Chapter 14

Control parameters

The parameters, discussed in this chapter (except the first section), are defined in the following routines of the file *Usrdef_Model.f90*:

- usrdef_init_params: setup of monitoring parameters (Section 14.2)
- usrdef_mod_params: switches, model parameters and attributes of forcing files (Sections 14.3-14.8)
- usrdef_MPI_partition: user-defined domain decomposition (Section 14.9)

14.1 File *defruns*

The program open this file at the start of the simulation(s) and is read linewise. Each line represents a separate run and contains the definitions of three parameters defined separated by a ',' The general systax is

```
runtitle,status,filename
```

where

runtitle the title of the simulation stored in the model parameter runtitle

status the status of the CIF

- '0' The CIF utility is switched off (both for reading and writing). This is the default condition.
- 'R' Model setup parameters are read from a CIF.
- 'W' Model setup parameters are written to a CIF.

filename Name of the CIF file. If not given, the default name TRIM(runtitle)//'.cifmodA' is taken. This parameter is obviously not used if status equals '0'

Defaults are taken (except for runtitle which must always be given) when the value is an empty string, one blank or several blanks. All blanks are ignored on the input line.

Consider the following example

```
conesA,,
conesA,R,
conesA,W,myciffile
```

In the example the first line initiates the run conesA without CIF, the second one reads the setup from the file *conesA.cifmodA*, the third writes the CIF data to the file *myciffile*.

Lines can be commented if the first character is a '!'. This replaces, for compatibility with the CIF syntax below, the '#' character used in previous versions.

The procedure is used to combine multiple simulations within one run:

1. The program opens the file at the start.
2. The first line is read.
3. A simulation is started with the given title.
4. When the simulation ends, a next line is read giving a new title and a next simulation initiates.
5. When there are no more lines to be read, the file is closed and the program terminates.

14.2 Parameters for monitoring

This section describes the parameters used to set up the monitoring and a few other general parameters. They are defined in usrdef_init_params. The routine is called by *all* processes.

14.2.1 Cold start

LOGICAL :: cold_start If .TRUE., the program executes model initialisation and finalisation, but does not enter the time loop (.FALSE.). If *defruns* contains multiple lines, a cold start is performed for each simulation. The option is useful for debugging.

14.2.2 Log files

INTEGER :: levprocs_ini(npworld) Determines the level of tracing of the *inilog* file for each process. Different levels can be defined for different files. If 0 (default value), no log file will be written. The *inilog* file only contains information about model initialisation and is closed as soon as the program enters the time loop. In parallel mode, the size (npworld) of the vector array equals the number of processes, initially defined within MPI_COMM_WORLD or equals 1 in the serial case.

INTEGER :: levprocs_run(npworld) Determines the level of tracing of the *runlog* file for each process. Different levels can be defined for different files. If 0 (default value), no log file will be written. The *runlog* file traces program execution during the time loop.

CHARACTER (LEN=leniofile) :: inilog_file Name of the *inilog* file. Default is TRIM(runtitle)//'.inilogA'. In parallel mode, the name is appended with the process id number.

CHARACTER (LEN=leniofile) :: runlog_file Name of the *runlog* file. Default is TRIM(runtitle)//'.runlogA'. In parallel mode, the name is appended with the process id number.

LOGICAL :: exitlog Writes an exit statement of the form 'num:R', where 'num' is the program level in the "log"-file on exit of a routine call if .TRUE. (.TRUE.).

INTEGER :: runlog_count Sets the number of time steps after which the *runlog* file is overwritten. Default is the total number of time steps (i.e. information is written at all time steps and the file is never overwritten).

14.2.3 Error files

INTEGER :: maxerrors Maximum allowed number of error messages within the *errlog* file. Default is MaxErrMsgs defined in *syspars.f90*.

INTEGER :: levprocs_err(npworld) Level of error checking for each process (0).

- 0 : error checking disabled (for a particular processor) and no file is created
- 1 : error checking enabled during initialisation phase only
- 2 : error checking enabled throughout the whole program

CHARACTER (LEN=leniofile) :: errlog_file Name of the *errlog* file. Default is TRIM(runtitle)//'.errlogA'. In parallel mode, the name is appended with the process id number.

14.2.4 Warning file

LOGICAL :: warning Disables/enables writing of a *warning* file (.TRUE.).

CHARACTER (LEN=leniofile) :: warlog_file Name of the *warning* file. Default is TRIM(runtitle)//'.warlogA'.

14.2.5 Timer file

INTEGER :: levtimer Determines the type of information in the timer report (0).

- 0: No timer report is written.
- 1: Writes the total execution time only.
- 2: Writes time information (in % of total time) for all "timers". In case of a parallel run, the information is written as follows: time on the master process, mean, minimum and maximum time over all processes.
- 3: The same as previous, but in case of a parallel run, the information is additionally written for each individual process. In the serial case, behaviour is as for case 2.

CHARACTER (LEN=leniofile) :: timing_file Name of the *timing* file. Default is TRIM(runtitle)//'.timingA'.

INTEGER :: timer_format Format for total execution time in the timer report (1).

- 1: seconds
- 2: minutes
- 3: hours
- 4: days

14.3 Dimensions of the process domain grid

The parameters below are used to setup a domain decomposition and are defined in usrdef_mod_params. The routine is called if ciffiles(icif_model) %status = '0' or 'W'

nprocs the actual number of processes to be used (1)

nprocsx X-dimension of the decomposed domain (0)

nprocsy Y-dimension of the decomposed domain (0)

- nprocsx and nprocsy are needed by the program for making a "simple" domain decomposition when the switch iopt_MPI_partit = 1. Otherwise, if iopt_MPI_partit=2, these parameters are determined by the program.

- nprocs must be defined if the decomposition is obtained from a data file or defined in usrdef_partition. In that case its value must match the size of the arrays nc1procs, nc2procs, nr1procs, nr2procs.

In case of a simple decomposition, each (but not all) of these three parameters may be zero. However, their values must be between 0 and npworld which is the number of processes in the MPI communicator MPI_comm_world or, equivalently, the number of processes defined in the script launching the program.

The program follows the following procedures

1. nprocsx and nprocsy are non-zero: nprocs is set to nprocsx×nprocsy

2. both nprocsx and nprocsy are zero: both values are set internally so that nprocsx×nprocsy = nprocs and |nprocsx-nprocsy| is minimal

3. nprocsx is non-zero, while nprocsy is zero: nprocsy = nprocs/nprocsx

4. nprocsy is non-zero, while nprocsx is zero: nprocsx = nprocs/nprocsy

Remarks

- Cases 2–4: If nprocs is zero, its value is set to npworld.

- Case 3–4: If no integer division is possible, an error is issued.

14.4 Model switches

A total of 83 switches is implemented. They are defined in usrdef_mod_params.

14.4.1 Model grid

iopt_grid_htype Type of horizontal grid (1).

1: uniform rectangular grid

2: non-uniform rectangular grid

3: curvilinear grid

iopt_grid_nodim Grid dimension (3).

1: 1-dimensional grid (water column model)

2: 2-dimensional grid (depth-averaged model without vertical structure)

3: 3-dimensional grid

iopt_grid_sph Type of coordinates (0).

0: Cartesian coordinates

1: spherical coordinates

iopt_grid_vtype Type of vertical grid (1).

1: uniform σ-grid

2: horizontally uniform and vertically non-uniform σ-grid

3: horizontally and vertically non-uniform σ-grid

iopt_grid_vtype_transf Type of vertical grid transformation (0).

0 : uniform vertical grid (iopt_grid_vtype=1) or user-defined

11: log-transformation (4.23) at the bottom following Davies & Jones (1991) if iopt_grid_vtype=2

12: log-transformation (4.24) at the surface following Davies & Jones (1991) if iopt_grid_vtype=2

13: transformation with enhanced resolution near the bottom and/or the bottom as defined in Burchard & Bolding (2002)

21: Song & Haidvogel (1994) transformation given by (4.33) and (4.35) if iopt_grid_vtype=3

14.4.2 Interpolation

iopt_arrint_hreg Disables/enables (0/1) the use of non-uniform weighted averages for interpolation in the horizontal of arrays on the model grid (0).

iopt_arrint_vreg Disables/enables (0/1) the use of non-uniform weighted averages for interpolation in the vertical of arrays on the model grid (0).

iopt_arrint_3D Selects dimension of mask or weight factor in some array interpolations

0: 2-D masks or weights

1: 3-D masks or weights

It is recommended to set the first two of these switches only for grids with highly irregular grid spacings.

14.4.3 Hydrodynamics

iopt_curr Type of current fields (2).

0: Currents and elevations are set to their default (zero) values and are not updated.

1: Currents and elevations are initialised but not updated in time.

2: Currents and elevations are initialised and updated in time.

iopt_curr_wfall Type of formulation for the settling of particulate matter (1).

1: settling enabled without correction terms

2: settling enabled with the correction terms (7.117)–(7.118) included

iopt_hydro_impl Disables/enables the implicit scheme (0).

01: The momentum equations are solved with the explicit (mode-splitting) scheme (default).

11: The momentum equations are solved using the implicit algorithm. The compiler option -DPETSC must be set.

14.4.4 Density

iopt_dens Evaluation of the density and expansion coefficients (0).

- 0: uniform density, zero expansion coefficients
- 1: density from the linear equation of state (4.108), expansion coefficients are uniform
- 2: from the McDougall *et al.* (2003) general equation of state (4.103)–(4.107) without pressure effects
- 3: from the McDougall *et al.* (2003) general equation of state (4.103)–(4.107) with pressure effects included

iopt_dens_grad Selects numerical algorithm for discretisation of the baroclinic pressure gradient (1).

- 0: gradient set to zero
- 1: traditional σ-coordinate (second order) method
- 2: z-level method
- 3: method of Shchepetkin & McWilliams (2003)

iopt_sal Salinity update (0).

- 0: uniform (space and time) salinity field
- 1: salinity field initialised but not updated in time
- 2: salinity field initialised and updated in time

iopt_sal_sbc Type of surface boundary condition for salinity (0).

- 0: zero surface flux
- 1: surface flux given by (4.276)

iopt_temp Temperature update (0).

- 0: uniform (space and time) temperature field
- 1: temperature field initialised but not updated in time
- 2: temperature field initialised and updated in time

iopt_temp_optic Disables/enables (0/1) the optical module (1).

- 0: all solar radiation is assumed to be absorbed at the surface, i.e. the water column is considered as opaque
- 1: solar radiation is absorbed within the water column using specified values for the attenuation depths

iopt_temp_sbc Type of surface boundary condition for temperature (1).

1: Neumann condition using the model's surface heat flux formulations

2: Dirichlet using prescribed surface temperatures taken at the first grid point below the surface

3: Dirichlet using prescribed surface temperature taken at the surface itself

14.4.5 External modules

iopt_biolgy Disables/enables (0/1) the activation of an external biological module (not available) (0). Since no biological module is implemented in the current version, this default cannot be changed.

iopt_sed Disables/enables (0/1) the activation of an external sediment module (0).

14.4.6 Bottom boundary conditions

iopt_bstres_drag Formulation for the bottom drag coefficient C_{db}(3).

0: not used

1: spatially uniform value

2: spatially non-uniform obtained from a data file

3: using a spatially uniform roughness length

4: using a spatially non-uniform roughness length

iopt_bstres_form Type of formulation for the bottom stress (2).

0: bottom stress set to zero

1: linear bottom stress law (4.338) or (4.339)

2: quadratic bottom stress (4.340) or (4.339)

iopt_bstres_nodim Type of currents in the (linear or quadratic) bottom stress formulation (3).

2: depth-mean currents

3: 3-D current taken at the bottom grid cell

14.4.7 Advection

iopt_adv_scal Type of scheme for the advection of scalar quantities (3).

- 0: advection disabled
- 1: upwind scheme
- 2: Lax-Wendroff (explicit) in the horizontal, central (semi-implicit) in the vertical
- 3: TVD scheme

iopt_adv_tvd Type of limiting function for TVD scheme (1).

- 1: superbee limiter
- 2: monotone limiter

iopt_adv_turb Type of scheme for the advection of turbulence quantities (0).

- 0: advection disabled
- 1: upwind scheme
- 2: Lax-Wendroff (explicit) in the horizontal, central (semi-implicit) in the vertical
- 3: TVD scheme

iopt_adv_2D Type of scheme for the advection of 2-D transports (1).

- 0: advection disabled
- 1: upwind scheme
- 2: Lax-Wendroff (explicit) in the horizontal, central (semi-implicit) in the vertical
- 3: TVD scheme

iopt_adv_3D Type of scheme for the advection of 3-D currents (1).

- 0: advection disabled
- 1: upwind scheme
- 2: Lax-Wendroff (explicit) in the horizontal, central (semi-implicit) in the vertical
- 3: TVD scheme

iopt_scal_depos Discretisation for the deposition (vertical advective flux at the sea bed) of particulate matter (1).

- 0: Deposition flux is set to zero.

1: first order (upwind) scheme

2: second order scheme using extrapolation

Remarks

- The Lax-Wendroff/central scheme is non-monotone and should not be selected. This is illustrated with the cases ***cones*** and ***front*** (see Sections 23.1 and 23.2).
- The TVD scheme has the ability to retain sharp gradients, but consumes more CPU time compared to the upwind scheme.
- TVD is the recommended scheme for 3-D scalars. The advantage of using a TVD scheme is less evident for the 2-D mode which uses a much smaller time step than the 3-D mode. The faster upwind scheme can be recommended for the 2-D mode in most cases. TVD is recommended for resolving highly sheared 3-D currents, as occurring in e.g. frontal zones.
- Advection of turbulence is considered of less importance than the production and dissipation terms in the k, $k-\varepsilon$ and kl transport equations. It is recommended not to change the zero default value of iopt_adv_turb.
- The same limiting function applies for all transport equations solved with the TVD scheme.

14.4.8 Diffusion coefficients

iopt_hdif_coef Type of scheme for horizontal diffusion coefficients (0).

0: not used

1: spatially uniform

2: Smagorinsky formulation (4.80) for momentum and (4.81) for scalars

iopt_hdif_scal Disables/enables (0/1) horizontal diffusion in the scalar transport equations (0).

iopt_hdif_turb Disables/enables (0/1) horizontal diffusion in the turbulence transport equations (0).

iopt_hdif_2D Disables/enables (0/1) horizontal diffusion in the 2-D transport equations (0).

iopt_hdif_3D Disables/enables (0/1) horizontal diffusion in the 3-D current transport equations (0).

iopt_kinvisc Formulation for kinematic viscosity (0).

0: user-defined uniform value kinvisc_cst

1: ITTC (1978) relation (7.24)

iopt_vdif_coef Selects the (general) type of vertical diffusion scheme (3).

0: vertical diffusion disabled

1: uniform diffusion coefficient

2: algebraic formulation as described in Section 4.4.2.2

3: second order turbulence closure as described in Section 4.4.3

Remarks

- If horizontal diffusion is enabled, the Smagorinsky formulation, taken from LES modelling, is a more robust scheme compared to a constant diffusion coefficient.
- Horizontal diffusion of scalars may be potentially dangerous since it introduces spurious diapycnal mixing.
- Horizontal diffusion of turbulence variables is only introduced for historical reasons and compatibility with COHERENS V1, but has no real physical basis.

14.4.9 Turbulence schemes

iopt_turb_alg Type of algebraic scheme if iopt_vdif_coef = 2 (1).

1: Pacanowski-Philander formulation (4.132)–(4.135)

2: Munk-Anderson formulation (4.136)–(4.140)

3: flow dependent formulation as described in Section 4.4.2.2 with α given by (4.148)

4: flow dependent formulation as described in Section 4.4.2.2 with α given by (4.149)

5: flow dependent formulation as described in Section 4.4.2.2 with α given by (4.150)

6: parabolic profile (4.154)

iopt_turb_dis_bbc Type of bottom boundary condition for the dissipation rate ε (2).

1: Neumann condition (4.353)

2: Dirichlet condition (4.351)

iopt_turb_dis_sbc Type of surface boundary condition for the dissipation rate ε (2).

1: Neumann condition (4.284)

2: Dirichlet condition (4.281)

iopt_turb_iwlim Type of background mixing scheme as described in Section 4.4.3.6 (0).

0: using uniform background coefficients

1: using limiting conditions for turbulence parameters

2: the Large *et al.* (1994) scheme given by (4.227)–(4.228)

iopt_turb_kinvisc Selects type of background mixing mixing (0).

0: user-defined constant value vdifmom_cst

1: kinematic viscosity as selected by iopt_kinvisc

iopt_turb_lmix Mixing length formulation as described in Section 4.4.3.5 (4).

1: parabolic law (4.213)

2: "modified" parabolic law (4.214)

3: "Xing" formulation (4.215)

4: "Blackadar" asymptotic formulation (4.216)

iopt_turb_ntrans Number of transport equations as described in Section 4.4.3.4 (1).

0: zero-equation model (equilibrium or Mellor-Yamada level 2 method) with a mixing length selected by iopt_turb_lmix

1: turbulence energy equation with a mixing length selected by iopt_turb_lmix

2: k-ε of k-kl equation depending on the value of iopt_turb_param

iopt_turb_param Selects type of second turbulent variable (2).

1: mixing length l (k-l scheme)

2: dissipation rate ε (k-ε scheme)

iopt_turb_stab_form Selects type of stability function (3).

1: constant value (4.198)

2: Munk-Anderson form (4.199)

3: from RANS model as explained in Section 4.4.3.3

iopt_turb_stab_lev Selects level for stability functions if iopt_turb_stab_form = 3 (1).

1: quasi-equilibrium method (Section 4.4.3.3)

2: non-equilibrium method (Section 4.4.3.3)

iopt_turb_stab_mod Selects type of closure (RANS) model (4).

1: MY82-model (Mellor & Yamada, 1982)

2: KC94-model (Kantha & Clayson, 1994)

3: BB95-model (Burchard & Baumert, 1995)

4: HR82-model (Hossain & Rodi, 1982)

5: CA01-model (Canuto *et al.*, 2001)

6: CA02-model (Canuto *et al.*, 2001)

iopt_turb_stab_tke Formulation for the turbulent diffusion coefficient ν_k (or stability coefficient S_k) of turbulent energy (2).

1: constant value for S_k as given by equation (4.200)

2: S_k is taken as proportional to momentum stability function S_u as given by (4.201)

3: using the formulation of Daly & Harlow (1970) as given by (4.185) or (4.191) depending on the value of iopt_turb_stab_lev

iopt_turb_tke_bcc Type of bottom boundary condition for turbulence energy (2).

1: Neumann condition (4.352)

2: Dirichlet condition (4.351)

iopt_turb_tke_sbc Type of surface boundary condition for turbulence energy (2).

1: Neumann condition (4.283)

2: Dirichlet condition (4.281)

14.4.10 Drying/wetting scheme

iopt_fld Selects the type of drying/wetting scheme (0).

- 0: Drying/wetting disabled
- 1: Drying/wetting algorithm without dynamic masks
- 2: Drying/wetting algorithm using dynamic masks

14.4.11 Structures

iopt_dischr Disables/enables (0/1) discharge module.

iopt_drycel Disables/enables (0/1) dry cell module.

iopt_thndam Disables/enables (0/1) thin dam module.

iopt_weibar Disables/enables (0/1) weirs/barriers module.

14.4.12 Time integration

iopt_cor_impl Time-integration of the Coriolis term (1).

- 0: explicit
- 1: semi-implicit
- 2: implicit

iopt_vadv_impl Time-integration for vertical advection (1).

- 0: explicit
- 1: semi-implicit
- 2: implicit

iopt_vdif_impl Time-integration for vertical diffusion (2).

- 0: explicit
- 1: semi-implicit
- 2: implicit

14.4.13 Open boundary conditions

iopt_obc_advflux Type of open boundary condition for the cross-stream (2-D and 3-D) advective fluxes (see Section 5.3.16.2)

- 1: zero gradient condition
- 2: quasi-upwind scheme

iopt_obc_advrlx Disables/enables (0/1) the relaxation scheme for horizontal momentum advection (see Section 5.3.16.2)

0: relaxation scheme disabled (default)

1: relaxation scheme enabled. In that case the parameter distrlx_obc (representing the parameter d_{max}) must be defined by the user in usrdef_mod_params or in the CIF.

iopt_obc_bio (General) type of open boundary conditions for biological variables (0). Currently not implemented.

0: default conditions at all open boundaries

1: non-default conditions for at least one open boundary point

iopt_obc_int Disables/enables (0/1) momentum advection adjacent to open boundaries (0).

iopt_obc_invbar Disables/enables (0/1) inverse barometric effect at open boundaries (0).

iopt_obc_relax Disables/enables (0/1) open boundary relaxation as discussed in Section 4.10.3 (0).

iopt_obc_sal (General) type of open boundary conditions for salinity (0).

0: default conditions at all open boundaries

1: non-default conditions for at least one open boundary point

iopt_obc_sed (General) type of open boundary conditions for sediments (0).

0: default conditions at all open boundaries

1: non-default conditions for at least one open boundary point

iopt_obc_temp (General) type of open boundary conditions for temperature (0).

0: default conditions at all open boundaries

1: non-default conditions for at least one open boundary point

iopt_obc_2D (General) type of open boundary conditions for the 2-D mode (0).

0: default conditions at all open boundaries

1: non-default conditions for at least one open boundary point

iopt_obc_3D (General) type of open boundary conditions for the 3-D currents (0).

0: default conditions at all open boundaries

1: non-default conditions for at least one open boundary point

Note that the open boundary conditions automatically reduce to their defaults (see Section 4.10) and input of open boundary data is disabled if the appropriate switch is not set.

14.4.14 Tides

iopt_astro_anal Disables/enables (0/1) the use of astronomical arguments for harmonic analysis if iopt_astro_pars $>$ 0 and iopt_out_anal $=$ 1 (0).

iopt_astro_pars Enbables or disables the inclusion of astronomical arguments and nodal corrections in the harmonic expansions (4.230) and (4.354) (0).

0: astronomical argument set to zero, nodal factors set to 1, nodal phases set to zero

1: evaluate astronomical phases at a given time and reference longitude, nodal factors are set to 1, nodal phases set to zero

2: evaluate astronomical phases and nodal corrections at a given time and reference longitude

iopt_astro_tide Disables/enables (0/1) the inclusion of the astronomical tidal force in the momentum equations (0). This requires that the model uses a spherical grid (iopt_grid_sph=1).

14.4.15 1-D applications

iopt_sur_1D Disables/enables surface forcing (surface slopes and elevations) in case 1-D (iopt_grid_nodim=1) water column applications (0).

14.4.16 Surface forcing

iopt_meteo Disables/enables (0/1) meteorological input and evaluation of all surface fluxes (0).

iopt_meteo_heat Selects type of input data for the heat fluxes (0).

- 0: no input
- 1: air temperature T_a, relative humidity RH, cloud cover f_c
- 2: total (downward) non-solar surface heat flux, cloud cover f_c
- 3: total (downward) non-solar surface heat flux, surface solar radiance Q_{rad}
- 4: cloud cover f_c
- 5: surface solar radiance Q_{rad}

iopt_meteo_salflx Selects type of input data for the salinity flux (0).

- 0: no input
- 1: evaporation minus precipitation rate $E_{vap} - P_{rc}$
- 2: precipitation rate P_{rc}

iopt_meteo_stres Selects type of input data for the barotropic mode, i.e. surface stress and pressure (0).

- 0: no input
- 1: components of wind speed (U_{10},V_{10}) and (unless iopt_grid_nodim=1) atmospheric pressure P_a
- 2: components of surface stress (τ_s^u,τ_s^v) and (unless iopt_grid_nodim=1) atmospheric pressure P_a

iopt_waves Type of wave input wave input (0).

- 0: wave input disabled (default)
- 1: wave height, period and wave direction
- 2: wave height, period, velocity, excursion and direction

Remarks

- Note that all meteorological surface forcing is disabled if iopt_meteo=0. This means that all surface fluxes are automatically set to zero and the input of any meteorological data is disabled.

- A wave-current interaction module is currently not yet implemented. Wave input is only used for the sediment module.

14.4.17 Surface boundary conditions

iopt_sflux_cds Formulation for the neutral surface drag coefficient C_{ds} (0).

- 0: constant value as given by the parameter cds_cst (see below)
- 1: equation (4.286) from Large & Pond (1981)
- 2: equation (4.287) from Smith & Banke (1975)
- 3: equation (4.288) from Geernaert *et al.* (1986)
- 4: equation (4.289) from Kondo (1975)
- 5: equation (4.290) from Wu (1980)
- 6: equation (4.291) from Charnock (1955)

iopt_sflux_cehs Formulation for the neutral surface (heat) exchange coefficients C_e, C_h (0).

- 0: constant value as given by the parameter ces_cst or chs_cst (see below)
- 1: equation (4.292) from Large & Pond (1982)
- 2: equation (4.293) from Anderson & Smith (1981)
- 3: equation (4.294) from Kondo (1975)
- 4: equation (4.295) from Wu (1980)

iopt_sflux_strat Selects dependence of surface drag and exchange coefficients on atmospheric stratification effects (0).

- 0: no dependence
- 1: using the Kondo (1975) parameterisation (Section 4.8.2)
- 2: using Monin-Obukhov similarity theory (Section 4.8.3)

14.4.18 Nesting

iopt_nests Disables/enables (0/1) the writing of open boundary data for nested sub-grids (0).

14.4.19 MPI mode

iopt_MPI_abort
- 0: If an error is detected in a MPI routine, an error message will be written, but the program will not abort immediately.
- 1: If an error is detected in a MPI routine, an error message will be written and the program will abort immediately afterwards.

iopt_MPI_comm_all Communication type for "all to all" operations (2).
- 1: blocking, standard send
- 2: blocking, synchronous send
- 3: non-blocking, standard send
- 4: non-blocking, synchronous send

iopt_MPI_comm_coll Disables/enables (0/1) the use of MPI collective calls (0).

iopt_MPI_comm_exch Communication type for "exchange" operations (2).
- 1: blocking, standard send
- 2: blocking, synchronous send
- 3: non-blocking, standard send
- 4: non-blocking, synchronous send
- 5: send-receive blocking calls

iopt_MPI_comm_gath Communication type for "all to one" gather (combine) operations (2).
- 1: blocking, standard send
- 2: blocking, synchronous send
- 3: non-blocking, standard send
- 4: non-blocking, synchronous send

iopt_MPI_comm_scat Communication type for "one to all" scatter (distribute and copy) operations (2).
- 1: blocking, standard send
- 2: blocking, synchronous send
- 3: non-blocking, standard send
- 4: non-blocking, synchronous send

iopt_MPI_partit Selects the method for domain decomposition (1).

1: "simple" partition based on the values of nprocsx and nprocsy

2: decomposition obtained from an external data file or defined in usrdef_partition

iopt_MPI_sync Disables/enables (0/1) synchronisation calls at the end of a series of blocking or non-blocking operations (0).

Remarks

- The non-blocking options are not yet tested and should not be used in the current version of COHERENS.
- Synchronisation of communication calls may lower the CPU performance.

14.4.20 PETSc

iopt_petsc_precond Type of preconditioner used by PETSc (5). For details, see the PETSc User Manual.

1 : Jacobi (PCJACOBI)

2 : Block Jacobi (PCBJACOBI)

3 : SOR (and SSOR) (PCSOR)

4 : SOR with Eisenstat trick (PCEISENSTAT)

5 : Incomplete Cholesky (PCICC)

6 : Incomplete LU (PCILU)

7 : Additive Schwarz (PCASM)

8 : Linear solver (PCKSP)

9 : Combination of preconditioners (PCCOMPOSITE)

10: LU (PCLU)

11: Cholesky (PCCHOLESKY)

12: No preconditioning (PCNONE)

iopt_petsc_solver Type of solver used by PETSc (5). For details, see the PETSc User Manual.

1 : Richardson (KSPRICHARDSON)

2 : Chebychev (KSPCHEBYCHEV)

3 : Conjugate Gradient (KSPCG)
4 : Biconjugate Gradient (KSPBICG)
5 : Generalised Minimal Residual (KSPGMRES)
6 : BiCGSTAB (KSPBCGS)
7 : Conjugate Gradient Squared (KSPCGS)
8 : Transpose-Free Quasi-Minimal Residual (1) (KSPTFQMR)
9 : Transpose-Free Quasi-Minimal Residual (2) (KSPTCQMR)
10: Conjugate Residual (KSPCR)
11: Least Squares Method (KSPLSQR)
12: Shell for no KSP method (KSPPREONLY)

14.4.21 User output

iopt_out_anal Disables/enables (0/1) harmonic output (0).

iopt_out_avrgd Disables/enables (0/1) time averaged output (0).

iopt_out_tsers Disables/enables (0/1) time series output (1).

iopt_out_tsers Disables/enables (0/1) time series output (1).

14.4.22 NetCDF

iopt_CDF_abort 0: If an error is detected in a netCDF routine, an error message will be written, but the program will not abort immediately.

1: If an error is detected in a netCDF routine, an error message will be written and the program will abort immediately afterwards.

iopt_CDF_fill Disables/enables (0/1) the use of fill values (0).

iopt_CDF_format Selects the type netCDF file format (1).

1: classic format

2: 64-bit offset format

The different netCDF file formats are discussed in the netCDF User Manual.

14.5 Model parameters

All parameters in this section are defined in usrdef_mod_params.

14.5.1 Date and time parameters

CStartDateTime	Start date in string format ('yyyy/mm/dd;hh:mm:ss:mmm') of 23 characters. If the last 4 characters are omitted they are set to ':000' by default (?).
CEndDateTime	End date in string format. If the last 4 characters are omitted they are set to ':000' by default (?).
delt2d	Barotropic (2-D) time step (mode-splitting scheme) or time step for all 2-D/3-D transport equations (implicit scheme) [s] (?).
ic3d	number of 2-D time steps within one 3-D time step (1). If iopt_hydro_impl=1 or iopt_grid_nodim=1 or 2, ic3d is always 1.
icnodal	Time step (measured in units of delt2d) for an update of the nodal tidal factors and astronomical arguments if iopt_astro_pars >0. If zero, nodal corrections (amplitudes and phases) are evaluated at the initial time only (0).
time_zone	Time zone, i.e. the difference of the local time with respect to GMT [hours]. Difference is positive (negative) eastwards (westwards) from Greenwich (0).

Remarks

- If the 2-D time step is lower than 1000 seconds, its precision is 1 millisecond and decimal numbers from the fourth position after the decimal point will be discarded. If the time step is larger than 1000 seconds, its precision is 1 second and its decimal part is ignored. If iopt_hydro_impl=0, the 2-D time step is limited by the CFL condition (5.4) for surface gravity waves. The maximum allowed 2-D time step is written to the "log" file.

- The parameter time_ zone is of type REAL and must be between -12.0 and 12.0 and is used to reset the start and end dates to GMT where necessary. A time zone must be given for the calculation of solar radiance and the astronomical Greenwich argument at the local time when the start and end dates are not expressed in GMT.

- It is clear that ic3d only needs to be defined for 3-D applications (iopt_grid_nodim=3). Note that the 3-D time step is limited by the constraints (5.5), (5.6).

14.5.2 Grid parameters

nc number of grid cells in the X-direction (including an extra column along the eastern edge) (?)

nr number of grid cells in the Y-direction (including an extra column along the northern edge) (?)

nz number of grid cells in the vertical direction (?)

nosbu number of open sea boundaries at (West/East) U-nodes (0)

nosbv number of open sea boundaries at (South/North) V-nodes (0)

nrvbu number of river boundaries at (West/East) U-nodes (0)

nrvbv number of river boundaries at (South/North) V-nodes (0)

- nc and nr must be positive and are automatically (re)set to 3 for water column applications (iopt_grid_nodim=1).
- nz must be positive and is automatically (re)set to 1 for 2-D applications (iopt_grid_nodim=2).
- The last row and the last column of the computational domain represent dummy (land) points. The "phyical" (horizontal) dimension of the domain is therefore (nc-1)×(nr-1).

14.5.3 Other integer model parameters

fld_mask(nofldmasks) Enables (1) or disables (0) a specific mask criterium for the drying/flooding algorithm as given by equations (5.372)–(5.383). Default values are fld_mask(1)=1,fld_mask(2:)=0. The number of available criteria given by nofldmask equals 11 in the current implementation and cannot be changed by the user.

maxitsimp Largest allowed iteration number for the outer loop in the free surface corrector method (implicit scheme) (1)

nconastro number of constituents for the astronomical tidal forcing used when iopt_astro_tide=1 (0)

nconobc number of constituents for the open boundary tidal forcing if iopt_grid_nodim>1 or for the surface forcing if iopt_grid_nodim=1 (0)

norlxzones number of relaxation zones used when iopt_obc_relax = 1 (0)

nonestsets number of nested sub-grids used when iopt_nests = 1 (0)

numdis number of discharge locations

numdry number of dry cells

numthinu number of thin dams at U-nodes

numthinv number of thin dams at V-nodes

numwbaru number of weirs/barriers at U-nodes

numwbarv number of weirs/barriers at V-nodes

nowaitsecs number of seconds to wait between two read attempts [s] (0)

maxwaitsecs maximum allowed time spent in wait calls [s] (3600)

norestarts number of restart times (1)

ntrestart(1:norestarts) Restart time indices for writing of initial conditions. If a value equals int_fill, it will be replaced by the total number of 2-D time steps (int_fill).

ntobcrlx The relaxation period T_r, divided by the 2-D time step delt2d, (optionally) used to define the relaxation factor $\alpha_r(t)$, defined by (4.356), for the 2-D mode at open boundaries. For details see Section 4.10.1 (0).

idmaster Process id of the master process (0). Must be between 0 and nprocs-1.

index_obc(1:nconobc) Key ids of the tidal constituents used for the tidal forcing at open boundaries (?).

index_astro(1:nconastro) Key ids of the tidal constituents for the astronomical tidal forcing (?).

- nowaitsecs and maxwaitsecs are used in connection to the endfile attribute discussed in Section 14.7.2.
- norestarts must not exceed the value of the system parameter MaxRestarts defined in *syspars.f90*.
- Key ids for tidal constituents are defined in *tide.f90*.

14.5.4 Physical model parameters

The defaults of parameters marked with a "*" can be generally applied and should, in priciple, not be changed.

atmpres_ref* Reference atmospheric pressure P_{ref} [Pa] (101325.0)

bdragcoef_cst Constant bottom drag coefficient C_{db} when iopt_bstres_drag=1 [-] (0.0).

bdraglin	Bottom friction velocity k_{lin} used in the linear bottom friction law if iopt_bstres_form=1 [m/s] (0.0).
b_SH	Parameter b in the Song & Haidvogel (1994) vertical grid transformation (0.1)
ccharno*	Charnock's constant a used in Charnock's relation (4.291) [-] (0.014).
cds_cst	Constant surface drag coefficient C_{ds} when iopt_sflux_cds=0 [-] (0.0013).
ces_cst	Constant surface exchange coefficient C_e when iopt_sflux_cehs=0 [-] (0.0013).
chs_cst	Constant surface exchange coefficient C_h when iopt_sflux_cehs=0 [-] (0.0013).
ckar*	von Karman's constant κ [-] (0.4).
dcrit_fld	Critical water depth d_{crit} used in the drying/wetting algorithm [m] (0.1).
depmean_cst	Constant water depth used to set up a default bathymetry [m] (0.0).
depmean_flag	Data flag marking land points in the bathymetry [m] (0.0).
distrlx_obc	Maximum distance d_{max} (from the open boundaries) used in the relaxation factor (5.292) for momentum advection
dlat_ref	Reference latitude to be used for the Coriolis frequency in the case of a Cartesian grid [decimal degrees] (0.0).
dlon_ref	Reference longitude to be used for solar irradiance in the case of a Cartesian grid [decimal degrees] (0.0).
dlon_ref_anal	If iopt_astro_pars>0, harmonically analysed phases are taken with respect to the astronomical argument for this reference longitude at the central time [decimal degrees, positive East] (0.0).
dlon_ref_obc	If iopt_astro_pars > 0, phases at open boundaries are assumed to be taken with respect to the astronomical argument at this reference value [decimal degrees]. If zero, the reference longitude is taken at Greenwich (0.0).
dl_BB	Parameter d_l in the Burchard & Bolding (2002) vertical grid transformation (4.26) (1.5)
dmin_fld	Minimum water depth d_{min} used in the drying/wetting algorithm [m] (0.02).

dthd_fld	Threshold water depth d_{th} used in the mask criteria for drying and flooding (see Section 5.4.2) [m] (0.1).
du_BB	Parameter d_u in the Burchard & Bolding (2002) vertical grid transformation (4.26) (1.5)
dzetaresid_conv	Threshold value ϵ_{imp} used in the convergence criterium for the outer loop in the implicit scheme (free surface corrector method) (10^{-14})
gacc_ref	If different from real_fill, the acceleration of gravity, taken as horizontally uniform. Otherwise, g is evaluated as function of latitude using (4.58) [m/s^2] (real_fill).
hcrit_SH	Parameter h_{crit} in the Song & Haidvogel (1994) vertical grid transformation (0.1)
hdifmom_cst	Constant coefficient for horizontal momentum diffusion ν_H when iopt_hdif_coef=1 [m^2/s] (0.0).
hdifscal_cst	Constant coefficient for horizontal scalar diffusion λ_H when iopt_hdif_coef=1 [m^2/s] (0.0).
kinvisc_cst	Constant value for the kinematic viscosity [m/s^2] (10^{-6}).
optattcoef1_cst	Inverse optical attenuation depth (λ_1^{-1}) for the absorption of long-wave solar radiation as used in (4.59)[m^{-1}] (10.0).
optattcoef2_cst	Inverse optical attenuation depth (λ_2^{-1}) for the absorption of short-wave solar radiation as used in (4.59) [m^{-1}] (0.067).
opt_frac	Long-wave fraction R of surface solar radiance as used in (4.59) [-] (0.54).
petsc_tol	Relative tolerance used by PETSc for solving the linear system. (The parameters atol, dtol, maxits used by PETSc in the solution procedure are set to the PETSc defaults) (10^{-7}).
Rearth*	Mean radius of the Earth R [m] (6371000.0)
rho_air*	Air mass density ρ_a [kg/m^3] (1.2)
sal_ref	Reference salinity S_{ref} used if iopt_sal=0 or in the linear equation of state (4.108) or as default initial condition [PSU] (33.0).
sigstar_DJ	Parameter σ_* in the Davies & Jones (1991) vertical grid transformations (4.23) and (4.24) (0.0)
sig0_DJ	Parameter σ_0 in the Davies & Jones (1991) vertical grid transformations (4.23) and (4.24) (0.1)

smag_coef_mom*	Smagorinsky coefficient C_m for horizontal diffusion of momentum [-] (0.1).
smag_coef_scal*	Smagorinsky coefficient C_s for horizontal diffusion of scalars [-] (0.1).
specheat*	Specific heat of seawater c_p at constant pressure [J/kg/degC] (3987.5).
temp_min	Minimum temperature. If set to real_fill, the minimum is taken as the freezing point of sea water (see equation (4.49) which is a function of salinity [deg C] (0.0).
temp_ref	Reference temperature T_{ref} used if iopt_temp=0 or in the linear equation of state (4.108) or as default initial condition [deg C] (12.0).
theta_cor*	Implicity factor θ_c for the Coriolis term [between 0.0 and 1.0] (0.5).
theta_SH	Parameter θ in the Song & Haidvogel (1994) vertical grid transformation (8.0)
theta_vadv*	Implicity factor θ_a for vertical advection [between 0.0 and 1.0] (0.501).
theta_vdif*	Implicity factor θ_d for vertical diffusion [between 0.0 and 1.0] (1.0).
vdifmom_cst	Constant coefficient for vertical diffusion of momentum used if iopt_vdif_coef=1 or as background value if iopt_turb_iwlim=0 [m^2/s] (10^{-6}).
vdifscal_cst	Constant coefficient for vertical diffusion of scalars used if iopt_vdif_coef=1 or as background value if iopt_turb_iwlim=0 [m^2/s] (10^{-6}).
wbarrlxu	Time relaxation coefficient at U-node weirs/barriers (1.0)
wbarrlxv	Time relaxation coefficient at V-node weirs/barriers (1.0)
zbzoz0lim	Value of the limiting ratio ξ_{min} for z_b/z_0 [-] (2.0).
zrough_cst	Constant bottom roughness length z_0 when iopt_bstres_drag=3 [m] (0.0).

14.5.5 Turbulence model parameters

Parameters marked with a * have been calibrated from experimental data or obtained from turbulence theory. Their values should not be changed, unless the user has sufficient experience in turbulence modelling.

alpha_Black	constant α_1 in the Blackadar (1962) mixing length formulation (4.217) [0.2]
alpha_ma	parameter α_m in the Munk & Anderson (1948) scheme (4.136)–(4.139) [10.0]
alpha_pp	parameter α_p in the Pacanowski & Philander (1981) scheme (4.132)–(4.134) [5.0]
beta_ma	parameter β_m in the Munk & Anderson (1948) scheme (4.136)–(4.139) [3.33]
beta_Xing	attenuation factor β_1 in the Xing & Davies (1996) mixing length formulation (4.215) [2.0]
cnu_ad	parameter C_ν in equation (4.151) [2.0]
c1_eps*	constant $c_{1\varepsilon}$ in the shear production term of the ε-equation (4.205) [1.44]
c2_eps*	constant $c_{2\varepsilon}$ in the dissipation term of the ε-equation (4.205) [1.92]
c31_eps*	constant $c_{3\varepsilon}$ in the buoyancy sink term of the ε-equation (4.205) in case of stable stratification ($N^2 > 0$) [0.2]
c32_eps*	constant $c_{3\varepsilon}$ in the buoyancy source term of the ε-equation (4.205) in case of unstable stratification ($N^2 < 0$) [1.0]
c_sk*	Daly-Harlow parameter c_{sk} in (4.177) [0.15]
delta1_ad	parameter δ_1 in equation (4.143) [0.0]
delta2_ad	parameter δ_2 in equation (4.143) [0.0]
dissipmin*	numerical lower limit ε_{min} for ε [10^{-12} W/kg]
expmom_ma	parameter n_1 in the Munk & Anderson (1948) scheme (4.136)–(4.139) [0.5]
expmom_pp	parameter n_p in the Pacanowski & Philander (1981) scheme (4.132)–(4.134) [2.0]
expscal_ma	parameter n_2 in the Munk & Anderson (1948) scheme (4.136)–(4.139) [1.5]
e1_my*	constant E_1 in the shear production term of the kl-equation (4.209) [1.8]
e2_my*	constant E_2 in the wall proximity term (4.210) of the kl-equation (4.209) [1.33]
e3_my*	constant E_3 in the buocancy source/sink term of the kl-equation (4.209) [1.0]

k1_ad	parameter K_1 in equations (4.148) and (4.150) [0.0025]
k2_ad	parameter K_2 in equation (4.149) [2×10^{-5}]
lambda_ad	parameter $\lambda_\star$ in equation (4.146) [0.0 m]
omega1_ad	parameter ω_1 in equation (4.151) [10^{-4} s^{-1}]
riccrit_iw	critical Richardson number Ri_0 in the Large *et al.* (1994) background mixing scheme (4.227) [0.7]
r1_ad	parameter r_1 in equation (4.143) [1.0]
r2_ad	parameter r_2 in equation (4.143) [1.0]
sigma_k*	parameter σ_k used to define S_k in (4.201) [1.0]
skeps*	neutral value S_{k0} of the stability coefficient S_k in the k-ε model (see equation (4.200)) [0.09]
sq_my*	parameter S_q used to determine S_{k0} in the Mellor-Yamada model (see equation (4.202)) [0.2]
tkelim*	background limit k_{lim} for k (see equation (4.226)) [10^{-6} J/kg]
tkemin*	numerical lower limit k_{min} for k [10^{-14} J/kg]
vbmom_pp	parameter ν_{bp} in the Pacanowski & Philander (1981) scheme (4.132)–(4.134) [10^{-4} m^2/s]
vbscal_pp	parameter λ_{bp} in the Pacanowski & Philander (1981) scheme (4.132)–(4.134) [10^{-5} m^2/s]
vdifmom_iw	internal wave breaking diffusion coefficient ν_{T0} for momentum in the Large *et al.* (1994) background mixing scheme (4.227) [10^{-4} m^2/s]
vdifscal_iw	internal wave breaking diffusion coefficient λ_{T0} for scalars in the Large *et al.* (1994) background mixing scheme (4.227) [5×10^{-5} m^2/s]
vdifshear_iw	maximum mixing due to unresolved vertical shear ν_0^s in the Large *et al.* (1994) background mixing scheme (4.227) [0.005 m^2/s]
vmaxmom_ma	parameter ν_{max} in the Munk & Anderson (1948) scheme (4.136)–(4.139) [3.0]
vmaxscal_ma	parameter λ_{max} in the Munk & Anderson (1948) scheme (4.136)–(4.139) [4.0]
vmax_pp	parameter ν_{max} in the Pacanowski & Philander (1981) scheme (4.132)–(4.134) [3.0]
v0dif_ma	parameter ν_{0m} in the Munk & Anderson (1948) scheme (4.136)–(4.139) [0.06 m^2/s]

v0dif_pp	parameter ν_{0p} in the Pacanowski & Philander (1981) scheme (4.132)–(4.134) [0.01 m^2/s]
wfltke	surface wave factor c_w used in the surface flux condition (4.283) for turbulent energy [0.0]
zlmixmin*	numerical lower limit l_{min} for l [1.7×10^{-10} m]
zrough_bot	bottom roughness length z_{0b} in the mixing length formulation (4.212) [0.0 m]
zrough_sur	surface roughness length z_{0s} in the mixing length formulation (4.212) [0.0 m]

14.6 Parameters for surface data grids

Surface data grids are external grids where (e.g. meteorological) data are defined for the surface forcing. The parameters characterising a surface grid are stored into the 2-D array surfacegrids of DERIVED TYPE GridParams, defined by

```
TYPE :: GridParams
   LOGICAL :: rotated
   INTEGER :: nhtype, n1dat, n2dat
   REAL :: delxdat, delydat, gridangle, x0dat, y0dat, y0rot
END TYPE GridParams
TYPE (GridParams), DIMENSION(MaxGridTypes,2) :: surfacegrids
```

An element of the array surfacegrids can be generically represented as surfacegrids(igrd,ifil) where igrd is a key id, called the "grid descriptor" and ifil the "file number". The file number can take the value of 1 for external data intended for input and 2 for data written by the model to the external grid. The latter is intended for future applications and currently not implemented.

All parameters of this section are defined in usrfdef_mod_params.

14.6.1 Grid descriptors

The grid descriptor may take (in the current version) the following values:

igrd_model	model grid
igrd_meteo	meteorological external grid
igrd_sst	sea surface temperature external grid
igrd_waves	surface wave external grid

Identifying the model grid as an external grid seems rather strange at first sight. The intention is to provide the possibility to define a uniform rectangular grid with the parameters below.

14.6.2 Grid parameters

In the case of an external (meteorological, surface temperature, surface wave) grid, the following attributes must or may be defined

nhtype Type of the surface data grid.

- 0: single grid point
- 1: uniform rectangular grid
- 2: non-uniform rectangular grid
- 3: non-rectangular (curvilinear or non-structured)
- 4: the same as the model grid

n1dat X-dimension of the surface grid

n2dat Y-dimension of the surface grid

delxdat grid spacing in the X-direction (meters or degrees longitude) when nhtype=1

delydat grid spacing in the Y-direction (meters or degrees latitude) when nhtype=1

x0dat X-coordinate (meters or degrees longitude) of the lower left corner when nhtype=1

y0dat Y-coordinate (meters or degrees latitude) of the lower left corner when nhtype=1

Remarks

- If nhtype=1, all parameters need to be defined.
- If nhtype=2,3, only n1dat and n2dat need to be defined.
- If nhtype=4, then n1dat=nc and n2dat=nr, and no further definitions need to be made.
- The corner coordinates x0dat, y0dat and the grid spacings delxdat, delydat are given in meters or degrees longitude and latitude depending on whether iopt_grid_sph equals 0 or 1.

In the case of a model grid (grid descriptor igrd_model), the attribute nhtype equals the value of iopt_grid_htype (between 1 and 3) and n1dat, n2dat are given by the previously defined grid sizes nc, nr.

delxdat grid spacing in the X-direction (meters or degrees longitude) when iopt_grid_htype=1. In the case of a rotated grid the spacing is given in transformed coordinates.

delydat grid spacing in the Y-direction (meters or degrees latitude) when iopt_grid_htype=1. In the case of a rotated grid the spacing is given in transformed coordinates.

x0dat X-coordinate (meters or degrees longitude) of the reference location x_r or λ_r

y0dat Y-coordinate (meters or degrees longitude) of the reference location y_r or ϕ_r

rotated must be set to .TRUE. in case of a rotated grid. Default is .FALSE..

gridangle grid rotation angle α (see Section 4.1.3) (decimal degrees). Must be between 0 and 180^0.

y0rot transformed latitude of the reference location in case of a rotated grid (decimal degrees). Only used for spherical (rotated) grids.

Remarks

- delxdat and delydat are only needed for uniform rectangular grids.
- The meaning of the reference location for a rectangular grid is explained in Section 4.1.2.
- In case of a fully curvilinear grid, the model grid is defined in routine usrdef_grid. The only attributes which may be supplied are the coordinates of the reference location used for avoiding rounding errors. For details see Section 15.1.
- Grid rotation is only allowed for rectangular grids.

14.7 Attributes of forcing files

Model forcing requires the definition of pamameters and input data. They can be directly defined by the user in one of the usrdef_routines below or by reading them for some external file. Before these data can be obtained, a series of "file attributes" needs to be set by the user to inform the program

which parameters/data are needed and how they are accessed. These attributes are stored in the 3-D array modfiles of DERIVED TYPE FileParams, defined by

```
TYPE :: FileParams
   LOGICAL :: defined, info, opened, time_regular
   CHARACTER (LEN=1) :: form, status
   CHARACTER (LEN=leniofile) :: filename, pathname
   CHARACTER (LEN=lendesc) :: filedesc
   INTEGER :: endfile, header_type, iostat, iunit,lenrec, &
            &  maxrecs, nocoords, nodim, novars,timeid, &
            &  timerec, tskips, varid, zetaid
   INTEGER, DIMENSION(3) :: tlims
END TYPE FileParams
TYPE (FileParams), DIMENSION(MaxIOTypes,MaxIOFiles,2) :: &
                                                & modfiles
```

Only the underlined parameters can be defined by the user, the others are used internally in the program (e.g. iunit giving the FORTRAN file unit number).

An element of the array modfiles can be generically represented as modfiles(idesc,ifil,iotype) where idesc is the "file descriptor", ifil the "file number" and iotype represents input (output) data if 1 (2).

The meaning of the third index iotype is as follows. Almost all forcing data (except nesting) are input data, i.e. represented by an element of modfiles with iotype=1. By defining a corresponding output file with iotype=2 one has the possibility to re-write the same input data now in a COHERENS standard format. This file can be used as input within a subsequent run. The user then needs to change only the status atrtribute from 'N' to 'R' (see below). In case of nested output, iotype must take the value of 2.

Input data can be spread over multiple files for a given descriptor by specifying different file numbers. This is further discussed below. The maximum value of ifil is given by the system parameter MaxIOFiles defined in *syspars.f90*.

14.7.1 File descriptors

The following key ids are available as file descriptors.

io_mppmod parallel decomposition (ifil=1)

io_inicon initial conditions for the physical (ifil=ics_phys) and sediment model (ifil=ics_sed)

io_modgrd	model grid (ifil=1)
io_metgrd	surface meteorological grid (ifil=1)
io_sstgrd	sea surface temperature grid (ifil=1)
io_wavgrd	surface waves grid (ifil=1)
io_nstgrd	nested sub-grids (one file per sub-grid)
io_sedspc	specific arrays for the sediment module (see Section 19.1.2.3)
io_1uvsur	specifiers for 1-D surface forcing if ifil=1, forcing data if ifil=2
io_2uvobc	specifiers for 2-D mode open boundary forcing if ifil=1, open boundary data if ifil >1
io_3uvobc	specifiers for 3-D mode (baroclinic currents) open boundary forcing if ifil=1, open boundary data if ifil>1
io_salobc	specifiers for salinity open boundary forcing if ifil=1, open boundary data if ifil>1
io_tmpobc	specifiers for temperature open boundary forcing if ifil=1, open boundary data if ifil >1
io_sedobc	specifiers for sediment open boundary forcing if ifil=1, open boundary data if ifil >1
io_rlxobc	definitions of relaxation zones (ifil=1)
io_nstspc	specifiers for sub-grid nesting (ifil=1)
io_2uvnst	2-D open boundary data for nested sub-grids (one file per sub-grid)
io_3uvnst	3-D (baroclinic current) open boundary data for nested sub-grids (one file per sub-grid)
io_salnst	salinity open boundary data for nested sub-grids (one file per sub-grid)
io_tmpnst	temperature open boundary data for nested sub-grids (one file per sub-grid)
io_sednst	sediment open boundary data for nested sub-grids (one file per sub-grid)
io_metsur	meteorological data (ifil=1)
io_sstsur	SST data (ifil=1)
io_wavsur	wave data (ifil=1)
io_drycel	dry cell locations
io_thndam	thin dam locations

io_weibar	weirs/barriers locations and parameters
io_disspc	discharge specifiers
io_disloc	discharge locations
io_disvol	volume discharges
io_discur	momentum discharges
io_dissal	salinity discharge
io_distmp	temperature discharges

14.7.2 File parameters for input forcing (iotype=1)

status Status of the data file ('0').

'0' (zero): not defined

'N': user-defined

'R': COHERENS standard file

form File format.

'A': ASCII (portable, sequential)

'U': unformatted binary (non-portable, sequential)

'N': netCDF format (portable, non-sequential)

filename File name (including file path if needed).

tlims Start/end/step time indices (i.e. times measured in units of delt2d). These parameters are not directly used for reading the data, but to make updates after tlims(3)×delt2d seconds. If tlims(3)>0, time interpolation will be performed (see below).

info An "info" file with all header information will be created if .TRUE. (.FALSE.).

endfile Switch to decide what action needs to be taken when an end of file conditions occurs (0).

0: The program aborts with an error message

1: The program continues, no further attempt will be made to read data.

2: The program continues, a next attempt to read the data will be made after nowaitsecs seconds.

- Important to note that the status attribute equals '0' by default which means that the corresponding usrdef_ routine is not called by the program.

- The meaning of tlims is illustrated as follows for the case of meteorological forcing data. These data are used to evaluate the surface fluxes of momentum, heat and salinity and for the atmospheric pressure gradient in the momentum equations. All these quantities will be updated from time tlims(1) upto time tlims(2) at time intervals given by tlims(3). The data are read into the program with a date/time stamp which is saved. If tlims(3)>0, which is usally shorter than the time interval between two input dates, the meteo data are first linearly interpolated in time between their values at the most recent date, earlier than the current program time, and the earliest date, later than the current time. Since these dates are stored in memory, the program knows automatically when new data need to be read. If tlims(3)<0, the method is the same but without time interpolation, i.e. the data at the current program time are set to their values at the most recent date earlier than or equal to the program time. Although it is not absolutely necessary, it is recommended that tlims(3) is smaller than the time interval between two consecutive inputs. Note that if an element of the vector tlims is set to the undefined value int_fill, this value will be automatically replaced by the total number of 2-D time steps in the simulation nstep, which means that the corresponding time is set to the end date of the run.

- If endfile equals 2 and an "end of file condition" occurs during a read, the program waits for nowaitsecs seconds before make a next attempt. The total waiting time is given by maxwaitsecs after which the program aborts with an error message. The procedure is intended for making simulations in interactive mode. For example, assume that a main grid writes the open boundary data for a nested sub-grid. If the main and sub-grid are launched together and the former runs slower than the latter, the nested grid will wait for input from the main grid.

14.7.3 File parameters for output forcing (iotype=2)

status Status of the data file ('0').

'0': not defined

'W': a COHERENS standard file will be created

form File format.

'A': ASCII (sequential)

'U': unformatted binary (machine-dependent, sequential)

'N': netCDF format (portable, non-sequential)

filename File name (including file path if needed).

14.7.4 Other forcing attributes

Other relevant parameter components, not defined in usrdef_mod_params but used internally, are:

iunit File unit. This parameter is set internally and cannot be defined by the user.

iostat File I/O status

- -1: open error occurred
- 0 : file not opened
- 1 : file is open and file pointer is located at the start or before the end of the file
- 2 : file pointer is located at the end of the file (i.e. an EOF condition will occur on a next read)
- 3 : an end of file condition did occur

14.8 Parameters for user-defined output

A few general parameters need to be specified in usrdef_mod_params for user-defined output. They need to be defined in usrdef_mod_params. All other specifiers for user-defined output are to be defined in other *Usrdef_* files. For more details about the meaning of the parameters below, see Section 9.6.

nosetstsr number of time series file sets if iopt_out_tsers=1 (0)

nostatstsr number of time series output stations if iopt_out_tsers=1 (0)

novarstsr number of time series variables if iopt_out_tsers=1 (0)

nosetsavr number of time averaged file sets if iopt_out_avrgd=1 (0)

nostatsavr number of time averaged output stations if iopt_out_avrgd=1 (0)

novarsavr number of time averaged variables if iopt_out_avrgd=1 (0)

nosetsanal number of harmonic file sets if iopt_out_anal=1 (0)

nofreqsanal number of harmonic frequencies if iopt_out_anal=1 (0)

nostatsanal number of harmonic output stations if iopt_out_anal=1 (0)

novarsanal number of harmonic variables if iopt_out_anal=1 (0)

intitle title used to create names of model forcing files

outtitle title used to create names of user output files

14.9 Domain decomposition

The domain decomposition is defined in usrdef_partition. This routine is called in parallel mode by reader processes if iopt_MPI_partit=2 and modfiles(io_mppmod,1,1)%status='N'.

nc1procs(nprocs)	global X-index of lower/upper left cell of the process domains
nc2procs(nprocs)	global X-index of lower/upper right cell of the process domains
nr1procs(nprocs)	global Y-index of lower/upper left cell of the process domains
nr2procs(nprocs)	global Y-index of lower/upper right cell of the process domains