Chapter 20

User output

There exist 4 types of user output:

1. The source file *Usrdef_Time_Series.f90* defines setup parameters and data for time series output. Time series output is enabled when iopt_out_tsers=1 (default value).

2. The source file *Usrdef_Time_Averages.f90* defines setup parameters and data for time averaged output. Time averaged output is enabled when iopt_out_avrgd=1.

3. The source file *Usrdef_Harmonic_Analysis.f90* defines setup parameters and data for harmonic analysis and output. Harmonic analysis and output is enabled when iopt_out_anal=1.

4. User formatted output can be defined in *Usrdef_Output.f90*.

The usrdef_routines for each type are discussed in Sections 20.1-20.4. The output files not only contain the model data and metadata but also the coordinates of the output grid. In this way each output file can be considered as self-consistent and used by independent postprocessing programs for display and analysis. Coordinates of the output data grid are the subject of Section 20.5.

20.1 Time series output

The file *Usrdef_Time_Series.f90* contains the routines:

- usrdef_tsr_params: defines the output variables, file attributes, the space-time resolution of the output grid and other metadata

- usrdef_tsr0d_vals: values of the 0-D output variables
- usrdef_tsr2d_vals: values of the 2-D output variables
- usrdef_tsr3d_vals: values of the 3-D output variables

20.1.1 Specifiers for time series output

The following integer and derived type arrays can or must be defined here:

```
TYPE (VariableAtts), DIMENSION(novarstsr) :: tsrvars
INTEGER, DIMENSION(nosetstsr,novarstsr) :: ivarstsr
TYPE (FileParams), DIMENSION(nosetstsr) :: tsr0d, tsr2d, &
                                        & tsr3d, tsrgrd
TYPE (OutGridParams), DIMENSION(nosetstsr) :: tsrgpars
INTEGER, DIMENSION(nosetstsr,nostatstsr) :: lstatstsr
TYPE (StationLocs), DIMENSION(nostatstsr) :: tsrstatlocs
```

The array dimensions are previously defined in usrdef_mod_params and have the following meaning

nosetstsr	number of output "sets" as defined in usrdef_mod_params
novarstsr	total number of output variables as defined in usrdef_mod_params
nostatstsr	total number of stations as defined in usrdef_mod_params

The general meaning of the setup arrays is

tsrvars	attributes of the output variables
ivarstsr	variable indices
tsr0d	attributes of the 0-D output files
tsr2d	attributes of the 2-D output files
tsr3d	attributes of the 3-D output files
tsrgrd	attributes of the grid file (if needed)
tsrgpars	attributes of the output data grid
lstatstsr	stations labels
tsrstatlocs	station attributes

A more detailed discussion is given below.

20.1.1.1 variable attributes

Attributes of the output variables are stored in a DERIVED TYPE array of type VariableAtts[1]

```
TYPE :: VariableAtts
   CHARACTER (LEN=lenname) :: f90_name
   CHARACTER (LEN=lendesc) :: long_name, vector_name
   CHARACTER (LEN=lenunit) :: units
   CHARACTER (LEN=lennode) :: node
   INTEGER :: ivarid, klev, nrank, numvar, oopt
   REAL :: dep
END TYPE VariableAtts
```

Output variables and their attributes are then selected with the following arrays:

```
TYPE (VariableAtts), DIMENSION(novarstsr) :: tsrvars
INTEGER, DIMENSION(nosetstsr,novarstsr) :: ivarstsr
```

ivarid — Variable key id of the output variable (as defined in *modids.f90*). If zero, the variable is considered as user-defined. Otherwise, the key id should be supplied using its FORTRAN name (standard variable). The syntax is iarr_* where * is the variables's FORTRAN name (e.g. iarr_temp). A list of available key ids is given in Appendix E. Variables need to be defined in the following order: 0-D (if any), 2-D (if any), 3-D (if any).

f90_name — FORTRAN name of the variable (e.g. 'temp') (user-defined variables only)

long_name — long description of the variable, e.g. 'temperature' (user-defined variables only)

vector_name — If the variable is the component of a vector, the name of the vector, e.g. 'current' (user-defined variables only)

units — unit of the variable, e.g. 'm/s' (user-defined variables only)

nrank — variable rank (0, 2 or 3). Its value must be non-decreasing with the array index. The variables are ordened in the same sequence as ivarid. This means that the variables in the array tsrvars must be defined in the following order: first 0-D (if any), next 2-D (if any) and finally the 3-D variables (if any).

[1]The string lengths lenname, lendesc, lenunit, lennode are defined in *syspars.f90*.

numvar Variable number in case of multi-variable arrays, such as sediment fractions. The number then represents the last index of the data variable (e.g. fraction number).

ivarstsr Each file set contains a sub-set of the variables defined in tsrvars. The element ivarstsr(iset,ivar) maps, for set iset, the local variable index ivar into the corresponding array index in tsrvars.

In case of a standard variable (non-zero ivarid attribute), the following attributes may optionally be defined:

oopt If the rank of the result is different from the one implemented by the variable's rank, the rank attribute must be set to the rank of the result. For example, the domain average of a 3-D variable has a rank of 0. The attribute has one of the following values

- oopt_null No operator is applied (default).
- oopt_mean Result depends on the rank of the model variable and the rank of the output data.
  - If the rank of the result is 0, the output value is the domain average in case of a 3-D or the surface average in case of a 2-D variable. Land areas are excluded in the averaging.
  - If the rank of the output value is 2, the result is the depth averaged value.
- oopt_int Result depends on the rank of the model variable and the rank of the output data.
  - If the rank of the result is 0, the output value is the domain integrated value in case of a 3-D or the surface integrated value in case of a 2-D variable. Land areas are excluded in the integration.
  - If the rank of the output value is 2, the result is the depth integrated value.
- oopt_max Result depends on the rank of the model variable and the rank of the output data.
  - If the rank of the result is 0, the output value is the domain maximum in case of a 3-D or the surface maximum in case of a 2-D variable. Land areas are excluded.

- If the rank of the output value is 2, the result is the maximum over the water depth.

oopt_min Result depends on the rank of the model variable and the rank of the output data.

- If the rank of the result is 0, the output value is the domain minimum in case of a 3-D or the surface minimum in case of a 2-D variable. Land areas are excluded.
- If the rank of the output value is 2, the result is the minimum over the water depth.

klev Produces the value of a 3-D variable at the vertical level given by the attribute klev. Rank of the result is 2.

oopt_dep Produces the value of a 3-D variable using vertical interpolation at a specified depth given by the attribute dep. Rank of the result is 2.

oopt_klev Defines the output vertical level in case oopt equals oopt_klev.

dep Defines the output water depth (measured positively from the surface) in case oopt equals oopt_dep. Result is 0, if dep is larger than the total water depth at the output location.

node Used for 3-D variables defined at W-nodes on the model grid. If node is set to 'C' (default), the vertical profile of the variable is first interpolated at the C-node before the operator is applied. If set to 'W', the output variable must be defined at the W-node and no interpolation is performed. It is remarked that quantities defined at U- or V-nodes are always interpolated at the C-nodes before the operator is applied.

20.1.1.2 file attributes

File attributes are stored in a DERIVED TYPE array of type FileParams, defined in Section 14.7:

```
TYPE (FileParams), DIMENSION(nosetstsr) :: tsr0d, tsr2d, tsr3d, tsrgrd
```

The following file attributes can be defined:

defined The output file will be created only if this parameter is set to .TRUE.. Default is .FALSE..

form Format of the data file.

'A': ASCII

'U': unformatted binary

'N': netCDF

filename File name. If not defined, a default will be constructed by the program.

info An "info" file (ending with 'I') will be created if .TRUE. (default value).

header_type No heading information will be written if set to 0. Default is 2. This option is not available for netCDF ('N') files.

The arrays tsr0d, tsr2d, tsr3d are used for output of respectively 0-D, 2-D, 3-D output. They will contain metadata output if header_type is set to its default value or in case of a netCDF file. By default, the data file will contain the coordinates of the output grid and the times of output. The coordinate and time data can also be written to a separate "grid" file whose attributes are stored in tsrgrd. A selection can be made with the grid_file attribute described below.

20.1.1.3 output data grid

The output grid (in space and time) and its attributes are defined using the following DERIVED TYPE array.

```
TYPE :: OutGridParams
   LOGICAL :: gridded, grid_file, land_mask, time_grid
   CHARACTER (LEN=lentime) :: refdate
   INTEGER :: nodim, nostats, time_format
   INTEGER, DIMENSION(3) :: tlims, xlims, ylims, zlims
END TYPE OutGridParams
TYPE (OutGridParams), DIMENSION(nosetstsr) :: tsrgpars
```

gridded If .TRUE. (default), output data are defined on a sub-grid of the model grid (or the whole model grid). If .FALSE., the data are taken at a number of irregularly spaced locations ("station" data) defined below.

grid_file If .TRUE., the coordinates of the output grid will be written on a separate output file (defined by tsrgrd). Otherwise, they are written within the data file itself. Default is .FALSE.

land_mask A land mask will be applied if .TRUE. and gridded=.TRUE.. This means that the gridded data will be stored as a vector excluding land points. Advantage may be a significant reduction in disk space. Default is .FALSE..

time_grid If .TRUE., the data grid is taken as time-dependent (since the vertical positions in a σ-grid depend on time). Surface elevations will be written as an additional coordinate variable at each time step. Default if .FALSE., in which case the vertical positions are referred to the mean water level.

time_format Format of the time coordinate (0).

0: date/time in string fromat

1: seconds

2: minutes

3: hours

4: days

5: months

6: years

7: date in years

Cases 1–6 are numerical formats. Cases 0 and 7 are absolute times, the other ones relative to the reference date/time.

refdate Reference date/time for calculating relative times. If not given, refdate equals the first output date/time, rounded to the nearest minute, hour, ... depending on the value of time_format.

tlims Start/end/step time indices for data output. This means that output will be written at intervals of delt2d*tlims(3) seconds.

nodim Dimension of the output grid (0/2/3). For example, the dimension must be set to 3 to enable 3-D output.

nostats Number of data stations in case of non-gridded (station) data.

xlims Start/end/step X-index in case of gridded data. This defines the output sub-grid in the X-direction. (Option not available for 0-D or station output). Default values are (1,nc-1,1).

ylims Start/end/step Y-index in case of gridded data. This defines the output sub-grid in the Y-direction. (Option not available for station or 0-D output). Default values are (1,nr-1,1).

zlims Start/end/step Z-index in case of gridded data. This defines the output sub-grid in the Z-direction and applies for gridded and

non-gridded output. (Option only available for 3-D output). Default values are (1,nz,1).

20.1.1.4 station attributes

Station atrributes are defined by

```
TYPE :: StationLocs
   INTEGER :: ipos, jpos
   CHARACTER (LEN=lenname) :: name
END TYPE StationLocs
TYPE (StationLocs), DIMENSION(nostatstsr) :: tsrstatlocs
INTEGER, DIMENSION(nosetstsr,nostatstsr) :: lstatstsr
```

ipos (global) X-index of the output stations

jpos (global) Y-index of the output stations

name descriptive names of the stations. Default is 'Station X' where 'X' is the station's array index.

lstatstsr Station labels. Each file set may contain a sub-set of stations. The element lstatstsr(iset,istat) maps, for set iset, the local station index istat into the corresponding global array index in tsrstatlocs.

20.1.2 Time series output data

The usrdef_ routines below are only called to define the values of user-defined variables. Output data for standard variables are automatically generated by the program and evaluated at the C-nodes (except when the node attribute is set to 'W'). Interpolation is performed when necessary.

Important to note is that the program assumes that all (including user-defined) 2-D and 3-D output data are given at the C-nodes (unless the node attribute is set to 'W'). This means that variables, defined at other nodes, such as velocity components, need to be interpolated first at the C-nodes. Interpolation routines are provided by the program (see Section 31.1).

20.1.2.1 values of 0-D time series data

The subroutine usrdef_tsr0d_vals defines the 0-D (space-independent) data for time series output.

```
SUBROUTINE usrdef_tsr0d_vals(out0ddat,n0vars)
INTEGER, INTENT(IN) :: n0vars
REAL, INTENT(OUT), DIMENSION(n0vars) :: out0ddat
```

where

n0vars number of 0-D output variables

out0ddat 0-D output data. The variables are given in the same order as they are defined in the array tsrvars with the index 1 referring to the first 0-D variable and the index n0vars to the last variable in tsrvars.

20.1.2.2 values of 2-D time series data

The subroutine usrdef_tsr2d_vals defines the 2-D (horizontally dependent) data for time series output.

```
SUBROUTINE usrdef_tsr2d_vals(out2ddat,i,j,n2vars)
INTEGER, INTENT(IN) :: i, j, n2vars
REAL, INTENT(OUT), DIMENSION(n2vars) :: out2ddat
```

where

n2vars number of 2-D output variables

i X-index of the data location on the *local* model grid

j Y-index of the data location on the *local* model grid

out2ddat 2-D output data. The variables are given in the same order as they are defined in the array tsrvars with the index 1 referring to the first 2-D variable and the index n2vars to the last variable in tsrvars.

20.1.2.3 values of 3-D time series data

The subroutine usrdef_tsr3d_vals defines the 3-D (spatially dependent) data for time series output.

```
SUBROUTINE usrdef_tsr3d_vals(out3ddat,i,j,k,n3vars)
INTEGER, INTENT(IN) :: i, j, k, n3vars
REAL, INTENT(OUT), DIMENSION(n3vars) :: out3ddat
```

where

n3vars number of 3-D output variables

i X-index of the data location on the *local* model grid

j Y-index of the data location on the *local* model grid

k vertical index of the data location on the model grid

out3ddat 3-D output data. The variables are given in the same order as they are defined in the array tsrvars with the index 1 referring to the first 3-D variable and the index n3vars to the last variable in tsrvars.

20.2 Time averaged output

The file *Usrdef_Time_Averages.f90* contains the routines:

- usrdef_avr_params: defines the output variables, file attributes, the space-time resolution of the output grid and other metadata
- usrdef_avr0d_vals: values of the 0-D output variables
- usrdef_avr2d_vals: values of the 2-D output variables
- usrdef_avr3d_vals: values of the 3-D output variables

Time averaged output is defined in almost the exact way as time series output. Only differences are

- In all variable and routine names 'tsr' is replaced by 'avr'.
- The third element of the vector attribute tlims(3) is now used also to determine the averaging period.
- The output grid is always time independent.

For completeness all definitions are fully discussed below.

20.2.1 Specifiers for time averaged output

The following integer and derived type arrays can or must be defined here:

```
TYPE (VariableAtts), DIMENSION(novarsavr) :: avrvars
INTEGER, DIMENSION(nosetsavr,novarsavr) :: ivarsavr
TYPE (FileParams), DIMENSION(nosetsavr) :: avr0d, avr2d, &
                                        & avr3d, avrgrd
TYPE (OutGridParams), DIMENSION(nosetsavr) :: avrgpars
INTEGER, DIMENSION(nosetsavr,nostatsavr) :: lstatsavr
TYPE (StationLocs), DIMENSION(nostatsavr) :: avrstatlocs
```

The array dimensions are previously defined in usrdef_mod_params and have the following meaning

nosetsavr	number of output "sets" as defined in usrdef_mod_params
novarsavr	total number of output variables as defined in usrdef_mod_params
nostatsavr	total number of stations as defined in usrdef_mod_params

The general meaning of the setup arrays is

avrvars	attributes of the output variables
ivarsavr	variable indices
avr0d	attributes of the 0-D output files
avr2d	attributes of the 2-D output files
avr3d	attributes of the 3-D output files
avrgrd	attributes of the grid file (if needed)
avrgpars	attributes of the output data grid
lstatsavr	stations labels
avrstatlocs	station attributes

A more detailed discussion is given below.

20.2.1.1 variable attributes

Output variables and their attributes are selected with the following arrays:

```
TYPE (VariableAtts), DIMENSION(novarsavr) :: avrvars
INTEGER, DIMENSION(nosetsavr,novarsavr) :: ivarsavr
```

The array components and elements provide the following information

ivarid	Variable key id of the output variable (as defined in *modids.f90*). If zero, the variable is considered as user-defined. Otherwise, the key id should be supplied using its FORTRAN name (standard variable). The syntax is iarr_* where * is the variables's FORTRAN name (e.g. iarr_temp). A list of available key ids is given in Appendix E. Variables need to be defined in the following order: 0-D (if any), 2-D (if any), 3-D (if any).
f90_name	FORTRAN name of the variable, e.g. 'temp') (user-defined variables only)
long_name	long description of the variable (e.g. 'temperature'), user-defined variables only
vector_name	If the variable is the component of a vector, the name of the vector, e.g. 'current' (user-defined variables only). Otherwise, it is left undefined.
units	unit of the variable, e.g. 'm/s' (user-defined variables only)
nrank	variable rank (0, 2 or 3). Its value must be non-decreasing with the array index. The variables are ordened in the same sequence

as ivarid. This means that the variables in the array avrvars must be defined in the following order: first 0-D (if any), next 2-D (if any) and finally the 3-D variables (if any).

numvar Variable number in case of multi-variable arrays, such as sediment fractions. The number then represents the last index of the data variable (e.g. fraction number).

ivarsavr Each file set contains a sub-set of the variables defined in avrvars. The element ivarsavr(iset,ivar) maps, for set iset, the local variable index ivar into the corresponding array index in avrvars.

In case of a standard variable (non-zero ivarid attribute), the following attributes may optionally be defined:

oopt If the rank of the result is different from the one implemented by the variable's rank, the rank attribute must be set to the rank of the result. For example, the domain average of a 3-D variable has a rank of 0. The attribute has one of the following values

- oopt_null No operator is applied (default).
- oopt_mean Result depends on the rank of the model variable and the rank of the output data.
  - If the rank of the result is 0, the output value is the domain average in case of a 3-D or the surface average in case of a 2-D variable. Land areas are excluded in the averaging.
  - If the rank of the output value is 2, the result is the depth averaged value.
- oopt_int Result depends on the rank of the model variable and the rank of the output data.
  - If the rank of the result is 0, the output value is the domain integrated value in case of a 3-D or the surface integrated value in case of a 2-D variable. Land areas are excluded in the integration.
  - If the rank of the output value is 2, the result is the depth integrated value.
- oopt_max Result depends on the rank of the model variable and the rank of the output data.

- If the rank of the result is 0, the output value is the domain maximum in case of a 3-D or the surface maximum in case of a 2-D variable. Land areas are excluded.
- If the rank of the output value is 2, the result is the maximum over the water depth.

oopt_min Result depends on the rank of the model variable and the rank of the output data.

- If the rank of the result is 0, the output value is the domain minimum in case of a 3-D or the surface minimum in case of a 2-D variable. Land areas are excluded.
- If the rank of the output value is 2, the result is the minimum over the water depth.

oopt_klev Produces the value of a 3-D variable at the vertical level given by the attribute klev. Rank of the result is 2.

oopt_dep Produces the value of a 3-D variable using vertical interpolation at a specified depth given by the attribute dep. Rank of the result is 2.

klev Defines the output vertical level in case oopt equals oopt_klev.

dep Defines the output water depth (measured positively from the surface) in case oopt equals oopt_dep. Result is 0, if dep is larger than the total water depth at the output location.

node Used for 3-D variables defined at W-nodes on the model grid. If node is set to 'C' (default), the vertical profile of the variable is first interpolated at the C-node before the operator is applied. If set to 'W', the output variable must be defined at the W-node and no interpolation is performed. It is remarked that quantities defined at U- or V-nodes are always interpolated at the C-nodes before the operator is applied..

20.2.1.2 file attributes

File attributes are stored in a DERIVED TYPE array of type FileParams, defined in Section 14.7:

```
TYPE (FileParams), DIMENSION(nosetsavr) :: avr0d, avr2d, avr3d, avrgrd
```

The following file attributes can be defined:

defined — The output file will be written only if this parameter is set to .TRUE.. Default is .FALSE..

form — Format of the data file.

'A': ASCII

'U': unformatted binary

'N': netCDF

filename — File name. If not defined, a default will be constructed by the program.

info — An "info" file (ending with 'I') will be created if .TRUE. (default value).

header_type — No heading information will be written if set to 0. Default is 2. This option is not available for netCDF ('N') files.

The arrays avr0d, avr2d, avr3d are used for output of respectively 0-D, 2-D, 3-D output. They will contain metadata output if header_type is set to its default value or in case of a netCDF file. By default, the data file will contain the coordinates of the output grid and the times of output. The coordinate and time data can also be written to a separate "grid" file whose attributes are stored in avrgrd. A selection can be made with the grid_file attribute described below.

20.2.1.3 output data grid

The output grid (in space and time) and its attributes are defined using the following DERIVED TYPE array:

```
TYPE (OutGridParams), DIMENSION(nosetsavr) :: avrgpars
```

gridded — If .TRUE. (default), output data are defined on a sub-grid of the model grid (or the whole model grid). If .FALSE., the data are taken at a number of irregularly spaced locations ("station" data) defined below.

grid_file — If .TRUE., the coordinates of the output grid will be written on a separate output file (defined by avrgrd). Otherwise, they are written within the data file itself. Default is .FALSE.

land_mask — A land mask will be applied if .TRUE. and gridded=.TRUE.. This means that the gridded data will be stored as a vector excluding land points. Advantage may be a significant reduction in disk space. Default is .FALSE..

time_format Format of the time coordinate (0).

0: date/time in string fromat

1: seconds

2: minutes

3: hours

4: days

5: months

6: years

7: date in years

Cases 1–6 are numerical formats. Cases 0 and 7 are absolute times, the other ones relative to the reference date/time.

refdate Reference date/time for calculating relative times. If not given, refdate equals the first output date/time, rounded to the nearest minute, hour, ... depending on the value of time_format.

tlims The first two elements of this vector are the start and end time indices for data output and determine the start and end output date/times. The averaging period is defined by tlims(3). This means in practice that output will be written at regular intervals of delt2d*tlims(3) seconds which is the same as the period of averaging. Output times are defined in the middle of each averaging period.

nodim Dimension of the output grid (0/2/3). For example, the dimension must be set to 3 to enable 3-D output.

nostats Number of data stations in case of non-gridded (station) data.

xlims Start/end/step X-index in case of gridded data. This defines the output sub-grid in the X-direction. (Option not available for 0-D or station output). Default values are (1,nc-1,1).

ylims Start/end/step Y-index in case of gridded data. This defines the output sub-grid in the Y-direction. (Option not available for 0-D or station output). Default values are (1,nr-1,1).

zlims Start/end/step Z-index in case of gridded data. This defines the output sub-grid in the Z-direction and applies for gridded and non-gridded output. (Option only available for 3-D output). Default values are (1,nz,1).

20.2.1.4 station attributes

Station atrributes are defined by

```
TYPE (StationLocs), DIMENSION(nostatsavr) :: avrstatlocs
INTEGER, DIMENSION(nosetsavr,nostatsavr) :: lstatsavr
```

Arrays and array components represent the following

ipos (global) X-index of the output stations

jpos (global) Y-index of the output stations

name descriptive names of the stations. Default is 'Station X' where 'X' is the station's array index.

lstatsavr Station labels. Each file set may contain a sub-set of stations. The element lstatsavr(iset,istat) maps, for set iset, the local station index istat into the corresponding global array index in avrstatlocs.

20.2.2 Time averaged output data

The usrdef_ routines below are only called to define the values of user-defined variables. Output data for standard variables are automatically generated by the program and evaluated at the C-nodes (except when the node attribute is set to 'W'). Interpolation is performed when necessary.

Important to note is that the program assumes that all (including user-defined) 2-D and 3-D output data are given at the C-nodes (unless the node attribute is set to 'W'). This means that variables, defined at other nodes, such as velocity components, need to be interpolated first at the C-nodes. Interpolation routines are provided by the program (see Section 31.1).

20.2.2.1 values of 0-D time averaged data

The subroutine usrdef_avr0d_vals defines the 0-D (space-independent) data for time averaged output.

```
SUBROUTINE usrdef_avr0d_vals(out0ddat,n0vars)
INTEGER, INTENT(IN) :: n0vars
REAL, INTENT(OUT), DIMENSION(n0vars) :: out0ddat
```

where

n0vars number of 0-D output variables

out0ddat 0-D output data. The variables are given in the same order as they are defined in the array avrvars with the index 1 referring to the first 0-D variable and the index n0vars to the last variable in avrvars.

20.2.2.2 values of 2-D time averaged data

The subroutine usrdef_avr2d_vals defines the 2-D (horizontally dependent) data for time averaged output.

```
SUBROUTINE usrdef_avr2d_vals(out2ddat,i,j,n2vars)
INTEGER, INTENT(IN) :: i, j, n2vars
REAL, INTENT(OUT), DIMENSION(n2vars) :: out2ddat
```

where

n2vars	number of 2-D output variables
i	X-index of the data location on the *local* model grid
j	Y-index of the data location on the *local* model grid
out2ddat	2-D output data. The variables are given in the same order as they are defined in the array avrvars with the index 1 referring to the first 2-D variable and the index n2vars to the last variable in avrvars.

20.2.2.3 values of 3-D time averaged data

The subroutine usrdef_avr3d_vals defines the 3-D (spatially dependent) data for time averaged output.

```
SUBROUTINE usrdef_avr3d_vals(out3ddat,i,j,k,n3vars)
INTEGER, INTENT(IN) :: i, j, k, n3vars
REAL, INTENT(OUT), DIMENSION(n3vars) :: out3ddat
```

where

n3vars	number of 3-D output variables
i	X-index of the data location on the *local* model grid
j	Y-index of the data location on the *local* model grid
k	vertical index of the data location on the model grid
out3ddat	3-D output data. The variables are given in the same order as they are defined in the array avrvars with the index 1 referring to the first 3-D variable and the index n3vars to the last variable in avrvars.

20.3 Harmonic analysis

The file *Usrdef_Harmonic_Analysis.f90* contains the routines:

- usrdef_anal_freqs: defines the frequencies for harmonic analysis and their attributes
- usrdef_anal_params: defines the output variables, file attributes, the space-time resolution of the output grid and other metadata
- usrdef_anal0d_vals: values of the 0-D output variables
- usrdef_anal2d_vals: values of the 2-D output variables
- usrdef_anal3d_vals: values of the 3-D output variables

The definitions for harmonic output are similar to the ones for time series output. Main difference is that the output grid is time independent and additional information has to be given about the analysis itself and that the output files can be produced for residual data, amplitudes, phases and elliptic parameters.

20.3.1 Harmonic frequencies

The following arrays are defined here:

```
CHARACTER (LEN=7), DIMENSION(nofreqsanal) :: harm_freq_names
CHARACTER (LEN=lentime), DIMENSION(nosetsanal) :: cdate_time_ref
INTEGER, DIMENSION(nosetsanal) :: nofreqsharm
INTEGER, DIMENSION(nofreqsanal) :: index_anal
INTEGER, DIMENSION(nosetsanal,nofreqsanal) :: ifreqsharm
REAL, DIMENSION(nofreqsanal) :: harm_freq
```

The array sizes are defined in usrdef_mod_params and have the following meaning:

nosetsanal	number of output "sets" as defined in usrdef_mod_params
nofreqsanal	total number of frequencies as defined in usrdef_mod_params

The arrays provide the following information

index_anal	Key ids of the constituents used in harmonic analysis (as defined in *tide.f90*). If positive, harm_freq_names and harm_freq do not need to be defined. Default is 0.
nofreqsharm	number of frequencies per set
harm_freq	values of the frequencies [rad/s]
harm_freq_names	names of the frequencies names (e.g. 'M2')

ifreqsharm	Each set contains a sub-set of frequencies defined by index_anal. The element ifreqsharm(iset,ifreq) maps, for set iset, the local frequency ifreq into the corresponding "global" array index as defined in the global array index_anal.
cdate_time_ref	If set by the user, all harmonic phases are calculated with respect to this reference date. Otherwise, the reference date/time is taken with respect to the central time or to the astronomical phase at Greenwich depending on the value of iopt_astro_anal.

20.3.2 Specifiers for harmonic output

The following integer and derived type arrays can be defined here:

```
TYPE (VariableAtts), DIMENSION(novarsanal) :: analvars
INTEGER, DIMENSION(nosetsanal,novarsanal) :: ivarsanal
TYPE (FileParams), DIMENSION(nosetsanal) :: res0d, res2d, res3d, &
                                            & analgrd
TYPE (FileParams), DIMENSION(nosetsanal,nofreqsanal) :: &
          & amp0d, amp2d, amp3d, pha0d, pha2d, pha3d, ell2d, ell3d
TYPE (OutGridParams), DIMENSION(nosetsanal) :: analgpars
INTEGER, DIMENSION(nosetsanal,nostatsanal) :: lstatsanal
TYPE (StationLocs), DIMENSION(nostatsanal) :: analstatlocs
INTEGER, DIMENSION(nosetsanal,14) :: ivarsell
INTEGER, DIMENSION(nosetsanal,2) :: ivecell2d, ivecell3d
```

where the arrays sizes are defined in usrdef_mod_params:

nosetsanal	are the number of output "sets" as defined in usrdef_mod_params
novarsanal	are the total number of output variables as defined in usrdef_mod_params
nostatsanal	are the total number of stations as defined in usrdef_mod_params

Output parameters are defined by the following arrays

analvars	attributes of the output variables
ivarsanal	variable indices
res0d	attributes of the 0-D residual file(s)
amp0d	attributes of the 0-D amplitudes file(s)
pha0d	attributes of the 0-D phase file(s)
res2d	attributes of the 2-D residual file(s)

amp2d	attributes of the 2-D amplitudes file(s)
pha2d	attributes of the 2-D phase file(s)
ell2d	attributes of 2-D elliptical file(s)
res3d	attributes of the 3-D residual file(s)
amp3d	attributes of the 3-D amplitudes file(s)
pha3d	attributes of the 3-D phase file(s)
ell3d	attributes of 3-D elliptical file(s)
analgrd	attributes of the grid file (if needed)
analgpars	attributes of the output data grid
lstatsanal	stations labels
analstatlocs	station attributes
ivarsell	variable indices for elliptic parameters
ivecell2d	array indices (taken from analvars) for the X- and Y-component of 2-D elliptic vectors
ivecell3d	array indices (taken from analvars) for the X- and Y-component of 3-D elliptic vectors

A more detailed discussion is given below.

20.3.2.1 variable attributes

Output variables and their attributes are selected with the following arrays:

```
TYPE (VariableAtts), DIMENSION(novarsanal) :: analvars
INTEGER, DIMENSION(nosetsanal,novarsanal) :: ivarsanal
INTEGER, DIMENSION(nosetsanal,14) :: ivarsell
INTEGER, DIMENSION(nosetsanal,2) :: ivecell2d, ivecell3d
```

Arrays and array components have the following meaning

ivarid	Variable key id of the output variable (as defined in *modids.f90*). If zero, the variable is considered as user-defined. Otherwise, the key id should be supplied using its FORTRAN name (standard variable). The syntax is iarr_* where * is the variables's FORTRAN name (e.g. iarr_temp). A list of available key ids is given in Appendix E. Variables need to be defined in the following order: 0-D (if any), 2-D (if any), 3-D (if any).
f90_name	FORTRAN name of the variable, e.g. 'temp' (user-defined variables only)

long_name long description of the variable, e.g. 'temperature' (user-defined variables only)

vector_name If the variable is the component of a vector, the name of the vector, e.g. 'current' (user-defined variables only). Otherwise, it is left undefined.

units unit of the variable, e.g. 'm/s' (user-defined variables only)

nrank variable rank (0, 2 or 3). Its value must be non-decreasing with the array index. The variables are ordened in the same sequence as ivarid. This means that the variables in the array analvars must be defined in the following order: first 0-D (if any), next 2-D (if any) and finally the 3-D variables (if any).

numvar Variable number in case of multi-variable arrays, such as sediment fractions. The number then represents the last index of the data variable (e.g. fraction number).

ivarsanal Each file set contains a sub-set of the variables defined in analvars. The element ivarsanal(iset,ivar) maps, for set iset, the local variable index ivar into the corresponding array index in analvars.

ivarsell Has the same meaning as ivarsanal now for elliptic parameters (i.e. defines which elliptic parameters are written to output).

ivecell2d The array elements ivecell2d(iset,1) and ivecell2d(iset,2) are the array indices in analvars of the X- and Y-component of the vector used to describe the harmonic ellipses for 2-D output.

ivecell3d Array elements ivecell3d(iset,1) and ivecell3d(iset,2) are the array indices in analvars of the X- and Y-component of the vector used to describe the harmonic ellipses for 3-D output.

In case of a standard variable (non-zero ivarid attribute), the following attributes may optionally be defined:

oopt If the rank of the result is different from the one implemented by the variable's rank, the rank attribute must be set to the rank of the result. For example, the domain average of a 3-D variable has a rank of 0. The attribute has one of the following values

- oopt_null No operator is applied (default).
- oopt_mean Result depends on the rank of the model variable and the rank of the output data.
  - If the rank of the result is 0, the output value is the domain average in case of a 3-D or the

surface average in case of a 2-D variable. Land areas are excluded in the averaging.

- If the rank of the output value is 2, the result is the depth averaged value.

oopt_int Result depends on the rank of the model variable and the rank of the output data.

- If the rank of the result is 0, the output value is the domain integrated value in case of a 3-D or the surface integrated value in case of a 2-D variable. Land areas are excluded in the integration.
- If the rank of the output value is 2, the result is the depth integrated value.

oopt_max Result depends on the rank of the model variable and the rank of the output data.

- If the rank of the result is 0, the output value is the domain maximum in case of a 3-D or the surface maximum in case of a 2-D variable. Land areas are excluded.
- If the rank of the output value is 2, the result is the maximum over the water depth.

oopt_min Result depends on the rank of the model variable and the rank of the output data.

- If the rank of the result is 0, the output value is the domain minimum in case of a 3-D or the surface minimum in case of a 2-D variable. Land areas are excluded.
- If the rank of the output value is 2, the result is the minimum over the water depth.

oopt_klev Produces the value of a 3-D variable at the vertical level given by the attribute klev. Rank of the result is 2.

oopt_dep Produces the value of a 3-D variable using vertical interpolation at a specified depth given by the attribute dep. Rank of the result is 2.

klev Defines the output vertical level in case oopt equals oopt_klev.

dep — Defines the output water depth (measured positively from the surface) in case oopt equals oopt_dep. Result is 0, if dep is larger than the total water depth at the output location.

node — Used for 3-D variables defined at W-nodes on the model grid. If node is set to 'C' (default), the vertical profile of the variable is first interpolated at the C-node before the operator is applied. If set to 'W', the output variable must be defined at the W-node and no interpolation is performed.It is remarked that quantities defined at U- or V-nodes are always interpolated at the C-nodes before the operator is applied.

The elliptic parameters, available for output are: major axis and minor axis of the tidal ellipse, ellipticity, inclination of the ellipse, elliptic phase and magnitude of the cyclonic and anticyclonic current. The values of the elliptic parameters do not need to be specified by the user, but are automatically determined by the program using the variables's key id. A list of all key ids for elliptic variables is given in Table 20.1, including the dimension of the corresponding "current" vector, which may in principle be any kind of vector and whose components are known to the program through the arrays ivecell2d and ivecell3d. Since they can be defined both in 2-D and 3-D mode, there are 14 key ids available. This explains why the second dimension of ivarsell equals 14. Note that the f90_name, long_name, units of elliptic parameters are pre-defined by the program and cannot be reset by the user in the current implementation.

20.3.2.2 file attributes

File attributes are stored in a DERIVED TYPE array of type FileParams, defined in Section 14.7:

```
TYPE (FileParams), DIMENSION(nosetsanal) :: res0d, res2d, res3d, &
                                            & analgrd
TYPE (FileParams), DIMENSION(nosetsanal,nofreqsanal) :: &
      & amp0d, amp2d, amp3d, pha0d, pha2d, pha3d, ell2d, ell3d
```

The following file attributes can be defined:

defined — The output file will be written only if this parameter is set to .TRUE.. Default is .FALSE..

form — Format of the data file.

'A': ASCII

Table 20.1: List of available elliptic parameters (corresponding variable key id, description of the parameter and the dimension of the vector which determines the ellipse).

Key id	Description	Dimension
iarr_ellmaj2d	Major axis	2
iarr_ellmin2d	Minor axis	2
iarr_ellip2d	Ellipticity	2
iarr_ellinc2d	Inclination	2
iarr_ellpha2d	Elliptic phase	2
iarr_ellcc2d	Cyclonic component	2
iarr_ellac2d	Anticyclonic component	2
iarr_ellmaj3d	Major axis	3
iarr_ellmin3d	Minor axis	3
iarr_ellip3d	Ellipticity	3
iarr_ellinc3d	Inclination	3
iarr_ellpha3d	Elliptic phase	3
iarr_ellcc3d	Cyclonic component	3
iarr_ellac3d	Anticyclonic component	3

'U': unformatted binary

'N': netCDF

filename File name. If not defined, a default will be constructed by the program.

info An "info" file (ending with 'I') will be created if .TRUE. (default value).

header_type No heading information will be written if set to 0. Default is 2. This option is not available for netCDF ('N') files.

The arrays ⋆0d, ⋆2d, ⋆3d are used for output of respectively 0-D, 2-D, 3-D output. They will contain metadata output if header_type is set to its default value or in case of a netCDF file. By default, the data file will contain the coordinates of the output grid and the times of output. The coordinate and time data can also be written to a separate "grid" file whose attributes are stored in analgrd. A selection can be made with the grid_file attribute defined below.

20.3.2.3 output data grid

The output grid (in space and time) and its attributes are defined using the following DERIVED TYPE array:

```
TYPE (OutGridParams), DIMENSION(nosetsanal) :: analgpars
```

gridded If .TRUE. (default), output data are defined on a sub-grid of the model grid (or the whole model grid). If .FALSE., the data are taken at a number of irregularly spaced locations ("station" data) defined below.

grid_file If .TRUE., the coordinates of the output grid will be written on a separate output file (defined by analgrd). Otherwise, they are written within the data file itself. Default is .FALSE.

land_mask A land mask will be applied if .TRUE. and gridded=.TRUE.. This means that the gridded data will be stored as a vector excluding land points. Advantage may be a significant reduction in disk space. Default is .FALSE..

time_format Format of the time coordinate (0).

0: date/time in string fromat

1: seconds

2: minutes

3: hours

4: days

5: months

6: years

7: date in years

Cases 1–6 are numerical formats. Cases 0 and 7 are absolute times, the other ones relative to the reference date/time.

refdate — Reference date/time for calculating relative times. If not given, refdate equals the first output date/time, rounded to the nearest minute, hour, ... depending on the value of time_format.

tlims — The first two elements of this vector are the start and end time indices for data output and determine the start and end output date/times. The period of the harmonic analysis is defined by tlims(3). This means in practice that output will be written atregular intervals of delt2d*tlims(3) seconds which is the same as the period of the analysis. Output times are defined in the middle of each period (central time) of analysis.

nodim — Dimension of the output grid (0/2/3). For example, the dimension must be set to 3 to enable 3-D output.

nostats — Number of data stations in case of non-gridded (station) data.

xlims — Start/end/step X-index in case of gridded data. This defines the output sub-grid in the X-direction. (Option not available for 0-D output). Default values are (1,nc-1,1).

ylims — Start/end/step Y-index in case of gridded data. This defines the output sub-grid in the Y-direction. (Option not available for 0-D output). Default values are (1,nr-1,1).

zlims — Start/end/step Z-index in case of gridded data. This defines the output sub-grid in the Z-direction and applies for gridded and non-gridded output. (Option only available for 3-D output). Default values are (1,nz,1).

20.3.2.4 station attributes

Station atrributes are defined by

```
TYPE (StationLocs), DIMENSION(nostatsanal) :: analstatlocs
INTEGER, DIMENSION(nosetsanal,nostatsanal) :: lstatsanal
```

Arrays and array components have the following meaning

ipos (global) X-index of the output stations

jpos (global) Y-index of the output stations

name descriptive names of the stations. Default is 'Station X' where 'X' is the station's array index.

lstatsanal Station labels. Each file set may contain a sub-set of stations. The element lstatsanal(iset,istat) maps, for set iset, the local station index istat into the corresponding global array index in analstatlocs.

20.3.3 Harmonic output data

The usrdef_ routines below are only called to define the values of user-defined variables. Output data for standard variables are automatically generated by the program and evaluated at the C-nodes (except when the node attribute is set to 'W'). Interpolation is performed when necessary.

Important to note is that the program assumes that all (including user-defined) 2-D and 3-D output data are given at the C-nodes (unless the node attribute is set to 'W'). This means that variables, defined at other nodes, such as velocity components, need to be interpolated first at the C-nodes. Interpolation routines are provided by the program (see Section 31.1).

20.3.3.1 values of 0-D harmonic data

The subroutine usrdef_anal0d_vals defines the 0-D (space-independent) data for harmonic output.

```
SUBROUTINE usrdef_anal0d_vals(out0ddat,n0vars)
INTEGER, INTENT(IN) :: n0vars
REAL, INTENT(OUT), DIMENSION(n0vars) :: out0ddat
```

where

n0vars number of 0-D harmonic variables

out0ddat 0-D output data. The variables are given in the same order as they are defined in the array analvars with the index 1 referring to the first 0-D variable and the index n0vars to the last variable in analvars.

20.3.3.2 values of 2-D harmonic data

The subroutine usrdef_anal2d_vals defines the 2-D (horizontally dependent) data for harmonic output.

```
SUBROUTINE usrdef_anal2d_vals(out2ddat,i,j,n2vars)
INTEGER, INTENT(IN) :: i, j, n2vars
REAL, INTENT(OUT), DIMENSION(n2vars) :: out2ddat
```

where

n2vars	number of 2-D harmonic variables
i	X-index of the data location on the *local* model grid
j	Y-index of the data location on the *local* model grid
out2ddat	2-D output data. The variables are given in the same order as they are defined in the array analvars with the index 1 referring to the first 2-D variable and the index n2vars to the last variable in analvars.

20.3.3.3 values of 3-D harmonic data

The subroutine usrdef_anal3d_vals defines the 3-D (spatially dependent) data for harmonic output.

```
SUBROUTINE usrdef_anal3d_vals(out3ddat,i,j,k,n3vars)
INTEGER, INTENT(IN) :: i, j, k, n3vars
REAL, INTENT(OUT), DIMENSION(n3vars) :: out3ddat
```

where

i	X-index of the data location on the *local* model grid
j	Y-index of the data location on the *local* model grid
k	vertical index of the data location on the model grid
n3vars	number of 3-D harmonic variables
out3ddat	3-D output data. The variables are given in the same order as they are defined in the array analvars with the index 1 referring to the first 3-D variable and the index n3vars to the last variable in analvars.

20.4 User-defined output

The routine usrdef_output is intended for users who like to define output in their own format. The routine has to be programmed by the user at his/hers responsibility. Their are no general rules, except that the routine is called at each 2-D time step. Examples are defined for most test cases and can be found in the subdirectories of **setups**.

20.5 Output grid coordinates

Besides data of model variables, the program writes also the coordinates of the output data grid. They are written, either to the data file itself or to a separate file. The selection is made with the grid attribute grid_file discussed above. In this way, a postprogramming program can read data as well as their locations (on a geographic spherical or an arbitrary Cartesian grid). If the file is written in netCDF format, there is a further advantage since several graphical software programs (MATLAB, FERRET, ncview, ...) accept netCDF data without a need for an intermediate postprocessing program.

The following coordinate arrays are written, depending on the values of the grid attributes nodim and time_grid:

1. xout: X-coordinates of the data grid in meters or (fractional) degrees longitude depending on the value of iopt_grid_sph. Only when nodim>0.

2. yout: Y-coordinates of the data grid in meters or (fractional) degrees latitude depending on the value of iopt_grid_sph. Only when nodim>0.

3. zout: Z-coordinates ($\overline{z}$) in meters of the output grid based on the mean water depth. They are calculated using the formula (see equation (4.19)).
$$\overline{z} = h(\sigma - 1) \tag{20.1}$$
Only when nodim=3.

4. depout: Mean water depth h (bathymetry) in meters. Land points are set to zero. Only when nodim>0.

5. time: Output times written at each output time. Units are absolute (in string format) or numerical depending on the value of the attribute time_format (see above).

6. zetout: Water elevation ζ in meters. This array is written at all times when nodim=3 and time_grid=.TRUE..

The arrays depout and zetout are included for the following reasons:

- If output is written with land_mask=.TRUE., the data are not written in gridded format, but as a vector array excluding land points. The output grid can be recovered by a postprocessing program knowing the positions of the land areas where depout>0.
- If the output grid is taken as time-dependent, the exact vertical positions of the data are then obtained using

$$z = \overline{z}(1 + \zeta/h) + \zeta \tag{20.2}$$

  where $\overline{z}$, h and ζ are obtained from the data file. An easier alternative would be to write z at all times, but this would require the output of a 3-D array at each time, while in the present formulation only one 2-D array (ζ) is needed, giving a saving of disk space.

It may be remarked that in future releases of COHERENS different coordinate data may be implemented to comply with international data standards.