# Cut Cell Methods in Global Atmospheric Dynamics

**Dissertation**

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Jutta Adelsberger

aus

Moers

Bonn 2014

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Michael Griebel

2. Gutachter: Prof. Dr. Marc Alexander Schweitzer

Tag der Promotion: 12. Februar 2014

Erscheinungsjahr: 2014

# Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit der nächsten Generation von Techniken zur Simulation globaler dreidimensionaler Atmosphärenströmungen, die sich sowohl in Bezug auf die Modellierung, Gittergenerierung als auch Diskretisierung andeutet.

Anhand einer detaillierten Dimensionsanalyse der kompressiblen Navier-Stokes Gleichungen für klein- und großskalige Strömungen in der Atmosphäre leiten wir die kompressiblen Euler Gleichungen her, den sogenannten dynamischen Kern meteorologischer Modelle. In diesem Zusammenhang geben wir auch einen Einblick in die Multiskalenmodellierung und zeigen einen neuen numerischen Weg auf, reduzierte Atmosphärenmodelle herzuleiten und dabei eine Konsistenz im Modellierungs- und Diskretisierungsfehler zu erhalten.

Der Schwerpunkt dieser Arbeit liegt jedoch auf der Gittergenerierung. Im Hinblick auf immer feiner aufgelöste Vermessungen der Erdoberfläche und immer größere Rechnerkapazitäten sind die Methoden der Atmosphärentriangulierung neu zu bedenken. Insbesondere die weit verbreiteten geländefolgenden Koordinaten erweisen sich als nachteilig für hochaufgelöste Gitter, da diese den Fehler in der Druckgradientkraft und der hydrostatischen Inkonsistenz dieser Methode erheblich verstärken.

Nach einer detaillierten Analyse von Standardverfahren der vertikalen Atmosphärentriangulierung präsentieren wir die Cut Cell Methode als leistungsfähige Alternative. Wir konstruieren einen speziellen Cut Cell Ansatz mit zwei Stabilisierungsbedingungen und geben eine ausführliche Anleitung zur Implementation von Cut Cell Methoden in existierende Atmosphärencodes.

Zur Diskretisierung des dynamischen Kerns auf unseren so erzeugten Gittern bieten sich Finite Volumen Methoden an, da sie u.a. wegen ihrer Erhaltungseigenschaften besonders gut für die hyperbolischen Euler Gleichungen geeignet sind. Wir ergänzen die Finite Volumen Diskretisierung um ein neues nichtlineares Interpolationsschema des Geschwindigkeitsfeldes, das speziell an die Geometrie der Erde und der Atmosphäre angepasst ist.

Abschließend demonstrieren wir die Leistungsfähigkeit unseres Cut Cell Ansatzes in Kombination mit den dargestellten Diskretisierungs- und Interpolationsschemata anhand dreidimensionaler Simulationen. Wir verwenden Standardtestfälle wie einen Advektionstest und die Simulation einer Rossby-Haurwitz Welle und konstruieren weiterhin einen neuen Fall von Strömungen zwischen Hoch- und Tiefdruckgebieten, der geeignet ist, das Potential von Cut Cell Gittern und die Einflüsse verschiedener Effekte der Euler Gleichungen sowie der Topographie der Erde herauszustellen.

**Danksagung**

An dieser Stelle möchte ich mich bei allen bedanken, die mir in der Promotionszeit mit Rat und Tat zur Seite standen. Allen voran gilt mein Dank Prof. Dr. Michael Griebel für das interessante Thema, seine vielen Anregungen und Diskussionen sowie für die Bereitstellung von exzellenten Arbeitsbedingungen. Des weiteren bedanke ich mich herzlich bei Prof. Dr. Marc Alexander Schweitzer sowohl für die Übernahme des Zweitgutachtens als auch für seine stets offene Tür.

Besonderer Dank gilt all meinen Kollegen am Institut für Numerische Simulation für die freundschaftliche Atmosphäre und stete Hilfsbereitschaft. Insbesondere danke ich Christian Neuen, Alexander Rüttgers und Margrit Klitz für wertvolle Diskussionen und aufmerksames Korrekturlesen. Ein Dank gebührt außerdem Daniel Wissel für die schöne Zeit im gemeinsamen Büro sowie Ralph Thesen für seine Hilfe in allen Rechner- und Lebenslagen.

Nicht zuletzt möchte ich mich ganz herzlich bei Christian und meinen Eltern für all ihre Unterstützung und Ermutigung bedanken.

Bonn, im Januar 2014 Jutta Adelsberger

# Contents

# 1 Introduction

## Numerical Weather Forecast

Weather plays an important role in our day-to-day life, and thus its prediction has always been of special interest to mankind. In early times, a weather forecast relied on observations and experience. Only in the middle of the 20th century, first mathematical models were developed, based on physical laws and supported by meteorological measurements, with which a prediction of the weather could be computed.

These weather simulations are based on the description of atmospheric dynamics by natural laws. Thereby, quantities like wind velocity, air pressure, density, and temperature can be defined and a system of equations derived, which represents the mathematical formulation of the natural principles and which describes the temporal evolution of the aforementioned variables. In this way, a system of non-linear partial differential equations arises, for which no analytical solution is known and which thus has to be solved approximately.

Numerical weather forecasts are still an area of intensive research. The aim is to constantly improve the quality of the forecasts by means of the applied models, grids, and numerical techniques. Here, the rapid development of computing capacities plays a decisive role since they allow for more and more elaborate models and highly resolved computations. In recent years, this led to an increased renunciation of reduced models and simplifying methods in favor of more complex approaches which provide more reliable prognoses.

## Challenges of Atmospheric Dynamics

Currently, atmospheric research groups all over the world move towards the next generation of dynamical cores and their corresponding grids and numerical schemes. The dynamical core consists of the basic dynamic equations of fluid flow and forms the crucial part of any meteorological system from a numerical point of view. Due to the newly available computing capacities, reduced models for special scales, which were necessary for manageable time and memory ressources and which dominated the atmospheric com-

munity for a long time, are more and more abandoned. Instead, the full compressible Euler equations are favored, which describe any atmospheric flow on any scale and thus form the most general model possible. The hyperbolicity of these equations is a challenge on its own since the possible occurences of shocks complicate the analysis and the suitable numerical schemes exceedingly.

Another driving force for next generation models are new highly resolved global digital elevation models (GDEM) of the Earth's surface, which provide a never before seen resolution and accuracy. Only recently, a new freely available data set was released, called ASTER GDEM [Min09, AST09], with a spatial spacing of 1 arc-second or approximately 30 m. Although already very highly resolved, the elevation model will soon be outperformed by the data of TanDEM-X [Ger10, Ger13], a data acquisition of twin satellites of the German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt, DLR). The new GDEM is announced for 2014 with a resolution of 0.4 arc-seconds, which corresponds to approximately 12 m. Of course, the ability to make use of elevation data with such resolutions is also coupled to the development of high-performance parallel computers.

In this context, the grid generation of the atmosphere is also an area of intensive research. First of all, in horizontal direction, there is a demand for grids which cover the Earth's surface as evenly as possible and particularly avoid singularities at the poles. Moreover, the option of adaptivity should be incorporated so that e.g. rough terrain or special regions can be higher resolved. Germany's National Meteorological Service (Deutscher Wetterdienst, DWD) currently applies two coupled grids, a highly resolved local model LM [DSB11, DFH$^+$11] of Central Europe and a coarsely resolved global model GME [MLP$^+$02], which provides the boundary conditions for LM. Such a splitting of computational domains is accompanied by serious disadvantages, particularly the violation of conservation properties at the boundaries. Therefore, the DWD has made an effort to develop a next generation global model ICON [Bon04, GKZ11], which is based on an adaptively refined icosahedron and which shall be suitable for both weather and climate forecasts. Its first operational use was expected for 2013 but is still pending.

Generally, the tendency goes to the construction of "one grid for all", in respect of local and global grids, weather and climate applications, as well as atmosphere and ocean dynamics. The latter is realized in the recent Ocean-Land-Atmosphere Model OLAM [WA08, WA11], which includes both the flow in the atmosphere and in the ocean with their reciprocal effects. Such an approach is called a unified Earth System Model.

But not only the horizontal grid generation is constantly improved, the vertical principle is of even more interest. The highly anisotropic extensions of the atmosphere is a difficult challenge for the generation of a stable and manageable grid – even for today's high-performance computers. Atmospheric grids have long been solely dominated by terrain-following vertical coordinates which follow the curvature of the terrain and which are still widely used in nearly all present weather forecast systems. Only now, the drawbacks of this vertical principle become increasingly evident. Terrain-following coordinates suffer from a severe pressure gradient force error and hydrostatic inconsis-

tency. Usually, these have been damped by artificial diffusion terms, see e.g. [PST04], which change the originally hyperbolic equations in a generally unacceptable way. Nevertheless, the main problem of terrain-following coordinates nowadays is their inability to cope with very highly resolved horizontal triangulations. Namely, both the pressure gradient force error and the hydrostatic inconsistency depend on the skewness of cells and thus increase with finer mesh resolution since cells tend to be steeper for finer grids. With respect to the demand for higher and higher resolved computations, this is a serious drawback.

Less-known in atmospheric dynamics is the cut cell approach which constructs an orthogonal Cartesian grid with boundary cells cut by the terrain. Up to now, cut cells are predominantly used in applications with complex geometries [PB79, LeV88a, ICM03] and found their way into oceanic and atmospheric dynamics only recently [AHM97, SBJ$^+$06, WA08]. However, the application of cut cell techniques in today's weather forecast systems is still pending. A reason may be the so-called small cell problem which has to be dealt with in a suitable way. Typically, the boundary cells have arbitrary shapes and sizes since they are cut by the geometry. As known from the Courant-Friedrichs-Lewy criterion [CFL28], the time step necessary for a stable explicit or semi-implicit simulation procedure depends on the smallest cell of the grid. Therefore, an arbitrarily small cell leads to an arbitrarily small time step, and thus the computation process takes an arbitrarily long time. This impracticable restriction is the small cell problem which has to be circumvented.

Different remedies have been proposed for the small cell problem, but not all of them are suited for the application in atmospheric grids, and they are frequently also attached to other drawbacks. By studying the state-of-the-art, we received the impression that the following citation from 2009 is still up-to-date:

*"Although* [...] *there are cut-cell codes currently in use* [...]*, the approach is not mature and it is at the forefront of advanced research in universities and national laboratories."*
N. Nikiforakis, [Nik09]

Apart from the demanding mesh generation, the discretization schemes of the dynamical core are currently also undergoing a transformation. Reduced atmospheric models were often discretized by Finite Difference schemes, but the full compressible Euler equations represent conservation principles and thus require schemes which guarantee the conservation of mass, momentum, and energy. Here, the Finite Volume method [LeV02, Krö97] is a natural choice since it conserves quantities by construction. Furthermore, Finite Volumes are especially suitable for unstructured grids as well as hyperbolic equations since they are capable of representing discontinuous solutions.

Finally, testing the dynamical core of a three-dimensional general circulation model (GCM) with special grids and discretization schemes is not straightforward. The simulation results can neither be compared with analytical solutions since no non-trivial solutions are known, nor be verified by actual measurements because the dynamical core

is isolated from the physical parameterization. Therefore, model evaluations have to rely on intuition, experience, and model intercomparisons. For a long time, three-dimensional test cases were rare to find. Whereas a test suite for the two-dimensional shallow water equations has long been standardized [WDH+92], a set of three-dimensional benchmarks for atmospheric GCMs is only presently established together with a community devoted to the intercomparison of different GCMs, the so-called Dynamical Core Model Intercomparison Project [JLNT08, UJK+12]. Such efforts motivate the development of new techniques in a significant way.

### Contributions of this Thesis

Our own contributions in the context of next generation dynamical cores, atmospheric grids, and numerical schemes are as follows.

- We present a detailed dimensional analysis of the three-dimensional compressible Navier-Stokes equations for small- and large-scale flow in the atmosphere, reasoning the application of the full Euler equations. In this context, we also give an insight into multiscale modeling and a new numerical view at the derivation of reduced atmospheric models, which has the potential of simplifying the error analysis considerably due to a new consistency of the modeling and discretization error.

- The main focus of this thesis is a systematic comparison of vertical principles for atmospheric mesh generations and a thorough summary of the state-of-the-art in cut cell methods. We create a special cut cell approach with two stabilizing constraints and provide a comprehensive guideline for an implementation of cut cells into existing atmospheric codes, which has not been available so far.

- We accompany our Finite Volume discretization by a new interpolation scheme of the velocity field, formerly developed by the author in [Ade08], which is adapted to the geometry of the Earth and its atmosphere. Its quality is verified in further benchmark tests.

- We demonstrate the performance of cut cell grids in combination with our discretization and interpolation schemes in different stable simulation runs. Apart from two standard benchmarks, an advection test and a Rossby-Haurwitz wave, we construct a new benchmark case suitable for testing the dynamical core of a three-dimensional GCM. This test illustrates the capabilities of cut cell grids, different physical effects of the governing equations and the influence of the topography.

### Outline

The remainder of the thesis is organized as follows. We start in Section 2 with the modeling of the dynamical core of atmospheric dynamics. We derive the compressible Euler equations based on a detailed dimensional analysis for small- and large-scale atmospheric flow and complement them by turbulence modeling. Moreover, we provide

an insight into multiscale modeling and a numerical point of view at the derivation of reduced atmospheric models.

Grid generation is the central theme of this thesis. In Section 3, we concentrate on the horizontal triangulation of the Earth's surface based on global digital elevation models. Here, a cubed sphere approach with a bisection strategy and optional adaptivity leads to global grids of the Earth's topography.

Afterwards, in Section 4, we focus on the vertical grid generation. We review the common step-mountain and terrain-following approaches with their various drawbacks and present the cut cell approach as a capable alternative. A detailed comparison shows the superiority of the latter method, and we close the section with a comprehensive guideline for an implementation of cut cells into existing atmospheric codes together with two necessary stabilizing steps.

In Section 5, we discretize our governing equations in space and time by Finite Volumes and the implicit Euler method and thus derive a sparse system of linear equations for each variable and each time step. In this context, we present a new Earth interpolation scheme for the velocity field.

Our numerical approaches are verified by simulation runs in Section 6. An advection test, a benchmark with flow between high- and low-pressure areas as well as a Rossby-Haurwitz test case illustrate the capabilities of cut cell grids in contrast to terrain-following coordinates together with our discretization and interpolation schemes.

We finally conclude the thesis in Section 7 with a summary and an outlook to further interesting studies.

# 2 Atmospheric Modeling

A mathematical model is the basis of every numerical simulation. Such a model translates real phenomena into a mathematical problem such as a system of partial differential equations. Starting with fundamental laws of nature and their mathematical analogon and adding special forces or terms which depend on the desired application, we are able to formulate a model which allows us to numerically find an approximation to the solution.

A model which describes the dynamics of a planetary atmosphere or ocean in a rotating reference system is called general circulation model, abbreviated GCM. In this thesis, we are particularly interested in a GCM for atmospheric flows of the Earth, with weather forecasts as intended application. Such a model involves the basic dynamic equations of fluid flow, the well-known Navier-Stokes equations, with special contributions due to the Earth's gravity and rotation. The resulting set of equations forms the so-called dynamical core of atmospheric flows. Generally, a GCM may further consist of additional equations representing special properties depending on the actual application. In the thesis at hand, we focus on the dynamical core since it is the crucial part of any meteorological modeling from the numerical point of view.

In most of the literature and in actual forecast systems in use, simplified models of the dynamical core are used resulting in a reduction of the applications to special cases. The induced errors are often neglected although their impact is of utmost significance. Therefore, we feel a need to give a compact overview of atmospheric modeling with as few simplifying assumptions to the dynamical core as possible.

## 2.1. Governing Equations

First of all, we give a brief introduction to the governing equations and their derivations. For details, see [Ade08, KV03, GDN98].

In the following, let $\Omega \subset \mathbb{R}^3$ be a three-dimensional domain, $\boldsymbol{x} \in \Omega$ a position vector and $t \in [0, t_{\text{end}}]$ the time. The flow of a fluid in domain $\Omega$ at time $t$ is characterized by

the variables

$$\begin{aligned}
\boldsymbol{u}: &\quad \Omega \times [0, t_{\text{end}}] \to \mathbb{R}^3 && \text{velocity,}\\
\rho: &\quad \Omega \times [0, t_{\text{end}}] \to \mathbb{R} && \text{density,}\\
p: &\quad \Omega \times [0, t_{\text{end}}] \to \mathbb{R} && \text{pressure, and}\\
T: &\quad \Omega \times [0, t_{\text{end}}] \to \mathbb{R} && \text{temperature as energy substitute.}
\end{aligned}$$

All fluid flows are based upon the fundamental physical laws of conservation, namely the conservation laws of mass, momentum, and energy. The resulting system is called Navier-Stokes equation system.

### 2.1.1. Conservation of Mass

When particles are in motion, their mass is preserved; only the occupied volume – and thus the density – may change. The mass $M$ of a fluid occupying domain $\Omega_t := \Omega(t)$ at time $t$ can be expressed by the integral of the density of the fluid

$$M(t) = \int_{\Omega_t} \rho(\boldsymbol{x}, t)\, d\boldsymbol{x}.$$

Since mass of a moving fluid is maintained, the time derivative of mass has to vanish. Using Reynolds' transport theorem, this leads to

$$\int_{\Omega_t} \left(\rho_t + \nabla \cdot (\rho \boldsymbol{u})\right)(\boldsymbol{x}, t)\, d\boldsymbol{x} = 0 \tag{2.1}$$

with the shortened notation $\rho_t := \frac{\partial}{\partial t}\rho$. The equation holds for any domain $\Omega_t$ and thus in particular for arbitrarily small domains, too. This argument allows us to pass on to the differential form of the conservation law of mass

$$\rho_t + \nabla \cdot (\rho \boldsymbol{u}) = 0, \tag{2.2}$$

which is also called continuity equation.

### 2.1.2. Conservation of Momentum

The momentum of a solid body is defined as the product of its mass and its velocity. In the case of a fluid, the momentum $\boldsymbol{m}$ of a control volume $\Omega_t$ at time $t$ is written as

$$\boldsymbol{m}(t) = \int_{\Omega_t} \rho(\boldsymbol{x}, t)\boldsymbol{u}(\boldsymbol{x}, t)\, d\boldsymbol{x}.$$

Furthermore, according to Newton's Second Law, the temporal change of momentum equals the sum of all forces acting on the fluid. By elementary transformations we get

the differential form of the momentum equation

$$(\rho \boldsymbol{u})_t + \nabla \cdot (\rho \boldsymbol{u} \circ \boldsymbol{u}) + \nabla p = -\rho g \boldsymbol{k} - \nabla \cdot \boldsymbol{\tau}.$$

Here, $\boldsymbol{u} \circ \boldsymbol{u} := \boldsymbol{u}\boldsymbol{u}^\mathsf{T}$, $g$ denotes the gravitational constant of the Earth, $\boldsymbol{k}$ the unit normal vector of the Earth and $\boldsymbol{\tau}$ the viscous stress tensor which describes the molecular friction

$$\boldsymbol{\tau} := -\mu \left( \nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^\mathsf{T} - \frac{2}{3} (\nabla \cdot \boldsymbol{u}) \mathbb{1} \right) \tag{2.3}$$

with the dynamic viscosity $\mu$ and the identity matrix $\mathbb{1}$.

Note that we already added a term due to our special setting, namely the gravitational force $-\rho g \boldsymbol{k}$ acting downward to the center of the Earth. This is a simplification, since gravity actually varies throughout the Earth for different reasons. Before we take a closer look at these reasons in Figure 2.1, we will at first take into account another atmospheric consideration.

#### Rotating Reference Frame

So far, we used a fixed Cartesian coordinate system for describing the dynamics of the atmosphere and didn't consider that the total velocity of a particle consists of its relative velocity with respect to the surface of the Earth and its planetary angular velocity with which it rotates around the Earth's axis.

Physical phenomena are indeed independent of the choice of the coordinate system, but their description necessarily depends on the observer and hence the chosen reference frame. Since we live on the surface of the Earth and thus perceive and measure every velocity relative to the Earth's surface, it is natural to use a rotating coordinate system. Therefore, we choose in the following a Cartesian system which rotates in accordance with the Earth around its rotational axis $\boldsymbol{\Omega}$ with angular velocity $\|\boldsymbol{\Omega}\|$. So the velocity $\boldsymbol{u}$ is no longer an absolute velocity but a relative one in respect of the Earth's rotation.

Rotations imply changes of direction and thus accelerations. Therefore, a coordinate transformation to a moving system results in additional inertia force terms in the momentum equation, namely the Coriolis and the centrifugal force. For the derivation of these terms see [Ade08, Dut86].

#### Coriolis Force

The Coriolis force $-2\boldsymbol{\Omega} \times \rho \boldsymbol{u}$ is an inertia force in a rotating system which is only perceived by a co-moving observer. Force-free movements are always straight-lined, but in a rotating frame of reference, they appear curved for a co-moving observer. This curvature is accredited to the Coriolis force which acts perpendicular to the direction of motion and perpendicular to the rotational axis. Therefore, it has a horizontal as well as a vertical component which vanishes at the North and South Pole. So the Coriolis force deflects every movement in the atmosphere which is non-parallel to the Earth's axis.

Moreover, since the Earth rotates from west to east, the counterclockwise rotation causes a clockwise curvature of flow on the northern hemisphere and the clockwise rota-

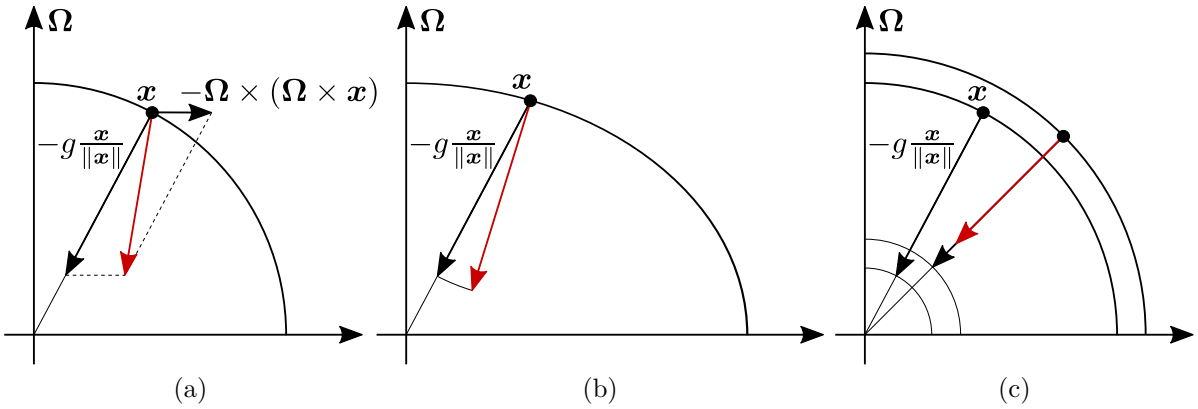



Figure 2.1.: Variation of gravity due to (a) the centrifugal force, (b) the shape and the inhomogeneity of mass of the Earth, and (c) the difference in height, i.e. the different distances of positions to the Earth's center.

tion a counterclockwise curvature on the southern hemisphere.

### Centrifugal Force

The centrifugal force $-\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \rho\boldsymbol{x})$, where $\boldsymbol{x}$ is a position vector, acts outwards and perpendicular to the rotational axis and varies with latitude. The force itself cannot be directly observed on Earth, instead we notice the resulting force consisting of gravitational and centrifugal force. In comparison, the centrifugal force is at least three orders of magnitude smaller than the gravitational force. Since the latter varies about $\pm 0.3\,\%$ anyway due to the shape of the Earth and the inhomogeneity of mass throughout the Earth and additionally about $0.3\,\%$ for an altitude difference of 10 km [Gil82], these effects are often neglected, compare Figure 2.1. Since computing capacities and measurements are not capable of representing the exact shape of the Earth and the modeling of the inhomogeneity of mass is very difficult as well, a spherical homogenous Earth has to be assumed. In this course, we also neglect the centrifugal force and assume that the gravitational force is always directed to the center of a spherical Earth.

So from now on, every velocity $\boldsymbol{u}$ means a relative velocity with respect to the Earth's rotation, and the momentum equation reads

$$(\rho\boldsymbol{u})_t + \nabla \cdot (\rho\boldsymbol{u} \circ \boldsymbol{u}) + 2\boldsymbol{\Omega} \times \rho\boldsymbol{u} + \nabla p = -\rho g\boldsymbol{k} - \nabla \cdot \boldsymbol{\tau}. \tag{2.4}$$

The first term is the temporal derivative, the second describes the convection, the third the Coriolis force, the fourth the pressure gradient force, the fifth the gravitational force and the sixth the molecular friction.

Note that the above coordinate transformation to a rotating system has no effect on scalar quantities, so every scalar equation remains the same, merely the velocity is

interpreted as relative to the Earth's rotation.

### 2.1.3. Conservation of Energy

The energy content $E$ of a control volume $\Omega_t$ can be expressed by the integral of the total energy $e$ per unit mass multiplied by the density $\rho$

$$E(t) = \int_{\Omega_t} \rho(\boldsymbol{x}, t) e(\boldsymbol{x}, t)\, d\boldsymbol{x}.$$

Note that $e$ is the specific total energy

$$e = e_{\text{kin}} + e_{\text{th}} = \frac{1}{2}\boldsymbol{u}^2 + c_v T \tag{2.5}$$

which consists of the sum of specific kinetic and thermal energy. Potential energy appears in form of gravity. Here, $c_v$ represents the specific heat capacity of dry air at constant volume.

Work done on a system changes its energy. Concretely, energy alters through, e.g., motion of particles, compression or expansion, shear forces, and heat conduction. Moreover, the energy variable $e$ can be exchanged for the temperature variable $T$ by relation (2.5). This leads to the temperature equation

$$c_v((\rho T)_t + \nabla \cdot (\rho \boldsymbol{u} T)) + p \nabla \cdot \boldsymbol{u} = \lambda \Delta T - \boldsymbol{\tau} : \nabla \boldsymbol{u} + Q. \tag{2.6}$$

For a detailed derivation see [Ade08, KV03]. Here, $\lambda$ is the thermal conductivity and $Q$ a source term consisting of effects of insolation. Furthermore, we use the notation of the Frobenius product

$$\boldsymbol{\tau} : \nabla \boldsymbol{u} := \nabla \cdot (\boldsymbol{\tau} \boldsymbol{u}) - \boldsymbol{u} \cdot (\nabla \cdot \boldsymbol{\tau}). \tag{2.7}$$

For all the atmosphere-dependent constants, see Appendix A.1.

### 2.1.4. Equation of State

Up to now, we derived a system of equations consisting of the conservations of mass (2.2), momentum (2.4), and temperature (2.6) with variables $\boldsymbol{u}$, $\rho$, $p$, and $T$. But this system is not closed since it has one more variable than equations to determine a unique solution. To close the system, an additional equation, i.e. a so-called equation of state, is necessary.

The dry atmosphere can be considered as an ideal gas mixture. Therefore, we assume the ideal gas law as equation of state which describes a functional relation between pressure, density, and temperature of an ideal gas or a gas mixture. It reads

$$p = \rho R_{\text{air}} T \tag{2.8}$$

with the gas constant $R_{\text{air}}$ for dry air. With this equation of state, the system is closed

and forms the dynamical core of atmospheric dynamics.

At this point, we summarize the dynamical core by repeating the equations of mass, momentum, temperature, and state

$$\begin{aligned} \rho_t + \nabla \cdot (\rho \boldsymbol{u}) &= 0 \\ (\rho \boldsymbol{u})_t + \nabla \cdot (\rho \boldsymbol{u} \circ \boldsymbol{u}) + 2\boldsymbol{\Omega} \times \rho \boldsymbol{u} + \nabla p &= -\rho g \boldsymbol{k} - \nabla \cdot \boldsymbol{\tau} \\ c_v((\rho T)_t + \nabla \cdot (\rho \boldsymbol{u} T)) + p \nabla \cdot \boldsymbol{u} &= \lambda \Delta T - \boldsymbol{\tau} : \nabla \boldsymbol{u} + Q \\ p &= \rho R_{\text{air}} T, \end{aligned} \tag{2.9}$$

which represent the compressible three-dimensional Navier-Stokes equations of fluid dynamics.

In this thesis, we consider the numerically most crucial dynamical core and thus the dynamics of dry air. Note however that the inclusion of humidity poses no problem since it only results in additional fairly simple equations with the constituents dry air, water vapor, liquid water, and frozen water as new variables [DSB11].

### 2.1.5. Boundary Conditions

The equations (2.9) are valid in the domain $\Omega$ representing the Earth's atmosphere. Thus, boundary conditions are necessary at the Earth's surface and at an artificial upper boundary in the stratosphere. We choose this second boundary as spherical shell at the height of about 24 km above the Earth's surface because this encloses already more than 90 % of the air and nearly the whole water vapor and thus the main weather influences. Note that we won't have to deal with artificial lateral boundaries since we always consider the global atmosphere.

Let $\nu$ be the outer normal direction and $\tau$ the tangential plane. For the density we specify a Neumann zero condition

$$\nabla_\nu \rho = 0 \tag{2.10}$$

at both boundaries to prohibit mass flow across the boundaries. For the velocity we choose slip conditions

$$\boldsymbol{u}_\nu = 0 \quad \text{and} \quad \nabla_\nu \boldsymbol{u}_\tau = 0 \tag{2.11}$$

and for the pressure a Neumann condition which balances the pressure gradient and the gravitational force

$$\nabla_\nu p = (-\rho g \boldsymbol{k})_\nu \tag{2.12}$$

at both boundaries. Furthermore, we attach the temperature with a Neumann zero condition at the upper boundary

$$\nabla_\nu T = 0, \tag{2.13}$$

whereas a Dirichlet condition depending on the insolation would be reasonable at the Earth's surface.

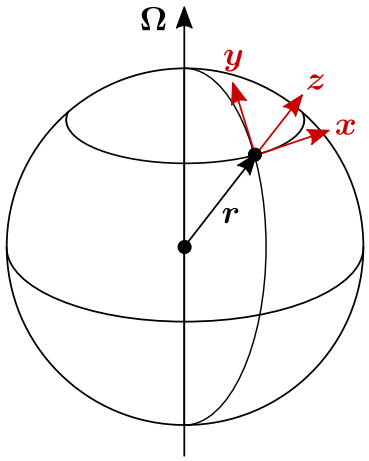



Figure 2.2.: Tangential Cartesian coordinate system with its origin at position vector $\boldsymbol{r}$, the $\boldsymbol{x}$-axis pointing towards the east, the $\boldsymbol{y}$-axis towards the north and the $\boldsymbol{z}$-axis radial away from the Earth's center.

## 2.2. Dimensional Analysis

Apart from the specially chosen equation of state, the system (2.9) consists of the compressible three-dimensional Navier-Stokes equations in a rotating frame of reference. Every variable therein is a physical quantity attached with its SI[1] unit. Now, we intend to eliminate the units and derive so-called dimensionless variables and equations. This representation allows us to compare the magnitude of each term and thus reveals its importance for our special atmospheric conditions. The approach is called dimensional analysis and is also known as "Π-theorem", see e.g. [Buc14, Bra57, Gör75, Bar96] and the references therein.

To account for anisotropic forces in a dimensional analysis, we introduce at first a tangential Cartesian coordinate system.

### 2.2.1. Tangential Cartesian Coordinates

Very anisotropic forces appear in the atmosphere like the gravitational, Coriolis, or pressure gradient force. To take these into account in a theoretical dimensional analysis, a splitting of equations and vectors in their horizontal and vertical components is reasonable. Hence, we apply a further coordinate transformation to the equation system (2.9).

The new coordinate system is chosen to be a local system with its origin at position vector $\boldsymbol{r}$, the $\boldsymbol{x}$-axis pointing towards the east, the $\boldsymbol{y}$-axis towards the north and the $\boldsymbol{z}$-axis radial away from the Earth's center. Therefore, it is still an orthogonal Cartesian

[1]Système international d'unités, abbreviated SI, international system of units consisting of the base units meter (m), kilogram (kg), second (s), ampère (A), kelvin (K), candela (cd) and mole (mol).

system, see Figure 2.2. An alternative to this tangential coordinate system would be a spherical system where each position vector is specified by its latitude and longitude, i.e. more precisely by its polar and azimuthal angle, and by its radial distance from the origin.

Let $_{\shortparallel}$ denote the component of a vector in the tangential plane $\boldsymbol{xy}$ and $_{\perp}$ the radial component in the direction of $\boldsymbol{z}$. Now, every vector as well as the nabla operator has to be written in the new coordinate system

$$\boldsymbol{u} = \boldsymbol{u}_{\shortparallel} + \boldsymbol{u}_{\perp}, \quad \nabla = \nabla_{\shortparallel} + \nabla_{\perp} \tag{2.14}$$

leading to the transformed equation system

$$\begin{aligned}
&\rho_t + (\nabla_{\shortparallel} + \nabla_{\perp}) \cdot (\rho(\boldsymbol{u}_{\shortparallel} + \boldsymbol{u}_{\perp})) = 0 \\
&(\rho \boldsymbol{u}_{\shortparallel})_t + \nabla_{\shortparallel} \cdot (\rho \boldsymbol{u}_{\shortparallel} \circ \boldsymbol{u}_{\shortparallel}) + \nabla_{\perp} \cdot (\rho \boldsymbol{u}_{\perp} \circ \boldsymbol{u}_{\shortparallel}) + 2(\boldsymbol{\Omega}_{\shortparallel} \times \rho \boldsymbol{u}_{\perp} + \boldsymbol{\Omega}_{\perp} \times \rho \boldsymbol{u}_{\shortparallel}) + \nabla_{\shortparallel} p \\
&\qquad = -(\nabla \cdot \boldsymbol{\tau})_{\shortparallel} \\
&(\rho \boldsymbol{u}_{\perp})_t + \nabla_{\shortparallel} \cdot (\rho \boldsymbol{u}_{\shortparallel} \circ \boldsymbol{u}_{\perp}) + \nabla_{\perp} \cdot (\rho \boldsymbol{u}_{\perp} \circ \boldsymbol{u}_{\perp}) + 2(\boldsymbol{\Omega}_{\shortparallel} \times \rho \boldsymbol{u}_{\shortparallel}) + \nabla_{\perp} p \\
&\qquad = -\rho g \boldsymbol{k} - (\nabla \cdot \boldsymbol{\tau})_{\perp} \\
&c_v((\rho T)_t + \nabla_{\shortparallel} \cdot (\rho \boldsymbol{u}_{\shortparallel} T) + \nabla_{\perp} \cdot (\rho \boldsymbol{u}_{\perp} T)) + p(\nabla_{\shortparallel} \cdot \boldsymbol{u}_{\shortparallel} + \nabla_{\perp} \cdot \boldsymbol{u}_{\perp}) \\
&\qquad = \lambda(\nabla_{\shortparallel} \cdot (\nabla_{\shortparallel} T) + \nabla_{\perp} \cdot (\nabla_{\perp} T)) - \boldsymbol{\tau} : \nabla \boldsymbol{u} + Q \\
&\qquad p = \rho R_{\text{air}} T.
\end{aligned} \tag{2.15}$$

Here, the second and third equations are the horizontal and vertical momentum equations. However, in (2.15) we still have to split the three terms connected with the viscous stress tensor $\boldsymbol{\tau}$. Since

$$\nabla \cdot \boldsymbol{\tau} = -\mu \left( \Delta \boldsymbol{u} + \frac{1}{3} \nabla (\nabla \cdot \boldsymbol{u}) \right), \tag{2.16}$$

these friction terms can we written as

$$(\nabla \cdot \boldsymbol{\tau})_{\shortparallel} = -\mu \Bigg( \underbrace{\left[ (\nabla_{\shortparallel} \cdot \nabla_{\shortparallel}) \boldsymbol{u}_{\shortparallel} + \frac{1}{3} \nabla_{\shortparallel} (\nabla_{\shortparallel} \cdot \boldsymbol{u}_{\shortparallel} + \nabla_{\perp} \cdot \boldsymbol{u}_{\perp}) \right]}_{=:\boldsymbol{r}_1} + \underbrace{\left[ (\nabla_{\perp} \cdot \nabla_{\perp}) \boldsymbol{u}_{\shortparallel} \right]}_{=:\boldsymbol{r}_2} \Bigg) \tag{2.17}$$

$$(\nabla \cdot \boldsymbol{\tau})_{\perp} = -\mu \Bigg( \underbrace{\left[ (\nabla_{\perp} \cdot \nabla_{\perp}) \boldsymbol{u}_{\perp} + \frac{1}{3} \nabla_{\perp} (\nabla_{\shortparallel} \cdot \boldsymbol{u}_{\shortparallel} + \nabla_{\perp} \cdot \boldsymbol{u}_{\perp}) \right]}_{=:\boldsymbol{r}_3} + \underbrace{\left[ (\nabla_{\shortparallel} \cdot \nabla_{\shortparallel}) \boldsymbol{u}_{\perp} \right]}_{=:\boldsymbol{r}_4} \Bigg). \tag{2.18}$$

For the splitting of the term $\boldsymbol{\tau} : \nabla \boldsymbol{u}$ in the temperature equation, a decomposition of $\boldsymbol{\tau}$

$$\begin{aligned}
\boldsymbol{\tau} = -\mu \Bigg( &\underbrace{\left[ \nabla_{\shortparallel} \boldsymbol{u}_{\shortparallel} + \nabla_{\perp} \boldsymbol{u}_{\perp} + (\nabla_{\shortparallel} \boldsymbol{u}_{\shortparallel} + \nabla_{\perp} \boldsymbol{u}_{\perp})^{\mathsf{T}} - \frac{2}{3} (\nabla_{\shortparallel} \cdot \boldsymbol{u}_{\shortparallel} + \nabla_{\perp} \cdot \boldsymbol{u}_{\perp}) \mathbb{1} \right]}_{=:\boldsymbol{\tau}_1} \\
&+ \underbrace{\left[ \nabla_{\shortparallel} \boldsymbol{u}_{\perp} + (\nabla_{\shortparallel} \boldsymbol{u}_{\perp})^{\mathsf{T}} \right]}_{=:\boldsymbol{\tau}_2} + \underbrace{\left[ \nabla_{\perp} \boldsymbol{u}_{\shortparallel} + (\nabla_{\perp} \boldsymbol{u}_{\shortparallel})^{\mathsf{T}} \right]}_{=:\boldsymbol{\tau}_3} \Bigg)
\end{aligned}$$

and of $\nabla \boldsymbol{u}$

$$\nabla \boldsymbol{u} = [\nabla_{\shortparallel} \boldsymbol{u}_{\shortparallel} + \nabla_{\perp} \boldsymbol{u}_{\perp}]_{=:\boldsymbol{\sigma}_1} + [\nabla_{\shortparallel} \boldsymbol{u}_{\perp}]_{=:\boldsymbol{\sigma}_2} + [\nabla_{\perp} \boldsymbol{u}_{\shortparallel}]_{=:\boldsymbol{\sigma}_3}$$

lead to

$$\begin{aligned} \boldsymbol{\tau} : \nabla \boldsymbol{u} = -\mu \Big( & [\boldsymbol{\tau}_1 : \boldsymbol{\sigma}_1 + \boldsymbol{\tau}_2 : \boldsymbol{\sigma}_3 + \boldsymbol{\tau}_3 : \boldsymbol{\sigma}_2]_{=:s_1} + [\boldsymbol{\tau}_1 : \boldsymbol{\sigma}_2 + \boldsymbol{\tau}_2 : \boldsymbol{\sigma}_1]_{=:s_2} \\ & + [\boldsymbol{\tau}_1 : \boldsymbol{\sigma}_3 + \boldsymbol{\tau}_3 : \boldsymbol{\sigma}_1]_{=:s_3} + [\boldsymbol{\tau}_2 : \boldsymbol{\sigma}_2]_{=:s_4} + [\boldsymbol{\tau}_3 : \boldsymbol{\sigma}_3]_{=:s_5} \Big). \end{aligned} \tag{2.19}$$

For a detailed derivation of the transformed equation system (2.15) see [Ade08].

## 2.2.2. Nondimensionalization

The equations are still assigned with units and thus dependent on the magnitude of each variable. To estimate and compare the order of magnitude of each term, a transition to relative quantities is necessary. This well-known principle is called nondimensionalization. The transition leads to dimensionless numbers, whose magnitudes are characteristic for the modeled phenomenon and which allow a direct comparison of the sizes of each term.

In general, we get a dimensionless quantity $\xi^*$ by dividing a dimensionful quantity $\xi$ by a reference value $\xi_{\text{ref}}$. This reference value is a known characteristic constant depending on the considered problem. Hence, we substitute in our equations each dimensionful variable $\xi$ by

$$\xi = \xi_{\text{ref}} \xi^* \tag{2.20}$$

and the derivatives in time and space by

$$\frac{\partial}{\partial t} = \frac{1}{t_{\text{ref}}} \frac{\partial}{\partial t^*}, \quad \nabla_{\shortparallel} = \frac{1}{l^{\shortparallel}_{\text{ref}}} \nabla^*_{\shortparallel}, \quad \nabla_{\perp} = \frac{1}{l^{\perp}_{\text{ref}}} \nabla^*_{\perp} \tag{2.21}$$

with the characteristic time scale $t_{\text{ref}}$ and the horizontal and vertical length reference values $l^{\shortparallel}_{\text{ref}}$ and $l^{\perp}_{\text{ref}}$. Furthermore, we postulate by reason of consistence

$$t_{\text{ref}} = \frac{l^{\shortparallel}_{\text{ref}}}{u^{\shortparallel}_{\text{ref}}} = \frac{l^{\perp}_{\text{ref}}}{u^{\perp}_{\text{ref}}}. \tag{2.22}$$

These substitutions result in equations written with dimensionless variables and derivatives instead of the dimensionful ones but with additional groups of constant reference values. Appropriately combining these reference values leads to dimensionless characteristic numbers, whose magnitudes represent the importance of each term they belong to.

The equation system can now be written in its dimensionless form. For ease of read-

ability, we omit the label $^*$ for all dimensionless variables and derivatives

$$Sr\rho_t + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{2.23}$$

$$\begin{aligned} Sr(\rho \boldsymbol{u}_{\shortparallel})_t + \nabla_{\shortparallel} \cdot (\rho \boldsymbol{u}_{\shortparallel} \circ \boldsymbol{u}_{\shortparallel}) + \nabla_{\perp} \cdot (\rho \boldsymbol{u}_{\perp} \circ \boldsymbol{u}_{\shortparallel}) + \frac{1}{Ro_1}(\boldsymbol{\Omega}_{\shortparallel} \times \rho \boldsymbol{u}_{\perp}) \\ + \frac{1}{Ro_2}(\boldsymbol{\Omega}_{\perp} \times \rho \boldsymbol{u}_{\shortparallel}) + \frac{1}{M_1^2}\nabla_{\shortparallel} p = \frac{1}{Re_1}\boldsymbol{r}_1 + \frac{1}{Re_2}\boldsymbol{r}_2 \end{aligned} \tag{2.24}$$

$$\begin{aligned} Sr(\rho \boldsymbol{u}_{\perp})_t + \nabla_{\shortparallel} \cdot (\rho \boldsymbol{u}_{\shortparallel} \circ \boldsymbol{u}_{\perp}) + \nabla_{\perp} \cdot (\rho \boldsymbol{u}_{\perp} \circ \boldsymbol{u}_{\perp}) + \frac{1}{Ro_3}(\boldsymbol{\Omega}_{\shortparallel} \times \rho \boldsymbol{u}_{\shortparallel}) \\ + \frac{1}{M_2^2}\nabla_{\perp} p = -\frac{1}{Fr^2}\rho \boldsymbol{k} + \frac{1}{Re_2}\boldsymbol{r}_3 + \frac{1}{Re_1}\boldsymbol{r}_4 \end{aligned} \tag{2.25}$$

$$\begin{aligned} &Sr(\rho T)_t + \nabla \cdot (\rho \boldsymbol{u} T) + (\gamma - 1) p \nabla \cdot \boldsymbol{u} \\ &\quad = \gamma \frac{1}{Pr}\left(\frac{1}{Re_1}\Delta_{\shortparallel} T + \frac{1}{Re_2}\Delta_{\perp} T\right) + (\gamma - 1)\left(\frac{M_1^2}{Re_1}s_1 + \frac{M_2^2}{Re_3}s_2 + \frac{M_1^2}{Re_3}s_3\right. \\ &\qquad \left. + \frac{M_2^2}{Re_1}s_4 + \frac{M_1^2}{Re_2}s_5\right) + (\gamma - 1)\, DaQ \end{aligned} \tag{2.26}$$

$$p = \rho T. \tag{2.27}$$

The dimensionless numbers are defined as follows [KV03]. The *Strouhal number* defines the ratio of advection time to reference time

$$Sr := \frac{l_{\text{ref}}/u_{\text{ref}}}{t_{\text{ref}}}, \tag{2.28}$$

the *Rossby number* the ratio of Earth's rotation time to advection time

$$Ro := \frac{u_{\text{ref}}}{2\|\boldsymbol{\Omega}\|_{\text{ref}} l_{\text{ref}}} = \frac{1}{2\|\boldsymbol{\Omega}\|_{\text{ref}}}\left(\frac{l_{\text{ref}}}{u_{\text{ref}}}\right)^{-1} \tag{2.29}$$

with

$$Ro_1 := Ro(u_{\text{ref}}^{\shortparallel}, l_{\text{ref}}^{\perp}), \quad Ro_2 := Ro(u_{\text{ref}}^{\shortparallel}, l_{\text{ref}}^{\shortparallel}), \quad Ro_3 := Ro(u_{\text{ref}}^{\perp}, l_{\text{ref}}^{\shortparallel}), \tag{2.30}$$

the *Mach number* the ratio of flow speed to sonic speed

$$M := \frac{u_{\text{ref}}}{\sqrt{p_{\text{ref}}/\rho_{\text{ref}}}} \tag{2.31}$$

with

$$M_1 := M(u_{\text{ref}}^{\shortparallel}), \quad M_2 := M(u_{\text{ref}}^{\perp}), \tag{2.32}$$

the *Reynolds number* represents the ratio of inertial to frictional forces

$$Re := \frac{\rho_{\text{ref}} u_{\text{ref}} l_{\text{ref}}}{\mu} \tag{2.33}$$

with

$$Re_1 := Re(u_{\text{ref}}^{\shortparallel}, l_{\text{ref}}^{\shortparallel}), \quad Re_2 := Re(u_{\text{ref}}^{\perp}, l_{\text{ref}}^{\perp}), \quad Re_3 := Re(u_{\text{ref}}^{\perp}, l_{\text{ref}}^{\shortparallel}), \tag{2.34}$$

the *Froude number* the ratio of flow speed to gravity wave speed

$$Fr := \frac{u_{\text{ref}}}{\sqrt{g l_{\text{ref}}}}, \tag{2.35}$$

the *Prandtl number* the ratio of kinematic viscosity to thermal conductivity

$$Pr := \frac{c_p \mu}{\lambda} = \frac{\gamma R_{\text{air}} \mu}{(\gamma - 1)\lambda}, \tag{2.36}$$

and finally the *Damköhler number* is defined as

$$Da := \frac{l_{\text{ref}} Q_{\text{ref}}}{u_{\text{ref}} p_{\text{ref}}} \tag{2.37}$$

and

$$\gamma := \frac{c_p}{c_v} \tag{2.38}$$

with $R_{\text{air}} = c_p - c_v$ and $c_p$ being the specific heat capacity of dry air at constant pressure. The indices of the Rossby, Mach, and Reynolds numbers in the equation system represent the horizontal or vertical reference values of the velocity or the length according to the above definitions.

### 2.2.3. Scale Analysis

Now, for the comparison of each term, we need to choose specific reference values. The Earth's atmosphere contains a vast spectrum of phenomena on different scales ranging from small turbulences to flow between low and high pressure areas and even to the global circulation of the atmosphere. In Table 2.1 the characteristic horizontal, vertical and temporal magnitudes of motions at different scales are listed.

To mirror this wide range of scales and to estimate the size range of each term we choose two sets of reference values. The first is a set of small-scale values at ground level and the second a set of large-scale values belonging to motions in the lower stratosphere.

| scale | $l^{\shortparallel}_{\text{ref}}$ in meters | $l^{\perp}_{\text{ref}}$ in meters | $t_{\text{ref}}$ in seconds |
|---|---|---|---|
| small | $\leq 10^2$ | $\leq 10^2$ | $\leq 10^2$ |
| convective | $10^3 - 10^4$ | $10^3 - 10^4$ | $10^2 - 10^3$ |
| meso | $10^5 - 10^6$ | $10^4$ | $10^4 - 10^5$ |
| large | $> 10^6$ | $10^4$ | $10^6$ |

Table 2.1.: Characteristic horizontal, vertical, and temporal magnitudes of motions at different atmospheric scales [Pic97].

Small-scale values at ground level:

$$\begin{aligned} l^{\shortparallel}_{\text{ref}} \sim 10^2\,\text{m}, \quad l^{\perp}_{\text{ref}} \sim 10^2\,\text{m}, \quad t_{\text{ref}} \sim 10^3\,\text{s}, \\ u^{\shortparallel}_{\text{ref}} \sim 10^{-1}\,\text{m}\,\text{s}^{-1}, \quad u^{\perp}_{\text{ref}} \sim 10^{-1}\,\text{m}\,\text{s}^{-1}, \\ \rho_{\text{ref}} \sim 10^0\,\text{kg}\,\text{m}^{-3}, \quad p_{\text{ref}} \sim 10^5\,\text{kg}\,\text{m}^{-1}\text{s}^{-2}. \end{aligned} \tag{2.39}$$

Large-scale values of the lower stratosphere:

$$\begin{aligned} l^{\shortparallel}_{\text{ref}} \sim 10^7\,\text{m}, \quad l^{\perp}_{\text{ref}} \sim 10^4\,\text{m}, \quad t_{\text{ref}} \sim 10^5\,\text{s}, \\ u^{\shortparallel}_{\text{ref}} \sim 10^2\,\text{m}\,\text{s}^{-1}, \quad u^{\perp}_{\text{ref}} \sim 10^{-1}\,\text{m}\,\text{s}^{-1}, \\ \rho_{\text{ref}} \sim 10^{-1}\,\text{kg}\,\text{m}^{-3}, \quad p_{\text{ref}} \sim 10^4\,\text{kg}\,\text{m}^{-1}\text{s}^{-2}. \end{aligned} \tag{2.40}$$

Here, the characteristic length scales are taken from Table 2.1. The small-scale values of $u^{\shortparallel}_{\text{ref}}$ and $u^{\perp}_{\text{ref}}$ represent calm air which is usually defined for velocities below $0.5\,\text{m/s}$. The large-scale values of $u^{\shortparallel}_{\text{ref}}$ and $u^{\perp}_{\text{ref}}$ are typical for jet streams in the upper troposphere and lower stratosphere where horizontal velocities up to $180\,\text{m/s}$ were measured. The time scales are chosen such that the postulation (2.22) is fulfilled.

Furthermore, pressure and density decrease with height by one order of magnitude, whereas the temperature sinks averagely from $288.15\,\text{K}$ ($15°\text{C}$) to $218.15\,\text{K}$ ($-55°\text{C}$). So there is no variation in the range of an order of magnitude and therefore, the reference temperature can be chosen for both reference sets as

$$T_{\text{ref}} = \frac{p_{\text{ref}}}{R_{\text{air}}\rho_{\text{ref}}} \sim 10^2\,\text{K}. \tag{2.41}$$

Likewise, the following values can be regarded as constant with height

$$\begin{aligned} \|\boldsymbol{\Omega}\|_{\text{ref}} \sim 10^{-4}\,\text{s}^{-1}, \quad g \sim 10\,\text{m}\,\text{s}^{-2}, \quad \mu \sim 10^{-5}\,\text{kg}\,\text{m}^{-1}\text{s}^{-1}, \\ \lambda \sim 10^{-2}\,\text{W}\,\text{m}^{-1}\text{K}^{-1}, \quad c_p \sim 10^3\,\text{J}\,\text{kg}^{-1}\text{K}^{-1}, \quad \gamma \sim 1.4. \end{aligned} \tag{2.42}$$

| Horizontal momentum equation | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Time | Non-linear | Coriolis | | Pressure | Friction | |
| | $Sr$ | 1 | $Ro_1^{-1}$ | $Ro_2^{-1}$ | $M_1^{-2}$ | $Re_1^{-1}$ | $Re_2^{-1}$ |
| small-scale, ground level | $10^0$ | $10^0$ | $10^{-1}$ | $10^{-1}$ | $10^7$ | $10^{-6}$ | $10^{-6}$ |
| large-scale, stratosphere | $10^0$ | $10^0$ | $10^{-2}$ | $10^1$ | $10^1$ | $10^{-13}$ | $10^{-7}$ |

| Vertical momentum equation | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Time | Non-linear | Coriolis | Gravitation | Pressure | Friction | |
| | $Sr$ | 1 | $Ro_3^{-1}$ | $Fr^{-2}$ | $M_2^{-2}$ | $Re_1^{-1}$ | $Re_2^{-1}$ |
| small-scale, ground level | $10^0$ | $10^0$ | $10^{-1}$ | $10^5$ | $10^7$ | $10^{-6}$ | $10^{-6}$ |
| large-scale, stratosphere | $10^0$ | $10^0$ | $10^4$ | $10^7$ | $10^7$ | $10^{-13}$ | $10^{-7}$ |

| Temperature equation | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Time | Non-lin. | Press. | Energy flux | | Friction | | | | |
| | $Sr$ | 1 | $\gamma - 1$ | $\gamma Pr^{-1} Re_1^{-1}$ | $\gamma Pr^{-1} Re_2^{-1}$ | $(\gamma-1)\cdot M_1^2 Re_1^{-1}$ | $(\gamma-1)\cdot M_2^2 Re_3^{-1}$ | $(\gamma-1)\cdot M_1^2 Re_3^{-1}$ | $(\gamma-1)\cdot M_2^2 Re_1^{-1}$ | $(\gamma-1)\cdot M_1^2 Re_2^{-1}$ |
| small | $10^0$ | $10^0$ | $10^{-1}$ | $10^{-6}$ | $10^{-6}$ | $10^{-14}$ | $10^{-14}$ | $10^{-14}$ | $10^{-14}$ | $10^{-14}$ |
| large | $10^0$ | $10^0$ | $10^{-1}$ | $10^{-13}$ | $10^{-7}$ | $10^{-15}$ | $10^{-18}$ | $10^{-12}$ | $10^{-21}$ | $10^{-9}$ |

Table 2.2.: Resulting magnitudes of the dimensionless numbers in the horizontal and vertical momentum equations (2.24) and (2.25) and the temperature equation (2.26) for the two reference sets (2.39) and (2.40).

Substituting all of these reference values into the equations, the dimensionless numbers and thus the magnitudes of each term result in the values listed in Table 2.2.

For both momentum equations, it is noteworthy that the terms describing the molecular friction are at least five orders of magnitude smaller than any other term. Therefore, it is justified to neglect the friction terms of the momentum equations which changes them to hyperbolic Euler equations.

Likewise, the friction terms in the temperature equation are vanishingly small, and even the energy flux terms with a difference of at least five orders of magnitude could be neglected. Indeed, the molecular friction is only in the laminar atmospheric boundary layer of importance [Fok03]. The phase diagram in Figure 2.3 illustrates this fact, namely that the friction terms become merely important at a length scale of centimeters.

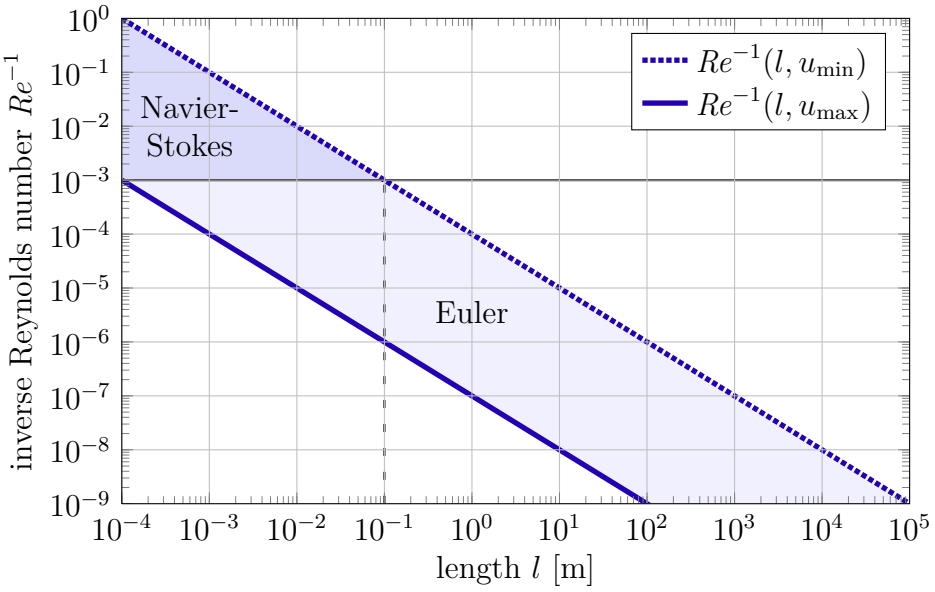



Figure 2.3.: Phase diagram for the inverse Reynolds number with $\rho_{\text{ref}} = 10^0\,\text{kg/m}^3$, $u_{\min} = 10^{-1}\,\text{m/s}$, and $u_{\max} = 10^2\,\text{m/s}$ in double logarithmic representation.

Neglecting the aforementioned terms, the equation system can now be written as

$$\begin{aligned}
Sr\rho_t + \nabla\cdot(\rho\boldsymbol{u}) &= 0\\
Sr(\rho\boldsymbol{u}_{\shortparallel})_t + \nabla_{\shortparallel}\cdot(\rho\boldsymbol{u}_{\shortparallel}\circ\boldsymbol{u}_{\shortparallel}) + \nabla_{\perp}\cdot(\rho\boldsymbol{u}_{\perp}\circ\boldsymbol{u}_{\shortparallel})&\\
+\frac{1}{Ro_1}(\boldsymbol{\Omega}_{\shortparallel}\times\rho\boldsymbol{u}_{\perp}) + \frac{1}{Ro_2}(\boldsymbol{\Omega}_{\perp}\times\rho\boldsymbol{u}_{\shortparallel}) + \frac{1}{M_1^2}\nabla_{\shortparallel}p &= 0\\
Sr(\rho\boldsymbol{u}_{\perp})_t + \nabla_{\shortparallel}\cdot(\rho\boldsymbol{u}_{\shortparallel}\circ\boldsymbol{u}_{\perp}) + \nabla_{\perp}\cdot(\rho\boldsymbol{u}_{\perp}\circ\boldsymbol{u}_{\perp})&\\
+\frac{1}{Ro_3}(\boldsymbol{\Omega}_{\shortparallel}\times\rho\boldsymbol{u}_{\shortparallel}) + \frac{1}{M_2^2}\nabla_{\perp}p &= -\frac{1}{Fr^2}\rho\boldsymbol{k}\\
Sr(\rho T)_t + \nabla\cdot(\rho\boldsymbol{u}T) + (\gamma-1)p\nabla\cdot\boldsymbol{u} &= (\gamma-1)DaQ\\
p &= \rho T
\end{aligned} \tag{2.43}$$

or after inverse transformation of the horizontal and vertical momentum equations into Cartesian coordinates as

$$\begin{aligned}
Sr\rho_t + \nabla\cdot(\rho\boldsymbol{u}) &= 0\\
Sr(\rho\boldsymbol{u})_t + \nabla\cdot(\rho\boldsymbol{u}\circ\boldsymbol{u}) + \frac{1}{Ro}\boldsymbol{\Omega}\times\rho\boldsymbol{u} + \frac{1}{M^2}\nabla p &= -\frac{1}{Fr^2}\rho\boldsymbol{k}\\
Sr(\rho T)_t + \nabla\cdot(\rho\boldsymbol{u}T) + (\gamma-1)p\nabla\cdot\boldsymbol{u} &= (\gamma-1)DaQ\\
p &= \rho T
\end{aligned} \tag{2.44}$$

or, for comparison with (2.9), in its dimensionful form as

$$\begin{aligned} \rho_t + \nabla \cdot (\rho \boldsymbol{u}) &= 0 \\ (\rho \boldsymbol{u})_t + \nabla \cdot (\rho \boldsymbol{u} \circ \boldsymbol{u}) + 2\boldsymbol{\Omega} \times \rho \boldsymbol{u} + \nabla p &= -\rho g \boldsymbol{k} \\ c_v((\rho T)_t + \nabla \cdot (\rho \boldsymbol{u} T)) + p \nabla \cdot \boldsymbol{u} &= Q \\ p &= \rho R_{\text{air}} T. \end{aligned} \tag{2.45}$$

Let us finally give a remark on low compressibility. Flows are usually characterized as low compressible if the Mach number, i.e. the quotient of flow speed and sonic speed, is in the whole domain smaller than 0.2. There is a vast amount of literature about flows with such low Mach numbers, also concerning atmospheric flows. Checking the Mach number for our setting in the troposphere and lower stratosphere, we immediately observe that the sonic speed decreases depending on the temperature – and hence the height – from 340 m/s at 288.15 K to 296 m/s at 218.15 K. Moreover, maximal wind speeds of 100 m/s at ground level and of 180 m/s for jet streams in the lower stratosphere can occur. The corresponding Mach numbers for these extreme cases are 0.3 and 0.6, respectively. Therefore, our problem is in general not low compressible, although for most of the domain rather small Mach numbers can be observed.

## 2.3. Multiscale Modeling

Our derived Euler equations can be regarded as full compressible inviscid flow equations in a rotating reference frame which are able to describe all atmospheric motions interacting on any time and length scales.

For the sake of completeness, we want to mention that various further reduced meteorological models exist in the literature, e.g. primitive hydrostatic equations, shallow water equations, or quasi-geostrophic equations to name the most prominent. All of these mostly independently developed models have been recently connected via a comprehensive multiscale ansatz and $\varepsilon$-analysis by Rupert Klein [Kle04, Kle08, Kle10, KVPR11, KV03].

In the following sections, we both wish to describe this popular ansatz and add a new numerical point of view to the development of reduced models.

### 2.3.1. Unified Approach to Reduced Meteorological Models

Atmospheric motions are always an interplay of various phenomena on diverse length and time scales. To separate single phenomena, many reduced models exist which are only valid on special scales. Historically, the development of such simplified models was also computationally necessary since former computer capacities were not able to solve the full compressible fluid equations with reasonable time and memory resources. But

| Coordinate scalings | Resulting model |
|---|---|
| $\mathcal{U}^{(i)}\left(\boldsymbol{x}_{\shortparallel}, \frac{z}{\varepsilon}, \frac{t}{\varepsilon}\right)$ | Linear small scale internal gravity waves |
| $\mathcal{U}^{(i)}\left(\boldsymbol{x}_{\shortparallel}, z, t\right)$ | Anelastic and pseudo-incompressible flows |
| $\mathcal{U}^{(i)}\left(\varepsilon^2\boldsymbol{x}_{\shortparallel}, z, \varepsilon t\right)$ | Gravity waves induced by Coriolis effects |
| $\mathcal{U}^{(i)}\left(\varepsilon^2\boldsymbol{x}_{\shortparallel}, z, \varepsilon^2 t\right)$ | Mid-latitude quasi-geostrophic flow |

Table 2.3.: Examples for coordinate scalings and their corresponding classical models according to [Kle04]. For denotation see (2.46) and (2.47).

nowadays, the trend is inverted by using as general as possible equations with preferably few assumptions – a trend we also pursued in Section 2.1 to 2.2.

Reduced models were mostly derived by physical argumentation in order to describe flows on special scales, for example flows on a large climate scale, a synoptic scale, or a small scale, compare also Table 2.1. But a mathematical framework for the interconnection of the diversity of existing models to the full compressible flow equations was missing for decades. In 2004, such a unified approach was developed in [Kle04] via multiscale asymptotics. In this way, the majority of reduced meteorological models can be derived as special asymptotic limit of the three-dimensional Euler equations in a rotating reference frame, see also [Kle08, Kle10].

Before describing this particular approach, let us note that the ansatz is based on the assumption that a natural scale separation exists for atmospheric flows of sufficiently large scale [Kle10] – a proposition which is controversially disputed, see e.g. [LTHS08]. Nevertheless, the union of physical observation and mathematical consistency is an important step for a better understanding and cooperation of meteorologists and applied mathematicians.

Multiscale asymptotics or so-called $\varepsilon$-analysis is a systematic way to gain the majority of the known simplified model equations of theoretical meteorology. In the following, we briefly describe the idea of the principle and otherwise refer to [KVPR11] for details.

As a first step, the dimensionless horizontal, vertical, and time coordinates $(\boldsymbol{x}_{\shortparallel}, z, t)$ with $z := \boldsymbol{x}_{\perp}$ are replaced by new scaled coordinates $(\boldsymbol{\xi}_{\shortparallel}, \zeta, \tau)$. The scaling is realized through an asymptotic expansion parameter $\varepsilon \ll 1$ whose power relating to each coordinate controls the resulting model. So the choice of scaling is essential. As an example, for quasi-geostrophic flow the scaling reads

$$\boldsymbol{\xi}_{\shortparallel} = \varepsilon^2 \boldsymbol{x}_{\shortparallel}, \quad \zeta = z, \quad \text{and} \quad \tau = \varepsilon^2 t. \tag{2.46}$$

Furthermore, each variable such as velocity, density, pressure, or temperature is asymp-

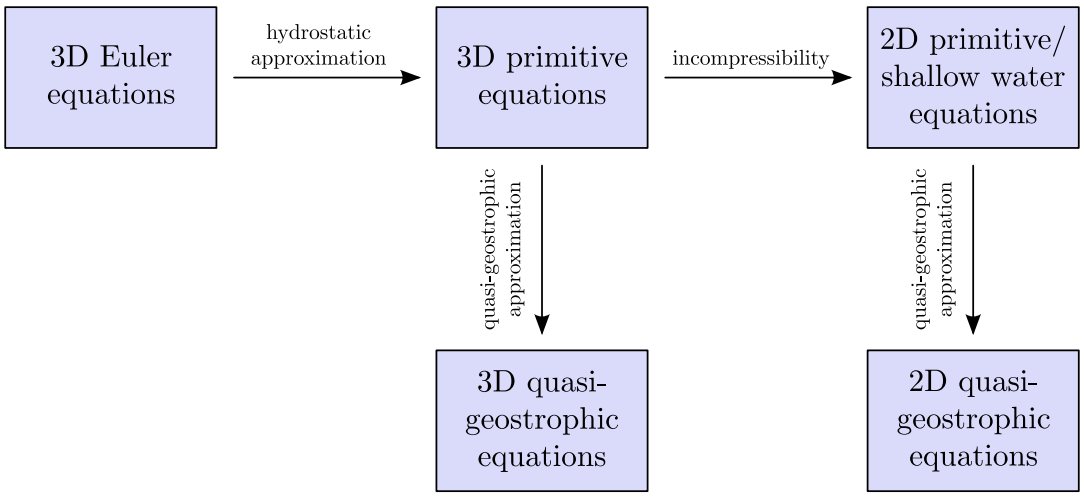



Figure 2.4.: Hierarchy of some reduced atmospheric models and their approximation steps with respect to the full three-dimensional Euler equations.

totically expanded by powers of $\varepsilon$

$$\mathcal{U}(\boldsymbol{x}_{\shortparallel}, z, t, \varepsilon) = \sum_i \varepsilon^i \mathcal{U}^{(i)}\left(\varepsilon^2 \boldsymbol{x}_{\shortparallel}, z, \varepsilon^2 t\right) \quad \text{with} \quad \mathcal{U} = (\boldsymbol{u}, \rho, p, T)^{\mathsf{T}}. \tag{2.47}$$

Next, the new variables and their derivatives are substituted into the full three-dimensional Euler equations resulting in a system which consists of equations with terms of different powers of $\varepsilon$. Since the equations shall be valid for arbitrarily small $\varepsilon$, too, each power of $\varepsilon$ has to vanish and thus satisfy an equation of its own. This cascade of equations finally leads to the reduced atmospheric model. For the example above with scaling (2.46), the (dimensionful) hydrostatic balance

$$\frac{\partial p}{\partial z} = -\rho g, \tag{2.48}$$

i.e. the balance of vertical pressure gradient force and gravitational force, is one of its results.

In this way, a multitude of different models can be derived, all depending on a special choice of scaling. In Table 2.3, exemplary scalings are listed and a comprehensive description of various reduced models can be found in [Kle10].

For some well-known simplified models, the corresponding physical assumptions and their dependencies are illustrated in Figure 2.4. Here, the main approximations are the above mentioned hydrostatic balance, the geostrophic balance which implies an equilibrium of horizontal Coriolis and pressure gradient forces, and the assumption of incompressibility which postulates divergence freedom of velocity and induces, together with the hydrostatic approximation, two-dimensional equations, also known as shallow water equations, see e.g. [Hen05] and the references therein.

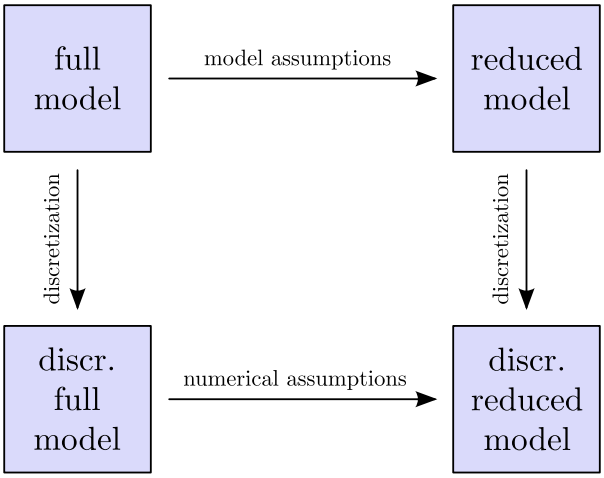



Figure 2.5.: Commutative diagram of the pathways from a full continuous model to a reduced discretized model.

### 2.3.2. Numerical Point of View

In the previous section, we described an asymptotic multiscale approach to derive simplified atmospheric models in a consistent mathematical way, which unifies existing models derived by physical observations of special phenomena. But this classical way of modeling is not the only one. We can also turn the ansatz upside down by taking on a numerical point of view.

The governing equations are continuous in their full as well as in their reduced form. The simplifying assumptions leading to the reduced models are continuous approximations, too, see the examples in Figure 2.4, and attached with a modeling error depending on $\varepsilon$. In the following process of discretizing the computational domain and the equations, a further error appears, this time depending on the mesh width $h$. Now, it is essential that the errors in $\varepsilon$ and $h$ are coherently balanced, i.e. that they have the same magnitude, because it would be pointless to resolve the domain finer and thus have a smaller discretization error than the modeling error already made and vice versa.

The idea is to change the order of simplifying the governing equations and discretizing them. Let us suppose the full Euler equations are first of all discretized implying an error depending on $h$. Then by numerical or geometrical approximations, the equation set is further simplified. In this way, the model assumptions can be interpreted as numerical assumptions on a full discretized model leading to the commutative diagram displayed in Figure 2.5. To our knowledge, this idea is pursued nowhere else in the classical literature.

The advantage of such "numerical modeling" is the implied consistency that both the discretization error and the modeling error depend on the mesh width $h$. So we only have to control the error in $h$, e.g. by refining the mesh width, and both the discretization and the modeling error decrease.

Let us illustrate the idea with an example, the so-called shallow water equations. Here,

we describe only the principle proceeding and do not work out the derivation in detail since it would be very technical and lengthy.

Starting with the full Euler equations in their Cartesian form, we first transform them to spherical or tangential coordinates in order to have a vertical coordinate direction. Then, we discretize them spatially, e.g. by Finite Volumes, Finite Elements, or Finite Differences, on a grid consisting of vertical atmospheric layers. For details on the generation of the grid, see the following Sections 3 and 4. Next, we assume that only one atmospheric layer exists, i.e. that we have only one degree of freedom in vertical direction accompanied by fixed boundary conditions. Thus, the vertical velocity vanishes which reduces the vertical momentum equation to the hydrostatic balance, which is one of the model assumptions needed for the shallow water equations, compare Figure 2.4. So in this way, we gained the hydrostatic model assumption by pure numerical construction. Analogously, a numerical approximation can be found to be interpreted as incompressibility assumption, so that the same discretized shallow water equations are finally derived.

At this point, we only state the commutativity of the diagram – i.e. that any physical model assumption can be interpreted as special numerical assumption leading to the same reduced discretized model – without a proof since it would shift the focus of this thesis. Nevertheless, we propose to study the connection in more detail since its inherent consistency has the potential of simplifying the error analysis considerably.

But as already stated, reduced models in atmospheric dynamics are increasingly being abandoned in favor of the full Euler equations. This trend is due to the ever increasing computing capacities which are available nowadays. The Euler equations describe any atmospheric flow on any scale, so they form the most general model possible. However, the quality of the solution, i.e. the resolved phenomena, depends heavily on the discretization. If the Euler equations are discretized with mesh width $h$, flow on a smaller scale than $h$ is no longer resolved. To compensate for it, turbulence modeling has to be applied, on which we concentrate in the following Section 2.4.

## 2.4. Turbulence

Due to the roughness of the Earth's surface, atmospheric flow is turbulent in air layers near the ground. However, the turbulences rapidly decrease with height, so that atmospheric flow can be regarded as generally laminar above a height of 1,000 – 1,500 m. This turbulent region up to 1,500 m above ground is called boundary layer of the atmosphere.

In principle, every kind of flow, whether it's laminar or turbulent, is being represented with our equation system (2.44). Therefore, the most obvious approach for the computation of atmospheric dynamics would be the choice of a sufficiently fine mesh, which resolves even the smallest eddies, and the solution of the equation system on this mesh. Such an approach is called direct numerical simulation (DNS). However, DNS is not practicable for the computation of atmospheric flow, since it would require a spatial res-

olution of the grid of about 1 cm and a temporal resolution of few seconds [Pic97], which by far exceeds the capacity of today's computers. Thus, we are forced to appropriately model the turbulence in the atmospheric boundary layer.

Turbulent flow involves heavy spatial and temporal fluctuations of the velocity field which can be captured neither by the numerical grid nor by the meteorological network. However, we are rather interested in mean values representing the large-scale trend of the flow than in each microscopically small eddy. Therefore, we aim at filtering irregular turbulent motions out and at appropriately modeling the effects of the fluctuations not resolved.

Since the Euler equations are non-linear, the average of a product differs from the product of the average values. Thus, a filtering changes the original equations, so that they are no longer universally valid. Afterwards, they hold only for the smoothed values, i.e. on scales which are larger than the averaging interval.

In the following, two widely-used classes of turbulence models, Reynolds-Averaged Navier-Stokes and Large Eddy Simulation, are described and compared, showing that the latter has to be preferred for atmospheric dynamics.

### 2.4.1. Reynolds-Averaged Navier-Stokes

#### Temporal Filtering

The idea of Reynolds-Averaged Navier-Stokes (RANS) is the filtering of each equation with a temporal averaging operator. This implies a decomposition of every variable $\psi$ in its large-scale part $\overline{\psi}$ and its fluctuation $\psi'$ such that

$$\psi = \overline{\psi} + \psi' \tag{2.49}$$

with the large-scale part being a temporal average over the time interval of turbulence $\Delta t$

$$\overline{\psi}(\boldsymbol{x}, t) := \frac{1}{\Delta t} \int_{t-\Delta t/2}^{t+\Delta t/2} \psi(\boldsymbol{x}, s)\, ds. \tag{2.50}$$

If $\psi$ is a vector field, the operator has to be applied by components.

We further use a density-weighted average, a so-called Favre filter, for velocity and temperature. These variables $\zeta$ are analogously split in their weighted average $\widehat{\zeta}$ and the corresponding fluctuation $\zeta''$

$$\zeta = \widehat{\zeta} + \zeta'' \tag{2.51}$$

with the density-weighted average defined by

$$\widehat{\zeta} := \frac{\overline{\rho\zeta}}{\overline{\rho}}. \tag{2.52}$$

The filter (2.50) which is used for $\rho$ and $p$ and the weighted filter (2.52) used for $\boldsymbol{u}$ and $T$ are both linear operators. Thus, for two variables $\psi_1$ and $\psi_2$ and $\alpha \in \mathbb{R}$ holds

$$\overline{\psi_1 + \psi_2} = \overline{\psi}_1 + \overline{\psi}_2 \quad \text{and} \quad \overline{\alpha\psi_1} = \alpha\overline{\psi}_1. \tag{2.53}$$

Furthermore, the derivatives in time and space are commutable with the filter operator

$$\overline{\frac{\partial}{\partial t}\psi} = \frac{\partial}{\partial t}\overline{\psi} \quad \text{and} \quad \overline{\frac{\partial}{\partial x_i}\psi} = \frac{\partial}{\partial x_i}\overline{\psi} \quad \text{for} \;\; i = 1, 2, 3, \tag{2.54}$$

and we have

$$\overline{\psi_1\overline{\psi}_2} = \overline{\psi}_1\overline{\psi}_2 \tag{2.55}$$

since $\overline{\psi}_2$ acts as a constant with respect to the outer integral. The property (2.54) is only valid, if $\Delta t$ is fixed with no variation in time [GT99, Pie02], compare the corresponding remark in Section 2.4.2. Since these three characteristics also hold for the weighted average (2.52), we can directly conclude

$$\overline{\widehat{\zeta}\psi} = \widehat{\zeta}\overline{\psi}. \tag{2.56}$$

Using these properties, we apply the filter operator (2.50) to our equation system (2.45) which leads to

$$\begin{aligned} \overline{\rho}_t + \nabla \cdot (\overline{\rho}\widehat{\boldsymbol{u}}) &= 0 \\ (\overline{\rho}\widehat{\boldsymbol{u}})_t + \nabla \cdot (\overline{\rho}\widehat{\boldsymbol{u} \circ \boldsymbol{u}}) + 2\boldsymbol{\Omega} \times \overline{\rho}\widehat{\boldsymbol{u}} + \nabla\overline{p} &= -\overline{\rho}g\boldsymbol{k} \\ c_v((\overline{\rho}\widehat{T})_t + \nabla \cdot (\overline{\rho}\widehat{\boldsymbol{u}T})) + \overline{p\nabla \cdot \boldsymbol{u}} &= \overline{Q} \\ \overline{p} &= \overline{\rho}R_{\text{air}}\widehat{T}. \end{aligned} \tag{2.57}$$

For a detailed derivation see [Ade08]. Note that we use the dimensionful equation system for better readability.

Due to the choice of the density-weighted filtering for $\boldsymbol{u}$ and $T$, the continuity equation and the equation of state remain unchanged, i.e. they are identical for both the original variables and their large-scale filtered parts. Only the momentum and temperature equation differ because of their non-linearity. Therefore, a decomposition of the convective term in the momentum equation

$$\begin{aligned} \overline{\rho}\widehat{\boldsymbol{u} \circ \boldsymbol{u}} &= \overline{\rho}(\widehat{\widehat{\boldsymbol{u}} \circ \widehat{\boldsymbol{u}}} + \widehat{\widehat{\boldsymbol{u}} \circ \boldsymbol{u}''} + \widehat{\boldsymbol{u}'' \circ \widehat{\boldsymbol{u}}} + \widehat{\boldsymbol{u}'' \circ \boldsymbol{u}''}) \\ &= \overline{\rho}\widehat{\boldsymbol{u}} \circ \widehat{\boldsymbol{u}} + \underbrace{\overline{\rho}(\widehat{\widehat{\boldsymbol{u}} \circ \widehat{\boldsymbol{u}}} - \widehat{\boldsymbol{u}} \circ \widehat{\boldsymbol{u}})}_{\boldsymbol{\mathcal{L}}} + \underbrace{\overline{\rho}(\widehat{\widehat{\boldsymbol{u}} \circ \boldsymbol{u}''} + \widehat{\boldsymbol{u}'' \circ \widehat{\boldsymbol{u}}})}_{\boldsymbol{\mathcal{C}}} + \underbrace{\overline{\rho}\widehat{\boldsymbol{u}'' \circ \boldsymbol{u}''}}_{\boldsymbol{\mathcal{R}}} \end{aligned} \tag{2.58}$$

leads to additional terms, which can be combined and interpreted as a new stress tensor

$$\boldsymbol{\tau}^{\text{RANS}} := \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{C}} + \boldsymbol{\mathcal{R}} \tag{2.59}$$

representing a kind of "turbulent" friction with the so-called Leonard term $\boldsymbol{\mathcal{L}}$, cross term $\boldsymbol{\mathcal{C}}$, and Reynolds term $\boldsymbol{\mathcal{R}}$.

Analogously, we decompose the corresponding term in the temperature equation in

$$\begin{aligned}\overline{\rho}\widehat{\boldsymbol{u}T} &= \overline{\rho}(\widehat{\widehat{\boldsymbol{u}}\widehat{T}} + \widehat{\widehat{\boldsymbol{u}}T''} + \widehat{\boldsymbol{u}''\widehat{T}} + \widehat{\boldsymbol{u}''T''}) \\ &= \overline{\rho}\widehat{\boldsymbol{u}}\widehat{T} + \underbrace{\overline{\rho}(\widehat{\widehat{\boldsymbol{u}}\widehat{T}} - \widehat{\boldsymbol{u}}\widehat{T})}_{\boldsymbol{\mathcal{L}}_T} + \underbrace{\overline{\rho}(\widehat{\widehat{\boldsymbol{u}}T''} + \widehat{\boldsymbol{u}''\widehat{T}})}_{\boldsymbol{\mathcal{C}}_T} + \underbrace{\overline{\rho}\widehat{\boldsymbol{u}''T''}}_{\boldsymbol{\mathcal{R}}_T}\end{aligned} \tag{2.60}$$

with the heat flux

$$\boldsymbol{q}^{\mathrm{RANS}} := \boldsymbol{\mathcal{L}}_T + \boldsymbol{\mathcal{C}}_T + \boldsymbol{\mathcal{R}}_T. \tag{2.61}$$

Special treatment is also necessary for the pressure dilatation term in the temperature equation

$$\begin{aligned}\overline{p\nabla\cdot\boldsymbol{u}} &= \overline{p}\nabla\cdot\widehat{\boldsymbol{u}} + \overline{p\nabla\cdot\boldsymbol{u}''} \\ &= \overline{p}\nabla\cdot\widehat{\boldsymbol{u}} + \overline{p}\nabla\cdot\overline{\boldsymbol{u}''} + \overline{p'\nabla\cdot\boldsymbol{u}''}.\end{aligned} \tag{2.62}$$

So these three terms induce additional terms when filtered, which cannot be expressed by the mean parts of the variables alone. Instead, the fluctuations $\boldsymbol{u}''$, $T''$ and $p'$ occur as additional variables in the filtered equation system

$$\begin{aligned}\overline{\rho}_t + \nabla\cdot(\overline{\rho}\widehat{\boldsymbol{u}}) &= 0 \\ (\overline{\rho}\widehat{\boldsymbol{u}})_t + \nabla\cdot(\overline{\rho}\widehat{\boldsymbol{u}}\circ\widehat{\boldsymbol{u}}) + 2\boldsymbol{\Omega}\times\overline{\rho}\widehat{\boldsymbol{u}} + \nabla\overline{p} &= -\overline{\rho}g\boldsymbol{k} - \nabla\cdot\boldsymbol{\tau}^{\mathrm{RANS}} \\ c_v((\overline{\rho}\widehat{T})_t + \nabla\cdot(\overline{\rho}\widehat{\boldsymbol{u}}\widehat{T})) + \overline{p}\nabla\cdot\widehat{\boldsymbol{u}} &= \overline{Q} - c_v\nabla\cdot\boldsymbol{q}^{\mathrm{RANS}} - \overline{p\nabla\cdot\boldsymbol{u}''} \\ \overline{p} &= \overline{\rho}R_{\mathrm{air}}\widehat{T}.\end{aligned} \tag{2.63}$$

Therefore, the system is no longer closed and we need further equations, which describe the dependencies of the new unknowns in relation to the mean variables. These dependencies cannot be exactly specified, because the subtraction of the filtered equations from the original ones again leads to a non-closed system. Thus, we are reliant on empirical approaches, also called parameterizations.

### Reynolds Assumption

In the literature, the equations are at first further simplified by usually postulating the so-called Reynolds assumption

$$\overline{\psi'} = 0 \quad \text{and} \quad \widehat{\zeta''} = 0 \tag{2.64}$$

or the equivalent formulation

$$\overline{\overline{\psi}} = \overline{\psi} \quad \text{and} \quad \widehat{\widehat{\zeta}} = \widehat{\zeta}. \tag{2.65}$$

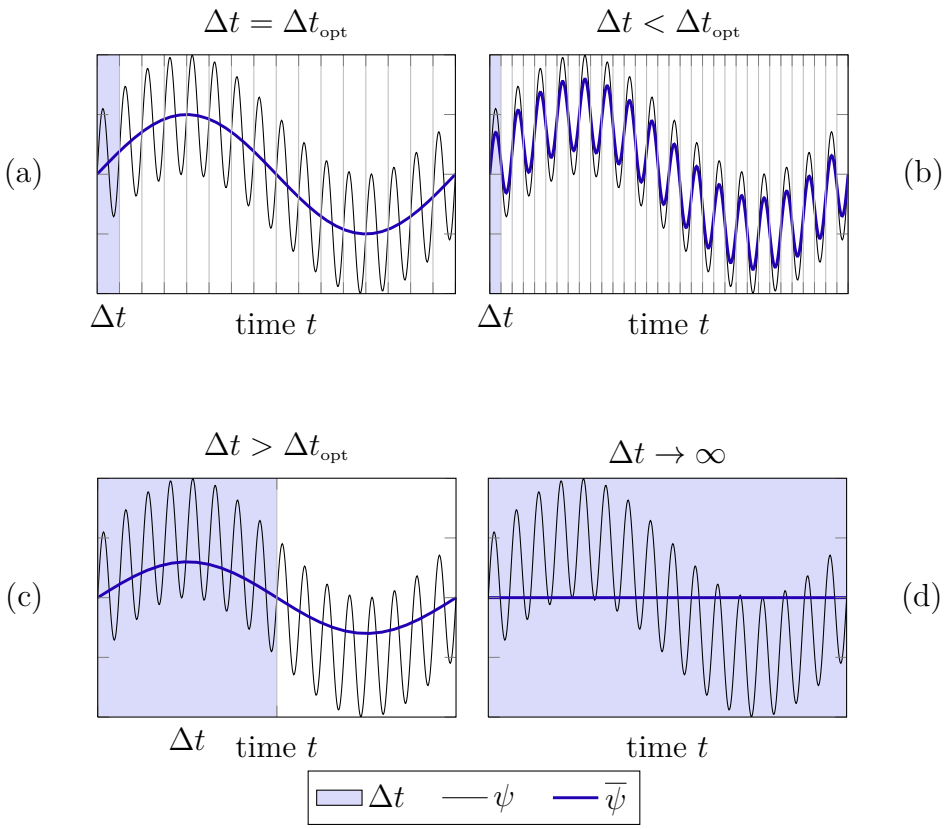



Figure 2.6.: Reynolds average $\overline{\psi}$ (thick line) of a function $\psi$ (thin line) with different filter widths $\Delta t$: (a) optimal, (b) too small, (c) too large, and (d) $\Delta t \to \infty$.

With this postulation, the Leonard and cross terms in $\boldsymbol{\tau}^{\text{RANS}}$ and $\boldsymbol{q}^{\text{RANS}}$ vanish.

However, the Reynolds assumption is in general not fulfilled, although it is widely assumed to be valid in the vast amount of literature concerning the dynamics of the atmosphere, and thus the Leonard and cross terms are consequently neglected. Most authors say – if the definition of the averaging operator (2.50) is not omitted anyway – that the averaging interval $\Delta t$ has to be chosen such that the Reynolds assumption is satisfied. But this is not always possible. The postulation indirectly requires a $\Delta t$ large enough to filter fast turbulent motions, but also small enough to preserve the large-scale trend of the variables. In other words, a spectral gap is being postulated, i.e. an explicit scale separation between turbulent and non-turbulent parts of the flow. For that, have a look at Figure 2.6. Plot (a) shows the optimal choice of $\Delta t$ with the existence of a spectral gap, for which the Reynolds assumption is fulfilled. In contrast, in (b) and (c) the assumption is not valid, since a second filtering would not yield the same result. In the case of (d), where $\Delta t \to \infty$, the flow gets stationary and the assumption holds in the limit.

As pointed out in [GMT00, GT99], a scale separation in the atmosphere is in general not possible. In fact, atmospheric measurements have shown that only for the vertical velocity such a postulation can be made. Of course, the case $\Delta t \to \infty$ is universally valid, but we are interested in a temporal evolution and not in a steady-state flow.

For lack of adequate and approved parameterizations for the Leonard and cross terms in respect of a temporal averaging and since we want to give an overview of usual approaches, we also neglect these terms at this point. But we explicitly point out that this is only valid for $\Delta t \to \infty$ and otherwise an error is introduced [GMT00, GT99].

Consequently, we now postulate the Reynolds assumption and thus get

$$\boldsymbol{\tau}^{\mathrm{RANS}} = \overline{\rho}\widehat{\boldsymbol{u}'' \circ \boldsymbol{u}''}, \tag{2.66}$$

$$\boldsymbol{q}^{\mathrm{RANS}} = \overline{\rho}\widehat{\boldsymbol{u}''T''}. \tag{2.67}$$

Therefore, in the momentum and temperature equations remain one additional turbulent term each, which has to be parameterized. The turbulent addition in the pressure dilatation term (2.62) represents the turbulent expansion power which is also generally neglected [Pic97].

### Prandtl's Mixing Length Model

One common approach for the parameterization of the remaining turbulent terms is based on Prandtl's mixing length theory [Oer04, Pic97]. The idea consists of an analogy between turbulent and molecular friction, which we will present in the following.

Inner friction arise from collisions of molecules. Those molecules pass through a free path, collide, and exchange momentum. The mean free path, i.e. the path which a molecule averagely traverses between two sequent collisions, is $10^{-7}$ m for air under normal conditions. The idea of Prandtl consists of an introduction of a mixing length $l$ for turbulent friction as well. Along this mixing length the momentum of a "turbulence parcel" is conserved before it is mixed with the environment and thus exchanges its momentum. The difficulty consists in the choice of the mixing length which should preferably be independent of the flow velocity.

A detailed derivation of Prandtl's ansatz and a description of limitations can be found in [Ade08, Pic97]. In summary, due to the dominance of turbulent shear flow in the atmospheric boundary layer and under consideration of Prandtl's mixing length theory, we get the approximations

$$\nabla \cdot \boldsymbol{\tau}^{\mathrm{RANS}} \cong -\frac{\partial}{\partial z}\left(A_M \frac{\partial \widehat{\boldsymbol{u}}_{\shortparallel}}{\partial z}\right), \tag{2.68}$$

$$\nabla \cdot \boldsymbol{q}^{\mathrm{RANS}} \cong -\frac{\partial}{\partial z}\left(A_H \frac{\partial \widehat{T}}{\partial z}\right) \tag{2.69}$$

for the additional turbulent terms. Here, $z$ is the vertical direction, $A_M$ the exchange

coefficient of momentum

$$A_M := \overline{\rho}\, l^2 \left| \frac{\partial \widehat{\boldsymbol{u}}_{\shortparallel}}{\partial z} \right| \tag{2.70}$$

with mean mixing length $l$, and $A_H$ the exchange coefficient of sensible heat which can be chosen as

$$A_H \cong 1.35 A_M \tag{2.71}$$

due to experimental evidence [Stu88].

Now, only the mean mixing length $l$ is still unknown but a multitude of heuristic parameterizations exists, e.g. a height-dependent ansatz

$$l \cong \kappa z \tag{2.72}$$

with the von Kármán constant $\kappa = 0.4$ or a variation with an upper limit for the mixing length according to Blackadar

$$l \cong \frac{\kappa z}{1 + \frac{\kappa z}{l_\infty}} \tag{2.73}$$

with an asymptotical mixing length $l_\infty = 500\,\mathrm{m}$ [DFH+11].

### Scale Analysis

Finally, the additional terms have to be transferred into their dimensionless form so that we can compare their magnitudes in terms of Section 2.2.3 and thus complete the modeling of turbulence. Note that the turbulent stress tensor only appears in the horizontal momentum equation due to the choice of modeling (2.68). With the corresponding prefactor and ansatz (2.72) we get the dimensionless form

$$\frac{l^{\shortparallel}_{\mathrm{ref}}}{\rho_{\mathrm{ref}} (\boldsymbol{u}^{\shortparallel}_{\mathrm{ref}})^2} \left( \nabla \cdot \boldsymbol{\tau}^{\mathrm{RANS}} \right)^* = \underbrace{\frac{l^{\shortparallel}_{\mathrm{ref}}}{l^{\perp}_{\mathrm{ref}}} \kappa^2}_{\boxed{\mathcal{O}(10^{-1})}} \left( -\frac{\partial}{\partial z^*} \left( \overline{\rho}^* z^{*2} \left| \frac{\partial \widehat{\boldsymbol{u}}^*_{\shortparallel}}{\partial z^*} \right| \frac{\partial \widehat{\boldsymbol{u}}^*_{\shortparallel}}{\partial z^*} \right) \right). \tag{2.74}$$

Again, the label $^*$ marks dimensionless quantities.

The analogous nondimensionalization of the turbulent heat flux leads to

$$\frac{l^{\shortparallel}_{\mathrm{ref}}}{\rho_{\mathrm{ref}} \boldsymbol{u}^{\shortparallel}_{\mathrm{ref}} T_{\mathrm{ref}}} \left( \nabla \cdot \boldsymbol{q}^{\mathrm{RANS}} \right)^* = \underbrace{1.35 \frac{l^{\shortparallel}_{\mathrm{ref}}}{l^{\perp}_{\mathrm{ref}}} \kappa^2}_{\boxed{\mathcal{O}(10^{-1})}} \left( -\frac{\partial}{\partial z^*} \left( \overline{\rho}^* z^{*2} \left| \frac{\partial \widehat{\boldsymbol{u}}^*_{\shortparallel}}{\partial z^*} \right| \frac{\partial \widehat{T}^*}{\partial z^*} \right) \right) \tag{2.75}$$

with the prefactor of the temperature equation.

Applying the small-scale reference values of Section 2.2.3, the additional turbulent terms in the horizontal momentum and temperature equation both have the magnitude $\mathcal{O}(10^{-1})$ at ground level and thus have to be accounted for, compare Table 2.2.

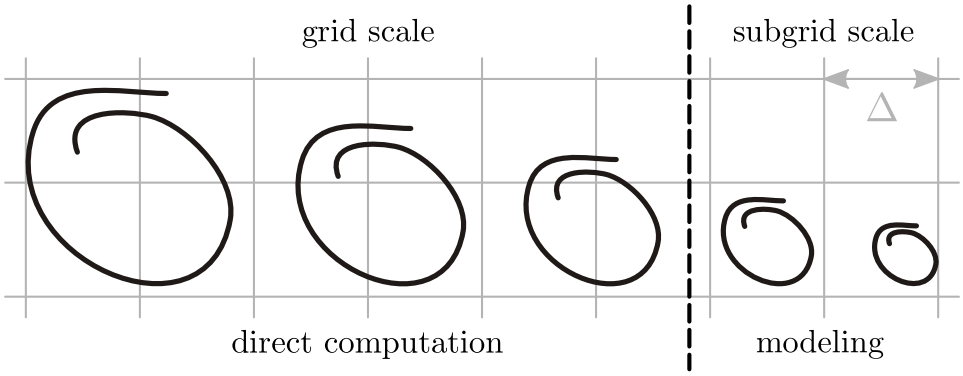



Figure 2.7.: LES concept adapted from [Bre02]. The filter width $\Delta$ separates the resolvable and thus direct computable eddies on the grid scale from the eddies on the subgrid scale.

## 2.4.2. Large Eddy Simulation

RANS turbulence modeling described in the previous section is based upon a temporal averaging of the flow equations. For the arising additional turbulent terms, a multitude of parameterizations have been proposed, neither of them claiming universal validity. Here, the crucial problem is that the whole spectrum of turbulent length scales has to be modeled due to the temporal averaging of the RANS approach and that this is very flow-dependent.

### Spatial Filtering

The idea of Large Eddy Simulation (LES) is a spatial instead of a temporal averaging, such that large eddies are resolved in contrast to small ones and thus only the influence of small eddies on grid scale eddies has to be modeled, see Figure 2.7. In this way, the difficulty of the RANS concept is restricted to the modeling of small-scale turbulences, for which simpler and more universal models can be chosen because of their characteristics like short life span, wide homogeneity, and isotropy. Thus, LES can be interpreted as a compromise between DNS and RANS.

In our description, we follow [Bre02, EHSZ92, MSCL91]. At first, we choose a spatial filter operator of the form

$$\overline{\psi}(\boldsymbol{x},t) := \int_{\Omega} G(\boldsymbol{x},\widetilde{\boldsymbol{x}},\Delta)\psi(\widetilde{\boldsymbol{x}},t)d\widetilde{\boldsymbol{x}}, \tag{2.76}$$

where $G$ is a filter function with filter width $\Delta$. A filter width with different values $\Delta_i$

in respect of the three spatial directions $i$ can be chosen as product

$$\Delta = (\Delta_1 \Delta_2 \Delta_3)^{\frac{1}{3}}. \tag{2.77}$$

Every variable $\psi$ can now be split into its filtered part on the grid scale (GS) $\overline{\psi}$ and its subgrid scale part (SGS) $\psi'$, which is not resolved by the grid and has to be modeled accordingly,

$$\psi = \overline{\psi} + \psi'. \tag{2.78}$$

A natural choice of $G$ is a top hat filter

$$G_i(x_i, \widetilde{x}_i, \Delta_i) = \begin{cases} \frac{1}{\Delta_i} & \text{for } |x_i - \widetilde{x}_i| \leq \frac{1}{2}\Delta_i \\ 0 & \text{else} \end{cases} \tag{2.79}$$

with

$$G(\boldsymbol{x}, \widetilde{\boldsymbol{x}}, \Delta) = \prod_{i=1}^{3} G_i(x_i, \widetilde{x}_i, \Delta_i), \tag{2.80}$$

which leads to an integration of the variable $\psi$ over the filter width $\Delta$. Note that the integral of $G$ itself is 1 and that in the limit $\Delta \to 0$ the filtered variable $\overline{\psi}$ converges to the original non-filtered $\psi$. Therefore, an LES with vanishing filter width becomes a DNS.

In LES, the actual choice of the filter width plays a decisive role. On the one hand, it should be as small as possible so that many eddies are resolved and only a small subgrid scale part has to be modeled. On the other hand, from a numerical point of view, the filter width should be much greater than the mesh width $h_i$ of the used grid to assure independence of the numerical method. As a compromise, filter widths in the range of $h_i \leq \Delta_i \leq 2h_i$ are often used in practice.

Application of the LES filtering to our equation system formally leads, analogously to the RANS filtering, to system (2.63). However, the filtered variables have to be interpreted in a different way. In RANS they represent temporal averages over a large time interval, whereas in LES they are spatial averages over a small volume.

Note that analogously to (2.52) we use a density-weighted average for $\boldsymbol{u}$ and $T$ and that we do not postulate the Reynolds assumption (2.64), which was already questionable in RANS. The permutability of the filter with the differential operator (2.54) is also not universally valid but leads to a commutation error. However, this error vanishes for special filter operators like the top hat filter (2.79) since it is of convolution type

$$G_i(x_i, \widetilde{x}_i, \Delta_i) = G_i(x_i - \widetilde{x}_i, \Delta_i = \text{const}) \tag{2.81}$$

if the filter width is chosen to be constant [Dah03]. The use of spatial variable filter widths, as they may be reasonable for complex turbulent flows, leads to an error of magnitude $\mathcal{O}(\Delta^2)$, which is acceptable for numerical methods up to second order [Bre02].

### Bardina-Smagorinsky Model

We now take a closer look at the subgrid scale stress tensor

$$\boldsymbol{\tau}^{\mathrm{SGS}} := \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{C}} + \boldsymbol{\mathcal{R}} \tag{2.82}$$

with the definitions of (2.58). The Leonard term $\boldsymbol{\mathcal{L}}$ describes the interaction of the grid scale eddies among each other inducing small-scale turbulence. Since this term is expressed only in filtered variables it can be directly computed and requires no modeling.

The cross term $\boldsymbol{\mathcal{C}}$ consists of the interactions between grid and subgrid scale and thus needs to be modeled. A common approach is

$$\boldsymbol{\mathcal{C}} \cong \overline{\rho}\left(\widehat{\boldsymbol{u}} \circ \widehat{\boldsymbol{u}} - \widehat{\widehat{\boldsymbol{u}}} \circ \widehat{\widehat{\boldsymbol{u}}}\right). \tag{2.83}$$

This represents the compressible analogon of the scale similarity model of Bardina, which is based on the idea that the smallest resolved eddies on the grid scale are similar to the largest non-resolved eddies on the subgrid scale [Bre02, EHSZ92, SEZH88].

The subgrid scale stress tensor $\boldsymbol{\mathcal{R}}$ is at first split into its anisotropic and isotropic part

$$\boldsymbol{\mathcal{R}} = \boldsymbol{\mathcal{R}}_A + \boldsymbol{\mathcal{R}}_I \tag{2.84}$$

with

$$(\cdot)_I := \frac{1}{3}\mathrm{tr}(\cdot)\mathbb{1} \quad \text{and} \quad (\cdot)_A := (\cdot) - (\cdot)_I. \tag{2.85}$$

Then, a compressible generalization of the well-known Smagorinsky model [Sma63, Bre02] leads to

$$\boldsymbol{\mathcal{R}}_A \cong -2C_A\Delta^2\overline{\rho}\left\|\widehat{\boldsymbol{S}}\right\|_F\left(\widehat{\boldsymbol{S}} - \frac{1}{3}\mathrm{tr}\left(\widehat{\boldsymbol{S}}\right)\mathbb{1}\right) \tag{2.86}$$

with the deformation tensor

$$\widehat{\boldsymbol{S}} := \frac{1}{2}\left(\nabla\widehat{\boldsymbol{u}} + (\nabla\widehat{\boldsymbol{u}})^{\mathsf{T}}\right) \tag{2.87}$$

and the Frobenius norm

$$\left\|\widehat{\boldsymbol{S}}\right\|_F := \sqrt{\sum_{i,j}\left|\widehat{S}_{ij}\right|^2}. \tag{2.88}$$

For the isotropic part, we get

$$\boldsymbol{\mathcal{R}}_I \cong \frac{2}{3}C_I\Delta^2\overline{\rho}\left\|\widehat{\boldsymbol{S}}\right\|_F^2\mathbb{1}. \tag{2.89}$$

In [SEZH88], the constants $C_A$ and $C_I$ are chosen to be 0.012 and 0.0066, respectively, resulting from a comparison with a DNS of compressible isotropic turbulence. But in principle, these constants are flow-dependent and vary both in time and space, see below.

Altogether, the model for the subgrid scale stress tensor reads as follows

$$\boldsymbol{\tau}^{\mathrm{SGS}} \cong \underbrace{\overline{\rho}\left(\widehat{\widehat{\boldsymbol{u}}\circ\widehat{\boldsymbol{u}}} - \widehat{\widehat{\boldsymbol{u}}}\circ\widehat{\widehat{\boldsymbol{u}}}\right)}_{=:\,\boldsymbol{\tau}_1} \underbrace{- 2C_A\Delta^2\overline{\rho}\left\|\widehat{\boldsymbol{S}}\right\|_F\left(\widehat{\boldsymbol{S}} - \frac{1}{3}\mathrm{tr}\left(\widehat{\boldsymbol{S}}\right)\mathbb{1}\right)}_{=:\,\boldsymbol{\tau}_2} \underbrace{- \frac{2}{3}C_I\Delta^2\overline{\rho}\left\|\widehat{\boldsymbol{S}}\right\|_F^2\mathbb{1}}_{=:\,\boldsymbol{\tau}_3}. \quad (2.90)$$

In the temperature equation, the subgrid scale heat flux

$$\boldsymbol{q}^{\mathrm{SGS}} := \boldsymbol{\mathcal{L}}_T + \boldsymbol{\mathcal{C}}_T + \boldsymbol{\mathcal{R}}_T \quad (2.91)$$

has to be modeled accordingly. Analogously to the cross term (2.83), we set

$$\boldsymbol{\mathcal{C}}_T \cong \overline{\rho}\left(\widehat{\boldsymbol{u}}\widehat{T} - \widehat{\widehat{\boldsymbol{u}}}\widehat{\widehat{T}}\right) \quad (2.92)$$

and for the Reynolds term

$$\boldsymbol{\mathcal{R}}_T \cong -\frac{C_A}{Pr_t}\Delta^2\overline{\rho}\left\|\widehat{\boldsymbol{S}}\right\|_F\nabla\widehat{T} \quad (2.93)$$

where $Pr_t$ denotes the turbulent Prandtl number which is often heuristically chosen to be 0.7. So, the subgrid scale heat flux can be modeled as

$$\boldsymbol{q}^{\mathrm{SGS}} \cong \underbrace{\overline{\rho}\left(\widehat{\widehat{\boldsymbol{u}}\widehat{T}} - \widehat{\widehat{\boldsymbol{u}}}\widehat{\widehat{T}}\right)}_{=:\,\boldsymbol{q}_1} \underbrace{- \frac{C_A}{Pr_t}\Delta^2\overline{\rho}\left\|\widehat{\boldsymbol{S}}\right\|_F\nabla\widehat{T}}_{=:\,\boldsymbol{q}_2}. \quad (2.94)$$

The other additional term $\overline{p\nabla\cdot\boldsymbol{u}''}$ represents the turbulent dilatation power which can be neglected based on DNS studies [EHSZ92, MSCL91, Pic97].

#### Germano's Dynamical Model

Since the parameters $C_A$, $C_I$, and $Pr_t$ actually vary in time and space, a dynamical adjustment during the simulation would be more reasonable than the choice of constant heuristic values. The idea of such a dynamical model originates from Germano [GPMC91]. Here, the parameters of a base model, that is in our case $C_A$, $C_I$, and $Pr_t$, are determined by evaluating the smallest resolved scales. The idea is akin to the model of Bardina where a similarity between the smallest resolved eddies on the grid scale and the largest non-resolved eddies on the subgrid scale is assumed.

The essential step of Germano's ansatz consists in the choice of a test filter with a wider width than the original $\Delta$, with which the contribution of the smallest still resolved scales are separated. Then, in short, equation systems for the sought parameters can be derived and solved e.g. by a least squares ansatz [Lil92]. In this way, dependent on space and time, we can dynamically calculate the values of $C_A$, $C_I$, and $Pr_t$. For more details see [Ade08, VGK94, ZSK93].

### Scale Analysis

Let us now summarize the LES-filtered equation system

$$\begin{aligned}
\overline{\rho}_t + \nabla \cdot (\overline{\rho}\widehat{\boldsymbol{u}}) &= 0 \\
(\overline{\rho}\widehat{\boldsymbol{u}})_t + \nabla \cdot (\overline{\rho}\widehat{\boldsymbol{u}} \circ \widehat{\boldsymbol{u}}) + 2\boldsymbol{\Omega} \times \overline{\rho}\widehat{\boldsymbol{u}} + \nabla\overline{p} &= -\overline{\rho}g\boldsymbol{k} - \boxed{\nabla \cdot \boldsymbol{\tau}^{\mathrm{SGS}}} \\
c_v \left( (\overline{\rho}\widehat{T})_t + \nabla \cdot (\overline{\rho}\widehat{\boldsymbol{u}}\widehat{T}) \right) + \overline{p}\nabla \cdot \widehat{\boldsymbol{u}} &= \overline{Q} - \boxed{c_v \nabla \cdot \boldsymbol{q}^{\mathrm{SGS}}} \\
\overline{p} &= \overline{\rho} R_{\mathrm{air}} \widehat{T}.
\end{aligned} \tag{2.95}$$

The framed terms are new compared to the original system (2.45) and both can be modeled with a mixed Bardina-Smagorinsky approach, the subgrid scale stress tensor with (2.90) and the subgrid scale heat flux with (2.94).

Finally, a nondimensionalization and scale analysis of the new terms complete our modeling process and allow for the comparison of the magnitudes with those of other effects in terms of Section 2.2.3. At first, we rewrite the deformation tensor (2.87) with horizontal and vertical dimensionless components of $\nabla$ and $\boldsymbol{u}$

$$\begin{aligned}
\widehat{\boldsymbol{S}} = \frac{u^{\shortparallel}_{\mathrm{ref}}}{2l^{\shortparallel}_{\mathrm{ref}}} \left( \nabla^*_{\shortparallel}\widehat{\boldsymbol{u}}^*_{\shortparallel} + \nabla^*_{\perp}\widehat{\boldsymbol{u}}^*_{\perp} + (\nabla^*_{\shortparallel}\widehat{\boldsymbol{u}}^*_{\shortparallel} + \nabla^*_{\perp}\widehat{\boldsymbol{u}}^*_{\perp})^{\mathsf{T}} \right) + \frac{u^{\perp}_{\mathrm{ref}}}{2l^{\shortparallel}_{\mathrm{ref}}} \left( \nabla^*_{\shortparallel}\widehat{\boldsymbol{u}}^*_{\perp} + (\nabla^*_{\shortparallel}\widehat{\boldsymbol{u}}^*_{\perp})^{\mathsf{T}} \right) \\
+ \frac{u^{\shortparallel}_{\mathrm{ref}}}{2l^{\perp}_{\mathrm{ref}}} \left( \nabla^*_{\perp}\widehat{\boldsymbol{u}}^*_{\shortparallel} + (\nabla^*_{\perp}\widehat{\boldsymbol{u}}^*_{\shortparallel})^{\mathsf{T}} \right).
\end{aligned}$$

Since the additional turbulent terms appear in the small-scale region at ground level, we can use the set of small-scale values chosen in Section 2.2.3, in particular $u^{\shortparallel}_{\mathrm{ref}} = u^{\perp}_{\mathrm{ref}}$ and $l^{\shortparallel}_{\mathrm{ref}} = l^{\perp}_{\mathrm{ref}}$, with which the nondimensionalization of $\widehat{\boldsymbol{S}}$ is independent of the horizontal and vertical splitting

$$\widehat{\boldsymbol{S}} = \frac{u^{\shortparallel}_{\mathrm{ref}}}{2l^{\shortparallel}_{\mathrm{ref}}} \left( \nabla^*\widehat{\boldsymbol{u}}^* + (\nabla^*\widehat{\boldsymbol{u}}^*)^{\mathsf{T}} \right). \tag{2.96}$$

With the abbreviations introduced in (2.90), it follows

$$\begin{aligned}
(\nabla \cdot \boldsymbol{\tau}_1)^*_{\shortparallel} &= \frac{\rho_{\mathrm{ref}}(u^{\shortparallel}_{\mathrm{ref}})^2}{l^{\shortparallel}_{\mathrm{ref}}} \left( \nabla^*_{\shortparallel} \cdot \left( \overline{\rho}^* \left( \widehat{\widehat{\boldsymbol{u}}^*_{\shortparallel} \circ \widehat{\boldsymbol{u}}^*_{\shortparallel}} - \widehat{\widehat{\boldsymbol{u}}^*_{\shortparallel}} \circ \widehat{\widehat{\boldsymbol{u}}^*_{\shortparallel}} \right) \right) + \nabla^*_{\perp} \cdot \left( \overline{\rho}^* \left( \widehat{\widehat{\boldsymbol{u}}^*_{\perp} \circ \widehat{\boldsymbol{u}}^*_{\shortparallel}} - \widehat{\widehat{\boldsymbol{u}}^*_{\perp}} \circ \widehat{\widehat{\boldsymbol{u}}^*_{\shortparallel}} \right) \right) \right), \\
(\nabla \cdot \boldsymbol{\tau}_1)^*_{\perp} &= \frac{\rho_{\mathrm{ref}}(u^{\perp}_{\mathrm{ref}})^2}{l^{\perp}_{\mathrm{ref}}} \left( \nabla^*_{\shortparallel} \cdot \left( \overline{\rho}^* \left( \widehat{\widehat{\boldsymbol{u}}^*_{\shortparallel} \circ \widehat{\boldsymbol{u}}^*_{\perp}} - \widehat{\widehat{\boldsymbol{u}}^*_{\shortparallel}} \circ \widehat{\widehat{\boldsymbol{u}}^*_{\perp}} \right) \right) + \nabla^*_{\perp} \cdot \left( \overline{\rho}^* \left( \widehat{\widehat{\boldsymbol{u}}^*_{\perp} \circ \widehat{\boldsymbol{u}}^*_{\perp}} - \widehat{\widehat{\boldsymbol{u}}^*_{\perp}} \circ \widehat{\widehat{\boldsymbol{u}}^*_{\perp}} \right) \right) \right), \\
(\nabla \cdot \boldsymbol{\tau}_2)^*_{\shortparallel} &= \frac{\Delta^2 C^{\mathrm{ref}}_A \rho_{\mathrm{ref}}(u^{\shortparallel}_{\mathrm{ref}})^2}{2(l^{\shortparallel}_{\mathrm{ref}})^2 l^{\shortparallel}_{\mathrm{ref}}} \nabla^*_{\shortparallel} \cdot \left( C^*_A \overline{\rho}^* \left\| \widehat{\boldsymbol{S}}^* \right\|_F \left( \widehat{\boldsymbol{S}}^* - \frac{1}{3}\mathrm{tr}\left( \widehat{\boldsymbol{S}}^* \right) \mathbb{1} \right) \right), \\
(\nabla \cdot \boldsymbol{\tau}_2)^*_{\perp} &= \frac{\Delta^2 C^{\mathrm{ref}}_A \rho_{\mathrm{ref}}(u^{\perp}_{\mathrm{ref}})^2}{2(l^{\perp}_{\mathrm{ref}})^2 l^{\perp}_{\mathrm{ref}}} \nabla^*_{\perp} \cdot \left( C^*_A \overline{\rho}^* \left\| \widehat{\boldsymbol{S}}^* \right\|_F \left( \widehat{\boldsymbol{S}}^* - \frac{1}{3}\mathrm{tr}\left( \widehat{\boldsymbol{S}}^* \right) \mathbb{1} \right) \right), \\
(\nabla \cdot \boldsymbol{\tau}_3)^*_{\shortparallel} &= \frac{\Delta^2 C^{\mathrm{ref}}_I \rho_{\mathrm{ref}}(u^{\shortparallel}_{\mathrm{ref}})^2}{6(l^{\shortparallel}_{\mathrm{ref}})^2 l^{\shortparallel}_{\mathrm{ref}}} \nabla^*_{\shortparallel} \cdot \left( C^*_I \overline{\rho}^* \left\| \widehat{\boldsymbol{S}}^* \right\|^2_F \mathbb{1} \right),
\end{aligned}$$

$$(\nabla \cdot \boldsymbol{\tau}_3)^*_\perp = \frac{\Delta^2 C_I^{\mathrm{ref}} \rho_{\mathrm{ref}} (u_{\mathrm{ref}}^\perp)^2}{6 (l_{\mathrm{ref}}^\perp)^2 l_{\mathrm{ref}}^\perp} \nabla^*_\perp \cdot \left( C_I^* \overline{\rho}^* \left\| \widehat{\boldsymbol{S}}^* \right\|_F^2 \mathbb{1} \right)$$

and analogously for the terms of the subgrid scale heat flux (2.94)

$$\begin{aligned}
(\nabla \cdot \boldsymbol{q}_1)^* &= \frac{\rho_{\mathrm{ref}} u_{\mathrm{ref}}^{\shortparallel} T_{\mathrm{ref}}}{l_{\mathrm{ref}}^{\shortparallel}} \left( \nabla^*_{\shortparallel} \cdot \left( \overline{\rho}^* \left( \widehat{\widehat{\boldsymbol{u}}^*_{\shortparallel} \widehat{T}^*} - \widehat{\widehat{\boldsymbol{u}}^*_{\shortparallel}} \widehat{\widehat{T}^*} \right) \right) + \nabla^*_\perp \cdot \left( \overline{\rho}^* \left( \widehat{\widehat{\boldsymbol{u}}^*_\perp \widehat{T}^*} - \widehat{\widehat{\boldsymbol{u}}^*_\perp} \widehat{\widehat{T}^*} \right) \right) \right), \\
(\nabla \cdot \boldsymbol{q}_2)^* &= \frac{\Delta^2 C_A^{\mathrm{ref}} \rho_{\mathrm{ref}} u_{\mathrm{ref}}^{\shortparallel} T_{\mathrm{ref}}}{2 P r_t^{\mathrm{ref}} l_{\mathrm{ref}}^{\shortparallel} (l_{\mathrm{ref}}^{\shortparallel})^2} \left( \nabla^*_{\shortparallel} \cdot \left( \frac{C_A^*}{P r_t^*} \overline{\rho}^* \left\| \widehat{\boldsymbol{S}}^* \right\|_F \nabla^*_{\shortparallel} \widehat{T}^* \right) \right) \\
&\quad + \frac{\Delta^2 C_A^{\mathrm{ref}} \rho_{\mathrm{ref}} u_{\mathrm{ref}}^{\shortparallel} T_{\mathrm{ref}}}{2 P r_t^{\mathrm{ref}} l_{\mathrm{ref}}^{\shortparallel} (l_{\mathrm{ref}}^\perp)^2} \left( \nabla^*_\perp \cdot \left( \frac{C_A^*}{P r_t^*} \overline{\rho}^* \left\| \widehat{\boldsymbol{S}}^* \right\|_F \nabla^*_\perp \widehat{T}^* \right) \right).
\end{aligned}$$

Finally, we add the prefactors of the horizontal and vertical momentum equations and the temperature equation according to Section 2.2.2 to the corresponding terms. Thus, we get for the subgrid scale stress tensor in the horizontal momentum equation

$$\frac{l_{\mathrm{ref}}^{\shortparallel}}{\rho_{\mathrm{ref}} (u_{\mathrm{ref}}^{\shortparallel})^2} \left( \nabla \cdot \boldsymbol{\tau}^{\mathrm{SGS}} \right)^*_{\shortparallel} = \underbrace{1}_{\boxed{\mathcal{O}(10^0)}} (\nabla \cdot \boldsymbol{\tau}_1)^*_{\shortparallel} - \underbrace{\frac{\Delta^2 C_A^{\mathrm{ref}}}{2 (l_{\mathrm{ref}}^{\shortparallel})^2}}_{\boxed{\mathcal{O}(10^{-2})}} (\nabla \cdot \boldsymbol{\tau}_2)^*_{\shortparallel} - \underbrace{\frac{\Delta^2 C_I^{\mathrm{ref}}}{6 (l_{\mathrm{ref}}^{\shortparallel})^2}}_{\boxed{\mathcal{O}(10^{-3})}} (\nabla \cdot \boldsymbol{\tau}_3)^*_{\shortparallel} \tag{2.97}$$

and for the part in the vertical momentum equation

$$\frac{l_{\mathrm{ref}}^\perp}{\rho_{\mathrm{ref}} (u_{\mathrm{ref}}^\perp)^2} \left( \nabla \cdot \boldsymbol{\tau}^{\mathrm{SGS}} \right)^*_\perp = \underbrace{1}_{\boxed{\mathcal{O}(10^0)}} (\nabla \cdot \boldsymbol{\tau}_1)^*_\perp - \underbrace{\frac{\Delta^2 C_A^{\mathrm{ref}}}{2 (l_{\mathrm{ref}}^\perp)^2}}_{\boxed{\mathcal{O}(10^{-2})}} (\nabla \cdot \boldsymbol{\tau}_2)^*_\perp - \underbrace{\frac{\Delta^2 C_I^{\mathrm{ref}}}{6 (l_{\mathrm{ref}}^\perp)^2}}_{\boxed{\mathcal{O}(10^{-3})}} (\nabla \cdot \boldsymbol{\tau}_3)^*_\perp. \tag{2.98}$$

Analogously, the subgrid scale heat flux term in the temperature equation reads

$$\frac{l_{\mathrm{ref}}^{\shortparallel}}{\rho_{\mathrm{ref}} u_{\mathrm{ref}}^{\shortparallel} T_{\mathrm{ref}}} \left( \nabla \cdot \boldsymbol{q}^{\mathrm{SGS}} \right)^* = \underbrace{1}_{\boxed{\mathcal{O}(10^0)}} (\nabla \cdot \boldsymbol{q}_1)^* - \underbrace{\frac{\Delta^2 C_A^{\mathrm{ref}}}{2 P r_t^{\mathrm{ref}} (l_{\mathrm{ref}}^{\shortparallel})^2}}_{\boxed{\mathcal{O}(10^{-2})}} (\nabla \cdot \boldsymbol{q}_2)^*_{\shortparallel} - \underbrace{\frac{\Delta^2 C_A^{\mathrm{ref}}}{2 P r_t^{\mathrm{ref}} (l_{\mathrm{ref}}^\perp)^2}}_{\boxed{\mathcal{O}(10^{-2})}} (\nabla \cdot \boldsymbol{q}_2)^*_\perp. \tag{2.99}$$

For the determination of the magnitudes of the characteristic numbers, we choose the small-scale reference values at ground level of Section 2.2.3, $\Delta = 10^2$ m, $Pr_t^{\mathrm{ref}} = 0.7$, and the values $C_A^{\mathrm{ref}} = 0.012$ and $C_I^{\mathrm{ref}} = 0.0066$ [SEZH88].

All in all, the additional turbulent terms in the horizontal and vertical momentum equations and in the temperature equation have a magnitude of $\mathcal{O}(10^0)$ for small-scale flow at ground level. Therefore, they have a relevant impact on the equations, compare Table 2.2.

As final remark, we summarize in Table 2.4 the main differences between the RANS

| RANS | LES |
|---|---|
| main differences | |
| • Temporal averaging over a large time interval.<br>• Modeling of the whole turbulence spectrum necessary.<br>• Parameterizations very flow-dependent, not general. | • Spatial averaging over a small volume.<br>• Modeling of small-scale turbulences necessary.<br>• Parameterizations more general due to short life span, homogeneity and isotropy of small eddies. |
| modeling approaches | |
| • Reynolds assumption.<br>• Prandtl's mixing length model.<br>• Determination of the mixing length according to von Kármán, Blackadar. | • Scale similarity model of Bardina for $\boldsymbol{\mathcal{C}}$.<br>• Smagorinsky model for $\boldsymbol{\mathcal{R}}$.<br>• Dynamical determination of the parameters according to Germano. |

Table 2.4.: Main differences between the RANS and LES concept and examples of different modeling approaches.

and LES concept. The many advantages of the latter, especially the lesser amount of turbulences which have to be modeled and thus the greater universality, recommend its application to atmospheric dynamics.

# 3 Horizontal Grid Generation

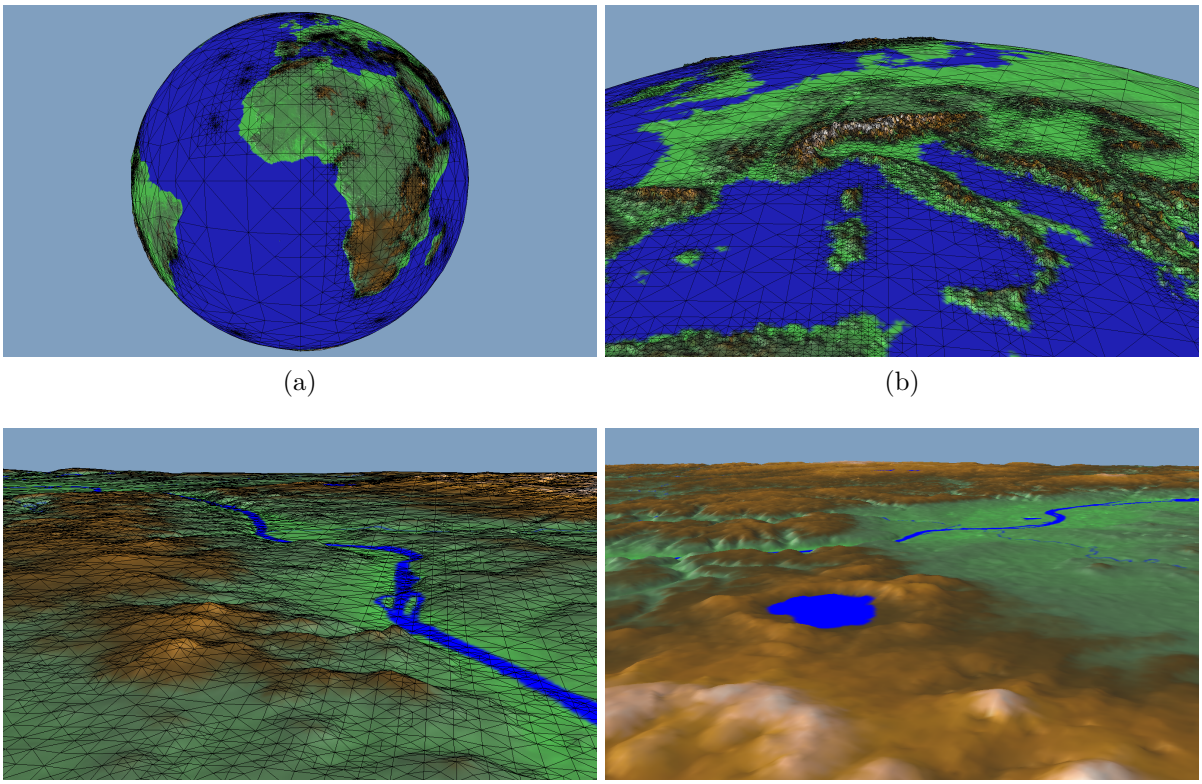
Due to the complexity of the Euler equations, no analytical solution is known so far, so the problem has to be approached numerically by computing approximated solutions in time and space. To this end, the continuous model derived in Section 2 has to be discretized, and thus also the continuous domain, i.e. the atmosphere, has to be substituted by a discrete grid consisting of interconnected points or cells, respectively. These points form the new domain of the discretized variables of the equation set.

Accounting for the anisotropic extensions of our computational domain, we split the generation of such an atmospheric grid in its horizontal and vertical part and start in this chapter with the horizontal triangulation of the lower boundary, that is the Earth's surface. As a prerequisite, we need a global topographical survey of the Earth, the corresponding data set being called global digital elevation model (GDEM). Only few such models are freely available, so we focus on the GTOPO30 [U.S96] and ASTER GDEM [Min09] data sets as basis for the meshes of this thesis.

Now, the numerous points of the given elevation model have to be selected and suitably connected to form a manageable mesh, for which we prescribe certain requirements. On the one hand, the spreading of the chosen grid points should be as uniform as possible and particularly avoid singularities at the poles. On the other hand, we want to enable adaptivity to permit higher resolutions for rough terrain. In this way, details of the landscape as well as heavier turbulences can be preserved, which are to be expected over rough terrain. Furthermore, it may be desirable to resolve interesting areas for certain applications higher but still maintain a global computational mesh.

In order to achieve this aim, we use principles of computer graphics [Ger03a] and take advantage of our prior work on an interactive visualization program of the Earth's surface. Previously, we transferred the concept of the SOAR Terrain Engine [LP02, Ger03b] for local rectangular terrain data sets to a global variant for the whole surface of the Earth [Ade05]. In Figure 3.1 different screenshots of results from our visualization program are shown.

In the following, we give a compact survey of the techniques used to generate a trian-



(a) (b)

(c) (d)

Figure 3.1.: Screenshots of our visualization program of the Earth [Ade05] based on the SOAR Terrain Engine. (a) View of the Earth, (b) view of Central Europe, (c) southbound view of the Siebengebirge and the Rhine Valley near Bonn and (d) westbound view of the Lake Laach in the Eifel region towards the Middle Rhine in Germany.

gulation of the Earth's orography. The extension of such a grid to the third dimension is then the focus of Section 4, in which the height of the atmosphere is added and accordingly discretized.

## 3.1. Global Digital Elevation Models

As mentioned before, only few global digital elevation models (GDEM) of the Earth exist because of the difficulty in generating reliable topographical surveys of the whole global surface. We choose as basis for the meshes of the thesis at hand the freely available data sets GTOPO30 [U.S96] and ASTER GDEM [Min09].

The GTOPO30 data set provides a global survey of the Earth's surface at a distance

| | GTOPO30 | ASTER GDEM | TanDEM-X |
|---|---|---|---|
| Horizontal spacing | 1,000 m | 30 m | 12 m |
| Horizontal resolution | 1,000 m (few excp.) | 120 m | 12 m |
| Vertical accuracy | ±30 m (60 % area), ±160 m (40 % area) at 90 % confidence | ±20 m at 95 % confidence | ±2 m at 90 % confidence |
| Coverage | global | 83° north – 83° south | global |
| Release year | 1996 | 2009 | 2014 |
| Data source | provided by eight organizations around the world | ASTER radiometer on NASA's Terra satellite | bistatic interferometer on twin satellites TerraSAR-X and TanDEM-X |
| Data acquisition | completed | ongoing | ongoing since 2010 |
| Distribution | USGS | METI/NASA | DLR |

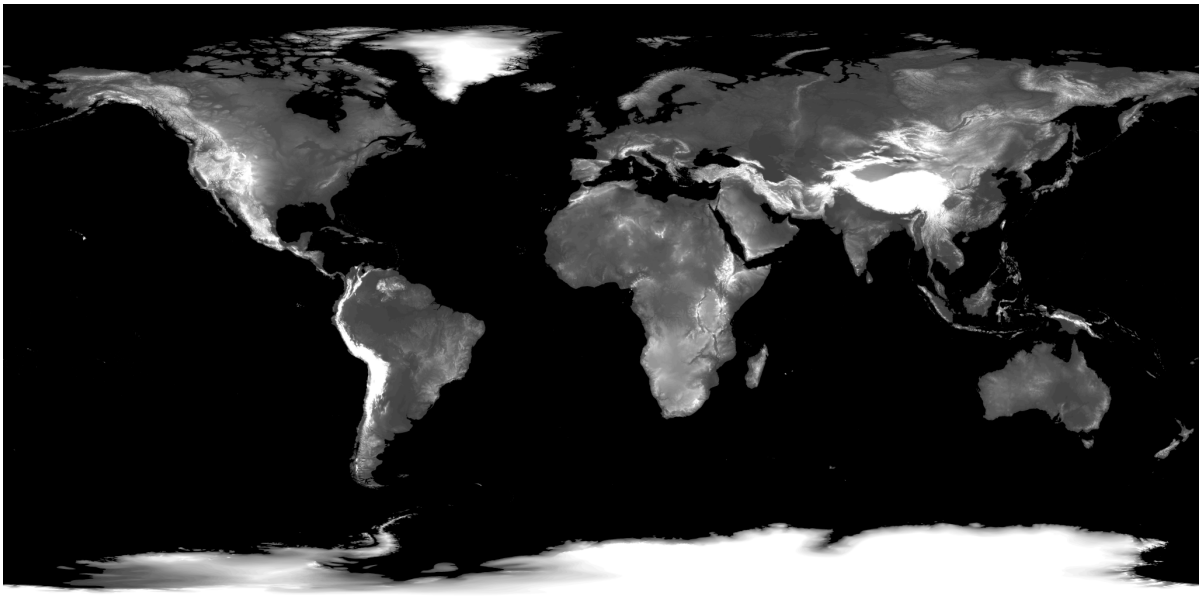
Table 3.1.: Comparison of the global digital elevation models GTOPO30 [U.S97], ASTER GDEM [AST09] and the announced TanDEM-X [Ger13].

of 30 arc-seconds which corresponds to a point spacing of about 1 km. It was created out of elevation data from eight different sources around the world, collected by the U.S. Geological Survey (USGS) and released in 1996 [U.S97].

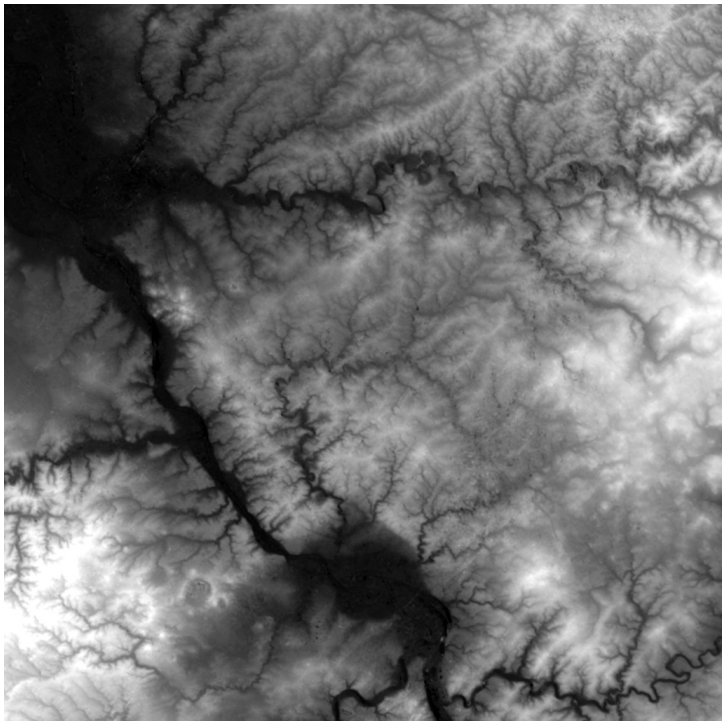
ASTER GDEM is a quite new data set released in 2009. It provides elevation data at a spatial spacing of 1 arc-second, or approximately 30 m, and the coverage of the Earth's surface is in between 83 degrees north and 83 degrees south. Since the digital elevation model is up-to-date it has still to be regarded as work in progress; the actual version one of ASTER GDEM is refered to as "research grade" [AST09] by its publishers, the Ministry of Economy, Trade and Industry of Japan (METI) and the National Aeronautics and Space Administration (NASA), who acknowledge artifacts and anomalies due to residual clouds and the matching algorithm used to generate the elevation values. Moreover, resolution inaccuracies are stated for parts of the released standard data. More precisely, the detail of topography resolvable in the ASTER GDEM appears to have a resolution of 3.8 arc-seconds corresponding to slightly less than 120 m [AST09].

Note here the distinction between the terms grid spacing and resolution. Even though the data may have a consistent grid spacing, the level of detail of the source data determines the actual resolution. So the topographic features that would be expected to be resolved at a particular spacing may not be represented in the elevation model due to the quality of the underlying source data.

Because of the described problems and ongoing development of the ASTER GDEM, we will basically use the GTOPO30 data set and employ the ASTER data only for areas

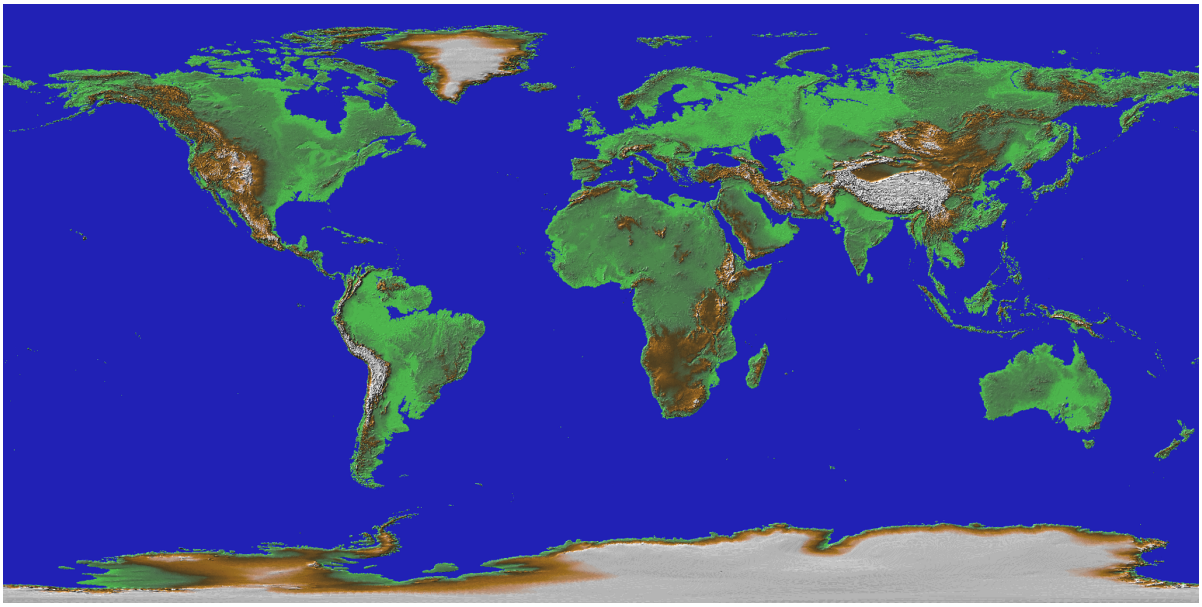


(a) GTOPO30

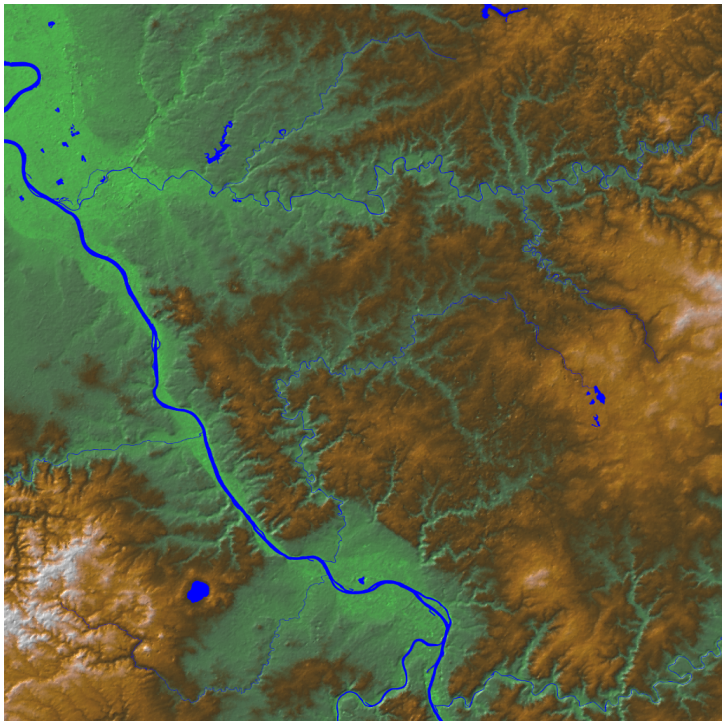


(b) ASTER GDEM

Figure 3.2.: (a) The GTOPO30 data set. (b) A detail of the ASTER GDEM showing the Rhineland between Bonn and Koblenz in Germany.



(c) GTOPO30



(d) ASTER GDEM

Figure 3.2.: Color-coded versions of (a) and (b). The color or the grayscale, respectively, refers to the height values.

where higher resolutions are desired. In Table 3.1, we summarize and compare the main features of these two GDEMs. In addition, Figure 3.2 shows the complete GTOPO30 elevation model and a detail of the ASTER GDEM.

Looking ahead, in 2014 the release of a new high-resolution GDEM is awaited. Led by the German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt, DLR), the mission TanDEM-X [Ger10, Ger13] is currently collecting elevation data via bistatic interferometry on the twin satellites TerraSAR-X and TanDEM-X. The announced GDEM will have a resolution of 12 m and is highly anticipated due to its accuracy and its entirely homogenous global mapping. As foresight, the announced features of TanDEM-X are also listed in Table 3.1.

## 3.2. Terrain Triangulation

The next step in our grid generation procedure is an appropriate connection and thus interpolation of the points given in the chosen elevation model. Such a triangulation forms a mesh consisting of e.g. polygonal elements which approximate the described surface.

The freedom in the choice of connection is limited by a few reasonable conditions. First of all, we require a regular or so-called saturated grid, which means that the whole domain is disjointly covered by a continuous mesh with no cracks or overlapping elements. Furthermore, we would like to avoid degenerated elements with very acute or obtuse angles. For keeping the mesh simple and preferably uniform, the polygons should rather possess a simple structure and as few different types of them as possible should exist. In addition, the triangulation technique should permit adaptivity so that higher resolutions for certain regions are possible while still maintaining a global regular grid.

Consideration of these requirements leads us to the choice of a triangular mesh generated by the so-called bisection method. We will firstly describe this technique for finite local domains and afterwards transfer it in Section 3.2.3 to our global setting.

### 3.2.1. Bisection Method

Let us now describe the well-known bisection method and its application to terrain triangulation in detail.

Consider a local quadratic domain of terrain. Then the coarsest triangular mesh consists of two right-angled triangles. The idea of the bisection method is that the next finer mesh is constructed by dividing every coarse triangle in halves. More precisely, we insert a new vertex, the so-called refinement vertex, in the middle of the longest edge of each triangle and connect it with the opposite corner. The concept is illustrated in Figure 3.3.

In this way, we get a hierarchy of regular uniform grids. In two-dimensional projection, that is without height values for each vertex, these grids consist of right-angled isosceles

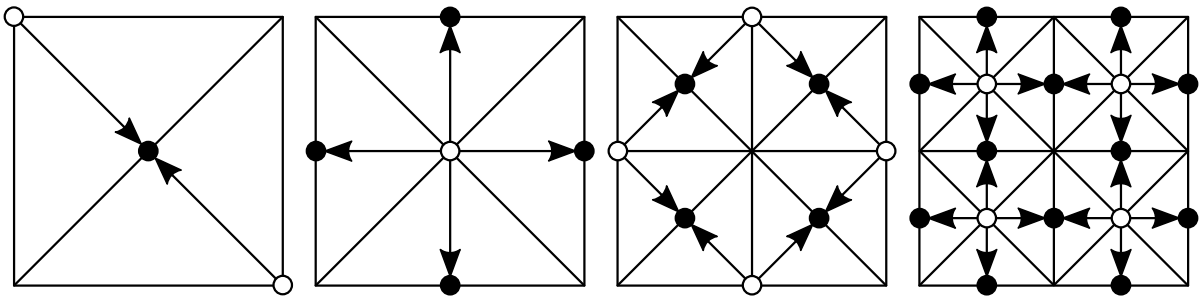


Figure 3.3.: Construction of different grid resolutions via the bisection method. Refinement vertices are marked black and their parents white. The arrows illustrate the parent-child relations.

triangles of identical size. So each grid is characterized by a fixed size of triangles and so they as well as the grid itself can be marked with a characteristical consecutive number, the so-called level. In our numbering, a higher level means a finer mesh and thus smaller triangles.

Figure 3.3 also illustrates a tree structure induced by the refinement technique. A newly inserted refinement vertex always belongs to two adjacent triangles. Boundaries can be neglected since we will transfer the method to the surface of a sphere where no horizontal boundaries exist. In this way, every refinement or child vertex is related to two triangles and has thus two parent vertices. These parent vertices have to exist in a mesh before a child vertex can be inserted. This is an important constraint for the inclusion of adaptivity as will be shown in the next section.

### 3.2.2. Terrain-Dependent Adaptivity

For a given refinement, basically two alternatives exist to get a higher accuracy in the approximation of the terrain. On the one hand, a global uniform refinement of every triangle leads to higher accuracy but with the drawback of high costs since the number of triangles double. On the other hand, with an adaptive method only those triangles are refined whose approximations of the terrain are unsatisfying in a certain mathematical sense. In this way, we can achieve the same global higher accuracy in the approximation with much less triangles than with a global uniform refinement. Therefore, we will now have a closer look at the possibility of including adaptivity in the bisection method of Section 3.2.1.

As introduced in the previous section, parent-child relations of vertices are important for an adaptive modification of the bisection technique. If a child vertex is inserted without both parent vertices already existing in the mesh, so-called hanging nodes or T-junctions appear. This is the case when two adjacent triangles with the same refinement vertex are not refined at the same time. It leads to cracks in the terrain since the triangulation is no longer continuous at this place, see Figure 3.4a.

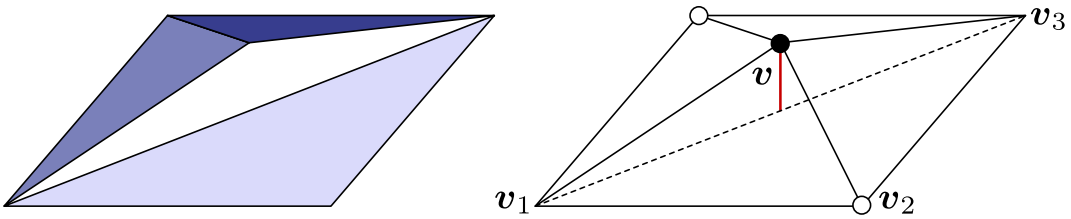



Figure 3.4.: (a) A hanging node. (b) One-level look-ahead error. The local change of height due to the insertion of the refinement vertex $\boldsymbol{v}$ is colored red.

In principle, starting with the coarsest mesh, we get an adaptive mesh by deciding for every triangle whether a refinement is necessary or not. Such a routine is called top-down since we start with the coarsest mesh and refine until a certain termination condition is fulfilled.

Let us now have a closer look at the core algorithm for the top-down construction of such an adaptive triangulation in Algorithm 1. Here, a metric defined on every triangle $T$, which will be specified later on, is the local termination condition. To avoid the above mentioned problem of hanging nodes, we now have to make special assumptions for this metric $\mu(T)$.

A first ansatz is the definition of the metric on each refinement vertex $\boldsymbol{v}(T)$, i.e. the midpoint of the triangles's longest edge, instead of each triangle $T$

$$\mu(T) = \mu(\boldsymbol{v}(T)). \tag{3.1}$$

In this way, two adjacent triangles with the same refinement vertex get the same metric and are thus always simultaneously refined. But this condition is not sufficient for the avoidance of hanging nodes since it does not require the existence of both parent vertices in the mesh before a new refinement vertex can be inserted. Therefore, a further so-called saturation condition has to be assumed

$$\mu(T) \geq \max\{\mu(\mathrm{Child}_1(T)), \mu(\mathrm{Child}_2(T))\}. \tag{3.2}$$

With this monotonicity criterion, hanging nodes in an adaptive mesh are excluded.

Now we turn to the question of the concrete construction of the metric $\mu$. In our case, the metric should represent a geometric criterion for refinement. Rough terrain should be resolved higher than flat terrain since the local change of height varies much more for a steeply sloping or ascending surface. So the metric should be a measure for the roughness of the terrain.

The insertion of a new refinement vertex leads to a local change of height in this point, that is, instead of the linear interpolation value, the exact height of this vertex is used. This difference in altitude, also called one-level look-ahead error, will serve as our

**Algorithm 1:** Top-down adaptive triangulation.

```
visit(Triangle T, Level l)
    if metric(T) ≤ ε or l = l_max then
        render(T)
    end
    else
        visit(Child1(T), l + 1)
        visit(Child2(T), l + 1)
    end
end
```

**Algorithm 2:** Bottom-up construction of a saturated metric for adaptivity.

```
assign_metric(Level l_min, Level l_max)
    for l ← l_max downto l_min do
        forall Triangles T on Level l do
            if l = l_max then
                μ(T) ← μ̃(T)
            end
            else
                μ(T) ← max{μ̃(T), μ(Child1(T)), μ(Child2(T))}
            end
        end
    end
end
```

measure for the geometric error, see Figure 3.4b.

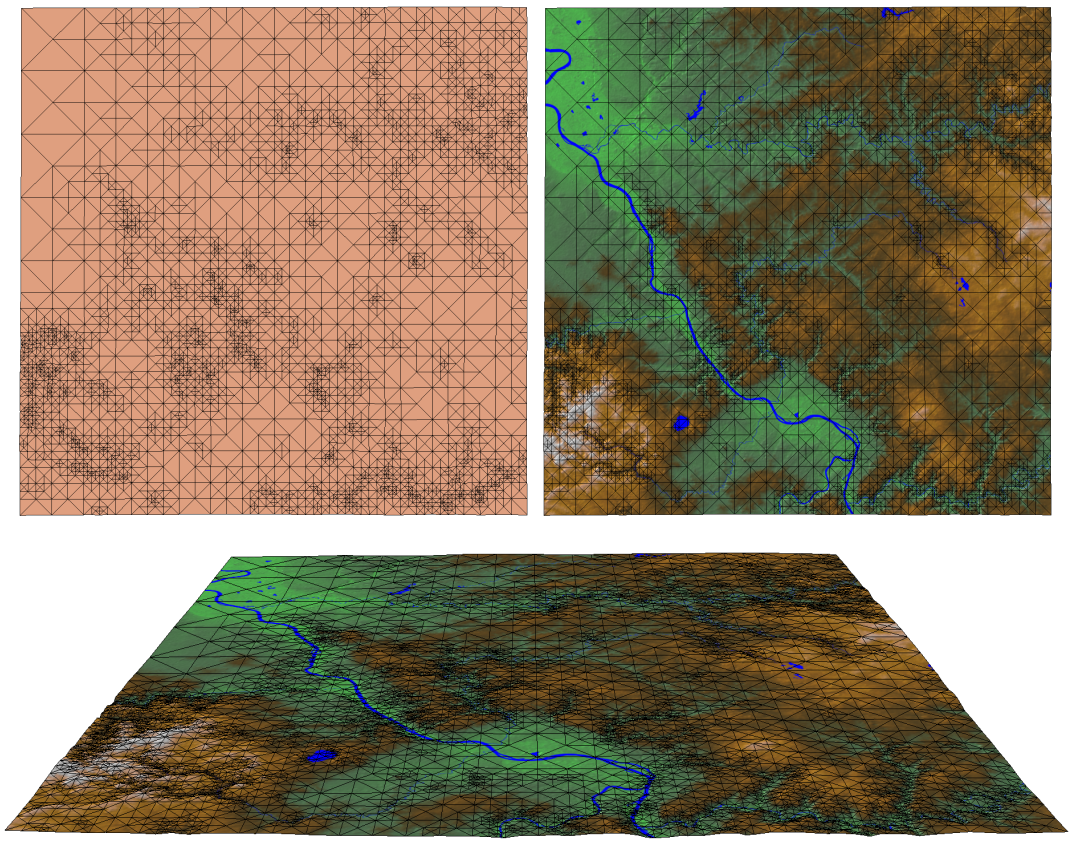
Let $\boldsymbol{v}$ be the refinement vertex of triangle $T = \Delta(\boldsymbol{v}^1, \boldsymbol{v}^2, \boldsymbol{v}^3)$. Here, $\boldsymbol{v}^1$ and $\boldsymbol{v}^3$ are the corners of the triangle which enclose the hypotenuse. The values $v_z$, $v_z^1$, and $v_z^3$ represent the height of the respective vertex. Then the metric is defined as

$$\tilde{\mu}(T) := \left| v_z - \frac{1}{2}\left(v_z^1 + v_z^3\right) \right|. \tag{3.3}$$

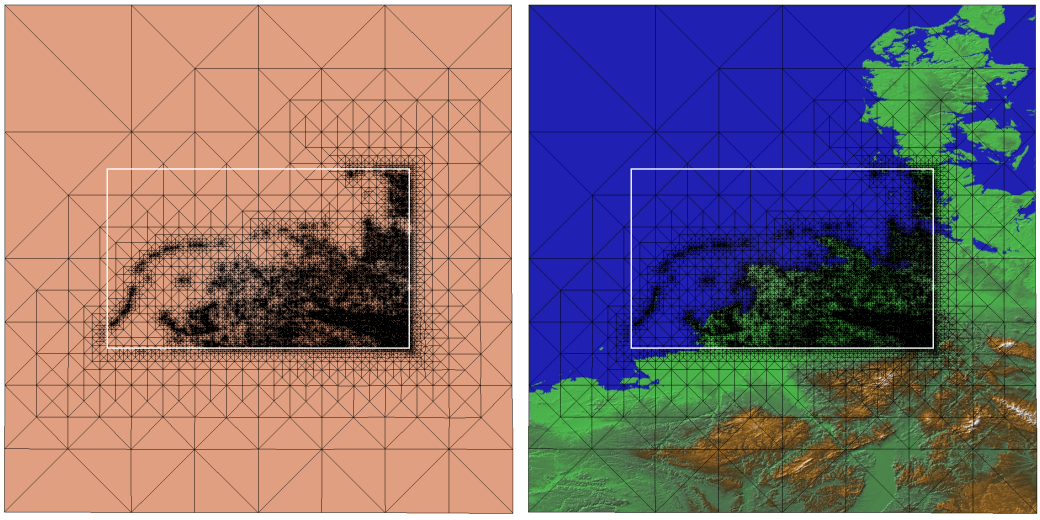
In general, this metric does not fulfill the saturation condition (3.2), but it can be corrected by a bottom-up traversing via the following formula

$$\mu(T) := \max\left\{\tilde{\mu}(T), \mu(\mathrm{Child}_1(T)), \mu(\mathrm{Child}_2(T))\right\}. \tag{3.4}$$

Bottom-up means a traversing of the mesh hierarchy from finest to coarsest level. It requires an initial global refinement to the finest level $l_{\max}$, on which we define $\mu(T) = \tilde{\mu}(T)$. In Algorithm 2, the construction of the metric $\mu$ for each triangle and each level



(a)



(b)

Figure 3.5.: Adaptive saturated meshes constructed with the bisection method. (a) Rhineland, terrain-dependent adaptivity. (b) Part of Central Europe, combination of regional adaptivity on the North Sea Coast and terrain-dependent adaptivity.

is summarized.

By using (3.3) for the computation of $\tilde{\mu}(T)$, we can construct an adaptive triangulation depending on the roughness of the terrain, see Figure 3.5a. Of course, many other metrics $\tilde{\mu}$ are also imaginable, e.g. for a visualization program, further refinements depending on the vicinity to the view point, the view direction, and the general visibility are reasonable. To this end, multiple metrics can be combined by simple algebraic combination [Ade05, Ger03b]. Furthermore, it would be interesting in our setting to resolve certain areas such as countries or continents higher than the rest of the world, which poses no problem for our described technique and can be applied straightforward, see Figure 3.5b. So, although we will use the geometric measure (3.3) in the following, keep in mind that it can be replaced with an area-dependent measure or even with a combination of these two.

### 3.2.3. Global Grid

Since we are now able to generate a uniform or adaptive triangulation on a local planar domain as described in the previous sections, we have to transfer it to the global surface of the Earth with a sphere as base.

The idea is straightforward. If a triangulation technique for planar height models of triangular or quadratic extent is known, we need to approximate the sphere through a polyhedron, preferably a convex polyhedron with regular polygonal faces, i.e. a Platonic solid. Then for each face we can revert to the known local method. Afterwards, the generated mesh at each face has to be projected to the surface of the sphere. Note that special care has to be taken at the edges of the polygons.

In this way, we get a direct transfer of a known local triangulation technique to a global domain and at the same time avoid special treatments of singularities, which would occur by using spherical coordinates for example. Note that in the global domain horizontal boundaries do not exist anymore, which will be a very important advantage for fluid dynamics later on.

Furthermore, the spreading of the vertices is still quite uniform after spherical transformation. Although the projection on a sphere distorts a uniform triangulation, the spreading can still be called nearly uniform and provides a good approximation of the sphere, especially for grids with a sufficiently small mesh width. Nevertheless, if desired, an even better spreading can be achieved if elevation data is chosen which is slightly relocated so that a projection upon the sphere would lead to a more uniform spreading. This is done by so-called spring adjustment methods based on spring dynamics [TTSG01].

Apart from that, our mesh circumvents a splitting of global and local grids as currently applied by Germany's National Meteorological Service (Deutscher Wetterdienst, DWD), which uses a highly resolved local model LM [DSB11, DFH$^+$11] of Central Europe and a coarsely resolved global model GME [MLP$^+$02], which provides the boundary conditions for LM. Such a splitting of computational domains is accompanied by serious disadvantages, particularly the violation of the conservation properties at the bound-

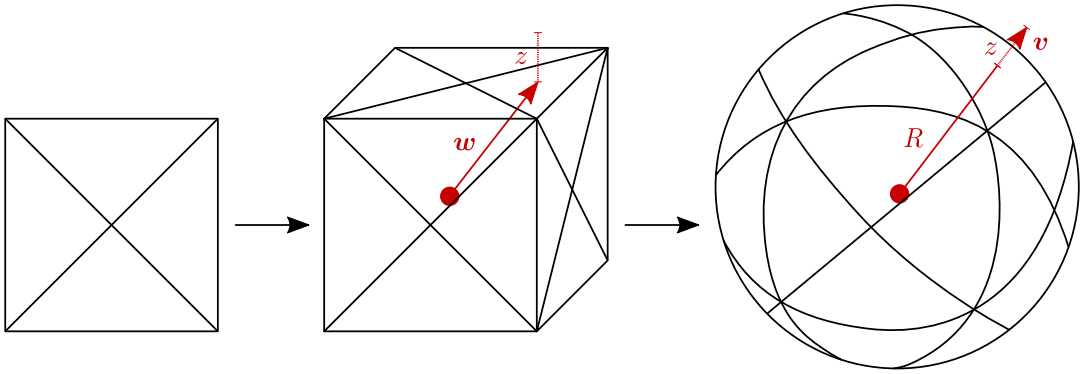



Figure 3.6.: Transfer of a local planar triangulation technique to a cube and then via projection to the surface of a sphere.

aries. Thus, our global adaptive grids have to be preferred. Currently, the DWD in cooperation with the Max Planck Institute for Meteorology (MPI-M) also develops a model which circumvents these drawbacks by constructing a global adaptive grid, the so-called ICON project [Bon04, GKZ11].

Let us now go into detail. As base polyhedron, we choose a cube, which allows for a straightforward transfer of the methods described in Sections 3.2.1 and 3.2.2. See Figure 3.6 for an illustration of the concept. The projection itself can be expressed by the following formula. To this end, let $m$ be half the side length of the cube, $R = \sqrt{3}m$ the radius of the circumscribed sphere and $\boldsymbol{w} = (x, y, m)^\mathsf{T}$ a point on the upper side of the surface of the cube with assigned height value $z$. The origin of the Cartesian coordinate system lies in the middle of the cube and the sphere. Then the projected point $\boldsymbol{v}$ of $\boldsymbol{w}$ including height $z$ is

$$\boldsymbol{v} = (R + z)\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}, \tag{3.5}$$

with Euclidean norm $\|\cdot\|$, compare Figure 3.6.

In this way, it is possible to assign each point of the global data set to a position in relation to the sphere and vice versa. So the triangulation technique can be directly transferred to a global mesh on a sphere. Note that in contrast to the given impression in Figure 3.6, which demonstrates the universal concept, we project each vertex of a horizontal grid to the sphere and then construct planar triangles between them. So we avoid dealing with curvilinear edges and faces which would lead to non-convex cells later on.

Depending on a given error bound for the refinement metric $\mu$, we are now able to generate a hierarchy of statically adaptive triangle grids of the Earth's surface. Figure 3.7c shows such test grids generated with metric (3.4). These grids have the characteristic that even with a coarse resolution of comparatively few elements, triangles of almost every level from the coarsest up to the finest appear. The different resolutions of the

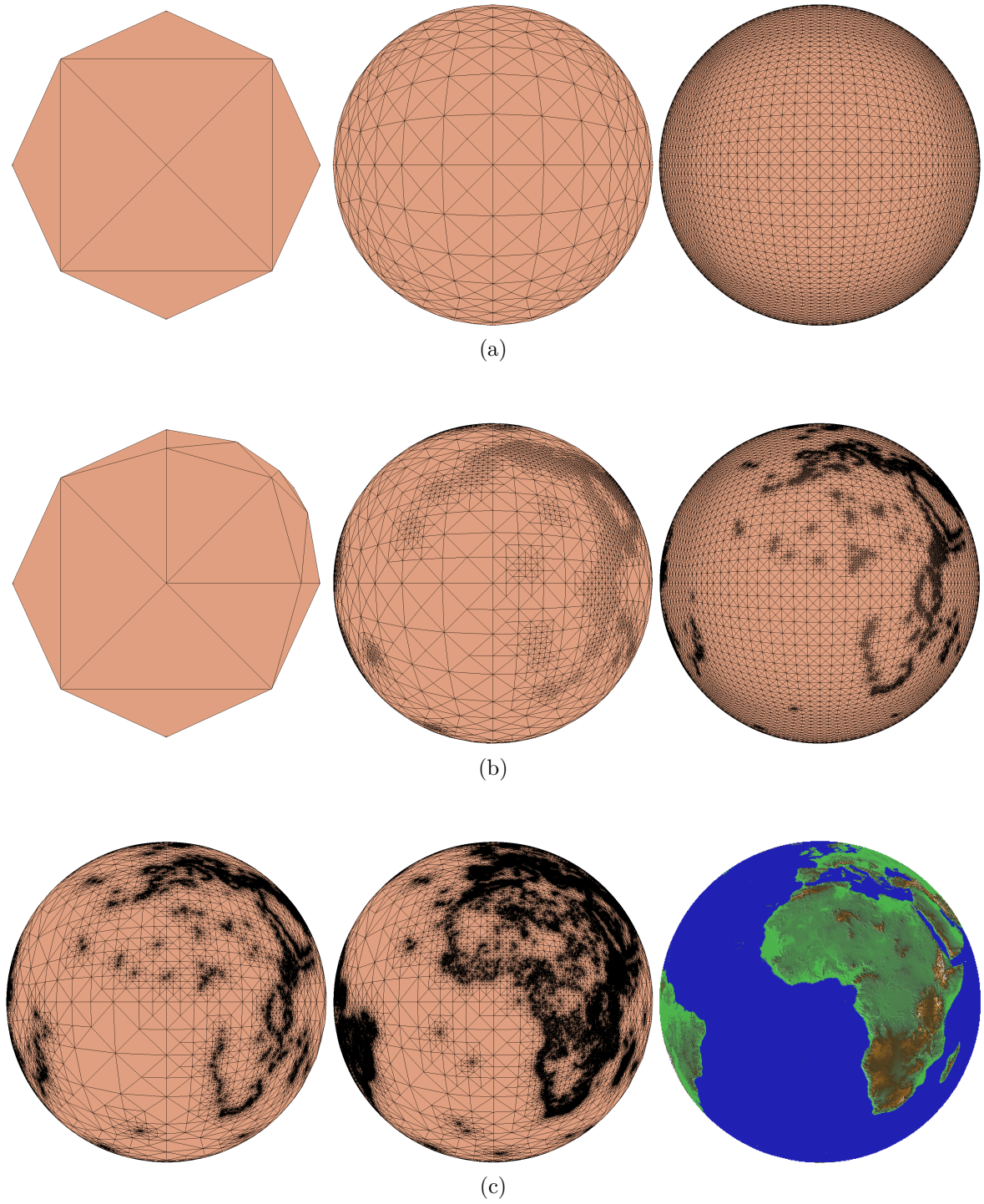


Figure 3.7.: Statically uniform and adaptive test grids of different resolutions. (a) Uniform grids. (b) Terrain-dependent adaptive grids with restriction of the range of levels to six. The statistics of these meshes are listed in Table 3.2. (c) Terrain-dependent adaptive grids without restriction of levels.

| Uniform grids of the Earth's surface | | | | | |
|---|---|---|---|---|---|
| grid | level | # vertices | # triangles | # edges | approx. mesh width |
| uniform01 | 1 | 14 | 24 | 36 | 6,036 km |
| uniform03 | 3 | 50 | 96 | 144 | 3,194 km |
| uniform05 | 5 | 194 | 384 | 576 | 1,621 km |
| uniform07 | 7 | 770 | 1,536 | 2,304 | 814 km |
| uniform09 | 9 | 3,074 | 6,144 | 9,216 | 407 km |
| uniform11 | 11 | 12,290 | 24,576 | 36,864 | 204 km |

| Adaptive grids of the Earth's surface | | | | | |
|---|---|---|---|---|---|
| grid | range of levels | # vertices | # triangles | # edges | $\mu$ |
| adaptive01 | 1 – 6 | 24 | 44 | 66 | 4,178.1 |
| adaptive03 | 3 – 8 | 80 | 156 | 234 | 3,636.5 |
| adaptive05 | 5 – 10 | 343 | 682 | 1,023 | 3,249.6 |
| adaptive07 | 7 – 12 | 1,386 | 2,768 | 4,152 | 2,475.9 |
| adaptive09 | 9 – 14 | 5,650 | 11,296 | 16,944 | 1,779.6 |
| adaptive11 | 11 – 16 | 21,854 | 43,704 | 65,556 | 1,276.6 |

Table 3.2.: Sizes of the six uniform and adaptive test grids of the Earth's surface as depicted in Figure 3.7a and 3.7b.

terrain and the depending rapid variations of triangle sizes have proven to be instability sources for simulations on such grids. Therefore, we enforce a further condition on our adaptive grids, namely a restriction on the range of levels appearing in a grid. The generated grids with a limit of six on the range of levels are depicted in Figure 3.7b. Moreover, the computational results on these meshes can be better compared to those on uniform meshes in which every triangle has the size of the coarsest or finest level of the corresponding adaptive mesh. Figure 3.7a and 3.7b show such triangulations of the Earth, and Table 3.2 lists these two sets of test grids and their associated sizes.

# 4

# Vertical Grid Generation

Now we turn toward the question of how to extend the horizontal triangulation of the Earth's surface of Section 3 to a mesh which covers the atmosphere. Based on two-dimensional triangles on the surface, we have to construct three-dimensional polyhedrons for a partition of the atmosphere into a grid of discrete cells.

For atmospheric dynamics, so-called step-mountain and terrain-following coordinates are the most common approaches, which we will present in the following and compare with the less-known cut cell approach. The superiority of the latter is illustrated in detail and accompanied by a comprehensive guideline for an implementation of cut cells into existing atmospheric codes.

## 4.1. Vertical Principle

As stated in Section 2.1.5, the relevant part of the atmosphere for our application is the area up to 24 km above the ground which covers the whole troposphere and a part of the lower stratosphere and thus encloses all the decisive weather influences.

An overview of the structure of the Earth's atmosphere is given in Figure 4.1. The different height layers are basically characterized by their alternating decreasing or increasing temperature profile. For basic orientations, typical human objects and natural phenomena are attached to the different layers at reference heights. In consideration of the logarithmic scale, we see that the weather takes place in a very small belt near the ground. The air above circa 24 km is already so thin and nearly without any water vapor that it has no significant effect on the weather anymore. In addition, we should emphasize how slim the atmosphere is in contrast to the extension of the Earth with its radius of about 6,371 km. In Figure 4.2, the correct proportion is depicted. Keep this in mind for our examination of the grid later on.

If we want to incorporate a given triangulation of the Earth's surface into an atmospheric grid by using the triangulation as its boundary mesh, the approach is evidently as follows. At first, consider a given triangulation on the sphere, i.e. the Earth without

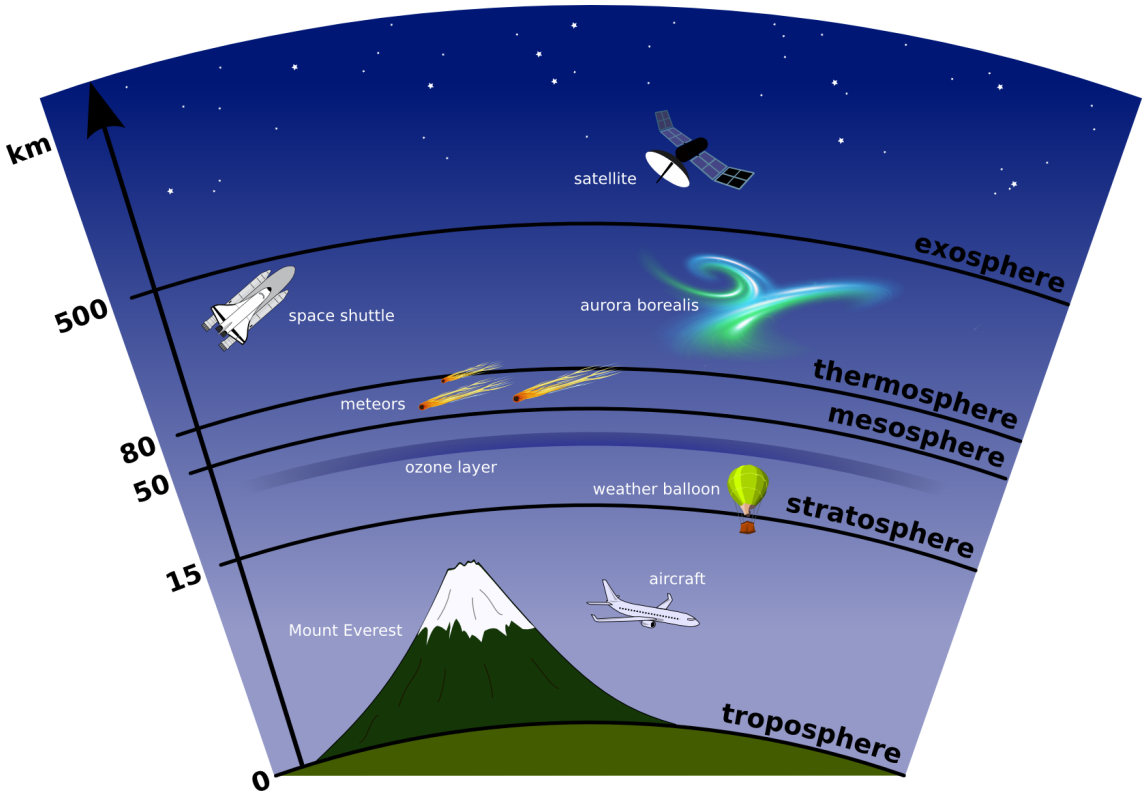



Figure 4.1.: Layers of the Earth's atmosphere in logarithmic scale and typical objects and phenomena at reference heights.

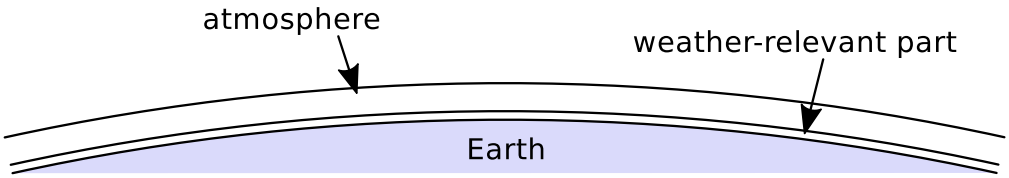
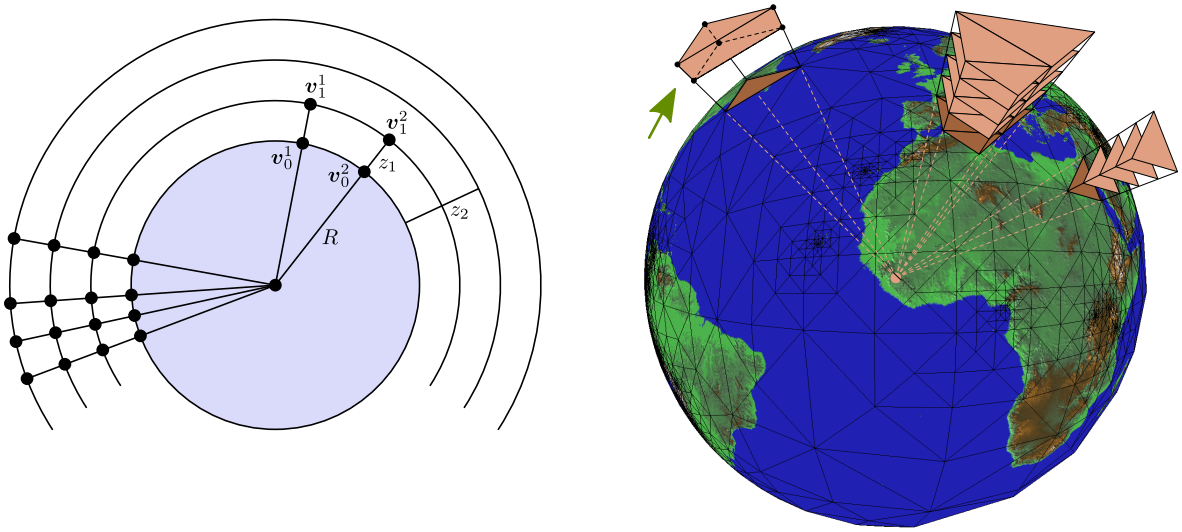



Figure 4.2.: Correct proportions of the Earth and its atmosphere up to 100 km above ground, defined as frontier to outer space (von Kármán line). The weather-relevant part up to 24 km includes the troposphere and lower stratosphere.




(a) (b)

Figure 4.3.: Principle of extending a horizontal grid on the Earth's surface via projection up into atmospheric layers. (a) Schematic view of (4.1) in 2D and (b) in 3D. This leads to truncated tetrahedral cells, a single one being marked by an arrow in (b).

any height values different from zero. Let $T = \Delta(\boldsymbol{v}_0^1, \boldsymbol{v}_0^2, \boldsymbol{v}_0^3)$ be such a triangle on the surface of the sphere with radius $R$, i.e. $\|\boldsymbol{v}_0^j\| = R \; \forall j$. We divide the atmosphere into spherical shells and radially project the vertices of the boundary mesh from the origin at the center of the sphere into these concentric layers. Formally stated, the projection of an arbitrary vertex $\boldsymbol{v}_0 := \boldsymbol{v}_0^j$ with $\|\boldsymbol{v}_0\| = R$ into layer $i$ with absolute altitude $z_i$ is achieved by

$$\boldsymbol{v}_i = (\|\boldsymbol{v}_0\| + z_i)\frac{\boldsymbol{v}_0}{\|\boldsymbol{v}_0\|} = \frac{R + z_i}{R}\boldsymbol{v}_0 = \left(1 + \frac{z_i}{R}\right)\boldsymbol{v}_0. \tag{4.1}$$

This principle is illustrated in Figure 4.3. In this way, we construct polyhedral cells of the form depicted in the upper left of Figure 4.3b, that is truncated tetrahedrons with parallel triangular faces.

So far, this procedure leads to a three-dimensional triangulated spherical shell around a sphere which represents a flat Earth without any terrain information. The next step is the incorporation of the height values of the Earth's surface into such a generated grid. In Section 3, we already described the two-dimensional triangulation of the Earth's surface. But the transfer of this terrain structure to the layers, or generally speaking to the vertical coordinate direction, is not well-defined. We rather have a free choice of how to fit the concentric layers to the surface structure of the Earth. Each choice leads to a special grid with different characteristics, especially regarding the shape of the resulting cells.

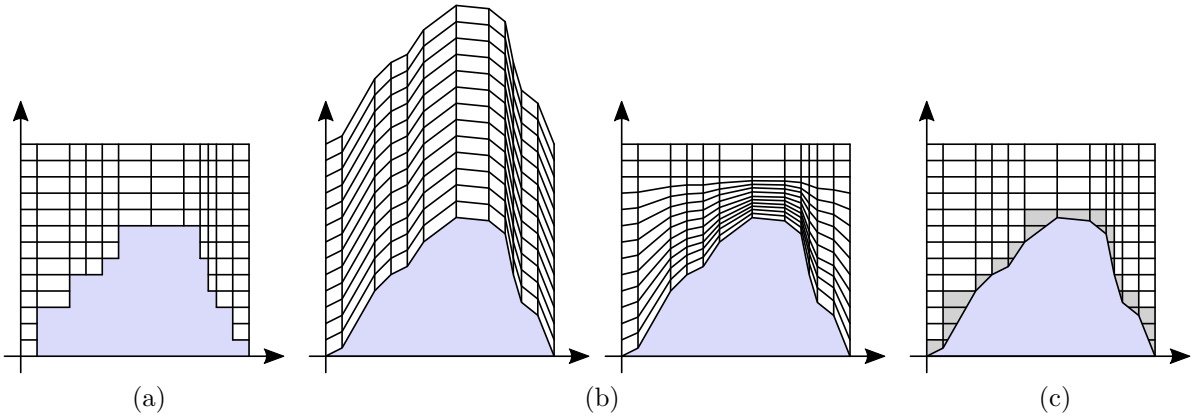



Figure 4.4.: Two-dimensional illustrations of different vertical coordinates over terrain. (a) Step-mountain approach, (b) terrain-following approach without and with vertical decay, and (c) cut cell approach.

In atmospheric and oceanic modeling, basically three vertical principles are used, each of them with various subtypes:

- the step-mountain approach,
- the terrain-following approach, and
- the cut cell approach.

A two-dimensional impression of the different vertical coordinates is given in Figure 4.4. At first sight, the main differences of these principles are obvious. The *step-mountain* or sometimes also called *$\eta$-coordinates* generate an orthogonal grid with exclusively rectangular cells. This causes terrain being represented as staircases, compare Figure 4.4a. *Terrain-following* or so-called *$\sigma$-coordinates* follow the curvature of the terrain whereby the originally rectangular cells are sheared, compressed, or stretched, as can be seen in Figure 4.4b. *Cut cells* are also called *shaved cells* in meteorological literature whereas the first term is predominantly used in the areas of mathematics, computer science, and engineering regarding complex geometries. Cut cells produce an orthogonal grid with boundary cells which are cut by the terrain, leading to deformed and occasionally very small cells at the Earth's surface, see Figure 4.4c. So we already note the main differences of these types of vertical coordinates.

## 4.2. Step-Mountain Approach

Due to their unrealistic representation of terrain via staircases, we will not consider step-mountain coordinates in the following. However, we summarize some of their characteristic advantages and disadvantages. For more details, we refer to [MJN$^+$88, Gal00, AC04] and the references therein.

Originally used in oceanic dynamics [Bry69, SM77] and later on adopted for atmospheric modeling as an alternative to the widely-used terrain-following coordinates, the step-mountain approach stands out due to its simplicity. Each atmospheric layer is horizontal or at least quasi-horizontal, depending on whether its exact location is chosen height-based or pressure-based. Thus, all cells are (quasi-)rectangular and each boundary segment either horizontally or vertically aligned, which is of great benefit for the stability of the discretization scheme. Moreover, such an orthogonal mesh do not suffer from the so-called pressure gradient force error, which will be addressed in detail in Section 4.3 in the context of terrain-following coordinates.

But apart from these favorable characteristics, step-mountain coordinates also involve some serious disadvantages. First of all, terrain is represented through staircases, so its approximation is of zeroth order. Especially for smooth topography, this is a very bad representation since all elevation changes occur as cliffs. Studies showed that the steps generate spurious disturbances concentrated in particular above step corners, which demand some kind of model damping for compensation.

A second even more significant effect of step-mountain coordinates is a constant underestimation of downwind flow on the lee side of mountains. This is due to the blocking effect of the steps for downward wind. It has been shown that this effect worsens exceedingly as horizontal resolution becomes increasingly fine. So the step-mountain approach leads to the following discrepancy. On the one hand, the approach is only reasonable in the limit of very high resolution because of the bad representation of topography, but on the other hand, fine resolutions lead to serious errors concerning the wind fields at ground level. Therefore, we neglect the step-mountain coordinates in the following.

## 4.3. Terrain-Following Approach

As the name already suggests, terrain-following coordinates follow the curvature of the underlying terrain. That is, spherical atmospheric layers are deformed by the topography of the Earth's surface. First introduced in [Phi57], this intuitive approach has become very popular in atmospheric modeling in the last decades and is still widely-used as the standard method in nearly all present weather forecast systems.

The motivation of introducing these coordinates was the advantageous effect that their lowest surface coincides with the orography. Figure 4.4b depicts the terrain-following principle in two dimensions and gives an impression of the two basic variants which we will examine in this thesis.

In principle, due to their popularity, various subtypes of terrain-following coordinates exist. These subtypes differ in the applied vertical decay functions which define the way of smoothing of the terrain-deformed atmospheric layers leading to spherical concentric layers in the upper part of the atmosphere. In Figure 4.4b, the difference between terrain-following coordinates without vertical decay and linear vertical decay with height are shown. The latter version is the most commonly used form and is also known as

$\sigma$-coordinate. Another major subtype are hybrid pressure-based coordinates [SB81] with constant pressure instead of constant height defining the atmospheric layers. These are particularly used in hydrostatic models and usually based on local pressure data adapted from measurements.

A further version with exponential decay of terrain features is offered by the so-called SLEVE (smooth level vertical) coordinates [SLF+02]. Here, a scale-dependent vertical decay and thus a non-local coordinate transformation is used so that small-scale topographic features decay much faster with height than their large-scale counterparts. Thus, SLEVE coordinates lead to a much smoother vertical grid structure than the other terrain-following principles. But although they suffer less from common disadvantages connected with the slope of coordinate surfaces, they still have the same problems typical for terrain-following approaches as will be seen in the following discussion.

So let us have a closer look at the advantages and disadvantages of terrain-following coordinates in general.

### 4.3.1. Advantages

First of all, the construction of a three-dimensional mesh with terrain-following vertical coordinates is quite straightforward with a given horizontal triangulation of the Earth's surface. Each grid point at the surface has to be radially projected up to a certain extent which depends on the vertical decay function and on the spacing of the atmospheric layers themselves. Since the polygons – or in our case the triangles – of the horizontal triangulation are already boundary faces of the lowest cell layer, such a projection directly leads to a three-dimensional mesh with truncated tetrahedrons as exclusive type of cells. Therefore, this construction is completely analogous to the described principle for a flat Earth as in Figure 4.3.

Suppose $\boldsymbol{v}_0$ is an arbitrary vertex of the horizontal grid on ground level, so its length can be split into $\|\boldsymbol{v}_0\| = R + H$ with Earth's radius $R$ and local topographic height $H = H(\boldsymbol{v}_0)$ above mean sea level. Note that height values above mean sea level may also be negative since the Earth's topography varies between $-418\,\mathrm{m}$ near the Dead Sea and 8,850 m at the top of Mount Everest. Again, let $\boldsymbol{v}_i$ be the corresponding vertex at layer $i$, and $z_i$ denotes the height of layer $i$ above mean sea level in the case of $H = 0$.

For terrain-following coordinates, (4.1) becomes

$$\boldsymbol{v}_i = (\|\boldsymbol{v}_0\| + z_i D(z_i, H)) \frac{\boldsymbol{v}_0}{\|\boldsymbol{v}_0\|} = \left(1 + \frac{z_i D(z_i, H)}{R + H}\right) \boldsymbol{v}_0, \tag{4.2}$$

where the decay function $D$ depends on the chosen scheme. Without any vertical decay, as in the left part of Figure 4.3b,

$$D(z_i, H) \equiv 1. \tag{4.3}$$

For the decay depicted in the right part, we define a boundary layer at height $Z$ above

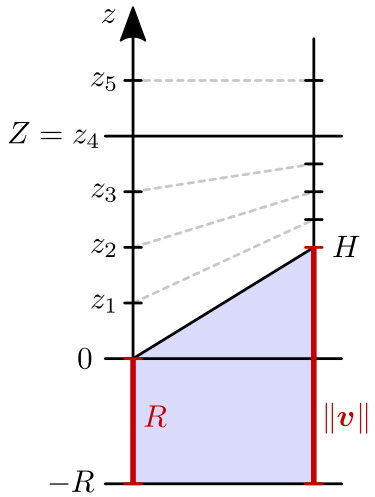



Figure 4.5.: Principle of linear vertical decay. The layers divide line segment $[0, Z]$ at the same ratio as they divide segment $[H, Z]$.

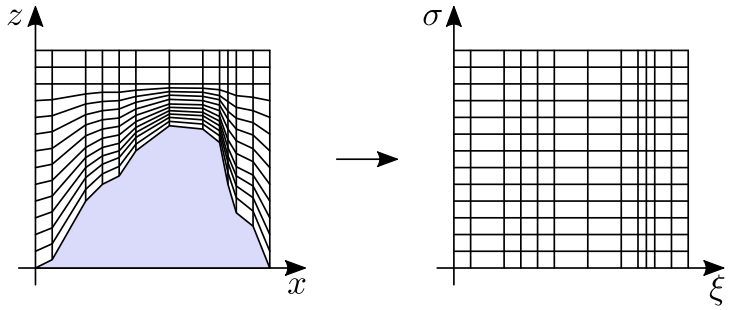



Figure 4.6.: Two-dimensional mapping from physical space $(x, z)$ to terrain-following space $(\xi, \sigma)$.

which the atmospheric layers should be spherical again. The idea of the shown linear decay is that, for a vertex with $H = 0$, layer $i$ with $z_i < Z$ divides the line segment $[0, Z]$ at the ratio of $z_i/Z$ and should divide segment $[H, Z]$ for a vertex with $H \neq 0$ by the same amount. This leads to the decay function

$$D(z_i, H) = \begin{cases} \frac{Z-H}{Z} & \text{if } z_i < Z \\ 1 & \text{if } z_i \geq Z \end{cases} \tag{4.4}$$

for the right part of Figure 4.3b. A schematic view of this construction is depicted in Figure 4.5.

This plainly constructed atmospheric grid implies a further advantage. Since the lowest level surface in a terrain-following setting coincides with the orography, a straightforward mapping upon a rectangular mesh is possible. More precisely, as it is illustrated in Figure 4.6, every vertex of a terrain-following mesh can be bijectively mapped upon a

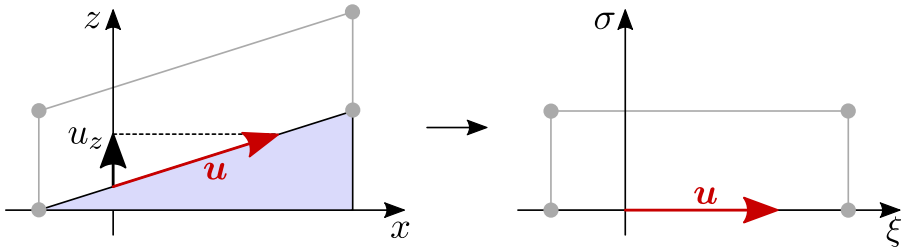



Figure 4.7.: Vertical velocity $u_z$ vanishes after mapping from physical space $(x, z)$ to terrain-following space $(\xi, \sigma)$.

regular rectangular grid. This is an important advantage, in particular if you desire the application of Finite Differences as discretization scheme.

Simultaneously, such a mapping leads to a simplification of the lower boundary condition. Figure 4.7 visualizes this circumstance. After transformation to terrain-following space, the vertical direction coincides with the normal direction of the terrain and so the vertical velocity vanishes at the level of topography and thus the boundary. Moreover, such a rectangular alignment of the lower boundary and the possibility of an unequal spacing of atmospheric levels simplify the coupling of dynamics with boundary and surface-layer parameterization schemes [SLF+02].

However, apart from these convenient properties, terrain-following coordinates also involve some serious disadvantages.

### 4.3.2. Shift of Difficulty

At first, let us bring to mind that the simplification of the boundary condition and the mesh structure through a coordinate transformation is gained by a shift of the computational difficulty to the modeling. Thus, such a tranformation simplifies the structure of the grid but simultaneously leads to more complex modeling equations since additional coordinate transformation terms are required. These terms add to the complexity of the discretization and solution part of the problem in a significant way.

So a coordinate transformation is always connected with a drawback. And since we will use Finite Volumes as discretization scheme, which can easily cope with unstructured meshes as opposed to Finite Differences, we would not benefit from the rectangular mesh structure in an essential way. Moreover, the advantage would rather be outweighed to a great extent by the complexity of the transformed modeling equations.

### 4.3.3. Pressure Gradient Force Error

The second most serious problem of terrain-following coordinates is the so-called pressure gradient force error, which is a long-known numerical discretization error already observed in [Kur68, Phi73, Jan77].

For illustration of this error-prone effect, let us have a closer look at the horizontal pressure gradient in a two-dimensional example. The transformation relations from physical space $(x, z)$ to terrain-following space $(\xi, \sigma)$ in the left case of Figure 4.4b, that is without vertical decay, reads

$$\begin{pmatrix} \xi \\ \sigma \end{pmatrix} = \begin{pmatrix} x \\ z - H(x) \end{pmatrix} \tag{4.5}$$

with the local topographic height $H(x)$. Then the horizontal pressure gradient can be written as

$$\begin{aligned} \frac{\partial p}{\partial x}(\xi, \sigma) &= \frac{\partial p}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial p}{\partial \sigma}\frac{\partial \sigma}{\partial x} \\ &= \frac{\partial p}{\partial \xi} + \frac{\partial p}{\partial \sigma}\frac{\partial}{\partial x}(z - H) \\ &= \frac{\partial p}{\partial \xi} + \frac{\partial p}{\partial \sigma}\left(-\frac{\partial H}{\partial x}\right) \\ &= \frac{\partial p}{\partial \xi} - \frac{\partial H}{\partial \xi}\frac{\partial p}{\partial \sigma} \end{aligned} \tag{4.6}$$

with the use of (4.5) and the chain rule for partial derivatives. Analogously, as depicted in the right case of Figure 4.4b, with an additional vertical decay function $D(x)$ and thus the mapping

$$\begin{pmatrix} \xi \\ \sigma \end{pmatrix} = \begin{pmatrix} x \\ (z - H(x))/D(x) \end{pmatrix} \tag{4.7}$$

the horizontal pressure gradient can be generally transformed into

$$\frac{\partial p}{\partial x}(\xi, \sigma) = \frac{\partial p}{\partial \xi} - \left(\frac{\sigma}{D}\frac{\partial D}{\partial \xi} + \frac{1}{D}\frac{\partial H}{\partial \xi}\right)\frac{\partial p}{\partial \sigma}. \tag{4.8}$$

So the horizontal pressure gradient consists of a sum of two terms, first the gradient of pressure along a constant $\sigma$-surface and the second involving the gradient of bottom topography.

Now consider the extreme case of a resting fluid which is in hydrostatic equilibrium, that is the vertical pressure gradient is balanced with the gravitational force

$$\frac{\partial p}{\partial z} = -\rho g. \tag{4.9}$$

Then the horizontal pressure gradient vanishes and so the two terms on the right hand side of (4.8) have to cancel. But especially near steep topography, these terms are large and opposite in sign as illustrated in Figure 4.8. This circumstance is quite error-prone and generally leads to a serious numerical truncation error due to large non-cancellations

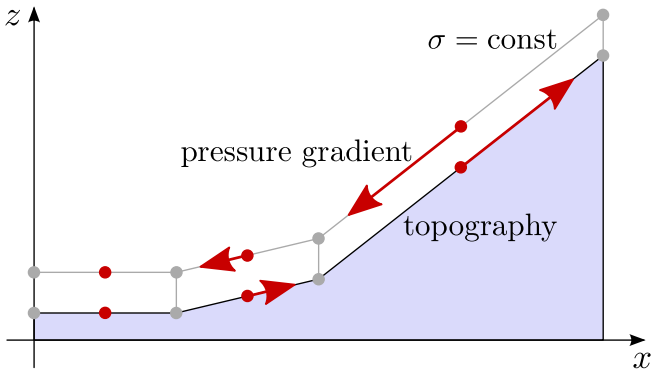



Figure 4.8.: Opposite direction of the pressure gradient along a constant $\sigma$-surface and the gradient of bottom topography for a resting fluid in hydrostatic balance.

of the two terms when evaluated using e.g. Finite Differences or Finite Volumes. This generates spurious accelerations in the horizontal which generally distort the forecast in an unacceptable way [Lan95, Han91, Sun76].

Moreover, the development in computer capacity and the associated tendency to use finer and finer meshes exacerbate the pressure gradient force error even more. This is due to the fact that cells tend to be steeper for finer spacings. And with steeper topography the discretization error from the horizontal pressure gradient terms increases more and more, as it is analytically shown in [BH93].

Many remedies have been proposed to mitigate this error, but none of them with the capability to eliminate the problem in a generally acceptable way. The first idea is based upon reducing the size of the two terms in (4.8) – and hence the truncation error – by subtracting a horizontally homogeneous reference profile from pressure in advance, first suggested in [Phi73, Gar73]. But this ansatz is only reasonable for limited areas and not for global domains where pressure varies too much [Han91, SBMB02]. Apart from that, a substantial error can still remain, see [BH93, Sun75] for analytical arguments.

Another remedy suggested and examined by [BH93, FB96] consists in computing the horizontal pressure gradient in Cartesian coordinates while maintaining $\sigma$-coordinates for any other quantity. The computation of the horizontal gradient in Cartesian coordinates requires interpolation of all necessary values in $\sigma$-coordinates. It is shown that this approach leads to smaller truncation errors but still introduces significant numerical diffusion near the bottom and reduces the stable parametric region in which it can be applied.

An alternative approach for reducing the error is a higher-order approximation of the pressure gradient [McC94]. Such an increase in the order of discretization produces improved results though not a complete elimination of the error, along with all the practical difficulties typically involved in higher-order schemes.

In summary, none of these basic approaches avoids the pressure gradient error in a both satisfying and general way.

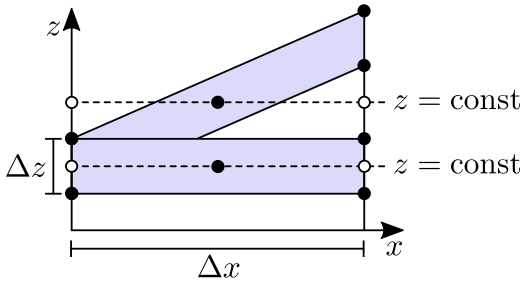



Figure 4.9.: Hydrostatically consistent (front) and inconsistent cell (behind) where vertical interpolation turns into extrapolation for points along constant $z$-surfaces; compare white points.

### 4.3.4. Hydrostatic Inconsistency

A further disadvantage of terrain-following coordinates, which may be intertwined with the previous one, is caused by so-called hydrostatic inconsistency. The term was first introduced in [Jan77] but is actually misleading as emphasized in [FB96]. It does not indicate a numerical inconsistency but rather an incoherent treatment of certain quantities whose approximations are needed at different stages in the computational process. If these quantities are inconsistently approximated, a substantial error occurs. Note that this error is an additional one, not to be mistaken for the discretization error of the pressure gradient force [MJ85, Han91].

To understand the source of the hydrostatic inconsistency error, we reconsider the splitting of the horizontal pressure gradient force in a sum of two terms (4.8). As it was first observed in [Jan77, Mes82], the numerical evaluation of these terms can be done in a hydrostatically inconsistent way. In the mentioned publications, several inconsistent examples are given, all of them having in common that different schemes are applied for the treatment of these two terms or of other interconnected terms. Typically, for repeated occurences of e.g. the hydrostatic equation or of quantities such as geopotentials, different discretizations or extrapolations are used, e.g. space centered schemes versus uncentered extrapolation schemes or the substitution of either average or correct values of the same quantity.

For clarification, let us assume in one step values of geopotentials at constant $\sigma$-surfaces are needed and later on at constant pressure- or $z$-surfaces. The chosen approximation scheme in the first step defines a special vertical profile of geopotential, for instance a piecewise linear function of $\xi$. For a hydrostatically consistent procedure it is now important to choose a scheme in the second step which yields values lying on the geopotential profile defined by the approximation of the first step [MJ85]. So for preserving hydrostatic consistency it is crucial to choose, either explicitly or implicitly, a consistent treatment of recurring terms in the computational process.

But this is only one part of the conditions for hydrostatic consistency. As pointed out in [Jan77, Mes82], even hydrostatically consistent schemes in the above sense do

not preserve their consistency in cases of very steep slopes and thin atmospheric layers. Problematic are cases as seen in Figure 4.9, where the evaluation of e.g. geopotential along constant $z$-surfaces ends up at points outside of the considered cell so that vertical interpolation turns into extrapolation, see also [SM03]. Such situations occur when the slope of $\sigma$-surfaces is too steep compared to the thickness of the vertical layer. Hence, both increasing the steepness and increasing the vertical resolution may lead to a violation of the hydrostatic consistency of the chosen scheme.

In order to express a criterion for hydrostatic consistency, let $\Delta x = \Delta\xi$ be the global or local horizontal mesh width and $\Delta z$ and $\Delta\sigma$ the analog vertical mesh sizes in Cartesian and terrain-following coordinates, respectively. With these notations, the so-called hydrostatic consistency condition is introduced in the context of oceanic dynamics in [Han91]

$$\left|\frac{\sigma}{D}\frac{\partial D}{\partial x}\right|\Delta x < \Delta\sigma \tag{4.10}$$

with the local ocean depth $D$ and $\sigma = z/D$. This coordinate transformation is a special case of our more general setting in (4.7) and (4.8) with $H \equiv 0$ and a special function $D$. For a generalization of the hydrostatic consistency condition (4.10), we need to postulate analogously for (4.7) with $H \neq 0$ and $D$ being the decay function

$$\left|\frac{\sigma}{D}\frac{\partial D}{\partial x} + \frac{1}{D}\frac{\partial H}{\partial x}\right|\Delta x < \Delta\sigma. \tag{4.11}$$

For this inequality, a practical reformulation is preferable by using

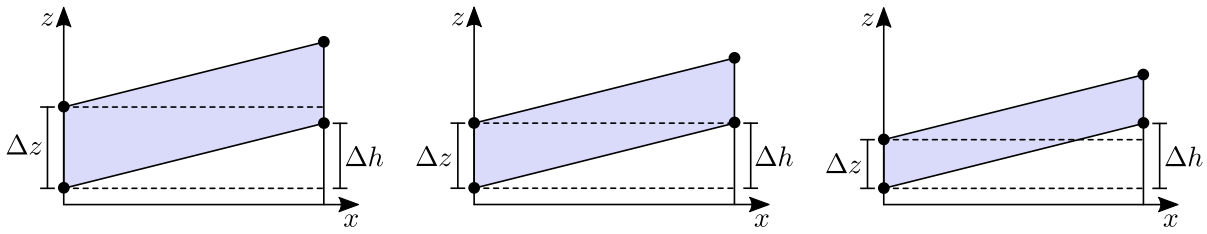
$$\frac{\partial H}{\partial x} \approx \frac{H(x_{k+1}) - H(x_k)}{\Delta x}$$

in a discrete sense, defining $\Delta h$ as the change of orography between one grid point and the next

$$\Delta h := \left|H(x_{k+1}) - H(x_k)\right|,$$

and assuming that the decay function $D$ is positive and independent of $x$. Then, the postulation can be written as

$$\begin{aligned}
&\left|\frac{\sigma}{D}\frac{\partial D}{\partial x} + \frac{1}{D}\frac{\partial H}{\partial x}\right|\Delta x < \Delta\sigma \\
\Leftrightarrow\quad &\left|\frac{1}{D}\frac{H(x_{k+1}) - H(x_k)}{\Delta x}\right|\Delta x < \Delta\left(\frac{z-H}{D}\right) \\
\Leftrightarrow\quad &\frac{1}{D}\frac{\left|H(x_{k+1}) - H(x_k)\right|}{\Delta x}\Delta x < \frac{1}{D}\Delta(z-H) \\
\Leftrightarrow\quad &\left|H(x_{k+1}) - H(x_k)\right| < \Delta z - \Delta_z H \\
\Leftrightarrow\quad &\Delta h < \Delta z
\end{aligned} \tag{4.12}$$




(a) $\Delta h < \Delta z$ (b) $\Delta h = \Delta z$ (c) $\Delta h > \Delta z$

Figure 4.10.: Examples of fulfillment and violation of the hydrostatic consistency condition $\Delta h < \Delta z$ (4.12).

with $\Delta_z H = \Delta_z H(x) = 0$. In particular, this hydrostatic consistency condition (4.12) also holds for mapping (4.5) where $D \equiv 1$. An equivalent formulation to our derived condition can be found in [SM03] in terms of cell vertices.

So a given vertical mesh increment demands a sufficiently small horizontal mesh width. Or the other way around, for a given horizontal mesh increment, a sufficiently large vertical mesh size is required as depicted in Figure 4.10.

The hydrostatic consistency condition is for instance postulated in the local model LM of Germany's National Meteorological Service (Deutscher Wetterdienst, DWD), where every cell has to satisfy (4.12) [SBMB02]. But generally, the postulation for gaining hydrostatic consistency is very restrictive and violated in most current models in operation, particularly in global models. Furthermore, the condition is even more difficult to hold for highly resolved meshes since smaller cells tend to be even steeper due to the underlying terrain.

Loosely speaking, the steepness of cells – also often called shear or skewness – causes hydrostatic inconsistency. If the skewness exceeds a certain degree in connection with high anisotropy of the cells, numerical errors induce serious stability problems.

Evidently, such inconsistent cells can be corrected in three different ways. As shown in Figure 4.11, the alternatives are a sufficient decrease in the horizontal mesh width, an increase in the vertical mesh size both lessening the anisotropy of the cell, or a decrease in the slope of the terrain and thus the shear of the cell. The latter is not applicable since it requires an unacceptable artificial change of orography. So the only remedy remains in the reduction of anisotropy which means a horizontal refinement of the mesh if we imply that a vertical coarsening should be avoided.

However, due to the extreme anisotropic extension of the atmosphere with a difference of three orders of magnitude regarding horizontal and vertical extent, compare Figure 4.2, a forced horizontal refinement to gain more isotropic cells is connected with a substantial and often impracticable computational increase. Therefore, hydrostatic inconsistency of terrain-following coordinates, particularly with regard to global atmospheric models, is an important and still unresolved problem.

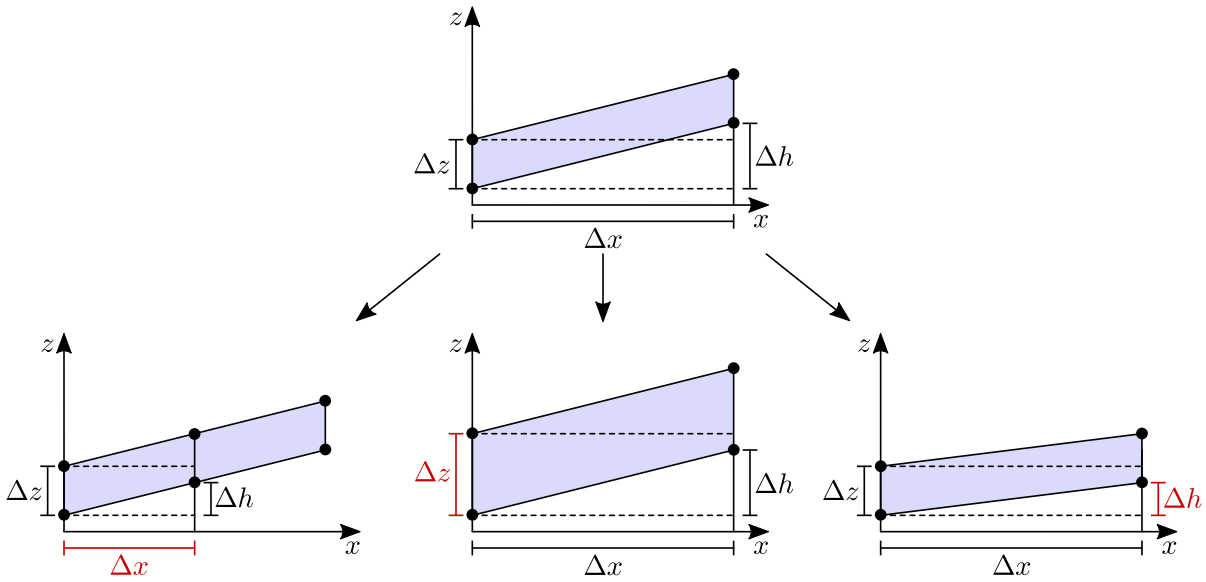



Figure 4.11.: Three possibilities of correcting a hydrostatically inconsistent cell (top): decreasing $\Delta x$ and thus $\Delta h$, increasing $\Delta z$, or decreasing $\Delta h$ (left to right).

### 4.3.5. Validations

These theoretical considerations can be validated by our data sets of the actual surface of the Earth listed in Table 3.2 combined with terrain-following coordinates as depicted in Figure 4.4b. First of all, we can state that the slight differences between simulations using terrain-following meshes with and without vertical decay are negligible. The advantages of regular cells at the top of smoothed grids are compensated by more anisotropic and thus even steeper cells near the ground, compare Figure 4.4b. Even the above mentioned SLEVE coordinates [SLF+02] with their exponential vertical decay suffer from the same decisive problem although in a slightly lesser way.

Furthermore, test runs with different resolutions of terrain reveal that simulations with coarse meshes tend to be more stable than those with fine meshes, which is a confirmation of the theory that finer meshes include steeper cells and thus suffer from the pressure gradient force error and the hydrostatic inconsistency in a more crucial way.

One mentioned – albeit unrealistic – remedy would be a decrease in the shear of cells by flattening the underlying terrain. Experiments indicate that an overall division of real height values by at least 1.5 leads to grids with stable simulation runs. Thus, if the terrain of the Earth would be flatter by one third, as it is illustrated in Figure 4.12, terrain-following coordinates would be unproblematic for atmospheric simulations. But of course, a distortion of orography to such great extent is unrealistic and entirely unacceptable.

So obviously, the only alternative for dealing with severely steep cells is the construction of less anisotropic cells. In further experiments based on the actual terrain of the

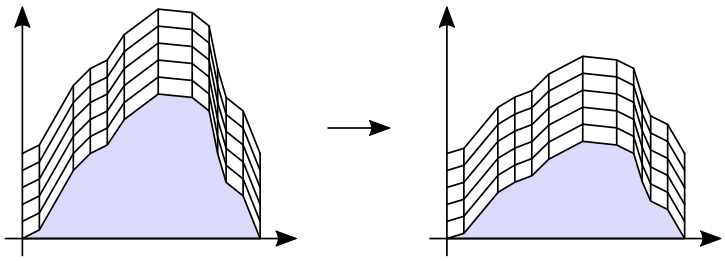


Figure 4.12.: Flattening of real terrain by one third results in hydrostatic consistency and thus in stable simulations for terrain-following coordinates.

| # of layers | spacing vertical | spacing horizontal | approx. # of cells (dof) |
|---|---|---|---|
| 1 | 24 km | 240 km | 9,000 |
| 2 | 12 km | 120 km | 70,000 |
| 4 | 6 km | 60 km | 570,000 |
| 8 | 3 km | 30 km | 4,500,000 |
| 16 | 1.5 km | 15 km | 36,000,000 |
| 32 | 0.75 km | 7.5 km | 290,000,000 |
| 64 | 0.375 km | 3.75 km | 2,300,000,000 |

Table 4.1.: Approximate number of required atmospheric cells (and thus degrees of freedom) estimated via (4.13) for one order of magnitude difference in vertical and horizontal spacing.

Earth, we constructed more isotropic cells by increasing the vertical as well as decreasing the horizontal mesh width according to Figure 4.11. Test runs with these grids show that a stable simulation is only possible when the difference between horizontal and vertical mesh width is at the most one order of magnitude. Larger differences lead to unstable simulations.

One order of magnitude difference in horizontal and vertical spacing is a very restrictive result, considering that the extent of the Earth's atmosphere in horizontal and vertical direction differs by three orders of magnitude with a surface area of about 510 million $\mathrm{km}^2$ (circumference of about 40,000 km, respectively) and a weather-relevant height of 24 km. More specifically, the restriction results in a great number of cells – identifiable with degrees of freedom (dof) – even for low vertical resolutions. In Table 4.1, we receive an impression of the rapid increase of cells depending on the number of vertical layers which restrict the horizontal mesh width. Here, the approximate number of cells $c$ for uniform

grids is being estimated by the formula

$$c = 4l\frac{510{,}000{,}000}{(\Delta x)^2} \tag{4.13}$$

as a function of the number of layers $l$, the horizontal spacing $\Delta x$ in km and the above mentioned surface area of the Earth in km$^2$. The factor 4 has its origin in the bisection method, compare Section 3.2.1.

Summing up, the promising remedy for the drawbacks of terrain-following coordinates by lessening the anisotropy of cells leads to a vast amount of cells for reasonable vertical resolutions, which exceeds by far today's computer capacities for a short-term weather forecast.

## 4.4. Cut Cell Approach

After the discussion of two well-known types of vertical coordinates in atmospheric dynamics, we introduce in this section a less-known third variant, the so-called cut cell approach. Cut cells combine the advantageous orthogonal grid of step-mountain approaches with the better resolution of orography achieved by terrain-following coordinates. The idea consists in a Cartesian grid covering the entire computational domain, and the terrain cutting parts out of the cells at the boundary, see Figure 4.4c.

Depending on the application, this mesh generation principle is known under different names and is frequently referred to as cut cell or shaved cell method, volume-fraction technique, Cartesian grid method, and embedded boundary or immersed boundary method.

Although known as a common technique for mesh generation of complex geometries for over three decades, cut cells found their way into oceanic and atmospheric dynamics only recently. The cut cell approach was originally developed in computational fluid dynamics for potential flow [PB79] and later on applied to the Euler equations [CSH86, LeV88a, BL89] as described in [ICM03]. A decade later, [AHM97] introduced cut cells for oceanic models with Finite Volume discretization and thus provided the basis for a transfer to atmospheric models, which were entirely dominated by terrain-following approaches so far. In [SBMB02] first tests of atmospheric flow over gently sloping terrain were successfully performed, followed by flow over realistic mountains in [SBJ$^+$06].

Nevertheless, the application of cut cell techniques in today's weather forecast systems is still pending. But Germany's National Meteorological Service (Deutscher Wetterdienst, DWD) in cooperation with the Max Planck Institute for Meteorology (MPI-M) has made an effort to develop a next generation global model for weather and climate forecasting based on an icosahedral grid and cut cells, the so-called ICON project [BKG$^+$04, Bon04, GKZ11]. Its first operational use was originally expected for 2013 but is still pending.

So although there are some convincing advantages to cut cell approaches, their spread-

ing took quite a while. A closer look at the advantages and difficulties might reveal possible reasons.

### 4.4.1. Advantages

Constructing a mesh for flows through, in, or around complex geometries can be arbitrarily difficult. In this context, the generation of unstructured meshes or rectangular body-fitted grids, which are comparable to our terrain-following approach, often proves to be non-universal and thus time-consuming and manpower-intensive because of the adjustments necessary for each special application.

So especially in industrial processes, a relatively automated and flexible mesh generation technique is desirable. Regarding these requirements, cut cell methods stand out due to their underlying Cartesian grid. The vast majority of the grid consists of regular Cartesian cells which are simple to construct and allow for a solver in its Cartesian and thus simplest possible form. Moreover, also the irregular cut cells can be straightforwardly constructed by basic geometric intersections.

This important advantage renders cut cell techniques interesting for a large area of applications. Not only the simulation of flows with complex shapes, moving boundaries, or topological changes, but also adaptive mesh refinements benefit from the straightforward construction, and thus complicated remeshing strategies are no longer necessary [YMUS99, KBN09].

Since we generate our fixed atmospheric grid as a preprocessing step, another advantage is of even more significance for our application. Cartesian grids imply horizontal mesh surfaces; so horizontal and vertical gradient evaluation separate naturally. With this separation no pressure gradient force error exists, one of the main difficulties of terrain-following coordinates, compare Section 4.3.3. Moreover, hydrostatic inconsistency, another important disadvantage of terrain-following approaches, see Section 4.3.4, poses no problem for horizontally aligned cells as well.

A further advantage of Cartesian grids is the avoidance of coordinate transformation terms so that the simplicity of the governing equations is maintained. Actually, the simplicity and universality of Cartesian grids are the true reasons for all of their favorable advantages.

### 4.4.2. Construction

Cut cells are regular cells which intersect the piecewise linearly approximated orography of the Earth. Thus, they vary in shape and have reduced volumes as well as certain reduced or vanishing face areas. The construction of these boundary elements is more complicated than the straightforward way of defining the remaining Cartesian cells or terrain-following cells. But nevertheless, their computation needs to be processed only once and involves basic geometric intersections. In Section 4.7, we will describe the construction process of cut cells for our atmospheric setting in detail.

With this process in mind, it is understandable that twofold opinions exist about wether or not the construction of cut cells is advantageous. On the one hand, the construction can be considered as a disadvantage, especially for those working with terrain-following environments, since it involves major modifications to existing terrain-following codes and parameterization packages, as stated e.g. in [SLF+02]. But on the other hand, it is undeniable that the construction process of cut cells is an advantage because of its universality and flexibility. The process can be efficiently automated and combined with adaptive mesh refinement, moving boundaries, and complicated geometries [LeV88a, YMUS99, KBN09]. Moreover, the construction itself is not complicated at all, if, as a start, a special case distinction is accomplished, compare Section 4.7.

### 4.4.3. Vertical Resolution

A further twofold aspect of the cut cell approach is the vertical resolution near the ground. In atmospheric models, a few kilometers above ground are typically higher resolved than the remaining domain. In terrain-following coordinates, an increase of resolution over the whole topography is simply done by a closer spacing of the grid layers and thus, only a few additional layers are necessary. For a Cartesian mesh, additional layers have to be spread over the whole range of topographic height, that is from $-418\,\mathrm{m}$ to $8{,}850\,\mathrm{m}$ above mean sea level. So a higher vertical resolution is much more expensive.

But, as argumented in [WA08], the expense is partially counterbalanced by the elimination of underground cells and the greater efficiency of horizontal gradient computation. Therefore, a cut cell model can have more layers than a terrain-following model while maintaining the same computational cost. Apart from that, higher resolution near the ground influences the quality of the terrain-following approach in a negative way because the pressure gradient force error and the hydrostatic consistency suffer from more anisotropic and sheared cells, compare the arguments in Section 4.3.

So far, we discussed several aspects of Cartesian cut cell methods, all of them advantageous or at least with no serious drawbacks. But there is one essential issue which has to be approached with care and treated in a suitable way, the so-called small cell problem.

### 4.4.4. Small Cell Problem

In a cut cell approach, the geometry cuts parts of the underlying Cartesian mesh out and leaves boundary cells of different shapes and sizes. Thus, a cut cell can be arbitrarily small as shown in Figure 4.13. We know from the Courant-Friedrichs-Lewy criterion [CFL28] that the time step necessary for a stable explicit simulation process depends on the smallest cell of the grid, compare also Section 5.3.3. For an arbitrarily small cell, the used time step has to be arbitrarily small, too, and thus the computation process takes an arbitrarily long time. This impracticable restriction is called the small cell problem, which has to be circumvented.

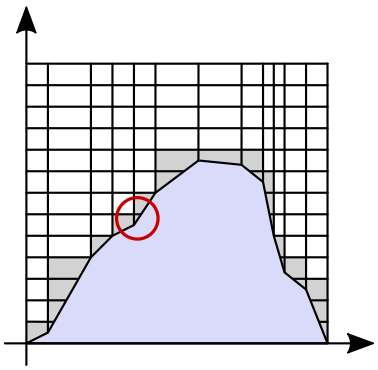


Figure 4.13.: Small cell problem: The gray colored cut cells can get arbitrarily small, see for example the encircled one.

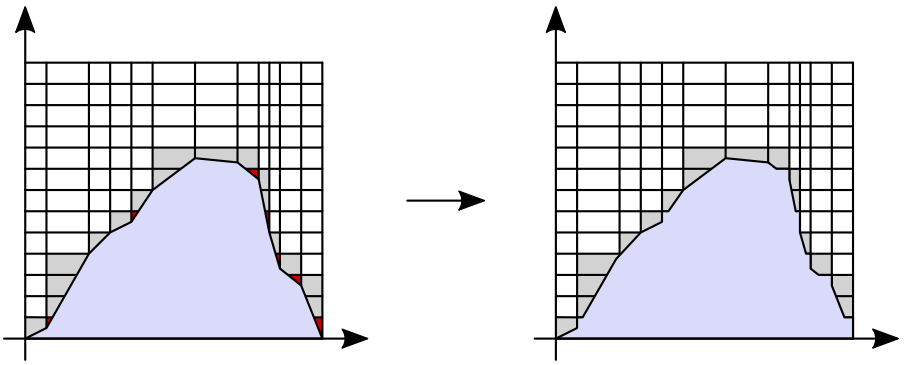


Figure 4.14.: Small cell remedy: Limitation of the minimum volume to half the size of the smallest regular cell. Smaller cells are discarded, i.e. filled with solid.

We will now give a thorough review of possible remedies proposed in the literature, which we feel is lacking in this comprehensive form.

#### Limitation of the Minimum Volume

The first apparent ansatz is a limitation of the minimum volume as used in e.g. [Adc95, AHM97, MAH+97]. Here, a threshold for the allowed size of cut cells is imposed, usually about half the size of the smallest regular cell, and every cell smaller than the threshold is discarded, i.e. filled with solid. As compensation, [Noh64] suggests to redistribute the mass and internal energy of a neglected cell to its neighbors. For an illustration of the principle, see Figure 4.14. Apart from filling small fluid cells with solid, it is also possible to empty correspondingly small solid parts of cells which generates new regular cells without a change in the order of magnitude of the error. This variant is depicted in Figure 4.15 and used in [Adc95].

In this way, tiny steps are added to the landscape, which cause small perturbations of the flow field with possibly wider influence on the solution in sum. [Noh64] for instance states fluctuations in the flow of a few percent. Moreover, the variant with filling and

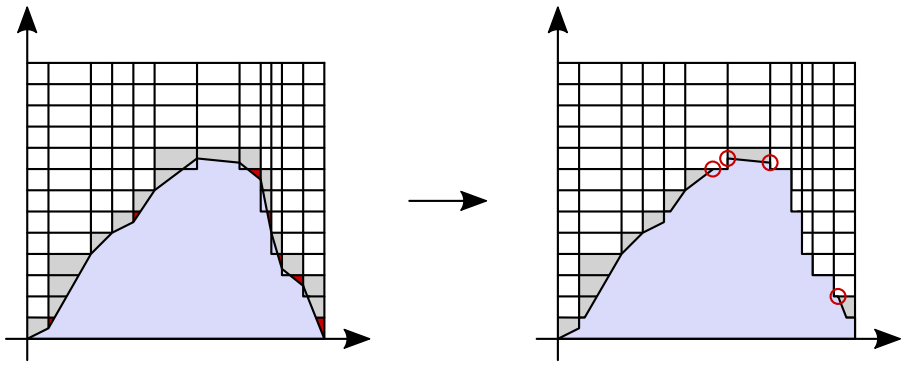


Figure 4.15.: Small cell remedy: Fill or empty a cut cell if its fluid or solid part is less than half the size of the smallest regular cell. Hanging nodes are encircled red.

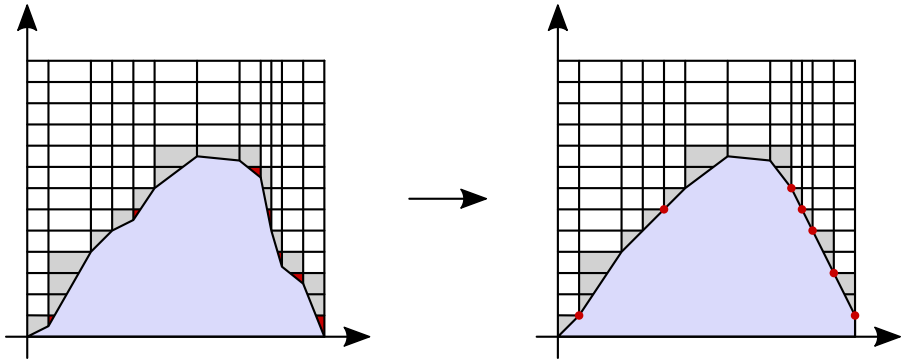


Figure 4.16.: Small cell remedy: Lifting or lowering of terrain nodes to Cartesian node locations so that small cut cells are eliminated. Shifted terrain nodes are marked by red dots.

emptying generates even more steps and leaves also hanging nodes in the mesh which have to be dealt with in a suitable way.

To avoid these tiny steps, a further variant of the limitation principle exists, where certain terrain nodes are shifted to near regular grid nodes such that small cells are eliminated. Also referred to as “iso-line lifting” in [Coi94], this ansatz results in smooth geometry without tiny steps at the expense of a greater overall change of the terrain. Note that lifting and lowering of terrain nodes may generate new small cells nearby and thus result in an iterative procedure if not even a vicious circle. See an example in Figure 4.16.

So all of these limitation approaches are connected with a change of the given geometry which influences the flow field especially near the boundary – a circumstance rather to be avoided for atmospheric applications. But nevertheless, this approach is a straightforward way of overcoming the small cell problem.

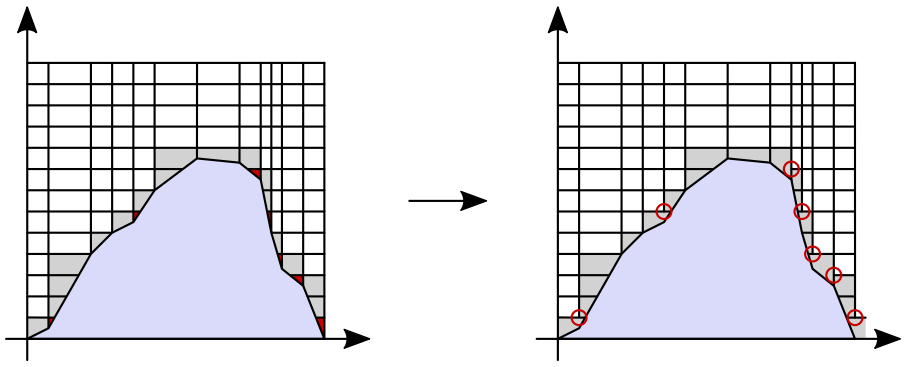


Figure 4.17.: Small cell remedy: Merging of small cells with their neighbors, which are chosen in direction of the boundary normal. Hanging nodes are encircled red.

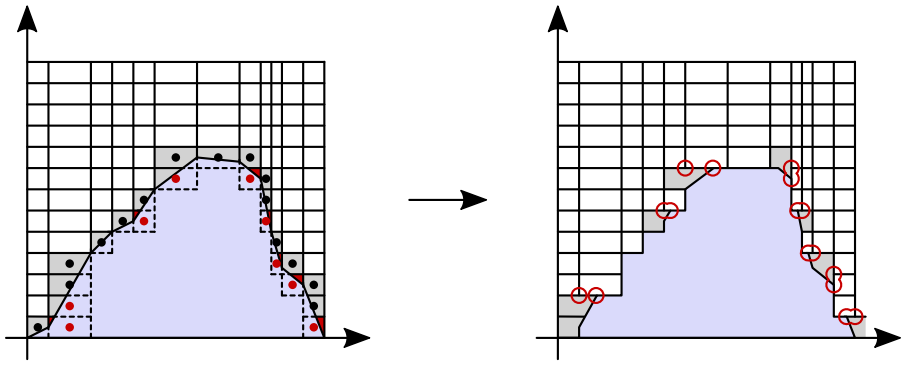


Figure 4.18.: Small cell remedy: Combination of discarding the solid part if the cell center is in the fluid (black dots) and merging with a neighbor if the center is in the solid (red dots). Hanging nodes are encircled red.

### Merging

A second idea consists in merging small cells with larger neighbor cells. In this way, the approximation of the boundary is maintained, i.e. the terrain does not need to be altered. Mentioned already in [Noh64] for moving boundaries, the idea was taken up by [CSH86, CvLP92, Qui94] and later on used extensively, especially in recent years [ICM03, YS08, YS10].

Small cut cells with a size less than half of the size of the smallest regular cell are merged with an adjacent cell, which is chosen in the direction of the boundary normal. So cells are combined horizontally on steep slopes and vertically on gentle slopes. The principle is illustrated in Figure 4.17. As depicted, merging of adjacent cells in a Cartesian mesh inherently leads to hanging nodes and thus to a problem on the geometric side. But apart from that, not only the geometry, but also the values defined in the center of each cell have to be merged, i.e. interpolated in a suitable way. A detailed description of the interpolation procedure can be found in [Qui94, CvLP92]. But gen-

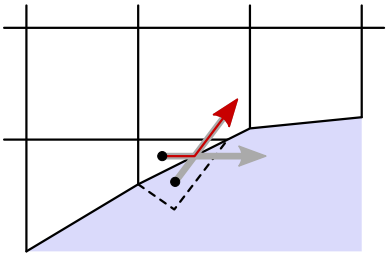


Figure 4.19.: Small cell remedy: A wave propagating out of the computational domain is matched by a reflected one, see [LeV88a].

erally, this interpolation is the main drawback of the merging ansatz. As emphasized in [WA08], the merging approach is impractical in the context of atmospheric dynamics since particularly the pressure varies that much even for adjacent cells, which still lie tens or hundreds of kilometers apart, that the values of neighboring cells cannot be readily combined.

Nevertheless, the ansatz is popular for many other applications of embedded boundary methods and thus a lot of variants exist. In [KAK03, HMS08] for instance, the method is transformed into and combined with, respectively, a cell-linking stategy where two cells are linked as master/slave pair. The slave cell's center is relocated to the master's center, while each cell still remains a distinct entity for the computation of certain values such as pressure, fluxes, or interpolation factors. This strategy circumvents problems associated with other merging approaches, such as a significant increase in complexity and a problematic formulation of a systematic merging algorithm in three space dimensions.

The two main hitherto described approaches of limitation and merging can also be combined. In [YMUS99], a distinction is drawn between cut cell centers located in the fluid or in the solid region. Those cells with a center in the fluid are reshaped to regular cells by discarding their solid parts, while cells with a solid center are combined with their neighbors, see Figure 4.18. This ansatz considerably reduces the amount of cut cells but also generates several hanging nodes as well as a quite rough approximation of the terrain with a lot of artificial staircases of various sizes. In this way, all the problems of step-mountain coordinates arise, compare Section 4.2.

### Wave Propagation

A further small cell remedy of completely different type is the wave propagation approach developed in [LeV88a, LeV88b, BL89]. The idea is to overcome the restriction of the time step due to the Courant-Friedrichs-Lewy criterion by allowing information to propagate through more than one cell in a single time step. To this end, boundary cells together with their variable values are reflected at the boundary to the outside of the domain. In the following simulation, the boundary is ignored and the computational domain extended. Due to the reflection of velocity, any flow or "wave" which leaves the original domain during a time step is matched by a mirrored wave, compare Figure 4.19.

The approach provides good results in one dimension but already for two dimensions

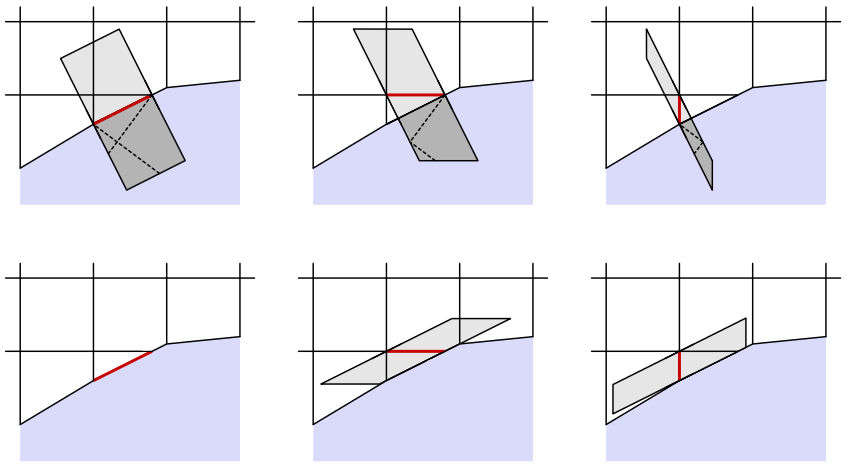


Figure 4.20.: Small cell remedy: Construction of $h$-boxes for each small cell face (red) in normal (top) and tangential (bottom) boundary direction, see [HBL05]. Dashed lines denote reflected cells beyond the computational domain.

occasional instabilities occur, which are observed but not explained in [BL90]. Furthermore, it is an expensive approach since necessary weights depend on the flow variables and thus need to be computed repeatedly. This is why a new approach is suggested.

#### $h$-Box Method

The $h$-box method [BL90, HBL05], is closely related to the wave propagation approach. Again, reflected cells beyond the boundary of the computational domain form the basis of the algorithm. The core consists of boundary-aligned boxes of regular cell length $h$ which are constructed for each small cell face, see Figure 4.20. As a preprocessing step, the intersections of the so-called $h$-boxes with regular cells are computed and thus area-weighted averages of all variables are assigned to the $h$-boxes. Based on these interpolated values, approximate fluxes at small cell faces can finally be evaluated.

Here, in contrast to the wave propagation approach, the interpolation weights can be preprocessed, i.e. have to be computed only once. Furthermore, the cancellation property of the fluxes of a single cell is confirmed in 2D and a better stability is gained, but nevertheless, cases are observed where “the results are not very accurate” [BL90].

#### Flux Difference Redistribution

Now we turn toward another small cell remedy which consists of flux difference redistribution and is originally based on ideas for shock tracking methods [CC87, PBC+95, ABCM97, CGKM06]. As a first step, a reference state is computed by treating all cut cells as regular cells; a variant of this ansatz can be found in [FJ98]. Secondly, a stable but non-conservative correction for cut cells is made. Redistributing the resulting conservation error, i.e. the difference in the mass increments, to nearby cells finally leads to a conservative scheme. The crucial point of this approach lies in the computation of the reference state whose error influences the quality of the correction and thus the resulting

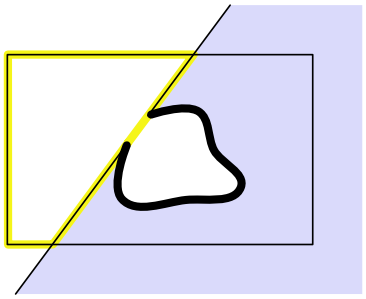


Figure 4.21.: Small cell remedy: Thin-wall approximation by increasing the cut cell volume (boldly framed) while the sizes of the reduced face areas remain (yellow), as constructed in [Hir93].

dynamics. As already admitted in [PBC+95], "poor choices for the reference state can clearly result in gross inaccuracies".

All of the last three described methods, that is wave propagation, $h$-boxes, and flux difference redistribution, originate from the context of the dynamics of two-dimensional Euler equations around complex geometries. In this area, cut cell methods were established much earlier than in our field of application, the dynamics of the atmosphere. Nevertheless, we now turn to an ansatz which is one of the rarely found cut cell approaches already applied to the three-dimensional Euler equations for atmospheric settings.

### Thin-Wall Approximation

The thin-wall approximation is based upon the interpretation of the small cell problem as distorted surface to volume ratio. For small cut cells, the ratio of their face areas to their cell volume is generally much greater than for regular cells. Numerically, this point of view is equivalent to reduced grid spacing and thus results in a restrictive CFL condition. As a remedy, [Hir93, Bon00, SBMB02, BS04] construct a correction of the distorted ratio by simply increasing the cell volume to an acceptable level without consistently changing the face areas. This approach is quite rude and may be interpreted as poking virtual holes in the terrain as shown in Figure 4.21. While generally stable, the method produces unsatisfactory solutions for some cases as pointed out in [WA08], where two detailed atmospheric examples are given.

In fact, the term "thin-wall" approximation does not refer to the small cell remedy itself but to a further assumption made to achieve that the cut cell's barycenter remains in its regular place. This is desirable because the regular cell center is located at the same height as its lateral neighbors and the horizontal and vertical gradients are separable. Therefore, as in [Bon00, SBMB02, BS04], it is additionally assumed that cut cells are regular but "hollow" cells. That is, the loss of volume due to the terrain part is uniformly distributed throughout the regular cell as if it were a porous medium. The same assumption is made for the faces of the cell which are regarded as differently permeable depending on the submerged fraction of each area. The principle is illustrated in Figure 4.22.

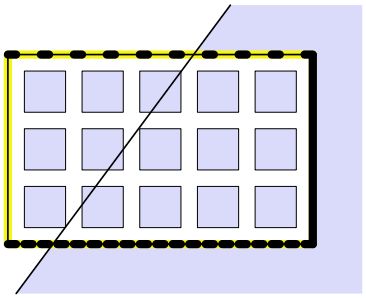


Figure 4.22.: Reduced cell volume (white) and reduced face areas (yellow) of a cut cell, as assumed in [WA08]. The reduction is uniform throughout the regular volume (squares) and over each face area (disrupted bold lines).

Naturally, such an indistinct shape of topography has impact on the flow dynamics near the boundary. This is of less consequence in oceanography where this representation is commonly used, whereas it should be only carefully applied in atmospheric dynamics since atmospheric flow near terrain is of special interest.

Note that the assumption of porous boundary cells on its own does not overcome the small cell problem. In [WA08], both the cut cell volume and the face areas keep their reduced sizes although they are uniformly distributed over the whole regular cell. So the small cell problem remains unaffected, compare Figure 4.22. In [WA08], due to the erroneous simulation of certain flow situations with the thin-wall approximation, the following remedy is preferred albeit not yet implemented.

#### Implicit Flux Balance

The implicit flux balance is a small cell remedy described by [LeV97, CL00]. The ansatz is based upon the wave propagation approach and also related to the flux difference redistribution. The idea consists in treating small cells as regular cells with small "capacity". As described in detail in [CL00], a so-called capacity function models that some cells are only partially available to the fluid as e.g. in a porous medium. In this way, the small cell problem is shifted from cells with small volume to cells with small capacity and thus the instability problem is overcome. In this respect, as reasoned in [WA08], it is essential that the inward and outward fluxes of a cell are tied to one another and thus implicitly balanced, so that the flow can traverse the entire small cell volume in less than one time step. The method is fully conservative, more precisely, the total mass is conserved by increasing the capacity of small cells if necessary. Thereby, inconsistency is unavoidable in this approach since artificial alterations of volumes are required for certain cells after all, as pointed out in [YS10].

#### 1D Flux Stabilization

Another small cell remedy by [KBN09] is based on 1D flux stabilization. Applied to two-dimensional atmospheric dynamics, the ansatz fits into a dimensionally split framework where each coordinate direction is solved independently by a fully explicit Riemann

solver. To overcome the restriction on the time step, the one-dimensional fluxes of cut cells are stabilized by extending their influence to the entire regular cell length. Beforehand, each cut cell face and thus each flux is decomposed into regions that are "shielded" and "unshielded" by the boundary to account for the necessity of representing irregular cut cells in one dimension. The approach is conservative but, as also stated in [KBN09], of limited suitability for practical atmospheric simulations since fully explicit schemes are inefficient due to the additional limitation of the time step by the speed of sound.

#### Asymptotic Approach

For the sake of completeness, a two-term asymptotic approach can also be used as small cell remedy, at least for elliptic equations with variable coefficients and discontinuities across an embedded interface, see [OSK09]. Such an asymptotic approach is best known from multiscale modeling. Here, the small cell problem is overcome by postulating the splitting

$$\boldsymbol{u}^{\pm} = \boldsymbol{u}^{\pm,0} + \varepsilon \boldsymbol{u}^{\pm,1}, \quad \varepsilon > 0$$

for the solutions $\boldsymbol{u}^{\pm}$ on both sides of a cut cell face, each provided with a specially chosen local coordinate system. The method is robust and allows for small cells down to machine accuracy, though it is of limited suitability for our hyperbolic atmospheric setting.

#### Fully Implicit Solution

Finally, we must not omit a completely different way of overcoming the small cell problem. As we know, the problem itself arises from the CFL criterion which restricts the time step due to explicit solution methods. By constructing a fully implicit solution procedure, no limitations on the time step exist any more and so the small cell problem is overcome – at the expense of a much more complicated and time-consuming solver which is much harder to implement. Such an ansatz is pursued by e.g. [MAB03] in the context of moving boundaries in a Cartesian grid.

Now we had a thorough review of the possibilities to circumvent the small cell problem of cut cell methods. As pointed out, not all of these remedies are suited for atmospheric applications or are attached with other drawbacks. So a thorough consideration of the different choices is essential. Regarding our framework as described in Section 5 and postulating that we wish to avoid any inconsistencies, the most straightforward ansatz of limitation of the minimum volume is the most promising. This assumption will further be justified by an examination of the mesh quality in the following section.

By summing up the state-of-the-art in cut cell methods, we received the impression that a citation of Nikos Nikiforakis in the year 2009 is still up-do-date. He stated in [Nik09] that although "there are cut-cell codes currently in use", the approach is "not mature" and "at the forefront of advanced research in universities and national laboratories".

## 4.5. Mesh Quality

Before summing up the advantages and disadvantages of the different types of vertical coordinates, certain measures of mesh quality have to be introduced in order to adequately compare the different approaches. In the literature, many measures can be found, see e.g. [TSW99] and the references therein, but they are usually fitted for special kinds of grids. So we are bound to construct our own appropriate measures for the atmospheric meshes we generated so far in Section 3 and 4.

As defined in [Fie00], we aim at so-called fair measures which have the following characteristics.

1. Completeness: The measure detects all degenerate cells.

2. Non-dimensionality: Similar cells have the same measure.

3. Boundedness: The measure cannot get arbitrarily large.

4. Normalization: The measure is positive between zero and one.

With this definition in mind, we construct some measures referring to important aspects of our specially generated grids.

### 4.5.1. Anisotropy

First of all, anisotropy is an essential characteristic of cells discretizing the Earth's atmosphere. As already emphasized at the beginning of Section 4 in Figure 4.2, the atmosphere is so thin in contrast to the extension of the planet's surface that the cells have to be anisotropic in order to be manageable by reasonable computer resources. We do not expect great differences in anisotropy for our vertical approaches but it is nevertheless an important aspect of the grids.

To construct a fair measure of anisotropy, the ratio between inscribed and circumscribed sphere of each cell has to be determined. The radius of the circumscribed sphere is defined as the maximum distance of the cell's barycenter to a corner. For the inscribed sphere of a convex polyhedron, the center can not be determined in such a straightforward way and generally differs from the barycenter, compare [SL04]. But nevertheless, it is a good approximation for our types of cells, so that we analogously define the radius of the inscribed sphere as minimum distance of the barycenter to a face, compare Figure 4.23.

So let $\boldsymbol{c}$ be the barycenter of cell $\mathcal{C}$ with corners $\boldsymbol{v}_i$, $i = 0, \ldots, n-1$, defined by

$$\boldsymbol{c} := \frac{1}{n} \sum_{i=0}^{n-1} \boldsymbol{v}_i. \tag{4.14}$$

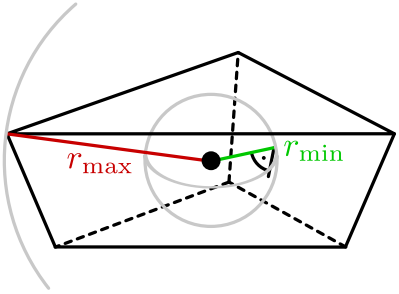



Figure 4.23.: Measure of anisotropy: Ratio of radii $r_{\max}$ and $r_{\min}$ of circumscribed and inscribed sphere as defined in (4.15) and (4.17).

Then the radius $r_{\max}$ of the circumscribed sphere is

$$r_{\max}(\mathcal{C}) = \max_i \{\|\boldsymbol{c} - \boldsymbol{v}_i\|\}\,. \tag{4.15}$$

For the inscribed sphere, we first construct Hesse's normal form of each face $F$

$$F = \left\{ \boldsymbol{x} \in \mathcal{C} \;\middle|\; \frac{\boldsymbol{n}_F}{\|\boldsymbol{n}_F\|} \cdot (\boldsymbol{x} - \boldsymbol{v}_F) = 0 \right\} \tag{4.16}$$

with normal vector $\boldsymbol{n}_F$ and arbitrary face corner $\boldsymbol{v}_F$. Then the radius $r_{\min}$ of the inscribed sphere is

$$r_{\min}(\mathcal{C}) = \min_F \left\{ \left| \frac{\boldsymbol{n}_F}{\|\boldsymbol{n}_F\|} \cdot (\boldsymbol{c} - \boldsymbol{v}_F) \right| \right\}, \tag{4.17}$$

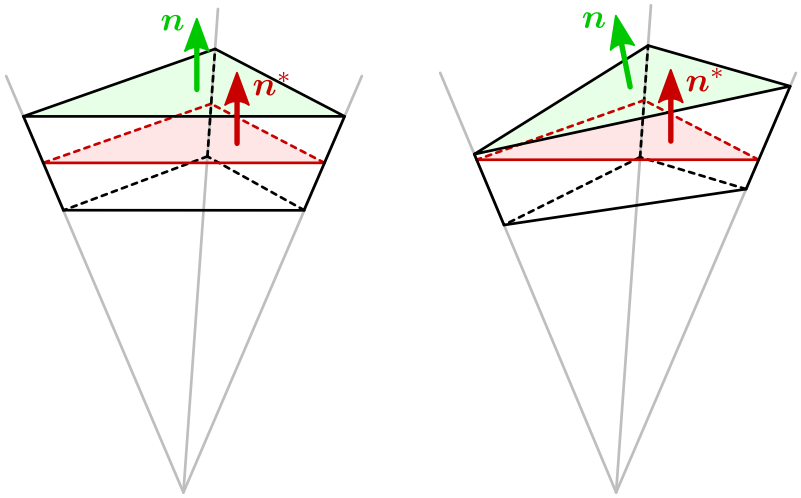
and we define the measure of anisotropy for each cell $\mathcal{C}$ as ratio

$$\nu_{\text{aniso}}(\mathcal{C}) := \frac{r_{\min}(\mathcal{C})}{r_{\max}(\mathcal{C})} \quad \in [0, 1]. \tag{4.18}$$

### 4.5.2. Orthogonality

With a measure of orthogonality we wish to make a statement on the horizontal and vertical alignment of each cell. By construction, all lateral cell faces are vertically aligned, compare Figure 4.3, so only the triangular faces at the top and bottom of each cell remain to be surveyed. They are horizontally aligned if the angle between the face's normal vector and the vector of the perfect vertical direction of the cell is zero, see Figure 4.24. We consider only top faces of each cell since bottom faces are either top faces of the lower neighbor or terrain faces for which an orthogonality statement is meaningless.

Let $\boldsymbol{c}$ be the cell's center, $\boldsymbol{n}$ the upwards pointing normal vector of the top triangular face and $\boldsymbol{v}_0$, $\boldsymbol{v}_1$, and $\boldsymbol{v}_2$ the face's vertices in counterclockwise numbering. To compute the upwards pointing vertical direction $\boldsymbol{n}^*$, we normalize the vertices and compute the




(a) (b)

Figure 4.24.: Measure of orthogonality: Angle between the normal vector $\boldsymbol{n}$ of the top face and the vertical direction $\boldsymbol{n}^*$. (a) Regular cut cell, (b) terrain-following cell. By construction, lateral cell faces are vertically aligned.

normal vector of the plane defined by these three points

$$\boldsymbol{n}^* = \left(\frac{\boldsymbol{v}_1}{\|\boldsymbol{v}_1\|} - \frac{\boldsymbol{v}_0}{\|\boldsymbol{v}_0\|}\right) \times \left(\frac{\boldsymbol{v}_2}{\|\boldsymbol{v}_2\|} - \frac{\boldsymbol{v}_0}{\|\boldsymbol{v}_0\|}\right).$$

For the sake of a direct interpretation as an angle, we omit the normalization of this measure and define

$$\nu^\circ_{\text{orth}}(\mathcal{C}) := \arccos \frac{\boldsymbol{n}(\mathcal{C}) \cdot \boldsymbol{n}^*(\mathcal{C})}{\|\boldsymbol{n}(\mathcal{C})\|\|\boldsymbol{n}^*(\mathcal{C})\|} \quad \in \left[0, \frac{\pi}{2}\right] \tag{4.19}$$

with the other three requirements of a fair measure still fulfilled. Normalized, the measure would be

$$\nu_{\text{orth}}(\mathcal{C}) := \cos \nu^\circ_{\text{orth}}(\mathcal{C}) = \frac{\boldsymbol{n}(\mathcal{C}) \cdot \boldsymbol{n}^*(\mathcal{C})}{\|\boldsymbol{n}(\mathcal{C})\|\|\boldsymbol{n}^*(\mathcal{C})\|} \quad \in [0, 1]. \tag{4.20}$$

### 4.5.3. Deformation

A further distinctive feature between cut cells and terrain-following cells is their degree of deformation. By construction, the latter follow the shape of the orography and are thus sheared, stretched, or compressed. As a measure, we choose the ratio of the cell's volume to the volume of the corresponding "ideal" cell. Such an ideal cell is horizontally aligned and at the same layer the actual cell belongs to. Figure 4.25 illustrates the principle. Again, this measure is only meaningful for regular cells.

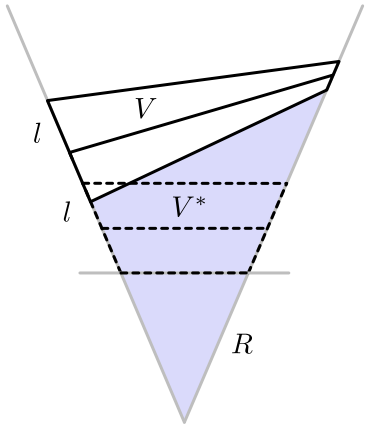



Figure 4.25.: Measure of deformation: Ratio between cell volume $V$ and the volume $V^*$ of the corresponding horizontally-aligned cell (dashed) at the same layer $l$. Compare also Figure 4.6.

Let $\mathcal{C}$ be the current cell, $V$ its volume and $V^*$ the volume of the ideal cell in the same column but at the height of the corresponding horizontally-aligned layer. Since the sheared, stretched, or compressed cell's volume can deviate from the ideal volume in both ways, we define the corresponding fair measure of deformation as

$$\nu_{\text{defor}}(\mathcal{C}) := \frac{|V(\mathcal{C}) - V^*(\mathcal{C})|}{V^*(\mathcal{C})} \quad \in [0, 1]. \tag{4.21}$$

### 4.5.4. Cut Cell Statistics

All three measures are not defined for cut cells since their shape and size are arbitrary and thus the measures fail to give useful information. Meaningful statements about cut cells, which we will collect for our concrete grids, are the number of cut cells $N^{\text{cut}}$ with respect to the total number of grid cells $N$ in percent

$$p_{\#}^{\text{cut}} := \frac{100 N^{\text{cut}}}{N} \quad \in [0\,\%, 100\,\%], \tag{4.22}$$

the number of small cut cells $N^{\text{small}}$, which are defined as smaller than half of the smallest regular cell, with respect to the total number of grid cells

$$p_{\#}^{\text{small}} := \frac{100 N^{\text{small}}}{N} \quad \in [0\,\%, 100\,\%], \tag{4.23}$$

the size of the terrain area $A^{\text{small}}$ of small cut cells, i.e. the sum of all areas of boundary faces which belong to small cells, with respect to the total terrain area $A$

$$p_{\text{area}}^{\text{small}} := \frac{100 A^{\text{small}}}{A} \quad \in [0\,\%, 100\,\%], \tag{4.24}$$

the smallest regular cell volume

$$V_{\text{min}}^{\text{reg}} := \min_{\mathcal{C}^{\text{reg}}} V(\mathcal{C}^{\text{reg}}), \tag{4.25}$$

the cut cells' smallest, largest, and average volume

$$V_{\text{min}}^{\text{cut}} := \min_{\mathcal{C}^{\text{cut}}} V(\mathcal{C}^{\text{cut}}),\ V_{\text{max}}^{\text{cut}} := \max_{\mathcal{C}^{\text{cut}}} V(\mathcal{C}^{\text{cut}}),\ V_{\varnothing}^{\text{cut}} := \frac{1}{N^{\text{cut}}} \sum_{\mathcal{C}^{\text{cut}}} V(\mathcal{C}^{\text{cut}}), \tag{4.26}$$

and the aspect ratio of inscribed to circumscribed sphere of each cut cell as measure of anisotropy

$$\nu_{\text{aniso}}^{\text{cut}}(\mathcal{C}^{\text{cut}}) := \frac{r_{\text{min}}(\mathcal{C}^{\text{cut}})}{r_{\text{max}}(\mathcal{C}^{\text{cut}})} \quad \in [0, 1]. \tag{4.27}$$

Again, the inscribed radius $r_{\text{min}}$ is computed as smallest distance of the centroid to a face and the circumscribed radius $r_{\text{max}}$ as largest distance of the centroid to a cell corner.

With all of these measures and percentages at hand, we now analyze our concrete grids. We choose our horizontal grids of Table 3.2 as basis and construct terrain-following and cut cell grids as described in Section 4.3 and 4.4. According to the constant bisection of horizontal mesh size in our test grid set, we choose one horizontal layer for the coarsest resolution and double the number of layers as the horizontal resolution is bisected. In this way, level 11 corresponds to 32 layers. Tables 4.2 and 4.3 give an overview of the resulting grid characteristics.

## 4.6. Comparison

At this point, we would like to sum up the advantages and disadvantages of the different types of vertical coordinates, which we examined in detail in the previous sections. First, we will have a closer look at the grid statistics of Tables 4.2 and 4.3 for our concrete atmospheric test meshes. Then, we will summarize the characteristic features of the different vertical coordinate approaches in a more general way and present a compact overview in Table 4.4.

Tables 4.2 and 4.3 give us insight into the dimensions and orders of magnitude of the grid characteristics for terrain-following and cut cell meshes with either horizontally uniform or adaptive triangulations. First of all, anisotropy is an inherent problem of the Earth's atmosphere and thus also of the cells in any kind of grid. Therefore, we always observe about three orders of magnitude difference in horizontal and vertical extent of

| | terrain-following<br>horizontally uniform | | | | | |
|---|---|---|---|---|---|---|
| level | 1 | 3 | 5 | 7 | 9 | 11 |
| layer | $1 \cdot 24\,\mathrm{km}$ | $2 \cdot 12\,\mathrm{km}$ | $4 \cdot 6\,\mathrm{km}$ | $8 \cdot 3\,\mathrm{km}$ | $16 \cdot 1.5\,\mathrm{km}$ | $32 \cdot 0.75\,\mathrm{km}$ |
| #cells | 24 | 192 | 1,536 | 12,288 | 98,304 | 786,432 |
| $\nu_{\mathrm{aniso,min,\varnothing}}$ | $2.4_{-3}$ | $2.3_{-3}$ | $2.4_{-3}$ | $2.4_{-3}$ | $2.4_{-3}$ | $2.4_{-3}$ |
| $\nu^{\circ}_{\mathrm{orth,max}}$ | 0.03° | 0.04° | 0.34° | 0.49° | 1.04° | 2.34° |
| $\nu_{\mathrm{defor,max}}$ | $1.9_{-1}$ | $7.7_{-2}$ | $2.7_{-2}$ | $7.5_{-3}$ | $1.9_{-3}$ | $1.5_{-3}$ |
| $V^{\mathrm{reg}}_{\mathrm{min}}$ | $3.3_{17}$ | $4.1_{16}$ | $4.0_{15}$ | $4.3_{14}$ | $5.0_{13}$ | $6.0_{12}$ |
| $V^{\mathrm{reg}}_{\mathrm{max}}$ | $3.3_{17}$ | $7.1_{16}$ | $1.3_{16}$ | $1.8_{15}$ | $2.4_{14}$ | $3.0_{13}$ |

| | cut cells<br>horizontally uniform | | | | | |
|---|---|---|---|---|---|---|
| level | 1 | 3 | 5 | 7 | 9 | 11 |
| layer | $1 \cdot 24\,\mathrm{km}$ | $2 \cdot 12\,\mathrm{km}$ | $4 \cdot 6\,\mathrm{km}$ | $8 \cdot 3\,\mathrm{km}$ | $16 \cdot 1.5\,\mathrm{km}$ | $32 \cdot 0.75\,\mathrm{km}$ |
| #cells | 24 | 192 | 1,536 | 12,285 | 98,141 | 782,320 |
| #cut | 100 %: 24 | 50 %: 96 | 25 %: 388 | 13 %: 1,580 | 7 %: 6,870 | 4 %: 30,694 |
| #small | 0 %: 0 | 0 %: 0 | 0 %: 0 | 0.2 %: 29 | 0.5 %: 448 | 0.6 %: 4,568 |
| $p^{\mathrm{small}}_{\mathrm{area}}$ | 0 % | 0 % | 0 % | 0.8 % | 3.1 % | 7.7 % |
| $\nu_{\mathrm{aniso,min,\varnothing}}$ | $2.4_{-3}$ | $2.3_{-3}$ | $2.4_{-3}$ | $2.4_{-3}$ | $2.4_{-3}$ | $2.4_{-3}$ |
| $\nu^{\mathrm{cut}}_{\mathrm{aniso,min}}$ | $2.3_{-3}$ | $1.8_{-3}$ | $1.2_{-3}$ | $1.1_{-5}$ | $6.7_{-5}$ | $2.2_{-5}$ |
| $\nu^{\mathrm{cut}}_{\mathrm{aniso,max}}$ | $2.4_{-3}$ | $2.4_{-3}$ | $2.6_{-3}$ | $2.9_{-3}$ | $4.3_{-3}$ | $7.0_{-3}$ |
| $\nu^{\circ}_{\mathrm{orth,max}}$ | 0° | 0° | 0° | 0° | 0° | 0° |
| $\nu_{\mathrm{defor,max}}$ | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| $V^{\mathrm{reg}}_{\mathrm{min}}$ | – | $4.1_{16}$ | $4.0_{15}$ | $4.3_{14}$ | $5.0_{13}$ | $6.0_{12}$ |
| $V^{\mathrm{reg}}_{\mathrm{max}}$ | – | $7.1_{16}$ | $1.3_{16}$ | $1.8_{15}$ | $2.4_{14}$ | $3.0_{13}$ |
| $V^{\mathrm{cut}}_{\mathrm{min}}$ | $3.1_{17}$ | $3.7_{16}$ | $3.1_{15}$ | $3.5_{10}$ | $8.3_{6}$ | $1.5_{6}$ |
| $V^{\mathrm{cut}}_{\mathrm{max}}$ | $3.2_{17}$ | $6.8_{16}$ | $1.2_{16}$ | $1.8_{15}$ | $2.4_{14}$ | $3.0_{13}$ |
| $V^{\mathrm{cut}}_{\varnothing}$ | $3.2_{17}$ | $5.3_{16}$ | $6.9_{15}$ | $7.9_{14}$ | $8.2_{13}$ | $7.5_{12}$ |

Table 4.2.: Grid statistics as defined in (4.18) – (4.27) for terrain-following vertical coordinates and cut cells based on the uniform horizontal grids of Table 3.2. Sole information about cut cells is shaded gray. The lower index of numbers denotes the exponent to the base 10.

| | terrain-following<br>horizontally adaptive | | | | | |
|---|---|---|---|---|---|---|
| level | 1–6 | 3–8 | 5–10 | 7–12 | 9–14 | 11–16 |
| layer | $1 \cdot 24\,\mathrm{km}$ | $2 \cdot 12\,\mathrm{km}$ | $4 \cdot 6\,\mathrm{km}$ | $8 \cdot 3\,\mathrm{km}$ | $16 \cdot 1.5\,\mathrm{km}$ | $32 \cdot 0.75\,\mathrm{km}$ |
| #cells | 44 | 312 | 2,728 | 22,144 | 180,736 | 1,398,528 |
| $\nu_{\mathrm{aniso,min,\varnothing}}$ | $4.3_{-3}$ | $4.5_{-3}$ | $5.4_{-3}$ | $5.7_{-3}$ | $6.0_{-3}$ | $6.0_{-3}$ |
| $\nu^{\circ}_{\mathrm{orth,max}}$ | 0.34° | 0.34° | 1.15° | 3.03° | 5.38° | 8.64° |
| $\nu_{\mathrm{defor,max}}$ | $1.9_{-1}$ | $7.7_{-2}$ | $2.7_{-2}$ | $7.5_{-3}$ | $1.9_{-3}$ | $1.9_{-3}$ |
| $V^{\mathrm{reg}}_{\mathrm{min}}$ | $2.3_{16}$ | $2.1_{15}$ | $1.8_{14}$ | $1.6_{13}$ | $1.7_{12}$ | $1.9_{11}$ |
| $V^{\mathrm{reg}}_{\mathrm{max}}$ | $3.3_{17}$ | $7.1_{16}$ | $1.3_{16}$ | $1.8_{15}$ | $2.4_{14}$ | $3.0_{13}$ |

| | cut cells<br>horizontally adaptive | | | | | |
|---|---|---|---|---|---|---|
| level | 1–6 | 3–8 | 5–10 | 7–12 | 9–14 | 11–16 |
| layer | $1 \cdot 24\,\mathrm{km}$ | $2 \cdot 12\,\mathrm{km}$ | $4 \cdot 6\,\mathrm{km}$ | $8 \cdot 3\,\mathrm{km}$ | $16 \cdot 1.5\,\mathrm{km}$ | $32 \cdot 0.75\,\mathrm{km}$ |
| #cells | 44 | 312 | 2,728 | 22,017 | 178,407 | 1,366,274 |
| #cut | 100 %: 44 | 50 %: 156 | 26 %: 698 | 14 %: 3,085 | 8 %: 14,796 | 5 %: 70,059 |
| #small | 0 %: 0 | 0 %: 0 | 0.6 %: 17 | 0.8 %: 180 | 1.0 %: 1,736 | 0.8 %: 11,604 |
| $p^{\mathrm{small}}_{\mathrm{area}}$ | 0 % | 0 % | 0.2 % | 0.2 % | 0.5 % | 0.6 % |
| $\nu_{\mathrm{aniso,min,\varnothing}}$ | $4.1_{-3}$ | $4.5_{-3}$ | $5.3_{-3}$ | $5.6_{-3}$ | $5.8_{-3}$ | $5.9_{-3}$ |
| $\nu^{\mathrm{cut}}_{\mathrm{aniso,min}}$ | $2.3_{-3}$ | $1.8_{-3}$ | $1.2_{-3}$ | $1.1_{-5}$ | $1.1_{-5}$ | $1.1_{-5}$ |
| $\nu^{\mathrm{cut}}_{\mathrm{aniso,max}}$ | $9.3_{-3}$ | $1.2_{-2}$ | $1.4_{-2}$ | $2.1_{-2}$ | $2.4_{-2}$ | $3.1_{-2}$ |
| $\nu^{\circ}_{\mathrm{orth,max}}$ | 0° | 0° | 0° | 0° | 0° | 0° |
| $\nu_{\mathrm{defor,max}}$ | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| $V^{\mathrm{reg}}_{\mathrm{min}}$ | – | $2.1_{15}$ | $1.8_{14}$ | $1.6_{13}$ | $1.7_{12}$ | $1.9_{11}$ |
| $V^{\mathrm{reg}}_{\mathrm{max}}$ | – | $7.1_{16}$ | $1.3_{16}$ | $1.8_{15}$ | $2.4_{14}$ | $3.0_{13}$ |
| $V^{\mathrm{cut}}_{\mathrm{min}}$ | $2.0_{16}$ | $2.0_{15}$ | $2.3_{13}$ | $3.6_{6}$ | $7.5_{3}$ | $1.4_{3}$ |
| $V^{\mathrm{cut}}_{\mathrm{max}}$ | $3.2_{17}$ | $6.8_{16}$ | $1.2_{16}$ | $1.7_{15}$ | $2.3_{14}$ | $3.0_{13}$ |
| $V^{\mathrm{cut}}_{\varnothing}$ | $1.9_{17}$ | $3.3_{16}$ | $3.8_{15}$ | $4.0_{14}$ | $3.7_{13}$ | $3.1_{12}$ |

Table 4.3.: Grid statistics as defined in (4.18) – (4.27) for terrain-following vertical coordinates and cut cells based on the adaptive horizontal grids of Table 3.2. Sole information about cut cells is shaded gray.

each cell. Remember our validation in Section 4.3, in which one order of magnitude was essential for a stable simulation with terrain-following coordinates, and the exorbitant increase in degrees of freedom and thus costs listed in Table 4.1. For arbitrarily shaped cut cells, the anisotropy of cells can both increase or decrease as expected and revealed in the minimal and maximal values $\nu^{\text{cut}}_{\text{aniso,min}}$ and $\nu^{\text{cut}}_{\text{aniso,max}}$. Here, the anisotropy varies between two and five orders of magnitude.

Greater differences between terrain-following and cut cell grids become evident regarding orthogonality and deformation. Meanwhile the regular cells in a cut cell mesh are perfectly Cartesian and thus orthogonal and undeformed, terrain-following grids suffer from both aspects. On the one hand, the cells are usually deformed with only a slight decrease for finer meshes and on the other hand, the deviation from orthogonality is distinctly increasing for our test grid sets by up to 9° for adaptive terrain-following meshes. The latter observation is concordant with our theoretical statement in Section 4.3 that the cells tend to be steeper for finer meshes and thus the pressure gradient force error and the hydrostatic inconsistency worsen exceedingly.

What is still lacking is information about the cut cells which, as we know, may get arbitrarily small. To get a feeling for them, we added a few quantitative statements on cut cells for our test grid sets. First of all, we can state that the percentage of cut cells in the grid is rapidly decreasing with refinement. In our lists, we end with about $4 - 5\,\%$ of cut cells. This is a quite small percentage compared to the statement in [ABM99] that for arbitrarily complex geometries typically $10 - 15\,\%$ of cells are cut. Moreover, the percentage of small cut cells is even considerably smaller and increases only marginally, ending with $0.6 - 0.8\,\%$ in our lists. This small percentage of problematic grid cells allows us to reconsider the option of limiting small cells by omitting them, compare Section 4.4.4.

A further indication for this is the area of the terrain belonging to small cut cells. For uniform refinement, this area increases up to $7.7\,\%$ in our lists, but the choice of horizontally adaptive grids reduces this amount significantly. Here, the terrain area of small cells increases very slowly with refinement, ending in our lists with the little amount of $0.6\,\%$ relating to the total terrain surface.

Furthermore, a comparison of minimal and maximal volumes confirms that extremely small cells, which arise with a size up to eight orders of magnitude smaller than minimal regular cells, are just outliers. The average volume of cut cells is only one order of magnitude below their maximum volume, whereas the minimum value is up to ten orders of magnitude lower. Additionally, the maximum volume almost coincides with the corresponding regular cell's volume. So we can summarize that small cells are indeed exceptions and most cut cells are only slightly smaller than regular cells and thus pose no problem at all.

Let us now turn to the comparison of step-mountain, terrain-following, and cut cell approaches based on the differences worked out in Sections 4.2 to 4.4.

The brief overview of step-mountain coordinates in Section 4.2 caused us to exclude

| | step-mountain | terrain-following | cut cells |
|---|---|---|---|
| Resolution of topography ... | zeroth order; flow error increases with finer meshes | ≥ first order | ≥ first order |
| Cell Alignment ..... | orthogonal | cells sheared, compressed/stretched | orthogonal + cut cells |
| Construction ..... | simple | simple | simple, cut cells once preprocessed |
| Coordinate transformation .. | no | yes | no |
| Pressure gradient force ..... | no error | error, increases with finer meshes | no error |
| Hydrostatic consistency ..... | no error | error, increases with finer meshes | no error |
| High vertical resolution near ground ..... | expensive | easy | expensive |
| Arbitrarily small cells ..... | no | no | yes |
| Universality, flexibility ..... | yes | no | yes |

Table 4.4.: Comparison of vertical coordinates for atmospheric dynamics. Disadvantageous features are shaded gray.

them from any further investigation because of their bad representation of topography and the implied spurious flow disturbances above step corners and constant underestimation of downwind flow. These errors even worsen for finer meshes which leads to a discrepancy with the need for high resolution to represent the terrain in an acceptable way.

A closer look at the widely-used terrain-following coordinates in Section 4.3 illustrated their advantageous simplicity through topographic alignment. But on the other hand, we saw that they are also accompanied by serious disadvantages like severe pressure gradient force and hydrostatic inconsistency errors, which are both increasing with finer meshes and for which no satisfying remedies exist.

A promising alternative is the cut cell approach presented in Section 4.4 in detail, which naturally circumvents the drawbacks of terrain-following coordinates. A serious restriction on the time step due to arbitrarily small cells turned out to be the main

problem attached to cut cells, but, as studied in the previous section, many remedies exist.

In Table 4.4, the different features of the three approaches are summarized in a compact way. Due to the favorable characteristics of cut cell methods and their still pending use in today's weather forecast systems, we will study their application more closely in the course of this thesis.

## 4.7. Our Vertical Scheme

As motivated in the previous sections, we wish to fathom the capabilities of a cut cell approach for the simulation of atmospheric dynamics in contrast to the widely-used terrain-following coordinates in recent weather forecast systems. The capabilities of cut cells are obvious, recall the detailed analysis in Section 4.4 and 4.6, so that they should find their belated way into atmospheric dynamics. Therefore, our choice for a discretized atmosphere is a special cut cell approach, which we will describe in detail in this section.

### 4.7.1. Construction of Atmospheric Cut Cells

Let us first have a closer look at the practical construction of cut cells in our atmospheric setting. In this part, we intend to provide a comprehensive guideline for an implementation of cut cells into existing atmospheric codes, which has not been available so far.

To this end, we combine the methods for a global bisection-based triangulation of the Earth's surface as described in Section 3.2 and the vertical projection of the triangulation into atmospheric layers explained in Section 4.1 with the cut cell approach. The first two methods are integrated in practically every weather forecast system. Thereby, the concrete triangulation of the Earth's topography is of no significance for the further procedure, and the layered partition of the atmosphere is the obvious choice because of its thin extension and the dominance of stratified flow in the atmosphere. Moreover, with this ansatz, a straightforward implementation of terrain-following coordinates is possible. However, we will now incorporate cut cells in the same environment.

We start with a clarification what "regular cells" and "cut cells" mean in our setting. Since we aim at simulating atmospheric dynamics on a global grid, compare Section 3.2.3, our regular three-dimensional Cartesian cells are not cuboids or prisms but rather truncated tetrahedrons, as we already observed in Section 4.1. Figure 4.3 illustrates the construction principle of these truncated tetrahedral cells for a flat Earth without any terrain. This so-constructed mesh is our background Cartesian grid whose cells are already sufficient for discretizing the interior of the atmospheric domain. So again, we choose height as vertical coordinate and gain planar horizontal layers, i.e. each regular cell is perpendicularly aligned to the radial direction of its cell center.

The new part of the construction is related to the cut cells at the boundary of the domain. Based on our horizontal terrain triangulation described in Section 3, the topo-

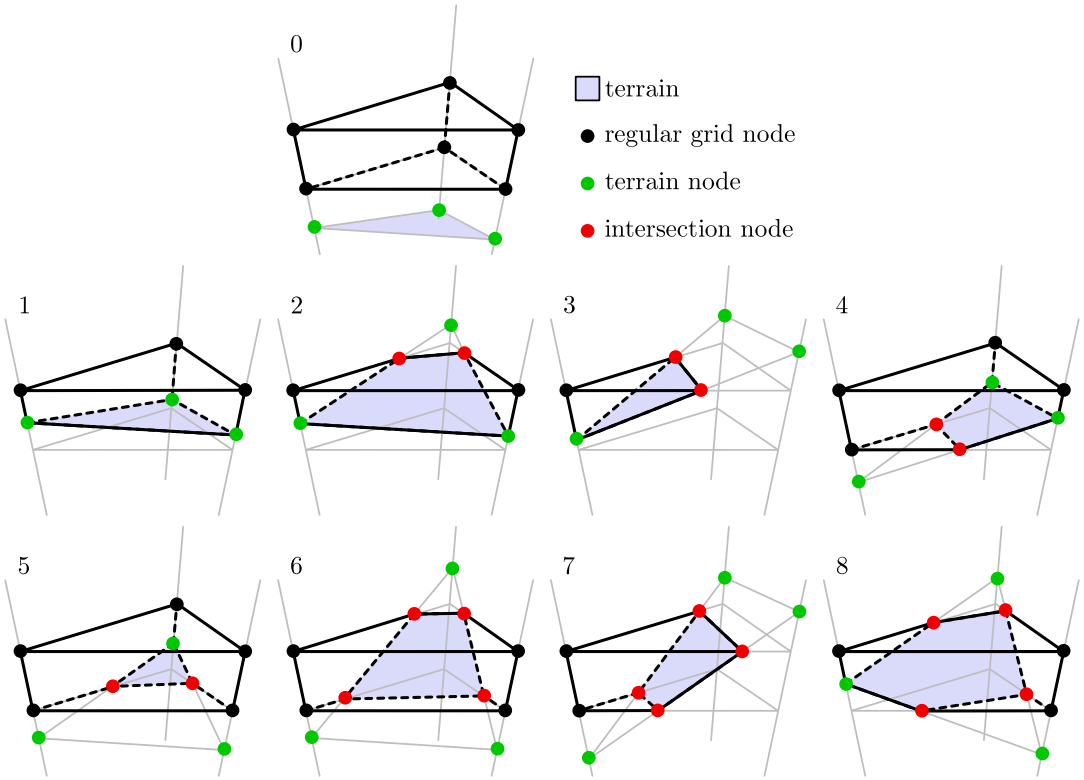



Figure 4.26.: Different types of cut cells. A planar terrain triangle can intersect a regular truncated tetrahedral cell (top) in eight different ways (bottom rows). Each case implies the cell's horizontal rotation and reflection.

graphy of the Earth is piecewise linearly approximated and thus intersects certain regular cells. These intersections leave so-called cut cells behind, which vary in shape and have reduced volumes as well as reduced or vanishing face areas.

Before we turn to the construction process in detail, have a look at Figure 4.26. Here, the eight different types of cut cells are depicted. Since all height values of the terrain are defined at the vertical edges of each cell column, compare Figure 4.3b, and the terrain in between is approximated linearly, these eight types are the only possible ways a regular truncated tetrahedral cell can be intersected. Thereby, each case implies the cut cell's horizontal rotation and reflection.

In the literature, only one exemplary type of cut cell is typically depicted and the troublesome but necessary distinction of cases is omitted [AHM97, SBJ+06, WA08]. For cubed cells a distinction can be found in [OSK09], but for cells different from cubes we feel a lack of a detailed practical description of the construction of cut cells throughout the literature. In [LBC+12], even an argument can be found stating that the distinction of cases is too complicated at all and is thus completely avoided. This is why we want to fill this gap now by describing in detail the construction of the different cut cells, for which only computations of basic geometric intersections are needed.

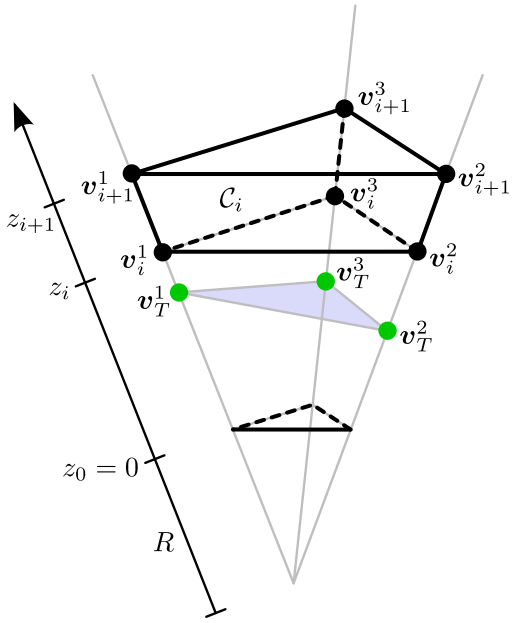



Figure 4.27.: Notation used for the construction of cut cells.

By construction, each terrain triangle is assigned to a column of regular grid cells. For each of these cells we have to decide whether the cell is completely above or below ground or whether it intersects the terrain triangle. In the latter case, the corresponding cut cell has to be constructed.

As depicted in Figure 4.27, let $T = (\boldsymbol{v}_T^1, \boldsymbol{v}_T^2, \boldsymbol{v}_T^3)$ be a terrain triangle with corners $\boldsymbol{v}_T^k$, $k \in \{1, 2, 3\}$, and $\mathcal{C}_i$ a regular cell of the corresponding column in layer $i$, $i \in \mathbb{Z}$, with three corners $(\boldsymbol{v}_i^1, \boldsymbol{v}_i^2, \boldsymbol{v}_i^3)$ at level height $z_i \in \mathbb{R}$ above mean sea level and three corners $(\boldsymbol{v}_{i+1}^1, \boldsymbol{v}_{i+1}^2, \boldsymbol{v}_{i+1}^3)$ at height $z_{i+1}$. The upper vertex indices correspond to the three edges of the considered cell column. Note that $\|\boldsymbol{v}_i^k\| = R + z_i$ for $i \in \mathbb{Z}$ with Earth's radius $R$ and $k \in \{1, 2, 3\}$.

The first distinctions are quite straightforward. If

$$\max_k \|\boldsymbol{v}_T^k\| \leq R + z_i, \tag{4.28}$$

then $\mathcal{C}_i$ is completely above ground and thus a regular truncated tetrahedron. On the other hand, if

$$\min_k \|\boldsymbol{v}_T^k\| \geq R + z_{i+1}, \tag{4.29}$$

then $\mathcal{C}_i$ is completely underground and can be eliminated from the computational domain. In any other case, $\mathcal{C}_i$ is a cut cell which intersects the boundary. Note that (4.28) and (4.29) include the cases with $\boldsymbol{v}_T^k = \boldsymbol{v}_i^k \; \forall\, k$ and $\boldsymbol{v}_T^k = \boldsymbol{v}_{i+1}^k \; \forall\, k$.

For further distinction, we determine the layer $l_k := l(\boldsymbol{v}_T^k)$ in which each terrain triangle node $\boldsymbol{v}_T^k$ is located. To this end, the height $\|\boldsymbol{v}_T^k\|$ of the triangle node has to be compared

to the level heights defined by the Cartesian mesh

$$l(\boldsymbol{v}_T^k) := j \quad \text{if} \quad R + z_j \leq \|\boldsymbol{v}_T^k\| < R + z_{j+1}. \tag{4.30}$$

Then, depending on their relative position to layer $i$ of cell $\mathcal{C}_i$, the eight types can be dinstinguished. The case numbering is consistent with Figure 4.26.

Case 1: Three nodes in layer $i$

$$l_k = i \quad \forall\, k. \tag{4.31}$$

Case 2: Two nodes in layer $i$, one higher

$$l_k \begin{cases} > i & \text{for exactly one } k, \\ = i & \text{else.} \end{cases} \tag{4.32}$$

Case 3: One node in layer $i$, two higher

$$l_k \begin{cases} = i & \text{for exactly one } k, \\ > i & \text{else.} \end{cases} \tag{4.33}$$

Case 4: Two nodes in layer $i$, one lower

$$l_k \begin{cases} < i & \text{for exactly one } k, \\ = i & \text{else.} \end{cases} \tag{4.34}$$

Case 5: One node in layer $i$, two lower

$$l_k \begin{cases} = i & \text{for exactly one } k, \\ < i & \text{else.} \end{cases} \tag{4.35}$$

Case 6: Two nodes lower than layer $i$, one higher

$$l_k \begin{cases} > i & \text{for exactly one } k, \\ < i & \text{else.} \end{cases} \tag{4.36}$$

Case 7: One node lower than layer $i$, two higher

$$l_k \begin{cases} < i & \text{for exactly one } k, \\ > i & \text{else.} \end{cases} \tag{4.37}$$

Case 8: One node in layer $i$, one lower, one higher

$$l_k \begin{cases} = i & \text{for exactly one } k, \\ < i & \text{for exactly one } k, \\ > i & \text{else.} \end{cases} \tag{4.38}$$

Now, depending on the cut cell type of $\mathcal{C}_i$, intersections of the terrain triangle with certain horizontal cell edges have to be computed. For a bottom-up traversal of each cell column, it is essential to recognize that only intersections at the top face of each cell have to be computed since those at the bottom face are already known due to the lower neighbor cell. Apart from that, an intersection node belongs not only to a cell and one of its vertical neighbors but also to a horizontal neighbor cell so that even further redundant computations can be avoided. Therefore, looking at Figure 4.26, only types 2, 3, 6, 7, and 8 involve new intersections, if any at all.

For the actual computation of upper intersection nodes of cell $\mathcal{C}_i$, we set up the plane equation of the terrain triangle

$$\boldsymbol{n} \cdot (\boldsymbol{x} - \boldsymbol{p}) = 0 \tag{4.39}$$

with normal vector $\boldsymbol{n} := (\boldsymbol{v}_T^1 - \boldsymbol{v}_T^2) \times (\boldsymbol{v}_T^1 - \boldsymbol{v}_T^3)$ and position vector $\boldsymbol{p} := \boldsymbol{v}_T^1$.

Next, we need to find the corner $\boldsymbol{v}_{i+1}^j$ of the regular cell $\mathcal{C}_i$ which is enframed by the pairwise appearing intersection nodes. For types 2, 6, and 8, it is the corner $\boldsymbol{v}_{i+1}^j$ with $j$ being the upper index of the terrain triangle node $\boldsymbol{v}_T^j$ which lies higher than layer $i$, compare (4.32), (4.36), and (4.38). For types 3 and 7, $j$ has to be the upper index of the node $\boldsymbol{v}_T^j$ which lies in or below layer $i$, compare (4.33) and (4.37).

Having found the central corner $\boldsymbol{v}_{i+1}^j$, the two cell edges defined by

$$\overline{\boldsymbol{v}_{i+1}^j \boldsymbol{v}_{i+1}^{k_1}} \text{ with } k_1 \neq j \quad \text{and} \quad \overline{\boldsymbol{v}_{i+1}^j \boldsymbol{v}_{i+1}^{k_2}} \text{ with } k_2 \neq j,\ k_2 \neq k_1 \tag{4.40}$$

are the lines intersecting the terrain triangle. Written as equations of a line

$$g_1 :\ \boldsymbol{x} = \boldsymbol{a} + \lambda \boldsymbol{r}_1 \quad \text{and} \quad g_2 :\ \boldsymbol{x} = \boldsymbol{a} + \mu \boldsymbol{r}_2 \tag{4.41}$$

with position vector $\boldsymbol{a} := \boldsymbol{v}_{i+1}^j$, scalars $\lambda, \mu \in \mathbb{R}$ and direction vectors $\boldsymbol{r}_1 := \boldsymbol{v}_{i+1}^j - \boldsymbol{v}_{i+1}^{k_1}$ and $\boldsymbol{r}_2 := \boldsymbol{v}_{i+1}^j - \boldsymbol{v}_{i+1}^{k_2}$ and substituted into plane equation (4.39), we gain the two intersections

$$\boldsymbol{s}_1 = \boldsymbol{a} - \frac{\boldsymbol{n} \cdot (\boldsymbol{a} - \boldsymbol{p})}{\boldsymbol{n} \cdot \boldsymbol{r}_1} \boldsymbol{r}_1 \quad \text{and} \quad \boldsymbol{s}_2 = \boldsymbol{a} - \frac{\boldsymbol{n} \cdot (\boldsymbol{a} - \boldsymbol{p})}{\boldsymbol{n} \cdot \boldsymbol{r}_2} \boldsymbol{r}_2. \tag{4.42}$$

Finally, we add these intersection nodes to the current cell and the appropriate vertical and horizontal neighbor cells in order to avoid redundant computations.

Now, we know every corner of the cut cell polyhedron so that, depending on the cut

cell type and the chosen data format, the polyhedron can be constructed and stored. Typically, such a data format consists for each cell of a list of faces, for each face of a list of points defining its corners in a circulating order and for each point in a list of coordinates.

Algorithm 3 summarizes the described construction procedure of the cut cell mesh in pseudocode. Starting with a Cartesian mesh and a terrain triangulation, the function `determine_layer_of_node` classifies in which layer each node of a terrain triangle is located by comparing its height to the level heights defined by the Cartesian mesh. The lowest terrain triangle node of each cell column defines which Cartesian cells are underground and can thus be deleted from the mesh structure, as done in function `delete_underground_cell`.

Then, the actual cut cells have to be constructed. For a fixed cell column, cut cells appear in the range of the minimum and maximum layer defined by the terrain triangle nodes. After excluding cell types without new possible intersection nodes and verifying that the node is not already computed by neighboring cells, the edge cutting the terrain triangle is determined and in function `intersect` the actual intersection of plane and line is calculated. Then, in function `add_intersection_to_cell_and_neighbors`, the new intersection node is stored and linked to the current cell and the appropriate adjacent cells in vertical and horizontal direction. This avoids redundant computations.

In function `determine_exact_case` the type of a cut cell is identified by comparisons of the layers of the terrain triangle nodes to the layer of the current cell as described in detail in (4.31) – (4.38). At last, with the type of the cut cell and all its corners known, the function `construct_polyhedron` constructs the current cut cell by defining its vertices, edges, and faces in a chosen data format.

In this way, an existing terrain-following code can be directly modified to a cut cell code with Algorithm 3 as a preprocessing step which has to be executed once in advance.

As a remark, cut cells have a nice effect on the horizontal resolution near the ground. Of course, the orography is still de facto linearly approximated based on the terrain triangles. But with the additional intersection nodes of cut cells we gain more sampling points of the surface and thus what appears to be a better horizontal resolution. Figure 4.28 depicts the effect.

Now, the question remains which remedy we choose for circumventing the small cell problem and thus the restriction on the time step size.

### 4.7.2. Circumventing Small Cells

In Section 4.4.4, we gave a thorough survey of known remedies for the small cell problem. Because of our general framework, not all of them are applicable and even more of them have considerable disadvantages, which is pointed out in detail in the aforementioned section. We concluded that the first apparent ansatz is also the most promising, namely the discarding of small troublesome cells by filling them with solid.

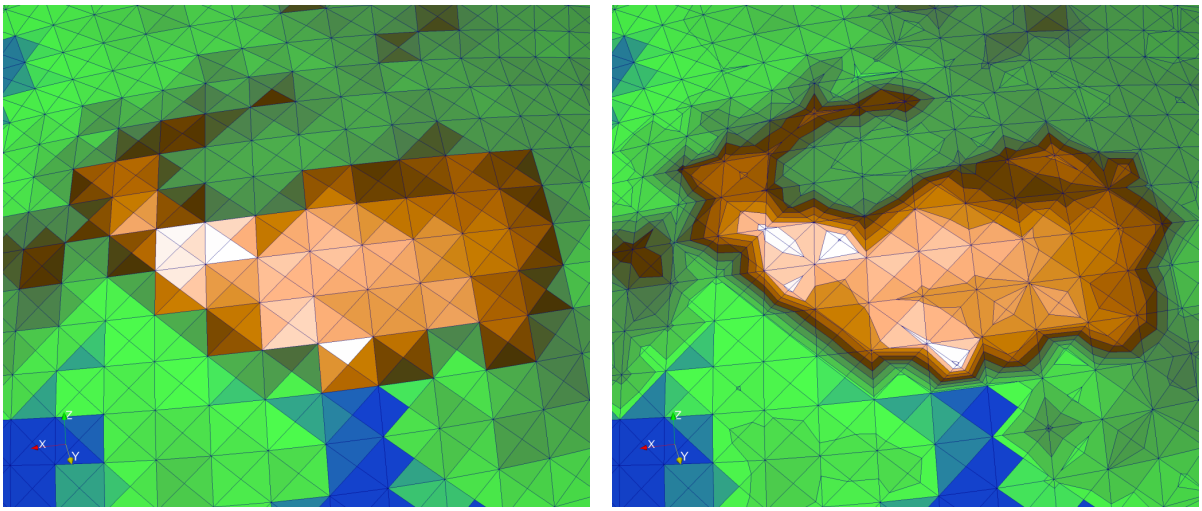
This estimation is supported by the statistics of our test grids studied in Section 4.6.

**Algorithm 3:** Construction of cut cells for a given Cartesian mesh and a terrain triangulation.

```
construct_cut_cells(Cartesian Mesh 𝒞, Terrain Triangulation 𝒯)
    forall Terrain Triangles T = (v_T^1, v_T^2, v_T^3) do
        forall k do
            Layer l_k ← determine_layer_of_node(𝒞, v_T^k)
        end
        forall Layer i < min_k l_k do
            delete_underground_cell(𝒞, T, i)
        end
        for Layer i ← min_k l_k to max_k l_k do
            // Case 2, 6, 8 or 3, 7
            if (l_j > i and l_k ≤ i ∀ k ≠ j) or (l_j ≤ i and l_k > i ∀ k ≠ j) then
                forall k ≠ j do
                    if Intersection not calculated by neighbors then
                        Normal Vector n ← (v_T^1 − v_T^2) × (v_T^1 − v_T^3)
                        Line Vector r ← v_{i+1}^j − v_{i+1}^k
                        Intersection s ← intersect(n, v_T^1, v_{i+1}^j, r)
                        add_intersection_to_cell_and_neighbors(𝒞, 𝒯, T, i, s)
                    end
                end
            end
            Case c ← determine_exact_case(i, l_1, l_2, l_3)
            construct_polyhedron(𝒞, T, i, c)
        end
    end
end

// Determination of the atmospheric layer of node v
Layer determine_layer_of_node(Cartesian Mesh 𝒞, Vector v)
    Height z ← ‖v‖− Earth's Radius R(𝒞)
    forall i do
        if z ≥ z_i(𝒞) and z < z_{i+1}(𝒞) then
            return i
        end
    end
end

// Intersection of plane n · (x − p) = 0 and line x = a + λr
Intersection intersect(Normal Vector n, Vector p, Vector a, Vector r)
    Scalar λ ← −n · (a − p)/(n · r)
    Intersection s ← a + λr
    return s
end
```



(a) (b)

Figure 4.28.: The Himalayas: Same horizontal resolution of the ground for (a) terrain-following coordinates and (b) cut cells. Color-coded is the height of each boundary face's barycenter.

We noticed that the percentage of small cut cells is very low, i.e. about 0.6 – 0.8 % of the total number of cells, and increases only marginally with mesh refinement. With respect to boundary cells, the percentage of small cells increases up to 15 – 17 % for our finest grids, i.e. 15 – 17 % of the terrain segments would be approximated by steps. At first sight, this seems to be much, but recall that the terrain area belonging to small cut cells does not correspond to this percentage. In fact, the amount of the terrain surface, which would be approximated by tiny steps, is much lower and increases especially for adaptive grids very slowly with refinement, ending in our lists with 0.6 % relating to the total terrain surface. So if we choose to circumvent the small cell problem by omitting these troublesome cells, only 0.6 % of the topography of the Earth would be approximated by zeroth order and 99.4 % still by first order.

So this is our first ansatz for a small cell remedy. Algorithm 4 describes the procedure of omitting cut cells if their volume is smaller than half of the volume of the smallest regular cell. The functions `determine_boundary_face` and `determine_neighbor_cell` as well as `define_as_boundary_face`, `delete_face_from_mesh`, and `delete_cell_from_mesh` depend on the data structure in which the mesh is organized and are thus not explicitly specified.

The core function of Algorithm 4 is `determine_volume`, stated below in Algorithm 5, in which the volume of each grid cell is calculated based on dissecting the convex polyhedron in tetrahedra or pyramids. To this end, let the cell have $n$ corners $\boldsymbol{v}_i$, $i = 0, \ldots, n-1$, and $m$ faces $F_j$, $j = 0, \ldots, m-1$, each face having $k_j \geq 3$ corners $\boldsymbol{v}_{j_0}, \ldots, \boldsymbol{v}_{j_{k_j-1}}$ in counterclockwise numbering when viewed from the outside and a normal vector $\boldsymbol{n}_j$

**Algorithm 4:** Limitation of the minimum volume of cut cells by omitting small cells for a given cut cell mesh $\mathcal{M} := \mathcal{C}^{\text{reg}} \dot{\cup} \mathcal{C}^{\text{cut}}$.

```
omit_small_cells(Cut Cell Mesh M)
    Volume V_min = ∞
    forall Regular Cells C_i^reg do
        Volume V ← determine_volume(C_i^reg)
        if V < V_min then
            V_min = V
        end
    end
    forall Cut Cells C_i^cut do
        Volume V ← determine_volume(C_i^cut)
        if V < V_min/2 then
            Face B ← determine_boundary_face(C_i^cut)
            delete_face_from_mesh(B, M)
            forall Faces F_j(C_i^cut)\B do
                Neighbor N ← neighbor_cell(F_j, C_i^cut, M)
                define_as_boundary_face(F_j, N, M)
            end
            delete_cell_from_mesh(C_i^cut, M)
        end
    end
end
```

**Algorithm 5:** Determination of the volume of a convex polyhedron $\mathcal{C}$ with $n$ vertices $\boldsymbol{v}_i$ and $m$ faces $F_j$ with $k_j$ vertices $\boldsymbol{v}_{j_0}, \ldots, \boldsymbol{v}_{j_{k_j-1}}$.

```
Volume determine_volume(Cell C)
    Volume V_sum ← 0
    Vector c ← (Σ_{i=0}^{n-1} v_i(C))/n    // centroid
    forall Faces F_j(C) do
        Normal Vector n ← (v_{j_2} − v_{j_0}) × (v_{j_1} − v_{j_0})
        n ← n/‖n‖
        Height h ← n · (c − v_{j_0})
        Area A ← (n · ((v_{j_0} × v_{j_{k_j-1}}) + Σ_{p=1}^{k_j-1}(v_{j_p} × v_{j_{p-1}})))/2
        Volume V ← Ah
        V_sum ← V_sum + V
    end
    return V_sum/3
end
```

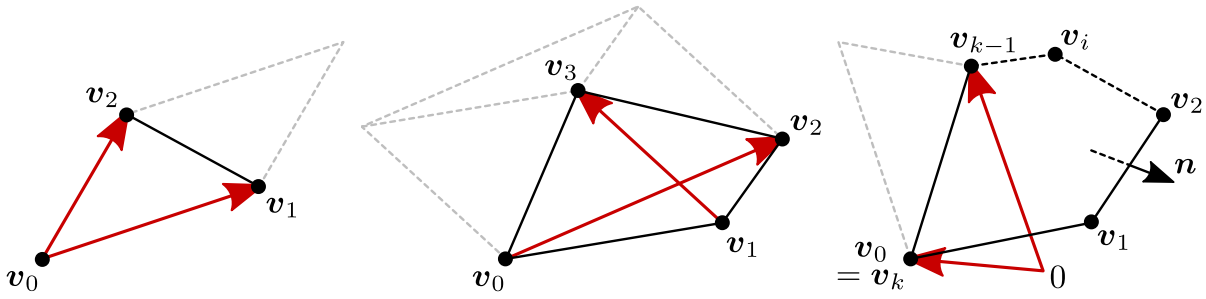



Figure 4.29.: Computing the area of polygons in 3D by means of cross products (red vectors) for (a) a triangle, (b) a quadrangle, and (c) a $k$-sided polygon.

pointing inside the cell with

$$\boldsymbol{n}_j := (\boldsymbol{v}_{j_2} - \boldsymbol{v}_{j_0}) \times (\boldsymbol{v}_{j_1} - \boldsymbol{v}_{j_0}). \tag{4.43}$$

Then, the dissection of the cell in disjoint pyramids is done by defining an arbitrary point inside the polyhedron, e.g. the centroid $\boldsymbol{c}$

$$\boldsymbol{c} := \frac{1}{n} \sum_{i=0}^{n-1} \boldsymbol{v}_i, \tag{4.44}$$

and connecting it with all of the corners. In this way, pyramids with different bases are defined, each base being a face of the polyhedron. By summing up all volumes of the pyramids, we gain the sought volume of the cell

$$V = \frac{1}{3} \sum_{j=0}^{m-1} A(F_j) h_j \tag{4.45}$$

with $A(F_j)$ being the area of face $F_j$ and $h_j$ the height of the corresponding pyramid with apex $\boldsymbol{c}$ and base $F_j$. The height is computed by inserting the centroid $\boldsymbol{c}$ in Hesse's normal form of face $F_j$

$$h_j = \frac{\boldsymbol{n}_j}{\|\boldsymbol{n}_j\|} \cdot (\boldsymbol{c} - \boldsymbol{v}_{j_0}) \,. \tag{4.46}$$

Now, only the calculation of the area of each $k$-sided polygonal face of the cell remains. For the special cases of a three-dimensional triangle and quadrangle, the area is obviously

$$A(F_j^{\text{tri}}) = \frac{1}{2} \left\| (\boldsymbol{v}_{j_2} - \boldsymbol{v}_{j_0}) \times (\boldsymbol{v}_{j_1} - \boldsymbol{v}_{j_0}) \right\| , \tag{4.47}$$

$$A(F_j^{\text{quad}}) = \frac{1}{2} \left\| (\boldsymbol{v}_{j_2} - \boldsymbol{v}_{j_0}) \times (\boldsymbol{v}_{j_3} - \boldsymbol{v}_{j_1}) \right\| , \tag{4.48}$$

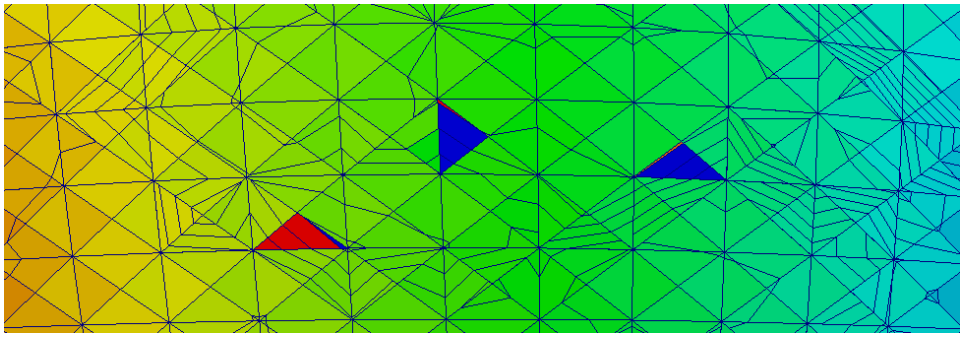


Figure 4.30.: Simulation instabilities in the density values in cells with very anisotropic boundary faces. Depicted is a detail of their color-coded distribution at ground level for a cut cell mesh, where small cells were already removed.

compare Figure 4.29a and 4.29b. A generalized formula for the area of $k$-sided polygons in three-dimensional space can be derived from Stokes' theorem and reads

$$A(F_j) = \frac{1}{2}\frac{\boldsymbol{n}_j}{\|\boldsymbol{n}_j\|} \cdot \sum_{p=0}^{k_j-1} (\boldsymbol{v}_{j_p} \times \boldsymbol{v}_{j_{p-1}}) \tag{4.49}$$

with $\boldsymbol{v}_{j_{-1}} := \boldsymbol{v}_{j_{k_j-1}}$, see [SE03] and the references therein. The principle is depicted in Figure 4.29c. So, the volume (4.45) of each cell in our mesh can now be calculated with the help of equations (4.44), (4.46), and (4.49) as done in Algorithm 5. Note that, for the sake of efficiency, the general formula (4.49) can be substituted in the case of three- and four-sided polygons – the most frequent cases in our meshes – by (4.47) and (4.48).

### 4.7.3. Further Mesh Improvement

Having discarded very small cells which restricted the time step size in a severe way, we detected a further source of instabilities in various simulation runs. From a certain resolution on, instabilities arise in cells with highly anisotropic boundary faces. Anticipating a simulation described in Section 6.2 in detail, Figure 4.30 shows such a situation. Here, in an otherwise smooth background distribution, very high and low density values appear in cells with extreme anisotropic acute-angled boundary faces and in their neighboring cells.

Therefore, we add a further criterion to eliminate those troublesome cells. Hence, we substitute Algorithm 4 by Algorithm 6, where two deletion criteria are incorporated. A cut cell is eliminated if, on the one hand, its volume is smaller than half of the smallest volume of a regular cell or, on the other hand, the boundary face's anisotropy underruns a certain threshold $\varepsilon > 0$. Due to the second criterion, it may happen that a cell is marked for deletion but its neighbor exactly underneath not. This can lead to re-entrant

**Algorithm 6:** Omitting small cells and cells with anisotropic boundary faces for a given cut cell mesh $\mathcal{M} := \mathcal{C}^{\text{reg}} \dot{\cup} \mathcal{C}^{\text{cut}}$.

```
omit_small_cells_and_with_aniso_bfaces(Cut Cell Mesh M, Anisotropy ε)
    Volume V_min = ∞
    forall Regular Cells C_i^reg do
        Volume V ← determine_volume(C_i^reg)
        if V < V_min then
            V_min = V
        end
    end
    forall Cut Cells C_i^cut do
        if exists(C_i^cut) then
            Volume V ← determine_volume(C_i^cut)
            Face B ← determine_boundary_face(C_i^cut)
            Anisotropy ν_aniso ← determine_face_anisotropy(B, M)
            if V < V_min/2 or ν_aniso < ε then
                delete_cell(C_i^cut, M)
            end
        end
    end
end

delete_cell(Cut Cell C, Cut Cell Mesh M)
    Face B ← determine_boundary_face(C)
    delete_face_from_mesh(B, M)
    forall Faces F_j(C)\B do
        Neighbor N ← neighbor_cell(F_j, C, M)
        if exists(N) then
            define_as_boundary_face(F_j, N, M)
            if underneath(N) then
                delete_cell(N)
            end
        end
        else
            delete_face_from_mesh(F_j, M)
        end
    end
    delete_cell_from_mesh(C, M)
end
```

corners in the terrain or even to hollow spaces. To circumvent these undesirable effects, we have to check whether neighbors underneath deleted cells exist and eliminate them, too. This is done in the recursive call of function `delete_cell` in Algorithm 6.

The anisotropy $\nu_{\text{aniso}}$ of a face is calculated analogously to Section 4.5.1 as ratio of inside to outside radius. Let $F$ be a lower boundary face with corners $\boldsymbol{v}_i$ in sequential numbering, $i = 0, \ldots, k-1$ and $\boldsymbol{v}_k := \boldsymbol{v}_0$. The face's barycenter is

$$\boldsymbol{c} := \frac{1}{k} \sum_{i=0}^{k-1} \boldsymbol{v}_i. \tag{4.50}$$

Then the radius $r_{\max}$ of the circumscribed sphere is defined as the maximum distance of the barycenter to a corner

$$r_{\max}(F) = \max_i \{\|\boldsymbol{c} - \boldsymbol{v}_i\|\}. \tag{4.51}$$

To determine the radius $r_{\min}$ of the inscribed sphere, we approximate its center as barycenter and thus need to calculate its minimum distance to the face's edges. This is done by an orthogonal projection of the center onto each edge interpreted as line

$$\boldsymbol{x} = \boldsymbol{v}_i + \lambda(\boldsymbol{v}_i - \boldsymbol{v}_{i+1}) \tag{4.52}$$

with $\lambda \in \mathbb{R}$. We intersect this line with the plane

$$\boldsymbol{n}_i \cdot (\boldsymbol{x} - \boldsymbol{c}) = 0 \tag{4.53}$$

with normal vector $\boldsymbol{n}_i := \boldsymbol{v}_i - \boldsymbol{v}_{i+1}$ and position vector $\boldsymbol{c}$, and thus get the intersection point $\boldsymbol{s}_i$ of the orthogonal projection

$$\boldsymbol{s}_i = \boldsymbol{v}_i + \frac{\boldsymbol{n}_i \cdot (\boldsymbol{c} - \boldsymbol{v}_i)}{\boldsymbol{n}_i \cdot \boldsymbol{n}_i} \boldsymbol{n}_i. \tag{4.54}$$

Then the radius of the inscribed sphere is the minimum distance of the barycenter to an intersection point

$$r_{\min}(F) = \min_i \{\|\boldsymbol{c} - \boldsymbol{s}_i\|\}, \tag{4.55}$$

and we define the measure of anisotropy analogously to (4.18) for each lower boundary face $F$ as ratio

$$\nu_{\text{aniso}}(F) := \frac{r_{\min}(F)}{r_{\max}(F)} \quad \in [0, 1]. \tag{4.56}$$

The procedure is summarized in function `determine_face_anisotropy` of Algorithm 7.

So we add the anisotropy criterion to our grid generation and omit cells with highly anisotropic boundary faces. These faces appear from grid level 7 on, although their deletion is not mandatory until level 11 for stable simulations of flow between high- and low-pressure areas as described in Section 6.2.

**Algorithm 7:** Determination of the anisotropy of a face $F$ with $k$ vertices $\boldsymbol{v}_i$ in sequential numbering, $i = 0, \dots k-1$, $\boldsymbol{v}_k := \boldsymbol{v}_0$.

```
Anisotropy determine_face_anisotropy(Face F)
    Distance d_min = ∞
    Distance d_max = −1
    Vector c ← (Σ_{i=0}^{k−1} v_i(F))/k    // face centroid
    // determine outside radius
    forall Vertices v_i(F) do
        Distance d ← ‖c − v_i‖
        if d > d_max then
            d_max ← d
        end
    end
    // determine inside radius
    forall Vertices v_i(F) do
        Vector n ← v_i − v_{i+1}
        Vector s ← v_i + (n · (c − v_i)/(n · n))n
        Distance d ← ‖c − s‖
        if d < d_min then
            d_min ← d
        end
    end
    return d_min/d_max
end
```

Note that this new criterion is not very restrictive; we choose the threshold $\varepsilon = 0.1$, i.e. the ratio of radii of inscribed to circumscribed sphere may not exceed 1:10. Thus, in respect of Table 4.2, we additionally delete 13 cells for level 7, 73 cells for level 9, and 413 cells for level 11, which corresponds to approximately 0.05 % of the total number of grid cells. Furthermore, the size of the terrain area approximated by steps increases only by 0.6 %.

Altogether, we are now able to take our cut cell grids as basis for the discretization of our governing equations in the following Section 5 and finally for stable simulations of atmospheric flow problems in Section 6.

# 5

# Finite Volume Discretization

After providing the atmosphere with a three-dimensional grid in Sections 3 and 4 and thus discretizing our computational domain, the model derived in Section 2 has to be discretized, too. In other words, the continuous model equations have to be transformed into a discrete set of equations which is only valid at the grid points. In this way, we are able to solve the equations numerically and simulate atmospheric dynamics over a chosen time interval.

For discretizing the Euler equations, we favor the so-called Finite Volume method [LeV02, Krö97] over other classical discretization schemes like Finite Differences and Finite Elements. One advantage is the method's flexibility to cope even with unstructured grids. But the decisive reason is the inherent conservation property of Finite Volumes. Our governing equations were derived from basic conservation laws stating that some quantities like mass, momentum, and energy are conserved at any rate, i.e. no mass, momentum, or energy in the system is lost or appears from nowhere. This characteristic is expressed in the integral formulation of each equation, compare for instance Section 2.1.1. Finite Volume methods choose this integral form as basis for discretization, resulting in equations which conserve certain quantities by construction. This favorable characteristic makes Finite Volumes superior for models derived from conservation laws. In addition, due to the use of the integral form, Finite Volume methods are especially suitable for hyperbolic equations such as the Euler equations since they are also capable of representing discontinuous solutions.

## 5.1. Basic Principle

We first describe the general principle of Finite Volume methods before starting to discretize our system of equations. First of all, the computational domain needs to be subdivided into finite regions or volumes, from which the name of the method is derived. These volumes can be identified with our grid cells constructed in the previous

sections. Above all, the conservation property of the equations should be maintained, in fact not only for the whole computational domain but also for every single cell. Thus, we formulate the equations for each control volume and compute the temporal alteration of the conserved quantity in every cell on average. In this way, we ensure that each conserved quantity may change only by in- and outflow over the boundary of a cell.

We use the cells of our grid constructed in Sections 3 and 4 directly as control volumes and allocate the cell average of any variable in the centroid of each cell – regardless of whether it is regular or cut. Thus, all variables share the same control volumes in what is called a non-staggered arrangement.

So let $\mathcal{C}$ be a cell with center $\boldsymbol{x}_\mathcal{C}$ and volume $V_\mathcal{C}$ and $F$ a face of $\mathcal{C}$ with outward pointing unit normal vector $\boldsymbol{n}_F$, center $\boldsymbol{x}_F$ and area $A_F$. $F$ can either be a boundary face or an interior face, the latter also defining a neighbor cell $\mathcal{N}$ with center $\boldsymbol{x}_\mathcal{N}$ and volume $V_\mathcal{N}$. Each variable $\phi$ is assigned an average value $\phi_\mathcal{C} := \phi(\boldsymbol{x}_\mathcal{C})$ in each centroid. Furthermore, we divide the time interval into constant time steps of length $\Delta t$ and assume that the control volumes stay fixed over time.

As already stressed, the values of each variable are defined as cell averages, i.e.

$$\begin{aligned} \phi_\mathcal{C} &= \frac{1}{V_\mathcal{C}} \int_\mathcal{C} \phi \, dV \\ \Leftrightarrow \quad \int_\mathcal{C} \phi \, dV &= V_\mathcal{C} \phi_\mathcal{C}. \end{aligned} \tag{5.1}$$

Moreover, we state that integrating the divergence of a vector field $\boldsymbol{\psi}$ over a control volume leads to a surface integral over the faces

$$\begin{aligned} \int_\mathcal{C} \nabla \cdot \boldsymbol{\psi} \, dV &= \int_{\partial\mathcal{C}} \boldsymbol{n} \cdot \boldsymbol{\psi} \, dS \\ &= \sum_F \int_F \boldsymbol{n}_F \cdot \boldsymbol{\psi} \, dS \end{aligned} \tag{5.2}$$

according to the theorem of Gauß. Consistently assuming that $\boldsymbol{\psi}_F := \boldsymbol{\psi}(\boldsymbol{x}_F)$ is an average value of $\boldsymbol{\psi}$ on face $F$, we get analogously to (5.1)

$$\begin{aligned} \boldsymbol{n}_F \cdot \boldsymbol{\psi}_F &= \frac{1}{A_F} \int_F \boldsymbol{n}_F \cdot \boldsymbol{\psi} \, dS \\ \Leftrightarrow \quad \int_F \boldsymbol{n}_F \cdot \boldsymbol{\psi} \, dS &= A_F \boldsymbol{n}_F \cdot \boldsymbol{\psi}_F. \end{aligned} \tag{5.3}$$

Apart from that, we postulate that we can approximate the average value of a product as product of its average values

$$(\phi^1 \phi^2)_\mathcal{C} \approx \phi^1_\mathcal{C} \phi^2_\mathcal{C}. \tag{5.4}$$

These principles are the basis of the following spatial discretization of our equations.

## 5.2. Spatial Discretization

Let us now derive the discretized form of the Euler equations in a compact way. For more details see e.g. [LeV02, Krö97, Jas96, Ade08].

### 5.2.1. Governing Equations

For the spatial discretization of our governing equations (2.45) – for ease of readability in dimensionful form –, we pursue the approach of switching to their integral formulation and applying the basic Finite Volume principles derived in Section 5.1. Note that the differential form of the equations actually arises from the integral formulation, which can be directly used in the discretization process.

#### Continuity Equation

Starting with the continuity equation (2.2)

$$\rho_t + \nabla \cdot (\rho \boldsymbol{u}) = 0,$$

we spatially integrate the equation over cell $\mathcal{C}$

$$\int_{\mathcal{C}} \rho_t \, dV + \int_{\mathcal{C}} \nabla \cdot (\rho \boldsymbol{u}) \, dV = 0 \tag{5.5}$$

or use the integral formulation (2.1) of the modeling process directly. The first term can be transformed with (5.1) in

$$\int_{\mathcal{C}} \rho_t \, dV = V_{\mathcal{C}} (\rho_t)_{\mathcal{C}}, \tag{5.6}$$

the further discretization of the temporal derivation will be done in the next section. For the second term, we use (5.2) and (5.3) and get

$$\int_{\mathcal{C}} \nabla \cdot (\rho \boldsymbol{u}) \, dV = \sum_F A_F \boldsymbol{n}_F \cdot (\rho \boldsymbol{u})_F. \tag{5.7}$$

The term $A_F \boldsymbol{n}_F \cdot (\rho \boldsymbol{u})_F =: \varphi_F$ is the so-called mass flux through face $F$. Its value has to be approximated by interpolations of $\rho$ and $\boldsymbol{u}$, on which we will focus in Section 5.2.2.

All in all, we get a semi-discretized form of the continuity equation

$$V_{\mathcal{C}} (\rho_t)_{\mathcal{C}} + \sum_F \varphi_F = 0. \tag{5.8}$$

#### Momentum Equation

After integrating the momentum equation (2.45)

$$(\rho \boldsymbol{u})_t + \nabla \cdot (\rho \boldsymbol{u} \circ \boldsymbol{u}) + 2\Omega \times \rho \boldsymbol{u} + \nabla p = -\rho g \boldsymbol{k}$$

over cell $\mathcal{C}$, we transform the temporal derivative as well as the convective term in an analog way as in the continuity equation

$$\int_{\mathcal{C}} (\rho \boldsymbol{u})_t \, dV = V_{\mathcal{C}} ((\rho \boldsymbol{u})_t)_{\mathcal{C}}, \tag{5.9}$$

$$\int_{\mathcal{C}} \nabla \cdot (\rho \boldsymbol{u} \circ \boldsymbol{u}) \, dV = \sum_F \varphi_F \boldsymbol{u}_F. \tag{5.10}$$

The Coriolis and gravitational forces are treated as so-called source terms. A source term is in this context a general function of variable $\phi$ which can typically not be written as temporal derivative, convection, or diffusion. The term is at first, if necessary, linearized and then with (5.1) and (5.4) discretized. For the Coriolis and gravitational terms, this procedure leads to

$$\int_{\mathcal{C}} \boldsymbol{\Omega} \times \rho \boldsymbol{u} \, dV = V_{\mathcal{C}} \boldsymbol{\Omega} \times \rho_{\mathcal{C}} \boldsymbol{u}_{\mathcal{C}}, \tag{5.11}$$

$$\int_{\mathcal{C}} \rho g \boldsymbol{k} \, dV = V_{\mathcal{C}} \rho_{\mathcal{C}} g \boldsymbol{k}. \tag{5.12}$$

For the discretization of the pressure gradient, the theorem of Gauß is again applicable since $\nabla p = \nabla \cdot (\mathbb{1} p)$ with identity matrix $\mathbb{1}$. This leads to the discretized form

$$\int_{\mathcal{C}} \nabla p \, dV = V_{\mathcal{C}} (\nabla p)_{\mathcal{C}} = \sum_F A_F \boldsymbol{n}_F p_F. \tag{5.13}$$

Again, this expression depends on values at faces, which have to be interpolated in an appropriate way. In order to circumvent this interpolation, we use a variant for the discrete computation of the pressure gradient, which is based on a least squares ansatz [Ope12]. Here, the idea is that a value $\phi_{\mathcal{C}}$ in $\mathcal{C}$ can be extrapolated to its neighbor $\mathcal{N}$ with the aid of the gradient in $\mathcal{C}$. Such an extrapolation induces an error if compared to the actual value $\phi_{\mathcal{N}}$ in $\mathcal{N}$. If we now minimize the sum of all squared and weighted errors of all neighboring cells, the resulting gradient in $\mathcal{C}$ should be a good approximation.

Concretely, we compute the distance tensor $\boldsymbol{D}$

$$\boldsymbol{D} := \sum_{\mathcal{N}} \frac{1}{\|\boldsymbol{d}_{\mathcal{N}}\|^2} \boldsymbol{d}_{\mathcal{N}} \boldsymbol{d}_{\mathcal{N}}^{\mathsf{T}} \tag{5.14}$$

in every cell $\mathcal{C}$ with $\boldsymbol{d}_{\mathcal{N}} := \boldsymbol{x}_{\mathcal{N}} - \boldsymbol{x}_{\mathcal{C}}$ being the distance vector of cell center $\boldsymbol{x}_{\mathcal{C}}$ to the centroid of its neighbor cell $\mathcal{N}$. Then, the pressure gradient is approximated by

$$(\nabla p)_{\mathcal{C}} = \sum_{\mathcal{N}} \frac{1}{\|\boldsymbol{d}_{\mathcal{N}}\|^2} \boldsymbol{D}^{-1} \boldsymbol{d}_{\mathcal{N}} (p_{\mathcal{N}} - p_{\mathcal{C}}). \tag{5.15}$$

This discretization depends only on cell values of the pressure and thus avoids any interpolation to faces.

All in all, the spatially discretized form of the momentum equation reads

$$V_{\mathcal{C}}((\rho \boldsymbol{u})_t)_{\mathcal{C}} + \sum_F \varphi_F \boldsymbol{u}_F + 2V_{\mathcal{C}} \boldsymbol{\Omega} \times \rho_{\mathcal{C}} \boldsymbol{u}_{\mathcal{C}} + V_{\mathcal{C}}(\nabla p)_{\mathcal{C}} = -V_{\mathcal{C}} \rho_{\mathcal{C}} g \boldsymbol{k} \tag{5.16}$$

with the least squares ansatz (5.15) for the pressure gradient and the mass flux $\varphi_F$, which will be specified in Section 5.2.2.

#### Temperature Equation

The terms of the temperature equation (2.45)

$$c_v((\rho T)_t + \nabla \cdot (\rho \boldsymbol{u} T)) + p \nabla \cdot \boldsymbol{u} = Q$$

are analogously discretized to the previous equations. Integrated over cell $\mathcal{C}$, we get for the temporal derivative and the convective term

$$\int_{\mathcal{C}} (\rho T)_t \, dV = V_{\mathcal{C}}((\rho T)_t)_{\mathcal{C}}, \tag{5.17}$$

$$\int_{\mathcal{C}} \nabla \cdot (\rho \boldsymbol{u} T) \, dV = \sum_F \varphi_F T_F, \tag{5.18}$$

for the pressure dilatation term

$$\int_{\mathcal{C}} p \nabla \cdot \boldsymbol{u} \, dV = V_{\mathcal{C}} p_{\mathcal{C}} \sum_F A_F \boldsymbol{n}_F \cdot \boldsymbol{u}_F, \tag{5.19}$$

and for the unspecified source term

$$\int_{\mathcal{C}} Q \, dV = V_{\mathcal{C}} Q_{\mathcal{C}}. \tag{5.20}$$

Altogether, the spatially discretized temperature equation reads

$$c_v \left( V_{\mathcal{C}}((\rho T)_t)_{\mathcal{C}} + \sum_F \varphi_F T_F \right) + V_{\mathcal{C}} p_{\mathcal{C}} \sum_F A_F \boldsymbol{n}_F \cdot \boldsymbol{u}_F = V_{\mathcal{C}} Q_{\mathcal{C}}. \tag{5.21}$$

The interpolated values $T_F$ and $\boldsymbol{u}_F$ are specified in Section 5.2.2.

#### Equation of State

Finally, also the equation of state (2.8)

$$p = \rho R_{\text{air}} T$$

is integrated over cell $\mathcal{C}$ and discretized according to (5.1) and (5.4)

$$p_{\mathcal{C}} = \rho_{\mathcal{C}} R_{\text{air}} T_{\mathcal{C}}. \tag{5.22}$$

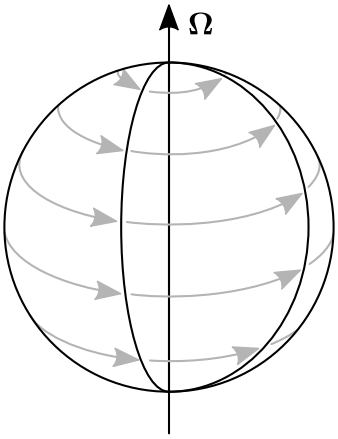



Figure 5.1.: A constantly rotating velocity field $\boldsymbol{\Omega} \times \boldsymbol{x}$ around the Earth with rotational axis $\boldsymbol{\Omega}$.

## 5.2.2. Interpolation Schemes

In Section 5.2.1, we observed that in the spatially discretized equations, we also need a few values at faces when in fact all variables are located at cell centers. So these face values $\phi_F$ have to be appropriately interpolated from the values $\phi_\mathcal{C}$ and $\phi_\mathcal{N}$ of the two adjacent cells. Here, we will use a common upwind scheme for scalar quantities as well as a new specially constructed Earth interpolation scheme for the flux velocity.

### Upwind Interpolation

The requirement of boundedness restricts our choice of an interpolation scheme for any scalar variable. Density, pressure, and temperature are quantities in a special scalar range and thus the interpolation should maintain this range and avoid e.g. negative or amplifying values. The obvious choice for assuring boundedness is the so-called upwind scheme, which takes either the value of cell $\mathcal{C}$ or that of cell $\mathcal{N}$ as face value depending on the direction of the flux through face $F$, i.e.

$$\phi_F = \begin{cases} \phi_\mathcal{C} & \text{if } \varphi_F \geq 0 \\ \phi_\mathcal{N} & \text{if } \varphi_F < 0. \end{cases} \tag{5.23}$$

In this way, we obtain interpolated values for $\rho_F$, $p_F$, and $T_F$ when required.

### Earth Interpolation

Now, only the computation of the mass flux $\varphi_F$ itself remains. Analogously to (5.4), we postulate

$$\varphi_F := A_F \boldsymbol{n}_F \cdot (\rho \boldsymbol{u})_F = A_F \boldsymbol{n}_F \cdot \rho_F \boldsymbol{u}_F \tag{5.24}$$

and thus reduce the problem to the interpolation of the velocity vector field $\boldsymbol{u}$.

The choice of an interpolation scheme for the velocity field is not as obvious as that of

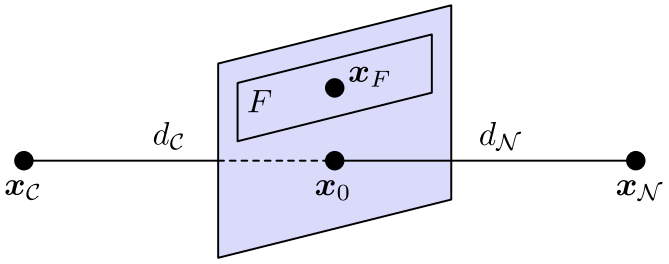



Figure 5.2.: Piercing point $\boldsymbol{x}_0$ of the line between cell centers $\boldsymbol{x}_{\mathcal{C}}$ and $\boldsymbol{x}_{\mathcal{N}}$ with the interjacent face plane.

the scalar variables. Linear interpolation may be the first intuitive idea, but this scheme is not able to cope with the non-orthogonality of grid cells and would generate values at wrong positions since the face center does not have to be in line with the adjacent cell centers, compare Section 4.1. Note that this is true even for cut cell grids of the global atmosphere since a perfectly orthogonal grid actually has to deviate slightly in order to adjust to the curvature of the Earth. So our focus lies on a non-linear interpolation scheme adjusted to the Earth's geometry, which we will construct in the following.

Consider in the first instance a velocity field which is constantly rotating around the Earth, as depicted in Figure 5.1. This velocity field

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{\Omega} \times \boldsymbol{x} \tag{5.25}$$

for all position vectors $\boldsymbol{x}$ arises from the rotation of the Earth with axis $\boldsymbol{\Omega}$. Otherwise uninfluenced, the field is characterized by three decisive properties. First, vectors on a fixed latitude at constant height are tangentially aligned and have the same length, meanwhile vectors on a fixed longitude at constant height share the same direction but differ in length, which varies linearly between zero at the poles and a maximum value of $\|\boldsymbol{\Omega}\|\|\boldsymbol{x}\|$ at the equator. Third, vectors on a radial line through the origin at the Earth's center have the same direction but their length increases linearly with the distance to the origin, starting there with zero velocity.

Now, the idea is to construct an interpolation scheme which interpolates the velocity field (5.25) perfectly by incorporating its characteristics. Let $\boldsymbol{u}_{\mathcal{C}}$ and $\boldsymbol{u}_{\mathcal{N}}$ be the velocity vectors at cell centers $\boldsymbol{x}_{\mathcal{C}}$ and $\boldsymbol{x}_{\mathcal{N}}$. Face $F$ with face center $\boldsymbol{x}_F$ connects the adjacent cells $\mathcal{C}$ and $\mathcal{N}$, and the interpolation vector $\boldsymbol{u}_F$ at face center $\boldsymbol{x}_F$ is sought. The piercing point $\boldsymbol{x}_0$ of the line $\boldsymbol{x} = \boldsymbol{x}_{\mathcal{C}} + \lambda(\boldsymbol{x}_{\mathcal{N}} - \boldsymbol{x}_{\mathcal{C}})$, $\lambda \in \mathbb{R}$, between $\boldsymbol{x}_{\mathcal{C}}$ and $\boldsymbol{x}_{\mathcal{N}}$ with the face plane $\boldsymbol{n}_F \cdot (\boldsymbol{x} - \boldsymbol{x}_F) = 0$ is

$$\boldsymbol{x}_0 = \boldsymbol{x}_{\mathcal{C}} + \frac{\boldsymbol{n}_F \cdot (\boldsymbol{x}_F - \boldsymbol{x}_{\mathcal{C}})}{\boldsymbol{n}_F \cdot (\boldsymbol{x}_{\mathcal{N}} - \boldsymbol{x}_{\mathcal{C}})}(\boldsymbol{x}_{\mathcal{N}} - \boldsymbol{x}_{\mathcal{C}}). \tag{5.26}$$

Moreover, the distances of $\boldsymbol{x}_0$ to the corresponding cell centers are $d_{\mathcal{C}} = \|\boldsymbol{d}_{\mathcal{C}}\| = \|\boldsymbol{x}_0 - \boldsymbol{x}_{\mathcal{C}}\|$ and $d_{\mathcal{N}} = \|\boldsymbol{d}_{\mathcal{N}}\| = \|\boldsymbol{x}_0 - \boldsymbol{x}_{\mathcal{N}}\|$. In Figure 5.2, the notation is visualized.

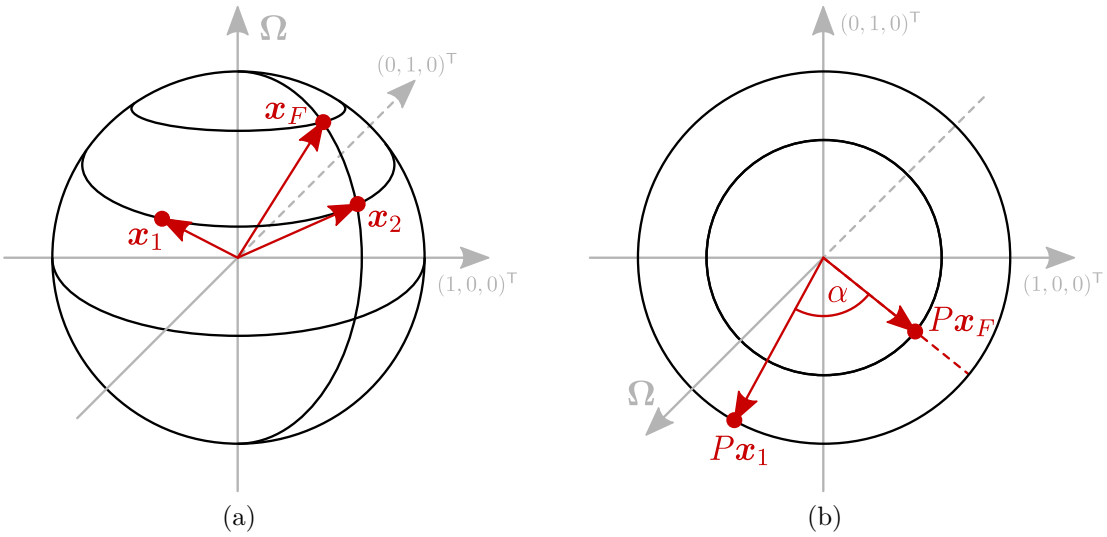



Figure 5.3.: (a) Latitude of $\boldsymbol{x}_1$ and longitude of $\boldsymbol{x}_F$ intersect in $\boldsymbol{x}_2$. (b) Projection of $\boldsymbol{x}_1$ and $\boldsymbol{x}_F$ in the equatorial plane.

With linear interpolation, we can compute any velocity vector between $\boldsymbol{x}_\mathcal{C}$ and $\boldsymbol{x}_\mathcal{N}$, in particular at the piercing point $\boldsymbol{x}_0$

$$\begin{aligned}\boldsymbol{u}_0 := \boldsymbol{u}(\boldsymbol{x}_0) &= \frac{d_\mathcal{N}}{d_\mathcal{C}+d_\mathcal{N}}\boldsymbol{u}_\mathcal{C} + \frac{d_\mathcal{C}}{d_\mathcal{C}+d_\mathcal{N}}\boldsymbol{u}_\mathcal{N} \\ &= \boldsymbol{u}_\mathcal{C} + \frac{d_\mathcal{C}}{d_\mathcal{C}+d_\mathcal{N}}\left(\boldsymbol{u}_\mathcal{N} - \boldsymbol{u}_\mathcal{C}\right). \end{aligned} \tag{5.27}$$

Following the idea that the new interpolation scheme should interpolate (5.25) perfectly, we scale and rotate the piercing point $\boldsymbol{x}_0$ according to the characteristics above to the location of $\boldsymbol{x}_F$. Applying the same scaling and rotation to its velocity $\boldsymbol{u}_0$, we obtain the interpolated velocity $\boldsymbol{u}_F$ at face center $\boldsymbol{x}_F$.

To this end, we start with scaling $\boldsymbol{x}_0$ to the length of $\boldsymbol{x}_F$ and apply the same scaling to $\boldsymbol{u}_0$, i.e.

$$\boldsymbol{x}_1 = \frac{\|\boldsymbol{x}_F\|}{\|\boldsymbol{x}_0\|}\boldsymbol{x}_0 \quad \text{and} \quad \boldsymbol{u}_1 := \boldsymbol{u}(\boldsymbol{x}_1) = \frac{\|\boldsymbol{x}_F\|}{\|\boldsymbol{x}_0\|}\boldsymbol{u}_0. \tag{5.28}$$

The auxiliary $\boldsymbol{x}_1$ is now located on the same spherical shell as $\boldsymbol{x}_F$ but still has to be rotated along latitude and longitude to the position of $\boldsymbol{x}_F$, compare Figure 5.3a.

Therefore, we first rotate $\boldsymbol{x}_1$ by $\alpha$ alongside its latitude to the longitude of $\boldsymbol{x}_F$. To determine the angle $\alpha$, a projection in the equatorial plane is applied, as depicted in

Figure 5.3b. Using the projection operator

$$P := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{5.29}$$

we calculate $\alpha$ by

$$\alpha := \sphericalangle(P\boldsymbol{x}_1, P\boldsymbol{x}_F) = \arccos \frac{P\boldsymbol{x}_1 \cdot P\boldsymbol{x}_F}{\|P\boldsymbol{x}_1\|\|P\boldsymbol{x}_F\|} \tag{5.30}$$

and eventually rotate $\boldsymbol{x}_1$ and $\boldsymbol{u}_1$ by $\alpha$ in counterclockwise direction with rotational axis $\boldsymbol{\Omega}/\|\boldsymbol{\Omega}\| = (0,0,1)^\mathsf{T}$, i.e.

$$\boldsymbol{x}_2 = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{x}_1 \quad \text{and} \quad \boldsymbol{u}_2 := \boldsymbol{u}(\boldsymbol{x}_2) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{u}_1. \tag{5.31}$$

Now, a linear scaling alongside the longitude still remains. Recalling the characteristics of velocity field (5.25), its scaling leads to the sought interpolated velocity

$$\boldsymbol{u}_F := \boldsymbol{u}(\boldsymbol{x}_F) = \frac{\|\boldsymbol{\Omega} \times \boldsymbol{x}_F\|}{\|\boldsymbol{\Omega} \times \boldsymbol{x}_2\|} \boldsymbol{u}_2. \tag{5.32}$$

Altogether, we gain $\boldsymbol{u}_F$ from the adjacent cell velocities $\boldsymbol{u}_\mathcal{C}$ and $\boldsymbol{u}_\mathcal{N}$ via the formula

$$\begin{aligned} \boldsymbol{u}_F &= \frac{\|\boldsymbol{\Omega} \times \boldsymbol{x}_F\|}{\|\boldsymbol{\Omega} \times \boldsymbol{x}_2\|} \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \frac{\|\boldsymbol{x}_F\|}{\|\boldsymbol{x}_0\|} \left( \boldsymbol{u}_\mathcal{C} + \frac{d_\mathcal{C}}{d_\mathcal{C} + d_\mathcal{N}} (\boldsymbol{u}_\mathcal{N} - \boldsymbol{u}_\mathcal{C}) \right) \\ &=: I(\boldsymbol{u}_\mathcal{C}, \boldsymbol{u}_\mathcal{N}) \end{aligned} \tag{5.33}$$

with (5.26) for $\boldsymbol{x}_0$, (5.30) for $\alpha$ and (5.31) for $\boldsymbol{x}_2$. So $I(\boldsymbol{u}_\mathcal{C}, \boldsymbol{u}_\mathcal{N})$ is our new Earth interpolation scheme, which interpolates the field $\boldsymbol{u} = \boldsymbol{\Omega} \times \boldsymbol{x}$ exactly and which we use for the velocity vector field $\boldsymbol{u}$ if values at faces are needed.

In [Ade08], we already presented a qualitative comparison of this new interpolation scheme for various flow profiles and showed its competitiveness with standard schemes such as linear or upwind interpolation.

### 5.2.3. Boundary Conditions

The interpolation schemes of the previous section were constructed for interior faces with two neighbor cells $\mathcal{C}$ and $\mathcal{N}$. But values at boundary faces with only one adjacent cell may also be required, so that we should consider the boundary conditions of Section 2.1.5 and discretize them accordingly.

Generally, boundary conditions can be classified into Dirichlet conditions, where the value of a variable is directly prescribed at each boundary face, and Neumann conditions, where the gradient in normal direction is specified. Let $F$ be a boundary face with center $\boldsymbol{x}_F$ belonging to cell $\mathcal{C}$ with center $\boldsymbol{x}_\mathcal{C}$, $\phi_F$ the value prescribed by Dirichlet conditions at face $F$ and $\boldsymbol{n}_F \cdot (\nabla\phi)_F$ the gradient in normal direction provided by Neumann conditions.

If Dirichlet conditions are specified, the value $\phi_F$ can be directly used when required e.g. in a sum over face values as customary for convective terms. If otherwise the face gradient is needed, it can be calculated by using the difference quotient

$$\boldsymbol{n}_F \cdot (\nabla\phi)_F = \frac{\phi_F - \phi_\mathcal{C}}{\|\boldsymbol{x}_F - \boldsymbol{x}_\mathcal{C}\|}. \tag{5.34}$$

For Neumann conditions, the value of the gradient in normal direction $\boldsymbol{n}_F \cdot (\nabla\phi)_F$ is known and can be directly used. If rather the value of the variable itself is required at the boundary face, we extrapolate

$$\begin{aligned} \phi_F &= \phi_\mathcal{C} + (\boldsymbol{x}_F - \boldsymbol{x}_\mathcal{C}) \cdot (\nabla\phi)_F \\ &\approx \phi_\mathcal{C} + \|\boldsymbol{x}_F - \boldsymbol{x}_\mathcal{C}\| \boldsymbol{n}_F \cdot (\nabla\phi)_F. \end{aligned} \tag{5.35}$$

### 5.2.4. Initial Values

Initial values in a discretized setting have to be defined in the variables' grid points, i.e. in the cell centers. For real atmospheric applications, measurement data from observation stations are initially used, which have to be appropriately interpolated to the grid points of the generated mesh. In our case, we are more interested in defining artificial benchmark data for numerical verification purposes. Apart from longitude and latitude, such a benchmark profile often depends on the height of each grid point.

Intuitively, the height $h$ of each point $\boldsymbol{x}$ can be determined by the vector's Euclidean length minus the radius $R$ of the Earth

$$h(\boldsymbol{x}) = \|\boldsymbol{x}\| - R. \tag{5.36}$$

But such an assignment of height values turns out to be inappropriate for cell centers, since the sphere is linearly approximated and thus only regular corners of each cell lie in the right altitude compared to the surface of the Earth, see Figure 5.4a. Note that even intersection nodes of cut cells have a faulty altitude if computed by (5.36).

To determine meaningful height values for both terrain-following and cut cell grids, we choose the following ansatz. Each cell column is assigned with a tangential plane of the base polyhedron. This plane has to be translated to the column point of choice, e.g. a cell center, for which a height value should be evaluated. Then, the intersection of the plane with an arbitrary regular cell node, interpreted as a line through the origin, defines the sought height. The principle is illustrated in Figure 5.4b and holds not only

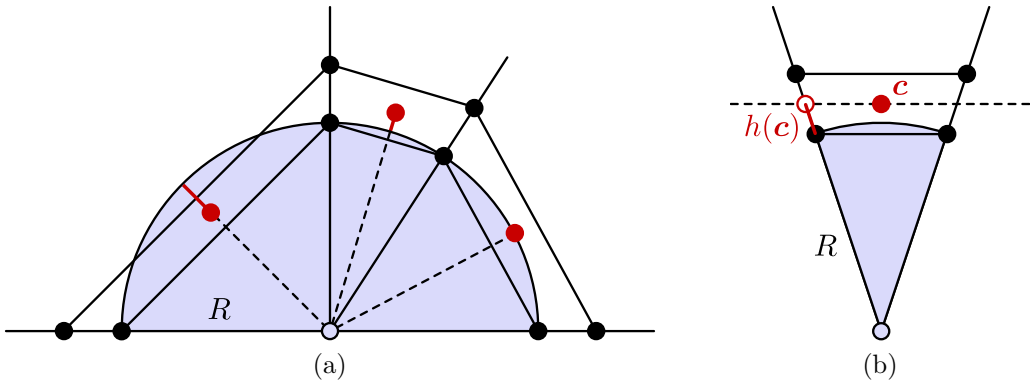



Figure 5.4.: Height evaluation with (a) (5.36) and (b) (5.39) of the lowest layer of a mesh without terrain. Cell corners are marked black, cell centers and their height above mean sea level red.

for regular cells but also for terrain-following and cut cells.

Let $F$ be a boundary face of the stratosphere, $R$ the radius of the Earth, $\boldsymbol{v}_i$ the corners of $F$ each scaled to the length of $R$ with $i \in \{0, 1, 2\}$, and $\mathcal{C}$ a cell with centroid $\boldsymbol{c}$ belonging to the column of $F$. Then we construct a plane $\boldsymbol{n} \cdot (\boldsymbol{x} - \boldsymbol{c}) = 0$ through $\boldsymbol{c}$ with normal vector

$$\boldsymbol{n} = (\boldsymbol{v}_1 - \boldsymbol{v}_0) \times (\boldsymbol{v}_2 - \boldsymbol{v}_0) \tag{5.37}$$

and compute its intersection with $\boldsymbol{v}_0$ interpreted as a line through the origin $\boldsymbol{x} = \lambda \boldsymbol{v}_0$ with $\lambda \in \mathbb{R}$. The intersection $\boldsymbol{s}$ reads

$$\boldsymbol{s} = \frac{\boldsymbol{n} \cdot \boldsymbol{c}}{\boldsymbol{n} \cdot \boldsymbol{v}_0} \boldsymbol{v}_0, \tag{5.38}$$

and its height above mean sea level as defined in (5.36) is the sought height of $\boldsymbol{c}$

$$h(\boldsymbol{c}) = h(\boldsymbol{s}). \tag{5.39}$$

This approach defines realistic height values at the cell centers independent of the grid's vertical scheme, its uniform or adaptive structure, and the inclusion of terrain.

## 5.3. Temporal Discretization

The spatial discretization of our continuous model led to equations (5.8), (5.16), (5.21), and (5.22), which still have to be discretized in time.

Therefore, let the time interval $[0, t_{\text{end}}]$ be split in discrete points with fixed time step $\Delta t$, which form the discrete set $\{n\Delta t \,|\, n \in [0, \lceil t_{\text{end}}/\Delta t \rceil]\}$. Then, we integrate each equation over a time interval $[n\Delta t, (n+1)\Delta t]$ and denote every variable $\phi$ with an upper

index representing the time step the variable belongs to

$$\phi^n := \phi(n\Delta t). \tag{5.40}$$

In the following, $\phi^n$ always represents the old known value of $\phi$ and $\phi^{n+1}$ the new value at the next time step.

Before turning to the discretization of the temporal integral, we further need an approximation of the temporal derivative in the continuity, momentum, and temperature equation, for which the common difference quotient

$$(\phi_t)_C = \frac{\phi_C^{n+1} - \phi_C^n}{\Delta t} \tag{5.41}$$

is appropriate.

However, for the time integral, various discretization schemes present themselves. Typically, we distinguish between an explicit scheme using only values from the former time step

$$\int_{n\Delta t}^{(n+1)\Delta t} \phi(t)\, dt = \mathcal{F}(\phi^n)\Delta t, \tag{5.42}$$

which is called explicit Euler method for the function $\mathcal{F}(\phi^n) := \phi^n$, an implicit scheme using unknown values from the new time step and thus leading to the solution of an equation system

$$\int_{n\Delta t}^{(n+1)\Delta t} \phi(t)\, dt = \mathcal{F}(\phi^{n+1})\Delta t, \tag{5.43}$$

which is known as implicit Euler method for $\mathcal{F}(\phi^{n+1}) := \phi^{n+1}$, and a combination of both

$$\int_{n\Delta t}^{(n+1)\Delta t} \phi(t)\, dt = \mathcal{F}(\phi^n, \phi^{n+1})\Delta t, \tag{5.44}$$

which is called Crank-Nicholson method for $\mathcal{F}(\phi^n, \phi^{n+1}) := \frac{1}{2}(\phi^n + \phi^{n+1})$. The latter is of second order and the other two are of first order. Our choice is the implicit Euler method since it proved to be more stable than the other two and moreover preserves the boundedness of the variables.

### 5.3.1. Governing Equations

Before applying the implicit Euler method (5.43) to our equation system, note that our governing equations are coupled and that $\phi$ is meant to be the variable for which the corresponding equation has to be solved. For each other variable, a known value should be used, i.e. either the value of the old time step $n$ or the new value at time step $n+1$ if it is already computed in the sequential solution procedure. In other words, assuming a solution procedure starting with the continuity equation, the newly calculated $\rho^{n+1}$ can – and should for stability reasons – already be used in the following solutions of the momentum and temperature equation.

With these remarks in mind, we now apply the temporal discretization to our system of equations. Starting with the semi-discretized continuity equation (5.8), we integrate over a time step, apply the temporal difference quotient (5.41) and the implicit Euler method (5.43) and thus get the discretized form

$$\begin{aligned} 0 &= \int_{n\Delta t}^{(n+1)\Delta t} \left( V_{\mathcal{C}}(\rho_t)_{\mathcal{C}} + \sum_F \varphi_F \right) dt \\ &\approx \int_{n\Delta t}^{(n+1)\Delta t} V_{\mathcal{C}} \frac{\rho_{\mathcal{C}}^{n+1} - \rho_{\mathcal{C}}^n}{\Delta t}\, dt + \int_{n\Delta t}^{(n+1)\Delta t} \sum_F A_F \boldsymbol{n}_F \cdot \rho_F \boldsymbol{u}_F \, dt \\ &\approx V_{\mathcal{C}}(\rho_{\mathcal{C}}^{n+1} - \rho_{\mathcal{C}}^n) + \Delta t \sum_F A_F \boldsymbol{n}_F \cdot \rho_F^{n+1} \boldsymbol{u}_F^n \\ \Rightarrow \quad 0 &= V_{\mathcal{C}} \frac{\rho_{\mathcal{C}}^{n+1} - \rho_{\mathcal{C}}^n}{\Delta t} + \sum_F A_F \boldsymbol{n}_F \cdot \rho_F^{n+1} \boldsymbol{u}_F^n. \end{aligned} \tag{5.45}$$

As already described, for the interpolated face values of density and velocity, we choose the upwind scheme (5.23) and the new Earth interpolation scheme (5.33), respectively. Solving this equation, we obtain density values $\rho^{n+1}$ at the new time step, which we can directly use to update the pressure variable by discretizing the equation of state (5.22)

$$p_{\mathcal{C}}^{n+1} = \rho_{\mathcal{C}}^{n+1} R_{\text{air}} T_{\mathcal{C}}^n. \tag{5.46}$$

The pressure gradient $(\nabla p)_{\mathcal{C}}^{n+1}$ is then constructed by a least squares ansatz as described in Section 5.2.1.

Analogously to the continuity equation, but not conducted in such detail, we proceed with the spatially discretized momentum equation (5.16), where we use the already known values of $\rho^{n+1}$ and $(\nabla p)^{n+1}$,

$$\begin{aligned} V_{\mathcal{C}} \rho_{\mathcal{C}}^{n+1} \frac{\boldsymbol{u}_{\mathcal{C}}^{n+1} - \boldsymbol{u}_{\mathcal{C}}^n}{\Delta t} + \sum_F \left( A_F \boldsymbol{n}_F \cdot \rho_F^{n+1} \boldsymbol{u}_F^n \right) \boldsymbol{u}_F^{n+1} + 2 V_{\mathcal{C}} \boldsymbol{\Omega} \times \rho_{\mathcal{C}}^{n+1} \boldsymbol{u}_{\mathcal{C}}^n + V_{\mathcal{C}} (\nabla p)_{\mathcal{C}}^{n+1} \\ = -V_{\mathcal{C}} \rho_{\mathcal{C}}^{n+1} g \boldsymbol{k}. \end{aligned} \tag{5.47}$$

Note that the explicit use of $\boldsymbol{u}^n$ in the convective and Coriolis term corresponds to a linearization of those non-linear terms.

Finally, the temperature equation (5.21) is discretized in time by again applying as many already known values of the new time step as possible

$$\begin{aligned} c_v \left( V_{\mathcal{C}} \rho_{\mathcal{C}}^{n+1} \frac{T_{\mathcal{C}}^{n+1} - T_{\mathcal{C}}^n}{\Delta t} + \sum_F \left( A_F \boldsymbol{n}_F \cdot \rho_F^{n+1} \boldsymbol{u}_F^{n+1} \right) T_F^{n+1} \right) + V_{\mathcal{C}} p_{\mathcal{C}}^{n+1} \sum_F A_F \boldsymbol{n}_F \cdot \boldsymbol{u}_F^{n+1} \\ = V_{\mathcal{C}} Q_{\mathcal{C}}^n. \end{aligned} \tag{5.48}$$

### 5.3.2. System of Linear Equations

Our governing equations (5.45) – (5.48) are now fully discretized in both space and time. In the process, each equation obtained the abstract form

$$a_{\mathcal{C}}\phi_{\mathcal{C}}^{n+1} + \sum_{\mathcal{N}} a_{\mathcal{N}}\phi_{\mathcal{N}}^{n+1} = f \tag{5.49}$$

with right hand side $f$ consisting of known values of the old time step as well as already computed values of the new time step due to the sequential solution procedure. For every cell such an equation exists, so that altogether we get a system of linear equations

$$A\boldsymbol{\phi}^{n+1} = \boldsymbol{f} \tag{5.50}$$

with $A$ being a sparse matrix with coefficients $a_{\mathcal{C}}$ on the diagonal and the neighbor values $a_{\mathcal{N}}$ in each corresponding row. $\boldsymbol{\phi}^{n+1}$ is the sought vector with new values of the variable $\phi$ for each cell and $\boldsymbol{f}$ is the known right hand side.

Altogether, by Finite Volume and implicit Euler discretization for each governing equation and each time step, we obtained a system of linear equations (5.50), which can be solved by an iteration scheme of choice.

### 5.3.3. Courant-Friedrichs-Lewy Criterion

Now, only the question of the time step size $\Delta t$ remains since the duration of the iteration process heavily depends on its size. Here, the renowned Courant-Friedrichs-Lewy (CFL) criterion [CFL28] originates, which we already mentioned in Section 4.4.4 as motivation for eliminating small cells in a cut cell approach.

Since we discretized some terms of our equation system explicitly, the choice of time step size is essential for the stability of the numerical scheme. Stability guarantees that small modifications of the input data or small errors in the approximation are not amplified in the solution process. Generally, instabilities occur if $\Delta t$ is too large, i.e. if $\Delta t$ violates the CFL criterion

$$\frac{\|\boldsymbol{u}\|\Delta t}{\Delta x} \leq 1 \tag{5.51}$$

written in its best-known one-dimensional form. This condition implies that a quantity is only permitted to move forward for up to one cell per time step, otherwise the conservation property of the scheme would be violated. So for a globally chosen time step size $\Delta t$, the worst-case condition

$$\Delta t \leq \frac{\Delta x_{\min}}{\|\boldsymbol{u}\|_{\max}} \tag{5.52}$$

or in the $N$-dimensional case [Taf95, PTVF07]

$$\Delta t \leq \frac{\Delta x_{\min}}{\sqrt{N}\|\boldsymbol{u}\|_{\max}}, \tag{5.53}$$

| Time step sizes (CFL) | | | | | | |
|---|---|---|---|---|---|---|
| level | 1 | 3 | 5 | 7 | 9 | 11 |
| terrain-following: $\Delta x_{\min}$, [m] | 24,000 | 12,000 | 6,000 | 3,000 | 1,500 | 750 |
| terrain-following: $\Delta t$, [s] | 80 | 40 | 20 | 10 | 5 | 2.5 |
| cut cells: $\Delta x_{\min}$, [m] | 18,287 | 9,848 | 3,316 | 868 | 230 | 44 |
| cut cells: $\Delta t$, [s] | 50 | 25 | 10 | 2.5 | 1 | 0.1 |

Table 5.1.: Time step sizes according to the CFL criterion (5.53) with $N = 3$ and $\|\boldsymbol{u}\|_{\max} = 180\,\mathrm{m/s}$ for our atmospheric test grids of Table 4.2. $\Delta x_{\min} = h$ for terrain-following coordinates and $\Delta x_{\min} = 2r_{\min}$ for cut cells with discarded smalls cells.

respectively, has to be fulfilled.

In our three-dimensional setting, the smallest mesh size $\Delta x_{\min}$ depends on the choice of the vertical scheme. For terrain-following coordinates, $\Delta x_{\min}$ is typically the height $h$ of the smallest horizontal layer, whereas for cut cell approaches, $\Delta x_{\min}$ may be arbitrarily small, resulting in the small cell problem we dealt with in Section 4.4.4. Furthermore, the characteristic maximal velocity $\|\boldsymbol{u}\|_{\max}$ of atmospheric flows may be chosen as $180\,\mathrm{m/s}$ which may be reached by jet streams in the upper troposphere. So the time step size has to be adjusted to each mesh to circumvent instabilities by fulfilling the CFL criterion, which is a necessary condition for convergence. For more details see [OF03] and the references therein. In Table 5.1, the resulting time step sizes are listed for our terrain-following and cut cell meshes as described in Sections 3 and 4. Here, we consider cut cell meshes with already eliminated small cells and determine $\Delta x_{\min} = 2r_{\min}$ as smallest inscribed diameter of a cell according to Section 4.5.1. Note that, for a better comparison later on, the time step sizes $\Delta t$ of cut cells in Table 5.1 are rounded down such that 1,000 s is a multiple of $\Delta t$.

As an aside, the above formulation of the CFL criterion only applies to semi-implicit discretization schemes. Fully explicit solvers generally require more restrictive CFL conditions, although these restrictions can be relaxed if the order of the temporal discretization scheme is increased [GJ08]. For explicit techniques, the condition takes the form

$$\Delta t \leq C \Delta x^{\alpha} \tag{5.54}$$

with $\Delta x \leq 1$, $C \in \mathbb{R}$ and $\alpha$ depending on the order of the time scheme. For instance, $\alpha = 2$ can be derived for the first order explicit Euler scheme and $\alpha = 4/3$ for the second order Runge-Kutta scheme, compare [Der12, SKD13]. For special cases like upwind schemes, even the space discretization order impacts stability, which leads to the

condition

$$\Delta t \leq \begin{cases} C\Delta x & \text{if } p \leq q \\ C\Delta x^{\frac{q(2p-1)}{p(2q-1)}} & \text{if } p > q \end{cases} \tag{5.55}$$

with space discretization order $p \in \mathbb{N}$ and time discretization order $q \in \mathbb{N}$ [Der12].

Generally, the concrete form of the CFL condition has to be carefully chosen and is often mismatched with the actual used scheme.

## 5.4. Convergence Theory

After discretizing the Euler equations and thus constructing sparse systems of linear equations which can be solved numerically, we have to give thought to the convergence property of Finite Volume methods as $\Delta t \to 0$ and $\Delta x \to 0$.

The history of Finite Volume methods goes back to the 1950s, so that they are comparatively young in contrast to Finite Difference and Finite Element methods. Although Finite Volumes are nowadays extensively used in engineering applications, their convergence theory is still far from completion.

The convergence theory for Finite Volume methods is most advanced for elliptic equations, because for a large class of problems cell-centered Finite Volumes yield the same stiffness matrix as Finite Elements such that the elaborate Finite Element theory can be transferred. Starting with one- and two-dimensional boundary value problems in [Hac89], the convergence theory was extended to $N$-dimensional elliptic problems in [Bey98] with special balance and regularity assumptions on the grid or dual box grid, respectively, see also [LeV02, Krö97].

But for parabolic and hyperbolic problems, in which we are specially interested, the theory is much less advanced. The most important problems for hyperbolic equations are that shocks complicate the analysis exceedingly and that their solution is not necessarily unique, so even if the scheme converges for diminishing mesh sizes and time steps, the limit may not be the correct physical one.

Following the overviews of up-to-date Finite Volume methods and their convergence theory in [EGH00, BO04], we give a short summary of current convergence results for hyperbolic equations. In general, convergence of a Finite Volume scheme can be deduced from its conservativity, a consistency of approximation, and some stability properties.

#### Scalar hyperbolic equation in one dimension

Starting with a scalar hyperbolic equation in one space dimension, convergence can be shown based on "strong bounded variation estimates" of the approximate solutions and the renowned theorem of Lax-Wendroff [LW60]. It states that if a scheme is conservative with continuous and consistent fluxes and if its approximate solutions form a bounded almost everywhere convergent sequence, then the limit must be a weak solution to the problem. In other words, if the scheme converges then its limit is a weak solution. In the linear case, this weak solution is unique, whereas in the non-linear case, the numerical

solutions have to additionally satisfy "entropy inequalities" to verify that the limit is a so-called "entropy weak solution". Generally, entropy weak solutions are widely assumed to be the unique physically correct solutions although this has only been rigorously proven in special cases [Ell04].

As an aside, the theorem of Lax-Wendroff originally only applies to uniform Cartesian grids but in the meantime, it has also been verified for unstructured but quasi-uniform grids in [Ell04]. Note furthermore that special attention has to be paid to boundary conditions since most of the proofs deal with initial data but without considering boundary conditions.

**Scalar hyperbolic equation in multiple dimensions**

The convergence theory in one dimension can be extended to the multidimensional case for tensor product Cartesian meshes [BO04] but seemingly not for general meshes, and an extension has only recently been undertaken as remarked in [EGH00]. Hence, for scalar hyperbolic equations in multiple dimensions, the approximate solutions can only be shown to fulfil some weaker estimates, namely "weak bounded variation estimates". Together with $L^\infty$ stability and satisfied entropy inequalities, an analog proposition to the Lax-Wendroff theorem can be derived, stating that such numerical approximations converge to the unique entropy weak solution.

**Systems of hyperbolic equations in multiple dimensions**

In order to include the Euler equations, a final extension of the convergence theory to hyperbolic systems in multiple dimensions would be necessary, a still open subject of ongoing research. If a consistent conservative scheme for systems could be shown to boundedly converge almost everywhere, then the theorem of Lax-Wendroff would imply that the limit is a weak solution and – in the presence of a discrete entropy condition – also an entropy solution. But such a confirmation is known to be difficult for hyperbolic systems, although observations of actual solutions of approved numerical schemes suggest that bounded almost everywhere convergence is rather common, also for important systems of gas dynamics like the Euler equations [Ell04]. Apart from that, the uniqueness of entropy solutions was only verified for scalar conservation laws in multiple dimensions [Kru70] with the two-dimensional Euler system as a counterexample [Ell03].

As a conclusion, only very few theoretical results are available and most of them are also limited to special cases, see for instance [GR96] and the references therein. Even for the continuous problem of hyperbolic equation systems little is known, so that the complex approximation schemes are primarily justified by numerical evidence rather than by a rigorous mathematical proof [EGH00, Ell04].

# 6 Numerical Simulations

Finally, we turn towards the concrete solution of our equation system. Having modeled and accordingly discretized the dynamical core of atmospheric dynamics, we now wish to test our specially constructed cut cell grids in different simulation runs. To this end, we incorporate our atmospheric grids, governing equations, discretizations, and interpolation schemes into the open source computational fluid dynamics (CFD) toolbox OpenFOAM [Ope13], see Appendix A.2 for more details.

Testing the dynamical core of three-dimensional general circulation models (GCM) is not straightforward. The simulation results can neither be compared with analytical solutions since no non-trivial solutions are known, nor be verified by actual measurements because the dynamical core is isolated from the physical parameterization and thus not a weather forecast in the classical sense. Therefore, model evaluations have to rely on intuition, experience, and model intercomparisons [JW06b].

Whereas a test suite for two-dimensional shallow water models has long been standardized [WDH+92], a set of three-dimensional benchmarks for atmospheric GCMs is only presently established [JLNT08] together with a community devoted to the intercomparison of different GCMs, the so-called Dynamical Core Model Intercomparison Project (DCMIP) [UJK+12].

Beforehand, three-dimensional test cases were rare to find. Early ones were either long-term benchmarks for climate models as suggested by [HS94, BD97] or generalizations of shallow water test cases like a three-dimensional version of the classical Rossby-Haurwitz test case proposed in [GR04]. One of the very few short-term tests, which were specially developed for a three-dimensional GCM, is the baroclinic instability test [JW06a, UMJS13], where a perturbation of a vertically-sheared basic state in geostrophic and hydrostatic balance triggers the evolution of a baroclinic wave in mid-latitudes.

Here, we aim at basic tests of the capabilities of cut cells grids and the new Earth interpolation scheme. Thus, we start with a fundamental advection test in Section 6.1 and increase the complexity by simulating counterbalancing flow between high- and

low-pressure areas in Section 6.2, where terrain-following grids fail by developing severe instabilities. An outlook to even more complex simulations is given by the Rossby-Haurwitz test case in Section 6.3.

## 6.1. Advection Test

As a first benchmark test, we solve a scalar transport equation, i.e. the continuity equation (2.2)

$$\rho_t + \nabla \cdot (\rho \boldsymbol{u}) = 0,$$

and study the behavior of a scalar $\rho$ under the influence of a fixed velocity field $\boldsymbol{u}$. This is both a test of the quality of the new Earth interpolation scheme as well as a test of the numerical accuracy of our method. It is also a standard benchmark test which can be found in many variants in the literature, e.g. in [WDH+92, KME02, JLNT08].

As described in detail in Section 5, we discretize the continuity equation with the Finite Volume method in space and the implicit Euler method in time and gain the discretized form (5.45)

$$V_C \frac{\rho_C^{n+1} - \rho_C^n}{\Delta t} + \sum_F A_F \boldsymbol{n}_F \cdot \rho_F^{n+1} \boldsymbol{u}_F^n = 0.$$

For computing the interpolated values $\rho_F^{n+1}$, we choose the upwind scheme (5.23) to assure boundedness and for $\boldsymbol{u}_F^n$ the new Earth interpolation scheme (5.33). As boundary conditions, we employ slip conditions (2.11) for the velocity field $\boldsymbol{u}$ and Neumann zero conditions (2.10) for the density $\rho$. In this way, mass flux across the boundaries is avoided. Compare Section 5.2.3 for the discretization of these conditions.

We solve the transport equation fully implicitly so that we are allowed to use large time steps with no restriction of the CFL criterion. Thus, we avoid the small cell problem and may choose the original cut cell grids without special treatment of very small cells.

### 6.1.1. Initial Values

The fixed velocity field for our benchmark test is a cyclic flow resulting from the rotation of the Earth

$$\boldsymbol{u}(\boldsymbol{x}) := \frac{1}{12} \boldsymbol{\Omega} \times \boldsymbol{x} \tag{6.1}$$

as depicted in Figure 5.1. With this choice, a complete rotation on each orbit takes the time

$$\frac{2\pi}{\frac{1}{12}\|\boldsymbol{\Omega}\|} \approx 12 \text{ days} = 1{,}036{,}800\,\text{s}, \tag{6.2}$$

which we choose as our end time of the simulation.

For the density $\rho$, we construct a $\cos^2$-cylinder as initial condition and expect this cylinder to be transported by the flow in twelve days around the Earth. Let $\boldsymbol{x} = (x_1, x_2, x_3)^\mathsf{T}$ be a position vector with distance

$$d(\boldsymbol{x}) := \sqrt{x_2^2 + x_3^2} \tag{6.3}$$

to the first coordinate axis and $r := 3{,}000{,}000\,\mathrm{m}$ the radius of the cylinder. Then, the initial field of density for each cell center $\boldsymbol{x}$ is defined as

$$\rho^0(\boldsymbol{x}) := \begin{cases} 1 + \cos^2\left(\frac{\pi d(\boldsymbol{x})}{2r}\right)\,\mathrm{kg/m^3} & \text{if } x_1 < 0 \text{ and } d \leq r, \\ 1\,\mathrm{kg/m^3} & \text{else.} \end{cases} \tag{6.4}$$

In this way, the center of the cylinder lies in the Pacific, i.e. at the negative first coordinate axis.

## 6.1.2. Simulation Results

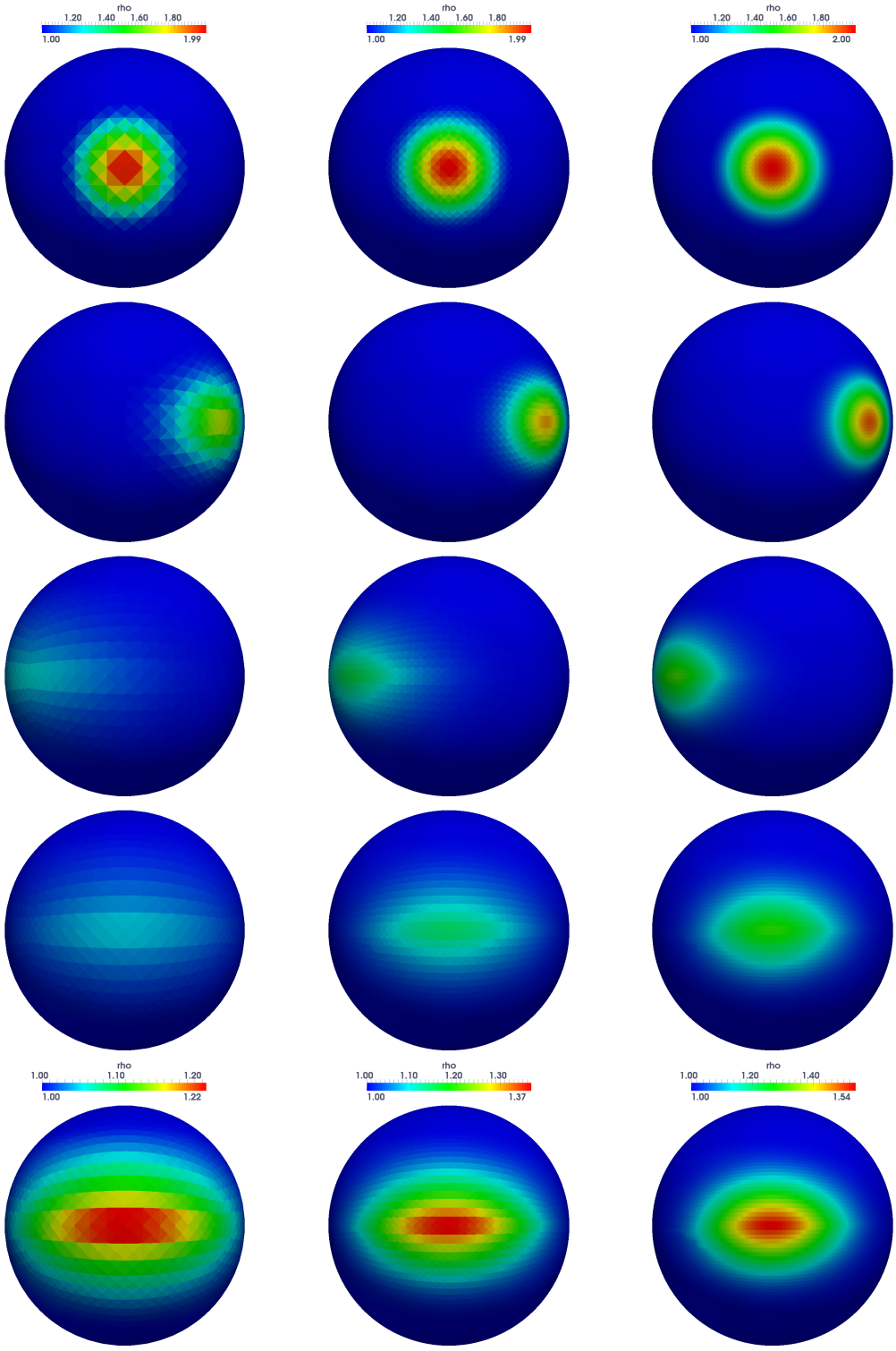
With this initial situation, we expect the density cylinder to be transported around the Earth and to arrive at the starting point after twelve days of simulated time. Without any diffusion and boundary friction, the cylinder should be perfectly conserved, but due to the approximate solutions on discrete grids, numerical diffusion typically occurs. This is expected to lead to a deformation of the circular base area of the cylinder to an ellipse extending in the flow direction with lower maxima of $\rho$. And that is exactly what we observe in our simulation. Figure 6.1 shows three cut cell grids of different resolutions and the course of the density cylinder at different points in time. With finer resolution, the deformation is significantly reduced so that in the limit a perfectly conserved circular cylinder can be expected.

A quantification for cut cell and terrain-following grids is depicted in Figure 6.2. Here, the absolute error is chosen as mean $l_2$-error defined as

$$e(\rho) := \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} \left(\rho(\boldsymbol{x}_i) - \rho_{\mathrm{ref}}(\boldsymbol{x}_i)\right)^2} \tag{6.5}$$

with $N$ being the number of grid cells. As reference values $\rho_{\mathrm{ref}}$, we use the initial state of $\rho$ at the finest grid, that is at grid level 13 with 32 layers. Of course, the evaluation of (6.5) requests a prolongation, i.e. an interpolation of $\rho$ at coarse grids to the reference mesh. For this step, we choose the Kriging algorithm, a method originating in geostatistics for the interpolation of measurement data. Here, in contrast to the common inverse distance weighting, also the spatial variance of data points is respected. For details see [Kri51, OW90].

As depicted in Figure 6.2, both cut cell and terrain-following grids show the same convergence behavior, although the asymptotic regime is not yet reached. But the ten-




(a) (b) (c)

Figure 6.1.: Transport of a $\cos^2$-cylinder of density with cut cell grids of level (a) 9, (b) 11, and (c) 13. Depicted is from top to bottom the density at 0 s, 129,600 s, 907,200 s, and 1,036,800 s with the final state rescaled in the bottom row.

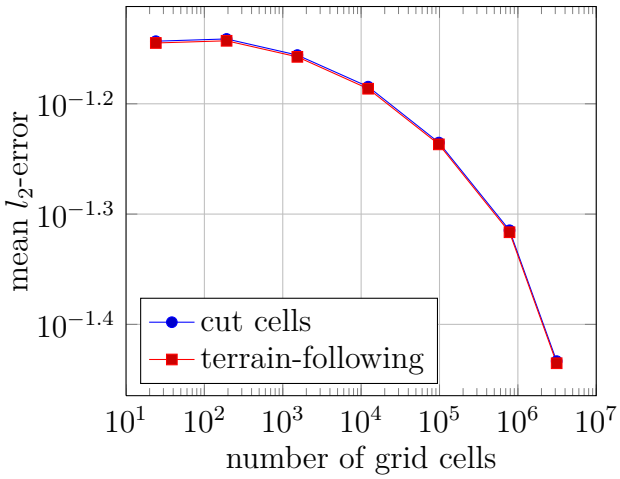
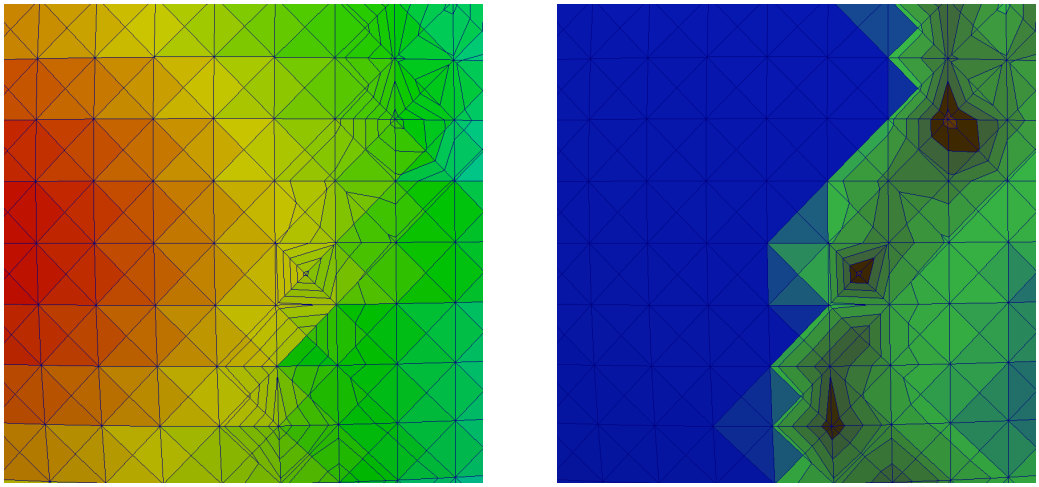



Figure 6.2.: Convergence plot of the transport of a $\cos^2$-cylinder of density in double logarithmic representation. Error (6.5) compares the final states of $\rho$ after one rotation with the initial state on the finest grid as reference solution.



(a) (b)

Figure 6.3.: Transport of a $\cos^2$-cylinder of density. Detail of the Andes with cut cells at the bottom of the grid. Colored is (a) the density and (b) the height.

dency is obvious and assures that simulations on finer grids would show the expected asymptotic convergence rate of one.

Finally, let us have a closer look at cut cells and their influence on the transport of $\rho$. Figure 6.3 shows a detail of the lower boundary with small cut cells and a great slope of terrain. The distribution of density is as smooth as desired and no significant disturbance is generated by the irregular cut cells, which is basically an effect of the adequate Earth interpolation scheme of the mass flux, compare the study in [Ade08]. Generally, the terrain itself has only marginal influence on the density distribution since after all the height of the cylinder extends to the stratosphere.

So this benchmark is a good first test of the principle capability of cut cell grids.

## 6.2. High- and Low-Pressure Areas

For the next benchmark step, we extend the continuity equation (2.2) by the momentum equation (2.45)

$$(\rho \boldsymbol{u})_t + \nabla \cdot (\rho \boldsymbol{u} \circ \boldsymbol{u}) + 2\boldsymbol{\Omega} \times \rho \boldsymbol{u} + \nabla p = -\rho g \boldsymbol{k}$$

and thus by a dynamic velocity field $\boldsymbol{u}$. To close the equation system, we add the equation of state (2.8)

$$p = \rho R_{\text{air}} T$$

with a fixed temperature field $T$. We solve the equations sequentially and use the newly computed variable directly in the next solution step. The following order has proven to be the most stable. First, we solve the discretized continuity equation (5.45)

$$V_{\mathcal{C}} \frac{\rho_{\mathcal{C}}^{n+1} - \rho_{\mathcal{C}}^{n}}{\Delta t} + \sum_F A_F \boldsymbol{n}_F \cdot \rho_F^{n+1} \boldsymbol{u}_F^n = 0$$

and gain $\rho^{n+1}$, with which we compute $p^{n+1}$ using the discretized equation of state (5.46)

$$p_{\mathcal{C}}^{n+1} = \rho_{\mathcal{C}}^{n+1} R_{\text{air}} T_{\mathcal{C}}.$$

Then, an update of the velocity $\boldsymbol{u}^{n+1}$ completes the sequence of one iteration step by solving the discrete momentum equation (5.47)

$$V_{\mathcal{C}} \rho_{\mathcal{C}}^{n+1} \frac{\boldsymbol{u}_{\mathcal{C}}^{n+1} - \boldsymbol{u}_{\mathcal{C}}^{n}}{\Delta t} + \sum_F \left( A_F \boldsymbol{n}_F \cdot \rho_F^{n+1} \boldsymbol{u}_F^n \right) \boldsymbol{u}_F^{n+1} + 2V_{\mathcal{C}} \boldsymbol{\Omega} \times \rho_{\mathcal{C}}^{n+1} \boldsymbol{u}_{\mathcal{C}}^n + V_{\mathcal{C}} (\nabla p)_{\mathcal{C}}^{n+1} = -V_{\mathcal{C}} \rho_{\mathcal{C}}^{n+1} g \boldsymbol{k},$$

where the variables density and pressure are already taken at the new time step. Again, for computing interpolated face values, we choose the upwind scheme (5.23) for $\rho_F^{n+1}$ and the new Earth interpolation scheme (5.33) for $\boldsymbol{u}_F^n$. Moreover, the pressure gradient $(\nabla p)_{\mathcal{C}}^{n+1}$ is constructed by a least squares ansatz as described in Section 5.2.1. As

boundary conditions, we implement those listed in Section 2.1.5 and discretize them according to Section 5.2.3.

The semi-implicit solution procedure demands a restriction on the time step size $\Delta t$. Thus, according to the CFL criterion studied in detail in Section 5.3.3, we employ the time step sizes of Table 5.1.

Note that we need a further stabilizing step already anticipated in Section 4.7.3 and Figure 4.30. Apart from omitting small cells, we have to further eliminate cells with very anisotropic boundary faces, i.e. with a greater ratio of inside to outside radius than 1:10. As already emphasized, this is not very restrictive since it only affects up to 0.05 % of the total number of grid cells. Moreover, for the current simulation, this stabilizing step would in fact only be mandatory from grid level 11 on.

### 6.2.1. Initial Values

Let us now have a look at the concrete initial values chosen for this benchmark test. We start with several high- and low-pressure areas evenly distributed throughout the atmosphere to examine the counterbalancing fluid flows which appear between those areas.

For simplicity, we choose a linear decay of pressure with height

$$p(h) = -3.75h + 10^5 \,\mathrm{Pa} \tag{6.6}$$

such that pressure decreases from $10^5$ Pa at mean sea level to $10^4$ Pa at 24 km. Initially, we postulate that the atmosphere is in hydrostatic balance (2.48), so the density has to be chosen as

$$\rho(h) = \frac{3.75}{g} \,\mathrm{kg/m^3} \tag{6.7}$$

and the fixed temperature profile due to the equation of state as

$$T(h) = \frac{g}{3.75 R_{\mathrm{air}}} \left(-3.75h + 10^5\right) \,\mathrm{K}. \tag{6.8}$$

Note that $h$ always refers to realistic height values as described in Section 5.2.4.

Now, we construct horizontally distributed high- and low-pressure areas in the following way. Let $\alpha \in [0, 2\pi)$ be the longitude and $\beta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ the latitude of each atmospheric point $\boldsymbol{x} = (x_1, x_2, x_3)^{\mathsf{T}}$. Longitude and latitude can be evaluated by the formulae

$$\alpha(\boldsymbol{x}) = \begin{cases} \pi + \arccos \frac{x_1}{\sqrt{x_1^2 + x_2^2}} & \text{if } x_1 \geq 0 \\ \pi - \arccos \frac{x_1}{\sqrt{x_1^2 + x_2^2}} & \text{if } x_1 < 0 \end{cases}, \tag{6.9}$$

$$\beta(\boldsymbol{x}) = \frac{\pi}{2} - \arccos \frac{x_3}{\|\boldsymbol{x}\|}. \tag{6.10}$$

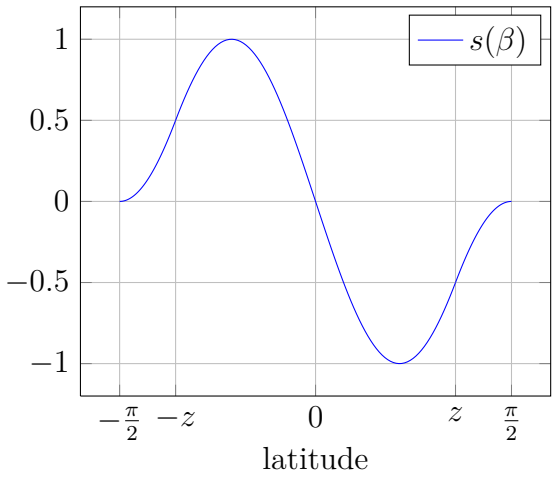



Figure 6.4.: The function $s(\beta)$ (6.11) which is smoothly decreasing towards the poles.

A function of the form $f(\alpha, \beta) = \sin(2\alpha)\sin(2\beta)$ would define two circular high- and low-pressure areas each on the northern and southern hemisphere, but since the large gradients of this function in the circumpolar regions should be avoided, we construct a sine function of latitude $\beta$ which is smoothly decreasing towards the poles

$$s(\beta) := \begin{cases} \sin\left(\frac{2n}{\pi}\beta + \pi - 2\arcsin\left(\frac{1}{2}\right)\right) + 1 & \text{if } \beta \in \left[-\frac{\pi}{2}, -z\right) \\ \sin\left(\frac{2n}{\pi}\beta - \pi + 2\arcsin\left(\frac{1}{2}\right)\right) - 1 & \text{if } \beta \in \left(z, \frac{\pi}{2}\right] \\ \sin\left(\frac{2n}{\pi}\beta + \pi\right) & \text{if } \beta \in [-z, z] \end{cases} \tag{6.11}$$

with

$$n := \frac{3\pi}{2} - 2\arcsin\left(\frac{1}{2}\right)$$

and the connection point

$$z := -\frac{\pi}{2n}\left(\arcsin\left(\frac{1}{2}\right) - \pi\right).$$

The function $s(\beta)$ is depicted in Figure 6.4.

Together with the above postulated atmospheric layering of density, pressure, and temperature, we get the initial values of high- and low-pressure areas

$$\rho^0 = \frac{3.75}{g} + \frac{1}{100}\sin(2\alpha)s(\beta)\,\mathrm{kg/m^3} \tag{6.12}$$

and

$$p^0 = \rho^0 R_{\text{air}} T \tag{6.13}$$

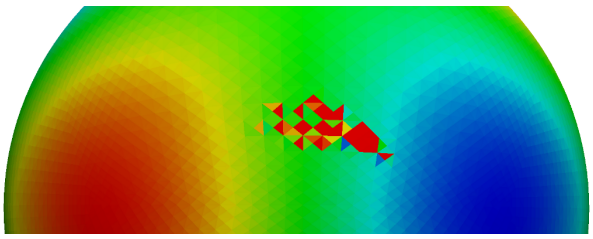


Figure 6.5.: Typical instabilities in a smooth density distribution for terrain-following grids as they arise with increasing intensity for finer resolutions.

with fixed height-dependent $T$ (6.8). Finally, the velocity field starts with

$$\boldsymbol{u}^0 = \vec{0}\,\mathrm{m/s} \tag{6.14}$$

and the end time of our simulation is 28,000 s $\approx$ 8 hours.

Before looking at the simulation results, a remark on the pressure gradient should be made. With the equation of state, the pressure gradient can be split in

$$\begin{aligned} \nabla p &= \nabla(\rho R_{\text{air}} T) \\ &= R_{\text{air}} T \nabla \rho + \rho R_{\text{air}} \nabla T. \end{aligned} \tag{6.15}$$

In this benchmark, the gradient of $T$ is analytically known as

$$\nabla T = -\frac{g}{R_{\text{air}}} \boldsymbol{k}. \tag{6.16}$$

So the second term in (6.15) equals the gravitational force, and the first term impels the flow due to the density gradient. Thus, we can write

$$\nabla p = R_{\text{air}} T \nabla \rho - \rho g \boldsymbol{k}, \tag{6.17}$$

and so it suffices to compute the density gradient $\nabla \rho$ in each iteration step with the least squares method, which provides good approximations for mostly horizontal gradients, and then compose the pressure gradient via (6.17).

### 6.2.2. Simulation Results

Let us now examine our simulation results. Note beforehand that terrain-following grids are not able to cope with this benchmark. From grid level 9 on, severe instabilities occur leading to an early abort of the simulation run, see Figure 6.5. Moreover, the instabilities intensify with finer mesh resolutions as we already anticipated in Section 4.3 due to the pressure gradient force error and hydrostatic inconsistency of these vertical coordinates. Therefore, in the following, we study only the stable simulations of our cut cell grids.

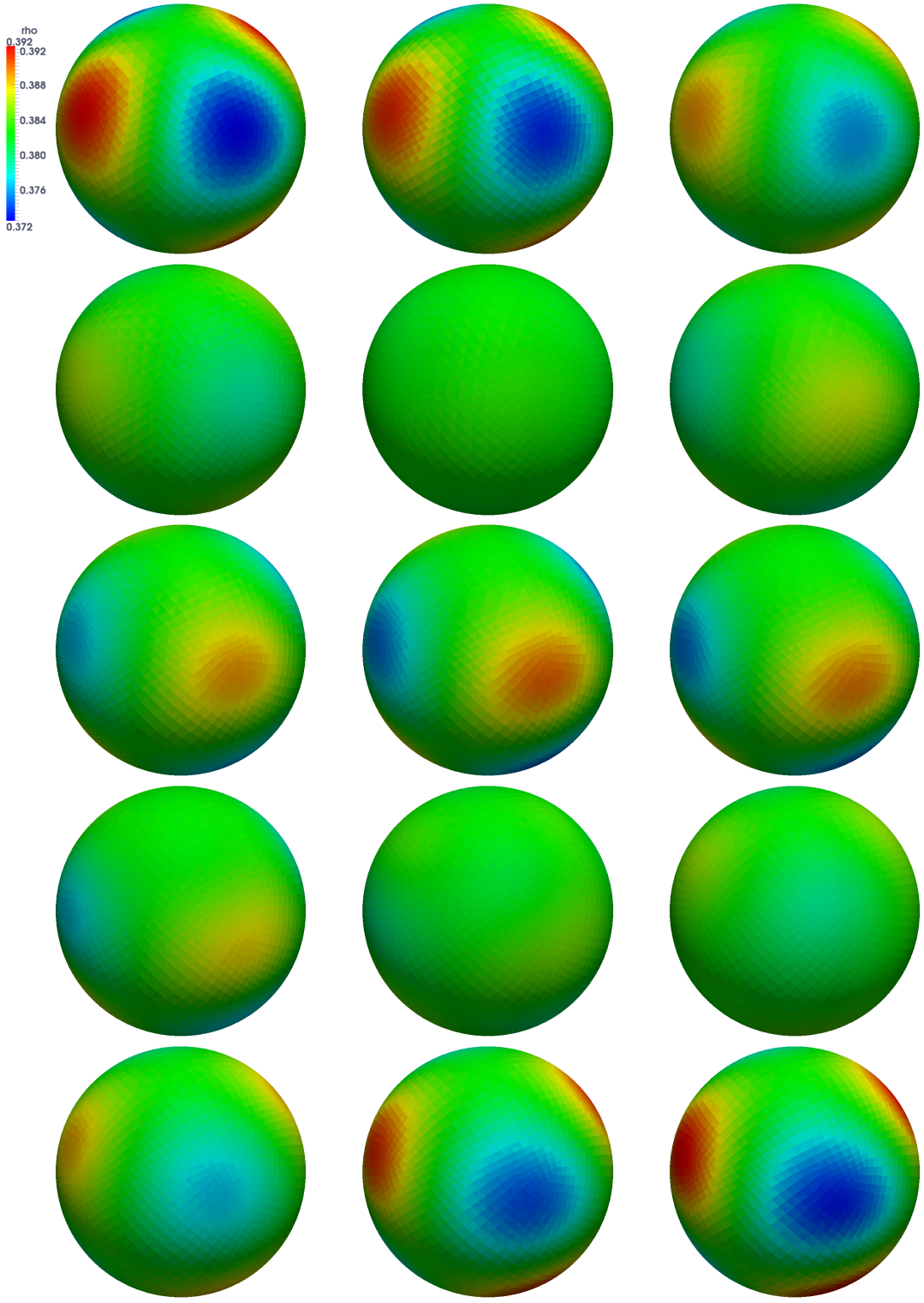



Figure 6.6.: Evolution of high- and low-pressure areas with cut cell grid 9. Depicted is from left to right and top to bottom the density at 0 s, 2,000 s, 4,000 s, ..., 28,000 s.

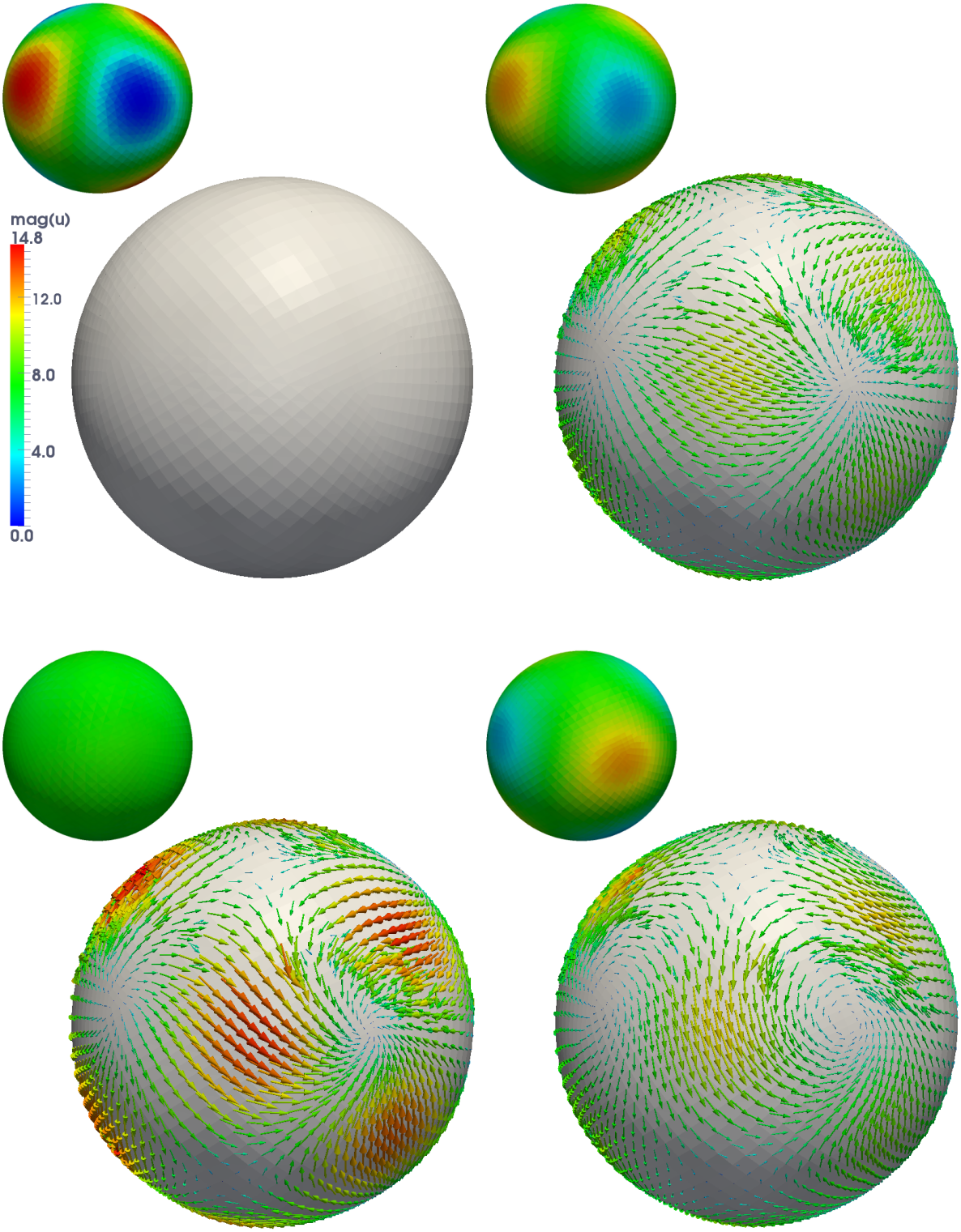



Figure 6.7.: Evolution of the flow between high- and low-pressure areas with cut cell grid 9. Depicted are from left to right and top to bottom the velocity vectors at 0 s, 4,000 s, 8,000 s, and 12,000 s accompanied by the small-sized corresponding density distributions.

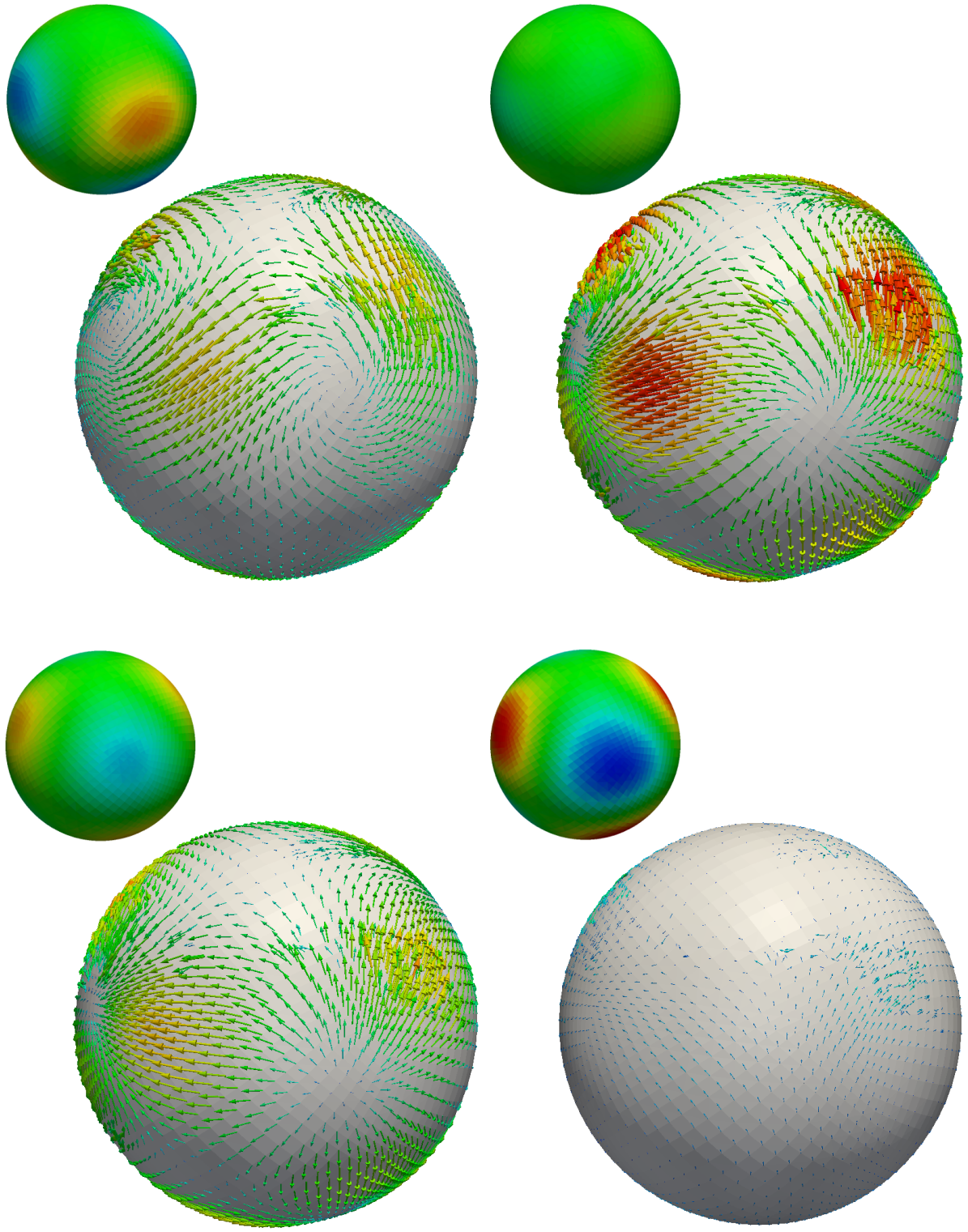
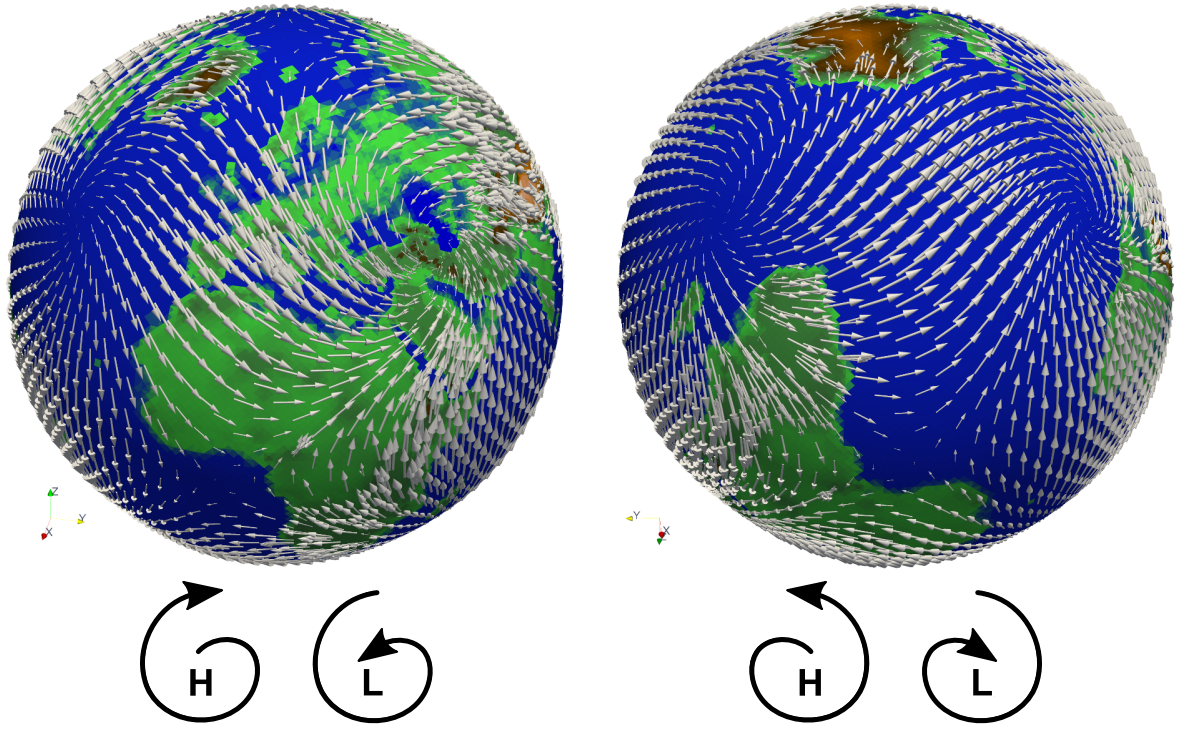


Figure 6.7.: Continuation of the preceding page with velocity vectors and density distributions at 16,000 s, 20,000 s, 24,000 s, and 28,000 s.




(a) Northern hemisphere (b) Southern hemisphere

Figure 6.8.: Effect of the Coriolis force on the rotating directions of high- and low-pressure areas on (a) northern and (b) southern hemisphere.

The evolution of density and thus the high- and low-pressure areas is shown in Figure 6.6 and the corresponding flow fields in Figure 6.7. As expected, counterbalancing flow is formed, streaming from high- towards low-pressure areas. The transport of density via this fluid flow leads to a balancing effect between the areas of different pressure. Due to the inherent inertia of the physical system, the flow does not stop immediately with equalized pressure but continues to stream, so that the high- and low-pressure areas appear again but now directly opposed. This influences the flow to that effect that the velocity decreases and the flow is finally converted in the opposite direction. The repetition of this process finally leads to one cycle during our simulated time as shown in Figure 6.6 and 6.7. The initial pressure areas appear again, but now slightly shifted in equatorial and counterclockwise direction. This shift is an effect of the Earth's rotation.

The influence of the Coriolis force on the flow can also be very well observed, see Figure 6.8. As already discussed in the modeling context of Section 2.1.2, the Coriolis force deflects every motion in the atmosphere which is non-parallel to the Earth's axis and has thus an effect on the rotating directions of high- and low-pressure areas on northern and southern hemisphere. On the northern hemisphere, flow streams clockwise out of a high-pressure area and counterclockwise into a low-pressure area. On the southern

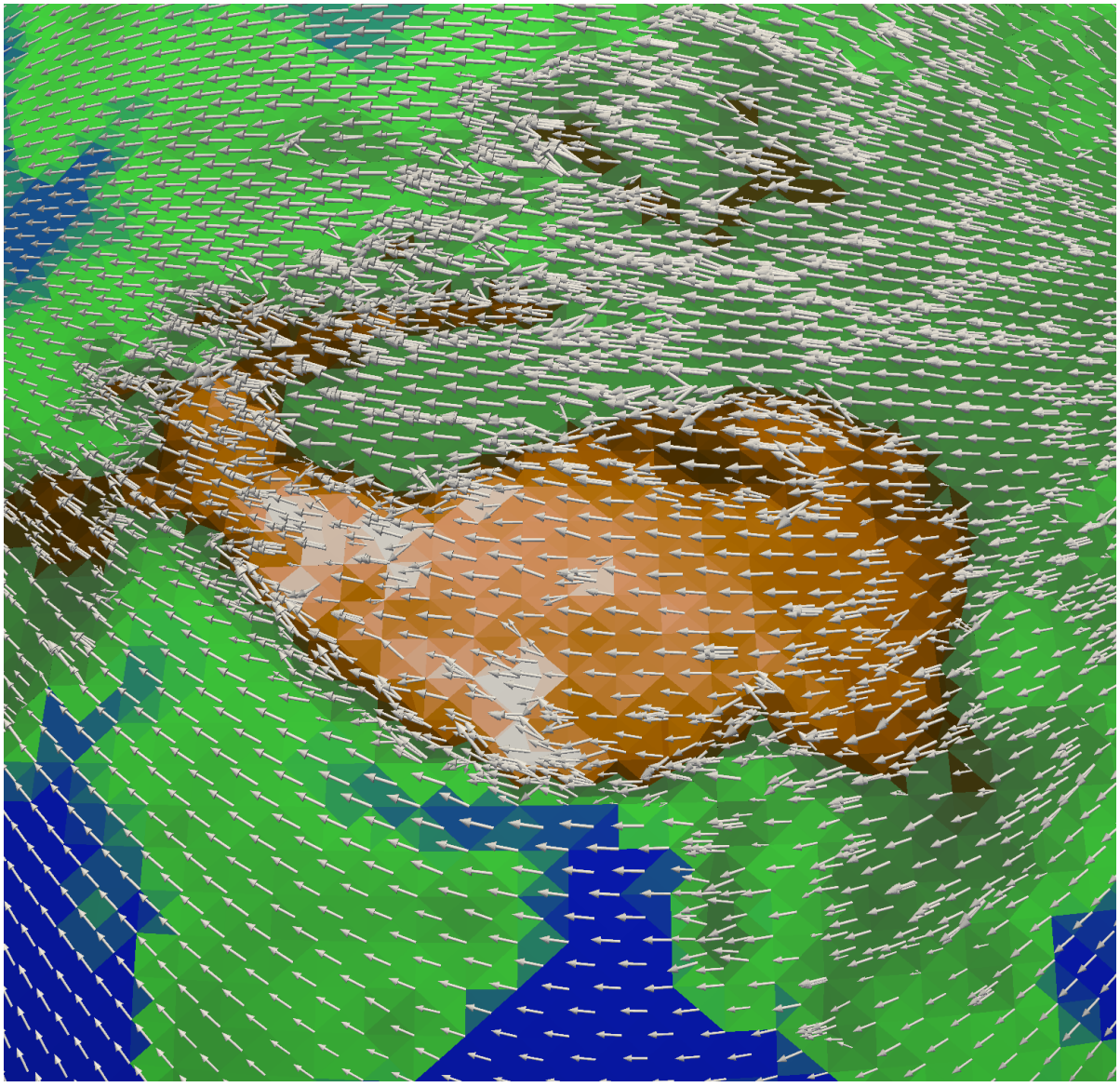


Figure 6.9.: Detail of the flow at ground level. The Himalayas deflect the overall background flow from east to west. Depicted is a simulation on a cut cell mesh of level 13 with 16 layers.

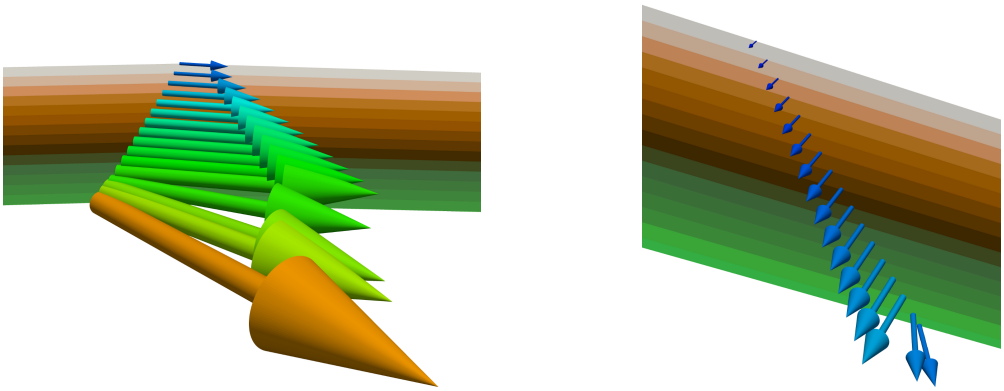


Figure 6.10.: Vertical slices of the atmosphere with height-colored layers. Near ground, the velocity vectors are deflected by the topography.

hemisphere, these directions are vice versa as can be seen in Figure 6.8.

Naturally, also the topography of the Earth has influence on the flow. In Figure 6.9, a detail of the flow at ground level is depicted, where the Himalayas form an obstacle for the overall background current from east to west. Thus, the flow is deflected by the Himalayas, especially at their horizontal boundaries where steep slopes of terrain force the flow into detours. Additionally, in Figure 6.10, vertical slices of the atmosphere above rough terrain are shown. Here, too, the deflection of velocity vectors at ground level can be observed. Note that the decrease of the velocities' magnitudes with height is due to the descreasing pressure.

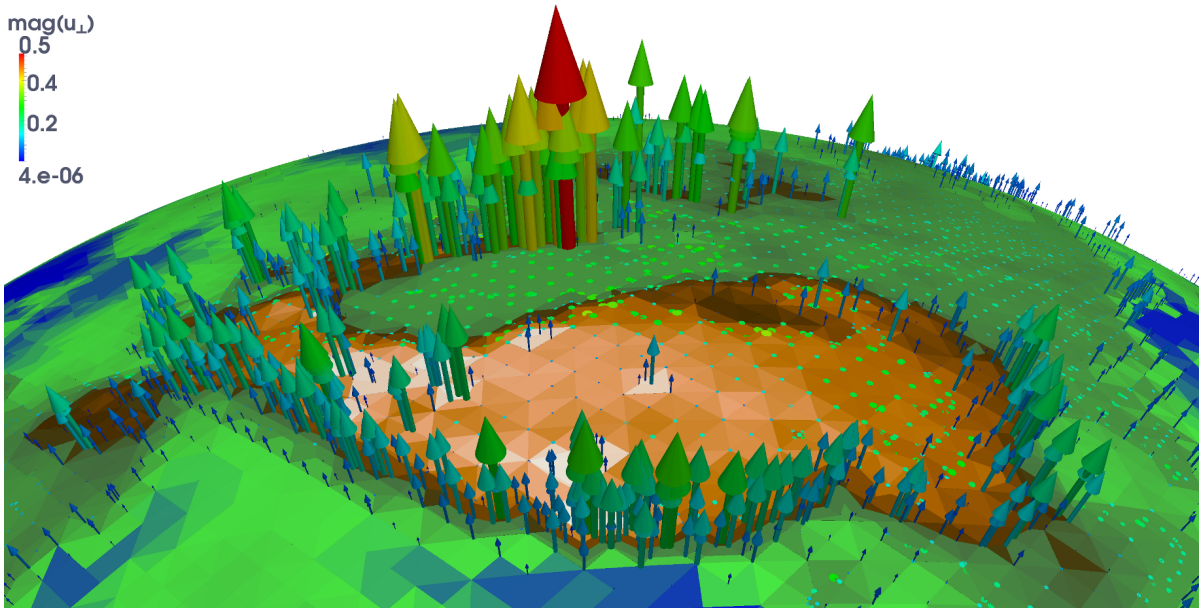
In this context, let us emphasize that the flow has also a vertical component. We started with initial fields in hydrostatic balance and a pure horizontal density gradient driving the overall flow. But in the course of time, the gradient develops also a vertical component due to the topography and thus disturbs the hydrostatic balance. As a result, especially in areas with rough terrain and steep slopes, the horizontal flow pattern is deflected in vertical direction. Since the vertical component $\boldsymbol{u}_\perp$ of the velocity field is comparatively small, we separately visualized it in Figure 6.11. The several colored dots on the ground, especially apparent in Figure 6.11a, are vectors pointing downwards.

Finally, we turn to the study of the convergence behavior of our method. On that account, we simulate this benchmark on each of our cut cell test grids (4.2) and compare the results at different points in time with the solutions on the finest grid.

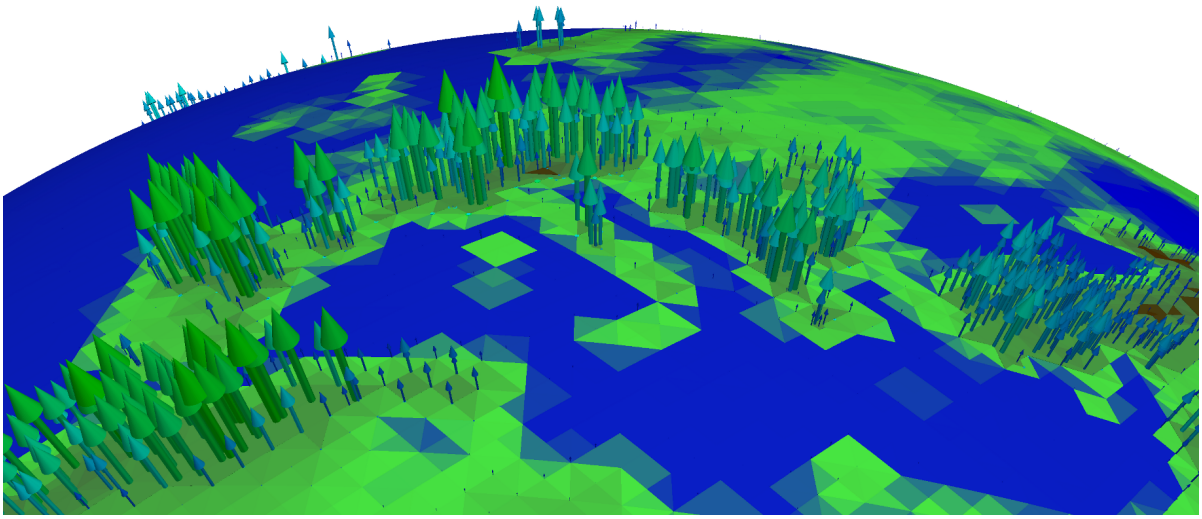
As already introduced in (6.5) in the context of the preceding simulation, we define the absolute error of $\rho$ as mean $l_2$-error

$$e(\rho) := \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} \left(\rho(\boldsymbol{x}_i) - \rho_{\text{ref}}(\boldsymbol{x}_i)\right)^2}. \tag{6.18}$$

$N$ is the number of grid cells and the reference solution $\rho_{\text{ref}}$ is chosen as solution on grid




(a)



(b)

Figure 6.11.: Vertical component $\boldsymbol{u}_\perp$ of the velocity field induced by the topography of (a) the Himalayas and (b) Europe. Depicted is a simulation on a cut cell mesh of level 13 with 16 layers.

level 11 with 32 layers. Analogously, the error of velocity $\boldsymbol{u}$ is defined as

$$e(\boldsymbol{u}) := \sqrt{\frac{1}{3N}\sum_{i=0}^{N-1}\left(\boldsymbol{u}(\boldsymbol{x}_i)-\boldsymbol{u}_{\text{ref}}(\boldsymbol{x}_i)\right)\cdot\left(\boldsymbol{u}(\boldsymbol{x}_i)-\boldsymbol{u}_{\text{ref}}(\boldsymbol{x}_i)\right)}$$
$$= \sqrt{\frac{1}{3N}\sum_{i=0}^{N-1}\sum_{j=0}^{2}\left(u_{ij}-u_{ij}^{\text{ref}}\right)^2} \tag{6.19}$$

with $\boldsymbol{u}(\boldsymbol{x}_i) = (u_{i0}, u_{i1}, u_{i2})^{\mathsf{T}}$ in cell center $\boldsymbol{x}_i$ and reference solution $\boldsymbol{u}_{\text{ref}}$ on grid level 11. The prolongation of $\rho$ and $\boldsymbol{u}$ to the reference mesh needed for the evaluation of these formulae is again chosen as Kriging method, compare Section 6.1.

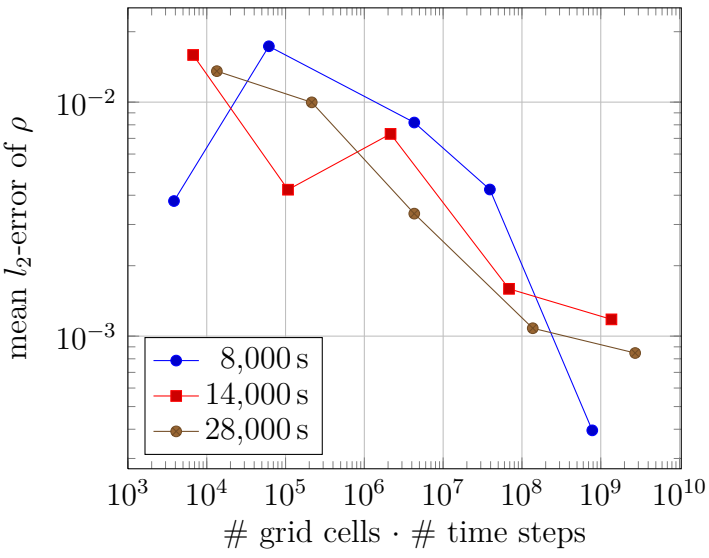
We evaluate the error at three characteristic points in time. 8,000 s is a turning point where the high- and low-pressure areas are balanced but the velocities reach their local maximum. At 14,000 s, the high- and low-pressure areas are opposed to the beginning and at 28,000 s, one cycle is finished, i.e. the range of $\rho$ is maximal but the velocities almost vanish.

Let us now take a look at the convergence plots of Figure 6.12 and 6.13. The first depicts the total complexity of $\rho$ and $\boldsymbol{u}$ where the time step sizes were adjusted to the grids according to Table 5.1, whereas the spatial complexity in the latter was evaluated with $\Delta t = 0.1$ s for each grid. First of all, we notice that the error of $\rho$ decreases faster but alternates more than that of $\boldsymbol{u}$. Furthermore, the translation of the error curves of $\boldsymbol{u}$ reflects the states of the velocity at the different points in time, which we have just described. At 28,000 s, the velocities are vanishingly small and thus the curve is the lowest, whereas the velocities reach their maximum at 8,000 s, which corresponds to the curve with the largest error.

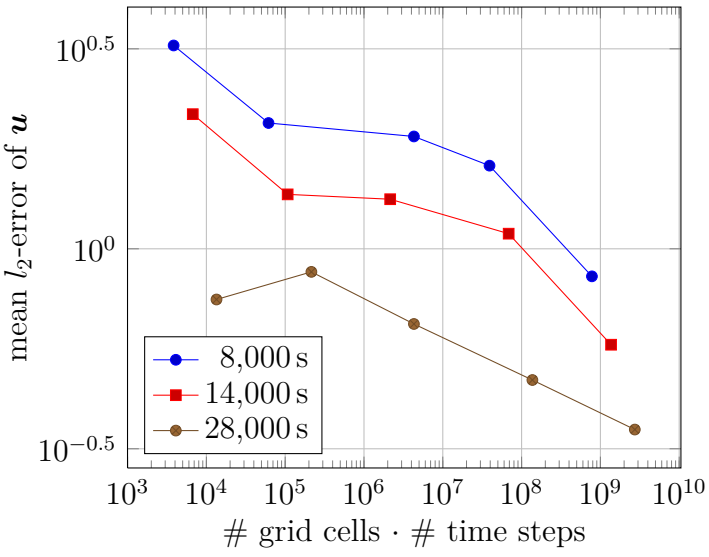
The gradients of these plots defined as slopes between consecutive points are listed in Table 6.1. The convergence rates for total and spatial complexity are significantly below 1.0, but the tendency of increasing descent of some of the curves both for $\rho$ and $\boldsymbol{u}$ suggests that the asymptotic regime is not yet reached. Simulations on finer grids should show a clearly steeper descent.

For the examination of the complexity in time, we choose a fixed grid, simulate the benchmark with a series of different time step sizes, and compare the results with a reference solution computed with an even smaller time step size. The results are shown in Figure 6.14. Here, we see perfect asymptotic behavior with a descent of 1.0 or even slightly more as listed in Table 6.1. This convergence rate confirms that the implicit Euler method we chose as time discretization scheme in Section 5.3 is a first order method.

Summing up, we gained stable simulation results with our cut cell grids constructed in Section 4.7 for the dynamical core consisting of continuity equation, momentum equation, and equation of state. Induced by initial fields of high- and low-pressure areas, we were able to observe the influences of pressure gradient force, convection, rotation of the Earth, Coriolis force, and topography on the fluid flow. Without any further simplifications or

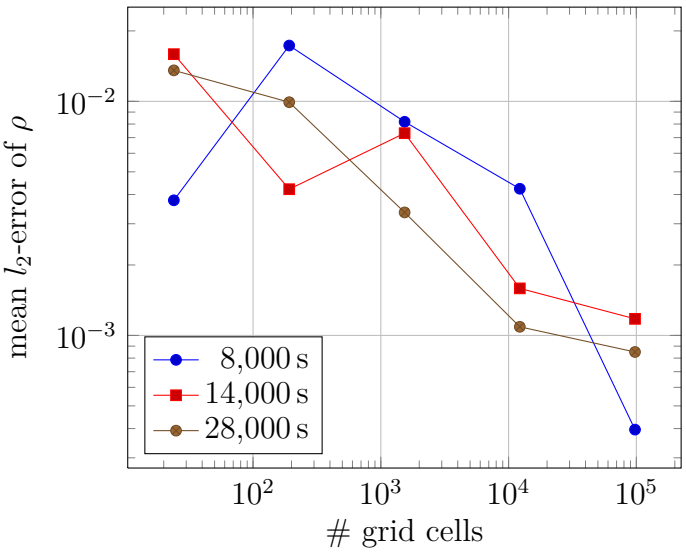



(a) Density $\rho$

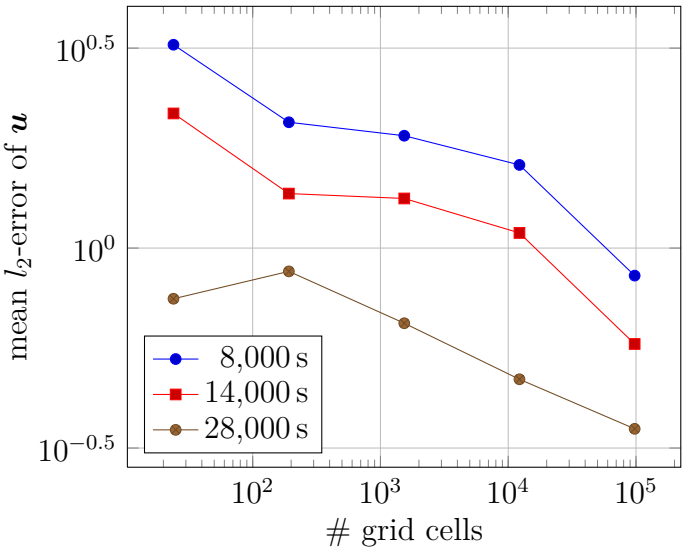



(b) Velocity $\boldsymbol{u}$

Figure 6.12.: Total complexity of (a) $\rho$ and (b) $\boldsymbol{u}$ for the simulation of high- and low-pressure areas in double logarithmic representation. The results on grid level 11 serve as reference solutions. The time step sizes were adjusted to the grids according to Table 5.1.

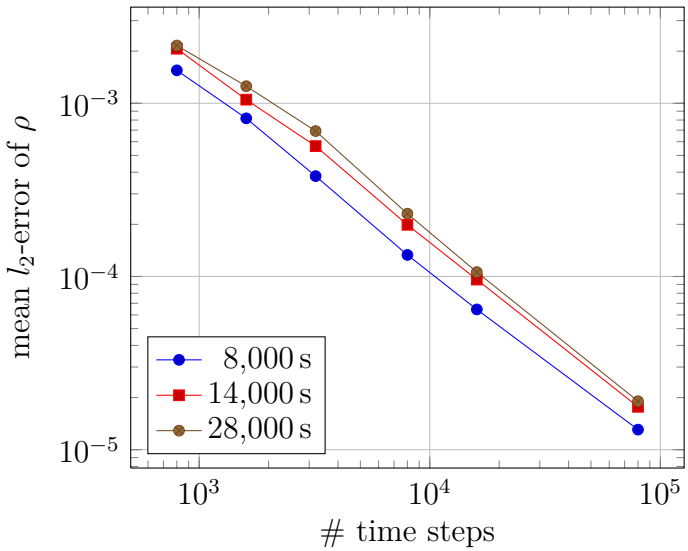



(a) Density $\rho$

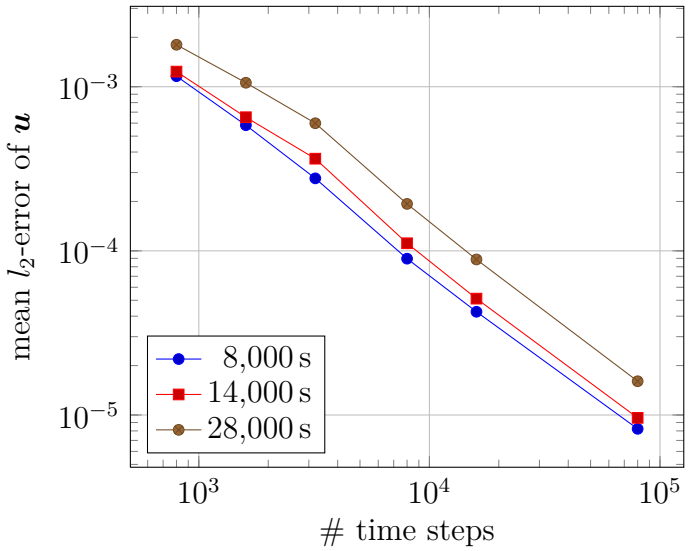



(b) Velocity $\boldsymbol{u}$

Figure 6.13.: Spatial complexity of (a) $\rho$ and (b) $\boldsymbol{u}$ in double logarithmic representation. The time step size $\Delta t = 0.1\,\mathrm{s}$ was fixed for each grid, and the results on grid level 11 serve as reference solutions.




(a) Density $\rho$




(b) Velocity $\boldsymbol{u}$

Figure 6.14.: Temporal complexity of (a) $\rho$ and (b) $\boldsymbol{u}$ in double logarithmic representation. The results on grid level 5 with $\Delta t = 10\,\text{s}$, $5\,\text{s}$, $2.5\,\text{s}$, $1\,\text{s}$, $0.5\,\text{s}$, and $0.1\,\text{s}$ are compared to reference solutions at $\Delta t = 0.01\,\text{s}$.

| Total complexity | | | | | | | |
|---|---|---|---|---|---|---|---|
| Gradient of the error of $\rho$ | | | | Gradient of the error of $\boldsymbol{u}$ | | | |
| grid | 8,000 s | 14,000 s | 28,000 s | grid | 8,000 s | 14,000 s | 28,000 s |
| 01 – 03 | 0.5 | −0.5 | −0.1 | 01 – 03 | −0.2 | −0.2 | 0.1 |
| 03 – 05 | −0.2 | 0.2 | −0.4 | 03 – 05 | −0.02 | −0.01 | −0.1 |
| 05 – 07 | −0.3 | −0.4 | −0.3 | 05 – 07 | −0.1 | −0.1 | −0.1 |
| 07 – 09 | −0.8 | −0.1 | −0.1 | 07 – 09 | −0.2 | −0.2 | −0.1 |

| Spatial complexity | | | | | | | |
|---|---|---|---|---|---|---|---|
| Gradient of the error of $\rho$ | | | | Gradient of the error of $\boldsymbol{u}$ | | | |
| grid | 8,000 s | 14,000 s | 28,000 s | grid | 8,000 s | 14,000 s | 28,000 s |
| 01 – 03 | 0.7 | −0.6 | −0.2 | 01 – 03 | −0.2 | −0.2 | 0.1 |
| 03 – 05 | −0.4 | 0.3 | −0.5 | 03 – 05 | −0.04 | −0.01 | −0.1 |
| 05 – 07 | −0.3 | −0.7 | −0.5 | 05 – 07 | −0.1 | −0.1 | −0.2 |
| 07 – 09 | −1.1 | −0.1 | −0.1 | 07 – 09 | −0.3 | −0.3 | −0.1 |

| Temporal complexity | | | | | | | |
|---|---|---|---|---|---|---|---|
| Gradient of the error of $\rho$ | | | | Gradient of the error of $\boldsymbol{u}$ | | | |
| $\Delta t$ in s | 8,000 s | 14,000 s | 28,000 s | grid | 8,000 s | 14,000 s | 28,000 s |
| 10 – 5 | −0.9 | −1.0 | −0.8 | 10 – 5 | −1.0 | −0.9 | −0.8 |
| 5 – 2.5 | −1.1 | −0.9 | −0.9 | 5 – 2.5 | −1.1 | −0.8 | −0.8 |
| 2.5 – 1 | −1.1 | −1.1 | −1.2 | 2.5 – 1 | −1.2 | −1.3 | −1.2 |
| 1 – 0.5 | −1.0 | −1.1 | −1.1 | 1 – 0.5 | −1.1 | −1.1 | −1.1 |
| 0.5 – 0.1 | −1.0 | −1.1 | −1.1 | 0.5 – 0.1 | −1.0 | −1.0 | −1.1 |

Table 6.1.: Gradient of the convergence plots for total, spatial, and temporal complexity for $\rho$ und $\boldsymbol{u}$ at time steps 8,000 s, 14,000 s, and 28,000 s. The gradient is computed between consecutive points.

modifications, terrain-following grids fail for this benchmark test and develop severe instabilities leading to an abort of the simulation.

## 6.3. Rossby-Haurwitz Wave

As an outlook, we present first results for a further standard benchmark, the Rossby-Haurwitz test case. Rossby-Haurwitz waves are analytical solutions of the non-linear barotropic vorticity equation on the sphere, first examined in [Ros40, Hau40]. These waves are no analytical solutions of the shallow water equations, but they approximately preserve their shape even in shallow water and primitive equation models. Therefore, since [Phi59], they have been frequently used for testing purposes and have thus become a standard benchmark of the shallow-water equations [WDH+92, Hei98]. A three-dimensional version for testing the dynamical core of today's GCMs was proposed in e.g. [GR04] and incorporated in the test suite of [JLNT08].

### 6.3.1. Initial Values

The Rossby-Haurwitz wave is a flow pattern which moves from west to east without change of shape in a non-divergent barotropic model [Hau40]. As initial conditions, we define in the following a Rossby-Haurwitz wave with wave number 4.

Let $\alpha \in [-\pi, \pi)$ be the longitude and $\beta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ the latitude of each atmospheric point $\boldsymbol{x} = (x_1, x_2, x_3)^\mathsf{T}$, which can be evaluated by the formulae

$$\alpha(\boldsymbol{x}) = \begin{cases} \arccos \frac{x_1}{\sqrt{x_1^2+x_2^2}} & \text{if } x_1 \geq 0 \\ -\arccos \frac{x_1}{\sqrt{x_1^2+x_2^2}} & \text{if } x_1 < 0 \end{cases}, \tag{6.20}$$

$$\beta(\boldsymbol{x}) = \frac{\pi}{2} - \arccos \frac{x_3}{\|\boldsymbol{x}\|}. \tag{6.21}$$

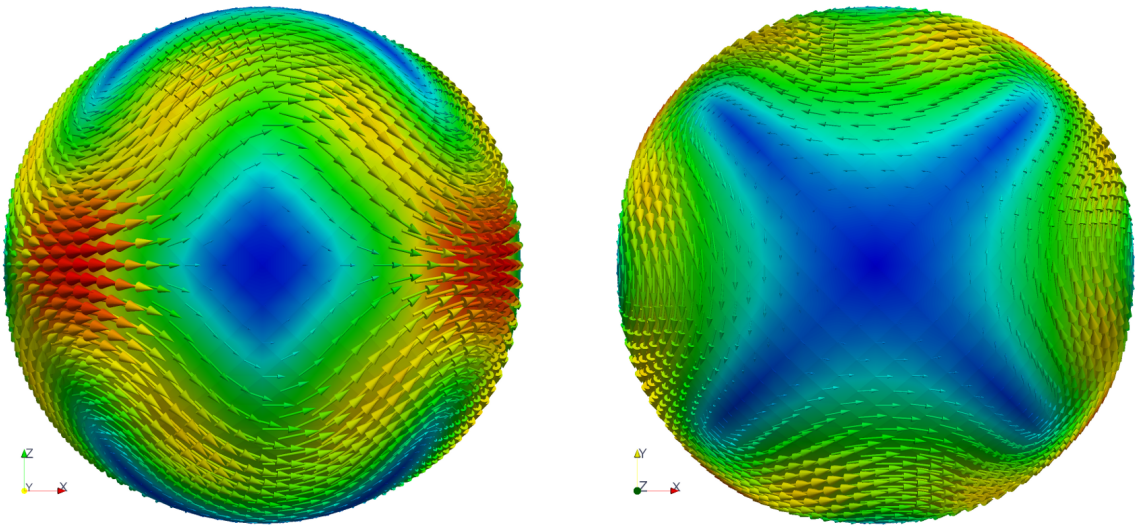
Note the shift in longitude compared to the definition in (6.9).

The velocity field of a Rossby-Haurwitz wave is composed of a zonal velocity $u_{\text{zonal}}$ in eastern direction

$$u_{\text{zonal}} = 50 \left( \cos\beta + \cos^3\beta (5 \sin^2\beta - 1) \cos(4\alpha) \right) \tag{6.22}$$

and a meridional velocity $u_{\text{merid}}$ in southern direction

$$u_{\text{merid}} = -200 \cos^3\beta \sin\beta \sin(4\alpha). \tag{6.23}$$



(a) (b)

Figure 6.15.: Initial velocity field $\boldsymbol{u}^0$ (6.25) of the Rossby-Haurwitz wave in (a) equatorial and (b) polar direction.

Altogether, we obtain the velocity field

$$\boldsymbol{u}(\boldsymbol{x}) = u_{\text{zonal}} \frac{\boldsymbol{\Omega} \times \boldsymbol{x}}{\|\boldsymbol{\Omega} \times \boldsymbol{x}\|} + u_{\text{merid}} \frac{\boldsymbol{x} \times (\boldsymbol{\Omega} \times \boldsymbol{x})}{\|\boldsymbol{x} \times (\boldsymbol{\Omega} \times \boldsymbol{x})\|}, \tag{6.24}$$

which is constant in height. A linear scaling with height, analogously to the previous benchmark, finally leads to the initial velocity field

$$\boldsymbol{u}^0(\boldsymbol{x}) = \frac{\|\boldsymbol{x}\|}{R} \boldsymbol{u}(\boldsymbol{x}) \text{ m/s} \tag{6.25}$$

with Earth radius $R$, see Figure 6.15.

For the initial values of density, pressure, and temperature, we choose the unperturbed fields (6.6), (6.7), and (6.8) of the preceding section.

### 6.3.2. Simulation Results

Here, we present first simulation results of the Rossby-Haurwitz test case as an outlook to further promising applications.

In Figure 6.15, the initial flow pattern of the Rossby-Haurwitz wave is depicted. For shallow water and primitive equations, the Rossby-Haurwitz wave approximately perserves its shape over time, while it moves from west to east around the Earth. For the full three-dimensional Euler equations, this cannot be expected. Nevertheless, we

observe an initial shift of the flow pattern in eastern direction, compare Figure 6.16, which depicts the evolution of the velocity magnitudes over time. Later on, the pattern changes and the flow is redirected.

In the highly resolved pictures of this figure, we can also observe the influence of the topography. Rough terrain leads to a deceleration or an acceleration of the fluid flow, as clearly seen in the color-coded velocity magnitudes of Figure 6.16, which displays a view of the Andes.

Note that for simplicity, we started with a uniform density distribution (6.7). Due to the inital wave pattern, a density profile develops which forms high- and low-pressure areas. Driven by those areas, the flow is redirected and so the wave pattern does not continue its initial eastern shift. This effect is shown in Figure 6.17, where the density distributions in combination with the velocity vectors are illustrated.

For the Rossby-Haurwitz test, terrain-following grids again fail. They develop rapidly amplifying instabilities which lead to an abort of the simulations after a short time, whereas the simulations with our cut cell grids are stable.

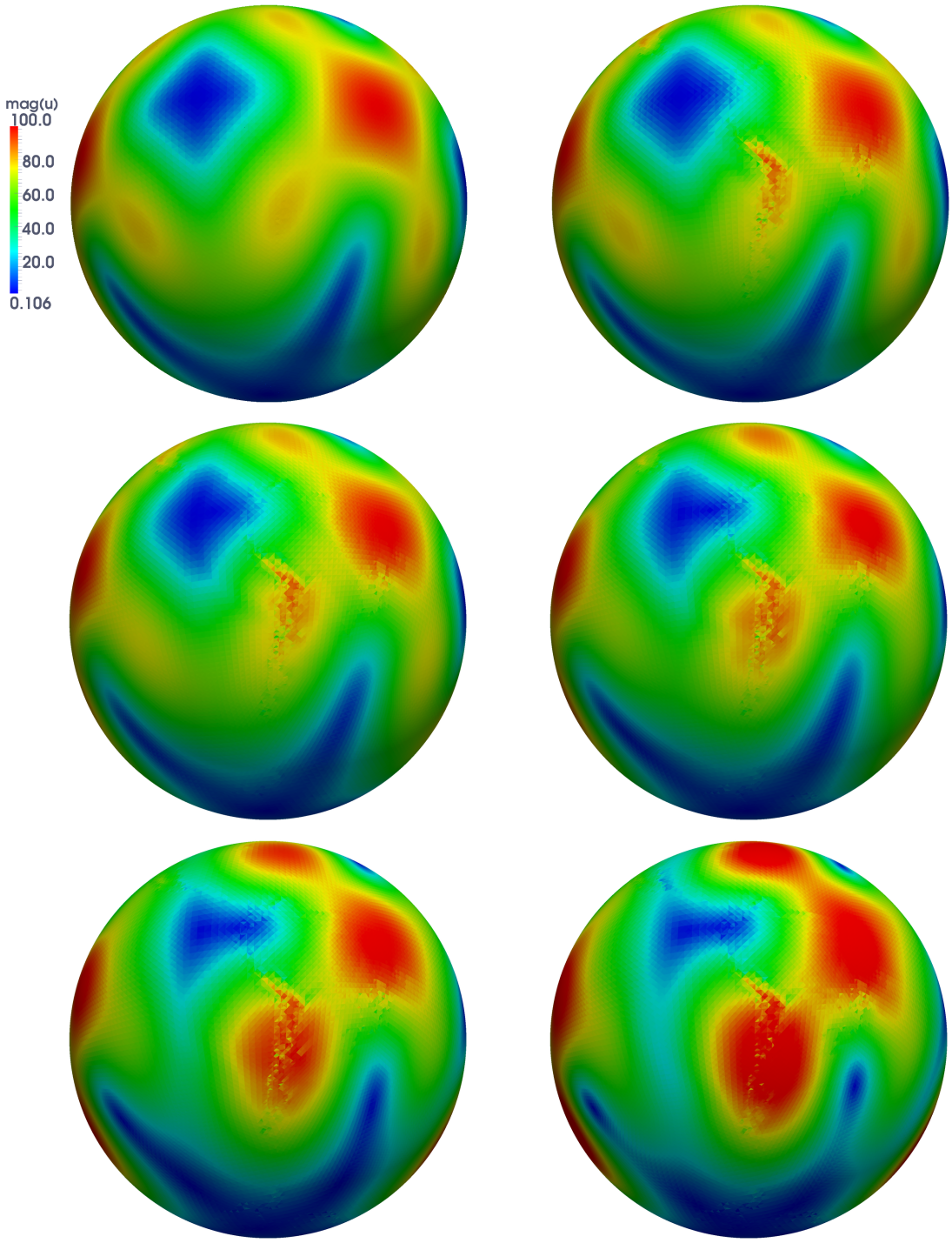



Figure 6.16.: Evolution of the Rossby-Haurwitz flow pattern at ground level with view of the Andes. Depicted is from left to right and top to bottom the velocity magnitude $\|\boldsymbol{u}\|$ at 0 s, 2,000 s, ..., 10,000 s. The simulation was conducted on cut cell grid 13 with 16 layers.

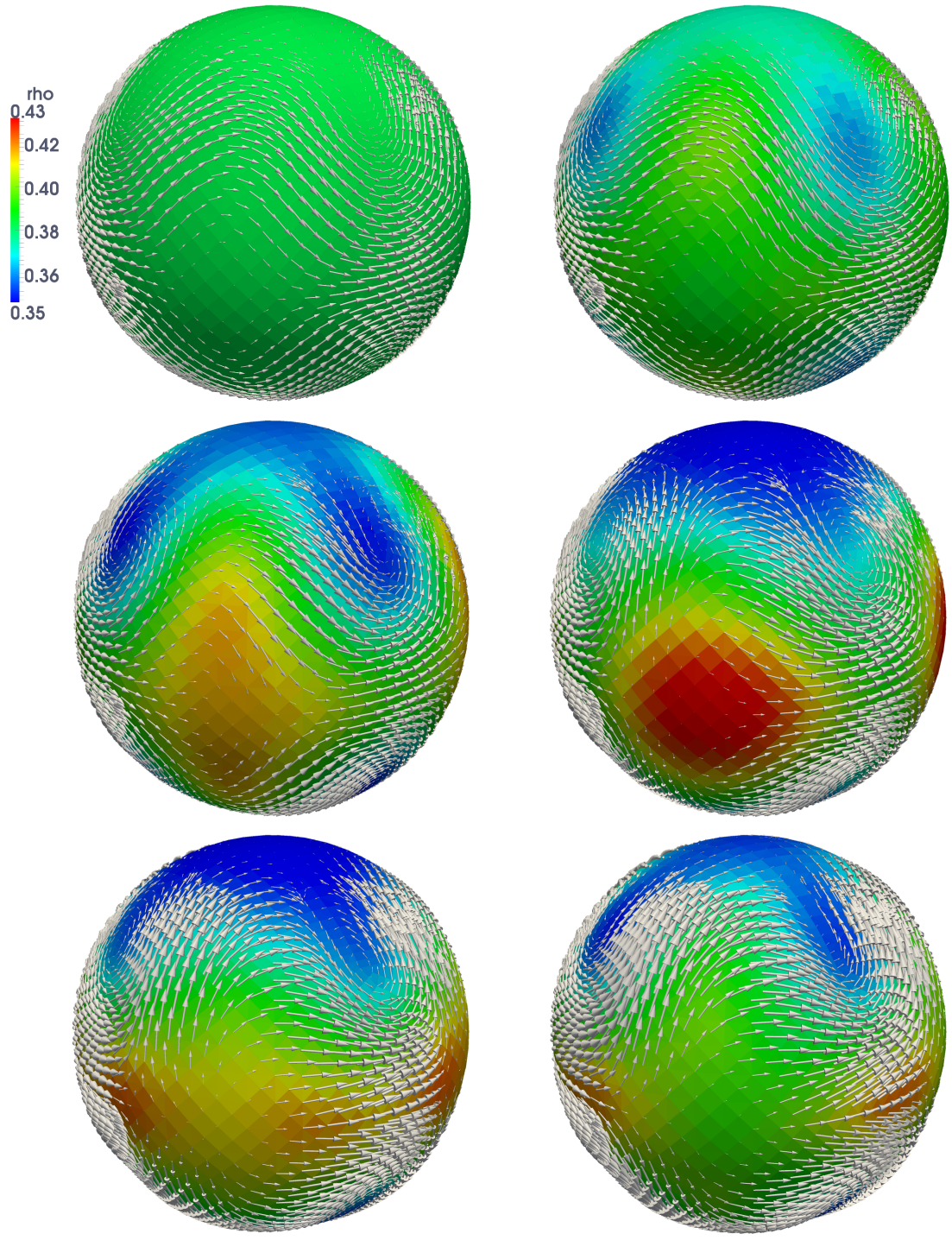



Figure 6.17.: Evolution of the Rossby-Haurwitz flow with cut cell grid 9. Depicted are from left to right and top to bottom the velocity vectors at 0 s, 5,000 s, ..., 25,000 s together with color-coded density distributions.

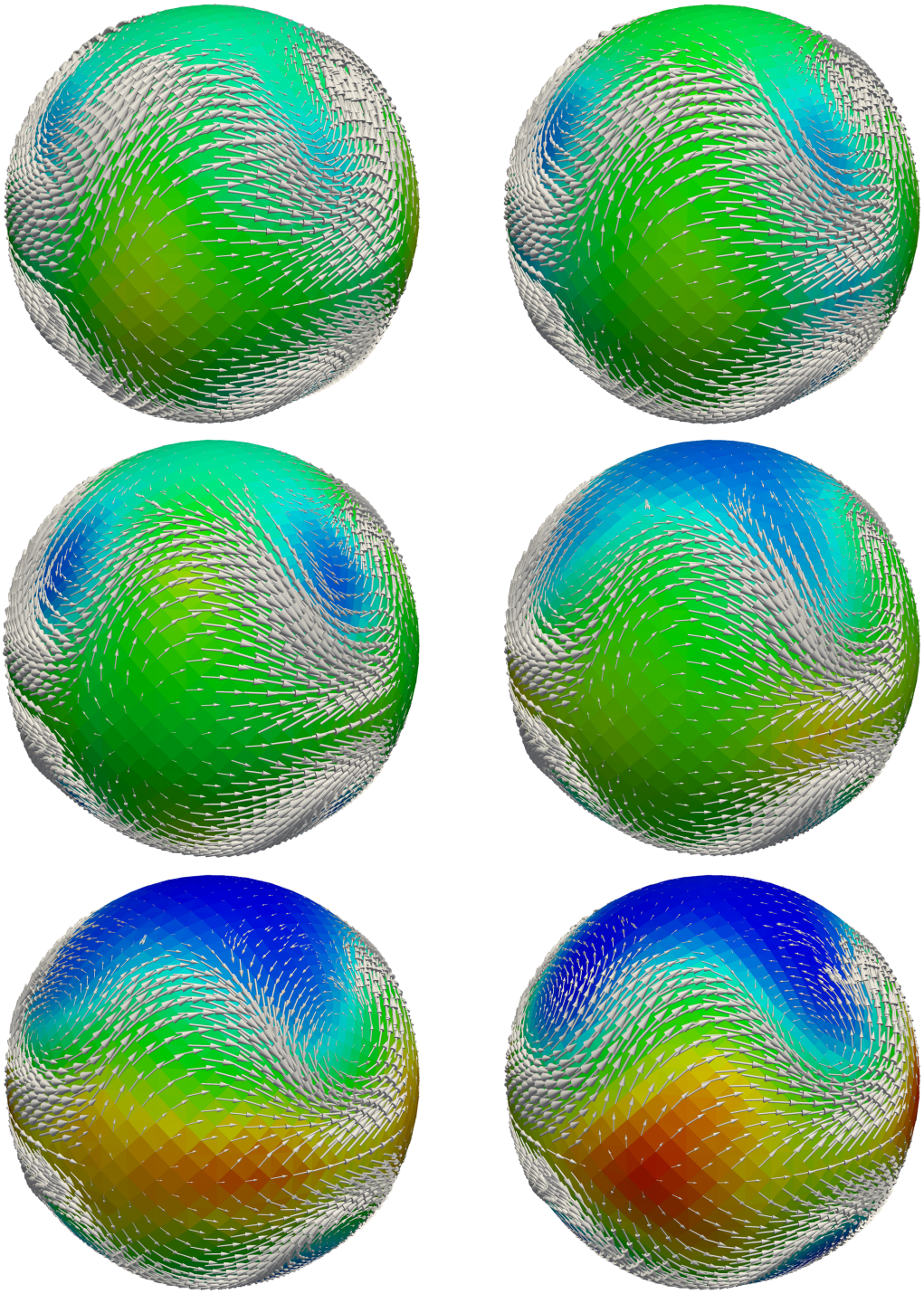


Figure 6.17.: Continuation of the preceding page with velocity vectors and density distributions at 30,000 s, 35,000 s, ..., 55,000 s.

# 7 Conclusion

**Summary**

In this thesis, we studied the modeling, grid generation, and numerical simulation of global atmospheric dynamics. To this end, we initially derived the dynamical core of general circulation models (GCM), i.e. the compressible Navier-Stokes equations, from physical conservation laws and reduced them to the compressible Euler equations based on a detailed dimensional analysis for small- and large-scale motions in the atmosphere. The full three-dimensional Euler equations describe any atmospheric flow on any scale and form the most general model possible.

Nevertheless, we also accounted for reduced meteorological models by giving an insight into multiscale modeling. In this context, we provided a new numerical point of view to the derivation of reduced models by interpreting model assumptions as numerical assumptions, which led to a consistent treatment of the involved errors. Generally, the quality of the solution of the dynamical core depends heavily on these errors. For instance, if the Euler equations are discretized with mesh width $h$, flow on a smaller scale than $h$ is no longer resolved. Therefore, we completed this part of the thesis with the modeling of turbulence and presented two standard methods, namely Reynolds Averaged Navier-Stokes and Large Eddy Simulation, to compensate for small eddies which cannot be resolved by the computational grid.

The resolution of the domain led to the main focus of this thesis, namely the grid generation of the Earth's atmosphere with its vast domain of $10^{19}\,\mathrm{m}^3$. To cover the atmosphere with a manageable grid, we accounted for its anisotropic extensions by splitting the grid generation in its horizontal and vertical part.

Based on the digital elevation models GTOPO30 and ASTER GDEM, we constructed triangulations of the Earth's surface with a classical bisection strategy and the option of including terrain- or region-dependent adaptivity. By projecting a cube's triangulation to the surface of a sphere, we obtained a polygonal approximation of the sphere and finally a global grid of the Earth's topography. In this way, our mesh circumvents singularities of a polar ansatz and the undesirable splitting of global and local grids as currently

applied by Germany's National Meteorological Service (Deutscher Wetterdienst, DWD).

The extension of the horizontal triangulation of the Earth's surface to a three-dimensional mesh which covers the atmosphere was the central question of this thesis. For atmospheric dynamics, step-mountain and terrain-following vertical coordinates are the most common approaches, the latter being widely used in nearly all present weather forecast systems. However, terrain-following coordinates are accompanied by serious drawbacks which we studied in detail, particularly the pressure gradient force error and the hydrostatic inconsistency, both of which increase with finer mesh resolution. We presented the less-known cut cell approach as capable alternative which circumvents the disadvantages of terrain-following coordinates in a natural way.

The cut cell approach is a common technique for mesh generation of complex geometries but found its way into atmospheric dynamics only recently. We described the ansatz in detail, illustrated its superiority to other vertical principles, and accompanied the description by a comprehensive guideline for an implementation of cut cells into existing atmospheric codes.

When dealing with explicit or semi-implicit discretization schemes, the time step size necessary for a stable simulation depends on the smallest cell of the grid, as expressed in the Courant-Friedrichs-Lewy (CFL) criterion. In a cut cell mesh, arbitrarily small cells may arise, restricting the time step size and thus the duration of the solution process to impractical sizes. Especially for up-to-date weather forecasts, the computational speed of the simulation runs is highly relevant. So a treatment of very small cells – the so-called small cell problem – was necessary.

Our approach included two stabilizing conditions imposed on the atmospheric cut cell grids, the omittance of very small cells as well as of cells with highly anisotropic boundary faces. The latter condition was justified by numerical evidence, and we showed that both criteria only marginally influence the overall grid since a considerably smaller percentage of cut cells arise for atmospheric grids than for arbitrarily complex geometries. By limiting small cells, the restriction on the time step size by the CFL criterion could be relaxed.

Next, we dealt with the spatial and temporal discretization of the dynamical core on the discretized computational domain. We chose the Finite Volume method as spatial discretization scheme because of its favorable characteristics concerning conservation properties and handling of unstructured grids. Together with the implicit Euler method as temporal discretization scheme and a new Earth interpolation for the velocity field, we derived a stable discretization of our governing equations. The resulting sparse system of linear equations could be solved by a preconditioned biconjugate gradient solver.

Since very few theoretical results are available for Finite Volume methods and their application to hyperbolic equations, we made an effort to validate our approach by thorough numerical evidence. Thus, our final step was the verification of the numerical scheme by simulation runs so that we could fathom the capabilities of cut cell grids together with our discretization and new Earth interpolation scheme.

We started with a fundamental advection test simulating the transport of a $\cos^2$-

cylinder of density under the influence of a cyclic velocity field. The cylinder was transported around the Earth and arrived at the starting point after twelve days of simulated time. For this test case with fixed velocity field and fully implicit solution procedure, cut cell and terrain-following grids showed similar results and convergence behaviors. Thus, it was a first test to illustrate the capability of cut cell grids and the quality of the new Earth interpolation scheme for the mass flux across cell faces.

Next, we added a dynamic velocity field and simulated counterbalancing flow between high- and low-pressure areas. With this simulation, we were able to observe several influences of our model equations – first of all the influences of the pressure gradient force and the convection, but also the impact of the Coriolis force, which affects the rotating directions of high- and low-pressure areas on northern and southern hemisphere, as well as further impacts of the Earth's rotation, leading to a shift of the initial pressure areas in equatorial and counterclockwise direction. Moreover, the effect of the topography on the flow at ground level and its deflections in horizontal and vertical direction could be studied.

For this benchmark, the stabilizing conditions of our cut cell grids were essential which led to a stable convergence behavior, whereas terrain-following grids completely failed by developing severe instabilities which intensified with finer mesh resolution and which resulted in early aborts of the simulations. The same effect appeared for our final test case, the so-called Rossby-Haurwitz wave. Here, we presented first stable results of our cut cell grids for this standard benchmark. Starting with the Rossby-Haurwitz wave flow pattern, we observed an initial movement of the pattern from west to east, the evolution of high- and low-pressure areas, and the influence of the topography.

In summary, this thesis combined a thorough modeling of the dynamical core of atmospheric dynamics and its special Finite Volume discretization with a detailed analysis of the mesh generation principles for the atmosphere. The long-neglected cut cell approach was applied to the atmospheric domain and compared to the widely-used terrain-following and step-mountain coordinates. Provided with special stabilizing constraints, the superiority of cut cell grids was emphasized and verified by first benchmark tests of the atmospheric setting with increasing complexity. Beyond that, we provided a comprehensive guideline for an implementation of cut cells into existing atmospheric codes, which has not been available so far.

### Outlook

Many interesting points for further studies still remain. In general, the performance of more complex benchmark tests would be the next step. Here, the presently established test suite for three-dimensional atmospheric GCMs by [JLNT08] together with the Dynamical Core Model Intercomparison Project (DCMIP) [UJK+12], a community devoted to the intercomparison of different GCMs, would be the best starting points. With e.g. the baroclinic instability or gravity wave test case, the capabilities of cut cell grids can be further illustrated and compared to results of other GCMs.

In this context, the parallelization of atmospheric codes is essential since it allows for simulation runs with highly resolved grids in reasonable computing time. Parts of our code already work in parallel. However, a fully parallel computation could employ even finer grids and allow for many more degrees of freedom. Therefore, our aim is to parallelize our code completely and prepare the program to be executable on high-performance parallel computers.

Another interesting topic concerns Section 2.3.2, in which we illustrated a numerical point of view at the derivation of reduced discretized atmospheric models. We interpreted model assumptions as numerical assumptions on a full discretized model leading to the commutative diagram displayed in Figure 2.5. The advantage of such "numerical modeling" is the implied consistency that both the discretization error and the modeling error depend on the mesh width $h$. A proof of the commutativity of the diagram is pending, and we propose to study the connection in more detail since its inherent consistency has the potential of simplifying the error analysis considerably.

# A

# Appendix

## A.1. Constants of Atmospheric Motions

We use the following constants according to [Gil82, Pic97]:

- $R = 6.371 \cdot 10^6\,\mathrm{m}$ radius of the Earth, defined as radius of the sphere with the same volume as the Earth,
- $\|\boldsymbol{\Omega}\| = 7.292 \cdot 10^{-5}\,\mathrm{s}^{-1}$ angular velocity of the Earth's rotation around the rotational axis $\boldsymbol{\Omega} = (0, 0, \|\boldsymbol{\Omega}\|)^\mathsf{T}$,
- $g = 9.807\,\mathrm{m\,s}^{-2}$ acceleration of gravity of the Earth,
- $\mu = \mu_{\mathrm{air}} = 1.7 \cdot 10^{-5}\,\mathrm{kg\,m}^{-1}\mathrm{s}^{-1}$ dynamical viscosity of dry air,
- $\lambda = \lambda_{\mathrm{air}} = 0.023\,\mathrm{W\,m}^{-1}\mathrm{K}^{-1}$ thermal conductivity of dry air,
- $R_{\mathrm{air}} = 287.04\,\mathrm{J\,kg}^{-1}\mathrm{K}^{-1}$ gas constant for dry air,
- $c_p = 1005\,\mathrm{J\,kg}^{-1}\mathrm{K}^{-1}$ specific heat capacity of dry air at constant pressure,
- $c_v = 718\,\mathrm{J\,kg}^{-1}\mathrm{K}^{-1}$ specific heat capacity of dry air at constant volume,
- $\gamma = c_p/c_v = 1.4$.

## A.2. OpenFOAM

In this thesis, we used the open source fluid dynamics software package OpenFOAM 2.2.2, the abbreviation standing for "**F**ield **O**peration **A**nd **M**anipulation" [Ope13, Ope12]. OpenFOAM is a computational fluid dynamics toolbox written in C++, which operates on arbitrarily unstructured polyhedral grids and is based on Finite Volume discretization. The various features of OpenFOAM are presented at http://www.openfoam.org.

We used OpenFOAM as basis for our atmospheric dynamics code, benefitting from the provided data structures, Finite Volume methods, and various iterative solvers. We incorporated our atmospheric grids in OpenFOAM and composed the atmospheric model, our discretization techniques, and the new Earth interpolation scheme in the available framework. Finally, we used the solving routines of the toolbox to solve the resulting sparse matrices in each time step. Concretely, the method of choice was an incomplete Cholesky preconditioned biconjugate gradient (BICCG) solver with an absolute residual tolerance of $10^{-16}$.

# Bibliography


[ABCM97] Almgren, A. S., J. B. Bell, P. Colella, and T. Marthaler: *A Cartesian Grid Projection Method for the Incompressible Euler Equations in Complex Geometries*. SIAM Journal on Scientific Computing, 18(5):1289–1309, 1997.

[ABM99] Aftosmis, M. J., M. J. Berger, and J. E. Melton: *Adaptive Cartesian Mesh Generation*. In Thompson, J. F., B. K. Soni, and N. P. Weatherill (editors): *Handbook of Grid Generation*, chapter 22. CRC Press, Boca Raton, Florida, 1999.

[AC04] Adcroft, A. and J.-M. Campin: *Rescaled Height Coordinates for Accurate Representation of Free-Surface Flows in Ocean Circulation Models*. Ocean Modelling, 7:269–284, 2004.

[Adc95] Adcroft, A.: *Numerical Algorithms for Use in a Dynamical Model of the Ocean*. PhD thesis, Imperial College, London, 1995.

[Ade05] Adelsberger, J.: *Interaktive Visualisierung von globalen Höhendaten unter Verwendung verbesserter Bounding Shapes*. Technical report, Institut für Numerische Simulation, Universität Bonn, 2005.

[Ade08] Adelsberger, J.: *Modellierung und Simulation von globalen Atmosphärenströmungen mit Finite-Volumen-Methoden auf adaptiven Polyedergittern*. Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, 2008.

[AHM97] Adcroft, A., C. Hill, and J. Marshall: *Representation of Topography by Shaved Cells in a Height Coordinate Ocean Model*. Monthly Weather Review, 125(9):2293–2315, 1997.

[AST09] ASTER GDEM Validation Team, METI/ERSDAC, NASA/LPDAAC, USGS/EROS: *ASTER Global DEM Validation, Summary Report*, 2009. http://www.ersdac.or.jp/GDEM/E/3.html.

[Bar96] Barenblatt, G. I.: *Scaling, Self-Similarity, and Intermediate Asymptotics*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 1996.

[BD97] Boer, B. J. and B. Denis: *Numerical Convergence of the Dynamics of a GCM*. Climate Dynamics, 13(5):359–374, 1997.

[Bey98] Bey, J.: *Finite-Volumen- und Mehrgitter-Verfahren für elliptische Randwertprobleme*. Advances in Numerical Mathematics. Teubner, Stuttgart, 1998.

[BH93] Beckmann, A. and D. B. Haidvogel: *Numerical Simulation of Flow around a Tall Isolated Seamount. Part I: Problem Formulation and Model Accuracy*. Journal of Physical Oceanography, 23(8):1736–1753, 1993.

[BKG+04] Bonaventura, L., L. Kornblueh, M. Giorgetta, E. Roeckner, T. Heinze, D. Majewski, P. Ripodas, B. Ritter, T. Ringler, and J. Baudisch: *The ICON Shallow Water Model: Scientific Documentation and Benchmark Tests*. Technical report, Max Planck Institute for Meteorology and Deutscher Wetterdienst, 2004. http://icon.enes.org.

[BL89] Berger, M. J. and R. J. LeVeque: *An Adaptive Cartesian Mesh Algorithm for the Euler Equations in Arbitrary Geometries*. AIAA Paper, American Institute of Aeronautics and Astronautics, AIAA-89-1930:1–7, 1989.

[BL90] Berger, M. J. and R. J. LeVeque: *Stable Boundary Conditions for Cartesian Grid Calculations*. Computing Systems in Engineering, 1(2–4):305–311, 1990. ICASE Report No. 90-37.

[BO04] Barth, T. and M. Ohlberger: *Finite Volume Methods: Foundation and Analysis*. In Stein, E., R. de Borst, and T. Hughes (editors): *Encyclopedia of Computational Mechanics*, volume 1, pages 439–473. John Wiley & Sons, Chichester, 2004.

[Bon00] Bonaventura, L.: *A Semi-implicit Semi-Lagrangian Scheme Using the Height Coordinate for a Nonhydrostatic and Fully Elastic Model of Atmospheric Flows*. Journal of Computational Physics, 158(2):186–213, 2000.

[Bon04] Bonaventura, L.: *The ICON Project: Development of a Unified Model Using Triangular Geodesic Grids*. Seminar at ECMWF, 2004. http://icon.enes.org.

[Bra57] Brand, L.: *The Pi Theorem of Dimensional Analysis*. Archive for Rational Mechanics and Analysis, 1:35–45, 1957.

[Bre02] Breuer, M.: *Direkte Numerische Simulation und Large-Eddy Simulation turbulenter Strömungen auf Hochleistungsrechnern.* Shaker Verlag, Aachen, 2002. Habilitationsschrift, Universität Erlangen-Nürnberg.

[Bry69] Bryan, K.: *A Numerical Method for the Study of the Circulation of the World Ocean.* Journal of Computational Physics, 4(3):347–376, 1969.

[BS04] Bitzer, H.-W. and J. Steppeler: *Documentation of the z-Coordinate Dynamical Core of LM.* Technical Report 6, Consortium for Small-Scale Modelling, 2004. http://www.cosmo-model.org.

[Buc14] Buckingham, E.: *On Physically Similar Systems; Illustrations of the Use of Dimensional Equations.* Physical Review, 4:345–376, 1914.

[CC87] Chern, I. L. and P. Colella: *A Conservative Front Tracking Method for Hyperbolic Conservation Laws.* Technical Report UCRL-97200, Lawrence Livermore National Laboratory Report, 1987.

[CFL28] Courant, R., K. Friedrichs, and H. Lewy: *Über die partiellen Differenzengleichungen der mathematischen Physik.* Mathematische Annalen, 100:32–74, 1928.

[CGKM06] Colella, P., D. T. Graves, B. J. Keen, and D. Modiano: *A Cartesian Grid Embedded Boundary Method for Hyperbolic Conservation Laws.* Journal of Computational Physics, 211(1):347–366, 2006.

[CL00] Calhoun, D. and R. J. LeVeque: *A Cartesian Grid Finite-Volume Method for the Advection-Diffusion Equation in Irregular Geometries.* Journal of Computational Physics, 157(1):143–180, 2000.

[Coi94] Coirier, W. J.: *An Adaptively-Refined, Cartesian, Cell-Based Scheme for the Euler and Navier-Stokes Equations.* PhD thesis, Department of Aerospace Engineering, University of Michigan, 1994.

[CSH86] Clarke, D. K., M. D. Salas, and H. A. Hassan: *Euler Calculations for Multielement Airfoils Using Cartesian Grids.* AIAA Journal, American Institute of Aeronautics and Astronautics, 24(3):353–358, 1986.

[CvLP92] Chiang, Y. L., B. van Leer, and K. G. Powell: *Simulation of Unsteady Inviscid Flow on an Adaptively Refined Cartesian Grid.* AIAA Paper, American Institute of Aeronautics and Astronautics, AIAA-92-0443:1–16, 1992.

[Dah03] Dahlström, S.: *Large Eddy Simulation of the Flow Around a High-Lift Airfoil.* PhD thesis, Department of Thermo and Fluid Dynamics, Chalmers University of Technology, Göteborg, Sweden, 2003.

[Der12] Deriaz, E.: *Stability Conditions for the Numerical Solution of Convection-Dominated Problems with Skew-Symmetric Discretizations.* SIAM Journal on Numerical Analysis, 50(3):1058–1085, 2012.

[DFH+11] Doms, G., J. Förstner, E. Heise, H.-J. Herzog, D. Mironov, M. Raschendorfer, T. Reinhardt, B. Ritter, R. Schrodin, J.-P. Schulz, and G. Vogel: *A Description of the Nonhydrostatic Regional COSMO Model, Part II: Physical Parameterization.* Consortium for Small-Scale Modelling, 2011. http://www.cosmo-model.org.

[DSB11] Doms, G., U. Schättler, and M. Baldauf: *A Description of the Nonhydrostatic Regional COSMO-Model, Part I: Dynamics and Numerics.* Consortium for Small-Scale Modelling, 2011. http://www.cosmo-model.org.

[Dut86] Dutton, J. A.: *The Ceaseless Wind, An Introduction to the Theory of Atmospheric Motion.* Dover Publications, New York, 1986.

[EGH00] Eymard, R., T. Gallouët, and R. Herbin: *Finite Volume Methods.* In Ciarlet, P. G. and J. L. Lions (editors): *Handbook of Numerical Analysis*, volume 7, pages 713–1018. Elsevier Science, Amsterdam, 2000.

[EHSZ92] Erlebacher, G., M. Y. Hussaini, C. G. Speziale, and T. A. Zang: *Toward the Large-Eddy Simulation of Compressible Turbulent Flows.* Journal of Fluid Mechanics, 238:155–185, 1992. ICASE Report No. 90-76, 1990.

[Ell03] Elling, V. W.: *A Possible Counterexample to Uniqueness of Entropy Solutions and Godunov Scheme Convergence.* Technical Report SCCM-03-05, Stanford University, California, 2003.

[Ell04] Elling, V. W.: *A Lax-Wendroff Type Theorem for Unstructured Grids.* PhD thesis, Stanford University, California, 2004.

[FB96] Fortunato, A. B. and A. M. Baptista: *Evaluation of Horizontal Gradients in Sigma-Coordinate Shallow Water Models.* Atmosphere-Ocean, 34(3):489–514, 1996.

[Fie00] Field, D. A.: *Qualitative Measures for Initial Meshes.* International Journal for Numerical Methods in Engineering, 47:887–906, 2000.

[FJ98] Forrer, H. and R. Jeltsch: *A Higher-Order Boundary Treatment for Cartesian-Grid Methods.* Journal of Computational Physics, 140(2):259–277, 1998.

[Fok03] Foken, T.: *Angewandte Meteorologie, Mikrometeorologische Methoden.* Springer, Berlin, 2003.

[Gal00] Gallus, W. A.: *The Impact of Step Orography on Flow in the Eta Model: Two Contrasting Examples.* Weather and Forecasting, 15(5):630–637, 2000.

[Gar73] Gary, J. M.: *Estimate of Truncation Error in Transformed Coordinate, Primitive Equation Atmospheric Models.* Journal of the Atmospheric Sciences, 30(2):223–233, 1973.

[GDN98] Griebel, M., T. Dornseifer, and T. Neunhoeffer: *Numerical Simulation in Fluid Dynamics, A Practical Introduction.* SIAM, Philadelphia, 1998.

[Ger03a] Gerstner, T.: *Multiresolution Visualization and Compression of Global Topographic Data.* GeoInformatica, 7(1):7–32, 2003. Shortened Version in Proceedings of Spatial Data Handling, IGU/GISc, 14-27, 2000.

[Ger03b] Gerstner, T.: *Top-Down View-Dependent Terrain Triangulation using the Octagon Metric.* Technical report, Institut für Numerische Simulation, Universität Bonn, 2003.

[Ger10] German Aerospace Center (DLR): *TanDEM-X, The Earth in Three Dimensions.* Deutsches Zentrum für Luft- und Raumfahrt, DLR, 2010. http://www.dlr.de/eo.

[Ger13] German Aerospace Center (DLR): *TanDEM-X Ground Segment, DEM Products Specification Document 3.0.* Deutsches Zentrum für Luft- und Raumfahrt, DLR, 2013. https://tandemx-science.dlr.de.

[Gil82] Gill, A. E.: *Atmosphere-Ocean Dynamics*, volume 30 of *International Geophysics Series.* Academic Press, Orlando, 1982.

[GJ08] Gilbert, J. C. and P. Joly: *Higher Order Time Stepping for Second Order Hyperbolic Problems and Optimal CFL Conditions.* In Glowinski, R. and P. Neittaanmäki (editors): *Partial Differential Equations*, volume 16 of *Computational Methods in Applied Sciences*, pages 67–93. Springer, New York, 2008.

[GKZ11] Giorgetta, M., P. Korn, and G. Zängl: *ICON: Developing a New Generation of Climate and Weather Forecasting Models, Research News.* Technical report, Max Planck Institute for Meteorology and Deutscher Wetterdienst, 2011.

[GMT00] Galmarini, S., F. Michelutti, and P. Thunis: *Estimating the Contribution of Leonard and Cross Terms to the Subfilter Scale from Atmospheric Measurements.* Journal of the Atmospheric Sciences, 57(17):2968–2976, 2000.

[Gör75] Görtler, H.: *Dimensionsanalyse: Theorie der physikalischen Dimensionen mit Anwendungen*. Ingenieurwissenschaftliche Bibliothek. Springer, Berlin, 1975.

[GPMC91] Germano, M., U. Piomelli, P. Moin, and W. H. Cabot: *A Dynamic Subgrid Scale Eddy Viscosity Model*. Physics of Fluids A, 3(7):1760–1765, 1991.

[GR96] Godlewski, E. and P.-A. Raviart: *Numerical Approximation of Hyperbolic Systems of Conservation Laws*, volume 118 of *Applied Mathematical Sciences*. Springer, New York, 1996.

[GR04] Giraldo, F. X. and T. E. Rosmond: *A Scalable Spectral Element Eulerian Atmospheric Model (SEE-AM) for NWP: Dynamical Core Tests*. Monthly Weather Review, 132(1):133–153, 2004.

[GT99] Galmarini, S. and P. Thunis: *On the Validity of Reynolds Assumptions for Running-Mean Filters in the Absence of a Spectral Gap*. Journal of the Atmospheric Sciences, 56(12):1785–1796, 1999.

[Hac89] Hackbusch, W.: *On First and Second Order Box Schemes*. Computing, 41(4):277–296, 1989.

[Han91] Haney, R. L.: *On the Pressure Gradient Force over Steep Topography in Sigma Coordinate Ocean Models*. Journal of Physical Oceanography, 21(4):610–619, 1991.

[Hau40] Haurwitz, B.: *The Motion of Atmospheric Disturbances on the Spherical Earth*. Journal of Marine Research, 3:254–267, 1940.

[HBL05] Helzel, C., M. J. Berger, and R. J. LeVeque: *A High-Resolution Rotated Grid Method for Conservation Laws with Embedded Geometries*. SIAM Journal on Scientific Computing, 26(3):785–809, 2005.

[Hei98] Heinze, T.: *Ein numerisches Verfahren zur Lösung der Flachwassergleichungen auf einer rotierenden Kugel mittels der Lagrange-Galerkin-Methode*. Diplomarbeit, Institut für Angewandte Mathematik, Universität Bonn, 1998.

[Hen05] Hense, A.: *Theoretische Meteorologie 1 & 2*. Rheinische Friedrich-Wilhelms-Universität Bonn, Fachbereich Meteorologie, 2005. Vorlesungsskript.

[Hir93] Hirt, C. W.: *Volume-Fraction Techniques: Powerful Tools for Wind Engineering*. Journal of Wind Engineering and Industrial Aerodynamics, 46–47:327–338, 1993.

[HMS08] Hartmann, D., M. Meinke, and W. Schröder: *An Adaptive Multilevel Multigrid Formulation for Cartesian Hierarchical Grid Methods.* Computers & Fluids, 37(9):1103–1125, 2008.

[HS94] Held, I. M. and M. J. Suarez: *A Proposal for the Intercomparison of the Dynamical Cores of Atmospheric General Circulation Models.* Bulletin of the American Meteorological Society, 75(10):1825–1830, 1994.

[ICM03] Ingram, D. M., D. M. Causon, and C. G. Mingham: *Developments in Cartesian Cut Cell Methods.* Mathematics and Computers in Simulation, 61(3–6):561–572, 2003.

[Jan77] Janjić, Z. I.: *Pressure Gradient Force and Advection Scheme Used for Forecasting with Steep and Small Scale Topography.* Contributions to Atmospheric Physics, 50:186–199, 1977.

[Jas96] Jasak, H.: *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows.* PhD thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London, 1996.

[JLNT08] Jablonowski, C., P. H. Lauritzen, R. D. Nair, and M. A. Taylor: *Idealized Test Cases for the Dynamical Cores of Atmospheric General Circulation Models: A Proposal for the NCAR ASP 2008 Summer Colloquium.* Technical report, National Center for Atmospheric Research, Boulder, Colorado, 2008.

[JW06a] Jablonowski, C. and D. L. Williamson: *A Baroclinic Instability Test Case for Atmospheric Model Dynamical Cores.* Quarterly Journal of the Royal Meteorological Society, 132(621C):2943–2975, 2006.

[JW06b] Jablonowski, C. and D. L. Williamson: *A Baroclinic Wave Test Case for Dynamical Cores of General Circulation Models: Model Intercomparisons.* Technical Report NCAR/TN-469+STR, National Center for Atmospheric Research, Boulder, Colorado, 2006.

[KAK03] Kirkpatrick, M. P., S. W. Armfield, and J. H. Kent: *A Representation of Curved Boundaries for the Solution of the Navier-Stokes Equations on a Staggered Three-Dimensional Cartesian Grid.* Journal of Computational Physics, 184(1):1–36, 2003.

[KBN09] Klein, R., K. R. Bates, and N. Nikiforakis: *Well-Balanced Compressible Cut-Cell Simulation of Atmospheric Flow.* Philosophical Transactions of the Royal Society A, 367(1907):4559–4575, 2009.

[Kle04] Klein, R.: *An Applied Mathematical View of Meteorological Modeling.* In *Applied Mathematics Entering the 21st Century: Invited Talks from the ICIAM 2003 Congress*, volume 116 of *SIAM Proceedings in Applied Mathematics*, 2004.

[Kle08] Klein, R.: *A Unified Approach to Meteorological Modelling Based on Multiple-Scales Asymptotics.* Advances in Geosciences, 15:23–33, 2008.

[Kle10] Klein, R.: *Scale-Dependent Models for Atmospheric Flows.* Annual Review of Fluid Mechanics, 42:249–274, 2010.

[KME02] Kaazempur-Mofrad, M. R. and C. R. Ethier: *An Efficient Characteristic Galerkin Scheme for the Advection Equation in 3-D.* Computer Methods in Applied Mechanics and Engineering, 191(46):5345–5363, 2002.

[Kri51] Krige, D. G.: *A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand.* Journal of the Chemical, Metallurgical and Mining Society of South Africa, 52:119–139, 1951.

[Krö97] Kröner, D.: *Numerical Schemes for Conservation Laws.* Advances in Numerical Mathematics. Wiley/Teubner, New York/Leipzig, 1997.

[Kru70] Kružkov, S. N.: *First Order Quasilinear Equations in Several Independent Variables.* Mathematics of the USSR-Sbornik, 10(2):217–243, 1970.

[Kur68] Kurihara, Y.: *Note on Finite Difference Expressions for the Hydrostatic Relation and Pressure Gradient Force.* Monthly Weather Review, 96(9):654–656, 1968.

[KV03] Klein, R. and S. Vater: *Mathematische Modellierung in der Klimaforschung.* Freie Universität Berlin, Fachbereich Mathematik und Informatik, 2003. Vorlesungsskript.

[KVPR11] Klein, R., S. Vater, E. Paeschke, and D. Ruprecht: *Multiple Scales Methods in Meteorology.* In *Asymptotic Methods in Fluid Mechanics: Survey and Recent Advances*, volume 523 of *CISM Courses and Lectures*, pages 127–196, 2011.

[Lan95] Landenfeld, T.: *Implementation of Topography in Ocean Models: An Analysis of Terrain-Following Coordinate Systems and a Shaved Cell Approach.* Master's thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, 1995.

[LBC+12] Lock, S.-J., H.-W. Bitzer, A. Coals, A. Gadian, and S. Mobbs: *Demonstration of a Cut-Cell Representation of 3D Orography for Studies of Atmospheric Flows over Very Steep Hills.* Monthly Weather Review, 140(2):411–424, 2012.

[LeV88a] LeVeque, R. J.: *Cartesian Grid Methods for Flow in Irregular Regions.* In Morton, K. W. and M. J. Baines (editors): *Numerical Methods for Fluid Dynamics III*, pages 375–382. Clarendon Press, Oxford, 1988.

[LeV88b] LeVeque, R. J.: *High Resolution Finite Volume Methods on Arbitrary Grids via Wave Propagation.* Journal of Computational Physics, 78(1):36–63, 1988.

[LeV97] LeVeque, R. J.: *Wave Propagation Algorithms for Multidimensional Hyperbolic Systems.* Journal of Computational Physics, 131(2):327–353, 1997.

[LeV02] LeVeque, R. J.: *Finite Volume Methods for Hyperbolic Problems.* Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.

[Lil92] Lilly, D. K.: *A Proposed Modification of the Germano Subgrid-Scale Closure Method.* Physics of Fluids A, 4(3):633–635, 1992.

[LP02] Lindstrom, P. and V. Pascucci: *Terrain Simplification Simplified: A General Framework for View-Dependent Out-of-Core Visualization.* IEEE Transactions on Visualization and Computer Graphics, 8(3):239–254, 2002.

[LTHS08] Lovejoy, S., A. F. Tuck, S. J. Hovde, and D. Schertzer: *Do Stable Atmospheric Layers Exist?* Geophysical Research Letters, 35(1):1–4, 2008.

[LW60] Lax, P. and B. Wendroff: *Systems of Conservation Laws.* Communications on Pure and Applied Mathematics, 13(2):217–237, 1960.

[MAB03] Murman, S. M., M. J. Aftosmis, and M. J. Berger: *Implicit Approaches for Moving Boundaries in a 3-D Cartesian Method.* AIAA Paper, American Institute of Aeronautics and Astronautics, AIAA-2003-1119:1–19, 2003.

[MAH+97] Marshall, J., A. Adcroft, C. Hill, L. Perelman, and C. Heisey: *A Finite-Volume, Incompressible Navier Stokes Model for Studies of the Ocean on Parallel Computers.* Journal of Geophysical Research, 102(C3):5753–5766, 1997.

[McC94] McCalpin, J. D.: *A Comparison of Second-Order and Fourth-Order Pressure Gradient Algorithms in a $\sigma$-Co-ordinate Ocean Model.* International Journal for Numerical Methods in Fluids, 18(4):361–383, 1994.

[Mes82] Mesinger, F.: *On the Convergence and Error Problems of the Calculation of the Pressure Gradient Force in Sigma Coordinate Models.* Geophysical and Astrophysical Fluid Dynamics, 19:105–117, 1982.

[Min09] Ministry of Economy, Trade and Industry of Japan (METI), National Aeronautics and Space Administration (NASA): *ASTER GDEM, Global Digital Elevation Model.* Earth Remote Sensing Data Analysis Center (ERSDAC), 2009. http://www.gdem.aster.ersdac.or.jp.

[MJ85] Mesinger, F. and Z. I. Janjić: *Problems and Numerical Methods of the Incorporation of Mountains in Atmospheric Models.* Lectures in Applied Mathematics, 22:81–120, 1985.

[MJN+88] Mesinger, F., Z. I. Janjić, S. Ničković, D. Gavrilov, and D. G. Deaven: *The Step-Mountain Coordinate: Model Description and Performance for Cases of Alpine Lee Cyclogenesis and for a Case of an Appalachian Redevelopment.* Monthly Weather Review, 116(7):1493–1518, 1988.

[MLP+02] Majewski, D., D. Liermann, P. Prohl, B. Ritter, M. Buchhold, T. Hanisch, G. Paul, W. Wergen, and J. Baumgardner: *The Operational Global Icosahedral-Hexagonal Gridpoint Model GME: Description and High-Resolution Tests.* Monthly Weather Review, 130:319–338, 2002.

[MSCL91] Moin, P., K. Squires, W. H. Cabot, and S. Lee: *A Dynamic Subgrid-Scale Model for Compressible Turbulence and Scalar Transport.* Physics of Fluids A, 3(11):2746–2757, 1991.

[Nik09] Nikiforakis, N.: *Mesh Generation and Mesh Adaptation for Large-Scale Earth-System Modelling.* Theme Issue, Philosophical Transactions of the Royal Society A, 367(1907):4473–4481, 2009.

[Noh64] Noh, W. F.: *CEL: A Time-Dependent, Two-Space-Dimensional, Coupled Eulerian-Lagrangian Code.* In Alder, B. J., S. Fernbach, and M. Rotenberg (editors): *Fundamental Methods in Hydrodynamics*, volume 3 of *Methods in Computational Physics*, pages 117–179. Academic Press, New York, London, 1964.

[Oer04] Oertel, H. (editor): *Prandtl's Essentials of Fluid Mechanics*, volume 158 of *Applied Mathematical Sciences.* Springer, New York, 2. edition, 2004.

[OF03] Osher, S. and R. Fedkiw: *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied Mathematical Sciences.* Springer, New York, 2003.

[Ope12] OpenFOAM: *The Open Source CFD Toolbox, Programmer's Guide 2.1.1*, 2012. http://www.openfoam.org.

[Ope13] OpenFOAM: *The Open Source CFD Toolbox, User Guide 2.2.2*, 2013. http://www.openfoam.org.

[OSK09] Oevermann, M., C. Scharfenberg, and R. Klein: *A Sharp Interface Finite Volume Method for Elliptic Equations on Cartesian Grids.* Journal of Computational Physics, 228(14):5184–5206, 2009.

[OW90] Oliver, M. A. and R. Webster: *Kriging: A Method of Interpolation for Geographical Information Systems.* International Journal of Geographical Information Systems, 4(3):313–332, 1990.

[PB79] Purvis, J. W. and J. E. Burkhalter: *Prediction of Critical Mach Number for Store Configurations.* AIAA Journal, American Institute of Aeronautics and Astronautics, 17(11):1170–1177, 1979.

[PBC+95] Pember, R. B., J. B. Bell, P. Colella, W. Y. Crutchfield, and M. L. Welcome: *An Adaptive Cartesian Grid Method for Unsteady Compressible Flow in Irregular Regions.* Journal of Computational Physics, 120(2):278–304, 1995.

[Phi57] Phillips, N. A.: *A Coordinate System Having Some Special Advantages for Numerical Forecasting.* Journal of Meteorology, 14(2):184–185, 1957.

[Phi59] Phillips, N. A.: *Numerical Integration of the Primitive Equations on the Hemisphere.* Monthly Weather Review, 87(9):333–345, 1959.

[Phi73] Phillips, N. A.: *Principles of Large Scale Numerical Weather Prediction.* In Morel, P. (editor): *Dynamic Meteorology*, pages 1–96. D. Reidel Publishing Company, Dordrecht, 1973.

[Pic97] Pichler, H.: *Dynamik der Atmosphäre.* Spektrum Akademischer Verlag, Heidelberg, 3. edition, 1997.

[Pie02] Pielke, R. A.: *Mesoscale Meteorological Modeling*, volume 78 of *International Geophysics Series.* Academic Press, San Diego, 2. edition, 2002.

[PST04] Polvani, L. M., R. K. Scott, and S. J. Thomas: *Numerically Converged Solutions of the Global Primitive Equations for Testing the Dynamical Core of Atmospheric GCMs.* Monthly Weather Review, 132:2539–2552, 2004.

[PTVF07] Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery: *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, 3. edition, 2007.

[Qui94] Quirk, J. J.: *An Alternative to Unstructured Grids for Computing Gas Dynamic Flows Around Arbitrarily Complex Two-Dimensional Bodies.* Computers & Fluids, 23(1):125–142, 1994. ICASE Report No. 92-7, 1992.

[Ros40] Rossby, C.-G.: *Planetary Flow Patterns in the Atmosphere.* Quarterly Journal of the Royal Meteorological Society, 66:68–87, 1940.

[SB81] Simmons, A. J. and D. M. Burridge: *An Energy and Angular-Momentum Conserving Vertical Finite-Difference Scheme and Hybrid Vertical Coordinates.* Monthly Weather Review, 109(4):758–766, 1981.

[SBJ+06] Steppeler, J., H.-W. Bitzer, Z. Janjić, U. Schättler, P. Prohl, U. Gjertsen, L. Torrisi, J. Parfinievicz, E. Avgoustoglou, and U. Damrath: *Prediction of Clouds and Rain Using a z-Coordinate Nonhydrostatic Model.* Monthly Weather Review, 134(12):3625–3643, 2006.

[SBMB02] Steppeler, J., H.-W. Bitzer, M. Minotte, and L. Bonaventura: *Nonhydrostatic Atmospheric Modeling using a z-Coordinate Representation.* Monthly Weather Review, 130(8):2143–2149, 2002.

[SE03] Schneider, P. J. and D. H. Eberly: *Geometric Tools for Computer Graphics.* The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling. Elsevier Science, San Francisco, 2003.

[SEZH88] Speziale, C. G., G. Erlebacher, T. A. Zang, and M. Y. Hussaini: *The Subgrid-Scale Modeling of Compressible Turbulence.* Physics of Fluids, 31(4):940–942, 1988.

[SKD13] Schneider, K., D. Kolomenskiy, and E. Deriaz: *Is the CFL Condition Sufficient? Some Remarks.* In Moura, C. A. de and C. S. Kubrusly (editors): *The Courant-Friedrichs-Lewy (CFL) Condition, 80 Years After Its Discovery*, pages 139–146. Springer, New York, 2013.

[SL04] Sahu, K. K. and A. K. Lahiri: *Finding the Insphere of a Convex Polyhedron: An Analytical Approach.* Philosophical Magazine, 84(12):1185–1196, 2004.

[SLF+02] Schär, C., D. Leuenberger, O. Fuhrer, D. Lüthi, and C. Girard: *A New Terrain-Following Vertical Coordinate Formulation for Atmospheric Prediction Models.* Monthly Weather Review, 130(10):2459–2480, 2002.

[SM77] Semtner, A. J., Jr. and Y. Mintz: *Numerical Simulation of the Gulf Stream and Mid-Ocean Eddies.* Journal of Physical Oceanography, 7(2):208–230, 1977.

[SM03] Shchepetkin, A. F. and J. C. McWilliams: *A Method for Computing Horizontal Pressure-Gradient Force in an Oceanic Model with a Non-Aligned Vertical Coordinate.* Journal of Geophysical Research, 108(C3):1–42, 2003.

[Sma63] Smagorinsky, J.: *General Circulation Experiments with the Primitive Equations, I. The Basic Experiment.* Monthly Weather Review, 91(3):99–164, 1963.

[Stu88] Stull, R. B.: *An Introduction to Boundary Layer Meteorology.* Atmospheric Sciences Library. Kluwer Academic Publishers, Dordrecht, 1988.

[Sun75] Sundqvist, H.: *On Truncation Errors in Sigma-System Models.* Atmosphere, 13(3):81–95, 1975.

[Sun76] Sundqvist, H.: *On Vertical Interpolation and Truncation in Connexion with Use of Sigma System Models*. Atmosphere, 14(1):37–52, 1976.

[Taf95] Taflove, A.: *Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Artech House, Boston, 1995.

[TSW99] Thompson, J. F., B. K. Soni, and N. P. Weatherill (editors): *Handbook of Grid Generation*. CRC Press, Boca Raton, Florida, 1999.

[TTSG01] Tomita, H., M. Tsugawa, M. Satoh, and K. Goto: *Shallow Water Model on a Modified Icosahedral Geodesic Grid by Using Spring Dynamics*. Journal of Computational Physics, 174(2):579–613, 2001.

[UJK+12] Ullrich, P. A., C. Jablonowski, J. Kent, P. H. Lauritzen, R. D. Nair, and M. A. Taylor: *Dynamical Core Model Intercomparison Project (DCMIP), Test Case Document*. Technical report, DCMIP Summer School, 2012.

[UMJS13] Ullrich, P. A., T. Melvin, C. Jablonowski, and A. Staniforth: *A Proposed Baroclinic Wave Test Case for Deep and Shallow-Atmosphere Dynamical Cores*. Quarterly Journal of the Royal Meteorological Society, pages 1–10, 2013.

[U.S96] U.S. Geological Survey: *GTOPO30, Global Digital Elevation Model*. Center for Earth Resources Observation and Science (EROS), 1996. http://eros.usgs.gov/#/Find_Data/Products_and_Data_Available/gtopo30_info.

[U.S97] U.S. Geological Survey: *GTOPO30 Documentation*. Center for Earth Resources Observation and Science (EROS), 1997. http://eros.usgs.gov/#/Find_Data/Products_and_Data_Available/GTOPO30.

[VGK94] Vreman, B., B. Geurts, and H. Kuerten: *On the Formulation of the Dynamic Mixed Subgrid-Scale Model*. Physics of Fluids, 6(12):4057–4059, 1994.

[WA08] Walko, R. L. and R. Avissar: *The Ocean-Land-Atmosphere Model (OLAM). Part II: Formulation and Tests of the Nonhydrostatic Dynamic Core*. Monthly Weather Review, 136(11):4045–4062, 2008.

[WA11] Walko, R. L. and R. Avissar: *A Direct Method for Constructing Refined Regions in Unstructured Conforming Triangular-Hexagonal Computational Grids: Application to OLAM*. Monthly Weather Review, 139(12):3923–3937, 2011.

[WDH+92] Williamson, D. L., J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber: *A Standard Test Set for Numerical Approximations to the Shallow Water Equations in Spherical Geometry*. Journal of Computational Physics, 102:211–224, 1992.

[YMUS99] Ye, T., R. Mittal, H. S. Udaykumar, and W. Shyy: *An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries.* Journal of Computational Physics, 156(2):209–240, 1999.

[YS08] Yamazaki, H. and T. Satomura: *Vertically Combined Shaved Cell Method in a z-Coordinate Nonhydrostatic Atmospheric Model.* Atmospheric Science Letters, 9(4):171–175, 2008.

[YS10] Yamazaki, H. and T. Satomura: *Nonhydrostatic Atmospheric Modeling Using a Combined Cartesian Grid.* Monthly Weather Review, 138(10):3932–3945, 2010.

[ZSK93] Zang, Y., R. L. Street, and J. R. Koseff: *A Dynamic Mixed Subgrid-Scale Model and its Application to Turbulent Recirculating Flows.* Physics of Fluids A, 5(12):3186–3196, 1993.

# Index