Consortium

for

Small-Scale Modelling

Technical Report No. 21

A new fast-waves solver

for the Runge-Kutta dynamical core

April 2013

DOI: 10.5676/DWD_pub/nwv/cosmo-tr_21

Deutscher Wetterdienst
MeteoSwiss
Ufficio Generale Spazio Aereo e Meteorologia
ΕΘΝΙΚΗ ΜΕΤΕΩΡΟΛΟΓΙΚΗ ΥΠΗΡΕΣΙΑ
Instytucie Meteorogii i Gospodarki Wodnej
Administratia Nationala de Meteorologie
ROSHYDROMET
Agenzia Regionale per la Protezione Ambientale del Piemonte
Agenzia Regionale per la Protezione Ambientale dell'Emilia-Romagna
Centro Italiano Ricerche Aerospaziali
Amt für GeoInformationswesen der Bundeswehr

www.cosmo-model.org

Editor: Massimo Milelli, ARPA Piemonte

A new fast-waves solver for the Runge-Kutta dynamical core

Michael Baldauf[*]

[*] Dr. Michael Baldauf
Deutscher Wetterdienst
Frankfurter Str. 135
63067 Offenbach am Main
Germany

Contents

1 Abstract

An essential part of the COSMO dynamical core is the fast waves solver. Here, the processes sound wave and gravity wave expansion are integrated and a stabilization of the whole time-splitting procedure by divergence damping is done. Sound and long gravity waves have high velocities and therefore are solved with a smaller time step (small compared to the larger time step of the advection process) and vertically implicit, to prevent from the strong time step reduction by flat grid elements in the vicinity of the ground.
This report describes a complete revision of the previously existing fast waves solver and incorporates several new aspects. First, all vertical discretizations are done with the proper distance weightings in the vertically stretched grid. Second, the so called strong conservation form of the divergence operator is used. Third, the possibility to use a fully 3D divergence damping should be possible. Fourth, the option to use a discretization of the horizontal pressure gradient, which promises to be more stable in steep terrain, should be implemented. Beyond these broader development goals several smaller improvements concerning e.g. the formulation of boundary conditions, the use of the reference state, more accurate formulation of the buoyancy term, or several other numerical operations have been done.
The report at first describes the discretization of the new fast waves solver to enable the model developer or user to understand the code structure. The second part motivates the chosen discretization by a truncation error analysis. Finally the third part assess the properties of the new solver against idealized test setups, where partially exact solutions are known.
This development is available since COSMO version 4.24 (together with INT2LM 1.20).

2 Introduction

This technical report describes a complete revision of the fast waves solver and thus the development of its new version. This new development is available since COSMO version 4.24 (available since July 2012).

The COSMO model was developed during the 1990's and at first went into operations at the end of 1999 at DWD (at this time named LM). Due to the planned usage of the COSMO model for convection-permitting resolutions, the decision to abandon the hydrostatic approximation was made from the beginning. The rising of massively parallel computers at this time further led to the decision not to use any anelastic approximation, but to use the fully compressible equation system. The latter equation system leads to a fast waves solver which is relatively easy to parallelize. G. Doms (**?**) developed the first fast waves solver for the LM, which mainly based on the solver of the MM5 (**?**, **?**), and further using ideas of the leapfrog based split explicit solver by **?** and **?**. A. Gassmann abandoned the 3-time level leapfrog scheme and implemented a 2-time level Runge-Kutta scheme (**?** and **?**). Additionally, she introduced e.g. the dynamical bottom boundary condition (**?**) in the fast waves solver. This at this time improved a lot the model behavior at the lower boundaries. J. Förstner implemented the Runge-Kutta scheme of **?** into the COSMO model and introduced T' instead of T into the fast waves solver. This resulted in a reduced noise in the vertical velocity field, mainly due to the better behavior of the temperature advection. Later on, G. Zängl has demonstrated that the introduction of vertical weightings in some of the explicit terms of the fast waves solver largely improves the pressure bias. This was one of the prerequisites to be able to use the Runge-Kutta dynamical core for the 7 km COSMO applications at the different COSMO services (e.g. COSMO-EU at DWD).

The main goals of this development project 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. **?** 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 **?**. This z-plane treatment should result in a more stable behavior in steep terrain.

These new features did not only require changes at almost every place in the fast waves code, but partially required a restructuring of the code logic, too. Together with the aim to introduce a bit more descriptive variable names, the decision was made, to reprogram the whole fast waves solver from scratch; a work which started at the end of 2010. This approach resulted in additional smaller improvements at other places, too, concerning e.g. the formulation of boundary conditions, the use of the reference state, a more accurate formulation of the buoyancy term, and others.

This report is subdivided into three parts. In the first part, sections 3-7, the discretization of the Euler equations, the setup and the solution of the vertically implicit tridiagonal equation for w is explained. To offer the reader an as much as possible consistent description, several repetitions of material presented in **?** (or the revised version **?**) have been unavoidable. But with this, the model developer (and model user) should be able to understand the code structure of the module `fast_waves_sc`.
To understand the embedding of the fast waves solver in the whole Runge-Kutta time integration framework, the reader should consult chapter 8 in **?**. Also not contained here is the stability analysis of the COSMO dynamical core. For this we refer to **?**. There, the reader will find reasons why the time integration scheme is done in the current manner.
In the second part (section 8), a motivation is given for one of the main development goals, namely the improved vertical discretization, by a truncation error analysis.
In the third part (9 and 10) a broad investigation of the properties of the new fast waves solver by idealized tests is performed. A few statements about model performance in real cases closes this part.

3 The Euler equations

The Euler equations in spherical, terrain following coordinates (λ, ϕ, ζ) with $\zeta = \zeta(\lambda, \phi, z)$, $r = r_{earth} + z$, can be written as

$$\frac{\partial u}{\partial t} = -\frac{1}{\rho}\,\frac{1}{r\cos\phi}\left(\frac{\partial p'}{\partial\lambda} + \frac{\partial\zeta}{\partial\lambda}\frac{\partial p'}{\partial\zeta}\right) + \frac{1}{\rho}\,\frac{1}{r\cos\phi}\left(\frac{\partial\,\alpha^h_{div}\,\rho\,D}{\partial\lambda} + \frac{\partial\zeta}{\partial\lambda}\frac{\partial\,\alpha^h_{div}\,\rho\,D}{\partial\zeta}\right) + \left.\frac{\partial u}{\partial t}\right|_{slow} \tag{1}$$

$$\frac{\partial v}{\partial t} = -\frac{1}{\rho}\,\frac{1}{r}\left(\frac{\partial p'}{\partial\phi} + \frac{\partial\zeta}{\partial\phi}\frac{\partial p'}{\partial\zeta}\right) + \frac{1}{\rho}\,\frac{1}{r}\left(\frac{\partial\,\alpha^h_{div}\,\rho\,D}{\partial\phi} + \frac{\partial\zeta}{\partial\phi}\frac{\partial\,\alpha^h_{div}\,\rho\,D}{\partial\zeta}\right) + \left.\frac{\partial v}{\partial t}\right|_{slow} \tag{2}$$

$$\frac{\partial w}{\partial t} = -\frac{1}{\rho}\,\frac{\partial\zeta}{\partial z}\frac{\partial p'}{\partial\zeta} + g\left(\frac{p_0}{p}\frac{T'}{T_0} - \frac{p'}{p} + \frac{p_0}{p}\frac{T}{T_0}\,q_x\right) + \frac{1}{\rho}\frac{\partial\zeta}{\partial z}\frac{\partial\,\alpha^v_{div}\,\rho\,D}{\partial\zeta} + \left.\frac{\partial w}{\partial t}\right|_{slow} \tag{3}$$

$$\frac{\partial p'}{\partial t} = -\frac{c_p}{c_V}p\,D + g\rho_0 w + \left.\frac{\partial p'}{\partial t}\right|_{slow} \tag{4}$$

$$\frac{\partial T'}{\partial t} = -\frac{R}{c_V}T\,D - \frac{\partial T_0}{\partial z}w + \left.\frac{\partial T'}{\partial t}\right|_{slow} \tag{5}$$

The denotations are the same as in **?**, in particular $T = T_0 + T'$ and $p = p_0 + p'$ are split into the stationary reference state and deviations. Here only the 'fast' processes are written down and grouped into sound (left group of terms on the rhs), buoyancy (middle group of terms in the w, p', and T'-equations) and artificial divergence damping (right group in the three momentum equations). $\left.\frac{\partial u}{\partial t}\right|_{slow}, \ldots, \left.\frac{\partial T'}{\partial t}\right|_{slow}$ denote the remaining ('slow') processes advection, Coriolis force and all physical parameterizations.

The occurring coordinate derivatives may be understood as

$$\frac{\partial\zeta}{\partial\lambda} \equiv \left.\frac{\partial\zeta}{\partial\lambda}\right|_z \equiv \frac{\partial\zeta(\lambda,\phi,z)}{\partial\lambda}, \quad \frac{\partial\zeta}{\partial\phi} \equiv \left.\frac{\partial\zeta}{\partial\phi}\right|_z \equiv \frac{\partial\zeta(\lambda,\phi,z)}{\partial\phi}, \quad \frac{\partial\zeta}{\partial z} \equiv \left.\frac{\partial\zeta}{\partial z}\right|_z \equiv \frac{\partial\zeta(\lambda,\phi,z)}{\partial z}.$$

We remind the abbreviations and relations valid for the COSMO coordinate transformation

$$\sqrt{g} := -\frac{\partial z}{\partial\zeta} = -\frac{1}{\frac{\partial\zeta}{\partial z}} \quad \Leftrightarrow \quad \frac{\partial\zeta}{\partial z} = -\frac{1}{\sqrt{g}}, \tag{6}$$

$$\left.\frac{\partial z}{\partial\lambda}\right|_\zeta = -\frac{\partial z}{\partial\zeta}\frac{\partial\zeta}{\partial\lambda} \quad \Leftrightarrow \quad \frac{\partial\zeta}{\partial\lambda} = \frac{1}{\sqrt{g}}\left.\frac{\partial z}{\partial\lambda}\right|_\zeta, \tag{7}$$

$$\left.\frac{\partial z}{\partial\phi}\right|_\zeta = -\frac{\partial z}{\partial\zeta}\frac{\partial\zeta}{\partial\phi} \quad \Leftrightarrow \quad \frac{\partial\zeta}{\partial\phi} = \frac{1}{\sqrt{g}}\left.\frac{\partial z}{\partial\phi}\right|_\zeta. \tag{8}$$

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 (**?**, eq. (3.122))

$$D \equiv \operatorname{div}\mathbf{v} = \frac{1}{r^2\cos\phi}\frac{1}{\sqrt{g}}\left[\frac{\partial}{\partial\lambda}\left(r\sqrt{g}\,u\right) + \frac{\partial}{\partial\phi}\left(r\cos\phi\,\sqrt{g}\,v\right) + \frac{\partial rZ}{\partial\zeta}\right] \tag{9}$$

with the definition

$$Z := r\cos\phi\,\sqrt{g}\,\dot{\zeta} = \frac{\partial z}{\partial\lambda}\,u + \frac{\partial z}{\partial\phi}\cos\phi\,v - r\cos\phi\,w. \tag{10}$$

Obviously, the divergence is expressed on 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

$$D = \frac{1}{r\cos\phi}\frac{1}{\sqrt{g}}\left[\frac{\partial}{\partial\lambda}\left(\sqrt{g}\,u\right) + \frac{\partial}{\partial\phi}\left(\cos\phi\,\sqrt{g}\,v\right) + \frac{\partial Z}{\partial\zeta}\right]. \tag{11}$$

It is advisable to introduce some other abbreviations

$$\tilde{d}_{hor} := \frac{\partial}{\partial\lambda}\left(\sqrt{g}\,u\right) + \frac{\partial}{\partial\phi}\left(\cos\phi\,\sqrt{g}\,v\right), \tag{12}$$

$$\tilde{d}_{vert} := \frac{\partial Z}{\partial\zeta}, \tag{13}$$

so we can write the divergence as

$$D = \frac{1}{r\cos\phi}\frac{1}{\sqrt{g}}\left[\tilde{d}_{hor} + \tilde{d}_{vert}\right]. \tag{14}$$

Furthermore we define

$$Z = Z_{hor} + Z_{vert} \tag{15}$$

with

$$Z_{hor} := Z_x + Z_y, \tag{16}$$

$$Z_x := \frac{\partial z}{\partial\lambda}\,u, \tag{17}$$

$$Z_y := \frac{\partial z}{\partial\phi}\cos\phi\,v, \tag{18}$$

$$Z_{vert} := -r\cos\phi\,w. \tag{19}$$

The artificial divergence damping has been written in a more efficient form - compared to the original idea of having a damping of the divergence[1] - analogous to **?**. This form allows a common discretization of $\tilde{p} = p' - \alpha_{div}^{h}\rho D$ in the horizontal momentum equations. To indicate the optional divergence damping in the w-equation, the diffusion coefficient α_{div}^{v} carries an upper index 'v'.

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

$$p = \rho\,R_d\,(1+q_x)\,T, \tag{23}$$

where

$$q_x := \left(\frac{R_v}{R_d} - 1\right)q_v - q_{cond}, \tag{24}$$

describes the so-called 'water loading' in the buoyancy term. From this we can derive the

[1] A more self-evident form of the divergence damping leads to the three momentum equations

$$\frac{\partial u}{\partial t} = -\frac{1}{\rho}\,\frac{1}{r\cos\phi}\left(\frac{\partial p'}{\partial\lambda} + \frac{\partial\zeta}{\partial\lambda}\frac{\partial p'}{\partial\zeta}\right) + \alpha_{div}^{h}\frac{1}{r\cos\phi}\left(\frac{\partial D}{\partial\lambda} + \frac{\partial\zeta}{\partial\lambda}\frac{\partial D}{\partial\zeta}\right) + \left.\frac{\partial u}{\partial t}\right|_{slow}, \tag{20}$$

$$\frac{\partial v}{\partial t} = -\frac{1}{\rho}\,\frac{1}{r}\left(\frac{\partial p'}{\partial\phi} + \frac{\partial\zeta}{\partial\phi}\frac{\partial p'}{\partial\zeta}\right) + \alpha_{div}^{h}\frac{1}{r}\left(\frac{\partial D}{\partial\phi} + \frac{\partial\zeta}{\partial\phi}\frac{\partial D}{\partial\zeta}\right) + \left.\frac{\partial v}{\partial t}\right|_{slow}, \tag{21}$$

$$\frac{\partial w}{\partial t} = -\frac{1}{\rho}\left(\frac{\partial\zeta}{\partial z}\frac{\partial p'}{\partial\zeta}\right) + g\left(\frac{p_0}{p}\frac{T'}{T_0} - \frac{p'}{p} + \frac{p_0}{p}\frac{T}{T_0}\,q_x\right) + \alpha_{div}^{v}\frac{\partial\zeta}{\partial z}\frac{\partial D}{\partial\zeta} + \left.\frac{\partial w}{\partial t}\right|_{slow}. \tag{22}$$

However the discretization of this form leads to much more inefficient code, because - in contrast to the above form - the divergence terms cannot be treated together with the pressure terms. Beyond this, both forms have the same slight tendency to instability in a Bretherton-transformed equation system (see (**?**, Appendix B)). Hence, there is no advantage in using this form.

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

$$
-g\frac{\rho'}{\rho} = +g\frac{p_0}{T_0}\frac{T}{p}\left(-\frac{T_0}{T}\frac{p'}{p_0} + \frac{T'}{T} + q_x\right) = +g\left(-\frac{p'}{p} + \frac{p_0}{p}\frac{T'}{T_0} + \frac{p_0}{p}\frac{T}{T_0}q_x\right), \tag{25}
$$

i.e. densities are expressed by pressure and temperature in an efficient manner. By the way, inserting $T' = T - T_0$ results in

$$
-g\frac{\rho'}{\rho} = +g\left(-1 + \frac{p_0}{p}\frac{T}{T_0}(1+q_x)\right). \tag{26}
$$

This form is used for the 'dynamical bottom boundary condition' (section 5.2.1). The reason is, that for parallelization no boundary exchange of T' is needed; instead an estimation of T by the starting value is used.

4 Discretization

4.1 Numerical operators for the spatial discretization

To define the numerical operators for the *spatial* discretizations we have to remember the distinct grid positions, where the different variables are located in the COSMO model. The main grid positions with integer indices (i, j, k) are those, where the scalars p, T, ρ, p_0, T_0, ρ_0, p', T', q_v, q_c, ... and the divergence D are located. The velocity components u are defined at the staggered grid position $(i+\frac{1}{2}, j, k)$, v at $(i, j+\frac{1}{2}, k)$, and w at $(i, j, k-\frac{1}{2})$. In the following, 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 1).

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 $(i, j+\frac{1}{2}, k-\frac{1}{2})$.

For the spatial discretization we introduce the following 2-point operators, leading to at most second order formulas. The limitation to two point operators in the vertical stems from the fact, that only a tridiagonal implicit equation system should occur. For the horizontal operations in principle higher order discretizations may be used (this is currently under investigation by A. Will and J. Ogaja at the University in Cottbus), but here only 2-point formulas are used, too.

We start with the spatial *averaging* operators. Horizontal averaging is done by

$$\overline{\psi}^{\lambda}\Big|_{i,j,k} \equiv A_\lambda \psi|_{i,j,k} \;:=\; \frac{1}{2}(\psi_{i-\frac{1}{2},j,k} + \psi_{i+\frac{1}{2},j,k}), \tag{27}$$

$$\overline{\psi}^{\phi}\Big|_{i,j,k} \equiv A_\phi \psi|_{i,j,k} \;:=\; \frac{1}{2}(\psi_{i,j-\frac{1}{2},k} + \psi_{i,j+\frac{1}{2},k}). \tag{28}$$

These are second order discretizations. In the following, the notation with an over-bar '$\overline{\psi}$' is mostly used for prefactors, whereas the operator notation '$A\psi$' is mostly used for the prognostic variables, for which an implicit equation system will be derived.

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

$$z^{(m)}_{i,j,k} := \frac{1}{2}\left(z^{(h)}_{i,j,k-\frac{1}{2}} + z^{(h)}_{i,j,k+\frac{1}{2}} \right). \tag{29}$$

Accordingly an averaging from half level variables (e.g. w) to the main level is done by arithmetic averaging:

$$\overline{\psi}^{\zeta}\Big|_{i,j,k} \equiv A_\zeta \psi|_{i,j,k} := \frac{1}{2}(\psi_{i,j,k-\frac{1}{2}} + \psi_{i,j,k+\frac{1}{2}}). \tag{30}$$

But averaging from main level variables (e.g. p', T', ...) to the half level position is done by a weighting

$$\overline{\psi}^{\zeta,N}\Big|_{i,j,k-\frac{1}{2}} \equiv A^N_\zeta \psi|_{i,j,k-\frac{1}{2}} := g_{i,j,k-\frac{1}{2}}\psi_{i,j,k} + (1-g_{i,j,k-\frac{1}{2}})\psi_{i,j,k-1} \tag{31}$$

with

$$g_{i,j,k-\frac{1}{2}} := \frac{z^{(h)}_{i,j,k-\frac{1}{2}} - z^{(h)}_{i,j,k-\frac{3}{2}}}{z^{(h)}_{i,j,k+\frac{1}{2}} - z^{(h)}_{i,j,k-\frac{3}{2}}}. \tag{32}$$

To be complete, we also note the extrapolation formula

$$\psi_{i,j,k+\frac{1}{2}} = -(1-g_{i,j,k-\frac{1}{2}})\,\psi_{i,j,k-1} + (2-g_{i,j,k-\frac{1}{2}})\,\psi_{i,j,k}. \tag{33}$$

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: $A^{\overline{N}}_\zeta$ (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

$$\delta_\lambda \psi|_{i,j,k} \;:=\; \frac{\psi_{i+\frac{1}{2},j,k} - \psi_{i-\frac{1}{2},j,k}}{\Delta\lambda}, \tag{34}$$

$$\delta_\phi \psi|_{i,j,k} \;:=\; \frac{\psi_{i,j+\frac{1}{2},k} - \psi_{i,j-\frac{1}{2},k}}{\Delta\phi}, \tag{35}$$

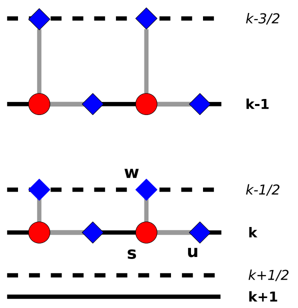

Figure 1: Denotation of the levels and the position of variables in the COSMO-model. Grey connection lines indicate staggered grid positions with the same indices in the COSMO program code.

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 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

$$\delta_\zeta \psi|_{i,j,k} := \frac{\psi_{i,j,k+\frac{1}{2}} - \psi_{i,j,k-\frac{1}{2}}}{\Delta\zeta}. \qquad (36)$$

The derivative of a main level variable to the half level position could be done in the same way

$$\delta_\zeta^N \psi|_{i,j,k-\frac{1}{2}} := \frac{\psi_{i,j,k} - \psi_{i,j,k-1}}{\Delta\zeta} \qquad (37)$$

with the argumentation, that from two 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':

$$\delta_\zeta^{(s,u)} \psi := A_\lambda \delta_\zeta A_\zeta^N \psi \qquad (38)$$

An equally accurate discretization is

$$\delta_\zeta^{(s,u)} \psi := \delta_\zeta A_\zeta^{\bar{N}} A_\lambda \psi \qquad (39)$$

For the vertical derivative 'u-Position to a scalar point' one can use

$$\delta_\zeta^{(u,s)} u := A_\lambda \delta_\zeta A_\zeta^{\bar{N}} u \qquad (40)$$

analogous to eq. (38) or alternatively

$$\delta_\zeta^{(u,s)} u := \delta_\zeta A_\zeta^N A_\lambda u. \qquad (41)$$

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.

- The following metric term occurring in the 'strong conservation'-form of the divergence is best discretized as

$$discr.^{(s)} \left[\frac{\partial z}{\partial \lambda} u \right] = \delta_\zeta \left(A_\lambda \left(\frac{\partial z}{\partial \lambda} A_\zeta^{\bar{N}} u \right) \right). \tag{42}$$

  Analogous for the y-direction.

- Metric terms of ∇p and ∇D

$$discr.^{(u)} \left[\frac{\partial \zeta}{\partial \lambda} \frac{\partial \psi}{\partial \zeta} \right] \tag{43}$$

  occur in the explicit integration of the horizontal momentum equations (hence, at the u- or v-position). The calculation of the metrical prefactor is done in any case by the partitioning

$$\frac{\partial \zeta}{\partial \lambda} = - \frac{\partial \zeta}{\partial z} \frac{\partial z}{\partial \lambda}. \tag{44}$$

  Here $\frac{\partial \zeta}{\partial z}$ is defined most naturally at the 's'-position and $\frac{\partial z}{\partial \lambda}$ most naturally at the 'uw'-position. Therefore

$$discr. \left[\frac{\partial \zeta}{\partial \lambda} \right] = - \left(A_\lambda \frac{\partial \zeta}{\partial z} \right) \cdot \left(A_\zeta \frac{\partial z}{\partial \lambda} \right). \tag{45}$$

  This results in

$$discr.^{(u)} \left[\frac{\partial \zeta}{\partial \lambda} \frac{\partial \psi}{\partial \zeta} \right] = - \left(A_\zeta \frac{\partial z}{\partial \lambda} \right) \cdot \left(A_\lambda \underbrace{\left(\frac{\partial \zeta}{\partial z} \delta_\zeta A_\zeta^N \psi \right)}_{= discr.^{(s)} \left[\frac{\partial \psi}{\partial z} \right]} \right). \tag{46}$$

  This means that a ζ-derivative can be expressed most naturally by a z-derivative. Analogous:

$$discr.^{(v)} \left[\frac{\partial \zeta}{\partial \phi} \frac{\partial \psi}{\partial \zeta} \right] = - \left(A_\zeta \frac{\partial z}{\partial \phi} \right) \cdot \left(A_\phi \underbrace{\left(\frac{\partial \zeta}{\partial z} \delta_\zeta A_\zeta^N \psi \right)}_{= discr.^{(s)} \left[\frac{\partial \psi}{\partial z} \right]} \right). \tag{47}$$

- For the z-derivative of a scalar variable at the s-position we use in the following

$$discr.^{(s)} \left[\frac{\partial \psi}{\partial z} \right] \equiv \delta_z^{(s,s)} \psi := \frac{\partial \zeta}{\partial z} \delta_\zeta A_\zeta^N \psi. \tag{48}$$

- We further remark that $\sqrt{g}$ at the 'w'-position can be better calculated directly by

$$\left. \frac{\partial z}{\partial \zeta} \right|_{k-\frac{1}{2}} = \frac{z_k^{(m)} - z_{k-1}^{(m)}}{1} = \frac{\frac{1}{2}\left(z_{k+\frac{1}{2}}^{(h)} + z_{k-\frac{1}{2}}^{(h)} \right) - \frac{1}{2}\left(z_{k-\frac{1}{2}}^{(h)} + z_{k-\frac{3}{2}}^{(h)} \right)}{1} = \frac{z_{k+\frac{1}{2}}^{(h)} - z_{k-\frac{3}{2}}^{(h)}}{2} \tag{49}$$

  instead of a weighted averaging, due to the special definition of the position of the main levels (hint by A. Will (personal communication)).

Some definitions of these operators may seem a bit ad hoc for the reader. A deeper reasoning for them is given in section 8, where truncation errors are calculated and those operators with the smallest error are chosen.

4.2 Discretization of the Euler equations

In the last section, spatial discretization operators have been defined, which are necessary for the numerical treatment of the Euler equations. The basic time integration idea behind the fast waves solver is the horizontally explicit-vertically implicit (HE-VI) scheme, where the horizontal integration is further done by a forward-backward scheme and the vertical implicit step is a general Crank-Nicholson scheme (**?**). It is therefore reasonable to define a general time averaging operator, here called a 'Crank-Nicholson time averaging operator', by

$$\hat{\beta}_i^{(a)} \phi := \beta_i^{(a)} \phi^{n+1} + \left(1 - \beta_i^{(a)}\right) \phi^n. \tag{50}$$

The upper index (a) denotes a process, e.g. the terms for sound expansion (s), the lower index i simply enumerates terms. This nomenclature for the Crank-Nicholson weights is adopted from **?**. 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 (51) \\
\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) + \\
&\quad + \overline{\frac{1}{\rho}}^{\phi} \, \overline{\frac{1}{r}}^{\phi} \left(\delta_\phi (\alpha_{div}^h \rho D_{(uv)}) - \overline{\frac{\partial z}{\partial \phi}}^{\zeta} A_\phi \left(\delta_z^{(s,s)} (\alpha_{div}^h \rho D_{(uv)}) \right) \right) + \left.\frac{\partial v}{\partial t}\right|_{slow} \qquad (52) \\
\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_{div}^v \rho D_{(w)}) + \left.\frac{\partial w}{\partial t}\right|_{slow} \qquad (53) \\
\frac{p'^{n+1} - p'^n}{\Delta t} &= -\frac{c_p}{c_V} p D_{(p)} + \hat{\beta}_3^b g \rho_0 \, A_\zeta w + \left.\frac{\partial p'}{\partial t}\right|_{slow} \qquad (54) \\
\frac{T'^{n+1} - T'^n}{\Delta t} &= -\frac{R}{c_V} T D_{(T)} - \hat{\beta}_4^b \frac{\partial T_0}{\partial z} A_\zeta w + \left.\frac{\partial T'}{\partial t}\right|_{slow} \qquad (55)
\end{aligned}$$

Here, different divergences $D_{(p)}$, $D_{(T)}$, $D_{(uv)}$, $D_{(w)}$ have been introduced, because they can carry different Crank-Nicholson weights (see sections 4.2.1).

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

$$\overline{\frac{1}{p}}^{\zeta,N} = \frac{1}{p_0(z_{k+1/2}) + A_\zeta^N p'}, \qquad \overline{\frac{p_0}{p}}^{\zeta,N} = \frac{1}{1 + \frac{A_\zeta^N p'}{p_0(z_{k+1/2})}}, \tag{56}$$

$$\overline{\frac{1}{T_0}}^{\zeta,N} = \frac{1}{T_0(z_{k+1/2})}, \qquad \overline{\frac{T}{T_0}}^{\zeta,N} = 1 + \frac{A_\zeta^N T'}{T_0(z_{k+1/2})}. \tag{57}$$

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)). Analogous to that, $\overline{\frac{1}{\rho}}^{\zeta,N}$ in eq. (53) is calculated with the aid of the ideal gas equation by the values of $T_0(z_{k+1/2}) + A_\zeta^N T'$ and $p_0(z_{k+1/2}) + A_\zeta^N p'$. In particular $\overline{\frac{\partial \zeta}{\partial z}}^{\zeta,N}$ is calculated by (49).

4.2.1 Discretization of the divergence

The horizontal contributions of the divergence can be discretized by

$$\tilde{d}_{hor} := \delta_\lambda(\overline{\sqrt{g}}^\lambda \, u) + \delta_\phi(\overline{\cos\phi}^\phi \, \overline{\sqrt{g}}^\phi \, v). \tag{58}$$

For the vertical contribution

$$\tilde{d}_{vert} := \delta_\zeta Z, \tag{59}$$

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

$$\begin{aligned} Z_{hor} &= Z_x + Z_y, & (60)\\ Z_x &= A_\lambda \left(\frac{\partial z}{\partial \lambda} \, A_\zeta^{\bar{N}} \, u \right), & (61)\\ Z_y &= A_\phi \left(\frac{\partial z}{\partial \phi} \, \cos\phi \, A_\zeta^{\bar{N}} \, v \right). & (62) \end{aligned}$$

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

$$Z_{vert} := -r \cos\phi \, w \tag{63}$$

can be calculated at the boundaries, too.

The different divergence terms are:

$$D_{(p)} = \frac{1}{r\cos\phi} \frac{1}{\sqrt{g}} \left[\hat{\beta}_3^s \, \tilde{d}_{hor} + \hat{\beta}_8^s \, \delta_\zeta Z_{hor} + \hat{\beta}_4^s \, \delta_\zeta Z_{vert} \right]. \tag{64}$$

$$D_{(T)} = \frac{1}{r\cos\phi} \frac{1}{\sqrt{g}} \left[\hat{\beta}_5^s \, \tilde{d}_{hor} + \hat{\beta}_9^s \, \delta_\zeta Z_{hor} + \hat{\beta}_6^s \, \delta_\zeta Z_{vert} \right]. \tag{65}$$

[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.

$$D_{(uv)} = \frac{1}{r\cos\phi}\,\frac{1}{\sqrt{g}}\left[\hat{\beta}_1^d\,\tilde{d}_{hor} + \hat{\beta}_5^d\,\delta_\zeta Z_{hor} + \hat{\beta}_2^d\,\delta_\zeta Z_{vert}\right]. \tag{66}$$

$$D_{(w)} = \frac{1}{r\cos\phi}\,\frac{1}{\sqrt{g}}\left[\hat{\beta}_3^d\,\tilde{d}_{hor} + \hat{\beta}_6^d\,\delta_\zeta Z_{hor} + \hat{\beta}_4^d\,\delta_\zeta Z_{vert}\right]. \tag{67}$$

4.2.2 Mahrer discretization of the horizontal pressure gradients

The basic idea behind the discretization of the horizontal pressure gradient by **?** 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 **?**, the then required *extrapolation* of p' often leads to similar instabilities as the conventional discretization. **?** 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 (**?**, **?**) 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 (see sections 9.4 and 9.5). 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 (section 8.2). 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.

5 Boundary conditions

5.1 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

$$\dot{\zeta} = 0, \tag{68}$$

which is equivalent to

$$w = \frac{1}{r\cos\phi}\frac{\partial z}{\partial\lambda}u + \frac{1}{r}\frac{\partial z}{\partial\phi}v. \tag{69}$$

As mentioned above, the vertically implicit solver needs this boundary condition at time level $n+1$. This is not a problem, because u^{n+1} and v^{n+1} are already known by the forward step of the horizontal momentum equations.

At the top boundary we can simply prescribe

$$\dot{\zeta}_{i,j,k=\frac{1}{2}} = 0 \quad \Rightarrow \quad w^{n+1}_{i,j,\frac{1}{2}} = 0. \tag{70}$$

At the lower boundary the free slip condition reduces to

$$w^{n+1}_{i,j,ke+\frac{1}{2}} = \frac{1}{r\cos\phi}\left[A_\lambda\left(\frac{\partial z}{\partial\lambda}\,u^{n+1}_{(sfc)}\right) + A_\phi\left(\frac{\partial z}{\partial\phi}\,\cos\phi\, v^{n+1}_{(sfc)}\right)\right], \tag{71}$$

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.

5.2 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.

5.2.1 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. Th 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] (**?**, section 5.4).

For three arbitrarily prescribed points

$$z_1, \quad z_2 = z_1 + h_1, \quad z_3 = z_1 + h_2, \tag{72}$$

one can derive a one-sided derivative formula of 2nd order for a function $f(z)$ in z_1 by

$$af(z_1) + bf(z_2) + cf(z_3) = (a+b+c)f + (bh_1 + ch_2)f' + \left(b\frac{h_1^2}{2} + c\frac{h_2^2}{2}\right)f'' + \dots. \tag{73}$$

This results in the weights

$$\begin{aligned} a &= -(b+c), && (74)\\ b &= \frac{1 - ch_2}{h_1}, && (75)\\ c &= \frac{h_1}{h_2(h_1 - h_2)}. && (76)\end{aligned}$$

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 boundary' (**?**), described in the next section 5.2.1.

At the top boundary a one sided derivative formula of first order is sufficient.

[3] Therefore the switch `ldyn_bbc=.FALSE.` has a different meaning in the two fast waves solver versions.

Dynamic bottom boundary condition for the pressure In section 5.4 of **?** the strong influence of the formulation of the lower boundary condition for the pressure p' was mentioned. Instead of a pressure extrapolation, **?** 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

$$\dot{\zeta} = \frac{u}{r\cos\phi} \left.\frac{\partial \zeta}{\partial \lambda}\right|_z + \frac{v}{r} \left.\frac{\partial \zeta}{\partial \phi}\right|_z + w\frac{\partial \zeta}{\partial z} = 0. \tag{77}$$

The vertical pressure gradient occurs in the three momentum equations, therefore we differentiate by time to obtain

$$\frac{\partial \dot{\zeta}}{\partial t} = \frac{1}{r\cos\phi} \left.\frac{\partial \zeta}{\partial \lambda}\right|_z \frac{\partial u}{\partial t} + \frac{1}{r} \left.\frac{\partial \zeta}{\partial \phi}\right|_z \frac{\partial v}{\partial t} + \frac{\partial \zeta}{\partial z}\frac{\partial w}{\partial t} = 0. \tag{78}$$

Here we insert the momentum equations (1)-(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

$$F_u := -\frac{1}{\rho}\frac{1}{r\cos\phi}\frac{\partial \tilde{p}}{\partial \lambda} + \left.\frac{\partial u}{\partial t}\right|_{slow} \tag{79}$$

$$F_v := -\frac{1}{\rho}\frac{1}{r}\frac{\partial \tilde{p}}{\partial \phi} + \left.\frac{\partial v}{\partial t}\right|_{slow} \tag{80}$$

$$F_w := +g\left(-1 + \frac{p_0}{p}\frac{T}{T_0}(1+q_x)\right) + \frac{1}{\rho}\frac{\partial \zeta}{\partial z}\frac{\partial(\alpha^v_{div} - \alpha^h_{div})\rho D}{\partial \zeta} + \left.\frac{\partial w}{\partial t}\right|_{slow}. \tag{81}$$

In F_w the form (26) of the buoyancy term was used. Sorting by $\frac{\partial \zeta}{\partial z}\frac{\partial p'}{\partial \zeta}$ finally results in

$$\frac{\partial \zeta}{\partial z}\frac{\partial p'}{\partial \zeta}\frac{1}{\rho}\left[\left(\frac{1}{r\cos\phi}\frac{\partial z}{\partial \lambda}\right)^2 + \left(\frac{1}{r}\frac{\partial z}{\partial \phi}\right)^2 + 1\right] = -\frac{1}{r\cos\phi}\frac{\partial z}{\partial \lambda}F_u - \frac{1}{r}\frac{\partial z}{\partial \phi}F_v + F_w. \tag{82}$$

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.

5.2.2 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. (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

$$\beta_8^s = \beta_4^s, \quad \text{and} \quad \beta_9^s = \beta_6^s. \tag{83}$$

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.

6 The linear equation system for the vertical velocity

6.1 Setup of the equation system

To get rid of the sound speed limitation, the Euler equations are solved vertically implicit. To derive the equation system we write the Euler equations in the following linear form

$$\mathbf{A} \begin{pmatrix} u^{n+1} \\ v^{n+1} \\ w^{n+1} \\ p'^{n+1} \\ T'^{n+1} \end{pmatrix} = \mathbf{b}^n$$

with

$$\mathbf{A} = \begin{pmatrix} \frac{1}{\Delta t}\mathbf{1} & 0 & 0 & \mathbf{A}_{up} & 0 \\ 0 & \frac{1}{\Delta t}\mathbf{1} & 0 & \mathbf{A}_{vp} & 0 \\ \mathbf{A}_{wu} & \mathbf{A}_{wv} & \mathbf{A}_{ww} & \mathbf{A}_{wp} & \mathbf{A}_{wT} \\ \mathbf{A}_{pu} & \mathbf{A}_{pv} & \mathbf{A}_{pw} & \frac{1}{\Delta t}\mathbf{1} & 0 \\ \mathbf{A}_{Tu} & \mathbf{A}_{Tv} & \mathbf{A}_{Tw} & 0 & \frac{1}{\Delta t}\mathbf{1} \end{pmatrix}, \tag{84}$$

and

$$\mathbf{b}^n = (b_u, b_v, b_w, b_p, b_T)^T. \tag{85}$$

Of course, nearly all the terms arising in the Euler equations are non-linear. But this non-linearity in the sound and buoyancy terms is small enough, that in every small time step, the coefficients can be considered as quasi constant. They can even be considered as constant through the whole fast waves integration procedure[4].

Here the following operators have been used for the horizontal momentum equations

$$\mathbf{A}_{up} = \overline{\frac{1}{\rho}}^{\lambda} \overline{\frac{1}{r\cos\phi}}^{\lambda} \left(\beta_1^s \, \delta_\lambda - \beta_7^s \overline{\frac{\partial z}{\partial \lambda}}^{\zeta} A_\lambda \delta_z^{(s,s)} \right), \tag{86}$$

$$\mathbf{A}_{vp} = \overline{\frac{1}{\rho}}^{\phi} \overline{\frac{1}{r}}^{\phi} \left(\beta_1^s \, \delta_\phi - \beta_7^s \overline{\frac{\partial z}{\partial \phi}}^{\zeta} A_\phi \delta_z^{(s,s)} \right), \tag{87}$$

In $\mathbf{A}$ we have already used the fact, that the artificial divergence damping in the u- and v-equation must be treated explicitly, i.e. $\beta_1^d = \beta_2^d = \beta_5^d = 0$, otherwise no tridiagonal

[4] There is an internal switch `l_small_pert_in_pT`. If it is set to `.TRUE.`, then the coefficients (i.e. the lhs) of this equation system is calculated only once per *large* time step. If it is set to `.FALSE.`, then they are set in every small time step. The latter setting may be an interesting, although efficiency reducing, option in heavy weather conditions, e.g. in the simulation of tornadoes.

system can be achieved. We will see below that the horizontal pressure gradient terms must be treated fully explicit, too; therefore, $\mathbf{A}_{up} = 0$ and $\mathbf{A}_{vp} = 0$.

For the vertical momentum equation we get

$$\mathbf{A}_{wu} \;=\; -\overline{\left(\frac{1}{\rho}\right)}^{\zeta,N} \overline{\frac{\partial \zeta}{\partial z}}^{\zeta,N} \delta_\zeta^N \left[\alpha_{div}^v\, \rho\, \frac{1}{r\cos\phi}\, \frac{1}{\sqrt{g}} \left(\beta_3^d\, \delta_\lambda(\overline{\sqrt{g}}^\lambda \cdot ...) + \beta_6^d\, \mathbf{B}_x\right)\right], \tag{88}$$

$$\mathbf{A}_{wv} \;=\; -\overline{\left(\frac{1}{\rho}\right)}^{\zeta,N} \overline{\frac{\partial \zeta}{\partial z}}^{\zeta,N} \delta_\zeta^N \left[\alpha_{div}^v\, \rho\, \frac{1}{r\cos\phi}\, \frac{1}{\sqrt{g}} \left(\beta_3^d\, \delta_\phi(\overline{\cos\phi\,\sqrt{g}}^\phi \cdot ...) + \beta_6^d\, \mathbf{B}_y\right)\right], \tag{89}$$

$$\mathbf{A}_{ww} \;=\; \frac{1}{\Delta t}\mathbf{1} + \overline{\left(\frac{1}{\rho}\right)}^{\zeta,N} \overline{\frac{\partial \zeta}{\partial z}}^{\zeta,N} \delta_\zeta^N \left[\alpha_{div}^v\, \rho\, \frac{1}{r\cos\phi}\, \frac{1}{\sqrt{g}}\, \beta_4^d\, \delta_\zeta(r\,\cos\phi\, \cdot ...)\right], \tag{90}$$

$$\mathbf{A}_{wp} \;=\; \overline{\left(\frac{1}{\rho}\right)}^{\zeta,N} \overline{\frac{\partial \zeta}{\partial z}}^{\zeta,N} \beta_2^s\, \delta_\zeta^N + g\beta_2^b \overline{\frac{1}{p}}^{\zeta,N} A_\zeta^N, \tag{91}$$

$$\mathbf{A}_{wT} \;=\; -g\beta_1^b\, \overline{\frac{p_0}{p}}^{\zeta,N} \overline{\frac{1}{T_0}}^{\zeta,N} A_\zeta^N, \tag{92}$$

for the pressure equation

$$\mathbf{A}_{pu} \;=\; +\frac{c_p}{c_v} p \frac{1}{r\cos\phi}\, \frac{1}{\sqrt{g}} \left[\beta_3^s\, \delta_\lambda \left(\overline{\sqrt{g}}^\lambda \cdot ...\right) + \beta_8^s\, \mathbf{B}_x,\right] \tag{93}$$

$$\mathbf{A}_{pv} \;=\; +\frac{c_p}{c_v} p \frac{1}{r\cos\phi}\, \frac{1}{\sqrt{g}} \left[\beta_3^s\, \delta_\phi \left(\overline{\cos\phi\,\sqrt{g}}^\phi \cdot ...\right) + \beta_8^s\, \mathbf{B}_y\right], \tag{94}$$

$$\mathbf{A}_{pw} \;=\; -\frac{c_p}{c_v} p \frac{1}{r\cos\phi}\, \frac{1}{\sqrt{g}}\, \beta_4^s\, \delta_\zeta(r\,\cos\phi\, \cdot ...) - \beta_3^b g \rho_0 A_\zeta, \tag{95}$$

and for the temperature equation

$$\mathbf{A}_{Tu} \;=\; +\frac{R}{c_v} T \frac{1}{r\cos\phi}\, \frac{1}{\sqrt{g}} \left[\beta_5^s\, \delta_\lambda \left(\overline{\sqrt{g}}^\lambda \cdot ...\right) + \beta_9^s\, \mathbf{B}_x\right], \tag{96}$$

$$\mathbf{A}_{Tv} \;=\; +\frac{R}{c_v} T \frac{1}{r\cos\phi}\, \frac{1}{\sqrt{g}} \left[\beta_5^s\, \delta_\phi \left(\overline{\cos\phi\,\sqrt{g}}^\phi \cdot ...\right) + \beta_9^s\, \mathbf{B}_y\right], \tag{97}$$

$$\mathbf{A}_{Tw} \;=\; -\frac{R}{c_v} T \frac{1}{r\cos\phi}\, \frac{1}{\sqrt{g}}\, \beta_6^s\, \delta_\zeta(r\,\cos\phi\, \cdot ...) + \beta_4^b \frac{\partial T_0}{\partial z} A_\zeta, \tag{98}$$

where operators for the calculation of $\delta_\zeta Z_{hor}$ have been introduced:

$$\mathbf{B}_x \;:=\; \delta_\zeta \left(A_\lambda \left(\frac{\partial z}{\partial \lambda}\, A_\zeta^{\bar{N}} \;\ldots\right)\right), \tag{99}$$

$$\mathbf{B}_y \;:=\; \delta_\zeta \left(A_\phi \left(\frac{\partial z}{\partial \phi}\, \cos\phi\, A_\zeta^{\bar{N}} \;\ldots\right)\right). \tag{100}$$

Using again the shallow atmosphere approximation ($r \approx r_{earth}$ in all the prefactors) we can cancel $r\cos\phi$ in $\mathbf{A}_{pw}$, $\mathbf{A}_{Tw}$, and $\mathbf{A}_{ww}$.

The right hand sides are

$$\begin{aligned}
b^n_{(u)} &= \frac{1}{\Delta t} u^n - \overline{\frac{1}{\rho}}^{\lambda} \overline{\frac{1}{r\cos\phi}}^{\lambda} \left((1-\beta^s_1)\delta_\lambda p'^n - (1-\beta^s_7) \overline{\frac{\partial z}{\partial \lambda}}^{\zeta} A_\lambda \left(\delta_z^{(s,s)} p'^n \right) \right) + \\
&\quad + \overline{\frac{1}{\rho}}^{\lambda} \overline{\frac{1}{r\cos\phi}}^{\lambda} \left(\delta_\lambda (\alpha^h_{div}\, \rho\, D^n_{(uv)}) - \overline{\frac{\partial z}{\partial \lambda}}^{\zeta} A_\lambda \left(\delta_z^{(s,s)} (\alpha^h_{div}\, \rho\, D^n_{(uv)}) \right) \right) + \left. \frac{\partial u}{\partial t} \right|_{slow}, \quad (101) \\
b^n_{(v)} &= \frac{1}{\Delta t} v^n - \overline{\frac{1}{\rho}}^{\phi} \overline{\frac{1}{r}}^{\phi} \left((1-\beta^s_1)\delta_\phi p'^n - (1-\beta^s_7) \overline{\frac{\partial z}{\partial \phi}}^{\zeta} A_\phi \left(\delta_z^{(s,s)} p'^n \right) \right) + \\
&\quad + \overline{\frac{1}{\rho}}^{\phi} \overline{\frac{1}{r}}^{\phi} \left(\delta_\phi (\alpha^h_{div}\, \rho\, D^n_{(uv)}) - \overline{\frac{\partial z}{\partial \phi}}^{\zeta} A_\phi \left(\delta_z^{(s,s)} (\alpha^h_{div}\, \rho\, D^n_{(uv)}) \right) \right) + \left. \frac{\partial v}{\partial t} \right|_{slow}, \qquad (102) \\
b^n_{(w)} &= \frac{1}{\Delta t} w^n - \overline{\frac{1}{\rho}}^{\zeta,N} (1-\beta^s_2) \overline{\frac{\partial \zeta}{\partial z}}^{\zeta,N} \delta^N_\zeta p'^n + \\
&\quad + g \left((1-\beta^b_1) \overline{\frac{p_0}{p}}^{\zeta,N} \overline{\frac{1}{T_0}}^{\zeta,N} A^N_\zeta T' - (1-\beta^b_2) \overline{\frac{1}{p}}^{\zeta,N} A^N_\zeta 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^N_\zeta (\alpha^v_{div}\, \rho\, D^n_{(w)}) + \left. \frac{\partial w}{\partial t} \right|_{slow}, \qquad (103) \\
b^n_{(p)} &= \frac{1}{\Delta t} p'^n - \frac{c_p}{c_V} p D^n_{(p)} + (1-\beta^b_3) g \rho_0\, A_\zeta w^n + \left. \frac{\partial p'}{\partial t} \right|_{slow}, \qquad (104) \\
b^n_{(T)} &= \frac{1}{\Delta t} T'^n - \frac{R}{c_V} T D^n_{(T)} - (1-\beta^b_4) \frac{\partial T_0}{\partial z} A_\zeta w^n + \left. \frac{\partial T'}{\partial t} \right|_{slow}, \qquad (105)
\end{aligned}$$

with the explicit divergence terms

$$\begin{aligned}
D^n_{(p)} &= \frac{1}{r\cos\phi} \frac{1}{\sqrt{g}} \left[(1-\beta^s_3)\, \tilde{d}^n_{hor} + (1-\beta^s_8)\, \delta_\zeta Z^n_{hor} + (1-\beta^s_4)\, \delta_\zeta Z^n_{vert} \right], \qquad (106) \\
D^n_{(T)} &= \frac{1}{r\cos\phi} \frac{1}{\sqrt{g}} \left[(1-\beta^s_5)\, \tilde{d}^n_{hor} + (1-\beta^s_9)\, \delta_\zeta Z^n_{hor} + (1-\beta^s_6)\, \delta_\zeta Z^n_{vert} \right], \qquad (107) \\
D^n_{(uv)} &= \frac{1}{r\cos\phi} \frac{1}{\sqrt{g}} \left[(1-\beta^d_1)\, \tilde{d}^n_{hor} + (1-\beta^d_5)\, \delta_\zeta Z^n_{hor} + (1-\beta^d_2)\, \delta_\zeta Z^n_{vert} \right], \qquad (108) \\
D^n_{(w)} &= \frac{1}{r\cos\phi} \frac{1}{\sqrt{g}} \left[(1-\beta^d_3)\, \tilde{d}^n_{hor} + (1-\beta^d_6)\, \delta_\zeta Z^n_{hor} + (1-\beta^d_4)\, \delta_\zeta Z^n_{vert} \right]. \qquad (109)
\end{aligned}$$

Here $\tilde{d}^n_{hor}$ is calculated by eq. (58) and Z^n_{hor} by eq. (60)-(62) (in both cases with u^n and v^n, Z^n_{vert} analogous to eq. (63) with w^n).

6.2 Solution of the tridiagonal equation system

The solution of the tridiagonal equation system is a tedious, but straightforward work. First, one only ends up with a tridiagonal equation system, if the following requirements for the implicit weightings are fulfilled:

$$\beta_1^{(s)} = 0, \quad \beta_7^{(s)} = 0, \tag{110}$$

and

$$\beta_1^{(d)} = 0, \quad \beta_2^{(d)} = 0, \quad \beta_5^{(d)} = 0, \quad \beta_7^{(d)} = 0. \tag{111}$$

This simply means, that the equations for u and v must be solved explicitly, i.e. in a pure forward sense:

$$u^{n+1} = \Delta t \, b^n_{(u)}, \tag{112}$$

$$v^{n+1} = \Delta t \, b^n_{(v)}. \tag{113}$$

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}. **?** determines optimal values of the remaining off-centering weights by a stability analysis to

$$\beta_3^{(s)} = \beta_5^{(s)} = 1, \quad \beta_2^{(s)} = \beta_4^{(s)} = \beta_6^{(s)} = 0.7, \tag{114}$$

$$\beta_1^{(b)} = \beta_2^{(b)} = \beta_3^{(b)} = \beta_4^{(b)} = 0.7, \tag{115}$$

$$\beta_3^{(d)} = \beta_4^{(d)} = 1.0. \tag{116}$$

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

$$A_{k-\frac{1}{2}} w_{k-\frac{3}{2}} + B_{k-\frac{1}{2}} w_{k-\frac{1}{2}} + C_{k-\frac{1}{2}} w_{k+\frac{1}{2}} = rhs_{k-\frac{1}{2}} \tag{117}$$

(A, B, C, rhs and w are further defined at (i,j)) with the coefficients

$$\begin{aligned}
A_{k-\frac{1}{2}} = & -a_{wp1\,|\,k-\frac{1}{2}} \cdot a_{pw1\,|\,k-1} \cdot r_c \\
& +a_{wp1\,|\,k-\frac{1}{2}} \cdot a_{pw2\,|\,k-1} \cdot \frac{1}{2} \\
& +a_{wp2\,|\,k-\frac{1}{2}} \cdot (1-g_{k-\frac{1}{2}}) \cdot a_{pw1\,|\,k-1} \cdot r_c \\
& -a_{wp2\,|\,k-\frac{1}{2}} \cdot (1-g_{k-\frac{1}{2}}) \cdot a_{pw2\,|\,k-1} \cdot \frac{1}{2} \\
& +a_{wT\,|\,k-\frac{1}{2}} \cdot (1-g_{k-\frac{1}{2}}) \cdot a_{Tw1\,|\,k-1} \cdot r_c \\
& -a_{wT\,|\,k-\frac{1}{2}} \cdot (1-g_{k-\frac{1}{2}}) \cdot a_{Tw2\,|\,k-1} \cdot \frac{1}{2} \\
& +\frac{1}{\Delta t} a_{ww1\,|\,k-\frac{1}{2}} \cdot a_{ww2\,|\,k-1} \cdot r_c
\end{aligned}$$

$$\begin{aligned}
B_{k-\frac{1}{2}} = & +a_{wp1\,|\,k-\frac{1}{2}} \cdot \left[a_{pw1\,|\,k} + a_{pw1\,|\,k-1}\right] \cdot r_c \\
& -a_{wp1\,|\,k-\frac{1}{2}} \cdot \left[a_{pw2\,|\,k} - a_{pw2\,|\,k-1}\right] \cdot \frac{1}{2} \\
& -a_{wp2\,|\,k-\frac{1}{2}} \cdot \left[-g_{k-\frac{1}{2}} \cdot a_{pw1\,|\,k} + (1-g_{k-\frac{1}{2}}) \cdot a_{pw1\,|\,k-1}\right] \cdot r_c \\
& -a_{wp2\,|\,k-\frac{1}{2}} \cdot \left[+g_{k-\frac{1}{2}} \cdot a_{pw2\,|\,k} + (1-g_{k-\frac{1}{2}}) \cdot a_{pw2\,|\,k-1}\right] \cdot \frac{1}{2} \\
& -a_{wT\,|\,k-\frac{1}{2}} \cdot \left[-g_{k-\frac{1}{2}} \cdot a_{Tw1\,|\,k} + (1-g_{k-\frac{1}{2}}) \cdot a_{Tw1\,|\,k-1}\right] \cdot r_c \\
& -a_{wT\,|\,k-\frac{1}{2}} \cdot \left[+g_{k-\frac{1}{2}} \cdot a_{Tw2\,|\,k} + (1-g_{k-\frac{1}{2}}) \cdot a_{Tw2\,|\,k-1}\right] \cdot \frac{1}{2} \\
& -\frac{1}{\Delta t} a_{ww1\,|\,k-\frac{1}{2}} \cdot \left[a_{ww2\,|\,k} + a_{ww2\,|\,k-1}\right] \cdot r_c \\
& +\frac{1}{\Delta t^2}
\end{aligned}$$

$$
\begin{aligned}
C_{k-\frac{1}{2}} \;=\; & -a_{wp1\,|\,k-\frac{1}{2}} \cdot a_{pw1\,|\,k} \cdot r_c \\
& -a_{wp1\,|\,k-\frac{1}{2}} \cdot a_{pw2\,|\,k} \cdot \frac{1}{2} \\
& -a_{wp2\,|\,k-\frac{1}{2}} \cdot g_{k-\frac{1}{2}} \cdot a_{pw1\,|\,k} \cdot r_c \\
& -a_{wp2\,|\,k-\frac{1}{2}} \cdot g_{k-\frac{1}{2}} \cdot a_{pw2\,|\,k} \cdot \frac{1}{2} \\
& -a_{wT\,|\,k-\frac{1}{2}} \cdot g_{k-\frac{1}{2}} \cdot a_{Tw1\,|\,k} \cdot r_c \\
& -a_{wT\,|\,k-\frac{1}{2}} \cdot g_{k-\frac{1}{2}} \cdot a_{Tw2\,|\,k} \cdot \frac{1}{2} \\
& +\frac{1}{\Delta t}\, a_{ww1\,|\,k-\frac{1}{2}} \cdot a_{ww2\,|\,k} \cdot r_c
\end{aligned}
$$

using the denotations

$$
a_{wp1} \;:=\; \overline{\left(\frac{1}{\rho}\right)}^{\zeta,N} \overline{\frac{\partial \zeta}{\partial z}}^{\zeta,N} \beta_2^s, \qquad a_{wp2} := g\, \beta_2^b\, \overline{\left(\frac{1}{p}\right)}^{\zeta,N}, \tag{118}
$$

$$
a_{ww1} \;:=\; \overline{\left(\frac{1}{\rho}\right)}^{\zeta,N} \overline{\frac{\partial \zeta}{\partial z}}^{\zeta,N}, \qquad a_{ww2} := \alpha_{div}^v\, \rho\, \frac{1}{r\cos\phi}\, \frac{1}{\sqrt{g}}\, \beta_4^d, \tag{119}
$$

$$
a_{wT} \;:=\; -g\, \overline{\left(\frac{p_0}{p}\right)}^{\zeta,N} \overline{\left(\frac{1}{T_0}\right)}^{\zeta,N} \beta_1^b, \tag{120}
$$

$$
a_{pw1} \;:=\; -\frac{c_p}{c_v}\, p\, \frac{1}{r\cos\phi}\, \frac{1}{\sqrt{g}}\, \beta_4^s, \qquad a_{pw2} := -\beta_3^b g \rho_0, \tag{121}
$$

$$
a_{Tw1} \;:=\; -\frac{R}{c_v}\, T\, \frac{1}{r\cos\phi}\, \frac{1}{\sqrt{g}}\, \beta_6^s, \qquad a_{Tw2} := \beta_4^b \frac{\partial T_0}{\partial z}, \tag{122}
$$

and

$$
r_c := \overline{r\,\cos\phi}^{\zeta}. \tag{123}
$$

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. (117) reads

$$
\begin{aligned}
rhs \;:=\; & -\mathbf{A}_{wp} \underbrace{\left[b_p^{(n)} - \left(\mathbf{A}_{pu} u^{n+1} + \mathbf{A}_{pv} v^{n+1}\right)\right]}_{=:b_{4,p}} \\
& -\mathbf{A}_{wT} \left[b_T^{(n)} - \left(\mathbf{A}_{Tu} u^{n+1} + \mathbf{A}_{Tv} v^{n+1}\right)\right] \\
& +\frac{1}{\Delta t} \left[b_w^{(n)} - \left(\mathbf{A}_{wu} u^{n+1} + \mathbf{A}_{wv} v^{n+1}\right)\right].
\end{aligned} \tag{124}
$$

We express some of the needed terms more explicitly by

$$
\mathbf{A}_{pu} u^{n+1} + \mathbf{A}_{pv} v^{n+1} \;=\; \frac{c_p}{c_v}\, p\, \frac{1}{r\cos\phi}\, \frac{1}{\sqrt{g}} \left[\beta_3^s\, \tilde{d}_{hor}^{n+1} + \beta_8^s\, \delta_\zeta Z_{hor}^{n+1}\right], \tag{125}
$$

$$
\mathbf{A}_{Tu} u^{n+1} + \mathbf{A}_{Tv} v^{n+1} \;=\; \frac{R}{c_v}\, T\, \frac{1}{r\cos\phi}\, \frac{1}{\sqrt{g}} \left[\beta_5^s\, \tilde{d}_{hor}^{n+1} + \beta_9^s\, \delta_\zeta Z_{hor}^{n+1}\right], \tag{126}
$$

$$
\mathbf{A}_{wu} u^{n+1} + \mathbf{A}_{wv} v^{n+1} \;=\; -\overline{\left(\frac{1}{\rho}\right)}^{\zeta,N} \overline{\frac{\partial \zeta}{\partial z}}^{\zeta,N} \delta_\zeta^N \left[\frac{\alpha_{div}^v\, \rho}{r\cos\phi}\, \frac{1}{\sqrt{g}} \left(\beta_3^d\, \tilde{d}_{hor}^{n+1} + \beta_6^d\, \delta_\zeta Z_{hor}^{n+1}\right)\right] \tag{127}
$$

where $\tilde{d}_{hor}^{n+1}$ is calculated via (58) and Z_{hor}^{n+1} via (60) (both with u^{n+1} and v^{n+1}).

The explicit solution for p'^{n+1} reads

$$p'^{n+1} = \Delta t \left(b_{4,p} - \mathbf{A}_{pw} w^{n+1} \right) \tag{128}$$

(this can be calculated without any further boundary conditions/treatments).

The explicit equation for T'^{n+1} directly follows from eq. (55) using p'^{n+1} from eq. (128).

7 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 $\alpha_{div}\Delta t/\Delta x^2 \leq 1/2$ (**?**), there is an additional constraint in tilted terrain.

To derive this constraint, one can start from the following 2D equations for the quasi-2D-divergence damping alone in terrain following coordinates

$$\frac{\partial u}{\partial t} = \alpha_{div}^h \left(\frac{\partial D}{\partial x} + \frac{\partial \zeta}{\partial x}\frac{\partial D}{\partial \zeta} \right), \tag{129}$$

$$\frac{\partial v}{\partial t} = \alpha_{div}^h \left(\frac{\partial D}{\partial y} + \frac{\partial \zeta}{\partial y}\frac{\partial D}{\partial \zeta} \right), \tag{130}$$

$$\frac{\partial w}{\partial t} = 0, \tag{131}$$

$$D = \frac{\partial u}{\partial x} + \frac{\partial \zeta}{\partial x}\frac{\partial u}{\partial \zeta} + \frac{\partial v}{\partial y} + \frac{\partial \zeta}{\partial y}\frac{\partial v}{\partial \zeta} + \frac{\partial \zeta}{\partial z}\frac{\partial w}{\partial \zeta}. \tag{132}$$

For an explicit discretization of these equations analogous to (51), a von-Neumann stability analysis is relatively straightforward and results in the sufficient condition

$$\alpha_{div}^h \, \Delta t \left\{ \frac{1}{\Delta x^2} \left(2 + \frac{\Delta x}{\Delta \zeta} \left| \frac{\partial \zeta}{\partial x} \right| \right)^2 + \frac{1}{\Delta y^2} \left(2 + \frac{\Delta y}{\Delta \zeta} \left| \frac{\partial \zeta}{\partial y} \right| \right)^2 \right\} \leq 2. \tag{133}$$

This can be reformulated by prescribing $\frac{\partial z}{\partial x}$ (at constant ζ). Due to the relations similar to (7) and (8) and the estimation $\Delta\zeta \approx \frac{\partial \zeta}{\partial z} \, \Delta z$, this can be written as

$$\alpha_{div}^h \, \Delta t \left\{ \frac{1}{\Delta x^2} \left(2 + \frac{\Delta x}{\Delta z} \left| \frac{\partial z}{\partial x} \right| \right)^2 + \frac{1}{\Delta y^2} \left(2 + \frac{\Delta y}{\Delta z} \left| \frac{\partial z}{\partial y} \right| \right)^2 \right\} \leq 2. \tag{134}$$

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

$$\frac{\Delta h}{\Delta z} \equiv \frac{\Delta x}{\Delta z}\frac{\partial z}{\partial x} \tag{135}$$

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 **?** or even as the recommendation of 0.3 by **?**. 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 (134) 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 (134). 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 (134). Consequently, the divergence damping coefficient never exceeds the value given by `xkd`.

8 Discretization errors in a stretched grid

8.1 General analysis procedure

Now, in a certain sense we go back to section 4.1 and answer the question, why the vertical weighting is advantageous and why the chosen operators are to favor compared to other choices. To this purpose, a truncation error analysis is carried out. But first, we have to consider the following question: how can we determine the convergence order or the truncation error for discretizations on a *stretched* grid?

Hence, we consider a (vertical) interval $[0, L]$ which is discretized by $N + 1$ grid points

$$z^{(h)}_{\frac{1}{2}} = L, \quad z^{(h)}_{1+\frac{1}{2}}, \quad \dots, \quad z^{(h)}_{N-\frac{1}{2}}, \quad z^{(h)}_{N+\frac{1}{2}} = 0. \tag{136}$$

In COSMO the position of the half levels is prescribed, therefore we write half indices and add an upper index (h). The grid mesh size in a grid box is $\Delta z_k := z^{(h)}_{k-\frac{1}{2}} - z^{(h)}_{k+\frac{1}{2}}$, $k = 1, .., N$.

In contrast to an equidistant grid there exist arbitrary many different possibilities to perform a grid refinement. In the following, two variants of grid refinements are analyzed, that lead to different orders of the truncation error.

8.1.1 Discretization error analysis - variant A

In this first variant, the grid stretching is described by a *coordinate transformation* $z(\zeta)$. The transformed coordinate ζ generates an equidistant grid:

$$\zeta_{k+\frac{1}{2}} = \left(k + \frac{1}{2} \right) \cdot \Delta\zeta, \quad k = 0, \dots, N. \tag{137}$$

With increasing grid refinement $\Delta\zeta \to 0$ all the grid mesh sizes $\Delta z_k \approx \left. \frac{\partial z}{\partial \zeta} \right|_{k+\frac{1}{2}} \cdot \Delta\zeta$ converge to 0, too. Hence, the convergence inspection takes place for $\Delta\zeta \to 0$.

With decreasing $\Delta\zeta$ the stretching function *locally* becomes increasingly linear. Consequently, with this approach to refine the grid one gets formally 2nd order truncation errors for all two-point discretizations.

In the following analysis one has to distinguish carefully between main and half levels; otherwise a restriction to 2-dimensional fields in (λ, z) is sufficient. The grid is horizontally defined by

$$\lambda_i = i \cdot \Delta\lambda \tag{138}$$

with $i = 0, 1, 2, \ldots$ (for scalar- or w-positions) or $i = 1/2, 3/2, 5/2, \ldots$ (for u-position), and vertically by the z-coordinates of the half levels

$$z^{(h)}_{i,k+\frac{1}{2}} = z\left(\lambda_i, \zeta_{k+\frac{1}{2}}\right) \tag{139}$$

(with the above mentioned stretching function) and the main levels

$$z^{(m)}_{i,k} = \frac{z^{(h)}_{i,k+\frac{1}{2}} + z^{(h)}_{i,k-\frac{1}{2}}}{2}. \tag{140}$$

Hence, the w-positions are in $(\lambda_i, z^{(h)}_{i,k+\frac{1}{2}})$, the scalars T, p are in $(\lambda_i, z^{(m)}_{i,k})$, and so on. In the following truncation error analysis these coordinates have to be inserted into the discretizations according to the grid positions of the variables. Then a Taylor expansion for $\Delta\lambda \to 0$ and $\Delta\zeta \to 0$ *around the position of the target point* (which itself can be a function of $\Delta\lambda$ and $\Delta\zeta$, too) must be carried out.

8.1.2 Discretization error analysis - variant B

To avoid the circumstance, that for increasing resolution $\Delta z_k \to 0$ the grid stretching becomes *locally* more and more linear, one can prescribe a constant grid stretching

$$\frac{\Delta z_k}{\Delta z_{k-1}} = s$$

instead. Without any loss of generality we can assume that $s > 1$, i.e. the grid becomes finer in the vicinity of $z^{(h)}_0 = 0$.

To this purpose we choose the grid points in

$$\begin{aligned}
z^{(h)}_{i,-1-\frac{1}{2}} &= h(\lambda_i) - \frac{1}{2}\Delta\zeta - \frac{1}{s}\Delta\zeta, \\
z^{(h)}_{i,-\frac{1}{2}} &= h(\lambda_i) - \frac{1}{2}\Delta\zeta, \\
z^{(h)}_{i,\frac{1}{2}} &= h(\lambda_i) + \frac{1}{2}\Delta\zeta, \\
z^{(h)}_{i,1+\frac{1}{2}} &= h(\lambda_i) + \frac{1}{2}\Delta\zeta + s\Delta\zeta,
\end{aligned} \tag{141}$$

Here the prescription of a slope by $h(x)$ is possible. The flattening of coordinate surfaces with increasing height is neglected. The position of the main levels again is defined by Eq. (140). The Taylor expansion is performed in analogy to variant A.

One has to note, that such a grid refinement does not converge *globally*! For the grid mesh size it follows by the geometric summation formula

$$\Delta z_k = \Delta z_1 \cdot s^{k-1} = \frac{L}{\sum_{l=0}^{N-1} s^l} \cdot s^{k-1} = \frac{L}{\frac{s^N-1}{s-1}} \cdot s^{k-1}. \tag{142}$$

In particular for the coarsest grid mesh size it follows

$$\Delta z_N = L\frac{s-1}{s-s^{1-N}} \overset{N\to\infty}{\to} L\frac{s-1}{s} \neq 0!$$

By this kind of grid refinement one inserts smaller and smaller grid boxes only on one end of the interval (at $z = 0$), without a significant refinement at the other end (at $z = L$) for the coarse grid boxes. This, of cause, is a bit odd situation for this variant of grid refinement.

8.2 Horizontal pressure gradient term in the u-equation

We inspect the discretization of the horizontal pressure gradient

$$\frac{\partial p(\lambda, z)}{\partial \lambda} = \frac{\partial p(\lambda, \zeta)}{\partial \lambda} + \frac{\partial \zeta}{\partial \lambda}\frac{\partial p}{\partial \zeta} = \frac{\partial p(\lambda, \zeta)}{\partial \lambda} - \frac{\partial z}{\partial \lambda}\frac{\partial p}{\partial z}.$$

Discretization error analysis - variant A

1. The discretization with weighted vertical interpolation delivers

$$\begin{aligned} \delta_\lambda p - \left(A_\zeta \frac{\partial z}{\partial \lambda}\right)\left(A_\lambda\left(\frac{\partial \zeta}{\partial z}\delta_\zeta A_\zeta^N p\right)\right) &= \left.\frac{\partial p}{\partial \lambda}\right|_z + \\ &+ d\zeta^2\left[-\frac{1}{6}\frac{\partial z}{\partial \lambda}\left(\frac{\partial z}{\partial \zeta}\right)^2\frac{\partial^3 p}{\partial z^3} - \frac{1}{4}\frac{\partial z}{\partial \lambda}\frac{\partial^2 z}{\partial \zeta^2}\frac{\partial^2 p}{\partial z^2}\right] + \\ &+ d\lambda^2\,[\ldots] + d\zeta\, d\lambda \cdot 0 + O(d\zeta^4, d\zeta^2 d\lambda^2, d\lambda^4). \end{aligned} \tag{143}$$

The discretization operators are defined in section 4.1. For the 2nd order term $\sim \Delta\zeta^2$ one has either a contribution from the curvature of $p(z)$ (first term) or a stretching (=curvature) in the grid $z(\zeta)$ (second term).

2. The same discretization, but without weighted averages delivers

$$\begin{aligned} \delta_\lambda p - \left(A_\zeta \frac{\partial z}{\partial \lambda}\right)\left(A_\lambda\left(\frac{\partial \zeta}{\partial z}\delta_\zeta A_\zeta p\right)\right) &= \left.\frac{\partial p}{\partial \lambda}\right|_z + \\ &+ d\zeta^2\left[-\frac{1}{6}\frac{\partial z}{\partial \lambda}\left(\frac{\partial z}{\partial \zeta}\right)^2\frac{\partial^3 p}{\partial z^3} - \frac{1}{2}\frac{\partial z}{\partial \lambda}\frac{\partial^2 z}{\partial \zeta^2}\frac{\partial^2 p}{\partial z^2}\right] + \\ &+ d\lambda^2\,[\ldots] + d\zeta\, d\lambda \cdot 0 + O(d\zeta^4, d\zeta^2 d\lambda^2, d\lambda^4). \end{aligned} \tag{144}$$

In the term $\sim \Delta\zeta^2$ the second term has a twice as large prefactor compared to the discretization above. The prefactor of the term $\sim d\lambda^2$ is the same as above, as expected.

3. One can think about a lot of other discretizations for the horizontal pressure gradient, for example with

$$\begin{aligned} \delta_\lambda p - \left(A_\lambda \frac{\partial \zeta}{\partial z}\right)\left(A_\zeta\left(\frac{\partial z}{\partial \lambda}A_\lambda \delta_\zeta p\right)\right) &= \left.\frac{\partial p}{\partial \lambda}\right|_z + \\ &+ d\zeta^2\left[-\frac{1}{6}\frac{\partial z}{\partial \lambda}\left(\frac{\partial z}{\partial \zeta}\right)^2\frac{\partial^3 p}{\partial z^3} - \frac{1}{2}\frac{\partial z}{\partial \lambda}\frac{\partial^2 z}{\partial \zeta^2}\frac{\partial^2 p}{\partial z^2} - \frac{1}{4}\frac{\partial z}{\partial \zeta}\frac{\partial^2 z}{\partial \lambda \partial \zeta}\frac{\partial^2 p}{\partial z^2}\right. \\ &\left. -\frac{1}{4}\frac{\frac{\partial^2 z}{\partial \lambda \partial \zeta}\frac{\partial^2 z}{\partial \zeta^2}}{\frac{\partial z}{\partial \zeta}}\frac{\partial p}{\partial z} - \frac{1}{4}\frac{\frac{\partial^3 z}{\partial \zeta^3}\frac{\partial z}{\partial \lambda}}{\frac{\partial z}{\partial \zeta}}\frac{\partial p}{\partial z}\right] + \\ &+ d\lambda^2\,[\ldots] + d\zeta\, d\lambda \cdot 0 + O(d\zeta^4, d\zeta^2 d\lambda^2, d\lambda^4). \end{aligned} \tag{145}$$

Obviously this is discretization generates larger errors.

4. The Mahrer (1984)-discretization can only be inspected for gentle slopes. In this case it follows

$$\begin{aligned} \frac{\partial p}{\partial \lambda} &= \left.\frac{\partial p}{\partial \lambda}\right|_z + d\zeta^2 \left[-\frac{1}{24} \frac{\partial z}{\partial \lambda} \left(\frac{\partial z}{\partial \zeta} \right)^2 \frac{\partial^3 p}{\partial z^3} \right] + \\ &+ \; d\zeta \, d\lambda \left[\frac{1}{16} \frac{\partial^2 z}{\partial \lambda^2} \frac{\partial z}{\partial \zeta} \frac{\partial^2 p}{\partial z^2} + \ldots \right] + d\lambda^2 [\ldots] + ... \end{aligned} \tag{146}$$

Though the term $\sim d\zeta^2$ is much smaller than in the above discretizations, another second order term arises which is proportional to $d\zeta \, d\lambda$. This could explain the slightly worse discretization properties in the vertical.

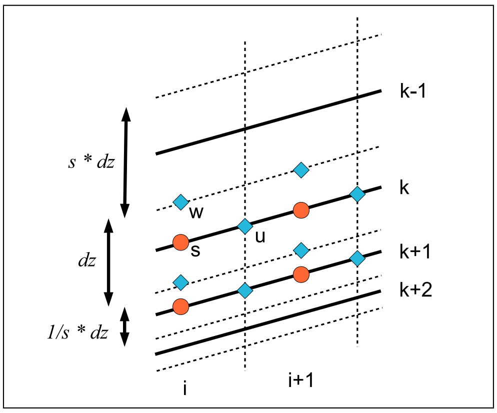

Figure 2: Grid positions and stretching in the 'discretization error analysis - variant B' for the horizontal pressure gradient $\partial p / \partial \lambda$.

Discretization error analysis - variant B

1. The discretization with weighted vertical interpolation delivers

$$\begin{aligned} \delta_\lambda p - \left(A_\zeta \frac{\partial z}{\partial \lambda} \right) \left(A_\lambda \left(\frac{\partial \zeta}{\partial z} \delta_\zeta A_\zeta^N p \right) \right) &= \left.\frac{\partial p}{\partial \lambda}\right|_z + \\ &+ \; dz \left[\frac{1}{8} \frac{\partial h}{\partial \lambda} \left(\frac{1}{s} - s \right) \frac{\partial^2 p}{\partial z^2} \right] + \\ &+ \; d\lambda^2 [...] + dz^2 [..] + O(dz^3, d\lambda^2 \, dz, ...) \end{aligned} \tag{147}$$

Now the leading truncation error is indeed of *first order* in dz.

2. The same discretization but without weighted averaging

$$\begin{aligned} \delta_\lambda p - \left(A_\zeta \frac{\partial z}{\partial \lambda} \right) \left(A_\lambda \left(\frac{\partial \zeta}{\partial z} \delta_\zeta A_\zeta p \right) \right) &= \left.\frac{\partial p}{\partial \lambda}\right|_z + \\ &+ \; dz \left[\frac{1}{4} \frac{\partial h}{\partial \lambda} \left(\frac{1}{s} - s \right) \frac{\partial^2 p}{\partial z^2} \right] + \\ &+ \; d\lambda^2 [...] + dz^2 [..] + ... \end{aligned} \tag{148}$$

Again the truncation error structure is the same as above but now the prefactor is twice as large.

3. The inappropriate discretization from above delivers

$$\begin{aligned}\delta_\lambda p - \left(A_\lambda \frac{\partial \zeta}{\partial z}\right)\left(A_\zeta \left(\frac{\partial z}{\partial \lambda} A_\lambda \delta_\zeta p\right)\right) = \left.\frac{\partial p}{\partial \lambda}\right|_z + \frac{\partial p}{\partial \zeta}\frac{\partial h}{\partial \lambda}\left(\frac{1}{2} - \frac{1}{4}\left(\frac{1}{s} + s\right)\right) + \\ + dz\,[...] + d\lambda^2\,[...] + dz^2\,[..] + ...\end{aligned} \tag{149}$$

   For $s \neq 1$ this don't even is a consistent discretization!

4. For the Mahrer (1984) discretization one gets:

$$\begin{aligned}\frac{\partial p}{\partial \lambda} &= \left.\frac{\partial p}{\partial \lambda}\right|_z + \\ &+ dz^2 \left[-\frac{1}{24}\frac{\partial h}{\partial \lambda}\frac{\partial^3 p}{\partial z^3}\right] + \\ &+ d\lambda dz\,[...] + d\lambda^2\,[...] + dz^2\,[..] + ...\end{aligned} \tag{150}$$

   The interesting result is, that no terms $\sim dz$ or $\sim dz\, d\lambda$ occur. So it is indeed a second order discretization!

Summarizing the results, it is clear, that the discretization with vertical weightings (option 1) is the best one. Further one can see, that discretizations with slightly wider stencils (option 3) in general lead to worse truncation errors. For the Mahrer discretization, the decision is a bit more difficult. Whereas in analysis variant A there occurs a mixed second order term, which is probably detrimental, in analysis variant B this is the only discretization which is really of second order.

8.3 Buoyancy term in the w-equation

Terms of the form T'/T_0 (or $p'/p0$) occuring in the buoyancy term of the w-equation.

Discretization error analysis - variant A

1. The discretization in the new fast waves solver uses weighted vertical interpolations for T' and calculates T_0 exactly at the w-position.

$$\frac{1}{T_0} A_\zeta^N T' = \frac{T'}{T_0} + d\zeta^2 \, \frac{1}{T_0}\left[\frac{1}{8}\left(\frac{\partial z}{\partial \zeta}\right)^2 \frac{\partial^2 T'}{\partial z^2}\right] + O(d\zeta^4). \tag{151}$$

2. use no weighting for the vertical averaging and calculate T_0 exactly at the w-position

$$\frac{1}{T_0} A_\zeta T' = \frac{T'}{T_0} + d\zeta^2 \, \frac{1}{T_0}\left[\frac{1}{8}\left(\frac{\partial z}{\partial \zeta}\right)^2 \frac{\partial^2 T'}{\partial z^2} + \frac{1}{4}\frac{\partial^2 z}{\partial \zeta^2}\frac{\partial T'}{\partial z}\right] + O(d\zeta^4). \tag{152}$$

   Compared to the above truncation error an additional term occurs, which stems from the curvature of the coordinate transformation. Therefore the weighted average is the better discretization.

3. use weighted averages both for T_0 and T':

$$\frac{1}{A_\zeta^N T_0} A_\zeta^N T' = \frac{T'}{T_0} + d\zeta^2 \, \frac{1}{T_0}\left[\frac{1}{8}\left(\frac{\partial z}{\partial \zeta}\right)^2 \frac{\partial^2 T'}{\partial z^2} - \frac{1}{8}\frac{1}{T_0}\frac{\partial^2 T_0}{\partial z^2}\left(\frac{\partial z}{\partial \zeta}\right)^2 T'\right] + O(d\zeta^4) \tag{153}$$

As expected, an additional term occurs when the reference state temperature exhibits a curvature. Since no statement about the sign of this term can be made, it is in general better to have no such term at all, i.e. the direct calculation of T_0 at this position is the better solution here.

4. Previous version without weighted averages

$$A_\zeta^N \left(\frac{T'}{T_0} \right) \quad = \quad \frac{T'}{T_0} + d\zeta^2 \, \frac{1}{T_0} \left[\frac{1}{8} \left(\frac{\partial z}{\partial \zeta} \right)^2 \frac{\partial^2 T'}{\partial z^2} + \dots \right] + O(d\zeta^4). \tag{154}$$

There arise several additional terms compared to the first version above.

Discretization error analysis - variant B Instead of (141) we use the following z-positions of the grid

$$\begin{aligned} z^{(h)}_{i,-1-\frac{1}{2}} &:= h(\lambda) - 2\frac{1}{s+1}dz, \\ z^{(h)}_{i,-\frac{1}{2}} &:= h(\lambda), \\ z^{(h)}_{i,+\frac{1}{2}} &:= h(\lambda) + 2\frac{s}{s+1}dz, \end{aligned} \tag{155}$$

with the (dimensionless) stretching factor s; this simply leads to more symmetric formula in s if applied to the w equation terms.

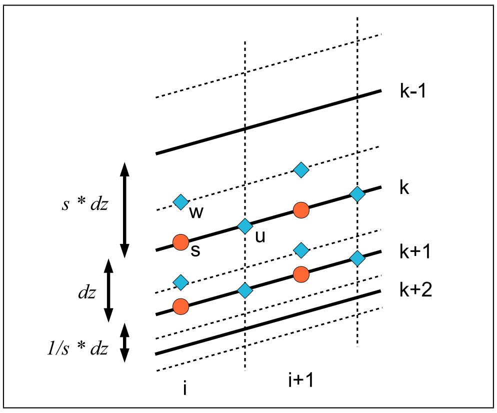

Figure 3: Grid positions and stretching in the 'discretization error analysis - variant B' for the buoyancy term in the w-equation.

1. use weighted vertical interpolations for T' and calculates T_0 exactly at the w-position

$$\frac{1}{T_0} A_\zeta^N T' \quad = \quad \frac{T'}{T_0} + dz \cdot 0 + dz^2 \left[\frac{1}{2} \frac{s}{(s+1)^2} \frac{1}{T_0} \frac{\partial^2 T'}{\partial z^2} \right] + O(dz^3). \tag{156}$$

Through the weighting this discretization is indeed of 2nd order in dz.

2. use no weighting for the vertical averaging and calculate T_0 exactly at the w-position

$$\frac{1}{T_0} A_\zeta T' \quad = \quad \frac{T'}{T_0} + dz \left[\frac{1}{2} \frac{s-1}{s+1} \frac{1}{T_0} \frac{\partial T'}{\partial z} \right] + dz^2 \left[\frac{1}{4} \frac{s^2+1}{(s+1)^2} \frac{1}{T_0} \frac{\partial^2 T'}{\partial z^2} \right] + O(dz^3) \tag{157}$$

This is only of first order for all $s \neq 1$..

3. use weighted averages both for T_0 and T'

$$\frac{1}{A_\zeta^N T_0} A_\zeta^N T' \;=\; \frac{T'}{T_0} + dz^2\,\frac{1}{2}\frac{s}{(s+1)^2}\left[\frac{1}{T_0}\frac{\partial^2 T'}{\partial z^2} - \frac{1}{T_0^2}\frac{\partial^2 T_0}{\partial z^2}T'\right] + O(dz^3). \tag{158}$$

   There arise additional terms.

4. Old version:

$$A_\zeta^N\left(\frac{T'}{T_0}\right) \;=\; \frac{T'}{T_0} + dz^2\,\frac{1}{T_0}\left[\frac{1}{2}\frac{\partial^2 T'}{\partial z^2}\frac{s}{(s+1)^2} + \ldots\right] + O(dz^3). \tag{159}$$

   There arise additional terms.

As in the section before, both analysis variants qualitatively deliver the same picture: the use of vertical weightings delivers discretizations with the smallest errors. In particular for variant B, there is a difference between a true second order discretization (option 1 with weightings) and an only first order discretization (option 2, without weightings). Further, the use of the exact values of the reference atmosphere is favorable.

8.4 The divergence term

$$\operatorname{div}\mathbf{v} = \frac{1}{g}\left[\frac{\partial g u(x,\zeta)}{\partial x} + \frac{\partial}{\partial\zeta}\left(\frac{\partial z}{\partial x}u - w\right)\right] \tag{160}$$

Discretization error analysis - variant A

1. $\frac{\partial Z_x}{\partial\zeta}$ is discretized by a weighted vertical average

$$\begin{aligned}\operatorname{div}\mathbf{v} \;=\; & \frac{\partial u(x,z)}{\partial x} + \frac{\partial w(x,z)}{\partial z} + d\zeta^2\left(-\frac{1}{4}\frac{\partial^2 z}{\partial x\partial\zeta}\frac{\partial z}{\partial\zeta}\frac{\partial^2 u}{\partial z^2} - \frac{1}{4}\frac{\partial^2 z}{\partial\zeta^2}\frac{\partial z}{\partial x}\frac{\partial^2 u}{\partial z^2}\right.\\ & \left. -\frac{1}{6}\left(\frac{\partial z}{\partial\zeta}\right)^2\frac{\partial z}{\partial x}\frac{\partial^3 u}{\partial z^3} + \frac{1}{24}\left(\frac{\partial z}{\partial\zeta}\right)^2\frac{\partial^3 w}{\partial z^3}\right) + dx^2\,() + \ldots\end{aligned} \tag{161}$$

2. $\frac{\partial Z_x}{\partial\zeta}$ is averaged without weights:

$$\begin{aligned}\operatorname{div}\mathbf{v} \;=\; & \frac{\partial u(x,z)}{\partial x} + \frac{\partial w(x,z)}{\partial z} + d\zeta^2\left(-\frac{1}{4}\frac{\partial^2 z}{\partial x\partial\zeta}\frac{\partial z}{\partial\zeta}\frac{\partial^2 u}{\partial z^2} - \frac{1}{2}\frac{\partial^2 z}{\partial\zeta^2}\frac{\partial z}{\partial x}\frac{\partial^2 u}{\partial z^2}\right.\\ & -\frac{1}{6}\left(\frac{\partial z}{\partial\zeta}\right)^2\frac{\partial z}{\partial x}\frac{\partial^3 u}{\partial z^3} + \frac{1}{24}\left(\frac{\partial z}{\partial\zeta}\right)^2\frac{\partial^3 w}{\partial z^3}\\ & \left. -\frac{1}{4}\frac{\partial\zeta}{\partial z}\frac{\partial^2 z}{\partial x\partial\zeta}\frac{\partial^2 z}{\partial\zeta^2}\frac{\partial u}{\partial z} - \frac{1}{4}\frac{\partial\zeta}{\partial z}\frac{\partial^3 z}{\partial\zeta^3}\frac{\partial z}{\partial x}\frac{\partial u}{\partial z}\right) + dx^2\,() + \ldots\end{aligned} \tag{162}$$

   One of the terms has a twice as large prefactor and two additional terms occur.

Discretization error analysis - variant B

1. $\frac{\partial Z_x}{\partial\zeta}$ is discretized by a weighted vertical average

$$\operatorname{div}\mathbf{v} \;=\; \frac{\partial u(x,z)}{\partial x} + \frac{\partial w(x,z)}{\partial z} + dz\left(\frac{1}{8}\frac{\partial h}{\partial x}\frac{\partial^2 u}{\partial\zeta^2}\left(\frac{1}{s} - s\right)\right) + O(dz^2, dx^2) \tag{163}$$

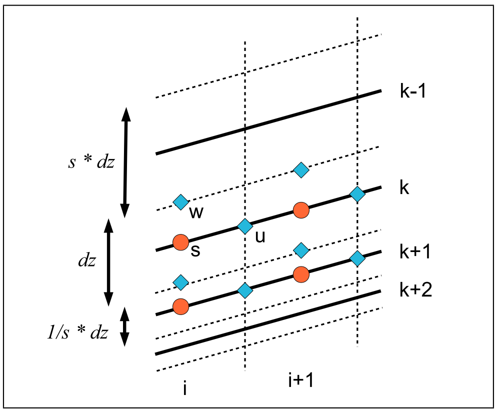

Figure 4: Grid positions and stretching in the 'discretization error analysis - variant B' for the divergence term in the scalar p'- or T'-equations.

2. $\frac{\partial Z_x}{\partial \zeta}$ is discretized without weightings

$$\begin{aligned} \operatorname{div} \mathbf{v} &= \frac{\partial u(x,z)}{\partial x} + \frac{\partial h}{\partial x}\frac{\partial u(x,z)}{\partial z}\left(\frac{1}{2} - \frac{1}{4}\left(s + \frac{1}{s}\right)\right) + \frac{\partial w(x,z)}{\partial z} + \\ &+ \; dz\left(\frac{1}{8}\frac{\partial h}{\partial x}\frac{\partial^2 u}{\partial \zeta^2}\left(\left(\frac{1}{s} - s\right) + \frac{1}{2}\left(\frac{1}{s^2} - s^2\right)\right)\right) + O(dz^2, dx^2) \qquad (164) \end{aligned}$$

For a stretched grid ($s \neq 1$) this is not a consistent discretization at all!

9 Idealized tests

In this section, the properties of the new fast waves solver are inspected using several idealized test setups. All the following simulations are performed with COSMO version 4.26.2 (concerning the dynamics this is similar to the more official version 4.27). If not otherwise stated, the switches for the new fast waves solver are `itype_fast_waves=2`, `ldyn_bbc=.FALSE.`, and `itype_bbc_w=114`. These settings will be denoted as 'FW2' in the following. The namelist values for the simulations with the old fast waves solver are `1`, `.TRUE.`, and `2`, respectively. The old fast waves solver settings are denoted by 'FW1'.

9.1 Linear gravity and sound wave expansion

In **?** (in the following abbreviated by BB13) an exact analytic solution for the expansion of linear gravity and sound waves for the compressible, non-hydrostatic Euler equations in an isothermal atmosphere was derived. This test case inspects almost all terms of the fast waves solver (with the exception of the 'horizontal' metric terms) together with the time integration scheme and the coupling with the advection process.

In the following simulations, the same setup as proposed for the small-scale test in BB13 is used. A weak warm bubble is set into a 10 km high and 300 km wide channel with periodic boundary conditions in the horizontal. The only difference is the definition of the grid. To demonstrate the benefit of the vertical weightings in the new fast waves solver a vertical grid stretching is introduced, with a grid stretching ratio of 1:10 between the finest vertical mesh size around $z = 5$ km and the coarsest mesh size around $z = 0$ and $z = 10$ km. Figures 5 show the grid for the first three chosen (horizontal) resolutions $\Delta x = 1000$ m, 500 m, and 250 m with 300 × 20, 600 × 40, and 1200 × 80 grid points, respectively, and the initial temperature perturbation T' of the weak warm bubble at $t = 0$. The time steps for these simulations are 20 s, 10 s, and 5 s, respectively, and analogous for the finer resolutions. During the expansion of the waves a background velocity field with $u_0 = 20$ m/s advects the waves to the right. Figure 6 shows the solution for T' and w of FW2 after 30 min. together with the analytic solution.

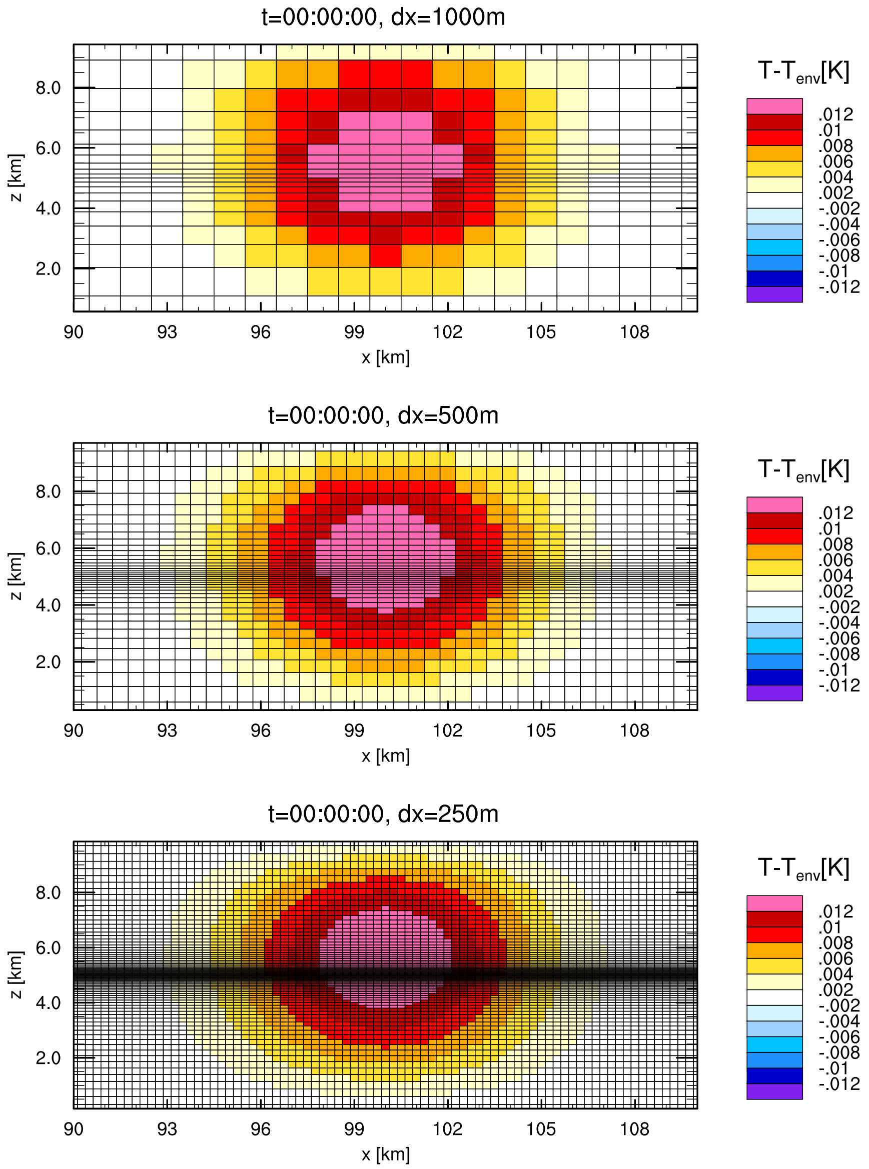

Figure 5: Grid and initial temperature perturbation $T'(t = 0)$ for the first three resolutions $\Delta x = 1000$ m, 500 m, and 250 m for the linear gravity/sound wave test.

The following results are obtained by COSMO 4.26.2 (with the additional implementation of the grid stretching in `src_artifdata.f90`). Figures 7 and 8 show the error norms of the simulated solution against the analytic solution after 30 min. The error norms

$$L_m(\psi) = \left(\frac{1}{N_x N_z} \sum_{i=1}^{N_x} \sum_{k=1}^{N_z} |\psi(x_i, z_k) - \psi_{\text{lin}}(x_i, z_k)|^m \right)^{1/m} , \quad m = 1, 2, \infty \qquad (165)$$

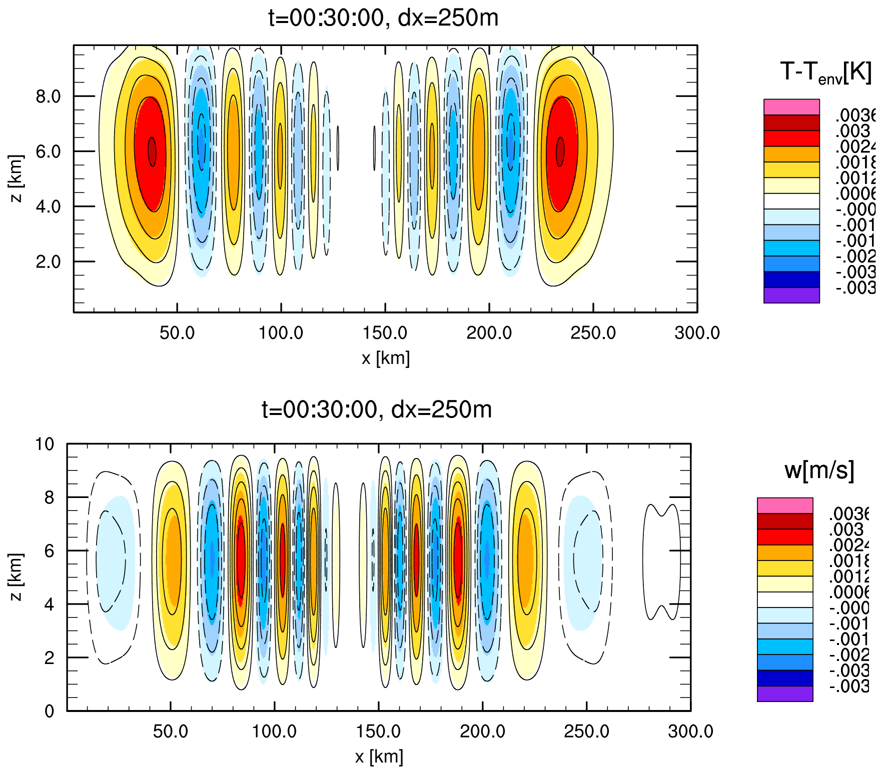

Figure 6: Temperature perturbation T' and vertical velocity w after 30 min. for $\Delta x = 250$ m horizontal resolution. Comparison between FW2-simulation (shaded) and analytic solution (lines).

for any field $\psi(x, z, t)$ are calculated as described in BB13 for the COSMO model; in particular the L_∞ norm is the usual maximum norm. In all cases, the errors of the new fast waves solver (FW2) are smaller compared to those of the old one (FW1). For the coarsest resolution the error of FW1 is nearly twice as large compared to FW2. One should notice that the more gentle slope for FW2 does not mean a smaller convergence rate. As one can see from the curvature of the lines, the simulation for the coarser resolutions is not yet in the convergence range. In contrast, these error norms show that the errors for not completely resolved structures are better with the vertical weightings in the new FW2. For very fine resolutions, at least for the L_∞-error of w, the error norms of FW1 and FW2 are nearly the same and result in a convergence rate of about 0.7 for T' and slightly higher for w. The reason for this behavior is, that for an increasing number of vertical grid points, the local grid stretching becomes increasingly linear. This can be seen in figures 5, too. Consequently, the importance of the weightings in the averages decreases.

9.2 2D linear flow over a series of hills

To test the proper inclusion of metric terms, 2-dimensional flow over a couple of hills is inspected. The test setup is chosen as in (**?**, section 5.b), but with a reduced mountain height of $h_{max} = 1$ m. This test combines hydrostatic and non-hydrostatic features of flow over mountains. An analytic solution for linear flow over hills for exactly the compressible, non-hydrostatic Euler equation system used in the COSMO model and in particular for this test setup is given in **?**.

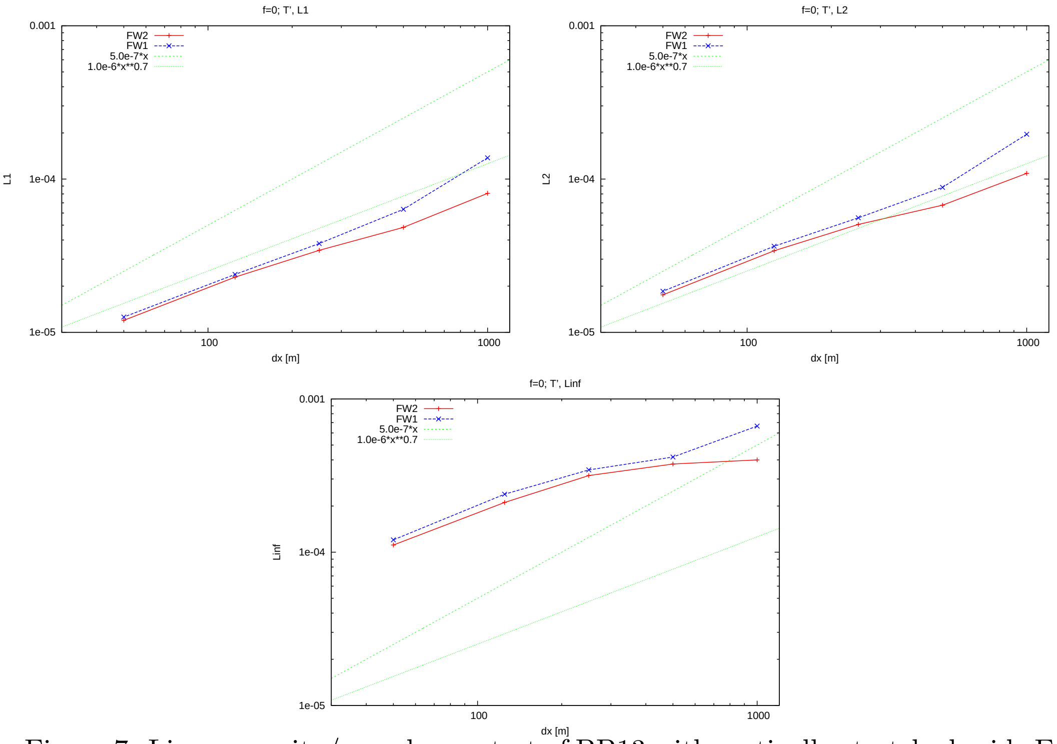

Figure 7: Linear gravity/sound wave test of BB13 with vertically stretched grid. Error norms of T' for the new fast waves solver (FW2, red) and the old one (FW1, blue) for different resolutions.

The vertical grid is stretched by a quadratic stretching function, i.e

$$z_k^{(h)} = H_{top} \cdot (\beta \eta^2 + (1-\beta)\eta), \qquad \eta = 1 - \frac{k-1}{ke}, \qquad k = 1, 2, \ldots, ke+1. \tag{166}$$

$(1+\beta)/(1-\beta)$ is roughly the ratio $\Delta z_{top}/\Delta z_{bottom}$. We choose $H_{top} = 25$ km and $\beta = 0.95$ and the following two resolutions:
1.) $\Delta x = 1000$ m and $ke = 50$ vertical levels,
2.) $\Delta x = 500$ m and $ke = 100$ vertical levels.
The tables 1 give an impression about the resulting vertical levels. All physical parameterizations, in particular the turbulence scheme, are switched off. Otherwise boundary layer effects would prevent from a correct comparison with the analytic solution.

The comparison for these two resolutions with the analytic solution is quite good (see figure 9). Both the hydrostatic pattern (higher above the hills) and the non-hydrostatic pattern (in the vicinity of the hills) is well represented by the simulation. The convergence to the correct solution can be seen also in the error measures between the simulations for the two different resolutions and the analytic solution (table 2). The integration area for the determination of the errors is chosen as far as possible away from the boundaries, therefore it was chosen horizontally in the range $[80, 120]$ km and vertically approximately in the range $[0.5, 7]$ km. The errors for the new fast waves solver are smaller compared to the old one. Again, as for the linear wave expansion in the section before, the differences become smaller, when the local grid stretching ratio becomes smaller. It is interesting, that the version with the isotropic 3D divergence damping is even better than with the quasi-3D version.

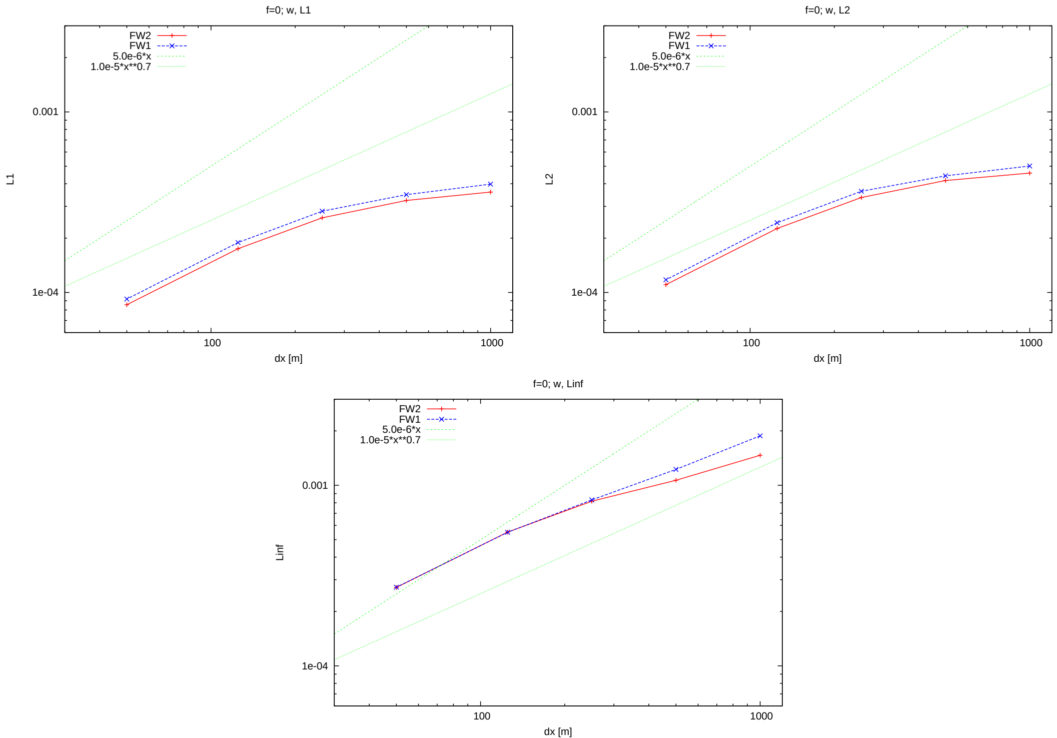

Figure 8: Linear gravity/sound wave test of BB13 with vertically stretched grid. Error norms of w for different resolutions.

9.3 3D linear flow over a hill

An analogous 3D test uses a single mountain with a rotational symmetric Gaussian shape

$$h(x) = h_{max} \cdot 2^{-\frac{x^2+y^2}{a^2}} \tag{167}$$

and a half width of $2a = 10000$ m. The hill is located exactly in the center of the simulation domain at the position $(100, 100)$ km.

Again the turbulence scheme and other parameterizations are switched off. The vertical stretching is done as in section 9.4. The inflow velocity is $u_0 = 20$ m/s, the atmospheric stratification $N = 0.01$ 1/s. The horizontal resolution is $\Delta x = \Delta y = 500$ m and a time step $\Delta t = 5$ s has been chosen.

Figure 10 shows horizontal cross sections of w after 24h simulation time. In this 3D test case the error norms of FW2 and FW1 (again against the analytic solution of **?**) are almost the same, with an improvement of only about one percent for FW2.

9.4 2D flow over steep mountains

To inspect the ability to stably simulate flow over steep mountains, a single 2D mountain (i.e. a mountain ridge) with a Gaussian shape

$$h(x) = h_{max} \cdot 2^{-\left(\frac{x}{a}\right)^2} \tag{168}$$

and a half width of $2a = 6000$ m is used. A constant inflow velocity of $u_0 = 20$ m/s and a dry atmosphere with a stable stratification $N = 0.01$ 1/s are prescribed. In the following different values for the height h_{max} are tested.

k	VCOORD(k)
1	25000.0000
2	24034.5000
3	23088.0000
4	22160.5000
...	
45	492.0000
46	362.5000
47	252.0000
48	160.5000
49	88.0000
50	34.5000
51	0.0000

k	VCOORD(k)
1	25000.0000
2	24514.8750
3	24034.5000
4	23558.8750
...	
95	160.5000
96	121.8750
97	88.0000
98	58.8750
99	34.5000
100	14.8750
101	0.0000

Table 1: vertical levels for the 2D linear flow over hills generated by eq. (166) for $ke = 50$ (left) and $ke = 100$ (right).

Δx	FW	$L_1 \cdot 10^{-5}$	$L_2 \cdot 10^{-5}$	$L_\infty \cdot 10^{-5}$
1000	FW1	4.377	7.749	34.88
1000	FW2	3.357	5.898	26.91
1000	FW2, 3D div.damp.	3.007	5.183	28.92
500	FW1	2.112	3.191	15.23
500	FW2	1.940	2.890	12.89
500	FW2, 3D div.damp.	1.273	1.780	7.56

Table 2: 2D linear flow over flat mountains. Error measures for two different resolutions in a stretched grid.

The horizontal grid mesh size of the 2-dimensional simulations is $\Delta x = 1000$ m. Therefore a time step $\Delta t = 10$ s has been chosen. The vertical grid is stretched again by the quadratic stretching function (166). Here we choose $H_{top} = 25$ km, $ke = 65$, and $\beta = 0.95$, leading to $\Delta z_{bottom} = 24.9$ m and $\Delta z_{top} = 745$ m, which are quite realistic values for model applications. The simulations run with a turbulence scheme switched on, to prevent from unphysically strong wave breaking.

The simulation with FW1 remains stable until $h_{max} = 2100$ m, but becomes unstable for $h_{max} = 2200$ m. This latter mountain height corresponds to a maximum slope angle $\alpha \approx 27°$ and a maximum step size of the orography $\max_i |h_i - h_{i+1}| \approx 500$ m. The instability develops in the bottom grid boxes, where high vertical velocities occur (Fig. 11).

In contrast, the simulation with the new fast waves solver FW2 remains stable until $h_{max} = 3100$ m. This corresponds to a maximum slope angle $\alpha \approx 35°$ and a maximum step size of the orography $\max_i |h_i - h_{i+1}| \approx 710$ m (left panel in Fig. 12). The same height can be achieved both for the quasi-3D and for the full 3D-divergence damping.

Of course, one cannot expect a large increase in stability over steeper mountains, because the main time integration scheme is the same in the new and the old fast waves solver. In particular the treatment of the metric correction terms is nearly the same. But nevertheless, the better discretization of vertical operations seems to help a bit in stability.

The same simulation with Mahrer discretization (see section 4.2.2) allows even much higher mountains until $h_{max} = 4800$ m. This corresponds to a maximum slope angle $\alpha \approx 47°$ and

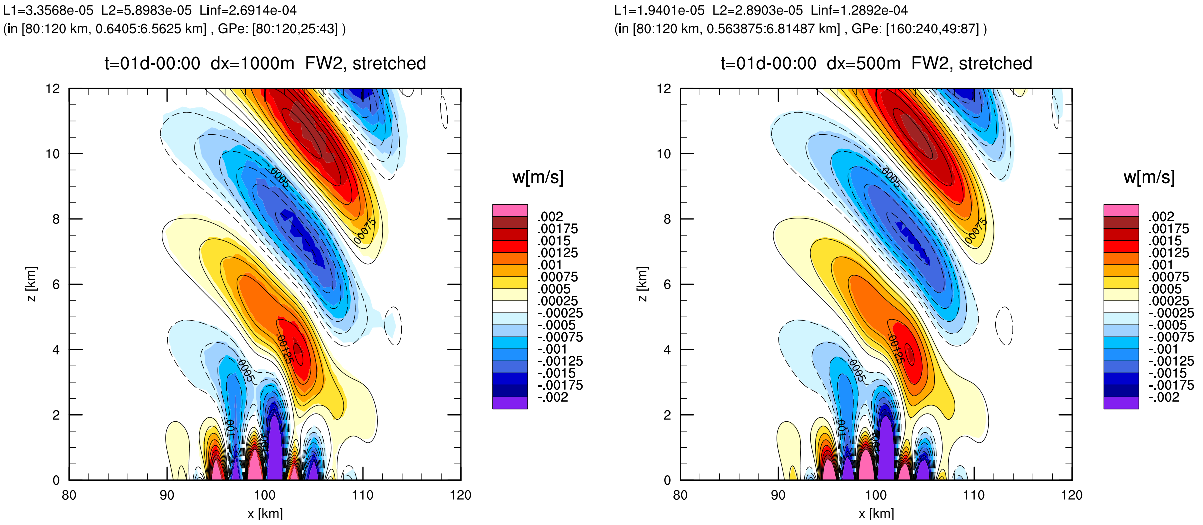

Figure 9: Linear flow over mountains (Schaer et al. 2002-setup) with vertical grid stretching by FW2. Vertical velocity w for $\Delta x = 1000$ m (left column) and $\Delta x = 500$ m (right column).

a maximum step size of the orography $\max_i |h_i - h_{i+1}| \approx 1100$ m (right panel in Fig. 12).

9.5 3D flow over steep mountains

To inspect stability over 3D steep mountains we use again a rotational symmetric Gaussian hill

$$h(x) = h_{max} \cdot 2^{-\frac{x^2+y^2}{a^2}} \tag{169}$$

with a half width of $2a = 6000$ m.

The horizontal resolution is $\Delta x = \Delta y = 1000$ m. Other model settings (vertical grid, time step, parameterizations, ...) are as in the section 9.4 before.

The simulation with FW1 is stable until a hill height of 1900 m (and becomes unstable for 2000 m). In contrast, the simulation with FW2 is stable until a hill height of 3000 m (and becomes unstable for 3100 m). Again much steeper slopes can be simulated with FW2 and the Mahrer discretization, here hill heights until 4600 m can be achieved (becomes unstable for 4700 m).

Compared to the values for the flow over 2D mountain ridges, all these maximum heights are a bit reduced for the 3D case. This indicates that more complex flows tend to reduce the maximum allowable slopes and orography steps.

9.6 Convective warm bubble test

The warm bubble test by **?** is a standard test case to investigate the model behavior in strong convective situations.

An elliptic shaped 3D warm bubble with horizontal radius of $x_d = y_d = 10000$ m, a vertical radius of $z_d = 1400$ m and a maximum temperature increment of 2 K is set into a prescribed moist atmosphere (see figure 13). The atmosphere has a prescribed velocity profile with no velocity at the bottom and an increasing velocity with height until $u_{max} = 30$ m/s (this value is used here).

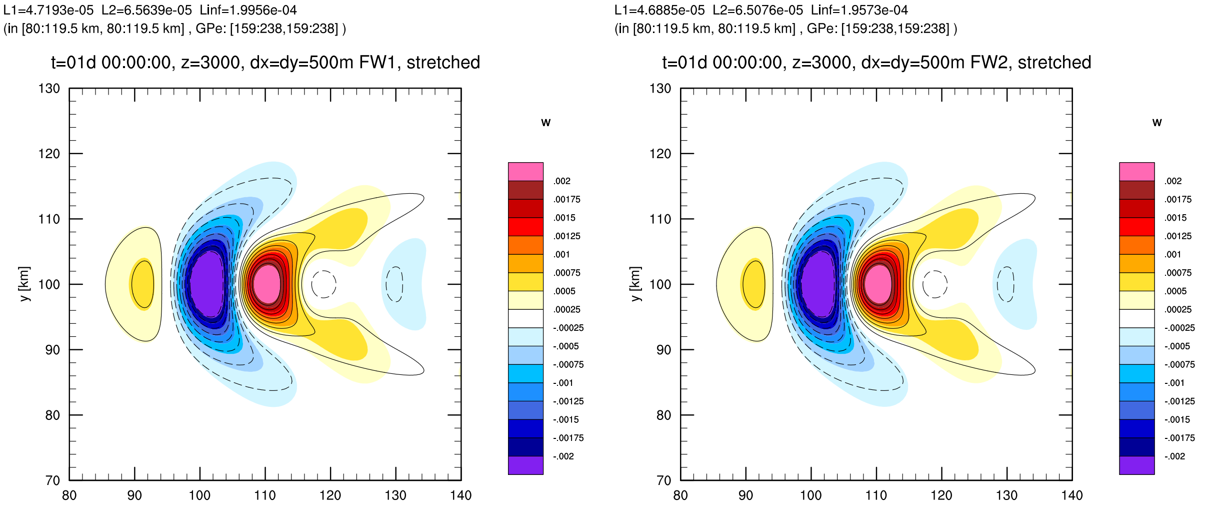

Figure 10: Vertical velocity w for linear Flow over a 3D Gaussian shaped mountain with height 1 m. Vertical velocity w simulated with the old (left) and new (right) fast waves solver (FW2) after 24 h. Cross section in $z = 3000$ m. Shaded: COSMO simulation, black lines: analytic solution

The COSMO setup has been chosen as close as possible to WK82; exceptions are a higher horizontal resolution of $\Delta x = \Delta y = 1000$ m, a time step of $\Delta t = 10$ sec., and the use of the 6-class graupel microphysics scheme (i.e. the prognostic variables q_v, q_c, q_i, q_r, q_s, q_g are used). The vertical grid points are prescribed again by (166) with $ke = 65$, $H_{top} = 22$ km, and $\beta = 0.95$. Figure 14 shows the specific mass concentrations after 30 min. and 1 h simulation time for three different model setups. The runs with FW1 generates higher maximum values of graupel and rain compared to the runs with FW2, whereas FW2 produces higher maxima of cloud ice and snow.

Figure 15 shows the time development of the maximum vertical velocity. The first cumulonimbus cloud with maximum updrafts develops after about 30 min. This structure then dissipates and generates new clouds later on. Generally these values are in a realistic range, nevertheless they are a bit higher than those of **?**. But one should notice, that their simulations have been done with $\Delta x = \Delta y = 2000$ m, probably leading to more diffusion compared to the higher resolution used here. FW2 produces slightly higher values of $w_{max}(t)$ than FW1, in particular for later times.

The simulations with FW2 have been performed with 2 setups: one with the standard quasi-3D divergence damping (i.e. $\alpha^v_{div} = 0$) and one with the full 3D divergence damping (i.e. $\alpha^v_{div} = \alpha^h_{div}$). The differences between these two versions are quite small. Therefore the slightly unphysical modification of the dispersion relation for gravity waves by the quasi-3D divergence damping mentioned in **?** seems to be not too strong in this case.

An interesting aspect in connection with divergence damping is the treatment of the first small time step. For the quasi-3D version it is absolutely necessary for real case simulations to perform *no* divergence damping in the first step; otherwise instabilities occur. These are obviously induced by complex terrain, because they haven't been neither seen in the stability analysis without orography (**?**) nor do they occur in idealized tests with simple orographies. In contrast to this, simulations with the full 3D divergence are independent from the treatment of the first small time step: they run stable, both if it is applied in the first step or not. [5]. In this respect, the full 3D divergence damping is also more satisfying

[5] To force the application of divergence damping also in the first small time step, one has to set the *internal*

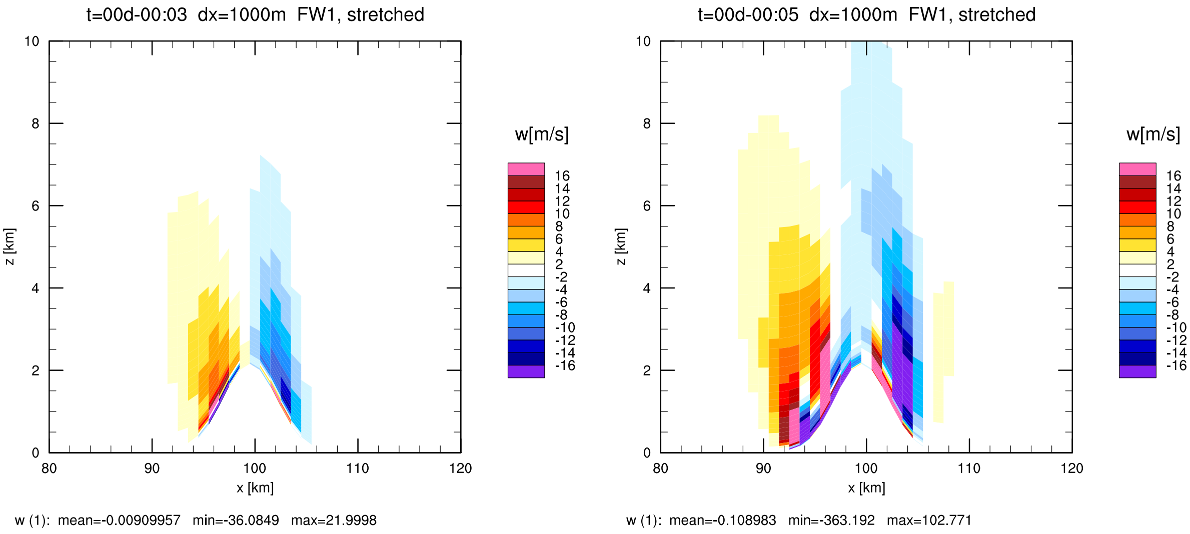

Figure 11: Flow over a 2D Gaussian shaped mountain with height 2200m. Vertical velocity w simulated with FW1 after 3 and 5 minutes. The run crashed only a few time steps later.

from a theoretical viewpoint than the quasi 3D version (the better dispersion properties of the full 3D version have been mentioned already in the introduction 2).

10 Real cases

The predominant feature of the new fast waves solver is its higher stability property. This has been demonstrated in the idealized tests in sections 9.4 and 9.5. There have been numerous examples of real case simulations which have been stabilized by using FW2. To list some of them:

The operational deterministic COSMO-DE and several COSMO-DE-EPS ensemble runs at '28.06.2011, 6 UTC' crashed by a shear instability after about 16 hours simulation time. Although this should be cured by a Smagorinsky diffusion, the runs with the new FW2 remained stable.

Another problem occurs, if the velocity fields near the ground become too noisy. This happened for the operational COSMO-DE run at '12.07.2011, 6 UTC', where q_r 'exploded' in the Alps. This could be cured by the more stable version of the tracer advection scheme, namely with the Strang-splitting variant of the Bott advection scheme. Otherwise, without this measure, the runs with the new FW2 remained stable, too.

Other examples were several experimental COSMO-DE runs with 65 levels which crashed during the summer 2012 period. (e.g. the '15.07.2011, 12 UTC' run crashed after only 215 time steps (i.e. after about 1.5 h)) FW2 experiments didn't show any crash during July and August 2012.

O. Fuhrer reported an operational COSMO-2 '16.06.2011, 0 UTC' run model crash after 1190 time steps (6.6 h). A. Seifert reported a model crash of an experimental COSMO-1 (resolution 0.01° (about 1.1km), 1700 * 1700 grid points) at the '24.08.2011 after only 10 time steps. Again these two runs have been stably simulated by FW2.

switch `l_no_div_damping_in_1st_step=.FALSE.` in module `fast_waves_sc`.
Attention: the runs with the 3D divergence damping *and* `l_no_div_damping_in_1st_step=.TRUE.` do *not work* in COSMO 4.27 due to a bug in the code. This will probably be fixed in version 4.28.

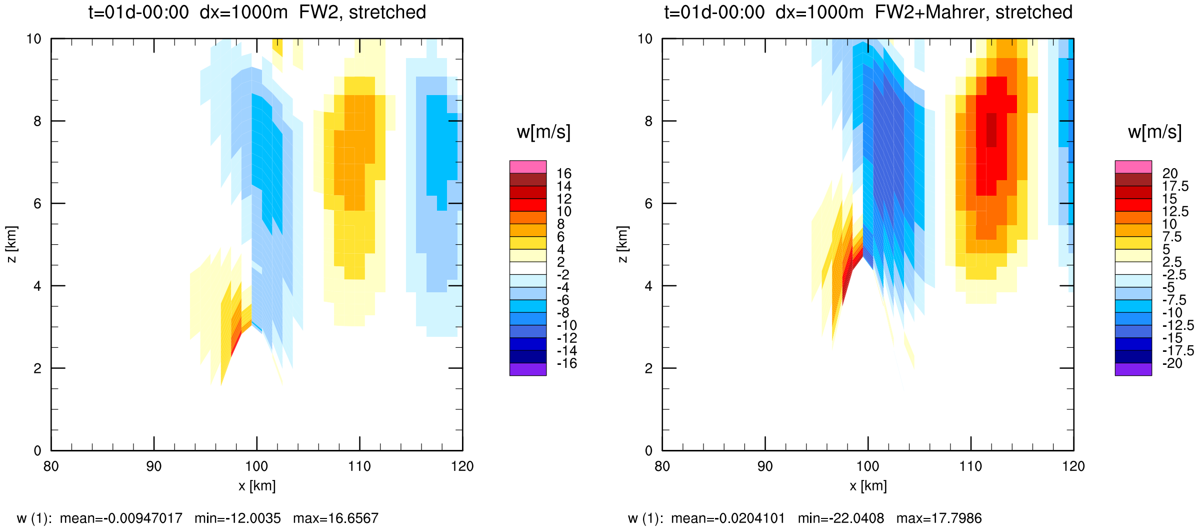

Figure 12: Flow over a 2D Gaussian shaped mountain. Vertical velocity w simulated with the new fast waves solver (FW2) after 24 h. Left: with conventional pressure discretization and a mountain height $h_{max} = 3100$ m. Right: with Mahrer (1984)-discretization and $h_{max} = 4800$ m.

Though a slightly higher accuracy of FW2 compared to FW1 has been shown in the idealized tests, the verification results for several periods at DWD didn't show larger deviations between these two versions. In particular during the winter period 2012/2013 there occured slight improvements in the pressure bias but a slight negative behavior in the cloud coverage. Nevertheless, the higher stability of the new fast waves solver led to the decision to introduce it both in COSMO-DE (2.8 km) and COSMO-EU (7 km) at 16 January 2013 after a 5 month testing period at DWD.

One should notice, that for a proper interpolation form driving initial and boundary data, the interpolation program INT2LM (since version 1.20) must use the same reference state as COSMO for the *half levels*, too. Therefore, the namelist switch `lanalyt_calc_T0p0` must be set to `.TRUE.` if `itype_fast_waves=2` is used (to be precise: this is only necessary, if the older reference atmosphere `irefatm=1` is used). The same holds for possible postprocessing programs, e.g. the `lmstat` feedback file generator.

11 Conclusions

The development of a new fast waves solver for the COSMO model has been described in this report. This new development is available since COSMO version 4.24 and should be used together with the interpolation program INT2LM since version 1.20. The slightly better accuracy properties and in particular the better stability properties in steeper terrain has been demonstrated. In the end, the new fast waves solver was introduced operationally at DWD for both model setups 2.8 km COSMO-DE (deterministic and ensemble mode) and 7 km COSMO-EU at 16 January 2013.

On the NEC SX9 (DWD) the new fast waves solver needs about 30 percent more computation time compared to the old one. This results in an increase of runtime of about 6 percent for the whole COSMO model. A certain increase in runtime is understandable due to the larger number of computations and storage loads by the additional vertical weightings. These values are also quite dependent from the computer architecture. On some Intel based computers

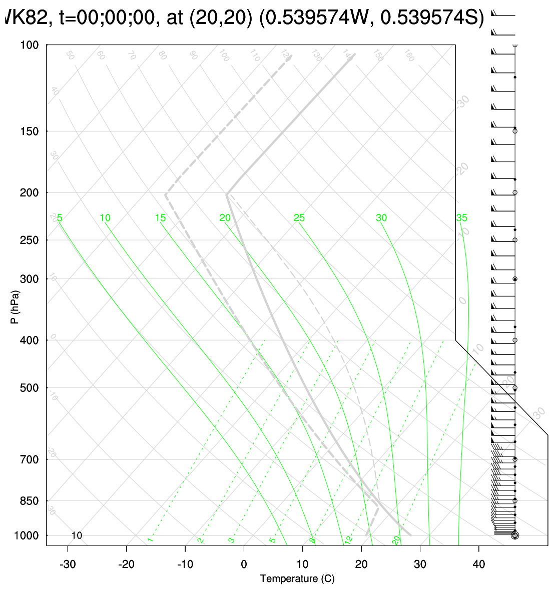

Figure 13: skew T - log p -digram for the environment atmosphere of the Weisman, Klemp (1982) test case.

these run times can be even higher, but a report from our Russian colleagues (running COSMO on an SGI ALTIX, built on Intel(R) Xeon(R) X5560 processors) indicates, that the runtime can even be much smaller with the new FW solver.

Of course, there is still enough work to be done. For example, until now it is not possible to use the exact boundary condition for Z and hence for the divergence, which is a bit unsatisfying. Though the Mahrer discretization (section 4.2.2) has been proven as advantageous in idealized steep mountain tests and as stable in many real case simulations, a detailed verification and assessment is still needed, possibly together with an improvement of the lower boundary treatment. Finally, the isotropic 3D divergence damping offers theoretical advantages and runs stable in several real test cases. Consequently, it should be further investigated. The full 3D divergence damping results in an about 10 percent longer runtime for the fast waves solver, resulting in an about 2 percent longer runtime for the whole COSMO model. Therefore, the question remains, if these additional computational costs justify its usage in operational applications.

The new solver was developed for the Runge-Kutta time integration scheme, which is now used by every NWP center of the COSMO consortium and by the most (if not all) climate runs of the climate CLM community. But in principle, there is no reason, why the new solver shouldn't be used for the 'old' leapfrog time integration scheme, too (a certain harmonization of other discretizations used in the leapfrog scheme and the vertical weightings introduced here is probably needed, before). Hence, there is a certain hope to replace the other fast waves solvers in future by this new development and therefore to reduce complexity and maintenance efforts in the COSMO code.

12 Acknowledgments

The author is grateful to Günther Zängl (Deutscher Wetterdienst), who initiated this project and helped a lot with many discussions about the topic. Many thanks also to Andreas Will

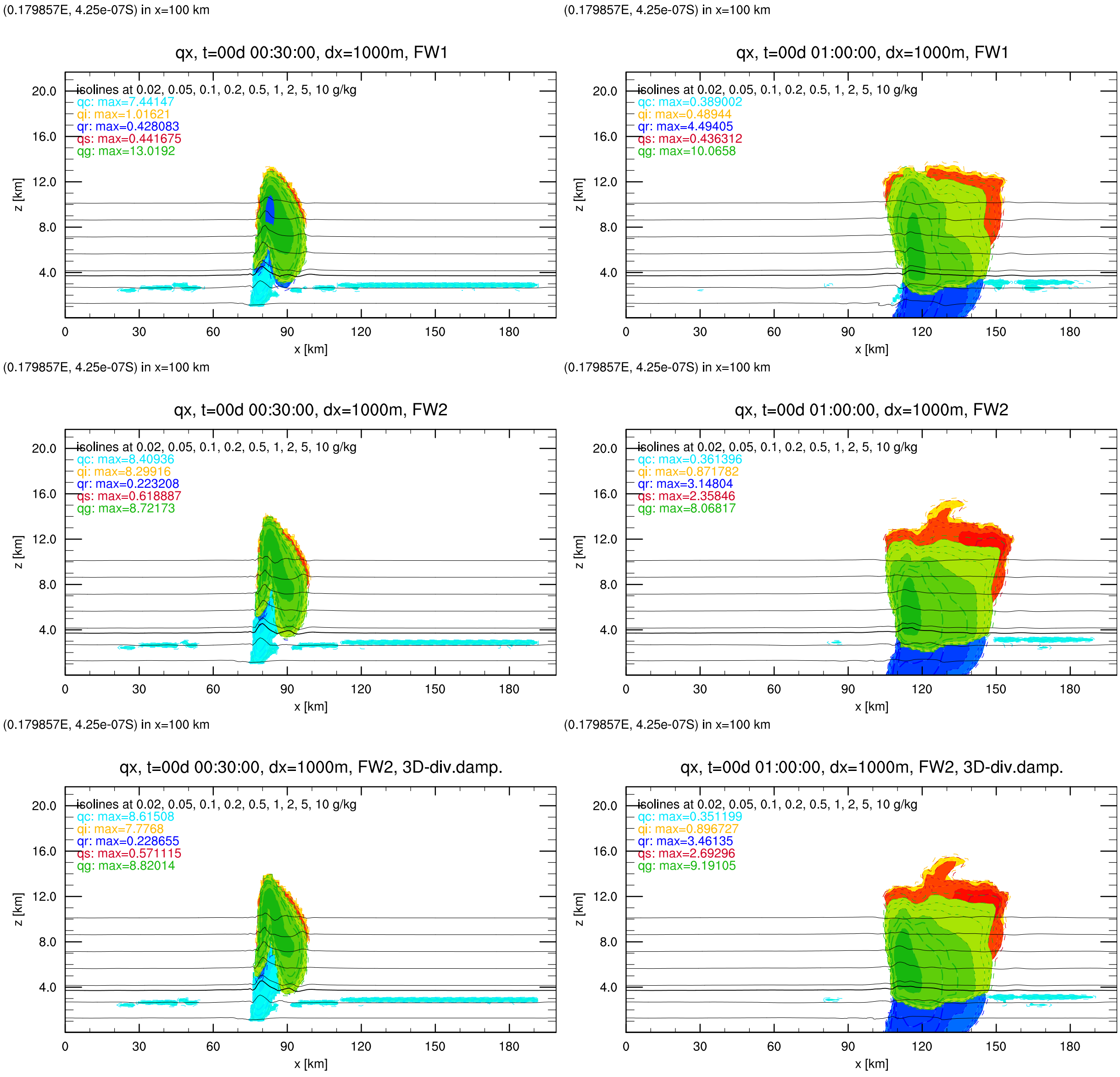

Figure 14: Weisman, Klemp (1982) test case with the 6-class graupel scheme. Specific masses of the microphysical constituents q_r, ... Left: after 30 min., Right: after 1 h. Top row: old FW1, middle row: new FW2, bottom row: new FW2 with 3D-divergence damping. Black lines: isolines of temperature; thick line is 0° C.

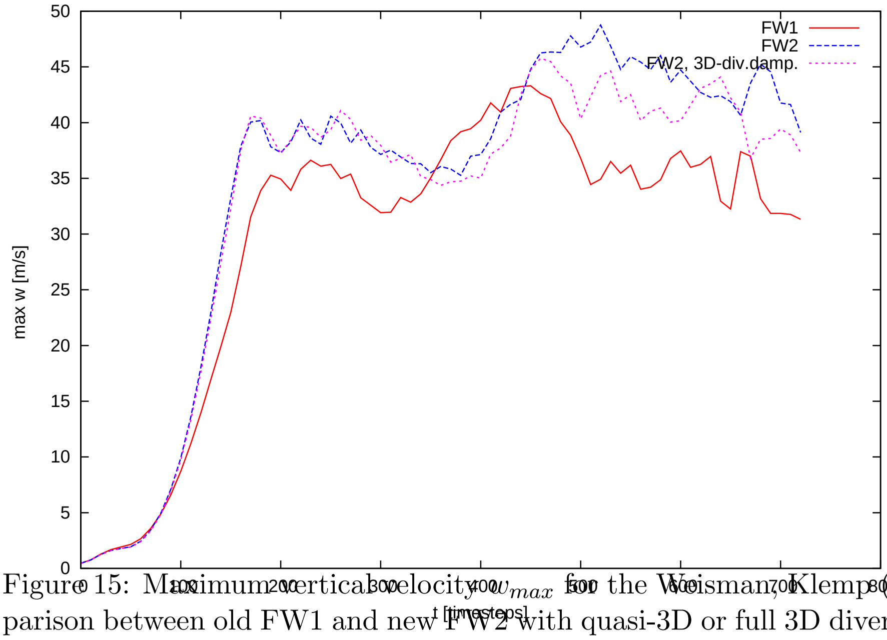

Figure 15: Maximum vertical velocity w_{max} for the Weisman, Klemp (1982) test case. Comparison between old FW1 and new FW2 with quasi-3D or full 3D divergence damping. Time step $\Delta t = 10$ sec.

(University of Cottbus) for many fruitful discussions and valuable hints. Further thanks go to Guy deMorsier and Oliver Fuhrer (MeteoSwiss), who helped a lot in testing the new fast waves solver for the COSMO-1 project. Many thanks also to Ulrich Schättler (Deutscher Wetterdienst), who has given many technical advices about the COSMO code.

List of COSMO Newsletters and Technical Reports

(available for download from the COSMO Website: www.cosmo-model.org)

COSMO Newsletters

No. 1: February 2001.

No. 2: February 2002.

No. 3: February 2003.

No. 4: February 2004.

No. 5: April 2005.

No. 6: July 2006.

No. 7: April 2008; Proceedings from the 8th COSMO General Meeting in Bucharest, 2006.

No. 8: September 2008; Proceedings from the 9th COSMO General Meeting in Athens, 2007.

No. 9: December 2008.

No. 10: March 2010.

No. 11: April 2011.

No. 12: April 2012.

No. 13: April 2013.

COSMO Technical Reports

No. 1: Dmitrii Mironov and Matthias Raschendorfer (2001):
Evaluation of Empirical Parameters of the New LM Surface-Layer Parameterization Scheme. Results from Numerical Experiments Including the Soil Moisture Analysis.

No. 2: Reinhold Schrodin and Erdmann Heise (2001):
The Multi-Layer Version of the DWD Soil Model TERRA_LM.

No. 3: Günther Doms (2001):
A Scheme for Monotonic Numerical Diffusion in the LM.

No. 4: Hans-Joachim Herzog, Ursula Schubert, Gerd Vogel, Adelheid Fiedler and Roswitha Kirchner (2002):
LLM ¯ the High-Resolving Nonhydrostatic Simulation Model in the DWD-Project LITFASS.
Part I: Modelling Technique and Simulation Method.

No. 5: Jean-Marie Bettems (2002):
EUCOS Impact Study Using the Limited-Area Non-Hydrostatic NWP Model in Operational Use at MeteoSwiss.

No. 6: Heinz-Werner Bitzer and Jürgen Steppeler (2004):
Documentation of the Z-Coordinate Dynamical Core of LM.

No. 7: Hans-Joachim Herzog, Almut Gassmann (2005):
Lorenz- and Charney-Phillips vertical grid experimentation using a compressible non-hydrostatic toy-model relevant to the fast-mode part of the 'Lokal-Modell'.

No. 8: Chiara Marsigli, Andrea Montani, Tiziana Paccagnella, Davide Sacchetti, André Walser, Marco Arpagaus, Thomas Schumann (2005):
Evaluation of the Performance of the COSMO-LEPS System.

No. 9: Erdmann Heise, Bodo Ritter, Reinhold Schrodin (2006):
Operational Implementation of the Multilayer Soil Model.

No. 10: M.D. Tsyrulnikov (2007):
Is the particle filtering approach appropriate for meso-scale data assimilation ?

No. 11: Dmitrii V. Mironov (2008):
Parameterization of Lakes in Numerical Weather Prediction. Description of a Lake Model.

No. 12: Adriano Raspanti (2009):
COSMO Priority Project "VERification System Unified Survey" (VERSUS): Final Report.

No. 13: Chiara Marsigli (2009):
COSMO Priority Project "Short Range Ensemble Prediction System" (SREPS): Final Report.

No. 14: Michael Baldauf (2009):
COSMO Priority Project "Further Developments of the Runge-Kutta Time Integration Scheme" (RK): Final Report.

No. 15: Silke Dierer (2009):
COSMO Priority Project "Tackle deficiencies in quantitative precipitation forecast" (QPF): Final Report.

No. 16: Pierre Eckert (2009):
COSMO Priority Project "INTERP": Final Report.

No. 17: D. Leuenberger, M. Stoll and A. Roches (2010):
Description of some convective indices implemented in the COSMO model.

No. 18: Daniel Leuenberger (2010):
Statistical analysis of high-resolution COSMO Ensemble forecasts in view of Data Assimilation.

No. 19: A. Montani, D. Cesari, C. Marsigli, T. Paccagnella (2010):
Seven years of activity in the field of mesoscale ensemble forecasting by the COSMO–LEPS system: main achievements and open challenges.

No. 20: A. Roches, O. Fuhrer (2012):
Tracer module in the COSMO model.

No. 21: M. Baldauf (2013):
A new fast-waves solver for the Runge-Kutta dynamical core.

COSMO Technical Reports

Issues of the COSMO Technical Reports series are published by the *COnsortium for Small-scale MOdelling* at non-regular intervals. COSMO is a European group for numerical weather prediction with participating meteorological services from Germany (DWD, AWGeophys), Greece (HNMS), Italy (USAM, ARPA-SIMC, ARPA Piemonte), Switzerland (MeteoSwiss), Poland (IMGW), Romania (NMA) and Russia (RHM). The general goal is to develop, improve and maintain a non-hydrostatic limited area modelling system to be used for both operational and research applications by the members of COSMO. This system is initially based on the COSMO-Model (previously known as LM) of DWD with its corresponding data assimilation system.

The Technical Reports are intended

- for scientific contributions and a documentation of research activities,
- to present and discuss results obtained from the model system,
- to present and discuss verification results and interpretation methods,
- for a documentation of technical changes to the model system,
- to give an overview of new components of the model system.

The purpose of these reports is to communicate results, changes and progress related to the LM model system relatively fast within the COSMO consortium, and also to inform other NWP groups on our current research activities. In this way the discussion on a specific topic can be stimulated at an early stage. In order to publish a report very soon after the completion of the manuscript, we have decided to omit a thorough reviewing procedure and only a rough check is done by the editors and a third reviewer. We apologize for typographical and other errors or inconsistencies which may still be present.

At present, the Technical Reports are available for download from the COSMO web site (www.cosmo-model.org). If required, the member meteorological centres can produce hard-copies by their own for distribution within their service. All members of the consortium will be informed about new issues by email.

For any comments and questions, please contact the editor:

Massimo Milelli
Massimo.Milelli@arpa.piemonte.it