

ICON Namelist Overview

November 15, 2012

Contents

| | | |
|----------|--|-----------|
| 1 | ICON Namelists | 4 |
| 1.1 | Scripts, Namelist files and Programs | 4 |
| 1.2 | Namelist parameters | 4 |
| 2 | Namelist parameters for grid generation | 5 |
| 2.1 | Namelist parameters defining the atmosphere grid | 5 |
| 2.1.1 | graph_ini (NAMELIST_GRAPH) | 5 |
| 2.1.2 | grid_ini (NAMELIST_GRID) | 5 |
| 2.1.3 | grid_options (NAMELIST_GRID) | 5 |
| 2.1.4 | plane_options (NAMELIST_GRID) | 6 |
| 2.1.5 | gridref_ini (NAMELIST_GRIDREF) | 6 |
| 2.2 | Namelist parameters defining the local grid generation | 8 |
| 2.2.1 | grid_geometry_conditions | 8 |
| 2.2.2 | local_grid_optimization | 9 |
| 2.2.3 | create_ocean_grid | 10 |
| 2.2.4 | torus_grid_parameters | 11 |
| 3 | Namelist parameters defining the ICON model | 11 |
| 3.1 | master_nml | 11 |
| 3.2 | master_model_nml (repeated for each model) | 11 |
| 3.3 | time_nml | 12 |
| 3.4 | parallel_nml | 13 |

| | | |
|----------|---|-----------|
| 3.5 | coupling_nml | 15 |
| 3.6 | run_nml | 15 |
| 3.7 | grid_nml | 18 |
| 3.8 | gridref_nml | 20 |
| 3.9 | prepicon_nml | 22 |
| 3.10 | remap_nml | 24 |
| 3.11 | input_field_nml | 24 |
| 3.12 | interpol_nml | 25 |
| 3.13 | dynamics_nml | 27 |
| 3.14 | ha_dyn_nml | 27 |
| 3.15 | nonhydrostatic_nml (relevant if run_nml:iequations=3) | 28 |
| 3.16 | sleve_nml (relevant if nonhydrostatic_nml:ivctype=2) | 31 |
| 3.17 | diffusion_nml | 32 |
| 3.18 | io_nml | 34 |
| 3.19 | output_nml | 36 |
| 3.20 | meteogram_output_nml | 40 |
| 3.21 | nh_pzlev_nml | 40 |
| 3.22 | transport_nml (used if run_nml/ltransport=.TRUE.) | 41 |
| 3.23 | nwp_phy_nml | 43 |
| 3.24 | radiation_nml | 45 |
| 3.25 | nwp_lnd_nml | 47 |
| 3.26 | echam_phy_nml | 48 |
| 3.27 | echam_conv_nml | 49 |
| 3.28 | vdiff_nml | 50 |
| 3.29 | turbdiff_nml | 50 |
| 3.30 | gw_hines_nml (Scope: lgw_hines = .TRUE. in echam_phy_nml) | 51 |
| 3.31 | ocean_physics_nml | 52 |
| 4 | Namelist parameters for testcases (NAMELIST_ICON) | 53 |
| 4.1 | ha_testcase_nml (Scope: ltestcase=.TRUE. and iequations=[0,1,2] in run_nml) | 53 |
| 4.2 | nh_testcase_nml (Scope: ltestcase=.TRUE. and iequations=3 in run_nml) | 55 |
| 5 | External data | 62 |
| 5.1 | extpar_nml (Scope: itopo=1 in run_nml) | 62 |

| | | |
|----------|---|-----------|
| 6 | External packages | 63 |
| 6.1 | art_nml | 63 |
| 7 | Information on vertical level distribution | 63 |

1 ICON Namelists

1.1 Scripts, Namelist files and Programs

Run scripts starting the programs for the grid generation and the models are stored in run/. These scripts write namelist files containing the specified Fortran namelists. Programs are stored in <icon home>/build/<architecture>/bin/.

Table 1: Namelist files

| Namelist file | Purpose | Made by script | Used by program |
|---------------------|---------------------|-------------------------|-----------------|
| NAMELIST_GRAPH | Generate graphs | create_global_grids.run | grid_command |
| NAMELIST_GRID | Generate grids | create_global_grids.run | grid_command |
| NAMELIST_GRIDREF | Gen. nested domains | create_global_grids.run | grid_command |
| NAMELIST_OCEAN_GRID | Gen. ocean grid | create_ocean_grid.run | grid_command |
| NAMELIST_TORUS_GRID | Gen. torus grid | create_torus_grid.run | grid_command |
| NAMELIST_ICON | Run ICON models | exp.<name>.run | control_model |

1.2 Namelist parameters

The following subsections tabulate all available Fortran namelist parameters by name, type, default value, unit, description, and scope:

- *Type* refers to the type of the Fortran variable, in which the value is stored: I=INTEGER, L=LOGICAL, R=REAL, C=character string
- *Default* is the preset value, if defined, that is assigned to this parameter within the programs.
- *Unit* shows the unit of the control parameter, where applicable.
- *Description* explains in a few words the purpose of the parameter.
- *Scope* explains under which conditions the namelist parameter has any effect, if its scope is restricted to specific settings of other namelist parameters.

Information on the file, where the namelist is defined and used, is given at the end of each table.

2 Namelist parameters for grid generation

2.1 Namelist parameters defining the atmosphere grid

2.1.1 graph_ini (NAMELIST_GRAPH)

| Parameter | Type | Default | Unit | Description | Scope |
|-------------|------|---------|------|--|-------|
| nroot | I | 2 | | root subdivision of initial edges | |
| grid_levels | I | 4 | | number of edge bisections following the root subdivision | |
| lplane | L | .FALSE. | | switch for generating a double periodic planar grid. The root level consists of 8 triangles. | |

Defined and used in: src/grid_generator/mo_io_graph.f90

2.1.2 grid_ini (NAMELIST_GRID)

| Parameter | Type | Default | Unit | Description | Scope |
|-------------|------|---------|------|--|-------|
| nroot | I | 2 | | root subdivision of initial edges | |
| grid_levels | I | 4 | | number of edge bisections following the root subdivision | |
| lplane | L | .FALSE. | | switch for generating planar grid. The root level consists of 8 triangles. | |

Defined and used in: src/grid_generator/mo_io_grid.f90

2.1.3 grid_options (NAMELIST_GRID)

| Parameter | Type | Default | Unit | Description | Scope |
|-------------|------|---------|------|---|-------|
| x_rot_angle | R | 0.0 | deg | Rotation of the icosahedron about the x-axis (connecting the origin and [0°E, 0°N]) | |

| Parameter | Type | Default | Unit | Description | Scope |
|----------------|------|---------|------|--|--------------------------|
| y_rot_angle | R | 0.0 | deg | Rotation of the icosahedron about the y-axis (connecting the origin and [90°E, 0°N), done after the rotation about the x-axis. | |
| z_rot_angle | R | 0.0 | deg | rotation of the icosahedron about the z-axis (connecting the origin and [0°E, 90°N), done after the rotation about the y-axis. | |
| itype_optimize | I | 4 | | Grid optimization type 0: no optimization 1: Heikes Randall 2: equal area 3: c-grid small circle 4: spring dynamics | |
| l_c_grid | L | .FALSE. | | C-grid constraint on last level | |
| maxlev_optim | I | 100 | | Maximum grid level where the optimization is applied | i_type_optimize = 1 or 4 |
| beta_spring | R | 0.90 | | tuning factor for target grid length | i_type_optimize = 4 |

Defined and used in: src/grid_generator/mo_io_grid.f90

2.1.4 plane_options (NAMELIST_GRID)

| Parameter | Type | Default | Unit | Description | Scope |
|-------------|------|---------|------|----------------------------------|---------------|
| tria_arc_km | R | 10.0 | km | length of triangle edge on plane | lplane=.TRUE. |

The number of grid points is generated by root level section and further bisections. The double periodic root level consists of 8 triangles.

The spatial coordinates are $-1 \leq x \leq 1$, and $-\sqrt{3}/2 \leq y \leq \sqrt{3}/2$. Currently the planar option can only be used as an *f*-plane.

Defined and used in: src/grid_generator/mo_io_grid.f90

2.1.5 gridref_ini (NAMELIST_GRIDREF)

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|-------------|-------|
|-----------|------|---------|------|-------------|-------|

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|-----------------|----------------|------|--|---------------|
| grid_root | I | 2 | | root subdivision of initial edges | |
| start_lev | I | 4 | | number of edge bisections following the root subdivision | |
| n_dom | I | 2 | | number of logical model domains, including the global one | |
| n_phys_dom | I | n_dom | | number of physical model domains, may be larger than n_dom (in this case, domain merging is applied) | |
| parent_id | I(n_phys_dom-1) | i | | ID of parent domain (first entry refers to first nested domain; needs to be specified only in case of more than one nested domain per grid level) | |
| logical_id | I(n_phys_dom-1) | i+1 | | logical grid ID of domain (first entry refers to first nested domain; needs to be specified only in case of domain merging, i.e. n_dom < n_phys_dom) | |
| l_plot | L | .FALSE. | | produces GMT plots showing the locations of the nested domains | |
| l_circ | L | .TRUE. | | Create circular (.T.) or rectangular (.F.) refined domains | |
| l_rotate | L | .FALSE. | | Rotates center point into the equator in case of l_circ = .FALSE. | lcirc=.FALSE. |
| write_hierarchy | I | 1 | | 0: Output only computational grids 1: Output in addition parent grid of global model domain (required for computing physics on a reduced grid) 2: Output all grids back to level 0 (required for hierarchical search algorithms) | |
| bdy_indexing_depth | I | max_rlccl (=8) | | Number of cell rows along the lateral boundary of a model domain for which the refin_ctrl fields contain the distance from the lateral boundary; needs to be enlarged when lateral boundary nudging is required for one-way nesting | |

| Parameter | Type | Default | Unit | Description | Scope |
|------------|------------|---------|------|--|---------------|
| radius | R(n_dom-1) | 30. | deg | radius of nested domain (first entry refers to first nested domain; needs to be specified for each nested domain separately) | lcirc=.TRUE. |
| hwidth_lon | R(n_dom-1) | 20. | deg | zonal half-width of refined domain (first entry refers to first nested domain; needs to be specified for each nested domain separately) | lcirc=.FALSE. |
| hwidth_lat | R(n_dom-1) | 20. | deg | meridional half-width of refined domain (first entry refers to first nested domain; needs to be specified for each nested domain separately) | lcirc=.FALSE. |
| center_lon | R(n_dom-1) | 90. | deg | center longitude of refined domain (first entry refers to first nested domain; needs to be specified for each nested domain separately) | |
| center_lat | R(n_dom-1) | 30. | deg | center latitude of refined domain (first entry refers to first nested domain; needs to be specified for each nested domain separately) | |

Defined and used in: `src/grid_generator/mo_gridrefinement.f90`

2.2 Namelist parameters defining the local grid generation

The ocean grids are created by the script `run/create_ocen_grid.run`

2.2.1 grid_geometry_conditions

| Parameter | Type | Default | Unit | Description | Scope |
|------------------|---------------------|---------|---------|--------------------------------|-------|
| no_of_conditions | I | 0 | | Number of geometric conditions | |
| patch_shape | I(no_of_conditions) | 0 | | 1=rectangle; 2=circle | |
| patch_center_x | R(no_of_conditions) | 0.0 | degrees | longitude of patch center | |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|-------------------------------|---------|---------|--|---------------|
| patch_center_y | R(no_of_ _condi- tions) | 0.0 | degrees | latitude of patch center | |
| rectangle_xradius | R(no_of_ _condi- tions) | 0.0 | degrees | half meridional extension of a rectangular patch | patch_shape=1 |
| rectangle_yradius | R(no_of_ _condi- tions) | 0.0 | degrees | half zonal extension of a rectangular patch | patch_shape=1 |
| circle_radius | R(no_of_ _condi- tions) | 0.0 | degrees | radius of a circular patch | patch_shape=2 |

Defined in `mo_grid_conditions.f90`

2.2.2 local_grid_optimization

| Parameter | Type | Default | Unit | Description | Scope |
|----------------------------------|------|---------|------|---|-------|
| use_optimization | L | .FALSE. | | Apply, or not, optimization | |
| use_edge_springs | L | .FALSE. | | Use spring dynamics | |
| prime_ref_length_coeff | R | 1.0 | | Spring length coefficient | |
| use_adaptive_ spring_length | L | .FALSE. | | Use adaptive spring length | |
| use_local_reference_ _length | L | .FALSE. | | Use locally adaptive spring length | |
| local_reference_ length_coeff | R | 0.0 | | Coefficient of local vs global spring length | |
| use_isotropy_force | L | .FALSE. | | Use isotropy force, tends to create symmetric triangles | |
| isotropy_rotation_coeff | R | 0.0 | | Coefficient of the rotational isotropy force | |
| isotropy_stretch_coeff | R | 0.0 | | Coefficient of the stretch isotropy force | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------------|------|---------|------|--|-------|
| optimize_vertex_depth | I | 1 | | For patches the min depth of the vertices that will be optimized. The boundary vertices have depth 0, the next level 1, etc. | |

Defined in `mo_local_grid_optimization.f90`

2.2.3 create_ocean_grid

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------------|------|---------|-----------------|---|--------------------|
| only_get_sea_land_mask | L | .false. | | .true.:returns the whole grid with a sea-land mask; .false.:returns only the ocean grid | |
| smooth_ocean_boundary | L | .true. | | .true.:smooths the ocean boundaries so no triabgle has two boundary edges; .false.:no smoothing | |
| input_file | C | | | name of the input grid file | |
| elevation_file | C | | | name of the file containing cell elevation values for the input_file | no_of_conditions=0 |
| elevation_field | C | | | name of the field containing the cell elevation values | no_of_conditions=0 |
| min_sea_depth | R | 0.0 | m (negative) | if cell elevation < min_sea_depth then the cell is consider sea | |
| set_sea_depth | R | 0.0 | m (negative) | if not 0, then sea cells are of set_sea_depth elevation | |
| set_min_sea_depth | R | 0.0 | m (negative) | if not 0, then sea cells have a maximum of set_min_sea_depth elevation | |
| edge_elev_interp_method | I | 2 | | compute edge elevation from cells using: linear interpolation=1; min value = 2 | |
| output_refined_ocean_file | C | | | name of the output refined ocean grid file | |

Defined in `mo_create_ocean_grid.f90`

2.2.4 torus_grid_parameters

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------------|------|---------|--------|--|-------|
| y_no_of_rows | I | | 4 | number of triangle rows of the torus grid, ≥ 2 | |
| x_no_of_columns | I | | 8 | number of triangle columns of the torus grid, ≥ 2 | |
| edge_length | R | m | 1000.0 | the triangle edge length | |
| x_center | R | m | 0.0 | the x coordinate of the torus center | |
| y_center | R | m | 0.0 | the y coordinate of the torus center | |
| out_file_name | C | | | the torus grid file name | |
| unfolded_torus_file_name | C | | | the unfolded torus grid file name (for plotting) | |
| ascii_filename | C | | | the unfolded torus grid ascii file name (for plotting) | |

Defined in `mo_create_torus_grid.f90`. See the run script `run/create_torus_grid.run`.

3 Namelist parameters defining the ICON model

Namelist parameters for the ICON models are organized in several thematic Fortran namelists controlling the experiment, and the properties of dynamics, transport, physics etc.

3.1 master_nml

| Parameter | Type | Default | Unit | Description | Scope |
|----------------|------|---------|------|--|-------|
| l_restart | L | .FALSE. | | If .TRUE.: Current experiment is started from a restart. | |
| model_base_dir | C | ' ' | | General path which may be used in file names of other name lists: If a file name contains the keyword "<path>", then this <code>model_base_dir</code> will be substituted. | |

3.2 master_model_nml (repeated for each model)

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------------------|------|--------------|------|---|-------|
| model_name | C | | | Character string for naming this component. | |
| model_namelist_filename | C | | | File name containing the model namelists. | |
| model_type | I | 0 | | Identifies which component to run. atmosphere=1, ocean=2, radiation=3, dummy_model=99 | |
| model_min_rank | I | 0 | | Start MPI rank for this model. | |
| model_max_rank | I | -1 | | End MPI rank for this model. | |
| model_inc_rank | I | 0 | | Stride of MPI ranks. | |
| model_restart_info_filename | C | restart.info | | Name (including full path) of the restart info file for this model | |

3.3 time_nml

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|------|------------------------|------|--|-------|
| dt_restart | R | 86400.*30. | s | Length of restart cycle in seconds. Note that the frequency of writing restart files is controlled by io_nml:dt_checkpoint. If the value of dt_checkpoint resulting from model default or user's specification is longer than dt_restart, it will be reset (by the model) to dt_restart so that at least one restart file is generated during the restart cycle. If dt_restart is larger than but not a multiple of dt_checkpoint, restart file will NOT be generated at the end of the restart cycle. | |
| calendar | I | 1 | | Calendar type: 0=Julian/Gregorian 1=proleptic Gregorian 2=30day/month,360day/year | |
| ini_datetime_string | C | '2008-09-01T00:00:00Z' | | Initial date and time of the simulation | |
| end_datetime_string | C | 2008-09-01T01:40:00Z' | | End date and time of the simulation | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|--|-------|
| | | | | Length of the run If "nsteps" in run_nml (see below) is positive, then nsteps*dtime is used to compute the end date and time of the run. Else the initial date and time, the end date and time, dt_restart, as well as the time step are used to compute "nsteps". | |

3.4 parallel_nml

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|------|---------|------|--|---------------------|
| nproma | I | 1 | | chunk length | |
| n_ghost_rows | I | 1 | | number of halo cell rows | |
| division_method | I | 1 | | method of domain decomposition 0: read in from file 1: use built-in geometric subdivision 2: use METIS | |
| division_file_name | C | | | Name of division file | division_method = 0 |
| ldiv_phys_dom | L | .TRUE. | | .TRUE.: split into physical domains before computing domain decomposition (in case of merged domains) (This reduces load imbalance; turning off this option is not recommended except for very small processor numbers) | division_method = 1 |
| p_test_run | L | .FALSE. | | .TRUE. means verification run for MPI parallelization (PE 0 processes full domain) | |
| l_test_openmp | L | .FALSE. | | if .TRUE. is combined with p_test_run=.TRUE. and OpenMP parallelization, the test PE gets only 1 thread in order to verify the OpenMP parallelization | p_test_run = .TRUE. |
| l_log_checks | L | .FALSE. | | if .TRUE. messages are generated during each synchronization step (use for debugging only) | |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|------|---------|------|--|---------------------|
| l_fast_sum | L | .FALSE. | | if .TRUE., use fast (not processor-configuration-invariant) global summation | |
| use_dycore_barrier | L | .FALSE. | | if .TRUE., set an MPI barrier at the beginning of the nonhydrostatic solver (do not use for production runs!) | |
| itype_exch_barrier | I | 0 | | 1: set an MPI barrier at the beginning of each MPI exchange call 2: set an MPI barrier after each MPI WAIT call 3: 1+2 (do not use for production runs!) | |
| iorder_sendrecv | I | 1 | | Sequence of send/receive calls: 1 = irecv/send 2 = isend/recv 3 = isend/irecv 4 = irecv/send with message size blocking | |
| exch_msgsize | I | 8192 | | Blocking size of exchange messages | iorder_sendrecv = 4 |
| itype_comm | I | 1 | | 1: use local memory for exchange buffers 2: use global memory for exchange buffers 3: asynchronous halo communication for dynamical core (NH tria only) | |
| num_io_procs | I | 0 | | Number of I/O processors (running exclusively for doing I/O) | |
| pio_type | I | 1 | | Type of parallel I/O. Only used if number of I/O processors greater number of domains. Experimental! | |
| nh_stepping_threads | I | 1 | | The number of OpenMP threads to be used by the non-hydrostatic dycore. Only used if the __OMP_RADIATION__ flag is set during compilation. Experimental! | |
| radiation_threads | I | 1 | | The number of OpenMP threads to be used by the radiation. Only used if the __OMP_RADIATION__ flag is set during compilation. Experimental! | |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------------|------|---------|------|--|-------|
| use_icon_comm | L | .FALSE. | | Enable the use of MPI bulk communication through the icon_comm_lib | |
| icon_comm_debug | L | .FALSE. | | Enable debug mode for the icon_comm_lib | |
| max_send_recv_buffer_size | I | 131072 | | Size of the send/receive buffers for the icon_comm_lib. | |
| use_sp_output | L | .FALSE. | | Enable this flag if output fields shall be gathered and written in single-precision. | |

Defined and used in: src/namelist/mo_parallel_nml.f90

3.5 coupling_nml

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|------|---------|------|---|-------|
| name | C | blank | | short name of the coupling field | |
| dt_coupling | I | 0 | s | coupling time step / coupling interval | |
| dt_model | I | 0 | s | model time step | |
| lag | I | 0 | | offset to coupling event in number of model time steps | |
| l_time_average | L | .FALSE. | | .TRUE.: time averaging between two coupling events | |
| l_time_accumulation | L | .FALSE. | | .TRUE.: accumulation of coupling fields in time between two coupling events | |
| l_diagnostic | L | .FALSE. | | .TRUE.: simple diagnostics (min, max, avg) for coupling fields is switched on | |
| l_activated | L | .FALSE. | | .TRUE.: activate the coupling of the respective coupling field | |

Defined and used in: src/namelist/mo_coupling_nml.f90

3.6 run_nml

| Parameter | Type | Default | Unit | Description | Scope |
|----------------------|------|---------|------|---|---|
| ldump_states | L | .FALSE. | | Dump patch/interpolation/grid refinement state of every patch (after subdivision in case of a parallel run) to a Netcdf file and exit program. | |
| lrestore_states | L | .FALSE. | | Restore patch/interpolation/grid refinement states from NetCDF dump files instead of calculating them. | |
| dump_filename | C | | | Filename of dump/restore files, default: "<path>dump_<proc><gridfile>". May contain the keyword <path> which will be substituted by <code>model_base_dir</code> , <proc> substituted by "procXofY_", and the grid filename <gridfile>. | |
| dd_filename | C | | | Filename of NetCDF domain decomposition dump files, default: "<path>dd_<gridfile>". May contain the keyword <path> which will be substituted by <code>model_base_dir</code> , and the grid filename <gridfile>. | |
| l_one_file_per_patch | L | .FALSE. | | Use one file per patch for all processors. This will decrease the amount of files used for dump/restore considerably, especially for massively parallel runs on hundreds or thousands of processors. Time for dumping will increase since the file has to be written sequentially, the time for restore should stay roughly the same, however. | ldump_states=.TRUE. or lrestore_states=.TRUE. |
| ldump_dd | L | .FALSE. | | Dump the domain decomposition (and a few related fields). This can be done either in a parallel run or in a single-CPU run. When done in a parallel run, the domain decomposition is for the number of parallel processes in use. When done in a single-CPU run, <code>nproc_dd</code> (see below) determines the number of processes for the decomposition. Uses always only one file per patch, | |

| Parameter | Type | Default | Unit | Description | Scope |
|----------------------|------------|---------|------|---|--|
| lread_dd | L | .FALSE. | | Read the domain decomposition when dumped with ldump_dd. | |
| nproc_dd | I | 1 | | Number of processors for the target domain decomposition (only relevant when running on a single processor). | ldump_dd = TRUE and a single processor run |
| nsteps | I | 0 | | number of time steps of this run. | |
| dtime | R | 600.0 | s | time step | |
| ltestcase | L | .TRUE. | | Idealized testcase runs | |
| ldynamics | L | .TRUE. | | Compute adiabatic dynamic tendencies | |
| iforcing | I | 0 | | Forcing of dynamics and transport by parameterized processes. Use positive indices for the atmosphere and negative indices for the ocean. 0: no forcing 1: Held-Suarez forcing 2: ECHAM forcing 3: NWP forcing 4: local diabatic forcing without physics 5: local diabatic forcing with physics -1: MPIOM forcing (to be done) | |
| ltransport | L | .FALSE. | | Compute large-scale tracer transport | |
| ntracer | I | 0 | | Number of advected tracers handled by the large-scale transport scheme | |
| lvert_nest | L | .FALSE. | | If set to .true. vertical nesting is switched on (i.e. variable number of vertical levels) | |
| num_lev | I(max_dom) | 31 | | Number of full levels (atm.) for each domain | lvert_nest=.TRUE. |
| nshift | I(max_dom) | 0 | | vertical half level of parent domain which coincides with upper boundary of the current domain | lvert_nest=.TRUE. |
| ltimer | L | .TRUE. | | TRUE: Timer for monitoring the runtime of specific routines is on (FALSE = off) | |
| timers_level | I | 1 | | | |
| activate_sync_timers | L | F | | TRUE: Timer for monitoring runtime of communication routines (FALSE = off) | |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------|------|------------------|------|--|----------------|
| msg_level | I | 10 | | controls how much printout is written during runtime. For values less than 5, only the time step is written. | |
| msg_timestamp | L | .FALSE. | | If .TRUE., precede output messages by time stamp. | |
| test_mode | I | 0 | | Setting a value larger than 0 activates a dummy mode in which time stepping is changed into just doing iterations, and MPI communication is replaced by copying some value from the send buffer into the receive buffer (does not work with nesting and reduced radiation grid because the send buffer may then be empty on some PEs) | iequations = 3 |
| output | C(:) | "vlist","totint" | | <p>Main switch for enabling/disabling components of the model output. One or more choices can be set (as an array of string constants). Possible choices are:</p> <ul style="list-style-type: none"> • "none": switch off all output; • "vlist" : old, vlist-based output mode; • "nml": new output mode (cf. <code>output_nml</code>); • "totint": computation of total integrals. <p>If the <code>output</code> namelist parameter is not set explicitly, the default setting "vlist","totint" is assumed.</p> | |

Defined and used in: src/namelists/mo_run_nml.f90

3.7 grid_nml

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|-------------|-------|
|-----------|------|---------|------|-------------|-------|

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|----------|---------|------|---|---------------|
| cell_type | I | 3 | | Cell type 3: triangular cells 4: quadrilateral cells (to be done) 6: pentagonal/hexagonal cells | |
| lplane | L | .FALSE. | | planar option | |
| corio_lat | R | 0.0 | deg | Center of the f-plane is located at this geographical latitude | lplane=.TRUE. |
| l_limited_area | L | .FALSE. | | | |
| grid_rescale_factor | R | 1.0 | | The geometry and the timestep will be multiplied by this factor. The angular velocity will be divided by this factor. | |
| lfeedback | L(n_dom) | .TRUE. | | Specifies if feedback to parent grid is performed. Setting lfeedback(1)=.false. turns off feedback for all nested domains; to turn off feedback for selected nested domains, set lfeedback(1)=.true. and set ".false." for the desired model domains | n_dom>1 |
| ifedback_type | I | 1 | | 1: incremental feedback 2: relaxation-based feedback | n_dom>1 |
| start_time | R(n_dom) | 0. | s | Time when a nested domain starts to be active (namelist entry is ignored for the global domain) | n_dom>1 |
| end_time | R(n_dom) | 1.E30 | s | Time when a nested domain terminates (namelist entry is ignored for the global domain) | n_dom>1 |
| patch_weight | R(n_dom) | 0. | | If patch_weight is set to a value > 0 for any of the first level child patches, processor splitting will be performed, i.e. every of the first level child patches gets a subset of the total number of processors corresponding to its patch_weight. A value of 0. corresponds to exactly 1 processor for this patch, regardless of the total number of processors. For the root patch and higher level childs, patch_weight is not used. However, patch_weight must be set to 0 for these patches to avoid confusion. | n_dom>1 |

| Parameter | Type | Default | Unit | Description | Scope |
|------------------------------|------|---------|------|---|-------|
| lredgrid_phys | L | .FALSE. | | If set to .true. is calculated on a reduced grid (= one grid level higher) | |
| dynamics_grid_filename | C | | | Array of the grid filenames to be used by the dycore. May contain the keyword <path> which will be substituted by <code>model_base_dir</code> . | |
| dynamics_parent_grid_id | I | | | Array of the indexes of the parent grid filenames, as described by the <code>dynamics_grid_filename</code> array. Indexes start at 1, an index of 0 indicates no parent. | |
| radiation_grid_filename | C | | | Array of the grid filenames to be used for the radiation model. Filled only if the radiation grid is different from the dycore grid. May contain the keyword <path> which will be substituted by <code>model_base_dir</code> . | |
| dynamics_radiation_grid_link | I | | | Array of the indexes linking the dycore grids, as described by the <code>dynamics_grid_filename</code> array, and the <code>radiation_grid_filename</code> array. It provides the link index of the <code>radiation_grid_filename</code> , for each entry of the <code>dynamics_grid_filename</code> array. Indexes start at 1, an index of 0 indicates that the radiation grid is the same as the dycore grid. Only needs to be filled when the <code>radiation_grid_filename</code> is defined. | |

Defined and used in: `src/namelists/mo_grid_nml.f90`

3.8 gridref_nml

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|------|---------|------|---|---------------------------|
| grf_intmethod_c | I | 2 | | Interpolation method for grid refinement (cell-based dynamical variables): 1: parent-to-child copying 2: gradient-based interpolation | <code>n_dom > 1</code> |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|------|---------|------|---|---------|
| grf_intmethod_ct | I | 2 | | Interpolation method for grid refinement (cell-based tracer variables): 1: parent-to-child copying 2: gradient-based interpolation | n_dom>1 |
| grf_intmethod_e | I | 4 | | Interpolation method for grid refinement (edge-based variables): 1: inverse-distance weighting (IDW) 2: RBF interpolation 3: combination gradient-based / IDW 4: combination gradient-based / RBF 5/6: same as 3/4, respectively, but direct interpolation of mass fluxes along nest interface edges | n_dom>1 |
| grf_velfbk | I | 1 | | Method of velocity feedback: 1: average of child edges 1 and 2 2: 2nd-order method using RBF interpolation | n_dom>1 |
| grf_scalfbk | I | 2 | | Feedback method for dynamical scalar variables (T, p_{sf}): 1: area-weighted averaging 2: bilinear interpolation | n_dom>1 |
| grf_tracfbk | I | 2 | | Feedback method for tracer variables: 1: area-weighted averaging 2: bilinear interpolation | n_dom>1 |
| grf_idw_exp_e12 | R | 1.2 | | exponent of generalized IDW function for child edges 1/2 | n_dom>1 |
| grf_idw_exp_e34 | R | 1.7 | | exponent of generalized IDW function for child edges 3/4 | n_dom>1 |
| rbf_vec_kern_grf_e | I | 1 | | RBF kernel for grid refinement (edges): 1: Gaussian 2: $1/(1 + r^2)$ 3: inverse multiquadric | n_dom>1 |
| rbf_scale_grf_e | R | 0.5 | | RBF scale factor for grid refinement (edges) | n_dom>1 |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|------|---------|------|--|-------------------------------------|
| denom_diffu_t | R | 135 | | Denominator for lateral boundary diffusion of temperature | n_dom>1 |
| denom_diffu_v | R | 200 | | Denominator for lateral boundary diffusion of velocity | n_dom>1 |
| l_mass_consvcorr | L | .TRUE. | | .TRUE.: Apply mass conservation correction in feedback routine | n_dom>1 |
| l_density_nudging | L | .TRUE. | | .TRUE.: Apply density nudging near lateral nest boundary | n_dom>1 .AND. lfeedback = .TRUE. |

Defined and used in: src/namelist/mo_gridref_nml.f90

3.9 prepicon_nml

The namelist prepicon_nml contains switches controlling the real-data initialization functionality of ICON. There are currently two ways of using it:

- Using the preprocessing tool prep_icon, it is possible to
 - ((a)) **generate the three-dimensional coordinate fields** needed by IFS2ICON if IFS2ICON is requested to do the horizontal and vertical interpolation from the IFS grid to the ICON grid
 - ((b)) **convert the hydrostatic set of variables provided by IFS2ICON** to the nonhydrostatic set of equations needed by ICONAM,
 - ((c)) **perform the vertical interpolation** to the ICON grid if IFS2ICON is requested to do only the horizontal interpolation step,
 - ((d)) **perform data remapping from a regular grid onto a triangular ICON grid and vice versa.**
- If ICONAM (iequations=3) is combined with NWP physics (iforcing=3), setting ltestcase=.false. activates functionality (c) while running the ICON executable.

Remark: When using prep_icon as a stand-alone tool, namelists must be defined in a file “NAMELIST_PREPICON”.

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|-------------|-------|
|-----------|------|---------|------|-------------|-------|

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|------------|---------|------|--|----------------|
| i_oper_mode | I | 1 | | Operating mode if the prep_icon executable is run: 1: generate coordinate fields 2: convert IFS2ICON output to NH prognostic variables 3: do vertical interpolation 4: grid remapping | |
| nlev_in | I | 91 | | number of model levels of input data | |
| nlevsoil_in | I | 4 | | number of soil levels of input data | |
| zpbl1 | R | 500.0 | m | bottom height (AGL) of layer used for gradient computation | |
| zpbl2 | R | 1000.0 | m | top height (AGL) of layer used for gradient computation | |
| l_w_in | L | .FALSE. | | Logical switch if vertical wind is provided as input | |
| l_sfc_in | L | .TRUE. | | Logical switch if surface fields are provided as input (mandatory when inwp_surface >0) | |
| l_sst_in | L | .FALSE. | | Logical switch. If true, the surface temperature of the water sea points is initialized with the SST provided in the ifs2icon file. If false, it is initialized with the skin temperature. If the SST is not provided in the ifs2icon file, l_sst_in is reset to false. | |
| l_zp_out | L | .FALSE. | | Logical switch for diagnostic output on pressure and height levels | prep_icon only |
| l_extdata_out | L | .FALSE. | | Logical switch to write extdata fields into output | prep_icon only |
| l_coarse2fine_mode | L(max_dom) | .FALSE. | | If true, apply corrections for coarse-to-fine mesh interpolation to wind and temperature | |
| ifs2icon_filename | C | | | Filename of IFS2ICON input file, default " <code><path>ifs2icon_R<nroot>B<jlev>_DOM<idom>.nc</code> ". May contain the keywords <code><path></code> which will be substituted by <code>model_base_dir</code> , as well as <code>nroot</code> , <code>jlev</code> , and <code>idom</code> defining the current patch. | |

Defined and used in: src/namelist/mo_prepicon_nml.f90

3.10 remap_nml

The namelist remap_nml is only processed when setting i_oper_mode=4 in prepicon_nml (cf. option (d)).

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|------|---------|------|---|---------------------------------|
| in_grid_filename | C | "" | | input grid file name | i_oper_mode=4 (prepicon_nml) |
| in_filename | C | "" | | input data file name (may be identical to input grid file name) | i_oper_mode=4 (prepicon_nml) |
| in_type | I | 1 | | Type of input grid: 1: regular grid; 2: ICON triangular grid | i_oper_mode=4 (prepicon_nml) |
| out_grid_filename | C | "" | | file containing output grid that is read as input for the remapping process | i_oper_mode=4 (prepicon_nml) |
| out_filename | C | "" | | output data file name | i_oper_mode=4 (prepicon_nml) |
| out_type | I | 2 | | Type of output grid: 1: regular grid; 2: ICON triangular grid | i_oper_mode=4 (prepicon_nml) |
| s_maxsize | I | 500 000 | | Maximum size of sequential list for very large stencils | i_oper_mode=4 (prepicon_nml) |
| l_have3dbuffer | L | .FALSE. | | Flag. True if a 3D buffer is used for output | i_oper_mode=4 (prepicon_nml) |

Defined and used in: src/prep_icon/mo_remap_io.f90

3.11 input_field_nml

The namelist input_field_nml is only processed when setting i_oper_mode=4 in prepicon_nml (cf. option (d)).

| Parameter | Type | Default | Unit | Description | Scope |
|------------|------|---------|------|-------------------------|---------------------------------|
| inputname | C | "" | | Name of input variable | i_oper_mode=4 (prepicon_nml) |
| outputname | C | "" | | Name of output variable | i_oper_mode=4 (prepicon_nml) |
| code | I | 0 | | GRIB input code number | i_oper_mode=4 (prepicon_nml) |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------|------|---------|------|-----------------------|---------------------------------|
| type_of_layer | C | "" | | GRIB input layer type | i_oper_mode=4 (prepicon_nml) |

Defined and used in: src/prep_icon/mo_remap_input.f90

3.12 interpol_nml

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|----------|--------------------------|------|--|----------------|
| llsq_lin_consv | L | .FALSE. | | conservative (T) or non-conservative (F) least-squares reconstruction for 2nd order (linear) transport | |
| llsq_high_consv | L | .TRUE. | | conservative (T) or non-conservative (F) least-squares reconstruction for high order transport | |
| lsq_high_ord | I | 3 | | polynomial order for high order reconstruction 1: linear 2: quadratic 30: cubic (no 3 rd order cross deriv.) 3: cubic | ihadv_tracer=4 |
| rbf_vec_kern_c | I | 1 | | Kernel type for reconstruction at cell centres: 1: Gaussian 3: inverse multiquadric | |
| rbf_vec_kern_e | I | 3 | | Kernel type for reconstruction at edges: 1: Gaussian 3: inverse multiquadric | |
| rbf_vec_kern_v | I | 1 | | Kernel type for reconstruction at vertices: 1: Gaussian 3: inverse multiquadric | |
| rbf_vec_scale_c | R(n_dom) | resolution- dependent | | Scale factor for RBF reconstruction at cell centres | |
| rbf_vec_scale_e | R(n_dom) | resolution- dependent | | Scale factor for RBF reconstruction at edges | |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|----------|----------------------|------|---|--------------------------------|
| rbf_vec_scale_v | R(n_dom) | resolution-dependent | | Scale factor for RBF reconstruction at vertices | |
| nudge_max_coeff | R | 0.02 | | Maximum relaxation coefficient for lateral boundary nudging | |
| nudge_efold_width | R | 2.5 | | e-folding width (in units of cell rows) for lateral boundary nudging coefficient | |
| nudge_zone_width | I | 8 | | Total width (in units of cell rows) for lateral boundary nudging zone | |
| i_cori_method | I | 3 | | Selector for tangential wind reconstruction method 1: Almut's method for tangential wind, but PV usage as in TRSK 2: method of Thuburn, Ringler, Skamarock and Klemp (TRSK) 3: Almut's method for tangential wind and PV usage | currently only for cell_type=6 |
| l_corner_vort | L | .TRUE. | | switch whether the rhombus averaged corner vorticity is averaged to the hexagon (.TRUE.) or the rhombi are directly averaged to the hexagon (.FALSE.) | i_cori_method=3 |
| l_intp_c2l | L | .FALSE. | | If .TRUE. directly interpolate from cell centers to lon-lat points, otherwise do gradient interpolation and reconstruction. | |
| rbf_dim_c2l | I | 10 | | stencil size for direct lon-lat interpolation: 4 = nearest neighbor, 13 = vertex stencil, 10 = edge stencil. | |
| l_mono_c2l | L | .FALSE. | | Monotonicity can be enforced by demanding that the interpolated value is not higher or lower than the stencil point values. | |

Defined and used in: src/namelist/mo_interp_nml.f90

3.13 dynamics_nml

This namelist is relevant if run_nml:ldynamics=.TRUE.

| Parameter | Type | Default | Unit | Description | Scope |
|----------------|------|------------|------|--|----------------------|
| iequations | I | 1 | | Equations and prognostic variables. Use positive indices for the atmosphere and negative indices for the ocean. 0: shallow water model 1: hydrostatic atmosphere, T 2: hydrostatic atm., θ -dp 3: non-hydrostatic atmosphere -1: hydrostatic ocean | |
| idiv_method | I | 1 | | Method for divergence computation: 1: Standard Gaussian integral. Hydrostatic atm. model: for unaveraged normal components, Non-hydrostatic atm. model: for averaged normal components 2: bilinear averaging of divergence | grid_nml:cell_type=3 |
| divavg_cntrwgt | R | 0.5 | | Weight of central cell for divergence averaging | idiv_method= 2 |
| sw_ref_height | R | 0.9*2.94e4 | gm | Reference height of shallow water model used for linearization in the semi-implicit time stepping scheme | |
| lcoriolis | L | .TRUE. | | Coriolis force | |

Defined and used in: src/namelist/mo_dynamics_nml.f90

3.14 ha_dyn_nml

This namelist is relevant if run_nml:ldynamics=.TRUE. and dynamics_nml:iequations=IHS_ATM_TEMP or IHS_ATM_THETA.

| Parameter | Type | Default | Unit | Description | Scope |
|--------------|------|---------|------|--|-------|
| itime_scheme | I | 4 | | Time integration scheme: 11: pure advection (no dynamics) 12: 2 time level semi implicit (not yet implemented) | |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|------|---------|------|---|--|
| | | | | 13: 3 time level explicit 14: 3 time level with semi implicit correction 15: standard 4th-order Runge-Kutta method (4-stage) 16: SSPRK(5,4) scheme (5-stage) | |
| ileapfrog_startup | I | 1 | | How to integrate the first time step when the leapfrog scheme is chosen. 1 = Euler forward; 2 = a series of sub-steps. | itime_scheme= 13 or 14 |
| asselin_coeff | R | 0.1 | | Asselin filter coefficient | itime_scheme= 13 or 14 |
| si_2tls | R | 0.6 | | weight of time step n+1. Valid range: [0,1] | itime_scheme=12 |
| si_expl_scheme | I | 2 | | scheme for the explicit part used in the 2 time level semi-implicit time stepping scheme. 1 = Euler forward; 2 = Adams-Bashforth 2nd order | itime_scheme=12 |
| si_cmin | R | 30.0 | m/s | semi implicit correction is done for eigenmodes with speeds larger than si_cmin | itime_scheme=14 and lsi_3d=.FALSE. |
| si_coff | R | 1.0 | | weight of the semi implicit correction | itime_scheme=14 |
| si_offctr | R | 0.7 | | | itime_scheme=14 |
| si_rtol | R | 1.0e-3 | | relative tolerance for GMRES solver | itime_scheme=14 |
| lsi_3d | L | .FALSE. | | 3D GMRES solver or decomposition into 2D problems | lshallow_water=.FALSE. and itime_scheme=14 |
| ldry_dycore | L | .TRUE. | | Assume dry atmosphere | iequations \in {1,2} |
| lref_temp | L | .FALSE. | | Set a background temperature profile as base state when computing the pressure gradient force | iequations \in {1,2} |

3.15 nonhydrostatic_nml (relevant if run_nml:iequations=3)

| Parameter | Type | Default | Unit | Description | Scope |
|--------------|------|---------|------|---|-------|
| itime_scheme | I | 4 | | Options for predictor-corrector time-stepping scheme: | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|----------|---------|------|---|--|
| | | | | 4: Contravariant vertical velocity is computed in the predictor step only, velocity tendencies are computed in the corrector step only (most efficient option) 5: Contravariant vertical velocity is computed in both substeps (beneficial for numerical stability in very-high resolution setups with extremely steep slopes, otherwise no significant impact) 6: As 5, but velocity tendencies are also computed in both substeps (no apparent benefit, but more expensive) | iequations=3 and cell_type=3 |
| rayleigh_type | I | 2 | | Type of Rayleigh damping 1: CLASSICAL (requires velocity reference state!) 2: Klemp (2008) type | cell_type=3 |
| rayleigh_coeff | R(n_dom) | 0.05 | | Rayleigh damping coefficient $1/\tau_0$ (Klemp, Dudhia, Hassiotis: MWR136, pp.3987-4004) | cell_type=3 |
| damp_height | R(n_dom) | 45000 | m | Height at which Rayleigh damping of vertical wind starts | |
| htop_moist_proc | R | 22500.0 | m | Height above which moist physics and advection of cloud and precipitation variables are turned off | |
| hbot_qvsubstep | R | 24000.0 | m | Height above which QV is advected with substepping scheme (must be larger than htop_moist_proc) | cell_type=3 and ihadv_tracer=22 or 32 |
| k2_updamp_coeff | R | 2.0e6 | | enhanced 2nd order diffusion coefficient in upper damping layer | cell_type=6, hdiff_order=3 (Smagorinski) |
| vwind_offctr | R | 0.15 | | Off-centering in vertical wind solver | cell_type=3 |
| ivctype | I | 2 | | Type of vertical coordinate: 1: Gal-Chen hybrid 2: SLEVE (uses sleve_nml) | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|------|---------|------|---|--------------------|
| iadv_rcf | I | 4 | | reduced calling frequency (rcf) for transport 1: no rcf (every dynamics-step) 2: transport every 2. step 4: ... Setting odd values (besides 1) requires l_nest_rcf = .TRUE. | |
| lhdiffrcf | L | .TRUE. | | .TRUE.: Compute diffusion only at advection time steps (in this case, divergence damping is applied in the dynamical core) | cell_type=3 |
| divdamp_fac | R | 0.004 | | Scaling factor for divergence damping | lhdiffrcf = .TRUE. |
| divdamp_order | I | 4 | | Order of divergence damping (2 or 4) | lhdiffrcf = .TRUE. |
| l_nest_rcf | L | .TRUE. | | Synchronize interpolation/feedback calls with advection (transport) time steps. l_nest_rcf is automatically reset to .FALSE. if iadv_rcf=1 | cell_type=3 |
| l_masscorr_nest | L | .FALSE. | | .TRUE.: Apply mass conservation correction also in nested domain | cell_type=3 |
| iadv_rhotheta | I | 2 | | Advection method for rho and rhotheta: 1: centred differences horiz. + vert. 2: 2nd order Miura horizontal 3: 3rd order Miura horizontal (not recommended) | cell_type=3 |
| igradp_method | I | 3 | | Discretization of horizontal pressure gradient: 1: conventional discretization with metric correction term 2: Taylor-expansion-based reconstruction of pressure (advantageous at very high resolution) 3: Similar discretization as option 2, but uses hydrostatic approximation for downward extrapolation over steep slopes 4: Cubic/quadratic polynomial interpolation for pressure reconstruction 5: Same as 4, but hydrostatic approximation for downward extrapolation over steep slopes | cell_type=3 |

| Parameter | Type | Default | Unit | Description | Scope |
|----------------|------|---------|------|--|---|
| l_zdiffu_t | L | .TRUE. | | .TRUE.: Compute Smagorinsky temperature diffusion truly horizontally over steep slopes | cell_type=3 .AND. hdiff_order=3/5 .AND. lhdiff_temp = .true. |
| thslp_zdiffu | R | 0.025 | | Slope threshold above which truly horizontal temperature diffusion is activated | cell_type=3 .AND. hdiff_order=3/5 .AND. lhdiff_temp=.true. .AND. l_zdiffu_t=.true. |
| thhgt_d_zdiffu | R | 200 | m | Threshold of height difference between neighboring grid points above which truly horizontal temperature diffusion is activated (alternative criterion to thslp_zdiffu) | cell_type=3 .AND. hdiff_order=3/5 .AND. lhdiff_temp=.true. .AND. l_zdiffu_t=.true. |
| exner_expol | R | 0.5 | | Temporal extrapolation (fraction of dt) of Exner function for computation of horizontal pressure gradient | cell_type=3 |
| l_open_abc | L | .FALSE. | | .TRUE.: Use open upper boundary condition (rather than w=0) to better conserve sea-level pressure in the presence of diabatic heating | cell_type=3 |
| ltheta_up_hori | L | .FALSE. | | upstream biased horizontal advection for theta (see also upstr_beta) | cell_type=6 |
| upstr_beta | R | 1.0 | | Selection of order for horiz. theta advection: 3rd order=1.0, 4th order=0.0 | cell_type=6 |
| gmres_rtol_nh | R | 1.0e-6 | | relative tolerance for convergence in gmres solver | cell_type=6 |

Defined and used in: src/namelist/mo_nonhydrostatic_nml.f90

3.16 sleve_nml (relevant if nonhydrostatic_nml:ivctype=2)

| Parameter | Type | Default | Unit | Description | Scope |
|---------------|------|---------|------|---|-------|
| min_lay_thckn | R | 50 | m | Layer thickness of lowermost layer; specifying zero or a negative value leads to constant layer thicknesses determined by top_height and nlev | |
| top_height | R | 23500.0 | m | Height of model top | |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------|------|---------|------|---|-------|
| stretch_fac | R | 1.0 | | Stretching factor to vary distribution of model levels; values <1 increase the layer thickness near the model top | |
| decay_scale_1 | R | 4000 | m | Decay scale of large-scale topography component | |
| decay_scale_2 | R | 2500 | m | Decay scale of small-scale topography component | |
| decay_exp | R | 1.2 | | Exponent of decay function | |
| flat_height | R | 16000 | m | Height above which the coordinate surfaces are flat | |
| lread_smt | L | .FALSE. | | read smoothed topography from file (TRUE) or compute internally (FALSE) | |

Defined and used in: src/namelist/mo_sleve_nml.f90

3.17 diffusion_nml

| Parameter | Type | Default | Unit | Description | Scope |
|-------------|------|---------------------|------|--|-------|
| lhdiff_temp | L | .TRUE. | | Diffusion on the temperature field | |
| lhdiff_vn | L | .TRUE. | | Diffusion on the horizontal wind field | |
| hdiff_order | I | 4 (hydro) 5 (NH) | | Order of ∇ operator for diffusion: -1: no diffusion 2: ∇^2 diffusion (not available for NH model on triangles!) 3: Smagorinsky ∇^2 diffusion (includes frictional heating for the hexagonal model if lhdiff_temp=.TRUE.) 4: ∇^4 diffusion 5: Smagorinsky ∇^2 diffusion combined with ∇^4 background diffusion as specified via hdiff_efdt_ratio defaults: 2 for hexagonal model, 4 for triangular model; for triangular NH model, 5 is strongly recommended! | |

| Parameter | Type | Default | Unit | Description | Scope |
|----------------------|------|--------------------------------|------|--|---|
| | | | | 24 or 42: ∇^2 diffusion from model top to a certain level (cf. k2_pres_max and k2_klev_max below); ∇^4 for the lower levels. | 24 and 42 currently allowed only in the hydrostatic atm model (run_nml:iequation = 1 or 2). |
| itype_vn_diffu | I | 1 | | Reconstruction method used for Smagorinsky diffusion: 1: u/v reconstruction at vertices only 2: u/v reconstruction at cells and vertices | iequations=3, hdiff_order=3 or 5 |
| itype_t_diffu | I | 1 | | Discretization of temperature diffusion: 1: $K_h \nabla^2 T$ 2: $\nabla \cdot (K_h \nabla T)$ | iequations=3, hdiff_order=3 or 5 |
| k2_pres_max | R | -99. | Pa | Pressure level above which ∇^2 diffusion is applied. | hdiff_order = 24 or 42, and run_nml:iequation = 1 or 2. |
| k2_klev_max | I | 0 | | Index of the vertical level till which (from the model top) ∇^2 diffusion is applied. If a positive value is specified for k2_pres_max, k2_klev_max is reset accordingly during the initialization of a model run. | hdiff_order = 24 or 42, and run_nml:iequation = 1 or 2. |
| hdiff_efdt_ratio | R | 1.0 | | ratio of e-folding time to time step (or 2* time step when using a 3 time level time stepping scheme) (only for triangles currently; for triangular NH model, values between 10 and 20 are recommended when using hdiff_order=5) | |
| hdiff_min_efdt_ratio | R | 1.0 (hydro) 15.0 (NH) | | minimum value of hdiff_efdt_ratio near model top | iequations=3 .AND. cell_type=3 .AND. hdiff_order=4 |
| hdiff_tv_ratio | R | 1.0 | | Ratio of diffusion coefficients for temperature and normal wind: $T : v_n$ | |
| hdiff_multfac | R | 1.0 | | Multiplication factor of normalized diffusion coefficient for nested domains | n_dom>1 |

| Parameter | Type | Default | Unit | Description | Scope |
|----------------|------|----------------------------------|------|--|--|
| hdiff_smag_fac | R | 0.15 (hydro) 0.025 (NH) | | Scaling factor for Smagorinsky diffusion | for triangles only with iequations=3, for hexagons with hdiff_order=3 |

Defined and used in: src/namelists/mo_diffusion_nml.f90

3.18 io_nml

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------------|------|--------------|------|--|-------|
| out_expname | C | 'IIIEEETTTT' | | Outfile basename | |
| out_filetype | I | 2 | | Type of output format: 1: GRIB1 (not yet implemented) 2: netCDF | |
| lkeep_in_sync | L | .FALSE. | | Sync output stream with file on disk after each timestep | |
| dt_data | R | 21600.0 | s | Output time interval | |
| dt_diag | R | 86400. | | diagnostic integral output interval | |
| dt_file | R | 2592000 | s | Time interval of triggering new output file | |
| dt_checkpoint | R | 2592000 | s | Time interval for writing restart files. Note that if the value of dt_checkpoint resulting from model default or user's specification is longer than time_nml:dt_restart, it will be reset (by the model) to dt_restart so that at least one restart file is generated during the restart cycle. | |
| lwrite_vorticity | L | .TRUE. | | write out averaged vorticity at vertices | |
| lwrite_initial | L | .TRUE. | | write out initial state | |
| lwrite_dblprec | L | .FALSE. | | write out double precision | |
| lwrite_oce_timestepping | L | .FALSE. | | write out intermediate ocean vars | |
| lwrite_divergence | L | .TRUE. | | write out divergence at cells | |

| Parameter | Type | Default | Unit | Description | Scope |
|------------------|------------|------------------------------|------|--|---|
| lwrite_omega | L | .TRUE. | | write out vertical velocity in pressure coords. | Always .FALSE. for nonhydrostatic and shallow water models |
| lwrite_pres | L | .TRUE. | | write out full level pressure | lshallow_water=.FALSE. |
| lwrite_z3 | L | .TRUE. | | write out geopotential on full levels | lshallow_water=.FALSE. |
| lwrite_tracer | L(ntracer) | .TRUE. | | write out tracer at cells | |
| lwrite_tend_phy | L | .TRUE. .FALSE. (Scope) | | Physics induced tendencies. | .TRUE. if iforcing=iecham .FALSE. else |
| lwrite_radiation | L | .FALSE. | | Radiation related fields. | Always .FALSE. if iforcing=inoforcing, iheldsuarez, ildf_dry |
| lwrite_precip | L | .FALSE. | | Precipitation | Always .FALSE. if iforcing=inoforcing, iheldsuarez, ildf_dry |
| lwrite_cloud | L | .FALSE. | | Cloud variables | Always .FALSE. if iforcing=inoforcing, iheldsuarez, ildf_dry |
| lwrite_tke | L | .TRUE. | | TKE | .FALSE. Always .FALSE. if iforcing=inoforcing, iheldsuarez, ildf_dry |
| lwrite_surface | L | .FALSE. | | surface variables | Always .FALSE. if iforcing=inoforcing, iheldsuarez, ildf_dry |
| lwrite_extra | L | .FALSE. | | debug fields | .TRUE. if inextra_2d / _3d > 0 .FALSE. else |
| inextra_2d | I | 0 | | Number of 2D Fields for diagnostic/debugging output. | iequations = 3 (to be done for 1, 2) |
| inextra_3d | I | 0 | | Number of 3D Fields for diagnostic/debugging output. | iequations = 3 (to be done for 1, 2) |

| Parameter | Type | Default | Unit | Description | Scope |
|----------------|------|---------|------|--|----------------------------|
| lflux_avg | L | .FALSE. | | if .FALSE. the output fluxes are accumulated from the beginning of the run if .TRUE. the output fluxes are average values from the beginning of the run, except of TOT_PREC that would be accumulated | iequations=3 iforcing=3 |
| itype_pres_msl | I | 1. | | Specifies method for computation of mean sea level pressure. 1: GME-type extrapolation, 2: stepwise analytical integration | |

Defined and used in: src/namelist/mo_io_nml.f90

3.19 output_nml

Please note: There may be several instances of output_nml in the namelist file, every one defining a list of variables with separate attributes for output.

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|--|-------|
| filetype | I | 4 | | One of CDI's FILETYPE_XXX constants. Possible values: 2 (=FILETYPE_GRB2), 4 (=FILETYPE_NC2), 5 (=FILETYPE_NC4) | |
| namespace | C | 'ECMWF' | | 'DWD' - DWD short names (or 'MPIMET', 'CMIP', 'ECMWF') Currently used for setting GRIB2 centre/subcentre information. RJ: For what exactly should that be used? | |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------|------|---------|------|--|--------|
| map_file | C | ' ' | | <p>File containig the mapping from internal names to names written to NetCDF. May contain the keyword <code><path></code> which will be substituted by <code>model_base_dir</code>.</p> <p>The format of this file:</p> <p>One mapping per line, first the internal name, then the name written to NetCDF, separated by an arbitrary number of blanks. The line may also start and end with an arbitrary number of blanks. Empty lines or lines starting with <code>#</code> are treated as comments.</p> <p>Names not covered by the mapping are output as they are.</p> <p>Note that the specification of output variables, e. g. in <code>ml_varlist</code>, is independent from this renaming.</p> | |
| mode | I | 2 | | <p>1 = forecast mode, 2 = climate mode</p> <p>In climate mode the time axis of the output file is set to <code>TAXIS_ABSOLUTE</code>. In forecast mode it is set to <code>TAXIS_RELATIVE</code>. Till now the forecast mode only works if the output is at multiples of 1 hour</p> | |
| taxis_tunit | I | 3 | | <p>3 = <code>TUNIT_HOUR</code> , 2 = <code>TUNIT_MINUTE</code></p> <p>Time unit of the <code>TAXIS_RELATIVE</code> time axis. For a complete list of possible values see <code>cdi.inc</code> Till now it only works for <code>taxis_tunit=3</code></p> | mode=1 |
| dom(:) | I | -1 | | <p>Array of domains for which this name-list is used. If not specified (or specified as -1 as the first array member), this name-list will be used for all domains.</p> <p>Attention: Depending on the setting of the parameter <code>l_output_phys_patch</code> these are either logical or physical domain numbers!</p> | |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|------|---------|------|--|-------|
| output_time_unit | I | 1 | | 1 = second, 2=minute, 3=hour, 4=day, 5=month, 6=year | |
| output_bounds(3,:) | R | None | | post-processing times in units defined by output_time_unit: start, end, increment. There may be specified several triples (up to 100) which must be in increasing order. | |
| steps_per_file | I | 100 | | Max number of output steps in one output file. If this number is reached, a new output file will be opened. | |
| include_last | L | .TRUE. | | Flag whether to include the last time step | |
| output_grid | L | .FALSE. | | Flag whether grid information is output (in NetCDF output) | |
| output_filename | C | None | | Output filename prefix (which may include path). Domain number, level type, file number and extension will be added, so specifying 'XXX' for output_filename you will end up in a name like XXX_DOM01_ML_0001.nc | |
| lwrite_ready | L | .FALSE. | | Flag if a "ready file" (sentinel file) should be written at the end of each output stage. Not yet implemented. | |
| ready_directory | C | None | | Output directory for ready files. Not yet implemented. | |
| ml_varlist(:) | C | None | | Name of model level fields to be output – or 'all'. | |
| pl_varlist(:) | C | None | | Name of pressure level fields to be output – or 'all'. | |
| hl_varlist(:) | C | None | | Name of height level fields to be output – or 'all'. | |
| il_varlist(:) | C | None | | Name of isentropic level fields to be output – or 'all'. | |
| p_levels(:) | R | None | hPa | pressure levels Not yet implemented. The pressure levels are currently always taken from array plevels in namelist nh_pzlev_nml. | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|------|---------|------|---|-------|
| h_levels(:) | R | None | m | height levels Not yet implemented. The height levels are currently always taken from array zlevels in namelist nh_pzlev_nml. | |
| i_levels(:) | R | None | K | isentropic levels Not yet implemented. The isentropic levels are currently always taken from array ilevels in namelist nh_pzlev_nml. | |
| remap | I | 0 | | interpolate horizontally, 0: none, 1: to regular lat-lon grid, 2: to Gaussian grids, (3:...) Currently only 0 and 1 are implemented. | |
| remap_internal | L | .FALSE. | | do interpolations online in the model or external (including triggering) Currently unused, interpolations are always done internally. | |
| reg_lon_def(3) | R | None | | if remap=1: start, increment, end longitude in degrees | |
| reg_lat_def(3) | R | None | | if remap=1: start, increment, end latitude in degrees | |
| gauss_tgrid_def | I | None | | if remap=2: triangular truncation (e.g.63 for T63) for which the Gauss grid should be used Currently unused since Gaussian grids are not implemented. | |
| north_pole(2) | R | 0,90 | | definition of north pole for rotated lon-lat grids. | |

Variable Groups: Using the "group:" keyword for the namelist parameters `ml_varlist`, `hl_varlist`, `pl_varlist`, sets of common variables can be added to the output:

| | |
|---|--|
| <code>group:atmo_ml_vars</code> | basic atmospheric variables on model levels |
| <code>group:atmo_pl_vars</code> , <code>group:atmo_zl_vars</code> | same set as <code>atmo_ml_vars</code> , but except pres and height, respectively |
| <code>group:nh_prog_vars</code> | additional prognostic variables of the nonhydrostatic model |
| <code>group:atmo_derived_vars</code> | derived atmospheric variables |
| <code>group:rad_vars</code> | |

group:precip_vars
group:cloud_diag
group:pbl_vars
group:phys_tendencies
group:land_vars
group:multisnow_vars
tile-averaged variables

Defined and used in: src/namelist/mo_name_list_output.f90

3.20 meteogram_output_nml

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|----------|-------------------------------------|------|---|-------|
| lmeteogram_enabled | L(n_dom) | .FALSE. | | Flag. True, if meteogram of output variables is desired. | |
| zprefix | C(n_dom) | "METEOGRAM_" | | string with file name prefix for output file | |
| ldistributed | L(n_dom) | .TRUE. | | Flag. Separate files for each PE. | |
| n0_mtgrm | I(n_dom) | 1 | | initial time step for meteogram output | |
| ninc_mtgrm | I(n_dom) | 1 | | output interval (in time steps) | |
| stationlist_tot | | 53.633, 9.983, 'Ham- burg' | | list of meteogram stations (triples with lat, lon, name string) | |

Defined and used in: src/namelist/mo_mtgrm_nml.f90

3.21 nh_pzlev_nml

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|---------------------------|--------------|
| nzlev | I | 10 | | number of height levels | iequations=3 |
| nplev | I | 10 | | number of pressure levels | iequations=3 |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|--|------|----------------------------|--|
| nilev | I | 3 | | number of isentropes | iequations=3 |
| zlevels | R | 10000, 9000, ..., 1000, 0 | m | array of height levels | iequations=3 level ordering from TOA to bottom |
| plevels | R | 100000, 90000, 80000, ..., 10000 | Pa | array of pressure levels | iequations=3 level ordering from TOA to bottom |
| ilevels | R | 340, 320, 300 | K | array of isentropic levels | iequations=3 level ordering from TOA to bottom |

Defined and used in: src/namelist/mo_nh_pzlev_nml.f90

3.22 transport_nml (used if run_nml/ltransport=.TRUE.)

| Parameter | Type | Default | Unit | Description | Scope |
|--------------|------------|------------|------|--|--|
| lvadv_tracer | L | .TRUE. | | TRUE : compute vertical tracer advection FALSE: do not compute vertical tracer advection | |
| ihadv_tracer | I(ntracer) | 2 5 | | Tracer specific method to compute horizontal advection: 0: no horiz. transport 1: upwind (1st order) 2: miura (2nd order, lin. reconstr.) 20: miura (2nd order, lin. reconstr.) with subcycling 3: miura3 (quadr. or cubic reconstr.) 22: combination of miura and miura with subcycling | if cell_type=3 if cell_type=3 lsq_high_ord ∈ [2,3] if cell_type=3 |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------|------------|---------|------|---|--|
| | | | | 32: combination of miura3 and miura with subcycling 4: FFSL (quadr. or cubic reconstr.) 5: up3 (3rd or 4th order upstream) | if cell_type=3 lsq_high_ord ∈ [2,3] if cell_type=6 |
| ivadv_tracer | I(ntracer) | 3 | | Tracer specific method to compute vertical advection: 0: no vert. transport 1: upwind (1st order) 2: muscl_cfl (2nd order, handles CFL > 1) 20: muscl (2nd order) 3: ppm_cfl (3 rd order, handles CFL > 1) 30: ppm (3rd order) | lvadv_tracer=TRUE |
| lstrang | L | .FALSE. | | splitting into fractional steps - second order Strang splitting (.TRUE.) - first order Godunov splitting (.FALSE.) | |
| ctracer_list | C | ” | | list of tracer names | |
| itype_hlimit | I(ntracer) | 3 4 | | Type of limiter for horizontal transport: 0: no limiter 3: monotonous flux limiter 4: positive definite flux limiter | ihadv_tracer ≠ 'iup3[4]' |
| itype_vlimit | I(ntracer) | 1 | | Type of limiter for vertical transport: 0: no limiter 1: semi-monotone slope limiter 2: monotonous slope limiter 4: positive definite flux limiter | |
| niter_fct | I | 1 | | number of iterations of monotone flux correction procedure | ihadv_tracer = 3, 32, 4 itype_hlimit = 3 |
| beta_fct | R | 1.0 | | factor for multiplicative spreading of range of permissible values (limiter) Tentative suggestion: beta_fct=1.0015 | ihadv_tracer = 3, 32, 4 itype_hlimit = 3 |
| iord_backtraj | I | 1 | | order of backward trajectory calculation: 1: first order | |

| Parameter | Type | Default | Unit | Description | Scope |
|----------------|------|---------|------|---|----------------------|
| | | | | 2: second order (iterative; currently 1 iteration hardcoded) | ihadv_tracer='miura' |
| igrad_c_miura | I | 1 | | Method for gradient reconstruction at cell center for 2nd order miura 1: Least-squares (linear, non-consv) 2: Green-Gauss | ihadv_tracer=2 |
| ivcfl_max | I | 5 | | determines stability range of vertical PPM-scheme in terms of the maximum allowable CFL-number | ivadv_tracer=3 |
| llsq_svd | L | .FALSE. | | use QR decomposition (FALSE) or SV decomposition (TRUE) for least squares design matrix A | |
| lclip_tracer | L | .FALSE. | | Clipping of negative values | |
| upstr_beta_adv | R | 1.0 | | parameter to select 3rd order (=1) or 4th order (=0) advection, or something inbetween (0..1) | ihadv_tracer=iup3 |

Defined and used in: src/namelist/mo_advection_nml.f90

3.23 nwp_phy_nml

The switches for the physics schemes and the time steps can be set for each model domain individually. If only one value is specified, it is copied to all child domains, implying that the same set of parameterizations and time steps is used in all domains. If the number of values given in the namelist is larger than 1 but less than the number of model domains, then the settings from the highest domain ID are used for the remaining model domains. If the time steps are not an integer multiple of the advective time step ($\text{dtime} \times \text{iadv_rcf}$), then the time step of the respective physics parameterization is automatically rounded to the next higher integer multiple of the advective time step.

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|----------------|---------|-------|--|------------------------|
| inwp_gscp | I (max_dom) | 1 | | cloud microphysics and precipitation 0: none 1: hydci (COSMO-EU microphysics) 9: Kessler scheme | run_nml/forcing = inwp |
| qi0 | R | 0.0 | kg/kg | cloud ice threshold for autoconversion | inwp_gscp=1 |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|----------------|----------------|-------|--|------------------------|
| <i>qc0</i> | R | 0.0 | kg/kg | cloud water threshold for autoconversion | inwp_gsep=1 |
| inwp_convection | I | 1 (max_dom) | | convection 0: none 1: Tiedtke/Bechtold convection | run_nml/forcing = inwp |
| inwp_cldcover | I (max_dom) | 3 | | cloud cover scheme for radiation 0: no clouds (only QV) 1: grid-scale clouds and QV 2: clouds from COSMO turbulence scheme 3: clouds from COSMO SGS cloud scheme | run_nml/forcing = inwp |
| inwp_radiation | I (max_dom) | 1 | | radiation 0: none 1: RRTM radiation 2: Ritter-Geleyn radiation | run_nml/forcing = inwp |
| inwp_satad | I | 1 | | saturation adjustment 0: none 1: | run_nml/forcing = inwp |
| inwp_turb | I (max_dom) | 1 | | vertical diffusion and transfer 0: none 1: COSMO diffusion and transfer 2: GME turbulence scheme (to be implemented) 3: EDMF-DUALM (work in progress) 4: ECHAM diffusion (currently for water only) | run_nml/forcing = inwp |
| inwp_sso | I (max_dom) | 1 | | subgrid scale orographic drag 0: none 1: (COSMO) Lott and Miller scheme | run_nml/forcing = inwp |
| inwp_gwd | I (max_dom) | 1 | | non-orographic gravity wave drag | run_nml/forcing = inwp |

| Parameter | Type | Default | Unit | Description | Scope |
|-------------------|----------------|---------|---------|--|------------------------|
| | | | | 0: none 1:Orr-Ern-Bechtold-scheme(IFS) | |
| inwp_surface | I (max_dom) | 1 | | surface scheme 0: none 1: TERRA | run_nml/forcing = inwp |
| ustart_raylfric | R | 160.0 | m/s | wind speed at which extra Rayleigh friction starts | inwp_gwd > 0 |
| efdt_min_raylfric | R | 10800. | s | minimum e-folding time of Rayleigh friction (effective for u > ustart_raylfric + 90 m/s) | inwp_gwd > 0 |
| latm_above_top | L (max_dom) | .FALSE. | | .TRUE.: take into account atmosphere above model top for radiation computation | inwp_radiation > 0 |
| itype_z0 | I | 1 | | Type of roughness length data used for turbulence scheme: 1 = including contribution from sub-scale orography, 2 = land-cover-related roughness only | inwp_turb > 0 |
| dt_conv | R (max_dom) | 600. | seconds | time interval of convection call currently each subdomain has the same value | run_nml/forcing = inwp |
| dt_ccov | R (max_dom) | dt_conv | seconds | time interval of cloud cover call currently each subdomain has the same value | run_nml/forcing = inwp |
| dt_rad | R (max_dom) | 1800. | seconds | time interval of radiation call currently each subdomain has the same value | run_nml/forcing = inwp |
| dt_sso | R (max_dom) | 1200. | seconds | time interval of sso call currently each subdomain has the same value | run_nml/forcing = inwp |
| dt_gwd | R (max_dom) | 1200. | seconds | time interval of gwd call currently each subdomain has the same value | run_nml/forcing = inwp |

Defined and used in: src/namelist/mo_atm_phy_nwp_nml.f90

3.24 radiation_nml

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|------------------------------------|------|---|-------|
| ldiur | L | .TRUE. | | switch for solar irradiation: .TRUE.:diurnal cycle, .FALSE.:zonally averaged irradiation | |
| nmonth | I | 0 | | 0: Earth circles on orbit 1-12: Earth orbit position fixed for specified month | |
| lyr_perp | L | .FALSE. | | .FALSE.: transient Earth orbit following VSOP87 .TRUE.: Earth orbit of year yr_perp of the VSOP87 orbit is perpetuated | |
| yr_perp | L | -99999 | | year used for lyr_perp = .TRUE. | |
| isolrad | I | 0 | | Insolation scheme 0: Use insolation defined in code. 1: Use insolation from external file containing the spectrally resolved insolation averaged over a year (not yet implemented) | |
| izenith | I | 3 4 (for iforcing = inwp) | | Choice of zenith angle formula for the radiative transfer computation. 0: Sun in zenith everywhere 1: Zenith angle depends only on latitude 2: Zenith angle depends only on latitude. Local time of day fixed at 07:14:15 for radiative transfer computation ($\sin(\text{time of day}) = 1/\pi$) 3: Zenith angle changing with latitude and time of day 4: Zenith angle and irradiance changing with season, latitude, and time of day (iforcing=inwp only) | |

| Parameter | Type | Default | Unit | Description | Scope |
|---|------|--|------|---|--|
| irad_h2o irad_co2 irad_ch4 irad_n2o irad_o3 irad_o2 irad_cfc11 irad_cfc12 irad_aero | I | 1 2 3 3 3 2 2 2 2 | | Switches for the concentration of radiative agents 0: 0. 1: prognostic variable 2: global constant 3: externally specified irad_aero = 5: Tanre aerosol climatology for run_nml/forcing = 3 (NWP) irad_aero = 6: Tegen aerosol climatology for run_nml/forcing = 3 (NWP) .AND. itopo =1 irad_o3 = 2: ozone climatology from MPI irad_o3 = 4: ozone clim for Aqua Planet Exp irad_o3 = 6: ozone climatology with T5 geographical distribution and Fourier series for seasonal cycle for run_nml/forcing = 3 (NWP) irad_o3 = 7: GEMS ozone climatology (from IFS) for run_nml/forcing = 3 (NWP) | Note: until further notice, please use irad_h2o = 1 irad_co2 = 2 and 0 for all the other agents for run_nml/forcing = 2 (ECHAM). |
| vmr_co2 vmr_ch4 vmr_n2o vmr_o2 vmr_cfc11 vmr_cfc12 | R | 353.9e-6 1693.6e-9 309.5e-9 0.20946 252.8e-12 466.2e-12 | | Volume mixing ratio of the radiative agents | |

Defined and used in: src/namelists/mo_radiation_nml.f90

3.25 nwp_lnd_nml

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|--|--------------------|
| nlev_snow | I | 1 | | number of snow layers for lmulti_snow=.true. | lmulti_snow=.true. |
| ntiles | I | 1 | | number of tiles | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|------|---------|------|---|----------|
| lsnowtile | L | .FALSE. | | .TRUE.: consider snow-covered and snow-free tiles separately | ntiles>1 |
| frlnd_thrhd | R | 0.5 | | fraction threshold for creating a land grid point | ntiles>1 |
| frlndtile_thrhd | R | 0.05 | | fraction threshold for retaining the respective tile for a grid point | ntiles>1 |
| frlake_thrhd | R | 0.5 | | fraction threshold for creating a lake grid point | ntiles>1 |
| frsea_thrhd | R | 0.5 | | fraction threshold for creating a sea grid point | ntiles>1 |
| nztlev | I | 2 | | used time integration scheme | |
| lmulti_snow | L | .FALSE. | | .TRUE. for use of multi-layer snow model | |
| idiag_snowfrac | I | 1 | | Type of snow-fraction diagnosis: 1 = based on SWE only, 2 = more advanced experimental method | |
| lseaice | L | .FALSE. | | .TRUE. for use of sea-ice model | |
| llake | L | .FALSE. | | .TRUE. for use of lake model | |

Defined and used in: src/namelist/mo_nwp_lnd_nml.f90

3.26 echam_phy_nml

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|--|---|
| lrad | L | .TRUE. | | Switch on radiation. | iforcing = 2 |
| lvdiff | L | .TRUE. | | Switch on turbulent mixing (i.e. vertical diffusion). | iforcing = 2 |
| lconv | L | .TRUE. | | Switch on cumulus convection. | iforcing = 2 |
| lcond | L | .TRUE. | | Switch on large scale condensation. | iforcing = 2 |
| lcover | L | .FALSE. | | .TRUE. for prognostic cloud cover scheme, .FALSE. for diagnostic scheme. | iforcing = 2 Note: lcover = .TRUE. runs, but has not been evaluated (yet) in ICON. |
| lgw_hines | L | .FALSE. | | .TRUE. for atmospheric gravity wave drag by the Hines scheme | iforcing = 2 |
| lssodrag | L | .FALSE. | | .TRUE. for subgrid scale orographic drag | iforcing = 2 Not implemeted yet |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|--------|---|-------------------------------------|
| llandsurf | L | .FALSE. | | .TRUE. for surface exchanges | iforcing = 2 Not implemented yet |
| lice | L | .FALSE. | | .TRUE. for sea-ice temperature calculation | iforcing = 2 Not implemented yet |
| lmeltpond | L | .FALSE. | | .TRUE. for calculation of meltponds | iforcing = 2 Not implemented yet |
| lhd | L | .FALSE. | | .TRUE. for hydrologic discharge model | iforcing = 2 Not implemented yet |
| lmlo | L | .FALSE. | | .TRUE. for mixed layer ocean | iforcing = 2 Not implemented yet |
| dt_rad | R | 3600. | second | time interval of full radiation computation | run_nml/iforcing = iecham |

Defined and used in: src/namelist/mo_echam_phy_nml.f90

3.27 echam_conv_nml

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|--|--------------------------------------|
| iconv | I | 1 | | Choice of cumulus convection scheme. 1: Nordeng scheme 2: Tiedtke scheme 3: hybrid scheme | iforcing = 2 .AND. lconv = .TRUE. |
| ncvmicro | I | 0 | | Choice of convective microphysics scheme. | iforcing = 2 .AND. lconv = .TRUE. |
| lmfpen | L | .TRUE. | | Switch on penetrative convection. | iforcing = 2 .AND. lconv = .TRUE. |
| lmfmid | L | .TRUE. | | Switch on midlevel convection. | iforcing = 2 .AND. lconv = .TRUE. |
| lmfdd | L | .TRUE. | | Switch on cumulus downdraft. | iforcing = 2 .AND. lconv = .TRUE. |
| lmfdudv | L | .TRUE. | | Switch on cumulus friction. | iforcing = 2 .AND. lconv = .TRUE. |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|---|-----------------------------------|
| cmftau | R | 10800. | | Characteristic convective adjustment time scale. | iforcing = 2 .AND. lconv = .TRUE. |
| cmfctop | R | 0.3 | | Fractional convective mass flux (valid range [0,1]) across the top of cloud | iforcing = 2 .AND. lconv = .TRUE. |
| cprcon | R | 1.0e-4 | | Coefficient for determining conversion from cloud water to rain. | iforcing = 2 .AND. lconv = .TRUE. |
| cminbuoy | R | 0.025 | | Minimum excess buoyancy. | iforcing = 2 .AND. lconv = .TRUE. |
| entrpen | R | 1.0e-4 | | Entrainment rate for penetrative convection. | iforcing = 2 .AND. lconv = .TRUE. |
| dlev | R | 3.e4 | Pa | Critical thickness necessary for the onset of convective precipitation. | iforcing = 2 .AND. lconv = .TRUE. |

Defined and used in: src/namelist/mo_echam_conv_nml.f90

3.28 vdiff_nml

| Parameter | Type | Default | Unit | Description | Scope |
|----------------|------|---------|------|--|-----------------|
| lsfc_mon_flux | L | .TRUE. | | Switch on surface momentum flux. | lvdiff = .TRUE. |
| lsfc_heat_flux | L | .TRUE. | | Switch on surface sensible and latent heat flux. | lvdiff = .TRUE. |

Defined and used in: src/namelist/mo_vdiff_nml.f90

3.29 turbdiff_nml

| Parameter | Type | Default | Unit | Description | Scope |
|------------|------|---------|------|---|---------------|
| itype_tran | I | 2 | | type of surface-atmosphere transfer | inwp_turb = 1 |
| imode_tran | I | 1 | | mode of surface-atmosphere transfer | inwp_turb = 1 |
| icldm_tran | I | 0 | | mode of cloud representation in transfer parametr | inwp_turb = 1 |
| imode_turb | I | 3 | | mode of turbulent diffusion parametrization | inwp_turb = 1 |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------|------|---------|------|--|-----------------------------------|
| icldm_turb | I | 2 | | mode of cloud representation in turbulence parametr | inwp_turb = 1 |
| itype_sher | I | 1 | | type of shear production for TKE | inwp_turb = 1 |
| ltkesso | L | .FALSE. | | calculation SSO-wake turbulence production for TKE | inwp_turb = 1 |
| ltkecon | L | .FALSE. | | consider convective buoyancy production for TKE | inwp_turb = 1 |
| lexpcor | L | .FALSE. | | explicit corrections of the implicit calculated turbul. diff. | inwp_turb = 1 |
| ltmpcor | L | .FALSE. | | consideration of thermal TKE-sources in the enthalpy budget | inwp_turb = 1 |
| lprfcor | L | .FALSE. | | using the profile values of the lowest main level instead of the mean value of the lowest layer for surface flux calulations | inwp_turb = 1 |
| lnonloc | L | .FALSE. | | nonlocal calculation of vertical gradients used for turbul. diff. | inwp_turb = 1 |
| lcpfluc | L | .FALSE. | | consideration of fluctuations of the heat capacity of air | inwp_turb = 1 |
| limpltkediff | L | .TRUE. | | consideration of fluctuations of the heat capacity of air | inwp_turb = 1 |
| itype_wcld | I | 2 | | type of water cloud diagnosis | inwp_turb = 1 |
| itype_synd | I | 2 | | type of diagnostics of synoptical near surface variables | inwp_turb = 1 |
| lconst_z0 | L | .FALSE. | | TRUE: horizontally homogeneous roughness lenght z0 | inwp_turb = 1 |
| const_z0 | R | 0.001 | m | value for horizontally homogeneous roughness lenght z0 | inwp_turb = 1 lconst_z0=.TRUE. |

Defined and used in: src/namelist/mo_turbdiff_nml.f90

3.30 gw_hines_nml (Scope: lgw_hines = .TRUE. in echam_phy_nml)

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|-------------|-------|
|-----------|------|---------|------|-------------|-------|

| Parameter | Type | Default | Unit | Description | Scope |
|---------------|------|---------|-------|---|----------------------|
| lheatcal | L | .FALSE. | | .TRUE.: compute drag, heating rate and diffusion coefficient from the dissipation of gravity waves .FALSE.: compute drag only | |
| emiss_lev | I | 10 | | Index of model level, counted from the surface, from which the gravity wave spectra are emitted | |
| rmscon | R | 1.0 | m/s | Root mean square gravity wave wind at the emission level | |
| kstar | R | 5.0e-5 | 1/m | Typical gravity wave horizontal wavenumber | |
| m_min | R | 0.0 | 1/m | Minimum bound in vertical wavenumber | |
| lrmscon_lat | L | .FALSE. | | .TRUE.: use latitude dependent rms wind - latitude >= lat_rmscon: use rmscon - latitude <= lat_rmscon_eq: use rmscon_eq - lat_rmscon_eq < latitude < lat_rmscon: use linear interpolation between rmscon_eq and rmscon .FALSE.: use globally constant rms wind rmscon | |
| lat_rmscon_eq | R | 5.0 | deg N | rmscon_eq is used equatorward of this latitude | lrmscon_lat = .TRUE. |
| lat_rmscon | R | 10.0 | deg N | rmscon is used polward of this latitude | lrmscon_lat = .TRUE. |
| rmscon_eq | R | 1.2 | m/s | is used equatorward of latitude lat_rmscon_eq | lrmscon_lat = .TRUE. |

Defined and used in: src/namelist/mo_gw_hines_nml.f90

3.31 ocean_physics_nml

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------------|------|---------|------|--|-------|
| i_sea_ice | I | 0 | | 1: Winton, 2:Semtner .FALSE.: compute drag only | |
| richardson_factor_tracer | I | 0.5e-5 | m/s | | |
| richardson_factor_veloc | I | 0.5e-5 | m/s | | |
| l_constant_mixing | L | .FALSE. | | | |

4 Namelist parameters for testcases (NAMELIST_ICON)

The ICON model code includes several experiments, so-called test cases, for the shallow water model as well as the 3-dimensional atmosphere. Depending on the specified experiment, initial conditions and boundary conditions are computed internally.

4.1 ha_testcase_nml (Scope: ltestcase=.TRUE. and iequations=[0,1,2] in run_nml)

| Parameter | Type | Default | Unit | Description | Scope |
|------------|------|---------|------|--|---|
| ctest_name | C | 'JWw' | | Name of test case: 'SW_GW': gravity wave 'USBR': unsteady solid body rotation 'Will_2': Williamson test 2 'Will_3': Williamson test 3 'Will_5': Williamson test 5 'Will_6': Williamson test 6 'GW': gravity wave (nlev=20 only!) 'LDF': local diabatic forcing test without physics 'LDF-Moist': local diabatic forcing test with physics initialised with zonal wind field 'HS': Held-Suarez test 'JWs': Jablonowski-Will. steady state 'JWw': Jablonowski-Will. wave test 'JWw-Moist': Jablonowski-Will. wave test including moisture 'APE': aqua planet experiment 'MRW': mountain induced Rossby wave 'MRW2': modified mountain induced Rossby wave 'PA': pure advection 'SV': stationary vortex 'DF1': deformational flow test 1 'DF2': deformational flow test 2 | lshallow_water=.TRUE. lshallow_water=.TRUE. lshallow_water=.TRUE. lshallow_water=.TRUE. lshallow_water=.TRUE. lshallow_water=.TRUE. lshallow_water=.FALSE. lshallow_water=.FALSE. and iforcing=4 lshallow_water=.FALSE., and iforcing=5 lshallow_water=.FALSE. lshallow_water=.FALSE. lshallow_water=.FALSE. lshallow_water=.FALSE. lshallow_water=.FALSE. lshallow_water=.FALSE. lshallow_water=.FALSE. lshallow_water=.FALSE. lshallow_water=.FALSE., ntracer = 2 |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|------|-----------|------------|---|--|
| | | | | 'DF3': deformational flow test 3 'DF4': deformational flow test 4 'RH': Rossby-Haurwitz wave test | lshallow_water=.FALSE. |
| rotate_axis_deg | R | 0.0 | deg | Earth's rotation axis pitch angle | ctest_name= 'Will_2', 'Will_3', 'JWs', 'JWw', 'PA', 'DF1234' |
| gw_brunt_vais | R | 0.01 | 1/s | Brunt Vaisala frequency | ctest_name= 'GW' |
| gw_u0 | R | 0.0 | m/s | zonal wind parameter | ctest_name= 'GW' |
| gw_lon_deg | R | 180.0 | deg | longitude of initial perturbation | ctest_name= 'GW' |
| gw_lat_deg | R | 0.0 | deg | latitude of initial perturbation | ctest_name= 'GW' |
| jw_uptb | R | 1.0 | m/s (?) | amplitude of the wave perturbation | ctest_name= 'JWw' |
| mountctr_lon_deg | R | 90.0 | deg | longitude of mountain peak | ctest_name= 'MRW(2)' |
| mountctr_lat_deg | R | 30.0 | deg | latitude of mountain peak | ctest_name= 'MRW(2)' |
| mountctr_height | R | 2000.0 | m | mountain height | ctest_name= 'MRW(2)' |
| mountctr_half_width | R | 1500000.0 | m | mountain half width | ctest_name= 'MRW(2)' |
| mount_u0 | R | 20.0 | m/s | wind speed for MRW cases | ctest_name= 'MRW(2)' |
| rh_wavenum | I | 4 | | wave number | ctest_name= 'RH' |
| rh_init_shift_deg | R | 0.0 | deg | pattern shift | ctest_name= 'RH' |
| ihs_init_type | I | 1 | | Choice of initial condition for the Held-Suarez test. 1: the zonal state defined in the JWs test case; other integers: isothermal state (T=300 K, ps=1000 hPa, u=v=0.) | ctest_name= 'HS' |
| lhs_vn_ptb | L | .TRUE. | | Add random noise to the initial wind field in the Held-Suarez test. | ctest_name= 'HS' |
| hs_vn_ptb_scale | R | 1. | m/s | Magnitude of the random noise added to the initial wind field in the Held-Suarez test. | ctest_name= 'HS' |
| lrh_linear_pres | L | .FALSE. | | Initialize the relative humidity using a linear function of pressure. | ctest_name= 'JWw-Moist', 'APE', 'LDF-Moist' |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|------|---------|------|--|--|
| rh_at_1000hpa | R | 0.75 | | relative humidity 0, 1 at 1000 hPa | ctest_name= 'JWw-Moist','APE', 'LDF-Moist' |
| limit_tracer_fv | L | .TRUE. | | Finite volume initialization for tracer fields | ctest_name='PA' |
| ape_sst_case | C | 'sst1' | | SST distribution selection 'sst1': Control experiment 'sst2': Peaked experiment 'sst3': Flat experiment 'sst4': Control-5N experiment 'sst_qobs': Qobs SST distribution exp 'sst_ice': Control SST distribution with -1.8 C above 64 N/S. | ctest_name='APE' |
| ildf_init_type | I | 0 | | Choice of initial condition for the Local diabatic forcing test. 1: the zonal state defined in the JWs test case; other: isothermal state (T=300 K, ps=1000 hPa, u=v=0.) | ctest_name= 'LDF' |
| ldf_symm | L | .TRUE. | | Shape of local diabatic forcing: .TRUE.: local diabatic forcing symmetric about the equator (at 0 N) .FALSE.: local diabatic forcing asym. about the equator (at 30 N) | ctest_name= 'LDF','LDF-Moist' |

Defined and used in: src/testcases/mo_ha_testcases.f90

4.2 nh_testcase_nml (Scope: ltestcase=.TRUE. and iequations=3 in run_nml)

| Parameter | Type | Default | Unit | Description | Scope |
|--------------|------|---------|------|--|-------|
| nh_test_name | C | 'jabw' | | testcase selection 'zero': no orography 'bell': bell shaped mountain at 0E,0N 'schaer': hilly mountain at 0E,0N | |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|---|--|
| | | | | <p>'jabw': Initializes the full Jablonowski Williamson test case.</p> <p>'jabw_s': Initializes the Jablonowski Williamson steady state test case.</p> <p>'jabw_m': Initializes the Jablonowski Williamson test case with a mountain instead of the wind perturbation (specify mount_height).</p> <p>'mrw_nh': Initializes the full Mountain-induced Rossby wave test case.</p> <p>'mrw2_nh': Initializes the modified mountain-induced Rossby wave test case.</p> <p>'mwbr_const': Initializes the mountain wave with two layers test case. The lower layer is isothermal and the upper layer has constant brunt vaisala frequency. The interface has constant pressure.</p> <p>'PA': Initializes the pure advection test case.</p> <p>'HS_nh': Initializes the Held-Suarez test case. At the moment with an isothermal atmosphere at rest (T=300K, ps=1000hPa, u=v=0, topography=0.0).</p> <p>'HS_jw': Initializes the Held-Suarez test case with Jablonowski Williamson initial conditions and zero topography.</p> <p>'APE_nh': Initializes the APE experiments. With the jabw test case, including moisture.</p> <p>'wk82': Initializes the Weisman Klemp test case</p> <p>'g_lim_area': Initializes a series of general limited area test cases: itype_atmos_ana determines the atmospheric profile, itype_anaprof_uv determines the wind profile and itype_topo_ana determines the topography</p> <p>'dcnip_rest_200': atmosphere at rest test (Schaer-type mountain)</p> <p>'dcnip_mw_2x': nonhydrostatic mountain waves triggered by Schaer-type mountain</p> | <p>l_limited_area = .TRUE.</p> <p>lcoriolis = .FALSE.</p> <p>lcoriolis = .FALSE.</p> |

| Parameter | Type | Default | Unit | Description | Scope |
|------------------------|------|-----------|---------|--|--|
| | | | | 'dcmip_gw_31' : nonhydrostatic gravity waves triggered by a localized perturbation (nonlinear) 'dcmip_gw_32' : nonhydrostatic gravity waves triggered by a localized perturbation (linear) 'dcmip_tc_51' : tropical cyclone test case with 'simple physics' parameterizations (not yet implemented) 'dcmip_tc_52' : tropical cyclone test case with full physics in Aqua-planet mode | l_limited_area = .TRUE. and lcoriolis = .FALSE. lcoriolis = .TRUE. lcoriolis = .TRUE. |
| jw_up | R | 1.0 | m/s | amplitude of the u-perturbation in jabw test case | nh_test_name='jabw' |
| u0_mrw | R | 20.0 | m/s | wind speed for mrw(2) and mwbr_const cases | nh_test_name='mrw(2)_nh' and 'mwbr_const' |
| mount_height_mrw | R | 2000.0 | m | maximum mount height in mrw(2) and mwbr_const | nh_test_name='mrw(2)_nh' and 'mwbr_const' |
| mount_half_width | R | 1500000.0 | m | half width of mountain in mrw(2), mwbr_const and bell | nh_test_name='mrw(2)_nh', 'mwbr_const' and 'bell' |
| mount_lonctr_mrw_deg | R | 90. | degrees | lon of mountain center in mrw(2) and mwbr_const | nh_test_name='mrw(2)_nh' and 'mwbr_const' |
| mount_latctr_mrw_deg | R | 30. | degrees | lat of mountain center in mrw(2) and mwbr_const | nh_test_name='mrw(2)_nh' and 'mwbr_const' |
| temp_i_mwbr_const | R | 288.0 | K | temp at isothermal lower layer for mwbr_const case | nh_test_name='mwbr_const' |
| p_int_mwbr_const | R | 70000. | Pa | pres at the interface of the two layers for mwbr_const case | nh_test_name='mwbr_const' |
| bruntvais_u_mwbr_const | R | 0.025 | 1/s | constant brunt vaissala frequency at upper layer for mwbr_const case | nh_test_name='mwbr_const' |
| mount_height | R | 100.0 | m | peak height of mountain | nh_test_name='bell' |

| Parameter | Type | Default | Unit | Description | Scope |
|---------------------|------|----------|-------|---|---|
| layer_thickness | R | -999.0 | m | thickness of vertical layers | If layer_thickness < 0, the vertical level distribution is read in from externally given HYB_PARAMS_XX. |
| n_flat_level | I | 2 | | level number for which the layer is still flat and not terrain-following | layer_thickness > 0 |
| nh_u0 | R | 0.0 | m/s | initial constant zonal wind speed | nh_test_name = 'bell' |
| nh_t0 | R | 300.0 | K | initial temperature at lowest level | nh_test_name = 'bell' |
| nh_brunt_vais | R | 0.01 | 1/s | initial Brunt-Vaisala frequency | nh_test_name = 'bell' |
| torus_domain_length | R | 100000.0 | m | length of slice domain | nh_test_name = 'bell', lplane=.TRUE. |
| rotate_axis_deg | R | 0.0 | deg | Earth's rotation axis pitch angle | nh_test_name= 'PA' |
| lhs_nh_vn_ptb | L | .TRUE. | | Add random noise to the initial wind field in the Held-Suarez test. | nh_test_name= 'HS_nh' |
| lhs_fric_heat | L | .FALSE. | | add frictional heating from Rayleigh friction in the Held-Suarez test. | nh_test_name= 'HS_nh' |
| hs_nh_vn_ptb_scale | R | 1. | m/s | Magnitude of the random noise added to the initial wind field in the Held-Suarez test. | nh_test_name= 'HS_nh' |
| rh_at_1000hpa | R | 0.7 | 1 | relative humidity at 1000 hPa | nh_test_name= 'jabw', nh_test_name= 'mrw' |
| qv_max | R | 20.e-3 | kg/kg | specific humidity in the tropics | nh_test_name= 'jabw', nh_test_name= 'mrw' |
| ape_sst_case | C | 'sst1' | | SST distribution selection 'sst1': Control experiment 'sst2': Peaked experiment 'sst3': Flat experiment 'sst4': Control-5N experiment 'sst_qobs': Qobs SST distribution exp. | nh_test_name='APE_nh' |
| limit_tracer_fv | L | .TRUE. | | Finite volume initialization for tracer fields | pure advection tests, only |
| lcoupled_rho | L | .FALSE. | | Integrate density equation 'offline' | pure advection tests, only |

| Parameter | Type | Default | Unit | Description | Scope |
|--------------------|------|---------|-------|---|---|
| qv_max_wk | R | 0.014 | Kg/kg | maximum specific humidity near the surface, range 0.012 - 0.016 used to vary the buoyancy | nh_test_name='wk82' |
| u_infty_wk | R | 20. | m/s | zonal wind at infinity height range 0. - 45. used to vary the wind shear | nh_test_name='wk82' |
| bub_amp | R | 2. | K | maximum amplitud of the thermal perturbation | nh_test_name='wk82' |
| bubctr_lat | R | 0. | deg | latitude of the center of the thermal perturbation | nh_test_name='wk82' |
| bubctr_lon | R | 90. | deg | longitude of the center of the thermal perturbation | nh_test_name='wk82' |
| bubctr_z | R | 1400. | m | height of the center of the thermal perturbation | nh_test_name='wk82' |
| bub_hor_width | R | 10000. | m | horizontal radius of the thermal perturbation | nh_test_name='wk82' |
| bub_ver_width | R | 1400. | m | vertical radius of the thermal perturbation | nh_test_name='wk82' |
| itype_atmo_ana | I | 1 | | kind of atmospheric profile: 1 piecewise N constant layers 2 piecewise polytropic layers | nh_test_name= 'g_lim_area' |
| itype_anaprof_uv | I | 1 | | kind of wind profile: 1 piecewise linear wind layers 2 constant zonal wind 3 constant meridional wind | nh_test_name= 'g_lim_area' |
| itype_topo_ana | I | 1 | | kind of orography: 1 schaer test case mountain 2 gaussian_2d mountain 3 gaussian_3d mountain any other no orography | nh_test_name= 'g_lim_area' |
| nlayers_nconst | I | 1 | | Number of the desired layers with a constant Brunt-Vaisala-frequency | nh_test_name= 'g_lim_area' and itype_atmo_ana=1 |
| p_base_nconst | R | 100000. | Pa | pressure at the base of the first N constant layer | nh_test_name= 'g_lim_area' and itype_atmo_ana=1 |
| theta0_base_nconst | R | 288. | K | potential temperature at the base of the first N constant layer | nh_test_name= 'g_lim_area' and itype_atmo_ana=1 |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------------|-------------------|----------------|------|---|---|
| h_nconst | R(nlayers_nconst) | 60000., 12000. | m | height of the base of each of the N constant layers | nh_test_name= 'g_lim_area' and itype_atmo_ana=1 |
| N_nconst | R(nlayers_nconst) | 60000. | 1/s | Brunt-Vaisala-frequency at each of the N constant layers | nh_test_name= 'g_lim_area' and itype_atmo_ana=1 |
| rh_nconst | R(nlayers_nconst) | 0.5 | % | relative humidity at the base of each N constant layers | nh_test_name= 'g_lim_area' and itype_atmo_ana=1 |
| rhgr_nconst | R(nlayers_nconst) | 0. | % | relative humidity gradient at each of the N constant layers | nh_test_name= 'g_lim_area' and itype_atmo_ana=1 |
| nlayers_poly | I | 2 | | Number of the desired layers with constant gradient temperature | nh_test_name= 'g_lim_area' and itype_atmo_ana=2 |
| p_base_poly | R | 100000. | Pa | pressure at the base of the first polytropic layer | nh_test_name= 'g_lim_area' and itype_atmo_ana=2 |
| h_poly | R(nlayers_poly) | 2000. | m | height of the base of each of the polytropic layers | nh_test_name= 'g_lim_area' and itype_atmo_ana=2 |
| t_poly | R(nlayers_poly) | 288., 213. | K | temperature at the base of each of the polytropic layers | nh_test_name= 'g_lim_area' and itype_atmo_ana=2 |
| rh_poly | R(nlayers_poly) | 0.2 | % | relative humidity at the base of each of the polytropic layers | nh_test_name= 'g_lim_area' and itype_atmo_ana=2 |
| rhgr_poly | R(nlayers_poly) | 5, 0. | % | relative humidity gradient at each of the polytropic layers | nh_test_name= 'g_lim_area' and itype_atmo_ana=2 |
| nlayers_linwind | I | 2 | | Number of the desired layers with constant U gradient | nh_test_name= 'g_lim_area' and itype_anaprof_uv=1 |

| Parameter | Type | Default | Unit | Description | Scope |
|----------------|--------------------|---------|------|---|---|
| h_linwind | R(nlayers_linwind) | 250. | m | height of the base of each of the linear wind layers | nh_test_name= 'g_lim_area' and itype_anaprof_uv=1 |
| u_linwind | R(nlayers_linwind) | 15. | m/s | zonal wind at the base of each of the linear wind layers | nh_test_name= 'g_lim_area' and itype_anaprof_uv=1 |
| ugr_linwind | R(nlayers_linwind) | 10. | 1/s | zonal wind gradient at each of the linear wind layers | nh_test_name= 'g_lim_area' and itype_anaprof_uv=1 |
| vel_const | R | 20. | m/s | constant zonal/meridional wind (itype_anaprof_uv=2,3) | nh_test_name= 'g_lim_area' and itype_anaprof_uv=2,3 |
| mount_lonc_deg | R | 90. | deg | longitud of the center of the mountain | nh_test_name= 'g_lim_area' |
| mount_latc_deg | R | 0. | deg | latitud of the center of the mountain | nh_test_name= 'g_lim_area' |
| schaer_h0 | R | 250. | m | h0 parameter for the schaer mountain | nh_test_name= 'g_lim_area' and itype_topo_ana=1 |
| schaer_a | R | 5000. | m | -a- parameter for the schaer mountain, also half width in the north and south side of the finite ridge to round the sharp edges | nh_test_name= 'g_lim_area' and itype_topo_ana=1,2 |
| schaer_lambda | R | 4000. | m | lambda parameter for the schaer mountain | nh_test_name= 'g_lim_area' and itype_topo_ana=1 |
| lshear_dcmip | L | FALSE | | run dcmip_mw_2x with/without vertical wind shear FALSE: dcmip_mw_21: non-sheared TRUE : dcmip_mw_22: sheared | nh_test_name= 'dcmip_mw_2x' |
| halfwidth_2d | R | 10000. | m | half lenght of the finite ridge in the north-south direction | nh_test_name= 'g_lim_area' and itype_topo_ana=1,2 |

| Parameter | Type | Default | Unit | Description | Scope |
|-----------|------|---------|------|---|---|
| m_height | R | 1000. | m | height of the mountain | nh_test_name= 'g_lim_area' and itype_topo_ana=2,3 |
| m_width_x | R | 5000. | m | half width of the gaussian mountain in the east-west direction half width in the north-south direction in the rounding of the finite ridge (gaussian_2d) | nh_test_name= 'g_lim_area' and itype_topo_ana=2,3 |
| m_width_y | R | 5000. | m | half width of the gaussian mountain in the north-south direction | nh_test_name= 'g_lim_area' and itype_topo_ana=2,3 |

Defined and used in: src/testcases/mo_nh_testcases.f90

5 External data

5.1 extpar_nml (Scope: itopo=1 in run_nml)

| Parameter | Type | Default | Unit | Description | Scope |
|----------------------|----------|----------|------|---|---------------------------|
| itopo | I | 0 | | 0: analytical topography/ext. data 1: topography/ext. data read from file | |
| n_iter_smooth_topo | I(n_dom) | 0 | | iterations of topography smoother | itopo = 1 |
| fac_smooth_topo | R | 0.015625 | | pre-factor of topography smoother | n_iter_smooth_topo > 0 |
| heightdiff_threshold | R(n_dom) | 3000. | m | height difference between neighboring grid points above which additional local nabla2 diffusion is applied | |
| l_emiss | L | .TRUE. | | read and use external surface emissivity map | itopo = 1 |
| extpar_filename | C | | | Filename of external parameter input file, default: "<path>extpar_<gridfile>". May contain the keyword <path> which will be substituted by model_base_dir. | |

Defined and used in: src/namelist/mo_extpar_nml.f90

6 External packages

6.1 art_nml

| Parameter | Type | Default | Unit | Description | Scope |
|--------------|------|---------|------|----------------------------------|-------|
| lart | L | .FALSE. | | main switch for ART-package | |
| lemi_volc | L | .FALSE. | | Emission of volcanic ash | |
| lconv_tracer | L | .FALSE. | | Convection of tracers | |
| lwash_tracer | L | .FALSE. | | Washout of tracers | |
| lrاد_volc | L | .FALSE. | | Radiative impact of volcanic ash | |
| lcld_tracer | L | .FALSE. | | Impact on clouds | |

Defined and used in: src/namelist/mo_art_nml.f90

7 Information on vertical level distribution

The hydrostatic and nonhydrostatic models need hybrid vertical level information to generate the terrain following coordinates. The hybrid level specification is stored in <icon home>/hyb_params/HYB_PARAMS_<nlev>. The **hydrostatic** model assumes to get **pressure based** coordinates, the **nonhydrostatic** model expects **height based** coordinates. For further information see <icon home>/hyb_params/README.