Chapter 33

Description of program variables

33.1 Currents

```
MODULE currents
REAL, DIMENSION(1-nhalo:ncloc+nhalo,1-nhalo:nrloc+nhalo) :: udvel, uvdvel_old,&
                                                          & vdvel, vdvel_old
REAL, DIMENSION(1-nhalo:ncloc+nhalo,0:nrloc+1) :: umvel, umvel_old
REAL, DIMENSION(0:ncloc+1,1-nhalo:nrloc+nhalo) :: vmvel, vmvel_old
REAL, DIMENSION(0:ncloc+1,nrloc) :: umpred
REAL, DIMENSION(ncloc,0:nrloc+1) :: vmpred
REAL, DIMENSION(ncloc+1,nrloc) :: udfvel
REAL, DIMENSION(ncloc,nrloc+1) :: vdfvel
REAL, DIMENSION(ncloc,nrloc) :: p2dbcgradatu, p2dbcgradatv, udevint, vdevint
REAL, DIMENSION(1-nhalo:ncloc+nhalo,1-nhalo:nrloc+nhalo,nz) :: uvel, uvel_old,&
                                                             & vvel, vvel_old
REAL, DIMENSION(2-nhalo:ncloc+nhalo,nrloc,nz) :: ufvel
REAL, DIMENSION(ncloc,2-nhalo:nrloc+nhalo,nz) :: vfvel
REAL, DIMENSION(0:ncloc,0:nrloc,nz+1) :: wvel
REAL, DIMENSION(ncloc,nrloc,nz) :: p3dbcgradatu, p3dbcgradatv, wphys
```

File
: *currents.f90*

Type
: Module

Purpose
: Velocity arrays

Description

p2dbcgradatu	X-component of the depth-integrated baroclinic pressure gradient $\overline{F_1^b}$	[m^2/s^2]
p2dbcgradatv	Y-component of the depth-integrated baroclinic pressure gradient $\overline{F_2^b}$	[m^2/s^2]
p3dbcgradatu	X-component of the baroclinic pressure gradient F_1^b	[m/s^2]
p3dbcgradatv	Y-component of the baroclinic pressure gradient F_2^b	[m/s^2]
udevint	Depth-integrated baroclinic terms $\overline{\delta D_{h1}} - \overline{\delta A_{h1}}$ in the U-equation	[m^2/s^2]
udfvel	X-component of the filtered depth-integrated current U_f	[m^2/s]
udvel	X-component of the depth-integrated current U	[m^2/s]
udvel_old	X-component of the depth-integrated current U at the previous 2-D time step	[m^2/s]
ufvel	X-component of the filtered 3-D current u_f	[m/s]
umpred	X-component of the depth-mean current $\overline{u}^p$ at the predictor step	[m/s]
umvel	X-component of the depth-mean current $\overline{u}$	[m/s]
umvel_old	X-component of the depth-mean current $\overline{u}$ at the old barotropic (mode-splitting scheme) or baroclinic (implicit scheme) time	[m/s]
uvel	X-component of the 3-D current u	[m/s]
uvel_old	X-component of the 3-D current u at the previous corrector (baroclinic) time step	[m/s]
vdevint	Depth-integrated baroclinic terms $\overline{\delta D_{h2}} - \overline{\delta A_{h2}}$ in the V-equation	[m^2/s^2]
vdfvel	Y-component of the filtered depth-integrated current V_f	[m^2/s]
vdvel	Y-component of the depth-integrated current V	[m^2/s]
vdvel_old	Y-component of the depth-integrated current V at the previous 2-D time step	[m^2/s]
vfvel	Y-component of the filtered 3-D current v_f	[m/s]
vmpred	Y-component of the depth-mean current $\overline{v}^p$ at the predictor step	[m/s]
vmvel	Y-component of the depth-mean current $\overline{v}$	[m/s]

vmvel_old	Y-component of the depth-mean current $\overline{v}$ at the old barotropic (mode-splitting scheme) or baroclinic (implicit scheme) time	[m/s]
vvel	Y-component of the 3-D current v	[m/s]
vvel_old	Y-component of the 3-D current v at the previous corrector (baroclinic) time step	[m/s]
wphys	Physical vertical velocity w	[m/s]
wvel	Transformed vertical velocity ω	[m/s]

33.2 Derived type definitions

```
MODULE datatypes
```

ExchComms

```
TYPE :: ExchComms
   LOGICAL :: sfirst
   INTEGER :: iddest, idsrce, tag
   INTEGER, DIMENSION(2) :: i1rcv, i2rcv, j1rcv, j2rcv
   INTEGER, DIMENSION(2) :: i1snd, i2snd, j1snd, j2snd
END TYPE ExchComms
```

File

datatypes.f90

Type

Derived type definition

Purpose

Parameters for exchange (send and receive) communications in parallel mode

Description

iddest	Destination process in case of a send operation
idsrce	Source process in case of a receive operation
i1rcv	Start (local) grid index in the X-direction for the receive operation
i1snd	Start (local) grid index in the X-direction for the send operation
i2rcv	End (local) grid index in the X-direction for the receive operation

i2snd	End (local) grid index in the X-direction for the send operation
j1rcv	Start (local) grid index in the Y-direction for the receive operation
j1snd	Start (local) grid index in the Y-direction for the send operation
j2rcv	End (local) grid index in the Y-direction for the receive operation
j2snd	End (local) grid index in the Y-direction for the send operation
sfirst	Switch which determines for a local domain whether a send operation is performed before a receive operation (see Section 11.4.3.2)
tag	Communication tag

FileParams

```
(MODULE datatypes)
TYPE :: FileParams
   LOGICAL :: defined, info, opened, time_regular
   CHARACTER (LEN=1) :: form, status
   CHARACTER (LEN=10) :: coherens_version
   CHARACTER (LEN=leniofile) :: filename, pathname
   CHARACTER (LEN=lendesc) :: filedesc
   CHARACTER (LEN=lentime-4) :: creation_date
   INTEGER :: endfile, header_type, iostat, iunit, lenrec, maxrecs, &
            & nocoords, nodim, novars, timeid, timerec, tskips, &
            & varid, zetaid
   INTEGER, DIMENSION(3) :: tlims
END TYPE FileParams
```

File

datatypes.f90

Type

Derived type definition

Purpose

Global file attributes

Description

coherens_version	Current COHERENS version
creation_date	Date and time of creation
defined	.TRUE. if the file is activated (status is different from '0')

endfile	Switch to select which action needs to be taken in case an end of file condition occurs during input
	0: The program aborts with an error message.
	1: The program continues. No further read attempt is made.
	2: The program waits until the data are available. If the data are not available after a given period of time, the program aborts.
filedesc	A character string with a description of the file contents
filename	File name
form	File format
	'A' ASCII
	'U' sequential unformatted binary
	'N' netCDF
	'D' direct access unformatted binary (currently not implemented)
header_type	Type of file header
	0: No header (not allowed for netCDF files)
	1: Indicates that the file contains forcing data.
	2: Indicates that the file contains user-defined output data.
	3: Indicates that the file contains user-defined output with output grid data only.
info	Header information is written to a separate information file if .TRUE.
iostat	File I/O status
	-1: unable to open file
	0 : The file is not opened
	1 : The file pointer is located before the end of the file
	2 : The file pointer is located at the end of the file. An end of file condition did not yet occurred.
	3 : The file pointer is located at the end of the file. An end of file condition did occur.

iunit	File unit number
lenrec	Record length (direct access files only)
maxrecs	Total amount of time records in case of a time series file
nocoords	Number of coordinate variables
nodim	Rank of the data variables (user output only)
novars	Number of data variables
opened	.TRUE. (.FALSE.) if the file is connected (not connected)
pathname	Preappended file path
status	File status
	'0' undefined, i.e. the file is not activated
	'N' the file is available in a user-defined (non-COHERENS) format for input
	'R' the file is available in COHERENS standard format for input
	'W' the file is created for output in COHERENS standard format
	'P' as 'W' but the data are appended to an already existing standard COHERENS file. The option is currently not implemented.
	'T' temporary file (for internal use only). The option is currently not used.
timeid	NetCDF variable id of the time coordinate variable
timerec	Current time record number
time_regular	.TRUE. if the file contains time series data at regular time intervals.
tlims	Start/End/Step time indices used in updating the data from a forcing file
tskips	Currently not implemented
varid	ID of a variable in the file (only used internally)
zetaid	NetCDF variable file ID of the surface elevation coordinate variable

GridParams

```
(MODULE datatypes)
TYPE :: GridParams
   LOGICAL :: rotated
   INTEGER :: nhtype, n1dat, n2dat
   REAL :: delxdat, delydat, gridangle, longpole, x0dat, y0dat, x0rot, y0rot
END TYPE GridParams
```

File

datatypes.f90

Type

Derived type definition

Purpose

Attributes of surface grids

Description

delxdat Spacing in the X-direction in case of a uniform rectangular grid [meters or degrees longitude]

delydat Spacing in the Y-direction in case of a uniform rectangular grid [meters or degrees latitude]

gridangle Rotation angle of a rotated model grid with respect to the reference grid

longpole Longitude of the new North pole in case of a rotated spherical rectangular grid [degrees longitude]

nhtype Type of data grid

- 0: single point data grid or data grid not defined
- 1: rectangular grid with uniform grid spacings in Cartesian or spherical coordinates (data grid is constructed from x0dat, y0dat, delxdat, delydat)
- 2: rectangular grid with non-uniform spacings
- 3: non-rectangular (curvilinear or non-structured) grid
- 4: data grid coincides with the model grid

n1dat Number of grid points in the X-direction or, in case of a non-rectangular grid, the total number of grid points

n2dat Number of grid points in the Y-direction or equal to one in case of a non-rectangular grid

rotated .TRUE. in case of a rotated rectangular model grid, .FALSE. otherwise.

x0dat X-coordinate of the lower left (Southwest) corner in case of a regular grid [meters or longitude]

x0rot new (rotated) X-coordinate of the lower left (Southwest) corner in case of a rotated spherical grid [degrees longitude]

y0dat Y-coordinate of the lower left (Southwest) corner in case of a regular grid [meters or latitude]

y0rot new (rotated) Y-coordinate of the lower left (Southwest) corner in case of a rotated spherical grid [fractional degrees latitude]

HRelativeCoords

```
(MODULE datatypes)
TYPE :: HRelativeCoords
   INTEGER :: icoord, jcoord
   REAL :: xcoord, ycoord
END TYPE HRelativeCoords
```

File

datatypes.f90

Type

Derived type definition

Purpose

Horizontal relative coordinates

Reference

Section 10.4.1

Description

icoord Integer part of the ξ_1 curvilinear coordinate with respect to the reference coordinate system

jcoord Integer part of the ξ_2 curvilinear coordinate with respect to the reference coordinate system

xcoord Fractional part of the ξ_1 curvilinear coordinate with respect to the reference coordinate system

ycoord Fractional part of the ξ_2 curvilinear coordinate with respect to the reference coordinate system

OutGridParams

```
(MODULE datatypes)
TYPE :: OutGridParams
   LOGICAL :: gridded, grid_file, land_mask, time_grid
   CHARACTER (LEN=1) :: status
   CHARACTER (LEN=lentime) :: enddate, refdate, startdate
   INTEGER :: ncout, nodim, nostats, nowetout, nrout, nstepout, nzout, &
             & time_format
   INTEGER, DIMENSION(3) :: tlims, xlims, ylims, zlims
   REAL :: deltout
END TYPE OutGridParams
```

File

datatypes.f90

Type

Derived type definition

Purpose

Parameters for user-defined output and attributes of output data grids

Description

deltout	Output time step. Unit is seconds in case of an absolute time. Otherwise, the unit is determined by the value of time_format.
enddate	Date and time of the last output
gridded	.TRUE. (.FALSE.) for gridded (non-gridded) output
grid_file	The coordinates of the output grid are written to a separate external file if .TRUE.
land_mask	The data are written with a land mask (i.e. without land points) if .TRUE.
ncout	X-dimension of the output grid in case of gridded data
nodim	Spatial dimension of the output grid (0,2,3)
nostats	Number of output stations in case of non-gridded data
nowetout	Number of wet output data in case a land mask is applied

nrout	Y-dimension of the output grid in case of gridded data
nstepout	Number of output time steps
nzout	Vertical dimension of the output grid
refdate	Reference date/time used when the time coordinate is written in a numerical format in which case it is defined as the time elapsed since this reference date.
startdate	Date and time of the first output
status	Status of the corresponding output files
	'W' data are written to a newly created file
	'P' data are appended to an existing file. This option is currently not yet available.
time_format	Format of the time coordinate. The first and last case are absolute times, the other ones relative times with respect to the refdate
	0: date/time in string format
	1: seconds
	2: minutes
	3: hours
	4: days
	5: months (one month is taken as 30.4375 days)
	6: years (one year is taken as 365.25 days)
	7: date in years
time_grid	Create a time-dependent grid if .TRUE. and nodim = 3.
tlims	Start/end/step output time index. The step value determines the period in case of time-averaged or harmonic output.
xlims	Start/end/step output space index in the X-direction in case of gridded data
ylims	Start/end/step output space index in the Y-direction in case of gridded data
zlims	Start/end/step output space index in the vertical direction

StationLocs

(MODULE datatypes)

```
TYPE :: StationLocs
   INTEGER :: ipos, jpos
   CHARACTER (LEN=lenname) :: name
END TYPE StationLocs
```

File
datatypes.f90

Type
Derived type definition

Purpose
Attributes of the output station locations in case of non-gridded output. The stations are assumed to be located at C-nodes.

Description

ipos C-node X-index of the stations

jpos C-node Y-index of the stations

name Name of the station

VariableAtts

```
(MODULE datatypes)
TYPE :: VariableAtts
   CHARACTER (LEN=lenname) :: f90_name
   CHARACTER (LEN=lendesc) :: long_name, vector_name
   CHARACTER (LEN=lenunit) :: units
   CHARACTER (LEN=lennode) :: node
   INTEGER :: data_type, ivarid, klev, nrank, numvar, oopt
   INTEGER, DIMENSION(4) :: shape
END TYPE VariableAtts
```

File
datatypes.f90

Type
Derived type definition

Purpose
Variable attributes

Description

f90_name	FORTRAN 90 name
long_name	Long descriptive name
vector_name	Associated vector name in case the variable denotes a vector component
units	Variable unit
node	Nodal location of the variable on the model grid. In case of user output, the nodal location on the output grid ('C' or 'W')
data_type	Data type of the variable (see Table 8.1)
ivarid	Variable key id (a zero means undefined)
klev	Vertical level used when oopt equals oopt_klev
nrank	Variable rank (0 for a scalar)
numvar	Variable number in case the last dimension of the array variable represents a variable dimension
oopt	Selects the operator to be applied in case the variable is selected for user output
	oopt_null No operator applied
	oopt_min Minimum value
	oopt_max Maximum value
	oopt_mean Spatially averaged value
	oopt_klev Value at a specified vertical level
	oopt_max Value at a specified depth below the surface
shape	Array shape (not defined in case of a scalar)
dep	Depth value used when oopt equals oopt_dep

VRelativeCoords

```
(MODULE datatypes)
TYPE :: VRelativeCoords
   INTEGER :: kcoord
   REAL :: zcoord
END TYPE VRelativeCoords
```

File

datatypes.f90

Type
Derived type definition

Purpose
Vertical relative coordinates

Description

kcoord Vertical level of the grid point along the reference grid just below the data point

zcoord Normalised vertical distance from the vertical level kcoord to the location of the data point (between 0 and 1)

33.3 Density

```
MODULE density
REAL, DIMENSION(0:ncloc+1,0:nrloc+1,nz) :: beta_sal, beta_temp, dens
REAL, DIMENSION(1-nhalo:ncloc+nhalo,1-nhalo:nrloc+nhalo,nz) :: sal, temp
```

File
density.f90

Type
Module

Purpose
Density arrays

Description

beta_sal	Salinity expansion coefficient β_S	[PSU^{-1}]
beta_temp	Temperature expansion coefficient β_T	[^{0}C^{-1}]
dens	Mass density ρ	[kg/m^3]
sal	Salinity S	[PSU]
temp	Temperature T	[^{0}C]

33.4 Water depths

```
MODULE depths
REAL, DIMENSION(0:ncloc+1,0:nrloc+1) :: depmeanatc, deptotatc
REAL, DIMENSION(ncloc,nrloc) :: deptotatc_old, deptotatc_err
```

```
REAL, DIMENSION(0:ncloc+1,nrloc) :: depmeanatu, deptotatu_old, deptotatu_prev
REAL, DIMENSION(1-nhalo:ncloc+nhalo,0:nrloc+1) :: deptotatu
REAL, DIMENSION(ncloc,0:nrloc+1) :: depmeanatv, deptotatv_old, deptotatv_prev
REAL, DIMENSION(0:ncloc+1,1-nhalo:nrloc+nhalo) :: deptotatv
REAL, DIMENSION(ncloc+1,nrloc+1) :: depmeanatuv, deptotatuv
REAL, DIMENSION(0:nc+1,0:nr+1) :: depmeanglb
REAL, DIMENSION(0:ncloc+1,0:nrloc+1) :: dzeta, zeta, zeta_old
```

File

depths.f90

Type

Module

Purpose

Water depths and sea surface elevations

Description

depmeanatc	Mean water depth at the C-nodes	[m]
depmeanatu	Mean water depths at the U-nodes	[m]
depmeanatuv	Mean water depth at the UV-nodes	[m]
depmeanatv	Mean water depth at the V-nodes	[m]
depmeanglb	Mean water depth (global) at C- (or -UV-nodes)	[m]
deptotatc	Total water depth at the C-nodes	[m]
deptotatc_err	Total water depth error $\delta_e H$ at the C-nodes	[m]
deptotatc_old	Total water depth at the C-nodes and old baroclinic time step	[m]
deptotatu	Total water depth at the U-nodes	[m]
deptotatuv	Total water depth at the UV-nodes	[m]
deptotatu_old	Total water depth at the U-nodes and old baroclinic time step	[m]
deptotatu_prev	Total water depth at the U-nodes and previous (outer) iteration with the implicit scheme	[m]
deptotatv	Total water depth at the V-nodes	[m]
deptotatv_old	Total water depth at the V-nodes and old baroclinic time step	[m]
deptotatv_prev	Total water depth at the V-nodes and previous (outer) iteration with the implicit scheme	[m]

dzeta	Free surface correction defined as the difference between the surface elevation at the next and previous iteration (implicit scheme only) [m]
zeta	Surface elevation [m]
zeta_old	Surface elevation at the start of the first (outer) iteration with the implicit scheme [m]

33.5 Diffusion

```
MODULE diffusion
REAL, DIMENSION(0:ncloc,0:nrloc) :: hdifcoef2datc
REAL, DIMENSION(ncloc+1,nrloc+1) ::  hdifcoef2datuv
REAL, DIMENSION(0:ncloc,0:nrloc,nz) :: hdifcoef3datc
REAL, DIMENSION(ncloc+1,nrloc,nz) :: hdifcoef3datu
REAL, DIMENSION(ncloc,nrloc+1,nz) :: hdifcoef3datv
REAL, DIMENSION(ncloc+1,nrloc+1,nz) :: hdifcoef3datuv
REAL, DIMENSION(0:ncloc,0:nrloc,nz+1) :: kinvisc, vdifcoefmom
REAL, DIMENSION(ncloc,nrloc,nz+1) :: vdifcoefscal, vdifcoeftke
```

File

diffusion.f90

Type

Module

Purpose

Horizontal and vertical diffusion coefficients

Description

hdifcoef2datc	Vertically integrated horizontal diffusion coefficient $\overline{\nu_H}$ at the C-nodes [m^3/s]
hdifcoef2datuv	Vertically integrated horizontal diffusion coefficient $\overline{\nu_H}$ at the UV-nodes [m^3/s]
hdifcoef3datc	Horizontal 3-D diffusion coefficient ν_H at the C-nodes [m^2/s]
hdifcoef3datu	Horizontal 3-D diffusion coefficient λ_H at the U-nodes [m^2/s]
hdifcoef3datuv	Horizontal 3-D diffusion coefficient ν_H at the UV-nodes [m^2/s]

hdifcoef3datv	Horizontal 3-D diffusion coefficient λ_H at the V-nodes	[m^2/s]
kinvisc	Kinematic viscosity	[m^2/s]
vdifcoefmom	Vertical diffusion coefficient ν_T for momentum at the W-nodes	[m^2/s]
vdifcoefscal	Vertical diffusion coefficient λ_T for scalars at the W-nodes	[m^2/s]
vdifcoeftke	Vertical diffusion coefficient ν_k for turbulent kinetic energy at the W-nodes	[m^2/s]

33.6 Fluxes

```
MODULE fluxes
REAL, DIMENSION(ncloc,nrloc) ::  bdragcoefatu, bdragcoefatv, bfricatu, &
                                & bfricatv, bstresatc, ces, chs, qlatent, &
                                & qlwave, qnonsol, qsensible, ssalflux, &
                                & sstresatc, ubstresatu, usstresatu, &
                                & vbstresatv, vsstresatv, zeros2d
REAL, DIMENSION(0:ncloc,0:nrloc) :: bdragcoefatc, cds, zroughatc
REAL, DIMENSION(0:ncloc,nrloc) :: usstresatc
REAL, DIMENSION(ncloc,0:nrloc) :: vsstresatc
REAL, DIMENSION(0:nwind,-ntemp:ntemp,nrelhum) :: cdstab, cestab, chstab
```

File

fluxes.f90

Type

Module

Purpose

Bottom and surface flux arrays

Description

bdragcoefatc	Bottom drag coefficient C_{db} at the C-nodes	
bdragcoefatu	Bottom drag coefficient C_{db} at the U-nodes	
bdragcoefatv	Bottom drag coefficient C_{db} at the V-nodes	
bfricatu	Bottom friction velocity $u_{\star b}$ at the U-nodes	[m/s]
bfricatv	Bottom friction velocity $u_{\star b}$ at the V-nodes	[m/s]
bstresatc	Magnitude of the bottom stress τ_b at the C-nodes	[m^2/s^2]

cds	Surface drag coefficient C_{ds}	
cdstab	Surface drag coefficient given in tabular form as function of wind speed, air minus sea temperature difference and relative humidity. The table is used for interpolation in case a Monin-Obokhov formulation is selected.	
ces	Exchange coefficient for the latent heat flux C_e (Dalton number)	
cestab	Dalton number given in tabular as function of wind speed, air minus sea temperature difference and relative humidity. The table is used for interpolation in case a Monin-Obokhov formulation is selected.	
chs	Exchange coefficient for the sensible heat flux C_h (Stanton number)	
chstab	Stanton number given in tabular form as function of wind speed, air minus sea temperature difference and relative humidity. The table is used for interpolation in case a Monin-Obokhov formulation is selected.	
qlatent	Latent surface heat flux	[W/m^2]
qlwave	Long-wave surface heat flux	[W/m^2]
qnonsol	Non-solar surface heat flux	[W/m^2]
qsensible	Sensible surface heat flux	[W/m^2]
ssalflux	Surface salinity flux	[PSU m/s]
sstresatc	Magnitude of the surface stress τ_s at the C-nodes	[m^2/s^2]
ubstresatu	X-component of the bottom stress τ_{b1} at the U-nodes	[m^2/s^2]
usstresatc	X-component of the surface stress τ_{s1} at the C-nodes	[m^2/s^2]
usstresatu	X-component of the surface stress τ_{s1} at the U-nodes	[m^2/s^2]
vbstresatv	Y-component of the bottom stress τ_{b2} at the V-nodes	[m^2/s^2]
vsstresatc	Y-component of the surface stress τ_{s2} at the C-nodes	[m^2/s^2]
vsstresatv	Y-component of the surface stress τ_{s2} at the V-nodes	[m^2/s^2]
zeros2d	Work space array with zero values	

zroughatc Bottom roughness length z_0 at the C-nodes [m]

33.7 Model grid arrays

```
MODULE grid
LOGICAL, DIMENSION(nobu) ::  westobu
LOGICAL, DIMENSION(nobv) ::  soutobv
LOGICAL, DIMENSION(nobx) ::  westobx
LOGICAL, DIMENSION(noby) ::  soutoby
LOGICAL, DIMENSION(ncloc,nrloc) :: maskatc_int
LOGICAL, DIMENSION(0:ncloc+1,0:nrloc+1) :: seapoint

INTEGER, DIMENSION(nobu) :: indexobu, iobu, jobu
INTEGER, DIMENSION(nobv) :: indexobv, iobv, jobv
INTEGER, DIMENSION(nobx) :: indexobx, iobx, jobx
INTEGER, DIMENSION(noby) :: indexoby, ioby, joby
INTEGER, DIMENSION(nobuloc_ext) :: iobuloc, jobuloc
INTEGER, DIMENSION(nobvloc_ext) :: iobvloc, jobvloc
INTEGER, DIMENSION(nobxloc_ext) :: iobxloc, jobxloc
INTEGER, DIMENSION(nobyloc_ext) :: iobyloc, jobyloc
INTEGER, DIMENSION(1-nhalo:ncloc+nhalo,1-nhalo:nrloc+nhalo) :: &
                                  & nodeatc, node2du, node2duv, node2dv
INTEGER, DIMENSION(1-nhalo:ncloc+nhalo,1-nhalo:nrloc+nhalo,nz) :: &
                                                & nodeatu, nodeatuv, nodeatv
INTEGER, DIMENSION(1-nhalo:ncloc+nhalo,1-nhalo:nrloc+nhalo,nz+1) :: &
                                                & nodeatuw, nodeatvw
REAL, DIMENSION(ncloc,nrloc) :: alphatc_fld, alphatu_fld, alphatv_fld, &
                                & gangleatc, &
                                & rlxobcatu, rlxobcatv
REAL, DIMENSION(ncloc+1,nrloc) :: gaccatu
REAL, DIMENSION(ncloc,nrloc+1) :: gaccatv
REAL, DIMENSION(ncloc+1,0:nrloc) :: coriolatu
REAL, DIMENSION(0:ncloc,nrloc+1) :: coriolatv
REAL, DIMENSION(0:nc+1) :: gdelxglb
REAL, DIMENSION(0:nr+1) :: gdelyglb
REAL, DIMENSION(0:ncloc+1,0:nrloc+1) :: gaccatc, gxcoord, gycoord
REAL, DIMENSION(0:ncloc,0:nrloc) :: gxlon, gylat
REAL, DIMENSION(0:nc+1,0:nr+1) :: gxcoordglb, gycoordglb
REAL, DIMENSION(0:nc+1,0:nr+1,nz+1) :: gscoordglb
REAL, DIMENSION(0:ncloc+1,0:nrloc+1,nz+1) :: gscoord, gscoordatuw, &
```

```
                                                & gscoordatvw, gscoordatw
REAL, DIMENSION(0:ncloc+1,0:nrloc+1,2:nz) :: delzatw
REAL, DIMENSION(0:ncloc+1,0:nrloc+1,nz) :: delzatc, gscoordatc, gscoordatu, &
                                           & gscoordatv
REAL, DIMENSION(nz+1) :: gsigcoord
REAL, DIMENSION(1-nhalo:ncloc+nhalo,1-nhalo:nrloc+nhalo) :: &
              & delxatc, delxatu, delxatuv, delxatv, &
              & delyatc, delyatu, delyatuv, delyatv
REAL, DIMENSION(0:ncloc+1,nrloc,nz) :: delzatu
REAL, DIMENSION(ncloc,0:nrloc+1,nz) :: delzatv
REAL, DIMENSION(ncloc+1,nrloc+1,nz) :: delzatuv
REAL, DIMENSION(ncloc+1,nrloc,2:nz) :: delzatuw
REAL, DIMENSION(ncloc,nrloc+1,2:nz) :: delzatvw
```

File

grid.f90

Type

Module

Purpose

Arrays for the model grid

Description

alphatc_fld	α-factor used in the drying/wetting algorithm at the C-nodes	
alphatu_fld	α-factor used in the drying/wetting algorithm at the U-nodes	
alphatv_fld	α-factor used in the drying/wetting algorithm at the V-nodes	
coriolatu	Coriolis frequency at the U-nodes	[rad/s]
coriolatv	Coriolis frequency at the V-nodes	[rad/s]
delxatc	Grid spacing in the X-direction at the C-nodes	[m]
delxatu	Grid spacing in the X-direction at the U-nodes	[m]
delxatuv	Grid spacing in the X-direction at the UV-nodes	[m]
delxatv	Grid spacing in the X-direction at the V-nodes	[m]
delyatc	Grid spacing in the Y-direction at the C-nodes	[m]
delyatu	Grid spacing in the Y-direction at the U-nodes	[m]
delyatuv	Grid spacing in the Y-direction at the UV-nodes	[m]

delzatc	Vertical grid spacing at the C-nodes	[m]
delzatu	Vertical grid spacing at the U-nodes	[m]
delzatuv	Vertical grid spacing at the UV-nodes	[m]
delzatuw	Vertical grid spacing at the UW-nodes	[m]
delzatv	Vertical grid spacing at the V-nodes	[m]
delzatvw	Vertical grid spacing at the VW-nodes	[m]
delzatw	Vertical grid spacing at the W-nodes	[m]
delyatv	Grid spacing in the Y-direction at the V-nodes	[m]
gaccatc	Acceleration of gravity at the C-nodes	[m^2/s]
gaccatu	Acceleration of gravity at the U-nodes	[m^2/s]
gaccatv	Acceleration of gravity at the V-nodes	[m^2/s]
gangleatc	Angle between the reference and local grid X-axis at C-nodes	[degrees]
gdelxglb	Grid spacings in the X-direction of a non-uniform rectangular grid (global array) [m or degrees]	
gdelyglb	Grid spacings in the Y-direction of a non-uniform rectangular grid (global array) [m or degrees]	
gscoord	Sigma coordinates of the UVW-nodes	
gscoordatc	Sigma coordinates of the C-nodes	
gscoordatu	Sigma coordinates of the U-nodes	
gscoordatuw	Sigma coordinates of the UW-nodes	
gscoordatv	Sigma coordinates of the V-nodes	
gscoordatvw	Sigma coordinates of the VW-nodes	
gscoordatw	Sigma coordinates of the W-nodes	
gscoordglb	Sigma coordinates of the W- or UVW-nodes (global array)	
gsigcoord	Sigma coordinates in case the vertical grid is horizontally uniform (iopt_grid_vtype=1)	
gxcoord	X-coordinates of the UV-nodes (local array)	[m or degrees longitude]
gxcoordglb	X-coordinates of the UV-nodes (global array)	[m or degrees longitude]
gxlon	Longitude coordinate at the C-nodes	[rad]
gycoord	Y-coordinates of the UV-nodes (local array)	[m or degrees latitude]

gycoordglb	Y-coordinates of the UV-nodes (global array) [m or degrees latitude]
gylat	Latitude coordinate at the C-nodes [rad]
indexobu	Maps the local open boundary indices at the U-nodes to their corresponding global ones.
indexobv	Maps the local open boundary indices at the V-nodes to their corresponding global ones.
indexobx	Maps the local open boundary indices at the X-nodes to their corresponding global ones.
indexoby	Maps the local open boundary indices at the Y-nodes to their corresponding global ones.
iobu	Global X-index of the West/East U-node open boundaries
iobuloc	Local X-index of the West/East U-node open boundaries
iobv	Global X-index of the South/North V-node open boundaries
iobvloc	Local X-index of the South/North V-node open boundaries
iobx	Global X-index of the West/East X-node open boundaries
iobxloc	Local X-index of the West/East X-node open boundaries
ioby	Global X-index of the South/North Y-node open boundaries
iobyloc	Local X-index of the South/north Y-node open boundaries
jobu	Global Y-index of the West/East U-node open boundaries
jobuloc	Local Y-index of the West/East U-node open boundaries
jobv	Global Y-index of the South/North V-node open boundaries
jobvloc	Local Y-index of the South/North V-node open boundaries
jobx	Global Y-index of the West/East X-node open boundaries
jobxloc	Local Y-index of the West/East X-node open boundaries
joby	Global Y-index of the South/north Y-node open boundaries
jobyloc	Local Y-index of the South/north Y-node open boundaries

maskatc_int .TRUE./.FALSE. at wet/dry C-node grid points

nodeatc Pointers at C-nodes

0: dry cell

1: wet cell

nodeatu Pointers at U-node cell faces

0: dry (land) cell face

1: coastal or solid structure boundary

2: interior wet U-node

3: open sea boundary

4: river open boundary

nodeatuv Pointer at corner (UV) nodes

0: at least two surrrounding U-nodes or at least two surrrounding V-nodes are dry

1: interior wet node, i.e. at most one surrounding U-node and at most one surrounding V-node is dry and none of the four surrounding velocity nodes are open boundaries

2: X-node open boundary, in which case at least one of the surrounding U-nodes is an open boundary while the other one is either a closed node or open boundary, but the node is not a Y-node open boundary

3: Y-node open boundary, in which case at least one of the surrounding V-nodes is an open boundary while the other one is either a closed node or open boundary, but the node is not an X-node open boundary

4: the node is both a X- and a Y-node open boundary

nodeatuw Pointer at UW-node cell faces

0: dry (land) cell face or bottom cell (1) or surface cell (nz+1)

1: coastal or solid structure boundary

2: interior wet UW-node

3: open sea boundary

4: river open boundary

nodeatv Pointers at V-node cell faces

0: dry (land) cell face

1: coastal or solid structure boundary

2: interior wet V-node

3: open sea boundary

4: river open boundary

nodeatvw Pointer at VW-node cell faces

0: dry (land) cell face or bottom cell (1) or surface cell (nz+1)

1: coastal or solid structure boundary

2: interior wet VW-node

3: open sea boundary

4: river open boundary

node2du Pointers at U-nodes for 2-D calculations

0: dry (land) cell face

1: coastal boundary

2: at least one U-node interface in the vertical is wet

3: open sea boundary

4: river open boundary

node2dv Pointers at V-nodes for 2-D calculations

0: dry (land) cell face

1: coastal boundary

2: at least one V-node interface in the vertical is wet

3: open sea boundary

4: river open boundary

node2duv Pointer at corner (UV) nodes for 2-D calculations

0: at least two surrrounding U-nodes or at least two surrrounding V-nodes are dry

1: at least one corner node in the vertical is an interior wet node, i.e. at most one surrounding U-node and at most one surrounding V-node (at the same vertical position) is dry and none of the four surrounding velocity nodes are open boundaries

2: X-node open boundary, in which case at least one of the surrounding U-nodes is an open boundary while the other one is either a closed node or open boundary, but the node is not a Y-node open boundary

3: Y-node open boundary, in which case at least one of the surrounding V-nodes is an open boundary while the other one is either a closed node or open boundary, but the node is not an X-node open boundary

4: the node is both a X- and a Y-node open boundary

rlxobcatu Relaxation factor for horizontal momentum advection at U-nodes

rlxobcatv Relaxation factor for horizontal momentum advection at V-nodes

seapoint .FALSE. for permanent land cells, .TRUE. otherwise

seapoint .FALSE. for permanent land cells, .FALSE. otherwise

soutobv .TRUE./.FALSE. at South/North V-open boundaries (global array)

soutoby .TRUE./.FALSE. at South/North Y-open boundaries (global array)

westobu .TRUE./.FALSE. at West/East V-open boundaries (global array)

westobx .TRUE./.FALSE. at West/East X-open boundaries (global array)

33.8 Model grid parameters

```
MODULE gridpars
!---grid dimensions
INTEGER :: nc = 0, nr = 0, nz = 0
INTEGER :: ncloc, nc1loc, nc2loc, nrloc, nr1loc, nr2loc
!---number of open boundary points
INTEGER :: nobu, nobv, nobx, noby
INTEGER :: nosbu = 0, nosbv = 0, nrvbu = 0, nrvbv = 0
INTEGER :: nobuloc, nobvloc, nobxloc, nobyloc
INTEGER :: nobuloc_ext, nobvloc_ext, nobxloc_ext, nobyloc_ext
INTEGER :: nosbuloc, nosbvloc, nrvbuloc, nrvbvloc
!---shortcuts for regular grid spacings
```

```
LOGICAL :: dXregX, dXregY, dYregX, dYregY, dXYreg, dZregZ, rotate_gvecs
!---halo dimensions
INTEGER, PARAMETER :: nhalo = 2
INTEGER :: nhdens, nhfvel, nhscal, nhturb, nh2vel, nh3vel
!---number of open boundary points
INTEGER :: noseaatc, noseaatcloc, nowetatc, nowetatcloc
!---number of sediment/biological variables used at open boundaries and for
!   nesting
INTEGER :: maxbiovars = 0, maxsedvars = 0
!---mean grid spacing
REAL :: delgrid
```

File

gridpars.f90

Type

Module

Purpose

Model grid parameters

Description

delgrid	Domain-averaged grid spacing for both X- and Y-direction [m]
dXregX	.TRUE. if the grid spacing in the X-direction is uniform in the X-direction
dXregY	.TRUE. if the grid spacing in the X-direction is uniform in the Y-direction
dXYreg	.TRUE. if all grid spacings are uniform in both X- and Y-directions
dYregX	.TRUE. if the grid spacing in the Y-direction is uniform in the X-direction
dYregY	.TRUE. if the grid spacing in the Y-direction is uniform in the Y-direction
dZregZ	.TRUE. for a uniform vertical vertical grid (iopt_grid_vtype=1)
maxbiovars	Maximum number of biological variables used at open boundaries and for nesting
maxsedvars	Maximum number of sediment fractions used at open boundaries and for nesting

nc	Number of grid cells in the X-direction on the global grid
ncloc	Number of grid cells in the X-direction on the local grid
nc1loc	Global X-index of cell (1,1) on the local grid
nc2loc	Global X-index of cell (ncloc,1) on the local grid
nhalo	Maximum size of a halo in all directions. The current value of 2 cannot be changed.
nhdens	Minimum required size of the halo needed for density arrays (1/2)
nhfvel	Minimum required size of the halo for horizontal advective currents (1/2)
nhscal	Minimum required size of the halo for scalar (non-density) arrays (1/2)
nhturb	Minimum required size of the halo for the turbulence transport arrays (0/1/2)
nh2vel	Minimum required size of the halo for the 2-D depth-integrated currents (1/2)
nh3vel	Minimum required size of the halo for the 3-D currents (1/2)
nobu	Global number of open boundary points at the U-nodes
nobuloc	Local number of open boundary points at the U-nodes
nobuloc_ext	Local number of interior open boundary points at the U-nodes including points within the first column of the eastern halo
nobv	Global number of open boundary points at the V-nodes
nobvloc	Local number of open boundary points at the V-nodes
nobvloc_ext	Local number of interior open boundary points at the V-nodes including points within the first row of the northern halo
nobx	Global number of open boundary points at the X-nodes
nobxloc	Local number of open boundary points at the X-nodes
nobxloc_ext	Local number of interior open boundary points at the X-nodes including points within the first column of the eastern halo
noby	Global number of open boundary points at the Y-nodes
nobyloc	Local number of open boundary points at the Y-nodes

nobyloc_ext	Local number of interior open boundary points at the Y-nodes including points within the first column of the northern halo
nosbu	Global number of open sea open boundary points at the U-nodes
nosbuloc	Local number of open sea open boundary points at the U-nodes
nosbv	Global number of open sea open boundary points at the V-nodes
nosbvloc	Local number of open sea open boundary points at the V-nodes
noseaatc	Number of sea (wet or dry) C-nodes on the global domain
noseaatcloc	Number of sea (wet or dry) C-nodes on the local domain
nowetatc	Number of active wet C-nodes on the global domain
nowetatcloc	Number of active wet C-nodes on the local domain
nr	Number of grid cells in the Y-direction on the global grid
nrloc	Number of grid cells in the Y-direction on the local grid
nrvbu	Global number of river open boundary points at the U-nodes
nrvbuloc	Local number of river open boundary points at the U-nodes
nrvbv	Global number of river open boundary points at the V-nodes
nrvbvloc	Local number of river open boundary points at the V-nodes
nr1loc	Global Y-index of cell (1,1) on the local grid
nr2loc	Global Y-index of cell (1,nrloc) on the local grid
nz	Number of grid cells in the vertical direction
rotate_gvecs	.TRUE. if vectors in the model grid are rotated with respect to the reference coordinate system

33.9 General and I/O parameters

`MODULE iopars`

General parameters

```
LOGICAL :: cold_start = .FALSE., next_simul = .FALSE.
INTEGER :: isimul = 0, nopenf = 0, nrecunit = 4, nosimul = 0
CHARACTER (LEN=lentitle) :: intitle, outtitle, runtitle
CHARACTER (LEN=lenname), DIMENSION(MaxProgLevels) :: procname
LOGICAL :: log_fill = log_undef
INTEGER :: int_fill = int_undef
REAL :: real_fill = real_undef, real_min = real_flag
```

File

iopars.f90

Type

Module

Purpose

General parameters mainly related to all kinds of input and output

Description

cold_start	If .TRUE., the program only executes the initialisation and finalisation phases, but does not enter the time loop.
intitle	Title for model forcing files
int_fill	Data flag for invalid integer data
isimul	Number of the simulation (as given by the corresponding line number in the *defruns* excluding comment lines)
log_fill	Data flag for invalid logical data
next_simul	When a new data line is read from the *defruns* file, its value is set to .TRUE. to start a new simulation or to .FALSE. to exit the program.
nopenf	Number of current connected files
nosimul	Number of simulations
nrecunit	Unit record size in bytes for direct access I/O (not used in the present implementation)
outtitle	Title for user output files
procname	Name of the subprogram at the current program level
runtitle	Simulation title
real_fill	Data flag for invalid real data

real_min Real data are considered as valid or invalid if they are greater than or lower than (or equal to) this critical threshold

Parameters for model forcing

```
(MODULE iopars)
INTEGER, DIMENSION(MaxIOTypes,2) :: maxdatafiles
INTEGER, DIMENSION(MaxCIFTypes) :: ciflinenum
TYPE (FileParams), DIMENSION(MaxCIFTypes) :: ciffiles
TYPE (FileParams), DIMENSION(MaxIOTypes,MaxIOFiles,2) :: modfiles
TYPE (GridParams), DIMENSION(MaxGridTypes,MaxGridFiles) :: surfacegrids
INTEGER, PARAMETER :: io_mppmod = 1, io_inicon = 2, io_fincon = 3, &
                    & io_modgrd = 4, io_metgrd = 5, io_sstgrd = 6, &
                    & io_wavgrd = 7, io_biogrd = 8, io_nstgrd = 9, &
                    & io_sedspc = 10, io_biospc = 11, io_1uvsur = 12, &
                    & io_2uvobc = 13, io_3uvobc = 14, io_salobc = 15, &
                    & io_tmpobc = 16, io_sedobc = 17, io_bioobc = 18, &
                    & io_rlxobc = 19, io_nstspc = 20, io_2uvnst = 21, &
                    & io_3uvnst = 22, io_salnst = 23, io_tmpnst = 24, &
                    & io_sednst = 25, io_bionst = 26, io_metsur = 27, &
                    & io_sstsur = 28, io_wavsur = 29, io_biosur = 30, &
                    & io_drycel = 31, io_thndam = 32, io_weibar = 33, &
                    & io_disspc = 34, io_disloc = 35, io_disvol = 36, &
                    & io_discur = 37, io_dissal = 38, io_distmp = 39

INTEGER, PARAMETER :: icif_defruns= 1, icif_model = 2, icif_bio = 3, &
                    & icif_sed = 4
INTEGER, PARAMETER :: ics_phys = 1, ics_bio = 2, ics_sed = 3
INTEGER, PARAMETER :: igrd_model = 1, igrd_meteo = 2, igrd_sst = 3, &
                    & igrd_waves = 4, igrd_bio = 5
CHARACTER (LEN=6), DIMENSION(MaxIOTypes) :: modfiles_desc = &
 & (/'mppmod','inicon','fincon','modgrd','metgrd','sstgrd','wavgrd','biogrd',&
   & 'nstgrd','sedspc','biospc','1uvsur','2uvobc','3uvobc','salobc','tmpobc',&
   & 'sedobc','bioobc','rlxobc','nstspc','2uvnst','3uvnst','salnst','tmpnst',&
   & 'sednst','bionst','metsur','sstsur','biosur','wavsur'/)
```

File

iopars.f90

Type
Module

Purpose
Parameters for model forcing

Description

ciflinenum	Number of the last input line read from a CIF
ciffiles	Attributes of CIFs
icif_*	Key ids of CIFs
icif_bio	CIF with setup parameters for the biological model
icif_defruns	*defruns* file
icif_model	CIF with model setup parameters
icif_sed	CIF with setup parameters for the sediment model
ics_*	Key ids for the file index in case the first index of modfiles equals io_inicon
ics_bio	initial conditions for biological arrays (currently not implemented)
ics_phys	initial conditions for physical arrays
ics_sed	initial conditions for sediment arrays
igrd_*	Key ids used for defining surface data grids
igrd_bio	surface grid for biological data (currently not implemented)
igrd_meteo	meteorological grid
igrd_model	(horizontal) model grid
igrd_sst	SST grid
igrd_waves	surface wave grid
io_*	File descriptor key ids
io_biogrd	surface grid for biological data
io_bionst	biological output for nesting
io_bioobc	definitions of open boundary conditions for biological variables (file number equals 1) or input of open boundary data (file number larger than 1).

io_biospc	(time-independent) arrays used for the setup of a biological model (currently not implemented)
io_biosur	biological surface data (currently not implemented)
io_disspc	discharge specifier arrays
io_disloc	discharge locations
io_disvol	volume discharge data
io_discur	discharge area and direction
io_dissal	salinity discharge data
io_distmp	temperature discharge data
io_drycel	dry cell locations
io_inicon	initial conditions
io_fincon	final conditions
io_metgrd	surface meteo grid
io_metsur	meteorological forcing data
io_modgrd	model grid, bathymetry and open boundary locations
io_mppmod	domain decomopistion
io_nstgrd	locations of nested sub-grid open boundaries
io_nstspc	number of open boundary locations at sub-grid nested grids
io_rlxobc	definitions of areas for the application of the relaxation open boundary scheme
io_salnst	salinity output for nesting
io_salobc	definitions of open boundary conditions for salinity (file number equals 1) or input of open boundary data (file number larger than 1)
io_sednst	sediment output for nesting
io_sedobc	definitions of open boundary conditions for sediment variables (file number equals 1) or input of open boundary data (file number larger than 1).

io_sedspc	(time-independent) arrays used for the setup of a sediment model (particle attributes in the COHERENS sediment model)
io_sstgrd	surface SST grid
io_sstsur	SST data
io_thndam	thin dams locations
io_tmpnst	temperature output for nesting
io_tmpobc	definitions of open boundary conditions for temperature (file number equals 1) or input of open boundary data (file number larger than 1)
io_wavgrd	surface wave grid
io_wavsur	surface wave data
io_weibar	weirs and barries locations and parameters
io_1uvsur	definitions of surface boundary conditions (file number equals 1) or input of surface data (file number greater than 1) in case the model is applied in water column (1-D) mode
io_2uvnst	2-D mode output for nesting
io_2uvobc	definitions of 2-D open boundary conditions (file number equals 1) or input of open boundary data (file number larger than 1)
io_3uvnst	3-D mode output for nesting
io_3uvobc	definitions of open boundary conditions for the baroclinic currents (file number equals 1) or input of open boundary data (file number larger than 1)
maxdatafiles	Largest file index (second dimension of modfiles) for a given file descriptor (first index of modfiles)
modfiles	Attributes of model forcing files. The first index is given by a file descriptor key id of the form io_* (see below), the second is the file index, the third equals 1 for input or 2 for output.
modfiles_desc	Descriptors of forcing files
surfacegrids	Attributes of surface data grids. The first index is given by a grid descriptor key id of the form igrd_* (see above), the second is the file index (currently equal to 1)

Parameters for user-defined output

```
(MODULE iopars)
!---time series
INTEGER :: nosetstsr = 0, nostatstsr = 0,  novarstsr = 0
INTEGER, DIMENSION(nosetstsr,novarstsr) :: ivarstsr
INTEGER, DIMENSION(nosetstsr,nostatstsr) :: lstatstsr
TYPE (FileParams), DIMENSION(nosetstsr) :: tsrgrd, tsr0d, tsr2d, tsr3d
TYPE (OutGridParams), DIMENSION(nosetstsr) :: tsrgpars
TYPE (StationLocs), DIMENSION(nostatstsr) :: tsrstatlocs
TYPE (VariableAtts), DIMENSION(novarstsr) :: tsrvars
!---time averages
INTEGER :: nosetsavr = 0, nostatsavr = 0, novarsavr = 0
INTEGER, DIMENSION(nosetsavr,novarsavr) :: ivarsavr
INTEGER, DIMENSION(nosetsavr,nostatsavr) :: lstatsavr
TYPE (FileParams), DIMENSION(nosetsavr) :: avrgrd, avr0d, avr2d, avr3d
TYPE (OutGridParams), DIMENSION(nosetsavr) :: avrgpars
TYPE (StationLocs), DIMENSION(nostatsavr) :: avrstatlocs
TYPE (VariableAtts), DIMENSION(novarsavr) :: avrvars
!---harmonic analysis
INTEGER :: nofreqsanal = 0, nosetsanal = 0, nostatsanal = 0, novarsanal = 0
CHARACTER (LEN=lentime), DIMENSION(nosetsanal) :: cdate_time_ref
CHARACTER (LEN=lenfreq), DIMENSION(nofreqsanal) :: harm_freq_names
INTEGER, DIMENSION(nofreqsanal) :: index_anal
INTEGER, DIMENSION(nosetsanal) :: nofreqsharm
INTEGER, DIMENSION(nosetsanal,nofreqsanal) :: ifreqsharm
INTEGER, DIMENSION(nosetsanal,novarsanal) :: ivarsanal
INTEGER, DIMENSION(nosetsanal,nostatsanal) :: lstatsanal
REAL, DIMENSION(nofreqsanal) :: harm_freq
TYPE (FileParams), DIMENSION(nosetsanal) :: analgrd, res0d, res2d, res3d
TYPE (FileParams), DIMENSION(nosetsanal,nofreqsanal) :: &
                           & amp0d, amp2d, amp3d, &
                           & pha0d, pha2d, pha3d, &
                           & ell2d, ell3d
TYPE (OutGridParams), DIMENSION(nosetsanal) :: analgpars
TYPE (StationLocs), DIMENSION(nostatsanal) :: analstatlocs
TYPE (VariableAtts), DIMENSION(novarsanal) :: analvars
!---elliptic parameters
INTEGER, DIMENSION(nosetsanal,14) :: ivarsell
INTEGER, DIMENSION(nosetsanal,2) :: ivecell2d, ivecell3d
TYPE (VariableAtts), DIMENSION(14) :: ellvars
```

```
!---output operators
INTEGER, PARAMETER :: oopt_null = 0, oopt_dep = 1, oopt_klev = 2, &
                    & oopt_min = 3, oopt_max = 4, oopt_mean = 5
```

File

iopars.f90

Type

Module

Purpose

Parameters for user-defined output

Description

analgpars	Attributes of the output grids for harmonic output
analgrd	Attributes of the grid file for harmonic output containing coordinate data only
analstatlocs	Attributes of all output stations for harmonic output (locations, names)
analvars	Attributes of all variables for harmonic output
amp0d	Attributes of the 0-D files for the harmonic output of amplitudes
amp2d	Attributes of the 2-D files for the harmonic output of amplitudes
amp3d	Attributes of the 3-D files for the harmonic output of amplitudes
avrgpars	Attributes of the output grids for time averaged output
avrgrd	Attributes of the grid file for time averaged output containing coordinate data only
avrstatlocs	Attributes of all output stations for time averaged output (locations, names)
avrvars	Attributes of all variables for time averaged output
avr0d	Attributes of the 0-D files for time averaged output
avr2d	Attributes of the 2-D files for time averaged output
avr3d	Attributes of the 3-D files for time averaged output

cdate_time_ref	Reference date with respect to which the analysed phases are given. If not defined, the phases are given with respect to the central analysis time or the Greenwich astronomical phase.
ellvars	Attributes of all possible elliptic variables (set by the program)
ell2d	Attributes of the 2-D files for harmonic output of elliptic parameters
ell3d	Attributes of the 3-D files for harmonic output of elliptic parameters
harm_freq	Frequencies used for analysis [radian/s]
harm_freq_names	Names of the frequencies used for harmonic analysis
ifreqsharm	Index mapping array for harmonic frequencies. The element ifreqsharm(iset,ifreq) maps, for set iset, the local frequency ifreq into the corresponding "global" array index as defined in the array index_anal.
index_anal	Key ids of the harmonic constituents. If zero, the corresponding frequencies need to be defined by the user as well.
ivarsanal	Index mapping array for harmonic output. The element ivarsanal(iset,ivar) maps, for set iset, the local variable index ivar into the corresponding array index in analvars.
ivarsavr	Index mapping array for time averaged output. The element ivarsavr(iset,ivar) maps, for set iset, the local variable index ivar into the corresponding array index in avrvars.
ivarsell	Index mapping array selecting the elliptic parameters for output. The element ivarsell(iset,ivar) maps, for set iset, the local variable index ivar into the corresponding array index in ellvars.
ivarstsr	Index mapping array for time series output. The element ivarstsr(iset,ivar) maps, for set iset, the local variable index ivar into the corresponding array index in tsrvars (see below).
ivecell2d	Indices of the elements in analvars representing the X- and Y-component of the 2-D elliptic vectors

ivecell3d	Indices of the elements in analvars representing the X- and Y-component of the 3-D elliptic vectors
lstatsanal	Station label for harmonic output. The element lstatsanal(iset,istat) maps, for set iset, the local index istat into the corresponding global array index in analstatlocs.
lstatsavr	Station label for time averaged output. The element lstatsavr(iset,istat) maps, for set iset, the local index istat into the corresponding global array index in avrstatlocs.
lstatstsr	Station label for time series output. The element lstatstsr(iset,istat) maps, for set iset, the local index istat into the corresponding global array index in tsrstatlocs.
nofreqsanal	Total number of frequencies for harmonic analysis
nofreqsharm	Number of frequencies used for each output set
nosetsanal	Number of output sets for harmonic output
nosetsavr	Number of output sets for time averaged output
nosetstsr	Number of time series output sets
nostatsanal	Total number of stations for harmonic output
nostatsavr	Total number of stations for time averaged output
nostatstsr	Total number of stations for time series output
novarsanal	Total number of variables for harmonic output
novarsavr	Total number of variables for time averaged output
novarstsr	Total number of variables for time series output
oopt_*	User output operators
	oopt_dep At a specified depth from the surface
	oopt_klev At a specified vertical level
	oopt_max Maximum value
	oopt_mean Spatially averaged value
	oopt_min Minimum value
	oopt_null No operator
pha0d	Attributes of the 0-D files for the harmonic output of phases

pha2d	Attributes of the 2-D files for the harmonic output of phases
pha3d	Attributes of the 3-D files for the harmonic output of phases
res0d	Attributes of the 0-D files for the residual harmonic output
res2d	Attributes of the 2-D files for the residual harmonic output
res3d	Attributes of the 3-D files for the residual harmonic output
tsrgpars	Attributes of the output grids for time series output
tsrgrd	Attributes of the grid file for time series output containing coordinate data only
tsrstatlocs	Attributes of all output stations for time series output (locations, names)
tsrvars	Attributes of all variables for time series output
tsr0d	Attributes of the 0-D files for time series output
tsr2d	Attributes of the 2-D files for time series output
tsr3d	Attributes of the 3-D files for time series output

Parameters for monitoring

```
(MODULE iopars)
!---log files
LOGICAL :: exitlog
CHARACTER (LEN=*), PARAMETER :: logfmt1 = '(I1,'':'',A)', &
                              & logfmt2 = '(I2,'':'',A)'
CHARACTER (LEN=1), PARAMETER :: logexit = 'R'
CHARACTER (len=leniofile) :: inilog_file, runlog_file
INTEGER :: iolog = 0, loglev1, loglev2, pglev, runlog_count
INTEGER, DIMENSION(levprocs) :: levprocs_ini, levprocs_run
!---error files
LOGICAL :: errchk
CHARACTER (len=leniofile) :: errlog_file
INTEGER :: errstat, ioerr = 0, maxerrors, nerrs = 0
INTEGER, DIMENSION(npworld) :: levprocs_err
INTEGER, PARAMETER :: &
  & ierrno_fopen = 1, ierrno_fclose = 2, ierrno_read = 3, ierrno_write = 4,&
```

```
  & ierrno_fend = 5, ierrno_input = 6, ierrno_inival = 7,ierrno_runval = 8,&
  & ierrno_alloc = 9, ierrno_arg = 10, ierrno_comms = 11, ierrno_MPI = 12, &
  & ierrno_CDF = 13
CHARACTER (LEN=lenerrcode), PARAMETER, DIMENSION(MaxErrCodes) :: error_code
!---warning file
LOGICAL :: warnflag, warning
CHARACTER (len=leniofile) :: warlog_file
INTEGER :: iowarn = 0
!---timer
LOGICAL :: timer = .FALSE.
CHARACTER (len=leniofile) :: timing_file
INTEGER :: levtimer = 0, maxwaitsecs = 3600, nowaitsecs = 0, npcc_max, &
         & npcc_rate, timer_format = 1
INTEGER (KIND=kndilong), DIMENSION(MaxTimers) :: nopcc
INTEGER, PARAMETER :: &
  & itm_hydro = 1, itm_1dmode = 2, itm_2dmode = 3, itm_3dmode = 4, &
  & itm_dens = 5, itm_temp = 6, itm_sal = 7, itm_init = 8, &
  & itm_trans = 9, itm_adv = 10, itm_hdif = 11, itm_vdif = 12, &
  & itm_phgrad = 13, itm_input = 14, itm_output = 15, itm_inout = 16, &
  & itm_com_coll = 17, itm_com_comb = 18, itm_com_copy = 19, &
  & itm_com_dist = 20, itm_com_exch = 21, itm_com_util = 22, itm_coms = 23, &
  & itm_MPI = 24, itm_CDF = 25, itm_arrint = 26, itm_user = 27, &
  & itm_nest = 28, itm_libs = 29, itm_astro = 30, itm_bconds = 31, &
  & itm_meteo = 32, itm_structs = 33, itm_wait = 34, itm_sed = 35, itm_bio = 3
!---timer descriptions
CHARACTER (LEN=20), DIMENSION(MaxTimers) :: desctimer
```

File

iopars.f90

Type

Module

Purpose

Parameters for monitoring

Description

desctimer	Strings written to the timer report for each (active) timer
errchk	Enables/disables error checking
errlog_file	Default name of the “errlog” file (appended by the process id in the parallel case)

error_code	Error message string corresponding to an error code number
errstat	Error status number as returned by MPI, netCDF and FORTRAN calls (e.g. ALLOCATE statement)
exitlog	Enables/disables the writing of the "exit" message to the "log" file when the program exits a routine
ierrno_⋆	Key ids for error messages
ierrno_alloc	allocation error
ierrno_arg	missing or invalid routine argument
ierrno_CDF	netCDF error
ierrno_comms	parallel communication error
ierrno_fclose	an error occurred when a file is closed
ierrno_fend	end of file condition
ierrno_fopen	an error occurred when a file is opened
ierrno_inival	invalid value for a setup or initial scalar or array parameter
ierrno_input	an error occurred when reading an invalid value for a parameter or array from a data file
ierrno_MPI	MPI error
ierrno_read	read error
ierrno_runval	invalid value for a scalar or array parameter during the time loop
ierrno_write	write error
inilog_file	Default name of the "inilog" file (appended by the process id in the parallel case)
ioerr	File unit of the "errlog" file
iolog	File unit of the "inilog" and "runlog" files
iowarn	File unit of the warning file
itm_⋆	Timer key ids
itm_adv	advection routines
itm_arrint	interpolation of model grid arrays
itm_astro	astronomical tide
itm_bconds	boundary conditions

itm_bio	biology (currently not activated)
itm_CDF	total of all netCDF calls
itm_coms	total of all communication calls
itm_com_coll	collect communication calls
itm_com_comb	combine communications calls
itm_com_copy	copy communication calls
itm_com_dist	distribute communication calls
itm_com_exch	exchange communication calls
itm_com_util	(parallel) utility communication calls
itm_dens	total of density (including temperature and salinity) calculations
itm_hdif	horizontal diffusion
itm_hydro	hydrodynamics
itm_init	initialisation procedures
itm_inout	total of input and output operations
itm_input	input operations
itm_libs	calls to library routines
itm_meteo	meteorological routines
itm_MPI	total of all MPI calls
itm_nest	nesting procedures
itm_output	output operations
itm_phgrad	baroclinic pressure gradient
itm_sal	salinity
itm_sed	sediment model
itm_structs	structures
itm_temp	temperature
itm_trans	transport routines
itm_user	calls to usrdef routines
itm_vdif	vertical diffusion (including turbulence modules)
itm_wait	wait calls
itm_1dmode	water column mode (1-D) calculations
itm_2dmode	2-D mode calculations

itm_3dmode 3-D mode calculations

levprocs_err Level of error checking

0: error checking disabled

1: error checking enabled during the initialisation phase only

2: error checking enabled during the whole simulation

levprocs_ini Maximum program level for writing a tracer information message to the "inilog" file when a routine is entered. Its value is determined for each process separately.

levprocs_run Maximum program level for writing a tracer information message to the "runlog" file when a routine is entered. Its value is determined for each process separately.

levtimer Defines the information in the timer report

0: no timer report is written

1: only the execution time is written

2: Execution time and time information (in percentage of total time) is written for all "timers". In case of a parallel simulation, the percentages are given for the process with the largest amount of time, the lowest amount of time, an average value over all processes and for the master process.

3: As the previous case, but the time percentages are now given for each individual process in addition. In the serial case, behaviour is as for case 2.

logexit Exit message string written to the "log" file on exiting of a routine

logfmt1 Format specification for writing the tracer information to the "log" file

logfmt2 As logfmt1 now for program levels higher than 9

loglev1 Maximum program level for writing a tracer information message when a routine is entered

loglev2 Maximum program level for writing a tracer information message when a routine is exited

maxerrors Maximum number of messages written to the "errlog" file

maxwaitsecs	Maximum allowed time (in seconds) for suspension of the program
nerrs	Number of detected errors
nopcc	Current number of process clock counts stored for each timer
nowaitsecs	Number of seconds to suspend the execution of the program during a wait call
npcc_max	Maximum number of clock counts which can be returned by the process clock
npcc_rate	Number of process clock counts per second
pglev	Current program level
runlog_count	Determines the number of (2-D) time steps after which the log-file will be re-written
runlog_file	Default name of the "runlog" file (appended by the process id in the parallel case)
timer	.TRUE. if levtimer>0 (see above)
timer_format	Unit of the execution time in the timer report 1: seconds 2: minutes 3: hours 4: days
timing_file	Default name of the time report file
warlog_file	Default name of the "warning" file
warnflag	Enables/disables the writing of a warning ("warlog") file by the master process (.TRUE. on the master only if warning is .TRUE.)
warning	Enables/disables the writing of warning messages

netCDF parameters

```
(MODULE iopars)
INTEGER :: char_NF90 = 0, clobber_NF90 = 0, fill_NF90 = 0, global_NF90 = 0,&
         & int_NF90 = 0, noerr_NF90 = 0, nofill_NF90 = 0, nowrite_NF90 = 0,&
         & offset_64bit_NF90 = 0, real_NF90 = 0, share_NF90 = 0,&
         & sizehint_default_NF90 = 0, unlimited_NF90 = 0, write_NF90 = 0
```

File
iopars.f90

Type
Module

Purpose
Aliases for parameters of the netCDF library

Description

char_NF90	Alias for NF90_char
clobber_NF90	Alias for NF90_clobber
fill_NF90	Alias for NF90_fill
global_NF90	Alias for NF90_global
int_NF90	Alias for NF90_int
noerr_NF90	Alias for NF90_noerr
nofill_NF90	Alias for NF90_nofill
nowrite_NF90	Alias for NF90_nowrite
offset_64bit_NF90	Alias for NF90_offset_64bit
real_NF90	Alias for NF90_real
share_NF90	Alias for NF90_share
sizehint_default_NF90	Alias for NF90_sizehint_default
unlimited_NF90	Alias for NF90_unlimited
write_NF90	Alias for NF90_write

33.10 Meteorological arrays

```
MODULE meteo
REAL, DIMENSION(0:ncloc,0:nrloc) :: atmpres, uwindatc, vwindatc, &
                                    & airtemp, sst
REAL, DIMENSION(ncloc,nrloc) :: cloud_cover, evapminprec, &
                               & evaporation, precipitation, relhum
```

File
meteo.f90

Type
Module

Purpose

Meteorological data variables

Description

airtemp	Air temperature interpolated in time and in space at the C-nodes [^{0}C]
atmpres	Atmospheric pressure interpolated in time and in space at the C-nodes [Pa]
cloud_cover	Cloud cover interpolated in time and in space at the C-nodes (between 0 and 1)
evapminprec	Evaporation minus precipitation rate interpolated in time and in space at the C-nodes [kg/m^2/s]
evaporation	Evaporation rate interpolated in time and in space at the C-nodes [kg/m^2/s]
precipitation	Precipitation rate interpolated in time and in space at the C-nodes [kg/m^2/s]
relhum	Relative humidity interpolated in time and in space at the C-nodes (between 0 and 1)
sst	Sea surface temperature [^{0}C]
uwindatc	X-component of the wind at 10m height interpolated in time and in space at C-nodes [m/s]
vwindatc	Y-component of the wind at 10m height interpolated in time and in space at C-nodes [m/s]

33.11 Key ids of model variables

```
MODULE modids
!---model grid
INTEGER, PARAMETER :: &
& iarr_alphatc_fld = 1, iarr_alphatu_fld = 2, iarr_alphatv_fld = 3,&
....
```

File

modids.f90

Type

Module

Purpose

Definitions of key ids for model variables. The key id name has the form iarr_⋆ where ⋆ is the FORTRAN name of the variable.

33.12 Nested sub-grids

```
MODULE nestgrids
INTEGER :: nonestsets = 0
INTEGER, DIMENSION(nonestsets) :: nestcoords, nohnstglbc, &
                                  & nohnstglbu, nohnstglbv
INTEGER, DIMENSION(nonestsets) :: nohnstatc, nohnstatu, nohnstatv, novnst
INTEGER, DIMENSION(nonestsets) :: inst2dtype, lbhnstatc, lbhnstatu, lbhnstatv
INTEGER, DIMENSION(nonestsets) :: nobionst, nosednst
INTEGER, DIMENSION(nprocs,nonestsets) :: nohnstcprocs, nohnstuvprocs
INTEGER, DIMENSION(maxbiovars,nonestsets) :: instbio
INTEGER, DIMENSION(maxsedvars,nonestsets) :: instsed
INTEGER, DIMENSION(nprocs,MAXVAL(nohnstcprocs),nonestsets) :: indexnstc
INTEGER, DIMENSION(nprocs,MAXVAL(nohnstuvprocs),nonestsets) :: indexnstuv
TYPE (HRelativeCoords), DIMENSION(SUM(nohnstatc)) :: hnstctoc
TYPE (HRelativeCoords), DIMENSION(SUM(nohnstatu)) :: hnstctou, hnstutou
TYPE (HRelativeCoords), DIMENSION(SUM(nohnstatv)) :: hnstctov, hnstvtov
TYPE (VRelativeCoords), DIMENSION(2,2,SUM(nohnstatc),&
                                  & MAXVAL(novnst)) :: vnstctoc
TYPE (VRelativeCoords), DIMENSION(2,2,SUM(nohnstatu),&
                                  & MAXVAL(novnst)) :: vnstutou
TYPE (VRelativeCoords), DIMENSION(2,2,SUM(nohnstatv),&
                                  & MAXVAL(novnst)) :: vnstvtov
```

File

nestgrids.f90

Type

Module

Purpose

Arrays for the setup of sub-grid nesting

Description

hnstctoc	Relative horizontal coordinates of the C-node sub-grid points with respect to the local C-node main grid

hnstctou — Relative horizontal coordinates of the U-node sub-grid points with respect to the local C-node main grid

hnstctov — Relative horizontal coordinates of the V-node sub-grid points with respect to the local C-node main grid

hnstutou — Relative horizontal coordinates of the U-node sub-grid points with respect to the local U-node main grid

hnstvtov — Relative horizontal coordinates of the V-node sub-grid points with respect to the local V-node main grid

indexnstc — Index mapping array for C-node points. Element indexnstc(iproc,lsub,iset) equals lglb-l1+1 where iproc is the process number, lglb the corresponding index in the global indexing system and l1 the index in the global indexing system of the first global point in set iset

indexnstuv — Index mapping array for U- and V-points. Element indexnstuv(iproc,lsub,iset) equals lglb-l1+1 where iproc is the process number, lglb the index in the global indexing system and l1 the index in the global indexing system of the first global point in set iset. Note that U-points are counted before V-node points.

instbio — Variable indices of the biological state variables used for nesting per set

instsed — Fraction indices of the sediment concentrations used for nesting per set

inst2dtype — Type of data used for 2-D nesting

1: transports and elevations

2: elevations

3: transports

lbhnstatc — The C-node sub-grid points are indexed globally over all sub-grids. The element lbhnstatc(iset) represents the global index of the first point in the global indexing system which is located on the local sub-domain.

lbhnstatu — The U-node sub-grid points are indexed globally over all sub-grids. The element lbhnstatu(iset) represents the global index of the first point in the global indexing system which is located on the local sub-domain.

lbhnstatv — The V-node sub-grid points are indexed globally over all sub-grids. The element lbhnstatv(iset) represents the

	global index of the first point in the global indexing system which is located on the local sub-domain.
nestcoords	Type of coordinates used in the setup 1: absolute coordinates 2: relative coordinates
nobionst	Number of nested biological state variables per nested sub-grid
nohnstatc	Local number of C-node (horizontal) sub-grid open boundary points
nohnstatu	Local number of U-node (horizontal) sub-grid open boundary points
nohnstatv	Local number of V-node (horizontal) sub-grid open boundary points
nohnstcprocs	Number of C-node sub-grid points for each sub-domain and each sub-grid
nohnstglbc	Global number of C-node (horizontal) sub-grid open boundary points
nohnstglbu	Global number of U-node (horizontal) sub-grid open boundary points
nohnstglbv	Global number of V-node (horizontal) sub-grid open boundary points
nohnstuvprocs	Number of U- and V-node sub-grid points for each sub-domain and each sub-grid
nonestsets	Number of sub-grid nests
nosednst	Number of nested sediment fractions per nested sub-grid
novnst	Number of vertical levels on the sub-grid
vnstctoc	Relative vertical coordinates of the C-node sub-grid points with respect to the local main C-node grid taken at the four points surrounding each sub-grid point.
vnstctou	Relative vertical coordinates of the C-node sub-grid points with respect to the local main U-node grid taken at the four points surrounding each sub-grid point.
vnstctov	Relative vertical coordinates of the C-node sub-grid points with respect to the local main V-node grid taken at the four points surrounding each sub-grid point.

33.13 Open boundary conditions

```
MODULE obconds
!---2-D open boundary conditions
INTEGER, DIMENSION(nobu) :: iloczobu, ityp2dobu
INTEGER, DIMENSION(nobv) :: iloczobv, ityp2dobv
!nofiles = maxdatafiles(io_2uvobc,1)
INTEGER, DIMENSION(2:nofiles) :: iobc2dtype, no2dobc
INTEGER, DIMENSION(nobu+nobv,2:nofiles) :: index2dobc
REAL, DIMENSION(nobu,2) :: ud2obu, zetobu
REAL, DIMENSION(nobv,2) :: vd2obv, zetobv
REAL, DIMENSION(nobu,nconobc) :: ud2obu_amp, ud2obu_pha, &
                                 & zetobu_amp, zetobu_pha
REAL, DIMENSION(nobv,nconobc) :: vd2obv_amp, vd2obv_pha, &
                                 & zetobv_amp, zetobv_pha
REAL, DIMENSION(nobu,2) :: obc2uvatu, obc2uvatu_old
REAL, DIMENSION(nobv,2) :: obc2uvatv, obc2uvatv_old
!---3-D open boundary forcing
REAL, DIMENSION(nobu,nz,2) :: obc3uvatu
REAL, DIMENSION(nobv,nz,2) :: obc3uvatv
REAL, DIMENSION(nobu,nz,0:2) :: obcsalatu, obctmpatu
REAL, DIMENSION(nobv,nz,0:2) :: obcsalatv, obctmpatv
!---surface forcing (1-D application)
INTEGER :: isur1dtype
REAL :: gxslope = 0.0, gyslope = 0.0
REAL, DIMENSION(nconobc) :: gxslope_amp, gxslope_pha, gyslope_amp, &
                          & gyslope_pha, zeta_amp, zeta_pha
```

File

obconds.f90

Type

Module

Purpose

Parameters and arrays used to define open boundary conditions for the 2-D and 3-D mode or surface boundary conditions (surface slope and elevations) for 1-D applications

Description

gxslope	X-component of the (barotropic) pressure gradient in case of a 1-D application	[m/s^2]

gxslope_amp Amplitudes of the X-component of the (barotropic) pressure gradient in case of a 1-D application [m/s^2]

gxslope_pha Phases of the X-component of the (barotropic) pressure gradient in case of a 1-D application [rad]

gyslope Y-component of the (barotropic) pressure gradient in case of a 1-D application [m/s^2]

gyslope_amp Amplitudes of the Y-component of the (barotropic) pressure gradient in case of a 1-D application [m/s^2]

gyslope_pha Phases of the Y-component of the (barotropic) pressure gradient in case of a 1-D application [rad]

iloczobu If the elevation has to be specified at a U-open boundary, the array selects the position of the specified elevation with respect to the open boundary.

0: not required

1: at the open boundary U-node

2: at the "nearest" C-node outside the domain

iloczobv The same as iloczobu now at V-open boundaries

index2dobc Each 2-D open boundary data file contains a sub-set of open boundary data points. The element index2dobc(idat,ifil) maps, for file ifil, the local data point idat into a corresponding global open boundary index (between 1:nobu for U- and nobu+1:nobu+nobv for V-open boundaries). The physical size of the first dimension for file ifil equals no2dobc(ifil).

iobc2dtype Identifies the type of variables within each 2-D open boundary data file

1: depth-integrated currents and elevations

2: elevations only

3: depth-integrated currents only

isur1dtype Identifies the variables within a 1-D surface forcing data file.

1: components of the pressure gradient and elevation

2: surface elevation

3: components of the pressure gradient

ityp2dobu Type of open boundary conditions at the U-nodes (0/13)

0 : clamped

1 : zero slope

2 : zero volume flux

3 : specified elevation

4 : specified transport

5 : radiation condition using shallow water speed

6 : Orlanski (1976) condition

7 : Camerlengo & O'Brien (1980)

8 : Flather (1976) with specified elevation and transport

9 : Flather with specified elevation

10: Røed & Smedstad (1984)

11: characteristic method with specified elevation and transport

12: characteristic method with specified elevation

13: characteristic method using a zero normal gradient condition

ityp2dobv — Type of open boundary condition at the V-nodes. Meaning is the same as for ityp2dobu with U replaced by V and West/East by South/North.

no2dobc — Number of input open boundary locations for each 2-D open boundary data file

obcsalatu — Storage array for salinity S in case the open boundary conditions at the U-nodes require the solution of a differential equation in time

obcsalatv — Storage array for salinity S in case the open boundary conditions at the V-nodes require the solution of a differential equation in time

obctmpatu — Storage array for temperature T in case the open boundary conditions at the U-nodes require the solution of a differential equation in time

obctmpatv — Storage array for temperature T in case the open boundary conditions at the V-nodes require the solution of a differential equation in time

obc2uvatu	Storage array for the X-component of the transport U in case the open boundary conditions at the U-nodes require the solution of a differential equation in time
obc2uvatu_old	Value of obc2uvatu at the first (outer) iteration with the implicit scheme
obc2uvatv	Storage array for the Y-component of the transport V in case the open boundary conditions at the V-nodes require the solution of a differential equation in time
obc2uvatv_old	Value of obc2uvatv at the first (outer) iteration with the implicit scheme
obc3uvatu	Storage array for the X-component of the baroclinic current δu in case the open boundary conditions at the U-nodes require the solution of a differential equation in time
obc3uvatv	Storage array for the Y-component of the baroclinic current δv in case the open boundary conditions at the V-nodes require the solution of a differential equation in time
ud2obu	X-component of the transport U at U-open boundaries as given by (4.354). The external term term is stored in elements (*,1), the full expression (including the harmonic expansion) in elements (*,2) [m^2/s]
ud2obu_amp	Amplitudes A_n in the harmonic expansion of the X-component of the transport U at U-open boundaries [m^2/s]
ud2obu_pha	Phases φ in the harmonic expansion of the X-component of the transport U at U-open boundaries [rad]
vd2obv	Y-component of the transport V at V-open boundaries as given by (4.354). The external term term is stored in elements (*,1), the full expression (including the harmonic expansion) in elements (*,2) [m^2/s]
vd2obv_amp	Amplitudes A_n in the harmonic expansion of the Y-component of the transport V at V-open boundaries [m^2/s]
vd2obv_pha	Phases φ in the harmonic expansion of the Y-component of the transport V at V-open boundaries [rad]
zeta_amp	Amplitudes of the surface elevation ζ in case of a 1-D application [m]
zeta_pha	Phases of the surface elevation ζ in case of a 1-D application [rad]

zetobu	Surface elevation at U-open boundaries as given by (4.354). The external term term is stored in elements (*,1), the full expression (including the harmonic expansion) in elements (*,2) [m]
zetobu_amp	Amplitudes A_n in the harmonic expansion of the surface elevation ζ at U-open boundaries [m]
zetobu_pha	Phases φ in the harmonic expansion of the surface elevation ζ at U-open boundaries [rad]
zetobv	Surface elevation at V-open boundaries as given by (4.354). The external term term is stored in elements (*,1), the full expression (including the harmonic expansion) in elements (*,2) [m]
zetobv_amp	Amplitudes A_n in the harmonic expansion of the surface elevation ζ at V-open boundaries [m]
zetobv_pha	Phases φ in the harmonic expansion of the surface elevation ζ at V-open boundaries [rad]

33.14 Optical arrays

```
MODULE optics
REAL, DIMENSION(ncloc,nrloc) :: optattcoef2, qrad
REAL, DIMENSION(ncloc,nrloc,nz+1) :: radiance
```

File

optics.f90

Type

Module

Purpose

Optical arrays

Description

optattcoef2	Inverse optical attenuation depth for the short-wave spectrum [m^{-1}]
qrad	Solar downard irradiance within the water column [W/m^2]
radiance	Solar irradiance incident on the surface [W/m^2]

33.15 Parameters for parallel processing

```
MODULE paralpars
LOGICAL :: parallel_set = .FALSE.
LOGICAL :: master, spare, worker
INTEGER :: idmaster = 0, nprocs = 1, nprocsx = 0, nprocsy = 0
INTEGER :: comm_work, icoordloc, idloc, iprocloc, jcoordloc, npworld
INTEGER, DIMENSION(2*nprocs) :: comprocs
INTEGER, DIMENSION(npworld) :: idprocs
INTEGER, DIMENSION(nprocs) :: icoordprocs, jcoordprocs, ncprocs, nc1procs, &
                            & nc2procs, nobuprocs, nobvprocs, nobxprocs, &
                            & nobyprocs, nodisprocs, nosbuprocs, nosbvprocs, &
                            & nowbaruprocs, nowbarvprocs, nrprocs, nrvbuprocs, &
                            & nrvbvprocs, nr1procs, nr2procs
INTEGER, DIMENSION(0:nprocsx+1,0:nprocsy+1) :: iddomain
INTEGER, DIMENSION(nobu,nprocs) :: indexobuprocs
INTEGER, DIMENSION(nobv,nprocs) :: indexobvprocs
INTEGER, DIMENSION(nobx,nprocs) :: indexobxprocs
INTEGER, DIMENSION(noby,nprocs) :: indexobyprocs
INTEGER, DIMENSION(numdis,nprocs) :: indexdisprocs
INTEGER, DIMENSION(numwbaru,nprocs) :: indexwbaruprocs
INTEGER, DIMENSION(numwbarv,nprocs) :: indexwbarvprocs
TYPE(ExchComms), DIMENSION(MaxHaloComms) :: halocomms
!---MPI definitions
INTEGER :: bsend_overhead_MPI = 0, comm_null_MPI = 0, comm_world_MPI = 0, &
         & proc_null_MPI = 0, undefined_MPI = 0
```

File

paralpars.f90

Type

Module

Purpose

Parameters for parallel applications

Description

bsend_overhead_MPI	Alias for MPI_bsend_overhead
comm_null_MPI	Alias for MPI_comm_null
comm_work	MPI communicator containing all worker (non-idle) processes

comm_world_MPI	Alias for MPI_comm_world
comprocs	Array defining the rank order for all-to-all communications (see Section 11.4.3.1)
halocomms	Parameters for exchange (send/receive) communications
icoordloc	Domain index in the X-direction of the local process on the parallel grid
icoordprocs	Array with the values of icoordloc for each process
iddomain	Process ranks as defined on the parallel domain grid
idloc	Rank of the local process
idmaster	Rank of the master process
idprocs	Array of process ranks (global)
indexdisprocs	Index mapping array at discharge locations. The element indexdisprocs(lloc,iproc) maps the local discharge position index lloc on the local process iproc to the corresponding global discharge position index.
indexobuprocs	Index mapping array at the U-open boundaries. The element indexobuprocs(lloc,iproc) maps the local open boundary index lloc on the local process iproc to the corresponding global open boundary index.
indexobvprocs	Index mapping array at the V-open boundaries. The element indexobvprocs(lloc,iproc) maps the local open boundary index lloc on the local process iproc to the corresponding global open boundary index.
indexobxprocs	Index mapping array at the X-open boundaries. The element indexobxprocs(lloc,iproc) maps the local open boundary index lloc on the local process iproc to the corresponding global open boundary index.
indexobyprocs	Index mapping array at the Y-open boundaries. The element indexobyprocs(lloc,iproc) maps the local open boundary index lloc on the local process iproc to the corresponding global open boundary index.

indexwbaruprocs	Index mapping array at U-node weir/barrier locations. The element indexwbaruprocs(lloc,iproc) maps the local U-node weir/barrier position index lloc on the local process iproc to the corresponding global U-node weir/barrier position index.
indexwbarvprocs	Index mapping array at V-node weir/barrier locations. The element indexwbarvprocs(lloc,iproc) maps the local U-node weir/barrier position index lloc on the local process iproc to the corresponding global V-node weir/barrier position index.
iprocloc	Local process number (=idloc+1)
jcoordloc	Domain index in the Y-direction of the local process on the parallel grid
jcoordprocs	Array with the values of jcoordloc for each process
master	.TRUE. on the master process, .FALSE. otherwise
ncprocs	Array with the X-dimension ncloc of each sub-domain (global)
nc1procs	Array with the values of nc1loc for each sub-domain (global)
nc2procs	Array with the values of nc2loc for each sub-domain (global)
nobuprocs	Array with the values of nobuloc for each process
nobvprocs	Array with the values of nobvloc for each process
nobxprocs	Array with the values of nobxloc for each process
nobyprocs	Array with the values of nobyloc for each process
nodisprocs	Array with the values of numdisloc for each process
nosbuprocs	Array with the values of nosbuloc for each process
nosbvprocs	Array with the values of nosbvloc for each process
nowbaruprocs	Array with the values of numwbaruloc for each process
nowbarvprocs	Array with the values of numwbarvloc for each process
nprocs	Total number of active (“worker”) processes used in the simulation
nprocsx	X-dimension of the parallel grid

nprocsy	Y-dimension of the parallel grid
npworld	Total number of processes in MPI_comm_world. This equals the number given in the script used to launch the program
nrprocs	Array with the Y-dimension nrloc of each sub-domain (global)
nr1procs	Array with the values of nr1loc for each sub-domain (global)
nr2procs	Array with the values of nr2loc for each sub-domain (global)
nrvbuprocs	Array with the values of nrvbuloc for each process
nrvbvprocs	Array with the values of nrvbvloc for each process
parallel_set	Enables/disables the parallel mode. The switch is automatically set by the program if the compiler option -DMPI is defined in *coherensflags.cmp*.
proc_null_MPI	Alias for MPI_proc_null
spare	.TRUE. if the local process is an idle (spare) process which is currently inactive
undefined_MPI	Alias for MPI_undefined
worker	.TRUE. for an active on (worker) process. Is the same as .NOT.spare.

33.16 Physical and numerical model parameters

```
MODULE physpars
!---general
REAL :: Rearth = 6371000.0, rho_air = 1.2, specheat = 3987.5
!---reference and minimum values
REAL :: atmpres_ref = 101325.0, beta_sal_ref, beta_temp_ref, density_ref, &
      & dlat_ref = 0.0, dlon_ref = 0.0, dlon_ref_anal = 0.0, &
      & dlon_ref_obc = 0.0, gacc_mean, gacc_ref = real_undef, &
      & sal_ref = 33.0, sst_ref = 12.0, temp_min = 0.0, temp_ref = 12.0
!---model grid
REAL :: b_SH = 0.1, dl_BB = 1.5, du_BB = 1.5, hcrit_SH = 200.0, &
      & sigstar_DJ = 0.0, sig0_DJ = 0.1, theta_SH = 8.0
!---diffusion coefficients
```

```
REAL :: hdifmom_cst = 0.0, hdifscal_cst = 0.0, kinvisc_cst = 1.0E-06, &
      & smag_coef_mom = 0.1, smag_coef_scal = 0.1, vdifmom_cst = 1.0E-06, &
      & vdifscal_cst = 1.0E-06
!---water depths
REAL :: depmean_cst = 0.0, depmean_flag = 0.0
!---inundation schemes
INTEGER, PARAMETER :: nofldmasks = 11
REAL :: dcrit_fld = 0.1, dmin_fld = 0.02, dthd_fld = 0.1
INTEGER, DIMENSION(nofldmasks) :: fld_mask
!---bottom/surface fluxes
REAL :: bdragcoef_cst = 0.0, bdraglin = 0.0, ccharno = 0.014, &
      & cds_cst = 0.0013, ces_cst = 0.00113, chs_cst = 0.00113, ckar = 0.4, &
      & zbtoz0lim = 2.0, zref_atm = 10.0, zrough_cst = 0.0
!---relaxation distance for momentum advection
REAL :: distrlx_obc = 0.0
!---open boundary conditions
REAL :: cgravratio = 0.03
!---optical parameters
REAL :: optattcoef1_cst = 10.0, optattcoef2_cst = 0.067, opt_frac = 0.54
!---parameters for exchange coefficients in tabular form
INTEGER :: nrelhum, ntemp, nwind
REAL :: drelhum = 0.05, dtempdif = 1.0, dtempmax = 5.0, dtempmin = -5.0, &
      & dwind = 0.25, relhummax = 1.0, relhummin = 0.5, uwindmax = 50.0, &
      & uwindmin = 3.0
!---implicit code
INTEGER :: itsimp = 0, maxitsimp = 1, noitsimp
REAL :: dzetaresid, dzetaresid_conv = 1.0E-14, petsc_tol = 1.0E-07
!---numerical
REAL :: theta_cor = 0.5, theta_vadv = 0.501, theta_vdif = 1.0
REAL, PARAMETER :: eps_adv = 1.0E-12
```

File

physpars.f90

Type

Module

Purpose

Physical and numerical model parameters

Description

atmpres_ref	Reference atmospheric pressure P_{ref}	[Pa]

bdragcoef_cst	Constant bottom drag coefficient C_{db} when iopt_bstres_drag=1
bdraglin	Bottom friction velocity k_{lin} used in the linear bottom friction law if iopt_bstres_form=1 [m/s]
beta_sal_ref	Reference value for the salinity contraction coefficient β_S [PSU^{-1}]
beta_temp_ref	Reference value for the temperature expansion coefficient β_T [^{0}C^{-1}]
b_SH	Parameter b in the Song & Haidvogel (1994) vertical grid transformation
ccharno	Charnock's constant a used in Charnock's relation (4.291)
cds_cst	Constant surface drag coefficient C_{ds} when iopt_sflux_cds=0
ces_cst	Constant surface exchange coefficient C_e when iopt_sflux_cehs=0
cgravratio	Ratio of the internal to the external wave speed (used in the open boundary condition (4.382)
chs_cst	Constant surface exchange coefficient C_h when iopt_sflux_cehs=0
ckar	von Karman's constant κ
dcrit_fld	Critical water depth d_{crit} used in the drying/wetting algorithm [m]
density_ref	Reference density ρ_0 [kg/m^3]
depmean_cst	Constant water depth used to set up a default bathymetry [m]
depmean_flag	Data flag marking land points in the bathymetry [m]
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 case of a Cartesian grid [degrees]
dlon_ref	Reference longitude to be used in case of a Cartesian grid [degrees, positive East]
dlon_ref_anal	If iopt_astro_anal=1, the harmonically analysed phases are taken with respect to the astronomical argument for this reference longitude. [degrees, positive East]

dlon_ref_obc — If iopt_astro_pars>0, the phases at open boundaries are assumed to be taken with respect to the astronomical argument at this reference value. If zero, the reference longitude is taken at Greenwich. [degrees, positive East]

dl_BB — Parameter d_l in the Burchard & Bolding (2002) vertical grid transformation (4.26)

dmin_fld — Minimum water depth d_{min} used in the drying/wetting algorithm [m]

drelhum — Interval of relative humidities taken to calculate the surface exchange coefficients using the Monin-Obukhov theory in tabular form

dtempdif — Interval of air minus sea temperature differences taken to calculate the surface exchange coefficients using the Monin-Obukhov theory in tabular form $[^0C]$

dtempmax — Maximum air minus sea surface temperature difference taken to calculate the surface exchange coefficients using the Monin-Obukhov theory in tabular form $[^0C]$

dtempmin — Minimum air minus sea surface temperature difference taken to calculate the surface exchange coefficients using the Monin-Obukhov theory in tabular form $[^0C]$

dthd_fld — Threshold water depth d_{th} used in mask criteria for the inundation scheme [m]

dwind — Interval of wind speeds taken to calculate the surface exchange coefficients using the Monin-Obukhov theory in tabular form [m/s]

du_BB — Parameter d_u in the Burchard & Bolding (2002) vertical grid transformation (4.26)

dzetaresid — Value of $\|\zeta'\|_\infty$ (free surface corrector method). Its value is saved at the last iteration until the next time step.

dzetaresid_conv — Threshold value ϵ_{imp} used in the convergence criterium for the outer loop (free surface corrector method)

eps_adv — Tolerance factor for the calculation of the flux ratios with the TVD advection scheme

fld_mask	Enables (0) or disables (1) a specific mask criterium
gacc_mean	Mean acceleration of gravity • If gacc_ref is defined, gacc_mean=gacc_ref. • If gacc_ref is undefined and the grid is Cartesian, gacc_mean is defined by the geodetic formula (4.58) at the latitude given by dlat_ref. • If gacc_ref is undefined and the grid is spherical, gacc_mean is defined by the geodetic formula (4.58) applied at each C-node point and averaged over the physical domain.
gacc_ref	If different from real_fill, the acceleration of gravity is taken as horizontally uniform. Otherwise, g is evaluated as function of latitude using (4.58) $[m/s^2]$.
hcrit_SH	Parameter h_{crit} in the Song & Haidvogel (1994) vertical grid transformation
hdifmom_cst	Constant coefficient for horizontal momentum diffusion ν_H when iopt_hdif_coef=1 $[m^2/s]$
hdifscal_cst	Constant coefficient for horizontal scalar diffusion λ_H when iopt_hdif_coef=1 $[m^2/s]$
itsimp	Current iteration number for the outer loop of the implicit scheme
kinvisc_cst	Constant value for the kinematic viscosity $[m/s^2]$
maxitsimp	Largest allowed iteration number for the outer loop (free surface corrector method)
noitsimp	Last iteration number of the outer loop (free surface corrector method).
nofldmasks	Number of available mask criteria
nrelhum	Number of relative humidities taken to calculate the surface exchange coefficients using the Monin-Obukhov theory in tabular form
ntemp	Number of air minus sea temperature differences taken to calculate the surface exchange coefficients using the Monin-Obukhov theory in tabular form
nwind	Number of wind speeds taken to calculate the surface exchange coefficients using the Monin-Obukhov theory in tabular form

optattcoef1_cst	Inverse optical attenuation depth (λ_1^{-1}) for the absorption of long-wave solar radiation as used in (4.59)[m^{-1}]
optattcoef2_cst	Inverse optical attenuation depth (λ_2^{-1}) for the absorption of short-wave solar radiation as used in (4.59) [m^{-1}]
opt_frac	Long-wave fraction R of the surface solar radiance as used in (4.59)
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).
Rearth	Mean radius of the Earth [m]
relhummax	Maximum relative humidity taken to calculate the surface exchange coefficients using the Monin-Obukhov theory in tabular form
relhummin	Minimum relative humidity taken to calculate the surface exchange coefficients using the Monin-Obukhov theory in tabular form
rho_air	Air mass density ρ_a [kg/m^3]
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]
sigstar_DJ	Parameter σ_* in the Davies & Jones (1991) vertical grid transformations (4.23) and (4.24)
sig0_DJ	Parameter σ_0 in the Davies & Jones (1991) vertical grid transformations (4.23) and (4.24)
smag_coef_mom	Smagorinsky coefficient C_m for horizontal diffusion of momentum
smag_coef_scal	Smagorinsky coefficient C_s for horizontal diffusion of scalars
specheat	Specific heat of seawater c_p at constant pressure [$J/kg/^0C$]
sst_ref	Reference sea surface temperature [^0C]
temp_min	Minimum temperature. If set to real_fill, the minimum is taken as the freezing point of sea water which is a function of salinity. [^0C]
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 [^{0}C]
theta_cor	Implicity factor θ_c for the Coriolis term (between 0.0 and 1.0)
theta_SH	Parameter θ in the Song & Haidvogel (1994) vertical grid transformation
theta_vadv	Implicity factor θ_a for vertical advection (between 0.0 and 1.0)
theta_vdif	Implicity factor θ_d for vertical diffusion (between 0.0 and 1.0)
uwindmax	Maximum wind speed taken to calculate the surface exchange coefficients using the Monin-Obukhov theory in tabular form [m/s]
uwindmin	Minimum wind speed taken to calculate the surface exchange coefficients using the Monin-Obukhov theory in tabular form [m/s]
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]
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]
zbtoz0lim	Value of the limiting ratio ξ_{min} for z_b/z_0
zref_atm	Reference height z_a for meteorological variables (taken by default at 10 m height) [m]
zrough_cst	Constant bottom roughness length z_0 taken when iopt_bstres_drag=3 [m]
zbtoz0lim	Value of the limiting ratio z_b/z_0 []

33.17 Relaxation zones

```
MODULE relaxation
INTEGER :: norlxzones = 0
INTEGER, DIMENSION(2) :: inodesrlx = 0
INTEGER, DIMENSION(norlxzones) :: idirrlx, iposrlx, ityprlx, &
                                & jposrlx, ncrlx, nrrlx
INTEGER, DIMENSION(ncloc,nrloc,2) :: indexrlxatc, indexrlxatuv
```

```
REAL, DIMENSION(ncloc,nrloc,2) :: rlxwghtatc, rlxwghtatuv
```

File

relaxation.f90

Type

Module

Purpose

Definitions for application of relaxation conditions near open boundaries

Description

idirrlx — Defines the orientation of each zone

1: western boundary

2: eastern boundary

3: southern boundary

4: northern boundary

indexrlxatc — Each C-node grid point within a zone must have an associated boundary location. The array stores both the corresponding zonal index and the (global) index of the corresponding open boundary index. If the grid point is outside all zones, its value is 0. The last array index refers to either U-node (1) or V-node (2) boundaries.

indexrlxatuv — Each velocity node grid point within a zone must have an associated boundary location. The array stores both the corresponding zonal index and the (global) index of the corresponding open boundary index. If the grid point is outside all zones, its value is 0. The last array index refers to either U-node (1) or V-node (2) boundaries.

inodesrlx — The first element disables (0) or enables (1) the application of relaxation condition for C-node (scalar quantities), the second element for quantities at velocity (U- or V-) nodes.

iposrlx — (Global) X-index of the southwest corner of each relaxation zone

ityprlx — Type of interpolation (weighting) scheme within the relaxation zones

1: linear

2: quadratic

3: hyperbolic

jposrlx (Global) Y-index of the southwest corner of each relaxation zone

ncrlx Sizes of the zones (number of C- or velocity-node grid points) in the X-direction

norlxzones Number of relaxation zones

nrrlx Sizes of the zones (number of C- or velocity-node grid points) in the Y-direction

rlxwghtatc Weight factors used in the interpolation for C-node quantities. If the last index is 1(2), weighting is performed with respect to U-node (V-node) boundaries.

rlxwghtatuv Weight factors used in the interpolation for U-node or V-node quantities. If the last index is 1(2), weighting is performed with respect to U-node (V-node) boundaries.

33.18 Structures and discharges

```
!---dry cells
INTEGER :: numdry
INTEGER, ALLOCATABLE, DIMENSION(numdry) :: idry, jdry

!---thin dams
INTEGER :: numthinu, numthinuloc, numthinv, numthinvloc
INTEGER, ALLOCATABLE, DIMENSION(numthinu) :: ithinu, jthinu
INTEGER, ALLOCATABLE, DIMENSION(numthinv) :: ithinv, jthinv
INTEGER, ALLOCATABLE, DIMENSION(numthinuloc) :: ithinuloc, jthinuloc
INTEGER, ALLOCATABLE, DIMENSION(numthinvloc) :: ithinvloc, jthinvloc

!---weirs/barriers/orifices
INTEGER :: numwbaru, numwbaruloc, numwbarv, numwbarvloc
INTEGER, ALLOCATABLE, DIMENSION(numwbaru) :: iwbaru, jwbaru
INTEGER, ALLOCATABLE, DIMENSION(numwbarv) :: iwbarv, jwbarv
INTEGER, ALLOCATABLE, DIMENSION(numwbaruloc) :: indexwbaru, iwbaruloc, jwbaru
INTEGER, ALLOCATABLE, DIMENSION(numwbarvloc) :: indexwbarv, iwbarvloc, jwbarv
REAL :: wbarrlxu = 1.0, wbarrlxv = 1.0
REAL, ALLOCATABLE, DIMENSION(:) :: oricoefu, oricoefv, &
                                 & oriheightu, oriheightv, &
```

```
                                    & orisillu, orisillv
REAL, ALLOCATABLE, DIMENSION(:) :: wbarcoefu, wbarcoefv, wbarcrestu, &
                                    & wbarcrestv, wbarmodlu, wbarmodlv
REAL, ALLOCATABLE, DIMENSION(:) :: wbarelossu, wbarelossv

!---discharges
INTEGER :: numdis, numdisloc, numdisloc_ext
LOGICAL, ALLOCATABLE, DIMENSION(:) : disflag
INTEGER, ALLOCATABLE, DIMENSION(:) :: idis, idisloc, indexdisloc, jdis, &
                                       & jdisloc, kdis, kdistype, mdistype
REAL, ALLOCATABLE, DIMENSION(:) :: disarea, disdir, disspeed, disvol, &
                                    & xdiscoord, ydiscoord, zdiscoord
```

File

structures.f90

Type

Module

Purpose

Model switches

Reference

Chapter 18

Description

disarea	Area over which the discharge takes place	[m^2]
disdir	Discharge direction	[radian]
disflag	.TRUE./.FALSE. if a discharge locations is located on a wet/dry cell	
disspeed	Discharge speed defined as the volume discharge rate divided by the cell area	[m^2]
disvol	Volume discharge rate	[m^3/s]
idis	Global X-indices of discharge locations	
idisloc	Local X-indices of discharge locations	
idry	X-indices of dry cells	
indexdisloc	Global indices of the local discharge points	
indexwbaru	Global indices of the local U-node weirs/barriers	
indexwbarv	Global indices of the local V-node weirs/barriers	

ithinu	Global X-indices of thin dams at U-nodes
ithinuloc	Local X-indices of thin dams at U-nodes
ithinv	Global X-indices of thin dams at V-nodes
ithinvloc	Local X-indices of thin dams at V-nodes
iwbaru	Global X-indices of weirs/barriers at U-nodes
iwbaruloc	Local X-indices of weirs/barriers at U-nodes
iwbarv	Global X-indices of weirs/barriers at V-nodes
iwbarvloc	Local X-indices of weirs/barriers at V-nodes
jdis	Global Y-indices of discharge locations
jdisloc	Local Y-indices of discharge locations
jdry	Y-indices of dry cells
jthinu	Global Y-indices of thin dams at U-nodes
jthinuloc	Local Y-indices of thin dams at U-nodes
jthinv	Global Y-indices of thin dams at V-nodes
jthinvloc	Local Y-indices of thin dams at V-nodes
jwbaru	Global Y-indices of weirs/barriers at U-nodes
jwbaruloc	Local Y-indices of weirs/barriers at U-nodes
jwbarv	Global Y-indices of weirs/barriers at V-nodes
jwbarvloc	Local Y-indices of weirs/barriers at V-nodes
kdis	Vertical grid indices of the discharges locations. If zero, discharges are taken as homogeneous over the vertical
kdistype	Selects type of vertical location of the discharge 0: Uniformly distributed over the vertical 1: At the bottom 2: At the surface 3: At a fixed distance from the sea bed 4: At a fixed distance from the sea surface
mdistype	Selects method for flagging of discharge points located on dry cells 0: Locations on dry (C-node) cells are taken as invalid (default) 1: Locations on dry (C-node) cells are moved to the nearest neighbouring wet cell, provided such cell is available

	2: Locations on dry (C-node) cells are moved to the nearest wet cell	
numdis	Global number of discharge locations	
numdisloc	Local number of discharge locations	
numdisloc_ext	Local number of discharge locations including points within the first column of the western and first row of the southern halo	
numdry	Global number of dry cells	
numthinu	Global number of thin dams at U-nodes	
numthinuloc	Local number of thin dams at U-nodes	
numthinv	Global number of thin dams at V-nodes	
numthinvloc	Local number of thin dams at V-nodes	
numwbaru	Global number of weirs/barriers at U-nodes	
numwbaruloc	Local number of weirs/barriers at U-nodes	
numwbarv	Global number of weirs/barriers at V-nodes	
numwbarvloc	Local number of weirs/barriers at V-nodes	
oricoefu	Discharge coefficient for orifices at U-nodes	$[m^{1/2}/s]$
oricoefv	Discharge coefficient for orifices at V-nodes	$[m^{1/2}/s]$
oriheightu	Orifice width at U-nodes	[m]
oriheightv	Orifice width at V-nodes	[m]
orisillu	Orifice height at U-nodes	[m]
orisillv	Orifice height at V-nodes	[m]
wbarcoefu	Discharge coefficient for weirs/barriers at U-nodes	$[m^{1/2}/s]$
wbarcoefv	Discharge coefficient for weirs/barriers at V-nodes	$[m^{1/2}/s]$
wbarcrestu	Height of weir crest at U-nodes	[m]
wbarcrestv	Height of weir crest at V-nodes	[m]
wbarelossu	Energy loss sink term at U-node weirs/barriers	[1/s]
wbarelossv	Energy loss sink term at V-node weirs/barriers	[1/s]
wbarmodlu	Modular limit at U-node weirs/barriers	
wbarmodlv	Modular limit at V-node weirs/barriers	
wbarrlxu	Time relaxation coefficient at U-node weirs/barriers	
wbarrlxv	Time relaxation coefficient at V-node weirs/barriers	

xdiscoord	X-coordinates of discharge locations [m or degrees longitude]
ydiscoord	Y-coordinates of discharge locations [m or degrees latitude]
zdiscoord	Vertical coordinates (distance from sea bed or surface) of discharge location [m]

33.19 Model switches

```
MODULE switches
!---grid
INTEGER :: iopt_grid_htype = 1, iopt_grid_nodim = 3, iopt_grid_sph = 0, &
         & iopt_grid_vtype = 1, iopt_grid_vtype_transf = 0
!---interpolation
INTEGER :: iopt_arrint_hreg = 0, iopt_arrint_vreg = 0, iopt_arrint_3D = 0
!---hydrodynamics
INTEGER :: iopt_curr = 2, iopt_curr_wfall = 1, iopt_mode_2D, iopt_mode_3D
!---density
INTEGER :: iopt_dens = 0, iopt_dens_grad = 1, iopt_sal = 0, iopt_sal_sbc = 0, &
         & iopt_temp = 0, iopt_temp_optic = 1, iopt_temp_sbc = 1
!---external modules
INTEGER :: iopt_biolgy = 0, iopt_sed = 0
!---bottom stress
INTEGER :: iopt_bstres_drag = 3, iopt_bstres_form = 2, iopt_bstres_nodim = 3
!---transport
INTEGER :: iopt_transp_full = 0
!---advection
INTEGER :: iopt_adv_scal = 3, iopt_adv_turb = 0, iopt_adv_tvd = 1, &
         & iopt_adv_2D = 1, iopt_adv_3D = 1
!---diffusion
INTEGER ::  iopt_hdif_coef = 0, iopt_hdif_scal = 0, iopt_hdif_turb = 0, &
          & iopt_hdif_2D = 0, iopt_hdif_3D = 0, iopt_kinvisc = 0,
          & iopt_vdif_coef = 3
!---turbulence
INTEGER :: iopt_turb_alg = 1, iopt_turb_dis_bbc = 2, iopt_turb_dis_sbc = 2, &
         & iopt_turb_iwlim = 0, iopt_turb_kinvisc = 0, iopt_turb_lmix = 4, &
         & iopt_turb_ntrans = 1, iopt_turb_param = 2, iopt_turb_stab_form = 3,
         & iopt_turb_stab_lev = 1, iopt_turb_stab_mod = 4, &
         & iopt_turb_stab_tke = 2, iopt_turb_tke_bbc = 2, &
         & iopt_turb_tke_sbc = 2
```

```
!---drying/wetting
INTEGER :: iopt_fld = 0
!---structures
INTEGER :: iopt_dischr = 0, iopt_drycel = 0, iopt_thndam = 0, iopt_weibar = 0
!---structures
!---explicit/implicit integration
INTEGER :: iopt_cor_impl = 1, iopt_hydro_impl = 0, iopt_scal_depos = 1, &
         & iopt_vadv_impl = 1, iopt_vdif_impl = 2
!---open boundary conditions
INTEGER :: iopt_obc_advflux = 1, iopt_obc_advrlx = 0, iopt_obc_invbar = 0, &
         & iopt_obc_relax = 0, iopt_obc_sal = 0, iopt_obc_sed = 0, &
         & iopt_obc_bio = 0, iopt_obc_temp = 0, iopt_obc_2D = 0, &
         & iopt_obc_3D = 0
!---astronomical tide
INTEGER :: iopt_astro_anal = 0, iopt_astro_pars = 0, iopt_astro_tide = 0
!---1-D applications
INTEGER:: iopt_sur_1D = 0
!---meteo surface forcing
INTEGER :: iopt_meteo = 0, iopt_meteo_heat = 0, iopt_meteo_salflx = 0, &
         & iopt_meteo_stres = 0
!---surface fluxes
INTEGER :: iopt_sflux_cds = 0, iopt_sflux_cehs = 0, iopt_sflux_strat = 0
!---surface waves
INTEGER :: iopt_waves = 0
!---nesting
INTEGER :: iopt_nests = 0
!---parallel processing (MPI)
INTEGER :: iopt_MPI, iopt_MPI_abort = 0, iopt_MPI_comm_all = 2, &
         & iopt_MPI_comm_coll = 0, iopt_MPI_comm_exch = 2, &
         & iopt_MPI_comm_full = 0, iopt_MPI_comm_gath = 2, &
         & iopt_MPI_comm_scat = 2, iopt_MPI_partit = 1, iopt_MPI_sync = 0
!---output
INTEGER :: iopt_out_anal = 0, iopt_out_avrgd = 0, iopt_out_tsers = 1
!---netCDF
INTEGER :: iopt_CDF, iopt_CDF_abort = 0, iopt_CDF_fill = 0, &
         & iopt_CDF_format = 1
!---PETSC
INTEGER :: iopt_petsc, iopt_petsc_precond = 5, iopt_petsc_solver = 5
!---verification procedure
INTEGER :: iopt_verif
```

File
switches.f90

Type
Module

Purpose
Model switches

Reference
Section 14.4

Description

iopt_adv_scal — Type of scheme for the advection of scalar quantities
- 0: advection disabled
- 1: upwind scheme
- 2: Lax-Wendroff (explicit) in the horizontal, central (semi-implicit) in the vertical
- 3: TVD scheme

iopt_adv_turb — Type of scheme for the advection of turbulence quantities.
- 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 the TVD scheme.
- 1: superbee limiter
- 2: monotone limiter

iopt_adv_2D — Type of scheme for the advection of 2-D transports.
- 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. 0: advection disabled 1: upwind scheme 2: Lax-Wendroff (explicit) in the horizontal, central (semi-implicit) in the vertical 3: TVD scheme
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.
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.
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
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.
iopt_astro_pars	Enables or disables the inclusion of astronomical arguments and nodal corrections in the harmonic expansions (4.230) and (4.354). 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. This requires that the model uses a spherical grid (iopt_grid_sph=1).
iopt_biolgy	Disables/enables (0/1) the biological module. Currently not implemented.

iopt_bstres_drag	Formulation for the bottom drag coefficient C_{db}. 0: not used 1: spatially uniform value 2: spatially non-uniform 3: using a spatially uniform roughness length 4: using a specified, spatially non-uniform roughness length
iopt_bstres_form	Type of formulation for the bottom stress. 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. 2: depth-mean currents 3: 3-D current taken at the bottom grid cell
iopt_CDF	Disables/enables (0/1) the use of netCDF format. The format of all user output is set by default to netCDF ('N') if switched on. This switch cannot be defined by the user, but is set automatically when -DCDF is defined as CPP compiler option.
iopt_CDF_abort	Disables/enables (0/1) automatic abortion of the program when an error is detected whithin a netCDF call.
iopt_CDF_fill	Disables/enables (0/1) the use of fill values in netCDF files.
iopt_CDF_format	Selects the type netCDF file format. 1: classic format 2: 64-bit offset format
iopt_cor_impl	Time-integration of the Coriolis term. 0: explicit 1: semi-implicit 2: implicit
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: settling enabled without correction terms

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

iopt_dens — Evaluation of the density and expansion coefficients.

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 the numerical algorithm for the discretisation of the baroclinic pressure gradient.

0: gradient set to zero

1: traditional σ-coordinate (second order) method

2: z-level method

3: method of Shchepetkin & McWilliams (2003)

iopt_disch — Disables/enables (0/1) the discharge module.

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

iopt_fld — Selects the type of drying/wetting scheme.

0: Drying/wetting disabled

1: Drying/wetting algorithm without dynamic masks

2: Drying/wetting algorithm using dynamic masks

iopt_grid_htype	Type of horizontal grid. 1: uniform rectangular grid 2: non-uniform rectangular grid 3: curvilinear grid
iopt_grid_nodim	Grid dimension. 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: Cartesian coordinates 1: spherical coordinates
iopt_grid_vtype	Type of vertical grid. 1: uniform σ-grid 2: non-uniform σ-grid in the vertical, uniform in the horizontal 3: non-uniform σ-coordinate grid in the horizontal and the vertical
iopt_grid_vtype_transf	Type of vertical grid transformation. 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
iopt_hdif_coef	Type of scheme for horizontal diffusion coefficients. 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.

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

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

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

iopt_hydro_impl — Disables/enables the implicit scheme.

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

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

iopt_kinvisc — Formulation for kinematic viscosity.

0: user-defined uniform value kinvisc_cst

1: ITTC (1978) relation (7.24)

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

iopt_meteo_heat — Selects type of input data for the heat fluxes.

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: 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: no input

1: components of the wind speed (U_{10},V_{10}) and (unless iopt_grid_nodim=1) the atmospheric pressure P_a

2: components of the surface stress (τ_s^u,τ_s^v) and (unless iopt_grid_nodim=1) the atmospheric pressure P_a

iopt_mode_2D — Status of the 2-D mode. Its value is set internally and cannot be changed by the user.

0: The 2-D mode is disabled. Transports U, V and surface elevations ζ are set to their (zero) default values and are not updated.

1: Transports and elevation are initialised, but not updated in time

2: Transports and elevations are initialised and updated in time

iopt_mode_3D — Status of the 3-D mode. Its value is set internally and cannot be changed by the user.

0: The 3-D current are set to their default (zero) values and are not updated.

1: The 3-D current is initialised, but not updated in time.

2: The 3-D current is initialised and updated in time.

iopt_MPI — Disables/enables (0/1) the use of parallel communications. This switch cannot be defined by the user, but is set automatically when -DMPI is defined as CPP compiler option in *coherens-flags.cmp*.

iopt_MPI_abort — Disables/enables (0/1) automatic abortion of the program when an error is detected whithin a MPI call.

iopt_MPI_comm_all — Communication type for "all to all" operations.

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.

iopt_MPI_comm_exch Communication type for "exchange" operations.

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_full Disables/enables (0/1) the exchange of 4-D arrays. Currently not implemented.

iopt_MPI_comm_gath Communication type for "all to one" gather (combine) operations.

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.

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: "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.

iopt_nests	Disables/enables (0/1) the writing of open boundary data for nested sub-grids.
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. Currently not implemented. 0: default conditions at all open boundaries 1: non-default conditions for at least one open boundary point
iopt_obc_invbar	Disables/enables (0/1) the inverse barometric effect at open boundaries.
iopt_obc_relax	Disables/enables (0/1) the open boundary relaxation as discussed in Section 4.10.3.
iopt_obc_sal	(General) type of open boundary conditions for salinity. 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: 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: 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: 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 baroclinic currents.
	0: default conditions at all open boundaries
	1: non-default conditions for at least one open boundary point
iopt_out_anal	Disables/enables (0/1) harmonic output.
iopt_out_avrgd	Disables/enables (0/1) time averaged output.
iopt_out_tsers	Disables/enables (0/1) time series output.
iopt_petsc	Disables/enables PETSC (0/1). This switch is only defined internally when -DPETSC is provided and cannot be reset by the user.
iopt_petsc_precond	Type of preconditioner used by PETSc. 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. 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)
iopt_sal	Salinity update. 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: zero surface flux 1: surface flux given by (4.276)
iopt_scal_depos	Discretisation for the deposition (vertical advective flux at the sea bed) of particulate matter. 0: Deposition flux is set to zero. 1: first order (upwind) scheme 2: second order scheme using extrapolation
iopt_sed	Disables/enables (0/1) the activation of an external sediment module.
iopt_sflux_cds	Formulation for the neutral surface drag coefficient C_{ds}. 0: constant value as given by the parameter cds_cst

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: constant value as given by the parameter ces_cst or chs_cst

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 the surface drag and exchange coefficients on atmospheric stratification effects.

0: no dependence

1: using the Kondo (1975) parameterisation (Section 4.8.2)

2: using the Monin-Obukhov similarity theory (Section 4.8.3)

iopt_sur_1D — Disables/enables surface forcing (surface slopes and elevations) in case of a 1-D (iopt_grid_nodim=1) water column application.

iopt_temp — Temperature update.

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.

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: Neumann 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

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

iopt_transp_full — Selects how a series of tranport variables are updated in the transport routine. Currently not implemented.

0: Each equation is updated separately.

1: The equations are simultaneously updated.

iopt_turb_alg — Type of algebraic scheme if iopt_vdif_coef=2.

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 the bottom boundary condition for the dissipation rate ε.

1: Neumann condition (4.353)

2: Dirichlet condition (4.351)

iopt_turb_dis_sbc — Type of the surface boundary condition for the dissipation rate ε.

1: Neumann condition (4.284)

2: Dirichlet condition (4.281)

iopt_turb_iwlim	Type of the background mixing scheme as described in Section 4.4.3.6. 0: using uniform background coefficients 1: using limiting conditions for turbulence parameters giving the background limits (4.225)–(4.226) 2: the Large *et al.* (1994) scheme given by (4.227)
iopt_turb_kinvisc	Selects type of background mixing mixing. 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. 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. 0: zero-equation model (equilibrium or Mellor-Yamada level 2 method) 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 the type of second turbulent variable. 1: mixing length l (k-l scheme) 2: dissipation rate ε (k-ε scheme)
iopt_turb_stab_form	Selects the type of stability function. 1: constant value (4.198) 2: Munk-Anderson form (4.199) 3: from a RANS model as explained in Section 4.4.3.3
iopt_turb_stab_lev	Selects the level for stability functions if iopt_turb_stab_form= 3. 1: quasi-equilibrium method (Section 4.4.3.3)

2: non-equilibrium method (Section 4.4.3.3)

iopt_turb_stab_mod — Selects the type of closure (RANS) model.

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.

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

2: S_k is taken as proportional to the 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)

iopt_turb_tke_bcc — Type of the bottom boundary condition for turbulence energy.

1: Neumann condition (4.352)

2: Dirichlet condition (4.351)

iopt_turb_tke_sbc — Type of the surface boundary condition for turbulence energy.

1: Neumann condition (4.283)

2: Dirichlet condition (4.281)

iopt_vadv_impl — Time-integration for vertical advection

0: explicit

1: semi-implicit

2: implicit

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

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
iopt_vdif_impl	Time-integration for vertical diffusion.
	0: explicit
	1: semi-implicit
	2: implicit
iopt_verif	Disables/enables (0/1) the use of the verification procedure (not documented). This switch cannot be defined by the user, but is set automatically when -DVERIF is added as CPP compiler option in *coherensflags.cmp*. The only effect of the switch is that a different output is produced when running a pre-defined test case.
iopt_waves	Type of wave input wave input.
	0: wave input disabled
	1: wave height, period and wave direction
	2: wave height, period, velocity, excursion and direction
iopt_weibar	Disables/enables (0/1) the weir/barrier module.

33.20 Constants and system parameters

```
MODULE syspars
!--kind parameters
INTEGER, PARAMETER :: kndchar = KIND('A'), kndlog = KIND(.TRUE.), &
                    & kndint = KIND(1), &
                    & kndilong = MAX(4,SELECTED_INT_KIND(10)), &
                    & kndreal = KIND(1.0), kndlong = 8, kndcmplx = 8
!---data types
INTEGER, PARAMETER :: char_type = 1, log_type = 2, int_type = 3, &
                    & longint_type = 4, real_type = 5, long_type = 6, &
                    & cmplx_type = 7
!---universal parameters
REAL, PARAMETER :: pi = 3.14159265, halfpi = 1.57079633, twopi = 6.28318531
REAL (KIND=kndlong), PARAMETER :: pi_d = 3.1415926535897932_kndlong
REAL (KIND=kndlong), PARAMETER :: halfpi_d = 1.5707963267948966_kndlong
REAL (KIND=kndlong), PARAMETER :: twopi_d = 6.2831853071795865_kndlong
```

```
REAL, PARAMETER :: enap = 2.718282
REAL, PARAMETER :: degtorad = pi/180.0, radtodeg = 180.0/pi
REAL, PARAMETER :: degtorad_d = pi_d/180.0_kndlong, &
                 & radtodeg_d = 180.0_kndlong/pi_d
!---tidal parameters
INTEGER, PARAMETER :: MaxAstroTides = 56, MaxConstituents = 77
!---random generators
INTEGER, PARAMETER :: MaxGenerators = 32
!---MPI communications
INTEGER, PARAMETER :: MaxHaloComms = 8
!---model variables
INTEGER, PARAMETER :: MaxModArids = 454, MaxBioArids = 24, MaxSedArids = 51,
                    & MaxTotArids = MaxModArids + MaxBioArids + MaxSedArids
!---model I/O
INTEGER, PARAMETER :: MaxCIFTypes = 4, MaxCIFVars = 50, MaxGridTypes = 5, &
                    & MaxGridFiles = 2, MaxIOFiles = 33, MaxIOTypes = 39, &
                    & MaxProgLevels  = 20, MaxRestarts = 10
!---monitoring files and error coding
INTEGER, PARAMETER :: MaxErrCodes = 13, MaxErrMesgs = 50, MaxTimers = 36
!---character string lengths
INTEGER, PARAMETER :: lencifline = 300, lencifvar = 120, lendesc = 120, &
                    & lenerrcode = 120, lenformat = 120, lenfreq = 7, &
                    & leniofile = 120, lenname = 31, lennode = 3, &
                    & lentime = 23, lentitle = 20, lenunit = 60, &
                    & lenversion = 10

!---cif file
CHARACTER (LEN=1), PARAMETER :: cifcom = '!', cifend ='#', cifsep =','

!---user output
LOGICAL, PARAMETER :: DegreesOut = .TRUE.
CHARACTER (LEN=lenversion), PARAMETER :: model_version = 'V2.6'
!---output formats
CHARACTER (LEN=lenformat), PARAMETER :: IntegerFormat='(50I11)',&
                                      & RealFormat='(50G16.7)'
!---undefined and zero values
LOGICAL, PARAMETER :: log_undef = .FALSE.
INTEGER, PARAMETER :: int_undef = -2147483647, izero = 0
INTEGER (KIND=kndilong), PARAMETER :: izero_d = 0_kndilong, &
                                    & longint_undef = -2147483647_kndilong
REAL, PARAMETER ::  real_undef = -9.9692099683868690E+36, &
```

```
                  & real_flag = -9.9692099683868690E+35, rzero = 0.0
REAL (KIND=kndlong), PARAMETER :: rzero_d = 0.0_kndlong
CHARACTER (LEN=lentime), PARAMETER :: cdatetime_undef = &
                                    & 'xxxx/xx/xx;00:00:00,000'
```

File

syspars.f90

Type

Module

Purpose

Constants and other system parameters. The Max* parameters are mostly intended for the dimensioning of arrays which cannot be declared as allocatable.

Description

DegreesOut	Determines the unit of phases and angles in user-defined output files .TRUE. degrees .FALSE. radians
IntegerFormat	Format string used for reading/writing integer data from/to a ASCII ('A') file in standard COHERENS format
MaxAstroTides	Maximum allowed number of constituents for the astronomical forcing
MaxBioArids	Maximum available number of key ids for biological array variables.
MaxCIFTypes	Maximum number of CIF files
MaxCIFVars	Maximum number of data variables on a CIF line
MaxConstituents	Maximum allowed number of tidal constituents at open boundaries
MaxErrCodes	Maximum number of key ids for error messages
MaxErrMesgs	Default maximum number of error messages
MaxGenerators	Maximum available number of random generators which can be simultaneously used in the program
MaxGridFiles	Maximum number of surface grid files for each grid type. Current value is 1.

MaxGridTypes	Maximum number of surface grid types
MaxHaloComms	Maximum available number of exchange communications (send or receive)
MaxIOFiles	Maximum number of forcing files for a given file descriptor key id.
MaxIOTypes	Maximum number of file descriptor key ids.
MaxModArids	Maximum available number of key ids for physical model array variables
MaxProgLevels	Maximum number of subprogram levels
MaxRestarts	Maximum number of output times for the writing of restart conditions
MaxSedArids	Maximum available number of key ids for sediment array variables.
MaxTimers	Maximum available number of timers
MaxTotArids	Maximum available number of key ids for all model arrays
RealFormat	Format string used for reading/writing real data from/to a ASCII ('A') file in standard COHERENS format
cdatetime_undef	Flag for an undefined date/time string
char_type	Type parameter for character variables
cifcom	Comment character on a CIF line
cifend	Character marking the end of a data block in a CIF
cifsep	Data separator character within a CIF line
cmplx_type	Type parameter for complex variables
degtorad	Factor to convert degrees to radians
degtorad_d	Factor to convert degrees to radians in double precision
enap	Euler's number $e = 2.718282$
halfpi	The number $\pi/2$
halfpi_d	The number $\pi/2$ in double precision
int_type	Type parameter for integer variables
int_undef	Flag for undefined or invalid integer values
izero	The number zero in single precision
izero_d	The number zero in long integer format

kndchar	Kind parameter for character variables
kndcmplx	Kind parameter for complex variables
kndilong	Kind parameter for long integer variables
kndint	Kind parameter for integer variables
kndlog	Kind parameter for logical variables
kndlong	Kind parameter for double precision real variables
kndreal	Kind parameter for real variables
lencifline	Maximum length of a data line in a CIF
lencifvar	Maximum length of a CIF data value in string format
lendesc	Maximum length of a long_name attribute
lenerrcode	Maximum length of an error code message
lenformat	Maximum length of a string format specification
lenfreq	Maximim length of a frequency name
leniofile	Maximum length for the name of a file in standard COHERENS format
lenname	Maximum length of a f90_name attribute
lennode	Maximum length of a node attribute
lentime	Length of a date/time string
lentitle	Maximum length of a simulation title
lenunit	Maximum length of the units attribute
lenversion	Maximum length of the model_version string
log_type	Type parameter for logical variables
log_undef	Flag for undefined or invalid logical data
longint_type	Type parameter for a long integer variable
longint_undef	Flag for undefined or invalid long integer values
long_type	Type parameter for double precision real variables
model_version	Current COHERENS version number in string format
pi	The number π
pi_d	The number π in double precision
radtodeg	Factor to convert radians to degrees
radtodeg_d	Factor to convert radians to degrees in double precision

real_flag	Flag below which input data are considered as invalid (default value). Actual parameter used in the program is real_min.
real_type	Type parameter for real variables
real_undef	Default flag for undefined or invalid real values. Actual parameter used in the program is real_fill.
rzero	The number zero in single precision real format
rzero_d	The number zero in double precision real format
twopi	The number 2π
twopi_d	The number 2π in double precision

33.21 Tidal forcing

```
MODULE tide
!---number of tidal constituents
INTEGER :: nconastro = 0, nconobc = 0
!---tidal indices
INTEGER, DIMENSION(MaxAstroTides) :: index_astro = 0
INTEGER, DIMENSION(MaxConstituents) :: index_obc = 0
!---nodal factors
REAL, DIMENSION(nconastro) :: fnode_astro
REAL, DIMENSION(nconobc) :: fnode_obc
!---tidal phases
REAL, DIMENSION(nconastro) :: phase_astro
REAL, DIMENSION(nconobc) :: phase_obc
!---tidal force
REAL, DIMENSION(ncloc,nrloc) :: fxastro, fyastro
!---key ids for tidal constituents
INTEGER, PARAMETER ::&
    & icon_MS0   =  1, icon_Sa    =  2, icon_Ssa   =  3, icon_058    =  4,&
    ...
!---tidal species index
INTEGER, DIMENSION(MaxConstituents) :: ispec_tides
DATA ispec_tides /12*0, 21*1, 22*2, 1*3, 1*2, 4*3, 5*4, 6*6, 5*8/
!---amplitudes of the equilibrium tide
REAL, DIMENSION(MaxAstroTides) :: astro_ampl = &
 & (/0.198419, 0.003103, 0.019542, 0.001142, 0.004239, 0.022191, 0.003677, &
 ..., 0.003455/)
!---amplitude corrections for the earth tide
```

```
REAL, DIMENSION(MaxAstroTides) :: astro_earth
DATA astro_earth /15*0.693, 0.6946, 0.6948, 0.6950, 0.6956, 0.693, 0.6962, &
 ...,  24*0.693/
!---tidal frequencies
REAL, DIMENSION(MaxConstituents) :: tidal_spectrum = &
&(/0.0000000E+00, 1.9909688E-07, 3.9821277E-07, 5.9730965E-07, &
 & ..., 5.8177642E-04/)
!---names of tidal frequencies
CHARACTER (LEN=lenfreq), DIMENSION(MaxConstituents) :: tidal_freq_names = &
    & (/'MS0    ' ,'Sa     ' ,'Ssa    ', '058    ', 'Msm    ', 'Mm     ',&
    &   ..., 'S8     '/)
```

File

tide.f90

Type

Module

Purpose

Parameters and arrays for tidal forcing

Description

astro_ampl	Amplitudes of the astronomical equilibrium tide [m]
astro_earth	Elasticity factor α_{qn} representing the effect of the Earth tides on the tidal force
fnode_astro	Nodal factors of the constituents used for the astronomical tidal force
fnode_obc	Nodal factors of the tidal constituents used at open boundaries
fxastro	X-component of the astronomical tidal force [m^2/s]
fyastro	Y-component of the astronomical tidal force [m^2/s]
icon_$\star$	Key ids of all tidal constituents
index_astro	Key ids of the constituents used in the calulcation of the astronomical force
index_obc	Key ids of the constituents used at the open boundaries
ispec_tides	Tidal species indices q for each frequency
nconastro	Number of tidal constituents used in the calculation of the astronomical force

nconobc	Number of tidal constituents used at open boundaries
phase_astro	Astronomical phases of the constituents used for the astronomical tidal force, at Greenwhich. Nodal corrections are included if iopt_astro_pars=2. [rad]
phase_obc	Astronomical phases of the constituents used at open boundaries with respect to the reference longitude dlon_ref_obc. Nodal corrections are included if iopt_astro_pars=2. [rad]
tidal_freq_names	Names of all pre-defined tidal frequencies
tidal_spectrum	Frequencies of all pre-defined tidal constituents [rad/s]

33.22 Time parameters

```
MODULE timepars
LOGICAL :: corrstep, metstepin, predstep, physinit, wavestep
CHARACTER (LEN=lentime) :: CDateTime, CEndDateTime = cdatetime_undef,&
                         & CStartDateTime = cdatetime_undef, ClockTime
INTEGER :: julianday, norestarts = 0, nstep = 0, nt = 0, ntobcrlx = 0
INTEGER :: ic3d = 1, icnodal = 0
INTEGER (KIND=kndilong) :: nosecsrun = 0
REAL :: delt2d, delt3d, time_zone = 0.0
INTEGER, DIMENSION(6) :: time_convert = (/1,60,3600,86400,2629800,31557600/)
INTEGER, DIMENSION(7) :: IDateTime, IEndDateTime, IStartDateTime
INTEGER, DIMENSION(12) :: days_in_month = &
                        & (/31,28,31,30,31,30,31,31,30,31,30,31/)
INTEGER, DIMENSION(13) :: monthdays = &
                        & (/0,31,59,90,120,151,181,212,243,273,304,334,365/)
INTEGER, DIMENSION(MaxRestarts) :: ntrestart = 0
```

File

timepars.f90

Type

Module

Purpose

Time parameters

Description

CDateTime	Current date and time in string format

CEndDateTime	End date and time in string format
CStartDateTime	Start date and time in string format
ClockTime	Date and time from the machine's internal real-time clock at the start of the simulation
IDateTime	Current date and time in integer vector format
IEndDateTime	End date and time in integer vector format
IStartDateTime	Start date and time in integer vector format
corrstep	Set to .TRUE. at the corrector time steps
days_in_month	Number of days in each month (without correction for leap years)
delt2d	Barotropic (2-D) time step (mode-splitting scheme) or time step for all transport equations (implicit scheme) [s]
delt3d	Baroclinic (3-D) time step, equal to delt2d in case of an implicit scheme [s]
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.
ic3d	Ratio of the 3-D (baroclinic) and 2-D time steps
julianday	Julian day (year day between 1 and 365 or 366)
metstepin	Set to .TRUE. at time steps when an update of surface fluxes is required.
monthdays	Day number of the first day in each month (without correction for leap years)
norestarts	Number of times for the writing of re-start initial conditions
nosecsrun	Number of seconds since the start of the simulation
nstep	Number of 2-D time steps
nt	Time index (time since the start of the simulation divided by delt2d)
ntobcrlx	The relaxation period T_r, divided by the 2-D time step delt2d, (optionally) used to define the relaxation factor $\alpha_r(t)$ (see equation (4.356)) for the 2-D mode at open boundaries.

ntrestart	The times during a simulation, measured in time indices, when re-start conditions have to be written
predstep	Set to .TRUE. at the predictor time steps
physinit	et to .TRUE. for (re)initialisation of physical conditions
time_convert	Factors for converting the time in seconds to another unit
time_zone	Time zone, i.e. the difference of the local time with respect to GMT. Difference is positive (negative) eastwards (westwards) from Greenwich. [hours]
wavestep	Set to .TRUE. at time steps when an update of surface wave data is required.

33.23 Turbulence model parameters

```
MODULE turbpars
!---stability functions
INTEGER :: ib22
REAL :: cfequil(5), cfstabtke(3), cfstab1(8), cfstab2(11)
REAL :: c_sk = 0.15, keps0, f0stabmom, f0stabscal, f0stabtke, &
      & skeps = 0.09, sq_my = 0.2
!---limiting conditions
REAL :: alphaM_max, alphaN_max, alphaN_min, alphaN_sl, &
      & dissipmin = 1.0E-012, tkelim = 1.0E-06, tkemin = 1.0E-014, &
      & zlmixmin = 1.7E-010
!---t.k.e.-equation
REAL :: sigma_k = 1.0, wfltke = 0.0
!---kl-equation
REAL :: e1_my = 1.8, e2_my = 1.33, e3_my = 1.0, sigma_kl
!---eps-equation
REAL :: c1_eps = 1.44, c2_eps = 1.92, c31_eps = 0.2, c32_eps = 1.0, sigma_eps
!---roughness lengths
REAL :: zrough_bot = 0.0, zrough_sur = 0.0
!---KPP scheme for internal waves
REAL :: riccrit_iw = 0.7, vdifmom_iw = 1.0E-04, vdifscal_iw = 5.0E-05, &
      & vdifshear_iw = 0.005
!---Pacanowski-Philander relations
REAL :: alpha_pp = 5.0, expmom_pp = 2.0, vbmom_pp = 1.0E-04, &
      & vbscal_pp = 1.0E-05, vmax_pp = 3.0, v0dif_pp = 0.01
```

```
!---Munk-Anderson relations
REAL :: alpha_ma = 10.0, beta_ma = 3.33, expmom_ma = 0.5, &
      & expscal_ma = 1.5, & vmaxmom_ma = 3.0, vmaxscal_ma = 4.0, v0dif_ma = 0.06
!---algebraic flow-dependent relations
REAL :: cnu_ad = 2.0, delta1_ad = 0.0, delta2_ad = 0.0, k1_ad =0.0025, &
      & k2_ad = 2.0E-05, lambda_ad = 0.0, omega1_ad = 1.0E-04, &
      & r1_ad = 1.0, r2_ad = 1.0
!---algebraic mixing length formulations
REAL :: alpha_Black = 0.2, beta_Xing = 2.0
```

File

turbpars.f90

Type

Module

Purpose

Parameters for turbulence schemes

Description

alpha_Black	constant α_1 in the Blackadar (1962) mixing length formulation (4.217)
alphaM_max	maximum value for the stability parameter α_M in case of stable stratification when the the non-equilibrium method is applied
alphaN_max	maximum value for the stability parameter α_N in case of stable stratification
alphaN_min	minimum value for the stability parameter α_N in case of unstable stratification
alphaN_sl	slope of the limiting curve for α_N in the case of a stable stratification
alpha_ma	parameter α_m in the Munk & Anderson (1948) scheme (4.136)–(4.139)
alpha_pp	parameter α_p in the Pacanowski & Philander (1981) scheme (4.132)–(4.134)
beta_ma	parameter β_m in the Munk & Anderson (1948) scheme (4.136)–(4.139)
beta_Xing	attenuation factor β_1 in the Xing & Davies (1996) mixing length formulation (4.215)

cfequil	coefficients C_d used in the equilibrium formulation (4.194–4.195) for the stability functions
cfstabtke	coefficients used in the non-equilibrium (4.185) or the quasi-equilibrium (4.191) method for the stability function S_k
cfstab1	coefficients C_b or C_c used in the quasi-equilibrium formulation (4.187) or (4.190) for the stability functions
cfstab2	coefficients C_a used in the non-equilibrium formulation (4.182) for the stability functions
cnu_ad	parameter C_ν in equation (4.151)
c1_eps	constant $c_{1\varepsilon}$ in the shear production term of the ε-equation (4.205)
c2_eps	constant $c_{2\varepsilon}$ in the dissipation term of the ε-equation (4.205)
c31_eps	constant $c_{3\varepsilon}$ in the buoyance sink term of the ε-equation (4.205) in case of stable stratification ($N^2 > 0$)
c32_eps	constant $c_{3\varepsilon}$ in the buoyancy source term of the ε-equation (4.205) in case of unstable stratification ($N^2 < 0$)
c_sk	Daly-Harlow parameter c_{sk} in (4.177)
delta1_ad	parameter δ_1 in equation (4.143)
delta2_ad	parameter δ_2 in equation (4.143)
dissipmin	numerical lower limit ε_{min} for ε [W/kg]
eps0	parameter ϵ_0 as defined by (4.282)
expmom_ma	parameter n_1 in the Munk & Anderson (1948) scheme (4.136)–(4.139)
expmom_pp	parameter n_p in the Pacanowski & Philander (1981) scheme (4.132)–(4.134)
expscal_ma	parameter n_2 in the Munk & Anderson (1948) scheme (4.136)–(4.139)
e1_my	constant E_1 in the shear production term of the kl-equation (4.209)
e2_my	constant E_2 in the wall proximity term (4.210) of the kl-equation (4.209)
e3_my	constant E_3 in the buocancy source/sink term of the kl-equation (4.209)

f0stabmom	neutral stability coefficient S_{u0} for momentum (quasi-equilibrium or equilibrium method)	
f0stabscal	neutral stability coefficient S_{b0} for scalars (quasi-equilibrium or equilibrium method)	
f0stabtke	constant stability coefficient S_{k0} for turbulent kinetic energy	
ib22	Equals 1 when the parameter $c_{22\beta} \neq 0$, 0 otherwise	
k1_ad	parameter K_1 in equations (4.148) and (4.150)	
k2_ad	parameter K_2 in equation (4.149)	
lambda_ad	parameter $\lambda_\star$ in equation (4.146)	[m]
omega1_ad	parameter ω_1 in equation (4.151)	[s^{-1}]
riccrit_iw	critical Richardson number Ri_0 in the Large *et al.* (1994) background mixing scheme (4.227)	
r1_ad	parameter r_1 in equation (4.143)	
r2_ad	parameter r_2 in equation (4.143)	
sigma_eps	the parameter σ_ε as obtained from (4.206)	
sigma_k	parameter σ_k used to define S_k in (4.201)	
sigma_kl	the parameter σ_{kl} obtained from (4.211)	
skeps	neutral value S_{k0} of the stability coefficient S_k in the k-ε model (see equation (4.200))	
sq_my	parameter S_q used to determine S_{k0} in the Mellor-Yamada model (see equation (4.202))	
tkelim	background limit k_{lim} for k (see equation (4.226))	[J/kg]
tkemin	numerical lower limit k_{min} for k	[J/kg]
vbmom_pp	parameter ν_{bp} in the Pacanowski & Philander (1981) scheme (4.132)–(4.134)	[m^2/s]
vbscal_pp	parameter λ_{bp} in the Pacanowski & Philander (1981) scheme (4.132)–(4.134)	[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)	[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)	[m^2/s]

vdifshear_iw	maximum mixing due to resolved vertical shear ν_0^s in the Large *et al.* (1994) background mixing scheme (4.227) [m^2/s]
vmaxmom_ma	parameter ν_{max} in the Munk & Anderson (1948) scheme (4.136)–(4.139)
vmaxscal_ma	parameter λ_{max} in the Munk & Anderson (1948) scheme (4.136)–(4.139)
vmax_pp	parameter ν_{max} in the Pacanowski & Philander (1981) scheme (4.132)–(4.134)
v0dif_ma	parameter ν_{0m} in the Munk & Anderson (1948) scheme (4.136)–(4.139) [m^2/s]
v0dif_pp	parameter ν_{0p} in the Pacanowski & Philander (1981) scheme (4.132)–(4.134) [m^2/s]
wfltke	surface wave factor c_w used in the surface flux condition (4.283) for turbulent energy
zlmixmin	numerical lower limit l_{min} for l [m]
zrough_bot	bottom roughness length z_{0b} in the mixing length formulation (4.212) [m]
zrough_sur	surface roughness length z_{0s} in the mixing length formulation (4.212) [m]

33.24 Turbulence arrays

```
MODULE turbulence
REAL, DIMENSION(1-nhalo:ncloc+nhalo,1-nhalo:nrloc+nhalo,nz+1) :: &
              & dissip, tke, tke_old, zlmix
REAL, DIMENSION(ncloc,nrloc,nz+1) :: buofreq2, shearfreq2
```

File

turbulence.f90

Type

Module

Purpose

Turbulence arrays

Description

buofreq2	Squared buoyancy frequency N^2 [$1/s^2$]

dissip	Dissipation of turbulent kinetic energy	[W/kg]
tke	Turbulent kinetic energy	[J/kg]
tke_old	Turbulent kinetic energy at the old 3-D time step	[J/kg]
shearfreq2	Squared shear frequency M^2	[1/s^2]
zlmix	Turbulent mixing length	[m]

33.25 Surface wave arrays

```
MODULE waves
REAL, ALLOCATABLE, DIMENSION(ncloc,nrloc) :: wavedir, wavefreq
REAL, ALLOCATABLE, DIMENSION(0:ncloc,0:nrloc) :: waveexcurs, waveheight, &
                                & wavenum, waveperiod, waveuvel, wavevel, wavevvel
```

File

waves.f90

Type

Module

Purpose

Surface wave arrays

Description

wavedir	Wave direction	[rad]
wavexcurs	Near-bottom wave excursion amplitude	[m]
wavefreq	Peak wave frequency	[rad/s]
waveheight	Significant wave height	[m]
wavenum	Wave number	[1/m]
waveperiod	Peak wave period at	[s]
waveuvel	Near-bottom wave orbital velocity in the X-direction	[m/s]
wavevel	Near-bottom wave orbital velocity	[m/s]
wavevvel	Near-bottom wave orbital velocity in the Y-direction	[m/s]