Consortium for Small-Scale Modelling

A Description of the

Nonhydrostatic Regional COSMO-Model

Part I

Dynamics and Numerics

G. Doms and M. Baldauf

COSMO 5.05 February 2018

www.cosmo-model.org

DOI: 10.5676/DWD_pub/nwv/cosmo-doc_5.05_I;

Publisher
Deutscher Wetterdienst
Business Area "Research and Development"
Frankfurter Straße 135
63067 Offenbach
www.dwd.de

Editors
Ulrich Schättler, DWD
Ulrich.Schaettler@dwd.de

Contents

1 Overview on the Model System **1**

1.1 General Remarks 1

1.2 Basic Model Design and Features 3

1.3 Single Precision Version 7

1.4 Organization of the Documentation 8

2 The Governing Equations **10**

2.1 Basic Dynamic Equations 10

2.2 Averaging the Basic Equations 16

2.2.1 Dynamic Equations for Turbulent Flow 16

2.2.2 Simplified Thermodynamics 19

2.3 Rotated Spherical Coordinates 22

2.3.1 Definition 22

2.3.2 The Transformed Dynamic Equations 25

2.3.3 Special Transformation Relations 26

2.4 The Model Base State 29

2.5 Terrain-following Coordinates 33

2.5.1 Basic Geometry and Definitions 33

2.5.2 Transformation Relations 37

2.5.3 The Vertical Coordinate 41

2.6 The Set of Model Equations 46

2.6.1 Dynamic Equations in Terrain-Following Coordinates 46

2.6.2 Modifications 48

2.6.3 The Final Set of Equations 50

3 Discretized Form of the Model Equations **54**

3.1 Model Grid Structure 54

3.1.1 Grid Definition and Staggering 54

3.1.2 Horizontal Numerical Operators 57

3.1.3 Grid Stretching 57

3.1.4 Vertical Numerical Operators 61

3.2 Mode Splitting 63

3.3 Discretization and Numerical Integration 65

3.3.1 The Time Integration Scheme 65

3.3.2 Finite Difference Algorithms Related to Fast Modes 67

3.3.3 Finite Difference Algorithms Related to Slow Modes 71

3.3.4 Outline of an Integration Step 81

3.4 Special Transport Schemes 83

3.4.1 Vertical Redistribution 83

3.4.2 Horizontal Advection 84

3.4.3 Vertical Advection 87

3.4.4 Transport of Precipitation 87

4 Initial and Boundary Conditions **89**

4.1 Initial Conditions 89

4.2 Lateral Boundary Conditions 90

4.2.1 Periodic Boundary Condition 90

4.2.2 Relaxation Boundary Condition 91

4.3 Top Boundary Conditions 92

4.4 Bottom Boundary Conditions 93

4.5 Initialization 94

4.6 Interactive Self-Nesting 97

4.7 Spectral Nudging 97

5 Numerical Smoothing **100**

5.1 Time filter 101

5.2 Computational Mixing . . . 101
5.2.1 Fourth order Horizontal Diffusion . . . 102
5.2.2 A Monotonic Diffusion Operator . . . 105
5.2.3 Fourth-order Horizontal Diffusion with Orographic Flux Limiting . . . 108
5.3 Background Mixing in Physical Space . . . 109
5.4 Upper Boundary Damping Layer . . . 110
5.5 Rayleigh Friction . . . 111
5.6 Filtering of Topographical Forcing . . . 112

6 Alternative Time Integration Schemes 115
6.1 A 3-D Semi-Implicit scheme . . . 115
6.1.1 Semi-implicit Time Discretization . . . 116
6.1.2 Implementation Details . . . 120

7 The Runge-Kutta dynamical core 129
7.1 Slow processes in the Runge-Kutta scheme . . . 131
7.1.1 Some implementation details of the Runge-Kutta scheme . . . 133
7.1.2 Advection for the RK dynamical core . . . 134
7.1.3 Coriolis terms . . . 136
7.2 Fast processes in the Runge-Kutta scheme . . . 137
7.2.1 Integration of the 'fast waves' . . . 137
7.2.2 Boundary conditions . . . 146
7.2.3 Stability of the divergence damping in tilted terrain . . . 148
7.3 Tracer transport . . . 149
7.3.1 The semi-Lagrangian scheme . . . 149
7.3.2 Bott-advection and related schemes . . . 150
7.4 Damping mechanisms . . . 153
7.4.1 Relaxation at the lateral boundaries . . . 153
7.4.2 Horizontal Smagorinsky Diffusion . . . 155
7.4.3 Targeted diffusion to avoid cold pools in narrow valleys . . . 156

References 157

Section 1

Overview on the Model System

1.1 General Remarks

The *COSMO-Model* is a nonhydrostatic limited-area atmospheric prediction model. It has been designed for both operational numerical weather prediction (NWP) and various scientific applications on the meso-β and meso-γ scale. The COSMO-Model is based on the primitive thermo-hydrodynamical equations describing compressible flow in a moist atmosphere. The model equations are formulated in rotated geographical coordinates and a generalized terrain following height coordinate. A variety of physical processes are taken into account by parameterization schemes.

Besides the forecast model itself, a number of additional components such as data assimilation, interpolation of boundary conditions from a driving host model, and postprocessing utilities are required to run the model in NWP-mode, climate mode or for case studies. The purpose of the *Description of the Nonhydrostatic Regional COSMO-Model* is to provide a comprehensive documentation of all components of the system and to inform the user about code access and how to install, compile, configure and run the model.

The basic version of the COSMO-Model (formerly known as *Lokal Modell (LM)*) has been developed at the *Deutscher Wetterdienst* (DWD). The COSMO-Model and the triangular mesh global gridpoint model ICON form – together with the corresponding data assimilation schemes – the NWP-system at DWD. The subsequent developments related to the COSMO-Model have been organized within COSMO, the *Consortium for Small-Scale Modeling*. COSMO aims at the improvement, maintenance and operational application of a nonhydrostatic limited-area modeling system, which is now consequently called the COSMO-Model. The meteorological services participating to COSMO at present are listed in Table 1.1.

For more information about COSMO, we refer to the web-site at `www.cosmo-model.org`.

The COSMO-Model is available free of charge for scientific and educational purposes, especially for cooperational projects with COSMO members. However, all users are required to sign an agreement with a COSMO national meteorological service and to respect certain conditions and restrictions on code usage. For questions concerning the request and the agreement, please contact the chairman of the COSMO Steering Committee. In the case of a planned operational or commercial use of the COSMO-Model package, special regulations

Table 1.1: COSMO: Participating Meteorological Services

DWD	Deutscher Wetterdienst, Offenbach, Germany
MeteoSwiss	Meteo-Schweiz, Zürich, Switzerland
ITAF-ReMet	Ufficio Generale Spazio Aero e Meteorologia, Roma, Italy
HNMS	Hellenic National Meteorological Service, Athens, Greece
IMGW	Institute of Meteorology and Water Management, Warsaw, Poland
ARPA-SIMC	Agenzia Regionale per la Protezione Ambientale del-lÂt Emilia-Romagna Servizio Idro Meteo Clima Bologna, Italy
ARPA-Piemonte	Agenzia Regionale per la Protezione Ambientale, Piemonte, Italy
CIRA	Centro Italiano Ricerche Aerospaziali, Italy
ZGeoBW	Zentrum für Geoinformationswesen der Bundeswehr, Euskirchen, Germany
NMA	National Meteorological Administration, Bukarest, Romania
RosHydroMet	Hydrometeorological Centre of Russia, Moscow, Russia
IMS	Israel Meteorological Service, Bet-Dagan, Israel

will apply.

The further development of the modeling system within COSMO is organized in Working Groups which cover the main research and development activities: data assimilation, numerical aspects, upper air physical aspects, soil and surface physics aspects, interpretation and applications, verification and case studies, reference version and implementation and predictability and ensemble methods. In 2005, the COSMO Steering Committee decided to define *Priority Projects* with the goal to focus the scientific activities of the COSMO community on some few key issues and support the permanent improvement of the model. For contacting the Working Group Coordinators or members of the Working Groups or Priority Projects, please refer to the COSMO web-site.

The COSMO meteorological services are not equipped to provide extensive support to external users of the model. If technical problems occur with the installation of the model system or with basic questions how to run the model, questions could be directed via email to `cosmo-support@cosmo-model.org`. If further problems occur, please contact the members of an appropriate Working Group. We try to assist you as well as possible.

The authors of this document recognize that typographical and other errors as well as dis-

crepancies in the code and deficiencies regarding the completeness may be present, and your assistance in correcting them is appreciated. All comments and suggestions for improvement or corrections of the documentation and the model code are welcome and may be directed to the authors.

1.2 Basic Model Design and Features

The nonhydrostatic fully compressible COSMO-Model has been developed to meet high-resolution regional forecast requirements of weather services and to provide a flexible tool for various scientific applications on a broad range of spatial scales. When starting with the development of the COSMO-Model, many NWP-models operated on hydrostatic scales of motion with grid spacings down to about 10 km and thus lacked the spatial resolution required to explicitly capture small-scale severe weather events. The COSMO-Model has been designed for meso-β and meso-γ scales where nonhydrostatic effects begin to play an essential role in the evolution of atmospheric flows.

By employing 1 to 3 km grid spacing for operational forecasts over a large domain, it is expected that deep moist convection and the associated feedback mechanisms to the larger scales of motion can be explicitly resolved. Meso-γ scale NWP-models thus have the principle potential to overcome the shortcomings resulting from the application of parameterized convection in current coarse-grid hydrostatic models. In addition, the impact of topography on the organization of penetrative convection by, e.g. channeling effects, is represented much more realistically in high resolution nonhydrostatic forecast models.

In the beginning, the operational application of the model within COSMO were mainly on the meso-β scale using a grid spacing of 7 km. The key issue was an accurate numerical prediction of near-surface weather conditions, focusing on clouds, fog, frontal precipitation, and orographically and thermally forced local wind systems. Since April 2007, a meso-γ scale version is running operationally at DWD by employing a grid spacing of 2.8 km. Applications with similar resolutions are now run by most COSMO partners. We expect that this will allow for a direct simulation of severe weather events triggered by deep moist convection, such as supercell thunderstorms, intense mesoscale convective complexes, prefrontal squall-line storms and heavy snowfall from wintertime mesocyclones.

The requirements for the data assimilation system for the operational COSMO-Model are mainly determined by the very high resolution of the model and by the task to employ it also for nowcasting purposes in the future. Hence, detailed high-resolution analyses have to be able to be produced frequently and quickly, and this requires a thorough use of asynoptic and high-frequency observations such as aircraft data and remote sensing data. Since both 3-dimensional and 4-dimensional variational methods tend to be less appropriate for this purpose, a scheme based on the observation nudging technique has been chosen for data assimilation from the beginning of the development. But in March 2017 the nudging scheme has been replaced by a new, more modern, ensemble-based method, called KENDA: Km-scale ENsemble Data Assimilation. Note, that KENDA, unlike the nudging scheme, is not available within the source code of the COSMO-Model.

Besides the operational application, the COSMO-Model provides a nonhydrostatic modeling framework for various scientific and technical purposes. Examples are applications of the model to large-eddy simulations, cloud resolving simulations, studies on orographic flow

systems and storm dynamics, development and validation of large-scale parameterization schemes by fine-scale modeling, and tests of computational strategies and numerical techniques. For these types of studies, the model should be applicable to both real data cases and artificial cases using idealized test data. Moreover, the model has been adapted by other communities for applications in climate mode (CCLM) and / or running an online coupled module for aerosols and reactive trace gases (ART).

Such a wide range of applications imposes a number of requirements for the physical, numerical and technical design of the model. The main design requirements are:

(i) use of nonhydrostatic, compressible dynamical equations to avoid restrictions on the spatial scales and the domain size, and application of an efficient numerical method of solution;

(ii) provision of a comprehensive physics package to cover adequately the spatial scales of application, and provision of high-resolution data sets for all external parameters required by the parameterization schemes;

(iii) flexible choice of initial and boundary conditions to accommodate both real data cases and idealized initial states, and use of a mesh-refinement technique to focus on regions of interest and to handle multi-scale phenomena;

(iv) use of a high-resolution analysis method capable of assimilating high-frequency asynoptic data and remote sensing data;

(v) use of pure Fortran constructs to render the code portable among a variety of computer systems, and application of the standard MPI-software for message passing on distributed memory machines to accommodate broad classes of parallel computers.

The development of the COSMO-Model was organized along these basic guidelines. However, not all of the requirements are fully implemented, and development work and further improvement is an ongoing task. The main features and characteristics of the present release are summarized below.

COSMO-ICON Physics

In the last months, several physical packages have been unified with their counterpart in ICON, to reduce the maintenance work for having two different versions of one parameterization. We refer to this developments as the *COSMO-ICON Physics*.

A major technical change to implement this unification was, to use the ICON data structure for the variables in the physics. This structure does not reflect a horizontal field with two dimensions, but collects the grid points in a vector (or a block). This is in contrast to the COSMO-Model, which uses the `(i,j)`-structure for horizontal fields.

This *blocked data structure* is explained in more detail in Appendix A of Part II, the Physical Parameterizations.

Not all options for the parameterizations have been ported to the blocked data structure. More details are given below in the *Physical Parameterizations*.

Dynamics

- **Model Equations** – Nonhydrostatic, full compressible hydro-thermodynamical equations in advection form. Subtraction of a hydrostatic base state at rest.
- **Prognostic Variables** – Horizontal and vertical Cartesian wind components, pressure perturbation, temperature, specific humidity, cloud water content. Optionally: cloud ice content, turbulent kinetic energy, specific water content of rain, snow and graupel.
- **Diagnostic Variables** – Total air density, precipitation fluxes of rain and snow.
- **Coordinate System** – Generalized terrain-following height coordinate with rotated geographical coordinates and user defined grid stretching in the vertical. Options for (i) base-state pressure based height coordinate, (ii) Gal-Chen height coordinate and (iii) exponential height coordinate (SLEVE) according to Schär et al. (2002).

Numerics

- **Grid Structure** – Arakawa C-grid, Lorenz vertical grid staggering.
- **Spatial Discretization** – Second-order finite differences. For the two time-level scheme also 1st and 3rd to 6th order horizontal advection (default: 5th order). Option for explicit higher order vertical advection.
- **Time Integration** – Two time-level 2nd and 3rd order Runge-Kutta split-explicit scheme after Wicker and Skamarock (2002) and a TVD-variant (Total Variation Diminishing) of a 3rd order Runge-Kutta split-explicit scheme. Option for a second-order leapfrog HE-VI (horizontally explicit, vertically implicit) time-split integration scheme, including extensions proposed by Skamarock and Klemp (1992). Option for a three time-level 3-d semi-implicit scheme (Thomas et al. (2000)) based on the leapfrog scheme.
- **Numerical Smoothing** – 4th-order linear horizontal diffusion with option for a monotonic version including an orographic limiter. Rayleigh damping in upper layers. 2-d divergence damping and off-centering in the vertical in split time steps.

Initial and Boundary Conditions

- **Initial Conditions** – Interpolated initial data from various coarse-grid driving models (ICON (and former GME), ECMWF, COSMO-Model) or from the continuous data assimilation stream (see below). Option for user-specified idealized initial fields.
- **Lateral Boundary Conditions** – 1-way nesting by Davies-type lateral boundary formulation. Data from several coarse-grid models can be processed (ICON (and former GME), IFS, COSMO-Model). Option for periodic boundary conditions.
- **Top Boundary Conditions** – Options for rigid lid condition and Rayleigh damping layer.
- **Initialization** – Digital-filter initialization of unbalanced initial states (Lynch et al. (1997)) with options for adiabatic and diabatic initialization.

Physical Parameterizations

- **Subgrid-Scale Turbulence** – Prognostic turbulent kinetic energy closure at level 2.5 including effects from subgrid-scale condensation and from thermal circulations. Option for a diagnostic second order K-closure of hierarchy level 2 for vertical turbulent fluxes (not ported to the blocked data structure). Option for calculation of horizontal turbulent diffusion in terrain following coordinates (3D Turbulence; tested in artificial setups).
- **Surface Layer Parameterization** – A Surface layer scheme (based on turbulent kinetic energy) including a laminar-turbulent roughness layer. Option for a stability-dependent drag-law formulation of momentum, heat and moisture fluxes according to similarity theory This option has not been ported to the blocked data structure. (Louis (1979)).

- **Grid-Scale Clouds and Precipitation** – Cloud water condensation and evaporation by saturation adjustment. Precipitation formation by a bulk microphysics parameterization including water vapour, cloud water, cloud ice, rain and snow with 3D transport for the precipitating phases. Option for a new bulk scheme including graupel. Option for a simpler column equilibrium scheme.
- **Subgrid-Scale Clouds** – Subgrid-scale cloudiness is interpreted by an empirical function depending on relative humidity and height. A corresponding cloud water content is also interpreted. Option for a statistical subgrid-scale cloud diagnostic for turbulence. This option has not been ported to the blocked data structure.
- **Moist Convection** – Tiedtke (1989) mass-flux convection scheme with equilibrium closure based on moisture convergence. Option for the current IFS Tiedtke-Bechtold convection scheme.
- **Shallow Convection** – Reduced Tiedtke scheme for shallow convection only.
- **Radiation** – δ two-stream radiation scheme after Ritter and Geleyn (1992) short and longwave fluxes (employing eight spectral intervals); full cloud-radiation feedback.
- **Soil Model** – Multi-layer version of the former two-layer soil model after Jacobsen and Heise (1982) based on the direct numerical solution of the heat conduction equation. Snow and interception storage are included.
- **Fresh-Water Lake Parameterization** – Two-layer bulk model after Mironov (2008) to predict the vertical temperature structure and mixing conditions in fresh-water lakes of various depths.
- **Sea-Ice Scheme** – Parameterization of thermodynamic processes (without rheology) after Mironov and Ritter (2004). The scheme basically computes the energy balance at the iceâĂŹs surface, using one layer of sea ice.
- **Terrain and Surface Data** – All external parameters of the model are available at various resolutions for a pre-defined region covering Europe. For other regions or grid-spacings, the external parameter file can be generated by a preprocessor program using high-resolution global data sets.

Data Assimilation

- **Former Method** – Continuous four-dimensional data assimilation based on observation nudging (Schraff (1996), Schraff (1997)), with lateral spreading of upper-air observation increments along horizontal surfaces. Explicit balancing by a hydrostatic temperature correction for surface pressure updates, a geostrophic wind correction, and a hydrostatic upper-air pressure correction.
- **Actual Method** – Ensemble data assimilation based on the LETKF (Local Ensemble Transform Kalman Filter) (Schraff et al. (2016))
- **Assimilated Atmospheric Observations** – Radiosonde (wind, temperature, humidity), aircraft (wind, temperature), wind profiler (wind), and surface-level data (SYNOP, SHIP, BUOY: pressure, wind, humidity). Optionally RASS (temperature), radar VAD wind, and ground-based GPS (integrated water vapour) data. Surface-level temperature is used for the soil moisture analysis only.
- **Radar derived rain rates** – Assimilation of near surface rain rates based on latent heat nudging (Stephan et al. (2008)). It locally adjusts the three-dimensional thermodynamical field of the model in such a way that the modelled precipitation rates should resemble the observed ones.
- **Surface and Soil Fields** – Additional two-dimensional intermittent analysis:
  - **Soil Moisture Analysis** – Daily adjustment of soil moisture by a variational method (Hess (2001)) in order to improve 2-m temperature forecasts; use of a Kalman-Filter-like background weighting.

- **Sea Surface Temperature Analysis** – Daily Cressman-type correction, and blending with global analysis. Use of external sea ice cover analysis.
- **Snow Depth Analysis** – 6-hourly analysis by weighted averaging of snow depth observations, and use of snowfall data and predicted snow depth.

Code and Parallelization

- **Code Structure** – Modular code structure using standard Fortran constructs.
- **Parallelization** – The parallelization is done by horizontal domain decomposition using a soft-coded gridline halo (2 lines for Leapfrog, 3 for the Runge-Kutta scheme). The *Message Passing Interface* software (MPI) is used for message passing on distributed memory machines.
- **Compilation of the Code** – For all programs a Makefile is provided for the compilation which is invoked by the Unix *make* command. Two files are belonging to the Makefile: `ObjFiles` is a list of files that have to be compiled and `ObjDependencies` contains all file dependencies. In addition it reads the file `Fopts`, which has to be adapted by the user to specify the compiler, compiler options and necessary libraries to link.
- **Portability** – The model can be easily ported to various platforms; current applications are on conventional scalar machines (UNIX workstations, LINUX and Windows-NT PCs), on vector computers (NEC SX series) and MPP machines (CRAY, IBM, SGI and others).
- **Model Geometry** – 3-d, 2-d and 1-d model configurations. Metrical terms can be adjusted to represent tangential Cartesian geometry with constant or zero Coriolis parameter.

1.3 Single Precision Version

From the beginning of the development, the COSMO-Model had been designed to be able to run in both precisions: single and double precision. Therefore, the real variables are all defined using a `KIND`-parameter, named `wp` (means: working precision) in the module `kind_parameters.f90` (earlier, this `KIND`-parameter was named `ireals`). Other `KIND`-parameters are `sp` (for single precision) and `dp` (for double precision). Before compiling the model, the user has to decide whether `wp` will be set to `sp` or to `dp`. This can be done with the compiler pragma `-DSINGLEPRECISION`. If this pragma is set, single precision will be used, otherwise double precision.

But in the first years of the COSMO-Model, only the double precision version was developed and tested, nobody ever used or tried a single precision run.

But single precision programs run faster on computers, because of less memory traffic, therefore MeteoSwiss tested to run the COSMO-Model also in single precision. Which did not work in the first instance. Some effort had to be put in adapting the model to work for single precision.

The main changes are:

- Epsilons, which are used in comparisons or to make divisions safe, are adapted to work in both precisions. Variables `repsilon` and `rprecision` have been introduced in module `data_constants.f90`.
- New variables `imp_single` and `imp_double` are added to specify an appropriate MPI data type.

- To avoid automatic conversions by the compiler, all (!) real constants (as `2.0`, `0.5`, etc.) are now written with the kind parameter as suffix: `2.0_wp`, `0.5_wp`, etc.). Further developments should follow this rule!
- The pragma `SINGLEPRECISION` is now used to choose single precision for the COSMO-Model during compilation. If it is not set, double precision is used.

It turned out, that the radiation cannot be run in single precision (at least the routines `coe_th`, `inv_th`, `coe_so`, `inv_so`). Therefore it was decided to run the subroutine `fesft` and all routines called below in double precision. The necessary variables are defined with the `KIND`-parameter `dp`.

1.4 Organization of the Documentation

For the documentation of the model we follow closely the *European Standards for Writing and Documenting Exchangeable Fortran 90-Code*. These standards provide a framework for the use of Fortran-90 in European meteorological organizations and weather services and thereby facilitate the exchange of code between these centres. According to these standards, the model documentation is split into two categories: external documentation (outside the code) and internal documentation (inside the code). The model provides extensive documentation within the codes of the subroutines. This is in form of procedure headers, section comments and other comments. The external documentation is split into seven parts, which are listed in Table 1.2.

Table 1.2: COSMO Documentation: A Description of the Nonhydrostatic Regional COSMO-Model

Part I:	Dynamics and Numerics
Part II:	Physical Parameterization
Part III:	Data Assimilation
Part IV:	Special Components and Implementation Details
Part V:	Preprocessing: Initial and Boundary Data for the COSMO-Model
Part VI:	Model Output and Data Formats for I/O
Part VII:	User's Guide

Parts I - III form the scientific documentation, which provides information about the theoretical and numerical formulation of the model, the parameterization of physical processes and the four-dimensional data assimilation. The scientific documentation is independent of (i.e. does not refer to) the code itself. Part IV will describe the particular implementation of the methods and algorithms as presented in Parts I - III, including information on the basic code design and on the strategy for parallelization using the MPI library for message passing on distributed memory machines (not available yet). The generation of initial and boundary conditions from coarse grid driving models is described in Part V. This part is a description of the interpolation procedures and algorithms used (not yet complete) as well as a User's Guide for the interpolation program INT2LM. In Part VI we give a description

of the data formats, which can be used in the COSMO-Model, and describe the output from the model and from data assimilation. Finally, the User's Guide of the COSMO-Model provides information on code access and how to install, compile, configure and run the model. The User's Guide contains also a detailed description of various control parameters in the model input file (in NAMELIST format) which allow for a flexible model set-up for various applications. All parts of the documentation are available at the COSMO web-site (http://www.cosmo-model.org/content/model/documentation/core/default.htm).

Section 2

The Governing Equations

Starting from first principles, the governing thermo-hydrodynamical equations of the LM are derived in Section 3.1. This basic set of equations comprises prognostic Eulerian equations for momentum, heat, total mass, mass of water substance and the equation of state. The impact of turbulent motions on the nonresolvable scales is taken into account by Reynolds averaging (Section 3.2). In Section 3.3 the set of equations is written in spherical coordinates using some metrical simplifications. A thermodynamic model base state is introduced as described in Section 3.4. The equations are then transformed to a nonorthogonal coordinate system using a generalized terrain following height coordinate (Section 3.5). The resulting set of model equations is finally summarized in Section 3.6.

2.1 Basic Dynamic Equations

To arrive at a suitable mathematical description of atmospheric flow, the atmosphere is considered as a multicomponent continuum which is constituted by dry air, water vapour, liquid water and water in solid state forming an ideal mixture. The liquid and solid forms of water may be further subdivided to represent various categories of water substance in the atmosphere as cloud droplets, raindrops, pristine ice crystals, rimed aggregates of crystals, graupel, hail, etc.

The system is subject to the external impact due to gravity and Coriolis forces. Internally, various processes due to heat, mass and momentum transfer as well as phase changes of water may take place. The basic conservation laws for momentum, mass and heat are then represented by the following budget equations:

$$\rho \frac{d\mathbf{v}}{dt} = -\nabla p + \rho \mathbf{g} - 2\boldsymbol{\Omega} \times (\rho \mathbf{v}) - \nabla \cdot \underline{\mathbf{t}} \tag{2.1}$$

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v} \tag{2.2}$$

$$\rho \frac{dq^x}{dt} = -\nabla \cdot \mathbf{J}^x + I^x \tag{2.3}$$

$$\rho \frac{de}{dt} = -p \nabla \cdot \mathbf{v} - \nabla \cdot (\mathbf{J}_e + \mathbf{R}) + \varepsilon. \tag{2.4}$$

The index x represents a specific constituent of the mixture. We use

$x = d$	for dry air,
$x = v$	for water vapour,
$x = l$	for liquid water and
$x = f$	for water in the solid (frozen) state, i.e. ice.

Generally, bold symbols are used to represent vectors and bold underlined symbols indicate dyadic tensors. The scalar and the vector product are indicated by $\cdot$ and $\times$, respectively. In Eqs. (2.1) - (2.4) and the subsequent sections, the following symbols and definitions are used:

t	time
p	pressure
T	temperature
ρ^x	partial density of mixture constituent x
$\rho = \sum_x \rho^x$	total density of the air mixture
$q^x = \rho^x/\rho$	mass fraction (specific content) of constituent x
$v = \rho^{-1}$	specific volume
e	specific internal energy
$h = e + pv$	specific enthalpy
$\mathbf{v}$	barycentric velocity (relative to the rotating earth)
I^x	sources/sinks of constituent x
$\mathbf{J}^x$	diffusion flux of constituent x
$\mathbf{J}_e$	diffusion flux of internal energy (heat flux)
$\mathbf{R}$	flux density of solar and thermal radiation
$\underline{\mathbf{t}}$	stress tensor due to viscosity
$\varepsilon = -\underline{\mathbf{t}} \cdot \cdot \nabla \mathbf{v}$	kinetic energy dissipation due to viscosity
$\mathbf{\Omega}$	constant angular velocity of earth rotation
$\mathbf{g}$	apparent acceleration of gravity
$d/dt = \partial/\partial t + \mathbf{v} \cdot \nabla$	total (Lagrangian) time derivative operator
$\partial/\partial t$	local (Eulerian) time derivative operator
∇	gradient (Nabla) operator

The set of equations (2.1)-(2.4) has been written in advection form using the Lagrangian time derivative for a more compact representation of the basic conservation laws. Because total mass is conserved, the rate of change of any mass specific quantity ψ can be formulated by

$$\rho \frac{d\psi}{dt} = \frac{\partial(\rho\psi)}{\partial t} + \nabla \cdot (\rho \mathbf{v} \psi), \tag{2.5}$$

using the budget operator $\partial(\rho...)/\partial t + \nabla \cdot (\rho \mathbf{v}...)$. With (2.5) the prognostic equations can easily be transformed to flux (or budget) form, if required.

The source-sink terms I^x of the constituents refer to processes whereby water undergoes phase changes, and to processes by which water is generated and lost in chemical reactions with the components of dry air. For mesoscale dynamical applications, chemical changes in water mass can be neglected. We thus set $I^d = 0$ in the budget equation for dry air. Then,

from the conservation of total mass and the definition of the mass fraction q^x, the following relations hold:

$$\begin{aligned} \sum_x q^x &= q^d + q^v + q^l + q^f = 1 \\ \sum_x \mathbf{J}^x &= \mathbf{J}^d + \mathbf{J}^v + \mathbf{J}^l + \mathbf{J}^f = 0 \\ \sum_x I^x &= I^v + I^l + I^f = 0. \end{aligned} \tag{2.6}$$

On condition that dry air and water vapour behave like ideal gases and that liquid water and ice are incompressible substances, the equation of state for a moist atmosphere reads

$$v \equiv \rho^{-1} = (R_d q^d + R_v q^v)\frac{T}{p} + v_l q^l + v_f q^f, \tag{2.7}$$

where R_d and R_v are, respectively, the gas constants for dry air and water vapour; v_l is the partial specific volume of water, v_f is the partial specific volume of ice. For meteorological applications, liquid water and ice contribute very little to the total specific volume because v_l and v_f as well as q^l and q^f are much smaller than 1 ($(v_l q^l + v_f q^f)/v \approx 10^{-6}$). The neglection of the term $v_l q^l + v_f q^f$ in Eq. (2.7) corresponds also to the usual definition of pressure as the sum of the partial pressures of the gaseous constituents of the mixture. Using this approximation, the equation of state is given by

$$\begin{aligned} p &= \rho(R_d q^d + R_v q^v)T \\ &= \rho R_d\{1 + (R_v/R_d - 1)q^v - q^l - q^f\}T \\ &= \rho R_d T_v, \end{aligned} \tag{2.8}$$

where T_v is the generalized virtual temperature

$$T_v = \{1 + (R_v/R_d - 1)q^v - q^l - q^f\}T. \tag{2.9}$$

In case of $q^l = 0$ and $q^f = 0$, (2.9) becomes identical to the traditional definition of virtual temperature.

In the basic set of equations, i.e. (2.1)-(2.4) and the equation of state (2.8), the temperature is a diagnostic variable that has to be determined from the internal energy e or, alternatively, from the enthalpy h if the budget equation for h

$$\rho\frac{dh}{dt} = \frac{dp}{dt} - \nabla\cdot(\mathbf{J}_e + \mathbf{R}) + \varepsilon \tag{2.10}$$

is used instead of Eq. (2.4). For numerical modelling purposes, however, it is more convenient to apply directly a prognostic equation for temperature which we will refer to as the heat equation. The heat equation is obtained from (2.10) by an expansion of the enthalpy $h(T, p, q^x) = \sum_x h_x q^x$ according to

$$\frac{dh}{dt} = \left(\frac{\partial h}{\partial T}\right)_{p,q^x}\frac{dT}{dt} + \left(\frac{\partial h}{\partial p}\right)_{T,q^x}\frac{dp}{dt} + \sum_x\left(\frac{\partial h}{\partial q^x}\right)_{T,p}\frac{dq^x}{dt}. \tag{2.11}$$

The partial specific enthalpies h_x are given by

$$h_x = h_x^0 + c_{px}(T - T_0), \tag{2.12}$$

where $T_0 = 273.15K$, h_x^0 is the specific enthalpy of constituent x at reference temperature T_0 and c_{px} is the specific heat of constituent x at constant pressure. The small variation of h_l and h_f with pressure has been neglected in (2.12). Using (2.12) yields

$$\begin{aligned}\left(\frac{\partial h}{\partial T}\right)_{p,q^x} &= c_p = \sum_x c_{px} q^x \\ \left(\frac{\partial h}{\partial p}\right)_{T,q^x} &= 0 \\ \left(\frac{\partial h}{\partial q^x}\right)_{T,p} &= h_x = h_x^0 + c_{px}(T - T_0),\end{aligned}$$

for the partial derivatives in (2.11) and from the budget equation (2.10) the heat equation results in the form

$$\rho c_p \frac{dT}{dt} = \frac{dp}{dt} + l_V I^l + l_S I^f - \nabla \cdot (\mathbf{J}_s + \mathbf{R}) - \sum_x c_{px} \mathbf{J}^x \cdot \nabla T + \varepsilon. \tag{2.13}$$

$\mathbf{J}_s$ is the reduced (or sensible) heat flux, l_V and l_S are, respectively, the latent heat of vapourization and the latent heat of sublimation. The difference of l_S and l_V is the latent heat of fusion, l_F. These are defined by

$$\begin{aligned}\mathbf{J}_s &= \mathbf{J}_e - \sum_x h_x \mathbf{J}^x \\ l_V &= L_V - (c_{pl} - c_{pv})(T - T_0) \\ l_S &= L_S - (c_{pf} - c_{pv})(T - T_0) \\ l_F &= L_F - (c_{pf} - c_{pl})(T - T_0),\end{aligned} \tag{2.14}$$

where $L_V = -(h_l^0 - h_v^0)$, $L_S = -(h_f^0 - h_v^0)$ and $L_F = L_S - L_V$ are, respectively, the values of l_V, l_S and l_F at the reference temperature T_0.

In comparison with the budget equation for enthalpy, the heat equation (2.13) reveals clearly the impact of phase transitions of water on the temperature. Additionally, the sensible heat flux $\mathbf{J}_s$, i.e. the heat flux reduced by the heat transported by the diffusion fluxes occurs instead of $\mathbf{J}_e$. The term $\sum_x c_{px} \mathbf{J}^x \cdot \nabla T$ takes the impact of heat advection by the diffusion fluxes into account.

In order to calculate the temperature from the heat equation (2.13), the total derivative of pressure has to be determined. In hydrostatic models using pressure or a pressure based function as vertical coordinate, dp/dt is related to the contravariant vertical velocity and can be diagnosed from the continuity equation. In case of nonhydrostatic models using a vertical coordinate based on geometrical height, however, dp/dt has to be calculated from a separate prognostic equation. This pressure tendency equation is obtained by taking the material derivative of the equation of state (2.8)

$$\frac{dp}{dt} = \frac{p}{\rho}\frac{d\rho}{dt} + \rho R_d T \frac{d\alpha}{dt} + \rho R_d (1 + \alpha)\frac{dT}{dt}, \tag{2.15}$$

where α abbreviates the moisture term

$$\alpha = (R_v/R_d - 1)q^v - q^l - q^f \tag{2.16}$$

in the definition of virtual temperature, i.e. $T_v = (1 + \alpha)T$. Inserting the continuity equation (2.2), the budget equations (2.3) for the moisture constituents and the heat equation (2.13)

in Eq. (2.15) yields

$$\left\{1-(1+\alpha)\frac{R_d}{c_p}\right\}\frac{dp}{dt} = -p\nabla\cdot\mathbf{v} + (1+\alpha)\frac{R_d}{c_p}Q_h + Q_m. \tag{2.17}$$

Q_h represents the diabatic heat production per unit volume of air,

$$Q_h = l_V I^l + l_S I^f - \nabla\cdot(\mathbf{J}_s + \mathbf{R}) - \sum_x c_{px}\mathbf{J}^x\cdot\nabla T + \varepsilon, \tag{2.18}$$

and Q_m describes the impact of changes in the concentrations of the humidity constituents on the pressure tendency:

$$Q_m = \rho R_d T\frac{d\alpha}{dt} = -R_v T(I^l + I^f) - R_v T\nabla\cdot\mathbf{J}^v - R_d T\nabla\cdot\mathbf{J}^d. \tag{2.19}$$

To arrive at a more compact form of the pressure tendency equation, we reformulate the term $(1+\alpha)R_d$ in (2.17) as

$$(1+\alpha)R_d = R_d q^d + R_v q^v = c_p - c_v, \tag{2.20}$$

where $c_v = \sum_x c_{vx}q^x$ is the specific heat of moist air at constant volume. For the liquid and solid forms of water there is no difference in the specific heat at constant pressure and constant volume (due to incompressibility of water and ice), i.e. $c_{vl} = c_{pl}$ and $c_{vf} = c_{pf}$. For the specific heat of dry air and vapour at constant volume the relations $c_{vd} = c_{pd} - R_d$ and $c_{vv} = c_{pv} - R_v$ hold (due to their behaviour as ideal gases). With (2.20) the pressure tendency equation can be written as

$$\frac{dp}{dt} = -(c_p/c_v)p\nabla\cdot\mathbf{v} + (c_p/c_v - 1)Q_h + (c_p/c_v)Q_m. \tag{2.21}$$

Thus, when the heat equation in the form (2.13) is used we have to replace the continuity equation by (2.21) to calculate the pressure tendency. Thereby, the total density becomes a diagnostic variable which is obtained from the equation of state. If, on the other hand, we wish to retain the continuity equation in the basic set of equations, an alternate form of the heat equation has to be applied. This form can be derived by direct expansion of the budget equation for internal energy or, equivalently, by inserting (2.21) in (2.13):

$$\begin{aligned}\rho c_v\frac{dT}{dt} &= -p\nabla\cdot\mathbf{v} + (l_V - R_v T)I^l + (l_S - R_v T)I^f - \nabla\cdot(\mathbf{J}_s + \mathbf{R})\\ &\quad - \sum_x c_{px}\mathbf{J}^x\cdot\nabla T - R_v T\nabla\cdot\mathbf{J}^v - R_d T\nabla\cdot\mathbf{J}^d + \varepsilon.\end{aligned} \tag{2.22}$$

Using the heat equation in the form (2.22), the continuity equation can be applied to predict total density. The pressure is then a diagnostic variable to be calculated from the equation of state. Thus two alternative sets of basic equations can be formulated which are summarized below.

- Set I

$$\begin{aligned}\rho\frac{d\mathbf{v}}{dt} &= -\nabla p + \rho\mathbf{g} - 2\boldsymbol{\Omega}\times(\rho\mathbf{v}) - \nabla\cdot\underline{\mathbf{t}}\\ \frac{dp}{dt} &= -(c_p/c_v)p\nabla\cdot\mathbf{v} + (c_p/c_v - 1)Q_h + (c_p/c_v)Q_m\\ \rho c_p\frac{dT}{dt} &= \frac{dp}{dt} + Q_h\\ \rho\frac{dq^x}{dt} &= -\nabla\cdot\mathbf{J}^x + I^x\\ \rho &= p\{R_d(1+\alpha)T\}^{-1}.\end{aligned} \tag{2.23}$$

- Set II

$$\begin{aligned}
\rho\frac{d\mathbf{v}}{dt} &= -\nabla p + \rho\mathbf{g} - 2\boldsymbol{\Omega}\times(\rho\mathbf{v}) - \nabla\cdot\underline{\mathbf{t}} \\
\frac{d\rho}{dt} &= -\rho\nabla\cdot\mathbf{v} \\
\rho c_v\frac{dT}{dt} &= -p\nabla\cdot\mathbf{v} + Q_h + Q_m \\
\rho\frac{dq^x}{dt} &= -\nabla\cdot\mathbf{J}^x + I^x \\
p &= \rho R_d(1+\alpha)T.
\end{aligned} \tag{2.24}$$

Provided that the phase transition rates I^l and I^f, the diffusion fluxes $\mathbf{J}^v$, $\mathbf{J}^l$ and $\mathbf{J}^f$, the sensible heat flux $\mathbf{J}_s$, the radiation flux $\mathbf{R}$ and the viscoseous momentum stress $\underline{\mathbf{t}}$ are known, both Set I and Set II form a closed set of equations to predict the variables of state, i.e. $\mathbf{v}$, T, p, ρ and the mass concentrations q^v, q^l and q^f.

For numerical modelling of nonhydrostatic compressible atmospheric flow the Set II of equations offers the advantage of a direct use of the continuity equation. Thus, by applying suitable numerical algorithms, it is easy to achieve exact conservation of total mass. However, this set has only rarely been used in practice. One - and may be the only - example is the pioneering work of Müller (1974) who based his two-dimensional model of deep moist convection on the Set II type of equations.

By the numerical solution of Set I, on the other hand, exact mass conservation is not guaranteed but depends largely on the accuracy of the numerical algorithms. Nevertheless, almost all nonhydrostatic compressible models are based on Set I, which is frequently rewritten using potential temperature and dimensionless pressure as dependent variables instead of T and p (e.g. Klemp and Wilhelmson (1978). The reason for this preference are special, numerical efficient schemes for the treatment of sound waves which propagate at high speed and require small time steps for stable integration. These schemes can be easily applied in the Set I type of equations but become untractable with Set II.

Besides the lack of exact mass conservation an additional disadvantage of Set I is the occurrence of the diabatic heating rate Q_h and the moisture source term Q_m in the pressure tendency equation. These terms are relevant for a correct description of the thermodynamical feedbacks of diabatic heating and the representation of thermal compression waves but frequently cause numerical problems. Thus, these terms are usually neglected in the pressure equation. Omitting Q_h and Q_m, however, is equivalent to introducing artificial source and sink terms in the continuity equation. The physical error caused by this simplification is probably small in case of applications to cloud or boundary layer modelling studies with very short simulation times. But significant problems may arise in case of applications to short range numerical weather prediction (NWP) with integration times up to 72 hours and in the associated data assimilation cycle.

The present version of LM is also based on the Set I of equations. In the plannings for future developments we consider to use Set II alternatively as basic equations for a later version of the model.

2.2 Averaging the Basic Equations

The set (2.23) of basic equations is defined in terms of differential operators and is thus, in a formal mathematical sense, only valid in the limit when the time interval δt and the spatial increments δx, δy and δz (defining a volume element $\delta V = \delta x \delta y \delta z$) approach zero. For a physically meaningful interpretation, however, the volume elements must not be infinitesimal points, but have to be large enough to contain a sufficiently large number of molecules to apply statistical thermodynamics. In other words, the spatial increments must be much larger than spacing between molecules but much smaller than macroscopic dimensions so that the differential terms do not vary within δt and δV. In the atmosphere, these criteria limit the direct application of Eqs.(2.23) to space scales on the order of about 1 cm and to time scales of about a second.

2.2.1 Dynamic Equations for Turbulent Flow

Mesoscale meteorological circulations have horizontal scales ranging from some 100 m up to 10 or 100 km and vertical scales up to 10 km. Obviously, the explicit simulation of such atmospheric flows with a numerical model using grid spacings on the order of 1 cm will never be possible. Therefore, it is necessary to average the basic equations over specified space and time scales. In the context of numerical simulations, these scales can be identified with the grid spacings and the time step of the numerical model. For a specified mesoscale circulation, the smaller the grid intervals, the better the numerical resolution of the flow.

Formally, averaging has to be done by summing up a large number of realizations of a specific ensemble of flows. In practice, we assume tacitly that the intervals for averaging are large enough to allow for scale separation, i.e. it is assumed that the average or mean value of a variable of the flow varies much more slowly in time and space than do the deviations from the average. The ensemble average can then be replaced by an integration over a specific time interval and space domain.

By performing this integration, any variable ψ of the flow can be decomposed according to

$$\psi = \overline{\psi} + \psi', \tag{2.25}$$

where

$$\overline{\psi} = \frac{1}{\Delta V \Delta t} \int \int \psi dt dV \tag{2.26}$$

represents the average of ψ over the finite time interval Δt and the volume element ΔV formed by the grid spacings Δx, Δy and Δz. By Reynolds convention, the average of the deviations ψ' is zero:

$$\overline{\psi'} = 0. \tag{2.27}$$

Apart from the decomposition (2.25) using the Reynolds average (2.26), we also use the decomposition

$$\psi = \widehat{\psi} + \psi'' \tag{2.28}$$

with

$$\widehat{\psi} = \overline{\rho\psi}/\overline{\rho} \qquad \text{and} \qquad \widehat{\psi''} = 0, \tag{2.29}$$

where $\widehat{\psi}$ is the mass weighted average (Hesselberg average) of ψ and ψ'' is the deviation of ψ from its mass weighted mean value. The mass weighted average is used for velocity and

for mass specific variables of state as enthalpy h, internal energy e and the concentrations q^x (see van Mieghem (1973)) for the choice of a suitable averaging operator in atmospheric turbulence theory).

The mean values $\overline{\psi}$ and $\widehat{\psi}$ are also called the grid scale values of ψ which describe the slowly varying resolvable part of the flow. The deviations ψ' and ψ'' from the corresponding mean value are often called the subgrid scale perturbations which describe the strongly fluctuating, nonresolvable part of the flow. In the budget equations, the feedback of the subgrid scale fluctuations on the resolvable flow is expressed by additional terms in the form of perturbation correlations resulting from averaging the budget equations. These correlation products represent the mean contribution of transports induced by the nonresolvable motions and are of crucial importance in numerical modelling. For any practical application, they have to be parameterized, i.e., they have to be formulated in terms of the grid scale variables.

If the grid spacings defining the domain ΔV for averaging are not too large, i.e. on the order of some 100 m, the fluctuating part of the flow can be identified with purely turbulent motions. Turbulent flows typically show stochastic characteristics and statistical theories can be used to describe them. However, the larger the horizontal grid spacings become, the more additional processes showing organized structures of increasing complexity are included in the subgrid scale fluxes. For grid spacings of the order of 10 km, e.g., shallow and deep moist convection have to be parameterized and for grid intervals typical for global modelling even mesoscale circulations on scales of about 100 km are not resolved.

Applying the averaging operator (2.26) to the budget equations for momentum (2.1), total mass (2.2), mass of the water constituents (2.3) and enthalpy (2.10) yields prognostic equations for the corresponding mean values:

$$\overline{\rho}\frac{\widehat{d}\widehat{\mathbf{v}}}{dt} = -\nabla\overline{p} + \overline{\rho}\mathbf{g} - 2\boldsymbol{\Omega}\times(\overline{\rho}\widehat{\mathbf{v}}) - \nabla\cdot(\underline{\overline{\mathbf{t}}} + \underline{\mathbf{T}}) \tag{2.30}$$

$$\frac{\widehat{d}\overline{\rho}}{dt} = -\overline{\rho}\nabla\cdot\widehat{\mathbf{v}} \tag{2.31}$$

$$\overline{\rho}\frac{\widehat{d}\widehat{q^x}}{dt} = -\nabla\cdot(\overline{\mathbf{J}^x} + \mathbf{F}^x) + \overline{I^x} \tag{2.32}$$

$$\overline{\rho}\frac{\widehat{d}\widehat{h}}{dt} = \frac{\widehat{d}\overline{p}}{dt} + B_h - \nabla\cdot(\overline{\mathbf{J}_e} + \mathbf{F}_h + \overline{\mathbf{R}}) + \overline{\varepsilon}. \tag{2.33}$$

The operator $\widehat{d}/dt$ assigns the Lagrangian time derivative with respect to the mass weighted barycentric velocity $\widehat{\mathbf{v}}$, i.e., $\widehat{d}/dt = \partial/\partial t + \widehat{\mathbf{v}}\cdot\nabla$, which is related to the budget operator according to

$$\overline{\rho}\frac{\widehat{d}\widehat{\psi}}{dt} = \frac{\partial(\overline{\rho}\widehat{\psi})}{\partial t} + \nabla\cdot(\overline{\rho}\widehat{\mathbf{v}}\widehat{\psi}). \tag{2.34}$$

Eq. (2.34) can be used to transform (2.30)-(2.33) from advection to flux form. In addition to the averages of the molecular fluxes and source terms the following correlation products describing subgrid scale transport processes occur in the set of equations:

$\underline{\mathbf{T}} = \overline{\rho\mathbf{v}''\mathbf{v}''}$	turbulent flux of momentum (the Reynolds stress tensor) ;
$\mathbf{F}^x = \overline{\rho\mathbf{v}''q^x}$	turbulent flux of constituent x ($\sum_x \mathbf{F}^x = 0$) ;
$\mathbf{F}_h = \overline{\rho\mathbf{v}''h}$	turbulent flux of enthalpy ;

$B_h = \overline{\mathbf{v}'' \cdot \nabla p}$ source term of enthalpy due to buoyant heat and moisture fluxes.

The term $B_h = \overline{\mathbf{v}'' \cdot \nabla p}$ in Eq. (2.33) can be interpreted as the work of the turbulent velocity fluctuations done against the pressure gradient force. Because the dominating part of ∇p is its hydrostatic component, this term is closely related to the buoyant heat and moisture fluxes. An approximate form of the buoyancy term B_h is given below.

In contrast to the budget equations, the averaging operator is not applied to thermodynamic equations of state. Here we use the diagnostic hypothesis that linear and nonlinear thermodynamic relations between the variables of state hold for their mean values in the same way as on the molecular scale Herbert (1975). Thus, the equation of state for the turbulent scales of motion reads

$$\begin{aligned} \overline{p} &= \overline{\rho} R_d \{1 + (R_v/R_d - 1)\widehat{q^v} - \widehat{q^l} - \widehat{q^f}\}\widehat{T} \\ &= \overline{\rho} R_d \widehat{T_v}. \end{aligned} \tag{2.35}$$

The heat equation is derived from the prognostic equation (2.33) for mean enthalpy using the diagnostic relation

$$\begin{aligned} \widehat{h} &= \sum_x \widehat{h_x}\widehat{q^x} \\ \text{with} \qquad \widehat{h_x} &= h_x^0 + c_{px}(\widehat{T} - T_0) \end{aligned}$$

for the partial specific enthalpies $\widehat{h_x}$ of the constituents x:

$$\overline{\rho}\widehat{c_p}\frac{\widehat{d}\widehat{T}}{dt} = \frac{\widehat{d}\overline{p}}{dt} + \overline{Q_h}. \tag{2.36}$$

The mean diabatic heating $\overline{Q_h}$ is given by

$$\begin{aligned} \overline{Q_h} &= \widehat{l_V}\overline{I^l} + \widehat{l_S}\overline{I^f} + B_h - \nabla \cdot (\overline{\mathbf{J}_s} + \mathbf{H} + \overline{\mathbf{R}}) \\ &\quad - \sum_x c_{px}(\overline{\mathbf{J}^x} + \mathbf{F}^x) \cdot \nabla\widehat{T} + \overline{\varepsilon}. \end{aligned} \tag{2.37}$$

$\widehat{l_V}$ and $\widehat{l_S}$ are, respectively, the latent heat of vapourization and of sublimation as defined by Eq. (2.14) but for the mean value $\widehat{T}$ of temperature. $\widehat{c_p} = \sum_x c_{px}\widehat{q^x}$ is the mean value of specific heat at constant pressure and $\mathbf{H}$ assigns the turbulent flux of sensible heat defined by

$$\mathbf{H} = \mathbf{F}_h - \sum_x \widehat{h_x}\mathbf{F}^x. \tag{2.38}$$

In analogy to the procedure in Section 3.1, the pressure tendency equation is obtained by taking the material derivative of the equation of state (2.35). By using (2.31), (2.32) and the heat equation (2.36) we arrive at

$$\frac{\widehat{d}\overline{p}}{dt} = -(\widehat{c_p}/\widehat{c_v})\overline{p}\nabla \cdot \widehat{\mathbf{v}} + (\widehat{c_p}/\widehat{c_v} - 1)\overline{Q_h} + (\widehat{c_p}/\widehat{c_v})\overline{Q_m}, \tag{2.39}$$

where $\widehat{c_v} = \sum_x c_{vx}\widehat{q^x}$ is the mean value of the specific heat at constant volume and the moisture source term $\overline{Q_m}$ is defined by

$$\overline{Q_m} = -R_v\widehat{T}(\overline{I^l} + \overline{I^f}) - R_v\widehat{T}\nabla\cdot(\overline{\mathbf{J}^v} + \mathbf{F}^v) - R_d\widehat{T}\nabla\cdot(\overline{\mathbf{J}^d} + \mathbf{F}^d). \tag{2.40}$$

The equation of motion (2.30), the budget equations (2.32) for the water constituents, the heat equation (2.36), the pressure tendency equation (2.40) and the equation of state (2.35) form a general set to predict the evolution of the mean flow provided that all subgrid scale processes are parameterized in terms of the grid scale variables $\widehat{\mathbf{v}}$, $\overline{\rho}$, $\overline{p}$,$\widehat{T}$ and $\overline{q^x}$. Before we look at practical simplifications of this set, it is useful to reformulate the sensible heat flux $\mathbf{H}$ and the buoyancy term B_h contributing to the mean diabatic heating $\overline{Q_h}$.

By inserting the enthalpy in the definition of $\mathbf{F}_h$ and neglecting triple correlation products in the evaluation, it is easy to show that the sensible heat flux defined by (2.38) can be expressed as a correlation product of the velocity fluctuations and the temperature:

$$\mathbf{H} = \widehat{c_p}\overline{\rho\mathbf{v}''T} = \widehat{c_p}\overline{\rho\mathbf{v}''T''} \tag{2.41}$$

(since $\overline{\rho\mathbf{v}''\widehat{T}} = 0$). The buoyancy term B_h can be written as

$$B_h = \overline{p\mathbf{v}''\cdot\nabla\ln p} = R_d\overline{(\rho\mathbf{v}''T_v)\cdot\nabla\ln p}$$

and, by neglecting triple correlations, we finally arrive at the representation (with α from (2.16) and using (2.20))

$$\begin{aligned} B_h &= R_d(1+\widehat{\alpha})(\overline{\rho\mathbf{v}''T})\cdot\nabla\ln\overline{p} + R_d\widehat{T}(R_v/R_d-1)(\overline{\rho\mathbf{v}''q^v})\cdot\nabla\ln\overline{p} \\ &\quad -R_d\widehat{T}(\overline{\rho\mathbf{v}''q^l} - \overline{\rho\mathbf{v}''q^f})\cdot\nabla\ln\overline{p}, \\ \text{or} & \\ B_h &= (\widehat{c_p}-\widehat{c_v})/\widehat{c_p}\mathbf{H}\cdot\nabla\ln\overline{p} \\ &\quad +R_d\widehat{T}\{(R_v/R_d-1)\mathbf{F}^v - \mathbf{F}^l - \mathbf{F}^f\}\cdot\nabla\ln\overline{p}. \end{aligned} \tag{2.42}$$

for B_h in terms of the sensible heat flux $\mathbf{H}$ and the turbulent fluxes $\mathbf{F}^x$ of the various water phases.

This set of hydrodynamical equations for turbulent flow including the heat equation (2.36) with (2.42) for the buoyancy term has not yet been applied in full form for numerical modelling of mesoscale processes. For practical reasons, numerical models make often use of simplifications to the basic equations. These assumptions are typically justified with the scales of motion under consideration, as e.g., the hydrostatic approximation for large and mesoscale flow or the anelastic approximation for deep convection which filters the meteorologically unimportant sound waves. Another type of approximations is related to the fact that in the atmosphere air forms a very diluted mixture with respect to the water constituents. This allows to use an approximate form of the heat equation, which is discussed in the next Section.

2.2.2 Simplified Thermodynamics

Because the LM is designed to cover a broad range of spatial scales for various applications, we will not make use of any scale related assumption. Thus, the nonhydrostatic form of the equations allowing for compressibility will be retained. For simplicity, however, an approximate form of the heat equations is used. In detail, the following simplifications are presupposed for LM:

(a) *Treatment of molecular fluxes*
Mesoscale flow in the atmosphere is always characterized by a more or less turbulent state. In general, the turbulent fluxes of momentum, heat and moisture are much larger than the corresponding molecular fluxes. Thus, all molecular fluxes are neglected except for the diffusion fluxes of the liquid and solid forms of water. Due to microphysical growth processes water drops and ice crystals can become large enough to have a significant fall velocity relative to the air. This gravitational sedimentation flux is part of the molecular diffusion flux and must be retained to describe the fallout of precipitation. Thus, in the basic set of equations we set the viscous stress tensor and the molecular fluxes of sensible heat and of water vapour to zero and replace the fluxes of water and ice by the sedimentation fluxes:

$$\begin{aligned} \underline{\mathbf{t}} &= 0, \qquad \mathbf{J}_s = 0, \qquad \mathbf{J}^v = 0, \\ \mathbf{J}^l &\simeq \mathbf{P}^l = \rho q^l \mathbf{v}_T^l, \\ \mathbf{J}^f &\simeq \mathbf{P}^f = \rho q^f \mathbf{v}_T^f. \end{aligned} \tag{2.43}$$

$\mathbf{P}^l$ and $\mathbf{P}^f$ are, respectively, the precipitation fluxes of liquid water and ice. They depend on the mean fall velocities of the corresponding particles, i.e. their terminal velocities $\mathbf{v}_T^l$ and $\mathbf{v}_T^f$.

(b) *Approximations to the heat equation*
In the atmosphere, the water constituents contribute very little to the total mass of any volume of air. Typical maximum values of q^v are a few percent and q^l and q^f in general do not attain values larger than 10^{-3}. Thus, for numerical modelling purposes, it is a more than adequate simplification to approximate the specific heat of moist air by the specific heat of dry air. In effect, the impact of the diffusion fluxes of the water phases on changes in temperature can be neglected and the latent heat of vapourization and of sublimation can be replaced by their constant values at reference temperature T_0. Using these approximations, i.e.,

$$\begin{aligned} \widehat{c_p} &= \sum_x c_{px} \widehat{q^x} \simeq c_{pd}, \\ \widehat{l_V}(\widehat{T}) &\simeq L_V, \qquad \widehat{l_S}(\widehat{T}) \simeq L_S, \\ \mathbf{H} &\simeq c_{pd} \overline{\rho \mathbf{v}'' T}, \\ \sum_x c_{px} (\overline{\mathbf{J}_x} + \mathbf{F}_x) \cdot \nabla \widehat{T} &\simeq 0, \end{aligned} \tag{2.44}$$

we can treat moist air as though it were dry. The dominating impact of phase changes of water on temperature, however, is taken into account with sufficient accuracy.

(c) *Approximations to the pressure tendency equation*
Because the pressure tendency equation (2.39) replaces the continuity equation, any approximation to the heat equation has to be adapted carefully to guarantee an as close as possible conservation of total mass. An adequate approximation of the pressure tendency equation is

$$\frac{\widehat{d\overline{p}}}{dt} = -(c_{pd}/c_{vd}) \overline{p} \nabla \cdot \widehat{\mathbf{v}} + (c_{pd}/c_{vd} - 1) \overline{Q_h}, \tag{2.45}$$

where the impact on pressure due to changes in the concentrations of the water constituents resulting from diffusion fluxes and phase transitions has been neglected. The

approximation (2.45) induces a small artificial source/sink term in the continuity equation. However, it is supposed that this inconsistency has no significant impact for mesoscale numerical modelling.

(d) *Treatment of buoyant heat and moisture fluxes*
The buoyancy term (2.42) in the heat equation may be approximated by

$$B_h = R_d/c_{pd}\mathbf{H}\cdot\nabla\ln\overline{p} + R_d\widehat{T}\{(R_v/R_d-1)\mathbf{F}^v - \mathbf{F}^l - \mathbf{F}^f\}\cdot\nabla\ln\overline{p}. \tag{2.46}$$

Using the definition of potential temperature and of the scaled pressure variable π (the Exner function),

$$\theta = T/\pi, \qquad \pi = (p/p_{00})^{R_d/c_{pd}}, \tag{2.47}$$

where p_{00} is a constant reference pressure (usually set to 1000 hPa), the turbulent flux of sensible heat can alternatively be written in the more familiar form

$$\mathbf{H} = c_{pd}\overline{\rho\mathbf{v}''T} \simeq c_{pd}\overline{\pi}\overline{\rho\mathbf{v}''\theta}. \tag{2.48}$$

Here, the temperature fluctuations have been approximated by $T'' \simeq \overline{\pi}\theta''$ as for the turbulent scales of motion fractional fluctuations in pressure are generally much smaller than fractional fluctuations in temperature.

However, in the present version of LM temperature changes due to buoyant heat and moisture fluxes will be neglected completely, together with the mean dissipation rate due to viscouseous stresses:

$$B_h = 0, \qquad \overline{\varepsilon} = 0. \tag{2.49}$$

This approximation can be justified by recalling that the forcing function $\nabla\ln\overline{p}$ varies only very slowly with height. For any nonneutral stratification with $\mathbf{H} \neq 0$, the temperature changes caused by the vertical divergence of $\mathbf{H}$ will thus be much larger than those caused by the buoyancy term. The buoyancy term occurs also in the budget equation for turbulent kinetic energy (which we will not consider in this Section, see Part II of the documentation for details), but with opposite sign. If in a later version of LM this equation is added to the set of model equations for a prognostic treatment of turbulence, the buoyancy term should be retained in the heat equation to allow for energy conservation.

By introducing the approximations (2.43), (2.44), (2.45) and (2.49) in the general hydrothermodynamic equations derived above, we finally end up with the following set of equations describing the evolution of nonhydrostatic compressible mean flow. Here and in the following sections, the bar and hat symbols indicating mean values will be omitted for convenience.

$$\rho\frac{d\mathbf{v}}{dt} = -\nabla p + \rho\mathbf{g} - 2\boldsymbol{\Omega}\times(\rho\mathbf{v}) - \nabla\cdot(\underline{\mathbf{T}}) \tag{2.50}$$

$$\frac{dp}{dt} = -(c_{pd}/c_{vd})p\nabla\cdot\mathbf{v} + (c_{pd}/c_{vd}-1)Q_h \tag{2.51}$$

$$\rho c_{pd}\frac{dT}{dt} = \frac{dp}{dt} + Q_h \tag{2.52}$$

$$\rho\frac{dq^v}{dt} = -\nabla\cdot\mathbf{F}^v - (I^l + I^f) \tag{2.53}$$

$$\rho\frac{dq^{l,f}}{dt} = -\nabla\cdot(\mathbf{P}^{l,f} + \mathbf{F}^{l,f}) + I^{l,f} \tag{2.54}$$

$$\rho = p\{R_d(1+(R_v/R_d-1)q^v - q^l - q^f)T\}^{-1}. \tag{2.55}$$

Q_h represents the rate of diabatic heating/cooling and is given by

$$Q_h = L_V I^l + L_S I^f - \nabla \cdot (\mathbf{H} + \mathbf{R}). \tag{2.56}$$

LM uses Eqs. (2.50) - (2.55) as basic model equations. They form a complete set to predict the grid scale variables of state, i.e., $\mathbf{v}$, T, p, ρ, q^v, q^l and q^f, provided that all terms describing the impact of subgrid scale processes are known. These are the Reynolds stress tensor $\underline{\mathbf{T}}$, the turbulent flux of sensible heat $\mathbf{H}$, the turbulent fluxes of water vapour $\mathbf{F}^v$, of liquid water $\mathbf{F}^l$ and of ice $\mathbf{F}^f$, the precipitation fluxes of water $\mathbf{P}^l$ and of ice $\mathbf{P}^f$, the rates of phase changes I^l and I^f of water and ice, respectively, and the flux $\mathbf{R}$ of solar and thermal electromagnetic radiation. The calculation of these terms as functions of the grid scale variables, by so-called parameterization schemes, is described in Part II of the LM documentation.

The budget equations (2.54) for the liquid and solid forms of water may be further subdivided to represent nonprecipitating categories of water, as, e.g., cloud water and cloud ice with negligible sedimentation fluxes, and precipitating categories of water, as, e.g., rain, snow and graupel with large sedimentation fluxes and negligible turbulent fluxes. The concrete form of the corresponding budget equations can become quite complex and depends on the type of parameterization scheme which is chosen to represent the hydrological cycle in the atmosphere. For the next sections, we therefore use Eq. (2.54) as an abbreviation representing various types of parameterizations. See Part II for details on the parameterization of cloud microphysical processes.

2.3 Rotated Spherical Coordinates

The equations of fluid motion derived in the previous section are formulated with respect to the rotating earth and are thus valid for any coordinate system rotating with the earth. Obviously, the spherical coordinate system is the most natural and convenient way to take the spherical shape of the earth into account. With respect to practical applications on a large domain, however, spherical coordinates cause numerical problems due to the convergence of the meridians and the resulting pole singularities.

2.3.1 Definition

For limited area modelling different types of map projections, as the Polar Stereographic, the Lambert or the Mercator projection are frequently used. The use of rotated spherical coordinates has also become common practice. In the latter system, the pole is tilted and can be positioned such that the equator runs through the centre of the model domain. Thus, problems resulting from the convergence of the meridians can be minimized for any limited area model domain on the globe. Especially, for a very small domain with negligible impact of the curvature of the earth's surface, the equations become identical to those for a tangential Cartesian coordinate system.

LM uses a rotated spherical coordinate system. In order to obtain the appropriate equations of motion, we have to make two coordinate transformations. The first one transforms from a Cartesian system (X, Y, Z) with the origin located at the earth's centre and the Z-axis oriented along the axis of the earth's rotation, i.e. pointing toward the geographical North Pole, to a new Cartesian system $(\tilde{X}, \tilde{Y}, \tilde{Z})$. The origin of this new system is also located

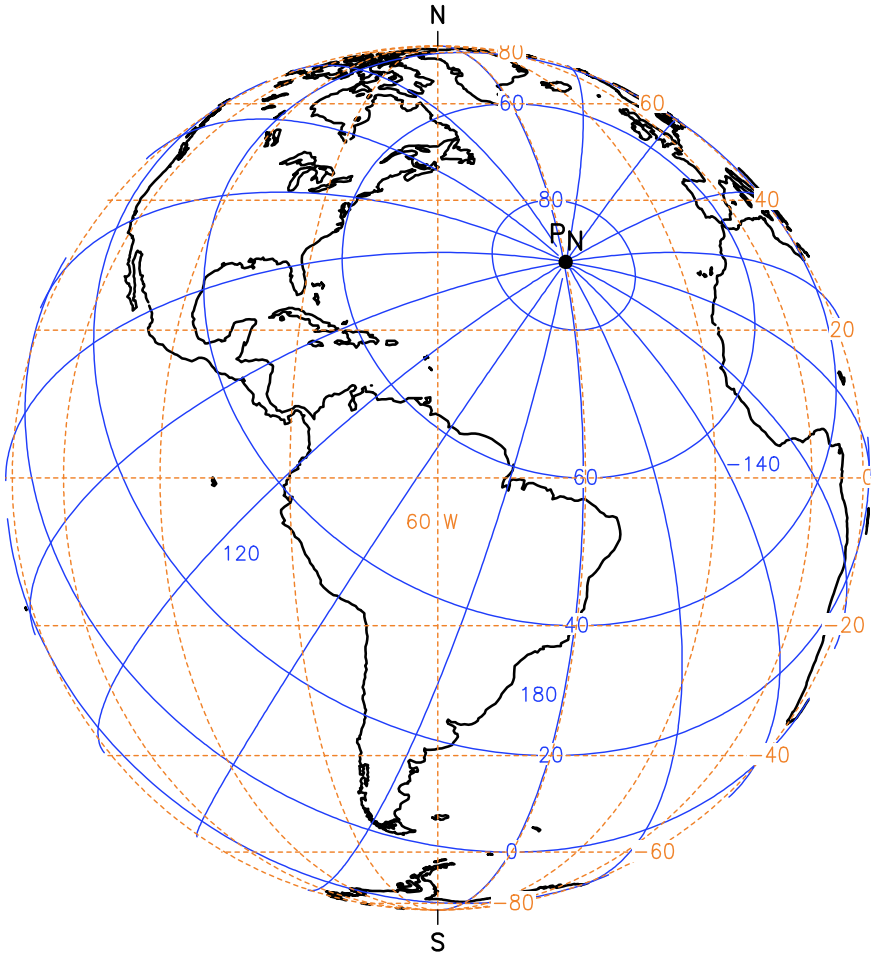

Figure 2.1: Rotated longitude and latitude (full lines, at a contour interval of 20°) for a spherical coordinate system with the North pole shifted to the point P_N with geographical coordinates $\lambda_g^N = 40°W$ and $\varphi_g^N = 30°N$. Broken lines indicate longitude and latitude of the geographical system (at a contour interval of 20°). The rotated 0°/180° meridians are conform with the $140°W/40°E$ geographical meridians. The equator of the rotated grid runs along the southern edge of South America and enters the northern hemisphere in the Pacific. All subdomains centred along the rotated equator will have a minimum of curvature effects from the convergence of meridians.

at the earth's centre, but the $\tilde{Z}$-axis is tilted against the Z-axis. By defining the $\tilde{Z}$-axis to point from the centre to a point $P_N = (\lambda_g^N, \varphi_g^N)$ in which λ_g^N is geographical longitude and φ_g^N is geographical latitude of the point, the transformation is uniquely specified. P_N defines the north pole of the rotated coordinate system. In the second step we transform from the $(\tilde{X}, \tilde{Y}, \tilde{Z})$-system to orthogonal spherical coordinates (λ, φ, r) in which λ is longitude, φ is latitude and r is the distance from the earth's centre. Both λ and φ are defined with respect to the rotated $\tilde{Z}$-axis. It is convenient to define the rotated meridian which runs through both the geographical and the rotated North Pole as the 0° meridian. An example is shown in Figure 2.1.

Because the first transformation involves only a simple rotation of a Cartesian system, the metrics of the rotated (λ, φ, r)-coordinate system is completely determined by the second transformation. Both steps can be done with standard mathematical transformation tech-

niques. The resulting equations are formally identical to those obtained for the nonrotated spherical coordinate system but the geographical longitude λ_g and geographical latitude φ_g are replaced by λ and φ, respectively. Details on the transformation can be found in standard textbooks on dynamic meteorology, e.g. Dutton (1976).

Most meteorological models do not apply the full form of the transformed equations of motion but make use of some metrical simplifications, which we will also apply for LM. First, all impacts resulting from the nonspherical shape of the earth's surface are neglected and the apparent gravity acceleration is assumed to be constant and perpendicular to surfaces of constant radius, i.e.

$$\mathbf{g} \cong -g(\mathbf{r}/r), \tag{2.57}$$

where g is the constant mean value of absolute gravity acceleration. Second, as the vertical extent of the model is confined to the troposphere and lower stratosphere, the height of any point above the surface will be much smaller than the radius of the earth, and we have to a close approximation

$$r = a + z \cong a. \tag{2.58}$$

a is the mean radius of the earth's surface and z is the geometrical height above mean sea level. Thus, wherever the distance r occurs as a factor in the transformed equations it can be approximated by a. By replacing the differential variation of r by the differential variation in z, $\partial r = \partial z$, the equations can then be written in terms of z as independent vertical coordinate instead of r. The approximation (2.58) supposes that all spherical surfaces of constant vertical coordinate z have the same curvature. This has two important consequences: (a) a number of metrical accelerations appearing in the equations of motion have to be neglected, and (b) the Coriolis acceleration has to be modified for consistency reasons (the Coriolis effect due to as well as on vertical motion must be neglected), taking a much simpler form.

By applying the metrical simplification, the orthogonal base vectors $\mathbf{q}_i$ and the Jacobian of the transformation, $\sqrt{G^s}$, of the modified (λ, φ, z) coordinate system are given by

$$\begin{aligned} \mathbf{q}_1 &= a\cos\varphi \mathbf{e}_\lambda \\ \mathbf{q}_2 &= a\mathbf{e}_\varphi \\ \mathbf{q}_3 &= \mathbf{e}_z = \mathbf{r}/r \\ \sqrt{G^s} &= a^2\cos\varphi. \end{aligned} \tag{2.59}$$

$\mathbf{e}_\lambda$, $\mathbf{e}_\varphi$ and $\mathbf{e}_z$ are the normalized unit vectors in the corresponding directions λ, φ and z, respectively. $\sqrt{G^s}$ is related to the metric tensor $\mathcal{G}^s$ of the spherical coordinate system according to

$$\sqrt{G^s} = \sqrt{\det(\mathcal{G}^s)}$$

where the elements g^s_{ij} of $\mathcal{G}^s$ are given by the scalar product of the base vectors $\mathbf{q}_i$, i.e. $g^s_{ij} = \mathbf{q}_i \cdot \mathbf{q}_j$. Thus, $\mathcal{G}^s$ is to be interpreted as a 3×3 matrix, with an obvious role in the measurement of distances, and not as a physical tensor. Because the $\mathbf{q}_i$ form an orthogonal vector base, all off-diagonal terms of the metric tensor $\mathcal{G}^s$ are zero, the diagonal elements are

$$g^s_{11} = a^2\cos^2\varphi, \qquad g^s_{22} = a^2, \qquad g^s_{33} = 1. \tag{2.60}$$

LM uses the representation of vectors and spatial differential operators with respect to the normalized base. The g^s_{ii} are used to reformulate the equations resulting from the direct transformation correspondingly. Thus, any vector $\mathbf{A}$ with components A^i in the $\mathbf{q}_i$-base is

rewritten with physical components $\tilde{A}^i$ using the normalized set of base vectors:

$$\mathbf{A} = \sum_n A^n \mathbf{q}_n = \tilde{A}^\lambda \mathbf{e}_\lambda + \tilde{A}^\varphi \mathbf{e}_\varphi + \tilde{A}^z \mathbf{e}_z,$$

$$\text{with} \qquad \tilde{A}^\lambda = \sqrt{g^s_{11}} A^1, \qquad \tilde{A}^\varphi = \sqrt{g^s_{22}} A^2, \qquad \tilde{A}^z = \sqrt{g^s_{33}} A^3. \tag{2.61}$$

The physical components of the velocity vector $\mathbf{v}$ in the rotated spherical coordinate system are assigned in an obvious manner with u for the zonal wind velocity, v for the meridional wind velocity and w for the vertical velocity:

$$u \equiv \tilde{v}^\lambda = a \cos\varphi \dot{\lambda}, \qquad v \equiv \tilde{v}^\varphi = a\dot{\varphi}, \qquad w \equiv \tilde{v}^z = \dot{z} = \dot{r}. \tag{2.62}$$

2.3.2 The Transformed Dynamic Equations

Using general transformation relations and applying the metrical simplification for the spherical coordinate system, the Nabla operator is formulated by

$$\nabla = \frac{\mathbf{e}_\lambda}{a \cos\varphi} \frac{\partial}{\partial \lambda} + \frac{\mathbf{e}_\varphi}{a} \frac{\partial}{\partial \varphi} + \mathbf{e}_z \frac{\partial}{\partial z} \tag{2.63}$$

and the divergence of a vector $\mathbf{A}$ using its physical components $\tilde{A}^i$ is calculated from

$$\nabla \cdot \mathbf{A} = \frac{1}{a \cos\varphi} \left(\frac{\partial \tilde{A}^\lambda}{\partial \lambda} + \frac{\partial}{\partial \varphi} (\tilde{A}^\varphi \cos\varphi) \right) + \frac{\partial \tilde{A}^z}{\partial z}. \tag{2.64}$$

The basic prognostic set (2.50) - (2.54) of hydrodynamic equations then takes the following form in the (λ, φ, z)-system:

$$\begin{aligned}
\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u - \frac{uv}{a} \tan\varphi - fv &= -\frac{1}{\rho a \cos\varphi} \frac{\partial p}{\partial \lambda} + M_u \\
\frac{\partial v}{\partial t} + \mathbf{v} \cdot \nabla v + \frac{u^2}{a} \tan\varphi + fu &= -\frac{1}{\rho a} \frac{\partial p}{\partial \varphi} + M_v \\
\frac{\partial w}{\partial t} + \mathbf{v} \cdot \nabla w &= -\frac{1}{\rho} \frac{\partial p}{\partial z} - g + M_w \\
\frac{\partial p}{\partial t} + \mathbf{v} \cdot \nabla p &= -(c_{pd}/c_{vd}) pD + (c_{pd}/c_{vd} - 1) \rho c_{pd} Q_T \\
\frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T &= \frac{1}{\rho c_{pd}} \left(\frac{\partial p}{\partial t} + \mathbf{v} \cdot \nabla p \right) + Q_T \\
\frac{\partial q^v}{\partial t} + \mathbf{v} \cdot \nabla q^v &= -(S^l + S^f) + M_{q^v} \\
\frac{\partial q^{l,f}}{\partial t} + \mathbf{v} \cdot \nabla q^{l,f} - \frac{1}{\rho} \frac{\partial P_{l,f}}{\partial z} &= S^{l,f} + M_{q^{l,f}} \\
\rho &= p \{ R_d (1 + (R_v/R_d - 1) q^v - q^l - q^f) T \}^{-1}.
\end{aligned} \tag{2.65}$$

The advection operator is defined by

$$\mathbf{v} \cdot \nabla = \frac{1}{a \cos\varphi} \left(u \frac{\partial}{\partial \lambda} + v \cos\varphi \frac{\partial}{\partial \varphi} \right) + w \frac{\partial}{\partial z}, \tag{2.66}$$

and D denotes the three-dimensional wind divergence to be calculated from

$$D \equiv \nabla \cdot \mathbf{v} = \frac{1}{a \cos\varphi} \left(\frac{\partial u}{\partial \lambda} + \frac{\partial}{\partial \varphi}(v \cos\varphi) \right) + \frac{\partial w}{\partial z}. \tag{2.67}$$

The Coriolis parameter $f = 2\Omega \sin\varphi_g$ depends on the rotated (λ, φ)-coordinates and on the geographical latitude φ_g^N of the rotated pole.

$$f = 2\Omega \left(\cos\varphi_g^N \cos\varphi \cos\lambda + \sin\varphi \sin\varphi_g^N \right). \tag{2.68}$$

Since the set (2.65) is written in advection form, the following symbols have been introduced for convenience: The M-terms denote the source terms due to turbulent mixing,

$$\begin{aligned} M_u &\equiv -\frac{1}{\rho}(\nabla \cdot \underline{\mathbf{T}}) \cdot \mathbf{e}_\lambda, \quad M_v \equiv -\frac{1}{\rho}(\nabla \cdot \underline{\mathbf{T}}) \cdot \mathbf{e}_\varphi, \quad M_w \equiv -\frac{1}{\rho}(\nabla \cdot \underline{\mathbf{T}}) \cdot \mathbf{e}_z, \\ M_{q^x} &\equiv -\frac{1}{\rho}\nabla \cdot \mathbf{F}^x, \quad M_T \equiv -\frac{1}{\rho c_{pd}}\nabla \cdot \mathbf{H}, \end{aligned} \tag{2.69}$$

S^l and S^f represent the cloud microphysical sources and sinks per unit mass of moist air,

$$S^{l,f} \equiv \frac{1}{\rho} I^{l,f}, \tag{2.70}$$

and P_l and P_f denote the absolute values of the gravitational diffusion fluxes of water and ice, the precipitation fluxes,

$$\mathbf{P}^{l,f} = -P_{l,f}\mathbf{e}_z = -\rho q^{l,f} |\mathbf{v}_T^{l,f}| \mathbf{e}_z. \tag{2.71}$$

Q_T is the diabatic heating term in the prognostic equation for temperature,

$$Q_T \equiv \frac{1}{\rho c_{pd}} Q_h = \frac{L_V}{c_{pd}} S^l + \frac{L_S}{c_{pd}} S^f + M_T + Q_r, \tag{2.72}$$

where Q_r denotes the temperature change due to convergence/divergence of the flux of solar and thermal electromagnetic radiation:

$$Q_r \equiv -\frac{1}{\rho c_{pd}} \nabla \cdot \mathbf{R}. \tag{2.73}$$

The formulation and calculation of the mixing terms M, the terms $S^{l,f}$ and $P_{l,f}$ from the hydrological cycle and the radiative heating rate Q_r will be discussed separately in Part II of the documentation.

2.3.3 Special Transformation Relations

For some pre- and postprocessing purposes it might be necessary to transform the rotated horizontal coordinates (λ, φ) to the geographical longitude/latitude (λ_g, φ_g) and vice versa.

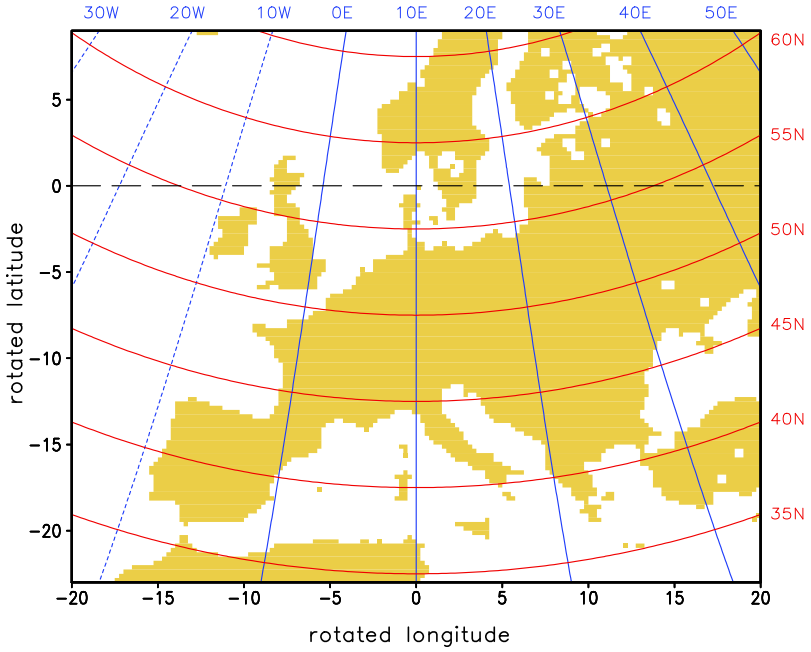

Figure 2.2: Geographical longitude (blue) and latitude (red) for a rotated grid with pole coordinates $\varphi_g^N = 32.5°$ and $\lambda_g^N = -170.0°$. The equator of the rotated grid is indicated by the dashed line. The rotated 0° meridian is conform with the 10°E geographical meridian.

The corresponding transformation relations are

$$\begin{aligned} \varphi &= \arcsin\left\{\sin\varphi_g \sin\varphi_g^N + \cos\varphi_g \cos\varphi_g^N \cos(\lambda_g - \lambda_g^N)\right\} \\ \lambda &= \arctan\left\{\frac{\cos\varphi_g \sin(\lambda_g - \lambda_g^N)}{\cos\varphi_g \sin\varphi_g^N \cos(\lambda_g - \lambda_g^N) - \sin\varphi_g \cos\varphi_g^N}\right\} \\ \text{and} & \\ \varphi_g &= \arcsin\left\{\sin\varphi \sin\varphi_g^N + \cos\varphi \cos\lambda \cos\varphi_g^N\right\} \\ \lambda_g &= \arctan\left\{\frac{\cos\varphi \sin\lambda}{\sin\varphi_g^N \cos\varphi \cos\lambda - \sin\varphi \cos\varphi_g^N}\right\} + \lambda_g^N \end{aligned} \tag{2.74}$$

Figure 2.2 illustrates the position of the geographical longitude and latitude for a rotated grid running from $-20°$ to $+20°$ in λ-direction (counting positive 'east' to the 0° rotated meridian) and from $-23°$ to 9° in φ-direction (counting positive 'north' to the rotated equator). The geographical coordinates of the rotated North Pole are specified as $\varphi_g^N = 32.5°$ (counting positive 'north' to the equator) and $\lambda_g^N = -170.0°$ (counting positive east to the Greenwich meridian). The inverse trigonometric functions used in the transformation relations (2.74) are not unique for all combinations of angles. In order to get a unique forward and backward transformation, we require that the geographical coordinates of the rotated pole are specified in the interval $-180° \leq \lambda_g^N \leq 180°$ and $0° \leq \varphi_g^N \leq 90°$.

The transformation of physical vector components and unit base vectors from the geographical to the rotated coordinate system is done with a matrix $\mathcal{P}$ which can be derived from the corresponding Jacobian matrix by normalization with the elements of the metric tensor. $\mathcal{P}$

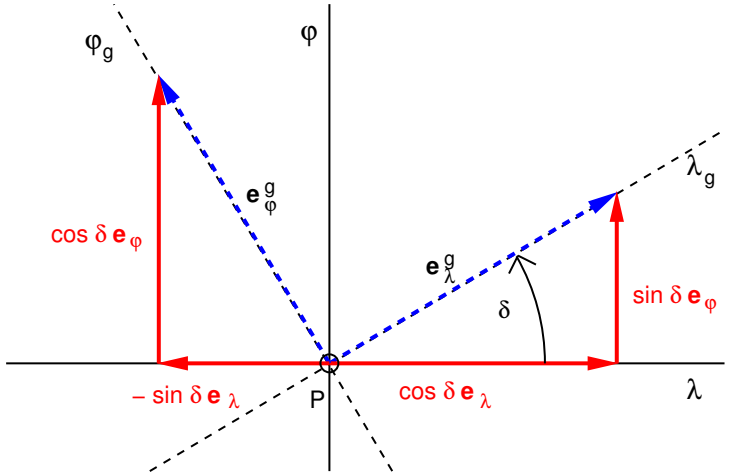

Figure 2.3: Transformation of the unit base vectors by a local rotation of angle δ around point P. To construct the base vectors of the geographical system, we use (2.77) to get $\mathbf{e}^g_\lambda = cos\delta\mathbf{e}_\lambda + sin\delta\mathbf{e}_\varphi$ and $\mathbf{e}^g_\varphi = -sin\delta\mathbf{e}_\lambda + cos\delta\mathbf{e}_\varphi$. Corresponding physical vector components transform accordingly. A similar illustration can be sketched for the reverse transformation.

is an orthonormal matrix with elements P_{ij} given by

$$\mathcal{P} = \begin{pmatrix} \cos\delta & -\sin\delta & 0 \\ \sin\delta & \cos\delta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{2.75}$$

where δ is the local angle formed by the geographical and the rotated meridians at a point. By defining δ to count positive for an anti-clockwise local rotation of the (λ, φ)-system to the (λ_g, φ_g) system, we have

$$\begin{aligned} \delta &= \arctan\left\{ \frac{\cos\varphi_g^N \sin\lambda}{\cos\varphi\sin\varphi_g^N - \cos\varphi_g^N \sin\varphi\cos\lambda} \right\}, \\ &= \arctan\left\{ \frac{\cos\varphi_g^N \sin(\lambda_g^N - \lambda_g)}{\cos\varphi_g \sin\varphi_g^N - \sin\varphi_g \cos\varphi_g^N \cos(\lambda_g^N - \lambda_g)} \right\}. \end{aligned} \tag{2.76}$$

The transformation is performed by applying the relations

$$\begin{aligned} \mathbf{e}_i &= \sum_n P_{in}\mathbf{e}^g_n \quad , \qquad \tilde{A}^i = \sum_n P_{in}\tilde{A}^g_n \,, \\ \mathbf{e}^g_i &= \sum_n P_{ni}\mathbf{e}_n \quad , \qquad \tilde{A}^i_g = \sum_n P_{ni}\tilde{A}_n \,. \end{aligned} \tag{2.77}$$

to the unit vectors $\mathbf{e}_i$ and to the physical components $\tilde{A}^i$ of a vector $\mathbf{A}$. Here, the index g denotes base vectors and vector components in the nonrotated geographical coordinate system, and $i = 1$ corresponds to the λ-direction, $i = 2$ to the φ-direction and $i = 3$ to the z-direction. Eqs. (2.77) represent a simple local rotation by the angle δ on the "horizontal" z-surfaces while the vertical components are not affected. This is illustrated in Figure (2.3).

In practice, the zonal and the meridional components of the wind velocity, u and v, are most frequently transformed from or to the corresponding components u^g and v^g in geographical

coordinates. From (2.77) and (2.75), these transformations are performed by

$$\begin{aligned} u &= u^g \cos\delta - v^g \sin\delta \\ v &= u^g \sin\delta + v^g \cos\delta \\ \text{and} & \\ u^g &= u \cos\delta + v \sin\delta \\ v^g &= -u \sin\delta + v \cos\delta. \end{aligned} \tag{2.78}$$

To calculate the components $\sin\delta$ and $\cos\delta$ of the matrix $\mathcal{P}$, it is not necessary to determine the angle of rotation by using the arctan-function in (2.76). A unique transformation is achieved by applying standard trigonometrical relations: denoting the nominator and denominator in (2.76) by

$$a = \cos\varphi_g^N \sin(\lambda_g - \lambda_g^N)\,, \quad b = \cos\varphi_g \sin\varphi_g^N - \sin\varphi_g \cos\varphi_g^N \cos(\lambda_g - \lambda_g^N)\,,$$

we simply have

$$\sin\delta = \frac{a}{\sqrt{a^2+b^2}}\,, \quad \cos\delta = \frac{b}{\sqrt{a^2+b^2}}\,. \tag{2.79}$$

2.4 The Model Base State

In LM, the thermodynamic variables are defined as the sums of base-state variables and deviations from the base state. The base or reference state is prescribed to be horizontally homogeneous, i.e. depending only on the height above the surface, time invariant and hydrostatically balanced.

The separation of the state variables in a base-state value and a deviation has a long tradition in nonhydrostatic models which are based on the anelastic approximation Dutton and Fichtl (1969). In obtaining the anelastic set of dynamic equations, linearization approximations with respect to the reference state are made: State variables that appear in factors of certain terms are replaced by their base-state values on the assumption that the fractional deviations are very small (on the order of one percent). Thus, models based on the anelastic system will only give reasonable results for cases in which the deviations are small initially and, moreover, remain small over the period of a simulation. Consequently, the base state has to be chosen very carefully in order to approximate the initial state of the atmosphere as closely as possible.

Some nonhydrostatic models based on the full compressible set of equations make also use of linearization approximations with respect to the base state, e.g., the MC2 model Tanguay et al. (1990) and the ARPS model Xue et al. (1995). Such linearizations simplify the equations considerably for the practical numerical integration, but confine the validity of the simulations to ranges which are consistent with the anelastic approximation.

Since LM is designed for operational NWP on a large domain covering an area of about 2000 km $\times$ 2000 km, fractional deviations from a horizontally homogeneous base state will be on the order of 10 percent. Deviations of this order of magnitude are too large for a useful application of linearization assumptions related to the anelastic approximation. For the basic dynamics of LM we thus follow the concept of the MM5 model Dudhia (1993) and do not a priori make use of any linearization approximations. In principle, this allows for an arbitrary

specification of the reference state which, in consequence, can be chosen less carefully and in a more general way than in anelastic type of models.

The main effect of introducing a reference state is the removal of horizontal base-state pressure gradient terms in the equation of motion. For not too large deviations of pressure from reference pressure, the removal of these terms reduces the computational error in the calculation of the pressure gradient force in case of sloping coordinate surfaces. Such sloping surfaces result from the use of a terrain-following vertical coordinate, as described in the next section.

By introducing the base state, any grid-scale thermodynamic variable ψ can be formally written as

$$\psi(\lambda, \varphi, z, t) = \psi_0(z) + \psi'(\lambda, \varphi, z, t). \tag{2.80}$$

The suffix zero indicates the base state value and the prime denotes the grid-scale deviation. As the turbulent fluxes have already been indicated by separate symbols, there is no danger of confusing ψ' with the turbulent fluctuations in Section 3.2. LM assumes the reference state to be dry and at rest. Thus we have:

$$u_0(z) = 0\,,\ v_0(z) = 0\,,\ w_0(z) = 0\,, \qquad q_0^v(z) = 0\,,\ q_0^l(z) = 0\,,\ q_0^f(z) = 0\,. \tag{2.81}$$

The model variables can then be written as:

$$\begin{aligned}
u(\lambda, \varphi, z, t) &= u'(\lambda, \varphi, z, t) \\
v(\lambda, \varphi, z, t) &= v'(\lambda, \varphi, z, t) \\
w(\lambda, \varphi, z, t) &= w'(\lambda, \varphi, z, t) \\
T(\lambda, \varphi, z, t) &= T_0(z) + T'(\lambda, \varphi, z, t) \\
p(\lambda, \varphi, z, t) &= p_0(z) + p'(\lambda, \varphi, z, t) \\
\rho(\lambda, \varphi, z, t) &= \rho_0(z) + \rho'(\lambda, \varphi, z, t) \\
q^x(\lambda, \varphi, z, t) &= q^{x\prime}(\lambda, \varphi, z, t), \qquad x = v, l, f.
\end{aligned} \tag{2.82}$$

T_0, p_0 and ρ_0 are related by the equation of state and the base state atmosphere is prescribed to be hydrostatically balanced:

$$\begin{aligned}
p_0 &= \rho_0 R_d T_0 \\
\frac{\partial p_0}{\partial z} &= -g\rho_0 = -\frac{g p_0}{R_d T_0}.
\end{aligned} \tag{2.83}$$

The vertical profiles of reference temperature and pressure are obtained from an integration of (2.83) by assuming, as proposed by Dudhia (1993), a constant rate of increase in temperature with the logarithm of pressure:

$$\frac{\partial T_0}{\partial \ln p_0} = \beta. \tag{2.84}$$

Performing the integration with the boundary values $p_{SL} = p_0(z = 0)$ and $T_{SL} = T_0(z = 0)$ for reference pressure and temperature at mean sea level $z = 0$, respectively, yields:

$$p_0(z) = \begin{cases} p_{SL} \exp\left\{-\frac{T_{SL}}{\beta}\left(1 - \sqrt{1 - \frac{2\beta g z}{R_d T_{SL}^2}}\right)\right\} & \text{if } \beta \neq 0 \\ p_{SL} \exp\left(-\frac{g z}{R_d T_{SL}}\right) & \text{if } \beta = 0 \end{cases}$$

$$T_0(z) \quad = \quad T_{SL}\sqrt{1-\frac{2\beta g z}{R_d T_{SL}^2}}. \tag{2.85}$$

The case $\beta = 0$ corresponds to an isothermal atmosphere. For $\beta > 0$ the reference atmosphere has a finite height z_{max} given by $z_{max} = R_d T_{SL}^2/(2\beta g)$. The top of the model domain has then to be positioned below this maximum value in order to avoid unrealistical low reference temperatures in the vicinity of the upper boundary. For the parameters p_{SL}, T_{SL} and β defining the base state, LM uses the default values

$$p_{SL} = 1000hPa, \quad T_{SL} = 288.15K, \quad \beta = 42K,$$

which result in reasonable vertical profiles of temperature, pressure and density throughout the troposphere. For the default parameters given above, the finite height of the reference atmosphere is at $z_{max} \simeq 29$ km. Figure (2.4) compares the default LM base state profiles of temperature and pressure against those for the US standard atmosphere. The differences are quite small, except for temperatures above the tropopause. In a later version of LM we intend to add other options besides the logarithmical profile (2.84) allowing for polytropic, isentropic and more generally defined reference atmospheres.

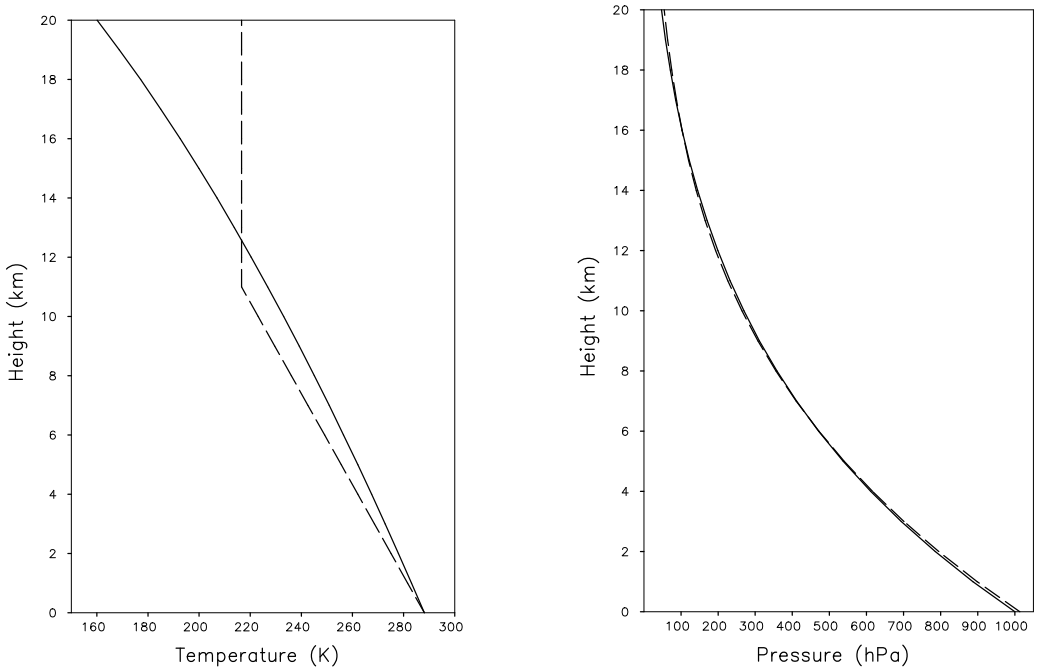

Figure 2.4: Profiles of reference temperature (left) and reference pressure (right) obtained by (2.85) for the LM default parameters of p_{SL}, T_{SL} and β (solid lines). The dashed lines indicate the profiles obtained from the US standard atmosphere.

LM uses the pressure perturbation as dependent model variable whereas the full form of the heat equation is retained. Because of the horizontal homogeneity of base-state pressure, the horizontal components of the pressure gradient force become

$$\frac{\partial p}{\partial \lambda} = \frac{\partial p'}{\partial \lambda}, \qquad \frac{\partial p}{\partial \varphi} = \frac{\partial p'}{\partial \varphi}, \tag{2.86}$$

and the advection of pressure reads

$$\mathbf{v} \cdot \nabla p = \mathbf{v} \cdot \nabla p' - g\rho_0 w. \tag{2.87}$$

The vertical acceleration due to the pressure gradient and gravity is rewritten as

$$-\frac{1}{\rho}\frac{\partial p}{\partial z} - g = -\frac{1}{\rho}\frac{\partial p'}{\partial z} - g\frac{\rho'}{\rho},$$

and the buoyancy term $g\rho'/\rho$ is expanded in the following form (using α from Eq. (2.16)):

$$\begin{aligned} g\frac{\rho'}{\rho} &= \frac{g}{\rho}\left(\frac{p}{R_d T_v} - \frac{p_0}{R_d T_0}\right) \\ &= \frac{g\rho_0}{(1+\alpha)\rho}\left(\frac{T_0 p}{T p_0} - 1 - \alpha\right) \\ &= \frac{g\rho_0}{(1+\alpha)\rho}\left(\frac{T_0 p'}{T p_0} - \frac{T'}{T} - \alpha\right) \end{aligned}$$

Here, the term $(1+\alpha)$ in the denominator represents a small correction of density due to the concentrations of the water constituents, which may readily be replaced by 1 because α is on the order of one percent. Thus,

$$\begin{aligned} -\frac{1}{\rho}\frac{\partial p}{\partial z} - g &= -\frac{1}{\rho}\frac{\partial p'}{\partial z} + B, \\ \text{where} \qquad & \\ B &= g\frac{\rho_0}{\rho}\left\{\frac{T'}{T} - \frac{T_0 p'}{T p_0} + \left(\frac{R_v}{R_d} - 1\right) q^v - q^l - q^f\right\} \end{aligned} \tag{2.88}$$

denotes the buoyant vertical acceleration. B from (2.88) resembles the buoyancy term resulting from the anelastic scale analysis. It is, however, not identical to the anelastic form because no scale approximations for the fractional temperature and pressure perturbations have been made.

The last two terms on the right hand side in the B-definition represent the effect of water loading. For liquid and solid hydrometeors suspended in the air or falling at terminal velocity, the gravity acceleration acting on the particles is balanced by aerodynamical forces of friction. Due to the conservation of barycentric momentum, these frictional forces in turn affect the moist air surrounding the particles by an acceleration which is equal in magnitude but opposite in sign. Clearly, as the equation of motion is formulated with respect to barycentric velocity, the effect of water loading is only implicitly taken into account. By the expansion (2.88) of the pressure gradient and gravity acceleration with respect to the dry base state, however, the physical mechanism of water loading, i.e. the generation of a downward vertical acceleration due to the weight of liquid and solid forms of water, is revealed explicitly.

Using Eqs. (2.86), (2.87) and (2.88) the set (2.65) of basic equations is now rewritten with p' as dependent variable:

$$\begin{aligned} \frac{\partial u}{\partial t} + \mathbf{v}\cdot\nabla u - \frac{uv}{a}\tan\varphi - fv &= -\frac{1}{\rho a\cos\varphi}\frac{\partial p'}{\partial \lambda} + M_u \\ \frac{\partial v}{\partial t} + \mathbf{v}\cdot\nabla v + \frac{u^2}{a}\tan\varphi + fu &= -\frac{1}{\rho a}\frac{\partial p'}{\partial \varphi} + M_v \end{aligned}$$

$$\begin{aligned} \frac{\partial w}{\partial t} + \mathbf{v}\cdot\nabla w &= -\frac{1}{\rho}\frac{\partial p'}{\partial z} + B + M_w \\ \frac{\partial p'}{\partial t} + \mathbf{v}\cdot\nabla p' - g\rho_0 w &= -(c_{pd}/c_{vd})pD + (c_{pd}/c_{vd}-1)\rho c_{pd} Q_T \\ \frac{\partial T}{\partial t} + \mathbf{v}\cdot\nabla T &= \frac{1}{\rho c_{pd}}\left(\frac{\partial p'}{\partial t} + \mathbf{v}\cdot\nabla p' - g\rho_0 w\right) + Q_T \end{aligned} \tag{2.89}$$

The set (2.89) is completed by the budget equations for the water constituents and the equation of state from (2.65), which are not affected by the introduction of the base state. Total pressure is calculated as the sum of base-state pressure and the predicted pressure perturbation, $p = p_0 + p'$, and the temperature perturbation occurring in the buoyancy term B is obtained from the reference value and the predicted total temperature, $T' = T - T_0$. Total density then results as a diagnostic variable from the equation of state.

In the next section, this set of equations is transformed to a nonorthogonal curvilinear coordinate system with a terrain following vertical coordinate ζ. The base-state variables T_0, p_0 and ρ_0 are independent of the horizontal coordinates λ and φ in the original z-system, but become functions of all three independent variables $(\lambda, \varphi, \zeta)$ in the ζ-system. Therefore, the base-state arrays in the model are three-dimensional. The reference state varies along the ζ-coordinate surfaces when these are not flat. This is usually true when terrain is included.

2.5 Terrain-following Coordinates

The set (2.89) of equations in the previous chapter is written in the curvilinear but orthogonal spherical coordinate system with geometrical height z above mean sea level as vertical coordinate. When surface terrain is included, the numerical solution of these equations becomes quite complex because of a costly formulation of the lower boundary conditions. A convenient method to overcome this problem is the transformation to a terrain-following coordinate system, where the lowest surface of constant vertical coordinate becomes conformal to the orography. Examples of such coordinates are the well known pressure-based *Sigma* coordinate for large-scale hydrostatic modelling and the *Gal-Chen* coordinate Gal-Chen and Sommerville (1975) for small-scale nonhydrostatic modelling.

2.5.1 Basic Geometry and Definitions

The LM model equations will be formulated in a generalized terrain-following coordinate system, where any monotonic function of geometrical height can be used as vertical coordinate. This generalized vertical coordinate is denoted by ζ. Since ζ is prescribed to be time independent, the resulting ζ-coordinate system represents a nondeformable system, where surfaces of constant ζ are fixed in physical space (in contrast to various pressure-based vertical coordinates used in hydrostatic models).

To set up the transformation relations more conveniently, we denote the coordinates of the original z-system by q^i,

$$\begin{aligned} q^1 &= \lambda \\ q^2 &= \varphi \\ q^3 &= z, \end{aligned} \tag{2.90}$$

and the coordinates of the new ζ-system with a^i. On the condition that a scalar function $a^3 = \zeta(\lambda, \varphi, z)$ exists, which is always monotonic in the variable z, the relations between the old coordinates q^i and the new coordinates a^i of the ζ-system are uniquely defined with the equations

$$\begin{aligned} a^1 &= \lambda = q^1 \\ a^2 &= \varphi = q^2 \\ a^3 &= \zeta(\lambda, \varphi, z). \end{aligned} \tag{2.91}$$

For the reverse transformation, q^3 in (2.90) is a unique function in the variable ζ, $q^3 = z(\lambda, \varphi, \zeta)$. The ζ-system defined by (2.91) is a special case of a fully three-dimensional curvilinear coordinate system, since the coordinate surfaces of constant λ and φ remain the same. However, terrain following coordinate surfaces, which become conformal to the orography at the ground, are accommodated by suitable vertical transformations. Figure 2.5 illustrates the geometry of the z-system and the ζ-system.

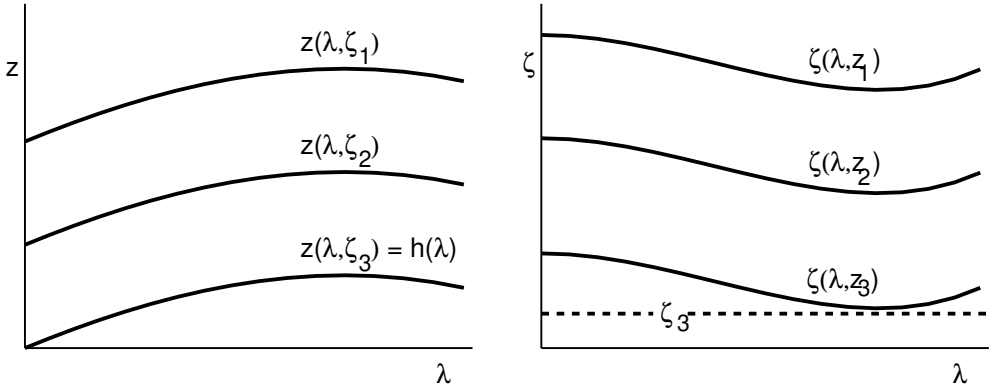

Figure 2.5: Left: geometric height $z(\lambda, \zeta)$ for various fixed values of the terrain following coordinate ζ. For a specific value of ζ (here ζ_3), the corresponding height $z(\lambda, \zeta)$ may be conformal with the height $h(\lambda)$ of the terrain. Right: coordinate values $\zeta(\lambda, z)$ for various fixed values of the geometrical height z. A fixed value of ζ (here ζ_3) represents the terrain height.

Because the ζ-system is curvilinear and nonorthogonal, we have to distinguish between covariant and contravariant base vectors and vector components. We will adopt the convention that subscripts indicate covariant quantities whereas superscripts indicate contravariant quantities. Covariant base vectors are tangent to the lines of constant curvilinear coordinates, contravariant base vectors are normal to the surfaces of constant coordinates. For the original (λ, φ, z)-system the two sets of base vectors are:

$$\begin{aligned} \mathbf{q}_1 &= a\cos\varphi \mathbf{e}_\lambda\,, & \mathbf{q}^1 &= (a\cos\varphi)^{-1}\mathbf{e}_\lambda\,, \\ \mathbf{q}_2 &= a\mathbf{e}_\varphi\,, & \mathbf{q}^2 &= (a)^{-1}\mathbf{e}_\varphi\,, \\ \mathbf{q}_3 &= \mathbf{e}_z\,, & \mathbf{q}^3 &= \mathbf{e}_z\,. \end{aligned} \tag{2.92}$$

Since the z-system is orthogonal, the covariant and contravariant base vectors have the same direction and there is no difference when normalized, i.e. unit vectors are used.

This is not true for the ζ-system, and any vector $\mathbf{A}$ may be represented in terms of either set of base vectors, using covariant or contravariant vector components. The covariant and

contravariant base vectors of the ζ-system are denoted by $\mathbf{a}_i$ and $\mathbf{a}^i$, respectively. For these base vectors the relation $\mathbf{a}_i \cdot \mathbf{a}^j = \delta_i^j$, where δ_i^j is the Kronecker delta, holds by definition. Denoting the contravariant and the covariant components of the vector $\mathbf{A}$ in the original orthogonal z-system by A^i and A_i, respectively, and those in the new nonorthogonal ζ-system by $\hat{A}^i$ and $\hat{A}_i$, $\mathbf{A}$ may be represented by

$$\begin{aligned} \mathbf{A} &= \sum_n A^n \mathbf{q}_n = \sum_n A_n \mathbf{q}^n \\ &= \sum_n \hat{A}^n \mathbf{a}_n = \sum_n \hat{A}_n \mathbf{a}^n. \end{aligned} \tag{2.93}$$

The mapping of the base vectors and the corresponding components of any vector is accomplished by the Jacobian matrix $\mathcal{J}^\zeta$ of the transformation with elements J_{ij}^ζ given by

$$J_{ij}^\zeta \equiv \frac{\partial a^j}{\partial q^i} \tag{2.94}$$

and the inverse Jacobian matrix $\mathcal{J}^z = (\mathcal{J}^\zeta)^{-1}$ with elements J_{ij}^z given by

$$J_{ij}^z \equiv \frac{\partial q^j}{\partial a^i}. \tag{2.95}$$

Base vectors and vector components transform contragredient to each other. For the set of covariant base vectors the transformation is given by

$$\begin{aligned} \mathbf{a}_i &= \sum_n J_{in}^z \, \mathbf{q}_n \quad ; \quad \mathbf{q}_i = \sum_n J_{in}^\zeta \, \mathbf{a}_n \\ \hat{A}^i &= \sum_n J_{ni}^\zeta \, A^n \quad ; \quad A^i = \sum_n J_{ni}^z \, \hat{A}^n \end{aligned} \tag{2.96}$$

and for the contravariant base the following relations hold:

$$\begin{aligned} \mathbf{a}^i &= \sum_n J_{ni}^\zeta \, \mathbf{q}^n \quad ; \quad \mathbf{q}^i = \sum_n J_{ni}^z \, \mathbf{a}^n \\ \hat{A}_i &= \sum_n J_{in}^z \, A_n \quad ; \quad A_i = \sum_n J_{in}^\zeta \, \hat{A}_n. \end{aligned} \tag{2.97}$$

For the transformation defined by (2.91) the Jacobian matrix and its inverse read

$$\mathcal{J}^\zeta = \begin{pmatrix} 1 & 0 & (\partial\zeta/\partial\lambda)_z \\ 0 & 1 & (\partial\zeta/\partial\varphi)_z \\ 0 & 0 & \partial\zeta/\partial z \end{pmatrix} \tag{2.98}$$

$$\mathcal{J}^z = \begin{pmatrix} 1 & 0 & (\partial z/\partial\lambda)_\zeta \\ 0 & 1 & (\partial z/\partial\varphi)_\zeta \\ 0 & 0 & \partial z/\partial\zeta \end{pmatrix}. \tag{2.99}$$

All computations will be done in the ζ-system, where the model variables are functions of the independent variables, the new coordinates $(\lambda, \varphi, \zeta)$. Consequently, it is convenient to

formulate the elements of the Jacobian matrix $\mathcal{J}^\zeta$ in terms of the elements of the inverse matrix $\mathcal{J}^z$, which can be easily calculated in the ζ-system. Applying the orthogonality relation $\sum_n J^z_{in} J^\zeta_{nj} = \delta^j_i$ yields

$$\begin{aligned} \left(\frac{\partial\zeta}{\partial\lambda}\right)_z &= -\left(\frac{\partial z}{\partial\zeta}\right)^{-1}\left(\frac{\partial z}{\partial\lambda}\right)_\zeta \\ \left(\frac{\partial\zeta}{\partial\varphi}\right)_z &= -\left(\frac{\partial z}{\partial\zeta}\right)^{-1}\left(\frac{\partial z}{\partial\varphi}\right)_\zeta \\ \frac{\partial\zeta}{\partial z} &= \left(\frac{\partial z}{\partial\zeta}\right)^{-1}. \end{aligned} \tag{2.100}$$

One can easily show that the inverse of the Jacobian matrix of the transformation from the z-system to the terrain-following coordinate system, $\mathcal{J}^z$, is related to the covariant metric tensor $\mathcal{G}^T$ according to

$$\begin{aligned} \sqrt{G^T} &\equiv \sqrt{\det\left(\mathcal{G}^T\right)} = \sqrt{G^s}\sqrt{G} \\ \text{where} \qquad \sqrt{G} &\equiv \left|\det\left(\mathcal{J}^z\right)\right| = \left|\frac{\partial z}{\partial\zeta}\right|. \end{aligned} \tag{2.101}$$

The elements g^T_{ij} of the matrix $\mathcal{G}^T$ are given by the scalar products of the corresponding covariant base vectors in the ζ-system, i.e. $g^T_{ij} = \mathbf{a}_i \cdot \mathbf{a}_j$. $\sqrt{G^T}$ denotes the total Jacobian of the ζ-system, which may be calculated as the product of the Jacobian of the spherical z-system, $\sqrt{G^s}$ from (2.59), and the Jacobian $\sqrt{G}$ of the transformation from the z- to the ζ-system. The latter is defined as the absolute value of the determinant of the inverse Jacobian matrix $\mathcal{J}^z$. This determinant is positive for a right-handed transformation, where ζ increases monotonically with z. For a left-handed transformation, ζ will decrease for increasing z and $\det(\mathcal{J}^z)$ becomes negative.

Most hydrostatic NWP-models use pressure-based vertical coordinates and thus left-handed coordinate systems. For this traditional reason, LM is also formulated in a left-handed terrain-following system, where ζ increases from the top of the model domain to the bottom and the ζ-surface corresponding to the lower boundary will become conformal to the terrain. Thus, by definition, $\partial z/\partial\zeta$ is always negative and

$$\sqrt{G} = \left|\det\left(\mathcal{J}^z\right)\right| = -\frac{\partial z}{\partial\zeta} > 0\,. \tag{2.102}$$

Denoting the nonunity and nonzero elements of the inverse Jacobian matrix by

$$\begin{aligned} J_\lambda &\equiv J^z_{13} = \left(\frac{\partial z}{\partial\lambda}\right)_\zeta, \\ J_\varphi &\equiv J^z_{23} = \left(\frac{\partial z}{\partial\varphi}\right)_\zeta, \\ J_\zeta &\equiv J^z_{33} = \frac{\partial z}{\partial\zeta} = -\sqrt{G}, \end{aligned} \tag{2.103}$$

the matrices (2.98) and (2.99) can be rewritten as

$$\mathcal{J}^\zeta = \begin{pmatrix} 1 & 0 & J_\lambda/\sqrt{G} \\ 0 & 1 & J_\varphi/\sqrt{G} \\ 0 & 0 & -1/\sqrt{G} \end{pmatrix} \tag{2.104}$$

$$\mathcal{J}^z = \begin{pmatrix} 1 & 0 & J_\lambda \\ 0 & 1 & J_\varphi \\ 0 & 0 & -\sqrt{G} \end{pmatrix}. \tag{2.105}$$

The geometry of the ζ-system is uniquely determined by the two sets of new base vectors. The covariant base vectors with respect to the original z-system result from (2.96) and (2.105). They are

$$\begin{aligned} \mathbf{a}_1 &= \mathbf{q}_1 + J_\lambda \mathbf{q}_3 = a\cos\varphi\,\{\mathbf{e}_\lambda + (J_\lambda/a\cos\varphi)\mathbf{e}_z\} &&= a\cos\varphi\tilde{\mathbf{a}}_1\,, \\ \mathbf{a}_2 &= \mathbf{q}_2 + J_\varphi \mathbf{q}_3 = a\,\{\mathbf{e}_\varphi + (J_\varphi/a)\mathbf{e}_z\} &&= a\tilde{\mathbf{a}}_2\,, \\ \mathbf{a}_3 &= -\sqrt{G}\mathbf{q}_3 = -\sqrt{G}\mathbf{e}_z &&= \sqrt{G}\tilde{\mathbf{a}}_3\,. \end{aligned} \tag{2.106}$$

Here, the $\tilde{\mathbf{a}}_i$ denote normalized base vectors of the ζ-system, where the roots of the metric coefficients g^s_{ii} of the original z-system have been used for normalization. $\tilde{\mathbf{a}}_3 = -\mathbf{e}_z$ is the unit vector in ζ-direction. However, $\tilde{\mathbf{a}}_1$ and $\tilde{\mathbf{a}}_2$ are not normalized to unity, i.e. they are not unit base vectors. The contravariant set of base vectors results from (2.97) and (2.104):

$$\begin{aligned} \mathbf{a}^1 &= \mathbf{q}^1 = (1/a\cos\varphi)\mathbf{e}_\lambda \\ \mathbf{a}^2 &= \mathbf{q}^2 = (1/a)\mathbf{e}_\varphi \\ \mathbf{a}^3 &= (J_\lambda \mathbf{q}^1 + J_\varphi \mathbf{q}^2 - \mathbf{q}^3)/\sqrt{G} \\ &= (J_\lambda \mathbf{e}_\lambda + J_\varphi \cos\varphi \mathbf{e}_\varphi - a\cos\varphi \mathbf{e}_z)/(a\cos\varphi\sqrt{G}) \end{aligned} \tag{2.107}$$

As clearly revealed by these sets of base vectors, the ζ-system is not an orthogonal system when the terrain is not flat ($J_\lambda \neq 0, J_\varphi \neq 0$). For the formulation of the model we will use the covariant set of normalized base vectors $\tilde{\mathbf{a}}_i$ (see Figure 2.6).

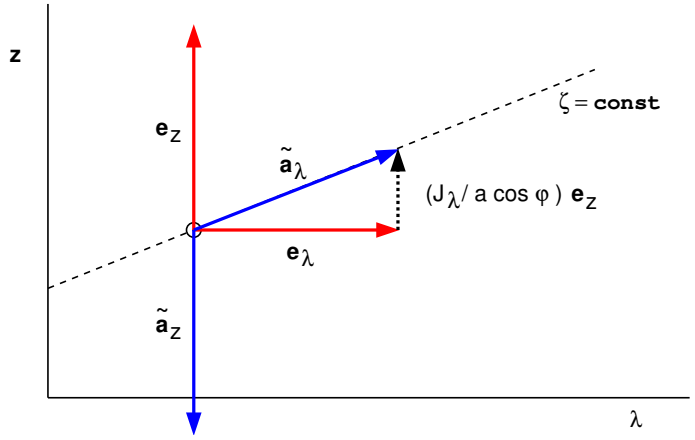

Figure 2.6: Unit vectors $\mathbf{e}_\lambda$ and $\mathbf{e}_z$ at a point on a surface of constant ζ (open circle) in a (λ, z)-plane (in red). The corresponding normalized base vectors $\tilde{\mathbf{a}}_\lambda$ and $\tilde{\mathbf{a}}_z$ of the ζ-system are indicated in blue. The dotted vector is the second vector component of $\tilde{\mathbf{a}}_\lambda = \mathbf{e}_\lambda + (J_\lambda/a\cos\varphi)\mathbf{e}_z$.

2.5.2 Transformation Relations

This subsection provides some useful relations to transform the set of basic equations from the z- to the ζ-system.

In LM, all vector variables and spatial differential operators are formulated with respect to the set of normalized covariant base vectors, i.e. $\tilde{\mathbf{a}}_1$, $\tilde{\mathbf{a}}_2$, $\tilde{\mathbf{a}}_3$. This allows to use the physical vector components of the original z-system instead of the contravariant components as dependent model variables, while doing all computations in the nonorthogonal terrain-following coordinate system. The only exception is the contravariant vertical velocity in the ζ-system, $\dot{\zeta}$, which becomes a diagnostic model variable.

(a) *Vector Components*
Any vector $\mathbf{A}$ with contravariant components A^i and physical components $\tilde{A}^i$ in the original z-system may be written in the ζ-system as

$$\mathbf{A} = \hat{A}^1\mathbf{a}_1 + \hat{A}^2\mathbf{a}_2 + \hat{A}^3\mathbf{a}_3.$$

Using (2.96) and (2.105), the contravariant vector components $\hat{A}^i$ are given by

$$\begin{aligned} \hat{A}^1 &= A^1 \\ \hat{A}^2 &= A^2 \\ \hat{A}^3 &= (J_\lambda A^1 + J_\varphi A^2 - A^3)/\sqrt{G}. \end{aligned} \tag{2.108}$$

Referring $\mathbf{A}$ to the normalized base vectors $\tilde{\mathbf{a}}_i$ yields

$$\mathbf{A} = \breve{A}^1\tilde{\mathbf{a}}_1 + \breve{A}^2\tilde{\mathbf{a}}_2 + \breve{A}^3\tilde{\mathbf{a}}_3, \tag{2.109}$$

where the normalized vector components $\breve{A}^i$ can be calculated in terms of the physical vector components $\tilde{A}^i$ of the z-system:

$$\begin{aligned} \breve{A}^1 &= \tilde{A}^1 \quad (= A^1 a\cos\varphi) \\ \breve{A}^2 &= \tilde{A}^2 \quad (= A^1 a) \\ \breve{A}^3 &= (J_\lambda/a\cos\varphi)\tilde{A}^1 + (J_\varphi/a)A^2 - A^3. \end{aligned} \tag{2.110}$$

Thus, by the representation (2.109) of $\mathbf{A}$, the horizontal vector components $\breve{A}^1$ and $\breve{A}^2$ become identical to the physical vector components in the original z-system.

(b) *Wind Velocity*
By applying (2.109) to the three-dimensional wind vector with the physical components u, v and w in the z-system,

$$\mathbf{v} = \dot{\lambda}\mathbf{q}_1 + \dot{\varphi}\mathbf{q}_2 + \dot{z}\mathbf{q}_3 = u\mathbf{e}_\lambda + v\mathbf{e}_\varphi + w\mathbf{e}_z,$$

$\mathbf{v}$ may be written in the ζ-system as

$$\mathbf{v} = u\tilde{\mathbf{a}}_1 + v\tilde{\mathbf{a}}_2 + \dot{\zeta}\sqrt{G}\tilde{\mathbf{a}}_3. \tag{2.111}$$

$\dot{\zeta}$ denotes the (nonnormalized) contravariant vertical velocity,

$$\dot{\zeta} = \frac{1}{\sqrt{G}}\left(\frac{J_\lambda}{a\cos\varphi}u + \frac{J_\varphi}{a}v - w\right), \tag{2.112}$$

which can be diagnosed from u, v and w. Figure 2.7 illustrates the representation of a wind vector $\mathbf{v}$ using orthogonal unit vectors of the z-system and normalized base vectors of the ζ-system.

(c) *Lagrangian Time Derivative*
The total time derivative of a scalar field function ψ is invariant and the general formulation for any set of coordinates a^i reads

$$\frac{d\psi}{dt} = \frac{\partial\psi}{\partial t} + \mathbf{v}\cdot\nabla\psi = \frac{\partial\psi}{\partial t} + \sum_n \dot{a}^n \frac{\partial\psi}{\partial a^n}.$$

For the ζ-system the Lagrangian time derivative becomes

$$\frac{d\psi}{dt} = \frac{\partial\psi}{\partial t} + \dot{\lambda}\frac{\partial\psi}{\partial\lambda} + \dot{\varphi}\frac{\partial\psi}{\partial\varphi} + \dot{\zeta}\frac{\partial\psi}{\partial\zeta}$$

and may be formulated using the physical wind components u and v:

$$\frac{d\psi}{dt} = \frac{\partial\psi}{\partial t} + \frac{1}{a\cos\varphi}\left(u\frac{\partial\psi}{\partial\lambda} + v\cos\varphi\frac{\partial\psi}{\partial\varphi}\right) + \dot{\zeta}\frac{\partial\psi}{\partial\zeta}. \tag{2.113}$$

(d) *Gradient Operator*
The Nabla operator applied to a scalar field function ψ forms an invariant vector which is represented by

$$\nabla\psi = \sum_n \mathbf{q}^n \frac{\partial\psi}{\partial q^n} = \sum_n \mathbf{a}^n \frac{\partial\psi}{\partial a^n}. \tag{2.114}$$

This general formulation uses contravariant base vectors, and the partial derivatives in (2.114) represent covariant vector components. Thus, partial derivatives with respect to the space coordinates will transform according to the rule (2.97) for covariant vector components. Using (2.105) for the inverse of the Jacobian matrix yields

$$\begin{aligned}
\left(\frac{\partial\psi}{\partial\lambda}\right)_z &= \left(\frac{\partial\psi}{\partial\lambda}\right)_\zeta + \frac{J_\lambda}{\sqrt{G}}\frac{\partial\psi}{\partial\zeta} \\
\left(\frac{\partial\psi}{\partial\varphi}\right)_z &= \left(\frac{\partial\psi}{\partial\varphi}\right)_\zeta + \frac{J_\varphi}{\sqrt{G}}\frac{\partial\psi}{\partial\zeta} \\
\frac{\partial\psi}{\partial z} &= -\frac{1}{\sqrt{G}}\frac{\partial\psi}{\partial\zeta}.
\end{aligned} \tag{2.115}$$

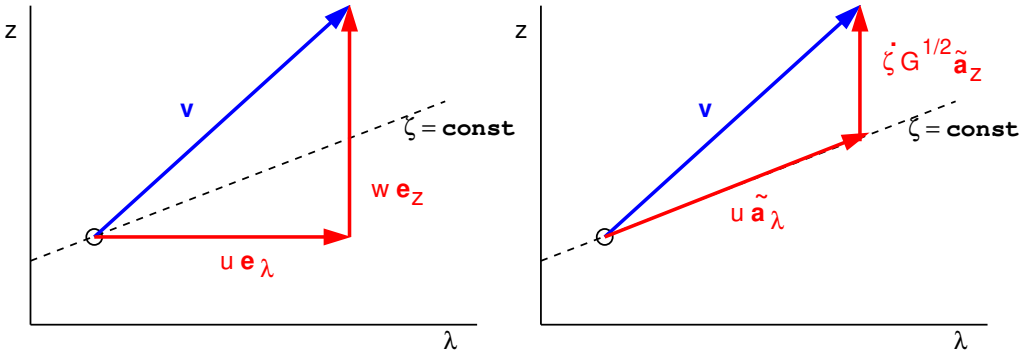

Figure 2.7: Wind vector $\mathbf{v}$ at a point (open circle) on a surface of constant ζ in a (λ, z)-plane. Left: vector decomposition along unit vectors $\mathbf{e}_\lambda$ and $\mathbf{e}_z$ of the orthogonal z-system. Right: vector decomposition along normalized base vectors $\tilde{\mathbf{a}}_\lambda$ and $\tilde{\mathbf{a}}_z$ of the ζ-system. In case of $\mathbf{v}$ being tangent to the ζ-surface, $\dot{\zeta} = 0$ and $\mathbf{v} = u\tilde{\mathbf{a}}_\lambda$; this is identical to $\mathbf{v} = u\mathbf{e}_\lambda + w\mathbf{e}_z$, since in this case $w = uJ_\lambda/a\cos\varphi$.

Especially, Eqs. (2.115) allow to formulate the vector $\nabla\psi$ with respect to the orthogonal basis $\mathbf{q}^i$, but to calculate the spatial derivatives within the terrain-following ζ-system. As will be described in Section 3.6, the equation of motion used in LM is formulated with respect to the orthogonal unit vectors $\mathbf{e}_\lambda$, $\mathbf{e}_\varphi$ and $\mathbf{e}_z$ of the spherical z-system. Thus, the transformation relations (2.115) can be used directly to transform the pressure gradient force according to the following representation of $\nabla\psi$:

$$\begin{aligned}\nabla\psi &= \frac{1}{a\cos\varphi}\left\{\left(\frac{\partial\psi}{\partial\lambda}\right)_\zeta + \frac{J_\lambda}{\sqrt{G}}\frac{\partial\psi}{\partial\zeta}\right\}\mathbf{e}_\lambda \\ &\quad + \frac{1}{a}\left\{\left(\frac{\partial\psi}{\partial\varphi}\right)_\zeta + \frac{J_\varphi}{\sqrt{G}}\frac{\partial\psi}{\partial\zeta}\right\}\mathbf{e}_\varphi - \frac{1}{\sqrt{G}}\frac{\partial\psi}{\partial\zeta}\mathbf{e}_z.\end{aligned} \tag{2.116}$$

The transformation relations (2.115) may also be written in an alternative form using the following expressions for the horizontal variation of the Jacobian $\sqrt{G}$:

$$\frac{\sqrt{G}}{\partial\lambda} = -\frac{\partial J_\lambda}{\partial\zeta}, \qquad \frac{\sqrt{G}}{\partial\varphi} = -\frac{\partial J_\varphi}{\partial\zeta}. \tag{2.117}$$

By applying (2.117) and the chain rule, (2.115) may be expanded to yield the strong conservation form

$$\begin{aligned}\sqrt{G}\left(\frac{\partial\psi}{\partial\lambda}\right)_z &= \frac{\partial}{\partial\lambda}\left(\sqrt{G}\psi\right)_\zeta + \frac{\partial}{\partial\zeta}\left(J_\lambda\psi\right) \\ \sqrt{G}\left(\frac{\partial\psi}{\partial\varphi}\right)_z &= \frac{\partial}{\partial\varphi}\left(\sqrt{G}\psi\right)_\zeta + \frac{\partial}{\partial\zeta}\left(J_\varphi\psi\right).\end{aligned} \tag{2.118}$$

(e) *Divergence of a Vector Field*
The divergence of any vector field $\mathbf{A}$ forms an invariant scalar. It is calculated in the ζ-system according to the general rule

$$\nabla\cdot\mathbf{A} = \frac{1}{\sqrt{G^T}}\sum_n \frac{\partial}{\partial a^n}\left(\sqrt{G^T}\hat{A}^n\right) \tag{2.119}$$

using the total Jacobian (2.101) of the transformation. The divergence of $\mathbf{A}$ may also be formulated with the contravariant vector components A^i of the orthogonal z-system instead of the contravariant components $\hat{A}^i$ of the ζ-system. From the transformation relations (2.108) for vector components the following representation of $\nabla\cdot\mathbf{A}$ in strong conservation form results:

$$\begin{aligned}\nabla\cdot\mathbf{A} &= \frac{1}{\sqrt{G^T}}\left\{\frac{\partial}{\partial\lambda}\left(\sqrt{G^T}A^1\right) + \frac{\partial}{\partial\zeta}\left(J_\lambda\sqrt{G^s}A^1\right)\right. \\ &\quad \left. + \frac{\partial}{\partial\varphi}\left(\sqrt{G^T}A^2\right) + \frac{\partial}{\partial\zeta}\left(J_\varphi\sqrt{G^s}A^2\right) - \frac{\partial}{\partial\zeta}\left(\sqrt{G^s}A^3\right)\right\}.\end{aligned} \tag{2.120}$$

By applying the relations from (2.117) for the Jacobian $\sqrt{G}$, $\nabla\cdot\mathbf{A}$ may be alternatively formulated in nonconservative form:

$$\begin{aligned}\nabla\cdot\mathbf{A} &= \frac{1}{\sqrt{G^s}}\left\{\frac{\partial}{\partial\lambda}\left(\sqrt{G^s}A^1\right) + \frac{J_\lambda}{\sqrt{G}}\frac{\partial}{\partial\zeta}\left(\sqrt{G^s}A^1\right)\right. \\ &\quad \left. + \frac{\partial}{\partial\varphi}\left(\sqrt{G^s}A^2\right) + \frac{J_\varphi}{\sqrt{G}}\frac{\partial}{\partial\zeta}\left(\sqrt{G^s}A^2\right) - \frac{1}{\sqrt{G}}\frac{\partial}{\partial\zeta}\left(\sqrt{G^s}A^3\right)\right\}.\end{aligned} \tag{2.121}$$

Both formulations (2.120) and (2.121) for $\nabla \cdot \mathbf{A}$ use the contravariant vector components of the original z-system, but the calculations are done within the ζ-system, i.e. the horizontal partial derivatives are computed along the sloping ζ-surfaces.

(f) *Wind Divergence*
We apply (2.120) and (2.121) to calculate the divergence of the three-dimensional wind field, $D = \nabla \cdot \mathbf{v}$, and use the physical vector components u, v and w with respect to the unit base vectors of the orthogonal z-system. The conservative form (2.121) yields

$$\begin{aligned} D &= \frac{1}{a\cos\varphi\sqrt{G}}\left\{\frac{\partial}{\partial\lambda}\left(\sqrt{G}u\right)+\frac{\partial}{\partial\zeta}\left(J_\lambda u\right)\right. \\ &\quad \left.+\frac{\partial}{\partial\varphi}\left(\sqrt{G}v\cos\varphi\right)+\frac{\partial}{\partial\zeta}\left(J_\varphi v\cos\varphi\right)\right\}-\frac{1}{\sqrt{G}}\frac{\partial w}{\partial\zeta}. \end{aligned} \tag{2.122}$$

and from (2.121) an alternative representation may be expanded:

$$\begin{aligned} D &= \frac{1}{a\cos\varphi}\left\{\frac{\partial u}{\partial\lambda}+\frac{J_\lambda}{\sqrt{G}}\frac{\partial u}{\partial\zeta}\right. \\ &\quad \left.+\frac{\partial}{\partial\varphi}\left(v\cos\varphi\right)+\cos\varphi\frac{J_\varphi}{\sqrt{G}}\frac{\partial v}{\partial\zeta}\right\}-\frac{1}{\sqrt{G}}\frac{\partial w}{\partial\zeta}. \end{aligned} \tag{2.123}$$

2.5.3 The Vertical Coordinate

For any practical computations in the terrain-following system, the vertical coordinate must be specified by a unique transformation relation $z = f(\lambda, \varphi, \zeta)$ to calculate the elements J_λ, J_φ and $\sqrt{G}$ of the Jacobian matrix. In order to keep the numerical formulation of the model equations independent from a specific choice on ζ, the coordinate transformation will be defined by a two-step procedure.

The first step involves a terrain-following transformation using a user-specified coordinate $\tilde{\zeta}$. In the second step, $\tilde{\zeta}$ is mapped to the computational coordinate ζ using a monotonic function m in the form $\tilde{\zeta} = m(\zeta)$. The transformation relation and the Jacobian read

$$\begin{aligned} z &= f(\lambda,\varphi,\tilde{\zeta}) \\ \tilde{\zeta} &= m(\zeta) \\ \sqrt{G} &= \frac{\partial z}{\partial\tilde{\zeta}}\frac{\partial\tilde{\zeta}}{\partial\zeta}. \end{aligned} \tag{2.124}$$

Since m can be any monotonic function, we define this function to map (by its inverse) the coordinate $\tilde{\zeta}$ to the index space used for the vertical discretization (see Section 4) with top-down increasing indices. Thus, this second step of the transformation defines the computational coordinate ζ in a unique way and additionally accommodates any user-specified grid-stretching in the $\tilde{\zeta}$-space. With respect to the discretization, the two-step transformation can be interpreted as a mapping of the irregular curvilinear grid associated with the terrain-following coordinate $\tilde{\zeta}$ in physical space onto a regular rectangular grid labeled by integers. These are the discrete values of the computational coordinate ζ with a vertical grid spacing of one. Figure 2.8 illustrates this two-step transformation.

LM offers three options for the terrain-following coordinate $\tilde{\zeta}$. The first one is a reference-pressure based coordinate denoted by η, the second one is a height based coordinate denoted

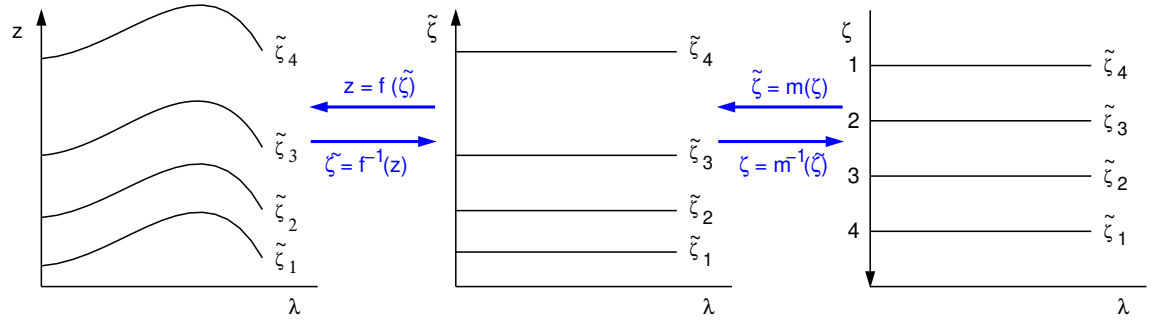

Figure 2.8: Mapping of an irregular curvilinear grid associated with the terrain following coordinate $\tilde{\zeta}$ onto a rectangular equidistant grid ζ labeled by integers.

by μ, and the third one is the height-based SLEVE (Smooth Level VErtical) coordinate according to Schär et al. (2002). Obviously, as base-state pressure and height are uniquely related by the hydrostatic equation, η represents also a terrain-following height coordinate, but the η-formulation might be more familiar to many meteorologist for traditional reasons.

Both vertical coordinates $\tilde{\zeta} = \eta$ and $\tilde{\zeta} = \mu$ are formulated to define a hybrid coordinate system with terrain-following coordinate lines between the surface terrain-height denoted by $h(\lambda, \varphi)$ and a height $z = z_F$ where the coordinate lines change back to flat horizontal lines. Between z_F and the top-height z_T of the model domain, all $\tilde{\zeta}$ coordinate lines correspond to lines of constant height in the original z-system. The reference pressures corresponding to $h(\lambda, \varphi)$, z_F and z_T are denoted by $p_0^s(\lambda, \varphi)$, p_F and p_T, respectively. These values can be calculated from Eq. (2.85).

(a) Pressure-based hybrid coordinate η

The reverse transformation for the pressure-based coordinate η is generally formulated as

$$p_0(\lambda, \varphi, \eta) \;=\; A(\eta) + B(\eta)\, p_0^s(\lambda, \varphi)\,. \tag{2.125}$$

The corresponding z-values result from Eq. (2.85), which finally defines the transformation function f in (2.124). η is chosen to be a normalized coordinate in p-direction running from the top of the model domain to the bottom. $\eta = \eta_T = p_T/p_{SL}$ corresponds to the top, where $p_{SL} = 1000$ hPa denotes a mean sea level pressure, and $\eta = 1$ corresponds to the surface. The mapping functions $A(\eta)$ and $B(\eta)$ are defined for a hybrid system according to

$$\begin{aligned} A(\eta) &= \begin{cases} p_{SL}\eta & \text{if } \eta_T \le \eta \le \eta_F\ , \\ p_F(1-\eta)/(1-\eta_F) & \text{if } \eta_F < \eta \le 1\ , \end{cases} \\ B(\eta) &= \begin{cases} 0 & \text{if } \eta_T \le \eta \le \eta_F\ , \\ (\eta-\eta_F)/(1-\eta_F) & \text{if } \eta_F < \eta \le 1\ . \end{cases} \end{aligned} \tag{2.126}$$

$\eta_F = p_F/p_{SL}$ refers to the interfacial height z_F separating the terrain-following part of the domain from the part which corresponds to the z-system. By default, p_F is set to 220 hPa and the top of the model domain is positioned at $p_T = 20$ hPa, i.e. $\eta_F = 0.220$ and $\eta_T = 0.020$. Other values or even other functional forms of the mapping functions $A(\eta)$ and $B(\eta)$ may be specified by the user.

The special case of a purely terrain-following, i.e. nonhybrid coordinate system is also included by the definition (2.125). Setting $\eta_F = \eta_T = 0$ in (2.126) yields the classical *Sigma*-coordinate in the form $p_0 = \sigma p_0^s$ with $\sigma = \eta$. However, for reasons discussed in Section 3.4, a domain with infinite height is not accommodated by the model. For a domain with finite top-height at η_T, a nonhybrid terrain-following system is obtained by setting $\eta_F = \eta_T$ in (2.126). This results in the transformation

$$p_0(\lambda, \varphi, \eta) \;=\; \frac{\eta - \eta_T}{1 - \eta_T} \left\{ p_0^s(\lambda, \varphi) - p_T \right\} + p_T \,.$$

Thus, by an additional transformation from η to a *Sigma*-coordinate σ_r according to

$$\sigma_r \;=\; \frac{\eta - \eta_T}{1 - \eta_T} \,,$$

which maps the η-interval $[\,\eta_T, 1\,]$ onto the σ_r-interval $[\,0, 1\,]$, the definition of the transformation relation $\sigma_r \equiv (p_0 - p_T)/(p_0^s - p_T)$ for the σ_r-coordinate as proposed by Dudhia (1993)) is revealed.

(b) Height-based hybrid coordinate μ

The μ-coordinate is based on the *Gal-Chen* coordinate but modified to allow for a hybrid system with an interfacial height z_F. μ is chosen to be a nonnormalized coordinate in z-direction, i.e. running from the surface terrain height where $\mu = 0$ to the top of the model domain where $\mu = \mu_T = z_T$. The μ-value for the interfacial height is $\mu_F = z_F$. The reverse transformation for μ is defined analogously to (2.125):

$$z(\lambda, \varphi, \mu) \;=\; a(\mu) + b(\mu)\, h(\lambda, \varphi) \,. \tag{2.127}$$

The mapping functions $a(\mu)$ and $b(\mu)$ are given by

$$\begin{aligned} a(\mu) \;&=\; \mu \,, \\ b(\mu) \;&=\; \begin{cases} 0 & \text{if } \mu_F \le \mu \le \mu_T \;, \\ (\mu_F - \mu)/\mu_F & \text{if } 0 \le \mu < \mu_F \;. \end{cases} \end{aligned} \tag{2.128}$$

The traditional *Gal-Chen* coordinate is obtained by setting z_F to the height z_T of the model domain, i.e. $\mu_F = \mu_T$. In contrast to the η-coordinate, the transformation (2.127) defines a right-handed coordinate system. As the computational coordinate ζ refers to a left-handed terrain-following system, the corresponding transformation becomes part of the mapping function $\mu = m(\zeta)$.

(c) Height-based hybrid SLEVE coordinate

Schär et al. (2002) have recently suggested a new terrain-following coordinate formulation (SLEVE: smooth level vertical) which produces a smooth computational mesh at mid and upper levels. Unlike traditional formulations, e.g. the well known pressure based Sigma coordinate η or the height based Gal-Chen coordinate μ, the new SLEVE coordinate transformation is characterized by a scale dependent, exponential vertical decay of the terrain structure. This allows for a fast decay of small-scale topography components, leading to a fast transition from terrain-following to smooth levels. The hybrid version of the SLEVE coordinate μ_s has been implemented and tested by Daniel Leuenberger from MeteoSwiss Leuenberger (2002).

Similar to the definition of the Gal-Chen coordinate, the height-based SLEVE coordinate μ_s is a non-normalized coordinate taking the values $\mu_s = 0$ at the terrain surface

h and $\mu_s = \mu_T = z_T$ at the model top. The inverse transformation for the SLEVE coordinate is given by

$$z(\lambda, \varphi, \mu_s) = a(\mu_s) + b_1(\mu_s) h_1(\lambda, \varphi) + b_2(\mu_s) h_2(\lambda, \varphi)\,, \tag{2.129}$$

where h_1 and h_2 denote the large-scale and the small-scale components of the topography $h(\lambda, \varphi)$. They satisfy the relation

$$h(\lambda, \varphi) = h_1(\lambda, \varphi) + h_2(\lambda, \varphi)\,. \tag{2.130}$$

The mapping functions $a(\mu_s)$ and $b_i(\mu_s)$ of a *hybrid version* of the SLEVE coordinate are given by

$$\begin{aligned} a(\mu_s) &= \mu_s\,, \\ b_i(\mu_s) &= \begin{cases} 0 & \text{if } \mu_F \le \mu_s \le \mu_T\,, \\ \dfrac{\sinh\{(\mu_F - \mu_s)/s_i\}}{\sinh(\mu_F/s_i)} & \text{if } 0 \le \mu_s < \mu_F\,, \end{cases} \end{aligned} \tag{2.131}$$

where the subscript $i = 1$ refers again to the large-scale part and $i = 2$ to the small-scale part of the topography. The decay constants s_i define the vertical decay rate of the respective topography component, i.e. at a height s_i the contribution of h_i to the level height has fallen to a factor of $1/e$ of the value at the surface $z = h$. $\mu_F = z_F$ denotes the height, where the terrain-following surfaces change to horizontal z-surfaces. The *original* SLEVE formulation is obtained by setting z_F to the height z_T of the model domain, i.e. $\mu_F = \mu_T$. It can be noted that the use of the hybrid SLEVE formulation mitigates the problem of the discontinuity in the determinant of the inverse Jacobian matrix at $\mu_s = \mu_F$ since $|\frac{\partial b_i}{\partial \mu_s}|_{\mu_s = \mu_F}$ is smaller compared with the Gal-Chen formulation.

Since any vertical coordinate transformation must be unique, the function (2.129) must be strictly monotone. A sufficient (but not necessary) condition can be expressed as

$$\frac{\partial z}{\partial \mu_s} \ge \gamma > 0\,, \tag{2.132}$$

where

$$\gamma = 1 - \frac{h_{1,max}}{s_1} \coth\left(\frac{\mu_F}{s_1}\right) - \frac{h_{2,max}}{s_2} \coth\left(\frac{\mu_F}{s_1}\right)\,. \tag{2.133}$$

Here, $h_{i,max}$ denote the maxima of the topography parts in the computational domain. From this condition it becomes clear that the choice of the decay rates s_1 and s_2 is not free but depends on the topography, the decomposition filter and the inter-facial height μ_F. In practice it is desirable that the *small-scale* component decays as fast as possible because these structures cause the most serious transformation errors. Therefore s_2 must be chosen as small as possible. Since the well resolved topography structures are much less responsible for errors, s_1 may be set to a larger value than s_2 resulting in slowly varying levels even in the upper part of the domain.

In the current LM implementation, a digital filter based on a 9-point averaging operator is used to calculate the large-scale part h_1 of the topography. The small-scale part h_2 is then defined as the difference to the original topography. For a 7 km grid spacing, 100 filter iterations are used to determine the large-scale part of the topography, s_1 and s_2 are set to 8000 m and 5000 m, respectively.

Figure 2.9 compares the position of the η- and μ hybrid coordinate levels over a bell-shaped mountain in physical space for the case of equidistant level increments in both η and μ. A base-state pressure of $p_F = 220hPa$ is specified to indicate the interfacial height z_F separating the terrain-following part of the domain from the z-system. Using the default parameters for the reference atmosphere, the value $z_F \simeq 11360m$ results from Eq. (2.85). For the μ-coordinate, an equidistant level increment results in an equidistant vertical distribution of levels in physical space over flat terrain. Over topography, the distance between model levels is compressed below z_F. This relative reduction of layer thickness over high topography results also in case of the pressure-based η-coordinate. In this case, however, layer thickness increases with height over both mountainous and flat terrain.

Figure 2.10 illustrates the position of the μ- and the μ_s-coordinates in physical space for a bell-shaped mountain structure with a superimposed small-scale variation. In this example, the vertical decay rates have been chosen as $s_1 = 12000m$ for the large-scale part and as $s_2 = 2000m$ for the small-scale part of the topography. A rapid decay of the small-scale variations results in case of the SLEVE-coordinate, and at heights above about $4000m$ only the large-scale structure is dominantly reflected in the coordinate lines. The amplitude of these large-scale structures is also reduced exponentially with height. In contrast, the amplitude of both the small-scale and the large-scale variations decrease only linearly in case of the μ-coordinate, resulting in a noticeable distortion of the coordinate lines up to the interfacial height z_F. Further details on the grid structure and the set-up of the vertical grid are discussed in Section 4.

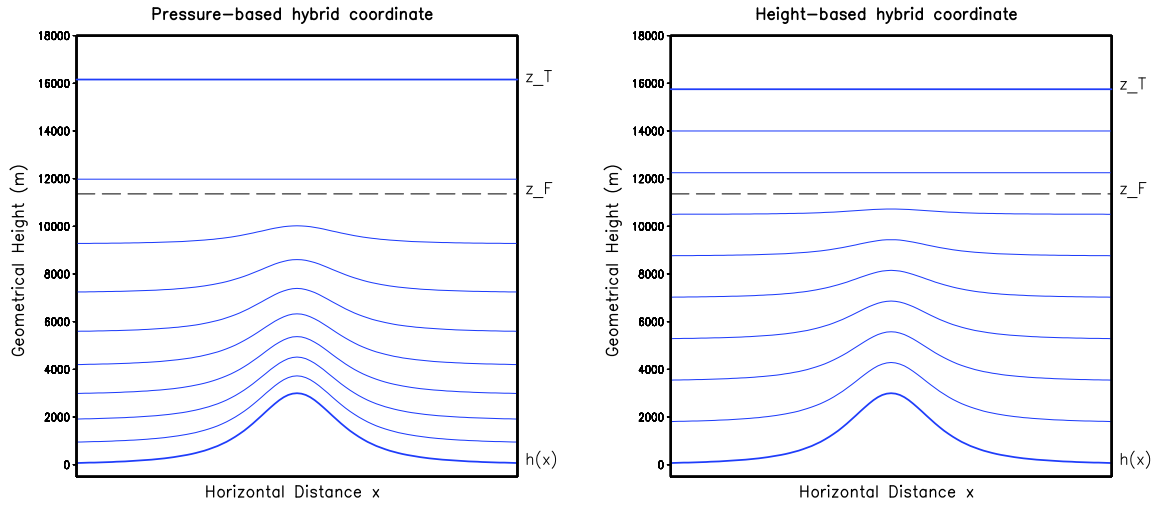

Figure 2.9: Terrain-following coordinate levels for a bell-shaped mountain with $3000m$ height. Left: Geometrical height of 10 η levels starting with $\eta_T = 0.1$ at the top (corresponding to a top height $z_T \simeq 16160m$) to $\eta = 1.0$ at the surface; the surface height is indicated as $h(x)$; the level increment is constant with $\Delta\eta = 0.1$, resulting in a non-equidistant vertical distribution of levels in physical space. Right: Geometrical height of 10 μ-levels starting with $\mu = 0$ at the surface to $\mu_T = z_T = 15750m$ at the top; the level increment is constant with $\Delta\mu = 1750m$, resulting in an equidistant distribution of levels in physical space for flat terrain. The same interfacial height z_F is used for both coordinates.

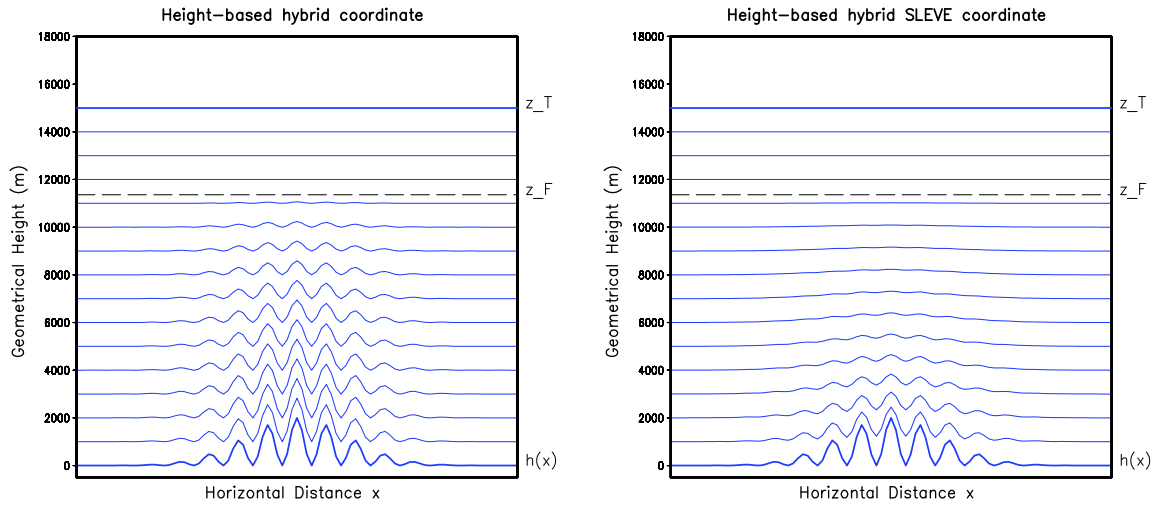

Figure 2.10: Terrain-following hybrid coordinate levels for a bell-shaped mountain with 2000m height and a half-width of 15 horizontal grid intervals, superimposed by a small-scale fluctuation with a 3 grid interval wavelength. The interfacial height is set to $z_F = 11360m$. Left: Geometrical height of 16 μ levels with a constant level increment of $\Delta\mu = 1000m$ from the surface to the top at $\mu_T = z_T = 15000m$. Right: Geometrical height of 16 μ_s levels of the SLEVE coordinate with a constant level increment of $\Delta\mu = 1000m$ from the surface to the top at $\mu_{s,T} = z_T = 15000m$. The vertical decay rates are $s_1 = 12000m$ for the large-scale part and $s_2 = 2000m$ for the small-scale part of the topography.

2.6 The Set of Model Equations

2.6.1 Dynamic Equations in Terrain-Following Coordinates

By applying the transformation relations from the previous section, the basic equations are written in the metrics of the terrain-following ζ-coordinate system. This is most easily done for the budget equations for mass specific scalars as the concentrations of the water constituents. Since these equations have been formulated in advection form, the Lagrangian time derivative (2.113) can directly be used to formulate the advection operator to the ζ-system.

Some difficulties are encountered when transforming the equation of motion. This equation has to be written in vector form as the directional dependence on the orientation of the base vectors has to be taken into account. As clearly pointed out by Sharman et al. (1988), two different decompositions of the equation of motion are possible:

(a) In the first method the equation of motion is decomposed along the coordinate directions of the terrain-following ζ-system by a projection onto the covariant base vectors $\mathbf{a}_i$. This yields prognostic equations for the contravariant velocities $\dot{\lambda}$, $\dot{\varphi}$ and $\dot{\zeta}$. However, the base vectors have also to be differentiated. Thus, the resulting momentum equations involve the Christoffel symbols from tensor analysis and take a rather complex, not very familiar form. The procedure is straightforward and this type of decomposition is described in detail in standard text books (e.g. Pielke (1984)). A major disadvantage of the method is that it is impossible to obtain a fully conservative form for the budget equation of momentum when using either covariant or contravariant velocity

components and base vectors.

(b) In the second method, which was independently proposed by Vinokur (1974) and Viviand (1974), the equation of motion is decomposed along the coordinate directions of the original z-system by a projection onto the orthogonal unit vectors $\mathbf{e}_i$. This yields prognostic equations for the physical velocity components u, v and w of the z-system which, however, are to be evaluated in the terrain-following ζ-system. Thereby, the velocity components become decoupled from the spatial variation of the base vectors $\mathbf{a}_i$. The resulting momentum equations take a simple form which looks very familiar since the physical vertical velocity w is a predicted quantity. Another advantage of this methodology, i.e. returning to the physical velocities u, v and w of the z-system while maintaining the metrics of the terrain-following coordinate system, is that it offers the possibility to derive a conservative form of the equation of motion.

After considering these aspects and related issues, we have chosen to follow the strategy (b), which is used in many other mesoscale models, to formulate the LM model equation in the terrain-following ζ-system. In the present version of the model, the advection form of the equations is preferred for the numerical solution and we thus will not make use of a conservative form of the equations. The advection form allows for an easy implementation of a robust and efficient time-splitting algorithm to solve the equations numerically (see Section 5). Clearly, conservation of mass is not guaranteed, but the related errors are believed to be small – at least for short time range integrations. A numerical scheme which is based on the flux-form of the equations has recently been developed for the WRF-model Klemp et al. (2003). The application of flux-form equations in LM will be considered for a future model version.

The transformation of the momentum equations according to method (b) involves rather lengthy algebraical and analytical calculations using the transformation relations from the previous sections as well as some tensor calculus. These are not reproduced here since we finally end up with the fact that the momentum equation in the ζ-system could have also been obtained by transforming the prognostic equations for the wind components in the z-system as if u, v and w were scalars (which they definitely are not).

Thus, the set of basic equations (2.89) is easily transformed to terrain-following coordinates by applying (2.113) for the total time derivative, (2.115) for the pressure gradient terms and for vertical derivatives, and (2.123) for the wind divergence D. This yields the following set of model equations in the $(\lambda, \varphi, \zeta)$-system:

$$
\begin{aligned}
\frac{\partial u}{\partial t} + \mathbf{v}\cdot\nabla u - \frac{uv}{a}\tan\varphi - fv &= -\frac{1}{\rho a\cos\varphi}\left(\frac{\partial p'}{\partial \lambda} + \frac{J_\lambda}{\sqrt{G}}\frac{\partial p'}{\partial \zeta}\right) + M_u \\
\frac{\partial v}{\partial t} + \mathbf{v}\cdot\nabla v + \frac{u^2}{a}\tan\varphi + fu &= -\frac{1}{\rho a}\left(\frac{\partial p'}{\partial \varphi} + \frac{J_\varphi}{\sqrt{G}}\frac{\partial p'}{\partial \zeta}\right) + M_v \\
\frac{\partial w}{\partial t} + \mathbf{v}\cdot\nabla w &= \frac{1}{\rho\sqrt{G}}\frac{\partial p'}{\partial \zeta} + B + M_w \\
\frac{\partial p'}{\partial t} + \mathbf{v}\cdot\nabla p' - g\rho_0 w &= -(c_{pd}/c_{vd})pD + (c_{pd}/c_{vd} - 1)\rho c_{pd} Q_T \\
\frac{\partial T}{\partial t} + \mathbf{v}\cdot\nabla T &= \frac{1}{\rho c_{pd}}\left(\frac{\partial p'}{\partial t} + \mathbf{v}\cdot\nabla p' - g\rho_0 w\right) + Q_T
\end{aligned}
\tag{2.134}
$$

$$\begin{aligned} \frac{\partial q^v}{\partial t} + \mathbf{v}\cdot\nabla q^v &= -(S^l + S^f) + M_{q^v} \\ \frac{\partial q^{l,f}}{\partial t} + \mathbf{v}\cdot\nabla q^{l,f} + \frac{1}{\rho\sqrt{G}}\frac{\partial P_{l,f}}{\partial \zeta} &= S^{l,f} + M_{q^{l,f}}. \end{aligned}$$

The total density ρ is calculated as a diagnostic variable from the equation of state, (2.55). The advection operator is defined by

$$\mathbf{v}\cdot\nabla = \frac{1}{a\cos\varphi}\left(u\frac{\partial}{\partial\lambda} + v\cos\varphi\frac{\partial}{\partial\varphi}\right) + \dot{\zeta}\frac{\partial}{\partial\zeta} \tag{2.135}$$

where the contravariant vertical velocity $\dot{\zeta}$ is diagnosed from the predicted physical velocity components u, v and w according to Eq. (2.112). The divergence of the wind field, D, results from (2.123) in terms of u, v and w. The formulation of the mixing terms M_ψ and the diabatic heating rate as well as their transformation to the ζ-system will be discussed in subsequent Sections.

The metrics of the terrain-following coordinate system is taken into account by the Jacobian $\sqrt{G}$ and the elements J_λ and J_φ of the inverse Jacobian matrix. For the numerical formulation of the model, these terms will not be evaluated explicitly by analytical differentiation of a specific analytical form $z = f(\lambda, \varphi, \zeta)$ of the transformation, but are provided as three-dimensional fields which are calculated numerically. In this way, the model code becomes independent from a specific choice for the vertical coordinate.

2.6.2 Modifications

In the present version of LM, the set (2.134) of prognostic equations is applied in a slightly different form. The modifications to obtain the model equations are summarized below.

(1) *Formulation of the metric terms*
The metric terms of the ζ-system, $\sqrt{G}$, J_λ and J_φ, are formulated in terms of the base-state variables p_0 and ρ_0. Thus, no additional three-dimensional arrays for these metric terms have to be defined in the model code and some core memory can be saved. Using the hydrostatic stratification of the base state, the metric terms are expanded as follows

$$\begin{aligned} -\sqrt{G} &= \frac{\partial z}{\partial\zeta} = \frac{\partial z}{\partial p_0}\frac{\partial p_0}{\partial\zeta} = -\frac{1}{g\rho_0}\frac{\partial p_0}{\partial\zeta} \\ J_\lambda &= \frac{\partial z}{\partial\lambda} = \frac{\partial z}{\partial p_0}\frac{\partial p_0}{\partial\lambda} = -\frac{1}{g\rho_0}\frac{\partial p_0}{\partial\lambda} \\ J_\varphi &= \frac{\partial z}{\partial\varphi} = \frac{\partial z}{\partial p_0}\frac{\partial p_0}{\partial\varphi} = -\frac{1}{g\rho_0}\frac{\partial p_0}{\partial\varphi}. \end{aligned}$$

We denote the variation of reference pressure with ζ by $\sqrt{\gamma}$ for abbreviation (and to indicate that it is closely related to the Jacobian $\sqrt{G}$),

$$\sqrt{\gamma} \equiv \frac{\partial p_0}{\partial\zeta} \tag{2.136}$$

and rewrite the metric terms in the form

$$\begin{aligned} \sqrt{G} &= \frac{1}{g\rho_0}\sqrt{\gamma} \\ \frac{J_\lambda}{\sqrt{G}} &= -\frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \lambda} \\ \frac{J_\varphi}{\sqrt{G}} &= -\frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \varphi}. \end{aligned} \tag{2.137}$$

(2) *Advection of horizontal momentum*
The terms of horizontal advection in the equations for u and v are combined with the metric term involving $\tan\varphi$ and with the Coriolis term and formulated in terms of the kinetic energy of horizontal motion, E_h, and the vertical component of absolute vorticity, V_a. E_h and V_a are defined as mass specific quantities,

$$E_h = \frac{1}{2}\left(u^2 + v^2\right) \tag{2.138}$$

$$V_a = \frac{1}{a\cos\varphi}\left\{\frac{\partial v}{\partial \lambda} - \frac{\partial}{\partial \varphi}\left(u\cos\varphi\right)\right\} + f. \tag{2.139}$$

With (2.138) and (2.139) the terms on the left hand side of the u- and v-equations from the set (2.134) may be expanded as

$$\mathbf{v}\cdot\nabla u - \frac{uv}{a}\tan\varphi - fv = \frac{1}{a\cos\varphi}\frac{\partial E_h}{\partial \lambda} - vV_a + \dot{\zeta}\frac{\partial u}{\partial \zeta} \tag{2.140}$$

$$\mathbf{v}\cdot\nabla v + \frac{u^2}{a}\tan\varphi + fu = \frac{1}{a}\frac{\partial E_h}{\partial \varphi} + uV_a + \dot{\zeta}\frac{\partial v}{\partial \zeta} \tag{2.141}$$

These relations correspond to a form of the equations for horizontal momentum that is often used in hydrostatic models (e.g. the former DWD models EM and DM). They allow to apply a special finite difference scheme which conserves enstrophy in case of strictly horizontal motion. By this scheme, the generation of nonlinear instability from aliasing is suppressed.

(3) *Approximations to the pressure equation*
In the prognostic equation for the pressure perturbation the source term due to diabatic heating will be neglected. The divergence term is usually the dominating term for most meteorological applications on the mesoscale, whereas the diabatic heating term is usually small. This approximation is used in most nonhydrostatic compressible models (e.g. Klemp and Wilhelmson (1978), Dudhia (1993)), Xue et al. (1995)). Thus, the pressure tendency equation from (2.134) takes the more simple form

$$\frac{\partial p'}{\partial t} + \mathbf{v}\cdot\nabla p' - g\rho_0 w = -(c_{pd}/c_{vd})pD \tag{2.142}$$

which is also substituted for the pressure tendency term on the right hand side of the heat equation.

As has been shortly discussed in Section 3.1, this simplification may have a detrimental impact on the behaviour of compression waves and on mass conservation, but the overall physical error is probably small. Dudhia (1993)) has pointed out, that the neglection of the diabatic heating term in the pressure tendency equation might be even advantageous for models with a rigid upper boundary. With this boundary condition, the

atmosphere is not free to expand vertically, as would be expected, and heating within the model domain will cause a larger temperature increase (by the factor c_{pd}/c_{vd}) than in case of free expansion. Thus, omitting the heating term remedies the overheating caused by vertical confinement of the model domain.

The general role of the heating term in the pressure equation is to increase the pressure locally and thereby force expansions which generate gravity and compression waves. Dropping this term is equivalent to neglect the small part of the velocity field that is directly forced by pressure changes due to diabatic heating, while allowing the temperature to change as if free expansion at constant pressure were occurring. This is also consistent with the thermodynamic assumption that the cloud condensation/ evaporation process operates at constant pressure (see Part II of the documentation).

2.6.3 The Final Set of Equations

By applying the formulations (2.137), (2.140) and (2.141) as well as the approximation (2.142) to the set (2.134) of prognostic equations, we finally arrive at the following form of the model equations.

- *Horizontal wind velocity*

$$\begin{aligned}\frac{\partial u}{\partial t} &= -\left\{\frac{1}{a\cos\varphi}\frac{\partial E_h}{\partial \lambda} - vV_a\right\} - \dot{\zeta}\frac{\partial u}{\partial \zeta} \\ &\quad -\frac{1}{\rho a\cos\varphi}\left(\frac{\partial p'}{\partial \lambda} - \frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \lambda}\frac{\partial p'}{\partial \zeta}\right) + M_u\end{aligned} \tag{2.143}$$

$$\begin{aligned}\frac{\partial v}{\partial t} &= -\left\{\frac{1}{a}\frac{\partial E_h}{\partial \varphi} + uV_a\right\} - \dot{\zeta}\frac{\partial v}{\partial \zeta} \\ &\quad -\frac{1}{\rho a}\left(\frac{\partial p'}{\partial \varphi} - \frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \varphi}\frac{\partial p'}{\partial \zeta}\right) + M_v\end{aligned} \tag{2.144}$$

- *Vertical wind velocity*

$$\begin{aligned}\frac{\partial w}{\partial t} &= -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial w}{\partial \lambda} + v\cos\varphi\frac{\partial w}{\partial \varphi}\right)\right\} - \dot{\zeta}\frac{\partial w}{\partial \zeta} \\ &\quad +\frac{g}{\sqrt{\gamma}}\frac{\rho_0}{\rho}\frac{\partial p'}{\partial \zeta} + M_w \\ &\quad +g\frac{\rho_0}{\rho}\left\{\frac{(T-T_0)}{T} - \frac{T_0 p'}{T p_0} + \left(\frac{R_v}{R_d} - 1\right)q^v - q^l - q^f\right\}\end{aligned} \tag{2.145}$$

- *Perturbation pressure*

$$\begin{aligned}\frac{\partial p'}{\partial t} &= -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial p'}{\partial \lambda} + v\cos\varphi\frac{\partial p'}{\partial \varphi}\right)\right\} - \dot{\zeta}\frac{\partial p'}{\partial \zeta} \\ &\quad +g\rho_0 w - \frac{c_{pd}}{c_{vd}}pD\end{aligned} \tag{2.146}$$

- *Temperature*

$$\begin{aligned}\frac{\partial T}{\partial t} &= -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial T}{\partial \lambda}+v\cos\varphi\frac{\partial T}{\partial \varphi}\right)\right\}-\dot{\zeta}\frac{\partial T}{\partial \zeta}\\ &\quad -\frac{1}{\rho c_{vd}}pD+Q_T \end{aligned} \tag{2.147}$$

- *Water vapour*

$$\begin{aligned}\frac{\partial q^v}{\partial t} &= -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial q^v}{\partial \lambda}+v\cos\varphi\frac{\partial q^v}{\partial \varphi}\right)\right\}-\dot{\zeta}\frac{\partial q^v}{\partial \zeta}\\ &\quad -(S^l+S^f)+M_{q^v} \end{aligned} \tag{2.148}$$

- *Liquid and solid forms of water*

$$\begin{aligned}\frac{\partial q^{l,f}}{\partial t} &= -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial q^{l,f}}{\partial \lambda}+v\cos\varphi\frac{\partial q^{l,f}}{\partial \varphi}\right)\right\}-\dot{\zeta}\frac{\partial q^{l,f}}{\partial \zeta}\\ &\quad -\frac{g}{\sqrt{\gamma}}\frac{\rho_0}{\rho}\frac{\partial P_{l,f}}{\partial \zeta}+S^{l,f}+M_{q^{l,f}} \end{aligned} \tag{2.149}$$

- *Total density of air*

$$\rho = p\{R_d(1+(R_v/R_d-1)q^v-q^l-q^f)T\}^{-1} \tag{2.150}$$

The modified Jacobian $\sqrt{\gamma}$, the kinetic energy E_h of horizontal motion and the absolute vorticity V_a are defined by (2.136), (2.138) and (2.139), respectively. The contravariant vertical velocity $\dot{\zeta}$ and the divergence of the wind field D are diagnostic variables within this set of equations. They are calculated from

$$\dot{\zeta} = -\frac{1}{\sqrt{\gamma}}\left(\frac{u}{a\cos\varphi}\frac{\partial p_0}{\partial \lambda}+\frac{v}{a}\frac{\partial p_0}{\partial \varphi}+g\rho_0 w\right), \tag{2.151}$$

and

$$\begin{aligned} D &= \frac{1}{a\cos\varphi}\left\{\frac{\partial u}{\partial \lambda}-\frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \lambda}\frac{\partial u}{\partial \zeta}+\frac{\partial}{\partial \varphi}(v\cos\varphi)\right.\\ &\quad \left.-\frac{\cos\varphi}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \varphi}\frac{\partial v}{\partial \zeta}\right\}-\frac{g\rho_0}{\sqrt{\gamma}}\frac{\partial w}{\partial \zeta}\,. \end{aligned} \tag{2.152}$$

The equations (2.143) - (2.150) form a complete set to predict the thermodynamic variables of state, i.e. the model variables u, v, w, T, p', ρ, q^v, q^l and q^f, provided that the various mixing terms M_ψ, the cloud microphysical source and sink terms S^l and S^f as well as the associated precipitation fluxes P^l and P^f, and the radiative heating term Q_r are known. The latter is part of the total diabatic heating according to (2.72),

$$Q_T = \frac{L_V}{c_{pd}}S^l+\frac{L_S}{c_{pd}}S^f+M_T+Q_r\,.$$

The calculation of these terms as functions of the model variables is done by corresponding parameterization schemes. Part II of the documentation describes the parameterization of shortwave and longwave electromagnetic radiation as well as the treatment of grid scale clouds and precipitation.

The mixing terms M_ψ, as defined by (2.69), represent the impact from subgrid scale transport processes. Clearly, the scale of these nonresolvable processes depends on the domain for averaging, i.e. on the grid spacing that is used to solve the model equations by finite differencing. The larger the grid spacing becomes, the more organized nonturbulent processes are to be included in the M_ψ-terms. For applications on the meso-β scale, moist convection is the dominating subgrid scale transport process that has to be parameterized with an appropriate independent scheme.

Thus, we split the mixing terms M_ψ into two parts: One describes the impact of small scale turbulent diffusion and the other describes the impact of organized moist convection. The discretization and calculation of the turbulent mixing terms is discussed in Section 4.3.3 and the parameterization of moist convection is documented in Part II.

Apart from these physical subgrid scale processes, the model provides some additional schemes for numerical smoothing and for relaxation and mixing in the lateral and upper boundary zones. For simplicity, we include these terms formally also in M_ψ. Thus, the mixing term M_ψ for a prognostic model variable ψ is decomposed as follows:

$$M_\psi = M_\psi^{TD} + M_\psi^{MC} + M_\psi^{LB} + M_\psi^{CM} + M_\psi^{RD}, \tag{2.153}$$

where the individual terms have the following meaning:

M_ψ^{TD}	tendency due to small scale turbulent mixing,
M_ψ^{MC}	tendency due to subgrid scale moist convection,
M_ψ^{LB}	lateral boundary relaxation term for one-way nesting of the model,
M_ψ^{CM}	source term representing computational mixing,
M_ψ^{RD}	tendency of ψ due to a Rayleigh damping scheme applied within the upper boundary.

The relaxation scheme for one way-nesting and the calculation of the corresponding tendency M_ψ^{LB} is described in Section 5. Various schemes for numerical smoothing and the Rayleigh damping scheme are discussed in Section 6, including the numerical treatment of the mixing terms M_ψ^{CM} and M_ψ^{RD}.

The default set-up for the model physics uses a diagnostic turbulence closure. That is, the diffusion coefficients K_m^v (for momentum) and K_h^v (for heat and moisture), which couple the vertical turbulent fluxes with the corresponding vertical gradients via flux-gradient relations, are determined in terms of wind shear and thermal stability, i.e. in terms of the predicted grid-scale model variables. Meanwhile, another option for a new LM turbulence parameterization is available. This scheme is based on the second-order equations and utilizes a prognostic equation for the mean sub-grid scale turbulent kinetic energy e_t. The diffusion coefficients are directly related to the predicted e_t, modulated by appropriate thermal stability functions. By applying the new scheme, the set of model equations (2.143) - (2.150) is extended by an additional prognostic equation for e_t in the form:

- *Turbulent Kinetic Energy*

$$\frac{\partial e_t}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial e_t}{\partial \lambda} + v\cos\varphi\frac{\partial e_t}{\partial \varphi}\right)\right\} - \dot{\zeta}\frac{\partial e_t}{\partial \zeta} + K_m^v \frac{g\rho_0}{\sqrt{\gamma}}\left\{\left(\frac{\partial u}{\partial \zeta}\right)^2 + \left(\frac{\partial v}{\partial \zeta}\right)^2\right\} + \frac{g}{\rho\theta_v}F^{\theta_v} - \frac{\sqrt{2}e_t^{3/2}}{\alpha_M l} + M_{e_t} \quad (2.154)$$

Here, l denotes a turbulent length scale, F^{θ_v} is the buoyant heat flux, α_M is a dissipation constant and M_{e_t} denotes the mixing of turbulent kinetic energy. The calculation of these terms, the algorithms to determine the diffusion coefficients K_m^v and K_k^v, and the numerical techniques to solve the e_t-equation (2.154) are described in Part II of the LM Documentation.

The numerical technique to solve the model equations (2.143) - (2.150) is by finite differencing. The model grid, the spatial discretization of the equations and the time integration scheme are discussed in the next section.

Section 3

Discretized Form of the Model Equations

The finite difference method is applied to solve the continuous model equations derived in the previous section. The following sections describe the model grid, the discretization of the prognostic equations in space and time, and the numerical algorithms to integrate the resulting finite difference equations.

3.1 Model Grid Structure

3.1.1 Grid Definition and Staggering

The model equations (2.143) - (2.150) have been formulated in a terrain-following coordinate system using a generalized vertical coordinate ζ. This general form of the transformation is employed to map the irregular grid associated with the terrain-following system in physical space onto a rectangular as well as regular computational grid. Thus, constant increments

$$\begin{aligned} \Delta\lambda &: \text{grid-spacing in } \lambda\text{-direction,} \\ \Delta\varphi &: \text{grid-spacing in } \varphi\text{-direction,} \\ \Delta\zeta &: \text{grid-spacing in } \zeta\text{-direction,} \end{aligned}$$

of the independent variables are used to set up the computational grid. The mapping function m from Eq. (2.124) for the two-step coordinate transformation is used to map any user-specified terrain-following coordinate to the computational coordinate ζ. To simplify the notation, we set the vertical grid-spacing equal to one:

$$\Delta\zeta = 1\,. \tag{3.1}$$

The computational $(\lambda, \varphi, \zeta)$-space is then represented by a finite number of grid points (i, j, k), where i corresponds to the λ-direction, j to the φ-direction and k to the ζ-direction. The position of the grid points in the computational space is defined by

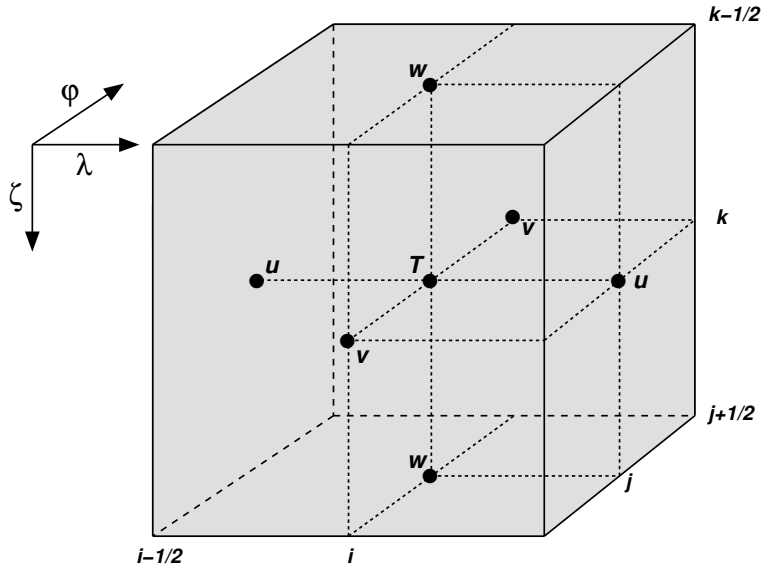

Figure 3.1: A grid box volume $\Delta V = \Delta x \Delta\lambda \Delta\varphi$ showing the Arakawa-C/Lorenz staggering of the dependent model variables.

$$\begin{aligned} \lambda_i &= \lambda_0 + (i-1)\,\Delta\lambda\,, & i &= 1, \cdots, N_\lambda \\ \varphi_j &= \varphi_0 + (j-1)\Delta\varphi, & j &= 1, \cdots, N_\varphi \\ \zeta_k &= k\,, & k &= 1, \cdots, N_\zeta\,. \end{aligned} \tag{3.2}$$

N_λ denotes the number of grid points in λ-direction, N_φ the number of points in the φ-direction and N_ζ the number of points in the ζ-direction. λ_0 and φ_0 define the south-western corner of the model domain with respect to the rotated geographical coordinates (λ, φ). Thus, $i = 1$ and $i = N_\lambda$ correspond, respectively, to the western and the eastern boundaries of the domain. Accordingly, the southern and the northern borderlines are given by $j = 1$ and $j = N_\varphi$.

Every grid point (i, j, k) represents the centre of an elementary rectangular grid volume with side lengths $\Delta\lambda$, $\Delta\varphi$ and $\Delta\zeta$. The grid-box faces are located halfway between the grid points in the corresponding directions, i.e. at $\lambda_{i\pm1/2}$, $\varphi_{j\pm1/2}$ and $\zeta_{k\pm1/2}$. The grid-box faces in vertical direction are usually refereed to as the half levels. These interfacial levels separate the model layers from each other. The model layers labeled by integers k are also denoted as main levels.

The top boundary of the model domain is defined to be the half level ($\zeta = 1/2$) above the uppermost model layer ($\zeta = 1$). At the lower boundary, the ζ-coordinate surface becomes conformal to the terrain height. The half level ($\zeta = N_\zeta + 1/2$) below the first model layer above the ground ($\zeta = N_\zeta$) defines the lower boundary of the model.

The model variables are staggered on an Arakawa-C/Lorenz grid with scalars defined at the centre of a grid box and the normal velocity components defined on the corresponding box faces (see Figure 3.1). This spatial arrangement of the prognostic and diagnostic variables is indicated by subscripts referring to the position within the grid:

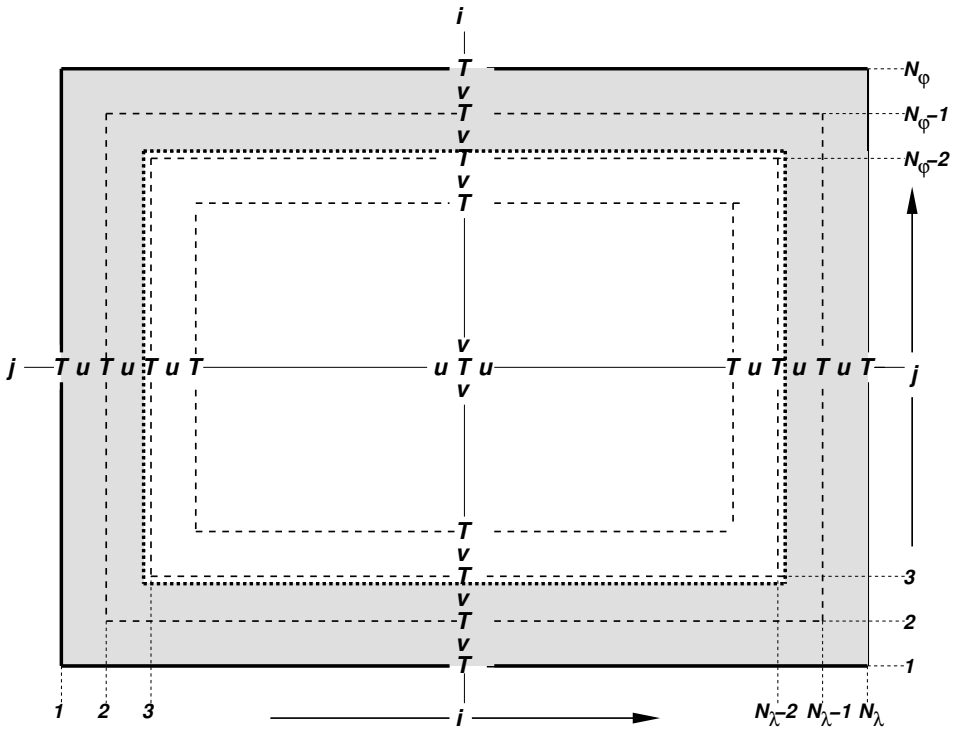

Figure 3.2: Horizontal model domain for $N_\lambda \times N_\varphi$ grid points and an offset of $N_{off} = 2$ for the position of the physical boundaries (dotted). The computational boundaries are shaded; the integration is done for variables in the interior computational domain ($i = 3, \cdots, N_\lambda - 2$ and $j = 3, \cdots, N_\varphi - 2$).

$$\begin{aligned} \psi_{i,j,k} &= \psi(\lambda_i,\, \varphi_j,\, \zeta_k) \qquad \text{for} \quad \psi = T, p', \rho, q^v, q^l, q^f\,, \\ u_{i\pm1/2,j,k} &= u(\lambda_i \pm \Delta\lambda/2,\, \varphi_j,\, \zeta_k)\,, \\ v_{i,j\pm1/2,k} &= v(\lambda_i,\, \varphi_j \pm \Delta\varphi/2,\, \zeta_k)\,, \\ w_{i,j,k\pm1/2} &= w(\lambda_i,\, \varphi_j,\, \zeta_k \pm 1/2)\,. \end{aligned} \tag{3.3}$$

In order to implement boundary conditions and to apply the domain decomposition strategy for code parallelization in a convenient way, the horizontal extent of the computational domain is chosen to be smaller than the total domain size. The lateral physical boundaries are positioned with a spatial offset from the outer boundaries to the interior. This offset is

$$\begin{aligned} N_{off}\Delta\lambda - \Delta\lambda/4 \qquad & \text{in } \lambda\text{-direction and} \\ N_{off}\Delta\varphi - \Delta\varphi/4 \qquad & \text{in } \varphi\text{-direction,} \end{aligned}$$

where N_{off} denotes the number of grid intervals used to define the position of the physical boundaries. By default, N_{off} is set to 2 (larger but not smaller numbers for N_{off} may be specified by the user).

All grid points interior to the physical boundary constitute the computational (or model interior) domain, where the model equations are integrated numerically. These are points with subscripts (i, j) running from $i = N_{off} + 1, \cdots, N_\lambda - N_{off}$ and $j = N_{off} + 1, \cdots, N_\varphi - N_{off}$. The extra points outside the interior domain constitute the computational boundaries. At these points, all model variables are defined and set to specified boundary values, but no

computations are done. For $N_{off} = 2$, we have two extra lines of grid points adjacent to each physical boundary (see Fig. 3.2). On this grid, fourth order computational mixing for all prognostic variables can be readily calculated at all points in the interior domain.

3.1.2 Horizontal Numerical Operators

To represent the governing equations in finite difference form we define the following notations for horizontal averaging and differencing.

$$\begin{aligned} \overline{\psi}^{n\lambda} &= \frac{1}{2}\{\psi(\lambda + n\Delta\lambda/2) + \psi(\lambda - n\Delta\lambda/2)\} \\ \overline{\psi}^{n\varphi} &= \frac{1}{2}\{\psi(\varphi + n\Delta\varphi/2) + \psi(\varphi - n\Delta\varphi/2)\} \\ \delta_{n\lambda}(\psi) &= \frac{1}{n\Delta\lambda}\{\psi(\lambda + n\Delta\lambda/2) - \psi(\lambda - n\Delta\lambda/2)\} \\ \delta_{n\varphi}(\psi) &= \frac{1}{n\Delta\varphi}\{\psi(\varphi + n\Delta\varphi/2) - \psi(\varphi - n\Delta\varphi/2)\} \end{aligned} \tag{3.4}$$

n is an integer (usually $n = 1$) and ψ is a dependent model variable which may be defined at the centre of a grid box or at the u, v or w location on the grid-box faces. Multiple averages will be denoted by

$$\overline{\psi}^{\lambda,\varphi} = \overline{(\overline{\psi}^{\varphi})}^{\lambda} = \overline{(\overline{\psi}^{\lambda})}^{\varphi}.$$

The following rules apply to the horizontal averaging and differencing operators:

$$\begin{aligned} \delta_x \overline{\psi}^{x} &= \overline{(\delta_x \psi)}^{x}, \\ \chi\delta_x\psi &= \delta_x(\overline{\psi}^{x}\chi) - \overline{(\psi\delta_x\chi)}^{x}, \\ \delta_x(\chi\psi) &= \overline{\chi}^{x}\delta_x\psi + \overline{\psi}^{x}\delta_x\chi, \end{aligned} \tag{3.5}$$

where x denotes λ or φ. ψ and χ represent dependent model variables.

3.1.3 Grid Stretching

With respect to vertical differencing and averaging the grid stretching in physical space has to be taken into account. Grid stretching is defined by the mapping function m in the transformation relation (2.124) for the computational coordinate ζ.

We define this mapping in a discrete form using a table, which relates the $N_\zeta + 1$ values of the half-level coordinate $\zeta_{k+1/2}$ to specific values of the terrain-following η- or of the μ- or μ_s-coordinate. This specification is done by the user and a nonequidistant (but of course monotonic) spacing in η, μ or μ_s may be chosen. In order to obtain a reasonable division of the model atmosphere into a given number of layers, N_ζ, it is useful to invert the corresponding transformation relations (2.125), (2.127) or (2.129). The half-level values $\eta_{k+1/2}$, $\mu_{k+1/2}$ or $(\mu_s)_{k+1/2}$ can then easily be specified in terms of base-state pressure or geometrical height with respect to flat terrain. The procedure to set up the base-state variables on the computational grid is as follows.

(a) Pressure-based hybrid coordinate η

In case of the pressure-based coordinate η, first the mapping functions $A(\eta)$ and $B(\eta)$ according to (2.126) are evaluated using the specified half-level values $\eta_{k+1/2}$. Then the reference pressure at the surface is calculated from the terrain height $h(\lambda,\varphi)$ using (2.85).

$$(p_0^s)_{i,j} = \begin{cases} p_{SL} \exp\left\{-(T_{SL}/\beta)\left[1-\sqrt{1-(2\beta g h_{i,j})/(R_d T_{SL}^2)}\right]\right\} & \text{if } \beta \neq 0 \ , \\ p_{SL} \exp\left\{-(g h_{i,j})/(R_d T_{SL})\right\} & \text{if } \beta = 0 \ . \end{cases} \tag{3.6}$$

The reference pressure on half levels may then be computed directly from the definition (2.125) of the transformation:

$$(p_0)_{i,j,k+1/2} = A(\eta_{k+1/2}) + B(\eta_{k+1/2})\,(p_0^s)_{i,j} \ , \qquad k = 0, \cdots, N_\zeta \ , \tag{3.7}$$

where $(p_0)_{1/2} = p_T$ corresponds to the top of the model domain and $(p_0)_{N_\zeta+1/2} = p_0^s$ to the lower boundary. The position of the half levels in physical space is then obtained analytically by inverting (2.85).

$$z_{i,j,k+1/2} = \begin{cases} R_d \ln(p_{SL}/p_{0\,i,j,k+1/2}) \left\{T_{SL} - \beta \ln(p_{SL}/p_{0\,i,j,k+1/2})/2\right\}/g & \text{if } \beta \neq 0 \ , \\ R_d T_{SL} \ln(p_{SL}/p_{0\,i,j,k+1/2})/g & \text{if } \beta = 0 \ . \end{cases} \tag{3.8}$$

(b) Height-based hybrid coordinate μ and SLEVE coordinate μ_s

In case of the modified Gal-Chen coordinate μ, the mapping functions $a(\mu)$ and $b(\mu)$ are first calculated from (2.128) using the specified half-level values $\mu_{k+1/2}$. The height of the half-level coordinate surfaces results directly from the definition (2.127) of the transformation:

$$z_{i,j,k+1/2} = a(\mu_{k+1/2}) + b(\mu_{k+1/2})\, h_{i,j} \ , \qquad k = 0, \cdots, N_\zeta \ . \tag{3.9}$$

For the SLEVE coordinate μ_s, the mapping functions $a(\mu)$, $b_1(\mu_s)$ and $b_2(\mu_s)$ are first calculated from specified half-level values $(\mu_s)_{k+1/2}$ using the (2.131). The height of the half-level coordinates results directly from the transformation (2.129):

$$z_{i,j,k+1/2} = a(\mu_{s\,k+1/2}) + b_1(\mu_{s\,k+1/2})\, h_{1\,i,j} + b_2(\mu_{s\,k+1/2})\, h_{2\,i,j} \ , \qquad k = 0, \cdots, N_\zeta \ . \tag{3.10}$$

The corresponding half-level base-state pressure is then obtained analytically from (2.85).

$$p_{0\,i,j,k+1/2} = \begin{cases} p_{SL} \exp\left\{-T_{SL}\left[1-\sqrt{1-(2\beta g z_{i,j,k+1/2})/(R_d T_{SL}^2)}\right]/\beta\right\} & \text{if } \beta \neq 0 \ , \\ p_{SL} \exp\left\{-(g z_{i,j,k+1/2})/(R_d T_{SL})\right\} & \text{if } \beta = 0 \ . \end{cases} \tag{3.11}$$

Thus, for all three types of vertical coordinates, the reference pressure and the geometrical height of the model half-levels are obtained analytically from the transformation relation and from the base-state temperature profile. $(p_0)_{k+1/2}$ and $z_{k+1/2}$ are the basic quantities to define the vertical grid structure.

Table 3.1: Values of the computational coordinate ζ (level index) related to a non-equidistant specification of the pressure based hybrid coordinate η at half-levels $k + 1/2$ for a 35-layer (36 half-levels) default set-up of LM. Other vertical level distributions can be specified by the user. For any specified $\eta_{k+1/2}$ distribution, the corresponding values of A and B are calculated by (2.126). Base state pressure p_0 and geometrical height z are then evaluated by (3.7) and (3.8). In the Table, p_0 and z refer to flat topography.

ζ	η	A (hPa)	B	p_0 (hPa)	z (m)
1/2	0.0200	20.0000	0.0000	20.0	23588.50
1 + 1/2	0.0400	40.0000	0.0000	40.0	20780.46
2 + 1/2	0.0650	65.0000	0.0000	65.0	18461.85
3 + 1/2	0.0930	93.0000	0.0000	93.0	16565.38
4 + 1/2	0.1230	123.0000	0.0000	123.0	14975.57
5 + 1/2	0.1540	154.0000	0.0000	154.0	13627.79
6 + 1/2	0.1850	185.0000	0.0000	185.0	12482.03
7 + 1/2	0.2160	216.0000	0.0000	216.0	11481.97
8 + 1/2	0.2480	212.1026	0.0359	248.0	10565.29
9 + 1/2	0.2810	202.7949	0.0782	281.0	9716.17
10 + 1/2	0.3150	193.2051	0.1218	315.0	8923.03
11 + 1/2	0.3510	183.0513	0.1679	351.0	8156.79
12 + 1/2	0.3880	172.6154	0.2154	388.0	7434.32
13 + 1/2	0.4250	162.1795	0.2628	425.0	6767.00
14 + 1/2	0.4620	151.7436	0.3103	462.0	6146.45
15 + 1/2	0.4990	141.3077	0.3577	499.0	5566.14
16 + 1/2	0.5360	130.8718	0.4051	536.0	5020.83
17 + 1/2	0.5730	120.4359	0.4526	573.0	4506.26
18 + 1/2	0.6100	110.0000	0.5000	610.0	4018.92
19 + 1/2	0.6470	99.5641	0.5474	647.0	3555.89
20 + 1/2	0.6830	89.4103	0.5936	683.0	3126.36
21 + 1/2	0.7180	79.5385	0.6385	718.0	2726.74
22 + 1/2	0.7520	69.9487	0.6821	752.0	2354.04
23 + 1/2	0.7840	60.9231	0.7231	784.0	2016.09
24 + 1/2	0.8130	52.7436	0.7603	813.0	1719.78
25 + 1/2	0.8390	45.4103	0.7936	839.0	1461.68
26 + 1/2	0.8620	38.9231	0.8231	862.0	1238.96
27 + 1/2	0.8830	33.0000	0.8500	883.0	1039.98
28 + 1/2	0.9030	27.3590	0.8756	903.0	854.19
29 + 1/2	0.9220	22.0000	0.9000	922.0	680.91
30 + 1/2	0.9400	16.9231	0.9231	940.0	519.53
31 + 1/2	0.9560	12.4103	0.9436	956.0	378.28
32 + 1/2	0.9700	8.4615	0.9615	970.0	256.34
33 + 1/2	0.9820	5.0769	0.9769	982.0	153.00
34 + 1/2	0.9920	2.2564	0.9897	992.0	67.71
35 + 1/2	1.0000	0.0000	1.0000	1000.0	0.00

Table 3.1 shows as an example for the set-up of the vertical grid in case of a default specification of η values for 36 model half-levels. Some other defaults for higher or coarser vertical resolution are also available in the model (for both pressure-based η coordinates and height based μ and μ_s coordinates), but arbitrary coordinate values for different resolutions and

grid stretching can be easily specified by the user. Table 3.2 shows the corresponding pressure and height values at full levels, as well as the depth and the stretching ratio of the layers.

Table 3.2: Values of pressure based hybrid coordinate η at full levels related to the computational coordinate ζ (full-level index) for the 35-layer (36 half-levels) default set-up of LM as specified in Table 3.1. The η values, the base-state pressure p_0 and the geometrical height z at full-levels are obtained as arithmetic means of the corresponding half-level values. The pressure thickness Δp_0, the height thickness Δz, and the stretching ratio $s_t = \Delta z_k / \Delta z_{k+1}$ of the layers are also indicated. p_0, z, Δp_0, Δz and s_t refer to flat topography.

ζ	η	p_0 (hPa)	Δp_0 (hPa)	z (m)	Δz (m)	s_t
1	0.0300	30.0	20.0	22184.48	2808.03	1.21
2	0.0525	52.5	25.0	19621.16	2318.61	1.22
3	0.0790	79.0	28.0	17513.62	1896.47	1.19
4	0.1080	108.0	30.0	15770.48	1589.81	1.18
5	0.1385	138.5	31.0	14301.68	1347.78	1.18
6	0.1695	169.5	31.0	13054.91	1145.76	1.15
7	0.2005	200.5	31.0	11982.00	1000.05	1.09
8	0.2320	232.0	32.0	11023.63	916.68	1.08
9	0.2645	264.5	33.0	10140.73	849.13	1.07
10	0.2980	298.0	34.0	9319.60	793.14	1.04
11	0.3330	333.0	36.0	8539.91	766.24	1.06
12	0.3695	369.5	37.0	7795.56	722.47	1.08
13	0.4065	406.5	37.0	7100.66	667.33	1.08
14	0.4435	443.5	37.0	6456.73	620.54	1.07
15	0.4805	480.5	37.0	5856.30	580.31	1.06
16	0.5175	517.5	37.0	5293.49	545.31	1.06
17	0.5545	554.5	37.0	4763.54	514.57	1.06
18	0.5915	591.5	37.0	4262.59	487.34	1.05
19	0.6285	628.5	37.0	3787.40	463.03	1.08
20	0.6650	665.0	36.0	3341.13	429.53	1.07
21	0.7005	700.5	35.0	2926.55	399.62	1.07
22	0.7350	735.0	34.0	2540.39	372.71	1.10
23	0.7680	768.0	32.0	2185.06	337.95	1.14
24	0.7985	798.5	29.0	1867.93	296.30	1.15
25	0.8260	826.0	26.0	1590.73	258.11	1.16
26	0.8505	850.5	23.0	1350.32	222.72	1.12
27	0.8725	872.5	21.0	1139.47	198.98	1.07
28	0.8930	893.0	20.0	947.08	185.79	1.07
29	0.9125	912.5	19.0	767.55	173.28	1.07
30	0.9310	931.0	18.0	600.22	161.38	1.14
31	0.9480	948.0	16.0	448.91	141.25	1.16
32	0.9630	963.0	14.0	317.31	121.95	1.18
33	0.9760	976.0	12.0	204.67	103.33	1.21
34	0.9870	987.0	10.0	110.35	85.29	1.26
35	0.9960	996.0	8.0	33.85	67.71	-

In order to meet the condition of a hydrostatically balanced base-state stratification using finite vertical differences, the reference density ρ_0 will be defined on main levels, i.e. at the centre position (i, j, k) of a grid box. ρ_0 is calculated from the following finite difference

formulation of the hydrostatic equation,

$$(\rho_0)_k \;=\; \frac{1}{g}\frac{(p_0)_{k+1/2}-(p_0)_{k-1/2}}{z_{k-1/2}-z_{k+1/2}}\,, \tag{3.12}$$

where the subscripts (i,j) indicating the position on the horizontal grid have been omitted. ρ_0 is interpreted as a constant mean base-state density within the layer k, resulting in a piecewise linear decrease of reference pressure with height.

3.1.4 Vertical Numerical Operators

Using the definitions from above, the vertical position of the main levels in physical space may be defined as the arithmetic mean of both the half-level pressure p_0 and the geometrical height z. Consequently, for vertical averaging and differencing at main levels, the centred operators

$$\begin{aligned} \overline{(\psi)_k}^{\,\zeta} &= \frac{1}{2}(\,\psi_{k+1/2}+\psi_{k-1/2}) \\ \delta_\zeta(\psi)_k &= \psi_{k+1/2}-\psi_{k-1/2} \end{aligned} \tag{3.13}$$

will be applied. Using (3.13) for the definition of the base-state pressure and the height yields

$$\begin{aligned} (p_0)_k &= \overline{(p_0)_k}^{\,\zeta} = \frac{1}{2}\Big(\,(p_0)_{k+1/2}+(p_0)_{k-1/2}\Big)\,, \\ (z)_k &= \overline{(z)_k}^{\,\zeta} = \frac{1}{2}(\,z_{k+1/2}+z_{k-1/2})\,. \end{aligned} \tag{3.14}$$

The hydrostatic equation (3.12) may then be written in the form

$$\delta_\zeta(p_0)_k \;=\; -g(\rho_0)_k\,\delta_\zeta(z)_k\,. \tag{3.15}$$

The base-state temperature is defined at model main levels. T_0 is interpreted as a local value and calculated from the equation of state using the layer average density ρ_0 from (3.12) and the reference pressure p_0 from (3.14) at the centre of the layer:

$$(T_0)_k \;=\; \frac{(p_0)_k}{R_d(\rho_0)_k}\,. \tag{3.16}$$

The Jacobian $\sqrt{G}$ of the terrain-following coordinate transformation and the related quantity $\sqrt{\gamma}$ from (2.141) are evaluated at the centre position (i,j,k) of a grid box. Their discretized form reads

$$\begin{aligned} (\sqrt{G})_k &= -\,\delta_\zeta(z)_k \;=\; z_{k-1/2}-z_{k+1/2}\,, \\ (\sqrt{\gamma})_k &= \delta_\zeta(p_0)_k \;=\; (p_0)_{k+1/2}-(p_0)_{k-1/2}\,. \end{aligned} \tag{3.17}$$

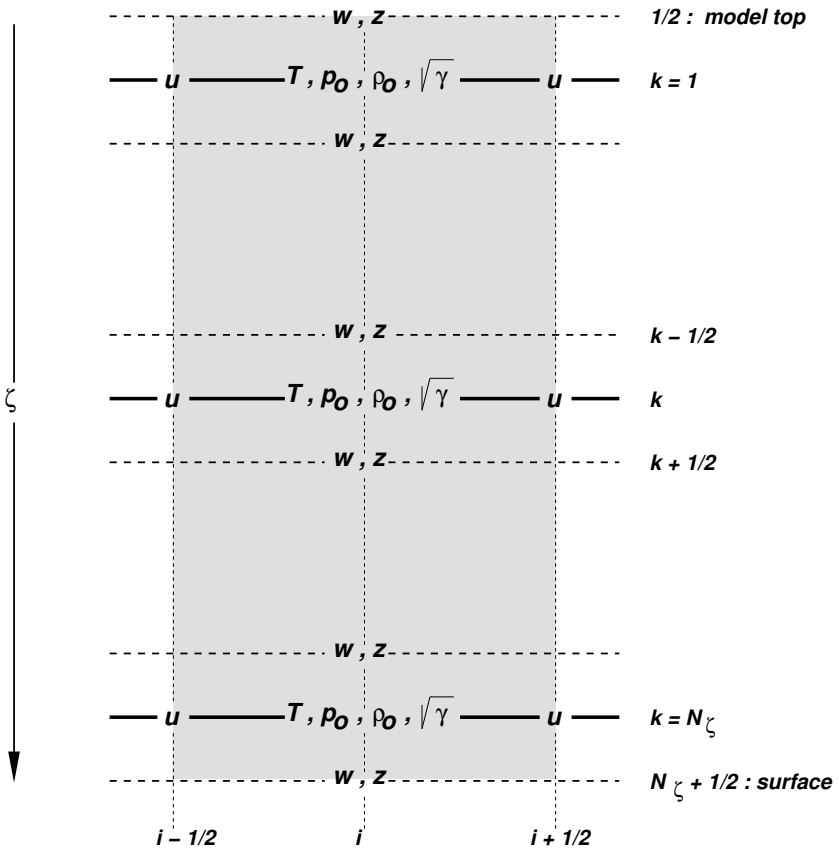

Figure 3.3: Vertical staggering of variables and metric terms in a grid box column with N_ζ layers. Dashed lines are the model half levels separating the main levels (full lines).

$\sqrt{G}$ and $\sqrt{\gamma}$ are related by the equation of state:

$$(\sqrt{G})_k \;=\; \frac{1}{g(\rho_0)_k}(\sqrt{\gamma})_k\,. \tag{3.18}$$

Figure 3.3 shows the vertical staggering of model variables as well as base state variables and metric terms used in the discretization.

The centred operator (3.13) will not be applied to the vertical averaging of main-level variables onto half levels, since, in case of grid stretching, the half levels are noncentred in physical space with respect to the adjacent main levels. Instead of an arithmetic mean, we use the averaging operator

$$\overline{(\psi)}^{\zeta}_{k+1/2} \;=\; \frac{(\sqrt{\gamma})_{k+1}\psi_k + (\sqrt{\gamma})_k\psi_{k+1}}{(\sqrt{\gamma})_{k+1} + (\sqrt{\gamma})_k}\,, \tag{3.19}$$

which corresponds to a mass-weighted linear vertical interpolation of the main-level values ψ_k and ψ_{k+1}.

Vertical differencing at half levels uses the centred operator (3.13) with respect to the equidistant computational coordinate ζ,

$$\delta_\zeta(\psi)_{k+1/2} \;=\; \psi_{k+1} - \psi_k\,, \tag{3.20}$$

but the factor $\sqrt{\gamma}$ occurring in the derivatives with respect to base-state pressure is replaced by the pressure thickness of the half levels:

$$(\sqrt{\gamma})_{k+1/2} = \frac{1}{2}\{(\sqrt{\gamma})_k + (\sqrt{\gamma})_{k+1}\} = (p_0)_{k+1} - (p_0)_k \,. \tag{3.21}$$

Thus, vertical derivatives with respect to base-state pressure are discretized by

$$\left(\frac{1}{\sqrt{\gamma}}\frac{\partial \psi}{\partial \zeta}\right)_k : \quad \frac{1}{(\sqrt{\gamma})_k}\delta_\zeta(\psi)_k = \frac{\psi_{k+1/2} - \psi_{k-1/2}}{(p_0)_{k+1/2} - (p_0)_{k-1/2}} \tag{3.22}$$

on main levels and by

$$\left(\frac{1}{\sqrt{\gamma}}\frac{\partial \psi}{\partial \zeta}\right)_{k+1/2} : \quad \frac{1}{(\sqrt{\gamma})_{k+1/2}}\delta_\zeta(\psi)_{k+1/2} = \frac{\psi_{k+1} - \psi_k}{(p_0)_{k+1} - (p_0)_k} \tag{3.23}$$

on half levels of the model grid. An alternative discretization of vertical derivatives on main levels makes use of a two grid-interval difference operator:

$$\begin{aligned}\left(\frac{1}{\sqrt{\gamma}}\frac{\partial \psi}{\partial \zeta}\right)_k : \quad \frac{1}{(\sqrt{\gamma})_k}\delta_{2\zeta}(\psi)_k &= \frac{\psi_{k+1} - \psi_{k-1}}{(\sqrt{\gamma})_{k+1/2} + (\sqrt{\gamma})_{k-1/2}} \\ &= \frac{\psi_{k+1} - \psi_{k-1}}{(p_0)_{k+1} - (p_0)_{k-1}} \,.\end{aligned} \tag{3.24}$$

The formulation (3.24) will be applied to the metric terms in the prognostic equations for horizontal momentum and the diagnostic equation for horizontal divergence.

3.2 Mode Splitting

Because the governing nonhydrostatic equations describe a compressible model atmosphere, meteorologically unimportant sound waves are also part of the solution. As acoustic waves are very fast, their presence severely limits the time step of explicit time integration schemes. In order to improve the numerical efficiency, the mode-splitting time integration method proposed by Klemp and Wilhelmson (1978) is employed.

This technique is based on a separation of the prognostic equations into terms which are directly related to acoustic wave modes and into terms which refer to comparatively slowly varying modes of motion. The time step for stable explicit integration of the slow modes is then subdivided into a number of small time steps, and the acoustically active terms are updated every small time step while all other terms related to the slow modes are computed only once every big time step. Consequently, only the small time step size is limited by the stability criterion for sound wave propagation. As not the complete but only a reduced set of equations is evaluated on the small time steps, the mode-splitting technique makes the explicit time integration more efficient.

To apply the time-splitting method to the model equations (2.143) - (2.150), the reduced set describing the fast acoustic modes has to be formulated first. Sound wave modes are related to the pressure gradient force terms in the equation for horizontal and vertical wind velocity

and to the divergence term in the pressure tendency equation. The buoyancy term due to pressure perturbations in the w-equation and the vertical advection of base-state pressure are also included in the small time step updating, as these terms have been found to be responsible for certain high frequency oscillations. The moisture equations are assumed to have no high-frequency terms contributing to acoustic waves.

According to a proposal by Skamarock and Klemp (1992), the terms related to vertical gravity wave propagation will also be included in the reduced set. This involves evaluating the thermal buoyancy term in the equation for the vertical velocity and an additional stepping of the heat equation with respect to the divergence term. Usually, the large time interval for the integration of the slow modes is limited by a stability criterion based on advective and gravity wave speeds. For applications on the meso-β scale with horizontal grid-spacings of about 10 km, the gravity wave condition becomes restrictive. Thus, by including the gravity wave modes in the reduced set, the stability criterion is shifted to the small time step. This extension of the mode-splitting integration scheme enhances the efficiency in case of model applications on the meso-β scale significantly.

The reduced set, which becomes subject to a split time integration using small time steps, is revealed by rewriting the model equations (2.143) - (2.147) for the wind components, the perturbation pressure and the temperature in the following form.

- *Horizontal momentum*

$$\frac{\partial u}{\partial t} = -\frac{1}{\rho a \cos\varphi}\left(\frac{\partial p'}{\partial \lambda} - \frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \lambda}\frac{\partial p'}{\partial \zeta}\right) + f_u \tag{3.25}$$

$$\frac{\partial v}{\partial t} = -\frac{1}{\rho a}\left(\frac{\partial p'}{\partial \varphi} - \frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \varphi}\frac{\partial p'}{\partial \zeta}\right) + f_v \tag{3.26}$$

- *Vertical momentum, perturbation pressure and temperature*

$$\frac{\partial w}{\partial t} = \frac{g}{\sqrt{\gamma}}\frac{\rho_0}{\rho}\frac{\partial p'}{\partial \zeta} + g\frac{\rho_0}{\rho}\left\{\frac{(T-T_0)}{T} - \frac{T_0 p'}{T p_0}\right\} + f_w \tag{3.27}$$

$$\frac{\partial p'}{\partial t} = g\rho_0 w - \frac{p c_{pd}}{c_{vd}}\left\{D_h - \frac{g\rho_0}{\sqrt{\gamma}}\frac{\partial w}{\partial \zeta}\right\} + f_{p'} \tag{3.28}$$

$$\frac{\partial T}{\partial t} = -\frac{p}{\rho c_{vd}}\left\{D_h - \frac{g\rho_0}{\sqrt{\gamma}}\frac{\partial w}{\partial \zeta}\right\} + f_T \tag{3.29}$$

Here, the threedimensional wind divergence D from (2.152) has been separated into a contribution D_h due to horizontal divergence and a contribution due to the variation of vertical velocity with height. D_h is given by

$$D_h = \frac{1}{a\cos\varphi}\left\{\frac{\partial u}{\partial \lambda} - \frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \lambda}\frac{\partial u}{\partial \zeta} + \frac{\partial}{\partial \varphi}(v\cos\varphi) - \frac{\cos\varphi}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \varphi}\frac{\partial v}{\partial \zeta}\right\}. \tag{3.30}$$

The f_ψ-terms in (3.25) - (3.29) denote the tendencies due to the slow modes. These are constituted by the advection and Coriolis terms, the mixing terms, the diabatic heating term and the contribution of water substance to the buoyancy term. During the sub-step integration for each big time step the f_ψ-terms are kept constant. The prognostic equations for the water constituents are solved completely on the large time step.

3.3 Discretization and Numerical Integration

3.3.1 The Time Integration Scheme

To illustrate the mode-splitting time integration scheme used in LM, we consider the model equations in the symbolic form

$$\frac{\partial \psi}{\partial t} = s_\psi + f_\psi \,, \tag{3.31}$$

where ψ denotes a prognostic model variable, f_ψ the forcing terms due to the slow modes and s_ψ the source terms which are related to the acoustic and gravity wave modes. The basic time discretization is by large time steps Δt. A superscript n, the time-step counter for the numerical integration, will denote the discrete time level $t = t_0 + n\Delta t$, where t_0 is some initial time for the integration. The numerical time integration uses the well-known leapfrog method, a three-level time differencing scheme. By this method, the variables are stepped forward from time level $t - \Delta t$ to time level $t + \Delta t$ where the right hand side of the corresponding prognostic equations are usually evaluated at the mid-time level t.

We consider first the case of prognostic equations with no acoustically active terms, i.e. the equations for the water constituents where $s_\psi = 0$. The f_ψ-term contains contributions from advection and mixing as well as microphysical source terms. Horizontal advection is treated explicitly and evaluated at the mid-time level n, whereas diffusion evaluated at time level n is unstable with respect to explicit leapfrog integration. Thus, the computational mixing terms are formulated with respect to a simple forward scheme, i.e. they are evaluated at the time level $n-1$. For stability reasons, vertical advection and vertical turbulent diffusion are treated implicitly by a *Crank-Nicolson* scheme involving the time levels $n-1$ and $n+1$. This results in a vertically coupled set of equations which is abbreviated by

$$\frac{\psi^{n+1} - \psi^{n-1}}{2\Delta t} = f_\psi^n = f_\psi(\psi^{n-1}, \psi^n, \psi^{n+1}) \,, \tag{3.32}$$

and solved by Gaussian elimination. Details are discussed in Section 4.3.3.

In case of prognostic equations with acoustically active terms, i.e. the equations for u, v, w, p' and T, the leapfrog interval $2\Delta t$ for the integration from time level $n-1$ to time level $n+1$ is subdivided into a number N_s of small time steps with size $\Delta\tau$:

$$2\Delta t = N_s \Delta\tau \,. \tag{3.33}$$

N_s must be chosen such that the small time step size $\Delta\tau$ satisfies the linear stability condition for sound wave propagation. The prognostic equation (3.31) is then sub-integrated within the leapfrog interval $2\Delta t$ using a two-level time differencing scheme:

$$\frac{\psi^{\nu+1} - \psi^{\nu}}{\Delta\tau} = s_\psi^\nu + f_\psi^n \,. \tag{3.34}$$

The superscript ν indicates the time level for the small time step integration, where $\nu = 0$ corresponds to the initial time level $n-1$ and $\nu + 1 = N_s$ to the final time level $n+1$. The slow-mode forcing f_ψ^n is evaluated only once every big time step and kept constant throughout the small time steps, whereas the fast-mode terms contained in s_ψ^ν are calculated every small step. Figure (3.4) illustrates the basic idea of the Klemp-Wilhelmson time-splitting scheme.

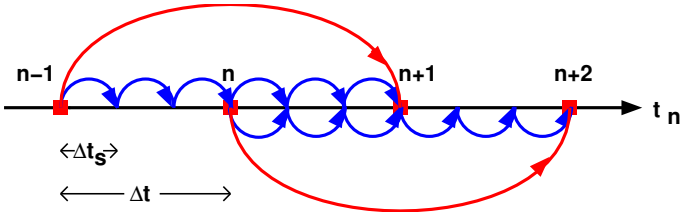

Figure 3.4: The Klemp-Wilhelmson time-splitting algorithm

For stability reasons, the f_ψ^n-term is computed in a similar way as in (3.32), using an implicit scheme for vertical advection and vertical diffusion. However, since the final value of the prognostic variable, ψ^{n+1}, is not known before the small time step integration is completed, the numerical formulation is based on tendency splitting. Thus, f_ψ^n is evaluated from the separately determined tendency due to slow-mode forcing,

$$f_\psi^n \;=\; \frac{\tilde{\psi}^{n+1} - \psi^{n-1}}{2\Delta t} \;=\; f_\psi(\psi^{n-1},\, \psi^n,\, \tilde{\psi}^{n+1}\,)\,, \tag{3.35}$$

where the slow-mode tendency is solved for $\tilde{\psi}^{n+1}$. Details on the calculation of the f-terms will be described in Section 4.3.3.

For the small time step integration there are two options. The first is the traditional fully explicit forward-backward scheme. By this method, the momentum equations (3.25) - (3.27) for u, v and w are first integrated one small time step using a forward scheme relative to the pressure gradient terms. Then the equations (3.28) - (3.29) for pressure perturbation and temperature are integrated forward on the same step using a backward scheme relative to the divergence term, i.e. the threedimensional wind divergence is evaluated with the new updated velocity components.

The other option is a modified *Crank-Nicolson* scheme which is based on time averaging of the vertical velocity and the pressure perturbation in the equations for w and p'. The scheme results in a coupled vertically implicit formulation of the w- and p'-equations. This algorithm is absolutely stable with respect to vertical propagation of acoustic waves. Thus, the small time step size becomes independent from the vertical grid spacing allowing for a much larger size of the small time step when the horizontal to vertical grid spacing ratio is large. This is usually the case for meso-γ and meso-β NWP-applications, where the horizontal grid spacing ranges from 1 km to 15 km and the vertical resolution in the atmospheric boundary layer is on the order of 100 m. For the spatial discretization on the C-grid, the stability condition of the vertically implicit scheme on $\Delta\tau$ reads

$$\Delta\tau \;\leq\; \frac{\Delta s}{\sqrt{2}c_s}\,, \tag{3.36}$$

where Δs is the horizontal grid spacing ($\Delta s = a\Delta\lambda$ for $\Delta\varphi = \Delta\lambda$ and $\cos\varphi \simeq 1$) and c_s is the sound wave speed. We use a constant temperature $T_{sw} = 303$ K to estimate c_s from

$$c_s \;=\; \sqrt{R_d(c_{pd}/c_{vd})T_{sw}}\,. \tag{3.37}$$

With this estimate, a suitable number N_s of small time steps for the integration within the leapfrog interval is derived from (3.33) for any specified large time step size Δt.

3.3.2 Finite Difference Algorithms Related to Fast Modes

The spatial discretization of the acoustic and gravity wave terms in the reduced set (3.25) - (3.29) uses second-order accurate differencing. As mentioned in the previous section, the vertical velocity and the pressure perturbation in the fast-mode forcing functions are subject to time averaging with respect to the small time step interval $\Delta\tau$. Following Ikawa (1988) and Dudhia (1993), the time averaging operator is defined as

$$\overline{\psi}^{\tau} = \frac{1}{2}\,(1+\beta_{sw})\,\psi^{\nu+1} + \frac{1}{2}\,(1-\beta_{sw})\,\psi^{\nu}\,. \tag{3.38}$$

The parameter β_{sw} determines the time-weighting, where $\beta_{sw} = 0$ gives a time-centred average (as used in the original Klemp-Wilhelmson scheme) and positive values $\beta_{sw} > 0$ give a bias towards the future small time step level $\nu+1$. Positive biasing can be used for an efficient damping of vertically propagating acoustic modes.

To simplify the notation, we define

$$\beta^{+} = \frac{1}{2}(1+\beta_{sw}) \qquad \text{and} \qquad \beta^{-} = \frac{1}{2}(1-\beta_{sw}) \;=\; 1-\beta^{+}\,, \tag{3.39}$$

and write the time average of vertical velocity and pressure perturbation in the form

$$\begin{aligned} \overline{w}^{\tau} &= \beta^{+} w^{\nu+1} + \beta^{-} w^{\nu}\,, \\ \overline{p'}^{\tau} &= \beta^{+} p'^{\nu+1} + \beta^{-} p'^{\nu}\,. \end{aligned} \tag{3.40}$$

(a) Finite Difference Equations

The finite difference form of the equations (3.25) - (3.29) related to the fast modes is as follows. Total density, total pressure and temperature as factors of the pressure gradient, buoyancy and divergence terms are kept fixed to their leapfrog mid-time level values during the small time step integration.

- *Horizontal momentum*

$$\frac{u^{\nu+1}-u^{\nu}}{\Delta\tau} = -\frac{1}{\overline{\rho^n}^{\lambda} a\cos\varphi}\left\{\delta_\lambda\, p'^{\nu} - \frac{1}{\overline{(\sqrt{\gamma})}^{\zeta,\lambda}}\delta_\lambda(p_0)\,\delta_{2\zeta}\overline{(p'^{\nu})}^{\lambda}\right\} + f_u^n \tag{3.41}$$

$$\frac{v^{\nu+1}-v^{\nu}}{\Delta\tau} = -\frac{1}{\overline{\rho^n}^{\varphi} a}\left\{\delta_\varphi\, p'^{\nu} - \frac{1}{\overline{(\sqrt{\gamma})}^{\zeta,\varphi}}\delta_\varphi(p_0)\,\delta_{2\zeta}\overline{(p'^{\nu})}^{\varphi}\right\} + f_v^n \tag{3.42}$$

- *Vertical momentum*

$$\begin{aligned} \frac{w^{\nu+1}-w^{\nu}}{\Delta\tau} = {} & \frac{g}{\sqrt{\gamma}}\,\frac{\overline{\rho_0}^{\zeta}}{\overline{\rho^n}^{\zeta}}\left\{\beta^{+}\delta_\zeta(p')^{\nu+1} + \beta^{-}\delta_\zeta(p')^{\nu}\right\} \\ & - g\,\frac{\overline{\rho_0}^{\zeta}}{\overline{\rho^n}^{\zeta}}\left\{\overline{\left(\frac{T_0\beta^{+}}{T^n p_0}\right)(p')^{\nu+1}}^{\zeta} + \overline{\left(\frac{T_0\beta^{-}}{T^n p_0}\right)(p')^{\nu}}^{\zeta}\right\} \\ & + g\,\frac{\overline{\rho_0}^{\zeta}}{\overline{\rho^n}^{\zeta}}\left\{\overline{\left(\frac{T^{\nu}-T_0}{T^n}\right)}^{\zeta}\right\} + f_w^n \end{aligned} \tag{3.43}$$

- *Pressure perturbation*

$$\begin{aligned}\frac{p'^{\nu+1}-p'^{\nu}}{\Delta\tau} &= g\rho_0\left\{\beta^+\left(\overline{w}^{\zeta}\right)^{\nu+1}+\beta^-\left(\overline{w}^{\zeta}\right)^{\nu}\right\}\\ &+\frac{g\rho_0}{\sqrt{\gamma}}\left(\frac{p^n c_{pd}}{c_{vd}}\right)\left\{\beta^+\delta_\zeta(w^{\nu+1})+\beta^-\delta_\zeta(w^{\nu})\right\}\\ &-\left(\frac{p^n c_{pd}}{c_{vd}}\right)D_h^{\nu+1}+f_{p'}^n \end{aligned} \tag{3.44}$$

- *Temperature*

$$\begin{aligned}\frac{T^{\nu+1}-T^{\nu}}{\Delta\tau} &= \frac{g\rho_0}{\sqrt{\gamma}}\left(\frac{p^n}{\rho^n c_{vd}}\right)\left\{\beta^+\delta_\zeta(w^{\nu+1})+\beta^-\delta_\zeta(w^{\nu})\right\}\\ &-\left(\frac{p^n}{\rho^n c_{vd}}\right)D_h^{\nu+1}+f_T^n \end{aligned} \tag{3.45}$$

These finite difference equations are evaluated at the locations of the corresponding variables on the staggered grid. The horizontal wind divergence D_h is defined at the centre (i,j,k) of a grid box and is computed using the updated velocity components u and v from time level $\nu+1$. The discretization is given by

$$\begin{aligned}D_h^{\nu+1} &= \frac{1}{a\cos\varphi}\left\{\delta_\lambda\left(u^{\nu+1}\right)-\overline{\frac{1}{\left(\sqrt{\gamma}\right)^{\zeta,\lambda}}\delta_\lambda(p_0)\,\delta_{2\zeta}(u^{\nu+1})}^{\lambda}\right.\\ &\left.+\delta_\varphi\left(v^{\nu+1}\cos\varphi\right)-\overline{\frac{\cos\varphi}{\left(\sqrt{\gamma}\right)^{\zeta,\varphi}}\delta_\varphi(p_0)\,\delta_{2\zeta}(v^{\nu+1})}^{\varphi}\right\}. \end{aligned} \tag{3.46}$$

In Eqs. (3.43) -(3.45) time averaging is performed on several terms with β^+ and β^- as weighting factors. The vertically explicit option is revealed by setting $\beta^+ = 0$ (and thus $\beta^- = 1$) in the w-equation and $\beta^+ = 1$ (and thus $\beta^- = 0$) in the pressure and the heat equation. In this case, the momentum equations for u, v and w are stepped forward for one small time step $\Delta\tau$ from time level ν to time level $\nu+1$ using the initial pressure values p'^{ν} to calculate the pressure gradient terms. Then the p'- and the T- equations are stepped forward using the updated velocities $(u,v,w)^{\nu+1}$ for the computation of the divergence term.

For the vertically implicit scheme with $\beta_{sw} \geq 0$ the small time step size becomes independent of the vertical resolution of the model, which is important for numerical efficiency. However, Skamarock and Klemp (1992) have shown that this scheme may be unstable to certain acoustic modes due to the interaction of the advection (evaluated on the big time step) and the propagation (evaluated on the small steps) of these waves. The β_{sw}-parameter can be used to damp the unstable modes. Ikawa (1988) has shown that the fully implicit scheme with $\beta_{sw} = 1$ is neutral to horizontally propagating waves but severely damps the vertical modes. Durran and Klemp (1983) and Dudhia (1993) found that β_{sw}-values in the range 0.2 - 0.4 are sufficient to effectively control the vertically unstable acoustic modes in practical applications. The default value used in LM is $\beta_{sw} = 0.4$.

Since the mode-splitting scheme is neutral to horizontally propagating sound waves, an additional numerical technique to control unstable horizontal wave modes must be introduced.

Skamarock and Klemp (1992) proposed threedimensional divergence damping to attenuate acoustic modes. Dudhia (1993) found that this method is equivalent to using a time-extrapolated value for the pressure perturbation in the equations for horizontal momentum. For simplicity reasons, the latter method is applied in LM. Thus, a small term

$$(p'_D)^\nu \;=\; -\,\alpha_{sw}\Delta\tau\frac{p^n c_{pd}}{c_{vd}}\left\{D_h^\nu - \frac{g\rho_0}{\sqrt{\gamma}}\delta_\zeta\left(\overline{w}^\tau\right)^{\nu-1}\right\}, \tag{3.47}$$

which is directly proportional to the threedimensional wind divergence calculated at the previous time step, is added onto the pressure perturbation in Eqs. (3.41) and (3.42) for u and v. The effect of this additional term is to slightly modify the phase between acoustic pressure and horizontal momentum oscillations, such that the wave is caused to decay. Obviously, the artificial term (3.47) tends to zero as the sound wave amplitudes become small. The time-extrapolation factor α_{sw} is set to 0.1 by default.

(b) Method of Solution

For the vertically implicit option, the Eqs. (3.43) and (3.44) for pressure perturbation and vertical velocity become coupled due to time averaging and have to be solved simultaneously for w and p' at the future time step $\nu+1$. The following procedure to solve the coupled set of equations is applied. First, after regrouping the unknown terms and evaluating the vertical averaging and differencing operators, the pressure and the w-equations are rewritten as

$$p_k'^{\,\nu+1} \;=\; p_k'^{\,\nu} + C_k^p + \beta^+ C_k^{p1}\left(w_{k'+1}^{\nu+1} - w_{k'}^{\nu+1}\right) + \beta^+ C_k^{p2}\left(w_{k'+1}^{\nu+1} + w_{k'}^{\nu+1}\right), \tag{3.48}$$

$$\begin{aligned} w_{k'}^{\nu+1} \;=\;& w_{k'}^{\nu} + C_{k'}^w + \beta^+ C_{k'}^{w1}\left(p_k'^{\,\nu+1} - p_{k-1}'^{\,\nu+1}\right) \\ & -\beta^+ C_{k'}^{w2}(\sqrt{\gamma})_{k-1}\left(\frac{T_0}{T^n}\frac{p'^{\nu+1}}{p_0}\right)_k - \beta^+ C_{k'}^{w2}(\sqrt{\gamma})_k\left(\frac{T_0}{T^n}\frac{p'^{\nu+1}}{p_0}\right)_{k-1}, \end{aligned} \tag{3.49}$$

where $k' = k - 1/2$ abbreviates the vertical coordinate index of the upper half-level corresponding to the model layer k. C_k^p and $C_{k'}^w$ denote the known terms on the right hand side of the p'- and the w-equation, respectively:

$$\begin{aligned} C_k^p \;=\;& (f_{p'}^n)_k\,\Delta\tau - \frac{c_{pd}}{c_{vd}}p_k^n\left(D_h^{\nu+1}\right)_k\Delta\tau \\ & +\beta^- C_k^{p1}\left(w_{k'+1}^{\nu} - w_{k'}^{\nu}\right) + \beta^- C_k^{p2}\left(w_{k'+1}^{\nu} + w_{k'}^{\nu}\right), \end{aligned} \tag{3.50}$$

$$\begin{aligned} C_{k'}^w \;=\;& (f_w^n)_{k'}\,\Delta\tau + \beta^- C_{k'}^{w1}\left(p_k'^{\,\nu} - p_{k-1}'^{\,\nu}\right) \\ & -\beta^- C_{k'}^{w2}(\sqrt{\gamma})_{k-1}\left(\frac{T_0}{T^n}\frac{p'^{\nu}}{p_0}\right)_k - \beta^- C_{k'}^{w2}(\sqrt{\gamma})_k\left(\frac{T_0}{T^n}\frac{p'^{\nu}}{p_0}\right)_{k-1} \\ & +C_{k'}^{w2}(\sqrt{\gamma})_{k-1}\left(\frac{T^\nu - T_0}{T^n}\right)_k + C_{k'}^{w2}(\sqrt{\gamma})_k\left(\frac{T^\nu - T_0}{T^n}\right)_{k-1}. \end{aligned} \tag{3.51}$$

The factors C^{p1}, C^{p2}, C^{w1} and C^{w2} result from vertical averaging and differencing of the terms on the right hand side of the p'- and w-equations. They have the following form:

$$C_k^{p1} \;=\; \frac{c_{pd}}{c_{vd}}\frac{g(\rho_0)_k p_k^n}{(\sqrt{\gamma})_k}\Delta\tau,$$

$$
\begin{aligned}
C_k^{p2} &= \frac{g}{2}(\rho_0)_k\,\Delta\tau\,, \\
C_{k'}^{w1} &= \frac{2g\,\Delta\tau}{(\sqrt{\gamma})_{k-1}+(\sqrt{\gamma})_k}\left\{\frac{(\sqrt{\gamma})_{k-1}(\rho_0)_k+(\sqrt{\gamma})_k(\rho_0)_{k-1}}{(\sqrt{\gamma})_{k-1}(\rho^n)_k+(\sqrt{\gamma})_k(\rho^n)_{k-1}}\right\}, \\
C_{k'}^{w2} &= \frac{1}{2}C_{k'}^{w1}\,.
\end{aligned}
\tag{3.52}
$$

Inserting ${p'_k}^{\nu+1}$ from (3.48) into (3.49) eliminates the unknown pressure at the future time step in the w-equation, leaving $w^{\nu+1}$ as the only unknown. After considerable algebra, this yields a linear tridiagonal equation system for $w^{\nu+1}$ which is written in the form

$$
A_{k'}\,w_{k'-1}^{\nu+1}+B_{k'}\,w_{k'}^{\nu+1}+C_{k'}\,w_{k'+1}^{\nu+1}\;=\;D_{k'}\,,
\tag{3.53}
$$

where the matrix diagonals A, B and C, and the inhomogeneous term D are

$$
\begin{aligned}
A_{k'} &= -(\beta^+)^2C_{k'}^{w1}\alpha_{k'}^T\left(C_{k-1}^{p1}-C_{k-1}^{p2}\right), \\
B_{k'} &= 1+(\beta^+)^2C_{k'}^{w1}\left\{\alpha_{k'}^B\left(C_k^{p1}-C_k^{p2}\right)+\alpha_{k'}^T\left(C_{k-1}^{p1}+C_{k-1}^{p2}\right)\right\}, \\
C_{k'} &= -(\beta^+)^2C_{k'}^{w1}\alpha_{k'}^B\left(C_k^{p1}+C_k^{p2}\right), \\
D_{k'} &= w_{k'}^{\nu}+(f_w^n)_{k'}\,\Delta\tau \\
&\quad +C_{k'}^{w2}\left\{(\sqrt{\gamma})_{k-1}\left(\frac{T^\nu-T_0}{T^n}\right)_k+(\sqrt{\gamma})_k\left(\frac{T^\nu-T_0}{T^n}\right)_{k-1}\right\} \\
&\quad +C_{k'}^{w1}\left\{\alpha_{k'}^B\left({p'_k}^{\nu}+\beta^+C_k^p\right)-\alpha_{k'}^T\left({p'_{k-1}}^{\nu}+\beta^+C_{k-1}^p\right)\right\}.
\end{aligned}
\tag{3.54}
$$

The coefficients α^B and α^T in (3.54) are defined as

$$
\begin{aligned}
\alpha_{k'}^B &= 1-\frac{1}{2}(\sqrt{\gamma})_{k-1}\left(\frac{T_0}{T^np_0}\right)_k, \\
\alpha_{k'}^T &= 1+\frac{1}{2}\,(\sqrt{\gamma})_k\,\left(\frac{T_0}{T^np_0}\right)_{k-1}.
\end{aligned}
\tag{3.55}
$$

The matrix equation (3.53) can be solved for $w^{\nu+1}$ using a standard tridiagonal solver, provided that appropriate boundary conditions on the vertical velocity at the upper ($k'=1/2$) and lower ($k'=N_\zeta+1/2$) boundary of the model domain are specified. The solution method used in LM is based on usual Gaussian elimination and back-substitution.

The present version of LM supports only non-penetrative top and bottom boundary conditions. Thus the vertical velocity at the top boundary is set to zero, and w at the lower boundary is calculated from the horizontal velocities u and v at the lowest model layer and the terrain height, ensuring that the flow at the lower boundary follows the surface terrain.

Additionally, top and bottom boundary conditions for the perturbation pressure and the horizontal wind components have to be specified for the evaluation of the metric correction terms in the equation for horizontal momentum and in the diagnostic equation for the horizontal divergence. For u and v, free slip conditions are applied while an extrapolated boundary condition for the perturbation pressure is used. Details on the formulation of the top and bottom boundary conditions are discussed in Section 5.

After $w^{\nu+1}$ has been obtained from the solution of (3.53), it is substituted in (3.48) to yield the pressure perturbation $p'^{\nu+1}$ at the future time step. Also, the heat equation (3.45) can then be stepped to obtain $T^{\nu+1}$:

$$\begin{aligned} T_k^{\nu+1} &= T_k^{\nu} + (f_T^n)_k \, \Delta\tau - \frac{p_k^n}{c_{vd}\rho_k^n} \left(D_h^{\nu+1}\right)_k \Delta\tau \\ &\quad + \frac{C_k^{p1}}{c_{pd}\rho_k^n} \left\{ \beta^- \left(w_{k'+1}^{\nu} - w_{k'}^{\nu}\right) + \beta^+ \left(w_{k'+1}^{\nu+1} - w_{k'}^{\nu+1}\right) \right\} . \end{aligned} \tag{3.56}$$

3.3.3 Finite Difference Algorithms Related to Slow Modes

Prior to the small time step integration described above, the forcing terms f_ψ due to the slow modes have to be evaluated by finite differencing at the centre time level n of the leapfrog scheme. These terms are

$$\begin{aligned} f_u^n &= \left(\frac{\partial u}{\partial t}\right)_{ha} + \left(\frac{\partial u}{\partial t}\right)_{va} + M_u \\ f_v^n &= \left(\frac{\partial v}{\partial t}\right)_{ha} + \left(\frac{\partial v}{\partial t}\right)_{va} + M_v \\ f_w^n &= \left(\frac{\partial w}{\partial t}\right)_{ha} + \left(\frac{\partial w}{\partial t}\right)_{va} + B_q + M_w \\ f_{p'}^n &= \left(\frac{\partial p'}{\partial t}\right)_{ha} + \left(\frac{\partial p'}{\partial t}\right)_{va} \\ f_T^n &= \left(\frac{\partial T}{\partial t}\right)_{ha} + \left(\frac{\partial T}{\partial t}\right)_{va} + \left(\frac{L_V}{c_{pd}} S^l + \frac{L_S}{c_{pd}} S^f\right) + Q_r + M_T \\ f_q^n &= \left(\frac{\partial q}{\partial t}\right)_{ha} + \left(\frac{\partial q}{\partial t}\right)_{va} + (S^q) + M_q \, . \end{aligned} \tag{3.57}$$

The terms $(\partial\psi/\partial t)_{ha}$ and $(\partial\psi/\partial t)_{va}$ denote the contribution of horizontal and, respectively, vertical advection to the slow-mode forcing. As described in Section 3.6, the horizontal advection terms, the metric terms related to the spherical coordinate system and the Coriolis terms in the prognostic equations for u and v have been reformulated to a combined formulation in terms of kinetic energy E_h and absolute vorticity V_a of horizontal motion. The terms $(\partial u/\partial t)_{ha}$ and $(\partial v/\partial t)_{ha}$ in (3.57) represent this combined form. Thus, referring to the model equations (2.143) - (2.150), the differential formulation of the advection terms is

$$\begin{aligned} \left(\frac{\partial \psi}{\partial t}\right)_{ha} &= -\frac{1}{a\cos\varphi}\left(u\frac{\partial \psi}{\partial \lambda} + v\cos\varphi\frac{\partial \psi}{\partial \varphi}\right) \\ \left(\frac{\partial \psi}{\partial t}\right)_{va} &= -\dot{\zeta}\frac{\partial \psi}{\partial \zeta} \end{aligned} \tag{3.58}$$

for all prognostic model variables, except for the u- and v-tendencies due to horizontal advection:

$$\begin{aligned} \left(\frac{\partial u}{\partial t}\right)_{ha} &= -\frac{1}{a\cos\varphi}\frac{\partial E_h}{\partial \lambda} + vV_a \\ \left(\frac{\partial v}{\partial t}\right)_{ha} &= -\frac{1}{a}\frac{\partial E_h}{\partial \varphi} - uV_a\,. \end{aligned} \tag{3.59}$$

The kinetic energy E_h and absolute vorticity V_a of horizontal motion is defined by Eq. (2.139).

The contribution B_q of water substance to the buoyancy in the forcing term f_w for vertical velocity is averaged from the model main levels onto the half levels, where w is defined, by applying the interpolation operator (3.19). This term is calculated explicitly using the concentration of water constituents from the current time level n:

$$B_q = g\frac{\overline{\rho_0}^{\,\zeta}}{\overline{\rho^n}^{\,\zeta}}\left\{\overline{\left(R_v/R_a - 1\right)q^v - q^l - q^f}^{\,\zeta}\right\}^n . \tag{3.60}$$

In Eqs. (3.57) the term f_q represents the slow-mode forcing for the various water constituents. The vertical divergence of the precipitation fluxes has not been included, since, in the present version of the model, the precipitating water phases (rain and snow) are not treated prognostically but are calculated from diagnostic budget equations. Thus, only the nonprecipitating water constituents (water vapour, cloud water and cloud ice) are subject to explicit time integration. These are indicated by the subscript q, and S_q denotes the corresponding microphysical sources and sinks. The parameterization of the S_q-terms and the numerical scheme to solve the diagnostic equations for the precipitating water phases will be discussed in Part II of the LM documentation.

The discretization of the advection terms and the numerical treatment of the other source terms contributing to the slow-mode forcing is discussed below.

(a) Mass-weighted Velocities

With respect to the numerical formulation of the advection terms it is convenient to define mass-weighted velocities according to

$$\begin{aligned} U_k &\equiv \overline{(\sqrt{\gamma})_k}^{\,\lambda} u_k\,, \\ V_k &\equiv \overline{(\sqrt{\gamma})_k}^{\,\varphi} v_k\,, \\ W^c_{k+1/2} &\equiv \overline{(\sqrt{\gamma})_{k+1/2}}^{\,\zeta} (\dot{\zeta})_{k+1/2}\,, \end{aligned} \tag{3.61}$$

where the positions of U, V and W^c on the staggered grid are the same as those declared for u, v and $\dot{\zeta}$. The weighting factor in (3.61) is $\sqrt{\gamma}$, i.e. base-state pressure thickness, which corresponds to the mean reference mass of a model layer. W^c denotes the weighted contravariant vertical velocity $\dot{\zeta}$, which has to be calculated diagnostically from the predicted physical velocities u, v and w according to (2.151). The discretized form of W^c is

$$W^c_{k+1/2} = -\frac{1}{a\cos\varphi\,\overline{(\sqrt{\gamma})}^{\,\zeta}_{k+1/2}}\left\{\overline{\left(\overline{U}^{\,\zeta}\delta_\lambda p_0\right)}^{\,\lambda}_{k+1/2} + \overline{\left(\overline{V\cos\varphi}^{\,\zeta}\delta_\varphi p_0\right)}^{\,\varphi}_{k+1/2}\right\}$$

$$- g \, \overline{(\rho_0)}^{\zeta}_{k+1/2} w_{k+1/2} \, . \tag{3.62}$$

The weighted contravariant vertical velocity is evaluated for the current time level n of the leapfrog scheme using the corresponding values of u, v and w.

As clearly pointed out by Clark (1977), a proper discretization of the diagnostic equation for the contravariant vertical velocity is very important in obtaining a correct kinetic energy budget in an anelastic model. However, we have not yet examined this type of sensitivity with respect to the compressible model equations of LM.

(b) Horizontal Advection

Horizontal advection is treated explicitly, i.e. the corresponding tendencies are evaluated by taking both the advection velocities (u and v) and the transported quantities at the centred time level n of the leapfrog scheme. The spatial discretization uses second-order accurate finite differences.

To calculate the tendency terms (3.59) for horizontal momentum, the kinetic energy E_h and the absolute vorticity V_a have to be specified first. E_h is defined at the centre position (i, j, k) of a grid box and calculated as

$$E_h \;=\; \frac{1}{2}\left\{\overline{(u^2)}^{\,\lambda} + \frac{1}{\cos\varphi}\overline{(v^2\cos\varphi)}^{\,\varphi}\right\} . \tag{3.63}$$

V_a is defined at the corners of an elementary grid area, i.e. at the position $(i+1/2, j+1/2, k)$, and is discretized by

$$V_a \;=\; f + \frac{1}{a\cos\varphi}\left\{\delta_\lambda v + \delta_\varphi(u\cos\varphi)\right\} . \tag{3.64}$$

where the Coriolis parameter f has also been defined at the $(i+1/2, j+1/2, k)$- position on the grid. The finite difference form of the horizontal advection terms for u and v reads

$$\begin{aligned}
\left(\frac{\partial u}{\partial t}\right)^n_{ha} &= -\frac{1}{a\cos\varphi}\delta_\lambda\left(E^n_h\right) + \frac{\overline{(V^n\cos\varphi)}^{\,\lambda,\varphi}}{\cos\varphi}\overline{\left(\frac{V^n_a}{\overline{\sqrt{\gamma}}^{\,\lambda,\varphi}}\right)}^{\,\varphi} , \\
\left(\frac{\partial v}{\partial t}\right)^n_{ha} &= -\frac{1}{a}\delta_\varphi\left(E^n_h\right) - \overline{(U^n)}^{\,\lambda,\varphi}\overline{\left(\frac{V^n_a}{\overline{\sqrt{\gamma}}^{\,\lambda,\varphi}}\right)}^{\,\lambda} .
\end{aligned} \tag{3.65}$$

For prognostic variables ψ defined at the centre (i, j, k) of a grid box, i.e. $\psi = p'$, T or q, the horizontal advection terms are discretized by

$$\left(\frac{\partial \psi}{\partial t}\right)^n_{ha} = -\frac{1}{\sqrt{\gamma}a\cos\varphi}\left\{\overline{U^n(\delta_\lambda\psi^n)}^{\,\lambda} + \overline{V^n\cos\varphi(\delta_\varphi\psi^n)}^{\,\varphi}\right\} . \tag{3.66}$$

As can be easily shown, this formulation is equivalent to a conservative flux form with respect to base-state density ρ_0 (or base-state pressure thickness $\sqrt{\gamma}$) plus a small correction term related to threedimensional divergence. In an anelastic model, this additional correction term would be zero. The formulation of the advective tendencies by a pure flux form using the time dependent density ρ as weighting factor for the velocities would yield a full mass conserving discretization. However, such a scheme may not be practical because acoustic density fluctuations would appear in the advection terms requiring a shorter time step to treat them adequately.

Horizontal advection for the vertical velocity is treated similar to (3.66), but the mass-weighted horizontal velocities U and V are first interpolated to the half-level position $k+1/2$ of w using the averaging operator (3.19):

$$\left(\frac{\partial w}{\partial t}\right)^n_{ha} = -\frac{1}{\overline{\sqrt{\gamma}}^{\,\zeta}a\cos\varphi}\left\{\overline{\overline{U^n}^{\,\zeta}(\delta_\lambda w^n)}^{\,\lambda} + \overline{\overline{V^n}^{\,\zeta}\cos\varphi(\delta_\varphi w^n)}^{\,\varphi}\right\} . \tag{3.67}$$

(c) Vertical Advection

For the numerical formulation of the vertical advection terms there are two options. The first is an explicit scheme where the advective tendencies are evaluated using the variables at the centre time level n of the leapfrog integration. The spatial discretization is second-order accurate and reads

$$\left(\frac{\partial \psi}{\partial t}\right)_{va}^{n} = -\frac{1}{\sqrt{\gamma}}\overline{(W^c \delta_\zeta \psi^n)}^{\zeta} \tag{3.68}$$

for the scalar variables $\psi = p'$, T and q,

$$\begin{aligned} \left(\frac{\partial u}{\partial t}\right)_{va}^{n} &= -\frac{1}{\overline{\sqrt{\gamma}}^{\lambda}}\overline{\left(\overline{W^c}^{\lambda}\delta_\zeta u^n\right)}^{\zeta} \\ \left(\frac{\partial v}{\partial t}\right)_{va}^{n} &= -\frac{1}{\overline{\sqrt{\gamma}}^{\varphi}}\overline{\left(\overline{W^c}^{\varphi}\delta_\zeta v^n\right)}^{\zeta} \end{aligned} \tag{3.69}$$

for the components of the horizontal wind velocity, and

$$\left(\frac{\partial w}{\partial t}\right)_{va}^{n} = -\frac{1}{\overline{\sqrt{\gamma}}^{\zeta}}\overline{\left(\overline{W^c}^{\zeta}\delta_\zeta w^n\right)}^{\zeta} \tag{3.70}$$

for the vertical velocity. These finite difference formulations make use of the mass-weighted contravariant vertical velocity $W^c = \sqrt{\gamma}\dot{\zeta}$, which is evaluated from Eq. (3.62) for the current time level n.

The evaluation of (3.68) - (3.70) at the uppermost and the lowest model layer requires boundary conditions on W^c. As already mentioned in the previous section on the fast modes, non-penetrative boundary conditions are imposed at the upper ($\zeta = 1/2$) and lower ($\zeta = N_\zeta + 1/2$) boundaries, i.e.

$$\begin{aligned} W^c_{1/2} &= 0\,, \\ W^c_{N_\zeta+1/2} &= 0\,. \end{aligned} \tag{3.71}$$

In addition to (3.71), top and bottom boundary conditions on the physical vertical velocity w are required to calculate the advection term (3.70). These are derived from Eq. (3.71) by imposing free slip conditions on the horizontal velocity at the upper and lower boundaries. Details are described in Section 5.

The other option is a modified *Crank-Nicolson*-scheme which is based on time averaging of the advected quantities on the right hand side of (3.68) - (3.70) using the time levels $n-1$ and $n+1$ of the current time step n. The averaging operator is defined by

$$\overline{(\psi^n)}^{t,v} = \beta_v^+ \,\tilde{\psi}^{n+1} + \beta_v^- \,\psi^{n-1} \tag{3.72}$$

for any prognostic model variable ψ, where

$$\beta_v^+ = \frac{1}{2}(1+\beta_v) \qquad \text{and} \qquad \beta_v^- = \frac{1}{2}(1-\beta_v) = 1-\beta_v^+ \tag{3.73}$$

are the weighting coefficients. The parameter β_v determines the time-weighting. The scheme is unconditionally stable for $\beta_v > 0$. $\beta_v = 0$ gives a time-centred scheme and $\beta_v = 1$ corresponds to a fully implicit scheme. For stability reasons, as the vertical velocities can become large in case of applications on the meso-γ scale, the centred implicit scheme with $\beta_v = 0$ is used by default.

Since the final value ψ^{n+1} of a prognostic variable in an integration step is not known at the set-up time n for the slow tendencies, the time average in (3.72) is formulated with respect to a provisional value $\tilde{\psi}^{n+1}$. This value would result from the leapfrog integration if no fast-mode terms were present. Thus, the formulation is not completely implicit but quasi-implicit with respect to the slow modes. The forcing term f_ψ is then calculated according Eq. (3.35), where the tendencies due to vertical advection on the right hand side are given by

$$\left(\frac{\partial \psi}{\partial t}\right)^n_{va} = -\frac{\beta_v^+}{\sqrt{\gamma}}\overline{\left(W^c \delta_\zeta \tilde{\psi}^{n+1}\right)}^\zeta - \frac{\beta_v^-}{\sqrt{\gamma}}\overline{(W^c \delta_\zeta \psi^{n-1})}^\zeta \tag{3.74}$$

for the scalar variables $\psi = p'$, T and q, and by

$$\begin{aligned}
\left(\frac{\partial u}{\partial t}\right)^n_{va} &= -\frac{\beta_v^+}{\overline{\sqrt{\gamma}}^\lambda}\overline{\left(\overline{W^c}^\lambda \delta_\zeta \tilde{u}^{n+1}\right)}^\zeta - \frac{\beta_v^-}{\overline{\sqrt{\gamma}}^\lambda}\overline{\left(\overline{W^c}^\lambda \delta_\zeta u^{n-1}\right)}^\zeta \\
\left(\frac{\partial v}{\partial t}\right)^n_{va} &= -\frac{\beta_v^+}{\overline{\sqrt{\gamma}}^\varphi}\overline{\left(\overline{W^c}^\varphi \delta_\zeta \tilde{v}^{n+1}\right)}^\zeta - \frac{\beta_v^-}{\overline{\sqrt{\gamma}}^\varphi}\overline{\left(\overline{W^c}^\varphi \delta_\zeta v^{n-1}\right)}^\zeta \\
\left(\frac{\partial w}{\partial t}\right)^n_{va} &= -\frac{\beta_v^+}{\overline{\sqrt{\gamma}}^\zeta}\overline{\left(\overline{W^c}^\zeta \delta_\zeta \tilde{w}^{n+1}\right)}^\zeta - \frac{\beta_v^-}{\overline{\sqrt{\gamma}}^\zeta}\overline{\left(\overline{W^c}^\zeta \delta_\zeta w^{n-1}\right)}^\zeta
\end{aligned} \tag{3.75}$$

for the components of the wind velocity.

The procedure results in a coupled tridiagonal set of equations for the variables in a vertical column. Details on the method of solution are described below. The upper and lower boundary conditions are the same as for the vertically explicit option.

(d) Vertical Turbulent Diffusion

Similar to vertical advection, LM offers options for an explicit or an implicit numerical treatment of vertical diffusion. The differential form of the turbulent mixing terms is given by flux-gradient relations

$$\begin{aligned}
M_T^{TD} &= \frac{1}{\rho\sqrt{G}}\frac{\partial}{\partial\zeta}\left(\frac{\rho\pi K_h^v}{\sqrt{G}}\frac{\partial\theta}{\partial\zeta}\right), \qquad M_{q^x}^{TD} = \frac{1}{\rho\sqrt{G}}\frac{\partial}{\partial\zeta}\left(\frac{\rho K_h^v}{\sqrt{G}}\frac{\partial q^x}{\partial\zeta}\right), \\
M_u^{TD} &= \frac{1}{\rho\sqrt{G}}\frac{\partial}{\partial\zeta}\left(\frac{\rho K_m^v}{\sqrt{G}}\frac{\partial u}{\partial\zeta}\right), \qquad M_v^{TD} = \frac{1}{\rho\sqrt{G}}\frac{\partial}{\partial\zeta}\left(\frac{\rho K_m^v}{\sqrt{G}}\frac{\partial v}{\partial\zeta}\right),
\end{aligned}$$

for temperature (M_T^{TD}), moisture variables ($M_{q^x}^{TD}$) and horizontal momentum (M_u^{TD} and M_v^{TD}), using a corresponding turbulent diffusion coefficient for heat (K_h^v) and for momentum (K_m^v). π denotes the Exner pressure function.

The transport coefficients are calculated by a diagnostic turbulence closure method by default. Optionally, a prognostic equation for turbulent kinetic energy may be applied (TKE-scheme, see Eq. (2.154)) to determine the diffusion coefficients. In this case, the flux-gradient

relations for the heat and moisture fluxes take a different form under cloudy conditions, since a sub-grid scale condensation scheme is also applied. Details on the TKE scheme and on the related numerical treatment are described in Part II of the LM Documentation. Perturbation pressure as well as vertical velocity are not subject to turbulent vertical mixing. For the actual numerical solution, the diffusion coefficients for momentum and heat – K_m^v and K_h^v – are replaced by the modified coefficients K_M^v and K_H^v from (5.21) to take computational background mixing in physical space into account.

The spatial discretization uses second-order accurate vertical differencing. K_M^v and K_H^v are defined at the $(i, j, k + 1/2)$-position on the staggered grid, i.e. at the half level position of the vertical velocity. Using (3.18) to express the Jacobian $\sqrt{G}$ in terms of $\sqrt{\gamma}$ yields the finite difference forms

$$\begin{aligned} M_T^{TD} &= g \left(\frac{\rho_0}{\rho^n}\right) \frac{1}{\sqrt{\gamma}} \delta_\zeta \left(\hat{H}^3\right), \qquad M_q^{TD} = g \left(\frac{\rho_0}{\rho^n}\right) \frac{1}{\sqrt{\gamma}} \delta_\zeta \left(F_q^3\right), \\ M_u^{TD} &= g \overline{\left(\frac{\rho_0}{\rho^n}\right)}^{\lambda} \frac{1}{\overline{\sqrt{\gamma}}^{\lambda}} \delta_\zeta \left(\tau^{13}\right), \qquad M_v^{TD} = g \overline{\left(\frac{\rho_0}{\rho^n}\right)}^{\varphi} \frac{1}{\overline{\sqrt{\gamma}}^{\varphi}} \delta_\zeta \left(\tau^{23}\right), \end{aligned} \tag{3.76}$$

where $\hat{H}^3$ denotes the vertical flux of sensible heat normalized by c_{pd} (as this factor cancels in the heat equation). The turbulent fluxes of sensible heat, moisture (F_q^3) and momentum (τ^{13} and τ^{23}) are defined at model half levels and at the horizontal position corresponding to the transported quantity.

The turbulent fluxes in (3.76) may not be evaluated with the variables from the current time level n, because the leapfrog scheme is unstable with respect to parabolic friction terms. Thus, we use a modified *Crank-Nicolson*-scheme for vertical diffusion which involves the time levels $n - 1$ and $n + 1$ by applying the averaging operator

$$\overline{(\psi^n)}^{t,d} = \beta_d^+ \, \tilde{\psi}^{n+1} + \beta_d^- \, \psi^{n-1} \tag{3.77}$$

to the diffused quantities. β_d^+ determines the time weighting and $\beta_d^- = 1 - \beta_d^+$. The finite difference formulation for the fluxes is

$$\begin{aligned} \hat{H}^3 &= g \frac{\overline{(\rho_0 \rho^n)}^{\zeta}}{\overline{\sqrt{\gamma}}^{\zeta}} \overline{\pi^n}^{\zeta} K_H^v \left(\beta_d^+ \, \delta_\zeta \tilde{\theta}^{n+1} + \beta_d^- \, \delta_\zeta \theta^{n-1}\right), \\ F_q^3 &= g \frac{\overline{(\rho_0 \rho^n)}^{\zeta}}{\overline{\sqrt{\gamma}}^{\zeta}} K_H^v \left(\beta_d^+ \, \delta_\zeta \tilde{q}^{n+1} + \beta_d^- \, \delta_\zeta q^{n-1}\right), \\ \tau^{13} &= \frac{g}{\overline{\sqrt{\gamma}}^{\zeta,\lambda}} \overline{\overline{(\rho_0 \rho^n)}^{\zeta} K_M^v}^{\lambda} \left(\beta_d^+ \, \delta_\zeta \tilde{u}^{n+1} + \beta_d^- \, \delta_\zeta u^{n-1}\right), \\ \tau^{23} &= \frac{g}{\overline{\sqrt{\gamma}}^{\zeta,\varphi}} \overline{\overline{(\rho_0 \rho^n)}^{\zeta} K_M^v}^{\varphi} \left(\beta_d^+ \, \delta_\zeta \tilde{v}^{n+1} + \beta_d^- \, \delta_\zeta v^{n-1}\right). \end{aligned} \tag{3.78}$$

The vertically explicit option is revealed by setting $\beta_d^+ = 0$ (i.e. $\beta_d^- = 1$). In this case, the values of the diffusion coefficients are limited according to a stability criterion depending on

the large time step and the vertical grid spacing:

$$K^v_{H,M} \leq \frac{1}{8}\frac{(\delta_\zeta z)^2}{\Delta t} = \frac{1}{8\Delta t}\left(\frac{\overline{\sqrt{\gamma}}^{\zeta}}{g\overline{\rho_0}^{\zeta}}\right)^2 . \tag{3.79}$$

Near the surface, where the vertical grid spacing is usually chosen to be much smaller than in the free atmosphere, this stability condition may severely limit the turbulent fluxes to unrealistic small values.

For $\beta_d^+ > 0$ the scheme becomes implicit with respect to the provisional values $\tilde{\psi}^{n+1}$. This yields a coupled tridiagonal set of equations for the variables in a vertical column which may be combined with the set resulting from the vertically implicit option for vertical advection. Details on the method of solution are described below.

A linear stability analysis of the implicit scheme shows that unconditionally stable solutions are obtained for $\beta_d^+ \geq 0.5$. However, when the diffusion coefficients are very large, high frequency oscillation of period $4\Delta t$ may be excited. Stable and non-oscillating solutions of the linear diffusion equation are obtained for the full implicit option using $\beta_d^+ = 1$. Practical applications in NWP-models have shown that, even if β_d^+ is set to one, $4\Delta t$ oscillation are frequently excited due to nonlinear effects, especially near the surface where the turbulent fluxes may become large in case of unstable thermal stratification.

The time weighting parameter β_d^+ can be used to control these oscillations. It has been found that super-implicit weighting factors $\beta_d^+ > 1$ efficiently damp the $4\Delta t$ noise, but also have a detrimental impact to the physical solution. Thus, we make the weighting factor to be a function of the computational vertical coordinate (β_d^+ is defined at half-levels to achieve mass conservation) and assign super-implicit weights only near the surface. By default, β_d^+ is set to 1.2 at the lowest model level, and gradually decreasing values are specified with increasing height in the planetary boundary layer. In the free atmosphere, a constant value of $\beta_d^+ = 0.75$ is used.

The calculation of the mixing terms in Eq. (3.76) requires boundary conditions on the turbulent fluxes at the top ($\zeta = 1/2$) and the bottom ($\zeta = N_\zeta + 1/2$) half level of the model. At the upper boundary, zero flux conditions are specified:

$$\begin{aligned}
(\tau^{13})_{\zeta=1/2} &= 0\,, \\
(\tau^{23})_{\zeta=1/2} &= 0\,, \\
(\hat{H}^3)_{\zeta=1/2} &= 0\,, \\
(F^3_q)_{\zeta=1/2} &= 0\,.
\end{aligned} \tag{3.80}$$

At the lower boundary, the turbulent fluxes of momentum, heat and moisture are specified using the surface flux parameterizations as described in Part II of the LM documentation; the surface fluxes of liquid and solid forms of water are set to zero:

$$\begin{aligned}
(\tau^{13})_{\zeta=N_\zeta+1/2} &= \tau^{13}_{sfc} = -\rho\, C^d_m |\mathbf{v}_h|\, u\,, \\
(\tau^{23})_{\zeta=N_\zeta+1/2} &= \tau^{23}_{sfc} = -\rho\, C^d_m |\mathbf{v}_h|\, v\,, \\
(\hat{H}^3)_{\zeta=N_\zeta+1/2} &= \hat{H^3_{sfc}} = -\rho\, C^d_h |\mathbf{v}_h|\, (\theta\pi_{sfc} - T_{sfc})\,, \\
(F^3_{q^v})_{\zeta=N_\zeta+1/2} &= (F^3_{q^v})_{sfc} = -\rho\, C^d_q |\mathbf{v}_h|\, (q^v - q^v_{sfc})\,,
\end{aligned} \tag{3.81}$$

$$
\begin{aligned}
(F^3_{q^l})_{\zeta=N_\zeta+1/2} &= 0\,, \\
(F^3_{q^f})_{\zeta=N_\zeta+1/2} &= 0\,.
\end{aligned}
$$

In (3.81), u, v, θ and q^v are, respectively, the horizontal velocity components, potential temperature and specific humidity at the lowest grid level above the surface ($\zeta = N_\zeta$), and $|\mathbf{v}_h|$ is the absolute horizontal wind speed at the same level. C_m^d denotes the drag coefficient for momentum exchange at the ground and C_h^d is the bulk-areodynamical transfer coefficient for turbulent heat and moisture exchange at the surface. For the calculation of C_m^d and C_h^d see Part II. The air density ρ is evaluated from the surface values of temperature, pressure and humidity. Temperature and specific humidity at the ground are either provided by the soil model or externally specified. The surface pressure is determined by hydrostatic extrapolation of the total pressure $p = p_0 + p'$ at the first model layer above the ground.

(e) Other Contributions to Slow-Mode Forcing

Besides horizontal advection, vertical advection and vertical diffusion, the cloud microphysical source terms S_q, the radiative heating Q_r, sub-grid scale moist convection M_ψ^{MC}, computational mixing M_ψ^{CM}, lateral boundary relaxation M_ψ^{LB} and Rayleigh damping M_ψ^{RD}at the upper boundary contribute to the slow mode forcing f_ψ.

The physical forcings due to radiation and due to moist convection are also evaluated at time level $n-1$. The calculation of these terms by corresponding parameterization schemes is described in subsequent Part II of the documentation.

S_q contains various source terms related to grid-scale clouds and precipitation (see Part II). These terms are also evaluated at time level $n-1$, except for the condensation and evaporation rate of cloud water, which is calculated by a saturation adjustment scheme. By this method, the updated values of temperature, water vapour and cloud water at time level $n+1$ (updated due to all processes except condensation and evaporation) are isobarically adjusted to yield a saturated thermodynamic state. Details on the adjustment scheme are described in Part II.

Also, for stability reasons, the computational mixing terms M_ψ^{CM}, the boundary relaxation terms M_ψ^{LB} and the Rayleigh damping terms M_ψ^{RD} will not be included in the slow-mode forcing. These terms are integrated sequentially by using the Marchuk (1975) splitting method, using initially the updated variables at time level $n+1$ resulting from the small time step integration followed by saturation adjustment. The numerical formulation of these terms is discussed in Sections 6.2, 5.2.2 and 6.4, respectively.

(f) Method of Solution

The implicit option on vertical advection and diffusion yields a simultaneous equation for the provisional values $\tilde{\psi}^{n+1}$ of the prognostic variables for each vertical column in grid-point space. By evaluating the finite difference equations described above, we obtain a linear tridiagonal equation system, which may be written in the general form

$$
A_k\,\tilde{\psi}^{n+1}_{k-1} + B_k\,\tilde{\psi}^{n+1}_{k} + C_k\,\tilde{\psi}^{n+1}_{k+1} = D_k\,, \tag{3.82}
$$

for variables defined on main levels indicated by k ($k = 1, \cdots, N_\zeta$). Boundary conditions will be included in the inhomogeneous term D_k such that $A_1 = 0$ and $C_{N_\zeta} = 0$. Thus we have a

tridiagonal matrix structure of dimensions $(N_\zeta) \times (N_\zeta)$.

$$\begin{bmatrix} B_1 & C_1 & & & & & \\ A_2 & B_2 & C_2 & & & & \\ & A_3 & B_3 & C_3 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & A_{N_\zeta-1} & B_{N_\zeta-1} & C_{N_\zeta-1} \\ & & & & & B_{N_\zeta} & C_{N_\zeta} \end{bmatrix} \begin{bmatrix} \tilde{\psi}_1 \\ \tilde{\psi}_2 \\ \tilde{\psi}_3 \\ \cdot \\ \cdot \\ \tilde{\psi}_{N_\zeta-1} \\ \tilde{\psi}_{N_\zeta} \end{bmatrix}^{n+1} = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ \cdot \\ \cdot \\ D_{N_\zeta-1} \\ D_{N_\zeta} \end{bmatrix}$$

The matrix diagonals have the form

$$\begin{aligned} A_k &= \beta_v^+ A_k^v + (\beta_d^+)_{k-1/2} A_k^d , \\ C_k &= \beta_v^+ C_k^v + (\beta_d^+)_{k+1/2} C_k^d , \\ B_k &= \frac{1}{2\Delta t} - A_k - C_k' , \end{aligned} \tag{3.83}$$

where

$$\begin{aligned} A_k^v &= -\frac{1}{2\sqrt{\gamma}_k} W_{k-1/2}^c , \quad k = 2, \cdots, N_\zeta \\ C_k^v &= +\frac{1}{2\sqrt{\gamma}_k} W_{k+1/2}^c , \quad k = 1, \cdots, N_\zeta - 1 \end{aligned} \tag{3.84}$$

$$\begin{aligned} A_k^d &= -g^2 \left(\frac{\rho_o}{\rho}\right)_k \frac{1}{\sqrt{\gamma}_k} \frac{\overline{(\rho_0 \rho)}^\zeta_{k-1/2}}{\overline{\sqrt{\gamma}}^\zeta_{k-1/2}} K_{k-1/2}^v , \quad k = 2, \cdots, N_\zeta \\ C_k^d &= -g^2 \left(\frac{\rho_o}{\rho}\right)_k \frac{1}{\sqrt{\gamma}_k} \frac{\overline{(\rho_0 \rho)}^\zeta_{k+1/2}}{\overline{\sqrt{\gamma}}^\zeta_{k+1/2}} K_{k+1/2}^v , \quad k = 1, \cdots, N_\zeta - 1 \end{aligned}$$

The term C_k' in the main diagonal B_k (3.83) is equal to C_k for all model layers $k < N_\zeta$, but for the lowest layer C_k' is defined by

$$C_k' = -g \left(\frac{\rho_o}{\rho}\right)_k \frac{(\beta_d^+)_{k+1/2}}{\sqrt{\gamma}_k} \rho_{k+1/2} |\mathbf{v}_h|_k C^d , \quad k = N_\zeta , \tag{3.85}$$

according to the lower boundary conditions from Eq. (3.81). The inhomogeneous term D_k contains all explicit contributions to the slow-mode forcing (except those from condensation-evaporation, computational mixing, lateral boundary relaxation and Rayleigh damping):

$$\begin{aligned} D_k &= \frac{\psi^{n-1}}{2\Delta t} + \left(\frac{\partial \psi}{\partial t}\right)_{ha}^n + M_\psi^{MC} + M_\psi^{CM} + Q \\ &\quad + \left(\beta_v^- A_k^v + \{\beta_d^-\}_{k-1/2} A_k^d\right) \left(\psi_k^{n-1} - \psi_{k-1}^{n-1}\right) \\ &\quad - \left(\beta_v^- C_k^v + \{\beta_d^-\}_{k+1/2} C_k^d\right) \left(\psi_{k+1}^{n-1} - \psi_k^{n-1}\right) \end{aligned} \tag{3.86}$$

Other forcings than horizontal advection, moist convection and computational mixing are abbreviated by Q. For the lowest model layer $k = N_\zeta$, the last term in (3.86) is replaced by

$$- C'_k(\beta_d^- / \beta_d^+)_{k+1/2} \left(\psi_{sfc}^{n-1} - \psi_k^{n-1}\right) - C'_k \psi_{sfc}^{n+1} \tag{3.87}$$

to include the lower boundary conditions.

Eq.(3.82) is solved for $\tilde{\psi}^{n+1}$ using a standard tridiagonal solver based on Gaussian elimination. Finally, the slow-mode tendencies are calculated from

$$f_\psi^n = \frac{\tilde{\psi}^{n+1} - \psi^{n-1}}{2\Delta t} . \tag{3.88}$$

According to the finite difference formulations (3.76) and (3.78), the factors in the diagonals A_k and C_k for $\psi = u$ and $\psi = v$ are evaluated at the corresponding positions on the C-grid. In case of $\psi = T$, the forcing for the turbulent flux of sensible heat is by the vertical gradient of potential temperature. Thus, the tridiagonal system has to be modified accordingly. To avoid a coupling of the T- and the p'-equations, the potential temperature is calculated using the pressure at the centred time level. The tridiagonal matrix equation for the slow-mode tendency of the vertical velocity takes also a slightly different form due to vertical averaging.

3.3.4 Outline of an Integration Step

As mentioned in the previous subsection, not all terms contributing to the tendency due to slow modes are considered by the forcing function f_ψ, which is used in the small time step sub-integration of the equations. The remaining terms are integrated subsequent to time splitting using the Marchuk splitting method Marchuk (1975).

To illustrate this method, we rewrite the model equations in the symbolic form (3.31) as

$$\frac{\partial \psi}{\partial t} = s_\psi + f_\psi^{TS} + S_\psi^c + M_\psi^{CM} + M_\psi^{LB} + M_\psi^{RD} . \tag{3.89}$$

s_ψ denotes the terms related to the fast modes and f_ψ^{TS} represents the slow-mode tendencies except for cloud condensation and evaporation (S_ψ^c), computational mixing (M_ψ^{CM}), lateral boundary relaxation (M_ψ^{LB}) and Rayleigh damping at the upper boundary (M_ψ^{RD}).

The idea of the Marchuk method is to split a complex system of equations into a number of simpler subsystems representing different processes. These are then solved consecutively one at a time, where different numerical schemes may be applied for each subsystem. Starting with given initial values of the variables, the integration of the first subsystem yields updated variables which are then used as initial conditions to integrate the second subsystem (thus, the Marchuk splitting is much different from the mode splitting scheme described above). This procedure is continued until the last subsystem is integrated yielding the final values of the variables at time level $n + 1$.

Applying the Marchuk method to the model equations (3.89) results in the following organization of time stepping.

(1) *Calculation of the Tendencies due to Slow Modes*
Given the values of the variable at time level $n - 1$ and n, the slow-mode forcing f_ψ^{TS} has to be set up first. The calculation of this term has been described in Section 4.3.3.

(2) *Mode-Splitting Time Integration*
In the second step, the first two subsystem from (3.89),

$$\frac{\partial \psi}{\partial t} = s_\psi + f_\psi^{TS}, \tag{3.90}$$

are then integrated simultaneously using the mode-splitting time scheme as described in Section 4.3.1 and 4.3.2. After a number of small time steps, this algorithm yields provisional values ψ^* of the variables at time level $n+1$.

(3) *Saturation Adjustment*
Using the updated variables ψ^* as initial values, the terms related to condensation and evaporation are then integrated in the third subsystem:

$$\frac{\partial \psi}{\partial t} = S_\psi^c. \tag{3.91}$$

The saturation adjustment scheme described in Part II of the LM Documentation is used to integrate (3.91) resulting in new provisional values ψ^{**} for time level $n+1$.

(4) *Computational Mixing*
The next step integrates the fourth subsystem from (3.89),

$$\frac{\partial \psi}{\partial t} = M_\psi^{CM}(\psi), \tag{3.92}$$

i.e. the provisional variables ψ^{**} from the previous step are updated for the computational mixing (horizontal diffusion) terms M_ψ^{CM} to yield new provisional values ψ^{***}. These terms M_ψ^{CM} are treated explicitly, their formulations and finite difference representation is discussed in Section 6.2.

(5) *Lateral Boundary Relaxation*
In the following step, the fifth subsystem from (3.89) is integrated in time,

$$\frac{\partial \psi}{\partial t} = M_\psi^{LB}(\psi), \tag{3.93}$$

i.e. the provisional variables ψ^{***} from the previous step are updated for the lateral boundary relaxation terms M_ψ^{LB} to yield new provisional values ψ^{****}. As described in Section 5.2.2, an implicit time scheme is used to integrate (3.93) for stability reasons.

(6) *Rayleigh Damping*
Finally, the last subsystem from (3.89) considering the Rayleigh damping in the upper model layers,

$$\frac{\partial \psi}{\partial t} = M_\psi^{RD}(\psi), \tag{3.94}$$

is integrated similar to step (5) by an implicit scheme (see Section 6.4) using the updated variables ψ^{****} as initial values. This yields the final values of the variables valid at time level $n+1$ and completes a time step.

This basic form of an integration cycle is reflected in the structure of the program code. Details on the implementation of the time integration scheme are discussed in Part IV of the LM-documentation.

3.4 Special Transport Schemes

The numerical treatment of 3-d transport using Leapfrog time integration with centred spatial differences will introduce small-scale oscillations due to the Gibbs phenomenon. Negative oscillations, however, are problematic for positive definite variables such as specific humidity, cloud water, cloud ice, rain, snow or the turbulent kinetic energy. Setting negative undershots simply to zero will result in a mean increase of the corresponding variable, that is in a continuous increase in mass. Thus, special schemes must be applied to guarantee a positive numerical solution.

It is also important to note that the equations for the constituents of the hydrological cycle, (2.148) and (2.149), and for turbulent kinetic energy (2.154) are numerically decoupled from the scheme concerning the integration of the thermodynamic equations (2.143) - (2.147) for momentum, temperature and perturbation pressure due to the mode-splitting approach described in Section 4.3. Thus, numerical integration schemes other than Leapfrog with centred spatial differences may be applied for the hydrological cycle.

3.4.1 Vertical Redistribution

The default model set-up utilizes the LM standard cloud microphysics scheme (see Part II of the documentation) which predicts cloud water (q^c) and specific humidity (q^v) and diagnoses the precipitation fluxes of rain and snow. In order to avoid negative values of q^c and q^v resulting from the time-integration scheme as described in Section 4.3.3, a simple 'hole-filling' algorithm based on vertical redistribution is applied. That is, whenever a negative value occurs in a layer k, it is set back to zero and the corresponding mass deficit is subtracted from the mass in the layer $k+1$ below.

Let $\tilde{q}_k^{n+1}$ denote the provisional concentration of water vapour or cloud water at time level $n+1$ in a layer k as resulting from the inversion of the tridiagonal matrix (3.82). Then the recursive algorithm to obtain the final values q_k^{n+1} starts at the top level $k=1$ and may be written as

$$\begin{aligned} q_k^{n+1} &= \max\left\{0,\, \tilde{q}_k^{n+1} + \Delta m_{k-1}/(\rho\Delta z)_k\right\}, \\ \Delta m_k &= \min\left\{0,\, q_k^{n+1}(\rho\Delta z)_k\right\}, \end{aligned} \tag{3.95}$$

for $k = 1, \ldots, N_\zeta$. Δm_k is the mass deficit in a layer with density ρ and thickness Δz. Density is evaluated at time level n and the start value for the mass deficit is set to $\Delta m_0 = 0$. The algorithm assures that the vertical integral of ρq, i.e the total mass of cloud water or water vapour per unit area is conserved during redistribution – as long as the mass deficit at the bottom layer turns out to be zero, $\Delta m_{N_\zeta} = 0$. In this case we have

$$\sum_{k=1}^{N_\zeta} \rho_k q_k^{n+1} \Delta z_k = \sum_{k=1}^{N_\zeta} \rho_k \tilde{q}_k^{n+1} \Delta z_k \,.$$

If Δm_{N_ζ} is negative, the corresponding mass amount is artificially introduced in the model atmosphere and mass conservation is violated. This simple hole-filling algorithm results in a strong vertical smoothing of q but can introduce small-scale two-grid-interval noise in the horizontal direction since negative values resulting from horizontal transport are corrected by

vertical redistribution. However, the scheme is a computationally cheap method to achieve a positive definite numerical solution.

3.4.2 Horizontal Advection

Transport of cloud ice (q^i) is much more crucial to numerical errors in the advection algorithm since depositional growth is calculated by a non-equilibrium approach in contrast to the instantaneous saturation adjustment for cloud water condensation-evaporation. Thus, a special scheme for horizontal transport should be used to predict q^i. We have implemented a variant of the Lin and Rood (1996) (LR hereafter) algorithm for this purpose. This algorithm extends arbitrary 1-d flux-form transport schemes to multi-dimensions and allows for relatively large time steps.

To outline shortly the LR-scheme, we consider the generic equation for 2-d horizontal transport

$$\frac{\partial \rho q}{\partial t} + \nabla_h \cdot (\mathbf{v}_h \rho q) \;=\; 0 \tag{3.96}$$

of a tracer constituent with mass fraction q. Without specifying details of the 1-d scheme, let X and Y denote 1-d operators representing the increments of the partial density field $Q = \rho q$ from timelevel n to $n+1$ due to the negative fluxes in x-direction and y-direction, respectively. A straightforward method to advance the solution would be to apply the operators simultaneously,

$$Q^{n+1} \;=\; Q^n + X(Q^n) + Y(Q^n)\,. \tag{3.97}$$

Such a scheme, however, is unstable due to the neglect of cross-derivative terms (or splitting terms) in the Taylor series expansions for the fluxes. One method to stabilize the scheme is to introduce the cross-derivative terms explicitly (Smolarkiewicz (1982); Schlesinger (1985)). However, the maximum stable Courant number is reduced significantly. Another method is based on sequential directional time-splitting Crowley (1968), where the X-operator is first applied in the x-direction, followed by the Y-operator applied in y-direction with the result of the x-transport as input:

$$Q^x \;=\; Q^n + X(Q^n)\,, \tag{3.98}$$

$$Q^{n+1} \;=\; Q^x + Y(Q^x)\,, \tag{3.99}$$

Substituting (3.98) into (3.99) yields

$$Q^{n+1} \;=\; Q^n + X(Q^n) + Y(Q^n) + YX(Q^n)\,, \tag{3.100}$$

where $YX(Q^n)$ is an abbreviation for $Y\{X(Q^n)\}$. This is just the splitting-term required to stabilize the scheme. The YX-operator has the form of a cross-derivative term, but as the scheme (3.100) results from directional splitting, it does not reduce the stable Courant-number in multi-dimensional applications. Actually, the linear CFL stability condition is $\max(c_x, c_y) \leq 1$ instead of the more stringent condition $c_x + c_y \leq 1$ of other multi-dimensional schemes (c_x and c_y denote the Courant numbers in x- and y-direction, respectively). This is the major advantage of directional splitting. However, a directional bias is introduced: Performing the y-advection first followed by x-advection results in an equally valid scheme

$$Q^{n+1} \;=\; Q^n + X(Q^n) + Y(Q^n) + XY(Q^n)\,, \tag{3.101}$$

which is antisymmetric to (3.100) with respect to the splitting term. Both schemes (3.100) and (3.101) are only first-order accurate in space due to the existence of the directional

bias. Second-order accuracy can be obtained by a symmetric sequence of the sequential splitting method Strang (1968). LR derive a symmetric form of the splitting scheme by simply averaging the two antisymmetric schemes (3.100) and (3.101):

$$Q^{n+1} = Q^n + X(Q^n) + Y(Q^n) + \frac{1}{2}\{YX(Q^n) + XY(Q^n)\} . \tag{3.102}$$

This algorithm eliminates the directional bias but is computational more expensive than either (3.100) or (3.101). To reduce total operation counts, LR rewrite (3.102) in the form

$$Q^{n+1} = Q^n + X\left\{Q^n + \frac{1}{2}Y(Q^n)\right\} + Y\left\{Q^n + \frac{1}{2}X(Q^n)\right\} . \tag{3.103}$$

As pointed out by LR, all three schemes (3.100), (3.101) and (3.103) satisfy mass-conservation due to the use of flux-form operators, and satisfy the directional-splitting stability condition $\max(c_x, c_y) \leq 1$. But consistency is not guaranteed: In case of a uniform Q-field in incompressible flow with $\nabla_h \cdot \mathbf{v}_h = 0$ there will be no change in partial density Q. All three schemes, however, predict a change proportional to the splitting term. To eliminate this error, LR replace the inner operators in (3.103) by their advective form counterparts denoted by X_a and Y_a:

$$Q^{n+1} = Q^n + X\left\{Q^n + \frac{1}{2}Y_a(Q^n)\right\} + Y\left\{Q^n + \frac{1}{2}X_a(Q^n)\right\} . \tag{3.104}$$

The contributions from inner advective form operators will be zero for a spatially uniform Q-field such that the consistency requirement is met. Since the outer operators are still in flux form, mass conservation is still guaranteed. Also – because the splitting terms are of higher order in time – it is sufficient to replace the two inner advective operators by first-order upstream operators to achieve an overall second-order accuracy. Equation (3.104) is the basic form of the general forward-in-time LR transport algorithm. The advantage is that it is highly flexible since any 1-d transport scheme including monotonicity constraints can be utilized, whereas many other methods for multidimensional advection are scheme dependent (e.g. Bott (1993); LeVeque (1993); Colella (1990)). Also, the extension to a semi-Lagrangian flux-form scheme with large time steps is possible (see LR). However, this option is not used in the current version of LM.

The budget equations (2.149) for the constituents of the hydrological cycle are written in advection form. In order to apply the LR-algorithm, we rewrite the horizontal advection term into a flux-divergence term and subtract the resulting wind-divergence term. The generic equation for horizontal transport of q then reads

$$\frac{\partial q}{\partial t} + \nabla_h \cdot (\mathbf{v}_h q) - q \nabla_h \cdot \mathbf{v}_h = 0 , \tag{3.105}$$

where we apply the LR-scheme (3.104) for the flux-divergence term with q instead of Q as the transported variable. At a gridpoint (i, j) the discretized form of the algorithm is written as

$$q^{n+1} = q^n + X(q^y) + Y(q^x) + q^n D \Delta t . \tag{3.106}$$

Here, D denotes the discretized wind divergence term

$$D = \frac{1}{a \cos\varphi}\{\delta_\lambda (u) + \delta_\varphi (v \cos\varphi)\} , \tag{3.107}$$

q^y and q^x are the intermediate values of q updated by the inner advection operators Y_a and X_a, respectively. A simple upstream operator is used for this purpose. Denoting the Courant

numbers in the horizontal directions at the cell centre (i, j) by

$$c_\lambda = \frac{\Delta t}{a\cos\varphi\Delta\lambda}(\overline{u}^{\lambda}), \qquad c_\varphi = \frac{\Delta t}{a\cos\varphi\Delta\varphi}(\overline{vcos\varphi}^{\varphi}), \tag{3.108}$$

we have for the intermediate values q^y and q^x:

$$q^y_{i,j} = q^n_{i,j} - \frac{1}{4}\{(c_\varphi)_{i,j}(q_{i,j+1} - q_{i,j-1}) - |(c_\varphi)_{i,j}|(q_{i,j+1} - 2q_{i,j} + q_{i,j-1})\}, \tag{3.109}$$

$$q^x_{i,j} = q^n_{i,j} - \frac{1}{4}\{(c_\lambda)_{i,j}(q_{i+1,j} - q_{i-1,j}) - |(c_\lambda)_{i,j}|(q_{i+1,j} - 2q_{i,j} + q_{i-1,j})\}. \tag{3.110}$$

For the outer flux-divergence operators X and Y a second-order Lax-Wendroff method with a monotonic flux-limiter has been implemented. Formally, we represent the operators as

$$X_{i,j} = -\frac{\Delta t}{a\cos\varphi}\delta_\lambda F^\lambda, \quad Y_{i,j} = -\frac{\Delta t}{a\cos\varphi}\delta_\varphi(cos\varphi F^\varphi). \tag{3.111}$$

In Equation (3.111), the Lax-Wendroff fluxes F^λ and F^φ are defined at the corresponding grid interfaces:

$$\begin{aligned} F^\lambda_{i+1/2,j} = {} & \max(0,u)\left\{q^y_{i,j} + 0.5(1-c_\lambda) * C_{i-1/2,j}\right\} + \\ & \min(0,u)\left\{q^y_{i+1,j} - 0.5(1+c_\lambda) * C_{i+1/2,j}\right\}, \end{aligned} \tag{3.112}$$

$$\begin{aligned} F^\varphi_{i,j+1/2} = {} & \max(0,v)\left\{q^x_{i,j} + 0.5(1-c_\varphi) * C_{i,j-1/2}\right\} + \\ & \min(0,v)\left\{q^x_{i,j+1} - 0.5(1+c_\varphi) * C_{i,j+1/2}\right\}. \end{aligned} \tag{3.113}$$

Here, the Courant numbers c_λ and c_φ are defined at the corresponding cell interfaces (i.e. as in (3.108), but without averaging), and C denotes the flux limiters. At present the monotonized centred (MC) limiter of van Leer (1977) is implemented. By defining the ratio of the slopes of the intermediate solution at the cell interfaces by

$$r_{i+1/2,j} = \frac{q^y_{i+2,j} - q^y_{i+1,j}}{q^y_{i+1,j} - q^y_{i,j}}, \qquad r_{i,j+1/2} = \frac{q^x_{i,j+2} - q^x_{i,j+1}}{q^x_{i,j+1} - q^x_{i,j}}, \tag{3.114}$$

the MC-limiter used in (3.112) and (3.113) can be formulated as

$$\begin{aligned} C_{i+1/2,j} &= \max\left\{0, \min\left(2,\, 2r_{i+1/2,j},\, \frac{1+r_{i+1/2,j}}{2}\right)\right\}, \\ C_{i,j+1/2} &= \max\left\{0, \min\left(2,\, 2r_{i,j+1/2},\, \frac{1+r_{i,j+1/2}}{2}\right)\right\}. \end{aligned} \tag{3.115}$$

This version of the scheme is applied to calculate horizontal advection of cloud ice, if the corresponding parameterization scheme is switched on. The recent model version allows also to apply a 3-d transport scheme for rain and snow. In this case, the horizontal transport of q^r and q^s is also treated by the LR-scheme with the MC-limiter.

In order to combine a two-timelevel forward-in-time integration scheme for q^i with the default three-timelevel Leapfrog scheme for the other variables, and to guarantee a positive definite solution, a Marchuk-splitting approach is applied. To illustrate the procedure, let us write the prognostic equation (2.149) for cloud ice as

$$\frac{\partial q^i}{\partial t} + \mathbf{v}_h \cdot \nabla_h q^i + \dot{\zeta}\frac{\partial q^i}{\partial \zeta} = S_i + M^{TD}_{q^i} + M^{LB}_{q^i} + M^{RD}_{q^i}. \tag{3.116}$$

Horizontal diffusion is not applied to the cloud ice field, $M_{q^i}^{CM} = 0$, and with the present convection scheme there is no feedback from cumulus convection, $M_{q^i}^{MC} = 0$. In the first integration step for (3.116), q^i is updated due to horizontal advection from timelevel n to timelevel $n+1$ as described above. These values are used as input to the scheme for vertical advection in a second step (see below). The resulting total advection tendency is then used as input the right hand side of the tridiagonal system (3.82) to solve implicitly for turbulent vertical diffusion $M_{q^i}^{TD}$ (see Section 4.3.3). The resulting values are then used to calculate the microphysical source terms S_i. The updated values from this step are finally subject to further updates due to lateral boundary forcing $M_{q^i}^{LB}$ and Rayleigh damping $M_{q^i}^{RD}$.

3.4.3 Vertical Advection

The vertical transport of cloud ice (and of rain and snow, if the 3-d transport scheme is switched on) is calculated as a second step in a Marchuk splitting with the variables resulting from horizontal transport as input. This introduces a directional bias which, however, is very small since horizontal transport dominate the vertical one on the meso-β scale. The advantage of this splitting is that the solution remains positive as long as a monotonic scheme is applied for both the horizontal and the vertical direction.

Analogue to the horizontal advection term, the vertical advection term in (3.116) is decomposed into a vertical flux-divergence term and a divergence term for the contravariant vertical velocity. For the actual computation, we use the mass-weighted vertical velocity W^c as defined in (3.61). Thus, the following subset of 3.116) is solved during Marchuk splitting:

$$\sqrt{\gamma}\frac{\partial q}{\partial t} + \frac{\partial W^c q}{\partial \zeta} - q\frac{\partial W^c}{\partial \zeta} = 0\,. \tag{3.117}$$

The procedure to solve (3.117) based on a Lax-Wendroff method with a MC-limiter as described in Section 4.4.2, but now with the fluxes and flux operator referring to the vertical direction, and the Courant numbers being evaluated for the mass-weighted contravariant vertical velocity W^c. In order to avoid instabilities in case of very large vertical velocities, the Courant number for vertical transport is artificially limited to a value of one.

3.4.4 Transport of Precipitation

The operational scheme for gridscale precipitation assumes column equilibrium for precipitating constituents rain and snow (see Part II of the LM Documentation). That is, sedimentation can be considered to be a fast process compared to the characteristic time scale of advection. By increasing the model's spatial resolution, this assumption gets more and more unrealistic. With the characteristic values of 5 m/s for the terminal velocity speed of rain and of 1 m/s for snow, the time scales for sedimentation of rain and snow are estimated as 500 s and 2500 s, respectively, by assuming a vertical fall distance of typically 2500 m for frontal precipitation. At 15 m/s wind speed, the corresponding horizontal displacements are about 8 km for rain and 40 km for snow – an effect that should be taken into account for grid spacings of 10 km or smaller. Horizontal transport is particularly important for the generation of lee-side precipitation, which is not sufficiently recognized by the operational LM. When going to the meso-γ scale, deep convection becomes resolved explicitly. Here, all instationary effects of cloud development, and especially the vertical advection of the precip-

itation phases must be considered to simulate realistically the life-cycle of deep convective cells.

In order to take these effects into account, the diagnostic equations for the precipitation fluxes P_x, where x stands for r (rain) or s (snow), respectively, have to be replaced by full prognostic equations (2.149) for the mixing ratios q^x. Considering the large terminal fall velocities of this hydrometeors, all mixing terms M_{q^x} will be neglected in the 3-d transport equations for rain and snow. Also, it is sufficient to specify simple outflow boundary conditions on the outflow boundaries, and zero boundary conditions on the inflow boundaries. With this simplification the prognostic equations for q^r and q^s read

$$\frac{\partial q^x}{\partial t} + \mathbf{v}_h \cdot \nabla_h q^x + \dot{\zeta}\frac{\partial q^x}{\partial \zeta} + \frac{1}{\rho\sqrt{G}}\frac{\partial}{\partial \zeta}(\rho v_x^T q^x) \;=\; S_x \,. \tag{3.118}$$

Here, v_x^T denotes the terminal fall velocity, which is a nonlinear function of q^x. The transformation of the current diagnostic parameterization scheme into a prognostic scheme using (3.118) requires three steps:

(a) a reformulation of the microphysical processes S_x in terms of q^x as dependent model variables (instead of P_x),

(b) an accurate and positive definite advection scheme, and

(c) a numerically efficient treatment of the sedimentation term.

Since the precipitation fluxes and the mixing ratios are uniquely related by basic parameterization assumptions, the reformulation of the source terms is straightforward (see Part II of the documentation). For threedimensional advection, the Lin and Rood (1996) algorithm as described in Section 4.4.2 and 4.4.3 is used.

The numerical treatment of sedimentation turns out to be more difficult: in case of precipitation falling through thin model layers near the surface, the Courant number may become larger than one. An implicit scheme, however, cannot be applied because the sedimentation velocity is a nonlinear function of the mixing ratio. Also, a semi-Lagrangian technique will be difficult to apply since the source terms have to be taken into account to allow for microphysical interactions during fallout. For the current testversion of the prognostic scheme, a relatively efficient integration method for (3.118) has been developed Gaßmann (2002). It is based on symmetric Strang process-splitting combined with local time-splitting for the sedimentation process, i.e. splitting is applied only for those layers where the local Courant number exceeds the stability limit. For details of the scheme, see Gaßmann (2002).

Meanwhile, an alternative numerical treatment of the sedimentation term is in evaluation. It is based on a quasi-linear approximation to the terminal velocity combined with an implicit upstream estimate for the fallout flux. Since the large numerical diffusion associated with the upstream approximation is not of importance due to the fact that fallout is very fast, the solutions with this scheme turn out to be very similar to the more accurate time-splitting scheme. However, the implicit upstream approach is by far more efficient. Given a successful testing, the new scheme will replace the time-splitting method in a future version of the model.

Section 4

Initial and Boundary Conditions

In a limited area model, only the lower boundary is physical. The boundaries at the top and the sides of the model domain are usually artificial. Various conditions at these boundaries may be specified. The conditions should be chosen to make sense in the context of a specific application.

With respect to the simulation of real data cases as well as for NWP purposes, it is important to use open or inflow-outflow lateral boundary conditions to allow the atmosphere in the model interior domain to interact with the external environment. In the present version of LM, only an externally specified time dependent forcing is taken into account by a one-way interactive nesting method. Another option is the use of periodic boundary conditions, which may be used for specific scientific applications.

We plan to implement additional options on lateral boundary conditions in near future. These will include Orlanski-type wave-radiation open boundary conditions Orlanski (1976) and a two-way interactive scheme for self-nesting of the model to focus on regions of interest. Work on the latter scheme, which will be based on the formulation in the MM5-model Grell et al. (1994) is in progress.

At the upper boundary, only non-penetrative conditions are supported at present. A Rayleigh damping scheme may be applied to suppress wave reflections from the rigid top boundary (see Section 6.4). Work on the implementation of a wave-radiation upper boundary conditions based on the formulation of Klemp and Durran (1983) has been started.

4.1 Initial Conditions

Typically, the initial conditions for a regional model are provided by interpolation from a coarse grid analysis or forecast. For operational applications and real data simulations the initialized analyses of the global model GME of DWD are used for this purpose. The actual interpolation is done in a separate pre-processor program, GME2LM, which is described in Part V of the LM-documentation. At present, the GME2LM is extended to provide also initial conditions derived from the ECMWF model or from an LM analysis or forecast at coarser resolution (for one-way self-nesting).

A four-dimensional data assimilation cycle based on a nudging analysis scheme can be in-

stalled for operational NWP with the LM at COSMO meteorological services. In this case, the initial conditions come from the continuous LM assimilation stream and only boundary data have to be provided by GME forecasts. The LM data assimilation system is described in Part III of this documentation.

For various research applications as well as for model testing and evaluation, the LM provides a capability to handle idealized cases using user-defined artificial initial and boundary data. For these types of application, periodic lateral boundary conditions can be specified optionally. Additionally, a 2-dimensional model configuration can be used.

4.2 Lateral Boundary Conditions

4.2.1 Periodic Boundary Condition

The periodic boundary condition assumes that the solution of the model equations replicates itself indefinitely outside of the computational domain. Thus, the solution at a distance d to the west (north) of the computational domain western (northern) boundary equals the solution at the same distance d to the west (north) of the eastern (southern) boundary.

According to the grid definition in Section 4.1.1, the grid-point indices of the computational domain run from $i = N_{off}+1, \cdots, N_\lambda - N_{off}$ in λ-direction and from $j = N_{off}+1, \cdots, N_\varphi - N_{off}$ in φ-direction, where N_{off} is an offset-parameter (usually set to 2) which assigns the external model domain outside to the lateral physical boundaries. Denoting the first and the last grid-point indices in λ-direction by

$$N_W \;=\; N_{off}+1 \quad , \quad N_E \;=\; N_\lambda - N_{off} \tag{4.1}$$

and in φ-direction by

$$N_S \;=\; N_{off}+1 \quad , \quad N_N \;=\; N_\varphi - N_{off} \tag{4.2}$$

yields the following periodic boundary conditions. ψ denotes scalars defined at the centre of a grid box as well as the vertical velocities w and $\dot{\zeta}$.

At the western and eastern boundaries:

$$\begin{aligned}
\psi_{N_W-1,j,k} &= \psi_{N_E\quad,j,k}\,, & \qquad \psi_{N_E+1,j,k} &= \psi_{N_W\quad,j,k}\,,\\
\psi_{N_W-2,j,k} &= \psi_{N_E-1,j,k}\,, & \qquad \psi_{N_E+2,j,k} &= \psi_{N_W+1,j,k}\,,\\
v_{N_W-1,j,k} &= v_{N_E\quad,j,k}\,, & \qquad v_{N_E+1,j,k} &= v_{N_W\quad,j,k}\,,\\
v_{N_W-2,j,k} &= v_{N_E-1,j,k}\,, & \qquad v_{N_E+2,j,k} &= v_{N_W+1,j,k}\,,\\
u_{N_W-\frac{1}{2},j,k} &= u_{N_E-\frac{1}{2},j,k}\,, & \qquad u_{N_E+\frac{1}{2},j,k} &= u_{N_W+\frac{1}{2},j,k}\,,\\
u_{N_W-\frac{3}{2},j,k} &= u_{N_E-\frac{3}{2},j,k}\,, & \qquad u_{N_E+\frac{3}{2},j,k} &= u_{N_W+\frac{3}{2},j,k}\,.
\end{aligned} \tag{4.3}$$

At the southern and northern boundaries:

$$
\begin{aligned}
\psi_{i,N_S-1,k} &= \psi_{i,N_N\ \ ,k}\,, & \psi_{i,N_N+1,k} &= \psi_{i,N_S\ \ ,k}\,,\\
\psi_{i,N_S-2,k} &= \psi_{i,N_N-1,k}\,, & \psi_{i,N_N+2,k} &= \psi_{i,N_S+1,k}\,,\\
v_{i,N_S-\frac{1}{2},k} &= v_{i,N_N-\frac{1}{2},k}\,, & v_{i,N_N+\frac{1}{2},k} &= v_{i,N_S+\frac{1}{2},k}\,,\\
v_{i,N_S-\frac{3}{2},k} &= v_{i,N_N-\frac{3}{2},k}\,, & v_{i,N_N+\frac{3}{2},k} &= v_{i,N_S+\frac{3}{2},k}\,,\\
u_{i,N_S-1,k} &= u_{i,N_N\ \ ,k}\,, & u_{i,N_N+1,k} &= u_{i,N_S\ \ ,k}\,,\\
u_{i,N_S-2,k} &= u_{i,N_N-1,k}\,, & u_{i,N_N+2,k} &= u_{i,N_S+1,k}\,.
\end{aligned}
\tag{4.4}
$$

4.2.2 Relaxation Boundary Condition

When we use a regional model for NWP purposes, information on the variables at the lateral boundaries and their time evolution must be specified by an external data set. These external data may be obtained by interpolation from a forecast run of another model or from a coarser resolution run of LM. At present, boundary data from the operational hydrostatic global model GME are supported. The actual interpolation is done in the separate pre-processor program GME2LM, which is described in Part V of the LM Documentation. The GME2LM is currently extended to provide also boundary conditions derived from ECMWF model forecasts or from LM forecasts at coarser resolution. Time dependent relaxation boundary conditions can then be used to force the solution at the lateral boundaries using the external data. This method is also referred to as one-way interactive nesting.

Nesting a high-resolution limited area model in a low-resolution driving model causes numerical problems, since the time evolution of the model variables is based on a system of equations that can differ from that of the driving model. The problems are related to a non-unique information transfer between the models at the boundaries, attributable to differences in the spatial resolution as well as to the use of different sets of model equations. E.g., information that cannot be transferred from the regional model to the driving model is reflected back from the boundaries and can severely contaminate the high-resolution model variables. This leads to the generation of numerical noise, which can propagate from the lateral boundaries inward to the centre of the model domain.

A simple and effective solution to this problem is to apply a sponge to the model variables within a relaxation zone close to the boundaries. In this zone, the variables of the high-resolution model are gradually modified to blend them with the driving model variables. In this way, the information transfer problem is cured, since information near the lateral boundaries is no longer generated by the high-resolution model but determined by the values of the low-resolution driving model.

A relaxation boundary condition similar to that discussed by Davies (1976) and Davies (1983) is used for one-way nesting. With this scheme, an extra forcing term M_ψ^{LB} is added to the right hand side of the prognostic equations. The tendency due to lateral boundary forcing is of the form

$$
\left(\frac{\partial \psi}{\partial t}\right)_{LB} = M_\psi^{LB} = -\mu_b\,(\psi - \psi_b) \tag{4.5}
$$

where ψ_b is the externally specified value of a prognostic variable ψ and μ_b is the relaxation coefficient. (4.5) is applied to all prognostic model variables except for the vertical velocity. A free slip lateral boundary condition is specified for w.

In LM, the external variables ψ_b are defined as three-dimensional fields which are available at discrete specified times T_N with time interval ΔT (the default value is $\Delta T = 1hr$). Within each time interval for boundary updating, ψ_b^n at time level n is a value linearly interpolated between two times T_N and T_{N+1} where external data sets are available:

$$\psi_b^n \;=\; \psi_b(T_N) + \frac{\psi_b(T_{N+1}) - \psi_b(T_N)}{\Delta T}(n\Delta t - T_N)\,, \quad T_N \le n\Delta t \le T_{N+1}\,. \tag{4.6}$$

The time integration of the boundary tendency (4.5) is by Marchuk splitting and uses provisional values ψ^* of the variables as initial conditions, which are updated due to all other adiabatic and diabatic processes (see Section 4.3.4). Because the relaxation coefficient μ_b may become large in the boundary zone, an implicit integration scheme is applied. The time discretization reads

$$\psi^{n+1} \;=\; \psi^* - 2\Delta t\mu_b\left(\psi^{n+1} - \psi_b^{n+1}\right)\,, \tag{4.7}$$

which is rewritten in the form

$$\psi^{n+1} \;=\; \psi^* - \alpha_b\left(\psi^* - \psi_b^{n+1}\right)\,, \tag{4.8}$$

where

$$\alpha_b \;=\; \frac{2\Delta t\mu_b}{1 \;+\; 2\Delta t\mu_b} \tag{4.9}$$

is a normalized attenuation function for lateral boundary relaxation. $\alpha_b = 1$ at the boundary and for all grid points outside the computational domain. Within the boundary zone, α_b decreases gradually and becomes zero in the free interior zone of the model domain such that the predicted values ψ^* are not affected by relaxation.

The values of the relaxation parameter μ_b can be chosen to minimize gravity wave reflection from the lateral boundaries. In LM, we use a formulation of Kallberg (1977) for the attenuation function α_b:

$$\alpha_b \;=\; 1 \;-\; \tanh\left(\frac{d}{2\Delta s}\right)\,. \tag{4.10}$$

Δs is the horizontal grid spacing and d is the distance from the lateral boundary. The values of the attenuation function α_b are calculated separately for the scalar grid points and the u- and v- grid points because of their different distances to the physical boundaries. Clearly, $\alpha_b = 1$ for $d = 0$ and $\alpha_b \to 0$ for $d \gg \Delta s$. Usually, a boundary zone depth of about 8 grid points is significantly affected by relaxation updating.

4.3 Top Boundary Conditions

The upper boundary of the model domain is defined as the top half level with constant computational coordinate $\zeta = 1/2$. According to the coordinate transformations discussed in Section 3.5.3, this level corresponds to a flat surface in physical space with fixed height z_T above mean sea level.

Non-penetrative boundary conditions are imposed at the upper boundary, i.e. the $\zeta = 1/2$ surface is treated as a rigid lid by setting the contravariant vertical velocity $\dot{\zeta}$ to zero. As

the $\zeta = 1/2$ level is flat, the physical vertical velocity also vanishes here:

$$\begin{aligned} \left(\dot{\zeta}\right)_{i,j,k=1/2} &= 0\,, \\ (w)_{i,j,k=1/2} &= 0\,. \end{aligned} \tag{4.11}$$

For the components of the horizontal velocity, the temperature and the water substances, the free-slip condition

$$(\delta_\zeta \psi)_{k=1/2} = 0 \tag{4.12}$$

is imposed. This is equivalent to the no friction, no subgrid scale flux conditions specified in Eq. (3.80). Thus, with (4.11) and (4.12) there is no mass transfer across the upper boundary.

The free-slip condition (4.12) is also applied to calculate the metric correction terms in the diagnostic equation (3.46) for the horizontal wind divergence. In order to evaluate the metric terms of the pressure gradient force in the equations (3.41) and (3.42) for horizontal momentum, an additional boundary condition on the perturbation pressure is required. We impose an extrapolated boundary condition for p', which is based on the assumption of a constant vertical gradient at the upper boundary. This condition may be formulated as

$$\left\{\frac{1}{\sqrt{\gamma}}\delta_\zeta(p')\right\}_{k=1/2} = \left\{\frac{1}{\sqrt{\gamma}}\delta_\zeta(p')\right\}_{k=1+1/2}\,. \tag{4.13}$$

Obviously, the additional boundary condition (4.13) is unnecessary in case of hybrid coordinate system with a number of flat model levels adjacent to the upper boundary, since the metric correction terms vanish there.

4.4 Bottom Boundary Conditions

The lower boundary of the model domain is defined as the bottom half level with constant computational coordinate $\zeta = N_\zeta + 1/2$. According to the coordinate transformation discussed in Section 3.5.3, this level follows the surface terrain with height $z_s = h(\lambda, \varphi)$ in physical space.

For the components of the horizontal velocity, the temperature and the water substances, friction boundary conditions are imposed. The subgrid scale turbulent fluxes are specified in terms of external surface variables and the values of the model variables at the lowest main level above the ground. These conditions have been formulated by Eq. (3.81).

The lower boundary is non-penetrative with respect to grid-scale mass fluxes. Thus, the contravariant vertical velocity must vanish:

$$\left(\dot{\zeta}\right)_{i,j,k=N_\zeta+1/2} = 0\,. \tag{4.14}$$

According to the diagnostic equation (2.112) for $\dot{\zeta}$, the vertical velocity w in case of a flow following the coordinate surfaces is given by

$$w = \frac{J_\lambda}{a\cos\varphi}\,u + \frac{J_\varphi}{a}\,v\,. \tag{4.15}$$

To evaluate (4.15) at the surface, we assume free slip conditions on the horizontal wind velocity. This yields the following discretized form of the lower boundary condition on w:

$$w_{N_\zeta+1/2} = \frac{1}{a\cos\varphi}\left\{\overline{(\delta_\lambda h)\,u_{N_\zeta}}^{\lambda} + \overline{(\delta_\varphi h)\,v_{N_\zeta}\cos\varphi}^{\varphi}\right\} . \tag{4.16}$$

The free-slip condition on u and v used in (4.16) is also applied to calculate the metric correction terms in the diagnostic equation (3.46) for the horizontal wind divergence. In order to evaluate the metric terms of the pressure gradient force in the equations (3.41) and (3.42) for horizontal momentum, an additional boundary condition on the perturbation pressure is required. Similar to the top boundary, we impose an extrapolated boundary condition for p', which is based on the assumption of a constant vertical gradient at the lower boundary. This condition may be formulated as

$$\left\{\frac{1}{\sqrt{\gamma}}\delta_\zeta(p')\right\}_{k=N_\zeta+1/2} = \left\{\frac{1}{\sqrt{\gamma}}\delta_\zeta(p')\right\}_{k=N_\zeta-1/2} , \tag{4.17}$$

where the right hand side is known from model interior values of perturbation pressure.

Numerical experimentation has shown that the model solution is very sensitive to the formulation of the lower boundary condition on pressure, i.e. relatively small changes to the extrapolation scheme may have a large impact on the flow structure near the surface. This is typical for overspecified boundary conditions. We plan to implement an alternative condition on pressure, which is consistent with the kinematic boundary condition (4.16) on the vertical velocity, in a future version of the model.

4.5 Initialization

Initial data for an NWP-model, which are obtained by an intermittent analysis scheme in the assimilation cycle or by interpolation from a coarse grid driving model, typically contain unbalanced information for the mass and wind field. This will give rise to spurious high-frequency oscillations of large amplitude during the first hours of the model integration. Thus, the initial data must be modified to reduce the unbalanced gravity and sound wave components to a realistic level. This process is called initialization. A number of initialization schemes have been developed to control the high-frequency oscillations, a well known example is the implicit normal mode initialization scheme used in many hydrostatic models. These schemes, however, are very complex and their extensions for use in a nonhydrostatic model framework is not straightforward. A relatively simple alternative approach based on time filtering was proposed by Lynch (1990): the high frequencies are removed by applying a digital filter to a short time series of the prognostic model variables obtained by an integration from the initial data.

The theoretical background of digital filtering is described in Lynch and Huang (1992), and shortly summarized as follows. In order to filter out the high-frequency components of a continuous function $f(t)$, one may proceed as follows: (i) calculate the Fourier transform $F(\omega)$ of $f(t)$, (ii) set the Fourier coefficients of the high frequencies to zero, and (iii) calculate the inverse transform of the modified $F(\omega)$. Step (ii) is performed by multiplying $F(\omega)$ by an appropriate weighting function $H(\omega)$. Typically, $H(\omega)$ is a step function

$$H(\omega) = \begin{cases} 1\,, & |\omega| \le |\omega_c|\,; \\ 0\,, & |\omega| > |\omega_c|\,, \end{cases} \tag{4.18}$$

where ω_c is a cutoff frequency (i.e. higher frequencies are to be filtered out). These three steps are equivalent to a convolution of $f(t)$ with $h(t)$, denoted by $(h * f)(t)$. Here, $h(t) = \sin(\omega_c t)/\pi t$ is the inverse Fourier transform of $H(\omega)$. That is, to filter $f(t)$ one calculates the integral

$$f^*(t) = (h * f)(t) = \int_{-\infty}^{+\infty} h(\tau) f(t-\tau) d\tau \,, \tag{4.19}$$

where f^* denotes the filtered input function f. Filtering of a discrete sequence of values of f, $\{f_n\} = \{\ldots, f_{-2}, f_{-1}, f_0, f_1, f_2, \ldots\}$ valid at times $t_n = n\Delta t$ proceeds in the same way by using discrete Fourier transforms operating on the discrete frequency $\theta = \omega\Delta t$. The convolution of the discrete $\{h_n\}$ with $\{f_n\}$ yields

$$f_n^* = (h * f)_n = \sum_{k=-\infty}^{k=+\infty} h_k f_{n-k} \tag{4.20}$$

for the filtered input values of f. h_n is the discrete inverse transform of the step function (4.18) for filtering,

$$h_n = \frac{\sin n\theta_c}{n\pi} \,, \tag{4.21}$$

where $\theta_c = \omega_c \Delta t = 2\pi\Delta t/\tau_c$ denotes the digital cutoff frequency for a given cutoff time period τ_c. In practice, the summation must be truncated at some finite value N of k, which gives rise to Gibbs oscillations. These may be reduced by means of a window function w_n multiplying the weights h_n. An example is the Lanczos window defined by

$$w_n = \frac{\sin\{n\pi/(N+1)\}}{n\pi/(N+1)} \,. \tag{4.22}$$

Thus, an approximation to the low-frequency part of $\{f_n\}$ is given by

$$f_n^* = \sum_{k=-N}^{k=+N} h_k w_k f_{n-k}. \tag{4.23}$$

The discrete convolution (4.23) for the time span $T_s = 2N\Delta t$ is formally identical to a non-recursive digital filter, where the output depends on both past and future values of the input, but not on other output values. Thus, also other digital filters than those defined by (4.21) and (4.22) may be used.

Given a sequence of model values, the application of digital filtering according to (4.23) is straightforward. However, if the filtered values shall be valid at initial time $t = 0$, specific set-ups must be designed such that model variables for times $t < 0$ are generated and that (4.23) can be calculated by running sums during a model run. Lynch and Huang (1992) developed an adiabatic initialization scheme for the HIRLAM model by applying a simple non-recursive filter to a sequence of values of model variables centred on the initial time $t = 0$. These values were generated by two short adiabatic model integrations, one forwards and one backwards from $t = 0$. Huang and Lynch (1993) (hereafter HL93) described a method to incorporate diabatic effects: the backward adiabatic integration is followed by a diabatic forecast of twice the length, and the values generated by the diabatic forward integration are processed by the filter. This results in filtered values valid at initial time $t = 0$. Other initialization techniques such as *finalization* and *launching*, as described in Lynch and Huang (1994)(hereafter LH94), are also possible but then the filtered fields are not applicable at initial time. These methods are not considered here.

LH94 proposed also an alternative method for diabatic initialization by digital filtering: the backward adiabatic integration was processed with a recursive filter, and the output values were used to initiate the forward diabatic integration, to which the filtering is applied again. This method also gives a filtered initial stage valid at time $t = 0$, but requires less CPU-time. Figure 4.1 illustrate the HL93 and LH94 methods for diabatic initialization by digital filtering.

The LH94 filtering approach of filtering both the the backwards and the forwards integrations is also possible using a more efficient non-recursive filter, as shown in Lynch et al. (1997)). This scheme has been implemented in LM. The extension to non-hydrostatic dynamics is very simple, it involves filtering of just two additional model variables – vertical velocity and perturbation pressure. The scheme uses a Dolph-Chebyshev window, and the resulting filter many properties of an optimal filter but is much easier to construct. For details about the filter design and the calculation of the filter weights w_n, see Lynch (1997).

The default set-up of the LM initialization is a diabatic digital filtering scheme as shown in Fig. 4.1b. Optionally, an adiabatic version may be used by specifying the second filter stage as an adiabatic forward integration. The backward integration is realized by simply changing the time step from Δt to $-\Delta t$. For both the forward and backward stage, the same timestep as in the free forecast is used. Care must be taken for horizontal diffusion during the backward integration: here, the sign of the corresponding diffusion coefficients has to be reversed in order to generate a smooth solution. Another problem arises from specifying appropriate boundary conditions for the backward model run. At present, the boundary fields for $t < 0$ are simply set to the initial conditions. Thus, the filter time span T_s should not exceed 1 hour of integration in order to get reasonable results along the boundaries.

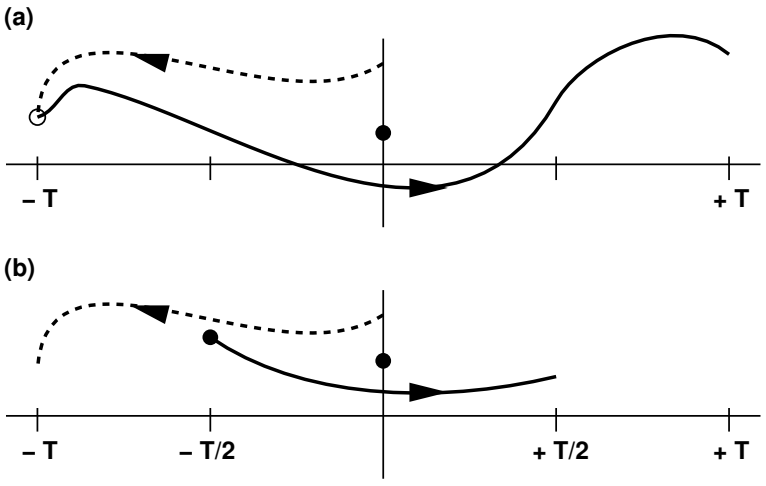

Figure 4.1: Schematic illustration of diabatic digital filtering procedures. Dashed lines indicate the adiabatic backward integration, full lines indicate the diabatic forward integration. (a): HL93 scheme. The adiabatic integration from t=0 to $t = -T$ is not filtered, but its terminal stage is used as initial condition (open circle) for the forward model run from $-T$ to T. The model variables from this run are subject to filtering with a span of $T_s = 2T$, and the output is valid at $t = 0$ (full circle).
(b) LH94 scheme. Both the backward and the forward integration are subject to a filter of span $T_s = T$. Filtering of the backward stage yields output valid at $t = -T/2$ (full circle). These values are used to initiate the forward diabatic integration from $t = -T/2$ to $t = T/2$. Filtering of this stage yields the final initialized variables valid at $t = 0$ (full circle)

.

This, however, implies to set the cutoff period τ_c to a value smaller than 1 hour to achieve reasonable filter characteristics. For the current meso-β-scale applications, we do not intend to filter out time signals with periods larger than 1 hour.

The LM initialization should only be used when starting from interpolated initial conditions, derived e.g. from GME or the ECMWF model. Starting the model from the continuous data assimilation suite based on nudging (see Part III of the documentation) does not require any initialization.

4.6 Interactive Self-Nesting

We plan to implement an additional option for self-nesting which is based on two-way interaction. With such a scheme, a coarse grid model solution is used to start and to provide boundary conditions for a fine grid, which is embedded in the coarse grid. The solution for the fine grid feeds back to the coarse grid, usually in a region where both grids overlap. The coupling of the two solutions is every time step of the coarse grid model solution. This type of nesting is similar to local static grid refinement techniques used in computational fluid dynamics.

A prototype scheme which is based on the nesting strategy as used in the MM5-model Grell et al. (1994) has been implemented in a test version of LM. The nesting proceeds in three major steps. First, at a given time t with solutions on both the coarse and fine grid, the coarse grid model solution is integrated for one coarse grid timestep Δt. Second, the prognostic variables on the coarse grid are interpolated to the fine grid boundaries and for the time interval $(t, t+\Delta t)$. The fine grid model solution is then advanced from t to Δt using several smaller time steps. For a grid aspect ration of 3, the small time step size is $\Delta t/3$. And third, the coarse grid variables are finally replaced with averaged fine grid values in the overlap region. The MM5 scheme uses a monotonic interpolation algorithm Smolarkiewicz and Grell (1992) for the interpolation to the fine grid. For the feedback stage, the fine grid values are directly inserted to the coarse grid points. A detailed description of the method is given by Grell et al. (1994).

A specific feature of the implementation is that multiple fine grids can be used. The first option is to use a number of fine grids of the same resolution within the coarse grid domain (but the fine grids may not overlap). The second option is to use telescoping grids which are successively nested into each other. The scheme allows also the use of high-resolution external parameters (e.g. topography, land-sea mask, soil-type etc.) within the fine grids, except for a transition zone along the lateral boundaries. At present, the prototype implementation is tested at HNMS. Thereafter, the scheme will be transfered to the model source code. A comprehensive description of the 2-way nesting scheme will be given in a future version of this documentation.

4.7 Spectral Nudging

Spectral nudging von Storch et al. (2000) is an approach to force the regional model to follow the steering provided by the driving model. In the “standard" approach Davies (1976) in current use, the steering takes place exclusively along the lateral boundaries in the spirit

of a classical boundary value problem. In the “spectral nudging" approach, the atmospheric state inside the integration area is also forced to accept the large-scales of the driving model whereas smaller scales are left to be determined by the regional model.

In the “spectral nudging" approach, the lateral “sponge forcing" is kept and an additional steering is introduced as described next.

Consider the expansion of a suitable COSMO quantity Ψ:

$$\Psi(\lambda,\phi,t) = \sum_{j=-J_m,k=-K_m}^{J_m,K_m} c^m_{j,k}(t) e^{ij\lambda/L_\lambda} e^{ik\phi/L_\phi} \tag{4.24}$$

with zonal coordinates λ, zonal wave-numbers j and zonal extension of the area L_λ. Meridional coordinates are denoted by ϕ, meridional wave-numbers by k, and the meridional extension by L_ϕ. t represents time. For COSMO, the number of zonal and meridional wave-numbers is J_m and K_m. A similar expansion is done for the driving model, which are given on a coarser grid. The coefficients of this expansion are labeled $c^a_{j,k}$, and the number of Fourier coefficients is $J_a < J_m$ and $K_a < K_m$. The confidence we have in the realism of the different scales of the re-analysis depends on the wavenumbers j and k and is denoted by $\eta_{j,k}$.

The model is then allowed to deviate from the state given by the re-analysis conditional upon this confidence. This is achieved by adding “nudging terms” in the spectral domain in both directions

$$\sum_{j=-J_a,k=-K_a}^{J_a,K_a} \eta_{j,k}(c^a_{j,k}(t) - c^m_{j,k}(t)) e^{ij\lambda/L_\lambda} e^{ik\phi/L_\phi} \tag{4.25}$$

In the following, we will use the nudging terms dependent on height. That is, our confidence in the reanalysis increases with height. On the other hand, we leave the regional model more room for its own dynamics at the lower levels where we expect regional geographical features are becoming more important. The better the confidence, the larger the $\eta_{j,k}$-values and the more efficient the nudging term.

Following F. et al. (1993) we use a height dependence nudging coefficient.

$$\eta^0(p) = \begin{cases} \alpha\left(1 - \frac{p}{p_{max}}\right)^2 & \text{for} \quad p < p_{max} \\ 0 & \text{for} \quad p > p_{max} \end{cases} \tag{4.26}$$

with p denoting pressure. We set $\eta_{j,k} = \eta^0$ for $j = 0 \ldots j_{max}$ north-south direction, $k = 0 \ldots k_{max}$ in the east-west direction and $\eta_{j,k} = 0$ otherwise. j_{max} and k_{max} are the maximum wave numbers which are determined by

$$\begin{aligned} j_{max} &= \frac{N_{\varphi m}\Delta x_m}{N_{amax}\Delta x_a} \\ k_{max} &= \frac{N_{\lambda m}\Delta x_m}{N_{amax}\Delta x_a} \end{aligned} \tag{4.27}$$

where $N_{\varphi m}$, $N_{\lambda m}$ are the number of COSMO grid points in meridional and zonal direction, respectively. Δx_m is the maximum grid width of the specific COSMO configuration and Δx_a that one for the driving model. N_{amax} denotes the minimum number of grid points needed to resolve a wave (i.e. five in general, see Pielke (2002)).

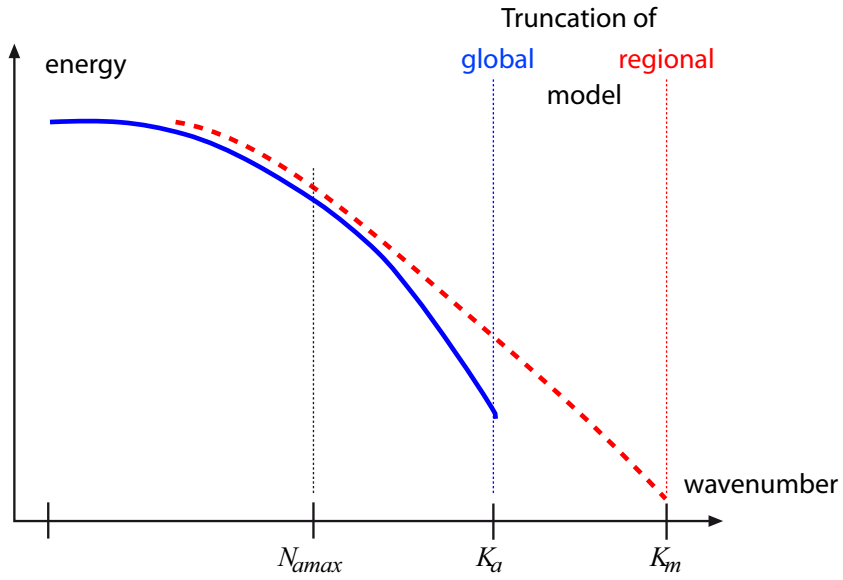

Figure 4.2: Schematic sketch of the energy spectrum of global and regional models.

Spectral nudging is switched off by default. It may be controlled by the following options in namelist GRIBIN:

Name	Type	Definition / Purpose / Comments	Default
`lspecnudge`	`Logical`	switch for spectral nudging	`.FALSE.`
`yvarsn`	`CHAR`	list of quantities for spectral nudging	`'U ','V '`
`isc_sn`	`INT`	wave numbers i-direction, i.e. k_{max}	2
`jsc_sn`	`INT`	wave numbers i-direction, i.e. j_{max}	2
`pp_sn`	`REAL`	lowest pressure level in hPa for spectral nudging, i.e. p_{max}	850.
`alpha_sn`	`REAL`	amplification factor for spectral nudging, , i.e. α (0. <=alpha_sn<= 1.)	0.05

Section 5

Numerical Smoothing

The subgrid scale fluxes of momentum, heat and moisture provide a spatial smoothing of the prognostic fields. However, this physical type of smoothing is related to turbulent processes and is efficient only in regions where sufficiently strong turbulence occurs. In regions with stable thermal stratification the turbulent fluxes are zero or negligible small and there is no turbulent mixing. For these cases a small amount of additional mixing is required to prevent the initiation and the subsequent growth of nonlinear instabilities and to control small-scale computational noise at or near the two grid interval ($2\Delta x$) wavelength by artificial damping. Numerical noise of such type is continuously generated by numerical dispersion, especially within the Leapfrog time integration using centred difference operators for advection. Also, weak nonlinear numerical instabilities resulting from aliasing can contribute to the generation of small-scale noise, and physical processes, which in general act discontinuously at single grid points, will always introduce noise at the $2\Delta x$-scale. The formation of non-resolvable small-scale structures will also arise from topographical forcing and from surface inhomogeneities.

Artificial damping can be achieved by modifying the diffusion coefficients for turbulent mixing. In the model turbulent mixing is formulated to operate along the coordinate lines of the orthogonal z-system (see Part II of the LM documentation). Therefore, this method is referred to as background mixing in physical space. Another technique to control small-scale noise is to introduce an additional artificial mixing term on the right hand side of the prognostic equations. This method is referred to as computational mixing, which in LM is formulated to operate along the coordinate lines of the nonorthogonal ζ- system.

To ensure numerical stability, all computational mixing terms are evaluated by a forward-in-time integration scheme in the context of Marchuk-splitting (see Section 4.3.4), since the explicit evaluation at time-level $n-1$ was found to destabilize the solution when the model is run close to the advective CFL-limit. Thus, the time integration of these terms is only conditionally stable and the constraints on the time step size imposed by artificial mixing can be more severe than by other processes. In LM the stability of the time integration is automatically ensured by limiting the values of the coefficients for computational diffusion according to the corresponding stability criterion for the specified model time step.

In the following subsections the time filter to avoid high frequency oscillations and various options on spatial numerical smoothing are discussed. Additionally, a Rayleigh damping formulation is described, that has been implemented to suppress gravity wave reflection at the upper boundary of the model.

5.1 Time filter

When all adiabatic and diabatic processes have been treated by the time integration procedure at time level t, the time step is complete and the new values of the prognostic model variables at time $t + \Delta t$ are known. A time filter is then applied to the variables at the mid-time level t of the leapfrog scheme. The purpose of this filter is to control rapid oscillations and instabilities of the computational mode, which are generated by the leapfrog integration. Thereby, the temporal decoupling of the physical and the computational modes of the numerical solution is avoided.

In LM we use the Robert-Asselin time filter (Robert (1966); Asselin (1972), which is defined as follows:

$$\tilde{\psi}^n = \psi^n + \varepsilon_A(\psi^{n+1} - 2\psi^n + \tilde{\psi}^{n-1}). \tag{5.1}$$

ψ represents one of the prognostic model variables and the tilde denotes the filtered value of ψ. The $n-1$, n and $n+1$ superscripts indicate the corresponding time levels $t - \Delta t$, t and $t + \Delta t$, respectively. ε_A denotes the Asselin filter coefficient. Both too large (e.g. $\varepsilon_A = 0.25$) and too small a value (e.g. $\varepsilon_A = 0.001$) may damp the physical mode excessively. Moderate values of the filter coefficient in the range $0.05 \leq \varepsilon_A \leq 0.2$ result in a strong damping of the computational mode with only a small impact on the physical mode of the numerical solution. The value $\varepsilon_A = 0.15$ is used as a default in LM.

When the time integration scheme is run very close to the stability limit for the large time step, the Robert-Asselin filter can destabilize the numerical solution Déqué and Cariolle (1986). In these cases, either the value of the filter coefficient or of the time step has to be reduced to recover stability. For practical applications, a reduction of the large time step is recommended.

Time filtering does not only damp the computational mode efficiently and prevents a temporal decoupling of the numerical solution, but is also of crucial importance for a stabilization of the time-split algorithm which is used for the integration of the equations. Skamarock and Klemp (1994) have shown that the interaction between propagating and advecting acoustic modes can introduce severe constraints on the maximum allowable time step. These constraints are removed by applying the time filter (5.1). It serves to stabilize the time-split method and masks the fact that the stability of the individual small and large time-step schemes does not ensure the stability of the coupled integration scheme.

The time filter (5.1) is applied to all prognostic model variables, i.e. the thermodynamic variables u, v, w, T, p' and the concentrations of the water constituents q^v, q^l and q^f. In cloudy regions, the application of the filter can disrupt the saturation equilibrium, which is imposed as a closure condition to calculate condensation and evaporation of cloud water by a saturation adjustment method (see Part II of the LM documentation). Thus, subsequent to time filtering, the saturation adjustment scheme is applied once again to the filtered variables.

5.2 Computational Mixing

Numerical smoothing is designed to remove shortwave grid-scale noise from the fields. Such small scale noise is usually of computational origin and is introduced by, e.g., the advection

scheme which generates overshoots and undershoots, or the numerical representation of the energy cascade which is related to nonlinear instability. Computational mixing is therefore designed as a spatial filter operating on the sloping coordinate lines of the ζ-system and not as a mathematical representation of a physical diffusion process. The latter would require a complete transformation of the Laplace operator to the ζ-system to take the direction of diffusive fluxes in physical space correctly into account. This is computationally expensive. Thus, the formulation of the computational mixing terms is done with a pseudo-orthogonal diffusion operator. In case of high-resolution modelling with steep slopes of the model layers, however, a 2-D numerical diffusion along terrain-following model layers will imply a large but unwanted vertical mixing (e.g. between mountain tops and narrow valleys), especially in an atmosphere with strong stratification of the thermodynamic variables. To tackle this problem, a simple orographic flux-limiter has been implemented which reduces numerical diffusion with increasing steepness of the topography.

LM offers three options for 4th-order computational mixing: (i) regular linear horizontal diffusion, (ii) a monotonic version of the horizontal diffusion scheme, and (iii) the monotonic version with a simple flux-limiter to reduce numerical diffusion with increasing steepness of the topography. These schemes are described in the following subsections. Vertical computational mixing is not considered by these schemes, but some vertical smoothing is done by background mixing in the physical space (see Section 6.3). It is important to note that the computational mixing should be as small as possible because of its non-physical origin. Otherwise the physical solution may be affected in a detrimental way.

5.2.1 Fourth order Horizontal Diffusion

Higher order computational mixing is mostly realized by introducing an additional operator to the right hand side of the prognostic equations:

$$\frac{\partial \psi}{\partial t} = S(\psi) + (-1)^{m/2+1} \alpha_m \nabla^m \psi \,, \tag{5.2}$$

where ψ is one of the prognostic variables and S represents all physical and dynamical source terms for ψ. The last term in Eq.(5.2) is the added linear diffusion term where m ($m = 2, 4, 6, \cdots$) denotes the (even) order of computational diffusion and α_m is the corresponding diffusion coefficient of order m.

The higher the order of the diffusion operator, the more scale selectively wavelength components are damped. Figure 5.1 shows the amplification factor (here: damping factor) for a one dimensional version of Eq. (5.2) using a forward in time and centred in space finite difference scheme for integration, plotted against the normalized wavelength $k\Delta x$ where k is the wavenumber and Δx is the grid spacing. For each of the second, fourth and sixth order schemes the corresponding diffusion coefficient α_m is chosen so that the shortest waves representable on a discrete grid ($2\Delta x$, i.e. $k\Delta x = \pi$) are damped completely within one integration time step.

As can be seen from Fig. 5.1, all schemes damp not only the two grid interval wavelength, but also longer wave components in a wave spectrum. For example, the amplitude of a $4\Delta x$ wave is reduced by a factor of 0.5 in the second order scheme, and longer waves are also damped significantly. The fourth and sixth order scheme have much less damping for these longer waves, which contain physical meaningful information in a simulation and should not

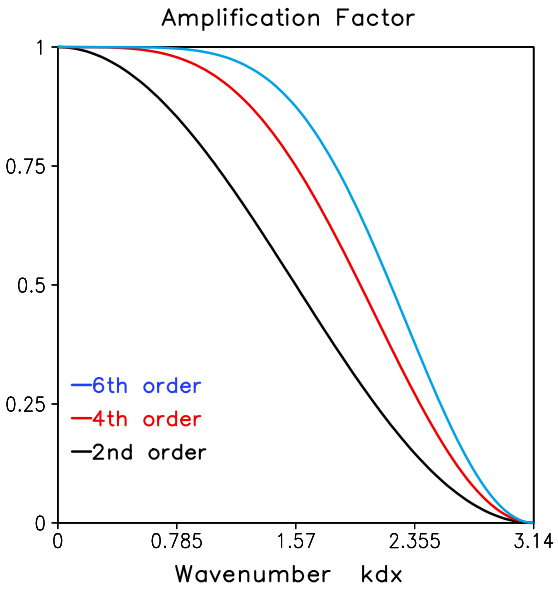

Figure 5.1: Amplification factor plotted against the normalized wavenumber.

be affected by a numerical filter. Clearly, higher order schemes provide a more scale selective damping.

Many models use the 4th-order linear computational diffusion scheme because of its relative computational efficiency compared to the 6th order scheme and of its superior damping properties compared to the second order scheme. The 4th-order linear scheme is also applied in LM and we do not plan to implement a 6th or even higher order filter. Schemes of higher than 2nd order, however, have unwanted side effects. The 2nd order scheme has a Laplacian form and thus is always monotonic: by transferring higher values of the field into regions with smaller values no new minima or maxima can be created. This is not the case for the 4th, 6th and even higher order schemes, where spatial oscillations are introduced (Gibbs phenomenon), as can be seen in Fig. 5.2.

Figure 5.2 shows the one dimensional numerical solution of Eq.(5.2)(with $S = 0$) for 2nd, 4th and 6th order diffusion applied to an initial rectangular structure. The second order scheme (left) is strongly diffusive even for the large scale structure. At the end of the integration ($t = 100$) the amplitude of the entire square wave is significantly reduced. The 4th (middle) and 6th order (right) schemes retain the initial structure much better, and at the lateral edges, where short wavelengths components dominate, the sharp gradients are maintained in the solution (somewhat better by using the 6th order scheme).

However, over- and undershoots are generated by the numerical solution. The amplitude of these unphysical, both positive and negative spatial oscillations is increased by increasing the order of the diffusion operator. That is, going to higher than 4th-order diffusion will not cure this side effect. Obviously, the $2\Delta x$ waves are removed from the solution, but at the cost of *introducing new noise on the resolvable scales*. In the context of a full simulation model these oscillations will interact nonlinearly with the physics as well as with the dynamics (as they are on the resolvable scales) and thus can result in unforeseeable effects. Moreover, negative oscillations are unacceptable for positive definite fields such as cloud water content or specific humidity.

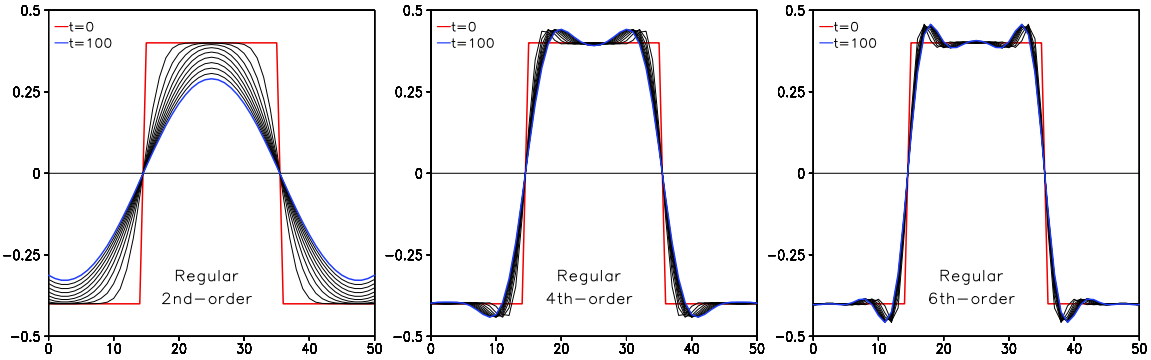

Figure 5.2: 1-D numerical solution of Eq.(1) for an initial square wave (red, $t = 0$) in case of 2nd (left), 4th (middle) and 6th order (right) linear diffusion. The domain has 51 grid points with a grid spacing of 1/50 m and the time step is 1 sec. The diffusion coefficients are chosen such that there is full damping of a $2\Delta x$ wave within one time step, i.e. $\alpha_m = (\Delta x/2)^m/\Delta t$. The solutions are plotted at 10 timestep intervals (thin lines).

Before we present a monotonic version of the 4th-order scheme for computational mixing in the following subsection, the implementation of the regular 4th-order horizontal diffusion is described. Let M_ψ^{CM} denote the computational mixing term of a model variable ψ. The evaluation of this term according to (5.2) for quasi-horizontal diffusion along model surfaces in lat/lon coordinates takes the form

$$M_\psi^{CM} = -\alpha_4 \frac{1}{a^2 \cos^2\varphi} \left\{ \frac{\partial^2 s_\psi}{\partial \lambda^2} + \cos\varphi \frac{\partial}{\partial \varphi} \left(\cos\varphi \frac{\partial s_\psi}{\partial \varphi} \right) \right\} \tag{5.3}$$

with

$$s_\psi = \frac{1}{a^2 \cos^2\varphi} \left\{ \frac{\partial^2 \tilde{\psi}}{\partial \lambda^2} + \cos\varphi \frac{\partial}{\partial \varphi} \left(\cos\varphi \frac{\partial \tilde{\psi}}{\partial \varphi} \right) \right\}$$

where α_4 is the diffusion coefficient for 4th-order mixing and $\tilde{\psi}$ is the diffused quantity. $\tilde{\psi} = \psi$ for all model variables except for temperature with $\tilde{T} = T - T_0$. This so-called slope correction for smoothing of temperature – where instead of the actual temperature the difference of temperature and a reference profile is diffused – tries to reduce the spurious biases resulting from quasi-horizontal mixing along steep model levels. Also, a reduction factor of the diffusion coefficient α_4 may be specified to apply a smaller smoothing for temperature and humidity. In the operational version of LM, we use a reduction factor of 0.75 for temperature and pressure diffusion, and of 0.5 for the diffusion of specific humidity and cloud water content.

To discretize (5.3) in finite difference form, it is convenient to define the nondimensional Laplace operator

$$\hat{\nabla}^2 \equiv \Delta\lambda^2 \delta_\lambda \left(\delta_\lambda \tilde{\psi} \right) + \cos\varphi \left(\frac{\Delta\lambda}{\Delta\varphi} \right)^2 \Delta\varphi^2 \delta_\varphi \left(\cos\varphi \delta_\varphi \tilde{\psi} \right) \tag{5.4}$$

with the horizontal finite difference operators δ_λ and δ_φ as in Section 4. The discretized form of the 4th-order horizontal diffusion term is now written as

$$M_\psi^{CM} = -\hat{\alpha}_4 \hat{\nabla}^2 \left(\hat{\nabla}^2 \tilde{\varphi} \right) \tag{5.5}$$

where

$$\hat{\alpha}_4 = \alpha_4/(a^4 \cos^4\varphi \Delta\lambda^4)$$

is a the normalized 4th-order diffusion coefficient of dimension 1/s. The calculation of the mixing terms given in (5.5) is straightforward by a two-step procedure. First, the nondimensional Laplacian of $\tilde{\psi}$, $\hat{\nabla}^2\tilde{\psi}$, is evaluated at interior gridpoints and at gridpoints just outside the physical boundary. This can be done without additional artificial boundary conditions, because the LM model domain is defined to have two rows/columns of gridpoints adjacent to each lateral boundary of the computational domain (see Section 4). In a second step, the Laplacian is then applied to the intermediate field $\hat{\nabla}^2\tilde{\psi}$ according to (5.5). The evaluation of these terms can now be done at all gridpoints inside the physical boundary.

The 4th-order horizontal diffusion terms are integrated in time using the explicit Euler-forward scheme. The corresponding linear stability analysis reveals the following condition for stable and nonoszillating solutions of symmetrical waves ($\Delta\lambda = \Delta\varphi$) at the equator of the rotated grid ($\cos\varphi = 1$):

$$0 \leq 128\Delta t\hat{\alpha}_4 \leq 1, \tag{5.6}$$

For a specified time step Δt, $\hat{\alpha}_4$ has to be limited according to (5.6) for stability reasons. In LM, the default value of the normalized diffusion coefficient $\hat{\alpha}_4$ is set to

$$\hat{\alpha}_4 = \frac{1}{2\pi^4\Delta t}. \tag{5.7}$$

This value results in a decrease of the amplitudes of the shortest resolvable waves with wavelengths $2\Delta\lambda$ and $2\Delta\varphi$ by about a factor of 1/3 within every large time step Δt. Other values may be specified by the user.

5.2.2 A Monotonic Diffusion Operator

In order to avoid the over- and undershoots generated by linear higher order diffusion, Xue (2000) proposed a scheme with a simple flux limiter. However, practical applications showed that this scheme is not strictly monotonic. For LM a new scheme based on multi-dimensional flux limiting has been developed Doms (2001). The formulation of the limiter factor closely follows the approach of Smolarkiewicz (1989).

The basic pre-requisite to derive the scheme is to rewrite the diffusion operator in Eq.(5.2) in a flux form. Omitting the source term S, the diffusive tendency then reads

$$\frac{\partial\psi}{\partial t} = (-1)^{m/2+1}\alpha_m\nabla^m\psi = -\nabla\cdot\mathbf{F}\,, \tag{5.8}$$

where $\mathbf{F}$ denotes the diffusive flux being defined by

$$\mathbf{F} \equiv (-1)^{m/2}\alpha_m\nabla(\nabla^{m-2}\psi)\,. \tag{5.9}$$

For 4th-order diffusion and in a rotated geographical coordinate system (λ, φ), the longitudinal and meridional flux components of $\mathbf{F}$ are given by

$$\mathbf{F}^\lambda = \frac{\alpha_4}{a\cos\varphi}\frac{\partial s_\psi}{\partial\lambda}\,, \quad \mathbf{F}^\varphi = \frac{\alpha_4}{a}\frac{\partial s_\psi}{\partial\varphi}\,, \tag{5.10}$$

with s_ψ as in Eq. (5.3). The numerical scheme for integrating the 2-D version of the diffusion equation (5.8) using a forward in time and centred in space scheme reads

$$\psi_{i,j}^{n+1} = \psi_{i,j}^{n} - (A_{i+1/2,j}^{n} - A_{i-1/2,j}^{n} + A_{i,j+1/2}^{n} - A_{i,j-1/2}^{n}), \tag{5.11}$$

where n denotes the time level, (i,j) are the gridpoint indices with i counting in λ-direction and j in φ-direction, and

$$A_{i\pm1/2,j}^{n} \equiv (F_{i\pm1/2,j}^{\lambda})^{n} \frac{\Delta t}{a\cos\varphi_j \Delta\lambda}, \quad A_{i,j\pm1/2}^{n} \equiv (F_{i,j\pm1/2}^{\varphi})^{n} \frac{\cos\varphi_{j\pm1/2}\Delta t}{a\cos\varphi_j \Delta\varphi}, \tag{5.12}$$

are the normalized 4th-order diffusive fluxes at cell interfaces calculated from Eq. (5.10). In case of a leapfrog time integration, Δt is replaced by $2\Delta t$ and the time levels have to be changed accordingly. The numerical scheme (5.11) can be rewritten as

$$\psi_{i,j}^{n+1} = \psi_{i,j}^{n} + A_{i,j}^{in} - A_{i,j}^{out}, \tag{5.13}$$

with $A_{i,j}^{in}$ denoting the sum of all fluxes into a grid cell and $A_{i,j}^{out}$ denoting the sum of all fluxes going out of a grid cell. Both the sum of the incoming and outgoing fluxes, $A_{i,j}^{in}$ and $A_{i,j}^{out}$, respectively, are defined to be positive. Using the notation

$$A_{.,.}^{+} = \max(A_{.,.}^{n}, 0), \quad A_{.,.}^{-} = \min(A_{.,.}^{n}, 0) \tag{5.14}$$

to indicate positive and negative fluxes at the cell interfaces, the sum of the incoming and outgoing fluxes is given by

$$\begin{aligned} A_{i,j}^{in} &= -A_{i+1/2,j}^{-} + A_{i-1/2,j}^{+} - A_{i,j+1/2}^{-} + A_{i,j-1/2}^{+}, \\ A_{i,j}^{out} &= +A_{i+1/2,j}^{+} - A_{i-1/2,j}^{-} + A_{i,j+1/2}^{+} - A_{i,j-1/2}^{-}. \end{aligned}$$

The difference $A_{i,j}^{in} - A_{i,j}^{out}$ is equivalent to the discretized total flux divergence. Since a monotonic numerical scheme shall not produce new minima or maxima within a time step, the monotonicity constraint is

$$\psi_{i,j}^{min} \le \psi_{i,j}^{n+1} \le \psi_{i,j}^{max}, \tag{5.15}$$

where $\psi_{i,j}^{min}$ is the minimum and $\psi_{i,j}^{max}$ is the maximum of the field variable ψ at the point (i,j) and the adjacent grid points at time level n:

$$\begin{aligned} \psi_{i,j}^{min} &= \min(\psi_{i,j}^{n}, \psi_{i+1,j}^{n}, \psi_{i-1,j}^{n}, \psi_{i,j+1}^{n}, \psi_{i,j-1}^{n}), \\ \psi_{i,j}^{max} &= \max(\psi_{i,j}^{n}, \psi_{i+1,j}^{n}, \psi_{i-1,j}^{n}, \psi_{i,j+1}^{n}, \psi_{i,j-1}^{n}). \end{aligned}$$

Using the monotonicity constraint (5.15) in the numerical scheme (5.13) yields

$$\psi_{i,j}^{min} \le \psi_{i,j}^{n} + A_{i,j}^{in} - A_{i,j}^{out} \le \psi_{i,j}^{max}. \tag{5.16}$$

Thus, the following conditions on the sum of the incoming and outgoing fluxes are sufficient to ensure monotonicity:

$$\begin{aligned} A_{i,j}^{in} &\le \psi_{i,j}^{max} - \psi_{i,j}^{n}, \\ A_{i,j}^{out} &\le \psi_{i,j}^{n} - \psi_{i,j}^{min}. \end{aligned}$$

With these conditions, the sum of all incoming fluxes cannot add more mass within one time step than that required to reach the maximum value $\psi_{i,j}^{max}$ and the sum of all outgoing fluxes

cannot remove more mass than that to fall to the minimum value $\psi_{i,j}^{min}$. These conditions can be rewritten in the form of flux limiter ratios

$$\begin{aligned} \beta_{i,j}^{in} &= \frac{\psi_{i,j}^{max} - \psi_{i,j}^{n}}{A_{i,j}^{in} + \epsilon}, \\ \beta_{i,j}^{out} &= \frac{\psi_{i,j}^{n} - \psi_{i,j}^{min}}{A_{i,j}^{out} + \epsilon}, \end{aligned} \tag{5.17}$$

where ϵ is a small number to avoid division by zero in the computer program. Whenever $\beta_{i,j}^{in} \geq 1$, the total incoming flux does not need to be modified. But if $\beta_{i,j}^{in} < 1$ the incoming fluxes are overestimated and the flux limiter has to be applied. Similar, if $\beta_{i,j}^{out} \geq 1$, the total outgoing fluxes will ensure monotonicity, but if $\beta_{i,j}^{out} < 1$, the outgoing fluxes are overestimated and have to be limited.

Since the total incoming and outgoing fluxes are additively composed of the fluxes at the cell interfaces, the limiter (5.17) is applied for each of these fluxes. And as the limiter for the incoming fluxes has been constructed independently of the limiter for the outgoing fluxes and because the outgoing flux at a cell interface is equivalent to the incoming flux at the adjacent cell, it is sufficient to simply take the minimum of both limiters. The application of the limiters (5.17) to the fluxes at the cell interfaces in the i- and j-direction can be written in a compact manner:

$$\begin{aligned} \tilde{A}_{i+1/2,j}^{n} &= \min\left(1, \beta_{i,j}^{out}, \beta_{i+1,j}^{in}\right) A_{i+1/2,j}^{+} + \min\left(1, \beta_{i+1,j}^{out}, \beta_{i,j}^{in}\right) A_{i+1/2,j}^{-}, \\ \tilde{A}_{i,j+1/2}^{n} &= \min\left(1, \beta_{i,j}^{out}, \beta_{i,j+1}^{in}\right) A_{i,j+1/2}^{+} + \min\left(1, \beta_{i,j+1}^{out}, \beta_{i,j}^{in}\right) A_{i,j+1/2}^{-}. \end{aligned} \tag{5.18}$$

For example, if the flux $A_{i+1/2,j}$ at the cell interface separating cell (i,j) and $(i+1,j)$ is positive, i.e. is an outgoing flux from cell (i,j) and thus an incoming flux for cell $(i+1,j)$, it will be limited with the minimum of the limiter $\beta_{i,j}^{out}$ of cell (i,j) and the limiter $\beta_{i+1,j}^{in}$ from cell $(i+1,j)$. By replacing the normalized fluxes A in the time integration scheme (5.11) with the limited normalized fluxes $\tilde{A}$ from Eq. (5.18), a monotonic diffusion scheme is finally obtained.

Figure 5.3 shows the solution of the one-dimensional square wave test using the 4th and 6th order monotonic scheme based on direct flux limiting according to (5.18). When compared to the solution of the regular higher order schemes as shown in Figure 5.2, the unwanted numer-

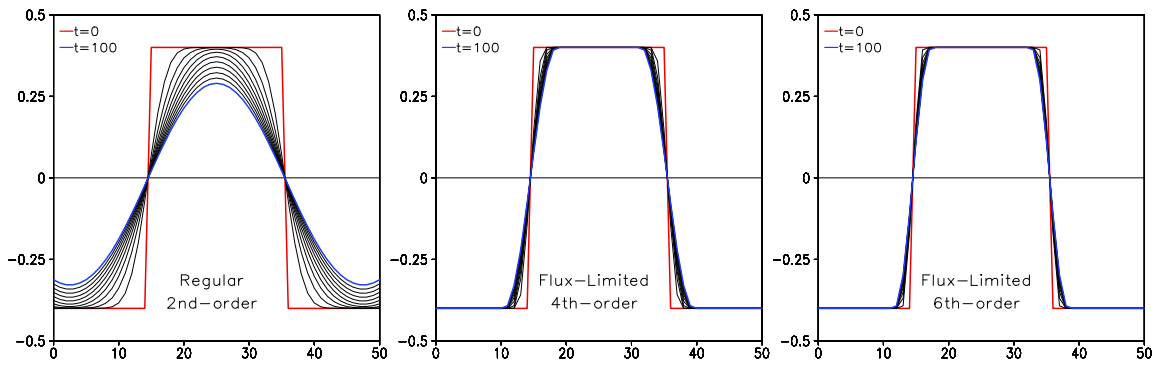

Figure 5.3: As in Fig. 2, but for 4th (middle) and 6th order (right) flux-limited monotonic linear diffusion.

ical effects associated with the Gibbs phenomena are no longer present, but the smoothing characteristic is maintained.

5.2.3 Fourth-order Horizontal Diffusion with Orographic Flux Limiting

Most mesoscale NWP models use a terrain-following sigma-type vertical coordinate and numerical smoothing is realized as horizontal diffusion, i.e. along quasi-horizontal surfaces of constant vertical coordinate. Over complex terrain, this may lead to systematic numerical biases which are induced by horizontal diffusion, as an unwanted vertical mixing in physical space will be implied. Clearly, the impact of this type of numerical error will increase with increasing steepness of the topography and is thus more noticeable in models with very high resolution.

For instance, if horizontal diffusion is applied directly to temperature in the prognostic equation for temperature, the smoothing will tend to cool the valleys and to heat the mountain tops in any stratification where the temperature decreases with height. Similarly, in an atmosphere with decreasing specific humidity with height, horizontal diffusion will dry the valley ground and moisten the adjacent mountains, which can have an unwanted positive feedback to precipitation formation over mountain tops.

Various modifications have been proposed to reduce the spurious biases resulting from quasi-horizontal mixing. Most models use a so-called slope correction for horizontal smoothing in the temperature equation, where instead of the actual temperature the difference of temperature and a reference profile is diffused. In LM, we use the model base state for this purpose. Errors from unwanted vertical mixing are then zero when the actual vertical temperature gradient is identical to the gradient of the base-state profile. However, when the profiles deviate from each other, large error may still be introduced (e.g. in very stable or unstable stratification). Slope corrections for other variables are normally not used.

Another popular method is the use of "true" horizontal diffusion, i.e. along surfaces of constant height or constant pressure (e.g. Ballard and Golding (1991), Zängl (2000)). Here, the values of the diffused variables at the adjacent gridpoints are interpolated to the height or pressure level of each gridpoint before diffusion is applied. Since the gridpoints usually have different heights, however, the diffusive fluxes at the cell interfaces will differ for neighboring gridpoints. Thus, mass conservation cannot be guaranteed with such a scheme. One could also try to locally reduce the value of the diffusion coefficient depending on the steepness of the orography (a commonly used measure is the Laplacian of topographical height). But here again the diffusive fluxes at the cell interface between two neighboring gridpoints will be different and mass can be artificially generated or lost.

Numerical smoothing schemes which introduce uncontrollable sources or sinks in momentum, heat or water mass are problematic, especially in an operational NWP model. With the flux-limited scheme discussed in the preceding Section, we have the opportunity to reduce the topography-induced biases without violating mass conservation: One can simply reduce the diffusive fluxes at the cell interfaces, depending on the steepness of the model surfaces. As the scheme is in flux-form, mass will be conserved automatically. And if the fluxes are artificially reduced before the monotonic flux limiters are applied, the monotonicity of the diffused field will also be guaranteed.

Various functional forms for orographic flux limiters have been tested. To keep things simple,

a simple quadratic form has been implemented, where the fluxes are gradually reduced with increasing steepness of the terrain-following coordinate surfaces and become zero when a threshold value for the height difference between neighboring grid points is exceeded. Defining this height differences at cell interfaces and denoting them by

$$\Delta h_{i+1/2,j} = |h_{i+1,j} - h_{i,j}|\,, \quad \Delta h_{i,j+1/2} = |h_{i,j+1} - h_{i,j}|\,, \tag{5.19}$$

where $h_{i,j}$ is the geometrical height of a grid point at a surface of constant vertical coordinate, the limiter function for the diffusive fluxes A in Eq. (5.11) is written as

$$\begin{aligned} A^n_{i+1/2,j} &= \max\{\,0\,, 1 - (\Delta h_{i+1/2,j}/H_{max})^2\} \cdot (F^{\lambda}_{i+1/2,j})^n \frac{\Delta t}{a\cos\varphi_j \Delta\lambda}\,, \\ A^n_{i,j+1/2} &= \max\{\,0\,, 1 - (\Delta h_{i,j+1/2}/H_{max})^2\} \cdot (F^{\varphi}_{i,j+1/2})^n \frac{\cos\varphi_{j+1/2}\Delta t}{a\cos\varphi_j \Delta\varphi}\,. \end{aligned} \tag{5.20}$$

Here, H_{max} is a threshold value for the maximal height difference where the diffusive fluxes become zero. In the operational version of LM, the value $H_{max} = 250$m is specified by default for the 7 km grid spacing. This value has to be adjusted for other horizontal resolutions.

Tests with the the orographic limiter (5.20) applied to the monotonic diffusion scheme described in Section 6.2.2 revealed a significant reduction of topography-induced biases and a noticeable improvement to the structure of the precipitation field above the Alps. Parallel test experiments with the modified horizontal diffusion scheme have been conducted at MeteoSwiss and at DWD for longer time periods. The statistical evaluation against observed surface weather parameters revealed a neutral impact of the new smoothing scheme to most parameters, but a beneficial impact to the predicted precipitation, especially in mountainous areas.

5.3 Background Mixing in Physical Space

This type of additional artificial mixing operates in physical space and thus becomes part of the parameterization scheme for turbulent processes. Background mixing is included by adding a user-specified constant to the diffusion coefficients for turbulent fluxes, or, with a similar effect, by setting a constant lower limit for the diffusion coefficients. The latter method is applied in LM.

Because turbulent mixing operates on the total fields, the impact of a background mixing coefficient is to diffuse also the base-state temperature and density stratification. This property may not be desirable in most cases and the coefficient for background diffusion should be chosen as small as possible. For too large values of the coefficient, the model will tend to establish a uniform adiabatic stratification, which masks and finally destroys the physical solution.

In the present version of LM, horizontal turbulent fluxes are neglected and only the dominating vertical turbulent fluxes are considered. Consequently, background mixing operates in the vertical direction only. The calculation of the coefficients for turbulent vertical diffusion of momentum and heat, which are denoted by K^v_m and K^v_h, respectively, and the treatment of the mixing terms is described in Part II of the documentation. Constant background mixing is introduced by setting lower limits according to

$$K^v_M = \max\left(K^v_m, K^v_{mb}\right)$$

$$K_H^v = \max\left(K_h^v, K_{hb}^v\right). \tag{5.21}$$

K_{mb}^v and K_{hb}^v are limiting constants which replace the values of the physical diffusion coefficients in case of weak turbulence. In this way a small background vertical diffusion is enabled. By default, these constants have the values $K_{mb}^v = 1.0$ m^2/s and $K_{hb}^v = 0.2$ m^2/s. If the vertical resolution of the model is increased, these values should be reduced following the guidelines for the assignment of the coefficients for horizontal computational mixing.

Because computational mixing, as formulated in Section 6.2 operates only along the sloping horizontal coordinate surfaces, the constant background mixing according to Eq. (5.21) is at present the only scheme for vertical smoothing. It has been found that this type of smoothing is very important to avoid a decoupling of the surface and the atmosphere in cases with strong thermal stability.

5.4 Upper Boundary Damping Layer

Enhanced damping can optionally be included in a number of model layers just below the upper boundary. The task of this type of damping is to absorb upward propagating wave disturbances and to suppress gravity wave reflection at the top boundary resulting from the rigid lid upper boundary condition, which is applied in the model. The prevention of wave energy reflection at the upper boundary is of crucial importance for a proper simulation of orographically induced flows. An alternative, more physically based method to avoid wave reflection is the application of a radiative upper boundary condition instead of the rigid lid condition. This option, however, is not yet implemented.

In LM enhanced damping near the top of the model domain is accomplished by Rayleigh friction terms which are added to the right hand side of the prognostic equations for momentum, temperature and pressure perturbation. These terms are formulated to damp the deviations from the corresponding boundary fields, which are defined as threedimensional time dependent fields and are obtained by interpolation from the driving model (see Section 5). Thus, the LM formulation of the damping layer tends to restore the externally specified boundary fields near the top of the domain.

The Rayleigh damping term of a variable ψ is denoted by M_ψ^{RD} and has the form

$$M_\psi^{RD} = -\mu_R(z)\left(\psi - \psi_b\right). \tag{5.22}$$

ψ_b is the externally specified boundary value of ψ, $\mu_R(z)$ is the vertical profile of the damping coefficient (i.e. the inverse e-folding time scale of damping at height z) and ψ represents u, v, w, T or p'. For the vertical velocity, $w_b = 0$ is specified.

In the present version of LM we apply a rather simple profile of the damping coefficient μ_R:

$$\mu_R(z) = \begin{cases} \dfrac{1}{2n_R\Delta t}\left\{1 - \cos\left(\pi\dfrac{z - z_D}{z_T - z_D}\right)\right\} & \text{for } z \geq z_D\,, \\ 0 & \text{for } z < z_D\,. \end{cases} \tag{5.23}$$

z_D is the height of the bottom of the damping layer and z_T the height of the model top boundary. The depth of the damping layer $(z_T - z_D)$ depends on the type of problem being

considered. In general, a layer depth of 1/3 of the total domain height or at least one vertical wavelength is recommended. Of course, the damping layer should be located above the part of the model where the solution is of interest. $n_R\Delta t$ is the e-folding time scale of damping at $z = z_T$. According to the profile function (5.23), the corresponding damping coefficient decreases to $\mu_R = 0$ at $z = z_D$.

Both z_D and n_R are disposable parameters. The default values are set to $z_D = 11000m$ and $n_R = 10$.

Similar to the treatment of the lateral boundary relaxation terms in Section 5, the time integration of the Rayleigh damping terms is done by an implicit scheme to ensure numerical stability. The prognostic equation of a model variable ψ is written as

$$\frac{\partial \psi}{\partial t} = f(\psi) + M_\psi^{RD} = f(\psi) - \mu_R \left(\psi - \psi_b\right),$$

where $f(\psi)$ represents all adiabatic and diabatic forcings as well as computational mixing and lateral boundary relaxation. The time discretization reads

$$\psi^{n+1} = \psi^{n-1} + f^n(\psi) 2\Delta t - 2\Delta t \mu_R \left(\psi^{n+1} - \psi_b^{n+1}\right),$$

which may be rewritten as

$$\psi^{n+1} = \tilde{\psi}^{n+1} - \frac{2\Delta t \mu_R}{1 + 2\Delta t \mu_R} \left(\tilde{\psi}^{n+1} - \psi_b^{n+1}\right). \tag{5.24}$$

Here, $\tilde{\psi}^{n+1} = \psi^{n-1} + 2\Delta t f^n(\psi)$ is a preliminary value of ψ which results from the integration of the prognostic equation taking all processes except Rayleigh damping into account. Thus, the final value of ψ is obtained from the preliminary value $\tilde{\psi}$ by a local updating according to Eq. (5.24).

5.5 Rayleigh Friction

For operational NWP-purposes, a Rayleigh friction term in the equations for the horizontal wind components u and v has been introduced. In the split-explicit time integration scheme described in Section 4.3, the big time step size Δt for stable integration is limited by the advective wind speed and may estimated by the CFL-condition

$$\Delta t \leq \frac{\Delta s}{\sqrt{2}\, v_a^{max}} \tag{5.25}$$

where Δs is the minimum horizontal grid spacing and v_a^{max} maximum absolute horizontal wind speed. Thus, for a given grid spacing, the time step has to be chosen from an estimate of the maximum v_a^{max} which will occur during the forecast. However, the maximum wind speed can become very large near the tropopause, especially during winter where v_a^{max} can exceed 130 m/s on some days.

In order to not restrict the time step for operational forecasts on the few events where v_a^{max} reaches such large values, Rayleigh friction is introduced in the u- and v-equations in the

form

$$\frac{\partial u}{\partial t} = \cdots - \epsilon_R u \tag{5.26}$$

$$\frac{\partial v}{\partial t} = \cdots - \epsilon_R v \tag{5.27}$$

in order to slow down the wind speed whenever a critical value is exceeded during the forecast. The Rayleigh friction coefficient ϵ_R is set to

$$\epsilon_R = \begin{cases} \dfrac{0.0005}{2\Delta t} \dfrac{(v_a - 0.95\, v_a^{cfl})}{0.05\, v_a^{cfl}} & \text{for } v_a \geq 0.95\, v_a^{cfl}\,, \\ 0 & \text{for } v_a < 0.95\, v_a^{cfl}\,. \end{cases} \tag{5.28}$$

Here, v_a is the maximum absolute wind speed at a time step within the model domain and v_a^{cfl} is the maximum wind speed allowed for stable integration. v_a^{cfl} is estimated from the CFL-condition (5.25) for a given grid spacing and time step, i.e. $v_a^{cfl} = \Delta s/(\sqrt{2}\Delta t)$. Thus, the time step should be chosen carefully to achieve a realistic (not too small) critical wind speed.

According to (5.28), Rayleigh friction is applied to the horizontal wind components whenever the wind speed exceeds 95% of v_a^{cfl} somewhere in the model domain. The damping coefficient then increases linearly from zero to 0.05% per $2\Delta t$ interval at $v_a = v_a^{cfl}$.

5.6 Filtering of Topographical Forcing

A long-standing problem in high-resolution modelling is the prediction of unrealistic precipitation fields in mountainous areas. During the first year of the operational NWP-application of LM, two main deficiencies became evident:

- extremely large amounts of precipitation occur directly over mountain tops and
- at the same time deep valleys and mountain lee-sides receive no precipitation at all, turning into a desert-like climate within the model's assimilation cycle.

These effects are most noticeable over the Alpine region, but are also noticeable over mid-range mountain areas. All in all, the predicted precipitation fields look very noisy in regions with steep topography.

A reason for this behaviour is the use of mean topography at the grid-scale Δx, which will result in 'singular' mountain tops and valley grounds being represented by one gridpoint only. Constant dynamical surface forcing of this type introduces a hopelessly inaccurate numerical feedback to the simulated flow, which then can affect the physics in a strange way. To investigate this effect, convergence tests of the solutions corresponding to mountain generated 3-D gravitational waves have been performed Gaßmann (2001). The conclusion of these tests is that topographical structures must be reasonably well resolved by the grid in order to obtain a quantitatively correct flow. Similar tests have been performed by Davies and Brown. (2001) for a 2-D flow.

Thus a weak filtering of topography, i.e. a removal of the very small-scale components from the wave spectrum, is necessary to allow for a more correct interaction of the dynamics

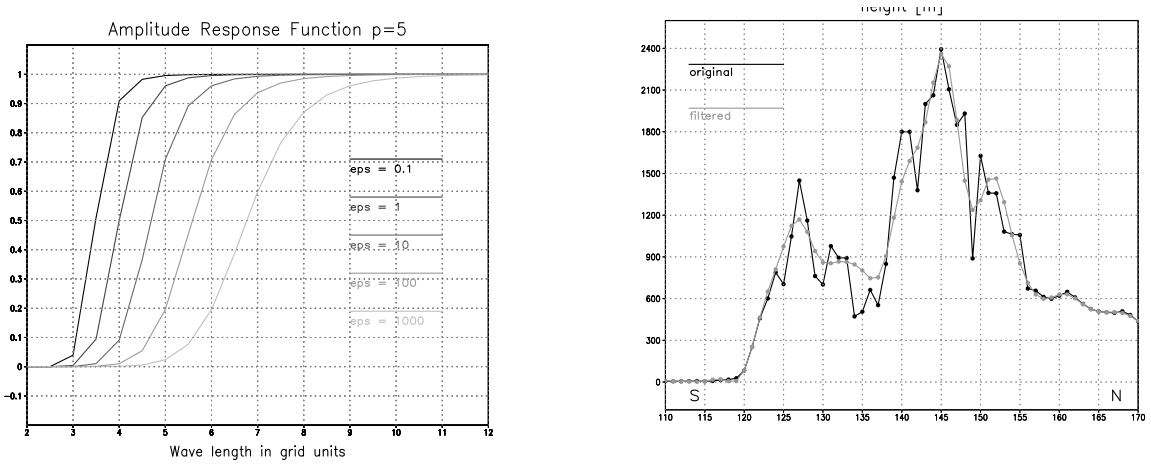

Figure 5.4: Left: Amplitude response function of the 10-th order Raymond filter for various values of the filter parameter ϵ. Right: South-north cross section of the LM-topography along the Brenner line for original (black) and filtered (grey) topography.

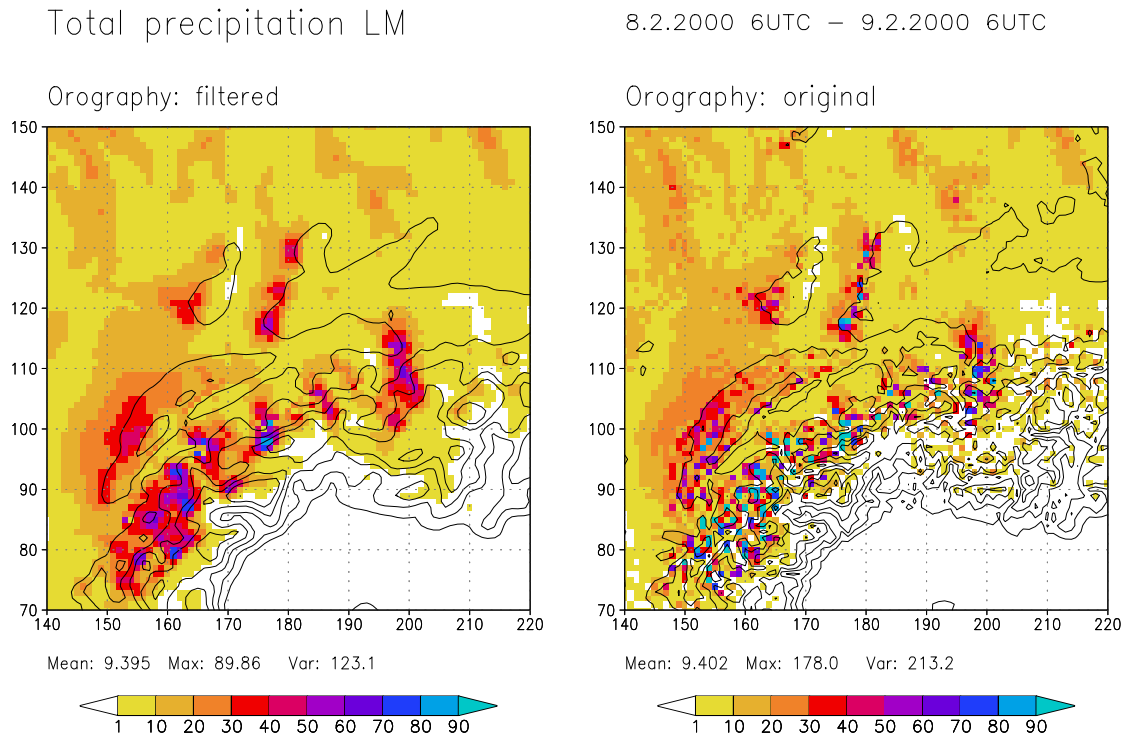

Figure 5.5: 24-h precipitation amount of a LM simulation starting at 8 February 2000 00 UTC. Left: Experimental run with filtered topography. Right: Operational run with unfilterd topography.

with the surface. A 10-th order Raymond (1988) filter using a filter parameter of $\epsilon = 0.1$ is applied for this purpose. Figure 5.4 (left) shows the amplitude response function for this filter. $2\Delta x$ and $3\Delta x$ wave components are filtered almost completely whereas $4\Delta x$ and larger scale components remain untouched. The impact of this weak filter on the LM topography is also shown in Figure 5.4 (right): all singular mountain and valley structures are removed.

The use of a filtered topography has a dramatic impact on the predicted precipitation field. An example is shown in Figure 5.5 for a test simulation starting at 8 February 2000 00 UTC. The routine run (right) exhibits the well known features with unrealistic minima and maxima of the 24-h accumulated precipitation amount in the Alps. A much smoother spatial distribution is predicted in the experimental run with filtered topography. All larger scale patterns and the area mean value of precipitation are very similar compared to the routine run, whereas the maximum and the variance of precipitation is reduced by about a factor of two. Outside the mountainous Alpine region, the differences are negligible.

Results from other simulations and the statistical evaluation of a quasi-operational test suite revealed an overall positive impact of using a weakly filtered topography. In general, there is no loss of meteorological relevant information and the precipitation patterns appear to be more realistic. We conclude that filtering of topography is necessary to avoid the formation of non-coherent dynamical structures generated by under-resolved surface forcing. Since winter 2000/2001, all operational runs of LM at COSMO meteorological centres use a filtered topography. For experimental runs with interpolated initial and boundary conditions, the filter can optionally be switched on and off.

Section 6

Alternative Time Integration Schemes

An alternative algorithm for time integration schemes has been implemented for optional use: a three-timelevel Leapfrog-based Eulerian 3-D semi-implicit scheme according to Thomas et al. (2000). This scheme has the potential for being more accurate at the same efficiency or being more efficient at the same accuracy than the default three-timelevel HE-VI scheme. However, the method is still under development and require further testing for operational application.

6.1 A 3-D Semi-Implicit scheme

As the default time integration scheme for LM, a variant of the Klemp and Wilhelmson (1978) time-splitting method has been implemented. The extensions proposed by Skamarock and Klemp (1992) are applied whereby horizontally propagating waves are treated explicitly with a forward-backward scheme and buoyancy oscillations are integrated with an implicit Crank-Nicholson method (see Section 4.3).

However, steep orography may provoke instabilities in such 1D semi-implicit schemes Steppeler (1995). The stability of these schemes has been analyzed by Ikawa (1988) and a stability criterion for MM5 was derived by Dudhia (1995). Fully 3D semi-implicit schemes can avoid such stability problems by treating all pressure gradient and divergence terms implicitly (Skamarock et al. (1997); Saito (1997)). A model comparison study by Saito et al. (1998) suggests that fully 3D semi-implicit schemes may be more cost-effective than split-explicit schemes at higher resolutions when the number of small time steps increases with the sound speed Courant number. Thus, a semi-implicit time scheme has been designed for the LM which does not depend on approximations such as mass lumping to simplify the resulting variable-coefficient elliptic boundary value problem. The resulting elliptic boundary value problem is solved with a minimal residual Krylov iterative method whose convergence rate is accelerated by line relaxation preconditioners (Skamarock et al. (1997) and Thomas et al. (1998)).

The development work and implementation of the semi-implicit scheme was conducted by Steve Thomas during his stay as guest scientist at DWD. We describe here only the numerical

formulation of the scheme. For tests of the scheme for idealized mountain wave flows and for real cases – including accuracy and efficiency considerations – we refer to Thomas et al. (2000).

6.1.1 Semi-implicit Time Discretization

(a) Governing Equations

In this section we derive a fully three-dimensional semi-implicit scheme for the LM model governing equations (2.143) - (2.147) which are rewritten in the mode-splitting form (see Section 4.2).

$$\frac{\partial u}{\partial t} + \frac{1}{\rho a \cos\varphi}\left(\frac{\partial p'}{\partial \lambda} - \frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \lambda}\frac{\partial p'}{\partial \zeta}\right) = f_u \tag{6.1}$$

$$\frac{\partial v}{\partial t} + \frac{1}{\rho a}\left(\frac{\partial p'}{\partial \varphi} - \frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \varphi}\frac{\partial p'}{\partial \zeta}\right) = f_v \tag{6.2}$$

$$\frac{\partial w}{\partial t} - \frac{g}{\sqrt{\gamma}}\frac{\rho_0}{\rho}\frac{\partial p'}{\partial \zeta} - g\frac{\rho_0}{\rho}\left\{\frac{p}{p_0}\left[\frac{T-T_0}{T} - \frac{p'}{p}\right]\right\} = f_w \tag{6.3}$$

$$\frac{\partial p'}{\partial t} - g\rho_0 w + \frac{pc_{pd}}{c_{vd}}\left\{D_h - \frac{g\rho_0}{\sqrt{\gamma}}\frac{\partial w}{\partial \zeta}\right\} = f_{p'} \tag{6.4}$$

$$\frac{\partial T}{\partial t} + \frac{R_d T}{c_{vd}}\left\{D_h - \frac{g\rho_0}{\sqrt{\gamma}}\frac{\partial w}{\partial \zeta}\right\} = f_T \tag{6.5}$$

The f-terms on the right-hand side denote the slow mode forcings, the terms on the left-hand side describe sound and gravity wave propagation. D_h denotes the horizontal wind divergence given by

$$D_h = \frac{1}{a\cos\varphi}\left\{\frac{\partial u}{\partial \lambda} - \frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \lambda}\frac{\partial u}{\partial \zeta} + \frac{\partial}{\partial \varphi}(v\cos\varphi) - \frac{\cos\varphi}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \varphi}\frac{\partial v}{\partial \zeta}\right\}. \tag{6.6}$$

For practical reasons, the factor $p/(\rho c_{vd})$ for the wind divergence in the thermodynamic equation (2.147) has been approximated by $R_d T/c_{vd}$. Also, the "dry" buoyancy term appearing in (2.145) has been rewritten in the equation (6.3) for the vertical velocity using the identity

$$B(T-T_0, p') := g\frac{\rho_0}{\rho}\left\{\frac{(T-T_0)}{T} - \frac{T_0 p'}{T p_0}\right\} = g\frac{\rho_0}{\rho}\left\{\frac{p}{p_0}\left[\frac{T-T_0}{T} - \frac{p'}{p}\right]\right\}. \tag{6.7}$$

The fast-mode forcing terms will be discretized by employing the time-tendency formulation of the semi-implicit scheme. Time discretization operators appearing in the Leapfrog or three-time-level semi-implicit scheme are given by

$$\delta_\tau(\psi) = \frac{\psi^{n+1} - \psi^{n-1}}{2\Delta\tau}, \tag{6.8}$$

$$\mu_\tau(\psi) = \frac{\psi^{n+1} + \psi^{n-1}}{2} = \Delta\tau\,\delta_\tau(\psi) + \psi^{n-1}. \tag{6.9}$$

Here, $\Delta\tau$ is the time step, δ_τ denotes the time differencing operator and μ_τ is the time averaging operator. Using (6.8) and (6.9), a fully 3D semi-implicit scheme applied to the LM

governing equations results in the following system of time and space discretized equations, where the linearization about the current time step n is based on p^n, T^n and ρ^n.

$$\begin{aligned}
\delta_\tau(u) + \mu_\tau\, \tilde{\delta}_x^{(1)}(p') &= f_u^n \\
\delta_\tau(v) + \mu_\tau\, \tilde{\delta}_y^{(1)}(p') &= f_v^n \\
\delta_\tau(w) + \mu_\tau\, \tilde{\delta}_z^{(1)}(p') - \mu_\tau\, B(T - T_0, p') &= f_w^n \\
\delta_\tau(p') + \left(\frac{p^n c_{pd}}{c_{vd}}\right) \mu_\tau D - \mu_\tau\, g\, \rho_0\, \mu_\zeta(w) &= f_{p'}^n \\
\delta_\tau(T) + \left(\frac{R_d T^n}{c_{vd}}\right) \mu_\tau D &= f_T^n
\end{aligned}$$

Using the notations from Section 4.1 for finite space differencing, the difference operators for the pressure gradient force are defined by

$$\begin{aligned}
\tilde{\delta}_x^{(1)}(p') &= \frac{1}{\overline{\rho^n}^{\lambda}\, a \cos\varphi} \left\{ \delta_\lambda(p') - \frac{1}{\overline{(\sqrt{\gamma})}^{\zeta,\lambda}}\, \delta_\lambda(p_0)\, \delta_{2\zeta}\overline{(p')}^{\lambda} \right\}, \\
\tilde{\delta}_y^{(1)}(p') &= \frac{1}{\overline{\rho^n}^{\varphi}\, a} \left\{ \delta_\varphi(p') - \frac{1}{\overline{(\sqrt{\gamma})}^{\zeta,\varphi}}\, \delta_\varphi(p_0)\, \delta_{2\zeta}\overline{(p')}^{\varphi} \right\}, \\
\tilde{\delta}_z^{(1)}(p') &= -\frac{g}{\sqrt{\gamma}}\, \frac{\overline{\rho_0}^{\zeta}}{\overline{\rho^n}^{\zeta}}\, \delta_\zeta(p')\,.
\end{aligned}$$

Operators appearing in the discretized divergence are given by

$$\begin{aligned}
\tilde{\delta}_x^{(2)}(u) &= \frac{1}{a \cos\varphi} \left\{ \delta_\lambda(u) - \mu_\lambda \frac{1}{\overline{(\sqrt{\gamma})}^{\zeta,\lambda}}\, \delta_\lambda(p_0)\, \delta_{2\zeta}\overline{(u)}^{\lambda} \right\}, \\
\tilde{\delta}_y^{(2)}(v) &= \frac{1}{a \cos\varphi} \left\{ \delta_\varphi(v\, \cos\varphi) - \mu_\varphi \frac{\cos\varphi}{\overline{(\sqrt{\gamma})}^{\zeta,\varphi}}\, \delta_\varphi(p_0)\, \delta_{2\zeta}\overline{(v)}^{\varphi} \right\}, \\
\tilde{\delta}_z^{(2)}(w) &= -\frac{g\rho_0}{\sqrt{\gamma}}\, \delta_\zeta(w)\,,
\end{aligned}$$

where the horizontal (D_h) and total divergence (D) are

$$D_h = \tilde{\delta}_x^{(2)}(u) + \tilde{\delta}_y^{(2)}(v), \qquad D = D_h + \tilde{\delta}_z^{(2)}(w)\,.$$

μ_λ and μ_φ denote the operators for horizontal averaging according to (3.4). In a similar way, we abbreviate the vertical averaging operator by μ_ζ. Vertical averaging is defined by (3.13) for transforming half-level values to full levels, and by (3.19) for averaging full-level values to half levels. For convenience we define also the weighted averaging operator $\tilde{\mu}_\zeta$ by

$$\tilde{\mu}_\zeta = \frac{\overline{\rho_0}^{\zeta}}{\overline{\rho^n}^{\zeta}}\, \mu_\zeta(w)\,. \tag{6.10}$$

The buoyancy must be averaged in the vertical momentum equation and thus we define the discretized B as

$$B(T - T_0, p') = g\, \tilde{\mu}_\zeta \left\{ \frac{p^n}{p_0} \left[\frac{T - T_0}{T^n} - \frac{p'}{p^n} \right] \right\}. \tag{6.11}$$

Following Skamarock et al. (1997), time averages are replaced by differences to avoid numerical cancellation and thus only time tendencies appear on the left-hand side. The remaining terms at time level $n-1$ are placed on the right-hand side of the equations (forming the explicit part of the discretized equations):

$$\delta_\tau(u) + \Delta\tau\ \tilde{\delta}_x^{(1)}\delta_\tau(p') = q_u\,, \tag{6.12}$$

$$\delta_\tau(v) + \Delta\tau\ \tilde{\delta}_y^{(1)}\delta_\tau(p') = q_v\,, \tag{6.13}$$

$$\delta_\tau(w) + \Delta\tau\ \tilde{\delta}_z^{(1)}\delta_\tau(p') - \Delta\tau\ \delta_\tau\ B(T-T_0,p') = q_w\,, \tag{6.14}$$

$$\delta_\tau(p') + \Delta\tau\left(\frac{p^n c_{pd}}{c_{vd}}\right)\delta_\tau D - \Delta\tau\ g\ \rho_0\ \mu_\zeta\delta_\tau(w) = q_{p'}\,, \tag{6.15}$$

$$\delta_\tau(T) + \Delta\tau\left(\frac{R_d T^n}{c_{vd}}\right)\delta_\tau D = q_T\,, \tag{6.16}$$

where the right-hand sides now become

$$\begin{aligned}
q_u &= f_u^n - \tilde{\delta}_x^{(1)}(p')^{n-1}\,,\\
q_v &= f_v^n - \tilde{\delta}_y^{(1)}(p')^{n-1}\,,\\
q_w &= f_w^n - \tilde{\delta}_z^{(1)}(p')^{n-1} + B(T-T_0,p')^{n-1}\,,\\
q_{p'} &= f_{p'}^n - \left(\frac{p^n c_{pd}}{c_{vd}}\right)D^{n-1} + g\ \rho_0\ \mu_\zeta(w)^{n-1}\,,\\
q_T &= f_T^n - \left(\frac{R_d T^n}{c_{vd}}\right)D^{n-1}\,.
\end{aligned}$$

Observe in particular that since $\delta_\tau\ T_0 = 0$,

$$\delta_\tau\ B(T-T_0,p') = B(\delta_\tau(T),\ \delta_\tau(p')) = \delta_\tau\ B(T,p')$$

and the full T, as opposed to the perturbation temperature $T' = T - T_0$, can be used in the thermodynamic equation.

(b) Discrete Wave Equation

To obtain an elliptic problem for the pressure perturbation p', we proceed to eliminate the buoyancy and divergence from the discretized system of equations (6.12) – (6.16). The buoyancy B is eliminated from the system of equations by first observing that from equations (6.15) and (6.16)

$$\frac{\delta_\tau(T)}{T^n} = -\Delta\tau\left(\frac{R_d}{c_{vd}}\right)\delta_\tau D + \frac{q_T}{T^n}\,, \tag{6.17}$$

$$\frac{\delta_\tau(p')}{p^n} = -\Delta\tau\left(\frac{c_{pd}}{c_{vd}}\right)\delta_\tau D + \Delta\tau\ g\ \frac{\rho_0}{p^n}\ \mu_\zeta\delta_\tau(w) + \frac{q_{p'}}{p^n}\,. \tag{6.18}$$

Subtracting (6.18) from (6.17) and then applying the relation

$$\frac{R}{c_v} = \frac{R_d}{c_{vd}} = \frac{c_{pd}-c_{vd}}{c_{vd}}\,,$$

we obtain

$$\frac{\delta_\tau(T)}{T^n} - \frac{\delta_\tau(p')}{p^n} + \Delta\tau\left[-\delta_\tau D + g\ \frac{\rho_0}{p^n}\ \mu_\zeta\delta_\tau(w)\right] = \frac{q_T}{T^n} - \frac{q_{p'}}{p^n}\,.$$

Given that

$$B(\delta_\tau(T),\ \delta_\tau(p')) = g\ \tilde{\mu}_\zeta\left\{\frac{p^n}{p_0}\left[\frac{\delta_\tau(T)}{T^n} - \frac{\delta_\tau(p')}{p^n}\right]\right\}\,,$$

it follows that

$$B(\delta_\tau(T),\ \delta_\tau(p')) + \Delta\tau\ g\ \tilde{\mu}_\zeta\ \left[\ -\frac{p^n}{p_0}\ \delta_\tau D + g\ \frac{\rho_0}{p_0}\ \mu_\zeta \delta_\tau(w)\ \right] = B(q_T, q_{p'})\,. \tag{6.19}$$

Next, we apply the operator $\Delta\tau g\tilde{\mu}_\zeta c_{vd}/(c_{pd}p_0)$ to Eq. (6.15) resulting in

$$\Delta\tau\ g\ \tilde{\mu}_\zeta\ \frac{c_{vd}}{c_{pd}}\ \frac{\delta_\tau(p')}{p_0} + \Delta\tau^2\ g\ \tilde{\mu}_\zeta\ \left[\ \frac{p^n}{p_0}\ \delta_\tau D - g\ \frac{c_{vd}}{c_{pd}}\ \frac{\rho_0}{p_0}\ \mu_\zeta \delta_\tau(w)\ \right] = \Delta\tau\ g\ \tilde{\mu}_\zeta\ \frac{c_{vd}}{c_{pd}}\ \frac{q_{p'}}{p_0}\,, \tag{6.20}$$

and also write equation (6.14) as

$$\delta_\tau(w) + \Delta\tau\ \tilde{\delta}_z^{(1)} \delta_\tau(p') - \Delta\tau\ B(\delta_\tau(T),\ \delta_\tau(p')) = q_w\,. \tag{6.21}$$

Finally, eliminate the buoyancy B from the equations by combining

$$\Delta\tau\ (6.19) + (6.20) + (6.21)$$

to obtain

$$\left[\ 1 + \Delta\tau^2\ \tilde{\mu}_\zeta\ N_0^2\ \mu_\zeta\ \right]\ \delta_\tau(w) + \Delta\tau\ \tilde{D}_z^{(1)}\ \delta_\tau(p') = q_w^*\,, \tag{6.22}$$

where

$$q_w^* = q_w + \Delta\tau\ g\ \tilde{\mu}_\zeta\ \left\{ \frac{p^n}{p_0}\ \left[\ \frac{q_T}{T^n} - \frac{R_d}{c_{pd}}\frac{q_{p'}}{p^n}\ \right] \right\}\,.$$

To eliminate the horizontal divergence D_h from the system of equations, form

$$\frac{c_{vd}}{c_{pd}}\ \frac{1}{p^n}\ (6.15) - \Delta\tau\ D_h(\ (6.12),\ (6.13)\ )$$

and thus obtain

$$\left[\ \frac{c_{vd}}{c_{pd}}\ \frac{1}{p^n} - \Delta\tau^2\ D_h(\ \tilde{\delta}_x^{(1)},\ \tilde{\delta}_y^{(1)}\ )\ \right]\ \delta_\tau(p') + \Delta\tau\ \tilde{D}_z^{(2)}\ \delta_\tau(w) = q'_{p'}\,, \tag{6.23}$$

where

$$q'_{p'} = \frac{c_{vd}}{c_{pd}}\ \frac{1}{p^n}\ q_{p'} - \Delta\tau\ D_h(\ q_u,\ q_v\ )\,.$$

Solve for $\delta_\tau(w)$ in (6.22) and substitute into (6.23) to the wave equation in the final form

$$\left[\ \frac{c_{vd}}{c_{pd}}\frac{1}{p^n} - \Delta\tau^2\ D_h(\ \tilde{\delta}_x^{(1)},\ \tilde{\delta}_y^{(1)}\ )\ \right]\ \delta_\tau(p') - \Delta\tau^2\ \tilde{D}_z^{(2)}\ \mathcal{N}^{-1}\ \tilde{D}_z^{(1)}\ \delta_\tau(p') = q_p^*\,. \tag{6.24}$$

Here, the right-hand side is defined by

$$q_p^* = q'_{p'} - \Delta\tau\ \tilde{D}_z^{(2)}\ \mathcal{N}^{-1}\ q_w^*\,.$$

The vertical operators $\mathcal{N}$, $\tilde{D}_z^{(1)}$ and $\tilde{D}_z^{(2)}$ appearing above are all defined in Section 7.1.2.

(c) Backsubstitution

Having solved the resulting elliptic problem $\mathcal{L}\,\delta_\tau(p') = q^*_{p'}$ from Eq. (6.24), the remaining tendencies are obtained by back substitution as follows:

$$\delta_\tau(u) = q_u - \Delta\tau\,\tilde{\delta}^{(1)}_x \delta_\tau(p')\,, \tag{6.25}$$

$$\delta_\tau(v) = q_v - \Delta\tau\,\tilde{\delta}^{(1)}_y \delta_\tau(p')\,, \tag{6.26}$$

$$\delta_\tau(w) = \mathcal{N}^{-1}\left[\, q^*_w - \Delta\tau\,\tilde{D}^{(1)}_z\,\delta_\tau(p')\,\right], \tag{6.27}$$

$$\delta_\tau(T) = q_T - \Delta\tau\left(\frac{R_d T^n}{c_{vd}}\right) D(\delta_\tau \mathbf{u})\,, \tag{6.28}$$

where $\mathbf{u} = (\,u,\,v,\,w\,)$. An off-centered time average can also be employed by replacing $\Delta\tau$ by

$$\Delta\tau^+ = (\,1 + \beta_{sw}\,)\,\Delta\tau = 2\beta\Delta\tau$$

on the left-hand side of the discretized governing equations. The traditional semi-implicit scheme uses the centered weighting $\beta = 1/2$ or $\beta_{sw} = 0$.

To compute the final values of the prognostic variables at time level $n+1$, we employ

$$(\psi)^{n+1} = (\psi)^{n-1} + 2\Delta\tau\delta_\tau(\psi)\,.$$

6.1.2 Implementation Details

(a) The Elliptic Operator

Operators appearing in the discrete formulation of the fully 3D semi-implicit scheme consist of both matrices and associated boundary conditions applied to a given field. We now proceed to develop the precise structure of the operators appearing in the discretized wave equation (6.24). Vertical operators are derived first. The operator $\tilde{D}^{(2)}_z$ is defined by

$$\tilde{D}^{(2)}_z = \left[\,\tilde{\delta}^{(2)}_z - g\,\frac{c_{vd}}{c_{pd}}\,\frac{\rho_0}{p^n}\,\mu_\zeta\,\right], \qquad \tilde{\delta}^{(2)}_z = -\frac{g\rho_0}{\sqrt{\gamma}}\,\delta_\zeta\,,$$

and moves data from model half levels $k' = k - 1/2$ to main levels k as follows.

$$\begin{aligned}
\left[\,\tilde{D}^{(2)}_z w_{k'}\,\right]_k &= \left[\,\tilde{\delta}^{(2)}_z - g\,\frac{c_{vd}}{c_{pd}}\,\frac{(\rho_0)_k}{(p^n)_k}\,\mu_\zeta\,\right] w_{k'} \\
&= \left[\,-\frac{g(\rho_0)_k}{(\sqrt{\gamma})_k}\,(w_{k'+1} - w_{k'}) - g\,\frac{c_{vd}}{c_{pd}}\,\frac{(\rho_0)_k}{(p^n)_k}\,\frac{(w_{k'+1} + w_{k'})}{2}\right] \\
&= \left[\,-\frac{g(\rho_0)_k}{(\sqrt{\gamma})_k} - \frac{g}{2}\,\frac{c_{vd}}{c_{pd}}\,\frac{(\rho_0)_k}{(p^n)_k}\right] w_{k'+1} + \left[\,\frac{g(\rho_0)_k}{(\sqrt{\gamma})_k} - \frac{g}{2}\,\frac{c_{vd}}{c_{pd}}\,\frac{(\rho_0)_k}{(p^n)_k}\right] w_{k'}\,.
\end{aligned}$$

The result can be written in the form of a recurrence formula

$$\left[\,\tilde{D}^{(2)}_z w_{k'}\,\right]_k = b^{(2)}_k w_{k'+1} - a^{(2)}_k w_{k'}\,, \tag{6.29}$$

where

$$b^{(2)}_k = -g(\rho_0)_k\left[\,\frac{1}{(\sqrt{\gamma})_k} + \frac{1}{2}\frac{c_{vd}}{c_{pd}}\frac{1}{(p^n)_k}\,\right], \quad a^{(2)}_k = -g(\rho_0)_k\left[\,\frac{1}{(\sqrt{\gamma})_k} - \frac{1}{2}\frac{c_{vd}}{c_{pd}}\frac{1}{(p^n)_k}\,\right],$$

or equivalently

$$b_k^{(2)} = -\frac{g(\rho_0)_k}{(\sqrt{\gamma})_k}\left[1+\frac{1}{2}\frac{c_{vd}}{c_{pd}}\frac{(\sqrt{\gamma})_k}{(p^n)_k}\right], \quad a_k^{(2)} = -\frac{g(\rho_0)_k}{(\sqrt{\gamma})_k}\left[1-\frac{1}{2}\frac{c_{vd}}{c_{pd}}\frac{(\sqrt{\gamma})_k}{(p^n)_k}\right].$$

The matrix structure of $\tilde{D}_z^{(2)}$ is lower bi-diagonal with dimensions $(N_\zeta+1)\times(N_\zeta+1)$.

$$\begin{bmatrix} b_0^{(2)} & & & & & & \\ -a_1^{(2)} & b_1^{(2)} & & & & & \\ & -a_2^{(2)} & b_2^{(2)} & & & & \\ & & \cdot & \cdot & & & \\ & & & \cdot & \cdot & & \\ & & & & -a_{N_\zeta-1}^{(2)} & b_{N_\zeta-1}^{(2)} & \\ & & & & & -a_{N_\zeta}^{(2)} & b_{N_\zeta}^{(2)} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \cdot \\ \cdot \\ w_{N_\zeta} \\ w_{N_\zeta+1} \end{bmatrix} = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \cdot \\ \cdot \\ p_{N_\zeta-1} \\ p_{N_\zeta} \end{bmatrix}$$

From dimensional analysis, we have as a final check that

$$-\frac{\sqrt{\gamma}}{p} \sim \Delta \ln p \sim \frac{g\Delta Z}{RT}, \quad -\frac{\sqrt{\gamma}}{\rho} \sim g\Delta Z, \quad -\frac{c_{vd}}{c_{pd}}\frac{\sqrt{\gamma}}{p}\frac{1}{\Delta Z} \sim \frac{c_{vd}}{c_{pd}}\frac{g}{RT} \sim \frac{g}{c_s^2},$$

and therefore the operator is of the correct form.

The operator $\tilde{D}_z^{(1)}$ is defined by

$$\tilde{D}_z^{(1)} = \left[\tilde{\delta}_z^{(1)} + \tilde{\mu}_\zeta \frac{c_{vd}}{c_{pd}}\frac{g}{p_0}\right], \quad \tilde{\delta}_z^{(1)} = -\frac{g}{\sqrt{\gamma}}\frac{\overline{\rho_0}^\zeta}{\overline{\rho^n}^\zeta}\delta_\zeta, \quad \tilde{\mu}_\zeta = \frac{\overline{\rho_0}^\zeta}{\overline{\rho^n}^\zeta}\mu_\zeta\,.$$

The operator moves data from full levels k to half levels $k' = k - 1/2$ according to

$$\left[\tilde{D}_z^{(1)} p'_k\right]_{k'} = \left[\tilde{\delta}_z^{(1)} + \tilde{\mu}_\zeta \frac{c_{vd}}{c_{pd}}\frac{g}{(p_0)_k}\right] p'_k\,.$$

In discrete form, we have

$$\left[\tilde{\delta}_z^{(1)} p'_k\right]_{k'} = -\frac{g}{(\sqrt{\gamma})_{k'}}\frac{(\overline{\rho_0}^\zeta)_{k'}}{(\overline{\rho^n}^\zeta)_{k'}}(p'_k - p'_{k-1})\,,$$

$$\left[\mu_\zeta \frac{(p')_k}{(p_0)_k}\right]_{k'} = \frac{1}{2(\sqrt{\gamma})_{k'}}\left\{(\sqrt{\gamma})_{k-1}\frac{(p')_k}{(p_0)_k} + (\sqrt{\gamma})_k\frac{(p')_{k-1}}{(p_0)_{k-1}}\right\},$$

where

$$(\sqrt{\gamma})_{k'} = \frac{1}{2}\left\{(\sqrt{\gamma})_{k-1} + (\sqrt{\gamma})_k\right\}.$$

Therefore, we can immediately write down the recurrence formula

$$\left[\tilde{D}_z^{(1)} p'_k\right]_{k'} = b_{k'}^{(1)} p'_k - a_{k'}^{(1)} p'_{k-1} \tag{6.30}$$

with coefficients

$$b_{k'}^{(1)} = -\frac{g}{(\sqrt{\gamma})_{k'}}\frac{(\overline{\rho_0}^\zeta)_{k'}}{(\overline{\rho^n}^\zeta)_{k'}}\left[1-\frac{1}{2}\frac{c_{vd}}{c_{pd}}\frac{(\sqrt{\gamma})_{k-1}}{(p_0)_k}\right], \quad a_{k'}^{(1)} = -\frac{g}{(\sqrt{\gamma})_{k'}}\frac{(\overline{\rho_0}^\zeta)_{k'}}{(\overline{\rho^n}^\zeta)_{k'}}\left[1+\frac{1}{2}\frac{c_{vd}}{c_{pd}}\frac{(\sqrt{\gamma})_k}{(p_0)_{k-1}}\right].$$

The matrix structure of $\tilde{D}_z^{(1)}$ is lower bi-diagonal with dimensions $(N_\zeta + 1) \times (N_\zeta + 1)$.

$$\begin{bmatrix} b_1^{(1)} & & & & & & \\ -a_2^{(1)} & b_2^{(1)} & & & & & \\ & -a_3^{(1)} & b_3^{(1)} & & & & \\ & & \cdot & \cdot & & & \\ & & & \cdot & \cdot & & \\ & & & & -a_{N_\zeta}^{(1)} & b_{N_\zeta}^{(1)} & \\ & & & & & -a_{N_\zeta+1}^{(1)} & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \cdot \\ \cdot \\ p_{N_\zeta} \\ p_{N_\zeta+1} \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \cdot \\ \cdot \\ w_{N_\zeta} \\ w_{N_\zeta+1} \end{bmatrix} \equiv \begin{bmatrix} w_{1/2} \\ w_{3/2} \\ w_{5/2} \\ \cdot \\ \cdot \\ w_{N_\zeta-1/2} \\ w_{N_\zeta+1/2} \end{bmatrix}$$

Vertical boundary conditions are handled by

$$\left[\tilde{D}_z^{(1)} \right]_{1/2} = b_1^{(1)} = 0, \qquad \left[\tilde{D}_z^{(1)} \right]_{N_{\zeta+1/2}} = \left\{ \alpha \, \tilde{\delta}_x^{(1)} + \beta \, \tilde{\delta}_y^{(1)} \right\} = -a_{N_{\zeta+1}}^{(1)}.$$

The form of the vector $\tilde{D}_z^{(2)} \, \tilde{D}_z^{(1)} p'$ is obtained from the matrix–vector product

$$\begin{bmatrix} b_0^{(2)} & & & & & & \\ -a_1^{(2)} & b_1^{(2)} & & & & & \\ & -a_2^{(2)} & b_2^{(2)} & & & & \\ & & \cdot & \cdot & & & \\ & & & \cdot & \cdot & & \\ & & & & -a_{N_\zeta-1}^{(2)} & b_{N_\zeta-1}^{(2)} & \\ & & & & & -a_{N_\zeta}^{(2)} & b_{N_\zeta}^{(2)} \end{bmatrix} \begin{bmatrix} 0 \\ b_2^{(1)} p'_2 - a_2^{(1)} p'_1 \\ b_3^{(1)} p'_3 - a_3^{(1)} p'_2 \\ \cdot \\ \cdot \\ b_{N_\zeta}^{(1)} p'_{N_\zeta} - a_{N_\zeta}^{(1)} p'_{N_\zeta-1} \\ -a_{N_\zeta+1}^{(1)} p'_{N_\zeta} \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_3 \\ \cdot \\ \cdot \\ x_{N_\zeta-1} \\ x_{N_\zeta} \end{bmatrix}$$

By expanding, we obtain for $k = 2, \ldots, N_\zeta - 1$:

$$\begin{aligned} \left[\tilde{D}_z^{(2)} \, \tilde{D}_z^{(1)} p' \right]_k &= -a_k^{(2)} \left(b_k^{(1)} p'_k - a_k^{(1)} p'_{k-1} \right) + b_k^{(2)} \left(b_{k+1}^{(1)} p'_{k+1} - a_{k+1}^{(1)} p'_k \right) \\ &= a_k^{(2)} a_k^{(1)} p'_{k-1} - \left(a_k^{(2)} b_k^{(1)} + b_k^{(2)} a_{k+1}^{(1)} \right) p'_k + b_k^{(2)} b_{k+1}^{(1)} p'_{k+1} \, . \end{aligned}$$

For model main levels $k = 1$ and $k = N_\zeta$:

$$\left[\tilde{D}_z^{(2)} \, \tilde{D}_z^{(1)} p' \right]_1 = b_1^{(2)} \left(b_2^{(1)} p'_2 - a_2^{(1)} p'_1 \right) ,$$

$$\begin{aligned} \left[\tilde{D}_z^{(2)} \, \tilde{D}_z^{(1)} p' \right]_{N_\zeta} &= -a_{N_\zeta}^{(2)} \left(b_{N_\zeta}^{(1)} p'_{N_\zeta} - a_{N_\zeta}^{(1)} p'_{N_\zeta-1} \right) - b_{N_\zeta}^{(2)} a_{N_\zeta+1}^{(1)} p'_{N_\zeta} \\ &= a_{N_\zeta}^{(2)} a_{N_\zeta}^{(1)} p'_{N_\zeta-1} - \left(a_{N_\zeta}^{(2)} b_{N_\zeta}^{(1)} + b_{N_\zeta}^{(2)} a_{N_\zeta+1}^{(1)} \right) p'_{N_\zeta} \, . \end{aligned}$$

The reference state Brunt-Väisällä frequency and sound speed are

$$N_0^2 = g^2 \, \frac{\rho_0}{p_0} \, \frac{R_d}{c_{pd}} = \frac{g^2}{c_{pd} T_0} \qquad c_0^2 = \frac{c_{pd}}{c_{vd}} \, \frac{p_0}{\rho_0} = \gamma \, R_d T_0 \, .$$

The Brunt-Väisällä operator is defined by

$$\mathcal{N} = \left[1 + \Delta\tau^2 \, \tilde{\mu}_\zeta \, N_0^2 \, \mu_\zeta \right] , \qquad \tilde{\mu}_\zeta = \frac{\overline{\rho_0}^\zeta}{\overline{\rho^n}^\zeta} \, \mu_\zeta \, .$$

The operator moves data from half levels k' to main levels k and then back to half levels k'. A three-term recurrence involving three vertical levels $k'-1$, k' and $k'+1$ results in a tridiagonal matrix.

$$[\,\mathcal{N} w_{k'}\,]_{k'} = \left[\, 1 + \Delta\tau^2 \, \frac{(\overline{\rho_0}^{\zeta})_{k'}}{(\overline{\rho^n}^{\zeta})_{k'}} \, \mu_\zeta \, (N_0^2)_k \, \mu_\zeta \,\right] w_{k'} \,, \qquad (N_0^2)_k = g^2 \, \frac{(\rho_0)_k}{(p_0)_k} \, \frac{R_d}{c_{pd}} \,,$$

where

$$\mu_\zeta \, w_{k'} = \frac{(w_{k'+1} + w_{k'})}{2} \,, \quad \mu_\zeta \, \psi_k = \frac{1}{2(\sqrt{\gamma})_{k'}} \, \{\, (\sqrt{\gamma})_{k-1} \, (\psi)_k + (\sqrt{\gamma})_k \, (\psi)_{k-1} \,\} \,.$$

Expanding individual terms, we obtain

$$\begin{aligned}
\mu_\zeta \, (N_0^2)_k \, (\overline{w}^{\zeta})_k &= \frac{1}{2(\sqrt{\gamma})_{k'}} \left\{ (\sqrt{\gamma})_{k-1} \, (N_0^2)_k \, (\overline{w}^{\zeta})_k + (\sqrt{\gamma})_k \, (N_0^2)_{k-1} \, (\overline{w}^{\zeta})_{k-1} \right\} \\
&= \frac{1}{2(\sqrt{\gamma})_{k'}} \left\{ (\sqrt{\gamma})_{k-1} \, (N_0^2)_k \, \frac{(w_{k'+1} + w_{k'})}{2} \right\} \\
&+ \frac{1}{2(\sqrt{\gamma})_{k'}} \left\{ (\sqrt{\gamma})_k \, (N_0^2)_{k-1} \frac{(w_{k'} + w_{k'-1})}{2} \right\} \\
&= \frac{1}{4(\sqrt{\gamma})_{k'}} \, (\sqrt{\gamma})_{k-1} \, (N_0^2)_k \, w_{k'+1} \\
&+ \frac{1}{4(\sqrt{\gamma})_{k'}} \left\{ (\sqrt{\gamma})_{k-1} \, (N_0^2)_k + (\sqrt{\gamma})_k \, (N_0^2)_{k-1} \right\} w_{k'} \\
&+ \frac{1}{4(\sqrt{\gamma})_{k'}} \, (\sqrt{\gamma})_k \, (N_0^2)_{k-1} \, w_{k'-1} \,.
\end{aligned}$$

The resulting three-term recurrence is given by

$$[\,\mathcal{N} w_{k'}\,]_{k'} = a_{k'} w_{k'-1} + b_{k'} w_{k'} + c_{k'} w_{k'+1} = \bar{w}_{k'} \,,$$

where

$$\begin{aligned}
c_{k'} &= \Delta\tau^2 \, \frac{(\overline{\rho_0}^{\zeta})_{k'}}{(\overline{\rho^n}^{\zeta})_{k'}} \, \frac{1}{4(\sqrt{\gamma})_{k'}} \, (\sqrt{\gamma})_{k-1} \, (N_0^2)_k \,, \\
b_{k'} &= 1 + \Delta\tau^2 \, \frac{(\overline{\rho_0}^{\zeta})_{k'}}{(\overline{\rho^n}^{\zeta})_{k'}} \, \frac{1}{4(\sqrt{\gamma})_{k'}} \left\{ (\sqrt{\gamma})_{k-1} \, (N_0^2)_k + (\sqrt{\gamma})_k \, (N_0^2)_{k-1} \right\} , \\
a_{k'} &= \Delta\tau^2 \, \frac{(\overline{\rho_0}^{\zeta})_{k'}}{(\overline{\rho^n}^{\zeta})_{k'}} \, \frac{1}{4(\sqrt{\gamma})_{k'}} \, (\sqrt{\gamma})_k \, (N_0^2)_{k-1} \,.
\end{aligned}$$

In particular, the matrix element weighting is such that

$$b_{k'} = 1 + a_{k'} + c_{k'} \,.$$

The matrix structure of $\mathcal{N}$ is tridiagonal with dimensions $(N_\zeta + 1) \times (N_\zeta + 1)$.

$$\begin{bmatrix}
1 & & & & & & & \\
a_2 & b_2 & c_2 & & & & & \\
 & a_3 & b_3 & c_3 & & & & \\
 & & . & . & . & & & \\
 & & & . & . & . & & \\
 & & & & a_{N_\zeta-1} & b_{N_\zeta-1} & c_{N_\zeta-1} & \\
 & & & & & a_{N_\zeta} & b_{N_\zeta} & c_{N_\zeta} \\
 & & & & & & & 1
\end{bmatrix}
\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ . \\ . \\ w_{N_\zeta-1} \\ w_{N_\zeta} \\ w_{N_\zeta+1} \end{bmatrix}
=
\begin{bmatrix} w_1 \\ \bar{w}_2 \\ \bar{w}_3 \\ . \\ . \\ \bar{w}_{N_\zeta-1} \\ \bar{w}_{N_\zeta} \\ w_{N_\zeta+1} \end{bmatrix}$$

An important property of $\mathcal{N}$ is that the product $\mathcal{N}w$ leaves the boundary conditions $w_{1/2}$ and $w_{N_\zeta+1/2}$ unmodified. Mass–lumping of the operator $\mathcal{N}$ is applied to simplify the preconditioner, with diagonal coefficients $b_{k'} = 1 + 2a_{k'} + 2c_{k'}$:

$$\bar{b}_{k'} = 1 + \Delta\tau^2 \, \frac{(\overline{\rho_0}^{\zeta})_{k'}}{(\overline{\rho^n}^{\zeta})_{k'}} \, \frac{1}{2(\sqrt{\gamma})_{k'}} \left\{ (\sqrt{\gamma})_{k-1} \, (N_0^2)_k + (\sqrt{\gamma})_k \, (N_0^2)_{k-1} \right\} .$$

(b) Lateral Boundary Conditions

Lateral boundaries are open with inflow/outflow determined by the normal components of the velocity. To a large extent the Arakawa 'C' grid and semi-implicit scheme determine the precise numerical form of boundary conditions. In the horizontal direction, the normal velocity components $u_{N_W-1/2,j,k}$ in the west, $u_{N_E+1/2,j,k}$ in the east, $v_{i,N_S-1/2,k}$ in the south and $v_{i,N_N+1/2,k}$ in the north intersect the boundary. For equations (6.12) – (6.16), the relations

$$\begin{aligned}
[\,\delta_\tau(u)\,]_{N_W-1/2,j,k} &= [\,q_u\,]_{N_W-1/2,j,k} = [\,\delta_\tau(u_e)\,]_{N_W-1/2,j,k} \\
[\,\delta_\tau(u)\,]_{N_E+1/2,j,k} &= [\,q_u\,]_{N_E+1/2,j,k} = [\,\delta_\tau(u_e)\,]_{N_E+1/2,j,k} \\
[\,\delta_\tau(v)\,]_{i,N_S-1/2,k} &= [\,q_v\,]_{i,N_S-1/2,k} = [\,\delta_\tau(v_e)\,]_{i,N_S-1/2,k} \\
[\,\delta_\tau(v)\,]_{i,N_N+1/2,k} &= [\,q_v\,]_{i,N_N+1/2,k} = [\,\delta_\tau(v_e)\,]_{i,N_N+1/2,k}
\end{aligned}$$

along with the implied Neumann boundary conditions below for the discrete wave equation, effectively close the numerical problem.

$$\begin{aligned}
\left[\,\tilde{\delta}_x^{(1)}(\delta_\tau(p'))\,\right]_{N_W-1/2,j,k} &= \left[\,\tilde{\delta}_x^{(1)}(\delta_\tau(p'))\,\right]_{N_E+1/2,j,k} = 0 \\
\left[\,\tilde{\delta}_y^{(1)}(\delta_\tau(p'))\,\right]_{i,N_S-1/2,k} &= \left[\,\tilde{\delta}_y^{(1)}(\delta_\tau(p'))\,\right]_{i,N_N+1/2,k} = 0
\end{aligned}$$

An implicit scheme for the pressure implies that the time tendency of the pressure gradient force must be zero at lateral boundaries so that the momentum equations are self-consistent. This is purely a numerical boundary condition to close the problem. Mass flux across lateral boundaries appears in right-hand side terms at time level $n-1$.

(c) Top and Bottom Boundary Conditions

In Section 5.2 the top and bottom boundary conditions on the contravariant $\dot{\zeta}$ and covariant w vertical velocity components have been specified. The model top at $k = 1/2$ is treated as a rigid lid where the vertical velocity vanishes:

$$\left(\,\dot{\zeta}\,\right)_{i,j,k=1/2} = (\,w\,)_{i,j,k=1/2} = 0\,.$$

The lower boundary is non-penetrative and the contravariant vertical velocity vanishes

$$\left(\,\dot{\zeta}\,\right)_{i,j,k=N_{\zeta+1/2}} = 0\,.$$

From equation (3.112) of Section 3.5, this implies that the vertical velocity w follows coordinate surfaces according to

$$w = \frac{J_\lambda}{a\cos\varphi}u + \frac{J_\varphi}{a}v\,.$$

To evaluate w at the surface $k = N_{\zeta+1/2}$, free-slip conditions on the horizontal wind components are assumed and these are extrapolated to the surface resulting in

$$w_{N_{\zeta+1/2}} = \frac{1}{a\cos\varphi}\left\{ \mu_\lambda(\delta_\lambda h)u_{N_\zeta} + \mu_\varphi(\delta_\varphi h)v_{N_\zeta}\cos\varphi \right\} . \tag{6.31}$$

The vertical velocity equation (6.14) does not apply at the top or bottom surfaces even though equations (6.15) and (6.16) require values at these points. In the current LM implementation, the values of u_{N_ζ} and v_{N_ζ} at time level $n+1$ are available in the split-explicit scheme. Thus, the values of $w_{1/2}$ and $w_{N_{\zeta+1/2}}$ may be computed using the above relations. Then w is computed implicitly at interior points using a tridiagonal solver.

In a fully implicit scheme for the pressure, the horizontal velocity components are defined implicitly through the horizontal momentum equations (6.12) and (6.13).

$$\delta_\tau(u) = q_u - \Delta\tau\,\tilde{\delta}_x^{(1)}\delta_\tau(p') , \tag{6.32}$$
$$\delta_\tau(v) = q_v - \Delta\tau\,\tilde{\delta}_y^{(1)}\delta_\tau(p') . \tag{6.33}$$

The time tendency of the vertical velocity equation (6.31) can be written in the general form

$$(\delta_\tau w)_{N_{\zeta+1/2}} = \alpha\,\delta_\tau(u_{N_\zeta}) + \beta\,\delta_\tau(v_{N_\zeta}) , \tag{6.34}$$

where α and β represent averaging operators and associated metric terms. Substituting equations (6.32) and (6.33) into (6.34) results in

$$(\delta_\tau w)_{N_{\zeta+1/2}} + \Delta\tau\left\{ \alpha\,\tilde{\delta}_x^{(1)} + \beta\,\tilde{\delta}_y^{(1)} \right\}(\delta_\tau p')_{N_\zeta} = \alpha\,(q_u)_{N_\zeta} + \beta\,(q_v)_{N_\zeta} . \tag{6.35}$$

Comparing (6.35) with (6.22) given below,

$$\mathcal{N}\,\delta_\tau(w) + \Delta\tau\,\tilde{D}_z^{(1)}\,\delta_\tau(p') = q_w^* ,$$

a vertical velocity equation satisfying the top and bottom boundary conditions is obtained. At the model lid,

$$\mathcal{N}(\delta_\tau w)_{1/2} = 0 ,$$

and therefore

$$\mathcal{N}_{1/2} = 1 , \quad \left[\tilde{D}_z^{(1)}\right]_1 = 0 , \quad (q_w^*)_{1/2} = 0 .$$

At the surface level $k = N_{\zeta+1/2}$,

$$\mathcal{N}(\delta_\tau w)_{N_{\zeta+1/2}} + \Delta\tau\,\tilde{D}_z^{(1)}\,(\delta_\tau p')_{N_{\zeta+1/2}} = (q_w^*)_{N_{\zeta+1/2}} ,$$

where

$$\mathcal{N}_{N_{\zeta+1/2}} = 1, \quad \left[\tilde{D}_z^{(1)}\right]_{N_{\zeta+1/2}} = \left\{ \alpha\,\tilde{\delta}_x^{(1)} + \beta\,\tilde{\delta}_y^{(1)} \right\}_{N_\zeta} , \quad (q_w^*)_{N_{\zeta+1/2}} = \alpha\,(q_u)_{N_\zeta} + \beta\,(q_v)_{N_\zeta} .$$

(d) Elliptic Solver

The elliptic problem (6.24) contains cross-derivative terms with variable coefficients and is solved using a GMRES Krylov type iteration (Saad and Schultz (1986), Saad (1993)). Smolarkiewicz and Margolin (1994), Smolarkiewicz and Margolin (1997) employ a right-preconditioned variant of the GCR algorithm of C. et al. (1983) for the pressure solver in

anelastic models. GMRES is mathematically equivalent to the GCR algorithm when the symmetric part of the elliptic operator $\mathcal{L}$ is negative-definite. In this case, both algorithms produce an identical sequence of iterates. Smolarkiewicz et al. (1997) propose a convergence test based on $\|\mathcal{L}\delta_\tau(p') - q^*_{p'}\| < \varepsilon_c$ where ε_c is an estimate of the RMS divergence. Skamarock et al. (1997) apply a similar test to the compressible equations and their approach is adopted in the LM solver. To accelerate convergence, a suitable preconditioning technique must be found. In essence, a preconditioner $\mathcal{P}$ is an approximate inverse of $\mathcal{L}$.

The relative magnitude of the horizontal and vertical parts of $\mathcal{L}$ is determined by the square of the grid aspect ratio $\Delta X^2/\Delta Z^2$ and can differ by several orders of magnitude when this ratio is large. For example, consider a quasi-hydrostatic LM model run with

$$\Delta X \approx a\Delta\lambda \approx 6\,\text{km}, \quad \Delta Z \sim -\frac{\sqrt{\gamma}}{g\rho} = 60\,\text{m} \quad \text{to} \quad \Delta Z \sim -\frac{\sqrt{\gamma}}{g\rho} = 600\,\text{m},$$

implying $\Delta X^2/\Delta Z^2$ varies from 10^2 up to 10^4. Therefore, an effective preconditioning strategy is to derive an implicit iteration using operator splitting in the vertical direction since these will be the dominant terms in the discretized system of equations for typical grid resolutions employed at meso-β and meso-γ scales. A vertical preconditioner will also be efficient at micro-α scales, however, a fully 3D scheme is preferable in the case of isotropic grids. Consider the simple two-dimensional diffusion equation

$$\frac{\partial u}{\partial \tau} = \nabla^2 u - r\,.$$

The vertical Alternating Direction Implicit (ADI) preconditioner proposed by Skamarock et al. (1997) is based on a forward-in-time integration scheme and a splitting of the discretized Laplacian into vertical and horizontal components. In discrete form,

$$(\, I - \beta_2\, \delta_{zz}\,)\; u^{\nu+1} = (\, I + \beta_1\, \delta_{xx}\,)\; u^{\nu} - \Delta\tau\, r\,, \tag{6.36}$$

where $\delta_{xx}u = u_{i+1,k} - 2u_{ik} + u_{i-1,k}$, $\delta_{zz}u = u_{i,k+1} - 2u_{ik} + u_{i,k-1}$, $\beta_1 = \Delta\tau/\Delta x^2$ and $\beta_2 = \Delta\tau/\Delta z^2$. The largest 'pseudo' time step $\Delta\tau < \Delta x^2/2$ is chosen so that the scheme remains stable when integrated to steady-state. Tridiagonal systems must be inverted along each vertical grid line in such an implicit scheme. When used as a preconditioner $\mathcal{P}$, one pseudo-time step or iteration of (6.36) is applied within each GMRES iteration.

A more implicit time integration scheme can be derived by including the diagonal terms appearing in δ_{xx} on the left hand side. In general, a preconditioned fixed-point iteration is derived from a matrix splitting $A = M - N$ and can be written in the equivalent forms

$$x^{\nu+1} = x^{\nu} + M^{-1}(b - Ax^{\nu}) = M^{-1}Nx^{\nu} + M^{-1}b$$

with iteration matrix $G = M^{-1}N = (I - M^{-1}A)$. The Jacobi and Gauss-Seidel iterations are based on the splitting $A = D - E - F$. D is diagonal, whereas E and F are strictly lower and upper triangular matrices. For Jacobi $M = D$, whereas $M = D - E$ in the Gauss-Seidel iteration. The forward-in-time, central-space scheme in 2D is

$$\frac{u^{\nu+1}_{ik} - u^{\nu}_{ik}}{\Delta\tau} = \frac{u^{\nu}_{i+1,k} - 2u^{\nu}_{ik} + u^{\nu}_{i-1,k}}{\Delta x^2} + \frac{u^{\nu}_{i,k+1} - 2u^{\nu}_{ik} + u^{\nu}_{i,k-1}}{\Delta z^2} - r_{ik}$$

Assume that $\Delta x = \Delta z$ so that the above scheme becomes

$$u^{\nu+1}_{ik} = u^{\nu}_{ik} + \frac{\Delta\tau}{\Delta x^2}\left[\, u^{\nu}_{i+1,k} + u^{\nu}_{i-1,k} + u^{\nu}_{i,k+1} + u^{\nu}_{i,k-1} - 4u^{\nu}_{ik} - \Delta x^2 r_i \,\right].$$

Setting $\Delta\tau/\Delta x^2 = 1/4$, the point Jacobi iteration in 2D is obtained:

$$4u_{ik}^{\nu+1} = u_{i+1,k}^{\nu} + u_{i-1,k}^{\nu} + u_{i,k+1}^{\nu} + u_{i,k-1}^{\nu} - \Delta x^2 r_i \,.$$

The above 'point' relaxation applies a weighted average of the four neighboring points to update the solution. Alternatively, a vertical line Jacobi relaxation scheme results from the implicit time-stepping scheme given below,

$$-u_{i,k-1}^{\nu+1} + 4u_{ik}^{\nu+1} - u_{i,k+1}^{\nu+1} = u_{i+1,k}^{\nu} + u_{i-1,k}^{\nu} - \Delta x^2 r_i \,, \tag{6.37}$$

and requires the solution of tridiagonal systems of equations. The LM preconditioner is a simple generalization of the splitting (6.37) to non-isotropic grids and a detailed derivation is provided below.

(e) Line Jacobi Preconditioner

The first step in the construction of a vertical line Jacobi preconditioner for the LM model is to derive an approximate form for $\mathcal{L}$ which still contains the dominant terms. Off–diagonal cross–derivative terms will be ignored in the preconditioner, since only the main diagonal is retained in the matrix splitting. Therefore, the first approximation to the elliptic operator drops terrain following terms in the horizontal gradient ∇ and divergence operators $(\nabla\cdot)$,

$$\left[\frac{c_{vd}}{c_{pd}} \frac{1}{p^n} - \Delta\tau^2 \, \nabla \cdot \frac{1}{\rho^n} \, \nabla \right] \delta_\tau(p') - \Delta\tau^2 \, \tilde{D}_z^{(2)} \, \mathcal{N}^{-1} \, \tilde{D}_z^{(1)} \, \delta_\tau(p') = q_{p'}^* \,.$$

In discrete form, we have

$$\nabla \cdot \frac{1}{\rho^n} \nabla = \frac{1}{a\cos\varphi} \delta_\lambda \frac{1}{\overline{\rho^n}^{\lambda}} \frac{1}{a\cos\varphi} \delta_\lambda + \frac{1}{a\cos\varphi} \delta_\varphi \cos\varphi \frac{1}{\overline{\rho^n}^{\varphi}} \frac{1}{a} \delta_\varphi \,.$$

To obtain the operator splitting, apply the operator to the pressure p

$$\left\{ \frac{1}{a\cos\varphi} \delta_\lambda \frac{1}{\overline{\rho^n}^{\lambda}} \frac{1}{a\cos\varphi} \delta_\lambda \, p \right\}_{ij} = \frac{1}{(\Delta\lambda \, a\cos\varphi_j)^2} \left[\frac{p_{i+1,j} - p_{ij}}{(\overline{\rho^n}^{\lambda})_{i+1/2,j}} - \frac{p_{ij} - p_{i-1,j}}{(\overline{\rho^n}^{\lambda})_{i-1/2,j}} \right] ,$$

and

$$\begin{aligned} \left\{ \frac{1}{a\cos\varphi} \delta_\varphi \cos\varphi \frac{1}{\overline{\rho^n}^{\varphi}} \frac{1}{a} \delta_\varphi \, p \right\}_{ij} &= \frac{1}{(a\Delta\varphi)^2 \cos\varphi_j} \left[\cos\varphi_{j+1/2} \frac{p_{i,j+1} - p_{ij}}{(\overline{\rho^n}^{\varphi})_{i,j+1/2}} \right] \\ &- \frac{1}{(a\Delta\varphi)^2 \cos\varphi_j} \left[\cos\varphi_{j-1/2} \frac{p_{ij} - p_{i,j-1}}{(\overline{\rho^n}^{\varphi})_{i,j-1/2}} \right] . \end{aligned}$$

The diagonal of the horizontal operator is therefore given by

$$\left[\frac{c_{vd}}{c_{pd}} \frac{1}{p^n} - \Delta\tau^2 \, d_{ij} \right] ,$$

where the diagonal coefficient $-d_{ij}$ appearing in the line Jacobi splitting is

$$\frac{1}{(\Delta\lambda \, a\cos\varphi_j)^2} \left[\frac{1}{(\overline{\rho^n}^{\lambda})_{i+1/2,j}} + \frac{1}{(\overline{\rho^n}^{\lambda})_{i-1/2,j}} \right] + \frac{1}{(a\Delta\varphi)^2 \cos\varphi_j} \left[\frac{\cos\varphi_{j+1/2}}{(\overline{\rho^n}^{\varphi})_{i,j+1/2}} + \frac{\cos\varphi_{j-1/2}}{(\overline{\rho^n}^{\varphi})_{i,j-1/2}} \right] .$$

Neumann boundary conditions are imposed along the lateral boundaries. For the eastern and western boundaries, $N_S \le j \le N_N$,

$$p_{N_E,j} - p_{N_E-1,j} = 0\,, \quad p_{N_W+1,j} - p_{N_W,j} = 0\,,$$

and the southern and northern boundaries, $N_E \le i \le N_W$

$$p_{i,N_S} - p_{i,N_S-1} = 0\,, \quad p_{i,N_N+1} - p_{i,N_N} = 0\,.$$

In the vertical direction, the form of $x = \tilde{D}_z^{(2)} \mathcal{N}^{-1} \tilde{D}_z^{(1)} p'$ with mass-lumping is

$$\begin{bmatrix} b_0^{(2)} & & & & & & \\ -a_1^{(2)} & b_1^{(2)} & & & & & \\ & -a_2^{(2)} & b_2^{(2)} & & & & \\ & & \cdot & \cdot & & & \\ & & & \cdot & \cdot & & \\ & & & & -a_{N_\zeta-1}^{(2)} & b_{N_\zeta-1}^{(2)} & \\ & & & & & -a_{N_\zeta}^{(2)} & b_{N_\zeta}^{(2)} \end{bmatrix} \begin{bmatrix} 0 \\ \bar{b}_2^{-1} \left(b_2^{(1)} p'_2 - a_2^{(1)} p'_1 \right) \\ \bar{b}_3^{-1} \left(b_3^{(1)} p'_3 - a_3^{(1)} p'_2 \right) \\ \cdot \\ \cdot \\ \bar{b}_{N_\zeta}^{-1} \left(b_{N_\zeta}^{(1)} p'_{N_\zeta} - a_{N_\zeta}^{(1)} p'_{N_\zeta-1} \right) \\ -a_{N_\zeta+1}^{(1)} p'_{N_\zeta} \end{bmatrix}$$

By expanding, we obtain for $k = 2, \ldots, N_\zeta - 1$

$$\begin{aligned} [\,x\,]_k &= -a_k^{(2)} \bar{b}_k^{-1} \left(b_k^{(1)} p'_k - a_k^{(1)} p'_{k-1} \right) + b_k^{(2)} \bar{b}_{k+1}^{-1} \left(b_{k+1}^{(1)} p'_{k+1} - a_{k+1}^{(1)} p'_k \right) \\ &= a_k^{(2)} \bar{b}_k^{-1} a_k^{(1)} p'_{k-1} - \left(a_k^{(2)} \bar{b}_k^{-1} b_k^{(1)} + b_k^{(2)} \bar{b}_{k+1}^{-1} a_{k+1}^{(1)} \right) p'_k + b_k^{(2)} \bar{b}_{k+1}^{-1} b_{k+1}^{(1)} p'_{k+1} \\ &= a_k^{(3)} p'_{k-1} + b_k^{(3)} p'_k + c_k^{(3)} p'_{k+1}\,. \end{aligned}$$

For model main levels $k = 1$ and $k = N_\zeta$

$$\begin{aligned} [\,x\,]_1 &= b_1^{(2)} \bar{b}_2^{-1} \left(b_2^{(1)} p'_2 - a_2^{(1)} p'_1 \right) \\ &= b_1^{(3)} p'_1 + c_1^{(3)} p'_2\,, \\ [\,x\,]_{N_\zeta} &= -a_{N_\zeta}^{(2)} \bar{b}_{N_\zeta}^{-1} \left(b_{N_\zeta}^{(1)} p'_{N_\zeta} - a_{N_\zeta}^{(1)} p'_{N_\zeta-1} \right) - b_{N_\zeta}^{(2)} a_{N_\zeta+1}^{(1)} p'_{N_\zeta} \\ &= a_{N_\zeta}^{(2)} \bar{b}_{N_\zeta}^{-1} a_{N_\zeta}^{(1)} p'_{N_\zeta-1} - \left(a_{N_\zeta}^{(2)} \bar{b}_{N_\zeta}^{-1} b_{N_\zeta}^{(1)} + b_{N_\zeta}^{(2)} a_{N_\zeta+1}^{(1)} \right) p'_{N_\zeta} \\ &= a_{N_\zeta}^{(3)} p'_{N_\zeta-1} + b_{N_\zeta}^{(3)} p'_{N_\zeta}\,. \end{aligned}$$

The preconditioner is based on solving tridiagonal systems in the vertical direction with Dirichlet boundary conditions on the pressure at $k = 0$ and $k = N_\zeta + 1$. Thus, coefficients of the above operator are computed only for interior points and the LU factorization of the vertical tridiagonal systems is performed once per time step in order to minimize computations. One further approximation in the preconditioner is to set

$$-a_{N_{\zeta+1}}^{(1)} = \left[\tilde{D}_z^{(1)} \right]_{N_{\zeta+1/2}} = \left\{ \alpha\, \tilde{\delta}_x^{(1)} + \beta\, \tilde{\delta}_y^{(1)} \right\} = 0\,.$$

Section 7

The Runge-Kutta dynamical core

In addition to the so called Leapfrog-scheme (Klemp and Wilhelmson (1978), mainly chapters 4 and 5) and a semi-implicit solver (Thomas et al. (2000), section 6.1) a new dynamical core for the solution of the compressible Euler-equations, called Runge-Kutta (RK) dynamical core, is available in the COSMO-model since about the start of the pre-operational test phase (August 2006) of the COSMO-DE (the convection permitting model with 2.8 km resolution). It bases on the time-splitting approach of Wicker and Skamarock (2002) with the intention that for successful convective-scale simulations a numerical solver is needed that produces much less numerical noise at the grid scale as the up to now used centered-difference schemes of the leapfrog solver.

The Euler equations. Since the Euler equations in the RK dynamical core are slightly different to their formulation in the leapfrog scheme, we here repeat them for spherical, terrain following coordinates (λ, ϕ, ζ) with $\zeta = \zeta(\lambda, \phi, z)$, $r = r_{earth} + z$ (for relations between the occurring coordinate derivatives we refer to eqns. (2.100)). The prognostic equations for the dynamical variables spherical wind components (u, v and w), and the deviations of temperature and pressure T', p' from a base state are split into a slow and a fast part. The slow part, denoted by the tendencies $\frac{\partial u}{\partial t}\Big|_{slow}, \ldots, \frac{\partial T'}{\partial t}\Big|_{slow}$, consists of the advection terms (section 7.1.2), the Coriolis terms (section 7.1.3) and tendencies from all the physical parameterisations Doms et al. (2004). The fast parts (see section 7.2) are the pressure gradient terms and the working terms in the T'- and p'-equation, thus leading to sound expansion, and the buoyancy terms, leading to the expansion of gravity waves.

$$\frac{\partial u}{\partial t} = -\frac{1}{\rho}\frac{1}{r\cos\phi}\left(\frac{\partial p'}{\partial \lambda} + \frac{\partial \zeta}{\partial \lambda}\frac{\partial p'}{\partial \zeta}\right) + D_{damp,u} + \left.\frac{\partial u}{\partial t}\right|_{slow} \tag{7.1}$$

$$\frac{\partial v}{\partial t} = -\frac{1}{\rho}\frac{1}{r}\left(\frac{\partial p'}{\partial \phi} + \frac{\partial \zeta}{\partial \phi}\frac{\partial p'}{\partial \zeta}\right) + D_{damp,v} + \left.\frac{\partial v}{\partial t}\right|_{slow} \tag{7.2}$$

$$\frac{\partial w}{\partial t} = -\frac{1}{\rho}\frac{\partial \zeta}{\partial z}\frac{\partial p'}{\partial \zeta} + g\left(\frac{p_0}{p}\frac{T'}{T_0} - \frac{p'}{p} + \frac{p_0}{p}\frac{T}{T_0}q_x\right) + D_{damp,w} + \left.\frac{\partial w}{\partial t}\right|_{slow} \tag{7.3}$$

$$\frac{\partial p'}{\partial t} = -\frac{c_p}{c_V} p\, D + g\rho_0 w + \left.\frac{\partial p'}{\partial t}\right|_{slow} \tag{7.4}$$

$$\frac{\partial T'}{\partial t} = -\frac{R}{c_V} T\, D - \frac{\partial T_0}{\partial z} w + \left.\frac{\partial T'}{\partial t}\right|_{slow} \tag{7.5}$$

The artificial divergence damping terms are abbreviated by

$$D_{damp,u} = \frac{1}{\rho}\frac{1}{r\cos\phi}\left(\frac{\partial\,\alpha^h_{div}\,\rho\,D}{\partial\lambda}+\frac{\partial\zeta}{\partial\lambda}\frac{\partial\,\alpha^h_{div}\,\rho\,D}{\partial\zeta}\right) \tag{7.6}$$

$$D_{damp,v} = \frac{1}{\rho}\frac{1}{r}\left(\frac{\partial\,\alpha^h_{div}\,\rho\,D}{\partial\phi}+\frac{\partial\zeta}{\partial\phi}\frac{\partial\,\alpha^h_{div}\,\rho\,D}{\partial\zeta}\right) \tag{7.7}$$

$$D_{damp,w} = \frac{1}{\rho}\frac{\partial\zeta}{\partial z}\frac{\partial\,\alpha^v_{div}\,\rho\,D}{\partial\zeta} \tag{7.8}$$

To indicate the optional divergence damping in the w-equation, too, the diffusion coefficient α^v_{div} carries an upper index 'v'.

The divergence is denoted by $D = \operatorname{div}\mathbf{v}$. One difference to the former version of the fast-waves solver is the use of its so-called strong conservation form (we here repeat eq. (2.122)):

$$D \equiv \operatorname{div}\mathbf{v} = \frac{1}{r^2\cos\phi}\frac{1}{\sqrt{g}}\left[\frac{\partial}{\partial\lambda}\left(r\sqrt{g}\,u\right)+\frac{\partial}{\partial\phi}\left(r\cos\phi\,\sqrt{g}\,v\right)+\frac{\partial rZ}{\partial\zeta}\right] \tag{7.9}$$

with the definition

$$Z := r\cos\phi\,\sqrt{g}\,\dot{\zeta} = \frac{\partial z}{\partial\lambda}\,u+\frac{\partial z}{\partial\phi}\cos\phi\,v-r\cos\phi\,w. \tag{7.10}$$

Obviously, the divergence is expressed on the one hand by derivatives in *terrain following* coordinates but on the other hand of the physical components u, v, w in *spherical* coordinates. Here one can cancel one factor r due to the shallow atmosphere approximation $r \approx r_{earth} = const.$, which is used throughout in the COSMO model

$$D = \frac{1}{r\cos\phi}\frac{1}{\sqrt{g}}\left[\frac{\partial}{\partial\lambda}\left(\sqrt{g}\,u\right)+\frac{\partial}{\partial\phi}\left(\cos\phi\,\sqrt{g}\,v\right)+\frac{\partial Z}{\partial\zeta}\right]. \tag{7.11}$$

It is advisable to introduce some other abbreviations

$$\tilde{d}_{hor} := \frac{\partial}{\partial\lambda}\left(\sqrt{g}\,u\right)+\frac{\partial}{\partial\phi}\left(\cos\phi\,\sqrt{g}\,v\right), \qquad \tilde{d}_{vert} := \frac{\partial Z}{\partial\zeta}, \tag{7.12}$$

so we can write the divergence as

$$D = \frac{1}{r\cos\phi}\frac{1}{\sqrt{g}}\left[\tilde{d}_{hor}+\tilde{d}_{vert}\right]. \tag{7.13}$$

Furthermore we define

$$Z = Z_{hor}+Z_{vert} \tag{7.14}$$

with

$$Z_{hor} := Z_x+Z_y, \tag{7.15}$$

$$Z_x := \frac{\partial z}{\partial\lambda}\,u, \qquad Z_y := \frac{\partial z}{\partial\phi}\cos\phi\,v, \tag{7.16}$$

$$Z_{vert} := -r\cos\phi\,w. \tag{7.17}$$

The buoyancy term in the w-equation has a slightly different form than in the former COSMO versions. To derive it we use the ideal gas equation for moist air

$$p = \rho\,R_d\,(1+q_x)\,T, \tag{7.18}$$

where

$$q_x := \left(\frac{R_v}{R_d} - 1\right) q_v - q_{cond}, \tag{7.19}$$

describes the so-called 'water loading' in the buoyancy term. From this we can derive the buoyancy term, expressed by p' and T', without any approximation

$$\begin{aligned}
-g\frac{\rho'}{\rho} = -g\frac{1}{\rho}(\rho - \rho_0) &= -g\frac{1}{\rho}\left(\frac{p}{R_d(1+q_x)T} - \frac{p_0}{R_d T_0}\right) \\
&= -g\frac{\rho_0}{\rho}\left(\frac{T_0}{p_0}\frac{p}{(1+q_x)T} - 1\right) \\
&= -g\frac{\rho_0}{\rho(1+q_x)}\left(\frac{T_0}{p_0}\frac{p'+p_0}{T} - (1+q_x)\right) \\
&= -g\frac{\rho_0}{\rho(1+q_x)}\left(\frac{T_0}{T}\frac{p'}{p_0} + \frac{T_0 - T}{T} - q_x\right) \\
&= +g\frac{\rho_0}{\rho(1+q_x)}\left(-\frac{T_0}{T}\frac{p'}{p_0} + \frac{T'}{T} + q_x\right)
\end{aligned}$$

In former versions of the fast waves solver, $(1+q_x)$ was neglected in the denominator because q_x is at most about 1%. Nevertheless one can avoid this neglect by

$$-g\frac{\rho'}{\rho} = +g\frac{p_0}{T_0}\frac{T}{p}\left(-\frac{T_0}{T}\frac{p'}{p_0} + \frac{T'}{T} + q_x\right) = +g\left(-\frac{p'}{p} + \frac{p_0}{p}\frac{T'}{T_0} + \frac{p_0}{p}\frac{T}{T_0}q_x\right), \tag{7.20}$$

i.e. densities are expressed by pressure and temperature in an efficient manner. By the way, inserting $T' = T - T_0$ results in

$$-g\frac{\rho'}{\rho} = +g\left(-1 + \frac{p_0}{p}\frac{T}{T_0}(1+q_x)\right). \tag{7.21}$$

This form is used for the 'dynamical bottom boundary condition' (section 7.2.2). The reason is, that for parallelization no boundary exchange of T' is needed; instead an estimation of T by the starting value is used.

7.1 Slow processes in the Runge-Kutta scheme

In the splitting idea of Wicker and Skamarock (2002) the tendency of the slow processes is calculated and added in each sub step of the fast processes. The methodology of the subcycling is described in section 3.2, with the difference, that now the deviation T' is used as a prognostic variable, instead of the absolute temperature T.

The horizontal advection of the dynamical variables is done by an upwind scheme of 5th order, where the tendencies in x and y-direction are added. To stabilize this scheme a Runge-Kutta RK3 time integration scheme is used as proposed in Wicker and Skamarock (2002). This scheme is formally of 2nd order but for linear problems of 3rd order and its combination with the 5th order advection is one of the most effective advection schemes of this type Baldauf (2008). The vertical advection tendencies are calculated by an implicit scheme with centered differences of 2nd order.

Here some sort of a flow diagram of the Runge-Kutta solver is sketched:

$$
\begin{aligned}
\text{solve: } \frac{\tilde{\Phi}-\Phi^n}{\frac{\Delta t}{3}} &= \beta A_z(\tilde{\Phi}) + (1-\beta)A_z(\Phi^n) \\
&\quad + A_x(\Phi^n) + A_y(\Phi^n) + C(\Phi^n) + P(\Phi^n) \\
\text{... and define its tendency: } L(\Phi^n) &:= \frac{\tilde{\Phi}-\Phi^n}{\frac{\Delta t}{3}} \\
\text{1. RK-substep: } \Phi^* &= \Phi^n + \frac{\Delta t}{3} L(\Phi^n) \\
\text{fast waves with slow tendency } S^* &:= \frac{\Phi^*-\Phi^n}{\Delta t/3}, \text{ starting at } \Phi^n \Rightarrow \Phi^* \\
\text{solve: } \frac{\tilde{\Phi}-[\alpha\Phi^n+(1-\alpha)\Phi^*]}{\frac{\Delta t}{2}} &= \beta A_z(\tilde{\Phi}) + (1-\beta)A_z(\Phi^*) \\
&\quad + A_x(\Phi^*) + A_y(\Phi^*) + C(\Phi^*) + P(\Phi^n) \\
\text{... and define its tendency: } L(\Phi^*) &:= \text{lhs. of the above expression}^1 \\
\text{2. RK-substep: } \Phi^{**} &= \Phi^n + \frac{\Delta t}{2} L(\Phi^*) \\
\text{fast waves with slow tendency } S^{**} &:= \frac{\Phi^{**}-\Phi^n}{\Delta t/2}, \text{ starting at } \Phi^n \Rightarrow \Phi^{**} \\
\text{solve: } \frac{\tilde{\Phi}-[\alpha\Phi^n+(1-\alpha)\Phi^{**}]}{\Delta t} &= \beta A_z(\tilde{\Phi}) + (1-\beta)A_z(\Phi^{**}) \\
&\quad + A_x(\Phi^{**}) + A_y(\Phi^{**}) + C(\Phi^{**}) + P(\Phi^n) \\
\text{... and define its tendency: } L(\Phi^{**}) &:= \text{lhs. of the above expression} \\
\text{3. RK-substep: } \Phi^{n+1} &= \Phi^n + \Delta t\, L(\Phi^{**}) \\
\text{fast waves with slow tendency } S^{***} &:= \frac{\Phi^{n+1}-\Phi^n}{\Delta t}, \text{ starting at } \Phi^n \Rightarrow \Phi^{n+1}
\end{aligned}
$$

A_x and A_y are spatial discretizations of the horizontal advection where an upwind scheme of 5th order is used Wicker and Skamarock (2002) more general operators are described in section 7.1.2). A_z is the spatial discretization of the vertical advection processes, a 2nd order centered difference scheme. For the weighting of the implicitness of the vertical advection $\beta = 1/2$ was chosen which therefore constitutes a true Crank-Nicholson scheme. C denotes the Coriolis terms and metric correction terms of the advection due to the earth curvature in the momentum equations. $P(\Phi^n)$ contains the tendencies of the physical parameterizations (with the exception of the microphysics) and are calculated once outside of the RK-scheme. The general stability properties of this time splitting together with the fast waves processes are inspected in Baldauf (2010).

Generally $\alpha = 1$ is chosen. This alters the property of the implicit scheme, but proved to be stable in a broad range of vertical advection Courant numbers, yet not been unconditionally stable.

The fast waves calculation consists of several steps with a small time step $\Delta\tau$:

$$\Phi^{\nu+1} = \Phi^{\nu} + \Delta\tau\, F(\Phi^{\nu}, \Phi^{\nu+1}) + \Delta\tau\, S \tag{7.22}$$

F is an abbreviation for the vertical implicit, horizontal forward-backward scheme for the sound and gravity wave expansion. S is the (constant) tendency of the appropriate slow processes.

A further extension of the new dynamical core is the possibility to abandon the shallow atmosphere approximation Petrik (2006). In this approximation metrical terms of the velocity advection, e.g. $uw \tan \phi / r$, and Coriolis terms, e.g. $2\Omega w \cos \phi$, which contain the vertical velocity are neglected in comparison to similar terms which contain the horizontal velocity component. Now all advection and Coriolis terms can be used. Whereas the advection terms seem to have no significant influence, this statement is not as clear for the Coriolis terms. At least in the vicinity of the equator these terms are even bigger than the traditionally used terms.

7.1.1 Some implementation details of the Runge-Kutta scheme

The following steps are performed in subroutine `org_runge_kutta` (in `src_runge_kutta.f90`):

- At the beginning the tendencies of the Rayleigh friction layer and the tendencies of the physical parameterizations of radiation, convection and the explicit parts of the turbulence (if 3D turbulence is switched on) are summed up:

```
utens  = Ray                + ut_conv            + 3Dturb_expl
vtens  = Ray                + vt_conv            + 3Dturb_expl
wtens  =                                         + 3Dturb_expl
ttens  =       sohr + thhr + tt_conv + (t_inc)   + 3Dturb_expl
pptens =
qvtens =                    + qv_conv            + 3Dturb_expl
```

- Then in `implicit_vert_diffusion_uvwt` the implicit (Crank-Nicholson) vertical turbulent tendencies are added to `utens`, `vtens`, `wtens`, `pptens` and `ttens`
- rename T to T'
- add Coriolis tendencies to `utens`, `vtens`, `wtens` (if this is not done in every RK-substep)
- Start of the Runge-Kutta substeps:
  - calculate the horizontal advection tendencies `uadvt`, `vadvt`, `wadvt`, `tadvt`, `ppadvt` (add. wcon) in subroutine `advection`
  - if not done above: add Coriolis tendencies to `uadvt`, `vadvt`, `wadvt`
  - if the implicit version of vertical advection is used, then call `complete_tendencies_uvwtpp`: Using the previously calculated tendencies from horizontal advection `uadvt`, ...) and the above mentioned tendencies due to physics and adiabatic processes of the dynamic variables (stored in `utens`, ...) the vertical advection is solved by a vertically implicit scheme. The resulting tendencies again are stored in `uadvt`, `vadvt`, `wadvt`, `ppadvt` and `tadvt`.
  - now insert these complete slow tendencies (`uadvt`, `vadvt`, `wadvt`, `ppadvt` and `tadvt` into the `fast_waves` solver.

  end of the Runge-Kutta substeps.
- if the latent heating from the previous time step was added, then substract it (it is later added in the cloud microphysics scheme))

- T now contains again $T_0 + T'$
- Advection of the moisture fields q_x, aerosol and gaseous components of COSMO-ART, and the TKE (in subroutine `advection_pd`)
- and then in an operator splitting manner perform the horizontal diffusion of q_x and the implicit vertical diffusion of q_v, q_c, q_i and TKE
- apply artificial diffusion:
  - apply targeted diffusion on T' to avoid cold pools
  - apply artificial horizontal diffusion to u, v, w, T', p'
  - if chosen: apply horizontal Smagorinsky diffusion on u and v
- call the saturation adjustment (number of iterations depends from $\dot{\zeta}$)

7.1.2 Advection for the RK dynamical core

As mentioned in section 3.3.2 the advection terms for any scalar quantity s are

$$\frac{\partial s}{\partial t} + (\mathbf{v} \cdot \nabla)s = \frac{\partial s}{\partial t} + A(s, \mathbf{v}) = \dots \tag{7.23}$$

with a 'scalar' advection operator in spherical coordinates (eq. (2.66))

$$A(s, \mathbf{v}) = \frac{u}{r \cos\phi}\frac{\partial s}{\partial \lambda} + \frac{v}{r}\frac{\partial s}{\partial \phi} + w\frac{\partial s}{\partial r} \tag{7.24}$$

or in spherical and terrain-following coordinates

$$A(s, \mathbf{v}) = \frac{u}{r \cos\phi}\frac{\partial s}{\partial \lambda} + \frac{v}{r}\frac{\partial s}{\partial \phi} + \dot{\zeta}\frac{\partial s}{\partial \zeta}. \tag{7.25}$$

u, v, w are velocity components along spherical, normalized base vectors.

For the velocity components

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = \dots, \tag{7.26}$$

the equations can be written as

$$\frac{\partial u}{\partial t} + A(u, \mathbf{v}) - \frac{uv}{r}\tan\phi + \frac{uw}{r} = \dots, \tag{7.27}$$

$$\frac{\partial v}{\partial t} + A(v, \mathbf{v}) + \frac{u^2}{r}\tan\phi + \frac{vw}{r} = \dots, \tag{7.28}$$

$$\frac{\partial w}{\partial t} + A(w, \mathbf{v}) - \frac{u^2 + v^2}{r} = \dots. \tag{7.29}$$

The rightmost terms in every equation can be switched off in COSMO, which belongs to the shallow atmosphere approximation.

According to eq. (2.112) the contravariant vertical velocity sounds

$$\dot{\zeta} = \frac{u}{r \cos\phi}\frac{\partial \zeta}{\partial \lambda} + \frac{v}{r}\frac{\partial \zeta}{\partial \phi} - \frac{1}{\sqrt{G}}w. \tag{7.30}$$

With the transformation rules (2.100) this can be rewritten to

$$\dot{\zeta} = \frac{1}{\sqrt{G}} \left(\frac{u}{r \cos\phi} \frac{\partial z}{\partial \lambda} + \frac{v}{r} \frac{\partial z}{\partial \phi} \right) - \frac{1}{\sqrt{G}} w. \tag{7.31}$$

The first term in (...) on the r.h.s is obviously a horizontal advection of the 'height-field' $z(x, y, \zeta)$ and therefore can be treated with the following advection operators.

Horizontal advection operators

Therefore we need horizontal advection operators in which the 'velocities' $u/(r\cos\phi)$ and v/r are inserted. In Wicker and Skamarock (2002) the advection operators in flux form are described. In the current version of the COSMO-model the above mentioned advection form is used. Therefore we determine the appropriate advection operators by comparing for a constant velocity u

$$\frac{\partial uq}{\partial x} \rightarrow \frac{F_{i+1/2} - F_{i-1/2}}{\Delta x} \tag{7.32}$$

$$u \frac{\partial q}{\partial x} \rightarrow A_i \tag{7.33}$$

In the follwing we list the available horizontal advection operators.

advection operator of 2nd order

$$A_i^{(2)} = \frac{\bar{u}_i}{2\,\Delta x} \left(q_{i+1} - q_{i-1} \right), \tag{7.34}$$

advection operator of 1st order

$$A_i^{(1)} = A_i^{(2)} - S \frac{|\bar{u}_i|}{2\,\Delta x} \left(q_{i+1} - 2q_i + q_{i-1} \right), \tag{7.35}$$

advection operator of 4th order

$$A_i^{(4)} = \frac{\bar{u}_i}{12\,\Delta x} \left(8(q_{i+1} - q_{i-1}) - (q_{i+2} - q_{i-2}) \right), \tag{7.36}$$

advection operator of 3rd order

$$A_i^{(3)} = A_i^{(4)} + S \frac{|\bar{u}_i|}{12\,\Delta x} \left(q_{i+2} - 4q_{i+1} + 6q_i - 4q_{i-1} + q_{i-2} \right), \tag{7.37}$$

advection operator of 6th order

$$A_i^{(6)} = \frac{\bar{u}_i}{60\,\Delta x} \left(45(q_{i+1} - q_{i-1}) - 9(q_{i+2} - q_{i-2}) + (q_{i+3} - q_{i-3}) \right), \tag{7.38}$$

advection operator of 5rd order

$$A_i^{(5)} = A_i^{(6)} - S \frac{|\bar{u}_i|}{60\,\Delta x} \left((q_{i+3} + q_{i-3}) - 6(q_{i+2} + q_{i-2}) + 15(q_{i+1} + q_{i-1}) - 20q_i \right). \tag{7.39}$$

S =sgn Δt denotes the sign of the timestep. The even order advection operators are pure centered difference schemes, whereas the odd order operators can be written as a sum of the next higher even order operator and a diffusion term.

The factor S is needed for the digital filtering initialization (DFI), where some timesteps into the past (i.e. with a negative timestep) are made. Without this factor the diffusion term would have a negative diffusion coefficient and therefore would become unstable.

The calculation of the averaged velocities $\bar{u}_i$ is done in dependence from the advected variable: for the advected velocity component u, one calculates $\bar{u}_{i+1/2} = \frac{1}{3}(u_{i-1/2} + u_{i+1/2} + u_{i+3/2})$. This procedure stabilizes momentum advection in sheared velocity fields.

For the advection of the scalar variables one can use $\bar{u}_i = \frac{1}{2}(u_{i-1/2} + u_{i+1/2})$. Recently U. Blahak found that for the odd order advection schemes it is recommendable to use this formula only in front of the centered difference advection operators,whereas one should use $|\bar{u}_i| := \frac{1}{2}(|u|_{i-1/2} + |u|_{i+1/2})$ in front of the diffusion operators.

In combination with the RK3 scheme the advection operator of 3rd order is stable for Courant numbers up to 1.62, advection 5th order ist stable up to 1.42 Wicker and Skamarock (2002), for an extended stability analysis see Baldauf (2008)).

For the two horizontal directions the tendencies of the two one-dimensional advection operators are added in every RK-substep (i.e. no operatorsplitting in the different directions).

Vertical advection operator

The vertical advection is done in an implicit manner using an advection operator

$$A_{z,k}(q) = \frac{d\zeta}{dt} \frac{q_{i,j,k+1} - q_{i,j,k-1}}{2\Delta\zeta} \tag{7.40}$$

7.1.3 Coriolis terms

The Coriolis terms in the spherical, terrain-following coordinate system are

$$\frac{\partial u}{\partial t} = \ldots + fv - f_c w \tag{7.41}$$

$$\frac{\partial v}{\partial t} = \ldots - fu \tag{7.42}$$

$$\frac{\partial w}{\partial t} = \ldots \quad + f_c w \tag{7.43}$$

with $f = 2|\mathbf{\Omega}| \sin\phi$ and $f_c = 2|\mathbf{\Omega}| \cos\phi$. Neglecting terms f_c is part of the 'shallow atmosphere' approximation. (see e.g. Petrik (2006) and references therein).

In the leapfrog dynamical core, the tendencies of the Coriolis terms are simply added to the tendencies of the physical parameterizations. This is formally not correct in the Runge-Kutta-scheme, because Coriolis terms would be integrated by an Euler forward (EF) scheme. The EF-scheme for the Coriolis terms is well known to be unstable with an amplification factor $1 + i\ \Delta t\ f$. Obviously, this is a very week instability: e.g. if a time step of 66 sec. is used, the amplitude of the variables increases by a factor of 10 % in about 4500 time steps (=78 h). Such slow instabilities are not so important for weather forecasting applications but possibly for long lasting climate runs.

Nevertheless, from a formal point of view a stable integration of the Coriolis terms should be achieved, that means, their tendencies should be calculated in every RK-substep like the horizontal advection terms. (this is done, if the internal parameter `l_Coriolis_every_RK_substep`

= .TRUE., whereas the 'leapfrog-like' treatment similar to the physical tendencies is chosen by .FALSE.). Moreover, despite the fact, that some spatial averaging is needed, the calculation of the tendencies is a very cheap operation. Every call of the subroutine `coriolis` takes currently about 0.2 % of a model run time.

It should be remarked that a 2-stage Runge-Kutta scheme is unstable, too. Only RK-schemes with order equal or higher 3 can stably integrate at least the shallow atmosphere Coriolis terms (precisely all so-called LC-RK-schemes Baldauf (2008) of order 3 or higher can provide this).

7.2 Fast processes in the Runge-Kutta scheme

The basic methodology to integrate the fast waves is mainly described in section 4.3.2. The main difference here is the use of T' instead of T as a prognostic variable. This results in an additional fast term $w\ \partial T_0/\partial z$, which contributes to the 'buoyancy' terms.

Since July 2012 (COSMO version 4.24), the fast waves solver again has been revised. The main original goals of this revision are the following:

- The consideration of the vertical grid stretching by introduction of appropriate weightings in *all* vertical discretizations of the fast waves solver. In particular their use in the *implicit* terms in the discretization of the Euler equations requires a complete re-derivation of the tridiagonal equation system.
- The usage of the 'strong conservation form' for the divergence operator. Though mainly developed for finite volume schemes with conservation properties, there was the hope to gain advantages by a more direct discretization of the metric terms and by a better formulation of the lower boundary conditions.
- The option for a fully 3-dimensional (3D) isotropic divergence damping instead of the 'traditional' quasi-3D version should be available. Gaßmann and Herzog (2007) have derived the dispersion relation of sound and gravity waves and have found a larger deviation from the correct one in the case of the quasi-3D version compared to the isotropic version.
- A further option should be the alternative discretization of the horizontal pressure gradients by the methodology of Mahrer (1984). This z-plane treatment should result in a more stable behavior in steep terrain.

The new module *fast_waves_sc* additionally contains other, smaller improvements, too, concerning e.g. the formulation of boundary conditions, the use of the reference state, a more accurate formulation of the buoyancy term. It should be emphasized, that this whole section is mainly an excerpt of the according COSMO technical report Baldauf (2013).

7.2.1 Integration of the 'fast waves'

To write down the discretization of the fast part of the Euler-equations (7.1)-(7.5) we define the following numerical spatial two-point operators, leading to at most second order formulas.

Here we will keep the half indices for the staggered velocity positions for clarification. One should notice that in the *COSMO code* the staggered grid positions $(i+\frac{1}{2},j,k)$, $(i,j+\frac{1}{2},k)$, and $(i,j,k-\frac{1}{2})$ (minus sign!) are denoted as `(i,j,k)` (this is indicated by the gray connection lines in Figure 7.1).

Horizontal averaging is done by

$$\overline{\psi}^{\lambda}\Big|_{i,j,k} \equiv A_\lambda \psi|_{i,j,k} := \frac{1}{2}(\psi_{i-\frac{1}{2},j,k} + \psi_{i+\frac{1}{2},j,k}), \tag{7.44}$$

$$\overline{\psi}^{\phi}\Big|_{i,j,k} \equiv A_\phi \psi|_{i,j,k} := \frac{1}{2}(\psi_{i,j-\frac{1}{2},k} + \psi_{i,j+\frac{1}{2},k}). \tag{7.45}$$

For the vertical averaging we have to keep in mind the special definition of the so called main levels (index k) and the half levels (index $k+\frac{1}{2}$). As stated above, in the COSMO model, the user can prescribe the grid positions of the half levels $z^{(h)}_{i,j,k-\frac{1}{2}}$. The height of the main levels is a simple arithmetic average

$$z^{(m)}_{i,j,k} := \frac{1}{2}\left(z^{(h)}_{i,j,k-\frac{1}{2}} + z^{(h)}_{i,j,k+\frac{1}{2}}\right). \tag{7.46}$$

Accordingly an averaging from half level variables (e.g. w) to the main level is done by arithmetic averaging:

$$\overline{\psi}^{\zeta}\Big|_{i,j,k} \equiv A_\zeta \psi|_{i,j,k} := \frac{1}{2}(\psi_{i,j,k-\frac{1}{2}} + \psi_{i,j,k+\frac{1}{2}}). \tag{7.47}$$

But averaging from main level variables (e.g. p', T', ...) to the half level position is done by a weighting

$$\overline{\psi}^{\zeta,N}\Big|_{i,j,k-\frac{1}{2}} \equiv A^N_\zeta \psi\Big|_{i,j,k-\frac{1}{2}} := g_{i,j,k-\frac{1}{2}}\psi_{i,j,k} + (1-g_{i,j,k-\frac{1}{2}})\psi_{i,j,k-1} \tag{7.48}$$

with

$$g_{i,j,k-\frac{1}{2}} := \frac{z^{(h)}_{i,j,k-\frac{1}{2}} - z^{(h)}_{i,j,k-\frac{3}{2}}}{z^{(h)}_{i,j,k+\frac{1}{2}} - z^{(h)}_{i,j,k-\frac{3}{2}}}. \tag{7.49}$$

To be complete, we also note the extrapolation formula

$$\psi_{i,j,k+\frac{1}{2}} = -(1-g_{i,j,k-\frac{1}{2}})\,\psi_{i,j,k-1} + (2-g_{i,j,k-\frac{1}{2}})\,\psi_{i,j,k}. \tag{7.50}$$

For vertical averages for u and v one has to use appropriate weights $g_{i+\frac{1}{2},j,k-\frac{1}{2}}$ or $g_{i,j+\frac{1}{2},k-\frac{1}{2}}$, respectively. The appropriate averaging operators are denoted by a bar above the N: $\overline{A^N_\zeta}$ (of course one should even distinguish between the u or v position; but this can be easily seen from the context). In the program code the denotations $g_{i,j,k-\frac{1}{2}}$ = `wgtfac(i,j,k)` , $g_{i+\frac{1}{2},j,k-\frac{1}{2}}$ = `wgtfac_u(i,j,k)` and $g_{i,j+\frac{1}{2},k-\frac{1}{2}}$ = `wgtfac_v(i,j,k)` (introduced by G. Zängl) are used.

Now we turn to spatial *derivatives*. For horizontal derivatives we can simply use the standard centered difference formulas

$$\delta_\lambda \psi|_{i,j,k} := \frac{\psi_{i+\frac{1}{2},j,k} - \psi_{i-\frac{1}{2},j,k}}{\Delta\lambda}, \tag{7.51}$$

$$\delta_\phi \psi|_{i,j,k} := \frac{\psi_{i,j+\frac{1}{2},k} - \psi_{i,j-\frac{1}{2},k}}{\Delta\phi}, \tag{7.52}$$

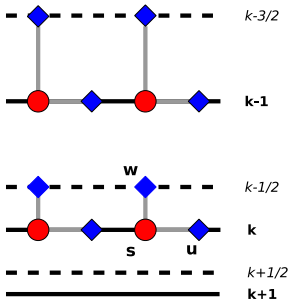

Figure 7.1: Denotation of the levels and the position of variables in the COSMO-model. Grey connection lines indicate staggered grid positions with the same indices in the COSMO program code.

and analogous for staggered variables.

In the case of vertical derivatives things become a little bit more complicated. Here one has to distinguish if the variable is either defined on the half or the main level and if the target point is on a half or a main level. The derivative of a half level variable at the position of the main level is obviously done by

$$\delta_\zeta \psi|_{i,j,k} := \frac{\psi_{i,j,k+\frac{1}{2}} - \psi_{i,j,k-\frac{1}{2}}}{\Delta\zeta}. \tag{7.53}$$

The derivative of a main level variable to the half level position could be done in the same way

$$\delta_\zeta^N \psi\Big|_{i,j,k-\frac{1}{2}} := \frac{\psi_{i,j,k} - \psi_{i,j,k-1}}{\Delta\zeta} \tag{7.54}$$

with the argumentation, that from two neighbouring points one can calculate a derivative only in one manner, independently from the position of the target point. But a second order formula can only be achieved if the target point is exactly in between the two main level points. For a decentered target point, the formula is only of first order accurate. Therefore, one should avoid this type of derivation operator, if possible.

Vertical derivative of a 'scalar variable to the u-Position':

$$\delta_\zeta^{(s,u)} \psi := A_\lambda \delta_\zeta A_\zeta^N \psi \tag{7.55}$$

For the vertical derivative 'u-Position to a scalar point' one can use

$$\delta_\zeta^{(u,s)} u := A_\lambda \delta_\zeta A_\zeta^{\bar{N}} u \tag{7.56}$$

analogous to eq. (7.55). Vertical derivatives 'scalar to v-position' or 'v-position to scalar' are discretized analogously.

Now we define operators for combinations or products of vertical derivatives. Such larger stencils occur in particular in the metric correction terms. For the grid position definition of the metric terms one has to notice that the half level positions $z^{(h)}$ (or 'hhl'), i.e. with the 'w'-position indices $(i, j, k - \frac{1}{2})$ are prescribed. Therefore, $\frac{\partial z}{\partial \zeta}$ and $\frac{\partial \zeta}{\partial z}$ or $\sqrt{g}$ and $1/\sqrt{g}$ are defined most naturally at the scalar ('s') position (i, j, k) (in this context it might by denoted

equally as the 'ww'-position), $\frac{\partial z}{\partial \lambda}$ is defined most naturally at the 'uw'-position $(i+\frac{1}{2}, j, k-\frac{1}{2})$, and $\frac{\partial z}{\partial \phi}$ most naturally at the 'vw'-position $(i, j+\frac{1}{2}, k-\frac{1}{2})$.

The following metric term occurring in the 'strong conservation'-form of the divergence is best discretized as

$$discr.^{(s)}\left[\frac{\partial z}{\partial \lambda}u\right] = \delta_\zeta\left(A_\lambda\left(\frac{\partial z}{\partial \lambda}A_\zeta^{\bar{N}}u\right)\right). \tag{7.57}$$

Analogous for the ϕ-direction.

Metric terms of ∇p and ∇D are discretized by

$$discr.^{(u)}\left[\frac{\partial \zeta}{\partial \lambda}\ \frac{\partial \psi}{\partial \zeta}\right] = -\left(A_\zeta\frac{\partial z}{\partial \lambda}\right)\cdot\left(A_\lambda\underbrace{\left(\frac{\partial \zeta}{\partial z}\delta_\zeta A_\zeta^N\psi\right)}_{=discr.^{(s)}\left[\frac{\partial \psi}{\partial z}\right]}\right). \tag{7.58}$$

This means that a ζ-derivative can be expressed most naturally by a z-derivative (again analogous for the ϕ-direction).

For the z-derivative of a scalar variable at the s-position we use the following

$$discr.^{(s)}\left[\frac{\partial \psi}{\partial z}\right] \equiv \delta_z^{(s,s)}\psi := \frac{\partial \zeta}{\partial z}\delta_\zeta A_\zeta^N\psi. \tag{7.59}$$

The basic time integration idea behind the fast waves solver is the horizontally explicit-vertically implicit (HE-VI) scheme, in which the horizontal integration is further done by a forward-backward scheme and the vertical implicit step is a general Crank-Nicholson scheme Klemp and Wilhelmson (1978). It is therefore reasonable to define a general time averaging operator, here called a 'Crank-Nicholson time averaging operator', by

$$\hat{\beta}_i^{(a)}\phi := \beta_i^{(a)}\phi^{n+1} + \left(1-\beta_i^{(a)}\right)\phi^n. \tag{7.60}$$

The upper index (a) denotes a process, e.g. (s) for sound expansion terms, the lower index i simply enumerates terms. This nomenclature for the Crank-Nicholson weights is adopted from Baldauf (2010). To limit the number of weights $\beta_i^{(a)}$, not for every term an own Crank-Nicholson weighting parameter is introduced. u- and v-terms, which are analogous in a process, get the same weighting parameter, because the both horizontal directions have often quite equal rights in real model applications and their appropriate grid stretching in a limited area model are not so different.

Now, the discretized Euler equations are expressed by the formerly defined operators:

$$\begin{aligned}
\frac{u^{n+1}-u^n}{\Delta t} &= -\overline{\frac{1}{\rho}}^{\lambda}\ \overline{\frac{1}{r\cos\phi}}^{\lambda}\left(\hat{\beta}_1^s\delta_\lambda p' - \hat{\beta}_7^s\overline{\frac{\partial z}{\partial \lambda}}^{\zeta}A_\lambda\left(\delta_z^{(s,s)}p'\right)\right) + \\
&\quad +\overline{\frac{1}{\rho}}^{\lambda}\ \overline{\frac{1}{r\cos\phi}}^{\lambda}\left(\delta_\lambda(\alpha_{div}^h\rho D_{(uv)}) - \overline{\frac{\partial z}{\partial \lambda}}^{\zeta}A_\lambda\left(\delta_z^{(s,s)}(\alpha_{div}^h\rho D_{(uv)})\right)\right) + \left.\frac{\partial u}{\partial t}\right|_{slow} \qquad (7.61)\\
\frac{v^{n+1}-v^n}{\Delta t} &= -\overline{\frac{1}{\rho}}^{\phi}\ \overline{\frac{1}{r}}^{\phi}\left(\hat{\beta}_1^s\delta_\phi p' - \hat{\beta}_7^s\overline{\frac{\partial z}{\partial \phi}}^{\zeta}A_\phi\left(\delta_z^{(s,s)}p'\right)\right) +
\end{aligned}$$

$$+\overline{\frac{1}{\rho}}^{\phi}\,\overline{\frac{1}{r}}^{\phi}\left(\delta_\phi(\alpha^h_{div}\rho D_{(uv)})-\overline{\frac{\partial z}{\partial\phi}}^{\zeta}A_\phi\left(\delta_z^{(s,s)}(\alpha^h_{div}\rho D_{(uv)})\right)\right)+\left.\frac{\partial v}{\partial t}\right|_{slow} \tag{7.62}$$

$$\begin{aligned}\frac{w^{n+1}-w^n}{\Delta t} &= -\overline{\frac{1}{\rho}}^{\zeta,N}\left(\hat\beta_2^s\overline{\frac{\partial\zeta}{\partial z}}^{\zeta,N}\delta_\zeta^N p'\right)+\\ &\quad +g\left(\hat\beta_1^b\overline{\frac{p_0}{p}}^{\zeta,N}\overline{\frac{1}{T_0}}^{\zeta,N}A_\zeta^N T'-\hat\beta_2^b\overline{\frac{1}{p}}^{\zeta,N}A_\zeta^N p'+\overline{\frac{p_0}{p}}^{\zeta,N}\overline{\frac{T}{T_0}}^{\zeta,N}\overline{q_x}^{\zeta,N}\right)+\\ &\quad +\overline{\frac{1}{\rho}}^{\zeta,N}\overline{\frac{\partial\zeta}{\partial z}}^{\zeta,N}\delta_\zeta^N(\alpha^v_{div}\rho D_{(w)})+\left.\frac{\partial w}{\partial t}\right|_{slow}\end{aligned} \tag{7.63}$$

$$\frac{p'^{n+1}-p'^n}{\Delta t} = -\frac{c_p}{c_V}pD_{(p)}+\hat\beta_3^b g\rho_0\,A_\zeta w+\left.\frac{\partial p'}{\partial t}\right|_{slow} \tag{7.64}$$

$$\frac{T'^{n+1}-T'^n}{\Delta t} = -\frac{R}{c_V}TD_{(T)}-\hat\beta_4^b\frac{\partial T_0}{\partial z}\,A_\zeta w+\left.\frac{\partial T'}{\partial t}\right|_{slow} \tag{7.65}$$

Here, different divergences $D_{(p)}$, $D_{(T)}$, $D_{(uv)}$, $D_{(w)}$ have been introduced, because they can carry different Crank-Nicholson weights (see section 7.2.1). Be aware, that each variable has a distinct time index, too, by the Crank-Nicholson operators $\hat\beta$. Only the additional slow processes $\left.\frac{\partial u}{\partial t}\right|_{slow},\ldots,\left.\frac{\partial T'}{\partial t}\right|_{slow}$ don't carry any time index: they are assumed to be constant during the small time step integration.

In deviation from the above defined operators, the following coefficient functions occurring in the buoyancy terms are calculated by

$$\overline{\frac{1}{p}}^{\zeta,N} = \frac{1}{p_0(z_{k+1/2})+A_\zeta^N p'},\qquad \overline{\frac{p_0}{p}}^{\zeta,N}=\frac{1}{1+\frac{A_\zeta^N p'}{p_0(z_{k+1/2})}}, \tag{7.66}$$

$$\overline{\frac{1}{T_0}}^{\zeta,N} = \frac{1}{T_0(z_{k+1/2})},\qquad \overline{\frac{T}{T_0}}^{\zeta,N}=1+\frac{A_\zeta^N T'}{T_0(z_{k+1/2})}. \tag{7.67}$$

The writing for the reference state variables T_0 and p_0 indicates that they can be calculated *exactly* at their grid position[2] (proposal by A. Will (personal communication)).

Discretization of the divergence The horizontal contributions of the divergence can be discretized by

$$\tilde d_{hor} := \delta_\lambda(\overline{\sqrt{g}}^{\lambda}u)+\delta_\phi(\overline{\cos\phi}^{\phi}\,\overline{\sqrt{g}}^{\phi}v). \tag{7.68}$$

For the vertical contribution

$$\tilde d_{vert} := \delta_\zeta Z, \tag{7.69}$$

Z, Z_x, Z_y, Z_{hor} and Z_{vert} are needed. They are most naturally defined at the w-position $(i,j,k-\frac{1}{2})$. However, one cannot directly use $\tilde d_{vert}$, because the implicit weighting must be considered. Z_{hor} is calculated in the subroutine `calc_Z_horiz`, and discretized by

$$Z_{hor} = Z_x+Z_y, \tag{7.70}$$

$$Z_x = A_\lambda\left(\frac{\partial z}{\partial\lambda}\,A_\zeta^{\bar N}\,u\right), \tag{7.71}$$

[2] Therefore the reference state variables have to be calculated not only on the main levels but on the half level positions, too. This must be done in all modules, which calculate them. In particular in the interpolation program `int2lm` one has to set the namelist switch `lanalyt_calc_p0T0=.TRUE.`, if `irefatm=1` is still used.

$$Z_y \quad = \quad A_\phi \left(\frac{\partial z}{\partial \phi} \cos\phi \, A_\zeta^{\bar{N}} \, v \right). \tag{7.72}$$

In this manner Z_{hor} can be calculated for $k = 2, 3, ..., ke$. At $k = 1$ (upper boundary) one immediately gets $Z_{hor} = 0$ (due to $\frac{\partial z}{\partial \lambda} = 0$, ...). At $k = ke + 1$ (lower boundary) u and v must be extrapolated. In contrast

$$Z_{vert} := -r \cos\phi \, w \tag{7.73}$$

can be calculated at the boundaries, too.

The different divergence terms are:

$$D_{(p)} = \frac{1}{r\cos\phi} \frac{1}{\sqrt{g}} \left[\hat{\beta}_3^s \, \tilde{d}_{hor} + \hat{\beta}_8^s \, \delta_\zeta Z_{hor} + \hat{\beta}_4^s \, \delta_\zeta Z_{vert} \right]. \tag{7.74}$$

$$D_{(T)} = \frac{1}{r\cos\phi} \frac{1}{\sqrt{g}} \left[\hat{\beta}_5^s \, \tilde{d}_{hor} + \hat{\beta}_9^s \, \delta_\zeta Z_{hor} + \hat{\beta}_6^s \, \delta_\zeta Z_{vert} \right]. \tag{7.75}$$

$$D_{(uv)} = \frac{1}{r\cos\phi} \frac{1}{\sqrt{g}} \left[\hat{\beta}_1^d \, \tilde{d}_{hor} + \hat{\beta}_5^d \, \delta_\zeta Z_{hor} + \hat{\beta}_2^d \, \delta_\zeta Z_{vert} \right]. \tag{7.76}$$

$$D_{(w)} = \frac{1}{r\cos\phi} \frac{1}{\sqrt{g}} \left[\hat{\beta}_3^d \, \tilde{d}_{hor} + \hat{\beta}_6^d \, \delta_\zeta Z_{hor} + \hat{\beta}_4^d \, \delta_\zeta Z_{vert} \right]. \tag{7.77}$$

Mahrer discretization of the horizontal pressure gradients The basic idea behind the discretization of the horizontal pressure gradient by Mahrer (1984) consists in using the gradient $\frac{\partial p'(\lambda,z)}{\partial \lambda}$ on z-planes instead of the conventional form $\frac{\partial p'(\lambda,\zeta)}{\partial \lambda} + \frac{\partial \zeta}{\partial \lambda} \frac{\partial p'(\lambda,\zeta)}{\partial \zeta}$ (equivalently for the ϕ-direction). To this purpose p' is interpolated vertically on the left- and right hand side of the target u-position. This means, at both columns (i, j) and $(i+1, j)$ p' is interpolated at the height $z_{i+\frac{1}{2},j,k}$ from the vertically nearest p'-values (analogous to the λ-direction). In this manner, the nearest values of p' are used in steep terrain. In contrast, in the above mentioned 'conventional' discretization, it can happen, that $\frac{\partial p'(\lambda,\zeta)}{\partial \zeta}$ is estimated from p'-values, which are (vertically) quite far away. This bears the risk of an instability, which does not occur in the Mahrer-approach.

A crucial point in the Mahrer discretization lies in the fact, that in the vicinity of steep terrain, interpolation is not longer possible, when $z_{i+\frac{1}{2},j,k}$ lies under the orography at least on one side of the u-position. As pointed out by Zängl (2012), the then required *extrapolation* of p' often leads to similar instabilities as the conventional discretization. Zängl (2012) describes a way, how to estimate the extrapolation by the hydrostatic approximation in the ICON dynamical core. Unfortunately, the need for a (quasi-)3-dimensional divergence damping necessary in the time-splitting approach (Skamarock and Klemp (1992), Baldauf (2010)) does not allow to transfer this idea to the COSMO fast waves solver (there is no hydrostatic approximation for the divergence).

However, in the new fast waves solver, an also linear extrapolation is used if $z_{i,j,ke} > z_{i+\frac{1}{2},j,k}$ (similar for the other neighboring orography heights). This indeed can help to increase stability in steep terrain in idealized test scenarios. But in real case applications, this approach

is not entirely satisfying. One reason could be the occurring of a mixed term in the Taylor expansion (see Baldauf (2013)). Therefore, though the Mahrer discretization has proven as a stable method in several real case runs (mainly for COSMO-DE), the conventional discretization is still recommended.

The tridiagonal equation system for the vertical velocity

The detailed derivation of the implicit vertical equation for w is derived in Baldauf (2013) Here we cite only the resulting equations. First of all, one only ends up with a tridiagonal equation system, if the following requirements for the implicit weightings are fulfilled:

$$\beta_1^{(s)} = 0, \quad \beta_7^{(s)} = 0, \quad \beta_1^{(d)} = 0, \quad \beta_2^{(d)} = 0, \quad \beta_5^{(d)} = 0, \quad \beta_7^{(d)} = 0. \tag{7.78}$$

This simply means, that the equations for u and v, (7.61) and (7.62), are solved explicitly, i.e. in a pure forward sense. This can be done at the beginning of the small time step. As said above, the updated values u^{n+1}, v^{n+1} are already available to formulate the boundary condition for the implicit equation system for w^{n+1}. Baldauf (2010) determines optimal values of the remaining off-centering weights by a stability analysis to

$$\beta_3^{(s)} = \beta_5^{(s)} = 1, \quad \beta_2^{(s)} = \beta_4^{(s)} = \beta_6^{(s)} = 0.7, \tag{7.79}$$

$$\beta_1^{(b)} = \beta_2^{(b)} = \beta_3^{(b)} = \beta_4^{(b)} = 0.7, \tag{7.80}$$

$$\beta_3^{(d)} = \beta_4^{(d)} = 1.0. \tag{7.81}$$

Further we use for the metric correction terms $\beta_8^{(s)} = \beta_4^{(s)}$, $\beta_9^{(s)} = \beta_6^{(s)}$, and $\beta_6^{(d)} = \beta_4^{(d)}$.

Finally the equation system can be formulated as a linear system of equations for w

$$A_{k-\frac{1}{2}} w_{k-\frac{3}{2}} + B_{k-\frac{1}{2}} w_{k-\frac{1}{2}} + C_{k-\frac{1}{2}} w_{k+\frac{1}{2}} = rhs_{k-\frac{1}{2}} \tag{7.82}$$

(A, B, C, rhs and w are further defined at (i,j)) with the coefficients

$$\begin{aligned}
A_{k-\frac{1}{2}} \;=\; & -a_{wp1\,|\,k-\frac{1}{2}} \cdot a_{pw1\,|\,k-1} \cdot r_c \\
& +a_{wp1\,|\,k-\frac{1}{2}} \cdot a_{pw2\,|\,k-1} \cdot \frac{1}{2} \\
& +a_{wp2\,|\,k-\frac{1}{2}} \cdot (1-g_{k-\frac{1}{2}}) \cdot a_{pw1\,|\,k-1} \cdot r_c \\
& -a_{wp2\,|\,k-\frac{1}{2}} \cdot (1-g_{k-\frac{1}{2}}) \cdot a_{pw2\,|\,k-1} \cdot \frac{1}{2} \\
& +a_{wT\,|\,k-\frac{1}{2}} \cdot (1-g_{k-\frac{1}{2}}) \cdot a_{Tw1\,|\,k-1} \cdot r_c \\
& -a_{wT\,|\,k-\frac{1}{2}} \cdot (1-g_{k-\frac{1}{2}}) \cdot a_{Tw2\,|\,k-1} \cdot \frac{1}{2} \\
& +\frac{1}{\Delta t}\, a_{ww1\,|\,k-\frac{1}{2}} \cdot a_{ww2\,|\,k-1} \cdot r_c
\end{aligned}$$

$$\begin{aligned}
B_{k-\frac{1}{2}} \;=\; & +a_{wp1\,|\,k-\frac{1}{2}} \cdot \left[a_{pw1\,|\,k} + a_{pw1\,|\,k-1}\right] \cdot r_c \\
& -a_{wp1\,|\,k-\frac{1}{2}} \cdot \left[a_{pw2\,|\,k} - a_{pw2\,|\,k-1}\right] \cdot \frac{1}{2} \\
& -a_{wp2\,|\,k-\frac{1}{2}} \cdot \left[-g_{k-\frac{1}{2}} \cdot a_{pw1\,|\,k} + (1-g_{k-\frac{1}{2}}) \cdot a_{pw1\,|\,k-1}\right] \cdot r_c
\end{aligned}$$

$$
\begin{aligned}
&-a_{wp2\,|\,k-\frac{1}{2}} \cdot \left[+g_{k-\frac{1}{2}} \cdot a_{pw2\,|\,k} + (1-g_{k-\frac{1}{2}}) \cdot a_{pw2\,|\,k-1}\right] \cdot \frac{1}{2} \\
&-a_{wT\,|\,k-\frac{1}{2}} \cdot \left[-g_{k-\frac{1}{2}} \cdot a_{Tw1\,|\,k} + (1-g_{k-\frac{1}{2}}) \cdot a_{Tw1\,|\,k-1}\right] \cdot r_c \\
&-a_{wT\,|\,k-\frac{1}{2}} \cdot \left[+g_{k-\frac{1}{2}} \cdot a_{Tw2\,|\,k} + (1-g_{k-\frac{1}{2}}) \cdot a_{Tw2\,|\,k-1}\right] \cdot \frac{1}{2} \\
&-\frac{1}{\Delta t}\, a_{ww1\,|\,k-\frac{1}{2}} \cdot \left[a_{ww2\,|\,k} + a_{ww2\,|\,k-1}\right] \cdot r_c \\
&+\frac{1}{\Delta t^2}
\end{aligned}
$$

$$
\begin{aligned}
C_{k-\frac{1}{2}} \;=\; & -a_{wp1\,|\,k-\frac{1}{2}} \cdot a_{pw1\,|\,k} \cdot r_c \\
& -a_{wp1\,|\,k-\frac{1}{2}} \cdot a_{pw2\,|\,k} \cdot \frac{1}{2} \\
& -a_{wp2\,|\,k-\frac{1}{2}} \cdot g_{k-\frac{1}{2}} \cdot a_{pw1\,|\,k} \cdot r_c \\
& -a_{wp2\,|\,k-\frac{1}{2}} \cdot g_{k-\frac{1}{2}} \cdot a_{pw2\,|\,k} \cdot \frac{1}{2} \\
& -a_{wT\,|\,k-\frac{1}{2}} \cdot g_{k-\frac{1}{2}} \cdot a_{Tw1\,|\,k} \cdot r_c \\
& -a_{wT\,|\,k-\frac{1}{2}} \cdot g_{k-\frac{1}{2}} \cdot a_{Tw2\,|\,k} \cdot \frac{1}{2} \\
& +\frac{1}{\Delta t}\, a_{ww1\,|\,k-\frac{1}{2}} \cdot a_{ww2\,|\,k} \cdot r_c
\end{aligned}
$$

using the denotations

$$a_{wp1} := \overline{\left(\frac{1}{\rho}\right)}^{\zeta,N} \overline{\frac{\partial\zeta}{\partial z}}^{\zeta,N} \beta_2^s, \qquad a_{wp2} := g\,\beta_2^b \overline{\left(\frac{1}{p}\right)}^{\zeta,N}, \tag{7.83}$$

$$a_{ww1} := \overline{\left(\frac{1}{\rho}\right)}^{\zeta,N} \overline{\frac{\partial\zeta}{\partial z}}^{\zeta,N}, \qquad a_{ww2} := \alpha_{div}^v\,\rho\,\frac{1}{r\cos\phi}\,\frac{1}{\sqrt{g}}\,\beta_4^d, \tag{7.84}$$

$$a_{wT} := -g\,\overline{\left(\frac{p_0}{p}\right)}^{\zeta,N} \overline{\left(\frac{1}{T_0}\right)}^{\zeta,N} \beta_1^b, \tag{7.85}$$

$$a_{pw1} := -\frac{c_p}{c_v}\,p\,\frac{1}{r\cos\phi}\,\frac{1}{\sqrt{g}}\,\beta_4^s, \qquad a_{pw2} := -\beta_3^b g\rho_0, \tag{7.86}$$

$$a_{Tw1} := -\frac{R}{c_v}\,T\,\frac{1}{r\cos\phi}\,\frac{1}{\sqrt{g}}\,\beta_6^s, \qquad a_{Tw2} := \beta_4^b \frac{\partial T_0}{\partial z}, \tag{7.87}$$

and

$$r_c := \overline{r\,\cos\phi}^{\zeta}. \tag{7.88}$$

In a_{pw1}, a_{Tw1}, and a_{ww2} one can cancel $r\cos\phi$ terms by the shallow atmosphere approximation and may set $r_c = 1$. (By efficiency reasons the averaging factor $1/2$ is defined into a_{pw2} and a_{Tw2} in the program code.)

The right hand side of eq. (7.82) reads

$$
\begin{aligned}
rhs \;:=\; & -\mathbf{A}_{wp} \underbrace{\left[b_p^{(n)} - \left(\frac{c_p}{c_v}\,p\,\frac{1}{r\cos\phi}\,\frac{1}{\sqrt{g}}\left[\beta_3^s\,\tilde{d}_{hor}^{n+1} + \beta_8^s\,\delta_\zeta Z_{hor}^{n+1}\right]\right)\right]}_{=:b_{4,p}} \\
& -\mathbf{A}_{wT}\left[b_T^{(n)} - \left(\frac{R}{c_v}\,T\,\frac{1}{r\cos\phi}\,\frac{1}{\sqrt{g}}\left[\beta_5^s\,\tilde{d}_{hor}^{n+1} + \beta_9^s\,\delta_\zeta Z_{hor}^{n+1}\right]\right)\right]
\end{aligned}
$$

$$+\frac{1}{\Delta t}\left[b_w^{(n)} - \left(-\overline{\left(\frac{1}{\rho}\right)}^{\zeta,N}\,\overline{\frac{\partial\zeta}{\partial z}}^{\zeta,N}\,\delta_\zeta^N\left[\frac{\alpha_{div}^v\,\rho}{r\cos\phi}\,\frac{1}{\sqrt{g}}\left(\beta_3^d\,\tilde{d}_{hor}^{n+1} + \beta_6^d\,\delta_\zeta Z_{hor}^{n+1}\right)\right]\right)\right], \quad (7.89)$$

where $\tilde{d}_{hor}^{n+1}$ is calculated via (7.68) and Z_{hor}^{n+1} via (7.70) (both with u^{n+1} and v^{n+1}). Furthermore, the operators

$$\mathbf{A}_{wp} = \overline{\left(\frac{1}{\rho}\right)}^{\zeta,N}\,\overline{\frac{\partial\zeta}{\partial z}}^{\zeta,N}\,\beta_2^s\,\delta_\zeta^N + g\beta_2^b\overline{\frac{1}{p}}^{\zeta,N}\,A_\zeta^N, \quad (7.90)$$

$$\mathbf{A}_{wT} = -g\beta_1^b\,\overline{\frac{p_0}{p}}^{\zeta,N}\,\overline{\frac{1}{T_0}}^{\zeta,N}\,A_\zeta^N, \quad (7.91)$$

$$\mathbf{A}_{pw} = -\frac{c_p}{c_v}p\,\frac{1}{r\cos\phi}\,\frac{1}{\sqrt{g}}\,\beta_4^s\,\delta_\zeta(r\,\cos\phi\,\cdot\ldots) - \beta_3^b g\rho_0 A_\zeta, \quad (7.92)$$

are used. Using again the shallow atmosphere approximation ($r \approx r_{earth}$ in all the prefactors) we can cancel $r\cos\phi$ in $\mathbf{A}_{pw}$,

The explicit right hand sides of the prognostic equations are

$$\begin{aligned} b_{(w)}^n &= \frac{1}{\Delta t}w^n - \overline{\frac{1}{\rho}}^{\zeta,N}\,(1-\beta_2^s)\overline{\frac{\partial\zeta}{\partial z}}^{\zeta,N}\,\delta_\zeta^N p'^n + \\ &\quad + g\left((1-\beta_1^b)\overline{\frac{p_0}{p}}^{\zeta,N}\,\overline{\frac{1}{T_0}}^{\zeta,N}\,A_\zeta^N T' - (1-\beta_2^b)\,\overline{\frac{1}{p}}^{\zeta,N}\,A_\zeta^N p' + \overline{\frac{p_0}{p}}^{\zeta,N}\,\overline{\frac{T}{T_0}}^{\zeta,N}\,\overline{q_x}^{\zeta,N}\right) + \\ &\quad + \overline{\frac{1}{\rho}}^{\zeta,N}\,\overline{\frac{\partial\zeta}{\partial z}}^{\zeta,N}\,\delta_\zeta^N(\alpha_{div}^v\,\rho\,D_{(w)}^n) + \left.\frac{\partial w}{\partial t}\right|_{slow}, \end{aligned} \quad (7.93)$$

$$b_{(p)}^n = \frac{1}{\Delta t}p'^n - \frac{c_p}{c_V}pD_{(p)}^n + (1-\beta_3^b)g\rho_0\,A_\zeta w^n + \left.\frac{\partial p'}{\partial t}\right|_{slow}, \quad (7.94)$$

$$b_{(T)}^n = \frac{1}{\Delta t}T'^n - \frac{R}{c_V}TD_{(T)}^n - (1-\beta_4^b)\frac{\partial T_0}{\partial z}\,A_\zeta w^n + \left.\frac{\partial T'}{\partial t}\right|_{slow}, \quad (7.95)$$

with the explicit divergence terms

$$D_{(p)}^n = \frac{1}{r\cos\phi}\,\frac{1}{\sqrt{g}}\left[(1-\beta_3^s)\,\tilde{d}_{hor}^n + (1-\beta_8^s)\,\delta_\zeta Z_{hor}^n + (1-\beta_4^s)\,\delta_\zeta Z_{vert}^n\right], \quad (7.96)$$

$$D_{(T)}^n = \frac{1}{r\cos\phi}\,\frac{1}{\sqrt{g}}\left[(1-\beta_5^s)\,\tilde{d}_{hor}^n + (1-\beta_9^s)\,\delta_\zeta Z_{hor}^n + (1-\beta_6^s)\,\delta_\zeta Z_{vert}^n\right], \quad (7.97)$$

$$D_{(w)}^n = \frac{1}{r\cos\phi}\,\frac{1}{\sqrt{g}}\left[(1-\beta_3^d)\,\tilde{d}_{hor}^n + (1-\beta_6^d)\,\delta_\zeta Z_{hor}^n + (1-\beta_4^d)\,\delta_\zeta Z_{vert}^n\right]. \quad (7.98)$$

Here $\tilde{d}_{hor}^n$ is calculated by eq. (7.68) and Z_{hor}^n by eq. (7.70)-(7.72) (in both cases with u^n and v^n, Z_{vert}^n analogous to eq. (7.73) with w^n).

The explicit solution for p'^{n+1} reads

$$p'^{n+1} = \Delta t\left(b_{4,p} - \mathbf{A}_{pw}w^{n+1}\right) \quad (7.99)$$

(this can be calculated without any further boundary conditions/treatments).

The explicit equation for T'^{n+1} directly follows from eq. (7.65) using p'^{n+1} from eq. (7.99).

7.2.2 Boundary conditions

The free slip condition

At the upper and lower boundaries the pure Euler equations (i.e. without friction/diffusion) possess only one physical boundary condition: the free slip condition, i.e. the velocity component perpendicular to the boundary must be zero

$$\dot{\zeta} = 0, \tag{7.100}$$

which is equivalent to

$$w = \frac{1}{r\cos\phi}\frac{\partial z}{\partial \lambda}u + \frac{1}{r}\frac{\partial z}{\partial \phi}v. \tag{7.101}$$

At the top boundary we can simply prescribe

$$\dot{\zeta}_{i,j,k=\frac{1}{2}} = 0 \quad \Rightarrow \quad w^{n+1}_{i,j,\frac{1}{2}} = 0. \tag{7.102}$$

At the lower boundary the free slip condition reduces to

$$w^{n+1}_{i,j,ke+\frac{1}{2}} = \frac{1}{r\cos\phi}\left[A_\lambda\left(\frac{\partial z}{\partial \lambda}\,u^{n+1}_{(sfc)}\right) + A_\phi\left(\frac{\partial z}{\partial \phi}\,\cos\phi\, v^{n+1}_{(sfc)}\right)\right], \tag{7.103}$$

where the $\cos\phi$-terms are arranged analogous to the strong conservation form of the divergence. $u^{n+1}_{(sfc)}$ and $v^{n+1}_{(sfc)}$ must be known at $k = ke + \frac{1}{2}$, therefore they must be extrapolated.

Boundary treatments

The following considerations does not concern true physical boundary conditions, but *boundary treatments* in the sense that the appropriate terms cannot longer be calculated by centered differences but need a different numerical treatment.

The vertical pressure gradient in the horizontal equations of motion. In the both horizontal momentum equations there is a need to have a different discretization of $\partial p'/\partial z$ or alternatively of $\partial\left(p' - \alpha^h_{div}\rho D\right)/\partial z$.

At the bottom boundary there exist two options. The first option is to use a one sided formula. Günther Zängl proposed to use a one sided derivative of *2nd order accuracy* (a first order formula is not accurate enough!) instead of the boundary treatment described in[3] (see 4.4).

For three arbitrarily prescribed points

$$z_1, \quad z_2 = z_1 + h_1, \quad z_3 = z_1 + h_2, \tag{7.104}$$

one can derive a one-sided derivative formula of 2nd order for a function $f(z)$ in z_1 by

$$af(z_1) + bf(z_2) + cf(z_3) = (a+b+c)f + (bh_1 + ch_2)f' + \left(b\frac{h_1^2}{2} + c\frac{h_2^2}{2}\right)f'' + \ldots. \tag{7.105}$$

[3] Therefore the switch `ldyn_bbc=.FALSE.` has a different meaning in the two fast waves solver versions.

This results in the weights

$$a = -(b+c), \tag{7.106}$$
$$b = \frac{1-ch_2}{h_1}, \tag{7.107}$$
$$c = \frac{h_1}{h_2(h_1-h_2)}. \tag{7.108}$$

This is active if the switch `ldyn_bbc=.FALSE.` is set. If it is set to `.TRUE.`, then the other possibility is used, the so called 'dynamical bottom boundary condition' Gaßmann (2004), described in the next paragraph.

At the top boundary a one sided derivative formula of first order is sufficient.

Dynamic bottom boundary condition for the pressure. In section 4.4 the strong influence of the formulation of the lower boundary condition for the pressure p' was mentioned. Instead of a pressure extrapolation, Gaßmann (2004) proposed a so called 'dynamic bottom (or lower) boundary condition' for p'. The goal is to determine $\partial p'/\partial\zeta$ in a manner *consistent* to the free-slip condition $\dot{\zeta}=0$. The starting point for the derivation is the bottom boundary condition $\dot{\zeta}=0$ at the surface $z=h_s(\lambda,\phi)$, explicitly written

$$\dot{\zeta} = \frac{u}{r\cos\phi}\left.\frac{\partial\zeta}{\partial\lambda}\right|_z + \frac{v}{r}\left.\frac{\partial\zeta}{\partial\phi}\right|_z + w\frac{\partial\zeta}{\partial z} = 0. \tag{7.109}$$

The vertical pressure gradient occurs in the three momentum equations, therefore we differentiate by time to obtain

$$\frac{\partial\dot{\zeta}}{\partial t} = \frac{1}{r\cos\phi}\left.\frac{\partial\zeta}{\partial\lambda}\right|_z\frac{\partial u}{\partial t} + \frac{1}{r}\left.\frac{\partial\zeta}{\partial\phi}\right|_z\frac{\partial v}{\partial t} + \frac{\partial\zeta}{\partial z}\frac{\partial w}{\partial t} = 0. \tag{7.110}$$

Here we insert the momentum equations (7.1)-(7.3) in the form

$$\begin{aligned}
\frac{\partial u}{\partial t} &= F_u - \frac{1}{\rho}\frac{1}{r\cos\phi}\frac{\partial z}{\partial\lambda}\left(-\frac{\partial\zeta}{\partial z}\frac{\partial\tilde{p}}{\partial\zeta}\right),\\
\frac{\partial v}{\partial t} &= F_v - \frac{1}{\rho}\frac{1}{r}\frac{\partial z}{\partial\phi}\left(-\frac{\partial\zeta}{\partial z}\frac{\partial\tilde{p}}{\partial\zeta}\right),\\
\frac{\partial w}{\partial t} &= F_w - \frac{1}{\rho}\frac{\partial\zeta}{\partial z}\frac{\partial\tilde{p}}{\partial\zeta},
\end{aligned}$$

with $\tilde{p}=p'-\alpha^h_{div}\rho D$ and

$$F_u := -\frac{1}{\rho}\frac{1}{r\cos\phi}\frac{\partial\tilde{p}}{\partial\lambda} + \left.\frac{\partial u}{\partial t}\right|_{slow} \tag{7.111}$$
$$F_v := -\frac{1}{\rho}\frac{1}{r}\frac{\partial\tilde{p}}{\partial\phi} + \left.\frac{\partial v}{\partial t}\right|_{slow} \tag{7.112}$$
$$F_w := +g\left(-1+\frac{p_0}{p}\frac{T}{T_0}(1+q_x)\right) + \frac{1}{\rho}\frac{\partial\zeta}{\partial z}\frac{\partial(\alpha^v_{div}-\alpha^h_{div})\rho D}{\partial\zeta} + \left.\frac{\partial w}{\partial t}\right|_{slow}. \tag{7.113}$$

In F_w the form (7.21) of the buoyancy term was used. Sorting by $\frac{\partial\zeta}{\partial z}\frac{\partial p'}{\partial\zeta}$ finally results in

$$\frac{\partial\zeta}{\partial z}\frac{\partial p'}{\partial\zeta}\frac{1}{\rho}\left[\left(\frac{1}{r\cos\phi}\frac{\partial z}{\partial\lambda}\right)^2 + \left(\frac{1}{r}\frac{\partial z}{\partial\phi}\right)^2 + 1\right] = -\frac{1}{r\cos\phi}\frac{\partial z}{\partial\lambda}F_u - \frac{1}{r}\frac{\partial z}{\partial\phi}F_v + F_w. \tag{7.114}$$

This delivers the needed value of $\frac{\partial \zeta}{\partial z}\frac{\partial p'}{\partial \zeta}$ at the lower boundary (for $k = ke$) in the u- and v-equation. The discretization of this expression and the averaging to the u and v position is straightforward. Nevertheless, several spatial interpolations with 4-point or even 8-point formulas are needed.

Boundary treatment of the divergence. In the 'strong conservation form' the horizontal derivatives may be calculated in every grid point by centered differences, even at the boundary points (see eq. (7.10)).

For the remaining vertical derivative $\frac{\partial Z}{\partial \zeta}$ one needs Z at the boundary, i.e. at $k = ke + 1/2$ and $k = 1/2$. For this two variants are possible. First it can be calculated by extrapolation; this is currently implemented.
Second, through the use of the *exact* boundary condition $Z = 0$. This is a theoretical advantage of the 'strong conservation form'. An important requirement for this is that the implicit weightings are chosen as

$$\beta_8^s = \beta_4^s, \quad \text{and} \quad \beta_9^s = \beta_6^s. \tag{7.115}$$

But nevertheless, there remains the problem that Z must be subdivided into horizontal and vertical parts, due to the vertically implicit solver. Therefore one cannot pose this exact boundary condition for the divergence itself. This probably is the reason, why this second boundary treatment does not work until now.

7.2.3 Stability of the divergence damping in tilted terrain

Though the divergence damping is in general necessary to stabilize the whole split-explicit time integration scheme, it can itself become unstable. Beyond the stability limitation $\alpha_{div}\Delta t/\Delta x^2 \leq 1/2$ (Baldauf (2010)), there is an additional constraint in tilted terrain (Baldauf (2013)):

$$\alpha_{div}^h \, \Delta t \left\{ \frac{1}{\Delta x^2}\left(2 + \frac{\Delta x}{\Delta z}\left|\frac{\partial z}{\partial x}\right|\right)^2 + \frac{1}{\Delta y^2}\left(2 + \frac{\Delta y}{\Delta z}\left|\frac{\partial z}{\partial y}\right|\right)^2 \right\} \leq 2. \tag{7.116}$$

This is a quite general stability condition not only for the divergence damping but for explicit discretizations of the diffusion equation in general. Because $\Delta h := \frac{\partial z}{\partial x}\,\Delta x$ is just the height jump along a ζ-coordinate line, the expression

$$\frac{\Delta h}{\Delta z} \equiv \frac{\Delta x}{\Delta z}\frac{\partial z}{\partial x} \tag{7.117}$$

describes the ratio between the height change Δh of a coordinate plane over one grid mesh size Δx and the vertical thickness Δz of the grid box.

Near the ground, where small $\Delta z \sim 20$ m are common, the ratio $\Delta h/\Delta z$ can be quite large even for rather gentle slopes. Therefore, in steep terrain the dimensionless value $xkd := \alpha_{div}/(c_s^2\Delta t)$ must be chosen much smaller than the recommended value of 0.1 by Wicker and Skamarock (2002) or even as the recommendation of 0.3 by Baldauf (2010). It is interesting that $\Delta h/\Delta z$ often achieves higher values for coarser resolutions, where the jumps from one grid box to another are larger than for finer resolutions. Hence, the steeper slopes occurring in fine scale model applications are not the limiting factor in this case.

The good news is that in practice, the limitation (7.116) must not be strictly fulfilled. One reason is, that the forward-backward scheme of the fast waves solver rather adds the tendencies of sound, buoyancy and divergence damping instead of doing an operator splitting. Therefore, in the COSMO code a namelist variable `divdamp_slope` was introduced to weaken up the condition (7.116). The true value of α^h_{div} is determined by

$$\alpha^h_{div} = \min\left(\texttt{xkd} \cdot c_s^2\,\Delta t,\ \texttt{divdamp\_slope} \cdot \alpha_{div,slope}\right),$$

where `xkd` is the namelist variable for divergence damping, and $\alpha_{div,slope}$ is the value determined by (7.116). Consequently, the divergence damping coefficient never exceeds the value given by `xkd`.

7.3 Tracer transport

For the advection of scalar tracer fields, i.e. all water constituents, possible aerosol particles and gaseous substances used in the framework of COSMO-ART, and the turbulent kinetic energy mainly two different schemes are available. One can either choose a fully 3-dimensional semi-Lagrangian scheme or a direction splitted finite volume scheme. For the latter several flux calculation formulations exist under which the flux calculation by Bott (1989a) is the standard choice.

The advantage of the finite volume scheme is the inherent mass conservation (at least for Courant numbers less than 1). The disadvantage is the direction splitting necessary to combine the one-dimensional schemes, which can lead to instabilities in strongly deformational flows. The semi-Lagrangian scheme does not suffer from those kinds of instabilities. But it is not mass conserving, which again in strongly deformational flows can negatively affect the quality of the simulation. Therefore currently the Bott scheme is used in most of the COSMO applications.

7.3.1 The semi-Lagrangian scheme

The semi-Lagrangian (SL) scheme mainly bases on the work of Staniforth and Côté (1991). An advection equation in the form

$$\frac{\partial q_l}{\partial t} + \vec{v}\nabla q_l = S_l \tag{7.118}$$

can be equivalently formulated as

$$\frac{dq_l}{dt} = S_l. \tag{7.119}$$

Here the lower index l denotes the kind of the tracer (e.g. water vapor $l = v$, cloud water $l = c$, ...) and S_l denotes all source terms or possible other transport terms like diffusion or sedimentation. Semi-Lagrangian advection splits the advection problem into two parts: firstly the determination from which space point a particle starts, i.e. the calculation of the backward trajectory. Secondly the determination of the tracer value in this point, usually by interpolation.

The backward trajectory is calculated with 2nd order accuracy... Baldauf and Schulz (2004).

The interpolation takes place in the transformed space which has a Cartesian grid structure. To construct the tri-cubic interpolation a polynomial $p(x, y, z)$ is searched with the property $p(i, j, k) = q_{i,j,k}$, where $i,\ j,\ k = -2,\ -1,\ 0,\ 1$, and $q_{i,j,k}$ is the value at the appropriate grid point. This problem can be reduced to the estimation of polynomials $P_k(x)$ of only one variable x with the property

$$P_i(j) = \begin{cases} 1 & : \quad i = j \\ 0 & : \quad i \neq j \end{cases}, \qquad i, j = -2, -1, 0, 1, \tag{7.120}$$

which is fulfilled by

$$P_{-2}(x) = \frac{1}{6}\,(x+1)\,x\,(x-1), \tag{7.121}$$

$$P_{-1}(x) = -\frac{1}{2}\,(x+2)\,x\,(x-1), \tag{7.122}$$

$$P_0(x) = \frac{1}{2}\,(x+2)\,(x+1)\,(x-1), \tag{7.123}$$

$$P_1(x) = -\frac{1}{6}\,(x+2)\,(x+1)\,x. \tag{7.124}$$

The polynomial p can then be constructed by

$$p(x, y, z) = \sum_{i,j,k=-2}^{1} P_i(x)\ P_j(y)\ P_k(z)\ q_{i,j,k}, \tag{7.125}$$

where x, y, z are the interpolation weights. The sum (7.120) runs over the neighboring 64 grid points of the backward trajectory starting point. These are much more points than are needed to construct a polynomial of only 3. order in 3 dimensions; for this task only 20 grid points would be necessary. But the high symmetry of equation (7.120) allows a quick way for calculating the sum. Therefore this SL variant is quite efficient. The tri-cubic interpolation generates quite good transport properties. Another advantage of this SL method is, that it calculates the interpolation in three dimensions in one step, therefore no splitting error occurs. However the lack of the conservation property can generate artefacts, if the flow deformation is very strong.

A further disadvantage is, that the method can produce negative undershoots for a positive definite field. A simple clipping of these negative values also can strongly violate the mass conservation and should be avoided. But for positive definite variables, a clipping of negative values is necessary. At least global conservation can be achieved by a simple multiplicative filling technique Rood (1987).

7.3.2 Bott-advection and related schemes

The advection equation for a scalar (e.g. a moisture specific mass) q_l

$$\frac{\partial q_l}{\partial t} + \vec{v}\nabla q_l = S_l \tag{7.126}$$

can be reformulated with the aid of the continuity equation to

$$\frac{\partial \rho_l}{\partial t} + \nabla(\rho_l \vec{v}) = \rho S_l. \tag{7.127}$$

with $\rho_l := \rho q_l$. Again the lower index l denotes the kind of the tracer (e.g. water vapor $l = v$, cloud water $l = c$, ...) and S_l denotes all source terms or possible other transport terms like diffusion or sedimentation.

For the occurring divergence one can use the 'strong conservation form' (eq. (2.120))

$$\begin{aligned}\nabla(\rho\vec{v}) &= \frac{1}{r^2\cos\phi}\frac{1}{\sqrt{g}}\left[\frac{\partial}{\partial\lambda}\left(r\sqrt{g}\rho u\right)+\frac{\partial}{\partial\phi}\left(r\cos\phi\sqrt{g}\rho v\right)+\frac{\partial}{\partial\zeta}\left(r^2\cos\phi\sqrt{g}\rho\dot{\zeta}\right)\right]\\ &= \frac{1}{\sqrt{g}}\left[\frac{1}{r\cos\phi}\frac{\partial}{\partial\lambda}\left(\sqrt{g}\rho u\right)+\frac{1}{r\cos\phi}\frac{\partial}{\partial\phi}\left(\cos\phi\sqrt{g}\rho v\right)+\frac{\partial}{\partial\zeta}\left(\sqrt{g}\rho\dot{\zeta}\right)\right] \qquad (7.128)\end{aligned}$$

with $\sqrt{g} = -\partial z/\partial\zeta$. Again the 'shallow atmosphere' approximation has been used, i.e. r is assumed constant.

If one-dimensional appropriate discretizations of the flux derivatives are available, then a numerical integration scheme can be constructed via an operator splitting. First calculate the densities

$$\rho_l^n = \rho^n \cdot q_l^n \qquad (7.129)$$

(here upper indices denote time levels). Integrate them and also the density itself (i.e. with $q_0 \equiv 1$) by operator splitting

$$\begin{aligned}\rho_l' &= \rho_l^n + \Delta t\ \hat{A}_x(\Delta t)\rho_l^n, &(7.130)\\ \rho_l'' &= \rho_l' + \Delta t\ \hat{A}_y(\Delta t)\rho_l', &(7.131)\\ \rho_l''' &= \rho_l'' + \Delta t\ \hat{A}_z(\Delta t)\rho_l'', &(7.132)\end{aligned}$$

where the numerical operators $\hat{A}_x$, $\hat{A}_y$, and $\hat{A}_z$ are discretizations of the 3 terms occurring in eq. (7.128). Afterwards calculate the specific mass

$$q_l''' = \frac{\rho_l'''}{\rho'''}. \qquad (7.133)$$

Finally the other source terms can again be taken into account by operator splitting e.g. by

$$q_l^{n+1} = q_l''' + \Delta t\ S_l''' \qquad (7.134)$$

(the microphysics contains also some implicit schemes, therefore this is only an example).

In the standard 'Bott2' scheme (and the related 'Bott4', 'PPM' and 'vanLeer') this sequence $\hat{A}_x$ - $\hat{A}_y$ - $\hat{A}_z$ is used in every odd time step, whereas in every even time step the reverse sequence $\hat{A}_z$ - $\hat{A}_y$ - $\hat{A}_x$ is used. This is a sort of a simplified Strang-splitting. Sometimes this can lead to $2\Delta t$ oscillations.

The drawback of this direction splitting is that it can lead to instabilities by a complete emptying of a grid box in one direction step, which cannot cured again by the following steps. This problem can be reduced to a certain degree if one uses the true Strang-splitting version $\hat{A}_z(\Delta t/2)$ - $\hat{A}_y(\Delta t/2)$ - $\hat{A}_x(\Delta t)$ - $\hat{A}_y(\Delta t/2)$ - $\hat{A}_z(\Delta t/2)$. (denoted as 'Bott2_Strang' (and appropriately 'Bott4_Strang', ... for the other schemes) This form is about 60% more time consuming than 'Bott2'. However it can increase the order of the time integration up to two, and, perhaps more important, it halves the time step for the z and y direction, which helps to reduce instabilities due to the direction splitting.

The use of the above flux form guarantees mass conservation of the appropriate tracer variable (at least for Courant numbers less than 1). The additional transport of the density itself delivers 'mass-consistency' in the sense that a constant field $q_l = const.$ remains constant if the wind divergence vanishes (however, this slightly perturbs the exact mass-conservation).

The Bott flux-derivative calculation

To discretize a one-dimensional advection equation in flux form

$$\frac{\partial \rho_l}{\partial t} + \frac{\partial c\rho_l}{\partial x} = 0 \tag{7.135}$$

Bott (1989a) used an idea going back to Tremback et al. (1987). In a finite volume discretization fluxes $\overline{F}^n_{j+1/2}$ and $\overline{F}^n_{j-1/2}$ occur at the boundaries of the grid box j. They can be assumed as averages over a time interval Δt

$$\overline{F}^n_{j+1/2} = \frac{1}{\Delta t}\int_{t^n}^{t^n+\Delta t} F_{j+1/2}\, dt. \tag{7.136}$$

For a purely advective flux $F = c\rho_l$ (where $c = \sqrt{g}u$ in the zonal direction, $c = \cos\phi\ \sqrt{g}v$ in the meridional direction, and $c = \sqrt{g}\dot{\zeta}$ in the vertical direction) and if one assumes for the moment that c would be constant, one can transform this time integral with the aid of $x' = x - ct$ in a space integral. Due to $\rho_l(x,t) = \rho_l(x-ct)$ it follows:

$$\int_{t^n}^{t^n+\Delta t} c\rho_l(x,t')dt' = -\int_{x_{j+1/2}}^{x_{j+1/2}-c\Delta t} \rho_l(x',t)dx' = -\int_{x_{j+1/2}-c\Delta t}^{x_{j+1/2}} \rho_l(x',t)dx'. \tag{7.137}$$

This spatial integral can be computed by approximating the integrand by an interpolation polynomial through the neighboring grid points.

Additionally the discretization has to be positive definite. This can be achieved if the fluxes are limited in a manner that

$$\frac{\Delta t}{\Delta x}(\overline{F}^n_{j+1/2} - \overline{F}^n_{j-1/2}) \le \phi^n_j \tag{7.138}$$

is fulfilled. This guarantees $\phi^{n+1}_j \ge 0$, i.e. a grid box mostly must be emptied.

Details of the interpolation formulas and the flux limitations are given in Bott (1989a) and Bott (1989b).

This scheme is only stable for Courant numbers less than 1. Skamarock (2006) proposed a general methodology to extend the stability range of one-dimensional finite volume schemes to arbitrary Courant numbers by shifting the contributions of whole grid cells (the 'integer' fluxes) between the starting point and the target grid cell. The remaining 'fractional' flux then is treated by the Bott scheme. However, this method destroys the conservation of the multi-dimensional advection scheme if one of the Courant numbers is greater than one.

The most recommended version of this whole scheme may be called by
`y_scalar_advect='BOTT2_Strang'`.

The deformation correction extension

One disadvantage of the original Bott (1989a) scheme (and of many other finite-volume formulations) is the fact that a constant density field does not remain constant in divergence-free but deformational flow fields, if the 'mass-consistency'-fixer described above is not applied.

To circumvent this problem Bott (2010) proposes to add a 'deformational correction' term in every direction step in equations (7.130)-(7.132) leading to

$$\rho_l' = \rho_l^n - \Delta t \frac{f_x^+(\rho_l^n) - f_x^-(\rho_l^n)}{\Delta x} + \Delta t\, \rho_l^n \frac{\partial u}{\partial x}, \tag{7.139}$$

$$\rho_l'' = \rho_l' - \Delta t \frac{f_y^+(\rho_l') - f_y^-(\rho_l')}{\Delta y} + \Delta t\, \rho_l^n \frac{\partial v}{\partial y}, \tag{7.140}$$

$$\rho_l^{n+1} = \rho_l'' - \Delta t \frac{f_z^+(\rho_l'') - f_z^-(\rho_l'')}{\Delta z} + \Delta t\, \rho_l^n \frac{\partial w}{\partial z} - \Delta t\, \rho_l^n \nabla \cdot \mathbf{v}. \tag{7.141}$$

f_x, f_y, and f_z are the numerical fluxes given by Bott (1989a) on the 'left ('-') and 'right' ('+') side of every 1D grid cell. The resulting scheme is still mass-conserving and Bott (2010) describes a procedure to keep not only the fluxes but also the additional deformational correction terms positive definite. Beyond this, one can easily show that an initial constant density field remains constant, if the added and subtracted divergence terms are compatible with the flux formulation.

To keep this scheme efficient also for Courant numbers larger than one, a local time-stepping scheme has been developed Baldauf (2019). Here, in every grid cell, for which the 1D Courant-number in direction j lies in the range $1 < C_j < 2$ (larger $C_j \geq 2$ are not allowed in the Runge-Kutta dynamical core), two time steps with the half Δt are performed. By a proper bookkeeping of the in- and outgoing fluxes and of the deformational correction terms, this local time-stepping is designed in a way that all the properties of the Bott (2010) scheme are maintained: it is still mass conserving, constant tracer density remains constant and it is positive definite.

Despite the 'keep the constant density' property of this scheme, the Bott (2010) proposal is not entirely mass-consistent in the stronger sense, i.e. that the scheme is consistent with the continuity equation (for total mass). Experience shows, that this stronger form of mass-consistency is necessary near the bottom boundary to produce realistic moisture fields. Therefore, a certain form of mass-consistency is achieved by again a parallel solution of the continuity equation with the same advection scheme. At the end of the advection process, the specific mass q_l is calculated with this artificially transported total mass density. As in the scheme `'BOTT2_Strang'` described above, this slightly violates exact mass-conservation.

This scheme may be called by `y_scalar_advect='BOTTDC2'`.

7.4 Damping mechanisms

7.4.1 Relaxation at the lateral boundaries

A general relaxation scheme for a field $\phi(\mathbf{r}, t)$ in the vicinity of a boundary sounds

$$\frac{\partial \phi}{\partial t} = -\mu_b(\mathbf{r}) \cdot (\phi - \phi_{ext}(\mathbf{r}, t)), \tag{7.142}$$

where the (space dependent) relaxation coefficient is expressed by a relaxation time τ and a spatial attenuation function f_r:

$$\mu_b(\mathbf{r}) = \frac{1}{\tau} \cdot f_r(\frac{d}{L}). \tag{7.143}$$

d is the distance from the boundary, L normalizes this distance.

It is convenient that the function f_r should have the properties $f_r(0) = 1$ and a monotonically decrease for increasing d. Ideally it falls to 0 in $d = L$: $f_r(1) = 0$; examples are

$$f_r(x) = |1-x|^\alpha, \quad \alpha > 1,\ x \le 1 \tag{7.144}$$
$$f_r(x) = \cos^2 \frac{\pi}{2} x, \quad x \le 1 \tag{7.145}$$

whereas it is set $f_r(x) = 0$ for $x > 1$. These functions also have the property $df_r/dx = 0$ at $x = 1$.
Other choices are

$$f_r(x) = e^{-\alpha\, x}, \quad \alpha > 0,\ x \le 1, \tag{7.146}$$
$$f_r(x) = 1 - \tanh(\alpha\, x), \quad \alpha > 0,\ x \le 1. \tag{7.147}$$

(Remark: these functions do not fulfill the property $f_r(1) = 0$).

Different time discretisations are possible:

- in an explicit manner for a 2-timelevel scheme

$$\phi^{n+1} = \phi^n - \Delta t\, \mu_b (\phi^n - \phi^n_{ext}) \tag{7.148}$$

- or in an implicit manner for a 2-timelevel scheme

$$\phi^{n+1} = \phi^n - \Delta t\, \mu_b (\phi^{n+1} - \phi^n_{ext}) \tag{7.149}$$

  but which can be solved explicitly using

$$\phi^{n+1} = \phi^n - \alpha_b (\phi^n - \phi^n_{ext}) \tag{7.150}$$

  with

$$\alpha_b = \frac{\Delta t\ \mu_b}{1 + \Delta t\ \mu_b} \tag{7.151}$$

- in an implicit manner for a 3-timelevel scheme

$$\phi^{n+1} = \phi^{n-1} - 2\Delta t\, \mu_b (\phi^{n+1} - \phi^n_{ext}) \tag{7.152}$$

  again this can be solved explicitly using

$$\phi^{n+1} = \phi^n - \alpha_b (\phi^n - \phi^n_{ext}) \tag{7.153}$$

  with

$$\alpha_b = \frac{2\Delta t\ \mu_b}{1 + 2\Delta t\ \mu_b} \tag{7.154}$$

  According to Kallberg (1977), eq. (7.147) is used directly for α_b.

The explicit scheme (for 2 timelevels) is stable for $0 \le \Delta t\ \mu_b \le 2$. Additionally, if one wants to avoid oscillating behaviour, one has to sharpen this to $0 \le \Delta t\ \mu_b \le 1$, or alternatively $\tau \ge \Delta t$.

In contrast to this the implicit variants are unconditionally stable. However, this is only achieved by a simple reduction of the dimensionless relaxation coefficient to values less than 1. Therefore, in the future only the explicit variant is used.

In the explicit case the function (7.146) has shown to give the best results (rather comparable to (7.144) or (7.147), whereas (7.145) seems to be a bad choice for lateral relaxation). $\alpha = 6$ gives a satisfying attenuation towards inside of the domain. The relaxation time τ can be prescribed as a multiple of the time step `crltau * dt`. Currently `crltau` $= 1$ is chosen; we remark that in this case $\phi^{n+1} = \phi^n_{ext}$ holds at the boundary.

The external field $\phi_{ext}(\mathbf{r}, t)$, towards which the relaxation takes place, is given as the spatial interpolation of the appropriate field in the driving model. Additionally it is the temporal mean of the fields at time levels n and $n + 1$.

7.4.2 Horizontal Smagorinsky Diffusion

In very rare events the artificial 4th order hyperdiffusion is not strong enough to prevent the model from a crash by horizontal shear instabilities. This shows that an additional, more physically based diffusion mechanism in the horizontal is needed. The nonlinear diffusion proposed by Smagorinsky (1963) is formally a purely horizontally acting ('harmonic') diffusion. In cartesian coordinates it reads

$$\frac{\partial u}{\partial t} + \vec{v} \cdot \nabla u = \ldots + K_{smag} \Delta u, \tag{7.155}$$

$$\frac{\partial v}{\partial t} + \vec{v} \cdot \nabla v = \ldots + K_{smag} \Delta v, \tag{7.156}$$

$$\frac{\partial w}{\partial t} + \vec{v} \cdot \nabla w = \ldots + 0, \tag{7.157}$$

with the diffusion coefficient:

$$K_{smag} = l_s^2 \cdot \sqrt{T^2 + S^2}, \tag{7.158}$$

$$T = \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}, \tag{7.159}$$

$$S = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}. \tag{7.160}$$

This means, that it contains both parts of the horizontal tension strain T and of the horizontal shearing strain S (on the sphere additional metric correction terms must be considered in T and S Smagorinsky (1993). However, those can be neglected for smaller scale model applications and with the main intention to prevent from model crashes by shear instabilities).

The length scale l_s (a sort of a mixing length) can be determined by the following argument: in any case a stability criterion

$$K_{smag} \cdot \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \leq \frac{1}{2} \tag{7.161}$$

must be fulfilled. One can simply set (motivated by 'numerical efficiency')

$$l_s^2 = \frac{c}{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}} \tag{7.162}$$

with a yet arbitrary 'Smagorinsky-constant' c. Then from stability constraint it follows, that the *dimensionless* diffusion coefficient

$$k_{smag} := c \cdot \Delta t \cdot \sqrt{T^2 + S^2} \tag{7.163}$$

must fulfill

$$k_{smag} \leq \frac{1}{2}. \tag{7.164}$$

For example for COSMO-DE with $\Delta x \approx \Delta y \approx 2800$ m and $\Delta t \approx 25$ s and for a shear of $\Delta u = 28$ m/s per grid box this results in $k_{smag} \approx c \cdot \Delta t \cdot \Delta u / \Delta y \approx c \cdot 0.25$.
To get the dimensional value K_{smag} one has to multiply by $\frac{1}{\Delta t} \cdot \frac{1}{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}} \approx 1.6 \cdot 10^5\ m^2/s$.
Smagorinsky (1963) proposes a value of about $c \approx 0.1$. If one uses this diffusion only as a mechanism to reduce shear instabilities without influencing to much the overall model behaviour, a value of $c \approx 0.03$ was found as appropriate for COSMO-DE simulations.

To discretize S and T by centered differences in a symmetric way, T is discretized to the scalar position (i, j) whereas S is discretized to the position $(i+1/2, j+1/2)$ ('uv'-position). Afterwards they are firstly squared (to prevent from annihilation by negative values) and secondly they are averaged to the u- and v-positions.

To avoid a 'double counting', i.e. that small wavelengths are diffused too strong by both Smagorinsky- and the 4th order (hyper)-diffusion, the hyperdiffusion coefficient is substracted with a weight of 0.5.

There is only one namelist switch (`l_diff_Smag`) to enable or disable this horizontal Smagorinsky diffusion. The two above mentioned parameters are only internal parameters (with the intention that only experienced users should modify them): the Smagorinsky constant `c_smag = 0.03` and the weighting to avoid double counting `weight_K_4th = 0.5`.

7.4.3 Targeted diffusion to avoid cold pools in narrow valleys

In rare cases, cold pools can develop in very narrow, steep valleys. One possible reason for that is the 5th order advection operator, which has the tendency to overshoot peaks in the field of a transported quantity. While this overshooting is often welcome and leads to an overall accurate scheme, it can be dramatically amplified in the temperature field, when a ramp like structure is recreated by steep orography.

To avoid such artificial cold pools (which can easily achieve negative temperatures of about -100^{circ}C) a targeted diffusion is applied. A criterion of the form

$$T'_{ijk} < \overline{T'}_{ijk} - 10K \tag{7.165}$$

where $\overline{T'}_{ijk} := \frac{1}{4}(T'_{i+1,j,k} + T'_{i-1,j,k} + T'_{i,j+1,k} + T'_{i,j-1,k})$ is the average of the 4 neighbouring grid points, is tested in the lowest 5 k-levels. If it is fulfilled, then a simple second order diffusion is applied in just this grid point. The diffusion coefficient is 1/4 of the maximum possible value (i.e. determined by stability).

By experience, the criterion is only seldomly fulfilled and only in very few grid points. Therefore, the only computational effort is the test of the above criterion (this fills a list of points, consequently the diffusion operation is applied only in these few points, too, and not on whole fields). The computational amount is only about 0.05% of the whole model run.

References

Asselin, R. (1972). Frequency filter for time integrations. *Mon. Wea. Rev. 100*, 487–490.

Baldauf, M. (2008). Stability analysis for linear discretisations of the advection equation with Runge-Kutta time integration. *J. Comp. Phys. 227*, 6638–6659.

Baldauf, M. (2010). Linear stability analysis of Runge-Kutta based partial time-splitting schemes for the Euler equations. *Mon. Wea. Rev. 138*, 4475–4496.

Baldauf, M. (2013). A new fast-waves solver for the Runge-Kutta dynamical core. COSMO Technical Report No. 21, Deutscher Wetterdienst, http://www.cosmo-model.org/content/model/documentation/techReports/default.htm.

Baldauf, M. (2019). Local time stepping for a mass-consistent and time-split advection scheme. *Quart. J. Roy. Met. Soc. 145*, 337–346.

Baldauf, M. and J.-P. Schulz (2004). Prognostic Precipitation in the Lokal Modell (LM) of DWD. COSMO Newsletter No. 4, Deutscher Wetterdienst. 177-180.

Ballard, S. P. and B. W. Golding (1991). Basic model formulation. Short range forecasting research. Technical Report Mesoscale Documentation Paper No. 4, UKMO, Met Office, Bracknell, UK.

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

Bott, A. (1989b). Reply (to Smolarkiewicz (1989) MWR). *Mon. Wea. Rev. 117*(11), 2263–2636.

Bott, A. (1993). The monotone area-preserving flux-form advection algorithm: Reducing the time-split error in the two-dimensional flow fields. *Mon. Wea. Rev. 121*, 2638–2641.

Bott, A. (2010). Improving the time-splitting errors of one-dimensional advection schemes in multidimensional applications. *Atm. Res. 97*, 619–631.

C., E. S., H. C. Elman, and M. H. Schultz (1983). Variational iterative methods for non-symmetric systems of linear equations. *SIAM J. Numer. Anal. 2*, 345–357.

Clark, T. L. (1977). A small-scale dynamic model using terrain following coordinate transformation. *J. Comp. Phys. 24*, 186–215.

Colella, P. (1990). Multidimensional upwind methods for hyperbolic conservation laws. *J. Comput. Phys. 87*, 171–200.

Crowley, W. P. (1968). Numerical Advection Experiments. *Mon. Wea. Rev. 96*, 1–11.

Davies, H. (1976). A lateral boundary formulation for multi-level prediction models. *Quart. J. R. Met. Soc. 102*, 405–418.

Davies, H. (1983). Limitations of some common lateral boundary schemes used in regional NWP models. *Mon. Wea. Rev. 111*, 1002–1012.

Davies, L. A. and A. R. Brown. (2001). Assessment of which scales of orography can be credibly resolved in a numerical model. *Quart. J. Roy. Meteor. Soc. 127*, 1225–1237.

Déqué, M. and D. Cariolle (1986). Some destabilizing properties of the Asselin time filter. *Mon. Wea. Rev. 114*, 880–884.

Doms, G. (2001). A scheme for monotonic numerical diffusion in the LM. COSMO Technical Report No. 3, Deutscher Wetterdienst.

Doms, G., F. J., H. E., H. H.-J., R. M., R. T., and V. G. (2004, June). *A Description of the Nonhydrostatic Regional Model LM, Part II: Physical Parameterization*. Offenbach, Germany: Deutscher Wetterdienst.

Dudhia, J. (1993). A nonhydrostatic version of the Penn State / NCAR mesoscale model: Validation tests and simulation of an Atlantic cyclone and cold front. *Mon. Wea. Rev. 121*, 1493–1513.

Dudhia, J. (1995). Reply to Comments on "A nonhydrostatic version of the Penn State / NCAR mesoscale model: Validation tests and simulation of an Atlantic cyclone and cold front". *Mon. Wea. Rev. 123*, 2573–2575.

Durran, D. and J. Klemp (1983). The effects of moisture on trapped mountain lee waves. *J. Atmos. Sci. 39*, 2490–2506.

Dutton, J. (1976). *The Ceaseless Wind*. McGraw-Hill Book Company.

Dutton, J. and G. Fichtl (1969). Approximate equations of motion for gases and liquids. *J. Atmos. Sci. 26*, 241–254.

F., G., M. M. R., and B. G. T. (1993). Development of second-generation regional climate model (RegCM2). Part II: Convective processes and assimilation of lateral boundary conditions. *Mon. Wea. Rev. 121*, 2814–2832.

Gal-Chen, T. and C. Sommerville (1975). On the use of a coordinate transform for the sulution of the Navier-Stokes equations. *J. Comput. Phys. 17*, 209–228.

Gaßmann, A. (2001). Filtering of LM-Orography. COSMO Newsletter No. 1, Deutscher Wetterdienst. 71-78.

Gaßmann, A. (2002). 3D-transport of precipitation. COSMO Newsletter No. 2, Deutscher Wetterdienst. 113-117.

Gaßmann, A. (2004). Formulation of the LM's Dynamical Lower Boundary Condition. COSMO Newsletter No. 4, Deutscher Wetterdienst. 155-158.

Gaßmann, A. and H.-J. Herzog (2007). A Consistent Time-Split Numerical Scheme Applied to the Nonhydrostatic Compressible Equations. *Mon. Wea. Rev. 135*, 20–36.

Grell, G., J. Dudhia, and D. Stauffer (1994). *A Description of the Fifth-Generation Penn State/NCAR Mesoscale Model (MM5)*. Boulder, Colorado, USA: National Centre for Atmosperica Research, Mesoscale and Microscale Meteorological Division.

Herbert, F. (1975). Irreversible Prozesse in der Atmosphäre; Teil 3. *Contr. Atm. Phys. 48*, 1–29.

Hess, R. (2001). Assimilation of screen-level observations by variational soil moisture analysis. *Meteorol. Atmos. Phys. 77*, 155–166.

Huang, X.-Y. and P. Lynch (1993). Diabatic digital filtering initialization: application to the HIRLAM model. *Mon. Wea. Rev. 121*, 589–603.

Ikawa, M. (1988). Comparison of Some Schemes for Nonhydrostatic Models with Orography. *J. Met. Soc. Japan 66*, 735–776.

Jacobsen, I. and E. Heise (1982). A new economic method for the computation of the surface temperature in numerical models. *Contr. Atmos. Phys. 55*, 128–141.

Kallberg, P. (1977). A lateral boundary formulation for multi-level prediction models. Internal Report No. 3, ECMWF Research Department, Shinfield Park, Reading, Berkshire RG2 9AX, England.

Klemp, J. and D. Durran (1983). An upper boundary condition permitting internal gravity wave radiation in numerical mesoscale models. *Mon. Wea. Rev. 111*, 430–444.

Klemp, J., W. Skamarock, and J. Dudhia (2003). Conservative Split-Explicit Time Integration Methods for the Compressible Nonhydrostatic Equations. *Mon. Wea. Rev. 135*, 2897–2913.

Klemp, J. and R. Wilhelmson (1978). The Simulation of Three-dimensional Convective Storm Dynamics. *J. Atmos. Sci. 35*, 1070–1096.

Leuenberger, D. (2002). The SLEVE coordinate in LM. (No. 2). 105-110.

LeVeque, R. J. (1993). High-resolution conservative algorithms for advection in incompressible flow. Technical Report 93-03, University of Washington, Seattle, WA. 40pp.

Lin, S. J. and R. Rood (1996). Multidimensional flux-form semi-lagrangian transport schemes. *Mon. Wea. Rev. 124*, 2046–2070.

Louis, J.-F. (1979). A parametric model of vertical eddy fluxes in the atmosphere. *Bound. Layer Meteor. 17*, 187–202.

Lynch, P. (1990). Initialization using a digital filter. Research activities in atmospheric and ocean modeling, no. 14, Working Group on Numerical Experimentation, WMO, Switzerland. CAS/JSC.

Lynch, P. (1997). The Dolph-Chebyshev window: a simple optimal filter. *Mon. Wea. Rev. 125*, 655–660.

Lynch, P., D. Girard, and V. Ivanovici (1997). Improving the efficiency of a digital filtering scheme. *Mon. Wea. Rev. 125*, 1976–1982.

Lynch, P. and X.-Y. Huang (1992). Initialization of the HIRLAM model using a digital filter. *Mon. Wea. Rev. 120*, 1019–1034.

Lynch, P. and X.-Y. Huang (1994). Diabatic initialization using recursive filters. *Tellus 46A*, 583–597.

Mahrer, Y. (1984). An Improved Numerical Approximation of the Horizontal Gradients in a Terrain-Following Coordinate System. *Mon. Wea. Rev. 112*, 918–922.

Marchuk, G. (1975). *Numerical Methods in Weather Prediction*. Academic Press. Comp., Dortrecht.

Mironov, D. (2008). Parameterization of lakes in numerical weather prediction. Description of a lake model. COSMO Technical Report 11, Consortium for Small-Scale Modelling, 41 pp. doi:10.5676/DWD_pub/nwv/cosmo-tr_11 (available from http://www.cosmo-model.org).

Mironov, D. and B. Ritter (2004). Testing the new ice model for the global NWP system GME of the German Weather Service. Technical documentation, 1220, WMO, WMO, Geneve, Switzerland.

Müller, E. (1974). *Numerische Simulation hochreichender Feuchtkonvektion auf der Basis der ursprünglichen Eulerschen Gleichungen*. Deutscher Wetterdienst, D-63004 Offenbach, Germany: Berichte des Deutschen Wetterdienstes Nr. 133.

Orlanski, I. (1976). A simple boundary condition for unbounded hyperpolic flows. *J. Comput. Phy. 21*, 251–269.

Petrik, R. (2006). Relevanz mesoskaliger Advektions- und Coriolisterme für das Meso-γ-Modell LMK.

Pielke, R. (1984). *Mesoscale Meteorological Modeling*. Academic Press, Inc.

Pielke, R. (2002). *Mesoscale Meteorological Modeling, Second Edition*. Academic Press, Inc.

Raymond, W. H. (1988). High-Order Low-Pass Implicit Tangent Filters for Use in Finite Area Calculations. *Mon. Wea. Rev. 116*, 2132–2141.

Ritter, B. and J.-F. Geleyn (1992). A comprehensive radiation scheme for numerical weather prediction models with potential applications in climate simulations. *Mon. Wea. Rev. 120*, 303–325.

Robert, A. (1966). The integration of a low-order spectral form of the primitive meteorological equations. *J. Meteor. Soc. Jap. 44*, 237–245.

Rood, A. J. (1987). Numerical Advection Algorithms and their Role in Atmospheric Transport and Chemistry Models. *Rev. Geophys. 24*(1), 71–100.

Saad, Y. (1993). A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Stat. Comp. 14*, 461–469.

Saad, Y. and M. Schultz (1986). GMRES: A Generalized Minimal Residual Algorithm for Solving Large Non-Symmetric Linear Systems. *SIAM J. Sci. Stat. Comp. 7*, 856–869.

Saito, K. (1997). Semi-implicit Fully Compressible Version of the MRI Mesoscale Nonhydrostatic Model. *Geophys. Mag. 2*, 109–137.

Saito, K., G. Doms, U. Schättler, and J. Steppeler (1998). 3-D Mountain Waves by the Lokal-Modell of DWD and the MRI Mesoscale Nonhydrostatic Model. *Papers in Meteorology and Geophysics 49*, 7–19.

Schär, C., D. Leuenberger, O. Fuhrer, D. Lüthi, and C. Girard (2002). A new terrain-following vertical coordinate formulation for atmospheric prediction models. *Mon. Wea. Rev. 130*, 2459–2480.

Schlesinger, R. E. (1985). Effects of upstream-biased third-order space correction terms on multidimensional Crowley advection schemes. *Mon. Wea. Rev. 113*, 1109–1130.

Schraff, C. (1996). Data assimilation and mesoscale weather prediction: A study with a forecast model for the Alpine region. Publication 56, Swiss Meteorological Institute, Zürich.

Schraff, C. (1997). Mesoscale data assimilation and prediction of low stratus in the Alpine region. *Meteorol. Atmos. Phys. 64*, 21–50.

Schraff, C., H. Reich, A. Schomburg, K. Stephan, A. Periáñez, and R. Potthast (2016). Kilometre-scale ensemble data assimilation for the COSMO model (KENDA). *Q. J. R. Meteorol. Soc. 142*, 1453–1472.

Sharman, R., T. Kelleer, and M. Wurtele (1988). Incompressible and anelastic flow simulations on numerically generated grids. *Mon. Wea. Rev. 116*, 1124–1136.

Skamarock, W. (2006). Positive-definite and Monotonic Limiters for Unrestricted-Time-Step Transport Schemes. *Mon. Wea. Rev. 134*, 2241–2250.

Skamarock, W. and J. Klemp (1992). The stability of time-split numerical methods for the hydrostatic and nonhydrostatic elasic equations. *Mon. Wea. Rev. 120*, 2109–2127.

Skamarock, W. and J. Klemp (1994). Efficiency and accuracy of the Klemp-Wilhelmson time-splitting technique. *Mon. Wea. Rev. 122*, 2623–2630.

Skamarock, W., P. Smolarkiewicz, and J. Klemp (1997). Preconditioned Conjugate Gradient Solvers for Helmholtz Equations in Nonhydrostatic Models. *Mon. Wea. Rev. 125*, 587–599.

Smagorinsky, J. (1963). General circulation experiments with the primitive equations. *Mon. Wea. Rev. 91*(3), 99–164.

Smagorinsky, J. (1993). Some historical remarks on the use of nonlinear viscosities. In *Large Eddy Simulation of Complex Engineering and geophysical Flows*, pp. 3–36. Cambridge University Press.

Smolarkiewicz, P. (1982). The multidimensional Crowley advection scheme. *Mon. Wea. Rev. 110*, 1986–1983.

Smolarkiewicz, P. (1989). Comment on "A positive definite advection scheme obtained by nonlinear renormalization of the advective fluxes". *Mon. Wea. Rev. 117*, 2626–2632.

Smolarkiewicz, P. and G. A. Grell (1992). A class of monotone interpolation schemes. *J. Comput. Phys. 101*, 431–440.

Smolarkiewicz, P., V. Grubisic, and L. Margolin (1997). On Forward-in-Time Differencing for Fluids: Stopping Criteria for Iterative Solutions of Anelastic Pressure Equations. *Mon. Wea. Rev. 125*, 647–654.

Smolarkiewicz, P. and L. Margolin (1994). Variational solver for elliptic problems in atmospheric flows. *Appl. Math. Comp. Sci. 4*, 527–551.

Smolarkiewicz, P. and L. Margolin (1997). On forward-in-time differencing for fluids: An Eulerian/semi-Lagrangian nonhydrostatic model for stratified flows. *Atmos. Ocean 35*, 127–152.

Staniforth, A. and J. Côté (1991). Semi-Lagrangian Integration Schemes for Atmospheric Models – A Review. *Mon. Wea. Rev. 119*, 2206–2223.

Stephan, K., S. Klink, and C. Schraff (2008). Assimilation of radar-derived rain rates into the convective-scale model COSMO-DE at DWD. *Quart. J. R. Meteor. Soc. 134*, 1315–1326.

Steppeler, J. (1995). Comments on "A nonhydrostatic version of the Penn State / NCAR mesoscale model: Validation tests and simulation of an Atlantic cyclone and cold front". *Mon. Wea. Rev. 123*, 2572.

Strang, G. (1968). On the construction and comparison of difference schemes. *SIAM J. Numer. Anal. 5*, 506–517.

Tanguay, M., A. Robert, and R. Laprise (1990). A semi-implicit semi-lagrangian fully compressible regional forecast model. *Mon. Wea. Rev. 118*, 1970–1980.

Thomas, S., C. Girard, R. Benoit, M. Desgagné, and P. Pellerin (1998). A new adiabatic kernel for the MC2 model. *Atmosphere-Ocean 36*, 241–270.

Thomas, S., C. Girard, G. Doms, and U. Schättler (2000). Semi-Implicit Scheme for the DWD Lokal-Modell. *Meteorol. Atmos. Phys. 73*, 105–125.

Tiedtke, M. (1989). A comprehensive mass flux scheme for cumulus parameterization in large-scale models. *Mon. Wea. Rev. 117*, 1779–1799.

Tremback, C., J. Powell, W. Cotton, and R. Pielke (1987). The Forward in Time Upstream Advection Scheme: Extension to Higher Orders. *Mon. Wea. Rev. 115*, 540–555.

van Leer, B. (1977). Toward the ultimate conservative difference scheme. Part IV: A new approach to numerical convection. *J. Comput. Phys. 23*, 276–299.

van Mieghem, J. (1973). *Atmosperic Energetics*. Dietrich Reimer Verlag, Germany.

Vinokur, M. (1974). Conservation equations of gas dynamics in curvilinear coordinate systems. *J. Comput. Phys. 14*, 105–125.

Viviand, H. (1974). Formes conservatives des equations de la dynamique des gaz. *La Recherce Aeronautique 1*, 65–66.

von Storch, H., L. H., and F. F. (2000). A spectral nudging technique for dynamical downscaling purposes. *Mon. Wea. Rev. 128*, 3664–3673.

Wicker, L. and W. Skamarock (2002). Time-splitting methods for elastic models using forward time schemes. *Mon. Wea. Rev. 130*, 2088–2097.

Xue, M. (2000). High-order monotonic numerical diffusion and smoothing. *Mon. Wea. Rev. 128*, 2853–2864.

Xue, M., K. Droegmeier, V. Wond, A. Shapiro, and K. Brewster (1995). *Advanced Regional Prediction System, Version 4.0, User's Guide*. Norman, OK 73019-0628 USA: Centre for Analysis and Prediction of Storms (CAPS), University of Oklahoma.

Zängl, G. (2000). A modified temperature diffusion scheme for simulations over complex terrain and its application to idealized Foehn simulations. *MAP Newsletter 13*.

Zängl, G. (2012, November). Extending the Numerical Stability Limit of Terrain-Following Coordinate Models over Steep Slopes. *Mon. Wea. Rev. 140*(11), 3722–3733.