The Meso-NH Atmospheric Simulation System: Scientific Documentation

Part I: Dynamics

Meso-NH version 6.0.0 – 19 March 2026

Acknowledgments

This volume contains contributions from P. Bechtold, S. Belair, Ph. Bougeault, J.M. Carrière, J. Cuxart, V. Ducrocq, C. Fischer, M. Georgelin, P. Héreil, J.-P. Lafore, C. Liousse, C. Mari, I. Mallet, P. J. Mascart, V. Masson, J.-P. Pinty, E. Richard, K. Suhre, J. Stein, P. Tulet, and J. Vilà-Guerau de Arellano. As editors, we would like to express our deep appreciation for the dedicated work of all contributors. The Meso-NH project is the achievement of a much larger team. Our thanks extend to all those who are not cited here, but have given their best to create this unique tool.

Philippe Bougeault and Patrick Mascart

Since the 2002 edition, in addition to the contributors cited above, thanks have to be extended to C. Augros, F. Auguste, D. Barbary, C. Barthe, S. Berthet, Y. Bouteloup, C. Bovalo, O. Caumont, J.-P. Chaboureau, M. Chong, A. Costes, F. Couvreux, T. Dauhut, G. Delautier, J. Escobar, O. Geoffroy, P.A. Joulin, C. Lac, P. Le Moigne, M. Leriche, Q. Libois, T. Lunet, M. Mandement, T. Marić, G. Molinié, T. Nagel, J., O. Nuissier, Pergaud, D. Ricard, Q. Rodier, R. Schoetter, F. Solmon, O. Thouron, M. Tomasini, B. Tsenova, B. Vié and F. Visentin.

Jean-Pierre Chaboureau

Chapter 1

General Presentation

Contents

1.1 Introduction

The Meso-NH Atmospheric Simulation System (Lafore et al. 1998, Lac et al. 2018) is a joint effort of Centre National de Recherches Météorologiques (Météo-France) and Laboratoire d'Aérologie (CNRS). It comprises several elements: **a numerical model** able to simulate the atmospheric motions, ranging from the large meso-alpha scale down to the micro-scale, with a **comprehensive physical package**; a flexible **file manager**; an ensemble of facilities to **prepare initial states**, either idealized or interpolated from real meteorological analyses or forecasts; a flexible **post-processing and graphical facility** to visualize the results; an ensemble of **unix procedures** to control these functions.

Some of the distinctive features of this ensemble are the followings: the model is based on an advanced set of anelastic systems. It allows for simultaneous simulation of several scales of motion, by the so-called "interactive grid-nesting technique". It allows for the in-line computation and accumulation of various terms of the budget of several quantities. It allows for the transport and diffusion of passive scalars, to be coupled with a chemical module. It uses the relatively new Fortran 90 compiler. It is tailored to be easily implemented on any Unix machine.

The documentation of this system is divided in three "books". The present "book 1" contains a **scientific introduction** to Meso-NH, discussing the system of equations, the geographical projections, the discretization, and the physical parameterizations. The "book 2" is the **algorithmic documentation** of the Fortran 90 code and of the Unix procedural package, with a summary of code organization, as well as the inputs, outputs, and actions of each subroutine. Finally, the

”book 3” is a **User Manual**, designed to allow a quick acquaintance of new users with the control parameters of the system.
Meso-NH is designed as a research tool for small and meso-scale atmospheric processes. It is freely accessible to any interested user, and we have tried to make it as ”user friendly” as possible, and as general as possible, although these two goals sometimes appeared contradictory.

1.2 The anelastic system

It is well known that the fast propagation of acoustic waves in the Euler equation system sets a very strong constraint on the computational time step. Generally speaking, three different methods have been developed to avoid this problem: (i) the use of two different time steps for acoustic and non-acoustic types of motions; (ii) the implicit treatment of some terms, which usually involves a linearization around some mean atmospheric state; (iii) finally the various ”anelastic” approximations. In presence of steep orography, the separation of the terms responsible for acoustic waves becomes much more difficult, and the problem is more complex. An extensive comparison of these method has been done by Ikawa (1988).

We have examined some of these method in great detail during the preliminary studies of the Meso-NH project, and concluded in favor of the ”anelastic” approach. In this approach, the acoustic waves are eliminated from the continuous set of equations by the use of a constant density profile instead of the actual fluid density in the continuity equation and in the momentum equation, except for the buoyancy term, which is the leading term of the approximation. The fluid becomes therefore formally incompressible, and the pressure is deduced from the solution of an elliptic equation. The continuity equation sets a strong constraint on the 3D velocity field, which is no longer synonymous of mass conservation. In fact the mass conservation must be insured by an additional equation, in a similar way as the surface pressure equation in hydrostatic models using the sigma coordinate system. A beneficial consequence is that the system has no external gravity wave, and the time discretization is entirely explicit, without introducing an overly restrictive constraint on the computational time step. This makes the code in general much more simple and conservative.

Known detrimental effects of the anelastic approximation are that: (i) the vertical velocities simulated by the model are slightly inaccurate (because the continuity and vertical momentum equations are approximate); (ii) pressure perturbations linked to the filtered acoustic waves may be transmitted instantaneously from one place to the other in the simulation domain, instead of traveling at the speed of sound. At the present time, there does not seem to exist any meteorological consequences of these effects, since the inaccuracy on the vertical velocity is smaller than the one induced by other sources (like uncertainty in the initial state and in the physical parameterizations), and since the acoustic waves are largely decoupled from meteorologically significant motions. We have therefore decided to adopt this method as the most simple allowing for an effective filtering of acoustic waves. However, at several places in the code, we have made provisions for a future possible change back to the fully compressible system, in case future research demonstrates some incapacity of the anelastic system.

Finally, among the various available versions of the anelastic approximation, we have implemented three formulations: the Lipps and Hemler (1982) anelastic system, the Wilhelmson and Ogura

(1972) ”modified anelastic equations”, and the ”pseudo–incompressible” system of Durran (1989), which offer some flexibility for the representation of meteorological flow depending on horizontal and vertical scales. A brief examination of the respective merits of these three systems is given in next chapter, but most of the rest of this manual simply illustrates the case of the Lipps and Hemler (1982) formalism for the sake of brevity.

1.3 Thermodynamic variables

The exact conservation of energy in presence of water, and especially liquid and ice phase, is a very difficult problem. The use of entropy as a prognostic thermodynamic variable appears to be the ultimate solution, however the computation of temperature, and the various mixing ratios of water from the entropy appears technically difficult, and subject to approximations. Alternative proposals by Pointin (1984), Hauf and Höller (1987), and Marquet (1991), do not get rid of these problems. Therefore, although we did not rule out the use of entropy in Meso-NH, we have decided that the first version of the model would rather use a ”dry” potential temperature as a thermodynamic variable. The evolution equation for this variable is an original one, and takes into account the effect of water vapor, and water phase changes, to a very good accuracy, we think.

1.4 Water

We adopted the use of mixing ratio, rather than specific content, for the various water phases, and other scalars in the model. The specific content is the ratio of the mass of a given substance to the *total mass* within a given volume. This total mass may vary because of the loss of water by precipitation, making the use of specific content tricky. The mixing ratio is the ratio of the mass of considered substance to the mass of *dry air* within the considered volume. The later is a true invariant. Therefore, we think that mixing ratio is a more appropriate quantity to conserve.

For the water, up to seven different mixing ratios can be carried by the model, corresponding to vapor, cloud liquid water, rain water, cloud ice, snow, graupel, and hail. The user may choose to use only a few of these, depending of his interest.

1.5 Geographical systems

Here the classical alternative is between the latitude/longitude system on the sphere, with possible stretching of the grid, and the conformal projection. Although all options may be maintained simultaneously in the code, we thought this would make it less easy to understand and maintain, and in view of the major foreseen applications, decided to code only the conformal projection system. This encompass a simple cartesian plan for very idealized applications. Since the curvature terms are retained, and the thin layer approximation may be relaxed, this does not preclude, in our opinion, the use of Meso-NH for the study of very large scale (though not global) processes. On the projection plane, the coordinates may be stretched, which allows for the use of effective variable resolution grid, much like the Canadian EFR System (Chouinard et al. 1994). In fact the combination in the same code of the grid-nesting capacity and the variable resolution option renders this tool rather unique to adress problems of scale interactions.

1.6 Vertical coordinate

Again, we were faced with a difficult choice between the classical Gal-Chen and Sommerville (1975) system of vertical coordinates, featuring a purely vertical, very simple transformation and allowing for an effective terrain following coordinate system, and the much more complicated, generalized curvilinear system, such as recently adopted in the ARPS model for instance. The major benefit expected from the generalized curvilinear transformation is to simplify the solution of the pressure problem in case of steep and nearly angulous topography. We have developed a prototype 2D version of the Meso-NH system, supporting both coordinate systems, in order to examine this point. Stein (personal communication 1995) has demonstrated that in most cases of interest for the foreseen use of Meso-NH, the Gal-Chen and Sommerville approach would do nearly as well as the generalized curvilinear system (as developed by Satomura 1989 for example), for a much reduced memory occupation and computation time. We therefore opted for the previous one, resulting in a huge simplification of the code. The major limitation of this system is not in terms of *slope*, since idealized flows over slopes of 70% have been simulated satisfactorily, but in terms of *slope discontinuity*: the topography used in the model must avoid a cliff-type behaviour. We feel this limitation would have been present with any other system anyway.

1.7 References

Chouinard, C., J. Mailhot, H. L. Mitchell, A. Staniforth, and R. Hogue, 1994: The Canadian Regional Data Assimilaton System: Operational and Research Applications. *Mon. Wea. Rev.*, **122**, 1306-1325.

Gal-Chen, T., and R. C. J. Sommerville, 1975: On the Use of a Coordinate Transformation for the Solution of the Navier-Stokes Equations. *J. Comput. Phys.*, **17**, 209-228.

Hauf, T., and H. Höller, 1987: Entropy and Potential Temperature. *J. Atmos. Sci.*, **44**, 2887-2901.

Ikawa, M., 1988: Comparison of Some Schemes for Nonhydrostatic Models with Orography. *J. Meteor. Soc. Jap.*, **66**, 753-776.

Lipps, F. B., and R. S. Hemler, 1982: A scale analysis of deep moist convection and some related numerical calculations. *J. Atmos. Sci.*, **39**, 2192-2210.

Marquet, P., 1991: On the Concept of Exergy and Available Enthalpy: Application to Atmospheric Anergetics. *Quart. J. Roy. Meteor. Soc.*, **117**, 449-475.

Pointin, Y., 1984: Wet Equivalent Potential Temperature and Enthalpy as Prognostic Variables in Cloud Modeling. *J. Atmos., Sci.*, **15**, 651-660.

Durran D. R., 1989: Improving the anelastic approximation, *J. Atmos. Sci.*, **46**, 1453–1461.

Lac, C., J.-P. Chaboureau, V. Masson, J.-P. Pinty, P. Tulet, J. Escobar, M. Leriche, C. Barthe, B. Aouizerats, C. Augros, P. Aumond, F. Auguste, P. Bechtold, S. Berthet, S. Bieilli, F. Bosseur, O. Caumont, J.-M. Cohard, J. Colin, F. Couvreux, J. Cuxart, G. Delautier, T. Dauhut, V. Ducrocq, J.-B. Filippi, D. Gazen, O. Geoffroy, F. Gheusi, R. Honnert, J.-P. Lafore, C. Lebeaupin Brossier, Q. Libois, T. Lunet, C. Mari, T. Maric, P. Mascart, M. Mogé, G. Molinié, O. Nuissier, F. Pantillon, P. Peyrillé, J. Pergaud, E. Perraud, J. Pianezze, J.-L. Redelsperger, D. Ricard, E. Richard, S. Riette, Q. Rodier, R. Schoetter, L. Seyfried, J. Stein, K. Suhre, M. Taufour, O. Thouron, S. Turner, A. Verrelle, B. Vié, F. Visentin, V. Vionnet, and P. Wautelet, 2018: Overview of the Meso-NH model version 5.4 and its applications, *Geosci. Model Dev.*, [**11**, 1929-1969

Lafore J.P., J. Stein, N. Asencio, P. Bougeault, V. Ducrocq, J. Duron, C. Fischer, P. Héreil, P. Mascart, V. Masson, J. P. Pinty, J. L. Redelsperger, E. Richard, J. Vilá-Guerau de Arellano, 1998: The Meso-NH atmospheric simulation system. Part I: adiabatic formulation and control simulations, *Ann. Geophysicae*, **16**, 90–109.

Lipps F., and R. S. Hemler, 1982: A scale analysis of deep moist convection and some related numerical calculations, *J. Atmos. Sci.*, **39**, 2192–2210.

Satomura, T., 1989: Compressible flow simulations on numerically generated grids. *J. Meteor. Soc. Japan*, **67**, 473-482.

Wilhelmson, R., and Y. Ogura, 1972: The pressure perturbation and the numerical modelling of a cloud, *J. Atmos. Sci.*, **29**, 1295–1307.

Chapter 2

Basic Equations

Contents

2.1 A family of anelastic approximations

As mentioned above in section 1.2, the size of time step that produces a stable numerical solution for explicit finite-difference approximation to the Navier-Stokes equations is restricted by the rapid propagation of sound waves. This time step limitation may be overcome by replacing the full Navier-Stokes equations by an approximation that filters out the sound waves. In search of such sound–proof systems, a wide family of anelastic approximation has been proposed, and all these systems assume that thermodynamic state variables exhibit only small departures from a suitably defined static reference state. The original form of the anelastic approximation was proposed by Batchelor (1953) and later refined by Ogura and Phillips (1962) as an improvement over the Boussinesq approximation more suitable for studying convection within deep atmospheric layers. This original anelastic system conserved modified forms of both total energy and Ertel potential vorticity (Nance and Durran 1994; Bannon 1995). Unfortunately, this original anelastic approximation only holds for an isentropic reference state, and is not really applicable to realistic meteorological investigation. Subsequently, this restriction was removed by Wihelmson and Ogura (1972) which replaced the constant reference–state potential temperature with a vertically varying reference–state potential temperature. This modified anelastic equation set improves the representation of deep convection but sacrifices potential vorticity and energy conservations (Bannon 1995). A major improvement was provided by the Lipps and Hemler (1982) equation set which allows an accurate representation of deep tropospheric phenomena, and simultaneously conserves modified forms of Ertel potential vorticity and total energy for finite amplitude disturbances (Bannon 1995). Finally, Durran (1989) has introduced the pseudo-incompressible approximation to obtain a better description of upper–level isothermal layers, while retaining a good conservation of appropriate forms of Ertel potential vorticity and total energy for finite amplitude displacements (Bannon 1995).

The current formulation of Meso–NH model allows for the use of three of these anelastic systems: the Lipps and Hemler (1982) system (LH hereafter), the traditional Wihelmson and Ogura (1972), a.k.a. the ”Modified Anelastic Equations” (MAE hereafter), and a simplified implementation of the pseudo-incompressible system of Durran (1989, DUR hereafter). The present chapter tries to describe these three implementations in an orderly way.

2.2 Preliminary definitions

We will use the following notations:

the rotation velocity of the earth $\mathbf{\Omega}$,
the temperature T,
the pressure P,
the reference value of the pressure $P_{00} = 100000 Pa$,
the total density of the moist air ρ,
the density of the dry fraction of the air ρ_d,
the specific heat at constant pressure of dry air C_{pd},
the specific heat at constant pressure of water vapor C_{pv},
the specific heat of liquid water C_l,
the specific heat of ice water C_i,
the gas constant for dry air R_d,

the gas constant for water vapor R_v,
the latent heat of vaporization L_v,
the latent heat of sublimation L_s,
the latent heat of melting L_m.

Various substances are measured by their mixing ratio $r_\bullet$, which is the mass of the substance within a given volume, divided by the mass of dry air within the same volume. In the current anelastic formulation, we assimilate the dry air mass and the reference state dry air mass (see below), thus the volumic mass may be recovered as $\rho_{dref} r_\bullet$.

The treatment of the different phases of water is flexible. Up to seven forms of water may be considered if the user wishes so, the mixing ratios being r_v for vapor, r_c for cloud liquid water, r_r for liquid rain, r_i for cloud ice, r_s for snow, r_g for graupels, and r_h for hail. The same mnemonics are used in the code, inasmuch as possible. If a user is not interested by such computations, he may decide to use only a subset of these variables. The computations concerning the remaining water variables will not be performed, resulting in lesser cost. The resulting equations may be obtained by setting the corresponding variables to zero in the following of this chapter.

Whenever necessary, we use the mixing ratio of total water substance

$$r_w = r_v + r_c + r_r + r_i + r_s + r_g + r_h \tag{2.1}$$

The specific heat at constant pressure of the moist air takes the general form

$$C_{ph} = C_{pd} + r_v C_{pv} + (r_c + r_r) C_l + (r_i + r_s + r_g + r_h) C_i. \tag{2.2}$$

An important quantity is the Exner function

$$\Pi = (P/P_{00})^{R_d/C_{pd}}. \tag{2.3}$$

We use the ”dry” potential temperature

$$\theta = \frac{T}{\Pi}. \tag{2.4}$$

The virtual temperature is defined as

$$T_v = T\,(1 + r_v R_v/R_d)/(1 + r_w), \tag{2.5}$$

and the virtual potential temperature as

$$\theta_v = \theta\,(1 + r_v R_v/R_d)/(1 + r_w). \tag{2.6}$$

$\boldsymbol{U}$ is the air velocity, and whenever necessary, its Cartesian components are called u, v, and w.

2.3 Reference state

All anelastic approximation systems rely on the hypothesis that the atmosphere will not depart very far from a ”reference state”, defined as an atmosphere at rest, in hydrostatic equilibrium, with horizontally uniform profiles of temperature $T_{ref}(z)$ and water vapor $r_{v\,ref}(z)$. No condensed water is considered in the reference state. For our application, the reference profiles are often chosen as the initial horizontal averages of actual fields over the expected domain of simulation. Any profile, however, may be used, but the inaccuracy of the computation increases if the reference state is far from the actual mean state. In the following the dependency in z only will be assumed for all quantities subscripted with $()_{ref}$.

The hydrostatic relation and the equation of state are used to derive the profiles of the virtual temperature and the Exner function of the reference state $T_{v\,ref}$, Π_{ref}:

$$T_{v\,ref} = T_{ref}(1 + r_{v\,ref}R_v/R_d)/(1 + r_{v\,ref}) \tag{2.7}$$

$$\frac{d(Log\Pi_{ref})}{dz} = -\frac{g}{C_{pd}T_{v\,ref}} \tag{2.8}$$

with the upper boundary condition $\Pi_{ref} = \Pi_{ref}^{top}$ at the model top $z = H.$

$\theta_{v\,ref}$ is then deduced from Π_{ref} and $T_{v\,ref}$ as

$$\theta_{v\,ref} = \frac{T_{v\,ref}}{\Pi_{ref}} \tag{2.9}$$

Finally, ρ_{ref} is deduced from the equation of state

$$\rho_{ref} = \frac{\Pi_{ref}^{C_{vd}/R_d}P_{00}}{R_d\theta_{v\,ref}} \tag{2.10}$$

and the density of the dry air fraction $\rho_{d\,ref}$ is deduced by

$$\rho_{d\,ref} = \frac{\rho_{ref}}{(1 + r_{v\,ref})} \tag{2.11}$$

Note that both the dry air and the water vapor must be considered to build the profile of the Exner function of the reference state, since each gas is subject to the partial pressure of the other one.

2.4 Equation of state

A primary difference between the three anelastic systems used in Meso–NH is the form adopted for the equation of state. Both the modified anelastic (MAE) and Lipps-Hemler (LH) systems follow the usual framework of the Boussinesq approximation by linearizing the equation of state. On the contrary, the Durran system (DUR) keeps the full equation of state, without any linearization.

Modified–anelastic and Lipps–Hemler systems

For the MAE and LH cases, the equation of state $p = \rho R_d T_v$ is replaced by the linearized form:

$$\rho' = \rho_{ref}\left(\frac{C_{vd}}{R_d}\frac{\Pi'}{\Pi_{ref}} - \frac{\theta'_v}{\theta_{v\,ref}}\right), \tag{2.12}$$

where $\rho' = \rho - \rho_{ref}, \Pi' = \Pi - \Pi_{ref}$, and $\theta'_v = \theta_v - \theta_{v\,ref}$.

Durran system

In the DUR case, contrary to most anelastic systems, the equation of state

$$p = \rho R_d T_v \tag{2.13}$$

is not linearized, and thus is used without any approximation.

2.5 Anelastic constraint

For all three anelastic system used in Meso–NH, the continuity equation is may be written as the approximated form:

$$\boldsymbol{\nabla} \cdot (\rho_{deff} \boldsymbol{U}) = 0, \tag{2.14}$$

where ρ_{deff} is a convenient compact notation for:

$$\rho_{deff} = \begin{cases} \rho_{dref}, & \text{(Lipps–Hemler)} \quad (2.15) \\ \rho_{dref}, & \text{(Mod. Anelastic Eq.)} \quad (2.16) \\ \rho_{dref}\theta_{vref}(1 + r_{vref}). & \text{(Durran)} \quad (2.17) \end{cases}$$

Strictly speaking, the Durran (1989) form of the anelastic constraint (equation 2.17) only holds in the adiabatic limit, and the rhs of (2.17) should be written $\dfrac{\mathcal{H}}{C_{pd}\Pi_{ref}}$ in the general case where $\mathcal{H}$ is the heating rate per unit volume. However, Durran (1989, p. 1454) argues that this generalized form reduces to (2.17) in the usual case where the anelastic assumption itself holds.

Also note that the dimensional definition of ρ_{deff} is slightly inconsistent. It corresponds to the dry air density of the reference state when the Lipps–Hemler deep convection anelastic (LH) or the Modified Anelastic Equation (MAE) approximations are used; but it includes an extra potential temperature factor when the pseudo-incompressible anelastic approximation (DUR) of Durran (1989) is used.

Also note that equation (2.14) represents a strong *geometrical* constraint on the wind field, called the *anelastic constraint*. This constraint is enforced by solving an elliptic equation for some pressure deviation function, that results from the combination of the continuity and momentum equations (see below).

2.6 Conservation of momentum

The system is written in a referential frame linked to the earth, with a rotation velocity $\boldsymbol{\Omega}$, and includes a yet un–specified momentum source term $\boldsymbol{\mathcal{F}}$. In Meso-NH, unlike most atmospheric models, the total density of the air will vary in function of the precipitation and evaporation of water. Therefore a consistent flux form of the equations can only be obtained by combining the Lagrangian forms and the equation of continuity for dry air. Furthermore, we want to allow for an easy transition to the compressible system, should this one be adopted in the future. Therefore, we use an equation of conservation for *the momentum of the dry air fraction* of the fluid:

$$\frac{\partial}{\partial t}(\rho_{deff}\boldsymbol{U})+\boldsymbol{\nabla}\cdot(\rho_{deff}\boldsymbol{U}\otimes\boldsymbol{U})+\rho_{deff}\boldsymbol{\mathcal{F}}_{\boldsymbol{\Pi}}+\rho_{deff}\boldsymbol{g}\frac{\theta_v-\theta_{v\,ref}}{\theta_{v\,ref}}+2\rho_{deff}\boldsymbol{\Omega}\wedge\boldsymbol{U}=\rho_{deff}\boldsymbol{\mathcal{F}}, \quad (2.18)$$

where $\rho_{deff}\boldsymbol{\mathcal{F}}_{\boldsymbol{\Pi}}$ is the pressure gradient force, which takes different forms in the three systems:

$$\rho_{deff}\boldsymbol{\mathcal{F}}_{\boldsymbol{\Pi}}=\begin{cases}\rho_{deff}\boldsymbol{\nabla}(C_{pd}\theta_{v\,ref}\Pi'), & \text{(Lipps-Hemler)} \quad (2.19)\\ \rho_{deff}C_{pd}\theta_{v\,ref}\boldsymbol{\nabla}\Pi', & \text{(Mod. Anelastic Eq.)} \quad (2.20)\\ \rho_{deff}C_{pd}\theta_v\boldsymbol{\nabla}\Pi'. & \text{(Durran)} \quad (2.21)\end{cases}$$

A distinctive property of the Lipps and Hemler (1982) system is that the reference state virtual potential temperature is pulled inside the pressure gradient term (2.19), whereas this factor appears outside of the gradient for both the MAE and Durran (1989) systems ((2.20) and (2.21)). For the MAE system also note that the virtual potential temperature of the reference state is used, whereas the Durran (1989) system directly retains the un–approximated virtual potential temperature in (2.21).

It is also worth noticing that the form adopted for the Durran (1989) system is identical to the momentum equation in the complete compressible system. Although the hydrostatic reference state have been removed, the pressure-gradient term has not been linearized. The Durran (1989) system is therefore expected to provide a particularly accurate description of the (compressible) momentum balance.

Apart from the differences in expressing the pressure gradient term, the other terms retain their usual form, respectively representing the time evolution, the advection, the buoyancy force, the Coriolis force, and the diabatic effects. Also note that the flux form of the equation is consistently kept throughout the model, insuring an integral conservation of momentum to a very good accuracy. For convenience, the model further uses θ_v rather than θ'_v, therefore, in the following equations, we will use $\theta_v-\theta_{v\,ref}$ to denote the fluctuation of buoyancy.

Finally, enforcing the anelastic constraint (2.14) leads to the improperly called ”pressure problem” to determine Π' (DUR and MAE) or $C_{pd}\theta_v\Pi'$ (LH), respectively. In order to simplify the expression of this problem, we introduce the dynamical source of momentum:

$$\boldsymbol{S}=-\boldsymbol{\nabla}\cdot(\rho_{deff}\boldsymbol{U}\otimes\boldsymbol{U})-\rho_{deff}\boldsymbol{g}\frac{\theta_v-\theta_{v\,ref}}{\theta_{v\,ref}}-2\rho_{deff}\boldsymbol{\Omega}\wedge\boldsymbol{U}+\rho_{deff}\boldsymbol{\mathcal{F}}, \quad (2.22)$$

whereby the momentum conservation equation may be rewritten:

$$\frac{\partial}{\partial t}(\rho_{deff}\boldsymbol{U})=\boldsymbol{S}-\left\{\begin{array}{c}\rho_{deff}\boldsymbol{\nabla}(C_{pd}\theta_{v\,ref}\Pi')\\ \rho_{deff}C_{pd}\theta_{v\,ref}\boldsymbol{\nabla}\Pi'\\ \rho_{deff}C_{pd}\theta_v\boldsymbol{\nabla}\Pi'\end{array}\right\}. \quad \begin{array}{r}\text{(Lipps-Hemler)} \quad (2.23)\\ \text{(Mod. Anelastic Eq.)} \quad (2.24)\\ \text{(Durran)} \quad (2.25)\end{array}$$

2.7 Hydrostatic equation

When an hydrostatic approximation of the vertical velocity equation is required, for instance within initialization routines or to compute diagnostic quantities, the hydrostatic form to be used also differs for the three equations systems.

With the pseudo–incompressible system of Durran (1989), as the vertical momentum equation is written without any approximation, the hydrostatic relation is exact and reads:

$$\frac{\partial \Pi'}{\partial z} = \frac{g}{C_{pd}\theta_v}\frac{\theta_v - \theta_{v\,ref}}{\theta_{v\,ref}}. \tag{2.26}$$

With both the MAE and Lipp–Hemler systems, the usual form of the hydrostatic relation $dp = -\rho g dz$ should not be used to establish diagnostic quantities related to this model. For the Lipps–Hemler (1982) system, by consistency with (2.18–2.19), one should use instead:

$$\frac{\partial (C_{pd}\theta_{v\,ref}\Pi')}{\partial z} = g\frac{\theta_v - \theta_{v\,ref}}{\theta_{v\,ref}}. \tag{2.27}$$

Whereas the MAE system leads to a formula similar to (2.26), except for the use of the reference state virtual potential temperature instead of the virtual potential temperature:

$$\frac{\partial \Pi'}{\partial z} = \frac{g}{C_{pd}\theta_{v\,ref}}\frac{\theta_v - \theta_{v\,ref}}{\theta_{v\,ref}}. \tag{2.28}$$

2.8 Thermodynamic equation

For the time being, the prognostic energy variable is the dry potential temperature θ. In order to account for the effects of moisture, its equation takes the following original form

$$\begin{aligned}\frac{\partial}{\partial t}(\rho_{deff}\theta) + \boldsymbol{\nabla}\cdot(\rho_{deff}\theta\,\boldsymbol{U}) = \rho_{deff}\left[\frac{R_d + r_v R_v}{R_d}\frac{C_{pd}}{C_{ph}} - 1\right]\frac{\theta}{\Pi_{ref}} w \frac{\partial \Pi_{ref}}{\partial z} \\ + \frac{\rho_{deff}}{\Pi_{ref}C_{ph}}\left[L_m(T)\frac{D(r_i + r_s + r_g + r_h)}{Dt} - L_v(T)\frac{Dr_v}{Dt} + \mathcal{H}\right].\end{aligned} \tag{2.29}$$

The terms on the right-hand side represent respectively the moist correction in absence of any phase change (derived from the conservation of total energy, with some approximations), the effects of phase changes, and the other diabatic effects (radiation and diffusion). The use of the reference state Exner function instead of the total one in the above equation allows for an effective decoupling of the pressure problem from the thermodynamical problem, leading to a great simplification of the resolution, while retaining an excellent accuracy.

2.9 Conservation of moisture

For any of the water substances $r_\star$, the conservation equation is written:

$$\frac{\partial}{\partial t}(\rho_{deff}r_\star) + \boldsymbol{\nabla}\cdot(\rho_{deff}r_\star\boldsymbol{U}) = \rho_{deff}\mathcal{Q}_\star, \tag{2.30}$$

where $\mathcal{Q}_\star$ stands for the effects of phase changes, sedimentation and diffusion. Note the flux form, and the use of ρ_{deff} (not ρ_{ref}), insuring the existence of integral conservation properties.

2.10 Conservation of passive scalars

Similarly, the model can carry an arbitrary number of passive scalars, following the equation:

$$\frac{\partial}{\partial t}(\rho_{deff}s_\star) + \boldsymbol{\nabla}\cdot(\rho_{deff}s_\star\boldsymbol{U}) = \rho_{deff}\mathcal{S}_\star, \tag{2.31}$$

where $\mathcal{S}_\star$ stands for the effects of diabatic and chemical processes.

2.11 Conservation of total mass

The total mass inside the model domain is the sum of the mass of dry air and the mass of water (other substances are neglected for the time being)

$$\mathcal{M} = \mathcal{M}_d + \mathcal{M}_w \tag{2.32}$$

The anelastic constraint (see above) does not supply the variation of the total mass of dry air $\mathcal{M}_d(t)$ inside the model domain, since it uses the reference density profile instead of the actual density. It is therefore necessary to use an additional equation. This one depends on the general conditions of the experiment:

- If periodic or wall boundary conditions are assumed, the total mass of dry air may not vary during a simulation. It is therefore specified once and for all at the beginning ($\mathcal{M}_d(t) = \mathcal{M}_d(t_0)$).
- If the model is driven by larger-scale meteorological information (forecasts or analyses), the time variation of the total mass of dry air is assumed to be entirely governed by the larger-scale fields. This is consistent with procedures currently used in other limited area models. In that case, the value $\mathcal{M}_d(t)$ of the total mass of dry air within the simulation volume is regularly updated by linear interpolation in time between the values deduced from the larger-scale fields.

On the other hand, the total mass of water $\mathcal{M}_w(t)$ can be computed at each time during the simulation, by a simple volume integration

$$\mathcal{M}_w = \int_V \rho_{dref} r_w dV \tag{2.33}$$

where V is the total volume of the model domain. Note that this results in the total mass varying with precipitation and evaporation of water, as it should, contrary to what was assumed in most models in the past. Also, when the model is forced by larger-scale fields, the evolution of the total mass of water in the model may be different from that in the forcing fields, since the representation of physical processes is different.

2.12 Pressure equation

For all three anelastic system implemented in Meso-NH, combining the anelastic constraint (2.14) and the momentum equations (2.23–2.25), some appropriate ”pressure function” can be retrieved by solving an elliptic problem:

$$\boldsymbol{\nabla}\cdot\left\{\begin{array}{c}\rho_{d\,eff}\boldsymbol{\nabla}(C_{pd}\theta_{v\,ref}\Pi')\\ \rho_{d\,eff}C_{pd}\theta_{v\,ref}\boldsymbol{\nabla}\Pi'\\ \rho_{d\,eff}C_{pd}\theta_{v}\boldsymbol{\nabla}\Pi'\end{array}\right\}=\boldsymbol{\nabla}\cdot\boldsymbol{S},\qquad\begin{array}{c}\text{(Lipps-Hemler)}\\ \text{(Mod. Anelastic Eq.)}\\ \text{(Durran)}\end{array}\qquad\begin{array}{r}(2.34)\\ (2.35)\\ (2.36)\end{array}$$

the details of the procedure, however, are system–dependent.

Lipps–Hemler system

With the Lipps–Hemler (1982) system, it is convenient to define the pressure function in (2.34), as:

$$\Phi = C_{pd}\theta_{v\,ref}\Pi', \tag{2.37}$$

and to solve directly (2.34) for Φ, which can be interpreted as a geopotential perturbation.

As the boundary condition for the elliptic equation (2.34) are Neuman ones (see below and Chapter 4) the solution Φ, hence the pressure, is determined only to an arbitrary constant. This is sufficient for the dynamical problem; however, for thermodynamic computations, it is often necessary to use the absolute value of the pressure. Let us assume that Φ_{fg} is a particular solution of the above elliptic problem. We want to determine the additional constant Φ_0 that will insure the correct absolute value of the pressure. This is achieved by using the total mass as defined above, which may be developed following:

$$\mathcal{M} = \int_V \rho\; dV = \int_V \rho_{ref}\; dV + \int_V \rho'\; dV = \mathcal{M}_{ref} + \int_V \rho'\; dV, \tag{2.38}$$

where $\mathcal{M}_{ref}$ is the mass of the reference state (including both dry air and water), computed once and for all at the beginning of each experiment.
As the linearized equation of state (2.12) replaces the usual ideal–gas law for density fluctuations in the Lipps–Hemler system, we have then

$$\mathcal{M}(t) = \mathcal{M}_{ref} + \int_V \rho_{ref}\left(\frac{C_{vd}}{R_d}\,\frac{1}{C_{pd}\theta_{v\,ref}}\,\frac{\Phi_{fg}+\Phi_0}{\Pi_{ref}} - \frac{\theta_v-\theta_{v\,ref}}{\theta_{v\,ref}}\right)dV \tag{2.39}$$

which is solved as

$$\Phi_0 = \frac{\mathcal{M}(t) - 2\times\mathcal{M}_{ref} + \int_V \dfrac{\rho_{ref}}{\theta_{v\,ref}}\left(\theta_v - \dfrac{C_{vd}}{R_d}\,\dfrac{\Phi_{fg}}{C_{pd}\Pi_{ref}}\right)dV}{\int_V \dfrac{\rho_{ref}}{\theta_{v\,ref}}\,\dfrac{C_{vd}}{R_d}\,\dfrac{1}{C_{pd}\Pi_{ref}}\,dV} \tag{2.40}$$

The resulting field of $\Phi_{fg}+\Phi_0$ is used to retrieve the absolute pressure

$$P = P_{00}\left(\Pi_{ref} + \frac{\Phi_{fg}+\Phi_0}{C_{pd}\theta_{v\,ref}}\right)^{C_{pd}/R_d} \tag{2.41}$$

Modified anelastic system

With the MAE system, solving the elliptic equation (2.35) directly provides the Exner function deviation Π' from the reference state, instead of the above defined Φ function.

However, the rest of the solution procedure follows the principles outlined for the Lipps–Hemler system. As the boundary condition for the elliptic equation (2.34) are Neuman ones, the Exner function Π' is determined to an arbitrary constant only, and the absolute value of the pressure is obtained as above. Assuming that Π'_{fg} is a particular solution of the elliptic problem, the additional constant Π_0 that will insure the correct absolute value of the pressure is derived from the total mass budget, written using (2.38) without changes.

As the linearized equation of state (2.12) replaces the usual ideal–gas law for density fluctuations in the MAE system, we have then

$$\mathcal{M}(t) = \mathcal{M}_{ref} + \int_V \rho_{ref} \left(\frac{C_{vd}}{R_d} \frac{\Pi'_{fg} + \Pi_0}{\Pi_{ref}} - \frac{\theta_v - \theta_{v\,ref}}{\theta_{v\,ref}} \right) dV \tag{2.42}$$

which is solved as

$$\Pi_0 = \frac{\mathcal{M}(t) - 2 \times \mathcal{M}_{ref} \; + \; \int_V \rho_{ref} \left(\frac{\theta_v}{\theta_{v\,ref}} - \frac{C_{vd}}{R_d} \frac{\Pi'_{fg}}{\Pi_{ref}} \right) dV}{\int_V \frac{\rho_{ref}}{\Pi_{ref}} \frac{C_{vd}}{R_d} \, dV} \tag{2.43}$$

The resulting field of $\Pi'_{fg} + \Pi_0$ is used to retrieve the absolute pressure

$$P = P_{00} \left(\Pi_{ref} + \Pi'_{fg} + \Pi_0 \right)^{C_{pd}/R_d} \tag{2.44}$$

Durran system

Finally, with the Durran (1989) system, solving the elliptic equation (2.36) directly provides the Exner function deviation Π' from the reference state. As it was the case for other systems in previous sections, the Neuman boundary conditions only determine Π' to an arbitrary constant, and this constant has to be obtained from a total mass budget to retrieve the absolute value of the pressure field. However, as the Durran (1989) system use the equation of state without linearization, a modified procedure is followed.

Let us assume that Π'_{fg} is a particular solution of the above elliptic problem. We want to determine the additional constant Π_0 that will insure the correct absolute value of the pressure. This is achieved by using the total mass as defined above:

$$\mathcal{M} = \int_V \rho \, dV. \tag{2.45}$$

Combining the ideal–gas equation of state and the Exner function definition:

$$\rho = \frac{\Pi^{C_{vd}/R_d} P_{00}}{R_d \theta_v} \tag{2.46}$$

the total mass of the model can be written as:

$$\mathcal{M}(t) = \int_V \frac{P_{00}}{R_d \theta_v} \left(\Pi_{ref} + \Pi'_{fg} + \Pi_0\right)^{C_{vd}/R_d} dV. \tag{2.47}$$

As $\Pi_{ref} + \Pi'_{fg} \gg \Pi_0$, a linearization of the Π_0 term allows to solve the above equation as

$$\Pi_0 = \left(\frac{\mathcal{M}_d(t) + \mathcal{M}_w(t) \ + \ \int_V \frac{P_{00}}{R_d \theta_v} \left(\Pi_{ref} + \Pi'_{fg}\right)^{C_{vd}/R_d} dV}{\int_V \frac{P_{00} C_{vd}}{R_d^2 \theta_v} dV} \right)^{R_d/C_{vd}} \tag{2.48}$$

Eventually, a second iteration is performed using the same equation to get an accurate value of Π_0. The resulting constant is used to retrieve the absolute pressure:

$$P = P_{00} \left(\Pi_{ref} + \Pi'_{fg} + \Pi_0\right)^{C_{pd}/R_d} \tag{2.49}$$

2.13 Boundary conditions

Top boundary

The model is assumed to be limited by a rigid horizontal lid, exerting a free slip condition on the atmosphere. The height H of this lid may be chosen by the user. In order to obtain realistic results, it is recommended to place H somewhere in the stratosphere. At $z = H$, the conditions imposed on the model are:

$$\boldsymbol{U} \cdot \boldsymbol{n} = 0 \tag{2.50}$$

$$\boldsymbol{\nabla} \left\{ \begin{array}{c} \Phi \\ \Pi' \\ \Pi' \end{array} \right\} \cdot \boldsymbol{n} = \boldsymbol{S} \cdot \boldsymbol{n} \qquad \begin{array}{c} \text{(Lipps-Hemler)} \\ \text{(Mod. Anelastic Eq.)} \\ \text{(Durran)} \end{array} \qquad \begin{array}{c} (2.51) \\ (2.52) \\ (2.53) \end{array}$$

where $\boldsymbol{n}$ is a vertical unit vector.

In order to prevent the reflexion of gravity waves on this lid, an absorbing layer may be activated, where the model prognostic variables are relaxed towards the large-scale values.

Bottom boundary

In the adiabatic formulation of the model, the lower boundary condition is defined as an insulated rigid lid, with free slip. This may be again formulated as

$$\boldsymbol{U} \cdot \boldsymbol{n} \ = \ 0 \tag{2.54}$$

$$\tag{2.55}$$

$$\boldsymbol{\nabla} \left\{ \begin{array}{c} \Phi \\ \Pi' \\ \Pi' \end{array} \right\} \cdot \boldsymbol{n} = \boldsymbol{S} \cdot \boldsymbol{n} \qquad \begin{array}{c} \text{(Lipps-Hemler)} \\ \text{(Mod. Anelastic Eq.)} \\ \text{(Durran)} \end{array} \qquad \begin{array}{c} (2.56) \\ (2.57) \\ (2.58) \end{array}$$

but $\boldsymbol{n}$, a unit vector normal to the earth surface, may not be vertical.

In the physical package of the model, the turbulent fluxes of heat, moisture, and momentum are computed in the usual way.

Lateral boundary

This is a more complicated problem. We allow for several types of conditions:

- Periodic conditions
- Rigid wall
- Free propagation of waves out of the simulation domain
- Evolution of the prognostic variables imposed by external information from larger scale
- In some cases, a mixture of the previous conditions can be used

This will be discussed in detail in Chapter 5.

2.14 Energy and potential vorticity conservation

A thorough discussion of the conservation properties for all three anelastic systems implemented in Meso-NH is given in Bannon (1995). Only a summary of the main conclusions is therefore given below, readers being referred to the original paper for derivations and details. Following Bannon (1995), we restrict the scope of this discussion to a *dry* atmosphere, and consider a finite amplitude flow with $\mathcal{H}$ representing the heating rate per unit volume, and $\boldsymbol{\mathcal{F}}$ the frictional forces. When a conservation property is existing, it will be given in a flux form, assuring that the volume integral of the relevant quantity is conserved for adiabatic inviscid flows in a closed domain for which the normal component of the velocity field vanishes on the boundaries.

Modified anelastic system

When the potential temperature of the reference state varies with height, neither a closed form of the energy, nor a consistent statement of potential vorticity conservation are possible. The case of the constant reference state potential temperature corresponds to the original anelastic equations (Ogura and Phillips 1962), which conserve both an energy form and an Ertel potential vorticity form (see Bannon 1995, and Durran 1989, p 1459).

Lipps–Hemler system

This system conserves both a form of energy, and an Ertel–like potential vorticity. Bannon (1995, p 2305) writes the total energy equation in flux form as:

$$\frac{\partial}{\partial t}(E_{LH}) + \boldsymbol{\nabla} \cdot [(E_{LH} + P')\boldsymbol{U}] = \mathcal{H} + \rho_{dref}\boldsymbol{U} \cdot \boldsymbol{\mathcal{F}}, \tag{2.59}$$

where P' is the pressure deviation from reference state, and E_{LH} is the appropriate form of total energy in the LH system:

$$E_{LH} = \rho_{dref} \left\{ \frac{1}{2} \boldsymbol{U} \cdot \boldsymbol{U} + gz + C_{pd}\theta \right\} \tag{2.60}$$

Note the use of the reference state density. Bannon (1995) further notes that the presence of C_{pd} instead of C_{vd} in the internal energy term reveals that in the LH system heating occur at constant pressure, as is required by this system's approximations.

The Lipps and Hemler (1982) anelastic form of Ertel's conservation theorem is:

$$\frac{D}{Dt}\left(PV_{LH}\right) = \frac{\boldsymbol{\omega}_a \cdot \boldsymbol{\nabla}\dot{\Theta}}{\rho_{dref}} + \frac{\boldsymbol{\nabla}\theta}{\rho_{dref}} \left(\boldsymbol{\nabla} \times \frac{\boldsymbol{\mathcal{F}}}{\rho_{dref}} \right) \tag{2.61}$$

where $\boldsymbol{\omega_a} = 2\boldsymbol{\Omega} + \boldsymbol{\nabla} \times \boldsymbol{U}$ is the absolute vorticity, $\dot{\Theta}$ the diabatic warming rate:

$$\dot{\Theta} = \frac{\theta\, \mathcal{H}}{\rho_{dref} C_{pd} T} \tag{2.62}$$

and the appropriate form of the potential vorticity is (Bannon 1995, p 2306):

$$PV_{LH} = \frac{\boldsymbol{\omega_a} \cdot \boldsymbol{\nabla}\theta}{\rho_{dref}}. \tag{2.63}$$

Note that the reference state density has to be used, not the true density.

Durran system

The pseudo–incompressible system of Durran (1989) also exhibits conservation properties for both a total energy and an Ertel–like potential vorticity.

Defining the pseudo–incompressible density:

$$\rho^\star = \rho_{dref} \frac{\theta_{ref}}{\theta} \tag{2.64}$$

The flux form of the total energy equation is (Durran, 1989):

$$\frac{\partial}{\partial t}(E_D) + \boldsymbol{\nabla} \cdot [(E_D + P^\star)\boldsymbol{U}] = \mathcal{H} + \rho^\star \boldsymbol{U} \cdot \boldsymbol{\mathcal{F}}, \tag{2.65}$$

where $P^\star = P_{ref} + \rho_{dref} C_{pd} \theta_{ref} \Pi'$, and E_D is the appropriate form of total energy in the Durran (1989) system:

$$E_D = \rho^\star \left\{ \frac{1}{2} \boldsymbol{U} \cdot \boldsymbol{U} + gz \right\} + \rho_{dref} C_{vd} T_{ref} \tag{2.66}$$

The pseudo–incompressible version of Ertel's conservation theorem is:

$$\frac{D}{Dt}\left(PV_D\right) = \frac{\boldsymbol{\omega}_a \cdot \boldsymbol{\nabla}\dot{\Theta}}{\rho^\star} + \frac{\boldsymbol{\nabla}\theta}{\rho^\star} \left(\boldsymbol{\nabla} \times \frac{\boldsymbol{\mathcal{F}}}{\rho} \right) \tag{2.67}$$

where PV_D is the appropriate form of the potential vorticity (Bannon 1995, p 2308):

$$PV_D = \frac{\boldsymbol{\omega_a} \cdot \boldsymbol{\nabla}\theta}{\rho^\star}. \tag{2.68}$$

Note that the pseudo–incompressible density has to be used, not the reference state one.

2.15 References

Bannon, P. R., 1995: Potential vorticity conservation, hydrostatic adjustment, and the anelastic approximation. *J. Atmos. Sci.*, **52**, 2302–2312

Batchelor, G. K., 1953: The condition for dynamical similarity of motions of frictionless perfect-gas atmosphere. *Quart. J. Roy. Meteor. Soc.*, **79**, 224–235.

Durran, D. R., 1989: Improving the anelastic approximation. *J. Atmos. Sci.*, **46**, 1453–1461.

Lipps, F., and R. S. Hemler, 1982: A scale analysis of deep moist convection and some related numerical calculations. *J. Atmos. Sci.*, **39**, 2192–2210.

Lipps, F., 1990: On the anelastic approximation for deep convection. *J. Atmos. Sci.*, **47**, 1794–1798.

Nance, L. B., and D. R. Durran, 1994: A comparison of the accuracy of three anelastic systems and the pseudo–incompressible system. *J. Atmos. Sci.*, **51**, 3549–3565.

Ogura, Y., and N. A. Phillips, 1962: Scale analysis of deep and shallow convection in the atmosphere. *J. Atmos. Sci.*, **19**, 173–179.

Wilhelmson, R., and Y. Ogura, 1972: The pressure perturbation and the numerical modelling of a cloud, *J. Atmos. Sci.*, **29**, 1295–1307.

Chapter 3

Coordinate Systems

Contents

In general, it will not be possible to develop the equations of the previous chapter in a simple Cartesian coordinate system, because of the earth sphericity, and of the underlying topography. In meteorology, a natural coordinate system is defined by the longitude λ, the latitude φ, and the distance from the earth center r (or the altitude above sea surface $z = r - a$, where a is the earth radius). The vector basis associated with this natural system will be called $(\boldsymbol{i}_0, \boldsymbol{j}_0, \boldsymbol{k})$. $\boldsymbol{i}_0$ points towards the east, $\boldsymbol{j}_0$ towards the north, and $\boldsymbol{k}$ is vertical. In Meso-NH, we prefer to work with a conformal projection allowing for rotation with respect to this natural basis. This allows more flexibility in the direction of the coordinate lines to study particular processes. This also simplifies the initialization of model runs with products of operational weather prediction systems, like the ARPEGE system of Meteo-France, or the ECMWF system.

3.1 Conformal projections and terrain following coordinates

We use a system of curvilinear coordinates $\widehat{x}, \widehat{y}, \widehat{z}$, defined in the following manner.

3.1.1 Vertical coordinate

The vertical coordinate $\hat{z}$ is a height-based terrain-following coordinate which can be alternatively selected as the classical Gal-Chen and Sommerville (1975) coordinate or as the SLEVE coordinate proposed by Schär et al. (2002) and modified by Leuenberger et al. (2010). The Gal-Chen and Sommerville coordinate is defined by

$$\hat{z} = H\frac{z - z_s}{H - z_s}, \tag{3.1}$$

where H is the height of the model top, and z_s the height of the local topography.
Being terrain following, the Gal-Chen and Sommerville coordinate allows a straightforward implementation of the lower boundary condition. However, in the presence of steep orography, the small-structures in the coordinate surface render the computation of advection inaccurate and imbalances in the discretization of the horizontal pressure gradient may lead to spurious motions over mountains. An easy way to alleviate these shortcomings is to use the so-called Smooth-Level Vertical-Coordinate (SLEVE) where small-scale features in the coordinate surfaces decay rapidly with height, thus limiting the existence of steep coordinate surfaces to the lowermost few kilometers above ground. This new transformation relies on the definition of two scale heights, which govern the vertical decay of the larger- and smaller-scale contributions of the topography, respectively. To this end, the topography is split into

$$z_s = z_{s1} + z_{s2} \tag{3.2}$$

where the suscripts 1 and 2 refer to large-scale and small-scale contributions, respectively. In practice, the large-scale contribution is obtained from the full topography by an appropriate smoothing operation. The SLEVE coordinate is then defined by the relationship

$$z = \hat{z} + z_{s1}\frac{\sinh[(H/s_1)^n - (\hat{z}/s_1)^n]}{\sinh[(H/s_1)^n]} + z_{s2}\frac{\sinh[(H/s_2)^n - (\hat{z}/s_2)^n]}{\sinh[(H/s_2)^n]} \tag{3.3}$$

where n is a real number, set to 1.15. The second and the third terms in the right-hand side of (3.3) govern the decay with height of large- and small-scale terrain features with the scale heights s_1

and s_2, respectively. The optimal choice of these scale heights depends upon the problem under consideration. For instance, for a topography reaching 1500 m and a model top height of 25 km, the values s_1 =15 km and s_2 =2.5 km represent a reasonable choice.

Since the transformation is purely vertical, $\boldsymbol{k}$ is also parallel to the $\widehat{z}$ lines in the physical space (see Fig. 3.1).

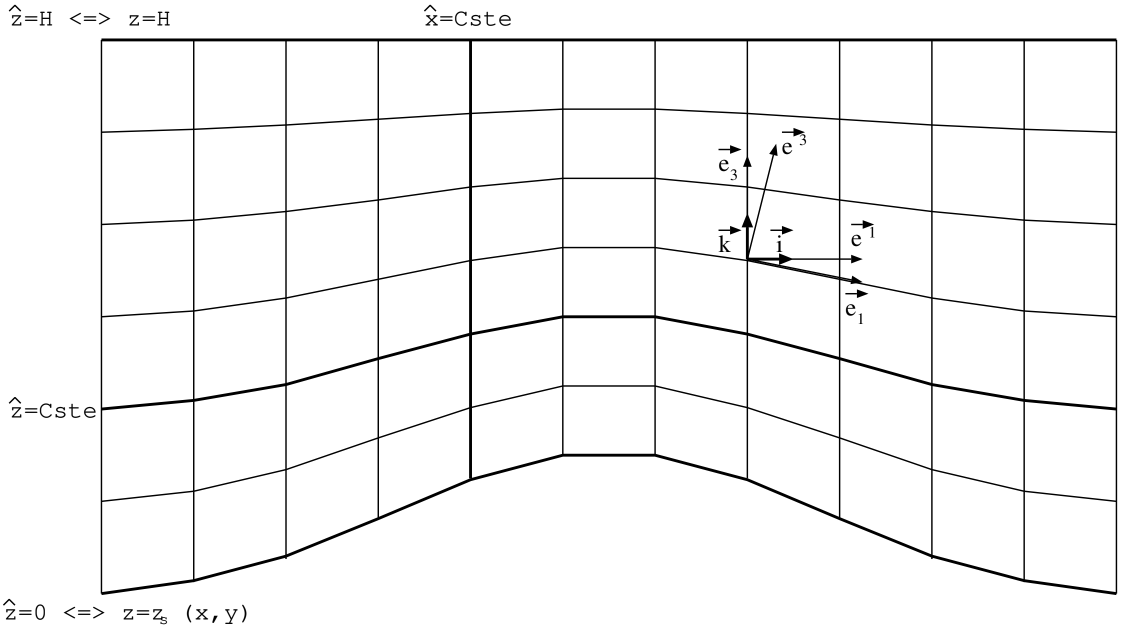


Figure 3.1: The Gal-Chen and Sommerville vertical coordinate and the Cartesian $(\boldsymbol{i}, \boldsymbol{j})$, covariant $(\boldsymbol{e}_i)$ and contravariant $(\boldsymbol{e}^i)$ bases.

On the other hand, $\widehat{x}$ and $\widehat{y}$ are the distances counted from an arbitrary origin in two arbitrary orthogonal directions on a conformal surface of projection (Fig. 3.2). The traces on the sphere of these coordinate lines define two orthogonal directions in each point. We will call hereafter $\boldsymbol{i}$ and $\boldsymbol{j}$ the horizontal, unit length vectors parallel to those directions. We note γ the angle between from $\boldsymbol{i}$ to $\boldsymbol{i}_0$. In general, this angle will vary with $\widehat{x}$ and $\widehat{y}$ because of the earth sphericity.

Clearly, $\boldsymbol{i}, \boldsymbol{j}$ and $\boldsymbol{k}$ form a local, Cartesian basis which is particularly interesting to develop the wind velocity vector. In the following, we will call u, v, w the components of $\boldsymbol{U}$ on this basis.

$$\boldsymbol{U} = u\boldsymbol{i} + v\boldsymbol{j} + w\boldsymbol{k} \tag{3.4}$$

It is the most natural decomposition of the wind in a local basis.

Three types of conformal projections are supported in Meso-NH: Lambert, polar stereographic, and Mercator. We recall here the formulae which allow to compute the coordinates $\widehat{x}, \widehat{y}$ from the latitude φ and the longitude λ of a given point, and conversely.

3.1.2 Polar stereographic and Lambert projections

The projection is defined by the conicity parameter K $(0 < K < 1)$, the earth radius a, a reference latitude ϕ_0, a reference longitude λ_0, an arbitrary angle of rotation β, and the coordinates of the

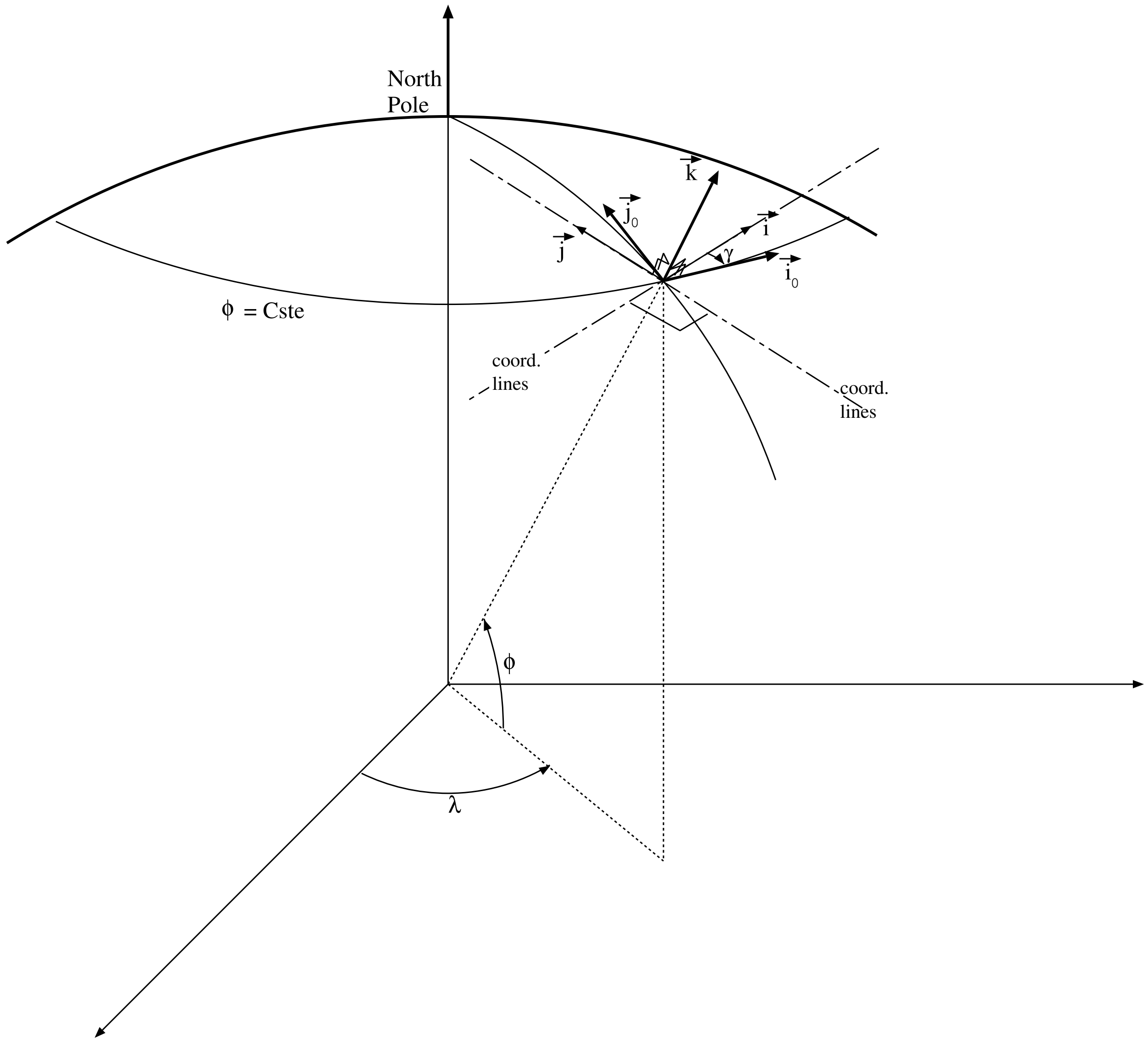


Figure 3.2: Principles of projections and notations on the sphere.

pole in the projection $\widehat{x}_0, \widehat{y}_0$ (Fig. 3.3). The useful formulae are

$$\begin{aligned}
\gamma &= K(\lambda - \lambda_0) - \beta \\
R &= \frac{a}{K}(\cos\varphi_0)^{1-K}(1+\sin\varphi_0)^K \left(\frac{\cos\varphi}{1+\sin\varphi}\right)^K \\
\widehat{x} &= \widehat{x}_0 + Rsin\gamma \\
\widehat{y} &= \widehat{y}_0 - Rcos\gamma
\end{aligned} \tag{3.5}$$

The polar stereographic projection corresponds to $K = 1$. The more general case of the Lambert conical projection is obtained for $0 < K < 1$.

The map scale factor, defined as the ratio of distances on the projection surface to distances on the sphere, is given by

$$m = \left(\frac{\cos\varphi_0}{\cos\varphi}\right)^{1-K} \left(\frac{1+\sin\varphi_0}{1+\sin\varphi}\right)^K \tag{3.6}$$

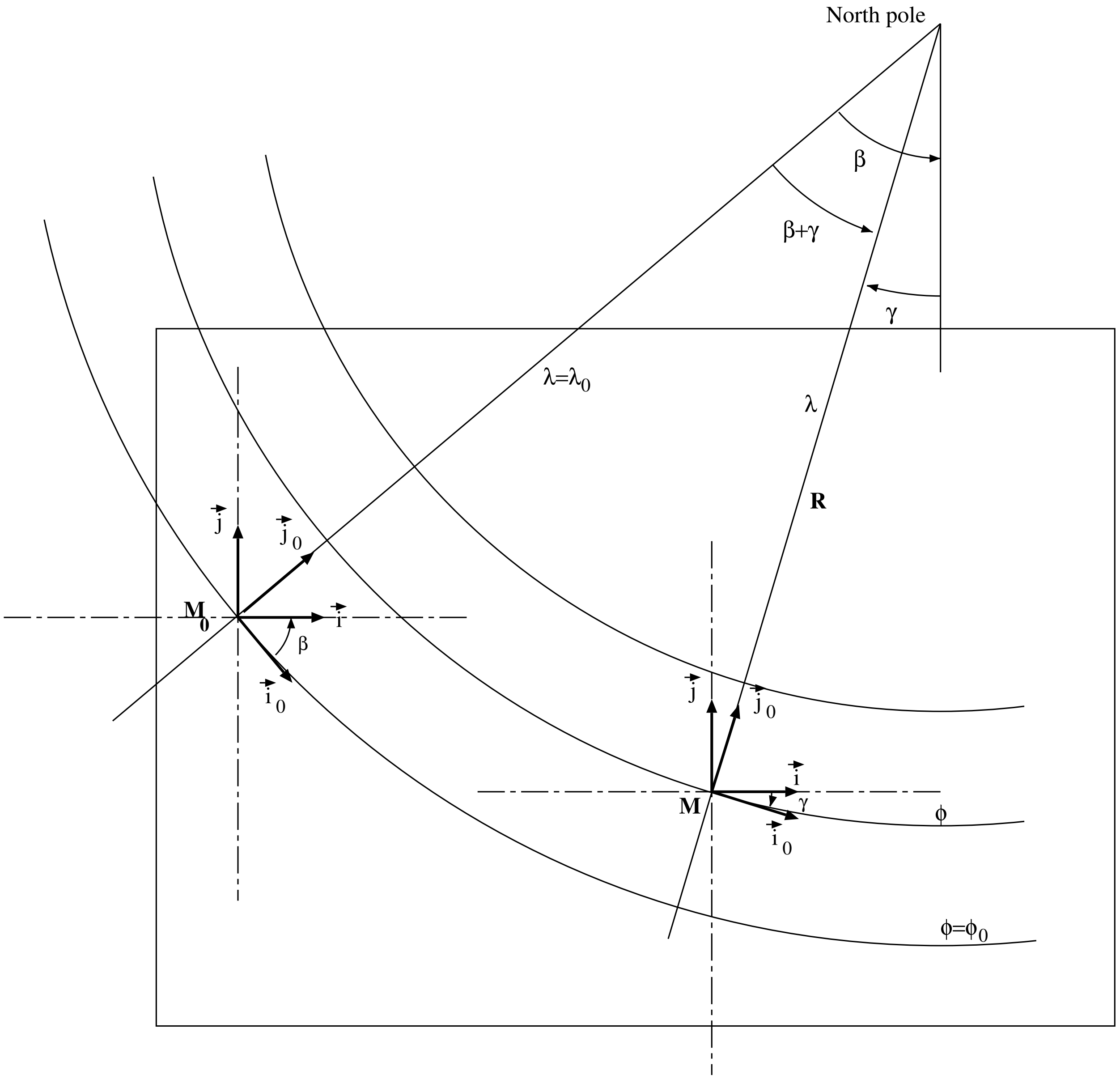


Figure 3.3: Polar stereographic and Lambert projections: M_0 is the point at the intersection of the reference latitude and longitude. In M_0, the angle from $\boldsymbol{i}$ to $\boldsymbol{i}_0$ is $-\beta$. In any other point M, this angle has a different value. Note that the North pole may be situated inside the model domain in the case of the polar stereographic projection, but not in the case of the Lambert projection.

3.1.3 Mercator projection

The projection is defined by a reference latitude ϕ_0, a reference longitude λ_0, an arbitrary angle of rotation β, and the coordinates of the origin in the projection $\widehat{x}_0, \widehat{y}_0$ (Fig. 3.4). The useful formulae are

$$
\begin{aligned}
\gamma &= -\beta \\
\widehat{x}' &= \widehat{x}_0 + a\cos\varphi_0(\lambda - \lambda_0) \\
\widehat{y}' &= \widehat{y}_0 - a\cos\varphi_0 \ln|\tan(\frac{\pi}{4} - \frac{\varphi}{2})| \\
\widehat{x} &= \widehat{x}'\cos\gamma - \widehat{y}'\sin\gamma \\
\widehat{y} &= \widehat{x}'\sin\gamma + \widehat{y}'\cos\gamma
\end{aligned}
\tag{3.7}
$$

For this projection, if β is chosen equal to zero, the local Cartesian basis coincides everywhere with the natural basis.
The map scale factor is given by

$$m = \frac{\cos\varphi_0}{\cos\varphi} \tag{3.8}$$

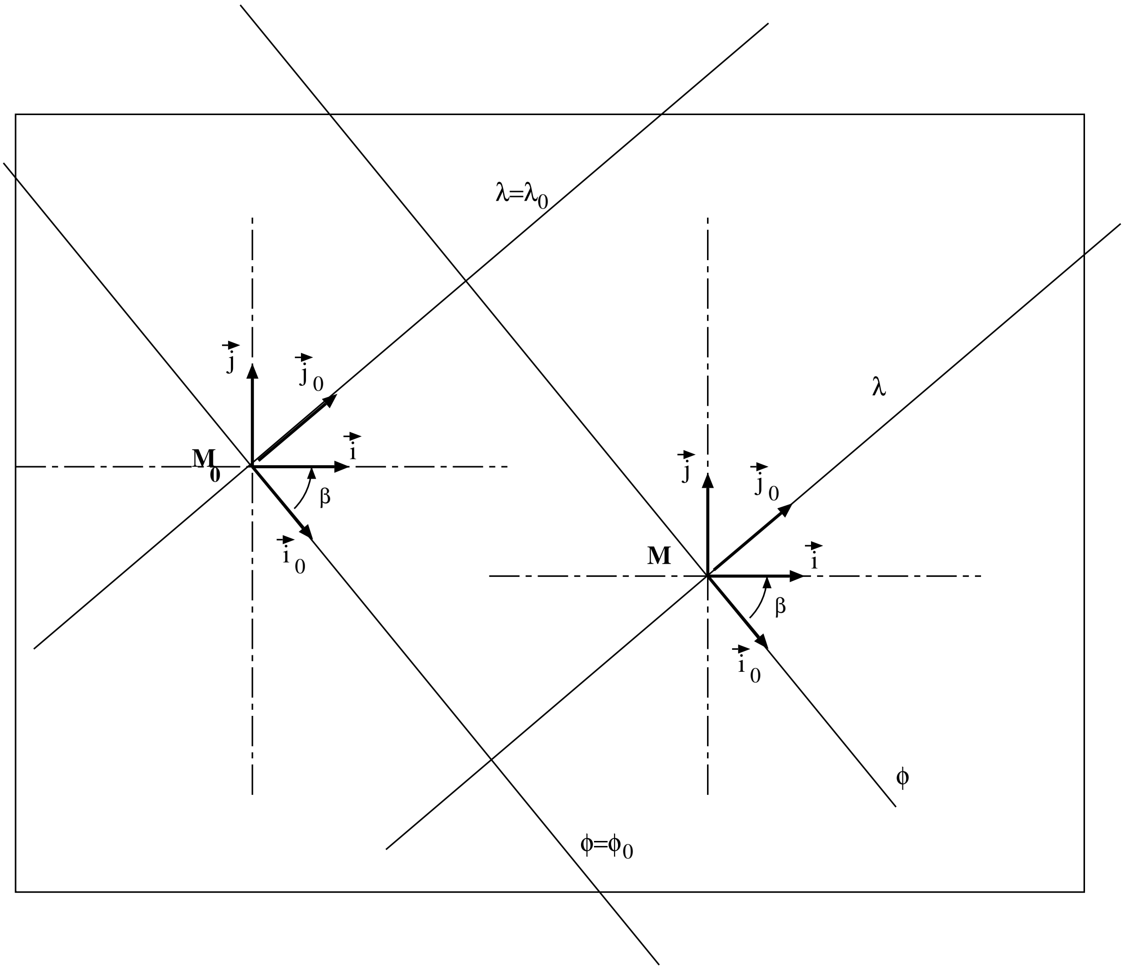


Figure 3.4: Mercator projection: the angle between $\boldsymbol{i}$ and $\boldsymbol{i}_0$ is independent of the position in the domain.

Note that allowing for $K = 0$ in the expression of the map scale factor of the Lambert projection will give exactly the same result. As far as the local metrics is concerned, the only useful informations are therefore m, K, and γ. These quantities will be used in the following equations, without further necessity to distinguish between the three types of projection.

3.2 Calculus in non orthogonal coordinates

In the physical space, the $\widehat{x}$ and $\widehat{z}$ coordinate lines are not orthogonal, because of the underlying topography z_s. Therefore, the coordinate system is not orthogonal and it will be necessary to introduce, beside the Cartesian basis, the covariant and contravariant bases to develop the tensor operators of Chapter 2. In Meso-NH, we follow the ideas of Viviand (1974) and Vinokur (1974), and use alternatively the Cartesian and covariant bases, in order to simplify the formulation of these operators. We now recall the main classical formulae to work with non orthogonal coordinate systems.

3.2.1 Metric coefficients

Let us call x, y, z the local distances on the sphere in the directions $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$.
We will use the classical notations $\widehat{d}_{\star\star}$ for the metric coefficients. Their values in the case of our transformation are given by:

$$\begin{aligned}
\widehat{d}_{xx} &= \frac{\partial x}{\partial \widehat{x}} = \frac{r}{am} \\
\widehat{d}_{yy} &= \frac{\partial y}{\partial \widehat{y}} = \frac{r}{am} \\
\widehat{d}_{zz} &= \frac{\partial z}{\partial \widehat{z}} = 1 - \frac{z_s}{H} \\
\widehat{d}_{zx} &= \frac{\partial z}{\partial \widehat{x}} = \frac{\partial z_s}{\partial \widehat{x}}(1 - \frac{\widehat{z}}{H}) \\
\widehat{d}_{zy} &= \frac{\partial z}{\partial \widehat{y}} = \frac{\partial z_s}{\partial \widehat{y}}(1 - \frac{\widehat{z}}{H}) \\
\widehat{d}_{xz} &= \frac{\partial x}{\partial \widehat{z}} = 0 \\
\widehat{d}_{yz} &= \frac{\partial y}{\partial \widehat{z}} = 0 \\
\widehat{d}_{xy} &= \frac{\partial x}{\partial \widehat{y}} = 0 \\
\widehat{d}_{yx} &= \frac{\partial y}{\partial \widehat{x}} = 0
\end{aligned} \tag{3.9}$$

3.2.2 Covariant basis

The basis of covariant vectors is defined as

$$\boldsymbol{e}_i = \frac{\partial \boldsymbol{r}}{\partial \widehat{x}^i} \tag{3.10}$$

(with $\widehat{x}^i = \widehat{x}$, $\widehat{y}$ or $\widehat{z}$). Those vectors are tangent to the coordinate lines (see Fig. 3.1). They can be projected to the Cartesian basis, resulting in

$$\begin{aligned}
\boldsymbol{e}_1 &= \frac{\partial \boldsymbol{r}}{\partial \widehat{x}} = \widehat{d}_{xx}\boldsymbol{i} + \widehat{d}_{zx}\boldsymbol{k} \\
\boldsymbol{e}_2 &= \frac{\partial \boldsymbol{r}}{\partial \widehat{y}} = \widehat{d}_{yy}\boldsymbol{j} + \widehat{d}_{zy}\boldsymbol{k} \\
\boldsymbol{e}_3 &= \frac{\partial \boldsymbol{r}}{\partial \widehat{z}} = \widehat{d}_{zz}\boldsymbol{k} \qquad (3.11)
\end{aligned}$$

3.2.3 Jacobian

The Jacobian of the transformation from $\widehat{x}, \widehat{y}, \widehat{z}$ to x, y, z is given by

$$\widehat{J} = \boldsymbol{e}_1 \cdot (\boldsymbol{e}_2 \wedge \boldsymbol{e}_3) \qquad (3.12)$$

or

$$\widehat{J} = \widehat{d}_{xx}\widehat{d}_{yy}\widehat{d}_{zz} = \left(\frac{r}{am}\right)^2 (1 - \frac{z_s}{H}) \qquad (3.13)$$

It is the ratio of the volumes in the transformed and physical spaces.

3.2.4 Contravariant basis

The basis of contravariant vectors is defined as

$$\boldsymbol{e}^{\,i} = \boldsymbol{\nabla}\widehat{x}^i \qquad (3.14)$$

These vectors are orthogonal to the surfaces $\widehat{x}^i$ =Const (see Fig. 3.1).
Since $\boldsymbol{e}^{\,i} \cdot \boldsymbol{e}_j = \delta_{ij}$, we also have:

$$\begin{aligned}
\boldsymbol{e}^{\,1} &= \frac{1}{\widehat{J}}(\boldsymbol{e}_2 \wedge \boldsymbol{e}_3) = \frac{1}{\widehat{d}_{xx}}\boldsymbol{i} \\
\boldsymbol{e}^{\,2} &= \frac{1}{\widehat{J}}(\boldsymbol{e}_3 \wedge \boldsymbol{e}_1) = \frac{1}{\widehat{d}_{yy}}\boldsymbol{j} \qquad (3.15) \\
\boldsymbol{e}^{\,3} &= \frac{1}{\widehat{J}}(\boldsymbol{e}_1 \wedge \boldsymbol{e}_2) = \frac{1}{J}\left(\widehat{d}_{xx}\widehat{d}_{yy}\boldsymbol{k} - \widehat{d}_{zx}\widehat{d}_{yy}\boldsymbol{i} - \widehat{d}_{zy}\widehat{d}_{xx}\boldsymbol{j}\right)
\end{aligned}$$

These formulae allow to find the expression of the contravariant components of any vector, as a function of its Cartesian components

$$\begin{aligned}
\boldsymbol{A} &= A^{c1}\boldsymbol{e}_1 + A^{c2}\boldsymbol{e}_2 + A^{c3}\boldsymbol{e}_3 \\
\boldsymbol{A} &= a^1\boldsymbol{i} + a^2\boldsymbol{j} + a^3\boldsymbol{k} \\
A^{c1} &= \frac{a^1}{\widehat{d}_{xx}} \qquad (3.16) \\
A^{c2} &= \frac{a^2}{\widehat{d}_{yy}} \\
A^{c3} &= \frac{a^3}{\widehat{d}_{zz}} - \frac{a^1\widehat{d}_{zx}}{\widehat{d}_{xx}\widehat{d}_{zz}} - \frac{a^2\widehat{d}_{zy}}{\widehat{d}_{yy}\widehat{d}_{zz}}
\end{aligned}$$

3.2.5 Spatial derivatives of the Cartesian basis

We will also use the expression of the spatial derivatives of the vectors of the Cartesian basis, due to the earth sphericity:

$$\begin{aligned}
\frac{\partial \boldsymbol{i}}{\partial \widehat{x}} &= \cos\gamma\,\frac{\widehat{d_{xx}}}{r\cos\varphi}\,(\sin\varphi - K)\,\boldsymbol{j} - \frac{\widehat{d_{xx}}}{r}\,\boldsymbol{k} \\
\frac{\partial \boldsymbol{i}}{\partial \widehat{y}} &= \sin\gamma\,\frac{\widehat{d_{yy}}}{r\cos\varphi}\,(\sin\varphi - K)\,\boldsymbol{j} \\
\frac{\partial \boldsymbol{j}}{\partial \widehat{x}} &= -\cos\gamma\,\frac{\widehat{d_{xx}}}{r\cos\varphi}\,(\sin\varphi - K)\,\boldsymbol{i} \\
\frac{\partial \boldsymbol{j}}{\partial \widehat{y}} &= -\sin\gamma\,\frac{\widehat{d_{yy}}}{r\cos\varphi}\,(\sin\varphi - K)\,\boldsymbol{i} - \frac{\widehat{d_{yy}}}{r}\,\boldsymbol{k} \\
\frac{\partial \boldsymbol{k}}{\partial \widehat{x}} &= \frac{\widehat{d_{xx}}}{r}\,\boldsymbol{i} \\
\frac{\partial \boldsymbol{k}}{\partial \widehat{y}} &= \frac{\widehat{d_{yy}}}{r}\,\boldsymbol{j}
\end{aligned} \tag{3.17}$$

3.2.6 Gradient

For any scalar quantity α,

$$\boldsymbol{\nabla}\alpha = \frac{\partial\alpha}{\partial\widehat{x}}\boldsymbol{e}^{\,1} + \frac{\partial\alpha}{\partial\widehat{y}}\boldsymbol{e}^{\,2} + \frac{\partial\alpha}{\partial\widehat{z}}\boldsymbol{e}^{\,3} \tag{3.18}$$

This vector may be projected on $(\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k})$ resulting in the Cartesian components of the gradient

$$\begin{aligned}
\frac{\partial\alpha}{\partial x} = \boldsymbol{i}\cdot\boldsymbol{\nabla}\alpha &= \frac{1}{\widehat{d_{xx}}}\frac{\partial\alpha}{\partial\widehat{x}} - \frac{\widehat{d_{zx}}}{\widehat{d_{xx}}\widehat{d_{zz}}}\frac{\partial\alpha}{\partial\widehat{z}} \\
\frac{\partial\alpha}{\partial y} = \boldsymbol{j}\cdot\boldsymbol{\nabla}\alpha &= \frac{1}{\widehat{d_{yy}}}\frac{\partial\alpha}{\partial\widehat{y}} - \frac{\widehat{d_{zy}}}{\widehat{d_{yy}}\widehat{d_{zz}}}\frac{\partial\alpha}{\partial\widehat{z}} \\
\frac{\partial\alpha}{\partial z} = \boldsymbol{k}\cdot\boldsymbol{\nabla}\alpha &= \frac{1}{\widehat{d_{zz}}}\frac{\partial\alpha}{\partial\widehat{z}}
\end{aligned} \tag{3.19}$$

3.2.7 Divergence

For any vector quantity $\boldsymbol{A}$, the contravariant components

$$\boldsymbol{A} = A^{c1}\boldsymbol{e}_1 + A^{c2}\boldsymbol{e}_2 + A^{c3}\boldsymbol{e}_3 = (\boldsymbol{A}\cdot\boldsymbol{e}^{\,1})\boldsymbol{e}_1 + (\boldsymbol{A}\cdot\boldsymbol{e}^{\,2})\boldsymbol{e}_2 + (\boldsymbol{A}\cdot\boldsymbol{e}^{\,3})\boldsymbol{e}_3 \tag{3.20}$$

allow to compute easily the divergence

$$\boldsymbol{\nabla}\cdot\boldsymbol{A} = \frac{1}{\widehat{J}}\left(\frac{\partial}{\partial\widehat{x}}(\widehat{J}A^{c1}) + \frac{\partial}{\partial\widehat{y}}(\widehat{J}A^{c2}) + \frac{\partial}{\partial\widehat{z}}(\widehat{J}A^{c3})\right) \tag{3.21}$$

3.2.8 Divergence of a tensor product

In order to avoid generating a large number of Christoffel symbols (quantities involved in the derivatives of the basis vectors), we use the contravariant components of $\boldsymbol{B}$ and the Cartesian components of $\boldsymbol{A}$

$$\begin{aligned}
\boldsymbol{B} &= B^{c\,1}\boldsymbol{e}_1 + B^{c\,2}\boldsymbol{e}_2 + B^{c\,3}\boldsymbol{e}_3 \\
\boldsymbol{A} &= \mathcal{A}^1\boldsymbol{i} + \mathcal{A}^2\boldsymbol{j} + \mathcal{A}^3\boldsymbol{k}
\end{aligned}$$

The result projected on the Cartesian basis reads

$$\begin{aligned}
\boldsymbol{i}\cdot\boldsymbol{\nabla}\cdot(\boldsymbol{B}\otimes\boldsymbol{A}) &= \frac{1}{\widehat{J}}\left(\frac{\partial}{\partial\widehat{x}}(\widehat{J}B^{c\,1}\mathcal{A}^1) + \frac{\partial}{\partial\widehat{y}}(\widehat{J}B^{c\,2}\mathcal{A}^1) + \frac{\partial}{\partial\widehat{z}}(\widehat{J}B^{c\,3}\mathcal{A}^1)\right) \\
&\quad -B^{c\,1}\mathcal{A}^3\frac{\widehat{d_{xx}}}{r} - B^{c\,1}\mathcal{A}^2\cos\gamma\frac{\widehat{d_{xx}}}{r\cos\varphi}(\sin\varphi - K) \\
&\quad -B^{c\,2}\mathcal{A}^2\sin\gamma\frac{\widehat{d_{yy}}}{r\cos\varphi}(\sin\varphi - K) \\
\boldsymbol{j}\cdot\boldsymbol{\nabla}\cdot(\boldsymbol{B}\otimes\boldsymbol{A}) &= \frac{1}{\widehat{J}}\left(\frac{\partial}{\partial\widehat{x}}(\widehat{J}B^{c\,1}\mathcal{A}^2) + \frac{\partial}{\partial\widehat{y}}(\widehat{J}B^{c\,2}\mathcal{A}^2) + \frac{\partial}{\partial\widehat{z}}(\widehat{J}B^{c\,3}\mathcal{A}^2)\right) \\
&\quad +B^{c\,2}\mathcal{A}^3\frac{\widehat{d_{yy}}}{r} + B^{c\,1}\mathcal{A}^1\cos\gamma\frac{\widehat{d_{xx}}}{rcos\varphi}(\sin\varphi - K) \\
&\quad +B^{c\,2}\mathcal{A}^1\sin\gamma\frac{\widehat{d_{yy}}}{rcos\varphi}(\sin\varphi - K) \\
\boldsymbol{k}\cdot\boldsymbol{\nabla}\cdot(\boldsymbol{B}\otimes\boldsymbol{A}) &= \frac{1}{\widehat{J}}\left(\frac{\partial}{\partial\widehat{x}}(JB^{c\,1}\mathcal{A}^3) + \frac{\partial}{\partial\widehat{y}}(\widehat{J}B^{c\,2}\mathcal{A}^3) + \frac{\partial}{\partial\widehat{z}}(\widehat{J}B^{c\,3}\mathcal{A}^3)\right) \\
&\quad -B^{c\,1}\mathcal{A}^1\frac{\widehat{d_{xx}}}{r} - B^{c\,2}\mathcal{A}^2\frac{\widehat{d_{yy}}}{r} \qquad (3.22)
\end{aligned}$$

For each equation, the first line contains the dominant terms, and the second and third lines show the additional terms due to the spatial variation of the vectors of the local basis. These terms are the *curvature terms*. The influence of non-orthogonality is also present through the use of the contravariant components of $\boldsymbol{B}$.

3.2.9 Coriolis force

Both vectors are expressed in basis $(\boldsymbol{i},\boldsymbol{j},\boldsymbol{k})$. We call $f = 2\Omega\sin\varphi$ and $f_\star = 2\Omega\cos\varphi$.

$$\begin{aligned}
2\boldsymbol{\Omega}\wedge\boldsymbol{U} &= (f_\star\cos\gamma w - fv)\boldsymbol{i} \\
&\quad +(fu + f_\star\sin\gamma w)\boldsymbol{j} \\
&\quad -(f_\star\cos\gamma u + f\sin\gamma v)\boldsymbol{k} \qquad (3.23)
\end{aligned}$$

3.3 Model equations in non orthogonal coordinates

3.3.1 Prognostic variables

Given the form of the divergence operator, the most convenient choice for the model prognostic variables is the product of the wind components, potential temperature, and mixing ratios by the dry density of the reference state, *and by the Jacobian of the coordinate system.*

To simplify the notations, we define $\widehat{\rho} = \rho_{dref}\widehat{J}$. The prognostic variables of the model are therefore $\widehat{\rho}u, \widehat{\rho}v, \widehat{\rho}w, \widehat{\rho}\theta, \widehat{\rho}r_*$, and $\widehat{\rho}s_*$.

For the water and passive scalars, the prognostic variable represents therefore (to the extent that ρ_{dref} is a good approximation of ρ_d) *the mass of substance within the grid volume*, which is a very simple and safe quantity to carry in a model.

3.3.2 The contravariant components of the wind

The contravariant components of the velocity vector $\boldsymbol{U} = U^c\boldsymbol{e}_1 + V^c\boldsymbol{e}_2 + W^c\boldsymbol{e}_3$ will be needed to express the advection operator. They may be computed as

$$U^c = \boldsymbol{U}\cdot\boldsymbol{e}^1 \tag{3.24}$$

$$V^c = \boldsymbol{U}\cdot\boldsymbol{e}^2 \tag{3.25}$$

$$W^c = \boldsymbol{U}\cdot\boldsymbol{e}^3 \tag{3.26}$$

Since (u, v) are the horizontal wind components in the basis $(\boldsymbol{i}, \boldsymbol{j})$ we have

$$U^c = \frac{u}{\widehat{d}_{xx}} \tag{3.27}$$

$$V^c = \frac{v}{\widehat{d}_{yy}} \tag{3.28}$$

$$W^c = \frac{w}{\widehat{d}_{zz}} - \frac{u\widehat{d}_{zx}}{\widehat{d}_{xx}\widehat{d}_{zz}} - \frac{v\widehat{d}_{zy}}{\widehat{d}_{yy}\widehat{d}_{zz}} \tag{3.29}$$

This simple computation is performed at the beginning of each model time step.

3.3.3 Momentum equation

$$\begin{aligned}
\frac{\partial}{\partial t}(\widehat{\rho}u) &= -\frac{\partial}{\partial\widehat{x}}(\widehat{\rho}U^c\,u) - \frac{\partial}{\partial\widehat{y}}(\widehat{\rho}V^c\,u) - \frac{\partial}{\partial\widehat{z}}(\widehat{\rho}W^c\,u) \\
&+ \widehat{\rho}uv\frac{\cos\gamma}{r\cos\varphi}(\sin\varphi - K) + \widehat{\rho}v^2\frac{\sin\gamma}{r\cos\varphi}(\sin\varphi - K) \\
&- \widehat{\rho}\frac{uw}{r} - \widehat{\rho}\frac{1}{\widehat{d}_{xx}}\frac{\partial\Phi}{\partial\widehat{x}} + \widehat{\rho}\frac{\widehat{d}_{zx}}{\widehat{d}_{xx}\widehat{d}_{zz}}\frac{\partial\Phi}{\partial\widehat{z}} \\
&- \widehat{\rho}f^*\cos\gamma w + \widehat{\rho}fv + \widehat{\rho}\boldsymbol{F}_v\cdot\boldsymbol{i}
\end{aligned} \tag{3.30}$$

$$
\begin{aligned}
\frac{\partial}{\partial t}(\widehat{\rho} v) &= -\frac{\partial}{\partial \widehat{x}}(\widehat{\rho} U^c\, v) - \frac{\partial}{\partial \widehat{y}}(\widehat{\rho} V^c\, v) - \frac{\partial}{\partial \widehat{z}}(\widehat{\rho} W^c\, v) \\
&- \widehat{\rho} u^2 \frac{\cos\gamma}{r\cos\varphi}(\sin\varphi - K) - \widehat{\rho} uv \frac{\sin\gamma}{r\cos\varphi}(\sin\varphi - K) \\
&- \widehat{\rho}\frac{vw}{r} - \widehat{\rho}\frac{1}{\widehat{d_{yy}}}\frac{\partial\Phi}{\partial\widehat{y}} + \widehat{\rho}\frac{\widehat{d_{zy}}}{\widehat{d_{yy}}\widehat{d_{zz}}}\frac{\partial\Phi}{\partial\widehat{z}} \\
&- \widehat{\rho} f^* \sin\gamma w - \widehat{\rho} f u + \widehat{\rho}\boldsymbol{F}_v \cdot \boldsymbol{j}
\end{aligned} \tag{3.31}
$$

$$
\begin{aligned}
\frac{\partial}{\partial t}(\widehat{\rho} w) &= -\frac{\partial}{\partial \widehat{x}}(\widehat{\rho} U^c\, w) - \frac{\partial}{\partial \widehat{y}}(\widehat{\rho} V^c\, w) - \frac{\partial}{\partial \widehat{z}}(\widehat{\rho} W^c\, w) \\
&+ \widehat{\rho} u^2 \frac{1}{r} + \widehat{\rho} v^2 \frac{1}{r} - \widehat{\rho}\frac{1}{\widehat{d_{zz}}}\frac{\partial\Phi}{\partial\widehat{z}} + \widehat{\rho} g \frac{\theta_v'}{\theta_v} \\
&+ \widehat{\rho} f^*(\sin\gamma v + \cos\gamma u) + \widehat{\rho}\boldsymbol{F}_v \cdot \boldsymbol{k}
\end{aligned} \tag{3.32}
$$

The right hand side of these equations has been rewritten in Chapter 2 as the sum of a dynamical source $\boldsymbol{S}_V$ and the pressure gradient. This is still valid, except that the source now contains the Jacobian. Note that those are still the Cartesian components of $\boldsymbol{S}_V$, i.e. the true horizontal and vertical components of the acceleration.

3.3.4 Thermodynamic equation

$$
\begin{aligned}
\frac{\partial}{\partial t}(\widehat{\rho}\theta) &= -\frac{\partial}{\partial \widehat{x}}(\widehat{\rho} U^c\, \theta) - \frac{\partial}{\partial \widehat{y}}(\widehat{\rho} V^c\, \theta) - \frac{\partial}{\partial \widehat{z}}(\widehat{\rho} W^c\, \theta) \\
&+\widehat{\rho}\left[\frac{R_d + r_v R_v}{R_d}\frac{C_{pd}}{C_{ph}} - 1\right]\frac{\theta}{\Pi_{ref}}\frac{w}{\widehat{d_{zz}}}\frac{\partial\Pi_{ref}}{\partial\widehat{z}} \\
&+\frac{\widehat{\rho}}{\Pi_{ref}C_{ph}}\left[L_m\frac{D(r_i + r_s + r_g + r_h)}{Dt} - L_v\frac{Dr_v}{Dt} + \mathcal{H}\right]
\end{aligned} \tag{3.33}
$$

3.3.5 Water and other scalars

The equations have exactly the same form as the thermodynamic equation, except for the source terms.

3.3.6 Continuity equation

It has the simple form, using the contravariant velocity components

$$
\frac{\partial}{\partial\widehat{x}}(\widehat{\rho}U^c) + \frac{\partial}{\partial\widehat{y}}(\widehat{\rho}V^c) + \frac{\partial}{\partial\widehat{z}}(\widehat{\rho}W^c) = 0. \tag{3.34}
$$

3.3.7 Pressure equation

The divergence operators appearing in the pressure equation may be evaluated with the contravariant components of the vectors $\boldsymbol{\nabla}\Phi$ and $\boldsymbol{S}_V$. The result is an elliptic equation, involving a quasi-Laplacian operator. This is discussed in Chapter 9.

3.4 Degenerated forms

3.4.1 Thin shell approximation

The depth of the atmosphere ($\simeq$ 30 km) is much smaller than the earth radius ($a \simeq$ 6000 km). Therefore, it is customary in meteorology to assume that the atmosphere is a thin shell ($z \ll a$ and $r \simeq a$). However, if this hypothesis is retained without care, the principle of conservation of angular momentum may be violated (Phillips 1966). It is necessary to make the additional hypothesis that the horizontal component of the earth rotation is negligible. In the Meso-NH model, we decided not to retain the thin shell approximation. However, for the purpose of comparison with models making this approximation, we introduce a flag δ_1, taking the value 1 in the general case and 0 in the case of the thin shell approximation.

3.4.2 Cartesian coordinate system

For many purposes, one may want to work in the much simpler Cartesian frame on a plane tangent to the sphere, and neglect the curvature terms (e.g. study of very small scale processes, or idealized studies of meso-scale processes). This may be obtained easily by setting $f = 2\Omega \sin\varphi_0$, $f^* = 2\Omega\cos\varphi_0$, $\widehat{d_{xx}} = 1$, $\widehat{d_{yy}} = 1$, $m = 1$, $r = a = \infty$. Note that contrary to the previous case, we may keep here $f^* \neq 0$ without violating the conservation of angular momentum. In order to obtain this very simple form, we introduce another flag δ_2 in the general system, taking the value 1 for the general case, and 0 for the Cartesian case.

The combination of the thin shell approximation and the Cartesian frame will supply the traditional f-plane approximation (no horizontal component of the earth rotation).

3.4.3 Flagged equations

The δ_1 and δ_2 flags modify the momentum equation in the following way:

$$\begin{aligned}
\frac{\partial}{\partial t}(\widehat{\rho} u) &= -\frac{\partial}{\partial \widehat{x}}(\widehat{\rho} U^c\, u) - \frac{\partial}{\partial \widehat{y}}(\widehat{\rho} V^c\, u) - \frac{\partial}{\partial \widehat{z}}(\widehat{\rho} W^c\, u) \\
&+ \delta_2 \widehat{\rho} uv \frac{\cos\gamma}{r\cos\varphi}(\sin\varphi - K) + \delta_2 \widehat{\rho} v^2 \frac{\sin\gamma}{r\cos\varphi}(\sin\varphi - K) \\
&- \delta_2\delta_1 \widehat{\rho}\frac{uw}{r} - \widehat{\rho}\frac{1}{\widehat{d_{xx}}}\frac{\partial\Phi}{\partial\widehat{x}} + \widehat{\rho}\frac{\widehat{d_{zx}}}{\widehat{d_{xx}}\widehat{d_{zz}}}\frac{\partial\Phi}{\partial\widehat{z}} \\
&- \delta_1 \widehat{\rho} f^* \cos\gamma w + \widehat{\rho} f v + \widehat{\rho}\boldsymbol{F}_v \cdot \boldsymbol{i}
\end{aligned} \tag{3.35}$$

$$\begin{aligned}
\frac{\partial}{\partial t}(\widehat{\rho} v) &= -\frac{\partial}{\partial \widehat{x}}(\widehat{\rho} U^c\, v) - \frac{\partial}{\partial \widehat{y}}(\widehat{\rho} V^c\, v) - \frac{\partial}{\partial \widehat{z}}(\widehat{\rho} W^c\, v) \\
&- \delta_2 \widehat{\rho} u^2 \frac{\cos\gamma}{r\cos\varphi}(\sin\varphi - K) - \delta_2 \widehat{\rho} uv \frac{\sin\gamma}{r\cos\varphi}(\sin\varphi - K) \\
&- \delta_2\delta_1 \widehat{\rho}\frac{vw}{r} - \widehat{\rho}\frac{1}{\widehat{d_{yy}}}\frac{\partial\Phi}{\partial\widehat{y}} + \widehat{\rho}\frac{\widehat{d_{zy}}}{\widehat{d_{yy}}\widehat{d_{zz}}}\frac{\partial\Phi}{\partial\widehat{z}} \\
&- \delta_1 \widehat{\rho} f^* \sin\gamma w - \widehat{\rho} f u + \widehat{\rho}\boldsymbol{F}_v \cdot \boldsymbol{j}
\end{aligned} \tag{3.36}$$

$$\begin{aligned}
\frac{\partial}{\partial t}(\widehat{\rho} w) &= -\frac{\partial}{\partial \widehat{x}}(\widehat{\rho} U^c\, w) - \frac{\partial}{\partial \widehat{y}}(\widehat{\rho} V^c\, w) - \frac{\partial}{\partial \widehat{z}}(\widehat{\rho} W^c\, w) \\
&+ \delta_2\delta_1 \widehat{\rho}\frac{u^2 + v^2}{r} - \widehat{\rho}\frac{1}{\widehat{d_{zz}}}\frac{\partial\Phi}{\partial\widehat{z}} + \widehat{\rho} g \frac{\theta_v'}{\widehat{\theta_v}} \\
&+ \delta_1 \widehat{\rho} f^* (\sin\gamma v + \cos\gamma u) + \widehat{\rho}\boldsymbol{F}_v \cdot \boldsymbol{k}
\end{aligned} \tag{3.37}$$

with

$$\begin{aligned}
\widehat{d_{xx}} &= \frac{a + \delta_1\delta_2 z}{am} \\
\widehat{d_{yy}} &= \frac{a + \delta_1\delta_2 z}{am} \\
r &= a + \delta_1 z \\
\widehat{J} &= \left(\frac{a + \delta_1 z}{am}\right)^2 (1 - \frac{z_s}{H})
\end{aligned} \tag{3.38}$$

The other equations of the model do not involve the flags.

3.5 References

Gal-Chen, T., and R. C. J. Sommerville, 1975: On the use of a coordinate transformation for the solution of the Navier-Stokes equations. *J. Comput. Phys.*, **17**, 209-228.

Phillips, N. A., 1966: The equations of motion for a shallow rotating atmosphere and the ”traditional approximation”. *J. Atmos. Sci*, **23**, 626-628

Leuenberger, D., M. Koller, O. Fuhrer, and C. Schär, 2010: A Generalization of the SLEVE Vertical Coordinate. *Mon. Wea. Rev.,*, **138,** 3683–3689, https://doi.org/10.1175/2010MWR3307.1

Schär, C., D. Leuenberger, O. Fuhrer, D. Lüthi, and C. Girard, 2003: A new terrain-following vertical coordinate formulation for atmospheric prediction models. *Mon. Wea. Rev.*, **130**, 2459-2480

Vinokur, M., 1974: Conservation equations of gas-dynamics in curvilinear coordinate systems. *J. Comp. Phys.*, **14**, 105-125.

Viviand, H., 1974: Formes conservatives des équations de la dynamique des gaz. *La Recherche Aérospatiale*, **1**, 65-66.

Chapter 4

Discretization

Contents

4.1 Stretching

For many reasons, it may be desirable to stretch the computational grid. On the vertical, it is customary to have a better resolution in the planetary boundary layer than in the atmosphere. On the horizontal, the use of stretched coordinates may allow to study the interaction between larger-scale and smaller-scale processes, or simply avoid problems at the lateral boundary. However, if the stretching is not done properly, it may result in a loss of accuracy.
In order to stretch the coordinates, we introduce a new system noted $(\overline{x}, \overline{y}, \overline{z})$, and related to $\widehat{x}, \widehat{y}, \widehat{z}$ by

$$d\overline{x} = \frac{d\widehat{x}}{\mathcal{D}_x(\widehat{x})} \tag{4.1}$$

$$\tag{4.2}$$

$$d\overline{y} = \frac{d\widehat{y}}{\mathcal{D}_y(\widehat{y})} \tag{4.3}$$

$$\tag{4.4}$$

$$d\overline{z} = \frac{d\widehat{z}}{\mathcal{D}_z(\widehat{z})} \tag{4.5}$$

The stretching functions $\mathcal{D}_x(\widehat{x}), \mathcal{D}_y(\widehat{y}), \mathcal{D}_z(\widehat{z})$ are defined independently in the three directions. The new metric coefficients d_{ij} can be computed as

$$\begin{aligned} d_{xx} &= \widehat{d}_{xx}\ \mathcal{D}_x(\widehat{x}) \\ d_{yy} &= \widehat{d}_{yy}\ \mathcal{D}_y(\widehat{y}) \\ d_{zz} &= \widehat{d}_{zz}\ \mathcal{D}_z(\widehat{z}) \\ d_{zx} &= \widehat{d}_{zx}\ \mathcal{D}_x(\widehat{x}) \\ d_{zy} &= \widehat{d}_{zy}\ \mathcal{D}_y(\widehat{y}) \end{aligned} \tag{4.6}$$

$$J = \widehat{J}\mathcal{D}_x(\widehat{x})\ \mathcal{D}_y(\widehat{y})\ \mathcal{D}_z(\widehat{z}) \tag{4.7}$$

and we define

$$\tilde{\rho} = \rho_{dref} J \tag{4.8}$$

In practice, the value of $\overline{x}$ is 1 on the first grid point, 2 to the second grid point, etc.. and $\mathcal{D}_x(\widehat{x})$ is equal to the local value of the grid distance on the surface of projection. Therefore d_{xx} and d_{zz} are precisely the horizontal and vertical grid distances in the physical space, and J is the volume of the grid box in the physical space. Thus, $\tilde{\rho}$ is the mass of dry air within each grid box, for the reference state.

The model equations in this new system have exactly the same expression as found in the previous chapter, substituting ($\widehat{x}$,$\widehat{y}$, $\widehat{z}$) by $(\overline{x}, \overline{y}, \overline{z})$. This reads

- Continuity equation

$$\frac{\partial}{\partial \overline{x}}(\tilde{\rho} U^c\ ) + \frac{\partial}{\partial \overline{y}}(\tilde{\rho} V^c\ ) + \frac{\partial}{\partial \overline{z}}(\tilde{\rho} W^c\ ) = 0. \tag{4.9}$$

- Momentum equation

$$
\begin{aligned}
\frac{\partial}{\partial t}(\tilde{\rho} u) \;=\;& -\frac{\partial}{\partial \overline{x}}(\tilde{\rho} U^c\, u) - \frac{\partial}{\partial \overline{y}}(\tilde{\rho} V^c\, u) - \frac{\partial}{\partial \overline{z}}(\tilde{\rho} W^c\, u) \\
&+ \delta_2 \tilde{\rho} uv \frac{\cos\gamma}{r\cos\varphi}(\sin\varphi - K) + \delta_2 \tilde{\rho} v^2 \frac{\sin\gamma}{r\cos\varphi}(\sin\varphi - K) \\
&- \delta_2 \delta_1 \tilde{\rho} \frac{uw}{r} - \tilde{\rho} \frac{1}{d_{xx}} \frac{\partial \Phi}{\partial \overline{x}} + \tilde{\rho} \frac{d_{zx}}{d_{xx} d_{zz}} \frac{\partial \Phi}{\partial \overline{z}} \\
&- \delta_1 \tilde{\rho} f^* \cos\gamma w + \tilde{\rho} f v + \tilde{\rho} \boldsymbol{F}_v \cdot \boldsymbol{i}
\end{aligned}
\tag{4.10}
$$

$$
\begin{aligned}
\frac{\partial}{\partial t}(\tilde{\rho} v) \;=\;& -\frac{\partial}{\partial \overline{x}}(\tilde{\rho} U^c\, v) - \frac{\partial}{\partial \overline{y}}(\tilde{\rho} V^c\, v) - \frac{\partial}{\partial \overline{z}}(\tilde{\rho} W^c\, v) \\
&- \delta_2 \tilde{\rho} u^2 \frac{\cos\gamma}{r\cos\varphi}(\sin\varphi - K) - \delta_2 \tilde{\rho} uv \frac{\sin\gamma}{r\cos\varphi}(\sin\varphi - K) \\
&- \delta_2 \delta_1 \tilde{\rho} \frac{vw}{r} - \tilde{\rho} \frac{1}{d_{yy}} \frac{\partial \Phi}{\partial \overline{y}} + \tilde{\rho} \frac{d_{zy}}{d_{yy} d_{zz}} \frac{\partial \Phi}{\partial \overline{z}} \\
&- \delta_1 \tilde{\rho} f^* \sin\gamma w - \tilde{\rho} f u + \tilde{\rho} \boldsymbol{F}_v \cdot \boldsymbol{j}
\end{aligned}
\tag{4.11}
$$

$$
\begin{aligned}
\frac{\partial}{\partial t}(\tilde{\rho} w) \;=\;& -\frac{\partial}{\partial \overline{x}}(\tilde{\rho} U^c\, w) - \frac{\partial}{\partial \overline{y}}(\tilde{\rho} V^c\, w) - \frac{\partial}{\partial \overline{z}}(\tilde{\rho} W^c\, w) \\
&+ \delta_2 \delta_1 \tilde{\rho} \frac{u^2 + v^2}{r} - \tilde{\rho} \frac{1}{d_{zz}} \frac{\partial \Phi}{\partial \overline{z}} + \tilde{\rho} g \frac{\theta'_v}{\overline{\theta}_v} \\
&+ \delta_1 \tilde{\rho} f^* (\sin\gamma v + \cos\gamma u) + \tilde{\rho} \boldsymbol{F}_v \cdot \boldsymbol{k}
\end{aligned}
\tag{4.12}
$$

- Thermodynamic equation

$$
\begin{aligned}
\frac{\partial}{\partial t}(\tilde{\rho} \theta) \;=\;& -\frac{\partial}{\partial \overline{x}}(\tilde{\rho} U^c\, \theta) - \frac{\partial}{\partial \overline{y}}(\tilde{\rho} V^c\, \theta) - \frac{\partial}{\partial \overline{z}}(\tilde{\rho} W^c\, \theta) \\
&+\tilde{\rho} \left[\frac{R_d + r_v R_v}{R_d} \frac{C_{pd}}{C_{ph}} - 1 \right] \frac{\theta}{\Pi_{ref}} \frac{w}{d_{zz}} \frac{\partial \Pi_{ref}}{\partial \overline{z}} \\
&+\frac{\tilde{\rho}}{\Pi_{ref} C_{ph}} \left[L_m \frac{D(r_i + r_s + r_g + r_h)}{Dt} - L_v \frac{D r_v}{Dt} + \mathcal{H} \right]
\end{aligned}
\tag{4.13}
$$

4.2 Location of the variables on the grid

We use a C-grid in the Arakawa convention (Mesinger and Arakawa 1976), both on the horizontal and on the vertical.

The horizontal grid is shown on Fig. 4.1. The "mass" points, located at the center of each grid element, are noted by $\bigcirc$, the u points by $\triangleright$, and the v points by $\triangle$. ζ is the vertical component of the vorticity. The corners of the shaded square all have the same integer coordinates (i, j).

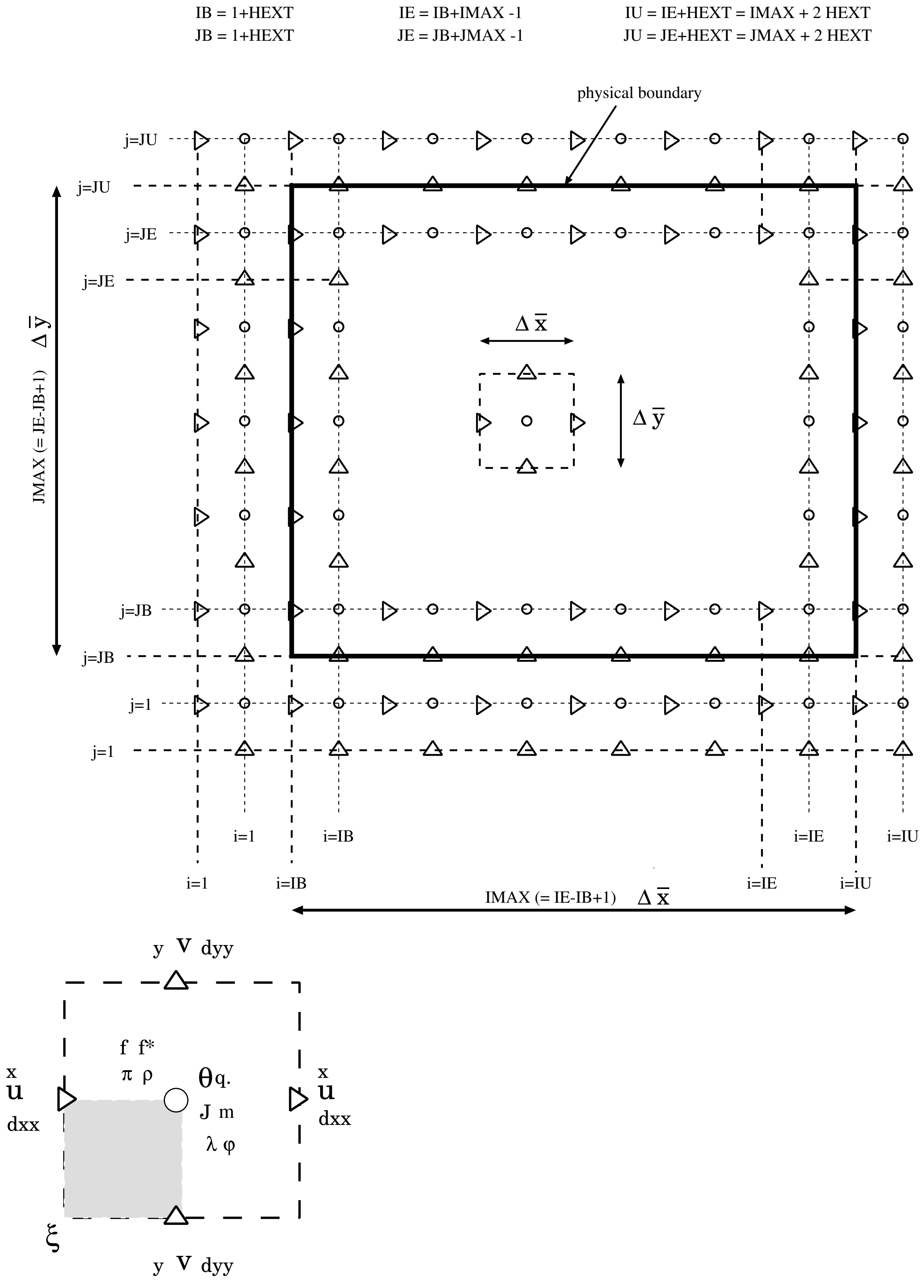


Figure 4.1: Discretization on the horizontal

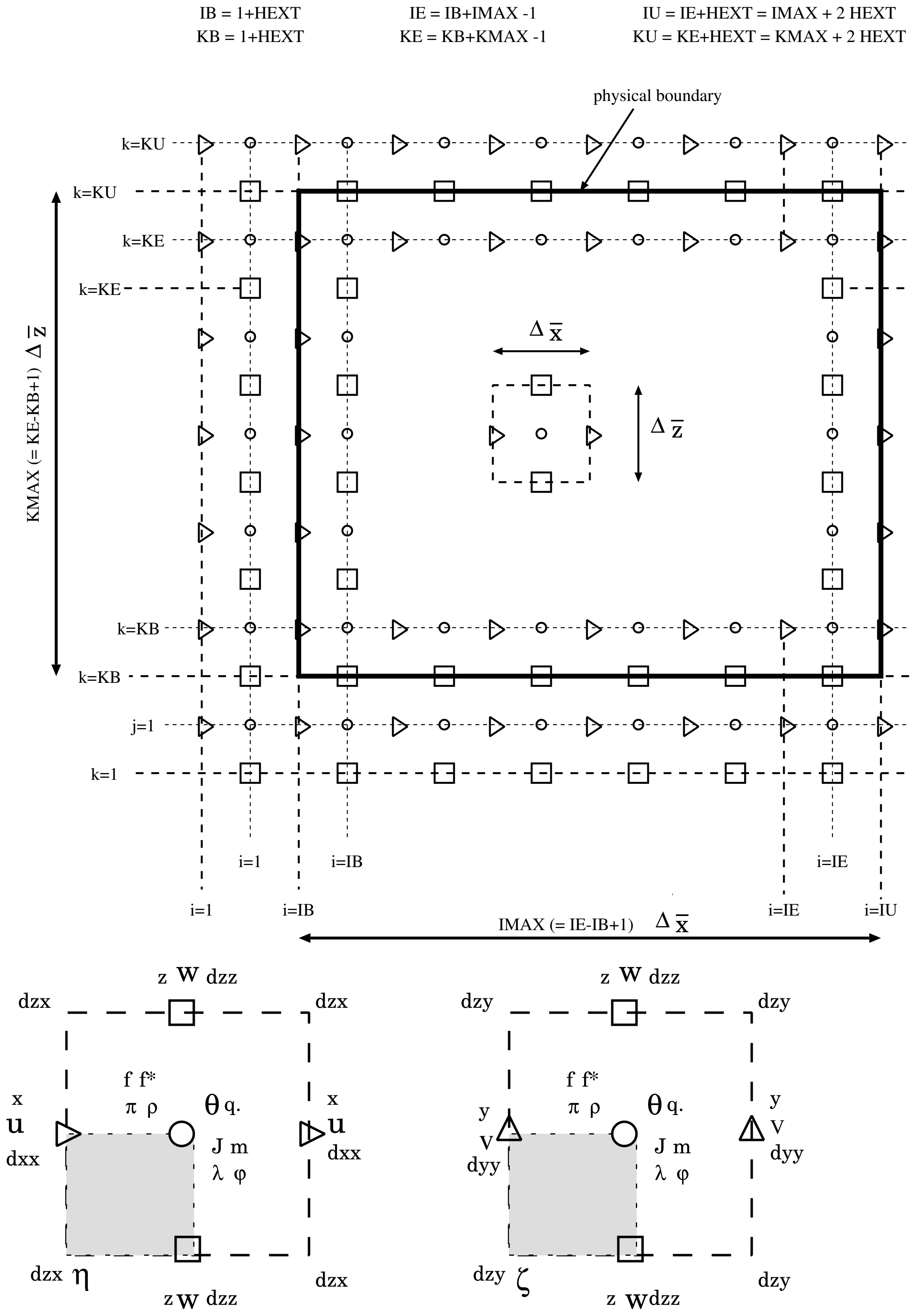


Figure 4.2: Discretization on the vertical

The vertical grid is shown on Fig. 4.2. Again the mass points are noted by ○, the u points by ▷, the v points by △, and the w points by □. ξ and η are the vorticity components along x and y. The corners of the shaded square all have the same integer coordinates.
$\widehat{x}, \widehat{y}, \widehat{z}$ and the metric coefficients d_{xx}, d_{yy}, d_{zz} are respectively located on u, v, and w points. On the other hand, d_{zx} and d_{zy} are located on the η and ξ vorticity points. The Jacobian J is located on the mass points.
The orography $z_s(i,j)$ is defined on the lowest w point.

4.3 Schuman operators

The discretization is based on second-order finite differences, and two point averages. We will adopt the traditional notations of Schuman: $\delta_x\alpha$, $\delta_y\alpha$ and $\delta_z\alpha$ for finite differences in the directions $\overline{x}$, $\overline{y}$, and $\overline{z}$; $\overline{\alpha}^x$, $\overline{\alpha}^y$, and $\overline{\alpha}^z$ for averaging in the same directions.
For instance, $\delta_x\alpha(i,j,k) = (\alpha(i{+}1,j,k) {-} \alpha(i,j,k))$ or $(\alpha(i,j,k) {-} \alpha(i{-}1,j,k))$, depending of the location of the variable α, $\overline{\alpha}^x(i,j,k) = (\alpha(i{+}1,j,k) {+} \alpha(i,j,k))/2$ or $(\alpha(i,j,k) {+} \alpha(i{-}1,j,k))/2$, and similarly for the other operators.
For the time evolution, we use the classical leap-frog explicit scheme. The operator of time derivation also follows Schuman notation:

$$\delta_t \overline{\alpha(i,j,k,t)}^t = \frac{\alpha(i,j,k,t+\Delta t) - \alpha(i,j,k,t-\Delta t)}{2\Delta t} \tag{4.14}$$

4.4 Grid generation

The parameters needed to generate the grids are:

- the projection parameters λ_0, φ_0, β and K.
- the latitude and longitude of one mass point to start the integration, i.e. the point $(i = 1, j = 1)$.
- the series of values of $\widehat{x}$ at u points, $\widehat{y}$ at v points, and $\widehat{z}$ at w points.

The following computations are performed:
1) The positions $\widehat{x}_m(i)$, $\widehat{y}_m(j)$, $\widehat{z}_m(k)$ of the mass points are deduced:

$$\begin{aligned} \widehat{x}_m(i) &= 0.5\,[\widehat{x}(i) + \widehat{x}(i+1)] \\ \widehat{y}_m(j) &= 0.5\,[\widehat{y}(j) + \widehat{y}(j+1)] \\ \widehat{z}_m(k) &= 0.5\,[\widehat{z}(k) + \widehat{z}(k+1)] \end{aligned} \tag{4.15}$$

2) The latitude and longitude of each mass point is retrieved:

Case of the polar stereographic or Lambert projection

Applying the general equations to the point (1,1) of the grid, one gets

$$\begin{aligned} \rho(1,1) &= \frac{a}{K}(\cos\varphi_0)^{1-K}(1+\sin\varphi_0)^K \left(\frac{\cos\varphi(1,1)}{1+\sin\varphi(1,1)}\right)^K \\ \gamma(1,1) &= K(\lambda(1,1) - \lambda_0) - \beta \end{aligned} \tag{4.16}$$

The pole coordinates may be deduced as

$$\begin{aligned} X_p &= \widehat{x}_m(1) - \rho(1,1)\sin\gamma(1,1) \\ Y_p &= \widehat{y}_m(1) + \rho(1,1)\cos\gamma(1,1) \end{aligned} \tag{4.17}$$

The longitude of any mass point (i,j) follows:

$$\lambda(i,j) = \frac{\beta + \arctan\left(-\dfrac{\widehat{x}_m(i) - X_p}{\widehat{y}_m(j) - Y_p}\right) + \epsilon\pi}{K} + \lambda_0 \quad [2\pi] \tag{4.18}$$

with

$$\begin{aligned} \epsilon &= 0 \text{ if } \widehat{y}_m(j) - Y_p \leq 0 \\ \epsilon &= 1 \text{ if } \widehat{y}_m(j) - Y_p > 0 \end{aligned}$$

while the latitudes are given by

$$\begin{aligned} \rho(i,j)^2 &= \left[\widehat{x}_m(i) - X_p\right]^2 + \left[\widehat{y}_m(j) - Y_p\right]^2 \\ \varphi(i,j) &= \frac{\pi}{2} - \arccos\left\{\frac{\left[a(\cos\varphi_0)^{1-K}\right]^{2/K}(1+\sin\varphi_0)^2 - \left[K^2\rho(i,j)^2\right]^{1/K}}{\left[a(\cos\varphi_0)^{1-K}\right]^{2/K}(1+\sin\varphi_0)^2 + \left[K^2\rho(i,j)^2\right]^{1/K}}\right\} \end{aligned} \tag{4.19}$$

Case of the Mercator projection

The latitudes and longitudes of the mass points are given by

$$\begin{aligned} \lambda(i,j) &= \frac{[\widehat{x}_m(i) - \widehat{x}_m(1)]\cos\gamma + [\widehat{y}_m(j) - \widehat{y}_m(1)]\sin\gamma}{a\cos\varphi_0} + \lambda(1,1) \\ \varphi(i,j) &= -\frac{\pi}{2} + 2\arctan\left\{\exp\left(-\frac{[\widehat{x}_m(1) - \widehat{x}_m(i)]\sin\gamma + [\widehat{y}_m(j) - \widehat{y}_m(1)]\cos\gamma}{a\cos\varphi_0}\right.\right. \\ &\quad \left.\left. + \ln\left[\tan\left(\frac{\pi}{4} + \frac{\varphi(1,1)}{2}\right)\right]\right)\right\} \end{aligned} \tag{4.20}$$

3) Once the latitudes and longitudes are known, the map scale factor, the angle γ, and the Coriolis parameters are easily computed for each mass point:

$$m = \left(\frac{\cos\varphi_0}{\cos\varphi}\right)^{1-K}\left(\frac{1+\sin\varphi_0}{1+\sin\varphi}\right)^K \tag{4.21}$$

$$\gamma = K(\lambda - \lambda_0) - \beta$$

$$f = 2\Omega\sin\varphi \tag{4.22}$$

$$f_* = 2\Omega\cos\varphi \tag{4.23}$$

4.5 Metric coefficients and Jacobian

$$
\begin{aligned}
z &= z_s + \widehat{z}\left(1 - \frac{z_s}{H}\right) && (4.24)\\
d_{xx} &= \overline{\left[\frac{\overline{(a+\delta_1\delta_2 z)}^z}{a}\,\frac{1}{m}\,\delta_x\widehat{x}\right]}^x && (4.25)\\
d_{yy} &= \overline{\left[\frac{\overline{(a+\delta_1\delta_2 z)}^z}{a}\frac{1}{m}\,\delta_y\widehat{y}\right]}^y && (4.26)\\
d_{zx} &= \delta_x z && (4.27)\\
d_{zy} &= \delta_y z && (4.28)\\
d_{zz} &= \overline{\delta_z z}^z && (4.29)\\
J &= \left(\frac{\overline{(a+\delta_1\delta_2 z)}^z}{a}\right)^2 \left(\frac{1}{m}\right)^2 \delta_x\widehat{x}\;\delta_y\widehat{y}\;\delta_z z && (4.30)
\end{aligned}
$$

4.6 Contravariant velocity components

$$
\begin{aligned}
\overline{\tilde{\rho}}^x U^c &= \overline{\tilde{\rho}}^x\,\frac{u}{d_{xx}} && (4.31)\\
\overline{\tilde{\rho}}^y V^c &= \overline{\tilde{\rho}}^y\,\frac{v}{d_{yy}} && (4.32)\\
\overline{\tilde{\rho}}^z W^c &= \frac{1}{d_{zz}}\left[\overline{\tilde{\rho}}^z w - \overline{\left(\overline{\left(\frac{\overline{\tilde{\rho}}^x u}{d_{xx}}\right)}^z d_{zx}\right)}^x - \overline{\left(\overline{\left(\frac{\overline{\tilde{\rho}}^y v}{d_{yy}}\right)}^z d_{zy}\right)}^y\right] && (4.33)
\end{aligned}
$$

4.7 Time derivatives

$$
\begin{aligned}
\frac{\partial}{\partial t}(\tilde{\rho}u) &\Longrightarrow \delta_t\left[\overline{\left(\overline{\tilde{\rho}}^x u\right)}^t\right] && (4.34)\\
\frac{\partial}{\partial t}(\tilde{\rho}v) &\Longrightarrow \delta_t\left[\overline{\left(\overline{\tilde{\rho}}^y v\right)}^t\right] && (4.35)\\
\frac{\partial}{\partial t}(\tilde{\rho}w) &\Longrightarrow \delta_t\left[\overline{\left(\overline{\tilde{\rho}}^z w\right)}^t\right] && (4.36)
\end{aligned}
$$

4.8 Advection

$$-\frac{\partial}{\partial \overline{x}}(\tilde{\rho} U^c\, u) \quad\Longrightarrow\quad -\delta_x\left[\overline{\left(\overline{\tilde{\rho}}^{x} U^c\right)}^{x}\,\overline{u}^{x}\right] \tag{4.37}$$

$$-\frac{\partial}{\partial \overline{y}}(\tilde{\rho} V^c\, u) \quad\Longrightarrow\quad -\delta_y\left[\overline{\left(\overline{\tilde{\rho}}^{y} V^c\right)}^{x}\,\overline{u}^{y}\right] \tag{4.38}$$

$$-\frac{\partial}{\partial \overline{z}}(\tilde{\rho} W^c\, u) \quad\Longrightarrow\quad -\delta_z\left[\overline{\left(\overline{\tilde{\rho}}^{z} W^c\right)}^{x}\,\overline{u}^{z}\right] \tag{4.39}$$

$$-\frac{\partial}{\partial \overline{x}}(\tilde{\rho} U^c\, v) \quad\Longrightarrow\quad -\delta_x\left[\overline{\left(\overline{\tilde{\rho}}^{x} U^c\right)}^{y}\,\overline{v}^{x}\right] \tag{4.40}$$

$$-\frac{\partial}{\partial \overline{y}}(\tilde{\rho} V^c\, v) \quad\Longrightarrow\quad -\delta_y\left[\overline{\left(\overline{\tilde{\rho}}^{y} V^c\right)}^{y}\,\overline{v}^{y}\right] \tag{4.41}$$

$$-\frac{\partial}{\partial \overline{z}}(\tilde{\rho} W^c\, v) \quad\Longrightarrow\quad -\delta_z\left[\overline{\left(\overline{\tilde{\rho}}^{z} W^c\right)}^{y}\,\overline{v}^{z}\right] \tag{4.42}$$

$$-\frac{\partial}{\partial \overline{x}}(\tilde{\rho} U^c\, w) \quad\Longrightarrow\quad -\delta_x\left[\overline{\left(\overline{\tilde{\rho}}^{x} U^c\right)}^{z}\,\overline{w}^{x}\right] \tag{4.43}$$

$$-\frac{\partial}{\partial \overline{y}}(\tilde{\rho} V^c\, w) \quad\Longrightarrow\quad -\delta_y\left[\overline{\left(\overline{\tilde{\rho}}^{y} V^c\right)}^{z}\,\overline{w}^{y}\right] \tag{4.44}$$

$$-\frac{\partial}{\partial \overline{z}}(\tilde{\rho} W^c\, w) \quad\Longrightarrow\quad -\delta_z\left[\overline{\left(\overline{\tilde{\rho}}^{z} W^c\right)}^{z}\,\overline{w}^{z}\right] \tag{4.45}$$

4.9 Curvature terms

$$\delta_2\tilde{\rho}uv\frac{\cos\gamma}{r\cos\varphi}(\sin\varphi - K) \quad\Longrightarrow\quad \delta_2\overline{\tilde{\rho}}^{x}u\,\overline{\left(\overline{v}^{y}\frac{\cos\gamma}{\cos\varphi}(\sin\varphi - K)\frac{1}{\overline{a+\delta_1 z}^{z}}\right)}^{x} \tag{4.46}$$

$$\delta_2\tilde{\rho}v^2\frac{\sin\gamma}{r\cos\varphi}(\sin\varphi - K) \quad\Longrightarrow\quad \delta_2\overline{\left(\overline{\left(\overline{\tilde{\rho}}^{y}v^2\right)}^{y}\frac{\sin\gamma}{\cos\varphi}(\sin\varphi - K)\frac{1}{\overline{a+\delta_1 z}^{z}}\right)}^{x} \tag{4.47}$$

$$-\delta_2\delta_1\tilde{\rho}\frac{uw}{r} \quad\Longrightarrow\quad -\delta_2\delta_1\overline{\tilde{\rho}}^{x}u\,\overline{\left(\frac{\overline{w}^{z}}{\overline{a+\delta_1 z}^{z}}\right)}^{x} \tag{4.48}$$

$$-\delta_2\tilde{\rho}u^2\frac{\cos\gamma}{r\cos\varphi}(\sin\varphi - K) \quad\Longrightarrow\quad -\delta_2\overline{\left(\overline{\left(\overline{\tilde{\rho}}^{x}u^2\right)}^{x}\frac{\cos\gamma}{\cos\varphi}(\sin\varphi - K)\frac{1}{\overline{a+\delta_1 z}^{z}}\right)}^{y} \tag{4.49}$$

$$-\delta_2\tilde{\rho}uv\frac{\sin\gamma}{r\cos\varphi}(\sin\varphi - K) \quad\Longrightarrow\quad -\delta_2\overline{\tilde{\rho}}^{y}v\,\overline{\left(\overline{u}^{x}\frac{\sin\gamma}{\cos\varphi}(\sin\varphi - K)\frac{1}{\overline{a+\delta_1 z}^{z}}\right)}^{y} \tag{4.50}$$

$$-\delta_2\delta_1\tilde{\rho}\frac{vw}{r} \quad\Longrightarrow\quad -\delta_1\delta_2\delta_1\overline{\tilde{\rho}}^{y}v\,\overline{\left(\frac{\overline{w}^{z}}{\overline{a+\delta_1 z}^{z}}\right)}^{y} \tag{4.51}$$

$$\delta_2\delta_1\tilde{\rho}\frac{u^2+v^2}{r} \quad\Longrightarrow\quad \delta_2\delta_1\overline{\left(\frac{\overline{\left(\overline{\tilde{\rho}}^{x}u^2\right)}^{x}+\overline{\left(\overline{\tilde{\rho}}^{y}v^2\right)}^{y}}{\overline{a+\delta_1 z}^{z}}\right)}^{z} \tag{4.52}$$

4.10 Coriolis force

$$
\begin{aligned}
-\delta_1 \tilde{\rho} f \cos\gamma w &\implies -\delta_1 \overline{(\tilde{\rho} f \cos\gamma \overline{w}^z)}^x && (4.53)\\
\tilde{\rho} f v &\implies \overline{(\tilde{\rho} f \overline{v}^y)}^x && (4.54)\\
-\delta_1 \tilde{\rho} f \sin\gamma w &\implies -\delta_1 \overline{(\tilde{\rho} f \sin\gamma \overline{w}^z)}^y && (4.55)\\
-\tilde{\rho} f u &\implies -\overline{(\tilde{\rho} f \overline{u}^x)}^y && (4.56)\\
\delta_1 \tilde{\rho} f (\sin\gamma v + \cos\gamma u) &\implies \delta_1 \overline{(\tilde{\rho} f \cos\gamma \overline{u}^x)}^z \ + \ \delta_1 \overline{(\tilde{\rho} f \sin\gamma \overline{v}^y)}^z && (4.57)
\end{aligned}
$$

4.11 Pressure gradient

$$
\begin{aligned}
-\tilde{\rho} \frac{1}{d_{xx}} \frac{\partial \Phi}{\partial \overline{x}} &\implies -\overline{\tilde{\rho}}^x \frac{\delta_x \Phi}{d_{xx}} && (4.58)\\
\tilde{\rho} \frac{d_{zx}}{d_{xx} d_{zz}} \frac{\partial \Phi}{\partial \overline{z}} &\implies \overline{\tilde{\rho}}^x \frac{1}{d_{xx}} \overline{\left(\overline{\left(\frac{\delta_z \Phi}{d_{zz}} \right)}^x d_{zx} \right)}^z && (4.59)\\
-\tilde{\rho} \frac{1}{d_{yy}} \frac{\partial \Phi}{\partial \overline{y}} &\implies -\overline{\tilde{\rho}}^y \frac{\delta_y \Phi}{d_{yy}} && (4.60)\\
\tilde{\rho} \frac{d_{zy}}{d_{yy} d_{zz}} \frac{\partial \Phi}{\partial \overline{z}} &\implies \overline{\tilde{\rho}}^y \frac{1}{d_{yy}} \overline{\left(\overline{\left(\frac{\delta_z \Phi}{d_{zz}} \right)}^y d_{zy} \right)}^z && (4.61)\\
-\tilde{\rho} \frac{1}{d_{zz}} \frac{\partial \Phi}{\partial \overline{z}} &\implies -\overline{\tilde{\rho}}^z \frac{\delta_z \Phi}{d_{zz}} && (4.62)
\end{aligned}
$$

4.12 Buoyancy term

$$
\tilde{\rho} g \frac{\theta'_v}{\overline{\theta}_v} \implies g \overline{\left(\tilde{\rho} \frac{\theta'_v}{\overline{\theta}_v} \right)}^z \qquad (4.63)
$$

4.13 Thermodynamic equation

$$
\begin{aligned}
\delta_t \left[\overline{(\tilde{\rho}\theta)}^t \right] \quad = \quad & -\delta_x \left[\overline{(\tilde{\rho}\theta)}^x U^c \right] - \delta_y \left[\overline{(\tilde{\rho}\theta)}^y V^c \right] - \delta_z \left[\overline{(\tilde{\rho}\theta)}^z W^c \right] \\
& + \tilde{\rho} \left[\frac{R_d + r_v R_v}{R_d} \frac{C_{pd}}{C_{ph}} - 1 \right] \frac{\theta}{\Pi_{ref}} \overline{\frac{w}{d_{zz}} \delta_z \Pi_{ref}}^z \\
& + \frac{\tilde{\rho}}{\Pi_{ref} C_{ph}} \left[L_m \frac{D(r_i + r_s + r_g + r_h)}{Dt} - L_v \frac{D r_v}{Dt} + \mathcal{H} \right] \qquad (4.64)
\end{aligned}
$$

4.14 Continuity equation

$$
\delta_x \left[\overline{\tilde{\rho}}^x U^c \right] + \delta_y \left[\overline{\tilde{\rho}}^y V^c \right] + \delta_z \left[\overline{\tilde{\rho}}^z W^c \right] = 0 \qquad (4.65)
$$

4.15 Pressure equation

For convenience, the discretization of the pressure equation is described in Chapter 9.

4.16 References

Mesinger, F., and A. Arakawa, 1976: Numerical methods used in atmospheric models. GARP Publications Series No. 17, WMO/ICSU Joint Organizing Committee, 64 pp.

Chapter 5

Lateral Boundary Conditions

Contents

The lateral boundary conditions (l.b.c. hereafter) of Meso-NH have been designed to offer various possibilities:

- different types of l.b.c. formulations (cyclic, rigid wall or open)
- coupling with large-scale fields provided by the ARPEGE or ECMWF models (either analyses or forecasts)
- two-way interactive gridnesting (detailed in the following chapter)

5.1 Principles and equations

5.1.1 Cyclic boundary condition (CLBCX or CLBCY = 'CYCL')

This widely used boundary condition is most simple to prescribe. For instance, giving the periodicity length L_x (corresponding to the model domain width) for the x-direction, the conditions imposed to all model variables α are

$$\alpha(x, y, z) = \alpha(x + L_x, y, z). \tag{5.1}$$

It should be noted that this option implies some specific treatments not only in the routine taking care of l.b.c., but at various places in the model. First, the normal velocity component at the physical boundaries is explicitly predicted by the equations. Second, the pressure solver accounts for this hypothesis by using complex FFTs, whereas for other types of l.b.c. the solver uses cosine FFTs. In fact the domain is physically unbounded in that case, as cyclic boundary condition assumes that the physical domain is infinitely large.

5.1.2 Rigid wall lateral boundary condition ('WALL')

For a free-slip rigid wall, a mirror type boundary condition is assumed. At lateral boundaries, the conditions imposed on the model become:

$$\boldsymbol{U} \cdot \boldsymbol{n} \;=\; 0 \tag{5.2}$$

where $\boldsymbol{n}$ is a horizontal unit vector normal to the lateral boundary. It results that normal velocity components are zero at the boundary, whereas all others variables are symmetric relative to the boundary,

$$\frac{\partial \alpha}{\partial n} = 0. \tag{5.3}$$

5.1.3 Wave-radiation open boundary ('OPEN')

There exists several variants of radiative, or open boundary conditions in the literature. For instance, some authors apply the wave-radiation boundary condition to all prognostic variables (Xue and Thorpe 1991), whereas others apply it only to the normal velocity component (Klemp and Wilhelmson 1978). For complex cases with ambient wind shear or with intense convection, the second approach works better.

For the present version of Meso-NH, we choose the simplest formulation, that gives from our experience, satisfactory results. The prognostic variables are separated in two groups: on one hand, the scalars, including the tangential velocity components; on the other hand the normal wind.

- **Case of scalars**

  To compute the advection, the basic need is the prescription of the flux at the boundary for each scalar. This is done depending on the sign of the normal velocity component.

  - *For outflow boundaries:* fluxes normal to the boundary are extrapolated from the interior using one-sided upstream differencing.

  - *For inflow boundaries:* each scalar is taken as its large-scale value LB at that boundary location.

- **Case of normal wind**

  For open l.b.c., normal velocity components u_n are computed by using a general Sommerfeld equation proposed by Carpenter (1982):

  $$\frac{\partial u_n}{\partial t} = \left(\frac{\partial u_n}{\partial t}\right)_{LB} - C^* \left(\frac{\partial u_n}{\partial x} - \left(\frac{\partial u_n}{\partial x}\right)_{LB}\right) \tag{5.4}$$

  where the subscript LB stands for large-scale value of the field and C^* denotes the phase speed of the perturbation field $u_n - (u_n)_{LB}$. The large scale gradient $(\partial u_n/\partial x)_{LB}$ and the time evolution $(\partial u_n/\partial t)_{LB}$ are specified by the coupling model. For idealized simulations including no larger-scale effects, they are of course set to zero.

  This formulation allows waves coming from the interior of the model domain to pass out freely through the boundary with minimal reflection. It also allows the large scale flow to force the evolution of the inner domain at large time scales. The last term is a relaxation that has been added to avoid a slow drift of the solution away from the LB field.

  Sophisticated methods may be used to evaluate the phase speed C^*, such as proposed by Orlanski (1976). As a first step however, we use the simple method proposed by Klemp and Wilhelmson (1978), reading

  $$C^* = u_n + C, \tag{5.5}$$

  where C is an adjustable, constant phase speed (20 to 50 ms^{-1} typically).

- **Pressure function**

  Prescribing the normal velocity components u_n at the boundaries, owing to the use of a Sommerfeld equation or another assumption (rigid wall one), infers the knowledge of the normal component of the gradient of the pressure function. The result is an elliptic equation for the pressure equation, with Neuman type boundary conditions when 'wall' or 'open' l.b.c. are used. This is discussed in Chapter 7.

- **Adjustment for mass conservation**

  Since the anelastic continuity equation

  $$\frac{\partial}{\partial \overline{x}}(\tilde{\rho} U^c\ ) + \frac{\partial}{\partial \overline{y}}(\tilde{\rho} V^c\ ) + \frac{\partial}{\partial \overline{z}}(\tilde{\rho} W^c\ ) = 0. \tag{5.6}$$

  is enforced throughout the domain, it is necessary to guarantee that it is still verified at the model domain scale. As the upper and lower boundaries are assumed to be rigid walls, the total mass flux through the lateral boundaries should be zero. Due to the use of the general Sommerfeld equation (Eq. 5.4), this requirement is not insured by open lateral boundary conditions. An adjustment is thus necessary for this l.b.c. type (see section 2.4).

5.1.4 Combinations of different types of l.b.c.

The three types of l.b.c. can be specified independently for each lateral boundary, with the only obvious limitation that if the cyclic boundary condition is chosen for one lateral boundary, the opposing side b.c. must also be cyclic. Note in particular that it is possible to mix a wall condition on one side and an open condition of the other side in the same direction. The combination of these different types of l.b.c. offers a wide range of geometrical configurations, such as a close tank, a channel, etc...

5.1.5 The coupling and external boundary conditions

One of the most difficult problems inherent to limited area models is the specification of the boundary conditions, that should achieve two partially contradictory goals:

1. evacuate through boundaries the physical (gravity...) and numerical waves generated by the LAM,
2. prescribe the large-scale evolution provided by the **coupling** models or analysis.

The wave-radiation open l.b.c are well designed to solve the first problem, but also to partly treat the coupling, owing to the use of modification proposed by Carpenter (1982). Another strategy of coupling used in most hydrostatic models, is the flow relaxation scheme (Davies 1976). Prognostic variables α in a marginal zone near the boundaries are forced to relax towards the large-scale fields LB on a time scale $1/K$, zero at the boundaries and inwards decreasing, using:

$$\frac{\partial \alpha}{\partial t} = \epsilon(\alpha) - K(\alpha - \alpha_{LB}) \tag{5.7}$$

where $\epsilon(\alpha)$ represents the sources terms of the α prognostic equation.
In practice, to treat the coupling the user has 2 possibilities:

1. either using the wave-radiation open l.b.c ('OPEN' version),
2. or activating the lateral sponge zones corresponding to the flow relaxation scheme (see section 6.3 of this book). To get this treatment the user must also force the phase speed to zero ($C^* = 0$). In that case the damping coefficient at the boundaries F_{SZ}^{max} must be maximum (relative to the time step). As tested by I. Mallet the shape of the variation of the damping coefficient in the rim zone may be important to reduce reflections.

In practice, the coupling can be performed by a combination of the wave-radiation open l.b.c with the flow relaxation scheme.
Concerning the large-scale fields LB, they are available on MNH files provided by the user at the segment simulation beginning. The Meso-NH code reads each CouPLing FILE [CPLFILE(n)] which must span the whole time duration of the segment. The current large-scale fields are linearly interpolated at each time step between the closest large-scale states given by the coupling files [CPLFILE(n) and CPLFILE(n+1)].

5.2 Discretization and implementation

5.2.1 Grid structure

Figure 5.1 shows the horizontal grid structure, including the outer points. By convention, the physical boundary of the simulation domain is shown by the thick line and corresponds to localization of the velocity components normal to the boundary. To implement the boundary conditions, it is convenient to define EXTra grid points outside the physical domain, both Horizontally and Vertically

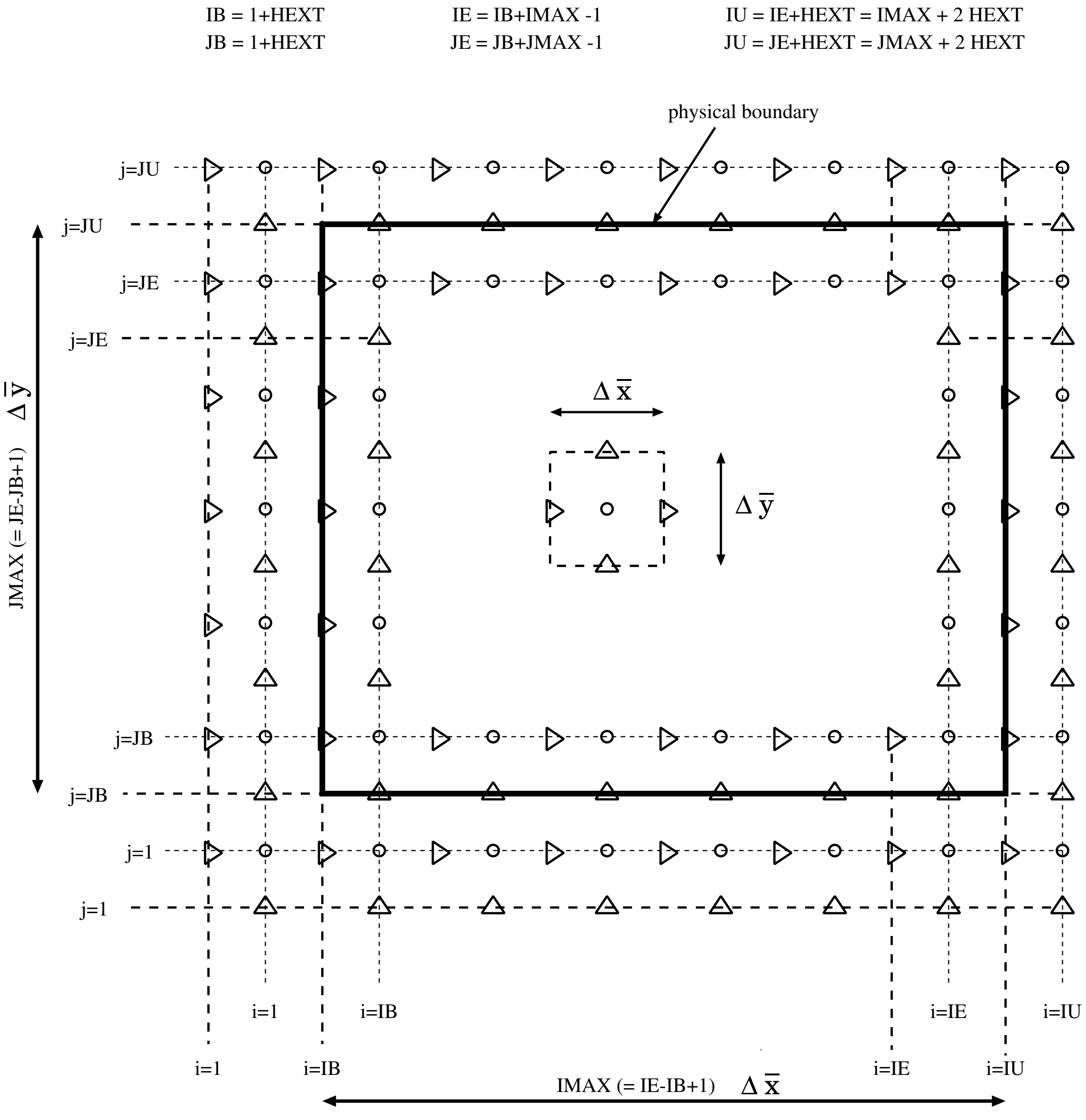


Figure 5.1: Horizontal grid structure for HEXT=1 outside the computation domain

(variables HEXT and VEXT respectively). The number of extra points in the direction normal to the boundary is parameterized. At present time, only the ”one extra point” case (HEXT=VEXT=1) is implemented, although provision has been made in many places in the code to facilitate further implementation of the more general option.

The strategy for coding the l.b.c. has been guided by the idea that the advection and diffusion terms should be calculated in the corresponding routines, without making any particular cases at the boundaries. The adequate values in the boundary zone, are therefore prepared by specialized routines (BOUNDARIES and RAD_BOUND). We take advantage of the fact that the computation of advection uses values at t, and the computation of diffusion uses values at $t-1$: the routine BOUNDARIES and RAD_BOUND are called only once per time step, and they set the boundary points to the values needed by the advection scheme for variables at t, and the values needed by the diffusion scheme for variables at $t-1$. Note that these values are, in general, different.

5.2.2 Cyclic l.b.c. (’CYCL’)

In case of cyclic l.b.c., the periodic lengths L_x and L_y involved in Eq. (5.1) correspond to the horizontal sizes of the simulation domain outlined by the thick line on Fig. 1. For instance, at the $\overline{x}$ boundaries, the cyclic l.b.c. are written for the three velocity components and the temperature:

```
DO JEXT=1,JPHEXT
  PRUT  (IIB-JEXT,:,:)   = PRUT  (IIE+1-JEXT,:,:)    !  Left side
  PRVT  (IIB-JEXT,:,:)   = PRVT  (IIE+1-JEXT,:,:)    !  =========
  PRWT  (IIB-JEXT,:,:)   = PRWT  (IIE+1-JEXT,:,:)
  PRTHT (IIB-JEXT,:,:)   = PRTHT (IIE+1-JEXT,:,:)

  PRUT  (IIE+JEXT,:,:)   = PRUT  (IIB-1+JEXT,:,:)    !  Right side
  PRVT  (IIE+JEXT,:,:)   = PRVT  (IIB-1+JEXT,:,:)    !  ==========
  PRWT  (IIE+JEXT,:,:)   = PRWT  (IIB-1+JEXT,:,:)
  PRTHT (IIE+JEXT,:,:)   = PRTHT (IIB-1+JEXT,:,:)
END DO
```

Note that the I prefix is used above in all indexes to conform to the DOCTOR norm.

5.2.3 Rigid wall l.b.c. (’WALL’)

For instance, at the left $\overline{x}$ boundary, the rigid wall l.b.c. reads for the three velocity components and the temperature:

```
PRUT  (IIB     ,:,:)   = 0.
DO JEXT=1,JPHEXT
  PRUT  (IIB-JEXT,:,:)   = -999                      !  not used
  PRVT  (IIB-JEXT,:,:)   = PRVT  (IIB-1+JEXT,:,:)
  PRWT  (IIB-JEXT,:,:)   = PRWT  (IIB-1+JEXT,:,:)
  PRTHT (IIB-JEXT,:,:)   = PRTHT (IIB-1+JEXT,:,:)
END DO
```

5.2.4 Wave-radiation open boundary ('OPEN')

a. Case of scalars (in routine BOUNDARIES)

For prognostic variables, except the wind component normal to the lateral boundary (hereafter subscript b), all source terms can be computed as for points inside the domain, except for advection and diffusion in the direction normal to the boundary, where the computations involve a point outside of the physical domain.

- **For outflow boundary at time t and $t - \Delta t$:** gradients normal to the boundary are extrapolated from the interior using one-sided upstream differencing. In fact it is equivalent to extrapolating each scalar at the location of the first extra point (hereafter subscript $b + 1$).

  $$\alpha^n_{b+1} = 2\alpha^n_b - \alpha^n_{b-1} \tag{5.8}$$

  For instance for the left $\overline{x}$ boundary and for variables at time t, it gives for the two tangential wind components and the temperature:

```
WHERE ( PRUT(IIB,:,:) <= 0. )                !  OUTFLOW condition

  PRVT  (IIB-1,:,:)   = 2.*PRVT  (IIB,:,:)   -PRVT  (IIB+1,:,:)
  PRWT  (IIB-1,:,:)   = 2.*PRWT  (IIB,:,:)   -PRWT  (IIB+1,:,:)
  PRTHT (IIB-1,:,:)   = 2.*PRTHT (IIB,:,:)   -PRTHT (IIB+1,:,:)
```

- **For inflow boundary at time t and $t - \Delta t$:** each scalar α^n (α^{n-1}) is taken as its large-scale value α^n_{LB} (α^{n-1}_{LB}) at the boundary location.

  $$\alpha^n_{b+1} = \alpha^n_{LB\,b+1} \tag{5.9}$$

  For instance for the left $\overline{x}$ boundary at time t, it gives for the two tangential wind components and the temperature:

```
ELSEWHERE                                    !  INFLOW  condition
  PVT  (IIB-1,:,:) = ZLBXVT   (1,:,:)
  PWT  (IIB-1,:,:) = ZLBXWT   (1,:,:)
  PTHT (IIB-1,:,:) = ZLBXTHT  (1,:,:)
```

b. Case of normal wind (in routine RAD_BOUND)

A semi-implicit numerical scheme is used to discretize the radiative equation (5.4), as recommended by Orlanski (1976). This reads

$$u^{n+1}_{n_b} = \frac{1 - r_L - K\Delta t}{1 + r_L + K\Delta t} u^{n-1}_{n_b} + \frac{2r_L}{1 + r_L + K\Delta t} u^n_{n_{b-1}} + \frac{2\Delta t}{1 + r_L + K\Delta t}\left(F^n_{LB} + K\, u^n_{LB_b}\right), \tag{5.10}$$

where n is the time level, and $r_L = C^*\, \Delta t/\Delta x$ is the normalized phase velocity, and F^n_{LB} the LB forcing term of Eq. (5.4). It should be noted that r_L must be positive to be outwards and results in a stable numerical scheme.

c. Adjustment for mass conservation (in routine MASS_LEAK)

The adjustment procedure simply consists in the following. First, compute the total mass flux through lateral boudaries by integrating the continuity equation (Eq. 5.6) over the whole domain of simulation:

$$LEAK = \left[\sum_{j=JB}^{JE} \sum_{k=KB}^{KE} \frac{1}{d_{xx}} \overline{\tilde{\rho}}^{x} u \; d\overline{y} d\overline{z} \right]_{i=IB}^{IE+1} + \left[\sum_{i=IB}^{IE} \sum_{k=KB}^{KE} \frac{1}{d_{yy}} \overline{\tilde{\rho}}^{y} v \; d\overline{x} d\overline{z} \right]_{j=JB}^{JE+1} \tag{5.11}$$

Second, as it is not strictly equal to zero, all normal wind components are corrected of the same value u_{stop} to eliminate this leak of mass (positive or negative at left or right lateral boundaries respectively).

$$u_{stop} = \frac{LEAK}{LINMASS} \tag{5.12}$$

where LINMASS is the mass per unit length in the direction normal to the open lateral boundaries (i.e. $\int \int_{S_{open}} \rho ds$).

$$\begin{aligned} LINMASS = & \left[\sum_{j=JB}^{JE} \sum_{k=KB}^{KE} \frac{1}{d_{xx}} \overline{\tilde{\rho}}^{x} \; d\overline{y} d\overline{z} \right]_{i=IB} + \left[\sum_{j=JB}^{JE} \sum_{k=KB}^{KE} \frac{1}{d_{xx}} \overline{\tilde{\rho}}^{x} \; d\overline{y} d\overline{z} \right]_{i=IE+1} \\ + & \left[\sum_{i=IB}^{IE} \sum_{k=KB}^{KE} \frac{1}{d_{yy}} \overline{\tilde{\rho}}^{y} \; d\overline{x} d\overline{z} \right]_{j=JB} + \left[\sum_{i=IB}^{IE} \sum_{k=KB}^{KE} \frac{1}{d_{yy}} \overline{\tilde{\rho}}^{y} \; d\overline{x} d\overline{z} \right]_{j=JB+1} \end{aligned} \tag{5.13}$$

The horizontal wind correction u_{stop} is applied only to open l.b., and LINMASS only accounts for these boundaries.

5.3 References

Carpenter, K. M., 1982: Note on the paper ”Radiation conditions for lateral boundaries of limited area numerical models”. *Quart. J. Roy. Meteor. Soc.*, **110**, 717-719.

Davies, H. C., 1976: A lateral boundary formulation for multi-level prediction models. *Quart. J. Roy. Meteor. Soc.*, **102**, 405-418.

Klemp, J. B. and R. B. Wilhelmson, 1978: The simulation of three-dimensional convective storm dynamics. *J. Atmos. Sci.*, **35**, 1070-1096.

Orlanski, I., 1976: A simple boundary condition for unbounded hyperbolic flows. *J. Comput. Phys.*, **21**, 251-269.

Redelsperger, J. L. and J. P. Lafore, 1988: A three-dimensional simulation of a tropical squall-line: Convective organization and thermodynamic vertical transport. *J. Atmos. Sci.*, **45**, 1334-1356.

Xue, M. and A. J. Thorpe, 1991: A mesoscale numerical model using nonhydrostatic pressure-based sigma coordinate equations: Model experiments with dry mountain flows. *Quart. J. Roy. Meteor. Soc.*, **119**, 1168-1185.

Chapter 6

Grid Nesting

Contents

6.1 Principles

There are many problems in atmospheric fluid dynamic in which higher spatial resolution is required in only a limited portion of the computation domain. The two-way interactive grid nesting of Clark and Farley (1984) is implemented in the Meso-NH code with some modifications to allow one to focus on such desired regions and obtain higher spatial resolution and greater computational efficiency (Stein et al. 2000).
The main principle is to execute in parallel several simulations on a nested grid whose a possible structure is illustrated in Fig. 6.1. The only restriction concerns the temporal and spatial resolution

ratio between models, that should be an integer. In the present example we have 4 nested models, with spatial resolution ratios of 2 and 4 between model 1 and model 2, and between model 2 and model 3 or 4, respectively.
The interactions between each fine mesh (FM) model and its coarse mesh (CM) model are treated through either one- or two-way interactive nesting.

- In the simpler one-way approach, only CM waves are allowed to enter and affect the FM model. It is simply performed by the use of interpolation operators **I** to derived boundary conditions of the FM model from its CM model.
- In the two-way approach, waves resolved by the FM model can also affect the CM model solution. It is done by operators **S** averaging FM data down to the CM resolution.

6.2 Basic choices

The following choices have been made:

- The nesting is restricted to the horizontal directions. Thus all models have the same vertical resolution and level numbering.
- The resolution ratio in the x- and y-direction can be different.
- Differing temporal resolutions are allowed (Fig. 6.1).
- A same CM model can monitor several FM models as illustrated by Fig. 6.1, where model 2 monitors models 3 and 4.
- No relative translation speed is allowed between CM and FM models.
- All models have the same x- and y-axis orientation, and work therefore in the same geographic projection system.
- As developed later, the FM topography averaged at the CM scale must be equal to the one of the CM model.
- Different levels of coupling are possible between models:
  level 2 for the two-way interaction of all variables,
  level 1 for the one-way interaction,
  level 0 for the no-interaction case (i.e. model is independent from the others).

6.3 Averaging and interpolation procedures

The original grid-nesting method proposed by Clark and Farley (1984) have been implemented with two major modifications concerning the averaging and interpolation procedures. First we preferred to inject the FM model results into the CM model through a relaxation term instead of a simple substitution. It allows to strictly respect the anelastic constraint independently of the averaging operator, and to get a fully consistent pressure field. Second the Clark and Farley interpolation is replaced by an algebraic interpolator, to avoid discontinuities from one grid box

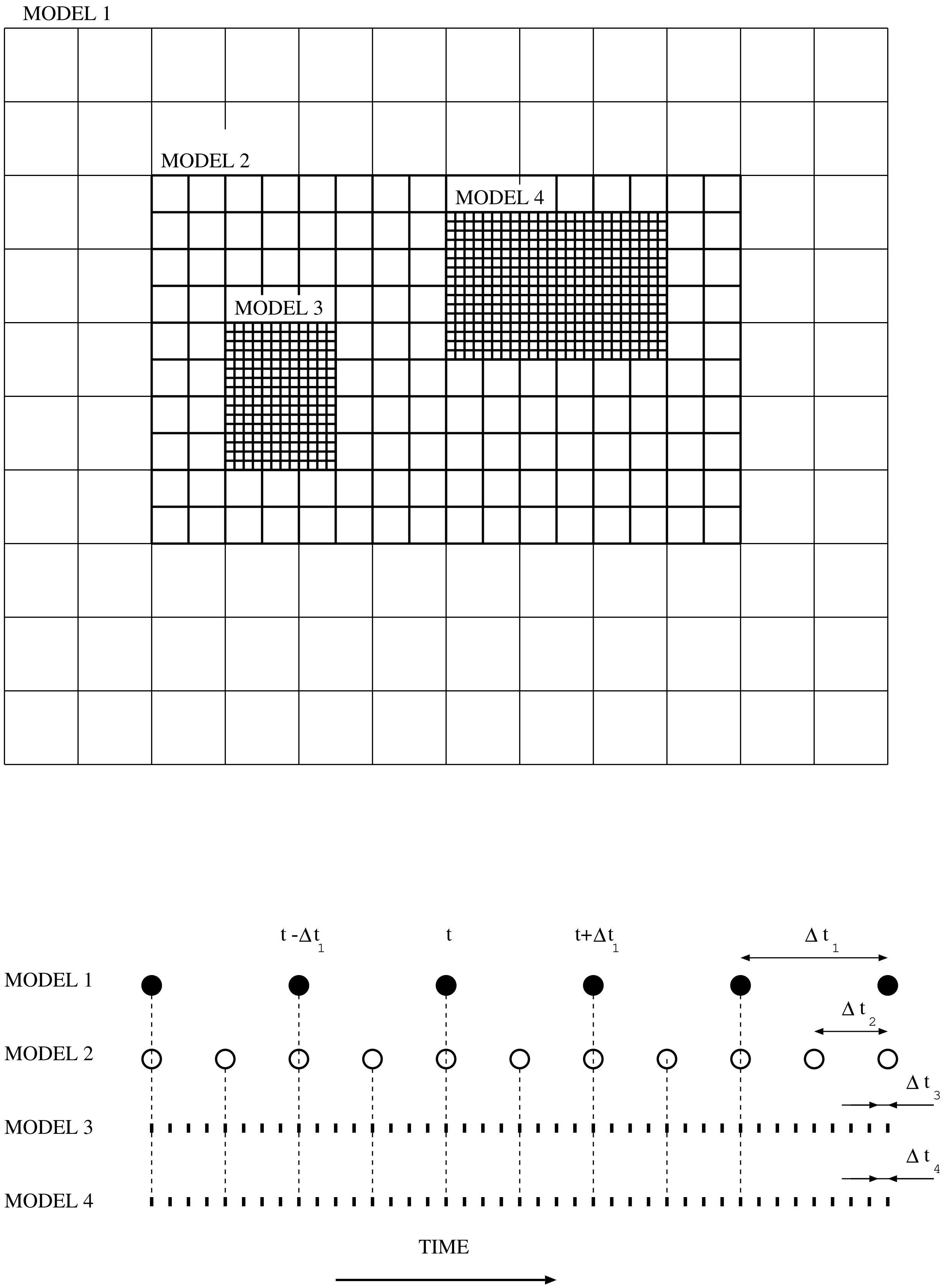


Figure 6.1: An example of nested grid-mesh horizontal and temporal structure.

to the next one. Contrary to results of Clark and Farley, our validation tests showed that the reversibility condition of Kurihara et al. (1979) between averaging and interpolation operators was not strictly necessary in our case.

6.3.1 Interpolation operators from coarse to fine meshes

The Bikhardt spatial interpolation

This algebraic interpolator (cited by de Floriani and Dettori 1981) uses 16 points. The interpolation formula at a point of coordinate (a,b) can be written as:

$$g(a,b) \; = \sum_{i=1}^{4} \sum_{j=1}^{4} f_{ij}\, B_i(a)\, B_j(b) \tag{6.1}$$

where f_{ij} corresponds to values at points (i,j) (Fig. 2) and the B_i are the following third order polynomials:

$$\begin{array}{rll} B_1 = & -0.5t^3 & +\, t^2 - 0.5t \\ B_2 = & +1.5t^3 & -2.5t^2 + 1 \\ B_3 = & -1.5t^3 & +\, 2t^2 + 0.5t \\ B_4 = & & +0.5t^2(t-1) \end{array} \tag{6.2}$$

obeying the properties $B_i(0) = 0$ except $B_2(0) = 1$ and $B_i(1) = 0$ except $B_2(1) = 1$, so that the interpolation surface passes exactly through the 4 most inner points (2,2), (2,3), (3,2) and (3,3). This Bikhardt operator is used to horizontally interpolate on $\overline{z}$ surface the CM models variables and LS fields to obtain the corresponding LB and LS fields of the FM model. Finally a linear vertical interpolation is performed on the LB and LS fields to account for the topography differences between CM and FM models.

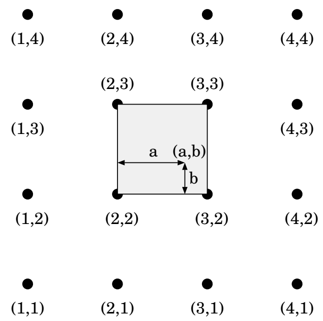


Figure 6.2: Configuration of points involved in the Bikhardt interpolation. The interpolation is valid in the shaded area.

Temporal interpolation

In case of differing time steps, the previous spatial interpolating is performed only when the two models are synchronized in time. In between a linear temporal interpolation is used to get LB and LS fields at the current time. In practice it is done by storing the LB and LS tendencies to allow its forward temporal integration.

Lateral boundaries conditions and other treatments

The previous interpolations allow to get the LB fields necessary to treat lateral boundaries as described in Chapter 5. As for a single model run different types of l.b.c. formulations (cyclic, rigid wall or open) can be used. For instance for open l.b.c. normal velocities are treated owing to the use of a modified Sommerfeld equation already presented in Chapter 5.

$$\frac{\partial u_n}{\partial t} = \left(\frac{\partial u_n}{\partial t}\right)_{LB} - C^* \left(\frac{\partial u_n}{\partial x} - \left(\frac{\partial u_n}{\partial x}\right)_{LB}\right) - K\ (u_n - u_{nLB}) \tag{6.3}$$

Also a lateral sponge zone may be applied to smoothly relax FM fields towards CM corresponding fields stored in the LB fields.

The LS fields previously prepared are eventually used for the numerical diffusion and the top absorbing layer (Chapter 8).

6.3.2 Averaging operators

Define Ψ to be a CM variable and ψ to be the equivalent variable in the FM. We define the following top hat averaging operators S, S_u and S_v for variables located at scalar, u and v points respectively:

$$S(\psi) = \frac{\sum_{i=1}^{X_{ratio}} \sum_{j=1}^{Y_{ratio}} (\tilde{\rho}_{FM}\psi)_{i,j}}{\sum_{i=1}^{X_{ratio}} \sum_{j=1}^{Y_{ratio}} (\tilde{\rho}_{FM})_{i,j}} \tag{6.4}$$

$$S_u(u) = \frac{\sum_{j=1}^{Y_{ratio}} \left(\overline{\tilde{\rho}_{FM}}^{x} u\right)_j}{\sum_{j=1}^{Y_{ratio}} \left(\overline{\tilde{\rho}_{FM}}^{x}\right)_j} \tag{6.5}$$

$$S_v(v) = \frac{\sum_{i=1}^{X_{ratio}} \left(\overline{\tilde{\rho}_{FM}}^{y} v\right)_i}{\sum_{i=1}^{X_{ratio}} \left(\overline{\tilde{\rho}_{FM}}^{y}\right)_i} \tag{6.6}$$

where $X_{ratio} = \dfrac{\Delta \overline{x}}{\Delta \overline{X}}$ and $Y_{ratio} = \dfrac{\Delta \overline{y}}{\Delta \overline{Y}}$ are the resolution ratio on the conformal plan in the x- and y-direction respectively.

The variables Ψ of the CM model are relaxed towards the averaged $S(\psi)$ values obtained from the FM model in the entire overlapping domain. The corresponding additional source for any prognostic variable Ψ is

$$S_{2W} = -K_{2W} \left[\Psi^{t-\Delta T} - S(\psi^{t-\Delta T})\right] \tag{6.7}$$

where subscript $2W$ stands for two-way interactive grid nesting and K_{2W} is the relaxation coefficient set by default to $1/(4\Delta T)$ with ΔT being the CM model time step. The mass continuity remains satisfied owing to the pressure solver passage although the averaging operators (6.4, 6.5 and 6.6) do not strictly guarantee this property.

6.4 Implementation of the nesting

In a first stage the grid-nesting technics has been implemented on a Cray 90, owing to the use of the Cray macrotasking facility to execute each model as a specific task. Nevertheless this approach appears to have several disadvantages; first macrotasking is specific to Cray computers, second the

efficiency of the nesting parallelization is weak, as each task is dependent from the others, third the model synchronization is quite delicate to code.
We finally preferred a ”sequential” approach, consisting in running at the same time only one model for only one time step. The coarser resolution models are run first. In case of multiprocessor run, each model time integration is distributed and performed on each processor.

6.5 Preparation of a nested simulation (spawning)

In order to conduct a grid-nesting simulation, each FM model must be first initialized by its CM model. This operation is performed by a program named SPAWNING, designed to generate initial fields by horizontal interpolation of fields provided by a given Meso-NH file. The input CM model for this interpolation task, will be named model 1, whereas the model 2 will refer to the one to be spawned (CM model). The main possibilities and limitations of the SPAWNING program are:

- The model 2 horizontal domain must be included in the model 1 domain.
- The resolution ratio between model 2 and 1 must be an integer, and can be different in the $\bar{x}$ and $\bar{y}$ directions.
- As regards other characteristics, the two domains remain identical, such as vertical grid, geometry or orientation.

NB: A different program, part of the PREP_REAL_CASE facilities (see the *”Initial fields for real case”* Chapter), can be used on the FM-File generated by the spawning, to change the vertical grid and modify the surface fields, including finer scale information.

6.5.1 Horizontal grid configuration

Figure 6.3 shows the horizontal grid structure of the models 1 and 2. The physical boundary of the two simulation domains (thick line on Fig. 6.3) corresponds to localization of the velocity component normal to the boundaries. The model 2 horizontal grid is defined by 3 input parameters:

1. the resolution ratios between models in the x and y-directions (DXRATIO and DYRATIO respectively) which must be integer,
2. the position of model 2 (XOR,YOR) relative to model 1 (see Fig. 6.3)
3. and the number of points in each horizontal direction.

The only constraint is that the number of points must be a multiple of the resolution ratio for each horizontal direction.

6.5.2 Interpolation operators from fine to coarse meshes

The basic task of the spawning program is to perform horizontal interpolations (at constant $\bar{z}$ level). Two different interpolation operators are used:

1. a simple linear interpolator, mainly to interpolate the $\widehat{x}$ and $\widehat{z}$ coordinates,
2. and the Bikhardt interpolation previously described for most variables.

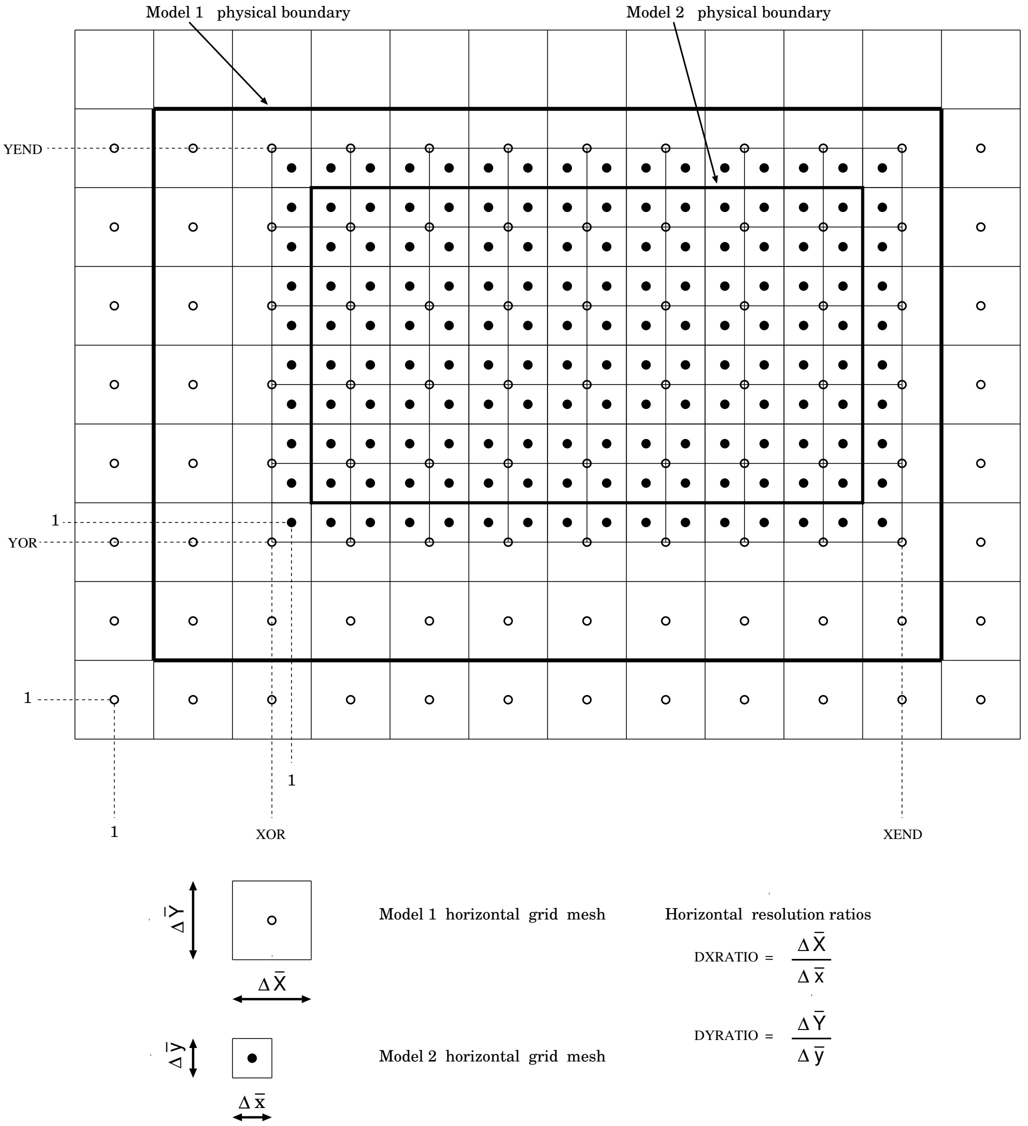


Figure 6.3: Horizontal structure of the grids for a resolution ratio of 2 between models 1 and 2.

6.5.3 Specific treatments for spawning

Grid structure and configuration

After initialization of model 1 from the corresponding FM-File, the program first initializes the model 2 configuration and spatio-temporal grid structure. The model configurations are identical except for the radiation which is not treated by the spawning, and the lateral boundary conditions which are deduced from the model 2 domain position. Concerning grid structure, the topography z_s is interpolated by the Bikhardt operator, whereas the coordinates are linearly interpolated. Some corrections are applied to z_s depending on the land-sea mask and on the model 1 topography, to be consistent, to avoid non-zero altitude at sea level, and to keep negative altitudes over land where it is the case at the model 1 scale.

3D fields

Each of the 3D prognostic or diagnostic fields of model 1 are interpolated independently, with correction of negative values for variables which must be, by definition, positive. Finally the pressure solver is used to enforce the anelastic constraint, accounting for new surface conditions due to the interpolated topography.

Surface fields

The interpolation of surface variables is more complicated, mainly due to the existence of non-physical values (flagged), soil characteristic over the sea for instance, and of discontinuities (at the sea coast). The method consists in first replacing the model 1 flagged values by physical values of the nearest point, then the interpolation is performed, and finally the flagged values are restored. Several corrections are also necessary to avoid overshoots, and to enforce certain constraints (consistency between the sea-land-lake masks or the clay-sand-silt indexes). Using such methods it is easy to perform a simulation at finer scale keeping the same surface characteristics (except the interpolation). If available, finer scale characteristics can be introduced with the use of the PREP_REAL_CASE facilities.

6.6 References

Clark, T. L., and R. D. Farley, 1984: Severe downslope windstorm calculations in two and three spatial dimensions using anelastic interactive grid nesting: a possible mechanism for gustiness. *J. Atmos. Sci.*, **41**, 329-350.

Chen C. 1991: A nested grid, non-hydrostatic, elastic model using a terrain-following coordinate transformation: the radiative-nesting boundary conditions. *Mon. Wea. Rev.*, **119**, 2852-2869.

De Floriani, A. and G. Dettori, 1981: An interpolation method for surfaces with tension. *Adv. Eng. Software*, **3**, 151-154.

Kurihara, Y., G. J. Tripoli and M. A. Bender, 1979: Design of a movable nested-mesh primitive equation model. *Mon. Wea. Rev.*, **107**, 239-249.

Stein J., E. Richard, J.-P. Lafore, J.-P. Pinty, N. Asencio and S. Cosma, 2000: High-resolution non-hydrostatic simulations of flash-flood episodes with grid-nesting and ice-phase parametrization. *Meteorol. Atmos. Phys.*, **72**, 101-110

Chapter 7

Advection Schemes

Contents

7.1 Introduction

Due to the computational C-grid, two kinds of advection schemes are available in Meso-NH, one for momentum and the other for scalar variables.
For the momentum variables, the available schemes are the centered schemes (4^{th} or 2^{nd} order) and the Weighted Essentially Non Oscillatory (WENO) schemes (5^{th} or 3^{rd} order).

For the scalar variables, the available scheme is the the Piecewise Parabolic Method (PPM) scheme. Note that the scalar variables have been classified into two groups: the meteorological variables (temperature, water substances, turbulent kinetic energy) and the other scalar variables (aerosol particles, gas concentrations, ...).

For the temporal schemes, the momentum variables are integrated with Explicit Runge-Kutta (ERK) methods for WENO schemes, and Leap-Frog (LF) or ERK temporal algorithm for the centered spatial schemes. All the rest of the model uses the Forward-In-Time (FIT) temporal integration, including the PPM transport scheme for the scalar variables. To increase the maximum Courant-Friedrichs-Lewy (CFL) number, an additional time-splitting is introduced for the PPM and WENO schemes.

These schemes are presented in more detail below. A comparison between spatial and temporal schemes for momentum, in terms of maximum CFL number and wall clock time to solution, completes the presentation and allows some recommendations.

7.2 Advection schemes for momentum

7.2.1 General approach

The transport scheme of momentum considers the equations of wind in their flux form to ensure conservation of momentum:

$$\begin{aligned}
\frac{\partial}{\partial t}(\tilde{\rho} u) &= -\frac{\partial}{\partial \overline{x}}(\tilde{\rho} U^c\, u) - \frac{\partial}{\partial \overline{y}}(\tilde{\rho} V^c\, u) - \frac{\partial}{\partial \overline{z}}(\tilde{\rho} W^c\, u), && (7.1)\\
\frac{\partial}{\partial t}(\tilde{\rho} v) &= -\frac{\partial}{\partial \overline{x}}(\tilde{\rho} U^c\, v) - \frac{\partial}{\partial \overline{y}}(\tilde{\rho} V^c\, v) - \frac{\partial}{\partial \overline{z}}(\tilde{\rho} W^c\, v), && (7.2)\\
\frac{\partial}{\partial t}(\tilde{\rho} w) &= -\frac{\partial}{\partial \overline{x}}(\tilde{\rho} U^c\, w) - \frac{\partial}{\partial \overline{y}}(\tilde{\rho} V^c\, w) - \frac{\partial}{\partial \overline{z}}(\tilde{\rho} W^c\, w). && (7.3)
\end{aligned}$$

Due to the conform horizontal projection and terrain-following vertical coordinates such as Gal-Chen and Somerville (1975), or Schär et al. (2002), the contravariant components of the wind U_c, V_c and W_c are introduced, corresponding to the components of the wind orthogonal to the coordinate lines (Fig. 7.1). This means that the advection scheme transports the directional momentum (the “advected ”field) by the contravariant components of the wind field (the “advector ”field).

To simplify, we consider only the x-momentum equation, and its x-derivative term:

$$\frac{\partial(\tilde{\rho} U^c u)}{\partial \overline{x}} = \frac{\partial(F_C(\tilde{\rho} U^c) F(u))}{\partial \overline{x}}. \tag{7.4}$$

$F_C(\tilde{\rho} U^c)$ contains the topologic terms, which integrate terrain transformations. The second flux $F(u)$ is calculated on the mesh point without considering any terrain transformation, using the advection method. All the other derivative terms are built with a similar methodology.

Because of the Arakawa C-grid, the advector (contravariant components) and the transported wind field (u, v and w) present different directions. The discrete form of the contravariant metric terms is of 4^{th} order on the vertical direction, in agreement with Klemp et al. (2003), and of 2^{nd} order on the horizontal directions.

Defining i as the spatial index in the x-direction and Δx the mesh step size, the derivative is written so:

$$\frac{\partial(\tilde{\rho} U^c u)_i}{\partial \overline{x}} = \frac{F_C(\tilde{\rho} U^c)_{i+1/2} F(u)_{i+1/2}}{\Delta x} - \frac{F_C(\tilde{\rho} U^c)_{i-1/2} F(u)_{i-1/2}}{\Delta x}. \tag{7.5}$$

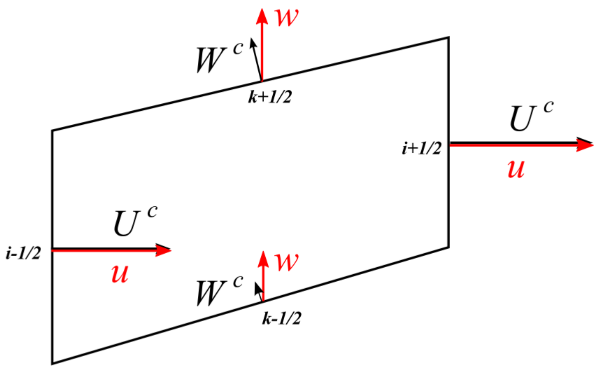


Figure 7.1: Representation of the wind (u, w) and its contravariant components (U_c, W_c) on the C-grid in the 2D vertical plane (x, z).

Then, a chosen discretization is applied to the flux terms F, following its definition using the flux formulation:

$$\frac{\partial u_i}{\partial x} = \frac{F(u)_{i+1/2} - F(u)_{i-1/2}}{\Delta x}. \tag{7.6}$$

Two different methods, each with two distinct orders, can be used to discretize F: a WENO discretization of 5^{th} or 3^{rd} order (WENO5 and WENO3, respectively), or a centered discretization of 4^{th} or 2^{nd} order (CEN4TH and CEN2ND, respectively).

7.2.2 Discretization with the WENO schemes

The WENO schemes owe their success to the use of a dynamic set of stencils, where a nonlinear convex combination of lower order polynomials adapts either to a higher order approximation at smooth parts of the solution, or to an upwind spatial discretization that avoids interpolation across discontinuities and provides the necessary dissipation for shock capturing (Jiang and Shu, 1995; Shu, 1998). They allow a better representation of the solution in presence of high gradients. Pressel et al. (2015) have shown the advantages of the WENO schemes from 3^{rd} through 11^{th} order for the transport of scalars and momentum over central difference schemes.

The first step consists on separating the velocity flux terms into positive and negative fluxes, using Lax-Friedrich flux splitting as in Shu (1998):

$$F_{\text{WENO}}(u)_{i+1/2} = f^+_{i+1/2} + f^-_{i+1/2}. \tag{7.7}$$

In the following development, only the reconstruction of the positive fluxes is described. The reader can refer to Wang and Spiteri (2007), Shu (1998) and Castro et al. (2011) for a more detailed description. The velocity fluxes are built using a stencil given by a Lagrangian interpolation using the velocity average on each cell.

For WENO5, the stencil is given by:

$$\begin{aligned} f^+_{i+1/2} = \gamma_0 &\left(\frac{2}{6}\bar{u}_{i-2} - \frac{7}{6}\bar{u}_{i-1} + \frac{11}{6}\bar{u}_i\right) \\ &+ \gamma_1 \left(-\frac{1}{6}\bar{u}_{i-1} + \frac{5}{6}\bar{u}_i + \frac{2}{6}\bar{u}_{i+1}\right) \\ &+ \gamma_2 \left(\frac{2}{6}\bar{u}_i + \frac{5}{6}\bar{u}_{i+1} - \frac{1}{6}\bar{u}_{i+2}\right) \end{aligned} \tag{7.8}$$

with the average value for the velocity defined by:

$$\bar{u}_i = \frac{1}{\Delta x_i} \int_{x_{i-1/2}}^{x_{i+1/2}} u(\xi) d\xi. \tag{7.9}$$

The strength of the WENO5 method relies on the choice of the WENO stencil weights γ_j. These allow to keep a non-oscillatory solution even in presence of shock or high gradient in the velocity field. They are given by:

$$\gamma_j = \frac{\alpha_j}{\alpha_0 + \alpha_1 + \alpha_2} \tag{7.10}$$

where the non-normalized stencil weights are:

$$\begin{aligned} \alpha_0 = \frac{1}{10}\left(\frac{1}{\epsilon + \beta_0}\right)^2, \ \alpha_1 &= \frac{6}{10}\left(\frac{1}{\epsilon + \beta_1}\right)^2, \\ \alpha_2 &= \frac{3}{10}\left(\frac{1}{\epsilon + \beta_2}\right)^2. \end{aligned} \tag{7.11}$$

The ϵ term is here to prevent the denominator to be null (it is set to 10^{-15} in the model). The β_j terms, also called indicators of smoothness, are the heart of essentially non-oscillatory (ENO) methods, from which the WENO schemes are extended. They are defined as:

$$\begin{aligned} \beta_0 &= \frac{13}{12}\left(\bar{u}_{i-2} - 2\bar{u}_{i-1} + \bar{u}_i\right)^2 + \frac{1}{4}\left(\bar{u}_{i-2} - 4\bar{u}_{i-1} + 3\bar{u}_i\right)^2 \\ \beta_1 &= \frac{13}{12}\left(\bar{u}_{i-1} - 2\bar{u}_i + \bar{u}_{i+1}\right)^2 + \frac{1}{4}\left(\bar{u}_{i-1} - \bar{u}_{i+1}\right)^2 \\ \beta_2 &= \frac{13}{12}\left(\bar{u}_i - 2\bar{u}_{i+1} + \bar{u}_{i+2}\right)^2 + \frac{1}{4}\left(3\bar{u}_i - 4\bar{u}_{i+1} + \bar{u}_{i+2}\right)^2. \end{aligned} \tag{7.12}$$

WENO5 reverts to WENO3 at the edges of the computational domain for open boundary conditions only.

For WENO3, the stencil is given by:

$$f^+_{i+1/2} = \gamma_0 \left(-\frac{1}{2}\bar{u}_{i-1} + \frac{3}{2}\bar{u}_i\right) + \gamma_1 \left(\frac{1}{2}\bar{u}_i + \frac{1}{2}\bar{u}_{i+1}\right). \tag{7.13}$$

The stencil weights are:

$$\gamma_j = \frac{\alpha_i}{\alpha_0 + \alpha_1} \tag{7.14}$$

where the non-normalized stencil weights are:

$$\alpha_0 = \frac{1}{3}\left(\frac{1}{\epsilon + \beta_0}\right)^2, \ \alpha_1 = \frac{2}{3}\left(\frac{1}{\epsilon + \beta_1}\right)^2 \tag{7.15}$$

and for the smoothness indicators:

$$\beta_0 = \left(-\bar{u}_{i-1} + \bar{u}_i\right)^2, \ \beta_1 = \left(-\bar{u}_i + \bar{u}_{i+1}\right)^2. \tag{7.16}$$

The computational cost of WENO3 is much lower than WENO5, but WENO3 is known to be strongly more diffusive (Tan et al., 2005). The use of explicit numerical diffusion applied to wind fields with WENO schemes is prohibited. WENO5 requires a halo of 3 points between adjacent processors (NHALO=3).

7.2.3 Discretization with centered schemes

For centered schemes, no flux decomposition is required. The fluxes are directly computed using a 4^{th}-order reconstruction for CEN4TH:

$$F_{\text{CEN4TH}}(u)_{i+1/2} = \frac{7(u_{i+1}+u_i)-(u_{i+2}+u_{i-1})}{12} \tag{7.17}$$

or a 2^{nd} order for CEN2ND:

$$F_{\text{CEN2ND}}(u)_{i+1/2} = \frac{u_{i+1}+u_i}{2} \tag{7.18}$$

CEN4TH reverts to CEN2ND at the edges of the computational domain for open boundary conditions only.

The centered schemes must be also combined with numerical diffusion operator of 4^{th} order in the model, in order to damp numerical energy accumulation in the shortest wavelengths. The 4^{th} order centered scheme presents a good accuracy (the effective resolution is of order of 5–6 Δx; Ricard et al., 2013) and it is easy to implement.

7.2.4 Temporal schemes for momentum

Once the space derivatives are estimated, a temporal discretization is used to integrate the fluxes in time from the current state to the next one. A common strategy to improve computational efficiency is to use explicit numerical schemes integrating the high-frequency modes with a small time step and lower-frequency modes with a larger time step. Wicker and Skamarock (2002) were among the first to use a 3^{rd}-order explicit Runge-Kutta (ERK) scheme. Higher order time-stepping combinations have been studying since them, especially combined with WENO spatial discretization (e.g. Wang and Spiteri, 2007).

For the WENO schemes employed in Meso-NH, some ERK methods of high order are implemented. An explicit splitting method can be added to improve a bit more the stability properties as presented below. For centered momentum schemes, only the Leap-Frog (LF) scheme and one specific centered ERK method are available.

Explicit RK method

Explicit Runge-Kutta (ERK) methods can be applied to the momentum transport, together with FIT time integration for the rest of the model (contravariant flux $F_C(\tilde{\rho}U^c)$ among others). The general temporal proceeding for one advection term in Eq. (7.1) will then be described.

Using the anelastic hypothesis ($\frac{\partial \rho}{\partial t} = 0$), we consider the tendency of a variable defined by its time-variation induced by the spatial term in the advection equation, noted T_u:

$$T_u = \tilde{\rho}\frac{\partial u}{\partial t}$$

, which is written in the discrete form using the FIT formulation (with n as the temporal index) as:

$$T_u^{n+1} = \tilde{\rho}\frac{u^{n+1}-u^n}{\Delta t} \tag{7.19}$$

The contravariant flux $F_C(U^c)$ (the “advector ” field) is kept constant along the time step, to satisfy the continuity equation. The ERK method is applied on the following equation:

$$\tilde{\rho}\frac{\partial u}{\partial t} = M(u) + S \tag{7.20}$$

where $M(u)$ is the discrete term defined in Eq. (7.1), and S represents the other terms of the momentum equation. As the solver pressure is called between the call to the momentum advection and the end of the time-step, the pressure contribution of the previous time step is added to the processes computed since the beginning of the current time step.
We consider a general s-stage ERK method defined by its Butcher coefficients:

$$\begin{array}{c|cccccc}
c_1 & & & & & & \\
c_2 & a_{21} & & & & & \\
c_3 & a_{31} & a_{32} & & & & \\
c_4 & a_{41} & a_{42} & a_{43} & & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & & \\
c_s & a_{s1} & a_{s2} & a_{s3} & \dots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & b_3 & \dots & b_{s-1} & b_s
\end{array}$$

The advection tendency follows:

$$\begin{aligned}
u_1^n &= u^n \\
u_k^n &= u^n \; + \; \Delta t \sum_{j=1}^{k-1} a_{k,j} \; \frac{M(u_j^n) + S}{\rho^n} \\
T_u^{n+1} &= \sum_{k=1}^{s} b_k \; \left(M(u_k^n) + S\right).
\end{aligned} \tag{7.21}$$

Adding the advection tendency to get u^{n+1} using Eq. (7.19) leads to the classical Runge-Kutta method applied to the momentum equation. The different explicit RK methods considered in Meso-NH are:

ERK with order 1 and 2 steps (RK21)

$$\begin{array}{c|cc}
0 & & \\
3/4 & 3/4 & \\
\hline
 & 0 & 1
\end{array}$$

SSP-RK with order 3 and 3 steps (RK33)

$$\begin{array}{c|ccc}
0 & & & \\
1 & 1 & & \\
1/2 & 1/4 & 1/4 & \\
\hline
 & 1/6 & 1/6 & 2/3
\end{array}$$

ERK with order 3 and 5 steps (RK53)

$$\begin{array}{c|ccccc}
0 & & & & & \\
1/7 & 1/7 & & & & \\
3/16 & 0 & 3/16 & & & \\
1/3 & 0 & 0 & 1/3 & & \\
2/3 & 0 & 0 & 0 & 2/3 & \\
\hline
 & 1/4 & 0 & 0 & 0 & 3/4
\end{array}$$

ERK order 4 (RKC4)

$$\begin{array}{c|cccc}
0 & & & & \\
1/2 & 1/2 & & & \\
1/2 & 0 & 1/2 & & \\
1 & 0 & 0 & 1 & \\
\hline
 & 1/6 & 1/3 & 1/3 & 1/6
\end{array}$$

RK33 requires more memory storage than the RKC4 and RK53 methods, as RK33 has a plain Butcher matrix. In contrast, RKC4 and RK53 have a diagonal Butcher matrix, so they only need to store one field in memory at each stage to compute the next one, justifying the classification of low-storage ERK methods.
WENO3 and WENO5 can be used with all the ERK schemes presented above.
CEN4TH can only be used with RKC4, or without RK time splitting with the classical Leap-Frog scheme. CEN2ND can only be used with the Leap-Frog scheme.

Additional time splitting method

To increase a bit more the maximum CFL number, an additional time-splitting can be activated for the wind advection with WENO. Considering one time step $[t_n, t_{n+1}]$, we divide it into L regular sub-steps $[t_l, t_{l+1}]$ with $t_n = t_0 < ... < t_l < t_{l+1} < ... < t_L = t_{n+1}$.
Once one value u^l is known (u^0 at first), the next value u^{l+1} is computed using Eq. (7.19) with $\Delta t = t_{l+1} - t_l$, that is:

$$T_u^{l+1} = \rho \frac{u^{l+1} - u^l}{\Delta t} \tag{7.22}$$

with u^{l+1} computed using all stages of the ERK method as described in Eq. (7.21). This process is repeated L times to compute the L tendencies (Fig. 7.2). At the end, we obtain the tendency on the original final time t_{n+1} by calculating the following average:

$$T_u = \frac{1}{L} \sum_{l=1}^{L} T_u^l. \tag{7.23}$$

The main interest of such an additional time splitting is to call the rest of the model (pressure solver, physics, chemistry, ...) less frequently.
When WENO schemes are applied to momentum transport (Fig. 7.2), the larger time step is applied to all the model components including the physics and the pressure solver, with the FIT temporal scheme. The advection of all variables is conducted with a constant advection momentum vector. A smaller time step is used for wind advection applying the ERK method on the subinterval.
With the centered momentum transport schemes, there is no additional time splitting. Therefore, when CEN4TH is applied to momentum transport with the RKC4 time marching, there is no sub-step l for momentum. When it is applied with the Leap-Frog scheme, there is neither ERK time-splitting nor sub-step l.

Comparison on the 2D hydrostatic mountain wave

A classical evaluation of the momentum advection schemes involves the steady-state solution of linear 2D hydrostatic flow over a single-peaked mountain with constant inflow, as in Durran and Klemp (1983) and Xue and Thorpe (1991). The profile of the symmetric Witch of Agnesi mountain is used as:

$$h(x) = h_{\max} \frac{a^2}{x^2 + a^2}$$

with $h_{\max} = 1\ m$ the height and $a = 10$ km the half-width of the mountain. The initial state of the atmosphere consists of a constant mean flow with $U = 20$ m s^{-1}, a ground potential temperature of $\theta = 250$ K, and a Brunt-Väisälä frequency $N = 0.02$ s^{-1}. The resolution is $\Delta x = 500$ m and $\Delta z = 250$ m, and the domain extends horizontally over 800 km and vertically over 30 km. A

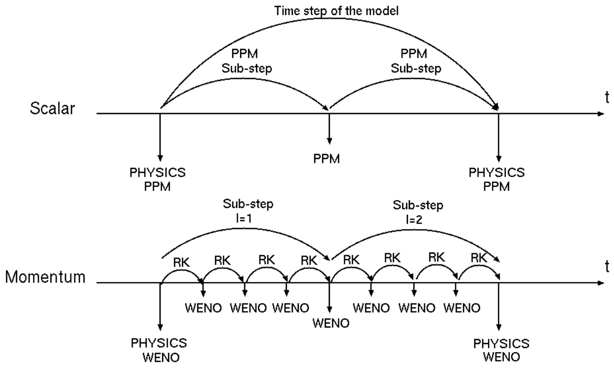


Figure 7.2: Representation of the time-marching in Meso-NH with WENO schemes for the transport of momentum and scalar variables.

Rayleigh damping layer is applied above 22 km. Figure 7.3 shows that the numerical (dashed grey) values of vertical velocity and the analytical (colored contours) ones compare well. The simulation is shown here only for WENO5 as the differences with the various advection schemes are too tiny to be visible.

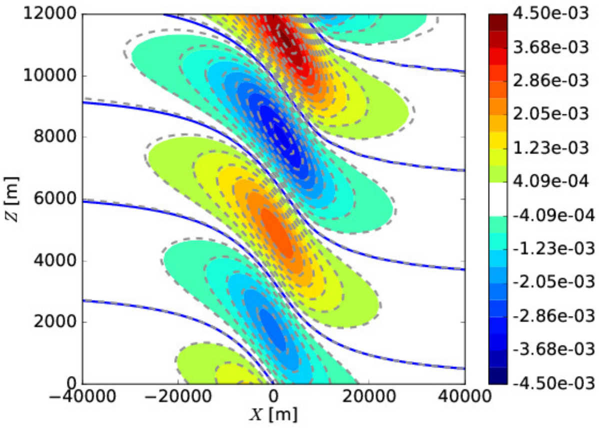


Figure 7.3: Case of linear hydrostatic mountain. Vertical cross-section of vertical velocity (m s^{-1}) after 10 h. Colored isovalues correspond to the analytical solution, and dashed grey lines to the numerical one, given here for WENO5 and RK53, with CFL= 0.4.

But there are some significant differences in terms of maximum CFL number and wall clock time to solution. A stability study, including the additional time-splitting for WENO schemes, is presented in Table 7.1. It should be noted that this case does not call any physics: only the dynamical sources are involved.

First, the additional time-splitting for WENO schemes is only presented with $L = 2$ as $L = 3$ does not improve the maximum CFL number.

Without any additional time-splitting for WENO schemes, the CFL numbers are maximum for CEN4TH/RKC4, and WENO5/WENO3 combined with RKC4/RK53. In contrast, the lowest CFL number is given by CEN4TH/LF. RK33 gives a maximum CFL number lower than RK53 and RKC4 with the WENO schemes (it requires more memory storage as explained above).

The computational cost is significantly lower with CEN4TH/RKC4, while WENO5 requires strong computational effort. WENO3/RKC4 costs almost the same than CEN4TH/RKC4 at equal CFL

number. Lastly, RKC4 and RK53 produce equivalent maximum CFL numbers with WENO schemes, but RKC4 is slightly cheaper with 4 stages instead of 5 for RK53.
With additional time-splitting for the WENO schemes, the maximum CFL number is significantly increased with RKC4/RK53 methods, up to 1.8 with WENO5 and 2.5 with WENO3. The impact is not obvious on the wall clock time to solution, because this case only involves dynamical processes. But it will be more significant on cases with physics, and even more with chemistry.

	LF	RK33	RK53	RKC4
CEN4TH	0.4			1.5
	1			*0.42*
WENO-3				
L=1		1.0	1.3	1.3
		0.74	*0.63*	*0.58*
L=2		2.1	2.5	2.5
		0.52	*0.57*	*0.54*
WENO-5				
L=1		1.0	1.4	1.4
		2.15	*1.95*	*1.76*
L=2		1.7	1.8	1.8
		1.99	*2.51*	*2.21*

Table 7.1: Case of linear hydrostatic flow for CEN4TH and for WENO schemes combined with ERK methods and additional time-splitting (L is the split number): maximum CFL number and computational cost for the maximum CFL number in italics to run 15000 s, considering the computational cost of CEN4TH/LF as the unit.

Recommendations

Other tests have shown that:

- CEN4TH/RKC4 presents the best effective resolution, and the less diffusive behaviour, followed by CEN4TH/LF and then WENO5/RK53-RKC4. This is crucial for LES of clouds: the entrainment of environmental at the cloud edges is higher with CEN4TH/RKC4 as the implicit diffusion is lower.
- WENO3 is excessively damping and is prohibited for LES as the effective resolution is of the order of $12\Delta x$.
- WENO5/RK53-RKC4 is well adapted to sharp gradients area, i.e. in complex shock-obstacle interactions with immersed boundary method.
- CEN4TH/LF is much more expensive than CEN4TH/RKC4

Therefore the recommendations could be:

- CEN4TH/RKC4 is recommended for LES.
- WENO5/RKC4 is recommended for simulations with expensive chemistry/physics, typically mesoscale simulations, or for simulations with sharp-gradient area.

7.3 Advection schemes for scalar variables: the PPM

7.3.1 Introduction

The PPM advection scheme is based on the piecewise parabolic method (PPM), a numerical technique developed in astrophysics for modeling fluid flows (Colella et al. 1984, Carpenter et al. 1990). The PPM scheme differs substantially from conventional grid-point methods like the 2^{nd}- and 4^{th}-order centered advection schemes. It is a finite-volume method in which some assumption is made about the structure of the approximate solution between the grid points. In the PPM case, piecewise continuous parabolas are fitted in each grid-cell. This design enables the scheme to handle sharp gradients and discontinuities very accurately. Although the constructed piecewise continuous functions would allow for the explicit calculation of derivatives, in practice the advective forcing is computed from the fluxes at the grid-cell edges. Various flux limiters can be used to ensure the scheme is monotonic. Monotonic schemes do not amplify extrema in the initial values and are, thus, also positive definite. Application of the PPM schemes to 3D problems requires density-corrected directional splitting and FIT time-marching.

Three different versions of the PPM advection scheme have been implemented in Meso-NH: the unrestricted scheme PPM_00 and two monotonic versions, PPM_01 and PPM_02. PPM_01 is based on the original scheme (Colella at al. 1984) with monotonicity constraints modified by Lin and Rood (1996) while PPM_02 uses the flux limiter developed by Skamarock (2006). PPM_02 permits extension to stable semi-Lagrangian integrations using CFL numbers larger than one. The PPM schemes are intended for advection of meteorological variables (e.g. temperature, water species, TKE) and passive scalars. All three versions have excellent mass-conservation properties and were found to be an order of magnitude more accurate than the centered 4th order (CEN4TH) scheme previously applied to scalar variables in idealized 2D tests. However, the implicit numerical diffusion associated with the monotonicity preserving methods in PPM_01 and PPM_02 can damp smooth, well resolved extrema. Therefore the use of explicit numerical diffusion applied to scalar fields should be prohibited when using the PPM schemes. Also, the flux limiters and the specific time marching require computational effort making the monotonic PPM schemes significantly more expensive than a classical centered scheme.

7.3.2 Description

The advective part of the scalar conservation equation can be written as:

$$\frac{\partial}{\partial t}(\tilde{\rho}\phi) = -\frac{\partial}{\partial x}(\tilde{\rho}U_c\phi) - \frac{\partial}{\partial y}(\tilde{\rho}V_c\phi) - \frac{\partial}{\partial z}(\tilde{\rho}W_c\phi) \tag{7.24}$$

where in the same way as for the transport of wind, the contravariant components of the wind field are the “advector ”field (U_c, V_c, W_c). To simplify the notation, we will consider in the following part (u, v, w) instead of (U_c, V_c, W_c), as if we were in a Cartesian framework. To solve this equation on a discrete grid using the PPM finite-volume method, an approximate solution for the scalar field ϕ is constructed between the grid points. The grid-point value ϕ_i of a scalar variable at a mass grid point i represents the average of the function $\phi(x)$ over the grid-cell, or the interval $x_{i-1/2} \leq x \leq x_{i+1/2}$, where the $x_{i-1/2}$ and $x_{1+1/2}$ are the neighboring momentum grid points. In the PPM case, the function $\phi(x)$ is a parabola which is constructed uniquely for each grid-cell using the cell mean (ϕ_i) and values at the cell edges ($\phi_{i-1/2}$ and $\phi_{i+1/2}$). Once the parabolas are known, we can calculate fluxes at the grid-cell edges, and the net scalar transport in each cell is equal to the flux divergences.

A solution (in one dimension) to Eq. (7.24) can be calculated from the FIT discretization:

$$(\tilde{\rho}\phi)_i^{t+\Delta t} = (\tilde{\rho}\phi)_i^t - \mathcal{F}_{x,i}(\phi^t) \tag{7.25}$$

where the operator $\mathcal{F}_{x,i}$ denotes the discrete flux divergence in the grid-cell i and is expressed as:

$$\mathcal{F}_{x,i}(\phi^t) = \frac{\Delta t}{\Delta x_i}\left[(\tilde{\rho}u)_{i+1/2} f(\phi^t)_{i+1/2} - (\tilde{\rho}u)_{i-1/2} f(\phi^t)_{i-1/2}\right] \tag{7.26}$$

where $F_{i\pm 1/2} = (\tilde{\rho}u)_{i\pm 1/2} f(\phi^t)_{i\pm 1/2}$ are the scalar fluxes at the grid-cell edges. These fluxes can be determined by integrating over the parabolas in each cell. The parabola in the grid-cell i (dropping the superscript t) is constructed from the cell average (ϕ_i) and left- and right-edge values ($\phi_{L,i}$ and $\phi_{R,i}$) in the following way:

$$\phi_i(\xi) = \phi_{L,i} + \xi\left[\Delta(\phi)_i + \phi_{6,i}\,(1-\xi)\right], \quad \xi = \frac{x - x_{i-1/2}}{\Delta x_i} \tag{7.27}$$

where $0 \leq \xi \leq 1$ ($x_{i-1/2} \leq x \leq x_{i+1/2}$) is the horizontal coordinate. The parameters of the parabola are:

$$\Delta(\phi)_i = \phi_{R,i} - \phi_{L,i} \tag{7.28}$$

$$\phi_{6,i} = 6\left[\phi_i - \frac{1}{2}\left(\phi_{R,i} + \phi_{L,i}\right)\right]. \tag{7.29}$$

The edge values are determined using the 4$^{\text{th}}$ order differencing:

$$\phi_{L,i} = \phi_{i-1/2} = \frac{1}{2}\left(\phi_i + \phi_{i-1}\right) - \frac{1}{6}\left(\delta(\phi_i) - \delta(\phi_{i-1})\right) \tag{7.30}$$

$$\phi_{R,i} = \phi_{i+1/2} = \frac{1}{2}\left(\phi_{i+1} + \phi_i\right) - \frac{1}{6}\left(\delta(\phi_{i+1}) - \delta(\phi_i)\right) \tag{7.31}$$

where the operator $\delta(\,)$ is defined as:

$$\delta(\phi_i) = \frac{1}{2}\left(\phi_{i+1} - \phi_{i-1}\right). \tag{7.32}$$

Finally, the scalar fluxes at the edges of grid-cells ($f(\phi)_{i\pm 1/2}$) are calculated as the average value of the scalar over the advective distance in each grid cell:

$$f(\phi)_{i+1/2} = \frac{1}{\Delta t u_{i+1/2}} \int_{x_{i+1/2}-\Delta t u_{i+1/2}}^{x_{i+1/2}} \phi_i(x)\,\mathrm{d}x, \quad \text{for} \quad u_{i+1/2} > 0 \tag{7.33}$$

Substituting Eqs. (7.27) through (7.32) we get:

$$f(\phi)^{+}_{i+1/2} = \phi_{R,i} - \frac{1}{2}Cr_{i+1/2}\left[\Delta(\phi)_i - \left(1 - \frac{2}{3}Cr_{i+1/2}\right)\phi_{6,i}\right] \tag{7.34}$$

for $Cr_{i+1/2} \geq 0$,

$$f(\phi)^{-}_{i+1/2} = \phi_{L,i+1} - \frac{1}{2}Cr_{i+1/2}\left[\Delta(\phi)_{i+1} + \left(1 + \frac{2}{3}Cr_{i+1/2}\right)\phi_{6,i+1}\right] \tag{7.35}$$

for $Cr_{i+1/2} < 0$. Here $Cr_{i+1/2} = u_{i+1/2}\Delta t/\Delta x$ is the CFL number.

The PPM_00 unrestricted scheme

The PPM advective fluxes $f_{i\pm1/2}$ can be expressed in a more concise way for more efficient calculation. Combining Eqs (7.27) through (7.32) with the expressions for the fluxes [Eqs. (7.34),(7.35)], we get:

$$\begin{aligned} f(\phi)^+_{i+1/2} &= \phi_{i+1/2} - Cr_{i+1/2}(\phi_{1+1/2} - \phi_i) \\ &\quad -Cr_{i+1/2}(1 - Cr_{i+1/2})(\phi_{i-1/2} - 2\phi_i + \phi_{i+1/2}) \end{aligned} \tag{7.36}$$

for $Cr_{i+1/2} = u_{i+1/2}\Delta t/\Delta x \geq 0$,

$$\begin{aligned} f(\phi)^-_{i+1/2} &= \phi_{i+1/2} + Cr_{i+1/2}(\phi_{1+1/2} - \phi_{i+1}) \\ &\quad +Cr_{i+1/2}(1 + Cr_{i+1/2})(\phi_{i+1/2} - 2\phi_{i+1} + \phi_{i+3/2}) \end{aligned} \tag{7.37}$$

for $Cr_{i+1/2} = u_{i+1/2}\Delta t/\Delta x < 0$. The values at the grid-cell edges (from Eqs. (7.30) and (7.31)) can be expressed as:

$$\phi_{i+1/2} = \left[7(\phi_{i+1} + \phi_i) - (\phi_{i+2} + \phi_{i-1})\right]/12. \tag{7.38}$$

The PPM_01 monotonic scheme

When sharp gradients or discontinuities are present in the scalar field, the piecewise continuous parabolas can have very steep gradients and produce non-physical values – overshoots (e.g. concentrations higher than 100 %) or undershoots (e.g. negative concentrations) somewhere inside the grid-cell. In those cases the calculated advective forcing (given by cell-edge fluxes (Eqs. 7.34) and (7.35)) would likely result in non-physical values in the scalar field after the advection step. To avoid the introduction of new extrema (overshoots or undershoots) in the domain, monotonicity constraints can be imposed on the advection scheme. In the PPM case, this can be achieved in two ways: modifying the parabolas or limiting the cell-edge fluxes.

In the original PPM scheme (Colella 1984) the monotonicity is ensured by modifying the parameters of the parabolas. Lin and Rood (1996) have proposed simplified monotonicity constraints that require fewer floating-point operations. The parabola parameters $\delta(\phi_i)$, $\Delta(\phi)_i$, $\phi_{6,i}$, $\phi_{L,i}$ and $\phi_{R,i}$ are modified in the following way. The average slope $\delta(\phi_i)$ in the i^{th} grid-cell is replaced by the modified version $\delta_m(\phi_i)$:

$$\begin{aligned} \delta_m(\phi_i) &= \text{sign}(\delta(\phi_i)) \times \\ &\quad \min\left[|\delta(\phi_i)|,\, 2(\phi_i - \phi_i^{\min}),\, 2(\phi_i^{\max} - \phi_i)\right] \\ \phi_i^{\min} &= \min(\phi_{i-1}, \phi_i, \phi_{i+1}) \\ \phi_i^{\max} &= \max(\phi_{i-1}, \phi_i, \phi_{i+1}) \end{aligned} \tag{7.39}$$

The first-guess parabola coefficients ($\Delta(\phi)_i$, $\phi_{6,i}$, $\phi_{L,i}$, $\phi_{R,i}$) are calculated using the modified $\delta_m(\phi_i)$. The parameters are then adjusted to eliminate overshoots and undershoots through the following algorithm:

```
IF    $\delta_m(\phi_i) = 0$
    $\hat{\phi}_{L,i} = \phi_i$,    $\hat{\phi}_{R,i} = \phi_i$,    $\hat{\phi}_{6,i} = 0$

ELSE IF    $\phi_{6,i}\Delta(\phi)_i < -(\Delta(\phi)_i)^2$
    $\hat{\phi}_{6,i} = 3(\phi_{L,i} - \phi_i)$,    $\hat{\phi}_{R,i} = \phi_{L,i} - \hat{\phi}_{6,i}$,    $\hat{\phi}_{L,i} = \phi_{L,i}$
```

```
ELSE IF   $\phi_{6,i}\Delta(\phi)_i > (\Delta(\phi)_i)^2$
    $\hat{\phi}_{6,i} = 3(\phi_{R,i} - \phi_i),\qquad \hat{\phi}_{L,i} = \phi_{R,i} - \hat{\phi}_{6,i},\qquad \hat{\phi}_{R,i} = \phi_{R,i}$
END IF
```

The parameters of the monotonized parabolas (labeled with ˆ) are used to calculate the advective fluxes defined in Eqs. (7.34) and (7.35).

The PPM_02 monotonic scheme

An alternative way of ensuring monotonicity is by correcting the cell-edge fluxes as in the standard FCT approach. This limiter is based on the work of Skamarock (2006) and Blossey and Durran (2008). The flux at the cell-edge is composed of a monotonicity-preserving upwind flux

$$F^{\mathrm{up}}_{i+1/2} = \begin{cases} (\tilde{\rho}u)_{i+1/2}\phi^t_i, & \text{for} \quad (\tilde{\rho}u)_{i+1/2} \geq 0 \\ (\tilde{\rho}u)_{i+1/2}\phi^t_{i+1}, & \text{otherwise} \end{cases} \tag{7.40}$$

and a higher order correction such that

$$F^m_{i+1/2} = F^{\mathrm{up}}_{i+1/2} + r_{i+1/2}F^{\mathrm{cor}}_{i+1/2} \tag{7.41}$$

where $0 \leq r_{i+1/2} \leq 1$ and $F^{\mathrm{cor}}_{i+1/2} = F^{\mathrm{ppm}}_{i+1/2} - F^{\mathrm{up}}_{i+1/2}$ is the difference between the PPM flux (calculated using Eqs. (7.36) or (7.37)) and the upstream flux. The resulting flux $F^m_{i+1/2}$ will produce a monotonic solution in the advection step.
To evaluate $r_{i+1/2}$ we first calculate an approximate "transported and diffused" solution using the upwind flux

$$(\tilde{\rho}\phi)^{td}_i = (\tilde{\rho}\phi)^t_i - \frac{\Delta t}{\Delta x_i}\left(F^{up}_{i+1/2} - F^{up}_{i-1/2}\right) \tag{7.42}$$

The sum of the correction fluxes directed out of the cell i is computed as $F^{\mathrm{out}}_i = \max(F^{\mathrm{cor}}_{i+1/2}, 0) - \min(F^{\mathrm{cor}}_{i-1/2}, 0)$, and the sum of the fluxes directed into the cell $i+1$ is calculated as $F^{\mathrm{in}}_{i+1} = \max(F^{\mathrm{cor}}_{i+1/2}, 0) - \min(F^{\mathrm{cor}}_{i+3/2}, 0)$. Finally, let

$$\phi^{\mathrm{max,min}}_i = \max, \min(\phi^t_{i-1}, \phi^t_i, \phi^t_{i+1}, \phi^{td}_{i-1}, \phi^{td}_i, \phi^{td}_{i+1}) \tag{7.43}$$

The re-normalization factor for monotonicity preservation is defined as:

$$r_{i+1/2} = \max\left[0, \min\left(1, \frac{[(\tilde{\rho}\phi)^{td}_i - \hat{\tilde{\rho}}_i\phi^{\mathrm{min}}_i]\Delta x_i}{\Delta t F^{\mathrm{out}}_i + \varepsilon}, \frac{[\hat{\tilde{\rho}}_{i+1}\phi^{\mathrm{max}}_{i+1} - (\tilde{\rho}\phi)^{td}_{i+1}]\Delta x_{i+1}}{\Delta t F^{\mathrm{in}}_{i+1} + \varepsilon}\right)\right] \tag{7.44}$$

for $F^{\mathrm{cor}}_{i+1/2} \geq 0$,

$$r_{i+1/2} = \max\left[0, \min\left(1, \frac{[\hat{\tilde{\rho}}_i\phi^{\mathrm{max}}_i - (\tilde{\rho}\phi)^{td}_i]\Delta x_i}{\Delta t F^{\mathrm{in}}_i + \varepsilon}, \frac{[(\tilde{\rho}\phi)^{td}_{i+1} - \hat{\tilde{\rho}}_{i+1}\phi^{\mathrm{min}}_{i+1}]\Delta x_{i+1}}{\Delta t F^{\mathrm{out}}_{i+1} + \varepsilon}\right)\right] \tag{7.45}$$

for $F^{\mathrm{cor}}_{i+1/2} < 0$. Here ε is a small parameter chosen to avoid division by zero, and $\hat{\tilde{\rho}}$ is the density as updated in the current advection step. More details about the density correction associated with directional splitting in 3D applications will be discussed later. In the final step the cell averages are updated to time $t + \Delta t$

$$(\tilde{\rho}\phi)^{t+\Delta t}_i = (\tilde{\rho}\phi)^{td}_i - \frac{\Delta t}{\Delta x_i}\left(r_{i+1/2}F^{\mathrm{cor}}_{i+1/2} - r_{i-1/2}F^{\mathrm{cor}}_{i-1/2}\right). \tag{7.46}$$

The PPM_02 scheme described here is fully monotonic. It can be modified to use semi-Lagrangian approximation to the flux divergence. This would extend the domain of dependence beyond the adjacent upstream grid-cell and allow stable computation for CFL numbers greater than unity. More details about the semi-Lagrangian extension can be found in e.g. Skamarock (2006) and Blossey and Durran (2008). For Eulerian integration with CFL number less than unity, the PPM_01 scheme is more computationally efficient.

Time marching and extension to three dimensions

To extend the 1D scalar advection scheme [Eq. (7.25)] to multiple dimensions, we follow the formulation of Easter (1993) where the mass conservation $\partial\tilde{\rho}/\partial t + \nabla \cdot (\tilde{\rho}\boldsymbol{V}) = 0$ is simultaneously integrated with the discrete version of the scalar transport [Eq. (7.25)]. This sort of density correction procedure is necessary with the PPM schemes because the mass flux $(\tilde{\rho}U)$ must be saved and updated for scalar advection at each grid-cell interface. The density correction is restricted to advection step at time $t + \Delta t$ and the model overall still uses the selected anelastic approximation. The mass conservation is discretized within a 3D formulation as:

$$\tilde{\rho}^{t+\Delta t} = \tilde{\rho}^{t} - \mathcal{F}_x(I) - \mathcal{F}_y(I) - \mathcal{F}_z(I), \tag{7.47}$$

where the vector $I \equiv 1$ and $\mathcal{F}_{x,y,z}$ denotes the discrete flux divergence in x, y and x directions, which is calculated using the non-monotonic PPM_00 scheme. The full 3D algorithm for simultaneous transport of the scalar and mass conservation is:

$$\begin{aligned}
(\tilde{\rho}\phi)^{*} &= (\tilde{\rho}\phi)^{t} - \mathcal{F}_x(\phi^{t}) \\
\tilde{\rho}^{*} &= \tilde{\rho}^{t} - \mathcal{F}_x(I) \\
\phi^{*} &= (\tilde{\rho}\phi)^{*}/\tilde{\rho}^{*} \\
(\tilde{\rho}\phi)^{**} &= (\tilde{\rho}\phi)^{*} - \mathcal{F}_y(\phi^{*}) \\
\tilde{\rho}^{**} &= \tilde{\rho}^{*} - \mathcal{F}_y(I) \\
\phi^{**} &= (\tilde{\rho}\phi)^{**}/\tilde{\rho}^{**} \\
(\tilde{\rho}\phi)^{t+\Delta t} &= (\tilde{\rho}\phi)^{**} - \mathcal{F}_z(\phi^{**}) \\
\tilde{\rho}^{t+\Delta t} &= \tilde{\rho}^{**} - \mathcal{F}_z(I) \\
\phi^{t+\Delta t} &= (\tilde{\rho}\phi)^{t+\Delta t}/\tilde{\rho}^{t+\Delta t}.
\end{aligned} \tag{7.48}$$

Here the flux-divergence operators $\mathcal{F}_{x,y,z}$ can be estimated using any of the described PPM schemes (PPM_00, PPM_01 or PPM_02), however the density corrections are always calculated using the non-monotonic scheme. The PPM_00 scheme is the most computationally efficient and accuracy of the correction is sufficient even for the advection of scalars with the monotonic schemes.

It can easily be seen that Eq. (7.48) collapses to Eq. (7.47) for $\phi = I$ and it is consistent (if ϕ is constant at initial time it remains constant). To achieve 2^{nd}-order accuracy of the time-split scheme [Eq. (7.48)] a form of Strang splitting (Strang 1968) is used. It consists of alternating the order of flux divergence operators between $x \rightarrow y \rightarrow z$ to $z \rightarrow y \rightarrow x$ at each time step.

PPM is a finite-volume method and the cell-edge fluxes estimated at the current model time t can only be used to calculate the scalar values at time $t + \Delta t$ and hence it intrinsically works with FIT time-marching.

Additional time splitting method

To increase a bit more the maximum CFL number, an additional time-splitting can be activated for the scalar advection ($LSPLIT\_CFL$), in order to increase the time step of the rest of the model and to follow a CFL strictly less than 1 for PPM (a threshold of $XSPLIT\_CFL = 0.8$ is proposed) (Fig. 7.2). This smaller time step for PPM can evolve during the run as a function of the CFL number.

7.4 References

Blossey, P. N. and D. R. Durran, 2008: Selective Monotonicity Preservation in Scalar Advection, *J. Comput. Phys.*, **227**, 5160—5183.

Carpenter, R. L., K. K. Droegemeier, P. R. Woodward, and C. E. Hane, 1990: Application of the Piecewise Parabolic Method (PPM) to Meteorological Modeling, *Mon. Wea. Rev.*, **118**, 586–612.

Castro, M., B. Costa, and W. S., Don, 2011: High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws, *J. Comput. Phys.*, **230**, 1766—1792.

Colella, P. and P. R. Woodward, 1984: The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations, *J. Comput. Phys.*, **54**, 174–201.

Durran, D. R., and J. B. Klemp, 1983: A compressible model for the simulation of moist mountain waves, *Mon. Wea. Rev.*, **111 (12)**, 2341-–2361.

Easter, R. C., 1993: Two Modified Versions of Bott's Positive-Definite Numerical Advection Scheme, *Mon. Wea. Rev.*, **121**, 297–304.

Gal-Chen, T., and R. C. Somerville, 1975: On the use of a coordinate transformation for the solution of the Navier-Stokes equations. *J. Comput. Phys.*, **17 (2)**, 209-–228.

Klemp, J. B., W. C. Skamarock, and O. Fuhrer, 2003: Numerical consistency of metric terms in terrain-following coordinates. *Mon. Wea. Rev.*, **131 (7)**, 1229-–1239.

Jiang, G.-S., and C.-W. Shu, 1995: Efficient implementation of weighted ENO schemes. Tech. rep., DTIC Document.

Lin, S. and R. B. Rood, 1996: Multidimensional Flux-Form Semi-Lagrangian Transport Schemes, *Mon. Wea. Rev.*, **124**, 2046–2070.

Pressel, K. G., C. M. Kaul, T. Schneider, Z. Tan, and S. Mishra, 2015: Large-eddy simulation in an anelastic framework with closed water and entropy balances. *J. Adv. Model. Earth Syst.*, **7 (3)**, 1425-–1456.

Ricard, D., C. Lac, S. Riette, R. Legrand, and A. Mary, 2013: Kinetic energy spectra characteristics of two convection-permitting limited-area models AROME and Meso-NH. *Quart. J. Roy. Meteor. Soc.*, **139 (674)**, 1327-–1341.

Rood, R. B., 1987: Numerical advection algorithms and their role in atmospheric transport and chemistry models. *Rev. Geophys.*, **25**, 71–100.

Schär, C., D. Leuenberger, O. Fuhrer, D. Lüthi, and C. Girard, 2002: A new terrain-following vertical coordinate formulation for atmospheric prediction models. *Mon. Wea. Rev.*, **130 (10)**, 2459-–2480.

Shu, C.-W., 1998: Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in Advanced Numerical Approximation of Nonlinear Hyperbolic Equations: Lectures given at the 2nd Session of the Centro Internazionale Matematico Estivo (C.I.M.E.) held in Cetraro, Italy, June 23–28, 1997 (ed. A. Quarteroni), Springer, Berlin, Heidelberg, Germany.

Skamarock, W. C., 2006: Positive-Definite and Monotonic Limiters for Unrestricted-Time-Step Transport Schemes, *Mon. Wea. Rev.*, **134**, 2241–2250.

Strang, G., 1968: On the Construction and Composition of Difference Schemes, *SIAM J. Numer. Anal.*, **5**, 506–517.

Tan, K.-A., R. Morison, and L. Leslie, 2005: A comparison of high-order explicit and non oscillatory finite difference advection schemes for climate and weather models. *Meteor. Atmos. Phys.*, **89 (1-4)**, 251-–267.

Wang, R., and R. J. Spiteri, 2007: Linear instability of the fifth-order WENO method. *SIAM J. Numer. Anal.*, **45 (5)**, 1871-–1901.

Wicker, L. J and W. C. Skamarock, 2002: Time-splitting methods for elastic models using forward time schemes. *Mon. Wea. Rev.*, **130**, 2088–2097.

Xue, M., and A. J. Thorpe, 1991: A mesoscale numerical model using the nonhydrostatic pressure based sigma-coordinate equations: Model experiments with dry mountain flows. *Mon. Wea. Rev.*, **119 (5)**, 1168-–1185.

Chapter 8

Numerical Diffusion Terms

Contents

As in most numerical models, it is necessary to prevent the occurrence of numerical waves due to the inaccurate representation of the dynamical processes and reflection at the top or lateral boundaries. This is done in a fairly classical way, through (i) a weak background diffusion, (ii) a top absorbing layer, and (iii) a lateral sponge zone. Note that in these three regions, the flow is relaxed towards the ”large-scale” values, which may be non uniform in space, and time dependent. For idealized runs, the user may of course choose uniform and steady large-scale values.

8.1 Background diffusion

8.1.1 Diffusion operator

A diffusion operator is applied to the *fluctuations* of the prognostic variables ϕ. The fluctuations are defined here as the departure from the large scale value ϕ_{LS}. The diffusion operator is a fourth-order operator (δ_{x^4}) used everywhere except at the first interior grid point where a second-operator (δ_{x^2}) is substituted in the case of non-periodic boundary conditions. The background diffusion source for any prognostic variable noted ϕ is

$$S_{BD} = -K4\big[\delta_{x^4}[\phi(t-\delta t) - \phi_{LS}] + \delta_{y^4}[\phi(t-\delta t) - \phi_{LS}]\big]$$

or

$$S_{BD} = +K2\big[\delta_{x^2}[\phi(t-\delta t) - \phi_{LS}] + \delta_{y^2}[\phi(t-\delta t) - \phi_{LS}]\big]$$

where $K2$ and $K4$ are positive coefficients, and ϕ_{LS} represents the Large Scale value of the considered variable.

8.1.2 Choice of the diffusion coefficient

Let us consider a single harmonic wave defined by:

$$\phi(x,t) = \Phi(t)e^{ikx}$$

where $\Phi(t)$ is the wave amplitude and k the wavenumber.
The application of a second-order diffusion operator during N time steps leads to:

$$\phi(x,t+N\Delta t) = \Phi(t)[1-2\frac{K_2\Delta t}{{d_{xx}}^2}(1-coskd_{xx})]^N$$

where Δt is the time step, ${d_{xx}}^2$ the grid interval, and $K2 = K_2/{d_{xx}}^2$ the diffusion coefficient. The time T_2 at which the initial wave is damped by e^{-1} is then:

$$T_2 = N\Delta t = \frac{-\Delta t}{\ln[1-2\frac{K_2\Delta t}{{d_{xx}}^2}(1-coskd_{xx})]}$$

which can be approximated by:

$$T_2 \simeq \frac{{d_{xx}}^2}{2K_2(1-coskd_{xx})}$$

The corresponding time in the case of a fourth-order diffusion operator is given by:

$$T_4 = N\Delta t = \frac{-\Delta t}{\ln[1-4\frac{K_4\Delta t}{{d_{xx}}^4}(1-coskd_{xx})^2]}$$

which can be approximated by:

$$T_4 \simeq \frac{{d_{xx}}^4}{4K_4(1-coskd_{xx})^2}$$

If k is the wavenumber associated to the nd_{xx} wavelength, T_2 and T_4 are given by

$$T_2(n) \simeq \frac{{d_{xx}}^2}{2K_2(1-cos(2\pi/n))} \tag{8.1}$$

$$T_4(n) \simeq \frac{{d_{xx}}^4}{4K_4(1-cos(2\pi/n))^2} \tag{8.2}$$

To set up the diffusion coefficients, it might be more convenient to specify T_2 or T_4 rather than K_2 or K_4. T_2 and T_4 can be more easily related to the physical processes being studied. From previous experience, $T_4(2)$ was set to 10-15 mn in the case of PBL convective rolls, 20-30 mn for moist convection, 1-2 hours for orographic flows.
For a specified wavelength n_0d_{xx}, an equivalent damping timescale with a second-order or a fourth-order diffusion scheme requires

$$K_4 = -\frac{K_2}{2}\frac{{d_{xx}}^2}{1-cos(2\pi/n_0)}. \tag{8.3}$$

The same arguments hold for the diffusion in the y direction.

In the code n_0 is set equal to 2 to select the highest wavenumber and so the user specifies $T_4(2)$. Noting that $K2 = K_{2x}/{d_{xx}}^2 = K_{2y}/{d_{yy}}^2$, $K4 = K_{4x}/{d_{xx}}^4 = K_{4y}/{d_{yy}}^4$, and according to (1), (2) and (3) $K4$ and $K2$ are then given by:

$$K4 = \frac{1}{16T_4(2)}$$

and

$$K2 = K4(1 - cos(2\pi/n_0)) = 4K4$$

8.1.3 Horizontal diffusion over mountains

Calculating the horizontal diffusion along terrain-following surfaces may introduce serious errors over mountainous terrain, particularly for atmospheric properties having a strong vertical gradient. Temperature diffusion along such surfaces, for example, tends to cool the air in valleys and to heat that air above mountains and likewise, diffusion of the water mixing ratio tends to dry the air in valleys and to moisten the atmosphere above mountains. To reduce the unphysical impact of horizontal diffusion applied on terrain following surfaces, the modifications suggested in Zängl (2002) have been implemented in the code. At model levels sufficiently far away from the ground, vertical interpolation is used to compute diffusion truly horizontally when the coordinate surfaces are sloping. Close to the ground, where truly horizontal computation of diffusion is not possible without intersecting the topography, diffusion is treated differently for momentum, moisture and temperature. A simple transition to diffusion along the coordinate surface is chosen for momentum. For the moisture variables, one-sided truly horizontal diffusion is used in combination with orography-adjusted diffusion along the $\hat{z}$ surfaces. This orography adjustment is achieved by strongly reducing the diffusion coefficient when the grid points involved in the diffusion computation are located at greatly different heights. For temperature, one sided horizontal diffusion is not used because it affects the slope wind circulation in an unphysical way. However, a temperature gradient is applied to the terrain-following part of diffusion so as to ensure that diffusion behaves neutrally with respect to the local vertical temperature gradient.

8.2 Top absorbing layer

To prevent spurious reflection from the model top boundary, an absorbing layer in which damping increases with height occupies the top fraction of the domain. A Rayleigh damping has been chosen, it is applied on the three components of the wind and on the thermodynamical variable. Only the perturbations of a variable from its local large scale values are damped on $\bar{z}$ surfaces. In the absorbing layer, the implicit damping source for any variable ϕ is written as:

$$S_{AL} = -K_{AL}(\bar{z})\big[\phi(t + \Delta t) - \phi_{LS}\big] = -KAL(\bar{z})\big[\phi(t - \Delta t) - \phi_{LS}\big]$$

where

$$KAL = \frac{K_{AL}(\bar{z})}{1 + 2\Delta t K_{AL}(\bar{z})}$$

and where $K_{AL}(\bar{z})$ is given by

$$K_{AL}(\bar{z}) = K_{AL}(H)\sin^2\left(\frac{\pi}{2}\frac{\bar{z}-\bar{z}_{alb}}{H-\bar{z}_{alb}}\right) \qquad for \quad \bar{z} \geq \bar{z}_{alb}$$

with $\bar{z}_{alb}$ the Gal-Chen and Sommerville height of the absorbing layer base and ϕ_{LS} the relaxation value of ϕ. In the first version of the model the relaxation fields are the initial fields and the maximum damping rate $K_{AL}(H)$ must be provided for each model run.

8.3 Lateral sponge zone

An additional sponge zone is inserted close to the lateral boundaries to either damp outward propagating waves or slowly incorporate inward propagating larger scale waves. A first order damping rate has been retained and its application to any prognostic variable ϕ leads to a source term of the form:

$$S_{SZ} = -K_{SZ}(\bar{x},\bar{y})\big[\phi(t+\Delta t)-\phi_{LS}\big] = -KSZ(\bar{x},\bar{y})\big[\phi(t-\Delta t)-\phi_{LS}\big]$$

where

$$KSZ = \frac{K_{SZ}(\bar{x},\bar{y})}{1+2\Delta t K_{SZ}(\bar{x},\bar{y})}.$$

The damping coefficient $KSZ(\bar{x},\bar{y})$ is non-zero in a rim zone of width $rim_{\bar{x}}$ and $rim_{\bar{y}}$ (in each x and y direction, respectively) following immediately the lateral boundaries. For example, near the left (x) lateral boundary, the damping coefficient has the generic form:

$$KSZ(\bar{x}) = \begin{cases} K_{SZ}^{max}\sin^2\left(\frac{\pi}{2}\frac{\bar{x}-rim_{\bar{x}}}{rim_{\bar{x}}}\right) & \text{if } 0 < \bar{x} < rim_{\bar{x}}, \\ 0 & \text{otherwise,} \end{cases}$$

In the four corners of the domain of simulation, there is a smooth transition between the pure $\bar{x}$ and pure $\bar{y}$ dependencies of the damping coefficient KSZ that is obtained in the following manner:

$$KSZ(\bar{x},\bar{y}) = \begin{cases} K_{SZ}^{max} & \text{if } 1 \leq d_{rim}^2, \\ K_{SZ}^{max}\sin^2\left(\frac{\pi}{2}d_{rim}\right) & \text{if } d_{rim}^2 \leq 1, \\ 0 & \text{if } \bar{x} \geq rim_{\bar{x}} \text{ and } \bar{y} \geq rim_{\bar{y}} \end{cases}$$

where $d_{rim} = \sqrt{\left(\frac{\bar{x}-rim_{\bar{x}}}{rim_{\bar{x}}}\right)^2 + \left(\frac{\bar{y}-rim_{\bar{y}}}{rim_{\bar{y}}}\right)^2}$. The maximum value of the relaxation coefficient K_{SZ}^{max} and the rim zone depths $rim_{\bar{x}}$ and $rim_{\bar{y}}$ are prescribed externally by the user.

8.4 References

Zängl, G., 2002: An improved method for computing horizontal diffusion in a sigma-coordinate model and its application to simulations over mountainous topography. *Mon. Wea. Rev.*, **130**, 1423-1432

Chapter 9

The Pressure Problem

Contents

9.1 Continuous pressure equation

9.1.1 Full elliptic problem

In the Meso-NH model formulation, the equations of motion are expressed as:

$$\frac{\partial \tilde{\rho} u}{\partial t} = S_x - \tilde{\rho}\frac{\partial \Phi}{\partial x} \tag{9.1}$$

$$\frac{\partial \tilde{\rho} v}{\partial t} = S_y - \tilde{\rho}\frac{\partial \Phi}{\partial y} \tag{9.2}$$

$$\frac{\partial \tilde{\rho} w}{\partial t} = S_z - \tilde{\rho}\frac{\partial \Phi}{\partial z} \tag{9.3}$$

where S_x, S_y and S_z are the source terms apart from the pressure terms; $\tilde{\rho} = \rho_{dref} J$; $\Phi = C_p\, \theta_{v\,ref}\, \Pi'$ is called here the "pressure" for short. The anelastic constraint reads

$$\frac{\partial \tilde{\rho} U^c}{\partial \overline{x}} + \frac{\partial \tilde{\rho} V^c}{\partial \overline{y}} + \frac{\partial \tilde{\rho} W^c}{\partial \overline{z}} = 0 \tag{9.4}$$

If we combine Eqs. (9.1)–(9.3) in order to build and nullify the divergence of the wind at the future instant in (9.4), we obtain a diagnostic equation for the pressure:

$$GDIV(\tilde{\rho}\overrightarrow{\nabla}\Phi) = GDIV\left(\overrightarrow{S} - \frac{\tilde{\rho}\boldsymbol{U}_{(t-\Delta t)}}{2\Delta t}\right) \tag{9.5}$$

where

$$GDIV(\boldsymbol{B}) = \frac{\partial B^{c1}}{\partial \overline{x}} + \frac{\partial B^{c2}}{\partial \overline{y}} + \frac{\partial B^{c3}}{\partial \overline{z}}$$

B^{c1}, B^{c2}, B^{c3} are the contravariant components of $\overrightarrow{B}$ and can be written as:

$$B^{c1} = \frac{B_1}{d_{xx}} \tag{9.6}$$

$$B^{c2} = \frac{B_2}{d_{yy}} \tag{9.7}$$

$$B^{c3} = \frac{1}{d_{zz}}\left[B_3 - \frac{B_1}{d_{xx}}d_{zx} - \frac{B_2}{d_{yy}}d_{zy}\right] \tag{9.8}$$

where (B_1, B_2, B_3) are the components of $\boldsymbol{B}$ along $(\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k})$.
$\overrightarrow{\nabla}$ is defined by: $\overrightarrow{\nabla} = \boldsymbol{i}\frac{\partial}{\partial x} + \boldsymbol{j}\frac{\partial}{\partial y} + \boldsymbol{k}\frac{\partial}{\partial z}$ where

$$\begin{aligned}
\frac{\partial}{\partial x} &= \frac{1}{d_{xx}}\frac{\partial}{\partial \overline{x}} - \frac{d_{zx}}{d_{xx}d_{zz}}\frac{\partial}{\partial \overline{z}} \\
\frac{\partial}{\partial y} &= \frac{1}{d_{yy}}\frac{\partial}{\partial \overline{y}} - \frac{d_{zy}}{d_{yy}d_{zz}}\frac{\partial}{\partial \overline{z}} \\
\frac{\partial}{\partial z} &= \frac{1}{d_{zz}}\frac{\partial}{\partial \overline{z}}
\end{aligned}$$

For convenience, we define the $QLAP$ operator by: $QLAP = GDIV(\tilde{\rho}\overrightarrow{\nabla})$.
Note that because an iterative method is used to solve the pressure equation, the residual term $GDIV(\tilde{\rho}\boldsymbol{U}_{(t-\Delta t)})$ must be kept and injected in the right-hand side of (9.5) to get the most accurate solution for Φ.

Top and bottom boundaries

For the highest and the lowest vertical level, a Neumann condition is imposed:

$$\tilde{\rho}\boldsymbol{n}\cdot\overrightarrow{\nabla}\Phi = \boldsymbol{n}\cdot\overrightarrow{S} \tag{9.9}$$

Lateral boundaries

We allow for several possibilities:

- For a periodic simulation domain, the lateral boundary conditions are a periodisation of the pressure field.
- For an open area, the u and v evolution on the boundaries are performed by the Sommerfeld's equations:

$$\frac{\partial\tilde{\rho}u}{\partial t} = -c_x\frac{\partial\tilde{\rho}u}{\partial x} \tag{9.10}$$

$$\frac{\partial\tilde{\rho}v}{\partial t} = -c_y\frac{\partial\tilde{\rho}v}{\partial y} \tag{9.11}$$

  where c_x and c_y are the phase velocities of the fastest waves in respectively x and y directions and determined by the Orlanski's method (1976) or fixed (Klemp and Wilhelmson 1978).

- If a Davies' scheme is used, the tendency on the boundary is given by:

$$\frac{\partial\tilde{\rho}u}{\partial t} = \left(\frac{\partial\tilde{\rho}u}{\partial t}\right)_{Large\ Scale} \tag{9.12}$$

$$\frac{\partial\tilde{\rho}v}{\partial t} = \left(\frac{\partial\tilde{\rho}v}{\partial t}\right)_{Large\ Scale} \tag{9.13}$$

- If we use a Carpenter's scheme (Carpenter 1982), the tendency on the boundary is expressed as:

$$\frac{\partial\tilde{\rho}u}{\partial t} = \left(\frac{\partial\tilde{\rho}u}{\partial t}\right)_{Large\ Scale} - c_x\left[\frac{\partial\tilde{\rho}u}{\partial x} - \left(\frac{\partial\tilde{\rho}u}{\partial x}\right)_{Large\ Scale}\right] \tag{9.14}$$

$$\frac{\partial\tilde{\rho}v}{\partial t} = \left(\frac{\partial\tilde{\rho}v}{\partial t}\right)_{Large\ Scale} - c_y\left[\frac{\partial\tilde{\rho}v}{\partial y} - \left(\frac{\partial\tilde{\rho}v}{\partial y}\right)_{Large\ Scale}\right] \tag{9.15}$$

Therefore, in every case, the temporal evolution of the horizontal momentum is known at the boundary by these equations and from (9.1)– (9.2), we can infer the normal component of the pressure gradient when the sources components $\boldsymbol{S} \cdot \boldsymbol{n}$ is known. The result is a non-homogeneous Neumann condition for the full elliptic problem. We can make this condition homogeneous if we replace in (9.1-9.2) the source terms S_x and S_y at the boundaries by the prescribed values of $\frac{\partial \tilde{\rho} u}{\partial t}$ and $\frac{\partial \tilde{\rho} v}{\partial t}$ respectively. This leads to an algebraic simplification in the pressure equation (9.5). The homogeneous boundary condition for the pressure field is:

$$\frac{\partial \Phi}{\partial n} = 0 \tag{9.16}$$

We want to solve efficiently and accurately the linear system (9.5) where the source term $\overrightarrow{S}$ is known and with the previous boundary conditions. As it is difficult and time-consuming to invert such a large system, we have implemented three iterative methods in order to offer a good compromise between versatility and efficiency. The first two methods are described in Golub and Meurant (1983): the first one consists in a conjugate gradient algorithm and the second one is the Richardson's method. The third method was suggested by Skamarock et al. (1997). The convergence of such algorithms is considerably speeded-up with the use of preconditioning techniques. Here, we precondition with the "flat" problem (Bernardet 1995), i.e. without orography, which compresses the eigenvalue spectrum of the original problem, leaving only a few eigenvalues significantly different from 1, resulting in much faster convergence. For all the methods, the first guess is also set equal to the solution of the flat problem.

9.1.2 Conjugate gradient algorithm

The Golub and Meurant's version of the conjugate gradient (CG) is devoted to symmetric matrices. As our problem is non symmetric, we use a generalization of the CG called ORTHOMIN, detailled in Young and Jea (1980). Considering the linear system:

$$Qx = y \tag{9.17}$$

where x is unknown. In our context, Q is the $QLAP$ operator defined before. Preconditionning the previous system with the "flat" problem gives:

$$F^{-1}Qx = F^{-1}y \tag{9.18}$$

where F^{-1} is the preconditioner with F defined as the $QLAP$ operator where the topography is neglected (flat problem).
The ORTHOMIN algorithm can be put in the following form (Kapitza and Eppel 1992):

$$\begin{aligned}
x^{(n+1)} &= x^{(n)} + \lambda_n p^{(n)} \\
\lambda_n &= \frac{(\delta^{(n)}, F^{-1}Qp^{(n)})}{(F^{-1}Qp^{(n)}, F^{-1}Qp^{(n)})} \\
\delta^{(n)} &= F^{-1}(y - Qx^{(n)}) \\
p^{(0)} &= \delta^{(0)} \\
p^{(n)} &= \delta^{(n)} + \alpha_n p^{(n-1)} \\
\alpha_n &= -\frac{\left(F^{-1}Q\delta^{(n)}, F^{-1}Qp^{(n-1)})\right)}{(F^{-1}Qp^{(n-1)}, F^{-1}Qp^{(n-1)})}
\end{aligned}$$

Here $x^{(n)}$ is the n-th iteration of x and (.,.) is the dot product of two vectors. Concerning the computing time, at each iteration, we have to invert three times the F matrix.

9.1.3 Richardson's method

This technique allows to divide by 3 the number of matrix inversion, compared to the CG algorithm. Furthermore, this method, including a relaxation factor α, allows an under-relaxation which is useful to improve the solver convergence in the context of simulations with large slopes. For example, for slopes greater than 1, preliminary tests have shown that the optimal value of α is equal to 1 for such simulations. If we assume the following linear system preconditioned with the inverse of the "flat" problem:

$$F^{-1}Qx = F^{-1}y \tag{9.19}$$

the Richardson's method reads:

$$x^{(n+1)} \;=\; x^{(n)} + \alpha(F^{-1}y - F^{-1}Qx^{(n)})$$

where $x^{(n)}$ is the n-th iteration of x.

9.1.4 Preconditioned conjugate-residual algorithm

The method was suggested by Skamarock et al. (1997). It is based on a conjugate-residual algorithm that is accelerated by a "flat Laplacian" preconditionner (see next section). It is interpreted physically as a finite-difference scheme for integrating, in a pseudo-time τ to a steady state, a damped oscillation equation with forcing equal to the residual error $\delta \equiv Qx - y$ of the linear elliptic problem $Qx = y$, that is:

$$\frac{\partial^2 Fx}{\partial \tau^2} + \frac{1}{T}\frac{\partial Fx}{\partial \tau} = \delta,$$

where T is an arbitrary damping coefficient and Fx the preconditioner operator so $F^{-1}x$ is the $QLAP$ operator. After initializing the residual δ^0, $p^0 = F^{-1}\delta^0$ and Qp^0, the resulting algorithm is used:

$$\begin{aligned}
\lambda &= -\frac{< \delta^{n-1}\dot{Q}p^{n-1} >}{< Qp^{n-1}\dot{Q}p^{n-1} >}\\
x^n &= x^{n-1} + \lambda p^{n-1}\\
\delta^n &= \delta^{n-1} + \lambda Qp^{n-1}\\
q^n &= F^{-1}\delta^n\\
\alpha &= -\frac{< Qq^n\dot{Q}p^{n-1} >}{< Qp^{n-1}\dot{Q}p^{n-1} >}\\
p^n &= q^n + \alpha p^{n-1}\\
Qp^n &= Qq^n + \alpha Qp^{n-1}
\end{aligned}$$

where $<\dot{}>$ is the inner product. The iteration stops when $||r^n|| \leq \epsilon$. The method is the most efficient converging one compared to the Richardson and CG algorithms. It is now set as the default value of pressure solver algorithm in Meso-NH (`CPRESOPT = 'CRESI'`).

9.1.5 Flat operator

Both iterative methods require the inversion of the operator $QLAP$ for the ”flat” problem. ”Flat” means the topography is neglected, which induces that $x = \overline{x}$, $y = \overline{y}$, $z = \overline{z}$. This involves that d_{zx}, d_{zy} vanish. Here, we consider a regular grid and the stretching functions $\mathcal{D}_x(\widehat{x})$, $\mathcal{D}_y(\widehat{y})$, $\mathcal{D}_z(\widehat{z})$ are neglected in order to use a FFT decomposition and will be implemented later. Thus, we obtain for the flat operator F:

$$F\Phi = \frac{\partial}{\partial \overline{x}}\left(\frac{1}{d_{xx}}\tilde{\rho}\frac{1}{d_{xx}}\frac{\partial \Phi}{\partial \overline{x}}\right) + \frac{\partial}{\partial \overline{y}}\left(\frac{1}{d_{yy}}\tilde{\rho}\frac{1}{d_{yy}}\frac{\partial \Phi}{\partial \overline{y}}\right) + \frac{\partial}{\partial \overline{z}}\left(\frac{1}{d_{zz}}\tilde{\rho}\frac{1}{d_{zz}}\frac{\partial \Phi}{\partial \overline{z}}\right) \tag{9.20}$$

In the flat problem, $\tilde{\rho}$ only depends on z, (9.20) can be rewritten as:

$$F\Phi =< \tilde{\rho} >_{x,y}\left[\frac{1}{< {d_{xx}}^2 >_{x,y}}\frac{\partial^2 \Phi}{\partial \overline{x}^2} + \frac{1}{< {d_{yy}}^2 >_{x,y}}\frac{\partial^2 \Phi}{\partial \overline{y}^2}\right] + \frac{\partial}{\partial Z}\left(< \tilde{\rho} >_{x,y}\frac{1}{< {d_{zz}}^2 >_{x,y}}\frac{\partial \Phi}{\partial Z}\right) \tag{9.21}$$

where $Z =< \overline{z} >_{xy}$.
We have applied a spatial averaging on a whole xy plan because $\tilde{\rho}$ and $\overline{z}$ are functions of z. When the orography is neglected, we must recover a function of only $\overline{z}$ (this is also necesary to lead to a separable problem). A Fast Fourier Transform (Schumann and Sweet 1988) is used for the inversion of the horizontal part of F. The form of this FFT depends on the lateral boundary conditions. The vertical part of F leads to a tridiagonal matrix and thus a classical inversion algorithm is applied (the double sweep method, see ”Numerical Recipes”).

9.2 Discrete pressure equation

9.2.1 Full elliptic problem

The continuous equation (9.5) is:

$$GDIV(\rho\overrightarrow{\nabla}\Phi) = GDIV\left(\overrightarrow{S} - \frac{\tilde{\rho}\boldsymbol{U}_{(t-\Delta t)}}{2\Delta t}\right) \tag{9.22}$$

where $\overrightarrow{\nabla}\Phi = \frac{\partial \Phi}{\partial x}\boldsymbol{i} + \frac{\partial \Phi}{\partial y}\boldsymbol{j} + \frac{\partial \Phi}{\partial z}\boldsymbol{k}$ is discretized by:

$$\frac{\partial \Phi}{\partial x} \rightarrow \frac{1}{d_{xx}}\left[\delta_x\Phi - \overline{d_{zx}\overline{\frac{\delta_z\Phi}{d_{zz}}}^{x}}^{z}\right] \tag{9.23}$$

$$\frac{\partial \Phi}{\partial y} \rightarrow \frac{1}{d_{yy}}\left[\delta_y\Phi - \overline{d_{zy}\overline{\frac{\delta_z\Phi}{d_{zz}}}^{y}}^{z}\right] \tag{9.24}$$

$$\frac{\partial \Phi}{\partial z} \rightarrow \frac{\delta_z\Phi}{d_{zz}} \tag{9.25}$$

where δ_x, δ_y, δ_z are finite difference operators.
The discrete form of the contravariant components of a vector $\boldsymbol{B}(B_1, B_2, B_3)$ is:

$$B^{c1} = \frac{B_1}{d_{xx}} \tag{9.26}$$

$$B^{c2} = \frac{B_2}{d_{yy}} \tag{9.27}$$

$$B^{c3} = \frac{1}{d_{zz}}\left[B_3 - \overline{\left(\overline{\left(\frac{B_1}{d_{xx}}\right)}^{z} d_{zx}\right)}^{x} - \overline{\left(\overline{\left(\frac{B_2}{d_{yy}}\right)}^{z} d_{zy}\right)}^{y}\right] \tag{9.28}$$

(B_1, B_2, B_3) are the components of $\boldsymbol{B}$ along $(\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k})$ and thus we have:

$$GDIV(\boldsymbol{B}) = \delta_x B^{c1} + \delta_y B^{c2} + \delta_z B^{c3} \tag{9.29}$$

We note $QLAP$ the numerical discretization of $QLAP$ and we obtain:

$$\begin{aligned}
QLAP(\Phi) = GDIV(\tilde{\rho}\boldsymbol{\nabla}\Phi) &= \delta_x\left[\overline{\tilde{\rho}}^{x}\frac{\delta_x\Phi}{{d_{xx}}^2}\right] - \delta_x\left[\overline{\tilde{\rho}}^{x}\frac{1}{{d_{xx}}^2}\overline{\left(d_{zx}\overline{\left(\frac{\delta_z\Phi}{d_{zz}}\right)}^{x}\right)}^{z}\right] \\
&+ \delta_y\left[\overline{\tilde{\rho}}^{y}\frac{\delta_y\Phi}{{d_{yy}}^2}\right] - \delta_y\left[\overline{\tilde{\rho}}^{y}\frac{1}{{d_{yy}}^2}\overline{\left(d_{zy}\overline{\left(\frac{\delta_z\Phi}{d_{zz}}\right)}^{y}\right)}^{z}\right] + \delta_z\left[\overline{\tilde{\rho}}^{z}\frac{\delta_z\Phi}{{d_{zz}}^2}\right] \\
&- \delta_z\left[\frac{1}{d_{zz}}\overline{\left\{d_{zx}\overline{\left\{\overline{\tilde{\rho}}^{x}\frac{\delta_x\Phi}{{d_{xx}}^2} - \overline{\tilde{\rho}}^{x}\frac{1}{d_{xx}}\overline{\left(d_{zx}\overline{\left(\frac{\delta_z\Phi}{d_{zz}}\right)}^{x}\right)}^{z}\right\}}^{z}\right\}}^{x}\right] \\
&- \delta_z\left[\frac{1}{d_{zz}}\overline{\left\{d_{zy}\overline{\left\{\overline{\tilde{\rho}}^{y}\frac{\delta_y\Phi}{{d_{yy}}^2} - \overline{\tilde{\rho}}^{y}\frac{1}{d_{yy}}\overline{\left(d_{zy}\overline{\left(\frac{\delta_z\Phi}{d_{zz}}\right)}^{y}\right)}^{z}\right\}}^{z}\right\}}^{y}\right]
\end{aligned}$$

For the first inner points, the calculation of the pseudo-Laplacian operator $QLAP$ requires the evaluation of pressure outside the physical domain.

Vertical boundaries

The discrete form of the boundary equation (9.9) is expressed as:

$$(\boldsymbol{S})^{c3} = \left(\tilde{\rho}\overrightarrow{\nabla}\Phi\right)^{c3} \tag{9.30}$$

where

$$\left(\tilde{\rho}\overrightarrow{\nabla}\Phi\right)^{c3} = \frac{1}{d_{zz}}\left[\left(\tilde{\rho}\frac{\partial\Phi}{\partial z}\right) - \overline{\left(\overline{\left(\frac{1}{d_{xx}}\left(\tilde{\rho}\frac{\partial\Phi}{\partial x}\right)\right)}^{z} d_{zx}\right)}^{x} - \overline{\left(\overline{\left(\frac{1}{d_{yy}}\right)\left(\tilde{\rho}\frac{\partial\Phi}{\partial y}\right)}^{z} d_{zy}\right)}^{y}\right] \tag{9.31}$$

and the pressure gradient expressions are given by (9.23-9.25). In (9.31), the vertical spatial average requires the points located at $z = -\Delta z/2$. In this case, we copy the value of the first pressure gradient over the surface:

$$\left(\frac{\partial \Phi}{\partial x}\right)_{-\frac{\Delta z}{2}} = \left(\frac{\partial \Phi}{\partial x}\right)_{\frac{\Delta z}{2}}$$
$$\left(\frac{\partial \Phi}{\partial y}\right)_{-\frac{\Delta z}{2}} = \left(\frac{\partial \Phi}{\partial y}\right)_{\frac{\Delta z}{2}}$$

Horizontal boundaries

Here, we have to consider how to evaluate the pressure gradient terms at the boundary (wind point), i.e. $i = 1 + jphext, i = imax + 1 + jphext, j = 1 + jphext, j = jmax + 1 + jphext$ and this for $kmax + 1 + jpvext > k > jpvext$. $jphext$ and $jpvext$ are the variables used to extend the number of horizontal and vertical outer points respectively ($jphext \geq 1$, $jpvext \geq 1$). We have to distinguish three cases:
•Cyclic case
The periodic condition reads:

$$\begin{aligned} \Phi(jphext, j, k) &= \Phi(imax + jphext, j, k) \\ \Phi(imax + 1 + jphext, j, k) &= \Phi(1 + jphext, j, k) \end{aligned} \tag{9.32}$$

in the x direction and

$$\begin{aligned} \Phi(i, jphext, k) &= \Phi(i, jmax + jphext, k) \\ \Phi(i, jmax + 1 + jphext, k) &= \Phi(i, 1 + jphext, k) \end{aligned} \tag{9.33}$$

in the y direction.
•Open and Davies' cases
The x lateral boundary condition $\frac{\partial \Phi}{\partial x} = 0$ for $i = 1 + jphext, i = imax + 1 + jphext$ $(jmax + 1 + jphext > j > jphext)$ becomes:

$$\frac{\delta_x \Phi}{d_{xx}} + \frac{1}{d_{xx}} \overline{\left(\overline{\left(\frac{\delta_z \Phi}{d_{zz}}\right)}^{x} d_{zx} \right)}^{z} = 0$$

The y lateral boundary condition $\frac{\partial \Phi}{\partial y} = 0$ for $j = 1 + jphext, j = jmax + 1 + jphext$ $(imax + 1 + jphext > i > jphext)$ becomes:

$$\frac{\delta_y \Phi}{d_{yy}} + \frac{1}{d_{yy}} \overline{\left(\overline{\left(\frac{\delta_z \Phi}{d_{zz}}\right)}^{y} d_{zy} \right)}^{z} = 0 \tag{9.34}$$

•Mixed boundary conditions
In this case, we have to combine the previous set of equations (9.32)-(9.34) to be consistent with the specification of the horizontal boundaries. For example, if we assume a periodic condition

along y and a Davies' condition along x we use (9.33) and (9.34).

Edges

The edges points at $k = jpvext$, (i.e. $\Phi(jphext, j, jpvext)$, $\Phi(imax + 1 + jphext, j, jpvext)$, $\Phi(i, jphext,\ jpvext)$, $\Phi(i, jmax + 1 + jphext, 1)$, where $imax + jphext + 1 > i > jphext$ and $jmax + jphext + 1 > j > jphext$) and their corresponding points located at the coordinate $k = kmax + 1 + jpvext$, are necessary for the calculation of the pseudo-Laplacian of the first inner grid points. So we have imposed boundary conditions consistent with the Neumann staggered condition defined for the horizontal boundaries. These conditions are expressed as:

- for $i = 1 + jphext, imax + 1 + jphext$ and $z = \Delta z/2, H - \Delta z/2$:

$$\frac{\partial \Phi}{\partial x} = 0$$

- for $i = 1 + jphext, imax + 1 + jphext$ and $z = 0, H$:

$$\delta_z \frac{\partial \Phi}{\partial x} = 0$$

- for $j = 1 + jphext, jmax + 1 + jphext$ and $z = \Delta z/2, H - \Delta z/2$:

$$\frac{\partial \Phi}{\partial y} = 0$$

- for $j = 1 + jphext, jmax + 1 + jphext$ and $z = 0, H$:

$$\delta_z \frac{\partial \Phi}{\partial y} = 0$$

The corner points, i.e. $\Phi(jphext, jphext, jpvext)$, $\Phi(imax + 1 + jphext, jphext, jpvext)$, $\Phi(jphext,\ jmax + 1 + jphext, jpvext)$, $\Phi(imax + 1 + jphext, jmax + 1 + jphext, jpvext)$ and their corresponding points at $k = kmax + 1 + jpvext$, are not required for the resolution of the pressure equation.

9.2.2 Flat operator

We want to invert the following system (9.35) where Y is known and F is the flat operator.

$$F\Phi = Y \tag{9.35}$$

The discretized form of the flat operator F (9.20) is written as:

$$F\Phi = < \tilde{\rho} >_{x,y} \left[\frac{1}{< {d_{xx}}^2 >_{xy}} \delta_{x^2}\Phi + \frac{1}{< {d_{yy}}^2 >_{xy}} \delta_{y^2}\Phi \right] + \delta_Z \overline{< \tilde{\rho} >_{x,y}}^{z} \frac{1}{< {d_{zz}}^2 >_{xy}} \delta_Z \Phi \tag{9.36}$$

The method consists in treating the horizontal part of F in the Fourier space. First, we compute the FFT of Y noted $\hat{Y}$. Then, we introduce the horizontal Fourier decomposition of F completed by its classical vertical part. This results in $imax \times jmax$ tridiagonal matrices where every matrix corresponds to a different horizontal mode. A classical tridiagonal matrix inversion is performed for each horizontal mode (m, n), which allows us to compute the solution of (9.35) in the Fourier space and denoted $\hat{\Phi}$. Finally, we apply an inverse FFT to obtain Φ in the physical space. As it is mentionned in the FFT documentation, the numbers of horizontal modes $imax$ and $jmax$ should be a composite number factorizable as a product of powers of 2, 3 and 5. Vectorization is achieved by doing parallel transforms.

Horizontal discretization

The FFT form depends on the lateral boundary conditions:
- Cyclic case

In the aim of no confusion with the complex number i ($i^2 = -1$), we adopt for a while the convention $(i, j, k) \rightarrow (I, J, K)$. With the boundary conditions (9.32)-(9.33), we have:

$$\Phi(I,J,K) = \sum_{m=0}^{imax-1} \sum_{n=0}^{jmax-1} \widehat{\Phi}_{m\,n}(K)\ \exp\left(i\frac{2\pi}{imax}mI\ +\ i\frac{2\pi}{jmax}nJ\right) \tag{9.37}$$

We denote $\tilde{\Phi}_{mn}(K) = \widehat{\Phi}_{mn}(K)\exp\left(i\frac{2\pi}{imax}mI\ +\ i\frac{2\pi}{jmax}nJ\right)$ where $\tilde{\Phi}_{m\,n}(K)$ is complex. Using the FFT decomposition (9.37), we have:

$$\delta_{x^2}\tilde{\Phi}_{mn} = -4\sin^2\left(\frac{\pi}{imax}m\right)\tilde{\Phi}_{mn}$$
$$\delta_{y^2}\tilde{\Phi}_{mn} = -4\sin^2\left(\frac{\pi}{jmax}n\right)\tilde{\Phi}_{mn}$$

The horizontal part of the operator F can now be written as:

$$<\tilde{\rho}>_{x,y}\left[\frac{1}{<{d_{xx}}^2>_{xy}}\delta_{x^2} + \frac{1}{<{d_{yy}}^2>_{xy}}\delta_{y^2}\right]\tilde{\Phi}_{mn} = b_{mn}\tilde{\Phi}_{mn} \tag{9.38}$$

where the eigenvalues are defined by:

$$b_{mn} = -4 <\tilde{\rho}>_{x,y}\left[\frac{\sin^2\left(\frac{\pi}{imax}m\right)}{<{d_{xx}}^2>_{xy}} + \frac{\sin^2\left(\frac{\pi}{jmax}n\right)}{<{d_{yy}}^2>_{xy}}\right]$$

where $m = 0,\ 1...imax-1$ and $n = 0,\ 1, ...jmax-1$.
- Open and Davies' cases

The degeneracy of the full discrete operator with orography gives:

$$\frac{\partial\Phi}{\partial x} \rightarrow \delta_x\Phi$$
$$\frac{\partial\Phi}{\partial y} \rightarrow \delta_y\Phi$$

We obtain an homogeneous problem by changing our variable Φ at the last point to have an homogeneous condition. Then, we employ a FFT cosine decomposition expressed as:

$$\Phi(I,J,K) = \sum_{m=0}^{imax-1} \sum_{n=0}^{jmax-1} \widehat{\Phi}_{m\,n}(K)\cos\left(\frac{(2I-1)\pi}{2\,imax}m\right) \times \cos\left(\frac{(2J-1)\pi}{2\,jmax}n\right) \tag{9.39}$$

In this case, the eigenvalues of F are expressed as:

$$b_{mn} = -4 <\tilde{\rho}>_{x,y}\left[\frac{\sin^2\left(\frac{\pi}{2\,imax}m\right)}{<{d_{xx}}^2>_{xy}} + \frac{\sin^2\left(\frac{\pi}{2\,jmax}n\right)}{<{d_{yy}}^2>_{xy}}\right]$$

where $m = 0, 1...imax-1$ and $n = 0, 1, ...jmax-1$.

- Mixed boundary conditions

Here, the decomposition of Φ results from a combination of (9.37) and (9.39). If we have a cyclic condition along x and an open condition along y, we have:

$$\Phi(I,J,K) = \sum_{m=0}^{imax-1} \sum_{n=0}^{jmax-1} \widehat{\Phi}_{m\,n}(K) \left[\exp\left(i\frac{2\pi}{imax}mI\right) \times \cos\left(\frac{(2J-1)\pi}{2\,jmax}n\right)\right] \tag{9.40}$$

and the eigenvalues are defined by:

$$b_{mn} = -4 < \tilde{\rho} >_{x,y} \left[\frac{\sin^2\left(\frac{\pi}{imax}m\right)}{< {d_{xx}}^2 >_{xy}} + \frac{\sin^2\left(\frac{\pi}{2\,jmax}n\right)}{< {d_{yy}}^2 >_{xy}}\right] \tag{9.41}$$

In the other case, if we have an open condition along x and a cyclic condition along y, the decomposition of Φ reads:

$$\Phi(I,J,K) = \sum_{m=0}^{imax-1} \sum_{n=0}^{jmax-1} \widehat{\Phi}_{m\,n}(K) \cos\left(\frac{(2I-1)\pi}{2\,imax}m\right) \times \exp\left(i\frac{2\pi}{jmax}nJ\right) \tag{9.42}$$

and the eigenvalues are expressed by:

$$b_{mn} = -4 < \tilde{\rho} >_{x,y} \left[\frac{\sin^2\left(\frac{\pi}{2\,imax}m\right)}{< {d_{xx}}^2 >_{xy}} + \frac{\sin^2\left(\frac{\pi}{jmax}n\right)}{< {d_{yy}}^2 >_{xy}}\right] \tag{9.43}$$

Vertical discretization

The discretized vertical operator is applied after the horizontal operator with boundary conditions derived from the full problem with orography:

$$\left(\tilde{\rho}\overrightarrow{\nabla}\Phi\right)^{c3} \rightarrow \frac{\overline{< \tilde{\rho} >_{x,y}}^{z}}{< {d_{zz}}^2 >_{xy}} \delta_z \Phi \tag{9.44}$$

Thus, we have:

$$F\tilde{\Phi}_{mn}(k) = a(k)\tilde{\Phi}_{mn}(k-1) + b(k)\tilde{\Phi}_{mn}(k) + c(k)\tilde{\Phi}_{mn}(k+1)$$

where

$$\begin{aligned}
a(k) &= \frac{< \tilde{\rho} >_{x,y}(k-1) + < \tilde{\rho} >_{x,y}(k)}{2\,\Delta z(k)^2} \\
b(k) &= -\frac{< \tilde{\rho} >_{x,y}(k-1) + < \tilde{\rho} >_{x,y}(k)}{2\,\Delta z(k)^2} - \frac{< \tilde{\rho} >_{x,y}(k) + < \tilde{\rho} >_{x,y}(k+1)}{2\,\Delta z(k+1)^2} + b_{mn} \\
c(k) &= \frac{< \tilde{\rho} >_{x,y}(k) + < \tilde{\rho} >_{x,y}(k+1)}{2\,\Delta z(k+1)^2}
\end{aligned}$$

Complete flat operator

Thus, the flat operator is expressed under the form of mn tridiagonal matrices of $kmax + 2 \times kmax + 2$ elements with the following form:

$$\mathbf{F_{mn}} = \begin{pmatrix} b(jpvext) & c(jpvext) & & \\ a(1+jpvext) & b(1+jpvext) & c(1+jpvext) & \\ & & & 0 \\ & & & \\ 0 & & & \\ & a(kmax+jpvext) & b(kmax+jpvext) & c(kmax+jpvext) \\ & & a(kmax+1+jpvext) & b(kmax+1+jpvext) \end{pmatrix}$$

The F_{mn} components do not depend on time and are stored in arrays of:
- $imax \times jmax \times (kmax + 2)$ elements for the b coefficients;
- $kmax + 2$ elements for the a and c coefficients.

All the matrices F_{mn} are invertible except for (m, n) = (0, 0) (the uniform mode) because the vertical Neumann conditions involve an evaluation of the pressure terms to within a constant. For this reason, this particular mode is inverted apart. Moreover, as all the F_{mn} vertical matrices are independent, this computation can easily be vectorized for the $m\,n$ different modes.

Algorithm performance

Both iterative methods presented above showed a good convergence for the solution of the elliptic equation. All in all, the Richardson method requires less computation time than the CG method, because it needs less matrix inversions.

Current limitations

At the present time, the algorithm is not adapted for large stretching of the horizontal grid. A practical maximum ratio of the larger to the smaller grid size is about two, which is sufficient to allow simulations on small domains in conformal projection.

9.3 References

Bernardet, P., 1995: The pressure term in the anelastic model: A symmetric elliptic solver for an Arakawa C grid in generalized coordinates. *Mon. Wea. Rev.*, **123**, 2474-2490.

Carpenter, K. M., 1982: Note on the paper ”Radiation conditions for lateral boundaries of limited area numerical models”. *Quart. J. Roy. Meteor. Soc.*, **110**, 717-719.

Kapitza, H. and D. P. Eppel, 1992: The non-hydrostatic meso-scale model GESIMA. Part I: Dynamical equations and tests *Beitr. Phys. Atmos.*, **65**, 129-146.

Klemp, J. B., and R. B. Wilhelmson, 1978: The simulation of three-dimensional convective storm dynamics. *J. Atmos. Sci.*, **35**, 1070-1096.

Golub, G. H., and G. A. Meurant, 1983: *Résolution des Grands Systèmes Linéaires*. Eyrolles, 329 pp.

Orlanski, I., 1976: A simple boundary condition of unbounded hyperbolic flows. *J. Comp. Phys.*, **21**, 251-269.

Schumann, U., and R. Sweet, 1988: Fast Fourier transforms for direct solution of Poisson’s equation with staggered boundary conditions. *J. Comput. Phys.*, **75**, 123-137

Skamarock, W. C., P. K. Smolarkiewicz, and J. B. Klemp, 1997: Preconditioned conjugate-residual solvers for Helmholtz equations in nonhydrostatic models. *Mon. Wea. Rev.*, **125**, 587-599.

Young, D. M. and K. C. Jea, 1980: Generalized conjugate–gradient acceleration of nonsymmetrizable iterative methods, *Linear Algebra Appl.*, **34**, 159-194.

Chapter 10

Immersed Boundaries

Contents

10.1 Introduction

This chapter has been adapted from Auguste et al. (2019) and presents the implementation of an Immersed Boundary Method (IBM) in Meso-NH to represent obstacles for simulations using a cartesian grid and flat terrain. Only the fundamental equations related to the IBM are given here, examples of IBM validation and application are given in Auguste et al. (2019).
Numerical solvers in Meso-NH enforce conservation on structured grids and hence cannot handle body fitted grids with steep topological gradients. The main idea behind the IBM is the detection of an interface separating a fluid region, where conservations laws are valid, from a solid region corresponding to the obstacle volume, by using different techniques (e.g. markers, LevelSet functions, local volume fraction, etc). Two main classes of IBM exist, which are based on the continuous and discrete forcing approaches, respectively. The continuous forcing approach was developed by Peskin (1972) for biomechanics applications and consists in the addition of a continuous artificial force in the momentum conservation equation that mimics the effect of the obstacles and drives the flow to relax to a no-slip condition at the wall of the obstacle. This approach and its variant developed by Goldstein et al. (1993) for a rigid interface can suffer from the lack of stiffness since the fluid-solid interface is generally spread over few cells, and the time step restriction. For the discrete approach, the boundary conditions are specified at the immersed interface. To simulate flows around non moving and rigid bodies, two subclasses of discrete approaches can be defined as in Mittal and Iaccarino (2005): direct or indirect approaches, depending on the forcing location (Pierson, 2015). Many types of discrete forcing exist, e.g. direct forcing in the fluid region near the interface as in Mohd-Yusof (1997), the immersed interface method (Leveque and Li, 1994), or the Cartesian grid method (Clarke et al., 1986). Depending on how to resolve the partial differential

equations, Cartesian grid methods (Ye et al., 1999) are written for finite-volume discretizations (Cut-Cell Technique, CCT) and for finite-difference discretizations (Ghost-Cell Technique, GCT) as in Tseng and Ferziger (2003). The CCT reshapes the cell cut by the interface to preserve mass, momentum and energy. Using GCT, the local spatial reconstruction is done inside the solid region. The latter technique has been successfully implemented by Lundquist et al. (2010, 2012) in the Weather Research and Forecasting (WRF) model.
The IBM implemented in Meso-NH is based on the discrete forcing approach. The fluid-solid interface is modelled by means of a LevelSet function (Sussman et al., 1994). The motivation behind this choice is that rigid and non-moving bodies in a turbulent flow shall be represented, and with sufficiently fine resolution to avoid the large dissipation inherent to the presumed spread interface. The GCT does not introduce source terms in the conservation equations modelling the fluid region so that boundary conditions are imposed at the interface and/or in the solid region. The only corrections to the physical model in the fluid region come from subgrid turbulent parameterizations. The idea is that in mesoscale atmospheric modelling applications, the IBM is used to resolve large obstacles such as buildings or mountains, whereas a subgrid drag model is used to handle unresolved obstacles such as vegetation (Aumond et al., 2013).

10.2 Representation of obstacles via a LevelSet function

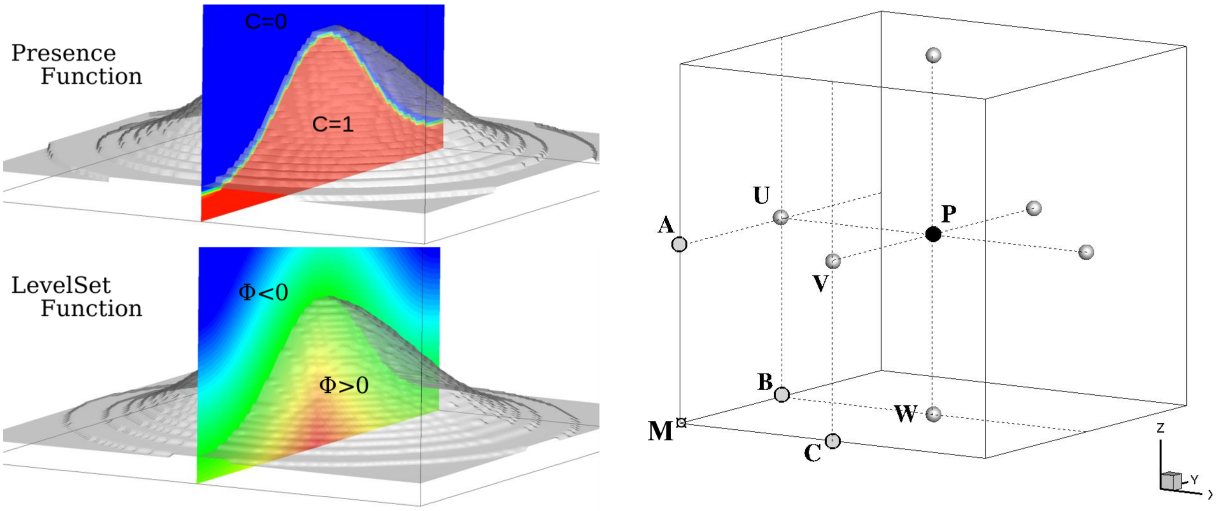
The numerical domain is divided in two regions: a fluid region where the equations of continuum mechanics hold, and a solid region embedding the obstacle where they do not. After comparing two methods to represent the obstacles (Fig. 10.1-a), one based on a local presence function, and one based on the LevelSet Function (LSF; Sussman et al., 1994; Kempe and Fröhlich, 2012), the LSF is chosen since it is able to capture the interface between the fluid and solid regions more accurately : The $\mid \phi \mid$ distance informs about the minimal distance to the fluid-solid interface and the ϕ-sign about the region nature: sgn$(\phi) > 0$ for the solid one; otherwise sgn$(\phi) < 0$. The vector $\mathbf{n}$ normal to the interface and its local curvature σ are defined such as $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$ and $\sigma = -\nabla.\mathbf{n}$. Fig. 10.1-a illustrates the continuous variation of LSF for an arbitrary bell shape interface. The LSF is estimated at the seven available point types per grid cell to limit the discretization errors (Fig. 10.1-b): at the mass point P where prognostic scalar variables are localized, at the three velocity nodes $U/V/W$, and the A/B/C vorticity nodes employed by turbulent variables. The points of the solid region acts as external points of the computational grid.

10.3 Modification of the equations using a Ghost Cut-Cell Technique

Let ψ^n be a prognostic variable of Meso-NH at the time $n\Delta t$ (Δt, the time step). The tendencies of the prognostic variables $\overline{\psi} = [\overline{\mathbf{u}}, \overline{\theta}, \overline{s}, (e)]$ can not be deduced from the conservation laws in the solid region. Expecting a correction due to IBM where $\phi \geq 0$, a general formulation of the tendencies is written as:

$$\frac{\partial}{\partial t} = \left.\frac{\partial}{\partial t}\right|_{csl} + \left.\frac{\partial}{\partial t}\right|_{ibm} \tag{10.1}$$

The RHS first term of Equation (10.1) is given by the conservation laws. The $\left.\frac{\partial}{\partial t}\right|_{ibm}$ tendency is the correction due to the GCT in the solid region and near the immersed interface satisfying the $\overline{\psi}$

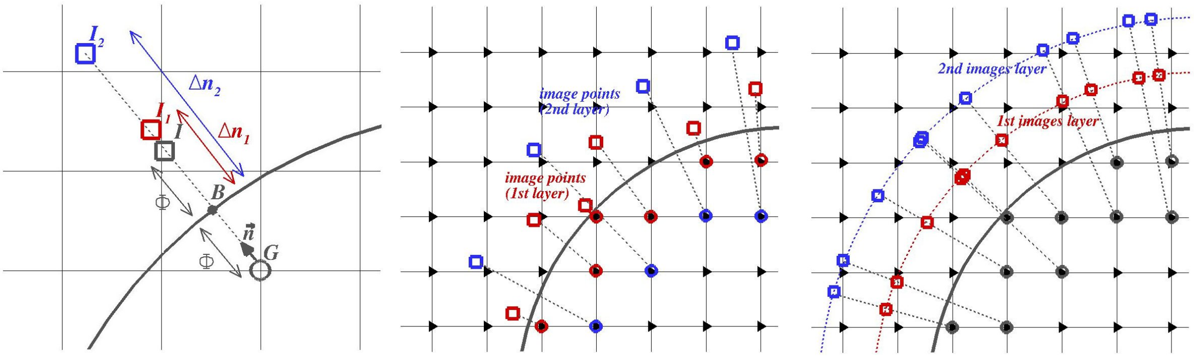


(a) (b)

Figure 10.1: *(a) Illustration of two ways to model a fluid-solid interface: the color code indicates the isocontours of the presence function C and the LevelSet function ϕ; (b) Definition of the point types per grid cell: M the geometric/mesh point, P the mass point, U/V/W the velocity nodes and A/B/C the vorticity nodes.*

desired boundary conditions at $\phi = 0$:

$$\left.\frac{\partial \overline{\psi}}{\partial t}\right|_{ibm} = -\left.\frac{\partial \overline{\psi}}{\partial t}\right|_{csl} + \frac{\overline{\psi}^{n+1} - \overline{\psi}^{n}}{\Delta t} \tag{10.2}$$



(a) (b) (c)

Figure 10.2: *(a) Node definitions acting in the Ghost-Cell Technique : the ghost (G), interface (B), normal vector ($\mathbf{n}$), mirror (I) and images (I_1, I_2). (b resp. c) Illustration of the classical (resp. new) GCT using the mirror (resp. images). Triangles correspond to one of the node types (see Fig. 10.1-b).*

The forced points are called ghost points and merely renamed ghosts. To estimate the variable $\overline{\psi}$ and for each ghost, the physical information is extracted near the interface and from the fluid region. The extension (grid stencil) of the forcing zone depends on the spatial accuracy of the numerical scheme. For example, Figure 10.2b-c show the case of a two-layers stencil in a two-dimensional grid. The characterization of the layer is done by a conditional loop applied direction by direction on the LSF. For a 2D case, the sign of $\phi(i,j).\phi(i,[j-k_l : j+k_l])$ and $\phi(i,j).\phi([i-k_l : i+k_l],j)$ is estimated. The integer value k_l determines the cells truncated by the interface: $k_l = 1$ (resp. $k_l = 2$) defines the first (resp. second) layer. The calculation of these ghost layers has a computational overhead due to data exchange among processors in parallel simulations. The stencil of the numerical schemes modelling the interface defines the k_l value. In order to limit this overhead, a low-order version of centered explicit-in-time schemes is employed when $\phi > -\Delta$. The CPU cost of the 'hybrid' advection scheme is largely compensated by the decrease of the ghosts number and parallel exchanges.

In the classical GCT (Tseng and Ferziger, 2003), the fluid information is obtained at a mirror point (noted I, merely renamed mirror) found in the normal direction to the interface in such a way that the interface node B is equidistant to I and G. Figure 10.2-a shows the characterization of one ghost G (of LSF value ϕ_G), its associated mirror I (of LSF value ϕ_I) and the interface node B ($\mathbf{GI} = 2\phi_G\mathbf{n}$). Figure 10.2-b illustrates several ghosts and mirrors. The $|\mathbf{IB}|$ distance depends on the forcing stencil and a problematic case regularly met in the mirror interpolation is the vicinity of ghosts with the interface ($\phi_G = -\phi_I << \Delta$, with Δ the space step) leading to a not well-posed condition.

The new GCT. To overcome this problem, we define image points (noted I_1 and I_2 in Fig. 10.2-a, merely renamed images) having a distance to the interface that depends only on the grid spacing: $\mathbf{GI_l} = l\Delta + \phi_G\mathbf{n}$ with $l = (1;2)$. Figure 10.2-a shows the images for one ghost. The new approach enforces a large enough value of the $|\mathbf{I_lB}|$ distance. Figure 10.2-b (resp. c) illustrates the classical (resp. new) GCT for several ghosts. Figure 10.2-b shows some mirror points associated with ghosts of the first layer in the vicinity of the interface. Figure 10.2-c shows that the new approach ensures that the image points are always located in the fluid region, irrespective of the ghost location. The definition of several images per ghost allows to build a profile of the $\overline{\psi}$ fluid information normal to the interface. $\overline{\psi}(I)$ is therefore recovered by a quadratic reconstruction f using the (B, I_1, I_2) points. Two distinct calculations of $f(\overline{\psi}(B), \overline{\psi}(I_1), \overline{\psi}(I_2))$ noted PLI^a and PLI^b are tested to build the Lagrange interpolation:

$$PLI^a(I) = [2L^a_G(I)\overline{\psi}(B) + L^a_{I_1}(I)\overline{\psi}(I_1) + L^a_{I_2}(I)\overline{\psi}(I_2)][1 + L^a_G(I)]^{-1} \tag{10.3}$$

$$PLI^b(I) = L^b_B(I)\overline{\psi}(B) + L^b_{I_1}(I)\overline{\psi}(I_1) + L^b_{I_2}(I)\overline{\psi}(I_2) \tag{10.4}$$

where $L^a(I)$ and $L^b(I)$ are the Lagrange polynomials:

$$L^a_{I_1}(I) = \left(\frac{2\Delta - \phi}{\Delta}\right)\left(\frac{2\phi}{\Delta + \phi}\right) \tag{10.5}$$

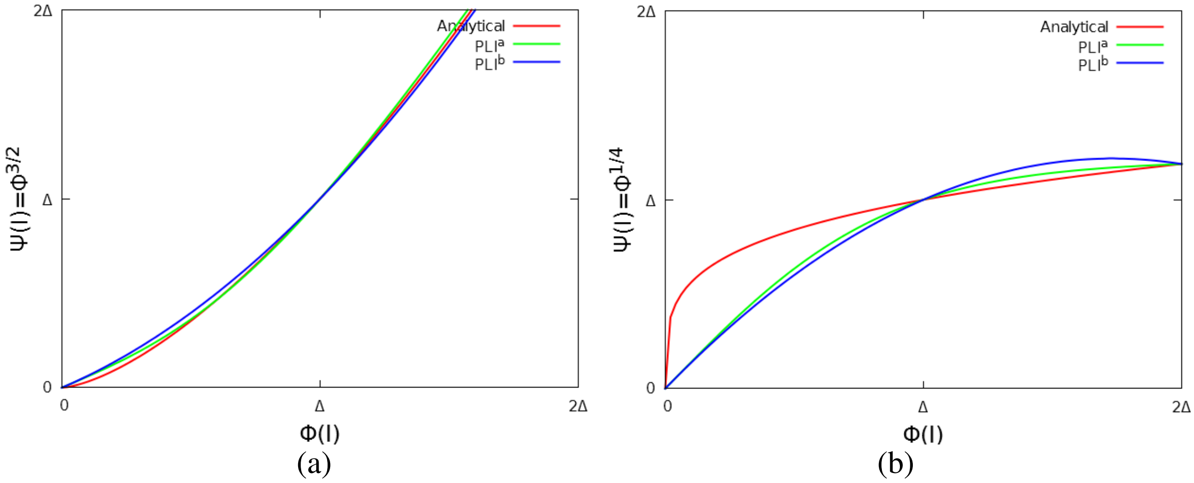
$$L^a_{I_2}(I) = \left(\frac{\phi - \Delta}{\Delta}\right)\left(\frac{2\phi}{2\Delta + \phi}\right) \tag{10.6}$$

$$L^a_G(I) = \left(\frac{\phi - \Delta}{\phi + \Delta}\right)\left(\frac{\phi - 2\Delta}{\phi + 2\Delta}\right) \tag{10.7}$$

$$L^b_{I_1}(I) = \left(\frac{2\Delta - \phi}{\Delta}\right)\left(\frac{\phi}{\Delta}\right) \tag{10.8}$$

$$L^b_{I_2}(I) = \left(\frac{\phi - \Delta}{\Delta}\right)\left(\frac{\phi}{2\Delta}\right) \tag{10.9}$$

$$L_B^b(I) = \left(\frac{\phi}{\Delta}\right)\left(\frac{\phi-2\Delta}{2\Delta}\right) \tag{10.10}$$



(a) (b)

Figure 10.3: *Quadratic interpolations of two analytical profiles* $\overline{\psi} = \phi^n$ *(red lines) using two image points at* $\phi = -[1;2]\Delta$ *and the interface node. Green (resp. blue) color corresponds to* L^a *(resp.* L^b*) polynomial results.*

The accuracy of an interpolation depends on the $\overline{\psi}$-profile. For example, a logarithmic evolution of the tangent velocity is expected in LES. Otherwise when the viscous layer is modelled, a linear evolution is expected. To compare the ability of each quadratic interpolation to approach a wide variety of profiles, the recover of power laws such as $\psi = \phi^{3/2}$ (Figure 10.3-a) and $\psi = \phi^{1/4}$ (Fig. 10.3-b) is studied. As it illustrates, PLI^a fits better the two analytical solutions and is therefore adopted. The GCT implementation is divided in four main steps: the fluid information recovery, the interface basis change, the interface condition and the ghost value.

The fluid information recovery. The values of $\overline{\psi}_{I_l}$ for the images located in a pure fluid cell (all corner nodes are in the fluid region) is recovered by a trilinear interpolation based on Lagrange Polynomials (LP):

$$L_i^{LP}(x_l) = \prod_{p=1,p\neq i}^{N} \frac{x_l - x_p}{x_i - x_p} \;\; ; \;\; L_j^{LP}(y_l) = \prod_{p=1,p\neq j}^{N} \frac{y_l - y_p}{y_j - y_p} \;\; ; \;\; L_k^{LP}(z_l) = \prod_{p=1,p\neq k}^{N} \frac{z_l - z_p}{z_k - z_p} \tag{10.11}$$

$$\overline{\psi}(x_l, y_l, z_l) = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} L_i^{LP}(x_l) L_j^{LP}(y_l) L_k^{LP}(z_l).\overline{\psi}(x_i, y_j, z_k) \tag{10.12}$$

For truncated cells with at least one corner node in the solid region, $\overline{\psi}_{I_l}$ is recovered using an inverse Distance Weighting (DW) interpolation:

$$\overline{\psi}(x_l) = \frac{\sum_{i=1}^{N} L_i^{DW}(x_l).\overline{\psi}(x_i)}{\sum_{i=1}^{N} L_i^{DW}(x_l)} \;\; ; \;\; \mid \mathbf{x_l} - \mathbf{x_i} \mid = \sqrt{(x_l - x_i)^2 + (y_l - y_i)^2 + (z_l - z_i)^2} \tag{10.13}$$

where $L_i^{DW}(x) =\mid x_l - x_i \mid^{-\alpha}$ ($\alpha = 1$). This formulation diverges when $x_i \to x_l$ and it is commonly adopted to impose $\overline{\psi}(x_l) = \overline{\psi}(x_i)$ when $\exists(x_i - x_l) \leq \epsilon$ (ϵ is an arbitrary parameter depending on the mesh discretization, $\epsilon << \Delta$). The 3D extension is direct with $\mid \mathbf{x_l} - \mathbf{x_i} \mid = \sqrt{(x_l - x_i)^2 + (y_l - y_i)^2 + (z_l - z_i)^2}$. This interpolation method has been chosen after comparison with Barycentric Lagrange and Modified Distance Weighting interpolations (Franke, 1982) and tests on the α coefficient. As the boundary condition is expressed in the interface frame and the grid is staggered, the non-collocation of the $\overline{\mathbf{u}}$ components implies firstly to interpolate three different classes of cells (with $U/V/W$ corners, Fig. 10.1-b) for each $U/V/W$ ghosts, secondly to build the change of frame matrix for which the proposed GCT presents an interest during the characterization of the direction tangent to the interface.

The interface basis change. The velocity vector $\overline{\mathbf{u}}$ defined in the Cartesian mesh basis at the images I_l ($\Delta n_1 = \Delta$ and $\Delta n_2 = 2\Delta$ in Fig. 10.2-a) is projected in the basis of the interface $(\mathbf{n}(B), \mathbf{t}(B), \mathbf{c}(B))$ in which the boundary conditions on each vector component are imposed. The normal direction to the obstacle is defined by Computing the LSF gradient. Otherwise, $(\mathbf{t}, \mathbf{c})$ are two arbitrary tangent directions. The tangent direction $\mathbf{t}$ is considered as the predominant tangent direction of the flow along the fluid-solid interface depending on the image values and defining the velocity vector such as $\overline{\mathbf{u}}(I_l) = \overline{u}_n(I_l)\mathbf{n} + \overline{u}_t(I_l)\mathbf{t}(I_l)$. The cotangent direction is defined such as:

$$\mathbf{c}(I_l) = \frac{\mathbf{n} \otimes \overline{\mathbf{u}}(I_l)}{\mid\mid \mathbf{n} \otimes \overline{\mathbf{u}}(I_l) \mid\mid} \quad ; \quad \mathbf{t}(I_l) = \mathbf{c}(I_l) \otimes \mathbf{n} \tag{10.14}$$

The $(\mathbf{n}, \mathbf{t}, \mathbf{c})$ basis at the interface is defined by considering or not the rotation of the tangent velocity with the distance to the interface:

$$\mathbf{t}(B) = \mathbf{t}(I_1) \text{ if no rotation} \quad ; \quad \mathbf{e_t}(B) = 2\mathbf{e_t}(I_1) - \mathbf{e_t}(I_2) \text{ if linear evolution} \tag{10.15}$$

Finally, the third component is $\mathbf{c}(B) = \mathbf{n} \otimes \mathbf{e_t}(B)$ and (inverse) projection is known.

The interface condition. Let $\overline{\psi}_B$ and $\Delta\frac{\partial\overline{\psi}}{\partial n}\big|_B$ the Dirichlet and Neumann conditions on $\overline{\psi}$ at boundary B. The general formulation of the boundary condition $\overline{\psi}(\phi = 0)$ is written as a Robin condition: $\overline{\psi}(\phi = 0) = k_r\overline{\psi}_B + (1 - k_r).(\overline{\psi}(\phi = -l\Delta) - l\Delta\frac{\partial\overline{\psi}}{\partial n}\big|_B)$. The switch between the Dirichlet condition and the Neumann condition is done through the coefficient $k_r \in [0 : 1]$. A Dirichlet condition, $(k_r; \overline{\psi}_B) = (1; 0)$ is imposed on the $\overline{\mathbf{u}}.\mathbf{n}$ velocity component normal to the interface arising from the impermeability hypothesis, and on the $\overline{\mathbf{u}}.\mathbf{t}$ component tangent to the interface for a no-slip hypothesis. A Neumann condition $((k_r; \frac{\partial\overline{\psi}}{\partial n}\big|_B) = (0; 0))$ is imposed for potential temperature, passive scalars, and subgrid kinetic energy to represent a no surface flux boundary condition, and to represent a free-slip case for $\overline{\mathbf{u}}.\mathbf{t}$. The $\frac{l\phi}{2}$-approximation in the location of the derivative term and the Neumann condition depending on the chosen image (in practice the selected image I_l is the closest one to the interface).
An interface condition depending on the characteristics of the surrounding fluid such as $\overline{\psi}(\phi = 0) = F(\overline{\psi}_{I_l}; \frac{\partial\overline{\psi}}{\partial n}\big|_{I_l})$ is a wall model. Using two (resp. three) images, simple wall models such as the constant (resp. linear) extrapolation of the $\overline{\psi}$ gradient is reached by the $\frac{\partial^2\overline{\psi}}{\partial n^2}\big|_{I_l} = 0$ (resp. $\frac{\partial^3\overline{\psi}}{\partial n^3}\big|_{I_l} = 0$) computation. The consistency between the tangent component to the interface of the resolved wind and the subgrid turbulence is the subject of Section 10.5.

The ghost value. Knowing $\overline{\psi}(\phi = 0)$ and $\overline{\psi}_{I_l} = \overline{\psi}(\phi = -l\Delta)$, $\psi(G)$ for a Dirichlet (resp. Neumann) condition is written such as :

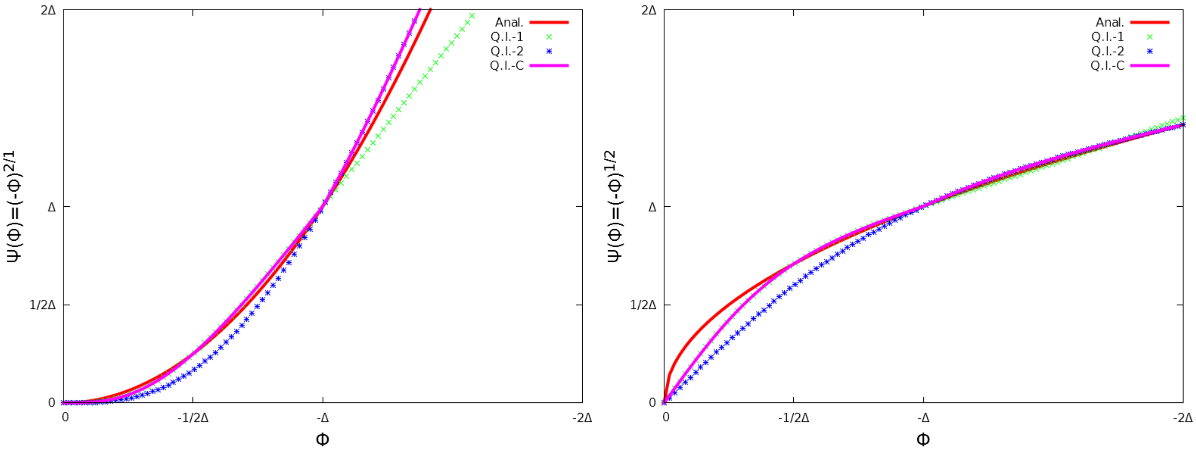


Figure 10.4: *Profile normal to the interface of two fluid informations* $\overline{\psi}$*: analytical solution (red lines), quadratic interpolation* QI_1 *using* $\overline{\psi}_{-\phi=[1/2;1]\Delta}$ *(green symbols),* QI_2 *using* $\overline{\psi}_{-\phi=[1;2]\Delta}$ *(blue symbols),* QI_C *as a combination of* QI_1 *and* QI_2 *(purple lines).*

$$\psi(G) = 2\psi(B) - \psi(I) \;\; (Dirichlet) \quad ; \quad \psi(G) = 2\phi \frac{d\psi}{dn}\bigg|_B + \psi(I) \;\; (Neuman) \tag{10.16}$$

In practice, three I_l images are defined for which the location are $\phi_{I_l} = -l\Delta$ with $l = [1/2;1;2]$. The choice of the images distance to the interface affects the results. To approach at best the expected solution, two quadratic interpolations depending on the used images and one combination of these quadratic interpolations are tested. Figure 10.4-a and Figure 10.4-b illustrate these interpolations by considering two analytical profiles (red lines): the quadratic interpolation QI_1 (resp. QI_2) is based on the images values located at $\phi = 1/2\Delta$ and $\phi = \Delta$ and plotted in green symbols (resp. at $\phi = \Delta$ and $\phi = 2\Delta$ plotted in blue symbols). Depending on the analytical profile, Figure 10.4 shows the influence of the image location choice. As expected, QI_1 (resp. QI_2) is less accurate than QI_2 (resp. QI_1) for $\overline{\psi}(\phi \in [-2\Delta : -\Delta])$ (resp. for $\overline{\psi}(\phi \in [-\Delta : 0])$). QI_C is the combination of QI_1 and QI_2 (purple line). QI_C preserves the advantage of each quadratic interpolation and when $\phi_G < \Delta$ (resp. $\phi_G > \Delta$), QI_1 (resp. QI_2) is used in the rest of the study. Knowing $\overline{\psi}^{n+1}(\phi^-)$ at the end of the Meso-NH temporal loop with QI_C, the $\overline{\psi}^{n+1}(\phi^+)$ profile is extrapolated from the fluid region to the solid region by applying an anti-symmetry $\overline{\psi}^{n+1}(\phi^+) = 2\overline{\psi}^{n+1}(0) - \overline{\psi}^{n+1}(\phi^-)$. The ghost value is estimated and the $\overline{\psi}$-gradient at the interface is also recovered.

10.4 Cut-Cell Technique and pressure solver

First looking at the RHS of the elliptic pressure equation, the $\frac{\partial(\rho_r \overline{\mathbf{u}}^*)}{\partial t}\Big|_{csl}$ coming from the resolution of the explicit-in-time schemes near the interface and in the solid regions badly affects the $\nabla.\overline{\mathbf{u}}^*$ computation. Therefore the fictive wind of the solid region can spread errors in the fluid region during the pressure resolution. A correction of the pressure solver when using the IBM is therefore required.

The elliptic problem is re-written as a resolution of the linear system $\mathcal{P}.\Psi^* = \mathcal{Q}$. In the standard Meso-NH version, $\nabla.\overline{\mathbf{u}}^* = \mathcal{Q}$ is estimated using a finite difference approach. To uncouple the

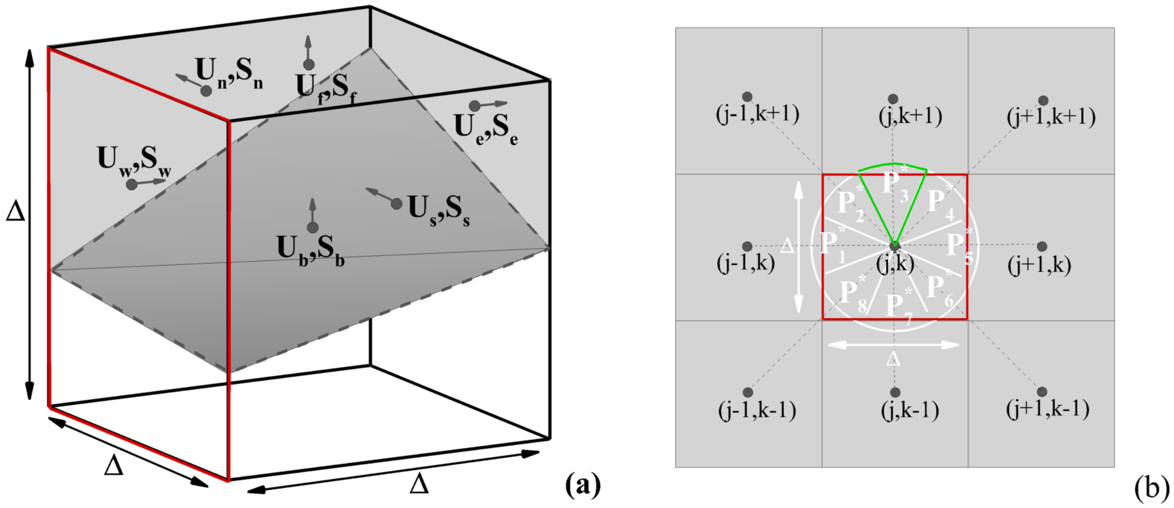


Figure 10.5: *(a) Momentum fluxes balance for an arbitrary truncated cell of volume $\mathcal{V}$ where the $\overline{u_i}^*$ velocities (U_i in the figure, $i = [e, w, n, s, b, f]$) are supported by the $\mathcal{S}_i$ surfaces in grey colour; the transparent volume is a part of the solid body. (b) Segmentation of the $\mathcal{S}_w$ arbitrary surface (red border) in eigth P^*_p pieces of cake (the border of P^*_3 is indicated in green).*

solid region from the fluid region, a null-divergence for pure solid cells is enforced and the balance of momentum fluxes is estimated by a finite volume approach for truncated cells (noted $\mathcal{Q}_{cct}$):

$$\mathcal{V}\nabla.\overline{\mathbf{u}}^* = \int_{\mathcal{V}_f} \nabla.\overline{\mathbf{u}}^* d\mathcal{V} + \int_{\mathcal{V}_s} \nabla\overline{\mathbf{u}}^* d\mathcal{V} = \sum \pm\overline{u_i}^*.\mathcal{S}_i = \sum \pm\widetilde{\Delta^2\overline{u_i}^*} \tag{10.17}$$

where $\mathcal{V} = \Delta^3$ is the cell volume, $\mathcal{V}_f$ (resp. $\mathcal{V}_s$) the fluid (resp. solid) part of $\mathcal{V}$, $\mathcal{S}_i$ the cell surfaces where i is the index of each surface orientation [e,w,n,s,b,f] as it illustrates in Figure 10.5-a.
According to the Green-Ostrogradski theorem, the $\overline{u_i}^*\mathcal{S}_i$ calculation is the classical way of a CCT Cut-Cell Technique (Yang et al., 1997; Kim et al., 2001) to estimate the velocity divergence. A similar approach is here performed re-building the flux $\widetilde{\Delta^2\overline{\mathbf{u}}^*}$ for truncated and solid cells.
The $\pm\widetilde{\Delta^2\overline{u_i}^*}$ calculation consists in a weighting of the out-fluxes and in-fluxes function of the fluid and cell surfaces ratio (Fig. 10.5-a). Figure 10.5-b gives an example of the west surface ($i = w$, red border) where $\widetilde{\Delta^2\overline{u_w}^*}(j,k)$ is calculated using the LSF value $\phi = \phi_w$ and the ones of the eight adjacent nodes $\phi_p(j \pm 1, k \pm 1)$. A disk of radius $\sqrt[-1]{\pi}\Delta$ is split in eight 'piece of cake' segments P^*_p ($p = [1 : 8]$). A LSF linear interpolation detects or not the interface location. In presence of an interface, its distance from the studied node is $0 < \delta_p < \sqrt[-1]{\pi}\Delta$. Knowing δ_p, the momentum fluxes balance is formulated for a non-moving body as p is the index of the 'piece of cake' and i the index of the cell surface:

$$\widetilde{\Delta^2\overline{u_i}^*} = \frac{\Delta^2}{8}[\sum_{p=1}^{8}\mathcal{H}(-\phi_p)\mathcal{H}(-\phi_i)\overline{\mathbf{u_i}}^* + \sum_{p=1}^{8}\mathcal{H}(-\phi_p\phi_i).|\mathcal{H}(-\phi_p) - \pi(\frac{\delta_p}{\Delta})^2|.(\mathcal{H}(-\phi_p)\overline{\mathbf{u}}^*_p + \mathcal{H}(-\phi_i)\overline{\mathbf{u_i}}^*)] \tag{10.18}$$

The four encountered cases correspond to a pure fluid cell $\widetilde{\Delta^2\overline{u_i}^*} = \frac{\Delta^2}{8}\sum_{p=1}^{8}\overline{\mathbf{u_i}}^*$ when $\phi_p < 0$

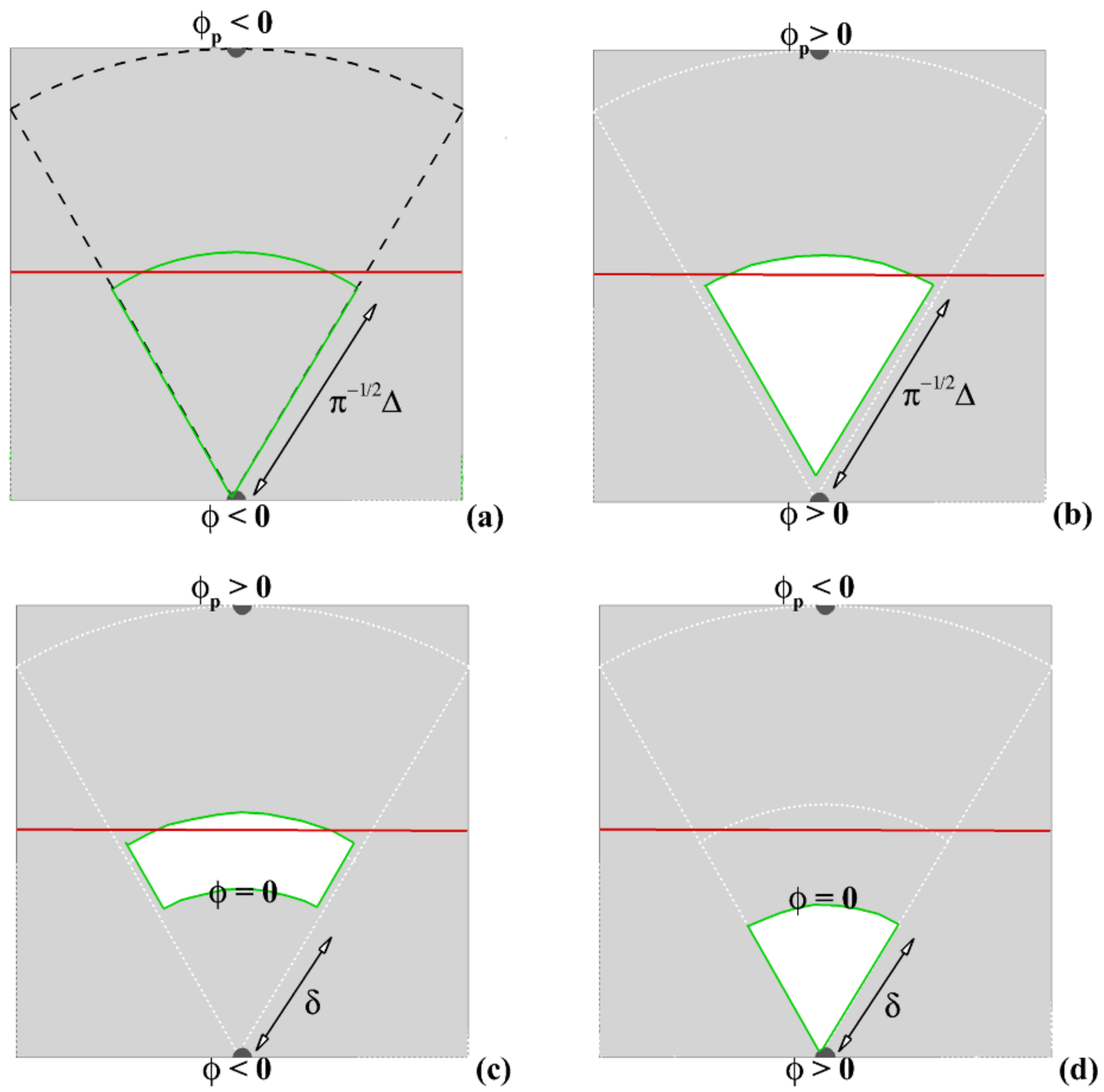


Figure 10.6: *(a,b,c,d)* $\pm\widetilde{\Delta^2\overline{u_i}^*}$ *calculations depending on the signs of* $\phi_i = (\phi)$ *and* ϕ_p *on an arbitrary piece of cake. The white (resp. grey) region corresponds to the solid (resp. fluid) one of* P_p^* *(same colour code as in Fig. 10.5).*

and $\phi_i < 0$ (Fig. 10.6-a); a pure solid cell $\widetilde{\Delta^2\overline{u_i}^*} = 0$ when $\phi_p > 0$ and $\phi_i > 0$ (Fig. 10.6-b); two types of truncated cells depending on the fluid/solid nature of the main node for which $\phi_p.\phi_i < 0$ (Fig. 10.6-c/d). Using Equation (10.18), Equation (10.17) is solved and leads to the RHS computation of the elliptic pressure equation.

Knowing $\mathcal{Q}_{cct}$, the reflection concerns now the $\mathcal{P}$ matrix to invert. The classical interface condition on the potential Ψ^* is a homogeneous Neumann condition $\frac{\partial\Psi^*}{\partial\phi} = 0$. Using a Boundary Fitted Method (BFM), the interface condition of the moving or non-moving body (Auguste, 2010) appears only on the border of a numerical domain. Using an IBM and without any impact of this interface condition on the $\mathcal{P}$-coefficients, the impermeability character of solid obstacles is not achieved. Due to the inversion of the horizontal part of $\mathcal{P}$ by a Fast Fourier Transform (Schumann and Sweet, 1988), the solution of calculating $\mathcal{P}_{cct}$ appears to be problematic. The adopted solution consists in an iterative procedure as used in Meso-NH for non-flat problems. The non-respect of the Ψ^*-condition in $\mathcal{P}$ leads to a not well-posed system and the iterative procedure goes to spread to the entire fluid domain the enforcement of the null-divergence imposed on solid cells. The solution of the pseudo-Poisson equation brings to $\Psi^* \rightarrow \Psi^{*M} = \sum\limits_{m=1}^{M} \mathcal{P}^{-1}.\mathcal{Q}^m_{cct}$ where M is the number of iterations. This number is limited by a convergence criterion, which is a compromise between incompressibility satisfaction and CPU cost.

10.5 Representation of subgrid turbulent exchange near the obstacles

It is known that the $\frac{l_m}{l_\epsilon} \rightarrow 1$ is a reasonable approximation in non-homogeneous, non-isotropic turbulence such as in the near-wall region. This approximation is indeed retained in the present IBM implementation, which assumes $l_m = l_\epsilon$ (hereafter noted l_m and called the mixing length). l_m is equal to the numerical cut-off space scale sufficiently far from the ground bringing to a $\Delta\sqrt{e}$ turbulent viscosity. Near the ground and following the Prandtl idea consisting in the assumption of the linear variation of l_m in the near-wall region, the upper limit of the mixing length is $min(kz, \Delta)$ (k is the von Kármán constant and z is the altitude).

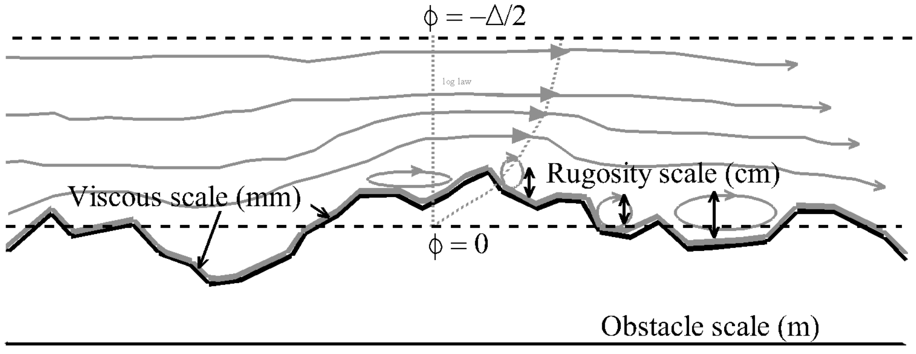


Figure 10.7: *Illustration of the unresolved physical processes near a non-idealized solid wall (black line) in an atmospheric context: the length scale based on the viscous effects (grey line) is drastically smaller than the roughness length. The roughness length approaches the scale of smallest eddies and governs the log-law profile.*

The turbulent characteristics are highly affected by a surface interaction. As a consequence and for LES, the subgrid turbulence scheme is modified in presence of immersed obstacles on the subgrid turbulent kinetic energy equation, mixing length computation and Reynolds stresses diagnosis.

The Subgrid Turbulent Kinetic Energy (e) condition. The explicit-in-time resolution of the prognostic equation for subgrid turbulent kinetic energy claims a GCT forcing and an interface condition on e. Commonly, the e profile is considered parabolical in the viscous sublayer (Bredberg, 2000; Craft et al., 2002) and constant in the inertial and wake/outer layers (Kalitzin et al., 2005; Capizzano, 2011). Due to the high turbulent Reynolds number $Re_t \approx \mathcal{O}(10^4 - 10^5)$

encountered, a homogeneous Neumann condition is applied at the immersed interface. The equilibrium between production and dissipation of e could be discussed and controverted; this choice acts as a first stage in the IBM development.

The near-wall correction of the mixing length. The von Kármán limitation due to immersed walls acts through the LSF and the upper limit on the mixing length l_m near the interface becomes $min(kz, -\phi, \Delta)$ with a banning of negative values in the solid region. Whatever the production of Subgrid Turbulent Kinetic Energy e, and the turbulent shear, the lower limit $l_m(-\phi \leq 0)$ induces a null value of the diagnosed surface fluxes. In addition, a singularity appears in the dissipative term $\rho_r K_\epsilon e^{3/2} l_m^{-1}$. By a pragmatic reasoning, the singularity due to $l_m^{-1}(\phi \rightarrow 0^-) \rightarrow \infty$ amounts to say that modelled length scales are smaller than the Kolmogorov scale $(\nu^3 \epsilon^{-1})^{\frac{1}{4}}$. Considering the Kolmogorov scale as modeled, the turbulence should vanish which is in contradiction with the dissipative term. In order to overcome this ill-posed problem, a lower limit has to be specified for l_m. In the study of atmospheric flows around buildings, a characteristic thickness of the viscous layer $H/\sqrt{Re}$ can be defined around a H bluff body for a Reynolds number based on the obstacle scale: $H \approx \mathcal{O}(10m); Re \approx \mathcal{O}(10^7)$. This thickness estimate is also proportional to $E\nu/u^*$ ($E \approx 9.8$ is commonly employed) where the friction velocity u^* is about the centimeter per second. Following these estimates, the length scale due to the viscous effects z_ν^{ib} belongs to the millimetres domain in the expected atmospheric cases. Looking after a building surface and its large heterogeneity (door, windows, surface characteristics), its roughness length z_0^{ib} is at least in the decimeter domain and $z_0^{ib} > z_\nu^{ib}$ (Illustration in Fig. 10.7). For low Re and smooth surfaces, $z_\nu^{ib} > z_0^{ib}$ could be encountered. Therefore, we assume $z^{ib} = max(z_0^{ib}, z_\nu^{ib})$ and that z^{ib} is related to the size of smallest unresolved eddies near walls (i.e. dissipative scale). The mixing length near wall is $z^{ib} < l_m < min(kz, -\phi, \Delta)$.

The turbulent fluxes correction. The $\overline{\psi}$-gradient and the turbulent diffusion $\mathcal{O}(z^{ib}\sqrt{e})$ prescribe the turbulent fluxes at the immersed interfaces. As a first step in the Meso-NH-IBM implementation, no-flux condition on the mean potential temperature is imposed such that the sensible heat flux is zero. Writing the mean velocity field at B such as $\overline{\mathbf{u}} = \overline{u_t}\mathbf{t}$, $\overline{u_t}(B)$ is needed to recover a gradient consistent with the turbulent shear. Considering the Prandtl (1925) or Kármán (1930) theories, the logarithmic profile is assumed in the vicinity of the wall according to $\overline{u}_t(z) = \frac{u^*}{k} ln(1 + \frac{z}{z^{ib}})$. Considering Δ as the limit of the resolved scales, most of the turbulent kinetic energy $\frac{1}{2}(\overline{u'^2} + \overline{v'^2} + \overline{w'^2})$ is contained in the subgrid one when $-\phi < \Delta$ and such as $K_{tke}\sqrt{e}$ with a constant $K_{tke} \gtrsim 1$. This assumption is reinforced by the homogeneous Neumann condition applied on e. This approach derives from the RANS (Reynolds-Averaged Navier Stokes) approaches and the velocity friction is formulated as $u^* = K_{tke}\sqrt[4]{C_\mu}\sqrt{e}$ where C_μ is a constant evolving between 0.03 (atmospheric applications) and 0.09 (fluid mechanics applications). Adding a damping function for the viscous cases (low turbulent Reynolds number, $Re_t < 20$), the tangent wall velocity at the interface is written such as:

$$\overline{u}_t(B) = \frac{K_{tke}\sqrt[4]{C_\mu}\sqrt{e(\phi = -\Delta/2)}}{k} ln\left(1 + \frac{\Delta}{z^{ib}}\left[1 - e^{-20 z^{ib} \Delta^{-1}}\right]\right) \tag{10.19}$$

Finally the pragmatic limitation $\overline{u}_t(B) \leq \overline{u_t}(\phi = -\Delta/2)$ operates if the e value is too high. The proposed dynamic wall-model evolves between the no-slip and free-slip conditions. If the subgrid turbulence is weak or if the physical problem is fully resolved, the viscous layer is well-modelled and $\overline{u_t}(B) \rightarrow 0$. Otherwise for an intense subgrid turbulence or a fully unresolved problem, the shear due to the wall presence is not perceived and $\frac{d\overline{u_t}}{dn}\big|_B \rightarrow 0$. The wall-model establishes an equilibrium between the production of e and the mean parietal friction. Note that the use of a log-law model near a singularity such as sharp edges and corners could be called into question. After numerical investigations for a single cube, $K_{tke}\sqrt[4]{C_\mu} \approx 1$ appears as a suitable choice.

10.6 References

Auguste, F., 2010: Instabilités de sillage et de trajectoire dans un fluide visqueux. *PhD thesis, University of Toulouse*, 313 p.

Auguste, F., G. Réa, R. Paoli, C. Lac, V. Masson, and D. Cariolle, 2019: Implementation of an immersed boundary method in the Meso-NH v5.2 model: applications to an idealized urban environment. *Geoscientific Model Development*, **12(6)**, 2607-2633.

Aumond, P., V. Masson, C. Lac, B. Gauvreau, S. Dupont, and M. Berengier, 2013: Including the drag effects of canopies: real case large-eddy simulation studies. *Bound.-lay. Met.*, **146**, 65-80.

Bredberg, J., 2000: On the wall boundary condition for turbulence models. *Chalmers University of Technology, Department of Thermo and Fluid Dynamics. Goteborg.*, **Internal Report 00/4**, 21 p.

Capizzano, F., 2011: Turbulent wall model for immersed boundary methods. *AIAA journal*, **49(11)**, 2367-2381.

Clarke, D. K., M. D. Salas, and H. A. Hassan, 1986: Euler calculations for multielement airfoils using Cartesian grids. *AIAA journal*, **24(3)**, 353-358.

Craft, T., S. Gant, A. Gerasimov, H. Iacovides, and B. Launder, 2002: Wall Functions Strategies for Use in Turbulent Flow CFD. *J. Heat transfer*, **1**, 3-14.

Franke, R., 1982: Scattered data interpolation, tests of some methods. *Math. Comput.*, **38(157)**, 181-200.

Goldstein, D., R. Handler, and L. Sirovich, 1993: Modeling a no-slip flow boundary with an external force field. *J. Comput. Phys.*, **105(2)**, 354-366.

Kalitzin, G., G. Medic, G. Iaccarino, and P. Durbin, 2005: Near-wall behavior of RANS turbulence models and implications for wall functions. *J. Comput. Phys.*, **204(1)**, 265-291.

Kármán, T. von, 1930: Mechanische Änlichkeit und Turbulenz. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 58-76.

Kempe, T., and J. Fröhlich, 2012: An improved immersed boundary method with direct forcing for the simulation of particle laden flows. *J. Comput. Phys.*, **231(9)**, 3663-3684.

Kim, J., D. Kim, and H. Choi, 2001: An immersed-boundary finite-volume method for simulations of flow in complex geometries. *J. Comput. Phys.*, **171(1)**, 132-150.

Leveque, R. J., and Z. Li, 1994: The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, **31(4)**, 1019-1044.

Lundquist, K. A., F. K. Chow, and J. K. Lundquist, 2010: An immersed boundary method for the Weather Research and Forecasting model. *Mon. Wea. Rev.*, **138(3)**, 796-817.

Lundquist, K. A., F. K. Chow, and J. K. Lundquist, 2012: An immersed boundary method enabling large-eddy simulations of flow over complex terrain in the WRF model. *Mon. Wea. Rev.*, **140(12)**, 3936-3955.

Mittal, R., and G. Iaccarino, 2005: Immersed Boundary Methods. *Annu. Rev. Fluid Mech.*, **37**, 239-261.

Mohd-Yusof, J., 1997: Combined immersed-boundary/B-spline method for simulations of flow in complex geometries. *Center for Turbulence Research Annual Research Briefs*, 317-327.

Peskin, C. S., 1972: Flow patterns around heart valves: a numerical method. *J. Comput. Phys.*, **10(2)**, 252-271.

Pierson, J.-L., 2015: Traversée d'une interface entre deux fluides par une sphère. *PhD thesis, University of Toulouse*, 226 p.

Prandtl, L., 1925: Bericht über Untersuchungen zur ausgebildeten Turbulenz. *Z. Angew. Math., Meth.*, **5**, 136-139.

Schumann, U., and R. Sweet, 1988: Fast Fourier Transforms for direct solution of Poisson's equations with staggered boundary conditions. *J. Comput. Phys.*, **75**, 123-137.

Sussman, M., P. Smereka, and S. Osher, 1994: A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, **114(1)**, 146-159.

Tseng, Y. H., and J. H. Ferziger, 2003: A ghost-cell immersed boundary method for flow in complex geometry. *J. Comput. Phys.*, **192(2)**, 593-623.

Yang, G., D. M. Causon, D. M. Ingram, R. Saunders, and P. Batten, 1997: A Cartesian cut cell method for compressible flows part A: Static body problems. *Aeronaut. J.*, **101(1002)**, 47-56.

Ye, T., R. Mittal, H. S. Udaykumar, and W. Shyy, 1999: An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *J. Comput. Phys.*, **156(2)**, 209-240.