Consortium

for

Small-Scale Modelling

Technical Report No. 24

Online Trajectory Module in COSMO: A short user guide

September 2014

Version 1.0

DOI: 10.5676/DWD_pub/nwv/cosmo-tr_24

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

www.cosmo-model.org

Editor: Massimo Milelli, ARPA Piemonte

Online Trajectory Module in COSMO:

A short user guide

by

***A. K. Miltenberger*[*], *A. Roches*[**], *S. Pfahl*[*], *H. Wernli*[*]**

[*]Institute for Atmospheric and Climate Science, ETHZ
Universitätstrasse 16
8092 Zürich
Switzerland
[**]C2SM
Universitätstrasse 16
8092 Zürich
Switzerland

Revision	**Modifications**	**Date**	**Author**
1.0	Initial Release	September 2014	Annette K. Miltenberger (ETHZ) Anne Roches (C2SM) Tobias Selz (LMU)

Abstract

An online trajectory module is implemented in COSMO 5.1 and higher versions. The module is a diagnostic tool that enables to compute trajectories (i.e., to follow air parcels) and to trace any prognostic variable as well as diagnostic variables along the trajectories. The trajectory position is updated at each model time step based on the model wind fields, which allows to calculate trajectories very accurately for high spatial resolutions.

1 Introduction

The Lagrangian description of atmospheric motion is a valuable approach to study numerous phenomena ranging from large spatial scales, as extratropical cyclones (e.g., Wernli and Davies, 1997), to small scales, as the entrainment in stratocumulus clouds (e.g., Stevens et al., 1996). It is also very common to investigate the dispersion of atmospheric pollutants (e.g., Weil et al., 2012).

While many trajectory models for offline computation are currently available, they rely on the output of numerical weather prediction models. To capture the spatial and temporal variability of the wind field, that can be represented in the Eulerian model at high spatial resolution, model output would be required at a minute-scale temporal resolution (about 5 min for 2.2 km grid spacing) for the trajectory calculation. The temporal resolution of the wind field data has been shown to significantly influence the trajectory path (e.g., Rössler et al. 1992). It thus appears very attractive for a NWP model to incorporate a module for online trajectory computation, which allows to use the wind field at every model timestep and hence allows a very accurate calculation of trajectories also for high resolution applications.

This technical report describes shortly the implementation of the online trajectory module in the official COSMO code and presents the technical aspects that are relevant for the user of the module. It is based on the user guide published as supplement material of Miltenberger et al. (2013). Discrepancies that can be observed between both documents reflect the changes made to the code between the first version and the version that is currently implemented in the official COSMO version 5.1 and higher.

For a detailed documentation, the reader should refer to Miltenberger et al. (2013), where the scientific rational, the workflow and an example of scientific application are provided. The module has been tested on several platforms and applied in multiple research projects, demonstrating its versatility and scientific soundness.

2 Scientific rational

The scientific reasoning for implementing an online trajectory module into an atmospheric model is thoroughly described in Miltenberger et al. (2013). This paper is freely accessible at *http://www.geosci-model-dev.net/6/1989/2013/gmd-6-1989-2013.pdf.*

This comprehensive scientific documentation provides also insights in the workflow and the trajectory integration scheme. It discusses error sources for the trajectory calculation and their relative importance and assesses the overall performance of the model. In addition a case study of an alpine foehn event is presented, demonstrating the scientific soundness of the method and the advantage of the online trajectory computation over offline trajectory computation using standard COSMO model output.

3 Implementation overview

The online trajectory module consists of two different source files, namely `data_traj.f90` and `src_traj.f90`. While the first file contains the definition of datatypes and the declaration of some variables used across the whole module (in particular some maximal values for the dimensioning of module arrays), the allocation of the module variables, the forward integration of the trajectories, the communication between the processors and the I/O subroutines are available in the second file. The interfaces to the rest of the COSMO code are slim: four calls to the organizing routine of the module in `lmorg.f90` and one call in `organize_data` as well as a global switch to activate/deactivate the module.

The module is controlled by a namelist and requires a text file containing the start positions of the trajectories (startfile).

In the following sections the technical details related to the use of the module and its implementation are described. Section 4 provides an overview of all relevant namelist swiches and section 5 explains how the start information of the trajectories should be specified. Reading these two sections should be enough to setup a trajectory computation. In section 6, more information about the output files produced by the trajectory module is provided while section 7 details the current implementation and should ease code comprehension and further extension.

4 Namelist settings

The online trajectory module can be switched on and off by an additional namelist parameter in the namelist group `RUNCTL` named `ltraj`, which is set to `FALSE` by default. The user-defined parameters for the trajectory calculation can be specified in a separate namelist named `INPUT_TRAJ` in a namelist group called `TRAJCTL`. These parameters are described below.

- `istart_mode_traj` : Specification mode of the trajectory start times:
  - `istart_mode_traj= 1` : A comma separated list of start times has to be given through the namelist parameter `hstart_traj` or `nstart_traj`. The list contains the start times in hours after model start (`hstart_traj`) or in model time steps (`nstart_traj`). Trajectories are always started at the same spatial locations for each specified start time. `nstart_traj` has priority over `hstart_traj` if both are specified.
  - `istart_mode_traj = 2` : Trajectories are started repeatedly at fixed time intervals from the same spatial locations. The first start time (in hours after the model start), the last start time (in hours after the model start) and the time interval (in hours) are specified through the namelist parameter `hcomb_start_traj` (a comma separated list of the three elements). The same triplet can be expressed in model time steps through the namelist parameter `ncomb_start_traj`. `ncomb_start_traj` has priority over `hcomb_start_traj` if both are specified.

  - istart_mode_traj = 3 : Trajectories are started repeatedly at the times and locations provided in the startfile. The start times should be provided in hours and the entries in the startfile should be sorted according to starting time. Except for nstop_traj resp. hstop_traj all other namelist parameters pertaining to the start times (hcomb_start_traj, ncomb_start_traj, hstart_traj, nstart_traj) have no effect.
- startfile_traj : Path of the start file. The path can be a relative path and is limited to 100 characters. A description of the start file is provided in section 5.
- hstop_traj or nstop_traj: Time at which the trajectory calculation stops in hours after model start or in model time steps, respectively. It only has to be specified if the trajectory calculation should stop before the model integration. It has to be a multiple of the model time step. nstop_traj has priority over hstop_traj if both are specified.
- ydir_traj : Directory to which the trajectory output will be written. The directory specification is limited to 100 characters.
- hinc_out_traj or ninc_out_traj: Output frequency of the trajectory positions (and traced variables) in hours or in number of model time steps respectively. hinc_out_traj has to be a multiple of the model time step and ninc_out_traj has to be a whole number. If both parameters are specified, ninc_out_traj has priority over hinc_out_traj.
- tracevar_traj : Names of the variables which should be traced along the trajectories. The naming should be consistent with the table in the COSMO source file src_setup_vartab.f90, which is also used for the COSMO model output.

An example of the namelist TRAJCTL is given in Listing 1.

Listing 1: Namelist example

```
cat > INPUT_TRAJ << end_input_traj
 &TRAJCTL
  nstop_traj= 4320,
  hstop_traj = 24.0,
  istart_mode_traj = 1,
    hstart_traj = 3.0,6.0,12.0,
    nstart_traj = 540,1080,2160,
    !hcomb_start_traj=
    !ncomb_start_traj=
  startfile_traj = './startfile_case_n.txt',
  ydir_traj = './output_traj',
  !hinc_out_traj = 0.0333,
  ninc_out_traj = 6,
  tracevar_traj = 'T','P','QV','QC','QR','RELHUM',
/
end_input_traj
```

5 Start file

A start file containing the start positions (and times) of the trajectories has to be provided. It is specified by the user via the namelist parameter startfile_traj (see section 4). It contains the rotated longitudes, the rotated latitudes and the height and optionally the times at which the trajectories are started.

It is a textfile and it must comply with strict formatting rules.

The first 3 lines are header lines:

- First line: It contains the reference date, i.e., the date at which the COSMO simulation starts. This line is not considered by the program.
- Second line: Description of the file data columns. It should either be "time lon lat z" or "lon lat z" (without quotes). Only the first four non blank characters of this line will be analyzed by the program and in case the first four characters match the keyword time, the program will read the first data column as being the start times and expect the second column to be the rotated longitudes. If the first four characters do not match the keyword time, the first data column is assumed to contain the rotated longitudes.
- Third line: It is not relevant for the program. It traditionally contains only "-------" symbols.

The following lines in the file are content lines. In case the keyword time is present on the second line, the program reads the first column as the start times, the second as the rotated longitudes, the third as the rotated latitudes and the fourth as the height. If the keyword time is not present on the second line, the first column is read as the rotated longitudes, the second as the rotated latitudes and the third as the height. Column delimiters are blank spaces. The longitudes have to range between -180.0 and 180.0 degrees and the latitudes can be comprised between -90.0 and 90.0 degrees. They are expressed in the same referential than COSMO (i.e., same rotated pole coordinates). The height has to be given in meters AMSL. If time data is provided, it must be in hours after model start. The entries in the startfile should be sorted according to starting time.

The start file is read in an endless loop and all lines (except the 3 header lines) are considered as data.

Make sure that your start file fulfills the format requirements explained above!

A startfile should look similar to the example provided in Listing 2.

Listing 2: Start file example

```
Reference Date 19870725_0000
time lon lat z
----------------
 10.00   -8.000  -2.500 9000.000
 10.00   -8.000  -2.500 9100.000
 10.00   -8.000  -2.500 9200.000
```

6 Output

For all trajectories, their position as well as the value of the traced variables are written to file at the output frequency specified by the user via the namelist parameter hinc_out_traj or ninc_out_traj (see section 4) in the output directory specified by the user (ydir_traj, see section 4). Only NetCDF format is supported.

The output is split in the following way: The trajectories that start together at a given start time are considered as a group. If the group is too large to be contained in one single file, the trajectories are further split over several files, according to their position in the start file. A limit of 2 GB is set as maximal file size in order to ease postprocessing. At all output time steps comprised between the start of the trajectory group and the end of the trajectory simulation (see namelist parameter `hstop_traj` or `nstop_traj` in section 4), the positions and traced variable values for the group of trajectories are written to one (or several) files. The files can thus have different lengths (for different start times).

One or several files `traj_tXXXXXX_pYYY.nc` are thus produced, where `XXXXXX` is the start time of the trajectory group expressed in minutes after the model start and `YYY` is the number of the file in the file set containing the trajectories starting at `XXXXXX` after the model start.

The output files follow the CF conventions. Some global attributes (reference date, duration of the simulation for the group of trajectories, coordinates of the rotated pole and output frequency) are written to each file. A variable `time` indicates the time elapsed since the model start in seconds. For each instance of `time` (first dimension) and for each trajectory identified by its ID (`id` as second dimension), the longitude, latitude and height (variables `longitude`, `latitude` and `z`) are written as well as the value of the traced variables. For each of these variables, the short name, the long name and the units are provided as attributes.

The header of an output file (here for 13507 trajectories starting after 120 minutes) thus looks like the example provided in Listing 3.

Listing 3: Output file example

```
netcdf traj_t000120_p001 {
dimensions:
        id = 13507 ;
        time = 541 ;
variables:
        float time(time) ;
                time:standard_name = "time" ;
                time:long_name = "time" ;
                time:units = "seconds" ;
        float longitude(time, id) ;
                longitude:standard_name = "grid_longitude" ;
                longitude:long_name = "rotated longitudes" ;
                longitude:units = "degrees" ;
        float latitude(time, id) ;
                latitude:standard_name = "grid_latitude" ;
                latitude:long_name = "rotated latitudes" ;
                latitude:units = "degrees" ;
        float z(time, id) ;
                z:standard_name = "height" ;
                z:long_name = "height above mean sea level" ;
                z:units = "m AMSL" ;
        float T(time, id) ;
                T:standard_name = "air_temperature" ;
                T:long_name = "temperature" ;
                T:units = "K" ;
        float P(time, id) ;
                P:standard_name = "air_pressure" ;
                P:long_name = "pressure" ;
                P:units = "Pa" ;
        float U(time, id) ;
                U:standard_name = "grid_eastward_wind" ;
                U:long_name = "U-component of wind" ;
                U:units = "m s-1" ;
        float QV(time, id) ;
                QV:standard_name = "specific_humidity" ;
                QV:long_name = "specific humidity" ;
                QV:units = "kg kg-1" ;
        float QI(time, id) ;
                QI:standard_name = "mass_fraction_of_cloud_ice_in_air
                    " ;
                QI:long_name = "specific cloud ice content" ;
                QI:units = "kg kg-1" ;
        float RELHUM(time, id) ;
                RELHUM:standard_name = "relative_humidity" ;
                RELHUM:long_name = "relative humidity" ;
                RELHUM:units = "%" ;

// global attributes:
                :ref_year = 2011 ;
                :ref_month = 11 ;
                :ref_day = 20 ;
                :ref_hour = 12 ;
                :ref_min = 0 ;
                :ref_sec = 0 ;
                :duration_in_sec = 10800. ;
                :pollon = -171.5 ;
                :pollat = 47.5 ;
                :output_timestep_in_sec = 20. ;
}
```

7 Implementation details

7.1 Workflow overview

An organizing routine `organize_traj` is called four times from `lmorg.f90`:

1. Once at the beginning of the initialization phase with the action `'input'`. The namelist `INPUT_TRAJ` is read as well as the start file. Necessary checks are performed in order to ensure that the specifications made by the user are meaningful. The user can proof the settings by inspecting the `YUSPECIF` file.

2. Once later on during the initialization phase with the action `'init'`. Arrays required for the whole trajectory simulation are allocated. A data structure is filled for all traced variables (see section 7.4). The rank of the eight surrounding processing elements for each processing element is defined in a parallel case. The position of the trajectories are initialized using the information read from the start file. Also, a status flag indicating if the trajectory has to be computed (`alive`), has left the domain (`dead`) or is not yet started (`embryonic`) is initialized using the start times read from or derived from the namelist settings (see section 4). In case the simulation is started from a restart file, the start positions are read directly from the restart file instead of the start file. The number of required output files is computed and the split of the trajectories among these files is determined. All NetCDF files are created and the global attributes are

written to the files. The variables are defined and their attributes written. In case the simulation is started from a restart file, this information is read from already existing output files. The NetCDF files will be kept open for the whole trajectory simulation.

3. Once at the end of each time step, where the other diagnostics are computed, with the action `'compute'`. For each compute time step of the trajectories (i.e., for each time step comprised between the start of the first trajectory and the end of the trajectory computation, both specified by the user via namelist parameters), it is checked for each trajectory, if it should be computed or not on each processor. In case it has to be computed, the forward integration is performed (see Miltenberger et al., 2013 for more details) and in case the trajectory leaves the processor domain after its integration, it is added to a buffer for communicating the new position to the relevant neighbor processor. After this has been done for all trajectories, the communication of all trajectories that have to be passed to a neighbor processor is performed. Finally output is performed, if the time step corresponds to an output time step.

4. Once at the end of the simulation with the action `'finalize'`. All NetCDF files are closed and the memory used for the trajectory simulation is freed.

In addition an organizing routine `organize_traj_restart('restart')` is called from `organize_data('results')` at the end of the time step in order to write out the trajectories to an extra restart file if needed.

7.2 Maximal values for array allocation

Some restrictions have to be set regarding the maximal size of arrays used in the module. All the maximal values are defined as parameters in the module `data_traj.f90`.
In case these maximal values have to be increased, one can simply change the values of the parameters listed below in `data_traj.f90` and test, if the model still works on the platform used.

The parameters are:

- `nmax_traj_starttime=100`: Maximal number of trajectory start times. It means that up to 100 values can be provided by the user through the namelist parameters `hstarts_traj` or `nstarts_traj` or that the namelist triplet `hcomb_start_traj` or `ncomb_start_traj` does not lead to more than 100 values. This parameter should be increased with caution!

- `nmax_traj_startpt=50000`: Maximal number of trajectories that can start at a given start time (from different locations). It thus means that a maximum of `100*50000=5 mio.` trajectories is imposed. Please note that so far, the module has been tested only with up to about 2 mio. of trajectories. If increasing this parameter or the previous one, intensive testing should be done.

- `nmax_traced_var=30`: Maximal number of variables that can be traced along the trajectories. This parameter could be increased, if more variables should be traced, but testing is recommended.

- `nmax_out_per_start_t=100`: Maximal number of output files per start time. It corresponds to the part `YYY` in the output filename `traj_tXXXXXX_pYYY.nc`. Please note that this could not be increased to more than 999 without modifying the naming scheme of the output files. At the moment, we cannot produce more than `100*100=10'000` output files (`nmax_traj_starttime` times `nmax_out_per_start_t`).

- `nmax_nc_size=2*2`^{30}: Maximal size of an output file. The file size is currently limited to 2 GB in order to ease post-processing. If increasing this value, some post-processing facilities might fail.
- `header_nc_size=2*2`^{20}: Reserved space for the header in an output file (2 MB). If more information is added to the header in the future, it might be necessary to increase this number.

7.3 Restrictions of use

The periodic boundary conditions are not yet implemented for the trajectories. The trajectory module can be used in combination with periodic boundary conditions (`lperi_x` and `lperi_y`) and the trajectories will be computed using periodic winds but the trajectories themselves are not cycled.

Other configurations are supported.

7.4 Traced variables

A new data type `trajtrace_type` is defined in `data_traj.f90` to hold the needed information to handle properly the traced variables. Most of the information is inherited from the `var` structure available in `src_setup_vartab.f90`. It contains the following elements:

- `name`: name of the traced variable
- `levtyp`: level type (GRIB related)
- `ntri`: time range indicator (GRIB related)
- `rank`: rank
- `p4`: pointer to a 4-dimensional field
- `p3`: pointer to a 3-dimensional field
- `idef_stat`: status indicating if the variable has been computed already or if it should be derived from other model variables
- `istag`: staggering flag (`0`: no staggering; `1`: zonal staggering; `2`: meridional staggering; `3`: vertical staggering)
- `units`: units (NetCDF related)
- `sdname`: standard name (NetCDF related)
- `lgname`: long name (NetCDF related)

Only COSMO model variables can be traced. The name of the traced variables must match exactly the name of the corresponding variable in the `var` structure (in `src_setup_vartab.f90`). Only 4-dimensional and 3-dimensional variables in space can be traced. The traced variables must be instantaneous values (i.e., have a time range indicator of 0), since it is not clear what should be traced in case of accumulated values or statistical quantities (mean, min, max). The variables can be on the half-levels or on the full-levels. They can be staggered in zonal direction (`U`) or in meridional direction (`V`). All variables that are valid at each time step in the model (`idef_stat=1` in `src_setup_vartab.f90`), i.e., all prognostic variables and several diagnostics, can be traced. The other model variables (`idef_stat=0` in `src_setup_vartab.f90`) must be derived before being written to the COSMO output. Only a subset of these variables has been implemented for tracing so far:

- `P`: total pressure
- `RELHUM`: relative humidity
- `POT_VORTIC`: potential vorticity

Other variables could easily be implemented by making use of the subroutines used in `src_output.f90` (e.g., `curl`).

At each output step the traced variables are interpolated to the trajectory position using a linear interpolation. The computed values for the traced variables as well as the trajectory ID and position (longitude, latitude and height) are gathered on processor 0, which writes the data to file.

7.5 Communication strategy

For each trajectory that leaves a processor, its ID as well as the three components of its position (longitude, latitude and height) are packed in a buffer dimensioned with the 8 neighbors and with the number of trajectories times the four needed pieces of information (ID, longitude, latitude and time). After having looped over all trajectories, actual communication is performed. For this, a "pairwise" non-blocking (`MPI_ISEND` and `MPI_IRECV`) communication strategy has been chosen. Two slices of the buffer for opposite neighbor processors (e.g., northern and southern neighbor) are communicated at a time. This is faster than communicating in each direction one after the other and requires less memory than a communication in all directions at a time. This solution is thus a compromise between performance in terms of communication speed and memory consumption. This could easily be changed if desired. The `MPI_IRECV` are pre-posted in order to improve performance.

7.6 Trajectory positions

The trajectory positions in the index space must be determined several times during a trajectory computation time step: To determine the winds at the trajectory position in order to update its position, to interpolate the orography at the trajectory position (in order to check if the trajectory has reached the ground), and finally to interpolate the traced variables to the trajectory location in case output must be performed. This computation is sensitive to the reference system and rounding errors thus occur when changing the domain decomposition. They are very small (typically differences of O(10^{-9}) for the longitudes and latitudes, O(10^{-6}) for height, O(10^{-7}) for traced temperature, O(10^{-8}) for traced wind or pressure, O(10^{-10}) for the traced humidity variables are observed) and do not affect the physical interpretation of the trajectories.

The required horizontal interpolation combined with the formulation of the lower boundary condition in the model (wind extrapolation) leads to trajectories that intersect the terrain, which is physically impossible. Any trajectory that reaches the ground is then lost for the rest of the simulation. In order to avoid unreasonable loss of trajectories, a trajectory that reaches the ground is reset to 10 meters above ground during the Euler forward integration (see Miltenberger et al., 2013 for more details). A hardcoded switch `ljump` controls this feature and could easily be set to `FALSE`, in order to let the trajectories leave the domain at the ground during the forward integration. Similarly, the value of the hardcoded parameter `hjump` that defines the height, at which the trajectories are reset (currently 10 meters above ground) can easily be modified.

References

Miltenberger A.K., Pfahl S., Wernli H., 2013: An online trajectory module (version 1.0) for the nonhydrostatic numerical weather prediction model COSMO. *Geosci. Model Dev.* **6**, 1989-2004, doi:10.5194/gmd-6-1989-2013.

Röessler, C. E., Paccagnella, T., Tibaldi, S., 1992: A three-dimensional atmospheric trajectory model: Application to a case study of Alpine lee cyclogenesis. *Meteorol. Atmos. Phys* **50**, 211-229.

Stevens B., Feingold G., Cotton W. R., Walko R. L., 1996: Elements of the microphysical structure of numerically simulated nonprecipitating stratocumulus. *J. Atmos. Sci.* **53**, 980-1006, doi:10.1175/1520-0469(1996)053<0980:EOTMSO>2.0.CO;2.

Weil J., Sullivan P.P., Patton E., Moeng C.-H., 2012: Statistical variability of dispersion in the convective boundary layer: Ensembles of simulations and observations. *Bound.-Lay. Meteorol.* **145**, 185-210, doi:10.1007/s10546-012-9704-y.

Wernli H., Davies H.C., 1997: A lagrangian-based analysis of extratropical cyclones. I: The method and some applications. *Q.J.R. Meteorol. Soc.* **123**, 467-489, doi:10.1002/qj.49712353811.

List of COSMO Newsletters and Technical Reports

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

COSMO Newsletters

No. 1: February 2001.

No. 2: February 2002.

No. 3: February 2003.

No. 4: February 2004.

No. 5: April 2005.

No. 6: July 2006.

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

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

No. 9: December 2008.

No. 10: March 2010.

No. 11: April 2011.

No. 12: April 2012.

No. 13: April 2013.

No. 14: April 2014.

COSMO Technical Reports

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

No. 22: C. Marsigli, T. Diomede, A. Montani, T. Paccagnella, P. Louka, F. Gofa, A. Corigliano (2013):
The CONSENS Priority Project.

No. 23: M. Baldauf, O. Fuhrer, M. J. Kurowski, G. de Morsier, M. Müllner, Z. P. Piotrowski, B. Rosa, P. L. Vitagliano, D. Wójcik, M. Ziemiański (2013):
The COSMO Priority Project 'Conservative Dynamical Core' Final Report.

COSMO Technical Reports

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

The Technical Reports are intended

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

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

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

For any comments and questions, please contact the editors:

Massimo Milelli
Massimo.Milelli@arpa.piemonte.it