Consortium for Small-Scale Modelling

A Description of the

Nonhydrostatic Regional COSMO-Model

Part V:

Preprocessing:
Initial and Boundary Data for the COSMO-Model

U. Schättler

INT2LM 1.23 **October 2013**

www.cosmo-model.org

Printed at Deutscher Wetterdienst, P.O. Box 100465, 63004 Offenbach, Germany

Contents

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 GME form – together with the corresponding data assimilation schemes – the NWP-system at DWD, which is run operationally since end of 1999. The subsequent developments related to the model have been organized within COSMO, the *Consortium for Small-Scale Modelling*. COSMO aims at the improvement, maintenance and operational application of a non-hydrostatic limited-area modelling 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 *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
USAM	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
AGeoBW	Amt für Geoinformationswesen der Bundeswehr, Euskirchen, Germany
NMA	National Meteorological Administration, Bukarest, Romania
RosHydroMet	Hydrometeorological Centre of Russia, Moscow, Russia

will apply.

The further development of the modelling system within COSMO is organized in Working Groups which cover the main research and development activities: data assimilation, numerical aspects, physical aspects, interpretation and applications, verification and case studies, reference version and implementation. 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 Work Package Coordinators or members of the Working Groups or Priority Projects, please refer to the COSMO web-site.

At present, the COSMO meteorological services are not equipped to provide extensive support to external users of the model. If problems occur in certain aspects, we would kindly ask you to contact the corresponding Work Package Coordinators or the current Scientific Project Manager. 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. Many NWP-models operate on hydrostatic scales of motion with grid spacings down to about 10 km and thus lack 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.

The present operational application of the model within COSMO is mainly on the meso-β scale using a grid spacing of 7 km. The key issue is 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. 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.

Besides the operational application, the COSMO-Model provides a nonhydrostatic modelling 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 modelling, 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 initial data.

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.

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** – Second-order leapfrog HE-VI (horizontally explicit, vertically implicit) time-split integration scheme by default, including extensions proposed by Skamarock and Klemp (1992). Option for a three time-level 3-d semi-implicit scheme (Thomas et al., 2000). Several Options for 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.
- **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 (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 (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. Preliminary option for calculation of horizontal turbulent diffusion in terrain following coordinates (3D Turbulence).
- **Surface Layer Parameterization** – A Surface layer scheme (based on turbulent kinetic energy) including a laminar-turbulent roughness layer. Option for a stability-dependent drag-law formulation of momentum, heat and moisture fluxes according to similarity theory (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.
- **Moist Convection** – Tiedtke (1989) mass-flux convection scheme with equilibrium closure based on moisture convergence. Option for the Kain-Fritsch (1992) convection scheme with non-equilibrium CAPE-type closure.
- **Shallow Convection** – Reduced Tiedtke scheme for shallow convection only.
- **Radiation** – δ two-stream radiation scheme after Ritter and Geleyn (1992) for 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. Option for the (old) two-layer soil model employing the extended force-restore method still included.
- **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

- **Basic Method** – Continuous four-dimensional data assimilation based on observation nudging, 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.

- **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), 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. 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** – The compilation of all programs is performed by a Unix shell script invoking the Unix *make* command. All dependencies of the routines are automatically taken into account by the script.
- **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-XT3, IBM-SP series, SGI ALTIX series).
- **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 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.

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

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:	Implementation Documentation
Part V:	Preprocessing: Initial and Boundary Data for the COSMO-Model
Part VI:	Postprocessing
Part VII:	User's Guide

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. Available postprocessing utilities will be described (in the future) in Part VI. 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 (*www.cosmo-model.org*).

Section 2

Introduction

This part of the documentation for the COSMO-Model is the description of the interpolation program INT2LM, which performs the interpolation from coarse grid model data to initial and/or boundary data for the COSMO-Model. The following coarse grid models are possible (at the moment):

- GME: the global DWD grid point model on a icosahedral grid.
- IFS: the global ECMWF spectral model.
- COSMO-Model: the COSMO-Model can be nested into itself.

It is also possible to process the data from other climate models (like ECHAM), but another *pre-pre-processor* is needed then. These *pre-pre-processor* are available from the CLM-Community, which operates the "CLimate Mode of the COSMO-Model".

Originally, INT2LM has been a joint development within COSMO and originates from the former GME2LM. The climate mode has been added by members of the CLM-Community. The development tasks were distributed as follows:

- DWD: parallel framework of the program; GME2LM
- ARPA-SIM: IFS2LM
- MeteoSwiss, DWD: LM2LM
- CLM-Community: climate mode and processing data from other climate models

This documentation is not yet complete. Missing are the description of the initial and boundary data that are necessary to run the COSMO-Model and the scientific documentation of the interpolation algorithms used. Nevertheless, it provides a User Guide of how to install the program and how to run it. Therefore it serves as a complete reference for all the NAMELIST groups and variables.

Section 3

The Interpolation Procedures

to be completed

Section 4

Installation of the INT2LM

This chapter explains the steps necessary to compile and run the interpolation program. Section 4.1 lists the external libraries that are necessary to run the program and what can be done, if these libraries are not available. Section 4.2 describes how to use the VCS (Version Control System: a programming environment tool developed at DWD) for working with the model. If the VCS is not available, the source code together with a Makefile for compiling and linking and scripts for running the model are provided. The next sections give detailed informations on how to prepare, compile, link and run the INT2LM.

4.1 External Libraries for the INT2LM

INT2LM uses external libraries for data I/O. 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`. INT2LM 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.

At DWD, a data base system can be used for this, which needs special routines. If these are not available, `dummy` routines are provided to satisfy the external references.

4.1.1 `libgrib1.a`:

As standard data format, the GRIB (Grided Binary), Version 1, is used. With `libgrib1.a`, data can be packed in / unpacked from grib code. 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.

DWD still uses a Grib file format, where all records are starting and ending with additional bytes, the so-called *controlwords*. An implementation of the Grib library is prepared that also deals with pure Grib files, that do not have these controlwords. But still we guarantee correct execution only, if the controlwords are used. To ensure this you have to set the environment variable

```
export LIBDWD_FORCE_CONTROLWORDS=1
```

4.1.2 `libcsobank.a`, `libsupplement.a`:

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.

4.1.3 `libgrib_api.a`:

Since Version 1.14, 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). The source code for grib_api is available from the web pages of ECMWF `http://www.ecmwf.int`. For INT2LM, grib_api Version 1.9.0 or higher is needed.

Usage of the grib_api 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 grib_api calls are not compiled and the library will not be linked to the binary.

4.1.4 `libnetcdf.a`:

Since Version 1.7, input and output of data can 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.

4.2 Working with the VCS

The Version Control System is a programming environment tool based on the Concurrent Version System (CVS). The programming environment consists of several shell scripts (or command procedures) that are accessible from an administrator directory (on DWD systems this directory is `/e/rhome/for0adm/vcscmd`; you can refer to this directory with the shell variable `$ADM`, if it is set properly). These command procedures serve to simplify tasks and contain safety features which may otherwise be easily forgotten.

External users having a collaboration with DWD can access the code of the COSMO-Model (and also of other models), the necessary scripts for installing the programming environment tool, and a description of that tool via ftp. A list of all command procedures together with a short explanation can be obtained with `$ADM/help`.

4.3 Preparing the Code

Source Code Administrator (for VCS)
As a source code administrator you have to provide the external code and libraries. They have to be created on your system and put to a special directory. They also have to be specified as `EXTOBJ` in `LinkLibs`, in order to link them to the object files of the COSMO-Model.

User (with VCS)
If working with the VCS you have to create your own workbench within a special directory (e.g. `$HOME/model`) with the command

`$ADM/workbench int2lm`.

The following files and subdirectories are created:

`./CompilerFlags`	To specify, which module is compiled with which set of compiler options.
`./FileNames`	To define the names of binaries and/or libraries.
`./LinkLibs`	To define the libraries for the link step.
`./Makefile`	Link to a makefile for compiling and linking.
`./Options`	To set the compiler and linker options.
`./Parallel`	To set the number of parallel tasks for compiling.
`./edid`	Script to edit the SCCS-decks.
`./mk_batch`	Script to submit a batch job (optional).
`./obj`	Directory containing object files of the files in `src`.
`./src`	Directory containing modified source files.
`./work`	Directory containing files you are working on.

Normally, correct defaults are set by your administrator. You can change `Options`, `Parallel` and `LinkLibs` according to your needs (see also the part for the *Source Code Administrator*).

User (without VCS)
If the VCS is not available, you have got a tar-file `int2lm_yymmdd_x.y`, where `yymmdd` describes the date in the form "Year-Month-Day" and `x.y` gives the version number as in the DWD Version Control System (VCS). 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.
`./runxx2yy`	Scripts to define the `NAMELIST` input and run the model for special coarse grid models *xx* and applications *yy*.
`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.

Here, also the source code for `mpe_io.f90` and the `dummies` for the data base system are included in `src`: `dummy_db.f90`. 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.

Before compiling and linking the program you should check and, if necessary, adapt the `KIND`-type parameters, which are used for selecting the precision of `REAL`-variables in the program and the precision of `INTEGER`-variables of the grib-library.

4.4 Compiling and Linking

Before compiling, check and adapt the necessary parameters (see above). All other input variables for the program can be determined before running the program with the `NAMELIST`-input (see Chapter 7). You have to choose the options for compiling the code in the file `Options` (if working within the VCS) or in `Fopts` (otherwise). See the User Guide of your computer system for necessary and/or desired options. Before linking check that the Grib library, necessary for the I/O, the external object files `mpe_io.o` (and `dummy_db.o`) and the necessary external libraries (see 4.1) are available.

The INT2LM 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` is provided, which has to be linked with the model then.

sequential	On single processor systems you can create a binary for sequential execution without using MPI. To avoid warning messages by the linker, a file `dummy_mpi.f90` is provided to satisfy the MPI external references.
parallel	On parallel computers with distributed memory you can create a binary for parallel execution, if MPI is available. You can also create a sequential binary, which can only run on one processor.

In the VCS environment the creation of one or more certain binaries is fixed. Ask your administrator, if you want to change the default. Outside the VCS you can choose the binary by modifying `Makefile`.

You can invoke a `make`-run by typing `make` *entry*. On batch-machines you can start a batch job for a `make`-run with `mk_batch` *entry*. Within VCS type `make help` for a list of available entries.

4.5 Running the Code

To run the code, an ASCII-file `INPUT` has to be provided that contains values for the `NAMELIST` variables. The form of this `INPUT` file is described in Chapters 6 and 7. This file is created by the provided run-scripts. See the manual for your system on how to invoke the binary created in the last step.

Section 5

Necessary Initial and Boundary Data for the COSMO-Model

This chapter lists all initial and boundary data which are necessary to run the COSMO-Model. Some of the data depend on special namelist settings in the COSMO-Model and the INT2LM, resp. This will be explained in detail.

5.1 Initial Data for the COSMO-Model

The data necessary to start the COSMO-Model can be divided into three groups:

5.1.1 External parameters:

The COSMO-Model needs information about the lower boundary of the domain and also of some background fields, like ozone content. The necessary information is either provided by an *external parameter* file (see also Section 6.2), which is produced for a certain region and resolution or computed directly in the INT2LM.

Constant external parameters for the surface

The following constant external parameters are needed in any case to start a simulation with the COSMO-Model. They are provided in an external parameter file.

`HSURF`	Height of surface topography
`FIS`	(alternatively) Geopotential of surface
`FR_LAND`	Fraction of land in the grid cell
`SOILTYP`	Soil type of the land (keys 0-9)
`Z0`	Roughness length

Additional external parameters are provided, that can be used in selected components of the COSMO-Model. Older external parameter files might not carry these informations, therefore usage of the corresponding fields can be controlled by namelist switches. The names of these switches are the same in INT2LM and in the COSMO-Model.

Subgrid scale orography scheme

The sub-grid scale orography (SSO) scheme by Lott and Miller (1997) has been implemented in the COSMO-Model (from Version 4.5 on). It is also included in the DWD global model. The scheme deals explicitly with a low-level flow which is blocked when the sub-grid scale orography is sufficiently high. For this blocked flow separation occurs at the mountain flanks, resulting in a form drag. The upper part of the low-level flow is lead over the orography, while generating gravity waves. The following external parameters are necessary to run the subgrid scale orography scheme:

`SSO_STDH`	standard deviation of subgrid scale orography [m]
`SSO_GAMMA`	anisotropy of the orography [-]
`SSO_THETA`	angle between the principal axis of orography and east [rad]
`SSO_SIGMA`	mean slope of subgrid scale orography [-]

The usage of the subgrid scale orography scheme is controlled by the Namelist switch `lsso`.

Topographical corrections in the radiation scheme

Topographical corrections for radiation calculations have been introduced. The following external parameters are necessary to compute these corrections.

`SKYVIEW`	sky view [1]
`SLO_ASP`	slope aspect [rad]
`SLO_ANG`	slope angle [rad]
`HORIZON`	horizont array: The horizont is splitted in `nhori` sectors

The usage of the topographical corrections is controlled by the Namelist switch `lradtopo`. The field `HORIZON` is treated as threedimensional array with `nhori` levels. `nhori` is also read as a namelist parameter.

NOTE: The topographical corrections are not yet available in the official code of the COSMO-Model.

External parameters for lakes

The usage of lake fraction and lake depth can be controlled by the Namelist switch `llake`. Up to now, the COSMO software cannot provide external parameters for lakes. Only some preliminary test files are available for Germany. Their usage can be switched off by setting `llake=.FALSE.` in INT2LM and the COSMO-Model. This is still the default.

`FR_LAKE`	lake fraction in a grid element [0,1]
`DEPTH_LK`	lake depth

Mimimum stomata resistance of plants

Up to Version 4.10 the multi-layer soil model of the COSMO-Model used a constant minimum value of stomatal resistance for plants. Now an external map can be read, that provides values for every grid point. Its usage is controlled by the Namelist switch `lstomata`.

`PRS_MIN`	mimimum stomata resistance of plants

Thermal radiative surface emissivity

Up to Version 4.10 a constant value was used for the thermal radiative surface emissivity. Now an external map can be read, that provides values for every grid point. Its usage is controlled by the Namelist switch `lemiss`.

`EMIS_RAD`	thermal radiative surface emissivity

Ground fraction covered by forests

The ground fraction covered by evergreen and deciduous forest, resp., can be used in the radiation scheme to determine the effect of snow covered forests on solar snow albedo. Their usage is controlled by the Namelist switch `lforest`.

`FOR_E`	ground fraction covered by evergreen forest
`FOR_D`	ground fraction covered by deciduous forest

Plant characteristics, ozone contents and aerosol types

These external fields are usually held constant for the duration of a simulation for numerical weather prediction. In climate simulations they are updated together with the boundaries.

Plant characteristics

The following fields have to be provided by the external parameter data set. Depending on the chosen options they are read by the INT2LM, processed if necessary, and passed on to the COSMO-Model.

PLCOV_MX	plant cover data set for vegetation time
PLCOV_MN	plant cover data set for time of rest
PLCOV12	12 monthly climatological mean values for plant cover
LAI_MX	leaf area index data set for vegetation time
LAI_MN	leaf area index data set for time of rest
LAI12	12 monthly climatological mean values for leaf area index
ROOTDP	root depth
NDVI_MRAT	ratio of monthly mean normalized differential vegetation index to annual maximum for 12 months

There are several options to compute these fields. They are controlled by the namelist parameter `itype_ndvi` and `itype_rootdp`, resp. Possible values for `itype_ndvi` are:

0: Data sets for vegetation and for rest are read from the external parameter file for plant cover and the leaf area index The actual values for a special day are computed by producing a sinus-type 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 normalized differential vegetation index (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 (provided by DWD in Grib1).

2: plant cover, leaf area index and roughness length for the COSMO-Model and for a special day are produced by using 12 monthly climatological mean values for plant cover, leaf area index and roughness length. These values are read from the external parameter data set for the COSMO-Model (provided by CLM in NetCDF).

For the root depth, just one dataset is given. The options for computing actual values are given by `itype_rootdp`:

0: input from external parameter for the COSMO-Model is taken and modified with an annual cycle.

1: input from external parameter for the COSMO-Model is taken as is but with a minimal value of 0.12.

2: input from external parameter for the COSMO-Model is taken and modified with an annual cycle. In addition, the values are adapted to ECOCLIMAP niveau.

3: input from external parameter for the COSMO-Model is taken and modified with an annual cycle but without maximum cut off.

4: input from external parameter for the COSMO-Model is taken without any modifications.

Ozone contents

VIO3	Vertical integrated ozone content
HMO3	Ozone maximum

Aerosol characteristics

The default treatment of aerosols in the COSMO-Model is by assuming constant values for aerosols in rural, urban or desert areas and over sea. Now, 12 monthly mean values of the following aerosol types can be read from the external parameters:

AER_SO4	Tegen (1997) aerosol type sulfate drops
AER_DUST	Tegen (1997) aerosol type mineral dust
AER_ORG	Tegen (1997) aerosol type organic
AER_BC	Tegen (1997) aerosol type black carbon
AER_SS	Tegen (1997) aerosol type sea salt

What type of aerosols should be used for the COSMO-Model is controlled by the namelist parameter `itype_aerosol`:

0: Default: constant values are assumed in the COSMO-Model. No external parameters are read in INT2LM.

1: The 12 monthly mean values are read by INT2LM and actual values for a special day are computed by a linear interpolation between the corresponding months.

5.1.2 Soil and surface variables:

Necessary surface variables

T_SNOW	Temperature of snow surface
W_SNOW	Water content of snow
W_I	Water content of interception water
QV_S	Specific water vapor content at the surface
T_S	Temperature of surface

In an assimilation cycle, these variables are governed by the COSMO-Model, with regular updates by external analyses for `T_SNOW, W_SNOW and W_I` (at 00, 06, 12 and 18 UTC) and for `T_S` (at 00 UTC). The external analysis for `T_S` updates the values only over sea.

If no assimilation cycle is used, the fields are interpolated from the coarse grid model.

Necessary soil variables

Which soil variables are necessary, depends on the usage of the soil model. For the multi-layer soil model, the following fields are necessary.

`T_SO`	Temperature of (multi-layer) soil levels
`W_SO`	Water content of (multi-layer) soil levels
`FRESHSNW`	Indicator for freshness of snow
`RHO_SNOW`	Prognostic snow density

In the COSMO-Model, the usage of the soil model is controlled by the namelist variable **`lmulti_layer`**`=.TRUE.`.

In INT2LM, the corresponding variable is **`lmulti_layer_lm`**`=.TRUE.`, which indicates, that the output of INT2LM has to be for the multi-layer soil model. There is also the namelist variable **`lmulti_layer_in`**`=.TRUE.`, which indicates, that the input (coarse grid) model also used a multi-layer soil model. This can only be true for the DWD models GME and COSMO-Model. No other model uses a comparable multi-layer soil model.

For the (old) two- or three-layer soil model, the following fields are necessary.

`T_M`	Temperature between upper and medium soil layer
`T_CL`	Temperature between medium and lower soil layer
`WG_1`	Water content of the upper soil layer
`WG_2`	Water content of the medium soil layer
`WG_3` (*)	Water content of the lower soil layer
`W_CL`	Climatological water content of the lowest soil layer

These fields will be read by the COSMO-Model, if **`lmulti_layer`**`=.FALSE.` is set. In the INT2LM, **`lmulti_layer_lm`**`=.FALSE.` has to be used correspondingly. The field `WG_3` only is necessary, if **`nlgw_ini`**`=3` in the namelist input for the COSMO-Model.

If an assimilation cycle is used, the soil fields are governed by the COSMO-Model (no matter, which soil model is used). There is the possibility to run a *Soil Moisture Analysis*, which is not really an external analysis, but adapts the soil moisture in the upper level in a way, that the temperature forecast is adjusted to the observations.

If no assimilation cycle is used, the fields are interpolated from the coarse grid model.

5.1.3 Atmospheric variables:

The following atmospheric variables are necessary to initialize a COSMO-Model forecast.

U	Zonal wind speed
V	Meridional wind speed
W	Vertical wind speed (defined on half levels)
T	Temperature
PP	Pressure deviation from a reference pressure
QV	Specific water vapour content
QC	Specific cloud water content

Since the start of the development of the COSMO-Model, more humidity variables have been added to the set of equation. Since values for these fields are not available in older data sets or from certain coarse grid models, their usage can be controlled by specific namelist settings. In the following table, the 3rd column gives the namelist variable in the COSMO-Model, the last column the corresponding namelist variable in the INT2LM. `ana` indicates, that the corresponding variable refers to the *analysis* or initial file.

QI	Specific cloud ice content	lana_qi	lprog_qi
QR	Specific rain content	lana_qr_qs	lprog_qr_qs
QS	Specific snow content	lana_qr_qs	lprog_qr_qs
QG	Specific graupel content	lana_qg	lprog_qg

In an assimilation cycle all these fields are updated by the nudging analysis. If no assimilation cycle is used, they are interpolated from the coarse grid model.

5.2 Boundary Data for the COSMO-Model

The necessary boundary data for the COSMO-Model are the atmospheric variables and some surface variables. Boundary values have to be provided also for the old two-layer soil model. Some special considerations have to be done for the climate mode.

5.2.1 Soil and surface variables:

Necessary surface variables

These fields are needed in any case:

`T_SNOW`	Temperature of snow surface
`W_SNOW`	Water content of snow
`QV_S`	Specific water vapor content at the surface

If the old two-layer soil model is used, also the following variables have to be provided:

`T_S`	Temperature of the surface
`T_M`	Temperature between upper and medium soil layer
`WG_1`	Water content of the upper soil layer
`WG_2`	Water content of the medium soil layer
`WG_3` (*)	Water content of the lower soil layer

The field `WG_3` only is necessary, if `nlgw_bd=3` in the namelist input for the COSMO-Model.

5.2.2 Atmospheric variables:

The following atmospheric variables are necessary boundary fields for the COSMO-Model.

`U`	Zonal wind speed
`V`	Meridional wind speed
`W` (*)	Vertical wind speed (defined on half levels)
`T`	Temperature
`PP`	Pressure deviation from a reference pressure
`QV`	Specific water vapour content
`QC`	Specific cloud water content

Depending on the variable `lw_freeslip` in the namelist input for the COSMO-Model, also the vertical wind speed has to be provided. If `lw_freeslip=.TRUE.`, a free-slip condition is implemented, which does NOT need boundary values. If `lw_freeslip=.FALSE.`, boundary values have to be provided. In INT2LM the namelist variable `lvertwind_bd=.TRUE.` has to be set in this case.

Whether boundary values can be provided by the coarse grid model for the other humidity variables, can be controlled by the namelist variables given in the next table (3rd column: COSMO-Model; last column: INT2LM).

QI	Specific cloud ice content	llb_qi	lprog_qi
QR	Specific rain content	llb_qr_qs	lprog_qr_qs
QS	Specific snow content	llb_qr_qs	lprog_qr_qs
QG	Specific graupel content	llb_qg	lprog_qg

5.2.3 Special considerations for the climate mode

If the COSMO-Model is run in climate mode, additional fields are necessary for the boundary updates, depending on the options chosen:

PLCOV	Plant cover
LAI	Leaf area index
ROOTDP	Root depth
VIO3	Vertical integrated ozone content
HMO3	Ozone maximum
T_S	Temperature of surface
	and only if `itype_aerosol = 2`
AER_SO4	Tegen (1997) aerosol type sulfate drops
AER_DUST	Tegen (1997) aerosol type mineral dust
AER_ORG	Tegen (1997) aerosol type organic
AER_BC	Tegen (1997) aerosol type black carbon
AER_SS	Tegen (1997) aerosol type sea salt

Section 6

Input Files for the INT2LM

The INT2LM requires several input files.

- An ASCII-file, called `INPUT`, that contains the namelist variables. The form of this file is described in Section 6.1 The namelist groups, the variables, their meanings and possible values are described in Chapter 7.
- 2 files with external parameters for the COSMO-Model and for the coarse grid model. These files can be in Grib(1) or NetCDF format.
- Files from the coarse grid model to compute the initial and/or boundary values. The name of these files are described in Section 6.5. These files can be in Grib(1) or NetCDF format. If GME is used and the GME-files contain data that were selected by a bitmap, this bitmap must also be provided to INT2LM.

6.1 File for Namelist Input

The INT2LM uses `NAMELIST`-input to specify runtime parameters. The parameters are splitted into the groups

- `CONTRL` – parameters for the model run
- `GRID_IN` – specifying the domain and the size of the coarse grid
- `LMGRID` – specifying the domain and the size of the COSMO-Model grid
- `DATABASE` – specifying a database job
- `DATA` – controlling the grib input and output
- `PRICTR` – controlling grid point output

The program provides default values for all parameters. To change a default value, an appropriate `NAMELIST` statement has to appear in the 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):

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
  startlat_tot = -10.4, startlon_tot = -3.025,
  pollat=32.5,          pollon=-170.0,
  dlat=0.025,           dlon=0.025,
  ielm_tot=72,          jelm_tot=92,
/
```

For a complete reference of all NAMELIST parameters see Chapter 7. An example INPUT-file can be seen in Figure 7.1.

6.2 External Parameters

For both models, the coarse grid input model and the COSMO-Model, external parameters are required to perform the interpolations. The necessary and / or available external parameters for the COSMO-Model are listed in Section 5.1

If the file with the external parameters for the COSMO-Model does not exist, they could be interpolated from the coarse grid model. But this works only for the configuration GME → COSMO. Note, that this is not recommended for practical simulations!

Necessary parameters for the coarse grid model

HSURF	Height of surface topography
FIS	(alternatively) Geopotential of surface
FR_LAND	Fraction of land in the grid cell
SOILTYP (*)	Soil type of the land (keys 0-9)

The soil type of some coarse grid models is not compatible to the soil types used in the COSMO-Model. These soil types are not used.

If external parameters for the COSMO-Model have to be interpolated from the coarse grid model, the following parameters are also necessary:

Z0	Roughness length
PLCOV	(Actual) plant cover during vegetation period
ROOTDP	(Actual) root depth

6.3 External Parameter Files for the COSMO-Model

External parameter files for the COSMO-Model are provided for different rotated coordinates, resolutions and domains. The following table shows the different files available from DWD's ftp-server with a short characterization. The filename contains information about the domain (`d0, d1, d3, d5`), the resolution in meters (_07000_ for about 7000 meters) and the size of the fields in grid points (`961x769`).

These files already contain the external parameters for the Subgrid Scale Orography scheme. External parameters for minimum stomata resistance of plants, the thermal radiative surface emissivity, the normalized differential vegetation index and mean values for the aerosol types are not yet provided. because the corresponding code parts are still under testing and evaluation at DWD. These parameters will be provided to the public later on.

lm_d0_02800_1605x1605.g1	dlat = dlon = 0.025	(*)
lm_d1_07000_961x769.g1	dlat = dlon = 0.0625	
lm_d1_07000_961x769_new.g1	dlat = dlon = 0.0625	(*)
lm_d1_07000_961x769.g1_2009121700	dlat = dlon = 0.0625	(**)
lm_d1_14000_481x385.g1	dlat = dlon = 0.125	
lm_d1_14000_481x385.g1_2009121700	dlat = dlon = 0.125	(**)
lm_d1_21000_321x257.g1	dlat = dlon = 0.1875	
lm_d1_21000_321x257.g1_2009121700	dlat = dlon = 0.1875	(**)
lm_d1_28000_241x193.g1	dlat = dlon = 0.25	
lm_d1_28000_241x193.g1_2009121700	dlat = dlon = 0.25	(**)
lm_d1_56000_121x113.g1	dlat = dlon = 0.5	
lm_d1_56000_121x113.g1_2009121700	dlat = dlon = 0.5	(**)
lm_d5_07000_965x773.g1	dlat = dlon = 0.0625	(*)
lm_d5_07000_965x773.g1_2009121700	dlat = dlon = 0.0625	(**)
lm_d5_14000_483x387.g1	dlat = dlon = 0.125	(*)
lm_d5_14000_483x387.g1_2009121700	dlat = dlon = 0.125	(**)

The domains `d1` are for a rotated grid with `pollat=32.5` and `pollon=-170.0`. Domains `d0` and `d5` are for a rotated grid with `pollat=40.0` and `pollon=-170.0`. `pollat` and `pollon` give the coordinates of the rotated north pole in real geographical coordinates. Files indicated by (*) are produced using a newer raw data set. Files indicated by (**) also are produced with the newer raw data set and they contain the SSO-parameters.

The area covered by the different domains is given in the next table.

	Rotated coordinates				Geographical coordinates			
	startlat	endlat	startlon	endlon	startlat	endlat	startlon	endlon
d0	-20.05	20.05	-20.05	20.05	27.11 N	63.69 N	11.21 W	56.59 E
d1	-38.75	9.25	-26.75	33.25	14.54 N	51.49 N	11.26 W	70.36 E
d5	-24.125	24.125	-30.125	30.125	20.00 N	60.16 N	19.17 W	77.01 E

External parameter files can be produced by DWD on request, if the domain (in rotated coordinates), the rotation (`pollat`, `pollon`) and the size of the domain (in grid points) is specified. Depending on the available raw data set, the highest possible resolution is about 2 km (`dlat=dlon=0.02`).

In the CLM-Community there is work going on to produce a preprocessor for the external parameters called PEP (Preparation of External Parameters). This preprocessor will use the DWD software, but add some own functionality, like the use of the ECOCLIMAP data set.

6.4 Available External Parameter Files for the GME

For DWD's global model GME, the following external parameter files are available on the ftp-server:

`invar_i128a`	resolution about 60 km
`invar_i192a`	resolution about 40 km
`invar_i192a_.new`	resolution about 40 km; new raw data set

The file `invar_i192a_.new` must be used for GME data after October, 24^{th}, 2007, 12 UTC!

6.5 Conventions for File Names

The initial and boundary fields needed for the model are provided either in Grib or in NetCDF format. Also for the output files, one can choose between Grib or NetCDF. Restart files are written in binary format with full precision. There is one file for the initial fields and also for every set of boundary fields. The following conventions apply for the filenames.

A file name for the COSMO-Model or the INT2LM has the general form

`yheader // ydate // yextension` (for Grib files),

or

`yheader // ydate // yextension // '.nc'` (for NetCDF files),

where `yheader`, `ydate` and `yextension` have the following meaning:

`yheader`: File header (3 characters)

- first character: specifies the model

  g: GME (global model)

  l: COSMO-Model

  e: ECMWF model IFS (Integrated Forecast System)

  c: A general (global) climate model

- second character:

  a: analysis file (uninitialized)

  i: analysis file (initialized)

  b: boundary file

  f: forecast files

  r: restart files

- third character: specifies the region covered by the data

  f: full model domain

  s: subdomain

`ydate`: There are two forms of specifying the date, either with the full date or relative to the start date:

- In the name of analysis files (second character in the header `a` or `i`) the full date is specified: `ydate` = `'yyyymmddhh'` with `yyyy`: year; `mm`: month; `dd`: day; `hh`: hour.

  Example:

`laf1992072100`	COSMO-Model, uninitialized analysis for full model domain from July, 21st, 1992.

- In forecast, boundary or restart files, `ydate` consists of a single character (the time unit of forecast range, `ytunit`), followed by a string.

  `ydate` = `ytunit // '`*string*`'`

  Depending on `ytunit`, the string has the following meaning:

  t: timestep mode: forecast range given in timesteps

  f: forecast mode: the forecast range is given in the form `ddhhmmss`, where `dd`: day, `hh`: hour, `mm`: minute, `ss`: second

  c: climate mode: the forecast range is given in the form `yyydddhh`, where `yyy`: year, `ddd`: day of the year, `hh`: hour

  d: day mode: the full date is given in the form `'yyyymmddhh'`, where `yyyy`: year; `mm`: month; `dd`: day; `hh`: hour

`yextension` (1 character, optional): Extension, e.g. data interpolated from model to pressure levels.

Examples:

`lbff00050000`	COSMO-Model, file with boundary values for hour 5
`lfff01233000`	COSMO-Model, forecast at day 1, 23 hours and 30 minutes.
`lrff01000000`	COSMO-Model, restart file for day 1.

Section 7

Namelist Input for INT2LM

The execution of INT2LM can be controlled by 6 `NAMELIST`-groups:

- `CONTRL` – parameters for the model run
- `GRID_IN` – specifying the domain and the size of the coarse grid
- `LMGRID` – specifying the domain and the size of the LM grid
- `DATABASE` – specification of database job
- `DATA` – controlling the grib input and output
- `PRICTR` – controlling grid point output

All `NAMELIST`-groups have to appear in the input file `INPUT` in the order given above. Every group is read in a special subroutine called `input_`*groupname*. These subroutines set default values for all parameters and check most parameters that have been changed for correctness and consistency.

The `NAMELIST` variables can be specified by the user in the run-scripts for the INT2LM, which then create the `INPUT` file. An example of `INPUT` is shown in Figure 7.1.

```
&CONTRL
 ydate_ini='2012051400', ydate_bd='2012051312',
 hstart=0.0, hstop=6.0, hincbound=1.0,
 linitial=.TRUE.,  lboundaries =.TRUE.,
 nprocx=4, nprocy=8, nprocio=0, lreorder=.FALSE.,
 yinput_model='GME',
 lfilter_oro=.TRUE., eps_filter=0.1,
 ilow_pass_oro=1, ilow_pass_xso=0, rxso_mask=0.0,
 lfilter_pp=.FALSE., lbalance_pp=.FALSE., norder_filter=5,
 lmulti_layer_in=.TRUE., lmulti_layer_lm=.TRUE., lprog_rho_snow=.TRUE.,
 lprog_qi=.TRUE., lprog_qr\_qs=.TRUE., luvcor=.TRUE.,
 lsso=.TRUE., lforest=.TRUE., llake=.FALSE., lbdclim=.FALSE.,
 itype_ndvi=0, idbg_level=2,
/
&GRID_IN
 ni_gme = 384, i3e_gme = 60, kcontrol_fi =15, ke_soil_in=7,
/
&LMGRID
 startlat_tot  = -20.0, startlon_tot  = -18.0,
 pollat=40.0,        pollon=-170.0,
 dlon=0.0625,        dlat=0.0625,
 ielm_tot=665,         jelm_tot=657,        kelm_tot=40,
 ke_soil_lm=7, ivctype=2, irefatm=2, delta_t=75.0, h_scal=10000.0,
/
&DATABASE
/
&DATA
 ie_ext=965, je_ext=773,
 ylmext_lfn='lm_d5_07000_965x773.sso.mol.g1',
 ylmext_cat='/e/rhome/routfor/routfox/lm/const/',
 yinext_lfn='invar.i384a',
 yinext_form_read='grb1',
 yinext_cat='/e/rhome/routfor/routfox/gme/const/',
 yin_form_read='grb1',
 yin_cat='/e/uscratch/uschaett/GME/data/',
 ybitmap_cat='/e/uscratch/uschaett/GME/bitmaps/'
 ybitmap_lfn='bitmp888',
 nbitmap=48000,
 ylm_cat='/e/uscratch/uschaett/COSMO_EU_input/'
 nprocess_ini = 131, nprocess_bd = 132,
 nl_soil_in=2, nl_soil_lm=2,
 l_ke_in_gds=.TRUE.,
/
&PRICTR
 lchkin=.TRUE., lchkout=.TRUE.,
/
```

Figure 7.1: Example file INPUT

7.1 CONTRL — Parameters for the Model Run

Initial time and forecast range

Name	Type	Meaning	Default
ydate_ini	CHAR	start of the forecast	' '
ydate_bd	CHAR	start of the forecast of the boundary model (if older forecast data shall be used)	' '
itype_calendar	LOG	to specify a certain type of calender 0: gregorian calendar (at the moment we still have the Julian calendar) 1: every year has 360 days 2: every year has 365 days	0
hstart	REAL	start of the forecast in hours	0.0
hstop	REAL	end of the forecast in hours	0.0
hincbound	REAL	time increment (in hours)	0.0
nincwait	INT	seconds to wait until next attempt if a ready file is not available	0
nmaxwait	INT	maximum seconds to wait until abort if a ready file is not available	0
ytrans_in	CHAR	directory for reading ready-files	' '
ytrans_out	CHAR	directory for writing ready-files	' '

Domain decomposition and parallelization

Name	Type	Meaning	Default
nprocx	INT	number of PEs in x-direction of the LM-grid	1
nprocy	INT	number of PEs in y-direction of the LM-grid	1
nprocio	INT	number of extra PEs for asynchronous I/O	0
nboundlines	INT	number of boundary lines of a subdomain	1
lreorder	LOG	if .TRUE., the PEs can be reordered for the cartesian MPI-communicator	.TRUE.
lasync_io	LOG	to run the model with extra (asynchronous) processors for I/O	.FALSE.
ldatatypes	LOG	if .TRUE., MPI-Datatypes for some communications shall be used	.FALSE.
ncomm_type	INT	type of communication	1

Basic Control

Name	Type	Definition / Purpose / Comments	Default
lgme2lm	LOG	eliminated from Version 1.14 on, now yinput_model='GME'	.TRUE.
lec2lm	LOG	eliminated from Version 1.14 on, now yinput_model='IFS'	.FALSE.
llm2lm	LOG	eliminated from Version 1.14 on, now yinput_model='GME'	.FALSE.
lum2lm	LOG	eliminated from Version 1.14 on, now yinput_model='UMR' and yinput_model='UMG'	.FALSE.
lcm2lm	LOG	eliminated from Version 1.14 on, now yinput_model='CM'	.FALSE.
yinput_model	CHAR*5	string to identify the input model; valid options: 'COSMO' : limited area model COSMO 'GFS' : Global Gorecast System (GFS) of NCAR 'GME' : Global Model (GME) of the DWD 'GSM' : Global Spectral Model (GSM) of the JMA 'HIRLM' : limited area model HIRLAM (introduced in 1.20) 'IFS' : Integrated Forecast System (IFS) of ECMWF 'UMG' : global Unified Model of the UK Met. Office (introduced in 1.20) 'UMR' : regional Unified Model of the UK Met. Office (introduced in 1.20) 'CM' : climate model standard format	' '
lanafg	LOG	eliminated from Version 1.9 on. Use analyses as input data (first guess)	.FALSE.
linitial	LOG	compute initial data for the COSMO-Model	.FALSE.
lboundaries	LOG	compute lateral boundaries for the COSMO-Model	.TRUE.
lbdclim	LOG	produce additional boundary fields that are needed for long term simulations.	.FALSE.
leps_bc	LOG	produce boundary data for ensemble mode	.FALSE.
lseaice	LOG	interpolate sea ice variables from GME to COSMO-Model	.FALSE.
luvcor	LOG	correct winds for given surface pressure tendency	.TRUE.

Name	Type	Definition / Purpose / Comments	Default
l_chemistry	LOG	switch to process additional chemistry fields Eliminated in Version 1.22 - Replaced by:	
l_art	LOG	switch to process additional fields for COSMO-ART	.FALSE.
l_art_nested	LOG	switch to process additional fields for COSMO-ART in case of nesting runs	.FALSE.
l_smi	LOG	switch for using an optional interpolation for the soil humidity	.FALSE.
lmixcld	LOG	switch for using an optional treatment of the humidity	.FALSE.
l_cressman	LOG	switch for using a cressman scheme during 'M'atch interpolation	.FALSE.
l_bicub_spl	LOG	switch for using a bicubic spline interpolation	.FALSE.
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.	2
lprintdeb_all	LOG	In most cases, the debug output is only written from one processor (with ID=0). With lprintdeb_all=.TRUE., all processors will print the debug output.	.FALSE.
ltime_proc	LOG	detailed timings of the program (per PE)	.FALSE.
ltime_mean	LOG	detailed timings of the program (mean value)	.FALSE.
lroutine	LOG	run the program as routine-job	.FALSE.

Controlling use of additional external parameters

Name	Type	Meaning	Default
lforest	LOG	read external parameters for the ground fraction of evergreen and deciduous forest	.FALSE.
lemiss	LOG	read a map from the external parameters for the thermal radiative surface emissivity	.FALSE.
lsso	LOG	read external parameters for the subgrid scale orography scheme	.FALSE.
lradtopo	LOG	read external parameters for the topographic corrections of radiation	.FALSE.
nhori	INT	number of sectors for the horizon array used by the topographic correction of the radiation	24
lstomata	LOG	read a map from the external parameters for the minimum stomata resistance of plants	.FALSE.
llake	LOG	read and process external parameters for lakes	.FALSE.
llake_coldstart	LOG	initialize prognostic lake variables for cold start	.FALSE.
lurban	LOG	read and process external parameters for an urban module. NOTE: An urban module is NOT official COSMO code but has to be implemented on your own!	.FALSE.

Control over variables that have to be written for the COSMO-Model

Name	Type	Definition / Purpose / Comments	Default
lvertwind_ini	LOG	compute vertical wind for initial data	.TRUE.
lvertwind_bd	LOG	compute vertical wind for boundary data	.FALSE.
lprog_qi	LOG	compute initial and boundary values for the cloud ice scheme	.FALSE.
lprog_qrqs	LOG	Renamed to lprog_qr_qs in Version 1.22	
lprog_qr_qs	LOG	compute initial and boundary values for rain and snow	.FALSE.
lprog_qg	LOG	compute initial and boundary values for the graupel scheme	.FALSE.
qvmin	REAL	security minimum value for water vapor	1E-12
qcmin	REAL	security minimum value for cloud water	1E-12
qimin	REAL	security minimum value for cloud ice content	1E-12
lmulti_layer_lm	LOG	compute fields for multi-layer soil model	.FALSE.
lmulti_layer_in	LOG	use multi-layer soil model variables from input fields	.FALSE.
lprog_rho_snow	LOG	read and interpolate the snow density to the COSMO-Model grid	.FALSE.
itype_w_so_rel	INT	to select the type of relative soil moisture input 0: use an artificial profile relative to the pore volume 1: relative to pore volume (read from coarse grid data) 2: relative to field capacity (read from coarse grid data) 3: the soil moisture profile in relation to the pore volume is kept constant below the deepest layer of the input soil model 4: the soil moisture profile related to field capacity is kept constant below the deepest layer of the input soil model	0
itype_t_cl	INT	to select the source for the deep soil (climatological) temperature 0: take t_cl from coarse grid model. 1: take t_cl from the external parameters for the COSMO-Model	0

Name	Type	Definition / Purpose / Comments	Default
itype_rootdp	INT	to select the treatment of the external parameter for root depth 0: input from external parameter for the COSMO-Model is taken and modified with an annual cycle. 1: input from external parameter for the COSMO-Model is taken as is but with a maximal value of 0.12. 2: input from external parameter for the COSMO-Model is taken and modified with an annual cycle. In addition, the values are adapted to ECOCLIMAP niveau. 3: input from external parameter for the COSMO-Model is taken and modified with an annual cycle but without maximum cut off. 4: input from external parameter for the COSMO-Model is taken without modifications.	0
itype_aerosol	INT	to select the treatment of plant cover and leaf area index: 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. 2: Additional external parameters for the aerosol types of sulfat, 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.	1

Name	Type	Definition / Purpose / Comments	Default
itype_ndvi	INT	to select the treatment of plant cover and leaf area index: 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 (provided by DWD in Grib1). 2: plant cover, leaf area index and z0 for the COSMO-Model and for a special day are produced by using 12 monthly climatological mean values for plant cover, leaf area index and roughness length. These values are read from the external parameter data set for the COSMO-Model (provided by CLM in NetCDF).	0
itype_albedo	INT	Switch to choose the type of solar surface albedo. This parameter has been introduced in Version 1.19. 1: surface albedo is a function of soiltype (method up to now and still default) 2: surface albedo is determined by two external fields for dry and for saturated soil. 3: A background albedo is prescribed by external fields, which give average values for every month. 4: The vegetation albedo is modified by forest fraction.	1
lt_cl_corr	LOG	if .TRUE., perform an alternative height correction for `t_cl`	.FALSE.

Name	Type	Definition / Purpose / Comments	Default
luse_t_skin	LOG	if .TRUE., use ECMWF skin temperature for surface	.FALSE.
lante_0006	LOG	if .TRUE., force to use ECMWF dataset before 27 June 2000	.FALSE.
lpost_0006	LOG	if .TRUE., force to use ECMWF dataset after 27 June 2000	.FALSE.

Treatment of orography and filtering

Name	Type	Meaning	Default
lfilter_oro	LOG	if .TRUE., filter the orography	.FALSE.
lfilter_pp	LOG	if .TRUE., filter the pressure deviation after vertical interpolation	.FALSE.
lbalance_pp	LOG	if .TRUE., compute a hydrostatic balanced pp after vertical interpolation in LM2LM	.FALSE.
eps_filter	REAL	parameter for filtering the orography	10.0
norder_filter	INT	order of the orography filtering	1
ilow_pass_oro	INT	type of low-pass filter for orography	1
numfilt_oro	INT	number of sequential applications of filter	1
ilow_pass_xso	INT	type of low-pass filter for extra smoothing of steep orography	0
numfilt_xso	INT	number of sequential applications of xso filter	1
lxso_first	LOG	do an extra smoothing of orography first	.FALSE.
rxso_mask	REAL	mask for extra smoothing of steep orography	0.0
rfill_valley	REAL	mask for valley filling	0.0
ifill_valley	REAL	type of valley filling	1
l_topo_z	LOG	additional smoothing of the topography for LM_Z	.FALSE.
llbc_smooth	LOG	run with a smooth orography transition at the lateral boundaries	.FALSE.
nlbc_smooth	INT	number of grid points for a smooth orography transition at the lateral boundaries	.FALSE.

7.2 GRID_IN — Specifying the Domain and the Size of the coarse Grid

grid_in contains variables that specify the size and resolution of the coarse grid.

Name	Type	Definition / Purpose / Comments	Default
ni_gme	INT	resolution of GME	128
i3e_gme	INT	number of levels in the vertical	51
kcontrol_fi	INT	control level for geopotential	15
ie_in_tot	INT	ie for input grid (total domain)	141
je_in_tot	INT	je for input grid (total domain)	71
ke_in_tot	INT	ke for input grid (total domain)	60
nlevskip	INT	number of missing levels in input grid	0
ke_soil_in	INT	number of levels in multi-layer soil input model	6
czml_soil_in	REAL	depth of bottom level of soil input layers. The default specification is (in cm) / 0.005, 0.02, 0.06, 0.18, 0.54, 1.62, 4.86, 14.58 /	see left
pcontrol_fi	REAL	pressure of control level for geopotential	-1.0
pollat_in	REAL	geographical latitude of rotated north pole (in degrees, north > 0)	90.0
pollon_in	REAL	geographical longitude of rotated north pole (in degrees, east > 0)	180.0
dlon_in	REAL	Mesh size in east-west direction	0.5
dlat_in	REAL	Mesh size in north-south direction	0.5
startlat_in_tot	REAL	latitude of the lower left grid point of the input domain (in degrees, north > 0, rotated coordinates)	-35.0
startlon_in_tot	REAL	longitude of the lower left grid point of the input domain (in degrees, east > 0, rotated coordinates)	-30.0
endlat_in_tot	REAL	latitude of the upper right grid point of the input domain (in degrees, north > 0, rotated coordinates)	0.0
endlon_in_tot	REAL	longitude of the upper right grid point of the input domain (in degrees, east > 0, rotated coordinates)	-40.0
p0sl_in	REAL	constant reference pressure on sea-level	10000.0
t0sl_in	REAL	constant reference temperature on sea-level	288.15

Name	Type	Definition / Purpose / Comments	Default
dt0lp_in	REAL	d (t0) / d (ln p0)	42.0
lushift_in	LOG	shift of u-velocity due to grid staggering	.FALSE.
lvshift_in	LOG	shift of v-velocity due to grid staggering	.FALSE.
east_add_in	INT	add an extra column to the east	0
west_add_in	INT	add an extra column to the west	0
south_add_in	INT	add an extra column to the south	0
north_add_in	INT	add an extra column to the north	0

7.3 LMGRID — Specifying the Domain and the Model Grid

lmgrid contains variables that specify the model domain in the rotated grid and the size of the total domain.

Name	Type	Definition / Purpose / Comments	Default
pollat	REAL	geographical latitude of rotated north pole (in degrees, north > 0)	32.5
pollon	REAL	geographical longitude of rotated north pole (in degrees, east > 0)	-170.0
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
dlon	REAL	Mesh size in east-west direction	0.0625
dlat	REAL	Mesh size in north-south direction	0.0625
startlat_tot	REAL	latitude of the lower left grid point of the total domain (in degrees, north > 0, rotated coordinates)	-14.375
startlon_tot	REAL	longitude of the lower left grid point of the total domain (in degrees, east > 0, rotated coordinates)	-6.875
ielm_tot	INT	number of gridpoints of the total domain in east-west direction	213
jelm_tot	INT	number of gridpoints of the total domain in north-south direction	213
kelm_tot	INT	number of gridpoints of the total domain in vertical direction	20
ke_soil_lm	INT	number of levels in multi-layer soil model for the COSMO-Model	6
czml_soil_lm	REAL	depth of bottom level of soil layers for the COSMO-Model. The default specification is / 0.005, 0.02, 0.06, 0.18, 0.54, 1.62, 4.86, 14.58 /	see left
czvw_so_lm	REAL	artificial volumetric soil water content profile for the COSMO-Model.	/0.75/

Name	Type	Definition / Purpose / Comments	Default
irefatm	INT	type of reference atmosphere 1: Default as used up to now 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.	1
lanalyt_calc_t0p0	LOG	if set to .TRUE., the values for the reference state of t_0 and p_0 are computed analytically. If it is set to .FALSE., they are only averaged between the half levels.	.FALSE.
ivctype	INT	kind of vertical coordinate system 1: reference pressure based hybrid coordinate. 2: height based hybrid Gal-Chen coordinate. 3: height based hybrid SLEVE coordinate. In comparison to option 2, there is an additional blending to a smoothed orography before blending to horizontal coordinates at the height given by the parameter vcflat below. (not extensively tested). The exact blending behaviour can be influenced by the parameters svc1 and svc2 below.	2
lnewVGrid	INT	to indicate, that a new vertical grid file HHL has to be created	.FALSE.
vcflat	REAL	coordinate value where system changes back to z-system	0.220

Name	Type	Definition / Purpose / Comments	Default
vcoord_d	REAL	vertical coordinate parameter list of pressure (ivctype=1) or height (ivctype=2 or ivctype=3) values. The usage of the fortran utility vcoord.f90 for preparation of the parameter values is recommended. For ivctype=1, values have to cover the range 0 to 1 in increasing order (σ coordinates). For ivctype=2, values are heights in m and have to be in decreasing order from the desired model top height down to 0.0 m. Some default sets of height values already exist in the source code for ivctype=1 and kelm_tot=20,32,35,40 and ivctype=2 and kelm_tot=20,35,40,50. If one of these sets should be used, only specify the respective kelm_tot and ivctype and leave vcoord_d out.	(missing)
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
svc1	REAL	decay rate for large-scale part of topography	10000.0
svc2	REAL	decay rate for small-scale part of topography	10000.0
nfltvc	REAL	number of filter applications for topo decomposition	100

7.4 DATABASE — Specification of Database Job

Name	Type	Meaning	Default
yinit_order	CHAR	string for initializing csodaban	'ak=nix'
yana_tab	CHAR	database for initial data	'*****'
ybd_tab	CHAR	database for boundary data	'*****'
nout_sockets	INT	number of sockets for database output per PE (0 means File-IO)	0
nin_sockets	INT	number of sockets for database input per PE (0 means File-IO; has to be ≤ 1)	0
iretry	INT	number of seconds to retry on database failure	0
ibackup_size	INT	size of incore backup space (in bytes) by a database failure	-1
ybackup_dir	CHAR	directory for outcore backup	'.'
idbg_level	INT	debug level for mpe_io	0

7.5 DATA – Controlling the Grib Input and Output

General control variables:

Name	Type	Meaning	Default
ncenter	INT	originating center identification	78
nprocess_ini	INT	generating process identification for initial values	131
nprocess_bd	INT	generating process identification for boundary values	132
nrbit	INT	pack-rate for the grib code (in bit)	16
nbitmap	INT	user dimension for bitmaps	6000*2
nl_soil_lm	INT	number of prognostic soil water levels for the COSMO data	2
nl_soil_in	INT	number of prognostic soil water levels for the input data	2
nvers	INT	for documenting purposes (mainly in GRIB-Code)	1
ymode_read	CHAR	specify open mode for reading	'r '
ymode_write	CHAR	specify open mode for writing	'w '
yvarini	CHAR	list of variables for LM initial data	
yvarbd	CHAR	list of variables for LM boundary data	
l_ke_in_gds	LOG	write the number of vertical levels explicitly to the grid description section	.TRUE.
ytunit_in	CHAR	time unit for input data	'f'
ytunit_out	CHAR	time unit for output data	'f'
yinput_type	CHAR	type of input data 'forecast' forecast data 'analysis' analysis data 'ana_init' initialized analysis data	'forecast'

Variables for external data and HHL-files (needed for GRIB2):

Name	Type	Meaning	Default
ylmext_cat	CHAR	directory of the external fields for LM/HM	' '
ylmext_lfn	CHAR	name of the file with the external fields for LM/HM	' '
ylm_hhl	CHAR	name of the vertical grid HHL-file that has to be processed in case of COSMO GRIB2 output files. This file has to be in the directory ylmext_cat	' '
yinext_cat	CHAR	directory of the external fields for GME	' '
yinext_lfn	CHAR	name of the file with the external fields for GME	' '
yin_hhl	CHAR	name of the vertical grid HHL-file that has to be read in case of COSMO GRIB2 input files. This file has to be in the directory yinext_cat	' '
ylmext_form_read	CHAR	input format of external data	'grb1'
yinext_form_read	CHAR	input format of external data from coarse grid 'grb1' input is read with DWD's Grib library and has to be GRIB1. 'apix' input is read with ECMWF's Grib library grib_api and can be Grib1 or Grib2. 'ncdf' input format is NetCDF.	'grb1'
ie_ext	INT	west-east size of fields with external parameters	1081
je_ext	INT	north-south size of fields with external parameters	1081

Variables for the models

Name	Type	Meaning	Default
yin_cat	CHAR	directory of the GME-fields	' '
yin_form_read	CHAR	input format of data from coarse grid	'grb1'
ybitmap_cat	CHAR	directory of an optional bitmap for GME data	' '
ybitmap_lfn	CHAR	name of the file with an optional bitmap for GME data	' '
ylm_cat	CHAR	directory of the LM/HM-fields	' '
ylm_form_write	CHAR	output format of COSMO-Model data 'grb1' data are written with DWD's Grib library in GRIB1. 'api1' data are written with ECMWF's Grib library grib_api in Grib1 'api2' data are written with ECMWF's Grib library grib_api in Grib2 'ncdf' data are written in NetCDF.	'grb1'
npstrframe	INT	thickness of output frames	8
lbd_frame	LOG	if .TRUE., boundary fields include only frames	.FALSE.

Additional specifications for NetCDF-IO:

yncglob_institution	CHAR	originating center name	'-'
yncglob_title	CHAR	title string for the output	'-'
yncglob_source	CHAR	program name and version	'-'
yncglob_contact	CHAR	contact e.g. email address	'-'
yncglob_project_id	CHAR	identification of the project of simulation	'-'
yncglob_experiment_id	CHAR	identification of the experiment of simulation	'-'
yncglob_references	CHAR	URL, report etc.	'-'
ncglob_realization	INT	number of the realization of the experiment	1

7.6 PRICTR — Controlling grid point output

Name	Type	Meaning	Default
nlev1pr	INT	k-index for printing the first model layer	10
nlev2pr	INT	k-index for printing the second model layer	20
igp_tot	INT	i-index for printing selected grid points (max. nmaxgp)	
jgp_tot	INT	j-index for printing selected grid points (max. nmaxgp)	
lprps	LOG	print some ps- and fis-fields	.FALSE.
lprt	LOG	print t at 2 levels (nlev1pr,nlev2)	.FALSE.
lpru	LOG	print u at 2 levels (nlev1pr,nlev2)	.FALSE.
lprv	LOG	print v at 2 levels (nlev1pr,nlev2)	.FALSE.
lprgrh	LOG	print grh at 2 levels (nlev1pr,nlev2)	.FALSE.
lprqv	LOG	print qv at 2 levels (nlev1pr,nlev2)	.FALSE.
lprqc	LOG	print qc at 2 levels (nlev1pr,nlev2)	.FALSE.
lprud	LOG	print ud (divergent wind correction)	.FALSE.
lprvd	LOG	print vd (divergent wind correction)	.FALSE.
lprdpdt	LOG	print dpdt (tendency of surface pressure)	.FALSE.
lprgp	LOG	print profiles at selected grid points	.FALSE.
lchkin	LOG	print check-values of input-fields	.FALSE.
lchkout	LOG	print check-values of output-fields	.FALSE.

7.7 EPSCTL — Characterizations for the Ensemble of Boundary Data

The namelist group EPSCTL is only read, if the switch leps_bc in the group CONTRL is set to .TRUE..

Name	Type	Meaning	Default
iepsmem_bc	INT	ID of the member in the ensemble of boundary conditions (must be ≥ 0.	-1
iepstyp_bc	INT	ID of the boundary ensemble generation type (must be ≥ 0.	-1
iepstot_bc	INT	total number of boundary ensemble members (must be ≥ 0.	0
lchk_bc_typ	LOG	if .TRUE., check member ID of input data	.FALSE.