂_t	partial differentiation operator with respect to time
∂_{x_i}	partial differentiation operator with respect to the ith spatial coordinate
$\nabla = (\partial_{x_1}, \partial_{x_2}, \partial_{x_3})^T$	partial differentiation operator with respect to all spatial coordinates (to form divergences and gradients)
∇_U	Jacobian operator (partial derivatives with respect to U)
∂X	boundary of a set X
$\overline{X} = X \cup \partial X$	closed set X, i.e. X including its boundary
$\mathring{X} = X \setminus \partial X$	opened set X, i.e. X without its boundary
$\int_X a(\underline{x})\, d\underline{x}$	integral of a field $a(\underline{x})$ over all points $\underline{x} \in X$ of a subset $X \subseteq \mathbb{R}^3$ of physical space; in this work a surface or volume integral
$\lvert X\rvert := \int_X d\underline{x}$	'content' of a set X: for cells and faces the specialised terms 'volume' and 'surface area' are used, respectively

Symbols of Variables & Constants

$\underline{c}_i := \int_{T_i} \underline{x}\, d\underline{x} / \lvert T_i\rvert$	centre of gravity of the cell T_i
$\underline{c}_{ij} := \int_{S_{ij}} \underline{x}\, d\underline{x} / \lvert S_{ij}\rvert$	centre of gravity of the face S_{ij}
$\mathsf{c} := \sqrt{\gamma \mathsf{p}/\rho}$	speed of sound
c_p	specific heat at constant pressure
c_V	specific heat at constant volume

$\mathsf{e} = \mathsf{u} + \lvert\underline{u}\rvert^2/2$	specific total energy
$F_l(U)$	advective flux in lth coordinate direction
$G_l(U)$	turbulent flux in lth coordinate direction
$\underline{g} = (g_1, g_2, g_3)^T$	Earth's acceleration of gravity
K_{ij}	turbulent exchange coefficient of the ith momentum components in the jth coordinate direction (and vice versa).
$K_{\mathsf{e}j}$	turbulent mixing coefficient of total energy e in the jth coordinate direction.
k	turbulent (sub-grid-scale) kinetic energy
l	turbulent mixing length (Section 3.3)
$\underline{n}_{ij}$	normal vector on the face S_{ij}
$P_i(\underline{x})$	recovery polynomial on cell T_i
p	pressure
$\mathsf{p}^{(0)}, \mathsf{p}^{(1)}$	basic state and derivation pressures: $\mathsf{p} = \mathsf{p}^{(0)} + \mathsf{p}^{(1)}$
$R := c_\mathsf{p} - c_V$	specific gas constant
S_{ij}	common face of cells T_i and T_j
$\mathsf{s} := \ln(\mathsf{p}/\rho^\gamma)$	specific entropy (thermodynamic)
T_i, T_j	cells of the computational grid
T	real temperature
$\mathsf{T}^{(0)}, \mathsf{T}^{(1)}$	basic state and derivation temperatures: $\mathsf{T} = \mathsf{T}^{(0)} + \mathsf{T}^{(1)}$
$U := (\rho, \rho\underline{u}, \rho\mathsf{e})^T$	the 'unknown': vector of conserved variables
$U_i := \int_{T_i} U \, d\underline{x}/\lvert T_i\rvert$	average value of U on cell T_i
$\underline{u} = (u_1, u_2, u_3)^T$	flow velocity
u_*	friction velocity
$\underline{\hat{u}} = (\hat{u}_1, \hat{u}_2, \hat{u}_3)^T$	flow velocity in rotated coordinates
u	specific inner energy
$\underline{x} = (x_1, x_2, x_3)^T$	a point in physical space designated by its position vector with the coordinates x_1, x_2 and x_3
δ_{ij}	Kronecker symbol, $\delta_{ij} = 1$ if $i = j$, and 0 otherwise
ϵ_{ijk}	Levi-Civita symbol, $\epsilon_{ijk} = +1$ if $(i,j,k) \in \{(1,2,3),(2,3,1),(3,1,2)\}$, $\epsilon_{ijk} = -1$ if $(i,j,k) \in \{(1,3,2),(2,1,3),(2,1,3)\}$, and 0 otherwise
$\gamma := c_\mathsf{p}/c_V$	ratio of specific heats
$\Theta := \mathsf{T}\,(\mathsf{p}/\mathsf{p}_\Theta)^{R/c_\mathsf{p}}$	potential temperature, $\mathsf{p}_\Theta = 1000\,\mathrm{hPa}$
$\Theta^{(0)}, \Theta^{(1)}$	basic state and derivation potential temperatures: $\Theta = \Theta^{(0)} + \Theta^{(1)}$
$\pi := (\mathsf{p}/\mathsf{p}_0)^{R/c_\mathsf{p}}$	Exner function
κ	von Kármán constant: a3m uses $\kappa = 0.4$

ρ	density
$\rho^{(0)}, \rho^{(1)}$	basic state and derivation densities: $\rho = \rho^{(0)} + \rho^{(1)}$
τ_{ij}	turbulent flux of the ith momentum components in the jth coordinate direction (and vice versa).
$\tau_{\mathsf{e}j}$	turbulent flux of total energy e in the jth coordinate direction.
$\underline{\Omega} = (\Omega_1, \Omega_2, \Omega_3)^T$	Earth's angular velocity

Bibliography

Abgrall, R., S. Lantéri and T. Sonar (**1999**). ENO approximations for compressible fluid dynamics. *Zeitschrift für Angewandte Mathematik und Mechanik* **79**, 3–28. `http://www3.interscience.wiley.com/cgi-bin/issuetoc?ID=350000%94`.

Atkinson, B. W. (**1981**). *Meso-scale Atmospheric Circulations*. Academic Press. ISBN 0-12-065960-3.

Berger, M. and J. Oliger (**1984**). Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics* **53**, 484–512.

Birnbaum, G. (**1998**). *Numerische Modellierung der Wechselwirkung zwischen Atmosphäre und Meereis in der arktischen Eisrandzone*. Berichte zur Polar und Meeresforschung **268**, Alfred-Wegener-Institut für Polar- und Meeresforschung, Bremerhaven.

Blackadar, A. K. (**1962**). The vertical distribution of wind and turbulent exchange in a neutral atmosphere. *Journal of Geophysical Research* **67**, 3095–3102.

Boris, J. P. and D. L. Book (**1973**). Flux-corrected transport I: SHASTA, a fluid transport algorithm that works. *Journal of Computational Physics* **11**, 38–69.

Bott, A. (**1989**). A positive definite advection scheme obtained by nonlinear renormalization of the advective fluxes. *Monthly Weather Review* **117**, 1006–1015.

Bott, A. (**1992**). Monotone flux limitation in the area-preserving flux-form advection algorithm. *Monthly Weather Review* **120**, 2592–2602.

Botta, N., R. Klein and A. Almgren (**1999**). *Dry Atmosphere Asymptotics*. PIK-Report **55**, Potsdam Institute for Climate Impact Research.

Carpenter, R. L., Jr., K. K. Droegemeier, P. R. Woodward and C. E. Hane (**1990**). Application of the piecewise parabolic method (PPM) to meteorological modeling. *Monthly Weather Review* **118**, 586–612.

Colella, P. and P. R. Woodward (**1984**). The piecewise parabolic method (PPM) for gas-dynamical simulations. *Journal of Computational Physics* **54**, 174–201.

Courant, R., K. Friedrichs and H. Lewy (**1928**). Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen* **100**, 32–74.

Cox, R., B. L. Bauer and T. Smith (**1998**). A mesoscale model intercomparison. *Bulletin of the American Meteorological Society* **79**, 265–283.

Deardorff, J. W. (**1978**). Efficient prediction of ground surface temperature and moisture, with inclusion of a layer of vegetation. *Journal of Geophysical Research* **83**, 1889–1903.

Dierer, S. (**1997**). *Validierung des mesoskaligen Transport- und Strömungsmodells METRAS*. Diplomarbeit, Fachbereich Geowissenschaften, Universität Hamburg.

Dierer, S. (**2002**). *Untersuchung von groß- und mesoskaligen Einflüssen auf die Entwicklung polarer Mesozyklonen mit Hilfe des Modells METRAS*. Doktorarbeit, Fachbereich Geowissenschaften, Universität Hamburg.

Dijkstra, E. W. (**1968**). Go to statement considered harmful. *Communications of the ACM* **11**, 147–148. `http://www.acm.org/classics/oct95/`.

Dolejší, V. (**1998**). Anisotropic mesh adaption for finite volume and finite element methods on triangular meshes. *Computing and Visualization in Science* **1**, 165–178.

Doms, G. and F. Herbert (**1985**). *Fluid- und Mikrodynamik in numerischen Modellen konvektiver Wolken*. Berichte des Instituts für Meteorologie und Geophysik **62**, Universität Frankfurt/Main.

Doms, G. and U. Schättler (**1999**). *The Nonhydrostatic Limited-Area Model LM (Lokal-Modell) of DWD*. Technical report, Deutscher Wetterdienst. Preliminary Version LM_F90.

Enquist, B. and S. Osher (**1981**). One-sided difference approximations for nonlinear conservation laws. *Mathematics of Computation* **36**, 321–351.

Ferziger, J. H. and M. Perić (**1999**). *Computational Methods for Fluid Dynamics*. Springer, Berlin, Heidelberg, New York, 2nd edition. ISBN 3-540-65373-2.

Fraedrich, K. and H. Flohn (**1966**). Tagesperiodische Zirkulation und Niederschlagsverteilung am Victoria-See (Ostafrika). *Meteorologische Rundschau* **19**, 157–165.

Germano, M. (**1992**). Turbulence: the filtering approach. *Journal of Fluid Mechanics* **238**, 325–336.

Grell, G. A., J. Dudhia and D. R. Stauffer (**1995**). *A Description of the Fifth-Generation Penn State/NCAR Mesoscale Model (MM5).* NCAR Technical Notice **NCAR/TN-398+STR**, National Center for Atmospheric Research. `http://www.mmm.ucar.edu/mm5/doc.html`.

Harten, A. (**1983**). High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics* **49**, 357–393.

Harten, A. (**1996**). Multiresolution representation of data: A general framework. *SIAM Journal on Numerical Analysis* **33**, 1205–1256. `http://links.jstor.org/`.

Harten, A. (**1997**). High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics* **135**, 260–278. `http://www.idealibrary.com/links/toc/jcph/135/2/0`, reprinted from Volume 49, Number 3, March 1983, pages 357–393.

Harten, A., B. Enquist, S. Osher and S. R. Chakravarthy (**1987**). Uniformly high order accurate essentially non-oscillatory schemes III. *Journal of Computational Physics* **71**, 231–303.

Hirsch, C. (**1988**a). *Numerical Computation of Internal and External Flows, Volume 1, Fundamentals of Numerical Methods.* Wiley, Chichester. ISBN 0-471-92385-0.

Hirsch, C. (**1988**b). *Numerical Computation of Internal and External Flows, Volume 2, Computational Methods for Inviscid and Viscous Flows.* Wiley, Chichester. ISBN 0-471-92452-0.

Holton, J. R. (**1992**). *An Introduction to Dynamic Meteorology*, volume 48 of *International Geophysics Series.* Academic Press, 3rd edition. ISBN 0-12-354355-X.

ISO/IEC (**1998**). *ISO/IEC 14882: Programming Languages – C++.* International Organization for Standardization / International Electrotechnical Commission. ISO standards document 14882:1998.

Jargon (**2002**). The jargon file version 4.3.1. `http://www.tuxedo.org/~esr/jargon/`, maintained by Eric S. Raymond.

Josuttis, N. M. (**1999**). *The C++ Standard Library: A Tutorial and Reference.* Addison Wesley. ISBN 0-201-37926-0. `http://www.josuttis.com/libbook/`.

Kapitza, H. and D. P. Eppel (**1992**). The non-hydrostatic mesoscale model GESIMA. part I: Dynamical equations and tests. *Contributions to Atmospheric Physics* **65**, 129–145.

Laird, N. F., D. A. R. Kristovich, X.-Z. Liang, R. W. Arritt and K. Labas (**2001**). Lake Michigan lake breezes: Climatology, local forcing, and synoptic environment. *Journal of Applied Meteorology* **40**, 409–424.

Larman, C. (**1998**). *Applying UML and Patterns: an Introduction to Object-Oriented Analysis and Design*. Prentice-Hall. ISBN 0-13-748880-7.

Lax, P. and B. Wendroff (**1960**). Systems of conservation laws. *Communications on Pure and Applied Mathematics* **13**, 217–237.

van Leer, B. (**1979**). Towards the ultimate conservative difference scheme, V. A second-order sequel to Godunov's method. *Journal of Computational Physics* **32**, 101–136.

van Leer, B. (**1997**). Towards the ultimate conservative difference scheme, V. A second-order sequel to Godunov's method. *Journal of Computational Physics* **135**, 229–248. `http://www.idealibrary.com/links/toc/jcph/135/2/0`, reprinted from Volume 32, Number 1, July 1979, pages 101–136.

Liu, X.-D., S. Osher and T. Chan (**1994**). Weighted essentially non-oscillatory schemes. *Journal of Computational Physics* **115**, 200–212.

Lüpkes, C. and K. H. Schlünzen (**1996**). Modelling the Arctic convective boundary-layer with different turbulence parameterizations. *Boundary-Layer Meteorology* **79**, 107–130.

Meister, A. and T. Sonar (**1998**). *Finite Volume Schemes for Compressible Fluid Flow*. Hamburger Beiträge zur Angewandten Mathematik, Reihe F **4**, Institut für Angewandte Mathematik, Universität Hamburg.

Mellor, G. L. and T. Yamada (**1974**). Hierarchy of turbulence closure models for planetary boundary layers. *Journal of the Atmospheric Sciences* **31**, 1791–1806.

Mesinger, F. and A. Arakawa (**1976**). *Numerical Methods used in Atmospheric Models*. GARP Publication Series **17**, Global Atmospheric Research Programme (GARP), WMO-ICSU Joint Organization Committee.

Monin, A. and A. Obukhov (**1954**). Basic turbulent mixing laws in the atmospheric surface layer. *Trudy Geofizecheskogo Instituta, Akademiya Nauk SSSR* **24**, 163–187.

Muzaferija, S. (**1994**). *Adaptive Finite Volume Method for Flow Prediction using Unstructured Meshes and Multigrid Approach*. Ph.D. thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London, UK.

Muzaferija, S. and D. Gosman (**1997**). Finite-volume CFD procedure and adaptive error control strategy for grids of arbitary topology. *Journal of Computational Physics* **138**, 766–787.

Neeman, H. J. (**1996**). *Autonomous Hierarchical Adaptive Mesh Refinement for Multiscale Simulations*. Ph.D. thesis, Department of Computer Schience, University of Illinois at Urbana-Champaign.

von Neumann, J. and R. D. Richtmyer (**1950**). A method for the numerical calculation of hydrodynamic shocks. *Journal Applied Physics* **21**, 380–385.

Niemeier, U. and K. H. Schlünzen (**1993**). Modelling steep terrain influences on flow patterns at the isle of Helgoland. *Contributions to Atmospheric Physics* **66**, 45–62.

Orlanski, I. (**1975**). A rational subdivision of scales for atmospheric processes. *Bulletin os the Americal Meteorological Society* **56**, 527–530.

Osher, S. and F. Solomon (**1982**). Upwind difference schemes for hyperbolic systems of conservation laws. *Mathematics of Computation* **38**, 339–374.

Pielke, R. A. (**1984**). *Mesoscale Meteorological Modeling*. Academic Press. ISBN 0-12-554820-6.

Pielke, R. A., W. R. Cotton, R. L. Walko, C. J. Tremback, W. A. Lyons, L. D. Grasso, M. E. Nicholls, M. D. Moran, D. A. Wesley, T. J. Lee and J. H. Copeland (**1992**). A comprehensive meteorological modeling system - RAMS. *Meteorology and Atmospheric Physics* **49**, 69–91.

Press, W. H., S. A. Teukolsky, W. T. Vettering and B. P. Flannery (**1992**). *Numerical Recipes in C*. Cambridge University Press, 2nd edition. ISBN 0-521-43108-5. `http://www.nr.com`.

Roache, P. J. (**1982**). *Computational Fluid Dynamics*. Hermosa, Albuquerque. Revised Printing.

Rogallo, R. S. and P. Monin (**1984**). Numerical simulation of turbulent flows. *Annual Reviews of Fluid Mechanics* **16**, 99–137.

von Salzen, K. and K. H. Schlünzen (**1999**a). A prognostic physico-chemical model of secondary and marine inorganic multicomponent aerosols I. Model description. *Atmospheric Environment* **33**, 567–576.

von Salzen, K. and K. H. Schlünzen (**1999**b). Simulation of the dynamics and composition of secondary and marine inorganic aerosols in the coastal atmosphere. *Journal of Geophysical Research* **104**, 30 201–30 217.

Schlünzen, H. (**1988**). *Das mesoskalige Transport- und Strömungsmodell METRAS – Grundlagen, Validierung, Anwendung*. Number A 88 in Hamburger Geophysikalische Einzelschriften, Geophysikalische Institute der Universität Hamburg and Max-Planck-Institut für Meteorologie.

Schlünzen, H., K. Bigalke, K.-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.

Schlünzen, K. H. (**1990**). Numerical studies on the inland penetration of sea breeze fronts at a coastline with tidally flooded mudflats. *Contributions to Atmospheric Physics* **63**, 243–256.

Schlünzen, K. H. (**1994**). Mesoscale modelling in complex terrain—an overview on the German nonhydrostatic models. *Contributions to Atmospheric Physics* **67**, 243–252.

Schlünzen, K. H. (**1996**). *Validierung hochauflösendenr Regionalmodelle*. Berichte aus dem Zentrum für Meeres- und Klimaforschung **30**, Zentrum für Meeres- und Klimaforschung, Universität Hamburg, Hamburg. Habilitationsschrift.

Schlünzen, K. H. (**1997**). On the validation of high-resolution atmospheric mesoscale models. *Journal of Wind Engineering and Industrial Aerodynamics* **67**, 479–492.

Schlünzen, K. H. and U. Krell (**1994**). Mean and local transport in air. In: Sündermann (editor), *Circulation and contaminant fluxes in the North Sea*, 317–344. Springer Verlag.

Sha, W., T. Kawamura and H. Ueda (**1991**). A numerical study on sea/land breezes as a gravity current: Kelvin-Helmholtz billows and inland penetration of the sea-breeze front. *Journal of the Atmospheric Sciences* **48**, 1649–1665.

Shapiro, R. (**1970**). Smoothing, filtering, and boundary effects. *Reviews of Geophysics and Space Physics* **8**, 359–387.

Shapiro, R. (**1971**). The use of linear filtering as a parameterization of atmospheric diffusion. *Journal of the Atmospheric Sciences* **28**, 523–531.

Sheng, L., K. H. Schlünzen and Z. Wu (**2000**). Three-dimensional numerical simulation of the mesoscale wind structure over Shandong Peninsula. *Acta Meteorologica Sinica* **14**, 96–107.

Shu, C.-W. (**1987**). TVB uniformly high-order schemes for conservation laws. *Mathmatics of Computation* **49**, 105–121.

Shu, C.-W. and S. Osher (**1988**). Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics* **77**, 439–471.

Simpson, J. E. (**1987**). *Gravity Currents: In the Environment and the Laboratory.* Ellis Horwood. ISBN 0-85312-972-X.

Simpson, J. E. (**1994**). *Sea Breeze and Local Winds.* Cambridge University Press. ISBN 0-521-45211-2.

Skamarock, W. C. (**1989**). Truncation error estimates for refinement criteria in nested and adaptive models. *Monthly Weather Review* **117**, 872–886.

Skamarock, W. C. and J. B. Klemp (**1992**). The stability of time-split numerical methods for the hydrostatic and nonhydrostatic elastic equations. *Monthly Weather Review* **120**, 2109–2127.

Skamarock, W. C. and J. B. Klemp (**1993**). Adaptive grid refinement for two-dimensional and tree-dimensional nonhydrostatic atmospheric flow. *Monthly Weather Review* **121**, 788–804.

Smolarkiewicz, P. K. (**1983**). A simple positive definite advection scheme with small implicit diffusion. *Monthly Weather Review* **111**, 479–486.

Sod, G. A. (**1978**). A survey of several finite difference schemes for hyperbolic conservation laws. *Journal of Computational Physics* **27**, 1–31.

Sonar, T. (**1997**). *Mehrdimensionale ENO Verfahren.* Teubner, Stuttgart. ISBN 3-519-02724-0.

Sonar, T. and E. Süli (**1998**). A dual graph-norm refinement indicator for finite volume approximations of the Euler equations. *Numerische Mathematik* **78**, 619–658. `http://link.springer.de/link/service/journals/00211/bibs/8078%004/80780619.htm`.

Spekreijse, S. P. (**1988**). *Multigrid solutions of the steady Euler equations*, volume 46 of *CWI Tract*. Centrum voor Wiskunde en Informatica, Amsterdam, Netherlands.

Stroustrup, B. (**1997**). *The C++ Programming Language*. Addison-Wesley, 3rd edition. ISBN 0-201-88954-4. `http://www.aw.com/cseng/titles/0-201-88954-4`.

Stull, R. B. (**1988**). *An Introduction to Boundary Layer Meteorology*. Number 13 in Atmospheric and Oceanographic Sciences Library, Kluwer Academic Publishers. ISBN 90-277-2768-6, 90-277-2769-4 (paperback).

Sündermann, A. (**1990**). *Die Anwendung von adaptiven Gittern in zwei einfachen Modellen für eine atmosphärische Rollenzirkulation*. Berichte aus dem Zentrum für Meeres- und Klimaforschung **5**, Zentrum für Meeres- und Klimaforschung, Universität Hamburg. Dissertation.

Tapp, M. C. and P. W. White (**1976**). A non-hydrostatic mesoscale model. *Quarterly Journal of the Royal Meteorological Society* **102**, 227–295.

Toro, E. F. (**1997**). *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer. ISBN 3-540-61676-4.

Tremback, C. J., J. Powell, W. R. Cotton and R. A. Pielke (**1987**). The froward-in-time upstream advection scheme: Extension to higher orders. *Monthly Weather Review* **115**, 540–555.

UML (**1997**). *UML 1.1 Notation Guide*. `http://www.rational.com/uml/resources/documentation/formats.j%tmpl`.

UML RTF (**1999**). *OMG Unified Modeling Language Specification, Version 1.3*. Object Management Group. `http://cgi.omg.org/cgi-bin/doc?ad/99-06-08`, oMG UML v. 1.3 is the UML RTF's proposed final revision to the UML 1.1 specification.

Wippermann, F. K. (**1981**). The applicability of several approximations in mesoscale modelling – a linear approach. *Contributions to Atmospheric Physics* **54**, 298–308.

Xue, M., K. Droegemeier and V. Wong (**2000**). The advanced regional prediction system (ARPS) - a multi-scale nonhydrostatic atmospheric simulation and prediction model. Part I: Model dynamics and verification. *Meteorology and Atmospheric Physics* **75**, 161–193.

Xue, M., K. K. Droegemeier, V. Wong, A. Shapiro and K. Brewster (**1995**). *ARPS User's Guide*. Center for the Analysis and Prediction of Storms, Norman, Oklahoma. `http://www.caps.ou.edu/ARPS/ARPS4.guide.html`, covering ARPS version 4.0.

Xue, M., K. K. Droegemeier, V. Wong, A. Shapiro, K. Brewster, F. Carr, D. Weber, Y. Liu and D. Wang (**2001**). The advanced regional prediction system (ARPS) - a multi-scale nonhydrostatic atmospheric simulation and prediction tool. Part II: Model physics and applications. *Meteorology and Atmospheric Physics* **76**, 143–165.

Yost, S. A. and P. Rao (**2000**). A moving boundary approach for one-dimensional free surface flows. *Advances in Water Resources* **23**, 373–382.