Chapter 17

Surface forcing and nesting

This chapter discusses the setup of the surface forcing needed for the application of surface boundary conditions and the procedures for defining nesting. The following routines are described in the next sections:

- usrdef_1dsur_spec: specifies the setup of boundary forcing for water column applications (1-D mode)
- usrdef_1dsur_data: defines the input of surface forcing data for water column applications
- usrdef_surface_absgrd: defines a surface data grid using absolute coordinates
- usrdef_surface_relgrd: defines a surface data grid using relative coordinates
- usrdef_surface_data: defines the input of surface data for a specific data grid
- usrdef_nstgrd_spec: general definitions for nesting (e.g. number of sub-grid data points)
- usrdef_nstgrd_abs: geographical (absolute) positions of the sub-grid (nested) data points
- usrdef_nstgrd_rel: relative coordinates of the sub-grid (nested) data points with respect to the model grid

The first two routines are contained in *Usrdef_Model.f90*, the next three in *Usrdef_Surface_Data.f90* and the last three in *Usrdef_Nested_Grids.f90* (see Table 13.1).

17.1 Water column surface forcing

The routines below are only used for 1-D applications (iopt_grid_nodim=1).

17.1.1 Surface forcing specifiers for the 1-D mode

Contrary to the 2-D and 3-D case, the surface elevation and the horizontal pressure gradient are in the case of a water column application (iopt_grid_nodim= 1) not calculated by the model but are considered as an external forcing at the surface instead. The subroutine usrdef_1dsur_spec describes specifier arrays for this water column surface forcing and is called if iopt_sur_1D=1 and modfiles(io_1uvsur,1,1)%status ='N'. The following arrays can be defined here:

gxslope_amp(nconobc) amplitudes of the X-component of the pressure gradient divided by ρ_0 [m^2/s]

gxslope_pha(nconobc) phases of the X-component of the pressure gradient [rad]

gyslope_amp(nconobc) amplitudes of the Y-component of the pressure gradient divided by ρ_0 [m^2/s]

gyslope_pha(nconobc) phases of the Y-component of the pressure gradient [rad]

zetadat_amp(nconobc) amplitudes of the surface elevation [m]

zetadat_pha(nconobc) phases of the surface elevation [rad]

isur1dtype selects the type of variables if they are supplied from an external data file, i.e. when (modfiles(io_1uvsur,2,1)%status = 'N' or 'R')

1: components of the pressure gradient and elevation

2: surface elevation

3: components of the pressure gradient

The size nconobc of the arrays denotes, in this case, the number of tidal constituents. Note also that the pressure gradient is normalised by the reference density ρ_0.

17.1.2 Surface forcing data for the 1-D mode

The data for the water column forcing are defined in the routine usrdef_1dsur_data. The routine is called if iopt_sur_1D=1 and modfiles(io_1uvsur,2,1)%status='N'. The routine is declared as follows:

```
SUBROUTINE usrdef_1dsur_data(ciodatetime,data1d,novars)
CHARACTER (LEN=lentime), INTENT(INOUT) :: ciodatetime
INTEGER, INTENT(IN) :: novars
REAL, INTENT(INOUT), DIMENSION(novars) :: data1d
```

where

novars the number of data variables depending on the value of isur1dtype

1: three data values (X- and Y-component of the pressure gradient and elevation)

2: one data value (elevation)

3: two data values (X- and Y-component of the pressure gradient)

The following information has to be obtained:

ciodatetime date/time in string format[1]

data1d forcing data. The array elements are:

1: X-component of the pressure gradient if isur1dtype=1 or 3, surface elevation if isur1dtype=2

2: Y-component of the pressure gradient if isur1dtype=1 or 3

3: surface elevation if isur1dtype=1

17.2 2-D surface forcing

17.2.1 Surface grid in absolute coordinates

The subroutine usrdef_surface_absgrd defines a surface data grid in "absolute" (geographical) coordinates. The X- and Y-coordinates are Cartesian or spherical depending on the value of the switch iopt_grid_sph. The routine is called when

- the grid is rectangular and non-uniform: surfacegrids(idgrd,ifil)%nhtype=2 where idgrd is the grid key id (see below)

- modfiles(iddesc,ifil,1)%status='N' where iddesc is the file descriptor and ifil the file index (see below)

[1]If the parameter time_zone is defined with a non-zero value, the time of the input data must be given in local time.

- the switch iopt_meteo is set to 1 in the case of a meteorological grid
- the switch iopt_temp_sbc equals 2 or 3 in the case of a SST (sea surface temperature) grid

```
SUBROUTINE usrdef_surface_absgrd(iddesc,ifil,n1dat,n2dat,xcoord,ycoord)
INTEGER, INTENT(IN) :: iddesc, ifil, n1dat, n2dat
REAL, INTENT(INOUT), DIMENSION(n1dat,n2dat) :: xcoord, ycoord
```

The INTENT(IN) arguments have the following meaning:

iddesc The file descriptor of the corresponding data file. The key id in parentheses below is the associated grid key id (idgrd).

io_metgrd surface meteo grid (igrd_meteo)

io_sstgrd surface grid for sea surface temperature (SST, igrd_sst)

io_wavgrd grid for surface waves (igrd_waves)

ifil file index. In the current version its value is 1.

n1dat X-dimension of the data grid equal to surfacegrids(idgrd,ifil)%n1dat

n2dat Y-dimension of the data grid equal to surfacegrids(idgrd,ifil)%n2dat

The following arrays need to be given here:

xcoord(n1dat,n2dat) X-coordinates [m or degrees longitude]

ycoord(n1dat,n2dat) Y-coordinates [m or degrees latitude]

17.2.2 Surface grid in relative coordinates

The subroutine usrdef_surface_relgrd defines a surface data grid in "relative" coordinates. Currently, the routine is only used to obtain the coordinates of the model C-grid points relative to a data grid.

The routine is called when

- The grid is non-rectangular: surfacegrids(idgrd,ifil)%nhtype=3 where idgrd is the grid key id (see below)
- modfiles(iddesc,ifil,1)%status='N' where iddesc is the file descriptor and ifil the file index (see below)
- The switch iopt_meteo is set to 1 in the case of a meteorological grid
- The switch iopt_temp_sbc equals 2 or 3 in the case of a SST grid

The relative coordinates are stored in a DERIVED TYPE array of type HRelativeCoords:

```
TYPE :: HRelativeCoords
   INTEGER :: icoord, jcoord
   REAL :: xcoord, ycoord
END TYPE HRelativeCoords
```

where icoord, jcoord are the indices in the X- and Y-direction of the grid cell containing the data point and xcoord, ycoord the normalised Cartesian coordinates (between 0 and 1) with respect to axes along the cell faces and origin at the lower left corner.

The routine is declared as

```
SUBROUTINE usrdef_surface_relgrd(iddesc,ifil,surfgridglb,nx,ny,&
                               & nonodes)
INTEGER, INTENT(IN) :: iddesc, ifil, nonodes, nx, ny
TYPE (HRelativeCoords), INTENT(INOUT), DIMENSION(nx,ny,nonodes)&
                               & :: surfgridglb
```

where

iddesc The file descriptor of the corresponding data file. The key id in parentheses below is the associated grid key id (idgrd).

- io_metgrd surface meteo grid (igrd_meteo)
- io_sstgrd surface grid for SST (igrd_sst)
- io_wavgrd grid for surface waves (igrd_waves)

ifil file index. In the current version its value is 1.

nx Currenly equal to the global X-dimension nc of the model grid.

ny Currenly equal to the global Y-dimension nr of the model grid.

nonodes Number of nodes. In the current implementation its value equals 1.

The relative coordinates, defined in this routine, must be stored in the array:

surfgridglb relative coordinates of the model C-node grid with respect to a data grid represented by iddesc

17.2.3 Surface forcing data

The surface forcing data are defined in usrdef_surface_data. The routine is called if

- modfiles(iddesc,ifil,1)%status='N' where iddesc is the file descriptor and ifil the file index (see below).
- The switch iopt_meteo is set to 1 in the case of a meteorological grid.
- The switch iopt_temp_sbc equals 2 or 3 in the case of a SST (sea surface temperature) grid.
- The switch iopt_waves is set to 1 in the case of a surface wave grid.

The routine is declared as

```
SUBROUTINE usrdef_surface_data(iddesc,ifil,ciodatetime,&
                             & surdata,n1dat,n2dat,novars)
CHARACTER (LEN=lentime), INTENT(INOUT) :: ciodatetime
INTEGER, INTENT(IN) :: iddesc, ifil, novars, n1dat, n2dat
REAL, INTENT(INOUT), DIMENSION(n1dat,n2dat,novars) :: surdata
```

where

iddesc The file descriptor of the corresponding data file:

io_metsur	surface meteo data
io_sstsur	surface SST data
io_wavsur	surface wave data

ifil The file index. In the current version its value is 1.

n1dat X-dimension of the data grid equal to surfacegrids(idgrd,ifil)%n1dat

n2dat Y-dimension of the data grid equal to surfacegrids(idgrd,ifil)%n2dat

novars Number of data variables. In case of meteorological data its value ranges from 2 to 7, in case of SST data its value is 1.

The number and type of meteorological data depends on the values of the switches iopt_meteo_stres, iopt_meteo_heat and iopt_meteo_salflx (see Section 14.4.16).

The data, to be defined, are:

ciodatetime date/time in string format[1]

surdata surface forcing data as defined on the surface data grid

Table 17.1: Input data required for surface forcing and criterion for input

name	description	unit	criterion
Meteorological data			
uwindatc	X-component of surface wind	m/s	iopt_meteo_stres= 1
vwindatc	Y-component of surface wind	m/s	iopt_meteo_stres= 1
ustresatc	X-component of surface stress	m^2/s^2	iopt_meteo_stres= 2
vstresatc	Y-component of surface stress	m^2/s^2	iopt_meteo_stres= 2
atmpres	atmospheric pressure	N/m^2	iopt_meteo_stres>0 nhtype>1
airtemp	air temperature	^0C	iopt_meteo_heat = 1
relhum	relative humidity	between 0 and 1	iopt_meteo_heat = 1
qnonsol	non-solar (upward) surface heat flux	W/m^2	iopt_meteo_heat = 2,3
cloud_cover	cloud cover	between 0 and 1	iopt_meteo_heat = 1,4
qrad	surface solar (downward) radiation	W/m^2	iopt_meteo_heat = 3,5
precipitation	precipitation rate	$kg/m^2/s$	iopt_meteo_salflx = 2
evapminprec	evaporation minus precipitation rate	$kg/m^2/s$	iopt_meteo_salflx = 1
Surface temperature data			
sst	sea surface temperature	^0C	iopt_temp_sbc = 2,3
Surface wave data			
waveheight	significant wave height	m	iopt_waves > 0
waveperiod	peak wave period	s	iopt_waves > 0
wavedir	wave direction	radians	iopt_waves > 0
wavevel	near bed wave velocity	m/s	iopt_waves = 2
waveexcurs	near bed wave excursion amplitude	m	iopt_waves = 2

A list of input forcing data is given in Table 17.1. In case of meteo input the following remaks apply:

- Meteo is only enabled when iopt_meteo = 1.
- The number and order of variables is determined by the switches iopt_meteo_stres, iopt_meteo_heat, iopt_meteo_salflx and the surface grid parameter nhtype.
- Wind velocities, atmospheric pressure, air temperature and relative humidity are assumed to be taken at a reference hight of 10 m.

17.3 Nesting

The objective of nesting is to interpolate values of specific model arrays to external data points, which constitute the open boundaries of one or more sub-grids nested within the model grid. The interpolated data are written to output files in standard COHERENS format. A different file is used for each type of variable and each subgrid. Writing of a file is enabled by setting modfiles(iddesc,iset,2)%status='W' in usrdef_mod_params where

iddesc the file descriptor which can take the following values:

- io_2uvnst 2-D mode variables (transport and surface elevation)
- io_3uvnst 3-D baroclinic current
- io_salnst salinity
- io_tmpnst temperature
- io_sednst sediment fractions

iset number of the nested sub-grid (between 1 and nonestsets)

It is assumed that the variables on the subdomain are defined on a C-grid like the main grid. The following five types of horizontal interpolation (depending on the type of variable) need to be considered:

- from model C-nodes to sub-grid C-nodes: 3-D scalars
- from model C-nodes to sub-grid U- or V-nodes: elevations
- from model U-nodes to sub-grid U-nodes: X-components of transports and currents
- from model V-nodes to sub-grid V-nodes: Y-components of transports and currents

To summarise, the nesting procedure is defined as follows:

1. Set iopt_nests to 1.
2. Define nonestsets as the number of nested sub-grids.
3. Define the number of data points in usrdef_nstgrd_spec.
4. Define the positions of the data points in usrdef_nstgrd_abs and/or usrdef_nstgrd_rel.
5. Enable the interpolation and writing of specific model variables by setting modfiles(iddesc,iset,2)%status='W'.

17.3.1 Sub-grid specifications

This section describes routine usrdef_nstgrd_spec where general parameters such as the number of data points and the type of coordinates are defined for each sub-grid. The routine is called if:

- the switch iopt_nests is set to 1.
- modfiles(io_nstspc,1,1)%status='N'

```
SUBROUTINE usrdef_nstgrd_spec
INTEGER, DIMENSION(nonestsets) :: nestcoords, nohnstglbc,  &
                    & nohnstglbu, nohnstglbv, novnst, inst2dtype
```

The argument arrays need to be defined for each sub-grid separately:

nestcoords Type of coordinates used to define the positions of the sub-grid open boundary points

1: absolute coordinates

2: relative coordinates

nohnstglbc number of C-node sub-grid points in the horizontal

nohnstglbu number of U-node sub-grid points in the horizontal

nohnstglbv number of V-node sub-grid points in the horizontal

novnst vertical dimension of the sub-grid

inst2dtype selects the type of data for 2-D nesting

1: transports and elevations

2: elevations

3: transports

In case of multi-variable scalar variables (such as concentrations of sediment fractions), additional specifier arrays need to be specified. For details, see Section 19.4.

17.3.2 Sub-grid locations in absolute coordinates

The positions of the data points for a sub-grid with index iset in absolute coordinates are obtained in usrdef_nstgrd_abs. The routine is called when

- iopt_nests is set to 1.

- modfiles(io_nstgrd,iset,1)%status='N'
- nestcoords(iset)=1
- nhdat>0 (see below)

The routine has the following arguments:

```
SUBROUTINE usrdef_nstgrd_abs(iset,nhdat,nzdat,xcoord,ycoord,zcoord,cnode)
CHARACTER (LEN=1), INTENT(IN) :: cnode
INTEGER, INTENT(IN) :: iset, nhdat, nzdat
REAL, INTENT(OUT), DIMENSION(nhdat) :: xcoord, ycoord
REAL, INTENT(OUT), DIMENSION(nhdat,nzdat) :: zcoord
```

where

iset the index number of the sub-grid

cnode grid node of the sub-grid data points which may take the values 'C', 'U', 'V'

nhdat number of sub-grid points in the horizontal, equal to the value of either nohnstglbc(iset), nohnstglbu(iset) or nohnstglbv(iset) depending on the value of cnode

nzdat number of data points in the vertical equal to novnst(iset)

Provided that the conditions above are fullfilled, the routine is called for each sub-grid at most 3 times with respectively cnode='C','U','V'.

The following coordinate arrays are to be defined:

xcoord X-coordinates of the sub-grid data points [meters or longitude]

ycoord Y-coordinates of the sub-grid data points [meters or latitude]

zcoord Z-coordinates of the sub-grid locations, defined as the negative distance to the mean surface level (only when nzdat>0) [m]

17.3.3 Sub-grid locations in relative coordinates

The positions of the data points for sub-grid with index iset in relative coordinates are obtained in usrdef_nstgrd_rel. The routine is called when

- iopt_nests is set to 1.
- modfiles(io_nstgrd,iset,1)%status='N'
- nestcoords(iset)=2

- nhdat>0 (see below)

As for the case of a surface data grid, the horizontal relative coordinates are stored in a DERIVED TYPE array of type HRelativeCoords:

```
TYPE :: HRelativeCoords
   INTEGER :: icoord, jcoord
   REAL :: xcoord, ycoord
END TYPE HRelativeCoords
```

where icoord, jcoord are the indices in the X- and Y-direction of the grid cell containing the data point and xcoord, ycoord the normalised Cartesian coordinates (between 0 and 1) with respect to axes along the cell faces and origin at the lower left corner.

The routine is declared as follows:

```
SUBROUTINE usrdef_nstgrd_rel(iset,nhdat,nzdat,nonodes,hnests,&
                           & zcoord,cnode)
CHARACTER (LEN=1), INTENT(IN) :: cnode
INTEGER, INTENT(IN) :: iset, nhdat, nonodes, nzdat
REAL, INTENT(OUT), DIMENSION(nhdat,nzdat) :: zcoord
TYPE (HRelativeCoords), INTENT(OUT), DIMENSION(nhdat,nonodes) :: hnests
```

The INTENT(IN) arguments have the following meaning

iset — the index number of the sub-grid

cnode — grid node of the sub-grid data points which may take the values 'C', 'U', 'V'

nhdat — number of sub-grid points in the horizontal, equal to the value of either nohnstglbc(iset), nohnstglbu(iset) or nohnstglbv(iset) depending on the value of cnode

nzdat — number of data points in the vertical equal to novnst(iset)

nonodes — the number of nodes for which relative coordinates need to be provided. Its value equals 1 or 2 depending on the value of cnode. This is further discussed below.

The positions are stored into the arrays

hnests — relative coordinates of the sub-grid ('C', 'U', 'V') locations with respect to either the C-, U- or V-node model grid

zcoord — Z-coordinates of the sub-grid locations, taken as the negative distance to the mean surface level (only when nzdat>0) [m].

Provided that nhdat> 0 and the other conditions above apply, the routine is called successively with cnode='C','U','V'. The value of nonodes and the meaning of hnests depends on cnode:

'C': One series of coordinates needs to be given (nonodes=1), representing interpolation from the C-node model grid to the C-node sub-grid locations.

'U': Two series of coordinates need to be given (nonodes=2)

1: interpolation from the U-node model grid to the U-node sub-grid locations

2: interpolation from the C-node model grid to the U-node sub-grid locations

'V': Two series of coordinates need to be given (nonodes=2)

1: interpolation from the V-node model grid to the V-node sub-grid locations

2: interpolation from the C-node model grid to the V-node sub-grid locations