Consortium for Small-Scale Modelling

A Description of the

Nonhydrostatic Regional COSMO-Model

Part VII

User's Guide

U. Schättler, G. Doms, and C. Schraff

COSMO 6.0 September 2021

www.cosmo-model.org

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

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

Contents

1 Overview on the Model System 1

1.1 General Remarks . . . 1

1.2 Basic Model Design and Features . . . 3

1.3 Single Precision Version . . . 7

1.4 Organization of the Documentation . . . 8

2 Introduction 10

3 Model Formulation and Data Assimilation 12

3.1 Basic State and Coordinate-System . . . 12

3.2 Differential Form of Thermodynamic Equations . . . 14

3.3 Horizontal and Vertical Grid Structure . . . 15

3.4 Numerical Integration . . . 18

3.4.1 Runge-Kutta: 2-timelevel HE-VI Integration . . . 19

3.4.2 Leapfrog: 3-timelevel HE-VI Integration . . . 20

3.4.3 Leapfrog: 3-timelevel Semi-Implicit Integration . . . 20

3.5 Physical Parameterizations . . . 21

3.5.1 Radiation . . . 21

3.5.2 Grid-scale Precipitation . . . 21

3.5.3 Moist Convection . . . 23

3.5.4 Vertical Turbulent Diffusion . . . 25

3.5.5 Parameterization of Surface Fluxes . . . 26

3.5.6 A subgrid-scale orography scheme . . . 26

3.5.7 Soil Processes . . . 26

3.6 Data Assimilation 29

4 Installation of the COSMO-Model 31

4.1 External Libraries for the COSMO-Model 31

4.1.1 `libgrib1.a`: 31

4.1.2 `libgrib_api.a, libgrib_api_f90.a` or `libeccodes.a, libeccodes_f90.a`: 32

4.1.3 `libnetcdf.a`: 32

4.1.4 `libmisc.a`: 33

4.1.5 `libcsobank.a`, `libsupplement.a`: 33

4.1.6 `libRTTOVxx.a`: 33

4.2 Preparing the Code 34

4.3 Compiling and Linking 34

4.4 Running the Code 35

5 Input Files for the COSMO-Model 36

5.1 Files for Namelist Input 36

5.2 Initial and Boundary Data 38

5.3 Observation Input Files 39

5.3.1 Templates for observation types for which Table-Driven Code Forms (TDCF) defined by WMO exist 41

5.3.2 Observation types with templates proposed or approved by WMO . . 50

5.3.3 Observation types without templates proposed by WMO 56

5.3.4 The blacklist file 59

6 Namelist Input for COSMO-Model 62

6.1 `LMGRID` — Model Domain and Reference Atmosphere 65

6.2 `RUNCTL` — Parameters for the Model Run 67

6.3 `DYNCTL` — Parameters for the Adiabatic Model 79

6.4 `PHYCTL` — Parameters for the Diabatic Model 90

6.4.1 Different Configurations for the COSMO-ICON Physics 111

6.5 `TUNING` — Parameters for tuning dynamics and physics 117

6.6 `DIACTL` — Parameters for Diagnostic Output 123

6.7 SATCTL — Controlling the Synthetic Satellite Images 127

6.8 INICTL — Parameters for the Model Initialization 131

6.9 EPSCTL — Controlling the Ensemble Prediction Mode 133

6.10 IOCTL — Controlling the Grib I/O 138

6.11 GRIBIN — Controlling the Grib Input 142

6.12 GRIBOUT — Controlling the Grib Output 146

References **152**

Section 1

Overview on the Model System

1.1 General Remarks

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

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

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

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

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

Table 1.1: COSMO: Participating Meteorological Services

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

will apply.

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

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

The authors of this document recognize that typographical and other errors as well as discrepancies in the code and deficiencies regarding the completeness may be present, and your assistance in correcting them is appreciated. All comments and suggestions for improvement or corrections of the documentation and the model code are welcome and may be directed to the authors.

1.2 Basic Model Design and Features

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

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

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

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

Besides the operational application, the COSMO-Model provides a nonhydrostatic modeling framework for various scientific and technical purposes. Examples are applications of the model to large-eddy simulations, cloud resolving simulations, studies on orographic flow systems and storm dynamics, development and validation of large-scale parameterization schemes by fine-scale modeling, and tests of computational strategies and numerical techniques. For these types of studies, the model should be applicable to both real data cases and artificial cases using idealized test data. Moreover, the model has been adapted by other communities for applications in climate mode (CCLM) and / or running an online coupled module for aerosols and reactive trace gases (ART).

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

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

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

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

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

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

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

COSMO-ICON Physics

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

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

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

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

Dynamics

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

Numerics

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

Initial and Boundary Conditions

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

Physical Parameterizations

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

- **Grid-Scale Clouds and Precipitation** – Cloud water condensation and evaporation by saturation adjustment. Precipitation formation by a bulk microphysics parameterization including water vapour, cloud water, cloud ice, rain and snow with 3D transport for the precipitating phases. Option for a new bulk scheme including graupel. Option for a simpler column equilibrium scheme.

- **Subgrid-Scale Clouds** – Subgrid-scale cloudiness is interpreted by an empirical function depending on relative humidity and height. A corresponding cloud water content is also interpreted. Option for a statistical subgrid-scale cloud diagnostic for turbulence. This option has not been ported to the blocked data structure.

- **Moist Convection** – Tiedtke (1989) mass-flux convection scheme with equilibrium closure based on moisture convergence. Option for the current IFS Tiedtke-Bechtold convection scheme.

- **Shallow Convection** – Reduced Tiedtke scheme for shallow convection only.

- **Radiation** – δ two-stream radiation scheme after Ritter and Geleyn (1992) short and longwave fluxes (employing eight spectral intervals); full cloud-radiation feedback.

- **Soil Model** – Multi-layer version of the former two-layer soil model after Jacobsen and Heise (1982) based on the direct numerical solution of the heat conduction equation. Snow and interception storage are included.

- **Fresh-Water Lake Parameterization** – Two-layer bulk model after Mironov (2008) to predict the vertical temperature structure and mixing conditions in fresh-water lakes of various depths.

- **Sea-Ice Scheme** – Parameterization of thermodynamic processes (without rheology) after Mironov and Ritter (2004). The scheme basically computes the energy balance at the ice's surface, using one layer of sea ice.

- **Terrain and Surface Data** – All external parameters of the model are available at various resolutions for a pre-defined region covering Europe. For other regions or grid-spacings, the external parameter file can be generated by a preprocessor program using high-resolution global data sets.

Data Assimilation

- **Former Method** – Continuous four-dimensional data assimilation based on observation nudging (Schraff (1996), Schraff (1997)), with lateral spreading of upper-air observation increments along horizontal surfaces. Explicit balancing by a hydrostatic temperature correction for surface pressure updates, a geostrophic wind correction, and a hydrostatic upper-air pressure correction.

- **Actual Method** – Ensemble data assimilation based on the LETKF (Local Ensemble Transform Kalman Filter) (Schraff et al. (2016))

- **Assimilated Atmospheric Observations** – Radiosonde (wind, temperature, humidity), aircraft (wind, temperature), wind profiler (wind), and surface-level data (SYNOP, SHIP, BUOY: pressure, wind, humidity). Optionally RASS (temperature), radar VAD wind, and ground-based GPS (integrated water vapour) data. Surface-level temperature is used for the soil moisture analysis only.

- **Radar derived rain rates** – Assimilation of near surface rain rates based on latent heat nudging (Stephan et al. (2008)). It locally adjusts the three-dimensional thermodynamical field of the model in such a way that the modelled precipitation rates should resemble the observed ones.

- **Surface and Soil Fields** – Additional two-dimensional intermittent analysis:
  - **Soil Moisture Analysis** – Daily adjustment of soil moisture by a variational method (Hess (2001)) in order to improve 2-m temperature forecasts; use of a Kalman-Filter-like background weighting.

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

Code and Parallelization

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

1.3 Single Precision Version

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

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

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

The main changes are:

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

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

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

1.4 Organization of the Documentation

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

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

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

Parts I - III form the scientific documentation, which provides information about the theoretical and numerical formulation of the model, the parameterization of physical processes and the four-dimensional data assimilation. The scientific documentation is independent of (i.e. does not refer to) the code itself. Part IV will describe the particular implementation of the methods and algorithms as presented in Parts I - III, including information on the basic code design and on the strategy for parallelization using the MPI library for message passing on distributed memory machines (not available yet). The generation of initial and boundary conditions from coarse grid driving models is described in Part V. This part is a description of the interpolation procedures and algorithms used (not yet complete) as well as a User's Guide for the interpolation program INT2LM. In Part VI we give a description of the data formats, which can be used in the COSMO-Model, and describe the output from the model and from data assimilation. Finally, the User's Guide of the COSMO-Model provides information on code access and how to install, compile, configure and run the model. The User's Guide contains also a detailed description of various control parameters in the model input file (in NAMELIST format) which allow for a flexible model set-up for various applications. All parts of the documentation are available at the COSMO web-site (http://www.cosmo-model.org/content/model/documentation/core/default.htm).

Section 2

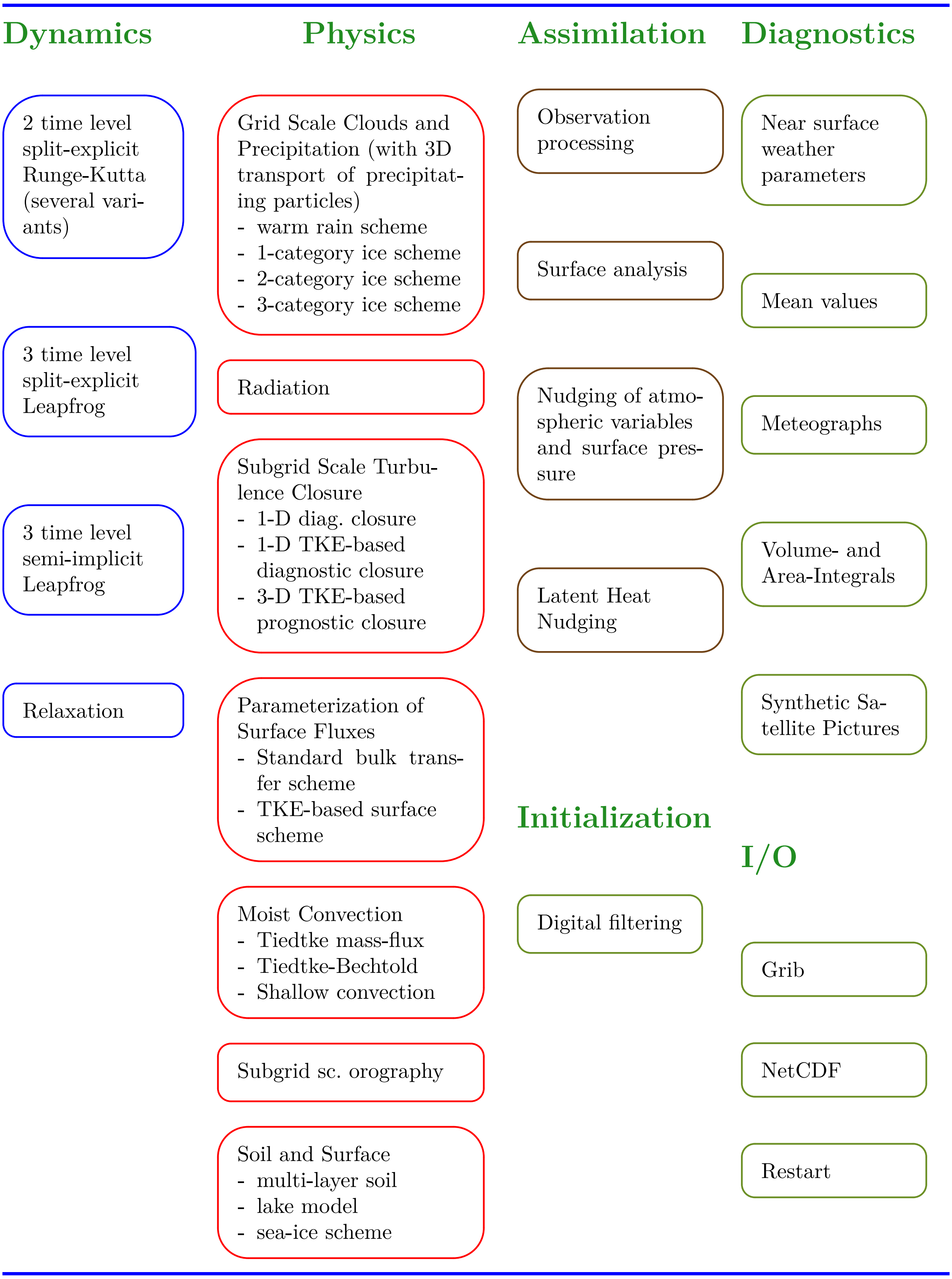
Introduction

The usage of the program package for the COSMO-Model is a rather complex task, both, for the experienced and even more for the non-experienced user. This User's Guide serves in a first instance as a complete reference for all the different NAMELIST groups and variables, with which the execution of the model can be controlled.

But first, an overview on the model formulation and the data assimilation is given in Section 3. The installation of the package is explained in Section 4, which also gives information on external libraries used in the COSMO-Model. Necessary input files of the model are listed in Section 5 and the detailed descriptions of all NAMELIST variables are given in Sections 6 and **??**.

Knowing the meaning of all NAMELIST-variables normally is not enough to find the way through the possible configurations of the model. Therefore, a description would be desirable that explains how the variables can be put together to give a meaningful setup, or which variable settings contradict each other or simply are not possible. Such a description is not available, but we tried to indicate dependencies of variables in the descriptions in Section 6. We also refer to the web-page of the CLM Community. They implemented a NAMELIST-tool, where you can find different practical setups for the climate and for the NWP mode: (https://wiki.coast.hzg.de/clmcom/tools-98599080.html).

Figure 2.1: Schematic view of the different COSMO-Model components


Dynamics
Physics
Assimilation
Diagnostics
2 time level split-explicit Runge-Kutta (several variants)
3 time level split-explicit Leapfrog
3 time level semi-implicit Leapfrog
Relaxation
Grid Scale Clouds and Precipitation (with 3D transport of precipitating particles)
- warm rain scheme
- 1-category ice scheme
- 2-category ice scheme
- 3-category ice scheme
Radiation
Subgrid Scale Turbulence Closure
- 1-D diag. closure
- 1-D TKE-based diagnostic closure
- 3-D TKE-based prognostic closure
Parameterization of Surface Fluxes
- Standard bulk transfer scheme
- TKE-based surface scheme
Moist Convection
- Tiedtke mass-flux
- Tiedtke-Bechtold
- Shallow convection
Subgrid sc. orography
Soil and Surface
- multi-layer soil
- lake model
- sea-ice scheme
Observation processing
Surface analysis
Nudging of atmospheric variables and surface pressure
Latent Heat Nudging
Initialization
Digital filtering
Near surface weather parameters
Mean values
Meteographs
Volume- and Area-Integrals
Synthetic Satellite Pictures
I/O
Grib
NetCDF
Restart

Section 3

Model Formulation and Data Assimilation

3.1 Basic State and Coordinate-System

The COSMO-Model is based on the primitive hydro-thermodynamical equations describing compressible nonhydrostatic flow in a moist atmosphere without any scale approximations. A basic state is subtracted from the equations to reduce numerical errors associated with the calculation of the pressure gradient force in case of sloping coordinate surfaces. The basic state represents a time-independent dry atmosphere at rest which is prescribed to be horizontally homogeneous, vertically stratified and in hydrostatic balance.

By introducing the basic state, the thermodynamic variables temperature (T), pressure (p) and density (ρ) can be formally written as the sum of a height dependent reference value and a space and time dependent deviation:

$$T = T_0(z) + T', \qquad p = p_0(z) + p', \qquad \rho = \rho_0(z) + \rho' \,, \tag{3.1}$$

where $T_0(z)$, $p_0(z)$ and $\rho_0(z)$ are related by the hydrostatic equation

$$\frac{\partial p_0}{\partial z} = -g\rho_0 = -\frac{g p_0}{R_d T_0} \tag{3.2}$$

and the equation of state, $p_0 = \rho_0 R_d T_0$. R_d is the gas constant of dry air. The vertical profile $T_0(z)$ of temperature can be specified arbitrary since we do not linearize the model equations with respect to the reference state.

In the first implementation of the COSMO-Model we prescribed a constant rate β for the temperature increase with the logarithm of pressure (as proposed by Dudhia (1993)), $\partial T_0 / \partial \ln p_0 = \beta$. The integration of the hydrostatic equation (3.2) with the boundary values $p_{SL} = p_0\,(z=0)$ and $T_{SL} = T_0\,(z=0)$ for the pressure and temperature at mean sea level $z = 0$ then yields the vertical profiles of the reference state:

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

$$T_0(z) \quad = \quad T_{SL}\sqrt{1-\frac{2\beta g z}{R_d T_{SL}^2}}. \tag{3.3}$$

For the three parameters p_{SL}, T_{SL} and β, which define the basic state, we use the default values p_{SL} = 1000hPa, T_{SL} = 288.15K and β = 42K. The variable names in the programs are `p0sl` (p_{SL}), `t0sl` (T_{SL}) and `dt0lp` (β), resp. This basic state is still available in the COSMO-Model and can be chosen as *Reference Atmosphere 1* (`irefatm=1`).

Since COSMO-Model 4.5, a new alternative reference atmosphere has been implemented, which can be chosen as *Reference Atmosphere 2* (`irefatm=2`). This reference atmosphere is based on the temperature profile

$$T_0(z) = T_{00} + \delta_T \cdot \exp(-z/h_{scal}), \tag{3.4}$$

with default values of T_{00} = 213.15K, δ_T = 75K and h_{scal} = 10km. In the model code, $T_{00} = T_{SL} - \delta_T$ = `t0sl - delta_t`. Thus, the reference atmosphere approaches an isothermal profile in the stratosphere, whereas the existing reference profile has an increasingly negative vertical temperature gradient in the stratosphere. The vertical extent of the model domain is no longer limited with the new reference atmosphere.

The new reference atmosphere needs two additional parameters δ_T (model variable `delta_t`) and h_{scal} (model variable `h_scal`). Default values are `delta_t=75.0` and `h_scal=10000.0`, resp.

The model equations are formulated with respect to a rotated lat/lon-grid with coordinates (λ, φ). The rotated coordinate system results from the geographical (λ_g, φ_g) coordinates by tilting the north pole (see Part I of the Documentation, *Dynamics and Numerics*). In the vertical, we use a generalized terrain-following height coordinate ζ, where any unique function of geometrical height can be used for transformation. Since ζ doesn't depend on time, the $(\lambda, \varphi, \zeta)$-system represents a non-deformable coordinate system, where surfaces of constant ζ are fixed in space - in contrast to the pressure based coordinate system of most hydrostatic models, where the surfaces of constant vertical coordinate move in space with changing surface pressure.

The transformation of the model equations from the orthogonal (λ, φ, z)-system to the non-orthogonal terrain-following $(\lambda, \varphi, \zeta)$-system is given by the three elements of the inverse Jacobian matrix $\mathcal{J}^z$,

$$J_\lambda \equiv J_{13}^z = \left(\frac{\partial z}{\partial \lambda}\right)_\zeta, \quad J_\varphi \equiv J_{23}^z = \left(\frac{\partial z}{\partial \varphi}\right)_\zeta, \quad J_\zeta \equiv J_{33}^z = \frac{\partial z}{\partial \zeta} = -\sqrt{G}. \tag{3.5}$$

The terrain-following ζ-system of the COSMO-Model is defined to be left-handed, i.e. the value of the ζ-coordinate increases with decreasing height z from the top of the model to the surface. Thus, J_ζ is always negative and equal to the negative absolute value ($\sqrt{G} = |\det(\mathcal{J}^z)|$) of the determinant of the inverse Jacobi matrix.

3.2 Differential Form of Thermodynamic Equations

By transforming the primitive hydro-thermodynamical equations to the $(\lambda, \varphi, \zeta)$ coordinate-system and subtracting the basic state, we achieve the following set of prognostic model equations for the three components u, v and w of the wind vector, the perturbation pressure p', the temperature T and the humidity variables q.

$$\begin{aligned}
\frac{\partial u}{\partial t} + \mathbf{v}\cdot\nabla u - \frac{uv}{a}\tan\varphi - fv &= -\frac{1}{\rho a\cos\varphi}\left(\frac{\partial p'}{\partial\lambda} + \frac{J_\lambda}{\sqrt{G}}\frac{\partial p'}{\partial\zeta}\right) + M_u \\
\frac{\partial v}{\partial t} + \mathbf{v}\cdot\nabla v + \frac{u^2}{a}\tan\varphi + fu &= -\frac{1}{\rho a}\left(\frac{\partial p'}{\partial\varphi} + \frac{J_\varphi}{\sqrt{G}}\frac{\partial p'}{\partial\zeta}\right) + M_v \\
\frac{\partial w}{\partial t} + \mathbf{v}\cdot\nabla w &= \frac{1}{\rho\sqrt{G}}\frac{\partial p'}{\partial\zeta} + B + M_w \\
\frac{\partial p'}{\partial t} + \mathbf{v}\cdot\nabla p' - g\rho_0 w &= -(c_{pd}/c_{vd})pD \\
\frac{\partial T}{\partial t} + \mathbf{v}\cdot\nabla T &= -\frac{p}{\rho c_{vd}}D + Q_T \\
\frac{\partial q^v}{\partial t} + \mathbf{v}\cdot\nabla q^v &= -(S^l + S^f) + M_{q^v} \\
\frac{\partial q^{l,f}}{\partial t} + \mathbf{v}\cdot\nabla q^{l,f} + \frac{1}{\rho\sqrt{G}}\frac{\partial P_{l,f}}{\partial\zeta} &= S^{l,f} + M_{q^{l,f}}
\end{aligned} \tag{3.6}$$

Here, the continuity equation has been replaced by an equation for p'. In Eqs. (3.6) a is the radius of the earth, c_{pd} and c_{vd} are the specific heat of dry air at constant pressure and constant volume, g is the gravity acceleration, f is the Coriolis parameter, R_v and R_d are the gas constants for water vapour and dry air. ρ is the density of moist air which is calculated as a diagnostic variable from the equation of state:

$$\rho = p\{R_d(1 + (R_v/R_d - 1)q^v - q^l - q^f)T\}^{-1}\,. \tag{3.7}$$

q^v is the specific humidity, q^l represents the specific water content of a category of liquid water (cloud or rain water) and q^f represents the specific water content of a category of frozen water (cloud ice, snow or graupel). The corresponding precipitation fluxes are denoted by P_l and P_f.

The terms M_ψ denote contributions from subgrid-scale processes as, e.g. turbulence and convection and Q_T summarizes the diabatic heating rate due to this processes. The various sources and sinks in the equations for the humidity variables due to microphysical processes of cloud and precipitation formation are denoted by S^l and S^f. The calculation of all these terms related to subgrid-scale processes is done by physical parameterization schemes. An overview of the schemes used in the COSMO-Model is given in Section 3.5.

The term B in the equation for the vertical velocity is the buoyant acceleration given by

$$B = g\frac{\rho_0}{\rho}\left\{\frac{T - T_0}{T} - \frac{p' T_0}{p_0 T} + \left(\frac{R_v}{R_d} - 1\right)q^v - q^l - q^f\right\}. \tag{3.8}$$

The advection operator in terrain-following coordinates is defined as

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

where $\dot{\zeta}$ is the contra-variant vertical velocity in the ζ-system:

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

D is the three-dimensional wind divergence which is calculated from

$$D = \frac{1}{a\cos\varphi} \left\{ \frac{\partial u}{\partial \lambda} + \frac{J_\lambda}{\sqrt{G}} \frac{\partial u}{\partial \zeta} + \frac{\partial}{\partial \varphi} (v \cos\varphi) + \cos\varphi \frac{J_\varphi}{\sqrt{G}} \frac{\partial v}{\partial \zeta} \right\} - \frac{1}{\sqrt{G}} \frac{\partial w}{\partial \zeta}.$$

In deriving the prognostic equation for the perturbation pressure from the continuity equation, a source term due to diabatic heating has been neglected. For most meteorological applications, this source term is much smaller than the forcing by divergence. This approximation is also used in many other nonhydrostatic simulation models.

3.3 Horizontal and Vertical Grid Structure

The model equations (3.6) are solved numerically using the traditional finite difference method. In this technique, spatial differential operators are simply replaced by suitable finite difference operators. The time integration is also by discrete stepping using a fixed timestep Δt.

The terrain-following coordinate system with the generalized vertical coordinate ζ allows to map the irregular grid associated with the terrain-following system in physical space onto a rectangular and regular computational grid. Thus, constant increments

$$\begin{aligned} \Delta\lambda &: \text{ grid-spacing in longitudinal direction,} \\ \Delta\varphi &: \text{ grid-spacing in latitudinal direction,} \\ \Delta\zeta &: \text{ grid-spacing in } \zeta\text{-direction } (\Delta\zeta = 1), \end{aligned}$$

of the independent variables are used to set up the computational grid. To simplify the notation, we set the vertical grid-spacing equal to one (see below). The discrete computational $(\lambda, \varphi, \zeta)$-space is then represented by a finite number of grid points (i, j, k), where i corresponds to the λ-direction, j to the φ-direction and k to the ζ-direction. The position of the grid points in the computational space is defined by

$$\begin{aligned} \lambda_i &= \lambda_0 + (i-1)\,\Delta\lambda\,, & i &= 1, \cdots, N_\lambda \\ \varphi_j &= \varphi_0 + (j-1)\Delta\varphi, & j &= 1, \cdots, N_\varphi \\ \zeta_k &= k\,, & k &= 1, \cdots, N_\zeta\,. \end{aligned} \tag{3.9}$$

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

Every grid point (i, j, k) represents the centre of an elementary rectangular grid volume with side lengths $\Delta\lambda$, $\Delta\varphi$ and $\Delta\zeta$. The grid-box faces are located halfway between the grid points in the corresponding directions, i.e. at $\lambda_{i\pm1/2}$, $\varphi_{j\pm1/2}$ and $\zeta_{k\pm1/2}$.

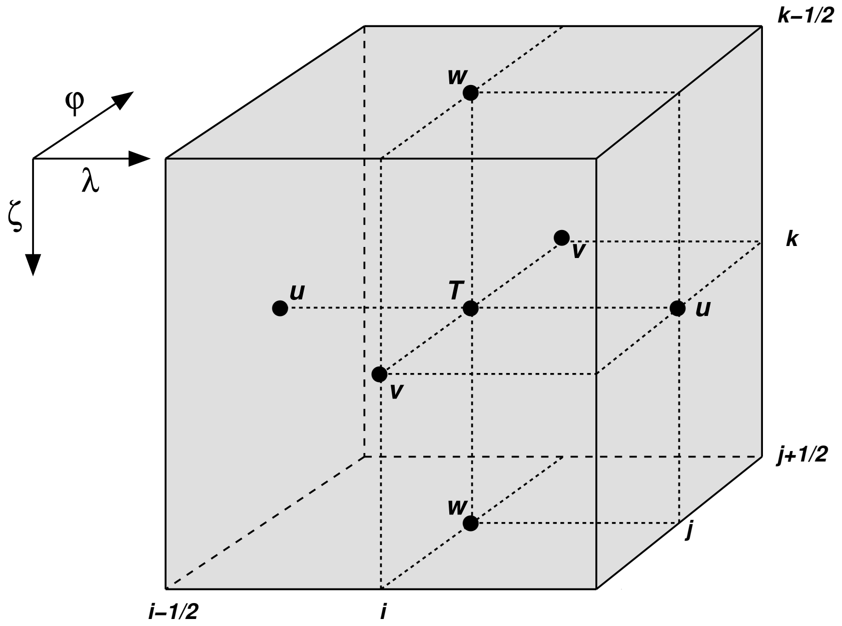


Figure 3.1: A grid box volume $\Delta V = \Delta\zeta\,\Delta\lambda\,\Delta\varphi$ showing the Arakawa-C/Lorenz staggering of the dependent model variables.

The model variables are staggered on an Arakawa-C/Lorenz grid with scalars (temperature, pressure and humidity variables) defined at the centre of a grid box and the normal velocity components defined on the corresponding box faces (see Figure 3.1). For a given grid spacing, this staggering allows for a more accurate representation of differential operators than in the A-grid, where all variables are defined at the same point. In general, we use second order centered finite difference operators, i.e. the numerical discretization error is reduced by a factor of four when we increase the resolution by a factor of two. For a detailed description of the numerical operators see Part I of the Documentation, *Dynamics and Numerics*.

The grid-box faces in vertical direction are usually referred to as the half levels. These interfacial levels separate the model layers from each other. The model layers labeled by integers k are also denoted as main levels. Thus, for a model configuration with N_ζ layers we have $N_\zeta + 1$ half levels. The top boundary of the model domain is defined to be the half level ($\zeta = 1/2$) above the uppermost model layer ($\zeta = 1$). At the lower boundary, the ζ-coordinate surface becomes conformal to the terrain height. The half level ($\zeta = N_\zeta + 1/2$) below the first model layer above the ground ($\zeta = N_\zeta$) defines the lower boundary of the model.

The discrete formulation of the model equations is independent on a specific choice for the vertical coordinate. This is achieved by a two-step transformation procedure: First we apply a transformation to a specific terrain-following system, where in principle any unique function of geometrical height z can be used. In the first implementation of the COSMO-Model, either a generalized sigma-type coordinate η based on base-state pressure (`ivctype=1`) or a generalized Gal-Chen coordinate μ based on height (`ivctype=2`) could be chosen. Later, two variants of the *Smooth LEvel VErtical coordinate* (SLEVE) have been added (`ivctype=3/4`).

In a second step this vertical coordinate is mapped onto the computational coordinate ζ with discrete coordinate values $\zeta_k = k$ and an equidistant grid spacing of $\Delta\zeta = 1$. The latter mapping is by a table which relates specific values of the terrain-following coordinate η or

μ to the $N_\zeta + 1$ values of the half-level values $\zeta_{k+1/2}$. In this way a user-defined vertical grid-stretching can be easily applied. Details on the set-up of the vertical grid are provided in Part I of the Documentation, *Dynamics and Numerics*.

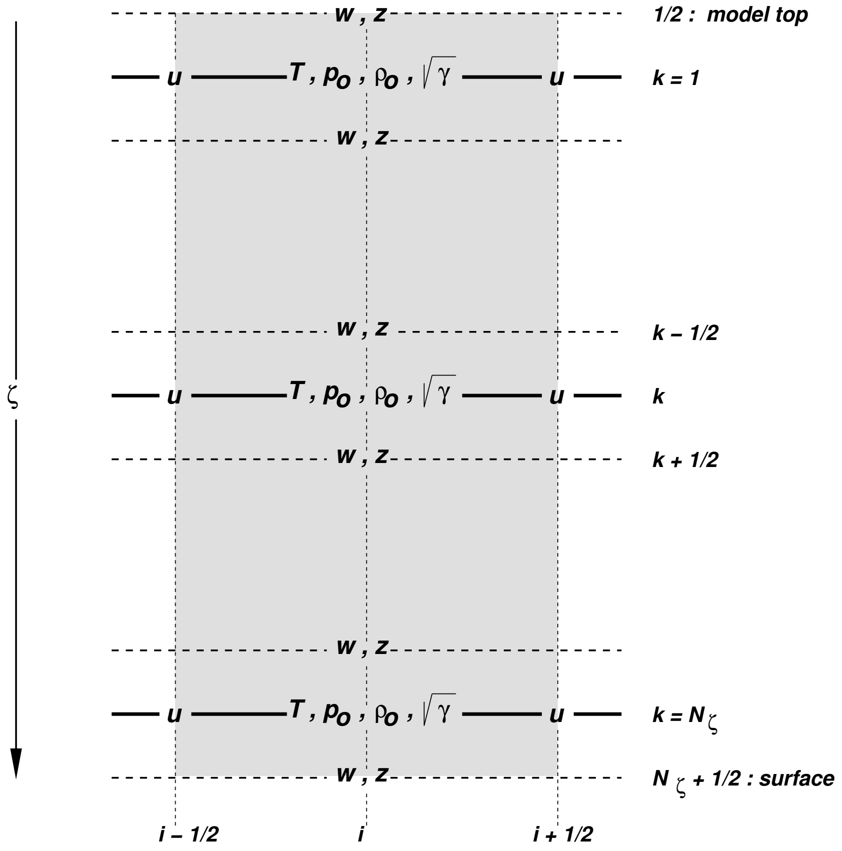


Figure 3.2: Vertical staggering of variables and metric terms in a grid box column with N_ζ layers. Dashed lines are the model half levels separating the main levels (full lines).

To render the model code independent on η or μ, all metric terms involving the three components (3.5) of the Jacobi-matrix are evaluated numerically on the computational grid. These terms are rewritten in the form

$$\sqrt{G} = \frac{1}{g\rho_0}\sqrt{\gamma}\,, \quad \frac{J_\lambda}{\sqrt{G}} = -\frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \lambda}\,, \quad \frac{J_\varphi}{\sqrt{G}} = -\frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \varphi}\,, \tag{3.10}$$

where $\sqrt{\gamma} \equiv \partial p_0/\partial\zeta$ denotes the change of base-state pressure with ζ. In discretized form we have

$$\begin{aligned} \sqrt{\gamma}_k &= (\Delta p_0)_k = (p_0)_{k+1/2} - (p_0)_{k-1/2}\,, \\ (p_0)_k &= \frac{1}{2}\{(p_0)_{k+1/2} + (p_0)_{k-1/2}\}\,. \end{aligned} \tag{3.11}$$

for $\sqrt{\gamma}$ and the base-state pressure p_0 on model main levels. Additionally, the height of model half levels $z_{k+1/2}$ resulting from the coordinate transformation is stored as a 3-D array HHL.

The base-state density on main levels then results from the discretized hydrostatic relation

$$(\rho_0)_k \;=\; \frac{1}{g}\frac{\sqrt{\gamma}_k}{z_{k-1/2} - z_{k+1/2}}$$

and the main level base-state temperature results from the equation of state. Fig. 3.2 illustrates the vertical staggering of model variables as well as base state variables and metric terms used in the discretization.

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

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

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

All grid points interior to the physical boundary constitute the computational (or model interior) domain, where the model equations are integrated numerically. These are points with subscripts (i, j) running from $i = N_{\text{off}}+1, \cdots, N_\lambda - N_{\text{off}}$ and $j = N_{\text{off}}+1, \cdots, N_\varphi - N_{\text{off}}$. The extra points outside the interior domain constitute the computational boundaries. At these points, all model variables are defined and set to specified boundary values, but no dynamical computations are done. For $N_{\text{off}} = 2$, we have two extra lines of grid points adjacent to each physical boundary (see Fig. 3.3).

3.4 Numerical Integration

Because the governing nonhydrostatic equations describe a compressible model atmosphere, meteorologically unimportant sound waves are also part of the solution. As acoustic waves are very fast, their presence severely limits the time step of explicit time integration schemes. In order to improve the numerical efficiency, the prognostic equations are separated into terms which are directly related to acoustic and gravity wave modes and into terms which refer to comparatively slowly varying modes of motion. This mode-splitting can formally be written in the symbolic form

$$\frac{\partial \psi}{\partial t} \;=\; s_\psi \,+\, f_\psi \,, \tag{3.12}$$

where ψ denotes a prognostic model variable, f_ψ the forcing terms due to the slow modes and s_ψ the source terms related to the acoustic and gravity wave modes. s_ψ is made up of the pressure gradient terms in the momentum equations, the temperature and pressure contributions to the buoyancy term in the w-equation and the divergence term in the pressure and the temperature equation. The subset of equations containing the s_ψ-terms is then integrated with a special numerical scheme. The COSMO-Model provides four different integration methods.

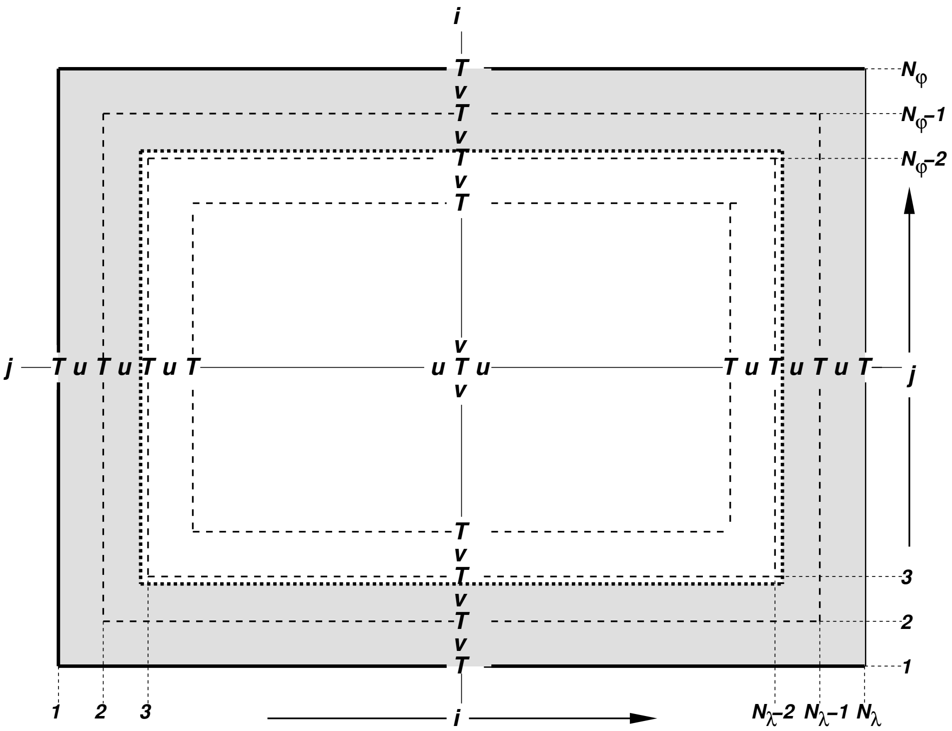


Figure 3.3: Horizontal model domain for $N_\lambda \times N_\varphi$ grid points and an offset of $N_{\text{off}} = 2$ for the position of the physical boundaries (dotted). The computational boundaries are shaded; the integration is done for variables in the interior computational domain ($i = 3, \cdots, N_\lambda - 2$ and $j = 3, \cdots, N_\varphi - 2$).

3.4.1 Runge-Kutta: 2-timelevel HE-VI Integration

Basic Namelist settings: `l2tls=.TRUE.; lsemi_imp=.FALSE.; irunge_kutta=1/2`
This scheme (with `irunge_kutta=1`) is used for the COSMO-DE and COSMO-EU.

This scheme has been implemented into the COSMO-Model as an alternative to the former default, the Leapfrog scheme, and can be combined with a forward-backward scheme for integrating the high-frequency modes of the elastic equations. The first (`irunge_kutta=1`) variant is the normal 3rd-order Runge-Kutta scheme used by Wicker and Skamarock (2002) whereas the second one is a total variation diminishing (TVD) variant of 3rd-order (Liu et al. (1994)) (`irunge_kutta=2`).

Different horizontal advection upwind or centered-differences schemes of 3rd- to 6th-order can be used – the operators are formulated in advection form. The vertical advection is normally treated in an implicit way using a Crank-Nicolson scheme and centered-differences in space. Most slow tendencies such as vertical diffusion, thermal/solar heating, parameterized convection and coriolis force are computed only once using values of the prognostic variables at time step n. These tendencies are fixed during the individual Runge-Kutta steps and contribute to the total slow-mode tendencies which are integrated in several small time steps together with the fast-mode tendencies in a time-splitting sense. In contradiction to this, the whole 3D-advection is computed in each Runge-Kutta step.

3.4.2 Leapfrog: 3-timelevel HE-VI Integration

Basic Namelist settings: `l2tls=.FALSE.; lsemi_imp=.FALSE.`

This method is a variant of the Klemp and Wilhelmson (1978) scheme which is based on a Leapfrog integration for the slow modes from time level $n-1$ to time level $n+1$ using an integration interval of $2\Delta t$. The slow mode tendencies are evaluated at time level n for horizontal advection using standard second order centered differences and at time level $n-1$ for most physical forcings. Vertical advection and vertical diffusion are calculated by a quasi-implicit scheme. The integration step is then subdivided into a number N_s of small time steps $\Delta\tau_s$ according to $2\Delta t = N_s \Delta\tau$ and the prognostic equations (3.12) are stepped forward according to

$$\psi^{\nu+1} = \psi^{\nu} + s_{\psi}^{\nu}\Delta\tau + f_{\psi}^{n}\Delta\tau\,. \tag{3.13}$$

Figure 3.4 illustrates the basic idea of the time-splitting scheme. In the integration of (3.13), sound waves are treated explicitly for horizontal directions using the forward-backward method while implicitly for the vertical direction (HE-VI). Thus, the small time step $\Delta\tau$ is limited by the CFL stability criterion for horizontal but not for vertical sound wave propagation. This makes the HE-VI scheme numerically very efficient for large grid aspect ratios, i.e. $\Delta x/\Delta z \gg 1$, which are typically used in meso-β and meso-γ applications.

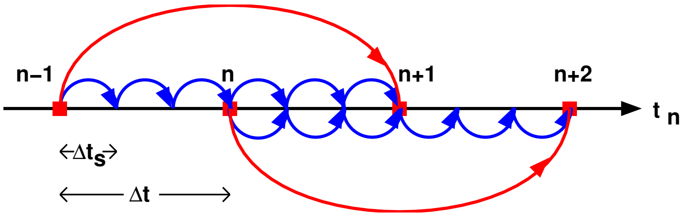


Figure 3.4: The time splitting algorithm

An additional 3-D divergence damping as well as a slight time off-centering in the vertical implicit formulation is applied to damp acoustic modes. On the big time step, the Asselin time filter and a 4th order horizontal diffusion are used for numerical smoothing. While this 3-timelevel HE-VI integration was the default time scheme of the COSMO-Model in the beginning, it has now been replaced by the 2-timelevel Runge-Kutta schemes..

3.4.3 Leapfrog: 3-timelevel Semi-Implicit Integration

Basic Namelist settings: `l2tls=.FALSE.; lsemi_imp=.TRUE.`

Because the HE-VI scheme integrates the horizontal momentum equations explicitly, steep orography may provoke instabilities in small-scale applications. Full 3D semi-implicit schemes can avoid such stability problems by treating all pressure gradient and divergence terms implicitly both vertically and horizontally (HI-VI-scheme) - thus, a small time step is not used. Moreover, 3D semi-implicit schemes have the potential to become more cost-effective than split-explicit schemes at higher resolution where the grid aspect ratio is more isotropic and where the number of small time steps increases with the sound speed Courant number for low Mach number flows.

The derivation of the scheme is based on the 3-timelevel Leapfrog integration and uses the time-tendency formulation to minimize cancellation errors. An elliptic equation for the pressure perturbation tendency

$$\mathcal{L}\,(\delta_\tau(p') = q_p$$

is obtained by forming the divergence of the momentum equations and eliminating the buoyancy terms. However, the use of a nonorthogonal curvilinear coordinate system results in an elliptic operator $\mathcal{L}$ containing cross-derivative terms with variable coefficients. A minimal residual Krylov iterative solver (GMRES) was thus chosen to solve for the perturbation pressure tendency. We found the convergence criterion proposed by Skamarock et al. (1997) to be both sufficient and a robust predictor of when the RMS divergence of the flow has stabilized. An efficient line-Jacobi relaxation preconditioner was developed having the property that the number of Krylov solver iterations grows slowly as the convergence parameter ε_c decreases. Once the solution for the pressure tencendy is known, the other variables are updated by back-substitution.

3.5 Physical Parameterizations

Some parts of the physics package of the COSMO-Model are adapted from the former operational hydrostatic model DM. Others have been widely rewritten or were replaced by new developments. This section gives a short overview on the parameterization schemes used. A detailed description is given in Part II of the Documentation, *Physical Parameterizations*.

3.5.1 Radiation

Basic Namelist settings: `lphys=.TRUE.; lrad=.TRUE.; hincrad=1.0`

To calculate the heating rate due to radiation we employ the parameterization scheme of Ritter and Geleyn (1992). This scheme is based on a δ-two-stream version of the general equation for radiative transfer and considers three shortwave (solar) and five longwave (thermal) spectral intervals. Clouds, aerosol, water vapour and other gaseous tracers are treated as optically active constituents of the atmosphere, which modify the radiative fluxes by absorption, emission and scattering.

As an extension to the original scheme, a new treatment of the optical properties of ice particles has been introduced which allows a direct cloud-radiative feedback with the predicted ice water content when using the cloud ice scheme for the parameterization of cloud and precipitation.

Numerically, the parameterization scheme is very cost-intensive. Thus, it is called only at hourly intervals during an operational forecast on the meso-β scale. The resulting short- and longwave heating rates are then stored and remain fixed for the following time interval. In case of high resolution simulations, the calling frequency of the radiation scheme can be increased to allow for a better representation of the interaction with the cloud field. The radiation can also be computed on a coarser grid to save computation time.

3.5.2 Grid-scale Precipitation

Basic Namelist settings: `lphys=.TRUE.; lgsp=.TRUE.`

The basic parameterization scheme for the formation of grid-scale clouds and precipitation is an adapted version of the DM-scheme. It is based on a Kessler-type bulk formulation and uses a specific grouping of various cloud and precipitation particles into broad categories of water substance. The particles in these categories interact by various microphysical processes which in turn have feedbacks with the overall thermodynamics. Microphysical processes are parameterized by corresponding mass transfer rates between the categories and are formulated in terms of the mixing ratios as the dependent model variables.

Besides water vapour in the gaseous phase three categories of water are considered by the default scheme:

- *cloud water* is in the form of small suspended liquid-phase drops. Cloud droplets are smaller than about 50 μm in radius and thus have no appreciable terminal fall speed relative to the airflow.
- *rain water* is in the form of liquid-phase spherical drops which are large enough to have a non-negligible fall velocity. An exponential Marshall-Palmer size-distribution is assumed for the raindrops and a drop terminal velocity depending only on drop diameter is prescribed.
- *Snow* is made up of large rimed ice particles and rimed aggregates which are treated as thin plates with a specific size-mass relation. Particles in this category have a non-negligible terminal velocity which is prescribed to depend only on particle size. An exponential Gunn-Marshall size-distribution is assumed.

The budget equation for the specific water contents q of the various categories (water vapour q^v, cloud water q^c, cloud ice q^i and graupel q^g, depending on the scheme used) take advective and turbulent transport into account and contain source and sink terms due to the microphysical processes of cloud and precipitation formation. For rain water q^r and snow q^s, only advective transport is considered. The following mass-transfer rates are considered by the scheme:

(a) condensation and evaporation of cloud water,

(b) the initial formation of rainwater by autoconversion and of snow by nucleation from the cloud water phase,

(c) the subsequent growth of the precipitation phases rain and snow by accretion, riming, deposition and shedding,

(d) evaporation of rainwater and sublimation of snow in subcloud layers and

(e) melting of snow to form rain and freezing of rain to form snow.

The impact of the vertical motion of rain and snow relative to the airflow due to the sedimentation of particles with their terminal velocities is also taken into account by the vertical divergence of the corresponding precipitation fluxes P_r and P_s. Figure 3.5 illustrates the microphysical processes considered by this parameterization scheme.

In contrast to the former diagnostic precipitation scheme with its assumption of column equilibrium for the precipitating particles, we now solve the complete prognostic equations

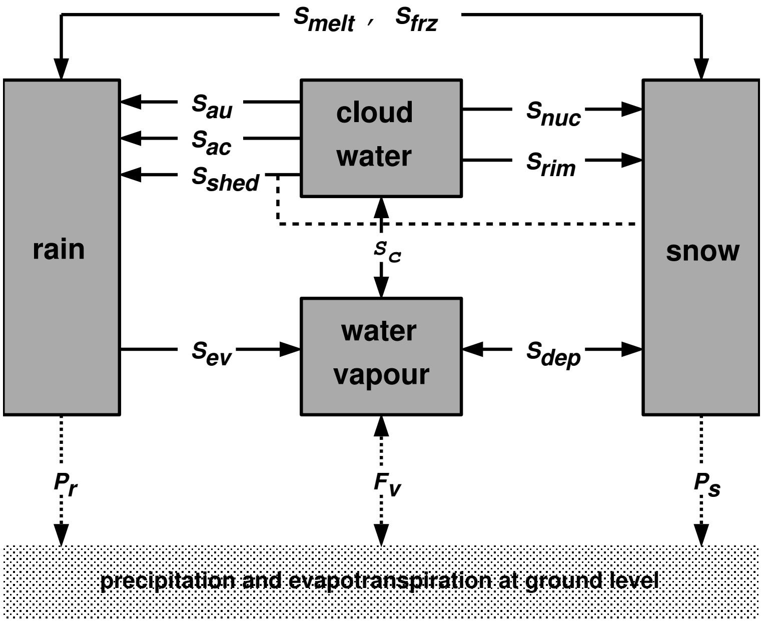


Figure 3.5: Hydrological cycle in the COSMO-Model cloud and precipitation scheme

for rain q^r and and snow q^s This approach is therefore applicable to the meso-γ and smaller scales.

These options are available for the microphysics parameterization scheme:

(a) `itype_gscp=1`: A *warm rain scheme* which is similar to the original Kessler (1969) scheme; all ice-phase processes are ignored.

(b) `itype_gscp=2`: The basic scheme described above.

(c) `itype_gscp=3`: An extension of the basic scheme which includes cloud ice as an additional prognostic variable (*cloud ice scheme*). The scheme allows for an explicit representation of ice clouds and a more complete simulation of precipitation formation in mixed phase clouds. This scheme is used in the COSMO-EU.

(d) `itype_gscp=4`: A graupel-scheme (in addition to cloud-ice) has been implemented recently. It allows for an explicit simulation of deep convective clouds. This is available only as a prognostic scheme. This scheme is used in the COSMO-DE.

The use of `itype_gscp=1/2` is not recommended for real cases. They are only used in idealized test cases.

3.5.3 Moist Convection

Basic Namelist settings: `lphys=.TRUE.; lconv=.TRUE.`

For model applications on the meso-α and meso-β scales down to grid spacings of 5-10 km, cumulus convection is a subgrid-scale process which requires a parameterized representation. And even on the meso-γ scale it turned out, that a parameterization of shallow convection still is necessary. The COSMO-Model offers three options:

(a) Mass flux Tiedtke scheme
Basic Namelist settings: `itype_conv=0`
The mass flux scheme of Tiedtke (1989), which is used in coarser grid applications (above 3 km), has been implemented for the meso-α and meso-β scale. This parameterization discriminates three types of moist convection: *shallow convection*, *penetrative convection* and *midlevel convection*, which are treated by different closure conditions. Both shallow and penetrative convection have their roots in the atmospheric boundary layer but they differ in vertical extent. Midlevel convection, on the other hand, has its roots not in the boundary layer but originates at levels within the free atmosphere.

As a closure condition, the Tiedtke scheme requires a formulation of the vertical mass flux at the convective cloud base in terms of the grid-scale variables. For shallow and penetrative convection, it is assumed that this mass flux is proportional to the vertically integrated moisture convergence between the surface and the cloud base. In case of midlevel convection, the mass flux is simply set proportional to the grid-scale vertical velocity.

Given the mass flux at cloud base, the vertical redistribution of heat, moisture and momentum as well as the formation of precipitation is then calculated by integrating a simple stationary cloud model for both updrafts and downdrafts. This finally allows to compute the convective tendencies, i.e. the feedback of the subgrid vertical circulation onto the resolved flow. The downdrafts are assumed to originate at the level of free sinking. As an additional closure condition, the downdraft mass flux in this level is set proportional to the updraft mass flux at cloud base via a coefficient γ_d, which is a disposable parameter. In the present version of the scheme γ_d is set to a constant value of 0.3. In subsaturated regions below cloud base, the precipitation in the downdrafts may evaporate with a parameterized rate. Depending on the temperature of the lowest model layer, the precipitation is interpreted as convective snow or rain.

(b) Tiedtke-Bechtold scheme
Basic Namelist settings: `itype_conv=2`
This scheme is a modernization of the Tiedtke-scheme and is used nowadays in the ECMWF IFS model. (see Bechtold et al. (2001), Bechtold et al. (2008), Bechtold et al. (2014)).

(c) A scheme for shallow convection
Basic Namelist settings: `itype_conv=3`
This scheme has been extracted from the Tiedtke scheme and can be used for the *convection permitting* scales. It is applied for the COSMO-DE.

The parameterization scheme is numerically very expensive. Thus, a timestep number increment can be specified for which the convection scheme is called. The convective tendencies are then stored and remain fixed for the following time steps.

Fractional Cloud Cover
In the parameterization schemes for grid-scale clouds and precipitation the condensation rate for cloud water is based on saturation equilibrium with respect to water. Consequently, a grid element is either fully filled with clouds at water saturation where $q^c > 0$ (relative humidity = 100%) or it is cloud free at water subsaturation where $q^c = 0$ (relative humidity $<$ 100%). The area fraction of a grid element covered with grid-scale clouds is thus a bivalued parameter which is either 1 or 0.

However, with respect to the calculation of radiative transfer but also for weather interpretation in postprocessing routines, it is useful to define a fractional cloud cover also for those grid boxes where the relative humidity is less than 100% and no grid-scale cloud water exists. The calculation of the fractional cloud cover σ_c in each model layer is calculated based on a traditional scheme which has been used in the former operational hydrostatic models EM/DM. σ_c is determined by an empirical function depending on the relative humidity, the height of the model layer and the convective activity. In addition to the EM/DM scheme, the contribution of convection to σ_c is assumed to depend on the vertical extent of the convection cell by prescribing a heuristic function. Also, a check for temperature inversions at the convective cloud tops is done to take anvils by an increase of σ_c in case of inversions into account.

3.5.4 Vertical Turbulent Diffusion

Basic Namelist settings: `lphys=.TRUE.; ltur=.TRUE.`

For vertical turbulent diffusion, several schemes are available:

a) 1-D diagnostic closure: has not been ported to blocked data structure!
Basic Namelist settings: `itype_turb=1; l3dturb=.FALSE.`
In the original EM/DM scheme, the vertical diffusion due to turbulent transport in the atmosphere is parameterized by a second-order closure scheme at hierarchy level 2.0 (Mellor and Yamada (1974); Müller (1981)). This results in a diagnostic closure where the turbulent diffusion coefficients are calculated in terms of the stability of the thermal stratification and the vertical wind shear. The impact of subgrid-scale effects on the heat and moisture fluxes due to condensation and evaporation of cloud water is not taken into account.

b) 1-D TKE based diagnostic closure:
Basic Namelist settings: `itype_turb=3; l3dturb=.FALSE.`
For the COSMO-Model, a new scheme has been developed, which is based on a prognostic equation for turbulent kinetic energy (TKE), that is a level 2.5 closure scheme. The new scheme includes the transition of turbulence which contributes mainly to the fluxes (diffusive turbulence) to very small scale (dissipative) turbulence by the action of small scale roughness elements, and the handling of non-local vertical diffusion due to the boundary layer scale turbulence. Most important seems to be the introduction of a parameterization of the pressure transport term in the TKE-equation, that accounts for TKE-production by subgrid thermal circulations. The whole scheme is formulated in conservative thermodynamic variables together with a statistical cloud scheme according to Sommeria and Deardorff (1977) in order to consider subgrid-scale condensation effects.

c) 3-D closure:
Basic Namelist settings: `itype_turb=5/7; l3dturb=.TRUE.`
The parameterization of subgrid-scale turbulent processes, also called a subgrid-scale (SGS) model, is of particular meaning for highly resolved LES-like model simulations. For resolutions reaching to the kilometer-scale, a more adequate turbulence parameterization scheme should be used. For both versions described above, there is the possibility to use a 3-D closure scheme. Up to now, this has been implemented into the COSMO-Model only for testing purposes.

COSMO-EU and COSMO-DE use the 1-D TKE based closure scheme.

3.5.5 Parameterization of Surface Fluxes

Basic Namelist settings: `lphys=.TRUE.; ltur=.TRUE.`

Mesoscale numerical modelling is often very sensitive to surface fluxes of momentum, heat and moisture. These fluxes provide a coupling between the atmospheric part of the model and the soil model. For both closure schemes described in Sec. 3.5.4, a special surface layer scheme can be applied.

a) A bulk-transfer scheme: has not been ported to blocked data structure!
Basic Namelist settings: `itype_tran=1`
For the 1-D diagnostic turbulence scheme, a stability and roughness-length dependent surface flux formulation based on Louis (1979) is implemented.

b) A TKE-based surface transfer scheme:
Basic Namelist settings: `itype_tran=2`
In context with the TKE-scheme, a revised and consistent formulation for the transport through the surface layer should be used. This surface scheme extends the TKE-equation to the constant flux layer and introduces an additional laminar layer just above the surface. This makes it possible to discriminate between the values of the model variables at the rigid surface (e.g. radiative surface temperatures) and values at the roughness height z_0 (lower boundary of the turbulent atmosphere). The Charnock formula to estimate the surface fluxes over sea is also reformulated using TKE.

COSMO-EU and COSMO-DE use the TKE based surface transfer scheme.

3.5.6 A subgrid-scale orography scheme

Basic Namelist settings: `lphys=.TRUE.; lsso=.TRUE.`

On larger domains it turned out that there is too little surface drag, causing an underestimation of the cross-isobar flow in the planetary boundary layer. To increase the surface drag can be accomplished by including subgrid scale orographic (SSO) effects. The SSO scheme by Lott and Miller (1997) was selected for the COSMO-Model. Its implementation in the COSMO-Model is described by Schulz (2008). It is also included e. g. in the global models at ECMWF or DWD and works here well.

3.5.7 Soil Processes

Basic Namelist settings: `lphys=.TRUE.; lsoil=.TRUE.`

The calculation of the surface fluxes requires the knowledge of the temperature and the specific humidity at the ground. The task of the soil model is to predict these quantities by the simultaneous solution of a separate set of equations which describes various thermal and hydrological processes within the soil. If vegetation is considered explicitly, additional exchange processes between plants, ground and air have to be taken into account.

a) The (multi-layer) soil model TERRA:
For land surfaces, the soil model TERRA provides the surface temperature and the specific humidity at the ground. The ground temperature is calculated by a direct solution of the heat conduction equation and the soil water content is predicted by the Richards equation. In the multi-layer version also the effect of freezing/melting of soil water/ice is included and a time dependent snow albedo is introduced. Evaporation from bare land surfaces as well as transpiration by plants are derived as functions of the water content, and - only for transpiration - of radiation and ambient temperature.

   Most parameters of the soil model (heat capacity, water storage capacity, etc.) strongly depend on soil texture. Five different types are distinguished: sand, sandy loam, loam, loamy clay and clay. Three special soil types are considered additionally: ice, rock and peat. Hydrological processes in the ground are not considered for ice and rock. Potential evaporation, however, is assumed to occur over ice, where the soil water content remains unchanged.

   The multi-layer concept avoids the dependence of layer thicknesses on the soil type. Additionally it avoids the use of different layer structures for the thermal and the hydrological sections of the model.

   Note: In COSMO-Version 5.05 the former two-/three-layer model has been eliminated.

b) The lake model FLake:
Basic Namelist settings: `lsoil=.TRUE.; llake=.TRUE.`
FLake (Fresh-water Lake), is a lake model (parameterisation scheme) capable of predicting the surface temperature in lakes of various depth on the time scales from a few hours to many years (see http://lakemodel.net for references and other information about FLake). It is based on a two-layer parametric representation (assumed shape) of the evolving temperature profile and on the integral budgets of heat and kinetic energy for the layers in question. The same concept is used to describe the temperature structure of the ice cover. An entrainment equation is used to compute the depth of a convectively-mixed layer, and a relaxation-type equation is used to compute the wind-mixed layer depth in stable and neutral stratification. Both mixing regimes are treated with due regard for the volumetric character of solar radiation heating. Simple thermodynamic arguments are invoked to develop the evolution equation for the ice thickness. The result is a computationally efficient bulk model that incorporates much of the essential physics. Importantly, FLake does not require re-tuning, i.e. empirical constants and parameters of FLake should not be re-evaluated when the model is applied to a particular lake. There are, of course, lake-specific external parameters, such as depth to the bottom and optical properties of water, but these are not part of the model physics.

   Using the integral approach, the problem of solving partial differential equations (in depth and time) for the temperature and turbulence quantities is reduced to solving ordinary differential equations for the time-dependent quantities that specify the temperature profile. FLake carries the equations for the mean temperature of the water column, for the mixed-layer temperature and its depth, for the temperature at the lake bottom, and for the shape factor with respect to the temperature profile in the lake thermocline (a stably stratified layer between the bottom of the mixed layer and the lake bottom). In case the lake is cover by ice, additional equations are carried for the ice depth and for the ice-surface temperature. The lake-surface temperature, i.e. the quantity that communicates information between the lake and the atmosphere, is equal to either the mixed-layer temperature or, in case the lake in question is covered by ice, to the ice-surface temperature. In the present configuration (a recommended choice), the heat flux through the lake water-bottom sediment interface is set to zero and a layer of snow over the lake ice is not considered explicitly. The effect of snow above the ice is accounted for parametrically through changes in the surface albedo with respect to solar radiation. Optionally, the bottom-sediment module and the snow module can be switched on. Then, additional equations are carried for the snow-surface temperature (temperature at the air-snow interface), for the snow depth, for the temperature at the bottom of the upper layer of bottom sediments penetrated by the thermal wave, and for the depth of that layer. Surface fluxes of momentum and of sensible and latent heat are computed with the operational COSMO-model surface-layer parameterization scheme. Optionally, a new surface-layer scheme can be used that accounts for specific features of the surface air layer over lakes.

In order to be used within the COSMO model (or within any other NWP or climate model), FLake requires a number of two-dimensional external-parameter fields. These are, first of all, the fields of lake fraction (area fraction of a given numerical-model grid box covered by lake water that must be compatible with the land-sea mask used) and of lake depth. Other external parameters, e.g. optical characteristics of the lake water, are assigned their default values offered by FLake. Since no tile approach is used in the COSMO model, i.e. each COSMO-model grid box is characterised by a single land-cover type, only the grid boxes with the lake fraction in excess of 0.5 are treated as lakes. Each lake is characterised by its mean depth. Deep lakes are currently treated with the *false bottom*. That is, an artificial lake bottom is set at a depth of 50 m. The use of such expedient is justified since, strictly speaking, FLake is not suitable for deep lakes (because of the assumption that the thermocline extends down to the lake bottom). However, as the deep abyssal zones typically experience no appreciable temperature changes, using the false bottom produces satisfactory results. A Global Land Cover Characterization (GLCC) data set (`http://edcdaac.usgs.gov/glcc`) with 30 arc sec resolution, that is about 1 km at the equator, is used to generate the lake-fraction filed. The filed of lake depth is generated on the basis of a data set (developed at DWD) that contains mean depths of a number of European lakes and of major lakes of the other parts of the world. Notice that, unless tile approach is used to compute the surface fluxes, only the lake-depth external parameter filed is actually required to use FLake within the COSMO model. Setting the lake depth to its actual value for the COSO-model grid boxes with the lake fraction in excess of 0.5, and to a negative value, say $-1m$, otherwise, unambiguously specifies the grid-boxes for which the lake-surface temperature should be computed.

c) A sea-ice scheme:
Basic Namelist settings: `lsoil=.TRUE., lseaice=.TRUE.`
The presence of sea ice on the ocean's surface has a significant impact on the air-sea interactions. Compared to an open water surface the sea ice completely changes the surface characteristics in terms of albedo and roughness, and therefore substantially changes the surface radiative balance and the turbulent exchange of momentum, heat and moisture between air and sea. In order to deal with these processes the COSMO model includes a sea ice scheme (Mironov (2008)).

COSMO-EU and COSMO-DE use the multi-layer soil model and the FLake-Model. The sea-ice scheme is only used in COSMO-EU.

3.6 Data Assimilation

Basic Namelist setting: `luseobs=.TRUE.`

The requirements for the data assimilation system for the operational model are mainly determined by the very high resolution of the model and by the task to employ it also for very short-range forecasting. Hence, detailed high-resolution analyses of the atmosphere have to be able to be produced frequently, and this requires a thorough use of asynoptic and high-frequency observations such as aircraft data and remote sensing data. Note that the synoptic scales are largely determined by the lateral boundary conditions provided by the driving model, and the main purpose of the assimilation scheme is to analyze the meso scales.

By design, 3-dimensional analysis methods tend to be less appropriate for this purpose. They do not allow to account for the exact observation time of asynoptic data, and they make it necessary to neglect most of the high-frequent data unless the analysis scheme is applied very frequently at significant computational costs. Moreover, the geostrophic approximation, a key ingredient of some of these schemes as used e.g. for the GME, is of limited validity on the meso scale. Therefore, 4-dimensional methods offer potential advantages since they include the model dynamics in the assimilation process directly. However, the 4-dimensional variational (4DVAR) method has been too expensive in the past for operational application considering the small amount of time available to produce the analyses and forecasts.

a) Observation Nudging —— Basic Namelist setting: `lnudge=.TRUE.`

Therefore, a scheme based on the observation nudging technique has been developed to define the atmospheric fields. It is based on an experimental nudging assimilation scheme which had been developed for the former hydrostatic model DM and its Swiss version SM (Schraff (1996); Schraff (1997)) and which compared favorably with the at that time operational Optimum Interpolation analysis of DM in a number of test cases. The scheme for COSMO has then been adapted to the nonhydrostatic modelling framework and runs on distributed memory machines using domain decomposition.

Nudging or Newtonian relaxation consists of relaxing the prognostic model variables towards prescribed values within a predetermined time window (see e.g. Davies and Turner (1977); Stauffer and Seaman (1990)). In the present scheme, nudging is performed towards direct observations which is more appropriate for high-resolution applications than nudging towards 3-dimensional analyses (Stauffer and Seaman (1994)). A relaxation term is introduced into the model equations, and the tendency for a prognostic variable $\psi(\mathbf{x}, t)$ is given by

$$\frac{\partial}{\partial t}\psi(\mathbf{x}, t) \quad = \quad F(\boldsymbol{\psi}, \mathbf{x}, t) \quad + \quad G_\psi \cdot \sum_{k_{(obs)}} W_k \cdot [\psi_k - \psi(\mathbf{x}_k, t)] \tag{3.14}$$

F denotes the model dynamics and physical parameterizations, ψ_k the value of the k^{th} observation influencing the grid point $\mathbf{x}$ at time t, $\mathbf{x}_k$ the observation location, G_ψ the constant so-called nudging coefficient, and W_k an observation-dependent weight which usually varies between 0 and 1. Neglecting the dynamics and physics and assuming a single observation with a constant weight W_k equal 1, the model value at the observation location relaxes exponentially towards the observed value with an e-folding decay rate of $1/G_\psi$ corresponding to about half an hour. The observational information is provided to the nudging scheme in the form of NetCDF observation input files which are described in Section 5.3.

In practical applications, the nudging term usually remains smaller than the largest term of the dynamics so that the dynamic balance of the model is not strongly disturbed. The coupling between the mass and wind field innovations is primarily induced implicitly by the model dynamics. If the assimilation process is successful the model fields will be close to dynamic balance at the beginning of the forecast, and an initialization step is not required.

b) Latent Heat Nudging —— Basic Namelist setting: `llhn=.TRUE.`

Radar-derived precipitation rates can be assimilated by an extra Latent Heat Nudging scheme (Stephan et al. (2008)). It computes additional temperature and humidity increments at each model column independently from each other. It is tuned and should be used only for convection-permitting model configurations (with horizontal mesh widths of ≤ 3 km). The observation input is gridded precipitation rates read in the form of extra Grib files. Further Grib files can be read optionally, containing a blacklist, and radar beam height maps utilised for bright band detection.

c) Analysis of surface and soil fields, outside COSMO code

In addition to the nudging-type assimilation schemes for the atmosphere, a set of 2-dimensional intermittent analysis schemes can be applied for some of the surface and soil fields in a full data assimilation cycle for the COSMO model. This comprises of a variational soil moisture analysis (Hess (2001)), a snow analysis, and a sea surface temperature (SST) analysis (including an analysis of sea ice cover). These analysis schemes are not integrated into the COSMO code, but are programs on their own. Therefore, they are not covered by this User's Guide, even though a scientific description or outline of them is included in the COSMO Documentation Part III on data assimilation.

d) Diagnostic surface analyses —— Basic Namelist setting: `lsurfa=.TRUE.`

In contrast, the data assimilation code of the COSMO model includes a module with a Cressman-type successive correction analysis scheme, which can be used to compute a set of other 2-dimensional surface-level analyses. This set comprises of a 2-m temperature, 2-m relative humidity, 10-m wind speed, and surface precipitation analysis. While the precipitation analysis is purely based on rain gauge (surface synoptic) data, the other analyses use the corresponding model field as a first guess purely to help defining the small-scale details that are not resolved by the surface observations. All these analyses are used only for diagnostic purposes. As an exception, the daytime 2-m temperature and optionally 2-m humidity analyses are used in the variational soil moisture analysis.

Note, that Nudging is no longer used at DWD for the regional data assimilation system, but an ensemble-based system is used now. This system, however, is not part of the COSMO-Model source code.

Section 4

Installation of the COSMO-Model

This chapter explains the steps necessary to compile and run the model. Section 4.1 lists the external libraries that are necessary to run different components of the model and what can be done, if these libraries are not available. The next sections give detailed informations on how to prepare, compile, link and run the COSMO-Model.

4.1 External Libraries for the COSMO-Model

For some components, the COSMO-Model uses external libraries. Usage of most of these libraries can be controlled by conditional compilation. To handle this, the C preprocessor (cpp) must be called. Most Fortran compilers activate the C preprocessor for files ending with a capital `F` in the suffix: `.F` or `.F90`. The COSMO-Model does not use capital letters in the suffix, therefore a special compiler option has to be set, to activate this preprocessor. Take a look to the manual of your compiler to find out about this option.

4.1.1 `libgrib1.a`:

The original implementation of INT2LM and the COSMO-Model used GRIB (Grided Binary), Version 1, as standard format. Coding and decoding of GRIB1 records can be done with the DWD GRIB1 library, the `libgrib1.a`. This library also contains C-routines to write data to and read it from disk. The Grib library is available from DWD and is provided together with the source code for the COSMO-Model. A short guide for the installation is included in the tar-file of the Grib library.

With GRIB1, DWD used a Grib file format, where all records are starting and ending with additional bytes, the so-called *controlwords*. To process these controlwords properly, you have to set the environment variable

```
export LIBDWD_FORCE_CONTROLWORDS=1
```

Usage of the DWD GRIB1 library can be controlled by conditional compilation and setting the macro `GRIBDWD`. If this macro is not set during compilation, the parts of the source code that do use `libgrib1.a` calls are not compiled and the library will not be linked to the binary.

4.1.2 `libgrib_api.a, libgrib_api_f90.a` or `libeccodes.a, libeccodes_f90.a:`

Since Version 4.28 another grib library can be used to read grib data. This is the `grib_api` (Application Programmer's Interface) from ECMWF. With this library it is possible to read and write also GRIB2 data (i.e. GRIB, Version 2). Since some time, `grib_api` is inclued in the `eccodes` framework, which also contains a `bufr`-library. From Version 5.05 on the COSMO-Model can also work with `eccodes` (INT2LM from Version 2.05 on). Source codes for both libraries together with installation instructions are available from the ECMWF web pages:
`http://www.ecmwf.int/en/computing/software`.

All following explanations are now for `eccodes`, but the same holds also for `grib_api`.

To use `eccodes` properly, two sets of definition files are necessary for running the programs: one contains all original definitions from ECMWF (`definitions`), the other set is compiled and maintained by DWD and must be used for the INT2LM and the COSMO-Model (`definitions.edzw`). Also, some special sample files are needed by both programs. Both sets of definition files and the necessary samples, which have to correspond to the `eccodes` version used, are distributed together with the INT2LM by DWD.

Please read more on `eccodes` and its usage in the Documentation, Part VI, *Model Output and Data Formats for I/O*.

Usage of the `eccodes` library can be controlled by conditional compilation and setting the macro `GRIBAPI`. If this macro is not set during compilation, the parts of the source code that do use `eccodes` calls are not compiled and the library will not be linked to the binary.

4.1.3 `libnetcdf.a:`

Since Version 3.18, input and output of data can also be done in the NetCDF format (Network Common Data Format). Using NetCDF requires an external library `libnetcdf.a`. The source code of this library can be downloaded from `http://www.unidata.ucar.edu` .

Usage of the NetCDF library can be controlled by conditional compilation and setting the macro `NETCDF`. If this macro is not set during compilation, the parts of the source code that do use NetCDF calls are not compiled and the library will not be linked to the binary.

NOTE:
The usage of NetCDF can only be avoided, if also the Nudging is switched off. If Nudging (the data assimilation) shall be used, a NetCDF library has to be available, because the observation processing is done via NetCDF.

4.1.4 `libmisc.a`:

Before Version 4.24 of the COSMO-Model, a library `libmisc.a` was needed for special applications in the data assimilation (the nudging). Since version 4.24 this library is not necessary any more.

4.1.5 `libcsobank.a`, `libsupplement.a`:

Before COSMO-Model 4.25 and INT2LM 2.1

The COSMO-Model and INT2LM use a tool for parallel asynchronous I/O from or to files or a data base system (only for Grib). The routines for that tool are grouped together in a module `mpe_io.f90`. In the VCS of DWD, `mpe_io.f90` is provided as an external module, hence it is not in the source code of the model library. `mpe_io.f90` uses the two libraries `libcsobank.a` and `libsupplement.a`.

For users outside DWD, `mpe_io.f90` has been included in the source code of the COSMO-Model and also in the INT2LM. To satisfy the calls from `mpe_io` to the data base system, an additional file `dummy_db.f90` is provided.

NEW:
Since COSMO-Model Version 4.25 and INT2LM Version 2.1, `mpe_io.f90` has been replaced by a modified version `mpe_io2.f90`, which does not support the database access any more. Therefore the usage of these external libraries is now obsolete.

4.1.6 `libRTTOVxx.a`:

Since Version 3.7 the COSMO-Model contains an interface to the RTTOV7-library (Radiative Transfer Model). This interface has been developed at the DLR Institute for Atmospheric Physics in Oberpfaffenhofen. Together with the RTTOV7-library it is possible to compute *synthetic* satellite images (brightness temperatures and radiances) derived from model variables for Meteosat5-7 and Meteosat Second Generation.

In subsequent versions also newer RTTOV libraries can be used:

- Between Versions 4.18 and 5.04e: RTTOV9: RTTOV9 has been eliminated now
- Since Version 4.26: RTTOV10: This library is used at DWD
- Since Version 5.05: RTTOV12: There still is no experience at DWD using COSMO with RTTOV12

The RTTOV model has been developed by UKMO et.al. in the framework of the ESA NWP-SAF. To use any version of the RTTOV model, a license is necessary. For getting this license, please contact `nwpsaf@metoffice.gov.uk`.

Usage of the RTTOV libraries can be controlled by conditional compilation and setting one of the macros `RTTOVxx`, with $\mathtt{x} \in \{7, 10, 12\}$. Note that RTTOV7 has been modified at DWD to be used in parallel programs. For the usage of RTTOV10 and RTTOV12, a special interface mo_rttov_ifc.f90 is necessary, which can also be obtained from DWD together with installation instructions.

If the license (and hence the RTTOV libraries) is not available, the corresponding macros must not be set. The computation of the synthetic satellite images is not possible then.

4.2 Preparing the Code

You have got a tar-file cosmo_yymmdd_x.y, where yymmdd describes the date in the form "Year-Month-Day" and x.y gives the version number. By de-taring, a directory is created with the following contents:

DOCS	Contains a short documentation of the changes in version x.
edid	Script to edit files in src and store them in work.
Fopts	Definition of the compiler options and also directories of libraries.
LOCAL	Contains several examples of Fopts-files for different computers.
Makefile	For compiling and linking the programs.
RUNSCRIPTS	Several examples of scripts (run_cosmo_xx or run_ideal_xx) to set the Namelist values for a configuration xx or for idealized test cases and start the program.
src	Subdirectory for the source code.
obj	Subdirectory where the object files are written.
ObjDependencies	Definition of the dependencies between the different source files.
Objfiles	Definition of the object files.
work	Subdirectory for intermediate files.

The directories ./obj and ./work are empty and can therefore get lost by the tar-process. If so, you have to create them again. In edid you have to adapt the pathnames if you want to work with it.

4.3 Compiling and Linking

You have to choose the options for compiling the code in Fopts. See the User Guide of your computer system for necessary and/or desired options. In particular you have to set the macros to select external libraries. Since COSMO-Model Version 5.1 you definitely have to set the macro -D__COSMO__! Before linking check that all necessary external libraries (see 4.1) are available.

Before compiling and linking the program you have to decide whether you want to run in single or in double precision. If you want to run in single precision, you have to define the macro -DSINGLEPRECSION in the Fopts-file. If you do not specify this macro, the default is to run the COSMO-Model in double precision.

NOTE:
It is not recommended to run the data assimilation in single precision!

All other input variables for the program can be determined before running the program with the NAMELIST-input (see Chapter 6).

The COSMO-Model is parallelized for distributed memory parallel computers using the domain decomposition technique and explicit message passing with the Message Passing Interface (MPI). Thus it can run on parallel platforms but also on sequential platforms where MPI is not available. For this purpose an additional module dummy_mpi.f90, together with a file mpif.h, are provided in the directory LOCAL. Place these files in the src-directory before compiling and linking. Then you can only create sequential binaries (see below).

You can invoke a make-run by typing make *entry*. The following *entries* are available. Note that the special *entry* has to correspond to the settings of the macros. seq or par correspond to the sequential or parallel mode, resp.

pur[par|seq] A pure binary without nudging and synthetic satellite images. -DNUDGING, -DRTTOVx and -DNETCDF must not be set.

all[par|seq] A full binary with nudging and synthetic satellite images, if -DNUDGING, -DNETCDF and -DRTTOVx are set.

nud[par|seq] A binary with nudging but without synthetic satellite images. -DNUDGING and -DNETCDF must be set and -DRTTOVx must not be set.

sat[par|seq] A binary without nudging but with synthetic satellite images. -DNUDGING must not be set and -DRTTOVx must be set.

More *entries* can be added on your own.

4.4 Running the Code

To run the code, several ASCII-files INPUT_*** have to be provided that contain values for the NAMELIST variables. The form of these INPUT_*** files is described in Chapters 5 and 6. They are created by the provided run-scripts.

The run-scripts also contain the command to invoke the binary, which you have to adapt. While a sequential binary usually can be invoked just by typing <name_of_binary>, a parallel binary must be invoked by a special tool. Often this tool is mpirun, but also other tools are possible. See the manual for your system on how to invoke the binary created in the last step.

Section 5

Input Files for the COSMO-Model

The COSMO-Model requires several input files.

- ASCII-files, called `INPUT_***` (see below for the exact filenames), that contain the namelist variables. The form of these files is described in Section 5.1. The namelist groups, the variables, their meanings and possible values are described in Chapter 6 for the model and in Chapter **??** for the data assimilation.

- GRIB or NetCDF files for the initial and boundary values. These files and their contents are explained in detail in Part V of the COSMO Documentation, the INT2LM User Guide (Preprocessing: Initial and Boundary Data for the COSMO-Model). Therefore we only give a short summary in 5.2. A description of the formats used (GRIB, NetCDF) can be found in Part VI of the COSMO Documentation (Model Output and Data Formats for I/O).

- NetCDF observation input files which contain the observational information for data assimilation (nudging) and for producing a NetCDF feedobs file. The purpose of the feedobs file is to serve as input to a LETKF analysis scheme or to verification tools. The NetCDF observation input files, including a blacklist file, are described in Section 5.3.

5.1 Files for Namelist Input

The COSMO-Model uses `NAMELIST`-input to specify runtime parameters. The parameters are splitted into several groups, which are distributed to the components. Table 5.1 lists the components, the groups and the corresponding `INPUT_***` files.

The last group of component Input / Output (`GRIBOUT`) can occur several times. Every group can determine a different list of variables for output and also different output steps.

The program provides default values for all parameters. To change a default value, an appropriate `NAMELIST` statement has to appear in the corresponding ASCII-file `INPUT_***`. The form of a `NAMELIST` statement depends on the specific platform you are using but is always similar to the following (refer to the Language Reference Manual of your system):

Table 5.1: NAMELIST-groups and INPUT_*** files

Component	Description	Group	INPUT file
Setup	specifying the domain and the size of the grid	/LMGRID/	INPUT_ORG
	parameters for the model run	/RUNCTL/	
	parameters for tuning variables	/TUNING/	
Dynamics	parameters for the adiabatic model	/DYNCTL/	INPUT_DYN
Physics	parameters for the diabatic model	/PHYCTL/	INPUT_PHY
Diagnostics	parameters for the diagnostic calculations	/DIACTL/	INPUT_DIA
	parameters for the satellite images	/SATCTL/	INPUT_SAT
Additionals	parameters for the initialization	/INICTL/	INPUT_INI
	parameters for controlling the EPS mode	/EPSCTL/	INPUT_EPS
Input /	controlling the I/O	/IOCTL/	INPUT_IO
Output	parameters for using DWD's database system	/DATABASE/	
	controlling the grib input	/GRIBIN/	
	controlling the grib output	/GRIBOUT/	
Assimilation	controlling the data assimilation	/NUDGING/	INPUT_ASS
	(optional extra groups for special operators)		

1. The ampersand (&) character, followed immediately by the name of the namelist group.
2. A sequence of zero or more

   `parameter=value,`

   statements.
3. / to terminate the NAMELIST group.

Example:
In the following example new values are set for the parameters in the Namelist group lmgrid:

```
&lmgrid
  startlon_tot=-10.4,    startlat_tot=-3.025,
  pollat=32.5,           pollon=-170.0,
  dlon=0.025,            dlat=0.025,
  ie_tot=72,             je_tot=92,
/
```

For a complete reference of all NAMELIST parameters see Chapters 6 (for the model) and ?? (for the data assimilation).

5.2 Initial and Boundary Data

To start a forecast, the files containing the initial and boundary data have to provide all meteorological fields necessary for running the model. These files are

- Initial data file, e.g. `laf2019070912`
- Boundary data files, e.g. `lbff00000000`, `lbff00010000`, etc.

For exact names of the files see Section 6.6 in the INT2LM User Guide (Preprocessing: Initial and Boundary Data for the COSMO-Model): *Conventions for File Names*.

When reading these files, the COSMO-MOdel checks whether all fields are present, otherwise the run will be aborted. Which fields are needed, depends on the configuration chosen, i.e. on the settings of special namelist variables.

For a complete list of all possible initial and boundary data please refer to Chap. 5 of the INT2LM User Guide.

5.3 Observation Input Files

This section describes the format of the **NetCDF file observation input** to the COSMO model and what has to be done, if these files are used as input to the nudging-type assimilation scheme inside COSMO, or to writing (NetCDF or ASCII (i. e. YUVERIF)) feedobs files for verification purposes, or to perform surface analyses based on observations (e. g. of 2-m temperature). The observation input described here currently relates to conventional observations only, and it does not describe the (GRIB) input for the latent heat nudging.

Please note that the former alternative to read observations from the '**AOF**' files has been eliminated in COSMO Version 5.04.

In the following items, general properties of the **NetCDF file observation input** are described:

- **Required files and file names:**

  The NetCDF observation input files have fixed file names which are given in the following sub-sections and begin with 'cdfin_'. For the observation file type 'SYNOP' for instance, the file name would be 'cdfin_synop'. For each observation file type, it is possible to have several input files, with suffix '.2', '.3', etc. for the file names (the suffix '.1' is not used). An additional suffix '.nc' is optional. Thus, for a second file for SYNOP data, the file name 'cdfin_synop.2.nc' would be possible.

  The existence of any of the NetCDF observation input files is optional. If there are no observations available of a certain type then the corresponding NetCDF file should be missing in the input directory. Any file with a corresponding file name must have the correct format (as described further below), and files with zero length are not allowed.

  In addition, a blacklist file with fixed name 'blklsttmp' containing a blacklist and a whitelist is read by COSMO. The existence of this file is mandatory (if itype_obfile = 2 and any of the namelist parameters 'lnudge', 'lverif', or 'lsurfa' is true).

  All these input files must reside in the directory given by the namelist parameter 'ycdfdir'.

- **Content of the NetCDF observation input files:**

  The NetCDF observation input files 'cdfin_'* are usually created by direct conversion of BUFR files (using the 'bufrx2netcdf' program). This means that they contain the same variables as the input BUFR files. Hence, if possible, all BUFR reports in a file should use exactly the same template in order to allow for a complete conversion into NetCDF. 'bufrx2netcdf' will only convert those observation reports in a BUFR file which have exactly the same template as the first report converted, and it will neglect the other reports. Therefore, if the BUFR reports do not have an identical template, several calls of 'bufrx2netcdf' (using the '-x' option to skip the previous reports with other templates) to a BUFR file are necessary to create several NetCDF files for the same observation file type. A BUFR file without any observations will result in a file 'cdfin_'* with zero length. Such a file must be deleted before starting running COSMO.

  COSMO expects the NetCDF files to contain specific variables. Some of the variables are mandatory while others are optional. This implies that the BUFR file prior to conversion into NetCDF must also contain these variables (at least the mandatory ones). Any BUFR files lacking these variables or containing the quantities in a different form with e.g. different variable names must first be converted into a BUFR file according to the specifications below prior to conversion into NetCDF.

The descriptions of the templates below applies to the NetCDF input files. However, the templates, in particular of BUFR Section 3, are based upon those of the BUFR files. Therefore, reference is made to descriptions of BUFR templates where available by WMO, and additional common sequence BUFR descriptors are included in the description even though they are absent in the NetCDF file. The specified variable types (int / float / char) relate only to the NetCDF file after conversion from BUFR.

- From **BUFR Section 1**, the following elements are **mandatory** for all observation types (i.e. COSMO will abort, if any of these are absent in the NetCDF file):

descriptor	type	variable name (NetCDF file)	meaning
	int	edition_number	BUFR edition number (usually = 4)
C-11	int	section1_centre	originating / generating data centre
C-12	int	section1_subcentre	e.g. processing centre for GPS reports
	int	section1_update_sequence_nr	upd. seq. number (station correction)
C-13	int	section1_data_category	WMO data category
C-13	int	section1_int_data_sub_category	internatl. (WMO) data sub category
	int	section1_date	year-month-day [YYYYMMDD]
	int	section1_time	hour-minute-second [HHMMSS]

The identifiers C-* denote BUFR-CREX Common Code Tables which are detailed in: http://www.wmo.int/pages/prog/www/WMOCodes/TDCFtables.html#TDCFtables, link 'Common Code Tables to Binary and Alphanumeric Codes'.

Observational reports which have missing values for 'section1_data_category' or for 'section1_int_data_sub_category' will be rejected. This also applies to GPS reports if 'section1_centre' or 'section1_subcentre' have missing values.

- From **BUFR Section 2**, the following elements are **optional**. This means that they are read and stored internally for all observation types, but it is not mandatory that they exist or contain non-missing values (i. e. COSMO will not abort, if they do not exist):

descriptor	type	variable name (in NetCDF file)	meaning
	int	section2_ikz	DWD-internal data base ID
	int	section2_decoding_date	year-month-day [YYYYMMDD]
	int	section2_decoding_time	hour-minute-second [HHMMSS]

- From **BUFR Section 3**, the variables read and used mandatorily or optionally are detailed in the following sub-sections for the different observation file types. The variable names in the NetCDF files equal the BUFR mnemonics for these variables.

  The 'ifxy' attribute for any variable in the NetCDF file is equal to the BUFR descriptor for that variable. These BUFR descriptors are described in http://www.wmo.int/pages/prog/www/WMOCodes/TDCFtables.html#TDCFtables, link 'BUFR/CREX Table B - Classification of elements' , and on the same page, the link 'Code and Flag Tables associated with BUFR/CREX Table B' provides the details of the code tables.

Observational reports are rejected unless they contain appropriate non-missing values for observation time (year, month, day, hour, minute), location (latitude, longitude), and for some observation types station altitude and some form of station identity.

Note that for some observation file types, the (BUFR and the) NetCDF files produced at DWD contain an additional variable for each (or most) of the existing variables of section 3. This variable contains a quality flag related to the value of the original variable. The variable name of this quality flag variable is equal to the name of the original variable plus a suffix 'Q'. However, all the quality flag variables of this type are obsolete, i.e. they are never needed, never used or read by COSMO, they always contain missing values only, and therefore they are never described hereafter.

In the following sub-sections of this section, the sequences of variables (templates) related to BUFR Section 3 are detailed for the different observation types. In the last sub-section, the blacklist file is described.

5.3.1 Templates for observation types for which Table-Driven Code Forms (TDCF) defined by WMO exist

The BUFR templates, which the NetCDF files described in this sub-section are based on, are described at
http://www.wmo.int/pages/prog/www/WMOCodes/TemplateExamples.html.

- **SYNOP, SYNOP MOBIL, and SHIP**

File names:				
File names:	for fixed land stations	(SYNOP)	:	'cdfin_synop'
	for mobile land stations	(SYNOP MOBIL)	:	'cdfin_synop_mob'
	for sea stations	(SHIP)	:	'cdfin_ship'.

The templates for observation type SYNOP and for SYNOP MOBIL follow the WMO common sequence descriptors '**TM 3 07 080**' resp. '**TM 3 07 090**'. These sequences, and also the use of the variables in COSMO, are identical to each other except for the report header (common sequences '3 01 090' esp. '3 01 092'). The template for SHIP follows the WMO common sequence descriptor '**TM 3 08 009**'.

Caution: The use of observation file type 'SYNOP MOBIL' in COSMO has not been tested yet due to a lack of testing opportunity because of a lack of data of this type. Its use is at the user's own risk. However, as the template is almost the same as for fixed land SYNOP, errors are unlikely. If errors occur, they should be reported to the author.

For all the 3 observation (file) types, the table below lists all the variables which are used by COSMO, plus some of the other variables, but it does not detail the variables for those common sequences which contain only variables that are not used by COSMO. For convenience, the table is split into a report header and three report body parts.

The use of the variables (first column of table) is defined as follows:

'need': COSMO asks stringently for this variable and will abort if variable is absent (but will not abort if values are equal to 'missing value')

'opt': variable exists and is read/used (but COSMO will not abort if it does not exist); 'used' means here that it is e. g. written to the feedobs file, but it does not imply active use in the data assimilation

'+': variable exists but is not read by COSMO

'−': descriptor (a BUFR common sequence, or a BUFR data description operator) exists only in the BUFR file, not in the NetCDF file

'(−)': descriptor itself exists only in the BUFR file, not in the NetCDF file; however the descriptor indicates a common sequence of variables, which *are* present in the NetCDF file, but which are not used by COSMO and hence are not detailed here

The existence (2^{nd} column 'L' for land stations, 3^{rd} column 'S' for SHIP sea stations) of the variables or descriptors for the different observation (file) types according to the above mentioned BUFR common sequence descriptors is defined as follows:

'+' : exists (for both types of land stations resp. for sea stations)

'f' : exists for fixed land stations, but not for mobile land stations

'm' : exists for mobile land stations, but not for fixed land stations

' ' : does not exist

use	L	S	WMO descriptor	type	mnemonics	meaning
−	f		3 01 090			Fixed surface station ID, time, horiz. + vertical coordinates
−	m		3 01 092			Mobile surface station ID, time, horiz. + vertical coordinates
−		+	3 01 093			Ship ID, movement, date / time, horiz. + vertical coordinates
−		+	3 01 036			Ship ID, movement, time, lat/lon
need	m	+	0 01 011	char*9	YDDDD	Ship or mobile land sta. identifier
+		+	0 01 012	int	MDS	Platform motion direction
+		+	0 01 013	int	NVS	Platform motion speed
+	m		0 01 003	int	MA	WMO region number
−	f		3 01 004			Surface station identification
need	f		0 01 001	int	MII	WMO block number
need	f		0 01 002	int	NIII	WMO station number
opt	f		0 01 015	char*20	YSOSN	Station or site name
need	+	+	0 02 001	int	NIX	Type of station
−	+		3 01 011			Year, month and day
need	+	+	0 04 001	int	MJJJ	Year
need	+	+	0 04 002	int	MMM	Month
need	+	+	0 04 003	int	MYY	Day
−	+		3 01 012			Hour, minute
need	+	+	0 04 004	int	MGG	Hour
need	+	+	0 04 005	int	NGG	Minute
−	+		3 01 021			Latitude and Longitude
need	+		0 05 001	float	MLAH	Latitude (high accuracy) [degree]
need	+		0 06 001	float	MLOH	Longitude (high accuracy) [deg.]
need		+	0 05 002	float	MLALA	Latitude (coarse accuracy) [deg.]

use	L	S	WMO descriptor	type	mnemonics	meaning
need		+	0 06 002	float	MLOLO	Longitude (coarse accur.) [deg.]
need	+	+	0 07 030	float	MHOSNN	Height of station above MSL (1)
need	+	+	0 07 031	float	MHOBNN	Height of barometer a. MSL (1)
opt	m		0 33 024	int	MSEQM	Station elevation quality mark
−	+		3 02 031			**Pressure** data
−	+	+	3 02 001			Pressure group
need	+	+	0 10 004	float	MPPP	Pressure
opt	+	+	0 10 051	float	MPPPP	Pressure reduced to MSL
opt	+	+	0 10 061	float	NPPP	3-hour pressure change
+	+	+	0 10 063	int	NA	Characteristic of press. tendency
+	+		0 10 062	float	NP24	24-hour pressure change
opt	+		0 07 004	float	MPN	Pressure (standard level)
opt	+		0 10 009	int	NHHHN	Geopotential height [gpm] of the standard level
−	+		3 02 035			Basic synoptic ’instanteous’ data
−		+	3 02 054			SHIP ’instanteous’ data
−	+		3 02 032			**Temperature + humidity** data
−		+	3 02 052			
opt	+	+	0 07 032	float	MHOSEN	Height of sensor above local ground / marine deck platform (for temp. + humidity meas.)
opt		+	0 07 033	float	MHAWAS	Height of sensor above water surface (temp. + hum. meas.)
need	+	+	0 12 101	float	MTDBT	Temperature / dry-bulb temperat.
+		+	0 02 039	int	MMOWTM	Method of wet-bulb temp. meas.
+		+	0 12 102	float	MTFNH	Wet-bulb temperature
need	+	+	0 12 103	float	MTDNH	Dew-point temperature
opt	+	+	0 13 003	int	MUUU	Relative humidity [%]
−	+		3 02 033			**Visibility** data
−		+	3 02 053			
+	+	+	0 07 032	float	MHOSEN0	Height of sensor a. ground (vis.)
opt		+	0 07 033	float	MHAWAS0	Height of sensor above water surface (for visibility)
opt	+	+	0 20 001	float	MVV	Horizontal visibility
−	+	+	3 02 034			**Precipitation** past 24 hours
+	+	+	0 07 032	float	MHOSEN1	Height of sensor a. gr. (precip.)
opt	+	+	0 13 023	float	MRR24	Total precipitation past 24 hours
−	+	+	3 02 004			**Cloud** data
opt	+	+	0 20 010	int	MN	Cloud cover (total) [%]
opt	+	+	0 08 002	int	MVTSU	Vertical significance
opt	+	+	0 20 011	int	MNH	Cloud amount (of low or middle clouds)
opt	+	+	0 20 013	float	NH	Cloud base height above surface
opt	+	+	0 20 012	int	MCC	Cloud type (low clouds C_L)
opt	+	+	0 20 012	int	MCC0	Cloud type (middle clouds C_M)
opt	+	+	0 20 012	int	MCC1	Cloud type (high clouds C_H)
opt	+	+	0 31 001	int	MDREP	Delayed descriptor replication
−	+	+	3 02 005			Individual cloud layers of masses
opt	+	+	0 08 002	int *n	MVTSU0	Vertical significance (2)

use	L	S	WMO descriptor	type	mnemonics	meaning
opt	+	+	0 20 011	int *n	MNH0	Cloud amount (2)
opt	+	+	0 20 012	int *n	MCC2	Cloud type (C) (2)
opt	+	+	0 20 013	float*n	NH0	Height of base of cloud (2)
(−)	+		3 02 036			Cloud with bases below station level
(−)	+		3 02 047			Direction of cloud drift
(−)	+		3 02 048			Direction and elevation of cloud
−	+		3 02 037			State of ground, snow depth, ground minimum temperature
opt	+		0 20 062	int	ME	State of ground (w. or w/o snow)
opt	+		0 13 013	float	NSSS	Total snow depth
+	+		0 12 113	float	MTGTGH	Ground min. temperat. past 12 hrs
(−)		+	3 02 055			Icing and ice
(−)		+	3 02 057			SHIP marine data
(−)		+	3 02 056			Sea surface temperature, depth
(−)		+	3 02 021			Waves
(−)		+	3 02 024			Wind waves
(−)		+	3 02 023			Swell waves
−	+		3 02 043			Basic synoptic ’period’ data
−		+	3 02 060			SHIP ’period’ data
−	+	+	3 02 038			**Present and past weather**
opt	+	+	0 20 003	int	NWW	Present weather
opt	+	+	0 04 024	int	MGGTP	Time period in hours
opt	+	+	0 20 004	int	MW1	Past weather 1
opt	+	+	0 20 005	int	MW2	Past weather 2
(−)	+		3 02 039			Sunshine data
−	+	+	3 02 040			**Precipitation** measurement
+	+	+	0 07 032	float	MHOSEN3	Height of sensor above ground / marine deck platform (precip.)
−	+	+	1 02 002			Replicate next 2 descript. twice
opt	+	+	0 04 024	int *2	MGGTP1	Time period in hours
opt	+	+	0 13 011	float*2	MRRR	Total precipitation / total water equivalent of snow
−	+		3 02 041			**Extreme temperature** data
−		+	3 02 058			
+	+	+	0 07 032	float	MHOSEN4	Height of sensor above ground / marine deck platf. (temper.)
opt		+	0 07 033	float	MHAWAS2	Height of sensor above water surface (for temperature)
opt	+		0 04 024	int	MGGTP2	Start of time period in hours
opt		+	0 04 024	int	MGGTP1	
opt	+		0 04 024	int	MGGTP3	End of time period in hours
opt		+	0 04 024	int	MGGTP2	
opt	+	+	0 12 111	float	MTXTXH	Maximum temperature over period specified
opt	+		0 04 024	int	MGGTP4	Start of time period in hours
opt		+	0 04 024	int	MGGTP3	
opt	+		0 04 024	int	MGGTP5	End of time period in hours
opt		+	0 04 024	int	MGGTP4	
opt	+	+	0 12 112	float	MTNTNH	Minimum temperature

use	L	S	WMO descriptor	type	mnemonics	meaning
						over period specified
−	+		3 02 042			**Wind** data
−		+	3 02 059			
+	+	+	0 07 032	float	MHOSEN5	Height of sensor a. ground/deck
opt		+	0 07 033	float	MHAWAS2	Height of sensor a. water surface
+	+	+	0 02 002	int	NIW	Type of instrumentation for wind measurement
opt	+	+	0 08 021	int	MTISI	Time significance (= 2 (time averaged))
opt	+	+	0 04 025	int	NGGTP	Time period (-10 min, or since a significant wind change)
need	+	+	0 11 001	int	NDNDN	Wind direction
need	+	+	0 11 002	float	NFNFN	Wind speed
+	+	+	0 08 021	int	MTISI0	Time significance (missing value)
−	+	+	1 03 002			Replicate next 3 descript. twice
opt	+	+	0 04 025	int *2	NGGTP0	Time period in minutes
+	+	+	0 11 043	int *2	NMWGD	Maximum wind gust direction
opt	+	+	0 11 041	float*2	NFXGU	Maximum wind gust speed
(−)	+		3 02 044			Evaporation data
(−)	+		3 02 045			Radiation data
(−)	+		3 02 046			Temperature change data

Table notes:

(1) Only one of the variables ’MHOBNN’ and ’MHOSNN’ is strictly needed to exist. ’MHOBNN’ is preferred to exist (and to be used if both variables exist and have non-missing values) because it should provide the precise height of the barometer for the pressure measurement, which is the observation with the most critical dependency on sensor (or ’station’) height.

(2) The use of ’*n’ in the variable type definition means that this variable has an additional dimension, i. e. several values may be present in one report. If the corresponding replication factors ’MEDRE’ or ’MDREP’ are zero for all reports in the NetCDF file, then the corresponding multi-dimensional variables do *not* need to exist (and probably will not exist) in the NetCDF file.

- **TEMP, TEMP MOBIL, and TEMP SHIP**

File names:	for fixed land stations	(TEMP)	:	’cdfin_temp’
	for mobile land stations	(TEMP MOBIL)	:	’cdfin_temp’
	for sea stations	(TEMP SHIP)	:	’cdfin_tempship’.

The template given by the common sequence descriptor ’**TM 3 09 052**’ is used for the observation types TEMP (fixed land stations), TEMP MOBIL (mobile land stations), as well as TEMP SHIP (sea stations). The table below lists all the variables of this template and their use in COSMO.

The use of the variables (first column of table) is defined as follows:

'need': COSMO asks stringently for this variable and will abort if variable is absent (but will not abort if values are equal to 'missing value')

'opt': variable exists and is read/used (but COSMO will not abort if it does not exist); 'used' means here that it is e. g. written to the feedobs file, but it does not imply active use in the data assimilation

'+': variable exists but is not read by COSMO

'–': descriptor (a BUFR common sequence, or a BUFR data description operator) exists only in the BUFR file, not in the NetCDF file

use	WMO descriptor	type	mnemonics	meaning
–	3 01 111			Identific. of launch site + instumentation
–	3 01 001			Station identification
need	0 01 001	int	MII	WMO block number (1)
need	0 01 002	int	NIII	WMO station number (1)
need	0 01 011	char*9	YDDDD	Ship or mobile land station identifier (1)
need	0 02 011	int	NRARA	Radiosonde type
opt	0 02 013	int	NSR	Solar and infrared radiation correction
opt	0 02 014	int	NSASA	Tracking technique / status of system
opt	0 02 003	int	NA4	Type of measuring equipment
–	3 01 113			Date / time of launch
opt	0 08 021	int	MTISI	Time significance (= 18 (launch time))
–	3 01 011			Year, month and day
need	0 04 001	int	MJJJ	Year
need	0 04 002	int	MMM	Month
need	0 04 003	int	MYY	Day
–	3 01 013			Hour, minute, second
need	0 04 004	int	MGG	Hour
need	0 04 005	int	NGG	Minute
+	0 04 006	int	MSEC	Second
–	3 01 114			Horiz. + vert. coord. of launch site
–	3 01 021			Latitude and Longitude
need	0 05 001	float	MLAH	Latitude (high accuracy) [degree]
need	0 06 001	float	MLOH	Longitude (high accuracy) [degree]
need	0 07 030	float	MHOSNN	Height of station above MSL (2)
need	0 07 031	float	MHOBNN	Height of barometer above MSL (2)
opt	0 07 007	int	MH	Height of release of sonde above MSL
opt	0 33 024	int	MSEQM	Station elevation quality mark
–	3 02 049			**Cloud** info reported with vert. soundings
opt	0 08 002	int	MVTSU	Vertical significance
opt	0 20 011	int	MNH	Cloud amount (of low or middle clouds)
opt	0 20 013	float	NH	Cloud base height above surface
opt	0 20 012	int	MCC	Cloud type (low clouds C_L)
opt	0 20 012	int	MCC0	Cloud type (middle clouds C_M)
opt	0 20 012	int	MCC1	Cloud type (high clouds C_H)
+	0 22 043	float	MTN00	Sea / water temperature
need	0 31 002	int	MEDRE	Extended delayed descriptor replicat. fac.
–	3 03 054			**Temperature, dewpoint** + **wind** data

use	WMO descriptor	type	mnemonics	meaning
				at a pressure level with sonde position
opt	0 04 086	int *n	NLTPD	Time displacement since launch time [s]
need	0 08 042	int *n	MEVSS	Extended vertical sounding significance
need	0 07 004	float*n	MPN	Pressure (vertical location) [Pa]
need	0 10 009	int *n	NHHHN	Geopotential height [gpm]
opt	0 05 015	float*n	MLADH	Latitude displacement since launch site
opt	0 06 015	float*n	MLODH	Longitude displacement since launch site
need	0 12 101	float*n	MTDBT	Temperature / dry-bulb temperature
need	0 12 103	float*n	MTDNH	Dew-point temperature
need	0 11 001	int *n	NDNDN	Wind direction [degree true]
need	0 11 002	float*n	NFNFN	Wind speed [m/s]
+	0 31 001	int	MDREP	Delayed descriptor replication factor
−	3 03 051			**Wind shear** data at a pressure level
+	0 04 086	int *n	NLTPD0	Time displacement since launch time [s]
+	0 08 042	int *n	MEVSS0	Extended vertical sounding significance
+	0 07 004	float*n	MPN0	Pressure (vertical location) [Pa]
+	0 05 015	float*n	MLADH0	Latitude displacement since launch site
+	0 06 015	float*n	MLODH0	Longitude displacement since launch site
+	0 11 061	float*n	NVBVB	Absolute wind shear in 1 km layer below
+	0 11 062	float*n	NVAVA	Absolute wind shear in 1 km layer above [m/s]

Table notes:

(1) Only either the pair of variables 'MII' and 'NIII' or the single variable 'YDDDD' is strictly needed to exist. If both exist and have non-missing values for a certain report, then the values of the pair 'MII' and 'NIII' are used.

(2) Only one of the variables 'MHOBNN' and 'MHOSNN' is strictly needed to exist. For radiosondes, 'MHOBNN' is found to be the pressure at the site where the ground check and calibration is done, whereas 'MHOSNN' coincides with the 'surface level' of the sounding. Therefore, for the three TEMP types and unlike for other observation types, 'MHOSNN' is preferred to exist (and be used if both variables have non-missing values).

(*n) '*n' in the variable type definition means that this variable has an additional dimension, used here for the vertical levels, and hence, several values may be present in one report. If the corresponding replication factors 'MEDRE' or 'MDREP' are zero for all reports in the NetCDF file, then the corresponding multi-dimensional variables do *not* need to exist (and probably will not exist) in the NetCDF file.

- **PILOT either with height or pressure as vertical coordinate**

File names:	for PILOT with height as vertical coordinate	:	'cdfin_pilot'
	for PILOT with pressure as vertical coordinate	:	'cdfin_pilot_p'.

Irrespective of the vertical coordinate used, the templates for PILOT (from fixed land stations), PILOT MOBIL (from mobile land stations), as well as PILOT SHIP (from sea stations) are identical. The common sequence descriptors '**TM 3 09 050**' and '**TM 3 09 051**' are used for PILOTs with pressure resp. height as vertical coordintate. These templates are identical to each other except for the vertical coordinate. ’**TM 3 09 050**’ contains pressure but lacks height as a variable, whereas for ’**TM 3 09 051**’, the variable pressure does not exist, and the vertical level is expressed as height.

COSMO is coded such that from the file ’cdfin_pilot_p’, in addition to the mandatory pressure variable, the height variable can also be read as an optional variable, and from ’cdfin_pilot’, in addtion to the mandatory height variable, the pressure variable is also read if present. This would allow for using PILOT reports which contain both pressure levels and height levels. *However, since the current version of ’bufrx2netcdf’ is not able to produce such ’mixed’ PILOT NetCDF files, this has never been tested.*

The table below, split into two parts, lists all the variables and their use in COSMO. The use of the variables (first column of the header table; columns ’Z’ and ’P’ of the body table for PILOT with height resp. pressure as vertical coordinate) is defined as follows:

’need’: COSMO asks stringently for this variable and will abort if variable is absent (but will not abort if values are equal to ’missing value’)

’opt’: variable exists and is read/used (but COSMO will not abort if it does not exist); ’used’ means here that it is e. g. written to the feedobs file, but it does not imply active use in the data assimilation

’(opt)’: variable does not exist in the template; if it does exist nevertheless then it is read and used by COSMO

’+’: variable exists but is not read by COSMO

’–’: descriptor (for a BUFR common sequence) exists only in the BUFR file, not in the NetCDF file

’ ’: variable does not exist at all

’Z’, ’P’	WMO descriptor	type	mnemonics	meaning
–	3 01 110			Identific. of launch site + instumentation
–	3 01 001			Station identification
need	0 01 001	int	MII	WMO block number (1)
need	0 01 002	int	NIII	WMO station number (1)
need	0 01 011	char*9	YDDDD	Ship or mobile land station identifier (1)
need	0 02 011	int	NRARA	Radiosonde type
opt	0 02 014	int	NSASA	Tracking technique / status of system
opt	0 02 003	int	NA4	Type of measuring equipment
–	3 01 113			Date / time of launch
opt	0 08 021	int	MTISI	Time significance (= 18 (launch time))
–	3 01 011			Year, month and day
need	0 04 001	int	MJJJ	Year
need	0 04 002	int	MMM	Month
need	0 04 003	int	MYY	Day
–	3 01 013			Hour, minute, second
need	0 04 004	int	MGG	Hour
need	0 04 005	int	NGG	Minute
+	0 04 006	int	MSEC	Second
–	3 01 114			Horiz. + vert. coord. of launch site
–	3 01 021			Latitude and Longitude
need	0 05 001	float	MLAH	Latitude (high accuracy) [degree]
need	0 06 001	float	MLOH	Longitude (high accuracy) [degree]

'Z', 'P'	WMO descriptor	type	mnemonics	meaning
need	0 07 030	float	MHOSNN	Height of station above MSL (2)
need	0 07 031	float	MHOBNN	Height of barometer above MSL (2)
opt	0 07 007	int	MH	Height of release of sonde above MSL
opt	0 33 024	int	MSEQM	Station elevation quality mark

'Z'	'P'	WMO descr.	type	mnemonics	meaning
need	need	0 31 002	int	MEDRE	Extended delayed descriptor replicat.
	–	3 03 050			**Wind** data at a pressure level
–		3 03 052			**Wind** data at a height level
opt	opt	0 04 086	int *n	NLTPD	Time displacem. since launch time [s]
need	need	0 08 042	int *n	MEVSS	Extend. vertic. sounding significance
(opt)	need	0 07 004	float*n	MPN	Pressure (vertical location) [Pa]
need	(opt)	0 07 009	int *n	NHHH	Geopotential height [gpm]
opt	opt	0 05 015	float*n	MLADH	Latitude displacem. since launch site
opt	opt	0 06 015	float*n	MLODH	Longitude displace. since launch site
need	need	0 11 001	int *n	NDNDN	Wind direction [degree true]
need	need	0 11 002	float*n	NFNFN	Wind speed [m/s]
+	+	0 31 001	int	MDREP	Delayed descriptor replication factor
	–	3 03 051			**Wind shear** data at a pressure level
–		3 03 053			**Wind shear** data at a height level
+	+	0 04 086	int *n	NLTPD0	Time displacem. since launch time [s]
+	+	0 08 042	int *n	MEVSS0	Extend. vertic. sounding significance
	+	0 07 004	float*n	MPN0	Pressure (vertical location) [Pa]
+		0 07 009	int *n	NHHH0	Geopotential height [gpm]
+	+	0 05 015	float*n	MLADH0	Latitude displacem. since launch site
+	+	0 06 015	float*n	MLODH0	Longitude displace. since launch site
+	+	0 11 061	float*n	NVBVB	Absolute shear in 1 km layer below
+	+	0 11 062	float*n	NVAVA	Absolute shear in 1 km layer above [m/s]

Table notes:

(1) Only either the pair of variables 'MII' and 'NIII' or the single variable 'YDDDD' is strictly needed to exist. If both exist and have non-missing values for a certain report, then the values of the pair 'MII' and 'NIII' are used.

(2) Only one of the variables 'MHOBNN' and 'MHOSNN' is strictly needed to exist. If both exist and have non-missing values for a certain report, then the values of 'MHOBNN' are used.

(*n) '*n' in the variable type definition means that this variable has an additional dimension, used here for the vertical levels, and hence, several values may be present in one report. If the corresponding replication factors 'MEDRE' or 'MDREP' are zero for all reports in the NetCDF file, then the corresponding multi-dimensional variables do *not* need to exist (and probably will not exist) in the NetCDF file.

5.3.2 Observation types with templates proposed or approved by WMO

The BUFR templates, which the NetCDF files described in this sub-section are based on, are described in (the lower part of)
http://www.wmo.int/pages/prog/www/WMOCodes/TemplateExamples.html
respectively for GPS zenith total delay and water vapour data at
http://egvap.dmi.dk/support/formats/egvap_bufr_v10.pdf .

- **BUOY**

  File name: 'cdfin_buoy'.

  The template follows the proposed WMO descriptor '**TM 3 08 008**'. The table below lists all the variables which are used by COSMO, plus only a subset of the variables that are not used by COSMO. For convenience, the table is split into a report header and a report body part.

  The use of the variables is defined as follows:

  'need': COSMO asks stringently for this variable and will abort if variable is absent (but will not abort if values are equal to 'missing value')

  'opt': variable exists and is read/used (but COSMO will not abort if it does not exist); 'used' means here that it is e. g. written to the feedobs file, but it does not imply active use in the data assimilation

  '+': variable exists but is not read by COSMO

  '−': descriptor (for a BUFR common sequence) exists only in the BUFR file, not in the NetCDF file

use	WMO descriptor	type	mnemonics	meaning
need	0 01 005	int	MABNN	Buoy / platform identifier
need	0 02 001	int	NIX	Type of station
+	0 02 036	int	NBOTY	Buoy type
opt	0 02 149	int	MTODB	Type of data buoy
−	3 01 011			Year, month and day
need	0 04 001	int	MJJJ	Year
need	0 04 002	int	MMM	Month
need	0 04 003	int	MYY	Day
−	3 01 012			Hour, minute
need	0 04 004	int	MGG	Hour
need	0 04 005	int	NGG	Minute
+	0 08 021	int	MTISI	Time significance (= 26 (time of last known position))
−	3 01 011			Year, month and day
−	3 01 012			Hour, minute
−	3 01 021			Latitude and Longitude
need	0 05 001	float	MLAH	Latitude (high accuracy) [degree]
need	0 06 001	float	MLOH	Longitude (high accuracy) [degree]
need	0 07 030	float	MHOSNN	Height of station above MSL (1)
+	0 01 012	int	MDS	Platform drift direction
+	0 01 014	float	MDSDS	Platform drift speed

use	WMO descriptor	type	mnemonics	meaning
opt	0 33 023	int	MQOBL	Quality of buoy location
−	3 02 021			Waves
−	3 06 004			Depths / salinities / temperatures
+	0 31 001	int	MDREP	Delayed descriptor replication factor
+	0 07 062	float*n	NZNZN	Depth below sea / water surface (2)
+	0 22 043	float*n	MTN00	Sea / water temperature (2)
+	0 22 062	float*n	MSNSN	Salinity (2)
−	3 06 005			Depths / directions / speeds of currents
−	3 02 001			Pressure and pressure change
need	0 07 031	float	MHOBNN	Height of barometer above MSL (1)
need	0 10 004	float	MPPP	**Pressure**
opt	0 10 051	float	MPPPP	Pressure reduced to MSL
opt	0 10 061	float	NPPP	3-hour pressure change
+	0 10 063	int	NA	Characteristic of pressure tendency
opt	0 07 032	float	MHOSEN	Height of sensor above marine deck platform (for temperature and humidity measurements)
+	0 07 033	float	MHAWAS	Height of sensor above water surface (for temperature and humidity)
need	0 12 101	float	MTDBT	**Temperature** / dry-bulb temperature
need	0 12 103	float	MTDNH	**Dew-point** temperature
opt	0 13 003	int	MUUU	Relative humidity [%]
+	0 07 032	float	MHOSEN0	Height of sensor above marine deck platform (for wind measurement)
+	0 07 033	float	MHAWAS0	Height of sensor above w. surf. (wind)
+	0 08 082	int	NACH2V	Artificial correction of sensor height to another value
+	0 07 033	float	MHAWAS1	Height of sensor above w. surf. (wind)
opt	0 08 021	int	MTISI1	Time significance (= 2 (time averaged))
opt	0 04 025	int	NGGTP	Time period in minutes
need	0 11 001	int	NDNDN	**Wind** direction
need	0 11 002	float	NFNFN	**Wind** speed
opt	0 04 025	int	NGGTP0	Time period in minutes
opt	0 11 041	float	NFXGU	Maximum wind gust speed
opt	0 04 024	int	MGGTP	Time period in hours
opt	0 13 011	float	MRRR	Total precipitation [kg/m2]

Table notes:

(1) Only one of the variables 'MHOBNN' and 'MHOSNN' are needed to exist. Preferred to exist is 'MHOBNN' (and to be used if both variables exist and have non-missing values) because it should provide the precise height of the barometer for the pressure measurement, which is the observation with the most critical dependency on sensor (or 'station') height.

(2) '*n' in the variable type definition means that this variable has an additional dimension, i. e. several values may be present in one report. If the corresponding replication factor 'MDREP' is zero for all reports in the NetCDF file, then these multi-dimensional variables do *not* need to exist (and probably will not exist) in the NetCDF file.

- **GPS (GNSS) zenith total delay and water vapour**

  File name: 'cdfin_gps_zenith'.

  GPS or GNSS (Global Navigation Satellite System: GPS, GLONASS, + GALILEI) reports on zenith total (path) delay (ZTD) are obtained via GTS from the UK Met Office as BUFR reports in a template which has been approved by WMO and has the (Table D) descriptor '**TM 3 07 022**'. These reports can be directly converted into NetCDF. The following table describes this template, and a description can also be found at : http://egvap.dmi.dk/support/formats/egvap_bufr_v10.pdf .

  The formats consist of the descriptors in the following table where

  'need' : COSMO asks stringently for this variable (and will abort if variable is absent)
  'opt' : variable exists and is read/used (but COSMO will not abort if it does not exist)
  '+' : variable exists but is not read by COSMO

GPS	descript.	type	mnemonics	meaning
need	0 01 015	char*20	YSOSN	Station or site name
need	0 04 001	int	MJJJ	Year
need	0 04 002	int	MMM	Month
need	0 04 003	int	MYY	Day
need	0 04 004	int	MGG	Hour
need	0 04 005	int	NGG	Minute
need	0 05 001	float	MLAH	Latitude (high accuracy) [degree]
need	0 06 001	float	MLOH	Longitude (high accuracy) [deg.]
need	0 07 001	int	MHP	Height of station [m]
+	0 08 021	int	MTISI	Time significance
need	0 04 025	int	NGGTP	Time period or displacement [min]
need	0 10 004	float	MPPP	Pressure [Pa]
need	0 12 001	float	MTN	Temperature / dry bulb temperature [K]
need	0 13 003	int	MUUU	Relative humidity [%]
need	0 33 038	int	NQFGD	Quality flags for ground-based GNSS data
+	0 08 022	int	MTOTN	Total number (accumulation / average)
+	0 02 020	int *n	MSACL	Satellite classification
+	0 01 050	int *n	MPTID	Platform transmitter ID number
+	0 05 021	float*n	MDA	Bearing or azimuth [degree_true]
need	0 07 021	float*n	MDE	Elevation [degree]
need	0 15 031	float*n	NADES	**Atmospheric path delay** in sat. signal [m]
need	0 15 032	float*n	NEERR	Estimated error in atmospheric path delay [m]
+	0 08 060	int	MSSMS	Sample scanning mode significance
+	0 15 033	float	NDPDL	Diff. in path delays for Limb views ... [m]
+	0 15 034	float	NEEPDD	Estimated error in path delay difference [m]
+	0 08 060	int	MSSMS0	Sample scanning mode significance
+	0 15 033	float	NDPDL0	Diff. in path delays for Limb views ... [m]
+	0 15 034	float	NEEPDD0	Estimated error in path delay difference [m]
need	0 15 035	float	NCZWV	Component of zenith path delay due to water vapour [m]
need	0 13 016	float	NWLN	**Precipitable water** [kg/m**2]
+	0 15 011	float	MLIED	LOG_10 of integrated electron density [log(1/m**2)]

Table note: '*n' in the variable type definition means that this variable has an additional dimension. Currently, this is set to a fixed value of 25.

- **GNSS zenith total delay**

  File name: 'cdfin_gnss_ztd'.

  The GPS data described in the previous section can also be processed by the GNSS STD operator. In contrast to the GPS IWV operator the STD operator is not part of the nudging assimilation system. The output of the STD operator is written to a specific feedobs file `fof_gnssgb_*` but the model state is not modified.

  In order to distinguish between both operators unique input file names are required even if the contents of the files is identical.

- **single-level AMDAR**

  File name: 'cdfin_amdar'.

  The template follows the proposed WMO descriptor '**TM 3 11 010**' and is described in the table below. The use of the variables is defined as follows:

  'need' : COSMO asks stringently for this variable (and will abort if variable is absent)
  'opt' : variable exists and is read/used (but COSMO will not abort if it does not exist)
  '+' : variable exists but is not read by COSMO
  '(+)' : descriptor exists only in the BUFR file, not in the NetCDF file

use	WMO descriptor	type	mnemonics	meaning
use	WMO descriptor	type	mnemonics	meaning
+	0 31 021		MADDF	Associated field significance (1)
−	3 11 005			Standard AMDAR report
need	0 01 008	char*8	YAIRN	Aircraft identification
+	0 01 023	int	NOSNO	Observation sequence number
−	3 01 021			Latitude and Longitude
need	0 05 001	float	MLAH	Latitude (high accuracy) [degree]
need	0 06 001	float	MLOH	Longitude (high accuracy) [degree]
−	3 01 011			Year, month and day
need	0 04 001	int	MJJJ	Year
need	0 04 002	int	MMM	Month
need	0 04 003	int	MYY	Day
−	3 01 013			Hour, minute and second
need	0 04 004	int	MGG	Hour
need	0 04 005	int	NGG	Minute
+	0 04 006	int	MSEC	Second
need	0 07 010	int	NFLEV	Flight level (2)
opt	0 08 009	int	NDEPF	Detailed phase of flight (3)
need	0 11 001	int	NDNDN	**Wind** direction
need	0 11 002	float	NFNFN	**Wind** speed
opt	0 11 031	int	MB	Degree of turbulence
opt	0 11 036	float	NMDEWX	Derived equivalent vertical gust speed
need	0 12 101	float	MTDBT	**Temperature** / dry-bulb T
+	0 33 025	int	MAIV	ACARS interpolated values
need	0 08 004	int	MPHAI	Phase of flight
need	0 02 064	int	MQARA	Wind quality/roll angle
opt	0 13 003	int	MUUU	Relative **humidity**
need	0 12 103	float	MTDNH	Dew-point temperature

use	WMO descriptor	type	mnemonics	meaning
opt	0 13 002	float	MMIXR	Mixing ratio
−	1 02 000			Delayed replication of 2 descriptors
+	0 31 001	int	MDREP	Delayed descriptor replication factor
+	0 11 075	float*n	MMPI	Mean turbulent intensity (EDR) (4)
+	0 11 076	float*n	MPTI	Peak turbulent intensity (EDR) (4)
+	0 11 037	int	MTUIN	Turbulence index (EDR)
+	0 11 039	int	NTIED	Extended time of occur. of peak EDR
+	0 11 077	int	NRED	EDR reporting interval
+	0 20 042	int	NAICE	Ice/no ice
+	0 20 043	float	NPLWC	Peak liquid water content
+	0 20 044	float	NALWC	Average liquid water content
+	0 20 045	int	NSLD	Supercooled water droplet conditions

Table notes:

(1) Variable 'MADDF' does not exist in the WMO descriptor 'TM 3 11 010', but is added at DWD by the decoding software.

(2) Flight level is defined relative to the ICAO standard sea level pressure and is readily converted to static air pressure using standard formulae (i. e. using the ICAO standard atmosphere). Hence, flight level is *not* the geometrical height.

(Once converted, the original resolution (either 100ft or 10ft) in the BUFR report is lost, hence it is desirable to disseminate the element in the received form.)

(3) This phase of flight table is expanded to indicate wind quality from roll angle (or roll and pitch combined) and also to indicate the method of ascent and descent observation interval selection either by time or pressure increments.

(4) The use of '*n' in the variable type definition means that this variable has an additional dimension, i. e. several values may be present in one report. If the corresponding replication factor 'MDREP' is zero for all reports in the NetCDF file, then the variables 'MMPI' and 'MPTI' do *not* exist in the NetCDF file.

- **multi-level AMDAR**

File name: 'cdfin_amdar_ml'.

Caution: The use of this observation file type in COSMO has not been tested thoroughly yet, mainly due to a lack of data of this type (at least over Europe). The use of it is at the user's own risk.

The template follows the proposed WMO descriptor '**TM 3 11 009**' and is described in the table below, which is split into two parts for convenience. There is also a proposed descriptor '**TM 3 11 008**' for aircraft profiles *without* latitude and longitude reported at each level. (Descriptor 'TM 3 11 008' equals 'TM 3 11 009' except that '3 11 007' is replaced by '3 11 006', which in turn is the same as '3 11 007' except that '3 01 021' is missing.) The use of the various variables is defined as follows:

'need' : COSMO asks stringently for this variable (and will abort if variable is absent)

'opt' : variable exists and is read/used (but COSMO will not abort if it does not exist)

'+' : variable exists but is not read by COSMO

'(+)' : descriptor exists only in the BUFR file, not in the NetCDF file

use	WMO descriptor	type	mnemonics	meaning
opt	0 01 033	int	MMIOGC	Identific. of orig./ generat. centre (1)
+	0 01 034	int	MMIOGS	Identific. of orig./ gen. sub-centre (1)
+	0 01 023	int	NOSNO	Observation sequence number (1)
need	0 01 008	char*8	YAIRN	Aircraft identification
−	3 01 011			Year, month and day
need	0 04 001	int	MJJJ	Year
need	0 04 002	int	MMM	Month
need	0 04 003	int	MYY	Day
−	3 01 013			Hour, minute and second
need	0 04 004	int	MGG	Hour
need	0 04 005	int	NGG	Minute
+	0 04 006	int	MSEC	Second
−	3 01 021			Latitude and Longitude
need	0 05 001	float	MLAH	Latitude (high accuracy) [degree]
need	0 06 001	float	MLOH	Longitude (high accuracy) [degree]
need	0 08 004	int	MPHAI	Phase of flight
−	1 01 000			Delayed replication of 1 descriptor
need	0 31 001	int	MDREP	Delayed descriptor replication factor
−	3 11 007			Aircraft ascent / descent profile data for 1 level with lat. / lon. indicated
need	0 07 010	int *n	NFLEV	Flight level (2, 3)
−	3 01 021			Latitude and Longitude
opt	0 05 001	float*n	MLAH0	Latitude (high accuracy) [degree] (3)
opt	0 06 001	float*n	MLOH0	Longitude (high accuracy) [degree] (3)
need	0 11 001	int *n	NDNDN	**Wind** direction (3)
need	0 11 002	float*n	NFNFN	**Wind** speed (3)
need	0 02 064	int *n	MQARA	Wind quality/roll angle (3)
need	0 12 101	float*n	MTDBT	**Temperature** / dry-bulb temperature (3)
need	0 12 103	float*n	MTDNH	**Dew-point** temperature (3)
+	0 13 002	float*n	MMIXR	Mixing ratio (1, 3)

Table notes:

(1) The Variables 'MMIOGC', 'MMIOGS', 'NOSNO', and 'MMIXR' do not exist in the WMO BUFR descriptor 'TM 3 11 009', but are added at DWD by the data base decoding software.

(2) Flight level is defined relative to the ICAO standard sea level pressure and is readily converted to static air pressure using standard formulae (i. e. using the ICAO standard atmosphere). Hence, flight level is *not* the geometrical height.

(Once converted, the original resolution (either 100ft or 10ft) in the BUFR report is lost, hence it is desirable to disseminate the element in the received form.)

(3) The use of '*n' in the variable type definition means that this variable has an additional dimension, here for vertical levels. If the corresponding replication factor 'MDREP' is zero for all reports in the NetCDF file, then these multi-dimensional variables do *not* need to exist (and probably will not exist) in the NetCDF file.

5.3.3 Observation types without templates proposed by WMO

- **ACARS**

  File names: 'cdfin_acars', 'cdfin_acars_uk', resp. 'cdfin_acars_us'.

  As a standard way implemented at DWD, ACARS can be read by COSMO in 2 different ways:

  1. A file type ('cdfin_acars_us') with BUFR obtained via GTS from ARINC Center 56 (USA, denoted as '_us' in the table below) plus another file type ('cdfin_acars_uk') with BUFR obtained via GTS from UK Met Office and Canada (denoted as '_uk'), or
  2. A single file type ('cdfin_acars') in a unified format (defined by DWD) which contains the reports from the two other files (denoted as 'unif.').

  The formats consist of the descriptors in the following table where

  'need': COSMO asks stringently for this variable (and will abort if variable is absent)

  'opt': variable exists and is read/used (but COSMO will not abort if it does not exist)

  '+': variable exists but is not read by COSMO

  '–': variable does not exist in this format

unif.	_uk	_us	descript.	type	mnemonics	meaning
opt	–	–	0 01 033	int	MMIOGC	Identific. of orig./ generat. centre
+	–	–	0 01 034	int	MMIOGS	Identific. of orig./ gen. sub-centre
need	need	need	0 01 008	char*8	YAIRN	Aircraft identification
–	+	+	0 01 006	char*8	YXXNN	Aircraft identification
+	–	–	0 01 023	int	NOSNO	Observation sequence number
–	–	opt	0 02 001	int	NIX	Type of station
–	–	+	0 02 002	int	NIW	Instrument type for wind measure.
–	+	–	0 02 061	int	NS1	Aircraft navigational system
need	need	need	0 05 001	float	MLAH	Latitude (high accuracy) [degree]
need	need	need	0 06 001	float	MLOH	Longitude (high accuracy) [deg.]
need	need	need	0 04 001	int	MJJJ	Year
need	need	need	0 04 002	int	MMM	Month
need	need	need	0 04 003	int	MYY	Day
need	need	need	0 04 004	int	MGG	Hour
need	need	need	0 04 005	int	NGG	Minute
+	–	+	0 04 006	int	MSEC	Second
–	need	–	0 07 002	float	MHHH	Height or altit. (vert. level) (1, 2)
need	–	opt	0 07 004	float	MPN	Pressure (vertical level) (1)
–	+	–	0 07 007	int	MH	Height
need	–	–	0 07 010	int	NFLEV	Flight level (1, 2)
–	–	need	0 10 070	int	MIAA	Indicated aircraft altitude (1, 2)
opt	–	–	0 08 009	int	NDEPF	Detailed phase of flight (3)
need	need	need	0 11 001	int	NDNDN	**Wind** direction
need	need	need	0 11 002	float	NFNFN	**Wind** speed

unif.	_uk	_us	descript.	type	mnemonics	meaning
opt	opt	–	0 11 031	int	MB	Degree of turbulence
–	+	–	0 11 032	float	MHBT	Height of base of turbulence
–	+	–	0 11 033	float	MHTT	Height of top of turbulence
opt	opt	–	0 11 036	float	NMDEWX	Derived equiv. vertical gust speed
–	need	need	0 12 001	float	MTN	**Temperature** / dry-bulb T (4)
need	–	–	0 12 101	float	MTDBT	Temperature / dry-bulb T (4)
+	–	+	0 33 025	int	MAIV	ACARS interpolated values
need	need	need	0 08 004	int	MPHAI	Phase of flight
need	need	need	0 02 064	int	MQARA	Wind quality/roll angle
opt	–	opt	0 13 003	int	MUUU	Relative **humidity**
opt	–	–	0 12 103	float	MTDNH	Dew-point temperature
opt	opt	opt	0 13 002	float	MMIXR	Mixing ratio
–	+	–	0 11 076	float	MPTI	Peak turbulent intensity (EDR)
+	+	–	0 20 041	int	MAICI	Airframe icing
opt	opt	opt	0 02 005	float	MPOTO	Precision of temperature observ.
+	+	+	0 02 062	int	NADRS	Type of aircraft data relay system
+	–	+	0 02 070	int	NOSLL	Original specif. of latit. / longit.
+	–	+	0 02 065	char*5	YAGRS	ACARS ground receiving system
opt	–	opt	0 33 026	int	MMRQ	Mixing ratio quality
–	–	+	0 04 015	int	NGGTI	Time increment
–	–	+	0 04 032	int	NGGTM	Duration rel. to following value
–	–	+	0 11 235	int	011235	unknown descriptor

Table notes:

(1) For the definition of the vertical level, it is in fact only required that at least one of the variables ’NFLEV’, ’MIAA’, ’NHHH’, or ’MPN’ exists in (any of) the NetCDF file(s) but it does not really matter which one(s). (And in order to use a report, the corresponding vertical level must not be a missing value.)

(2) Flight level is defined relative to the ICAO standard sea level pressure and is readily converted to static air pressure using standard formulae (i. e. using the ICAO standard atmosphere). Hence, flight level is *not* the geometrical height. In the above file types, the variables ’MIAA’ and ’NHHH’ also denote the same type of flight level as variable ’NFLEV’ and are therefore not geometrical height either.

(Once converted, the original resolution (either 100ft or 10ft) in the BUFR report is lost, hence it is desirable to disseminate the element in the received form.)

(3) This phase of flight table is expanded to indicate wind quality from roll angle (or roll and pitch combined) and also to indicate the method of ascent and descent observation interval selection either by time or pressure increments.

(4) For temperature, it is in fact only required that at least one of the variables ’MTDBT’ or ’MTN’ exists in (any of) the NetCDF file(s), but it does not really matter which one.

The unified ACARS format (’unif.’) produced at DWD (file type ’cdfin_acars’) contains additionally the following variables which do not occur in either of the ACARS formats (’_uk’ and ’_us’) obtained via GTS (except for variable ’MPTI’), but which are part of the AMDAR template proposed by WMO:

unif.	descript.	type	mnemonics	meaning
–	1 02 000			Delayed replication of 2 descriptors
+	0 31 001	int	MDREP	Delayed descriptor replication factor
+	0 11 075	float*n	MMPI	Mean turbulent intensity (EDR) (1)
+	0 11 076	float*n	MPTI	Peak turbulent intensity (EDR) (1)
+	0 11 037	int	MTUIN	Turbulence index (EDR)
+	0 11 039	int	NTIED	Extended time of occurrence of peak EDR
+	0 11 077	int	NRED	EDR reporting interval
+	0 20 042	int	NAICE	Ice/no ice
+	0 20 043	float	NPLWC	Peak liquid water content
+	0 20 044	float	NALWC	Average liquid water content
+	0 20 045	int	NSLD	Supercooled water droplet conditions

Table note:

(1) The use of '*n' in the variable type definition means that this variable has an additional dimension, i. e. several values may be present in one report. If the corresponding replication factor 'MDREP' is zero for all reports in the NetCDF file, then the variables 'MMPI' and 'MPTI' do *not* exist in the NetCDF file.

- **Wind Profiler, RASS temperature profiler, Radar VAD wind profiles**

File names: 'cdfin_wprof', 'cdfin_rass', resp. 'cdfin_radar_vad'.

Wind Profiler reports ('WP'), Radio Acoustic Sounding System temperature profile reports ('RASS'), and Radar Velocity Azimuth Display wind profile reports ('VAD') are usually obtained as BUFR reports with a variety of templates. At DWD, BUFR reports with a unified template for each of the 3 data types are produced from the original reports for subsequent conversion into NetCDF. These unified templates are described below.

The formats consist of the descriptors in the following table where

'need':	COSMO asks stringently for this variable (and will abort if variable is absent)
'opt':	variable exists and is read/used (but COSMO will not abort if it does not exist)
'+':	variable exists but is not read by COSMO
'–':	variable does not exist in this format

WP	RASS	VAD	descript.	type	mnemonics	meaning
opt	opt	opt	0 01 033	int	MMIOGC	Identif. of orig./ generat. centre
+	+	+	0 01 034	int	MMIOGS	Identif. of orig./ gen. sub-centre
need	need	need	0 01 001	int	MII	WMO block number
need	need	need	0 01 002	int	NIII	WMO station number
+	+	–	0 02 001	int	NIX	Type of station
need	need	need	0 04 001	int	MJJJ	Year
need	need	need	0 04 002	int	MMM	Month
need	need	need	0 04 003	int	MYY	Day
need	need	need	0 04 004	int	MGG	Hour
need	need	need	0 04 005	int	NGG	Minute

WP	RASS	VAD	descript.	type	mnemonics	meaning
need	need	need	0 05 002	float	MLALA	Latitude (coarse accuracy) [deg.]
need	need	need	0 06 002	float	MLOLO	Longitude (coarse accur.) [deg.]
need	need	need	0 07 001	int	MHP	Height of station [m]
opt	opt	–	0 10 018	char*5	YSSOSN	Short station or site name
opt	opt	opt	0 02 003	int	NA4	Type of measur. equipment used
opt	opt	–	0 25 021	int	MWCE	Wind computation enhancement
opt	opt	–	0 08 021	int	MTISI	Time significance
opt	–	–	0 04 026	int	MSETP	Time period or displacement [s]
need	need	need	0 31 001	int	MDREP	Delayed descript. replic. factor
opt	–	opt	0 33 002	int *n	MQINZ	Quality information
need	need	need	0 07 007	int *n	MH	Height [m]
need	–	need	0 11 001	int *n	NDNDN	**Wind** direction [degree]
need	–	need	0 11 002	float*n	NFNFN	**Wind** speed [m/s]
–	need	–	0 12 007	float*n	MTVIR	**Virtual temperature** [K]
opt	opt	opt	0 11 006	float*n	MWMPS	W-component [m/s]
opt	opt	–	0 21 030	int *n	NSINOR	Signal to noise ratio [db]
opt	opt	–	0 25 034	int *n	NWPQ	NOAA WP quality ctrl. results
opt	–	opt	0 11 050	float*n	NSTDFF	Stand. deviat. wind speed [m/s]
+	–	–	0 11 051	float*n	NSTDVF	Std. vertical wind speed [m/s]

Table notes:

- If ‘YSSOSN’ is present, then ‘MII’ and ‘NIII’ are not strictly mandatory.
- ’*n’ in the variable type definition means that this variable has an additional dimension, here for vertical levels. If the corresponding replication factor ’MDREP’ is zero for all reports in the NetCDF file, then these multi-dimensional variables do *not* need to exist (and probably will not exist) in the NetCDF file.
- Not all variables are listed here, that are present in some of the templates, but are not read and used by COSMO.
- Some of the variables that are used (but not needed) e.g. in the wind profiler file but are not present e.g. in the VAD file, could be added to the VAD file in the future and used without changing the COSMO code.

5.3.4 The blacklist file

The blacklist file contains both a blacklist and whitelist. The blacklist contains the stations with the variables and vertical ranges that are known to have a bad quality statistically. Observations of that variables and in that vertical ranges are then excluded permanently from active use in the assimilation. The whitelist contains all stations related to certain observation types, which are known to issue observations of good quality.

The difference in concept between the blacklist and the whitelist becomes important when a new station sends observations of a certain type via GTS and these data have unknown quality. These observations will be excluded from active use only if a whitelist exists for that observation type. Whitelists are useful for types where there are not too many stations and the risk that a new station will deliver data of minor quality is considered rather high. This often applies to remote sensing observation types.

A station (related to a certain observation type) can appear both on the whitelist and (several times) on the blacklist. The whitelist activates the station as a whole, and in the blacklist, certain vertical ranges for certain variables can be excluded then from active use.

The blacklist file is a formatted ASCII file and has a format as follow:

Format of Blacklist :

The first line is fixed, and the following lines are the entries in the blacklist, with following 10 columns:

1: station identity (’.’ as wildcard allowed)

2: observation type (1: surface-level, 2: aircraft, 4: buoy, 5: TEMP, 6: PILOT)

3,4: lower / upper limit (pressure in [hPa]) of blacklisted vertical range for geopotential

5,6: blacklisted vertical range for (horizontal) wind

7,8: blacklisted vertical range for temperature

9,10: blacklisted vertical range for humidity

```
2IIIIIII T P.G.U P.G.O P.W.U P.W.O P.T.U P.T.O P.D.U P.D.O
 01295    1  1100     0     0     0     0     0     0     0
 0210     1  1100     0     0     0     0     0     0     0
 9Z4Y48JP 2     0     0     0     0  1100     0     0     0
 ABX.     2     0     0  1100     0     0     0     0     0
 ABX..    2     0     0  1100     0     0     0     0     0
 ABX...   2     0     0  1100     0     0     0     0     0
 ABX....  2     0     0  1100     0     0     0     0     0
 17912    4  1100     0     0     0     0     0     0     0
 21523    4     0     0  1100     0     0     0     0     0
 07137    5  1100     0  1100     0  1100     0     0     0
 10437    5   300     0  1100     0     0     0     0     0
 10828    5  1100     0  1100     0     0     0     0     0
 10204    6     0     0   500     0     0     0     0     0
 10266    6     0     0   200     0  1100     0     0     0
 10384    6     0     0  1100   850     0     0     0     0
```

Immediately after the last line of the blacklist, the whitelist follows.

Standard format of Whitelist

```
    WHITELIST
03019     6    132
10135     6    132
10266     6    132
10394     6    132
47912     6    134
47945     6    134
72246     6    136
74341     6    136
10169     6    137
10204     6    137
10384     6    137
```

Alternative format of Whitelist

```
WHITELIST  6   132
03019     6    132
10135     6    132
10266     6    132
10394     6    132
WHITELIST  6   133
WHITELIST  6   134
47912     6    134
47945     6    134
WHITELIST  6   136
72246     6    136
74341     6    136
WHITELIST  6   137
10169     6    137
10204     6    137
10384     6    137
```

In the standard format, the first line is fixed, and the following lines are the entries in the whitelist, with the following 3 columns:

1: station identity
2: observation type (currently only 6: PILOT , which includes profilers)
3: observation code type (132: European wind profiler,
133: European RASS / SODAR,
134: Japan. wind profiler / RASS,
136: US wind profiler / RASS,
137: Radar VAD wind profiles)

The standard format assumes a whitelist exactly for those observation code types for which there are entries on whitelist. In the blacklist file used at DWD currently, no RASS profiler station is on the whitelist, which implies that COSMO will assume that no whitelist exists for RASS. Hence, all RASS reports would be used actively unless they are on the blacklist. This is indeed the case, i. e. the temperature profiles from all the known RASS stations are put on the blacklist, see e. g. station `10266` in the example above.

There is an alternative format for the whitelist which can be used (but is not yet as thoroughly tested). A line containing '`WHITELIST`' and observation type and code type precedes the whitelist for each code type, even if the whitelist for that code type is empty. Only this alternative format allows to use empty whitelists, and the above example in this alternative format is also shown above.

Section 6

Namelist Input for COSMO-Model

The execution of the COSMO-Model is controlled by 14 NAMELIST-groups:

- LMGRID – specifying the domain and the size of the grid
- RUNCTL – parameters for the model run
- DYNCTL – parameters for the adiabatic model
- PHYCTL – parameters for the diabatic model
- TUNING – parameters for tuning dynamics and physics
- DIACTL – parameters for the diagnostic calculations
- SATCTL – parameters for the satellite images
- INICTL – parameters for the initialization of model variables
- EPSCTL – controlling the ensemble prediction mode
- IOCTL – controlling the environment
- DATABASE – specification of database job
- GRIBIN – controlling the grib input
- GRIBOUT – controlling the grib output
- NUDGING – controlling the data assimilation

These NAMELIST-groups have to appear in the corresponding INPUT_* files. See Tab. 5.1 for the distribution of the NAMELIST-groups to the INPUT_* files. The INPUT_* files have to be in the directory, from where the model is started. Every group is read in a special subroutine called input_*groupname*. This subroutine also sets default values for all parameters and checks most parameters that have been changed for correctness and consistency.

The NAMELIST variables can be specified by the user in the run-script for the model, which then creates the INPUT_* files. An excerpt of this run script is shown in Figure 6.1 for the forecast part and in Figure **??** for the nudging part.

```
cat > INPUT_ORG << end_input_org
 &LMGRID
  startlat_tot  = -5.0,  startlon_tot  =  -5.0,
  pollat=40.0,        pollon=-170.0,
  dlon=0.025,         dlat=0.025,
  ie_tot=421,         je_tot=461,         ke_tot=50,
  irefatm=2,
 /
 &RUNCTL
  hstart = 0.0,   hstop = 2.0,    dt      = 25.0,    ydate_ini=’$DATE’,
  nprocx = 10,    nprocy = 20,    nprocio = 0,
  lphys      = .TRUE.,   luse_rttov    = .FALSE.,  luseobs = .TRUE.,
  lreproduce = .TRUE.,   itype_timing = 4,    ldatatypes = .TRUE.,
  ncomm_type=3, nboundlines= 3, idbg_level=2, ldiagnos   = .TRUE.,
 /
 &TUNING
  clc_diag  =   0.5, pat_len   = 750.0, tur_len   = 150.0,
  rlam_heat =   1.0, rlam_mom  =   0.0, rat_lam   =   1.0,
  rat_can   =   1.0, rat_sea   =   7.5, c_lnd     =   2.0,
 /
end_input_org

cat > INPUT_IO  << end_input_io
 &IOCTL
  lasync_io=.FALSE., ngribout=3, yform_read=’apix’,
 /
 &GRIBIN
  hincbound=1.0,
  lchkini    = .TRUE.,  lchkbd    = .TRUE.,  lbdana=.FALSE.,
  ydirini=’/gtmp/routfor/dat/initial/’,
  ydirbd=’/gtmp/routfor/dat/boundaries/’,
 /
 &GRIBOUT
    hcomb=0.0,3.0,1.0,
    lcheck=.TRUE.,
    ydir=’/gtmp/routfor/dat/output/’,
    yvarml=’U         ’,’V         ’,’W         ’,’T         ’,
    yvarpl=’T         ’,’RELHUM    ’,’U         ’,’V         ’,
    yvarzl=’T         ’,’RELHUM    ’,’U         ’,’V         ’,
    yvarsl=’SYNMSG’,
    plev=400.0,  500.0,  600.0,  700.0,  850.0, 950.0,  975.0, 1000.0,
    zlev=1000.0, 2000.0, 3000.0, 5000.0,
 /
end_input_io

cat > INPUT_DYN << end_input_dyn
 &DYNCTL
    l2tls=.TRUE., l_diff_Smag=.TRUE., y_scalar_advect=’BOTT2_STRANG’,
    irunge_kutta=1, irk_order=3, iadv_order=5, itype_fast_waves=2,
    itype_bbc_w=114, rlwidth=50000.0, itype_outflow_qrsg=2,
 /
end_input_dyn
```

Figure 6.1: Excerpt of run script from the COSMO-Model to create INPUT_* files

The following sections serve as a complete reference to all namelist variables used in the COSMO-Model. There is a section for every namelist group. Within the sections the descriptions of the variables is organized in subgroups, if the variables can be separated into different scopes. For example in the physics group `PHYCTL` there is a subgroup for every parameterization. All variables are displayed in tables, showing the name of the variable, their numerical type in the model, the definition including some comments and the default value.

A problem that less experienced COSMO users often face, is to find out, which namelist variables they have to adapt, if they want to change e.g. the domain or the resolution of their application. Therefore we introduced an additional 5th column to the tables, which should indicate the dependency of a variable on a special category. The following categories have been defined up to now:

M:	Mode of simulation (NWP, CLM, ITC, ART)
R:	Region
dx:	Horizontal resolution
dz:	Vertical resolution
dt:	Time step
BC:	Boundary conditions
P:	Performance

If you want to change for example your horizontal resolution, you should check for which variables a dependency on dx is listed in the tables below.

6.1 LMGRID — Model Domain and Reference Atmosphere

The namelist group LMGRID contains parameters that specify

- the lat-lon coordinates of the pole of the rotated grid,
- the position of the model domain within the rotated grid,
- the size and resolution of the model grid.
- and parameters for the reference atmosphere (since COSMO-Model Version 5.1)

The specifications of the parameters for this group are included in the file INPUT_ORG.

Domain size and model grid

Name	Type	Definition / Purpose / Comments	Default	Depend.
pollat	REAL	Geographical latitude of the rotated north pole (in degrees, north > 0); for a non-rotated lat-lon grid set pollat = 90.	32.5	R
pollon	REAL	Geographical longitude of the rotated north pole (in degrees, east > 0); for a non-rotated lat-lon grid set pollon = -180.	-170.0	R
polgam	REAL	Angle between the north poles of two rotated grids (in degrees, east > 0); necessary for transformation from one rotated grid to another rotated grid	0.0	R
dlat	REAL	'Meridional' (rotated lat-direction) grid spacing (in degrees).	0.008	dx
dlon	REAL	'Zonal' (rotated lon-direction) grid spacing (in degrees).	0.008	dx
startlat_tot	REAL	Latitude of the lower left scalar grid point of the total domain (in degrees, north > 0, rotated coordinates).	-7.972	R
startlon_tot	REAL	Longitude of the lower left scalar grid point of the total domain (in degrees, east > 0, rotated coordinates).	-1.252	R
ie_tot	INT	Number of gridpoints of the total domain in 'west-east' direction of the rotated coordinates.	51	dx,R
je_tot	INT	Number of gridpoints of the total domain in 'south-north' direction of the rotated coordinates.	51	dx,R
ke_tot	INT	Number of gridpoints of the total domain in vertical direction (i.e. number of layers or main-levels; there are ke_tot+1 half-levels).	20	dz

The specifications for the model domain and the grid size are compared to the values from the headers of the data files (the Grid Description Section (GDS) of the Grib files or the headers of NetCDF files) for the initial and boundary fields. If they do not correspond, the program will print an error message and abort.

Variables for the reference atmosphere

Since Version 5.1, there are also namelist variables to specify the reference atmosphere parameters, because these parameters cannot be written to the GRIB2 meta data any more.

When using GRIB1 or NetCDF, these parameters are specified in INT2LM and are written to the meta data of all atmospheric fields. The pressure field exchanged between INT2LM and the COSMO-Model or within a COSMO-Model assimilation cycle is the pressure deviation `PP`, because this field suffers less from the precision loss due to GRIB packing than the full pressure `P`. When the COSMO-Model reads the atmospheric fields, including `PP`, it gets the values for the reference atmosphere from the meta data, constructs the reference pressure field `P0` and then computes the full pressure `P`.

When using GRIB2, not the pressure deviation, but the full pressure field `P` is exchanged between INT2LM and the COSMO-Model or within a COSMO-Model assimilation cycle. This is done using a packing to 24 bits (while most other fields are packed to 16 bits). This gives a high enough precision for the values of the full pressure. Internally, the COSMO-Model is working with the pressure deviation as a prognostic variable, therefore it has to compute the reference pressure field `P0`, and for that has to know the reference atmosphere parameters. Therefore we introduced the following namelist variables; their names and meanings are identical to the corresponding namelist variables in the INT2LM.

Name	Type	Definition / Purpose / Comments	Default	Depend.
`irefatm`	`INT`	type of reference atmosphere 1: The reference atmosphere is based on a constant rate β for the temperature increase with the logarithm of pressure: $\partial T_0 / \partial \ln p_0 = \beta$. 2: The reference atmosphere is based on a temperature profile $$t_0(z) = (t_{0sl} - \Delta t) + \Delta t \cdot \exp\left(\frac{-z}{h_{scal}}\right),$$ where $z =$ `hhl(k)` is the height of a model grid point. If this option is used, the values for $\Delta t =$ `delta_t` and $h_{scal} =$ `h_scal` have also to be set.	2	dz
`p0sl`	`REAL`	constant reference pressure on sea-level	10000.0	
`t0sl`	`REAL`	constant reference temperature on sea-level	288.15	
`dt0lp`	`REAL`	d (t0) / d (ln p0)	42.0	
`delta_t`	`REAL`	temperature difference between sea level and stratosphere (for `irefatm=2`)	75.0	
`h_scal`	`REAL`	scale height (for `irefatm=2`)	10000.0	

Note that the reference atmosphere used in INT2LM and the one used in the COSMO-Model need not be the same, when the full pressure `P` is exchanged. The only dependency is: If the top of the atmosphere is above 27 km and INT2LM is using `irefatm=2`, then also the COSMO-Model has to use `irefatm=2`.

6.2 RUNCTL — Parameters for the Model Run

The namelist group RUNCTL contains the parameters that control the basic configuration of a model run. The specifications for the parameters in RUNCTL are included in the file INPUT_ORG.

The namelist parameters of this group are described in the subsections

- initial time, forecast range, time step and calendar used,
- basic control parameters,
- parameters for using an incremental analysis update,
- parameters related to artificial initial and boundary conditions,
- parameters for parallel and sequential execution,
- parameters for diagnostic min/max model output and
- parameters for debug purposes

Initial time, forecast range, time step and calendar used

Name	Type	Definition / Purpose / Comments	Default	Depend.
ydate_ini	CHAR	Initial date and time of the forecast. Since Version 4.24 there are two different formats: • ydate_ini = ’yyyymmddhh’, or • ydate_ini = ’yyyymmddhhmise’. where yyyy = year, mm = month, dd = day, hh = hour, mi = minutes and se = seconds. If the date is specified in the old format (still the default), all file names are also treated as usual. If the date is given with minutes and seconds, then all relevant files are also written with minutes and seconds in the file names. This will be necessary, if a sub-hourly analysis cycle will be used.	’ ’	
ydate_bd	CHAR	Start date and time of the forecast from which the boundary fields are used, specified by year-month-day-hour. (’yyyymmddhh’) as above. ydate_bd can be set different than ydate_ini, when the forecast, which produced the boundary files, started earlier than ydate_ini, for example, if forecasts from 00 UTC are used to drive a COSMO-Model forecast starting at 06 UTC. Note that ydate_bd always has to be older than ydate_ini! *If omitted,* ydate_bd *is set to* ydate_ini.	’ ’	

Name	Type	Definition / Purpose / Comments	Default	Depend.
ydate_end	CHAR	End date for the whole forecast. This is needed for climate simulations, where the forecast can be splitted into several runs on a computer and hstop only indicates the stop of a single run. The format is as above, i.e. either 'yyyymmddhh' or 'yyyymmddhhmise'. If omitted, ydate_end is not set and the end of the whole forecast is specified by hstop or nstop, resp.	' '	
lyear_360	LOG	To use a climatological year with 360 days. Eliminated: replaced by itype_calendar in Version 4.4		BC
itype_calendar	INT	To specify, which calendar is used during the forecast (introduced in Version 4.4): 0: gregorian calendar 1: every year has 360 days 2: every year has 365 days	0	BC
hstart	REAL	Number of the first hour relative to the initial time given by ydate_ini. For a regular forecast, hstart = 0.0. If hstart > 0.0, the model is in restart mode and it requires restart data (for Leapfrog from 2 consecutive time levels; for Runge-Kutta only from 1 time level) to resume a forecast run.	0.0	
hstop	REAL	Duration of the forecast in hours.	0.0	
nstop	INT	Number of time steps to be performed; determines the forecast range (nstop*dt) relative to the initial time. *can be specified alternatively to* hstop.	0	

Name	Type	Definition / Purpose / Comments	Default	Depend.
`dt`	`REAL`	Time step (in seconds); `dt` refers to the large time step size Δt associated with the Leapfrog or Runge-Kutta time integration of the slow modes. Since vertical advection and diffusion is calculated implicitly, the large time step can be determined from the linear stability criterion for horizontal advection $\Delta t \leq \frac{\Delta s}{\sqrt{2}\, v_{max}}\,,$ where Δs is the minimum grid spacing in physical space and v_{max} is the maximum absolute horizontal velocity during the forecast. For operational purposes, v_{max} is estimated as a high jet-stream velocity of about 120 m/s. The number N_s of small time steps $\Delta\tau$ per Leapfrog time intervall ($2\Delta t = N_s \Delta\tau$) is calculated automatically inside the program using the CFL stability criterion for horizontal acoustic wave propagation to estimate $\Delta\tau$. With the above value for v_{max}, we get 7 small steps per Leapfrog intervall (or 4 steps per Eulerian forward time step in the Runge-Kutta scheme). *Note: Both too small values ($N_s < 3$) and too large values ($N_s > 20$) for the number of small steps may have a detrimental impact to the solution.*	10.0	dx

Basic control parameters

The basic control parameters allow to switch on and off main features and/or packages of the model system.

Name	Type	Definition / Purpose / Comments	Default	Depend.
`lphys`	`LOG`	Main switch to include physical parameterizations. If .FALSE., the corresponding Namelist input from `PHYCTL` (see Section 6.4) is skipped and no physical parameterizations will be computed – except cloud condensation and evaporation by saturation adjustment. To supress this process, set `lcond = .FALSE.` in input group `DYNCTL` (see Section 6.3).	`.TRUE.`	
`ldiagnos`	`LOG`	Main switch to include diagnostic calculations. If this switch is set to `.TRUE.`, the variables in the group `DIACTL` should also be set (see Section 6.6)	`.FALSE.`	
`ldfi`	`LOG`	Main switch to run a model initialization by using the digital filtering technique. If this switch is set to `.TRUE.`, the variables in the group `INICTL` should also be set (see Section 6.8)	`.FALSE.`	IC

Name	Type	Definition / Purpose / Comments	Default	Depend.
luseobs	LOG	Main switch to use observations during the model run (for data assimilation purposes). To perform a free model forecast, luseobs is set to .FALSE. If this switch is set to .TRUE., the variables in the group NUDGING should also be set (see Section ??)	.FALSE.	M
leps	LOG	Main Switch to run the model in ensemble mode. Setting this switch has the effect that special GRIB meta data are set and / or processed during I/O. If lsppt is also set, special actions are taken during the stochastic perturbation of physics tendencies (SPPT). If this switch is set to .TRUE., the variables in the group EPSCTL should also be set (see Section 6.9)	.FALSE.	M
lsppt	LOG	Main Switch to run stochastic perturbation of physics tendencies (SPPT). If it is activated, related namelist variables are read from the namelist group /EPSCTL/.	.FALSE.	
luse_rttov	LOG	Main switch to compute synthetic satellite images. If this switch is set to .TRUE., the variables in the group SATCTL should also be set (see Section 6.7)	.FALSE.	
luse_radarfwo	LOG	To activate the radar forward operator.	.FALSE.	
lroutine	LOG	To indicate an operational run (if .TRUE.. This variable is used to set GRIB2 meta data.	.FALSE.	
l_cosmo_art	LOG	Main switch to compute the *Aerosol and Reactive Tracer* module. The COSMO-Model has to be compiled with -DCOSMOART to activate this module in the source code of the COSMO-Model. The source code of the ART module itself can be obtained from the *Karlsruhe Institute of Technology*.	.FALSE.	M
l_pollen	LOG	Main switch to compute the *Pollen* module. The COSMO-Model has to be compiled with -DPOLLEN to activate this module in the source code of the COSMO-Model. The source code of the Pollen module itself can be obtained from the *Karlsruhe Institute of Technology*.	.FALSE.	M
ltraj	LOG	Main switch to active the online trajectory module implemented by ETH Zürich.	.FALSE.	
llm	LOG	Running with zero vertical velocity as lower boundary condition in the fast_waves solver.	.FALSE.	

Parameters for the blocked physics

With the implementation of the unified COSMO-ICON physical parameterizations there is the need for a special data structure in the physics, because ICON uses a 2-dimensional data structure with *(number of grid points, vertical dimension)*.

This data format, which is called the *blocked data format*, has now been implemented in the COSMO-Model (starting in Version 5.01). A special `copy-in` / `copy-out` mechanism has been implemented to copy data between this new blocked structure and the basic 3-D data format of the COSMO-Model.

In an ideal world it is:

$$\texttt{nproma} \cdot \texttt{nblock} = (\texttt{iendpar - istartpar + 1}) \cdot (\texttt{jendpar - jstartpar + 1}),$$

where `nproma` is the *number of grid points* and `nblock` is the number of blocks necessary to hold all grid points from the original COSMO 2-D data structure `(ie,je)`. And because the world never is ideal, the last block will usually have a shorter length than `nproma`. In the program this length is denoted with `nlastproma`.

To specify the blocked data structure one can either specify the length of the first dimension, `nproma`, or the number of blocks `nblock`. In the COSMO-Model we implemented both variables as namelist variables, but specifying `nblock` has precedence over specifying `nproma`.

Name	Type	Definition / Purpose / Comments	Default
`nproma`	`INT`	To specify the number of grid points within one block of the blocked data structure. (Introduced in Version 5.01).	`16`
`nblock`	`INT`	To specify the number of blocks of the blocked data structure. (Introduced in Version 5.01).	`-1`

Parameters for using an incremental analysis update

Usually the model is started from an analysis file. When using the incremental analysis update (IAU), the model is started from a first guess (a forecast with lead time given by the namelist variable `nsec_fg` (in namelist group `/GRIBIN/`. The analysis increments are added to the model fields incrementally at each time step during the model integration over a certain period, given by namelist variable `peri_iau`.

Further details:

- The IAU updates the sum of water vapour and cloud water. After this, the water vapour and cloud water are balanced without applying any latent heating because the IAU updates reflect corrections due to observations and not due to simulated physical processes.
- Clipping is performed for all the other variables with zero lower bound (except for those related to temperature).
- In the case of the two-moment microphysics, the number concentrations of the hydrometeors are adjusted in order to preserve the particle size distributions. For zero prior hydrometeor contents and hence undefined prior number concentrations, the new concentrations are set consistent with assumptions in the Seifert/Beheng two-moment scheme.
- The IAU is not applied to the variables related to the snow cover and sea ice analyses for two reasons. Firstly, they have only an indirect influence on the atmospheric balance via the surface fluxes. Secondly, the analysis values and increments of ice and snow temperature and snow densitiy are not well defined in case of missing ice / snow in the analysis or first guess.

Name	Type	Definition / Purpose / Comments	Default
`itype_iau`	`INT`	To select the type of incremental analysis update (IAU): 0: no IAU is done (default) 1: IAU for all variables (except rel. to snow cover, sea ice) 2: IAU except for `QR`, `QS`, `QI`, `QG`, snow cover, sea ice (Introduced in Version 5.04d).	`0`
`peri_iau`	`REAL`	Period [s] over which IAU is applied. (Introduced in Version 5.04d).	`3600.0`

Parameters related to artificial initial and boundary conditions

The following parameters are only in effect if the control parameter `lartif_data` is set to `.TRUE.`. This parameter controls the basic type of the simulation: if it is set to `.TRUE.`,a run with artificial (or *idealized*) initial and boundary data is performed. If it is set to `.FALSE.`, a real-case simulation with initial and boundary data coming from a coarse-grid driving model or from an assimilation run is done.

If `lartif_data=.TRUE.` is chosen, there are parameters to control the type of lateral boundary conditions (relaxation, periodic, open) and to include/exclude metrical terms. Also, in this case another namelist `ARTIFCTL` will be read from the file `INPUT_IDEAL` and the namelist `GRIBIN` from file `INPUT_IO` has no effect and will be skipped. `ARTIFCTL` offers parameters for a wide variety of choices to configure idealized runs, such as definition of orography (idealized hills or pre user-provided ASCII files), specification of atmospheric and soil profiles for intitial and boundary conditions, and artificial convection triggers (*warm bubbles*, soil *hot spots*).

Name	Type	Definition / Purpose / Comments	Default	Depend.
`lartif_data`	LOG	If `.TRUE.`, the model runs with user-defined initial and boundary data. All Namelist input from `GRIBIN` is skipped and the corresponding parameters are not in effect. A Namelist group /ARTIFCTL/ has to be defined instead. If `.FALSE.`, the model expects initial and boundary data in Grib or NetCDF format. Control parameters for these files are contained in Namelist input group `GRIBIN`. (`lartif_data` has been renamed (named `lgen` before) and moved from group `GRIBIN` to `RUNCTL` in Version 4.8.)	`.FALSE.`	M
`l2dim`	LOG	Run a 2-dimensional model version. *To run the model in 2-d mode, the number of gridpoints in south-north direction must be set to* `je_tot = 5`.	`.FALSE.`	M
`lperi`	LOG	Eliminated in Version 4.17 and replaced by `lperi_x` and `lperi_y` (see below).	-	
`lperi_x`	LOG	Use periodic lateral boundary conditions in x-direction.	`.FALSE.`	
`lperi_y`	LOG	Use periodic lateral boundary conditions in y-direction.	`.FALSE.`	
`lcori`	LOG	moved to group /DYNCTL/ in Version 4.8		
`lmetr`	LOG	moved to group /DYNCTL/ in Version 4.8		
`lradlbc`	LOG	moved to group /DYNCTL/ in Version 4.8		

Also, a 1-dimensional set-up can be realized by setting the number of gridpoints in west-east direction to `ie_tot = 5` and choosing periodic lateral boundary conditions. Depending on the problem at hand, however, some additional modification of the dynamics code might be neccessary (e.g. to include a constant pressure gradient in terms of a geostrophic wind).

Note:
To generate artificial initial and boundary conditions for idealized model runs, the routines gen_ini_data (to specify initial data) and gen_bound_data (to specify boundary data) contained in the source file src_artifdata.f90 are used. The configuration of such idealized model runs can be done with the help of a large number of namelist parameters in the namelist ARTIFCTL in the file INPUT_IDEAL, which offers a wide variety of choices for orography, atmosphere and soil profiles for initial and boundary conditions and convection triggers. However, there is no unique way for this specification as it depends on the problem to be considered, and therefore it is possible to extend src_artifdata.f90 according to special needs.

The documentation of the namelist group ARTIFCTL would go beyond the scope of this User Guide, therefore an extra document will be provided for running idealized cases.

Parameters for parallel and sequential execution

This section contains parameters for the decomposition (parallelization) of the model domain and for generating additional information about timings for the different parts of the model run. Furthermore, the type of communication can be specified (see parameters ldatatypes, ncomm_type). How the different types of communications perform surely depends on the computer used. Therefore it would require some testing to find the optimal settings.

Name	Type	Definition / Purpose / Comments	Default	Depend.
nprocx	INT	Number of processors in the x-direction of the grid.	1	P
nprocy	INT	Number of processors in the y-direction of the grid.	1	P
nprocio	INT	Number of processors for GRIB I/O.	0	P
nprocio_radar	INT	Number of processors for doing asynchronous I/O within EMVORADO.	0	P
num_asynio _comm	INT	To choose the number of asynchronous I/O communicators for NetCDF. With several communicators it is possible to parallelize the output over the files to be written (the GRIBOUT namelists), so that several output files can be written simultaneously.	0	P
num_iope_ percomm	INT	To choose the number of asynchronous I/O processes per communicator for NetCDF I/O. With several processes per communicator it is possible to do a parallel writing of single files. This is only possible, if the parallel NetCDF library is available and the code has been compiled with the preprocessor directive -DPNETCDF.	0	P

Name	Type	Definition / Purpose / Comments	Default	Depend.
asynio_block _size	INT	To determine size of send buffer in netcdf_io.f90 (value given in MByte). This send buffer depends on the size of the COSMO domain. The default of 10 MB suffices for domain sizes of about 1100 × 1100 × 60 grid points. For larger domains, the size can be adjusted with this variable. Note that the maximum allocatable isend buffer (max_num_blocks) remains at 1 GB.	10	P
asynio_host _mem	INT	Total memory available for compute PE to store buffers (in units of GB).	1	P
asynio_pre-fetch_mem	INT	Total memory available for buffers on prefetch PE (in units of GB).	8	P
nboundlines	INT	Number of boundary lines at each side of a subdomain; specifies the size of the halo around each subdomain, where variables are updated by neighboring processors. Along nboundlines at the lateral boundaries of the total domain, the variables are set to boundary data from the driving host model. A minimum of 2 boundlines is required for the Leapfrog dynamical core. And a minimum of 3 boundlines is required for the Runge-Kutta dynamical core.	2	
ltime_proc	LOG	Eliminated; replaced by itype_timing in Version 4.4		
ltime_mean	LOG	Eliminated; replaced by itype_timing in Version 4.4		
itype_timing	INT	To specify, how a timing of the program should be done: 1: output hourly timings per task 2: output timings per task, summed up over all hours 3: output hourly mean values for all tasks 4: output mean values for all tasks, sum over all hours	4	
lreproduce	LOG	Ensures the computation of reproducible results in parallel mode.	.FALSE.	M
lreorder	LOG	If .TRUE., the numbering of the MPI-processes can be reordered for the cartesian MPI-communicator. This might give some performance benefits, depending on the computer platform used.	.FALSE.	P

Name	Type	Definition / Purpose / Comments	Default	Depend.
ldatatypes	LOG	Switch to choose between an explicit buffering (.FALSE.) or an implicit buffering (.TRUE.) using MPI-datatypes for the boundary exchange. Setting ldatatypes=.TRUE. might give some performance benefits, depending on the computer platform used.	.FALSE.	P
ltime_barrier	LOG	If .TRUE., some more barriers are called during the model run, to separate communications from computations in a very clean way. But without these barriers, the model will run faster.	.TRUE.	P
ncomm_type	INT	To choose a certain type of communication: 1: immediate send, blocking receive and wait on the sender side 2: immediate receive, blocking send and wait on the receiver side 3: using MPI_SENDRECV	1	

Parameters for diagnostic min/max model output

Name	Type	Definition / Purpose / Comments	Default	Depend.
hincmxt	REAL	Interval in hours for the period of validity of minimum and maximum values of the 2m-temperature (TMIN_2M and TMAX_2M), i.e. every hincmxt hours the corresponding arrays are reset to default values.	6.0	
hincmxu	REAL	Interval in hours for the period of validity of minimum and maximum values of the 10-m wind gusts (VMAX_10M), i.e. every hincmxu hours the corresponding array is reset to default values.	1.0	

Control parameters for debug purposes

Additional informations on the status of the model run can be obtained by increasing the debug level. However, this reduces the efficiency of the model run and blow up the standard ascii output enormously. Therefore these options are recommended to be used by model developers and/or in case of problems with the execution of a model run.

Name	Type	Definition / Purpose / Comments	Default	Depend.
idbg_level	INT	Selects the verbosity of ASCII output during a model run. The higher the value, the more debug output is written to standard output. To switch on/off debug output for special components, the additional flags ldebug_xxx have to be set to .TRUE. / .FALSE.	2	

Name	Type	Definition / Purpose / Comments	Default	Depend.
ldump_ascii	LOG	To save the ASCII-Files written during the run of the model to disk in every time step. This may cause a performance slow down on some machines and can be switched off. Then the files are only saved at the end of the run.	.TRUE.	
lprintdeb_all	LOG	In most cases, only the processor with ID = 0 will write the debug output. Setting the variable lprintdeb_all=.TRUE., all processors will print the debug output.	.FALSE.	
ldebug_dyn	LOG	To switch on/off the debug output for the dynamics.	.FALSE.	
ldebug_gsp	LOG	To switch on/off the debug output for the microphysics.	.FALSE.	
ldebug_rad	LOG	To switch on/off the debug output for the radiation.	.FALSE.	
ldebug_sso	LOG	To switch on/off the debug output for the SSO scheme. (Introduced in Version 5.04e).	.FALSE.	
ldebug_tur	LOG	To switch on/off the debug output for the turbulence.	.FALSE.	
ldebug_con	LOG	To switch on/off the debug output for the convection.	.FALSE.	
ldebug_soi	LOG	To switch on/off the debug output for the soil processes.	.FALSE.	
ldebug_io	LOG	To switch on/off the debug output the input and output.	.FALSE.	
ldebug_mpe	LOG	To switch on/off the debug output for the asynchronous I/O module mpe_io2.f90 (introduced in Version 4.27).	.FALSE.	
ldebug_dia	LOG	To switch on/off the debug output for the diagnostics.	.FALSE.	
ldebug_ass	LOG	To switch on/off the debug output for the assimilation.	.FALSE.	
ldebug_lhn	LOG	To switch on/off the debug output for the latent heat nudging.	.FALSE.	
ldebug_art	LOG	To switch on/off the debug output for the ART module (introduced in Version 4.9).	.FALSE.	
linit_fields	LOG	To switch on/off initialization of local variables (introduced in Version 4.9).	.FALSE.	

Miscellaneous control parameters

The following namelist variables have been added by the crCLIM group from the ETH Zürich. They provide the option, to perturb prognostic fields at start or on every time step during the simulation. Also, a plausibility check against the US standard atmosphere has

been introduced.
These variables have been introduced in Version 5.08.

Name	Type	Definition / Purpose / Comments	Default
itype_pert	INT	To add random perturbations to prognostic fields: 0: none (default) 1: at init 2: at every step	0
rperturb	REAL	Coefficient for adding perturbations.	1.0* rprec
l_t_check	LOG	if .TRUE., check plausibility of temperature against US standard atmosphere.	.FALSE.

Perturb Prognostic Fields:

With the perturbation coefficient, the method of perturbation is chosen:

(a) `rperturb` $\geq$ 0.0: relative perturbation with a specified magnitude;
necessary condition: `rperturb` $\geq$ `1.0 * rprecision`
Example: if `rperturb = 10`$^{-5}$ ($>$ `rprecision` for single and double precision) then the perturbation will be:

$$field = field * (1 + 10^{-5} * R),$$

where R is a random number from -1.0 to 1.0.

(b) `rperturb` $<$ 0.0: relative perturbation with a magnitude relative to precision
necessary condition: `rperturb` $\leq$ `-1.0`
Example: if `rperturb = -10.0` ($\leq$ -1.0 for single and double precision) then the perturbation will be:

$$field = field * (1 + 10.0 * 10^{-16} * R)$$

for double precision and

$$field = field * (1 + 10.0 * 10^{-7} * R)$$

for single precision where R is a random number from -1.0 to 1.0.

Note: The necessary conditions for (a) and (b) are checked in `src_setup.f90`, when reading the namelist variable `rperturb`. For single precision, `rprecision` is about 10^{-7}, for double precision it is about 10^{-16}.

Changes of Results: In case of `itype_pert` = 0, the results are not changed.

6.3 DYNCTL — Parameters for the Adiabatic Model

The namelist group DYNCTL contains control parameters for the numerical methods used to solve the thermo-hydro-dynamic model equations (the adiabatic part of the COSMO-Model) and to specify the boundary conditions and the numerical filtering for the COSMO solution, in particular close to the boundaries. The specifications for the parameters in DYNCTL are included in the file INPUT_DYN.

The namelist parameters of this group are described in the subsections

- main switches for the time integration,
- parameters for the semi-implicit time-integration scheme,
- parameters for the Runge-Kutta scheme,
- parameters for the lateral boundary conditions,
- horizontal diffusion,
- lower and upper boundary condition,
- additional numerical filters and
- spectral Nudging

Main switches for the time integration:

The main switch for the dynamics in the COSMO-Model is l2tls. Several options for the choice of numerical schemes are available with one of the schemes only. The more recent developments are implemented for the Runge-Kutta scheme (l2tls=.TRUE.) only.

In the first years of the COSMO-Model, the Leapfrog scheme was the only available dynamical core (l2tls = .FALSE.). But development for a 2 timelevel Runge-Kutta scheme started early. First, this scheme has been developed for very high resolutions below 4-5 km. But later it has been adapted also to the coarser resolutions. Now all applications can be run using the Runge-Kutta scheme and the Leapfrog scheme will no longer be supported.

Name	Type	Definition / Purpose / Comments	Default
l2tls	LOG	The default time-integration scheme is a 2 time-level Runge-Kutta scheme with time-split treatment of acoustic and gravity waves (l2tls = .TRUE.). Alternatively, a 3 time-level Leapfrog time-split scheme may be used (l2tls = .FALSE.).	.TRUE.
lcpp_dycore	LOG	This switch would activate the new code for the Runge-Kutta dynamical core, based on C++ template meta programming. But this code is not yet delivered for public use, since it is still under evaluation.	.FALSE.
leulag	LOG	To activate the EULAG dynamical core. To be able to run with the EULAG dynamical core, the COSMO-Model has to be compiled with the pragma -DEULAG and the ObjFiles.eulag and ObjDependencies.eulag have to be activated in the Makefile. Introduced in Version 5.07a.	.FALSE.

Name	Type	Definition / Purpose / Comments	Default
`l_euler_dynamics`	LOG	This variable allows to switch off/on the Euler dynamical core, i.e. advection, pressure gradient, divergence and buoyancy-terms for the u,v,w,T and p-equation. It does not affect tracers or the Coriolis force, i.e. even if it is set to `.FALSE.`, tracer advection and diffusion can take place and (if `lcori=.TRUE.`) Coriolis force can act. However, for the Leapfrog or semi-implicit dynamical core, the vertical advection of the dynamical variables is still active, even if `l_euler_dynamics=.FALSE.`.	`.TRUE.`
`lsemi_imp`	LOG	Instead of the standard Leapfrog time-split scheme, a Leapfrog scheme with semi-implicit treatment of acoustic and gravity waves may be used. This option works only with (`l2tls = .FALSE.`). *Caution: the semi-implicit scheme is not yet fully evaluated.*	`.FALSE.`
`lcond`	LOG	To include (`.TRUE.`) or exclude (`.FALSE.`) cloud water condensation and evaporation during the forecast.	`.TRUE.`
`lcori`	LOG	Run with Coriolis force (has been moved from group `/RUNCTL/` to here in Version 4.8.	`.TRUE.`
`lmetr`	LOG	Run with metric terms. If .FALSE., the spherical geometry of the grid is neglected and the Coriolis force is assumed to be constant on the model domain (f-plane approximation). (has been moved from group `/RUNCTL/` to here in Version 4.8.	`.TRUE.`
`lcori_deep`	LOG	To take coriolis terms ($\cos(\varphi)$) into account.	`.FALSE.`
`ladv_deep`	LOG	To use all metric advective terms.	`.FALSE.`
`l_satad_dyn_iter`	LOG	If set to `.FALSE.`, this disables the dynamical number of iterations for the saturation adjustment in the runge kutta dynamical core and a fixed number of 3 iterations is done then. (Introduced in Version 5.04a).	`.TRUE.`

Parameters for the semi-implicit time-integration scheme:

Name	Type	Definition / Purpose / Comments	Default
`ikrylow_si`	INT	Dimension of Krylow space for elliptic solver (GMRES).	20
`iprint_si`	INT	Selects output of print statistics of the elliptical solver. > 1: Print statistics in file `YUSOLVER`. $= 1$: No statistics are printed. < 1: Print statistics on the screen.	0
`maxit_si`	INT	Maximum number of iterations for elliptic solver.	200
`eps_si`	REAL	Precision limit for the convergence of the solution from the elliptic solver.	10^{-8}

Parameters for the Runge-Kutta scheme:

Name	Type	Definition / Purpose / Comments	Default
irunge_kutta	INT	Parameter to select a special Runge-Kutta scheme: 0: This option has been eliminated in Version 4.23 The dynamical core src_2timelevel.f90, associated with this option, has been removed from the code. 1: Use the dynamical core for the Runge-Kutta scheme from module src_runge_kutta.f90. 2: use the TVD-RK scheme (total variation diminishing)	1
irk_order	INT	Order of the Runge-Kutta scheme. Can be in the range from 1-3. The value 3 is recommended.	3
iadv_order	INT	Order of the horizontal advection scheme.	3
ieva_order	INT	Order of the explicit vertical advection scheme.	3
itheta_adv	INT	Option for using potential temperature as advected variable in the Runge-Kutta scheme: 0: Use $T\prime$ (perturbation temperature) for advection 1: Use $\theta\prime$ (perturbation potential temperature) 2: Use θ (full potential temperature)	0
ltadv_limiter	LOG	Switch to use a limiter for temperature advection (only for itheta_adv = 2)	.FALSE.
intcr_max	INT	Maximal integer courant number in courant-number independent advection	1
lva_impl_dyn	LOG	has been eliminated in Version 4.15. and replaced by y_vert_adv_dyn Switch to run with implicit vertical advection in the Runge-Kutta-scheme (.TRUE.) or not.	.TRUE.
y_vert_adv_dyn	CHAR	Choose an implicit vertical advection scheme for the dynamical variables u,v,w, T' and p' in the Runge-Kutta-scheme. This switch replaces and extends the old switch lva_impl_dyn. The following options can now be chosen: 'expl' explicit vertical advection, called in every RK substep. See also ieva_order. (corresponds to lva_impl_dyn=.FALSE.) 'impl2' 2nd order implicit vertical advection, called in every RK substep. (corresponds to lva_impl_dyn=.TRUE.) 'impl3' 3rd order implicit vertical advection, called at the end of the RK scheme	'impl2'

Name	Type	Definition / Purpose / Comments	Default
itype_fast_waves	INT	Parameter to select the treatment of fast waves: 1: Old scheme (from module fast_waves_rk.f90). 2: New scheme (from module fast_waves_sc.f90) with • proper weightings for all vertical discretizations, • divergence operator in strong conservation form, • (optionally) isotropic (fully 3D) divergence damping, • (optionally) Mahrer (1984) discretization. This is the new default (from Version 4.27 on).	2
lhor_pgrad_Mahrer	LOG	(Only for itype_fast_waves=2) The horizontal pressure gradient (i.e. in the u- and v-equations) is either calculated in the terrain-following system (.FALSE.) or by interpolation of p' to a horizontal plane (Mahrer (1984) MWR) (.TRUE.). (See also Baldauf (2013))	.FALSE.
divdamp_slope	REAL	(Only for itype_fast_waves=2) Exceed the theoretical slope stability criterion of the divergence damping. After a bug fix in Version 5.04a the range and the default value have changed: Possible range now is between 0.1 and 3.0. New Default is 1.0	~~20.0~~ 1.0
l_3D_div_damping	LOG	(Only for itype_fast_waves=2) The artificial divergence damping either acts only on the u- and v-equation (.FALSE.) or in a fully isotropic 3D manner (.TRUE.). The latter version is more time consuming (and less tested until now). (See also Baldauf (2013) and Baldauf (2014))	.FALSE.

Name	Type	Definition / Purpose / Comments	Default
itype_bbc_w	INT	Option for choosing bottom boundary condition for vertical wind: (before Version 4.12 this switch was named itype_lbc_w). For itype_fast_waves=1, allowed values are in the range 0-5: **0/1:** Runge-Kutta like method following iadv_order **2/3:** Second-order centered differences **4/5:** Fourth-order centered differences **0/2/4:** Quadratic extrapolation of horizontal wind to surface **1/3/5:** No extrapolation of horizontal wind to surface For itype_fast_waves=2, the settings 102, 104, 112, 114, 122 and 124 are possible. The meaning of the three digits 1ed are: **1** To make the value bigger than 100, to indicate the settings for itype_fast_waves=2. **e=0** No extrapolation of u and v to the bottom **e=1** Linear extrapolation of u and v to the bottom **e=2** Quadratic extrapolation of u and v to the bottom **d=2** Centered differences of 2nd order to discretize dh/dx and dh/dy **d=4** Centered differences of 4th order to discretize dh/dx and dh/dy	114
ldiabf_lh	LOG	Include diabatic forcing due to latent heat in the Runge-Kutta scheme.	.TRUE.
lsl_adv_qx	LOG	has been eliminated in Version 4.18. This is now set by the new parameter y_scalar_advect. (Switch to use semi-Lagrangian advection of the moisture variables in the Runge-Kutta-scheme (.TRUE.) or not.)	--
yef_adv_qx	CHAR	has been eliminated in Version 4.18. This is now set by the new parameter y_scalar_advect. (type of Euler-forward advection of the moisture variables (= 'vanLeer', 'PPM', 'Bott_2' 'Bott_4'))	--

Name	Type	Definition / Purpose / Comments	Default
`y_scalar_advect`	CHAR	String to specify the special horizontal advection type (only) for the Runge-Kutta dynamical core. This parameter replaces the 2 former parameters `yef_adv_qx` and `lsl_adv_qx`. The following table shows the possible settings of `y_scalar_advect` and how the old settings can be replaced (only for options that were also available before): `'SL3_MF'` SL-Advection with tricubic interpolation, multiplicative filling-option (`lsl_adv_qx=.TRUE.`, `yef_adv_qx='--'`) `'SL3_SC'` SL-Advection with simple clipping. `'SL3_SFD'` SL-Advection with tricubic interpolation, selective filling diffusion-option `'Bott2', 'Bott4'` Bott 2nd / 4th order finite-volume scheme (`lsl_adv_qx=.FALSE.`, `yef_adv_qx='Bott_x'`) `'Bott2_XYZYX', 'Bott4_XYZYX'` the same with modified sequence of Strang splitting. `'Bott2_Strang', 'Bott4_Strang'` Bott 2nd / 4th order finite-volume scheme with Strang splitting ('z-y-2x-y-z') `'Bott2_Strang_B', 'Bott4_Strang_B'` the same, but only works in the lowest `k_offset` levels of the atmosphere. `k_offset` is hardcoded with 5. `'BOTTDC2'` New option introduced in Version 5.05a. To choose Bott 2nd order finite-volume scheme with deformational correction and local time stepping. This method has similar stability characteristics as `'Bott2_Strang'` but is less expensive. `'vanLeer'` vanLeer type advection (`lsl_adv_qx=.FALSE.`, `yef_adv_qx='vanLeer'`) `'vanLeer_Strang'` vanLeer type advection with Strang splitting `'PPM'` PPM type advection (`lsl_adv_qx=.FALSE.`, `yef_adv_qx='PPM'`) `'PPM_Strang'` PPM type advection with Strang splitting `'MPDATA'` not yet available, work at MeteoCH `'NONE'` no tracer advection takes place NOTE: these strings are now case-insensitive (e.g. 'sl3_mF' or 'SL3_Mf' are both recognized).	`'BOTT2_STRANG'`

Parameters for the lateral boundary conditions:

Name	Type	Definition / Purpose / Comments	Default
`lw_freeslip`	`LOG`	Free-slip lateral boundary conditions are used for `w` by default in case of non-periodic Davies-type lateral boundaries. Otherwise, `w` must be specified along the boundaries. *It is recommended to use free-slip lateral boundary conditions for real-data simulations.*	`.TRUE.`
`lexpl_lbc`	`LOG`	eliminated in 5.1!: Now only the explicit formulation of the lateral boundary relaxation is possible. Chooses an explicit formulation of the lateral boundary relaxation.	`.TRUE.`
`crltau`	`REAL`	has been renamed in 5.1 to `crltau_inv`! To set the time factor for the explicit formulation of lateral boundary relaxation: $\tau_r = \texttt{crltau} \cdot \Delta t$.	`1.0`
`crltau_inv`	`REAL`	NOTE: Former name was `crltau` and the meaning really is $\texttt{crltau\_inv} = 1/\texttt{crltau}$. Factor for relaxation time: $1/\tau_r = \texttt{crltau\_inv} \cdot 1/dt$. The advantage of this replacement lies in the fact, that `crltau_inv=0` completely switches off the relaxation (this was not possible before).	`1.0`
`rlwidth`	`REAL`	If `lexpl_lbc = .TRUE.`, this parameter specifies the width of the relaxation layer in meters. It should be 10 to 15 times the grid mesh size in meters, but should not exceed 0.25 times the full domain size.	`85000.0`
`itype_outflow_qrsg`	`INT`	To choose the type of relaxation treatment for `qr, qs, qg`. (before Version 4.9 this switch was named `itype_lbcqx`). 1: `qr, qs, qg` are treated with the same lateral boundary relaxation as the other variables 2: no relaxation of `qr, qs, qg` is done at outflow boundary points	`1`
`itype_lbc_qrsg`	`INT`	To choose the type of lateral boundary treatment for `qr, qs, qg`, i.e., which values are used at the boundary zone. 1: A zero-gradient condition is used for `qr, qs, qg` 2: `qr, qs, qg` are set to 0.0 at the boundary zone 3: No presetting is done in the whole the boundary zone (must not be chosen for Leapfrog applications).	`1`
`lradlbc`	`LOG`	To run with radiative lateral boundary conditions (if `.TRUE.`), or with Davies conditions (if `.FALSE.`). (In Version 4.5 this switch has been moved from the group `RUNCTL` to `DYNCTL`).	`.FALSE.`

Name	Type	Definition / Purpose / Comments	Default
relax_fac	REAL	reduction factor for strength of lateral boundary relaxation (only if radiative lateral boundary conditions are used). (Switch has been introduced in Version 4.5).	0.01

Horizontal diffusion:

Name	Type	Definition / Purpose / Comments	Default
lhordiff	LOG	Main switch to include horizontal diffusion.	.TRUE.
itype_hdiff	INT	Parameter to select a scheme for horizontal diffusion. 1: Regular 4th-order linear scheme 2: Monotonic 4th-order linear scheme with orographic limiter	2
lhdiff_mask	LOG	has been eliminated in Version 4.12 and has been replaced by reduction factors for the interior and the boundary zone (see below).	-
l_diff_Smag	LOG	to switch on/off the Smagorinsky diffusion.	.TRUE.
hd_corr_u	REAL	eliminated in 4.12; replaced by hd_corr_u_[in,bd]	
hd_corr_u_bd	REAL	Reduction factor for the horizontal diffusion flux for u, v and w in the boundary zone of the domain.	0.25
hd_corr_u_in	REAL	Reduction factor for the horizontal diffusion flux for u, v and w in the interior zone of the domain.	0.25
hd_corr_t	REAL	eliminated in 4.12; replaced by hd_corr_t_[in,bd] Also, there are separate factors now for t and pp.	
hd_corr_t_bd	REAL	Reduction factor for the horizontal diffusion flux for t in the boundary zone of the domain.	0.0
hd_corr_t_in	REAL	Reduction factor for the horizontal diffusion flux for t in the interior zone of the domain.	0.0
hd_corr_p_bd	REAL	Reduction factor for the horizontal diffusion flux for pp in the boundary zone of the domain.	0.0
hd_corr_p_in	REAL	Reduction factor for the horizontal diffusion flux for pp in the interior zone of the domain.	0.0
hd_corr_q	REAL	eliminated in 4.12; replaced by hd_corr_q_[in,bd]	
hd_corr_q_[in,bd]	REAL	Renamed in 4.26 to hd_corr_trcr_[in,bd]	
hd_corr_trcr_bd	REAL	Reduction factor for the horizontal diffusion flux for tracers in the boundary zone of the domain.	0.0
hd_corr_trcr_in	REAL	Reduction factor for the horizontal diffusion flux for tracers in the interior zone of the domain.	0.0

Name	Type	Definition / Purpose / Comments	Default
hd_dhmax	REAL	Threshold value for the maximum height difference (m) between adjacent gridpoints in the orographic flux limiter. With increasing height difference, the diffusive fluxes are gradually decreased to become zero at hd_dhmax. Active only for itype_hdiff = 2. *Caution: The parameter* hd_dhmax *should be adjusted when changing the grid spacing.*	250.0
l_diff_cold_pools	LOG	Use targeted diffusion for cold pools. It should avoid unrealistic undershootings in the temperature field near the bottom, if the value of a grid point is more than thresh_cold_pool degrees Kelvin less than the average value of the surrounding.	.TRUE.
l_diff_cold_pools_uv	LOG	Use a different method for targeted diffusion for cold pools. Introduced in Version 5.05. For this scheme the threshold is computed over the 8 neighboring grid points instead of only 4 to detect a cold pool and a box-filter is applied on the detected thresholds. This increases the application domain for a detected cold pool from 1 to 9 grid points. To activate this new method, you also have to set l_diff_cold_pools = .TRUE.	.FALSE.
thresh_cold_pool	REAL	Threshold for targeted diffusion for cold pools. Reasonable values are thresh_cold_pool $\in [5.0, 20.0]$.	10.0

Lower and upper boundary condition:

Name	Type	Definition / Purpose / Comments	Default
lspubc	LOG	Option to use a sponge layer with Rayleigh damping in the upper levels of the model domain. If lspubc =.FALSE., a rigid lid upper boundary condition is used.	.TRUE.
itype_spubc	INT	To choose the type of Rayleigh damping in the upper levels. 1: chooses the damping against boundary fields; 2: chooses the damping against filtered forecast fields (can be chosen, if the boundary data is only available on frames. 3: New sponge layer near upper model boundary (according to Klemp et al.)	1
nfi_spubc2	INT	Number of applications of smoother for the determination of the large scale field used in the Rayleigh damping with itype_spubc=2	10
lrubc	LOG	Option to use a radiative upper boundary condition. *Not yet implemented.*	.FALSE.

Name	Type	Definition / Purpose / Comments	Default
rdheight	REAL	The bottom height (m) from which the Rayleigh sponge layer extends to the top of the model domain. A cosine damping profile with maximum damping at the top and zero damping at rdheight is assumed.	11000.0
nrdtau	INT	nrdtau*dt is the e-folding time for damping at the model top. Larger values of nrdtau result in weaker damping.	5
ldyn_bbc	LOG	To choose a dynamical bottom boundary condition.	.FALSE.

Additional numerical filters:

Name	Type	Definition / Purpose / Comments	Default
epsass	REAL	Value of the filter coefficient ε in the Asselin time-filter to damp the computational mode in the 3 time-level Leapfrog scheme (used only for l2tls=.FALSE.). *Caution: both too small ($\varepsilon < 0.01$) and too large values ($\varepsilon > 0.25$) may cause instabilities.*	0.15
alphaass	REAL	Weight for Williams 2009 modification to the Asselin time-filter ($0.5 <$ alphaass ≤ 1).	1.0
betasw	REAL	Value of the β-parameter used for time-weighting of the future values in the vertically implicit treatment of acoustic (sound) waves. $\beta = 0$ gives a time-centred average with no damping, $\beta = 1$ results in a fully implicit vertical scheme with strong damping of acoustic and gravity wave modes. *Slight positive off-centering is recommended to damp vertically propagating sound waves.*	0.40
betagw	REAL	Same as betasw, but used for gravity waves.	0.40
beta2sw	REAL	Same as betasw, but used in the p^*, T^* dynamics for sound waves.	0.40
beta2gw	REAL	Same as betagw, but used in the p^*, T^* dynamics for gravity waves.	0.40
xkd	REAL	Coefficient for divergence damping	0.1

Spectral Nudging

Name	Type	Definition / Purpose / Comments	Default
lspecnudge	LOG	to switch on/off spectral nudging	.FALSE.
yvarsn	CHAR	list of fields for spectral nudging (must be a subset of yvarbd	U, V
pp_sn	REAL	lowest pressure level for spectral nudging	850.0
alpha_sn	REAL	amplification factor for spectral nudging	0.05
isc_sn	INT	spectral nudging in i-direction	2
jsc_sn	INT	spectral nudging in j-direction	2
nincsn	INT	to define a time increment for calling the Spectral Nudging	1

Note: The spectral nudging has nothing to do with the Nudging, used as assimilation scheme in the COSMO-Model.

6.4 PHYCTL — Parameters for the Diabatic Model

The namelist group PHYCTL contains parameters controlling the physical parameterizations. All parameters are only active if the parameter lphys in Namelist /RUNCTL/ is set to .TRUE. in order to enable physical parameterizations. There is one main switch for each physical process to turn on/off this process and to activate additional parameters and sub-options for the corresponding parameterisation. The specifications for the parameters in /PHYCTL/ are included in the file INPUT_PHY.

The namelist parameters of this group are described in the subsections

- Grid-Scale Precipitation,
- Radiation,
- Moist Convection,
- Vertical turbulent diffusion,
- Surface layer fluxes,
- Soil Processes
- Subgrid Scale Orography

The switch to activate single column model runs is necessary in several schemes, therefore it is not assigned to a special group:

Name	Type	Definition / Purpose / Comments	Default
lscm	LOG	Main switch to run the single column model. Note, that up now to the single column model is not fully implemented in the COSMO-Model.	.FALSE.

The next switch works together with the saturation adjustment, which also occurs in several parts of the model.

Name	Type	Definition / Purpose / Comments	Default
htop_moist_proc	REAL	To switch off saturation adjustment in the higher stratosphere. It specifies the height, from which on saturation adjustment is switched off. This switch is only active for MESSy and is important for high atmosphere version only. Introduced in Version 5.07.	999999.0

Grid-Scale Precipitation

These parameters control the parameterization of grid scale precipitation. Note that the sub-grid scale precipitation is controlled by the convection parameterization.

Name	Type	Definition / Purpose / Comments	Default
lgsp	LOG	Main switch for including grid scale precipitation. If .TRUE., the model is run with a grid-scale precipitation scheme which computes the effects of precipitation formation on temperature and the prognostic moisture variables in the atmosphere (water vapour, cloud water, optionally cloud ice, rain, snow and graupel) as well as the precipitation fluxes of grid-scale rain and snow at the ground.	.TRUE.
itype_gscp	INT	Control parameter to select a specific scheme for grid scale parameterization: 1: Kessler-type warm rain parameterization scheme without ice-phase processes. 2: Kessler-type bulk formulation using cloud water, rain and snow. 3: Extension of the basic scheme with cloud water and cloud ice. 4: Graupel scheme with prognostic cloud water, cloud ice, and graupel.	3
lgsp_first	LOG	Switch to run grid scale precipitation at the beginning of time loop. Usually it is computed after the dynamics at the end of the time loop. This is only implemented for the new COSMO-ICON physics, where it can have performance benefits to run the whole parameterizations in one block.	.FALSE.
lsuper_coolw	LOG	Switch to activate effects of supercooled liquid water in the microphysics (only activated for graupel and cloudice scheme).	.TRUE.
lprogprec	LOG	This switch has been eliminated in Version 4.23. All simulations are now done with prognostic precipitation.	—
ltrans_prec	LOG	This switch has been eliminated in Version 4.23. All simulations are now done with prognostic precipitation.	—
ldiniprec	LOG	To switch on a diagnostic initialization of rain and snow, in case that no initial data are given.	.FALSE.

Up to COSMO-Model Version 3.6 the parameter itype_gscp was used to switch on/off a prognostic treatment of rain and snow, but only for the 2-timelevel Runge-Kutta scheme (for irunge_kutta = 0). Now only the specific kind of parameterization scheme can be chosen with itype_gscp and there are two additional parameters to control the prognostic precipitation.

Radiation

The following parameters control the Ritter-Geleyn radiation parameterization scheme of the COSMO-Model. Here, among others, the frequency of the calculation, the surface albedo, the background aerosols and the greenhouse gas concentration scenario can be specified.

Name	Type	Definition / Purpose / Comments	Default	Depend.
lrad	LOG	Main switch for including radiation. If .TRUE., the model is run with the radiation scheme which computes heating rates in the atmosphere (solar and thermal) and the energy balance (solar and thermal) at the ground. To save computing time, the radiation scheme will be called at certain time intervalls defined by nincrad or hincrad. Between two consecutive calls of the scheme, the radiative heating rates are kept constant.	.TRUE.	
nincrad	INT	Interval (in time steps) between two calls of the radiation scheme.	360	P,dt
hincrad	REAL	As nincrad, but time interval in hours. In general, an interval of 0.5 – 1 hour yields sufficient accuracy. *Can be specified alternatively to* nincrad.	0.0	P,dt
icldm_rad	INT	Parameter to select the mode of cloud representation (i.e. cloud cover, and cloud water and ice content) as input to the radiation parameterization. 0: No clouds are considered. 1: Only grid-scale clouds are considered. 2: Grid- and sub-grid scale water clouds are considered; cloud cover and water content are calculated according to a relative-humidity criterion (itype_wcld = 1) or a statistical closure (itype_wcld = 2). 3: Grid- and sub-grid scale (including convective) water and ice clouds are considered; cloud cover, water content and ice content are calculated according to the diagnosis based on a relative humidity scheme. (But note that also cloud ice is considered here!) 4: (at the moment) same as 3.	4	
lforest	LOG	Switch to choose the external parameter fields modifying the snow albedo of evergreen and deciduous forest. (itype_alb has to be chosen accordingly) Recommended: .TRUE.	.FALSE.	R
nradcoarse	INT	Has been eliminated in Version 5.08. If nradcoarse > 1, the radiation is computed on a coarser grid to save computation time. nradcoarse grid points in every horizontal dimension are combined. Maximal possible value is nboundlines.	1	P

Name	Type	Definition / Purpose / Comments	Default	Depend.
lradf_avg	LOG	Has been eliminated in Version 5.08. To average the radiative forcings when running on a coarser grid (nradcoarse $>$ 1). ???	.FALSE.	dx
lradtopo	LOG	To use topographic corrections for the radiation. (needs additional fields from int2lm).	.FALSE.	
nhori	INT	Number of sectors for the horizont used by the topographic corrections for the radiation.	24	P
ico2_rad	INT	Parameter to choose a special CO_2-concentration scenario: 0: constant CO_2-concentration of 360 ppm (default for weather prediction) 1: A1B scenario, only CO_2 is considered 2: A1B scenario, effective CO_2 is considered (i.e. CO_2 + CH_4 + N_2O 3: B1 scenario, only CO_2 is considered 4: B1 scenario, effective CO_2 is considered (i.e. CO_2 + CH_4 + N_2O 5: 6: 7: Scenario RCP2.6 8: Scenario RCP4.5 9: Scenario RCP6 10: Scenario RCP8.5 The following scenarios have been introduced in Version 5.09. 11: Scenario SSP1-1.9 12: Scenario SSP1-2.6 13: Scenario SSP2-4.5 14: Scenario SSP3-7.0 15: Scenario SSP3-7.0-lowNTCF 16: Scenario SSP4-3.4 17: Scenario SSP4-6.0 18: Scenario SSP5-3.4-over 19: Scenario SSP5-8.5	0	
lco2_stab	LOG	Option, to perform simulations with stabilized GHG forcings.	.FALSE.	
iy_co2_stab	LOG	(Only in effect, if lco2_stab = .TRUE.) To define the year, when GHG stabilization begins.	2001	
lemiss	LOG	Option, to use an external surface emissivity map (if set to .TRUE.). If lemiss is .FALSE. (default), a constant surface emissivity is assumed.	.FALSE.	

Name	Type	Definition / Purpose / Comments	Default	Depend.
`itype_aerosol`	`INT`	Switch to choose the type of aerosol map. 1: Tanre. (As it was up to Version 4.10 in the model.) Constant aerosol distributions are given for rural, urban, desert areas and the sea. 2: Tegen. A monthly aerosol climatology is used for sulfate drops, total dust, organic, black carbon and sea salt	`1`	
`itype_albedo`	`INT`	Switch to choose the type of solar surface albedo. This parameter has been introduced in Version 4.23. 1: surface albedo is a function of soiltype (method up to now and still default) 2: surface albedo is determined as a weighted mean value of two external fields for dry and for saturated soil. As a weighting factor the relative soil moisture saturation for the top soil layer is used. 3: A background albedo is prescribed by external fields, which give average values for every month. The actual surface albedo is determined by a linear interpolation between two monthly values, depending on the day. This is done in INT2LM. 4: The vegetation albedo is modified by forest fraction. Note, that for this type, also `lforest` has to be set to `.TRUE.`	`1`	

Moist Convection

These parameters specify the convection parameterization used. In particular at resolutions below 3 km the deep convection parameterization should be switched off, since this process is mainly a grid scale process for such high horizontal resolutions. It is possible to run with the shallow convection scheme then.

the model is run with a moist convection parameterization,

If a subgrid-scale (deep) convection scheme is used (`lconv=.TRUE.`), it computes the effect of moist convection on temperature, water vapour and horizontal wind in the atmosphere, and the precipitation rates of rain and snow at the ground. To save computing time, the convection scheme may not be called every time step, but at certain intervals defined by `nincconv`. Between two consecutive calls of the scheme, the convective tendencies and precipitation fluxes are kept constant.

There are two different schemes for deep convection: Tiedtke and (since version 5.04b) Tiedtke-Bechtold (the IFS convection scheme). A former implementation of Kain-Fritsch convection scheme has never been finished and was eliminated some time ago.

Name	Type	Definition / Purpose / Comments	Default
`lconv`	`LOG`	Main switch for including a subgrid-scale convection.	`.TRUE.`
`ltiedtke`	`LOG`	Eliminated ; replaced by itype_conv in Version 4.4	
`lkainfri`	`LOG`	Eliminated ; replaced by itype_conv in Version 4.4	
`lshallow`	`LOG`	Eliminated ; replaced by itype_conv in Version 4.4	
`itype_conv`	`INT`	To specify the type of convection parameterization 0: Tiedtke scheme 1: This option has been eliminated 2: Tiedtke-Bechtold scheme 3: Shallow convection based on Tiedtke scheme	`0`
`nincconv`	`INT`	Interval (in time steps) between two calls of the convection scheme.	`4`
`lconf_avg`	`LOG`	Switch to apply a horizontal smoothing of the convective forcings (moisture convergence, surface moisture flux and vertical velocity) prior to calling the convection scheme.	`.TRUE.`
`lcape`	`LOG`	Enables a CAPE-type closure within the Tiedtke convection scheme (not fully tested yet).	`.FALSE.`
`lctke`	`LOG`	Enables a turbulent kinetic energy closure within the Tiedtke scheme (not fully tested yet).	`.FALSE.`
`lconv_inst`	`LOG`	Switch to write instantaneous (`.TRUE.`) or aggregate (`.FALSE.`) values of `top_con` and `bas_con` to model output.	`.FALSE.`

Additional switches only for Tiedtke-Bechtold scheme

Note that these switches are only in effect if `itype_conv = 2`.

Name	Type	Definition / Purpose / Comments	Default
`lshallowconv_only`	LOG	To activate only the shallow convection from the Tiedtke-Bechtold scheme. Introduced in Version 5.04h.	`.FALSE.`
`ldetrain_conv_prec`	LOG	Detrain convective rain and snow for Tiedtke-Bechtold scheme. Introduced in Version 5.07.	`.FALSE.`
`icpl_aero_conv`	INT	Type of coupling between aerosols and convection scheme 0: No desription available yet. 1: No desription available yet. Introduced in Version 5.04b.	0
`icapdcycl`	INT	CAPE: diurnal cycle correction: 0: No CAPE diurnal cycle correction (IFS default prior to cy40r1, i.e. 2013-11-19) 1: CAPE - surface buoyancy flux (intermediate testing option) 2: CAPE - subcloud CAPE (IFS default starting with cy40r1) 3: Apply CAPE modification of (2) over land only, with additional restriction to the tropics Introduced in Version 5.04b.	0

Additional switches only for shallow convection scheme

Name	Type	Definition / Purpose / Comments	Default
`y_conv_closure`	CHAR	Type of shallow convection closure: Possible values are: `'standard'` : As it was up to now `'Boeing'` : Closure after Boeing. This is only possible for the shallow convection! Tiedtke and Tiedtke-Bechtold can only use the standard closure. Introduced in Version 5.04b.	`'standard'`

Vertical turbulent diffusion

There is one main switch to include turbulent diffusive fluxes, `ltur`, which activates both the parameterization of turbulent diffusion in the atmosphere and of surface-layer fluxes. A further selection of schemes can be done by the `itype_turb` and `itype_tran` parameters.

Since Version 5.04e (5.05) also the vertical diffusion can be computed in the physical parameterizations. Before this part has been computed in the dynamics. The new switch `itype_vdif` is used to choose when to run the vertical diffusion.

Since this version the new blocked version of the turbulence, which contains all ICON modifications, is running. As not all tests with the new ICON physics are satisfying, also the behaviour of the former implementation can be simulated, but not reproduced bit-identical. To choose the former settings for the turbulence, a namelist switch `loldtur` has been introduced in Version 5.04g.

Name	Type	Definition / Purpose / Comments	Default	Depend.
`ltur`	`LOG`	Main switch for including turbulent diffusion. If `.TRUE.`, the model is run with a turbulence parameterization which calculates the transport coefficients for momentum (K_m) and heat (K_h, also applied for water substance) in the atmosphere and the transfer coefficients at the ground (surface-layer). Over water, also the roughness length z_0 is computed. To save computing time, the exchange coefficients in the atmosphere may not be computed every time step, but at certain time intervals defined by `ninctura`.	`.TRUE.`	
`loldtur`	`LOG`	To switch to the *old* ijk-behaviour of the turbulence scheme: If `.TRUE.`, several internal switches in the turbulence code are set to simulate the *old* behaviour of the scheme. Introduced in Version 5.04g.	`.FALSE.`	
`ninctura`	`INT`	Increment in time steps for recalculating the transport coefficients K_m and K_h for vertical diffusion (`itype_turb = 1`) and for recalculating the stability functions S_m and S_h in case `itype_turb = 3` which are used to diagnose K_m and K_h from the predicted TKE. *Note: When running with the Leapfrog time-integration, `ninctura` should be an odd number to avoid using the same time family all the time.*	`1`	dt

Name	Type	Definition / Purpose / Comments	Default	Depend.
itype_vdif	INT	Parameter to choose the type of vertical diffusion calculation. -1: Turbulent diffusion is calculated by the previous routines of the dynamics. 0: Turbulent diffusion is computed just after the calculations of the turbulence model by using the same routine as in ICON from module turb_utilities.f90 applied to vertical profiles defined at (or interpolated to) horizontal mass positions. 1: Turbulent diffusion is computed at the end of the physics using collected explicit tendencies (and also using the same routines as in ICON). In this case the vertical diffusion is called separately for scalars (mass positions) and for winds (staggered wind positions). Introduced in Version 5.04a.	1	dx
lsflcnd	LOG	Run with a lower flux condition for vertical diffusion calculations. Introduced in Version 5.04a.	.TRUE.	
itype_turb	INT	Parameter to select the vertical turbulent diffusion parameterization. 1: Default diagnostic scheme. 2: Not used. 3: Prognostic TKE-based scheme, which includes effects from subgrid-scale condensation/evaporation. 4: Not used. 5/7: If a threedimensional turbulence scheme will be used.	3	dx
l3dturb	LOG	Switch to choose a 3D turbulence scheme (for (itype_turb = 5/7).	.FALSE.	dx
l3dturb_metr	LOG	To switch on/off the use of metric terms in the 3D turbulence.	.FALSE.	
lprog_tke	LOG	Switch to choose a prognostic treatment of TKE (for (itype_turb = 5/7).	.FALSE.	
lexpcor	LOG	Explicit corrections of the implicitly calculated turbulent heat and moisture fluxes due to effects from subgrid-scale condensation (only if itype_turb=3). *Should be set to .TRUE. to allow for a consistent treatment of diffusion coefficients and fluxes.*	.FALSE.	

Name	Type	Definition / Purpose / Comments	Default	Depend.
imode_turb	INT	Changed meaning in Version 5.04e; Mode of TKE equation in turbulence scheme. 0: Diagnostic equation. 1: Prognostic equation (default). 2: Prognostic equation (implicitely positive definite).	1	dt
icldm_turb	INT	Mode of cloud representation to take into account subgrid-scale condensation within the turbulence parameterization in case of itype_turb = 3. -1: Ignoring cloud water completely (pure dry scheme). 0: No clouds are considered. 1: Only grid-scale clouds are considered. 2: Grid- and sub-grid scale water clouds are considered; cloud cover and water content are calculated according to a relative-humidity criterion (itype_wcld = 1) or a statistical closure (itype_wcld = 2).	2	
ltmpcor	LOG	Consideration of thermal TKE-source in enthalpy budget (only if itype_turb=3).	.FALSE.	
lnonloc	LOG	Nonlocal calculation of vertical gradients used for turbulent diffusion (only if itype_turb=3).	.FALSE.	
lcpfluc	LOG	Consideration of fluctuations of the heat capacity of air (only if itype_turb=3).	.FALSE.	
lturhor	LOG	Switch to include horizontal turbulent diffusion. It has been eliminated in Version 4.10 and replaced by limpltkediff.	.FALSE.	
limpltkediff	LOG	Switch to include horizontal turbulent diffusion. Implemented in Version 4.10.	.TRUE.	
ltkesso	LOG	Switch, to calculate SSO-wake turbulence production for TKE. Changed default in Version 5.04e; Implemented in Version 4.10.	~~.FALSE.~~ .TRUE.	
ltkecon	LOG	Switch to consider convective buoyancy production for TKE. Implemented in Version 4.20.	.FALSE.	
ltkeshs	LOG	Switch to consider horizontal shear production for TKE. Introduced in Version 5.04a.	.FALSE.	

Name	Type	Definition / Purpose / Comments	Default	Depend.
itype_sher	INT	Type of shear production for TKE. Changed default and meaning of options in Version 5.04e; Implemented in Version 4.10. 0: Only vertical shear of horizontal wind. 1: Vertical shear of horizontal wind plus horizontal shear correction. 2: Vertical shear of horizontal wind, horizontal shear correction plus shear from vertical velocity.	~~1~~ 0	

Surface layer fluxes

These parameters control the calculation of the turbulent fluxes of latent and sensible heat.

Name	Type	Definition / Purpose / Comments	Default
`itype_tran`	INT	Main prameter to select a specific surface-layer parameterization. 1: Standard Louis-type scheme. 2: New TKE-based scheme including a laminar sub-layer.	2
`lfreeslip_sfc`	LOG	This functionality has been moved to the group `ARTIFCTL` in Version 4.17, but with a new name `lnosurffluxes_[m,h]`. (To switch on/off the surface momentum fluxes even if a turbulence scheme is used.)	--
		The following parameters apply only for the new surface layer scheme, i.e. only in case of `itype_tran = 2`.	
`imode_tran`	INT	Changed meaning and default in Version 5.04e; Mode of TKE-equation in transfer scheme: 0: Diagnostic equation. 1: Prognostic equation (default). 2: Prognostic equation (implicitely positive definite).	~~1~~ 0
`itype_wcld`	INT	Type of cloud water diagnosis. 1: Diagnosis based on a relative humidity scheme. 2: Diagnosis based on a statistical scheme.	2
`icldm_tran`	INT	New option (-1) and default in Version 5.04e; Mode of cloud representation to take into account subgrid-scale condensation within the new surface layer parameterization `itype_tran = 2`. -1: Ignoring cloud water completely (pure dry scheme). 0: No clouds are considered. 1: Only grid-scale clouds are considered. 2: Grid- and sub-grid scale water clouds are considered; cloud cover and water content are calculated according to a relative-humidity criterion (`itype_wcld = 1`) or a statistical closure (`itype_wcld = 2`).	~~0~~ 2
`itype_synd`	INT	Type of diagnosis of synoptic station values. 1: Interpolation of screen-level (2-m, 10-m) values using traditional similarity theory. 2: Interpolation of screen-level (2-m, 10-m) values based on profile relations used in the new surface-layer scheme.	1

Name	Type	Definition / Purpose / Comments	Default
`lprfcor`	`LOG`	Using the profile values of the lowest main level instead of the mean value of the lowest layer for surface flux calculations. *Not tested, should be set to* `.FALSE.`	`.FALSE.`

Soil Processes

These parameters control the parameterization of soil and vegetation processes. Mainly the configurations of the soil and vegetation model TERRA, of the lake model FLake and of the snow model are specified. The optimal configuration depends on the region investigated.

Note:
Additional external parameter fields are required by some of the methods used as additional input fields (e.g. for FLake). They can be provided using an appropriate version of INT2LM.

Name	Type	Definition / Purpose / Comments	Default	Depend.
lsoil	LOG	Main switch to include soil processes by running a soil model.	.TRUE.	M
lseaice	LOG	Main switch to activate the sea ice scheme at gridpoints covered by ice. **Note: This requires the temperature and the height of the sea-ice in the initial conditions**. These have to be provided by INT2LM from a driving model or from the assimilation cycle.	.TRUE.	R
llake	LOG	Main switch to include lake processes by running the lake model FLake. **This requires the lake fraction and lake depth in the initial conditions**. These have to be provided by INT2LM or from the data assimilation cycle.	.FALSE.	dx,R
lmulti_layer	LOG	Eliminated in Version 5.05 Switch to run the new multi-layer soil model TERRA_ML. If lmulti_layer = .FALSE, the standard soil model TERRA based on the two-layer EFR-method is used.	.TRUE.	
lmulti_snow	LOG	Eliminated in Version 5.11 Switch to run the multi-layer snow model. If lmulti_snow = .FALSE, the standard snow model within TERRA_ML is used. Introduced in Version 4.11. Remaining problems in COSMO.	.FALSE.	R
itype_snow	INT	Type of snow scheme to use (replaces lmulti_snow). 1: Traditional single layer scheme. 2: Multi-layer snow scheme by Ekaterina Machulskaya (former lmulti_snow). 3: Multi-layer snow scheme SNOWPOLINO.	1	

Name	Type	Definition / Purpose / Comments	Default	Depend.
itype_snow _start	INT	Type of cold start for SNOWPOLINO (only active for itype_snow = 3): 0: No cold start: fields for itype_snow = 3 have to be read from initial data file. 1: Hard cold start: all snow is removed and fields initiated to zero. 2: Soft cold start: state snow cover is estimated based on available analysis fields T_SNOW, W_SNOW and RHO_SNOW.	2	
nlgw	INT	Eliminated in Version 5.05 Only for lmulti_layer = .FALSE.: Number of prognostic soil water levels in the standard soil model TERRA, At present only nlgw = 2 and nlgw = 3 are implemented.	2	
lmelt	LOG	Switch to include melting processes within the soil when running the multi-layer soil model TERRA_ML. Recommended .TRUE. Removed in 5.0.	.FALSE.	R
lmelt_var	LOG	Sub-option for the lmelt parameter. IF lmelt_var = .TRUE, the soil freezing temperature is treated to be dependent on the water content, otherwise a constant freezing temperature is used. Recommended .TRUE. Removed in 5.0.	.FALSE.	
ke_soil	INT	Number of active soil layers in the new multi-layer soil model TERRA_ML. The total number of layers is ke_soil + 1.	7	
lstomata	LOG	Switch to use a minimum stomata resistance map for plants Introduced in Version 4.11.. If .FALSE. the standard value for mid latitudes is used. Addtional external parameter field RSMIN required.	.FALSE.	R
ke_snow	INT	Number of layers in the multi-layer snow model Introduced in Version 4.11.	2	
czml_soil(:)	REAL	Array to specify the main levels of the ke_soil + 1 soil layers (in meters). The default specification is / 0.005, 0.02, 0.06, 0.18, 0.54, 1.62, 4.86, 14.58 / *The maximum number of soil layers is limited to 20.*	see left	

Name	Type	Definition / Purpose / Comments	Default	Depend.
itype_trvg	INT	Parameter to select the type of parameterization for transpiration by vegetation. Recommended is 2. 1: Bucket version. 2: BATS version.	2	
itype_evsl	INT	Parameter to select the type of parameterization for evaporation of bare soil. Recommended is 2. 1: Bucket version. This option has been removed in Version 5.04e (with blocked version of TERRA). 2: BATS version. 3: Noilhan and Planton, 1989. 4: Resistance version.	2	
itype_root	INT	Parameter to select the type of root distribution Introduced in Version 4.11. Recommended is 1. 1: Uniform (Default) 2: Exponential (Arora & Boer, 2003)	1	
itype_canopy	INT	Parameter to select the type of canopy parameterization with respect to surface energy balance. Introduced in Version 5.06a. 1: Surface energy balance equation solved at the ground surface, canopy energetically not represented (Default) 2: Skin temperature formulation by Schulz and Vogel (2017), based on Viterbo and Beljaars (1995)	1	
cimpl	REAL	Stability parameter for the computation of the skin temperature. Introduced in Version 5.06a. Removed again in Version 5.09, as it is obsolete now due to modified formulation. The default value works well for COSMO domains with 2.2 km mesh size. For coarser grids and longer timesteps, lower values are appropriate, e.g. cimpl = 30.0 as in ICON. Reasonable values are cimpl $\in$ [4.0, 200.0].	120.0	
cskinc	REAL	Globally constant skin conductivity. Introduced in Version 5.06a. Reasonable values are cskinc $\in$ [10.0, 1000.0]. For cskinc < 0 an external parameter field SKC is read and used.	30.0	

Name	Type	Definition / Purpose / Comments	Default	Depend.
itype_heatcond	INT	Parameter to select the type of soil heat conductivity Introduced in Version 4.11. 1: Use average soil moisture 2: Take into account soil moisture/soil ice 3: Use a special (hardcoded) tuning factor to indirectly account for the impact of vegetation, which does not depend on soil moisture. (new option in Version 5.05).	1	
itype_mire	INT	Parameter to select a mire parameterization. Introduced in Version 5.06. 0: No parameterization 1: Mire parameterization following the approach from Yurova et al. (2014) Note: The mire parameterization only works with soil moisture dependent heat conductivity, i.e. itype_heatcond >= 2!	0	
czbot_w_so	REAL	to specify depth of bottom of last hydrological active soil layer	2.5	
itype_hydbound	INT	Parameter to select the type of hydraulic lower boundary Introduced in Version 4.11. 1: Allow for drainage but not diffusion 2: Rigid lid: not yet implemented 3: Ground water with drainage and diffusion	1	
itype_hydmod	INT	Type of soil water transport and ground water runoff: 0: Standard (default). 1: Modified formulation that allows ground water buildup (see Schlemmer et al. (2018)). Introduced in Version 5.08.	0	

Name	Type	Definition / Purpose / Comments	Default	Depend.
lsoil_init_fill	LOG	Filling of lowermost active soil layer at initialization. This determines if the groundwater should be filled up for the hydrologically passive layers or not. • If you start your run from a soil-moisture distribution derived using itype_hydmod=0, this variable should be set to .TRUE. • If you restart the model or initialize the model with a soil-moisture distribution derived using itype_hydmod=1 it should be set to .FALSE. Introduced in Version 5.08.	.FALSE.	
cwimax_ml	REAL	Maximum interception water content: cwimax_ml is used as a scaling parameter for maximum interception storage. With the default value the interception storage is turned off. In ICON 2.0.18 it is cwimax_ml=1.0E-4. (Introduced in Version 5.04e).	1.0E-6	
idiag_snowfrac	REAL	Method for diagnosis of snow-cover fraction: **1:** Old parameterization depending on soil water equivalent only. **2,20:** Advanced parameterization depending on snow depth, accounts also for vegetation and SSO. **3,30:** Similar to option 2, but somewhat less snow cover and limit over high vegetation. **4,40:** Same as option 3, but even more restrictive snow cover limit over high vegetation Extra for options 20, 30, 40: Artificially reduce snow-cover fraction in case of melting snow in order to reduce the ubiquitous cold bias in such situations. A minimum value to which the snow cover fraction is artificially reduced can be given with the namelist variable tune_minsnowfrac in the group /TUNING/. (Introduced in Version 5.04f).	1	

TERRA-URB

The urban-canopy land-surface scheme TERRA-URB has been implemented in the COSMO-Model. The TERRA-URB version used is 2.3, which is taken from the CCLM version COSMO-5.0-clm9. To run TERRA-URB, a *poor-man's* tile-approach has also been implemented.

For a documentation of TERRA-URB see the User Guide for TERRA-URB.

In the COSMO-Model, TERRA-URB can be controlled by the following namelist variables.

Name	Type	Definition / Purpose / Comments	Default
`lterra_urb`	LOG	To switch on/off the urban parameterization.	`.FALSE.`
`ntiles`	INT	Eliminated in Version 5.12. This only depends on `lterra_urb` and is set internally now. Number of tiles to be used. For TERRA-URB only 2 tiles can be used. If TERRA-URB is not running, `ntiles` must be set to 0.	0
`lurbfab`	LOG	To switch on/off the urban fabric (in a bulk approach).	`.TRUE.`
`itype_ahf`	INT	Type of anthropogenic heat: 0: no anthropogenic heat flux; 1: anthropogenic heat according to Flanner (2009).	1

`itype_kbmo_uf`	INT	Bluff-body thermal roughness length parametrization:
		0: standard SAI-based turbtran;
		1: Brutsaert/Kanda;
		2: Zilitinkevich.
			1

`itype_eisa`	INT	Type of evaporation from the impervious surface: 0: evaporation just like bare soil; 1: no evaporation; 2: PDF-based puddle evaporation (Wouters et al., 2015).	2

TERRA-URB requires information on the building material, for example the height of buildings or the area fraction of buildings in the urban tile. Up to now such information is not available and only default values are used for all cities. But work is in progress to derive corresponding external parameters.

INT2LM already implemented the possibility to read such external parameters, if available, and pass them on to the COSMO-Model. For every such field there now is a new namelist switch in the COSMO-Model, whether this field should be read from the initial data file or whether a real variable should be set to the whole domain.

If the following real namelist variables are set to -1.0, an external parameter field is read from the initial data set. Otherwise, the value given is taken for every grid point in the domain.

Name	Type	Definition / Purpose / Comments	Default
`curb_fr_bld`	`REAL`	Building area fraction.	`0.67`
`curb_h2w`	`REAL`	Street canyon height/width ratio.	`1.5`
`curb_h_bld`	`REAL`	Building height.	`15.0`
`curb_alb_th`	`REAL`	Effective termal albedo of building / pavement substrate.	`0.14`
`curb_alb_so`	`REAL`	Solar (short-wave) albedo of building / pavement substrate.	`0.10`
`curb_hcap`	`REAL`	Heat capacity.	`1.25E6`
`curb_hcon`	`REAL`	Material heat conductivity.	`0.777`
`curb_ahf`	`REAL`	Default annual-mean anthropogenic heat flux.	`-1.0`
`cskinc_urb`	`REAL`	Urban vegetation conductivity factor.	`1000.0`

Subgrid Scale Orography

These parameters control the parameterization of the effect of unresolved orography on the resolved scales of motion. The main effect is an additional surface drag over mountains.

Note:
Additional external parameter fields describing the subgrid scale orography are needed as input fields. They can be provided using an appropriate version of `INT2LM`.

Name	Type	Definition / Purpose / Comments	Default
`lsso`	`LOG`	Main switch to include subgrid scale orography processes	`.TRUE.`
`nincsso`	`INT`	Interval (in time steps) between two calls of the SSO scheme.	5

6.4.1 Different Configurations for the COSMO-ICON Physics

COSMO-ICON Parameterization Packages

One of the major developments in COSMO within the last years was the common COSMO-ICON Physics. The idea behind that development was to reduce the amount of maintenance and therefore to use the same source code packages in COSMO and in ICON. Besides the unification of the packages between COSMO and ICON, the technical modifications to use a blocked data structure were a big challenge.

With COSMO-Model Version 5.05 all physics packages are now in the blocked data format. The last two packages included from ICON, which are activated in COSMO Version 5.04e, are the turbulence and the surface schemes. The experiments done at DWD showed heterogeneous results, which might be due to the fact, that the turbulence and the surface scheme have significant differences to the schemes used up to now in COSMO. Alas the COSMO performance using the new ICON parts is worse than using the "old" schemes still in ijk-data format. One reason for this can be the fact that COSMO cannot take over all modifications from ICON, as for example the treatment of external parameters or the tile approach.

DWD therefore decided to choose a configuration, with which the behaviour of the former packages in ijk-data format can be reproduced rather closely. Of course, it is also possible to choose a configuration which is closer to the ICON like setup.

In the following we want to document these different setups: the conservative (or old-style COSMO physics) and the advanced (or new ICON style physics) approach. We start with an overview of the physical packages in COSMO and ICON, and which ones are shared. The main differences in these two setups are in the turbulence scheme and in the soil model TERRA. The different namelist settings for these schemes are discussed separately. Also in INT2LM there are necessary namelist settings to get the ICON-like behaviour of the physics.

Scheme	COSMO	ICON	blocked format implemented in COSMO Version
Microphysics	Prognostic water vapour, cloud water, ice, rain, snow, graupel (Doms, 2004; Seifert 2010)		5.01
Radiation	Ritter-Geleyn δ two-stream	RRTM	5.03
Subgrid Scale Orography	Lott and Miller (1997)		5.04e
Turbulence	prognostic TKE scheme (Raschendorfer)		5.04a
Surface Schemes	TERRA (Heise and Schrodin, 2002) FLake (Mironov) SeaIce (Mironov)		5.04e
Convection	Tiedtke (deep / shallow)	-	5.04b
	Tiedtke-Bechtold (only optional in COSMO)		

While the microphysics and the SSO scheme already were the same in ICON and in the old COSMO ijk-physics, there have been considerable modifications in ICON for the turbulence and the soil model TERRA. In the following we document how the namelist configuration can be chosen to either use a setup for reproducing the old ijk-schemes (as close as possible) or to use the new ICON modifications.

Namelist Settings for the Turbulence Scheme

In the turbulence scheme most modifications in ICON have been implemented as options. Therefore it is possible to choose between the former COSMO or the new ICON implementation. Several switches have been implemented for that. But some of these switches are not namelist variables but are hard-coded internally. In order to avoid re-compilation, a new namelist variable `loldtur` has been implemented in the COSMO namelist group `/PHYCTL/` to set these internal switches accordingly. In addition, some namelist variable have also to be chosen differently.

The following table lists the namelist variables in `/PHYCTL/` that have to be set for choosing the "OLD" (COSMO-like) configuration or the "NEW" (ICON-like) setup of the turbulence scheme.

/PHYCTL/	OLD	NEW	Explanation	Default
`loldtur`	`.TRUE.`	`.FALSE.`	If `.TRUE.`, internal switches are set to use former COSMO implementation of special algorithms in the turbulence scheme.	`.FALSE.`
`itype_vdif`	-1	-1	To choose the type of vertical diffusion. At the moment there are still problems using `itype_vdif` $\geq$ 0, which has been implemented with the new ICON version. Therefore only `itype_vdif = -1` can be recommended for both configurations (vertical diffusion done at the end of the dynamics).	1
`ltkeshs`	`.FALSE.`	`.TRUE.`	If `.TRUE.`, a separate horizontal shear production for TKE is considered. This was not the case in the OLD COSMO configuration.	`.TRUE.`
`itype_sher`	0	1	In the new code, `itype_sher = 1` would also consider a horizontal shear correction. As no horizontal shear production for TKE is computed (see above), we have to set `itype_sher = 0` for the OLD COSMO configuration (only vertical shear of horizontal wind).	0

/PHYCTL/	OLD	NEW	Explanation	Default
imode_tran	1	0	1 sets the mode of the TKE equation in the transfer scheme in the same way as it was done in the OLD COSMO implementation. ICON uses value 0.	0
icldm_tran	0	2	icldm_tran sets the mode of cloud representation in the transfer scheme. The old COSMO implementation did not consider clouds, this is chosen with value 0. In ICON (value 2) also a sub grid (turbulent) condensation is considered.	2

Also in the namelist group /TUNING/ there are some variables that are chosen differently in the ICON setup. At least compared to the settings that were used at DWD for COSMO-DE or for the former COSMO-EU. Of course the choice of these variables depend on special configurations and domains. The following table lists the variables that are now chosen at DWD for COSMO-D2 ("OLD") and for ICON ("NEW").

/TUNING/	OLD	NEW	Explanation	Default
tkhmin	0.4	0.75	Minimal diffusion coefficients [in m^2/s] for vertical scalar (heat) transport.	1
tkmmin	0.4	0.75	Minimal diffusion coefficients [in m^2/s] for vertical momentum transport.	1
rat_sea	20.0	7.0	Ratio of laminar scaling factors for heat over sea and land.	10.0
pat_len	500.0	750.0	Effective length scale of subscale surface patterns over land [in m].	100.0
tur_len	150.0	500.0	Asymptotic maximal turbulent distance [in m].	500.0
a_hshr	1.0	2.0	Length scale factor for separate horizontal shear production.	1.0
c_soil	1.0	1.75	Surface area density of the (evaporative) soil surface.	1.0

Namelist Settings for TERRA

Also in TERRA there have been considerable changes in the ICON version compared to the OLD COSMO implementation. Most of these modifications are not optional, so there is no way to mimic the OLD behaviour. Only for some algorithms, which already had several options, new ways have been implemented, which could be chosen now. Note that not all ICON options can be used in COSMO due to a different treatment of external parameters and the ICON tile approach.

The following table lists the namelist variables with which new ICON options could be chosen. The column "OLD" again lists the COSMO-D2 settings from DWD, which correspond to the former configuration of TERRA, while the column "NEW" lists the ICON settings. DWD did not make enough tests (independent from the turbulence settings) to judge the performance of the ICON settings in COSMO. But everybody is free to do so!

/TUNING/	OLD	NEW	Explanation	Default
itype_evsl	2	4	Type of parameterization of bare soil evaporation. Option 4 is the new implementation by Jan-Peter Schulz. Note that option 1 is no more available.	2
itype_heatcond	1	3	Type of soil heat conductivity: Option 1 uses an average soil moisture. New option 3 uses a special (hardcoded) tuning factor to indirectly account for the impact of vegetation, which does not depend on soil moisture.	1
itype_root	1	2	Type of root distribution: Option 1 assumes a uniform distribution used in the OLD COSMO configuration. Option 2, which assumes an exponential distribution, is used in ICON. Please note, that if you want to use option 2, you also have to use itype_rootdp = 4 in the INT2LM!	1
cwimax_ml	1.0E-6	0.0005	cwimax_ml is used as a scaling parameter for maximum interception storage. In the OLD COSMO implementation this was a constant with value 1.0E-6. Now it is a namelist variable. ICON uses a value of 5.0E-4 = 0.0005.	1.0E-6
idiag_snowfrac	1	20	Method for diagnosis of snow-cover fraction. (Further explanation necessary).	1

/TUNING/	OLD	NEW	Explanation	Default
lstomata	.FALSE.	.TRUE.	Minimum stomata resistance: if lstomata = .TRUE., an external parameter is read for the minimum stomata resistance. Note that you also have to run INT2LM then with lstomata = .TRUE.	.FALSE.

Namelist Settings for INT2LM

Some of the different settings in COSMO and ICON come from different treatment of external parameters. While ICON computes the external parameters within the model, COSMO gets its external parameters from INT2LM. For a comparable treatment in COSMO and ICON the following namelist variabes in INT2LM could be changed (in the group /CONTRL/).

The following table lists the external parameters, the namelist variable to control their behaviour and the values for the OLD COSMO and the new ICON settings together with some short explanations.

Namelist	OLD		NEW	
itype_ndvi works on: PLCOV LAI	0	Plant cover and leaf area index for the COSMO-Model and for a special day are produced by using the data sets for vegetation and for rest and modify these with an annual cycle.	1	Plant cover and leaf area index for the COSMO-Model and for a special day are produced by using only the data set for vegetation and an averaged ndvi ratio. This ndvi ratio is computed as a weighted mean between monthly mean values, which are taken from the external parameter data set for the COSMO-Model.
itype_rootdp works on ROOTDP	0	Input from external parameter for the COSMO-Model is taken and modified with an annual cycle.	4	Input from external parameter for the COSMO-Model is taken without modifications.

Namelist	OLD		NEW	
itype_aerosol works on AER_SO4 AER_DUST AER_ORG AER_BC AER_SS	1	No additional Fields for aerosol types are read from the external parameters. The COSMO-Model has to run with constant values for the aerosol distribution on rural areas, urban areas, desert areas or the sea (Tanre aerosol climatology).	2	Additional external parameters for the aerosol types of sulfate, mineral dust, organic, black carbon and sea salt are read as monthly mean values. Actual values for the current day are computed and given to the COSMO-Model (Tegen aerosol climatology).
lstomata works on RSMIN	.FALSE.	No external parameter is taken. A constant value for all grid points is determined in the COSMO-Model.	.TRUE.	Take a map from the external parameters for the minimum stomata resistance of plants.
lemiss works on EMIS_RAD	.FALSE.	No external parameter is taken. A constant value for all grid points is determined in the COSMO-Model.	.TRUE.	Take a map from the external parameters for the thermal radiative surface emissivity.

NOTE: Usage of SSO_STDH and SSO_SIGMA:

The new turbulence scheme now needs the external parameter SSO_STDH and the new scheme for TERRA needs SSO_SIGMA. Therefore the namelist variable lsso should be set to .TRUE. in any case when running INT2LM.

6.5 TUNING — Parameters for tuning dynamics and physics

The namelist group `TUNING` contains parameters that can be used to tune special components and packages of the parameterizations and dynamics. This namelist group is intended to be used mainly by the EXPERTS. The parameters can be used to adapt the behaviour of the model to special regions, applications and resolutions. The specifications for the parameters in `TUNING` are included in the file `INPUT_ORG`.

In the following tables some limitations and ranges for meaningful values of the different parameters are given in the form: values $\in [0, 1)$ where $[,]$ stands for *including* and $(,)$ for *excluding* a value.

Turbulence

Name	Type	Definition / Purpose / Comments	Default	Dep.
`tkesmot`	`REAL`	Time smoothing factor for TKE to reduce the time variability of the diffusion coefficient. Formel ???. Should be chosen as small as possible. (`tkesmot` $\in [0, 2]$)	`0.15`	???
`wichfakt`	`REAL`	Eliminated in Version 5.05; Vertical smoothing factor for explicit vertical diffusion coefficients. (`wichfakt` $\in (0, 0.5]$)	`0.0`	dt
`securi`	`REAL`	Eliminated in Version 5.05; Security factor for maximal diffusion coefficients for explicit vertical diffusion. (`securi` $\in (0, 1]$)	`0.85`	dz
`tkhmin`	`REAL`	Minimal diffusion coefficients for heat active in stable BL conditions. Changed default in Version 5.04e; (`tkhmin` $\in [0, 2]$)	~~`0.4`~~ `0.75`	dz,R
`tkmmin`	`REAL`	Minimal diffusion coefficients for momentum active in stable BL conditions. Changed default in Version 5.04e; (`tkmmin` $\in [0, 2]$)	~~`0.4`~~ `0.75`	dz,R
`z0m_dia`	`REAL`	Typical roughness length for a Synop station, which is used for the interpolation of screen-level values of the 10-m wind (instead of using the actual roughness length at the gridpoint) and T_{2M}. (`z0m_dia` $\in [0.001, 10]$). Modifies the T_2M calculation.	`0.2`	
`rat_lam`	`REAL`	Ratio of laminar boundary layer thickness for water wapour and sensible heat. Higher values allow to ... the Bowen ratio. (`rat_lam` $\in [0.1, 10]$)	`1.0`	
`rat_can`	`REAL`	Scaling factor for the calculation of the canopy height affecting the diagnostics of T_{2M} (`rat_can` $\in [0, 10]$). Removed in 5.0.	`1.0`	

Name	Type	Definition / Purpose / Comments	Default	Dep.
rat_sea	REAL	Ratio of laminar scaling factors for heat over sea and land. Changed default in Version 5.04e; (rat_sea $\in [1, 100]$).	~~20.0~~ 10.0	
pat_len	REAL	Length scale (m) of sub-scale surface patterns over land. Changed default in Version 5.04e; (pat_len $\in [0, 10000]$). Removed in 5.0.	~~500.0~~ 100.0	dx
tur_len	REAL	Maximal turbulent length scale (m). (tur_len $\in [0, 10000]$)	500.0	
c_lnd	REAL	Surface-area index of gridpoints over land (excluding leaf-area index) used in ???. (c_lnd $\in [1, 10]$). Formula: ???.	2.0	
c_sea	REAL	Surface-area index of gridpoints over sea used in ???. (c_sea $\in [1, 10]$). Formula: ???.	1.5	
c_soil	REAL	Surface-area index of the evaporating fraction of gridpoints over land used in ???. (c_soil $\in [0, \text{c\_lnd}]$). Formula ???.	1.0	
e_surf	REAL	Exponent to get the effective surface area used in ??? (Why needed additionally to c_{xxx}). (e_surf $\in [0.1, 10]$).	1.5	
rlam_heat	REAL	Scaling factor for the thickness of the laminar boundary layer for heat. Formula: ???. (rlam_heat $\in [0.1, 10]$)	1.0	
rlam_mom	REAL	Scaling factor for the thickness of the laminar boundary layer for momentum. Formula: ???. (rlam_mom $\in [0, 1]$)	0.0	
a_heat	REAL	Factor for turbulent heat transport. (a_heat $\in [0.01, 100]$) Removed in 5.0.	0.74	
a_mom	REAL	Factor for turbulent momentum transport. (a_mom $\in [0.01, 100]$) Removed in 5.0.	0.92	
a_hshr	REAL	Length scale factor for separate horizontal shear production of TKE. Changed default in Version 5.04e; Introduced in Version 4.10.	~~0.2~~ 1.0	
a_stab	REAL	Length scale factor for the stability correction. Introduced in Version 4.10.	0.0	
d_heat	REAL	Factor for turbulent heat dissipation. (d_heat $\in [0.01, 100]$). Removed in 5.0.	10.1	
d_mom	REAL	Factor for turbulent momentum dissipation. (d_mom $\in [0.01, 100]$). Removed in 5.0.	16.6	
c_diff	REAL	Factor for turbulent diffusion of TKE. (c_diff $\in [0, 10]$).	0.2	

Name	Type	Definition / Purpose / Comments	Default	Dep.
clc_diag	REAL	Cloud cover at saturation in statistical cloud diagnostic. (clc_diag $\in (0,1)$)	0.5	
q_crit	REAL	Critical value for normalized over-saturation. (q_crit $\in [1,10]$)	4.0	

Surface Schemes

Name	Type	Definition / Purpose / Comments	Default	Dep.
crsmin	REAL	Minimum value of stomatal resistance (used by the BATS approach for vegetation transpiration, itype_trvg=2). (crsmin $\in [50,200]$)	150.0	
tune_minsnowfrac	REAL	In case of melting snow the snow cover fraction is artificially reduced to this minimum value (in combination with idiag_snowfrac = 20/30/40 in group /PHYCTL/). (Introduced in Version 5.05).	0.125	
l_g	REAL	Scaling for ground water runoff for itype_hydmod=1. (Introduced in Version 5.08).	2.5	
soilhyd	REAL	Multiplication factor for hydraulic conductivity and diffusivity. Tested range soilhyd $\in [0.1, 10.0]$. (Introduced in Version 5.08).	1.0	
fac_rootdp2	REAL	Multiplication factor for prescribed root depth. Tested range fac_rootdp2 $\in [0.5, 5.0]$. (Introduced in Version 5.08).	1.0	

SSO Scheme

Name	Type	Definition / Purpose / Comments	Default	Dep.
gkdrag	REAL	Scaling factor for gravity wave drag in SSO scheme. Reasonable values: gkdrag $\in [0.0, 1.0]$. With a value of 1, the gravity wave drag in the SSO scheme is switched on. When reducing the value, the strength of the gravity wave drag is reduced. With a value of 0, the gravity wave drag in the SSO scheme is switched off.	0.075	

Name	Type	Definition / Purpose / Comments	Default	Dep.
gkwake	REAL	Scaling factor for low-level flow blocking in SSO scheme. Reasonable values: gkwake $\in [0.0, 1.0]$. With a value of 1, the low-level flow blocking in the SSO scheme is switched on. When reducing the value, the strength of the low-level flow blocking is reduced. With a value of 0, the low-level flow blocking in the SSO scheme is switched off.	0.5	

Radiation

Name	Type	Definition / Purpose / Comments	Default	Dep.
radqc_fact	REAL	Portion of grid scale qc + qr *seen* by the radiation. Reasonable values: radqc_fact $\in [0.5, 1.0]$. (Introduced in Version 5.04c).	0.5	
radqi_fact	REAL	Portion of grid scale qi *seen* by the radiation. Reasonable values: radqi_fact $\in [0.5, 1.0]$. (Introduced in Version 5.04c).	0.5	
radqs_fact	REAL	Portion of grid scale qs *seen* by the radiation. Reasonable values: radqs_fact $\in [0.5, 1.0]$. (Introduced in Version 5.06b).	0.5	
radqg_fact	REAL	Portion of grid scale qg *seen* by the radiation. Reasonable values: radqg_fact $\in [0.5, 1.0]$. (Introduced in Version 5.06b).	0.5	
uc1	REAL	Tuning variable for computing the rate of cloud cover in saturated conditions. (Introduced in Version 5.08).	0.8	

Microphysics

Name	Type	Definition / Purpose / Comments	Default	Dep.
qc0	REAL	Cloud water threshold for autoconversion.	0.0	
qi0	REAL	Cloud ice threshold for autoconversion.	0.0	
mu_rain	REAL	Shape parameter of rain drop size distribution. Reasonable values: mu_rain $\in [0.0, 5.0]$. Introduced in Version 4.5. ATTENTION: In Version 4.21 the default value has been changed from 0.5 to 0.0!	0.0	
rain_n0_factor	REAL	To reduce the evaporation of rain drops. Reasonable values: rain_n0_factor $\in [0.0, 1.0]$. Introduced in Version 4.14.	1.0	

Name	Type	Definition / Purpose / Comments	Default	Dep.
`cloud_num`	REAL	Cloud droplet number concentration.	5.0 E+08	
`v0snow`	REAL	Factor in the terminal velocity for snow. This was a local variable in the subroutines `hydci_pp` and `hydci_pp_gr` before, but had different values. To reproduce the results from the former version, the following values have to be set: `itype_gscp = 3` and LF: 15.0 `itype_gscp = 3` and RK: 25.0 `itype_gscp = 4` and RK: 20.0 Introduced in Version 4.14.	25.0	

Convection

Name	Type	Definition / Purpose / Comments	Default	Dep.
`entr_sc`	REAL	Mean entrainment rate for shallow convection. Introduced in Version 4.5.	0.0003	
`thick_sc`	REAL	Limit for convective clouds to be shallow (in Pa). Recommended values for `thick_sc`: `thick_sc` $\in [10000.0, 45000.0]$ Introduced in Version 4.18.	25000.0	
`thick_dc`	REAL	Threshold of cloud thickness for precipitating deep convection mode in Tiedtke deep convection parameterization. Reasonable values: `thick_dc` $\in [5000.0, 45000.0]$. (Introduced in Version 5.04c).	20000.0	
`cmfctop`	REAL	Relative mass flux above level of non-buoyancy for both shallow and deep convection parameterizations. Reasonable values: `cmfctop` $\in [0.1, 0.33]$. (Introduced in Version 5.04c).	0.33	
`cprcon_dc`	REAL	Conversion rate cloud water to precipitation for deep convection parameterization. Reasonable values: `cprcon_dc` $\in [0.0001, 0.001]$. (Introduced in Version 5.04c).	0.0002	
`Tmpmin_dc`	REAL	Minium temperature of the mixed-phase temperature range for deep convection parameterization [K]. Reasonable values: `Tmpmin_dc` $\in [230.0, 273.0]$. (Introduced in Version 5.04c).	250.16	

Additional switches only for Tiedtke-Bechtold scheme

These switches have been introduced with version 5.04b, when the Tiedtke-Bechtold scheme has been implemented.

Name	Type	Definition / Purpose / Comments	Default
tune_capdcfac_et	REAL	Fraction of CAPE diurnal cycle correction applied in the extratropics.	0.0
tune_rhebc_land	REAL	Relative humidity threshold for onset of evaporation below cloud base over land.	0.75
tune_rhebc_ocean	REAL	Relative humidity threshold for onset of evaporation below cloud base over sea.	0.85
tune_texc	REAL	Excess value for temperature used in test parcel ascent.	0.125
tune_qexc	REAL	Excess fraction of grid-scale QV used in test parcel ascent.	0.0125
tune_rcucov	REAL	Convective area fraction.	0.05
tune_entrorg	REAL	Entrainment parameter for deep convection valid at dx=20 km.	0.001825
tune_rhebc_land_trop	REAL	Relative humidity threshold for onset of evaporation below cloud base over land in the tropics.	0.7
tune_rhebc_ocean_trop	REAL	Relative humidity threshold for onset of evaporation below cloud base over sea in the tropics.	0.8
tune_rcucov_trop	REAL	Convective area fraction in the tropics.	0.05

Additional switches have been introduced with version 5.07:

Name	Type	Definition / Purpose / Comments	Default
tune_capdcfac_tr	REAL	Fraction of CAPE diurnal cycle correction applied in the tropics.	0.5
tune_rprcon	REAL	Coefficient for conversion of cloud water into precipitation.	1.4E-3
tune_rdepths	REAL	Maximum shallow convection depth (Pa).	2.0E4

6.6 DIACTL — Parameters for Diagnostic Output

DIACTL contains parameters to generate gridpoint and control ASCII output. These parameters are only in effect if the main switch ldiagnos in RUNCTL is set to .TRUE.. In this case, some additional time integrated fields – TDIV_HUM (vertically integrated divergence of specific humidity) and AEVAP_S (surface moisture flux) – are also calculated to allow for a mass-budget calculation based on GRIB-output. Currently, the following ASCII output can be generated:

- Grid point output (meteographs) in a specific form;
  results are written to files M_*stationname*.
- Control output for a quick-look monitoring of the model run;
  results are written to files YUPRMASS (for mass variables) and YUPRHUMI (for humidity variables), resp.

Note: Up to model version 3.4, two addtional forms of ASCII output could be generated: Diagnostics for various subdomains and differences between predicted and and boundary fields. Since these diagnostics can be easily calculated from GRIB-output, they will no longer be retained.

See Part VI of the COSMO-Model Documentation (*Model Output and Data Formats for I/O* for a further description of the ASCII output.

Parameters for control ouput

For control output, a number of variables are computed and written to the ASCII files YUPRMASS (for mass variables) and YUPRHUMI (for humidity variables). These variables are domain averages of quantities like surface pressure, surface pressure tendency, kinetic energy, dry static energy, moist static energy, cloud water content, absolute vertical velocity at certain levels, precipitation rates and accumulated precipitation, and domain maxima of absolute horizontal and vertical velocity. This allows for a quick-look control of the model run.

Name	Type	Definition / Purpose / Comments	Default
n0meanval	INT	Number of time step for the first call of control output. If n0meanval > nstop, no control output will be computed.	0
nincmeanval	INT	Interval (in time steps) between two calls of the control calculations.	10

Ouput for COSMO Testsuite

Name	Type	Definition / Purpose / Comments	Default
ltestsuite	LOG	To activate additional ASCII output, which is used and evaluated when running the COSMO Technical Testsuite.	.FALSE.

Control parameters for grid point output

Only one type of grid point output can be chosen, either the short meteographs (`lgpshort = .TRUE.`), the long grid point output (`lgplong = .TRUE.`) or the special grid point output for the physics (`lgpspec = .TRUE.`). The parameter `nmaxgp` (=100 per default) in the module `src_gridpoints.f90` gives the maximum number of gridpoints for which calculations can be done. For every grid point, the output is stored for all time steps to a file `M_`*stationname*, where *stationname* is a name that can be specified by namelist input. If no name is specified, the geographical coordinates are used instead.

Name	Type	Definition / Purpose / Comments	Default
`n0gp`	`INT`	Time step of the first grid point calculation. *Alternatively:*	`0`
`h0gp`	`REAL`	Same as `n0gp`, but time in hours.	`0.0`
`nincgp`	`INT`	Time interval (in time steps) between two calls for grid point output. *Alternatively:*	undefined
`hincgp`	`REAL`	Same as `nincgp`, but time in hours.	`0.0`
`station list_tot(:)`	`TYPE`	The list of stations for grid point output can be specified with a derived type declaration. The components of the type are: - `igp (INT)`: i-index - `jgp (INT)`: j-index - `rlatgp_tot (REAL)`: geographical latitude - `rlongp_tot (REAL)`: geographical longitude - `ystation_name (CHAR)`: name of the station	undefined
`lgpshort`	`LOG`	Calculate and print a short form of grid point output (1 line/step).	`.FALSE.`
`lgplong`	`LOG`	Calculate and print a long form of grid point output (about 1 page/step).	`.FALSE.`
`lgpspec`	`LOG`	Calculate and print a special form of grid point output for physics diagnostics.	`.FALSE.`

Computation of special diagnostics

Since Version 4.8 it is possible to choose different variants for special diagnostic computations.

Name	Type	Definition / Purpose / Comments	Default
itype_diag_t2m	INT	To specify the method for computing the 2m temperature 1: Computation with an exponential canopy profile, but with a diagnostic Prandtl layer interpolation even for scalars, using an adopted canopy layer resistance. 2: Computation with exponential canopy profile	1
itype_diag_gusts	INT	To specify the method for computing the maximal wind gusts 1: Dynamical gust derived from lowest model layer 2: Dynamical gust derived from 30 m 3: Dynamical gust derived after Brasseur 4: Similar to 1, but here the gust factor depends weakly on the mean wind speed at 10 meters. 5: Empirically trained statistical linear model. (Introduced in Version 5.06.)	1
lhailcast	LOG	To switch on/off a hailcast diagnostic. This implementation is taken from WRF; see Adams-Selin and Ziegler (2016) for further documentation. Introduced in Version 5.07a.	.FALSE.
ninchail	INT	To control the call frequency. It is recommended to call the hailcast diagnostic every 2 to 5 minutes, i.e. for a 10 second time step, set ninchail between 12 and 30. Introduced in Version 5.07a.	30
ldursun_mch	LOG	Use the Swiss thresholds for sunshine duration. This switch only determines the threshold and smooth transition of the sunshine duration. Switzerland is the only country which uses 200 W/m2 as a threshold instead of the 120 W/m2 which are defined by the WMO. Introduced in Version 5.07a.	.FALSE.

Computation of surface and volume integrals

Since Version 3.23, the COSMO-Model offers the possibility to calculate volume integrals of arbitrary fields over an arbitrary cuboid defined in the numerical (i.e. terrain-following) grid. Also surface integrals of arbitrary vector fields ('fluxes') over the surface of this cuboid can be computed. With the following set of namelist variables, the cuboid can be defined (in terms of grid point indices of the total domain) and these integrations can be switched on/off.

Name	Type	Definition / Purpose / Comments	Default
imin_integ	INT	Starting i-index of the cuboid.	1 + nboundlines
l_integrals	LOG	To switch on/off the computation of integrals.	.FALSE.
imax_integ	INT	Ending i-index of the cuboid.	ie_tot - nboundlines
jmin_integ	INT	Starting j-index of the cuboid.	1 + nboundlines
jmax_integ	INT	Ending j-index of the cuboid.	je_tot - nboundlines
kmin_integ	INT	Starting k-index of the cuboid	1
kmax_integ	INT	Ending k-index of the cuboid	ke_tot

6.7 SATCTL — Controlling the Synthetic Satellite Images

In August 2003 a project was started between DWD and the DLR Institute for Atmospheric Physics which aimed at the generation of synthetic satellite images within the COSMO-Model. The RTTOV (Radiative Transfer model for TIROS Operational Vertical sounder) model, Version 7, is used to compute radiances for satellite infrared or microwave nadir scanning radiometers from an atmospheric profile of temperature, variable gas concentrations, and cloud and surface properties.

From Version 4.18 up to Version 5.04e, also the use of RTTOV, Version 9, was possible, but has been removed again.

From Version 4.26 on, RTTOV 10 can be used. And in Version 5.05 also RTTOV 12 has been implemented.

The different RTTOV-libraries are included in the COSMO-Model by conditional compilation. Depending on which implementation is used, the corresponding macro has to be set when compiling the COSMO-Model (for example with -DRTTOV7):

`'RTTOV7'` to use the RTTOV7 library.
Note that the COSMO-Model does not use the official RTTOV7 library, but a modified one, which takes care of using this library within parallel programs. It also contains some optimizations for vector processors.

`'RTTOV9'` to use the RTTOV9 library. Eliminated in Version 5.05.

`'RTTOV10'` to use the RTTOV10 library.

`'RTTOV12'` to use the RTTOV12 library.

To use RTTOV10 or RTTOV12, another module, developed at DWD, is necessary as interface to this library: `mo_rttov_ifc.f90`. This module is available from DWD.

The RTTOV model can compute radiances and brightness temperatures of several instruments, which are located on diverse satellites. The implementation provided in the COSMO-Model only can compute the values for the following two instruments and channels specified in Table 6.43.

For every channel the following 4 products can be computed:

- cloudy brightness temperature
- clear-sky brightness temperature
- cloudy radiance
- clear-sky radiance

Note that the values for METEOSAT satellites can be computed, but be aware of the location of these satellites. Only MSG 2 is located such that it is useful for Europe.

Table 6.43: Instruments and channels for use in the COSMO-Model

Sensor	Satellite	Channel	Central Wavelength
MVIRI	METEOSAT [7]	1	WV6.4
MVIRI	METEOSAT [7]	2	IR11.5
SEVIRI	MSG [1-2]	4	IR3.9
SEVIRI	MSG [1-2]	5	WV6.2
SEVIRI	MSG [1-2]	6	WV7.3
SEVIRI	MSG [1-2]	7	IR8.7
SEVIRI	MSG [1-2]	8	IR9.7
SEVIRI	MSG [1-2]	9	IR10.8
SEVIRI	MSG [1-2]	10	IR12.1
SEVIRI	MSG [1-2]	11	IR13.4

Two fields have been implemented into the COSMO-Model to take care of the output of the synthetic satellite images. These fields can be specified by the shortnames

'SYNME7' output of the products for MVIRI, METEOSAT7,

'SYNMSG' output of the products for SEVIRI, MSG1 or MSG2.

Although special channel and products can be chosen via Namelist, the implementation is such that all channels for a product of a special instrument are computed.

Computation of the SynSat products can be controlled by the following namelist parameters:

Name	Type	Definition / Purpose / Comments	Default
itype_rttov	INT	To specify, which RTTOV Version shall be used. Possible values: 7,10,12. RTTOV9 has been removed in Version 5.05	7
num_sensors	INT	Number of sensors used during the calculation.	0
lcon_clw	LOG	To specify whether convective liquid water shall be used in the computations	.FALSE.
lsynsat	LOG	To activate computation of synthetic satellite images (default behaviour from former versions before 4.26).	.FALSE.
lobsrad	LOG	To activate satellite observation processing. This can only be done, if the model is compiled with -DRTTOV10!	.FALSE.
lread_ct	LOG	To read NWCSAF SEVIRI cloud type	.FALSE.
yclouddir	CHAR	Directory of cloud data.	"
linterp	LOG	Do an interpolation of p, t and q to half levels.	.TRUE.

Name	Type	Definition / Purpose / Comments	Default
sat_input_01	TYPE	Structure to specify characteristics for first instrument and to specify the products that shall be generated. Components are:	
	CHAR	Name of the satellite	all 'y'
	INT	Satellite identification	0
	CHAR	Name of the sensor	all 'y'
	INT	Number of channels used	0
	LOG	To generate clear sky radiance	.FALSE.
	LOG	To generate cloudy sky radiance	.FALSE.
	LOG	To generate clear sky brightness temperature	.FALSE.
	LOG	To generate cloudy sky brightness temperature	.FALSE.
sat_input_02	TYPE	Structure to specify characteristics for second instrument and to specify the products that shall be generated (same as above).	
nchan_input_01	INT	input channel list for first sensor	all 0
nchan_input_02	INT	input channel list for second sensor	all 0
emiss_input_01	REAL	To read emissivities for all channels of first instrument	all 0.0
emiss_input_02	REAL	To read emissivities for all channels of second instrument	all 0.0
		The following parameters apply only for the RTTOV10/12	
sat_long_01	REAL	Position of first satellite (longitude)	-999.0
sat_long_02	REAL	Position of second satellite (longitude)	-999.0
extrp_type	INT	Type of extrapolation above highest model level 0: constant 1: linear 2: extrapolate towards a climatological value	0
iceshape	INT	To specify, whether ice particles are 1: hexagonal 2: or ice aggregates	1

Name	Type	Definition / Purpose / Comments	Default
`iwc2effdiam`	INT	Type of conversion of ice water content to effective diameter of ice particles 1: Ou and Liou, 1995, Atmos. Res., 35, 127-138. 2: Wyser et al. (see McFarquhar et al. (2003)) 3: Boudala et al., 2002, Int. J. Climatol., 22, 1267-1284. 4: McFarquhar et al. (2003) NOTE: Only scheme 4 has been tested extensively!	4

6.8 INICTL — Parameters for the Model Initialization

The namelist group INICTL contains control parameters for the digital filter initialization (DFI). These parameters are only in effect if the main switch ldfi in /RUNCTL/ is set to .TRUE. The specifications for the parameters in /INICTL/ are included in the file INPUT_DYN. The DFI is relevant for NWP applications.

When the COSMO-Model starts with interpolated data from a coarse grid driving model – GME, IFS or the COSMO-Model itself – the initial data may contain unbalanced information for the mass and wind field. This may give rise to spurious high-frequency oscillations of high amplitude during the first hours of integration (dynamical adaptation). In this case the initial data can be modified by an initialization procedure in order to filter the unbalanced gravity and sound wave components to a realistic level. For this purpose, the DFI scheme has been implemented. By default, the initialization consists of a 1-hour adiabatic backward integration followed by a 1-hour diabatic forward integration of the model. Other settings can be chosen by the below namelist parameters.

Name	Type	Definition / Purpose / Comments	Default
ndfi	INT	Indicator for kind of filtering. 0: No filtering is done. 1: Foreward-stage filtering (launching) using a diabatic foreward integration. 2: Two-stage filtering using an adiabatic backward integration followed by a diabatic foreward integration.	0
nfilt	INT	Indicator for method of filtering. 1: Dolph-Chebyshev filter is used. *(no other filter is implemented)*	1
tspan	REAL	Time-span (in seconds) for the adiabatic and diabatic stages of the initialization. *Caution: The time-span has to be less or equal to the time, the first boundary data set is provided!*	3600.0
dtbak	REAL	Time-step (s) for the backcast filtering stage. *It is recommended to set* dtbak = dt.	90.0
dtfwd	REAL	Time-step (s) for the forecast filtering stage. *It is recommended to set* dtfwd = dt.	90.0
taus	REAL	Cuttoff time period (in seconds) of the filter. High-frequency components with periods less than taus are filtered. *It is recommended to set* taus *to a value smaller or equal to* tspan.	3600.0

Name	Type	Definition / Purpose / Comments	Default
itype_dfi_qx	INT	Treatment of moisture variables during the digital filtering initialization (active only for ndfi=1): 0: All moisture variables are filtered in the forward stage. 1: QV is filtered, other variables are initialized with instantaneous values valid at nhalf step of forward stage 2: QV is filtered but corrected to maintain saturated points at nhalf step of forward stage, other variables are initialized with instantaneous values valid at nhalf step of forward stage.	0
itype_dfi_soil	INT	Treatment of soil variables during the digital filtering initialization (active only for ndfi=1): 0: All soil variables are derived from the average of values at the initial and final step of forward stage. 1: All soil variables are initialized with instantaneous values valid at nhalf step of forward stage.	0

The recommended settings that were tested at the Italian Met Service are:

- itype_dfi_qx = 2
- itype_dfi_soil = 1

6.9 EPSCTL — Controlling the Ensemble Prediction Mode

The namelist group /EPSCTL/ contains control parameters for the ensemble prediction mode (EPS), e.g. to define the number of members, variations in input- and boundary data and variations in other namelist parameters to generate ensemble perturbations. The specifications for the parameters in /EPSCTL/ are included in the file INPUT_EPS.

Since Version 5.1 there are two more sets of variables in this namelist group, related to the stochastic perturbation of physics tendencies (SPPT) and the generation of random fields for SPPT.

The namelist parameters of this group are described in the subsections

- basic control parameters and
- parameters for changing input fields
- stochastic perturbation of physics tendencies (SPPT)
- generation of random fields for SPPT

Basic control parameters:

Name	Type	Definition / Purpose / Comments	Default
iepsmem	INT	ID of the member in the ensemble (ID ≥ 0) As a local solution for Grib 1, iepsmem is coded in product definition section 52 (ipds(52)).	-1
iepstot	INT	Total number of ensemble members (≥ 0) As a local solution for Grib 1, iepstot is coded in product definition section 51 (ipds(51)).	-1
iepstyp	INT	ID of the ensemble generation type (ID ≥ 0) As a local solution for Grib 1, iepstyp is coded in product definition section 50 (ipds(50)).	-1

Parameters for changing input fields:

Name	Type	Definition / Purpose / Comments	Default
fac_plcov	REAL	Modification factor for PLCOV	1.0
rmin_plcov	REAL	Lower limit for values of PLCOV	0.0
rmax_plcov	REAL	Upper limit for values of PLCOV	1.0
fac_lai	REAL	Modification factor for LAI	1.0
rmin_lai	REAL	Lower limit for values of LAI	0.0
rmax_lai	REAL	Upper limit for values of LAI	8.0
fac_rootdp	REAL	Modification factor for ROOTDP	1.0
rmin_rootdp	REAL	Lower limit for values of ROOTDP	0.0
rmax_rootdp	REAL	Upper limit for values of ROOTDP	2.0

Parameters for the stochastic perturbation of physics tendencies

The following variables are only in effect, if the namelist variable lsppt in group /RUNCTL/ is set to .TRUE..

Name	Type	Definition / Purpose / Comments	Default
itype_qxpert_rn	INT	Defines, which hydrometeor tendencies are perturbed: 0: qv only. 1: qv, qc, qi. 2: qv, qc, qi, qr, qs, qg.	0
itype_qxlim_rn	INT	Type of reduction/removal of the perturbation in case of negative (for qv, qx) or supersaturated (qv) values 0: No limitation of perturbed tendencies. 1: If new qv values are negative or super-saturated then tt T- and tt qv-tendencies are not perturbed; if new qc, qi, qr, qs, qg-values are negative then the corresponding tendencies are not perturbed.	0
ltargetdiff_mask	LOG	To switch off SPPT locally if targeted diffusion is active. (introduced in Version 5.04g)	.FALSE.
hinc_reset_pm	REAL	Increment for resetting pertstoph_mask (in hours) Introduced in Version 5.08.	0.0

Generation of random fields for SPPT

The following variables are only in effect, if the namelist variable lsppt in group /RUNCTL/ is set to .TRUE..

Name	Type	Definition / Purpose / Comments	Default
lhorint_rn	LOG	Random numbers defined on a random number horizontal coarse grid: .TRUE. The random numbers (as defined on a random number horizontal coarse grid) are horizontally bi-linearly interpolated to the COSMO-Model grid. .FALSE. The model grid points contained in the same random number coarse grid box have the same random number value.	.TRUE.
ltimeint_rn	LOG	Treatment of 3-D fields of random numbers: .TRUE. The 3-D fields of random numbers (available at *random number time steps*) are interpolated linearly in time. .FALSE. The random numbers remain constant in time until the next random number time step.	.TRUE.
lgauss_rn	LOG	To determine the distribution of random numbers: .TRUE. Use a gaussian distribution of random numbers. .FALSE. Use a uniform distribution of random numbers.	.TRUE.
itype_vtaper_rn	INT	Type of tapering near surface and in stratosphere: 0: No vertical tapering. 1: Prescribed tapering near surface and in stratosphere. 2: Prescribed tapering only in stratosphere. 3: Tapering according to namelist variable vtaper_rn.	1

Name	Type	Definition / Purpose / Comments	Default
imode_rn	INT	Stream of random numbers: 0: Use only one stream of random numbers (for each pattern) for all random number time steps; the same offset of the coarse grid relative to the lower left corner of the COSMO grid is then used for all random number time steps (for a given pattern). 1: Use a new stream of random numbers for every random number time step; this renders the 3-D random number fields valid for a given time reproducible even if produced by successive model runs, and enables temporal correlations in DA cycles across analysis steps; the offset of the coarse grid relative to the lower left corner of the COSMO grid is then different for each random number time step.	1
npattern_rn	INT	Number of random number patterns with different (4-D) correlation scales used for SPPT (must be ≤ 5).	1

For the following namelist parameters, npattern_rn values have to be specified:

Name	Type	Definition / Purpose / Comments	Default
hinc_rn	REAL	Random number time step (in [hrs]), i.e. hour increment for drawing a new 3-D field of random numbers; if imode_rn > 0, 24 hours must be divisible by hinc_rn; and if both hinc_rn and ninc_rn are given valid values (> 0) then the value of hinc_rn is used.	6.0
ninc_rn	INT	Random number time step (in [model timestep] units), i.e. timestep increment for drawing a new 3-D field of random numbers.	-1
nseed_rn	INT	External part of seed for random number generation.	0
nseed_rn2	INT	Only for imode_rn=1: External part of seed to generate 3-D random number fields valid for times later than the initial time of the model run.	0
dlat_rn	REAL	Random number coarse grid point distance in meridional direction (in [degrees]).	2.5
dlon_rn	REAL	Random number coarse grid point distance in zonal direction (in [degrees]).	2.5
stdv_rn	REAL	Standard deviation of the Gaussian distribution of random numbers.	0.5
range_rn	REAL	Upper limit imposed to absolute value of random numbers (the condition is: SUM(range_rn2) $<$ 1 (sum over npattern_rn), otherwise the perturbations may change the sign of physics tendencies, which may easily lead to instabilities).	stdv_rn_d $\cdot$ 2

For the following namelist parameter, ke values have to be specified (number of vertical model layers):

Name	Type	Definition / Purpose / Comments	Default
`vtaper_rn`	`REAL`	Only for `itype_vaper_rn` = 3: Externally specified function (by values ≤ 1 for each model layer from top down) for vertical tapering of the random number.	-

6.10 IOCTL — Controlling the Grib I/O

The namelist group /IOCTL/ contains some general parameters to control the Grib or NetCDF IO of a model run. It is the *top level* configuration namelist group for I/O. The specifications for the parameters in /IOCTL/ are included in the file INPUT_IO.

The namelist parameters of this group are described in the subsections

- basic control parameters,
- writing (and reading) Restart-Files,
- additional specifications for NetCDF-IO,
- reading and writing ”Ready-Files”,
- controlling the Soil Moisture Analysis,

Basic control parameters:

There are two options for prescribing initial and boundary conditions. If lartif_data (former name was lgen; the new switch now is in namelist group /RUNCTL/) is set to .TRUE., a run with user-defined artifical initial and boundary data is performed. Another namelist group /ARTIFCTL/ has to be specified in that case. But the default is a real-case model run with initial and boundary data coming from a coarse-grid driving model (initial data can also be generated by a continuous data assimilation using the nudging technique).

Name	Type	Definition / Purpose / Comments	Default
lgen	LOG	Eliminated in 4.8. Has been renamed to lartif_data and moved to group /RUNCTL/.	
lasync_io	LOG	If .TRUE., the model runs with extra processors for asynchronous IO.	.FALSE.
lprefetch_io	LOG	Enables reading of boundary files ahead of time, i.e. when the forthcoming I/O operation will be reading a GRIB file, then this can be done simultaneously with the preceding compute steps. Prefetching can only be enabled with true asynchronous I/O (lasync_io=.TRUE.).	.FALSE.
itype_gather	INT	To choose the type of gathering output fields: 1: (default) gather the fields by an extra communication per level 2: gather fields by one communication for nproc levels (with MPI_ALLTOALLV)	1
ngribout	INT	To specify how many NAMELIST-groups /GRIBOUT/ are contained for the model run. At least one block is necessary!	1

Name	Type	Definition / Purpose / Comments	Default
nvers	INT	The version number of a model run for documenting purposes. nvers is coded in the PDS of GRIB output files and is the only parameter to distinguish GRIB output files for the same case but coming from different model versions.	1
ncenter	INT	To specify the WMO-identification of generating center.	78
nsubcenter	INT	To specify the WMO-identification of the originating subcenter (to set GRIB2 metadata).	255
nlocaldefnr	INT	To specify the local definition number for GRIB local sections. The default value of -1 means, that no local section is present.	-1
yform_read	CHAR	Specifies the format of input files. The following formats are implemented: 'grb1' read GRIB1 data with DWD's GRIB library. 'apix' read GRIB 1/2 data with ECMWF's grib_api. 'ncdf' read NetCDF data.	'grb1'
yform_write	CHAR	In Version 4.18 this parameter has been moved to the group /GRIBOUT/. This gives the possibility, that it can be specified for every output group. (Specifies the format of output files (can be 'grb1' or 'ncdf')	'grb1'
yform_restart	CHAR	Introduced in Version 5.06b. Specifies the format for restart files. Up to Version 5.06a only binary restart files could be written. Now the option to write restart files in NetCDF has been introduced: 'bina' restart files will be written in binary format. 'ncdf' restart files will be written in NetCDF format.	'bina'
ymode_read	CHAR	Specifies the mode how files are opened for reading.	'r '
ymode_write	CHAR	Specifies the mode how files are opened for writing.	'w '
num_gribtabs	INT	Specifies the number of different GRIB tables used in COSMO-Model variable table.	6
lst_gribtabs(:)	LOG	Identifications of the different GRIB tables. Current table used are: 2, 201, 202, 203, 204, 205	see left
lbdclim	LOG	If .TRUE., use additional boundary fields that are needed for long term simulations.	.FALSE.
lbdsst	LOG	If .TRUE., use an additional boundary field to update the Sea Surface Temperature during NWP simulations	.FALSE.
ldwd_grib_use	LOG	If .TRUE., special DWD Grib settings are used	.TRUE.
l_ke_in_gds	LOG	If .TRUE., write number of vertical levels as a vertical coordinate parameter to the GDS grib section	.TRUE.

Name	Type	Definition / Purpose / Comments	Default
dbzthresh_ echotop	INT	(vector of at most 10 integers) Threshold(s) for which to compute ECHOTOPinM and ECHOTOP if one or both are specified in the list of model level output variables (yvarml in GRIBOUT).	(18, -1000, ..., -1000)

Writing (and reading) Restart-Files:

Name	Type	Definition / Purpose / Comments	Default
nhour_restart	INT	Triplet to specify start, stop and increment for writing restart files. Specifications are in full forecast hours.	12, 0, 12
ydir_restart	CHAR	Eliminated in 5.1; replaced by ydir_restart_in and ydir_restart_out	"
ydir_restart_in	CHAR	Directory for reading restart files.	"
ydir_restart_out	CHAR	Directory for writing restart files.	"
ytunit_restart	CHAR	Time unit to specify the file name for restart files.	'f'

Additional specifications for NetCDF-IO:

The Network Common Data Format (NetCDF) is designed to identify the data contained in a file in a unique way. Hereto the necessary informations need to be written into the file.

yncglob_institution	CHAR		'-'
yncglob_title	CHAR		'-'
yncglob_source	CHAR		'-'
yncglob_contact	CHAR		'-'
yncglob_project_id	CHAR		'-'
yncglob_experiment_id	CHAR		'-'
yncglob_references	CHAR		'-'
ncglob_realization	INT		1

Additional variables when using the Radar Forward Operator:

If the COSMO-Model is compiled to use the Radar Forward Operator EMVORADO (compile with -DRADARFWO), two additional namelist variables can be set to specify I/O directories for Mie lookup tables:

`ydir_mielookup_read`	CHAR	Reading Mie lookup tables	’-’
`ydir_mielookup_write`	CHAR	Writing Mie lookup tables	’-’

Reading and writing “Ready-Files” or ”Lockfiles”:

In an operational environment you sometimes have to run processes in parallel, although there are dependencies between these processes. For example, INT2LM usually runs in parallel with the COSMO-Model, but the COSMO-Model can only read a lateral boundary input file, if it has been written by INT2LM. Or a postprocessing job can only work with COSMO output fields, if the corresponding output file has been written completely.

For that purpose, the so called *ready files* have been implemented in INT2LM and the COSMO-Model, which work in the following way:
Process `A`, which provides output to other programs, writes a *ready file* after the output has been written completely. During execution, process `B`, which wants to read the output from process `A`, can check the existence of this *ready file* and can wait for a certain time interval, until it is available. Then the output of process `A` can be used.

In COSMO-Model Version 5.06 and INT2LM Version 2.06, an alternative approach has been implemented by the colleagues from MeteoSwiss, which works with *lockfiles*. In this approach, process `A` does not write its output directly to a file `outputfile`, but to a file with name `outputfile.lck`. Only if the file is fully written and is closed, the name is changed to `outputfile`. For this the (non-standard) Fortran intrinsic `RENAME` is used (which today is offered by nearly all Fortran compilers).

Namelist variables for Ready Files

`ytrans_in`	CHAR	Directory path for reading ready-files.	’ ’
`ytrans_out`	CHAR	Directory path for writing ready-files.	’ ’
`nincwait`	INT	Seconds to wait until the next read attempt for a lateral boundary file if the corresponding ready file is not available.	0
`nmaxwait`	INT	Maximum seconds to wait until abort if the ready file for the next lateral boundary file is not available.	0

Namelist variables for Lockfiles

`llockfiles`	LOG	to indicate, whether to use lockfiles (`.TRUE.`) or not (`.FALSE.`)	`.FALSE.`

Controlling the Soil Moisture Analysis:

`nsma_stat`	INT	Status for soil moisture analysis. `nsma_stat` is coded in the PDS of the grib-ed fields for soil moisture content (in the “local use area”). **0:** this is a “normal” soil moisture field. **20:** this is a field derived by soil moisture analysis.	0

6.11 GRIBIN — Controlling the Grib Input

The namelist group /GRIBIN/ contains parameters related to the input of initial and boundary files, which are an analysis file for the initialization of the model fields (containing also the *constant* parameter fields such as orography, soil type etc.), and boundary data files for certain equidistant boundary update time steps.

Note that the namelist group /GRIBIN/ applies to all possible input data formats (GRIB, NetCDF), not only to GRIB data. The specifications for the parameters in /GRIBIN/ are included in the file INPUT_IO.

Each namelist parameters of the list is described in one of the sections:

- control parameters for initial data,
- control parameters for boundary data and
- additional parameters for continuous data assimilation

Control parameters for initial data

Name	Type	Definition / Purpose / Comments	Default
ydirini	CHAR	Directory path of the initial data file – interpolated data from a driving model or analysis from a continuous COSMO-Model data assimilation.	' '
lchkini	LOG	Checking the initial data for min/max values.	.FALSE.
yvarini(:)	CHAR	has been eliminated in Version 4.12.	
lana_qi	LOG	If .TRUE., use the cloud-ice field contained in the initial data file as initial condition for cloud-ice; otherwise, initial conditions for cloud-ice are set in the model.	.FALSE.
lana_qr_qs	LOG	If .TRUE., use the values for rain and snow contained in the initial data file. Otherwise, values are set in the model.	.FALSE.
lana_qg	LOG	If .TRUE., use the values for the graupel scheme contained in the initial data file. Otherwise, values are set in the model.	.FALSE.
lana_tke	LOG	If .TRUE. use the values for tke contained in the initial data file. Otherwise, values are set in the model.	.FALSE.
lana_rho_snow	LOG	If .TRUE., values for the density of snow are provided in the initial data;	.FALSE.
nlgw_ini	INT	Eliminated in Version 5.05 Number of prognostic soil water levels in initial data. This parameter is only used for the "old" soil model.	2
nsec_fg	INT	Lead time (in [s], as integer) of first guess file for incremental analysis update. (Introduced in Version 5.04d).	3600

Control parameters for boundary data

Name	Type	Definition / Purpose / Comments	Default
hincbound	REAL	Interval in hours between two consecutive sets of boundary data (boundary update frequency). Within this time interval, boundary values are interpolated linearly in time. Only multiples of 0.25 are possible.	1.0
ydirbd	CHAR	Directory path of the boundary data files.	' '
lchkbd	LOG	Checking the boundary data for min/max values.	.FALSE.
yvarbd	CHAR	has been eliminated in Version 4.12.	
llb_qi	LOG	If .TRUE., take cloud-ice values contained in the lateral boundary data as boundary condition for cloud ice; otherwise, cloud ice boundary conditions are set in the model.	.FALSE.
llb_qr_qs	LOG	If .TRUE., take rain and snow values contained in the lateral boundary data as boundary condition.	.FALSE.
llb_qg	LOG	If .TRUE., take graupel values contained in the lateral boundary data as boundary condition.	.FALSE.
lbdana	LOG	If .TRUE., use analysis data as lateral boundary conditions.	.FALSE.
lic2bc	LOG	If .TRUE., use initial conditions as boundary data at timestep 0. (Introduced in Version 5.04d).	.FALSE.
sic2bc	REAL	Period (in [s]) in which the initial conditions are additionally used for the lateral boundary conditions (with weight 1 at time 0 decreasing linearly to zero at time sic2bc) (Introduced in Version 5.04d).	3600.0
hlastbc	REAL	Boundary fields valid later than this lead time [hour] are not read (boundary conditions beyond that time limit are obtained by linear extrapolation of boundary fields). (Introduced in Version 5.04d).	hstop+100
ytunitbd	CHAR	Time unit indicator in the boundary data file name (see Section 6.2). 't' forecast range given in timesteps 'f' the forecast range is given in the form ddhhmmss (where dd: day, hh: hour, mm: minute, ss: second) 'c' the forecast range is given in the form yyydddhh (yyy: year, ddd: day of the year, hh: hour) 'd' the forecast range is given in the form yyyymmddhh or (optionally) yyyymmddhhmise (can only be used, if hincbound is given in full hours) (Use 'd' for COSMO-CLM)	'f'
nlgw_bd	INT	Eliminated in Version 5.05 Number of prognostic soil water levels in boundary data. This parameter is only used for the "old" soil model.	2

Name	Type	Definition / Purpose / Comments	Default
lbd_frame	LOG	Normally, the boundary data are defined on the full 3-D model domain. IF lbd_frame = .TRUE., it is assumed that the boundary data are only defined on a lateral frame (e.g. when using boundary data obtained from ECMWF/IFS).	.FALSE.
npstrframe	INT	Number of grid-point rows and colums where lateral boundary data are defined when using the lbd_frame option.	8
ilevbotnoframe	INT	Number of vertical layer which separates frame and no-frame boundary data. For layers extending from the top of the model domain to ilevbotnoframe, the boundary data are defined for the full horizontal domain (this enables the application of a Rayleigh damping layer near the model top), and for layers from ilevbotnoframe to the lowest layer, the boundary data are defined on a frame.	0

Options for treatment of lateral boundary data

In Version 5.04d several options for the lateral boundary data have been implemented:

- Option to use the initial conditions for the lateral boundary data (namelist variable lic2bc) at the initial time with linearly decreasing weight during a pre-specified period (given by the namelist variable sic2bc.
- Option to discard boundary fields valid later than a certain lead time (given by namelist variable hlastbc). Boundary conditions after the last processed boundary field are obtained by linear extrapolation of the two last boundary fields.
- A non-negative check is introduced for tracers to allow for the linear extrapolation of boundary fields at the end of the model integration. This allows for a proper treatment of the lateral boundary condition at the last timestep also in the standard configuration of the model and avoids the jump in the surface pressure tendency at the last timestep seen e.g. in the operational runs.
- Note: This changes the model results very slightly. Only the last output step is affected.
- Bug fixes for the determination of nlastbound (and hence for the use of the lateral boundary conditions), if the interval of boundary data is not hourly or if the the first lateral boundary data are older than the model initial time.

Additional parameters for continuous data assimilation

Name	Type	Definition / Purpose / Comments	Default
`newbc`	INT	Number of times that boundary update is analysis after 1 hour.	360
`newbcdt`	INT	Time step increment of boundary update from analysis.	’ ’
`hnewbcdt`	REAL	Hour increment of boundary update from analysis.	0.0
`nincboufac`	INT	Factor to `nincbound` when new boundary update is analysis.	2
`lan_t_s`	LOG	Selection of analysed (tri=0) sea surface temperature.	`.FALSE.`
`lan_t_so0`	LOG	Selection of analysed (tri=0) sea surface temperature (for the multi-layer soil model.	`.FALSE.`
`lan_t_snow`	LOG	Selection of analysed snow temperature.	`.FALSE.`
`lan_t_cl`	LOG	Selection of analysed climatological soil temperature.	`.FALSE.`
`lan_w_snow`	LOG	Selection of analysed water content of snow.	`.FALSE.`
`lan_w_i`	LOG	Selection of analysed interception water.	`.FALSE.`
`lan_w_so`	LOG	Selection of analysed or external soil moisture.	`.FALSE.`
`lan_w_cl`	LOG	Selection of analysed climatological soil water content.	`.FALSE.`
`lan_vio3`	LOG	Selection of analysed vertical integrated ozone.	`.FALSE.`
`lan_hmo3`	LOG	Selection of analysed ozone maximum.	`.FALSE.`
`lan_plcov`	LOG	Selection of analysed plant cover.	`.FALSE.`
`lan_lai`	LOG	Selection of analysed leaf area index.	`.FALSE.`
`lan_rootdp`	LOG	Selection of analysed root depth.	`.FALSE.`
`lan_rho_snow`	LOG	Selection of analysed density of snow.	`.FALSE.`

The control of the parameters `lan_xxx` is as follows:
`.TRUE.`: Fields are used from external analyses or from an interpolated coarse grid field.
`.FALSE.`: Fields are used from the continous data assimilation of the COSMO-Model.

6.12 GRIBOUT — Controlling the Grib Output

How to use the NAMELIST group GRIBOUT is explained in more detail in Part VI of the COSMO-Model Documentation: *Model Output and Data Formats for I/O*.

Basic parameters for controlling the output:

Name	Type	Definition / Purpose / Comments	Default
ydir	CHAR	Directory of the output file.	' '
yvarml(:)	CHAR	List of model variables for output	
yvarpl(:)	CHAR	List of variables for pressure levels	
plev(:)	REAL	Pressure levels	
yvarzl(:)	CHAR	List of variables for z-levels	
zlev(:)	REAL	Z-levels	
yvarsl(:)	CHAR	List of satellite channels for output	
ngrib(:)	INT	List of output steps given in time steps	99999
ncomb(:)	INT	Triplets for building ngrib	99999
hgrib(:)	REAL	List of output steps given in hours	0.0
hcomb(:)	REAL	Triplets for building ngrib	0.0

Controlling the output domain and time unit indicator:

ytunit	CHAR	Time unit indicator in the output data file name (see Section 6.2).	'f'
ydomain	CHAR	Domain type (full or sub) indicator in the output data file name (see Section 6.2).	'f'
slon	REAL	Left longitude for a subdomain	startlon_tot
slat	REAL	Bottom latitude for a subdomain	startlat_tot
elon	REAL	Right longitude for a subdomain	endlon_tot
elat	REAL	Upper latitude for a subdomain	endlat_tot

Additional parameters for controlling the output:

Name	Type	Definition / Purpose / Comments	Default
ysystem	CHAR	This switch has been eliminated in Version 4.28 File, database or ecfs.	
ydbtype	CHAR	This switch has been eliminated in Version 4.28 Type of the database.	
yform_write	CHAR	Specifies the format of output files 'grb1' write GRIB1 data with DWD's GRIB library. 'api1' write GRIB 1 data with grib_api. 'api2' write GRIB 2 data with grib_api. 'ncdf' write NetCDF data. 'nc-4' Introduced in Version 5.09. write NetCDF data in NetCDF4 format	'grb1'
ypackingType	CHAR	Introduced in Version 5.06b. To choose GRIB 2 packing type. Up to Version 5.06a this was hard-coded as 'grid_simple'. For GRIB 2, additional packing types have been introduced, which offer an additional packing and therefore result in smaller Grib files (see comments below). The following packing types can now be used by INT2LM (2.06a) and COSMO (depends on configuration settings of eccodes!) 'grid_simple' default; as is up to now. 'grid_ccsds' CCSDS packing. 'grid_jpeg' JPEG 2000 Code Stream Format. 'grid_png' Using the png compression engine.	'grid_simple'
lcompress_netcdf	LOG	Introduced in Version 5.10. Determine whether compression is applied or not for a specific netcf file.	.FALSE.
nprocess_ini	INT	Generating process identification for initial data.	-999999
nprocess_bd	INT	Generating process identification for boundary data.	-999999
n_num	INT	This switch has been eliminated in Version 4.28 Counter for nests.	
nrbit	INT	Number of bits per value for grib packing. Usually, 16 bits give sufficient accuracy.	16
nunit_of_time	INT	Indicates the *Unit of Time* for Grib Code	1
lcheck	LOG	Checking the output data for min/max values.	.TRUE.
lanalysis	LOG	If .TRUE., analysis output files are written when the model runs in nudging mode.	.FALSE.
lwrite_tiles	LOG	If .TRUE., write all tiles of a tile variable to output.	.FALSE.

Name	Type	Definition / Purpose / Comments	Default
luvmasspoint	LOG	Enables interpolation of horizontal winds to mass grid points for output.	.FALSE.
loutput_q_densities	LOG	Determines the output units of the hydrometeor contents "QC", "QR", etc. If .FALSE., these are output as mass specific quantities (kg/kg), which is the model internal convention. If .TRUE., mass densities (kg/m^3) are output instead. In case of using the two-moment microphysical scheme, the units of the number contents "QNCLOUD", "QNRAIN", etc. (1/kg or 1/m^3) are chosen analogously.	.FALSE.
l_p_filter	LOG	Logical switch, if .TRUE., the fields on pressure levels are digitaly filtered to remove small-scale noise	.FALSE.
l_z_filter	LOG	Logical switch, if .TRUE., the fields on height levels are digitaly filtered to remove small-scale noise.	.FALSE.
lzint_above_ground	LOG	Introduced in Version 5.06b. To choose z-interpolation to levels above ground (default is to interpolate to levels above mean sea level).	.FALSE.
itype_vertint	INT	To specify the type of the vertical interpolation used to interpolate values to p- and / or z-level. 1: Cubic tension splines (previous method) 2: Linear interpolation (new in Version 4.17)	1
l_fi_ps_smooth	LOG	This switch has been renamed (see below), because what really is smoothed here is the mean sea level pressure!	--
l_fi_pmsl_smooth	LOG	Logical switch, if .TRUE., additional smoothing is done for mean sea-level pressure and geopotential in mountainous areas.	.FALSE.
l_fi_filter	LOG	Logical switch to filter the geopotential FI independent of the settings of l_p_filter.	.FALSE.
l_pmsl_filter	LOG	Logical switch to filter the mean sea level pressure PMSL independent of the settings of l_p_filter and / or l_z_filter	.TRUE.
ysuffix	CHAR	To add an optional suffix to the file names.	"

Notes on the GRIB 2 packing types:

The following packing types can now be used by INT2LM (2.06a) and COSMO (depends on configuration settings of eccodes!)

- 'grid_simple': default; as is up to now.
- 'grid_ccsds': The Consultative Committee for Space Data Systems uses a Golomb-Rice packing scheme, which is rather efficient in terms of size and encoding size for transmitting gridded data. This algorithm can also be applied to meteorological data and is available in GRIB 2 now. Note that this is a lossless packing, but acts upon the grid_simple packing, which looses accuracy.

- 'grid_jpeg': JPEG 2000 Code Stream Format (Joint Photographic Experts Group) is also a lossless compression method, which acts upon the grid_simple packing.

- 'grid_png': The png compression engine is based on the deflate method, which is a widely used, patent-free algorithm for universal, lossless data compression.

To use these additional packing types, they also have to be activated for eccodes (with -DENABLE_JPG=ON, -DENABLE_PNG=ON, -DENABLE_AEC=ON). See also the installation instructions for eccodes. The additional packing also takes additional computational time.

Additional variables when using the Radar Forward Operator:

If the COSMO-Model is compiled to use the Radar Forward Operator EMVORADO (compile with -DRADARFWO), an additional namelist variable dbz can be set to specify a namelist configuration for EMVORADO. The variable dbz is of TYPE dbzcalc_params, which is explained in the EMVORADO documentation.

dbz	TYPE	Namelist configuration for EMVORADO	

Handling of statistically processed fields:

The new diagnostic output variables for tracking of convective cells need a modified treatment of statistically processed fields, because they should be reset after every output step.

Up to now only the following methods are possible in the COSMO-Model: Statistical processing by

- summation or averaging:
  Fields are summed up from beginning of the simulation and never reset. Only in climate mode (lbdclim=.TRUE.) the fields are reset to 0.0 after an output step.

- Min / Max values:

  - Temperatures (TMIN_2M, TMAX_2M) are reset according to the namelist variable hincmxt in /RUNCTL/.
  - Wind speeds and gusts (e.g. VMAX_10M, VABS_10M, VGUST_DYN_10M, VGUST_CON_10M) are reset according to namelist variable hincmxu in /RUNCTL/.
  - the new MAX-variables for tracking of convective cells should be reset after every output. The difficulty here is, that these variables could occur in different GRIBOUT namelist groups.

A more generalized way has been implemented in the COSMO-Model, Version 5.05, how the resetting of statistically processed fields is organized.

All statistically processed fields are grouped into four categories.

Temperatures	TMIN_2M,	TMAX_2M,		
Winds	VMAX_10M,	VABSMX_10M,	VGUST_DYN_10M,	VGUST_CON_10M,
Summations /	T_2M_AV,	TD_2M_AV,	U_10M_AV,	V_10M_AV,
Averages	ASOB_S,	ATHB_S,	ASOB_T,	ATHB_T,
	ALHFL_S,	ASHFL_S,	AUMFL_S,	AVMFL_S,
	APAB_S,	ALHFL_BS,	ALHFL_PL,	ASWDIR_S,
	ASWDIFD_S,	ASWDIFU_S,	ATHD_S,	ATHU_S,
	ASOD_T,	AUSTR_SSO,	AVSTR_SSO,	AVDIS_SSO,
	AEVAP_S,	TOT_PREC,	SNOW_CON,	SNOW_GSP,
	RUNOFF_S,	RUNOFF_G,	SNOW_MELT,	DURSUN,
	TDIV_HUM,	RAIN_GSP,	RAIN_CON,	GRAU_GSP,
	HAIL_GSP,	DURSUN_M,	DURSUN_R,	
Cells	TCOND_MAX,	TCOND10_MX,	LPI_MAX,	DBZ_CTMAX,
	UH_MAX,	VORW_CTMAX,	W_CTMAX,	

For every category the type of resetting can be chosen. For that there are four new Namelist variables in /GRIBOUT/ to control the type of resetting. Allowable values for these variables are in principal:

0: no resetting at all,

1: reset after given interval,

2: reset after every output.

Which values are really applicable for which category is given in the table below.

Name	Type	Definition / Purpose / Comments	Default
ireset_temps	INT	Type of resetting for temperatures. 0: no resetting at all, 1: reset after given interval, 2: reset after every output.	1
ireset_winds	INT	Type of resetting for winds. 0: no resetting at all, 1: reset after given interval, 2: reset after every output.	1

Name	Type	Definition / Purpose / Comments	Default
ireset_sums	INT	Type of resetting for summations and averages 0: no resetting at all, 2: reset after every output.	0
ireset_cells	INT	Type of resetting for cell tracking variables 0: no resetting at all, 2: reset after every output.	2

The default values of the four new namelist variables in are in a way that the current behaviour of the COSMO-Model regarding statistically processed fields is retained.

References

Adams-Selin, R. D. and C. L. Ziegler (2016). Forecasting Hail Using a One-Dimensional Hail Growth Model within WRF. *Mon. Wea. Rev. 144*(12), 4919–4939.

Baldauf, M. (2013). A new fast-waves solver for the Runge-Kutta dynamical core. COSMO Technical Report No. 21, Deutscher Wetterdienst, http://www.cosmo-model.org/content/model/documentation/techReports/default.htm.

Baldauf, M. (2014). The COSMO model: towards cloud-resolving NWP. In *Recent Developments in Numerical Methods for Atmosphere and Ocean Modelling - Proceedings of a Seminar*, pp. 107–122. ECMWF, Reading, UK.

Bechtold, P., E. Bazile, F. Guichard, P. Mascart, and E. Richard (2001). A mass-flux convection scheme for regional and global models. *Quart. J. Roy. Meteor. Soc. 127*, 869–886.

Bechtold, P., M. Köhler, T. Jung, F. Doblas-Reyes, M. Leutbecher, M. Rodwell, F. Vitart, and G. Balsa (2008). Advances in simulating atmospheric variability with the ECMWF model: From synoptic to decadal time-scales. *Quart. J. Roy. Meteor. Soc. 134*, 1337–1351.

Bechtold, P., N. Semane, P. Lopez, J.-P. Chaboureau, A. Beljaars, and N. Bormann (2014). Representing equilibrium and non-equilibrium convection in large-scale models. *J. Atmos. Sci. 71*, 734–753.

Davies, H. and R. Turner (1977). Updating prediction models by dynamical relaxation: An examination of the technique. *Quart. J. R. Met. Soc. 103*, 225–245.

Dudhia, J. (1993). A nonhydrostatic version of the Penn State / NCAR mesoscale model: Validation tests and simulation of an Atlantic cyclone and cold front. *Mon. Wea. Rev. 121*, 1493–1513.

Hess, R. (2001). Assimilation of screen-level observations by variational soil moisture analysis. *Meteorol. Atmos. Phys. 77*, 155–166.

Jacobsen, I. and E. Heise (1982). A new economic method for the computation of the surface temperature in numerical models. *Contr. Atmos. Phys. 55*, 128–141.

Kessler, E. (1969). *On the Distribution and Continuity of water substance in the atmospheric circulations*, Volume 32 of *Meteorological Monographs Vol. 10*. American Meteorological Society, Boston.

Klemp, J. and R. Wilhelmson (1978). The Simulation of Three-dimensional Convective Storm Dynamics. *J. Atmos. Sci. 35*, 1070–1096.

Liu, X.-D., S. Osher, and T. Chan (1994). Weighted essentially non-oscillatory schemes. *J. Comput. Phys. 115*, 200–212.

Lott, F. and M. Miller (1997). A new subgrid-scale orographic drag parametrization: Its formulation and testing. *Q. J. R. Meteor. Soc. 123*, 101–127.

Louis, J.-F. (1979). A parametric model of vertical eddy fluxes in the atmosphere. *Bound. Layer Meteor. 17*, 187–202.

Lynch, P., D. Girard, and V. Ivanovici (1997). Improving the efficiency of a digital filtering scheme. *Mon. Wea. Rev. 125*, 1976–1982.

Mellor, G. and T. Yamada (1974). A hierarchy of turbulence closure models for planetary boundary layers. *J. Atm. Sc. 31*, 1791–1806.

Mironov, D. (2008). Parameterization of lakes in numerical weather prediction. Description of a lake model. COSMO Technical Report 11, Consortium for Small-Scale Modelling, 41 pp. doi:10.5676/DWD_pub/nwv/cosmo-tr_11 (available from http://www.cosmo-model.org).

Mironov, D. and B. Ritter (2004). Testing the new ice model for the global NWP system GME of the German Weather Service. Technical documentation, 1220, WMO, WMO, Geneve, Switzerland.

Müller, E. (1981). Turbulent flux parameterization in a regional-scale model. In *ECMWF Workshop on planetary boundary layer parameterization*, pp. 193–220. 25-27 November, 1981.

Ritter, B. and J.-F. Geleyn (1992). A comprehensive radiation scheme for numerical weather prediction models with potential applications in climate simulations. *Mon. Wea. Rev. 120*, 303–325.

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

Schlemmer, L., C. Schär, D. Lüthi, and L. Strebel (2018). A groundwater and runoff formulation for weather and climate models. *J. of Advances in Modeling Earth Systems 10*, 1809–1832.

Schraff, C. (1996). Data assimilation and mesoscale weather prediction: A study with a forecast model for the Alpine region. Publication 56, Swiss Meteorological Institute, Zürich.

Schraff, C. (1997). Mesoscale data assimilation and prediction of low stratus in the Alpine region. *Meteorol. Atmos. Phys. 64*, 21–50.

Schraff, C., H. Reich, A. Schomburg, K. Stephan, A. Periáñez, and R. Potthast (2016). Kilometre-scale ensemble data assimilation for the COSMO model (KENDA). *Q. J. R. Meteorol. Soc. 142*, 1453–1472.

Schulz, J.-P. (2008). Introducing sub-grid scale orographic effects in the COSMO model. *COSMO Newsletter 9*, 29–36. (Available at: http://www.cosmo-model.org).

Skamarock, W. and J. Klemp (1992). The stability of time-split numerical methods for the hydrostatic and nonhydrostatic elasic equations. *Mon. Wea. Rev. 120*, 2109–2127.

Skamarock, W., P. Smolarkiewicz, and J. Klemp (1997). Preconditioned Conjugate Gradient Solvers for Helmholtz Equations in Nonhydrostatic Models. *Mon. Wea. Rev. 125*, 587–599.

Sommeria, G. and J. W. Deardorff (1977). Subgrid-scale condensation in models of non-precipitating clouds. *J. Atmos. Sci. 34*, 344–355.

Stauffer, D. and N. Seaman (1990). Use of four-dimensional data assimilation in a limited-area mesoscale model. Part I: Experiments with synoptic-scale data. *Mon. Wea. Rev. 118*, 1250–1277.

Stauffer, D. and N. Seaman (1994). Multiscale four-dimensional data assimilation. *J. Appl. Meteor. 33*, 416–434.

Stephan, K., S. Klink, and C. Schraff (2008). Assimilation of radar-derived rain rates into the convective-scale model COSMO-DE at DWD. *Quart. J. R. Meteor. Soc. 134*, 1315–1326.

Thomas, S., C. Girard, G. Doms, and U. Schättler (2000). Semi-Implicit Scheme for the DWD Lokal-Modell. *Meteorol. Atmos. Phys. 73*, 105–125.

Tiedtke, M. (1989). A comprehensive mass flux scheme for cumulus parameterization in large-scale models. *Mon. Wea. Rev. 117*, 1779–1799.

Viterbo, P. and A. Beljaars (1995). An improved land surface parametrization scheme in the ECMWF model and its validation. *J. Clim. 8*, 2716–2748.

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

Yurova, A., M. Tolstykh, M. Nilsson, and A. Sirin (2014). Parameterization of mires in a numerical weather prediction model. *Water Resour. Res. 50*, 8982–8996.