

C++ - OCEAN

MATHIEU DUTOUR SIKIRIĆ

ABSTRACT. We explain here the C++ programs which are designed to work with oceanographic and meteorological data sets.

1. DESIGNS PRINCIPLES

- (1) **Namelist:** Namelists are used by many other oceanographic and meteorological programs (WWM, COSMO, WAM, etc.) and so the same file format is used by the C++ programs.
- (2) **No intermediate files:** It is common to have programs converting from format A to B and then using it. The problem of that approach is that the files in geoscience are usually big and this duplication is pointless.
- (3) **Access to all data files:** Interface have been designed for direct access to all kinds of data files in NETCDF and GRIB formats. Other file format will be added in the future if needed.
- (4) **No distinction between structured and unstructured:** Models come in all kind of shapes and the goal of the program is to accomodate them all.
- (5) **Common interfaces:** Same input structure between different programs
- (6) **Use NCL for plots:** NCL is probably the best language for making 2D graphics in geoscience and offers the largest set of possibilities. NCL is called by scripts written from the C++ program.

2. NAMELIST FORMAT

The input of the programs is done in namelist files that contains the information used. They are formed of blocks of the following kinds.

```
&PROC
  MODELNAME = "WWM"
  GridFile = "hgrid.gr3",
  HisPrefix = "WWM_output_",
/
```

The namelists are done according to following principles:

- (1) The beginning of a block is `&PROC` and the ending is `/`.
- (2) If a variable is not defined in a block then the default value is used.
- (3) A variable cannot be defined two times.
- (4) The blocks and variables can be ordered in anyway the users want. A block may be completely absent in which case all its variables are the default ones.
- (5) Everything after a `!` is considered as comment and not read.
- (6) Strings can be delimited by `"` or `'` or nothing, but endings cannot be mixed.
- (7) When inconsistencies are detected, clear error messages are printed.

3. MODEL DATA INPUT

The supported models are: `WWM`, `WWM_DAILY`, `WAM`, `COSMO`, `WW3`, `ROMS`, `GRIB_ECMWF`, `SCHISM_SFLUX`, `GRIB_GFS`, `GRIB_COSMO`, `GRIB_DWD`. The relevant values in the file are `GridFile`, `HisPrefix` and `MODELNAME`.

Currently the code supports the following models:

- (1) `WWM`: The Wind Wave Model III for wave modelling. 3 formats of grids are supported at the present moment: `.gr3`, (or `.ll` a synonym for `gr3` files), `.dat` (XFN format) and `.nc` (netcdf file). Example of input file:

```
MODELNAME = "WWM"
GridFile = "hgrid.gr3",
HisPrefix = "WWM_output_",
```

or

```
MODELNAME = "WWM"
GridFile = "system.dat",
HisPrefix = "WWM_output_",
```

or

```
MODELNAME = "WWM"
GridFile = "WWM_output_0001.nc",
HisPrefix = "WWM_output_",
```

- (2) `SCHISM_SFLUX` is the case of `SCHISM` sflux files. A typical construction would be

```
MODELNAME = "SCHISM_SFLUX"
GridFile = "sflux_air_1.001.nc",
HisPrefix = "sflux_air_1.",
```

For rain and long/short wave fluxes, use `sflux_prc_1.` and `sflux_rad_1.` Also, for the second forcing if present, use `sflux_air_2.` and similar.

- (3) **WWM_DAILY** is a variant of the above where the files are in disordered form, for example of the form `WWM_output_20160219_00.nc`.
- (4) **WAM**: Both in its structured and unstructured versions.

```
MODELNAME = "WAM"
GridFile = "WAM_output_0001.nc",
HisPrefix = "WWM_output_",
```

(the first history file contains the grid)

- (5) **WW3**: The Wavewatch III model both in its structured and unstructured versions.

```
MODELNAME = "WW3"
GridFile = "WAM_output_0001.nc",
HisPrefix = "WWM_output_",
```

- (6) **COSMO**: The atmospheric model and the netcdf output from the coupled version with ROMS and WAM.

```
MODELNAME = "COSMO"
GridFile = "COSMO_output_0001.nc",
HisPrefix = "COSMO_output_",
```

(the first history file contains the grid)

- (7) **ROMS** (Just ROMS or ROMS_IVICA): The structured model for data output.

```
MODELNAME = "ROMS"
GridFile = "roms_grid.nc",
HisPrefix = "ROMS_output_",
```

ROMS is for normal ROMS output, `ROMS_output.0001.nc`, etc. ROMS_IVICA is for files by date `ROMS_output.20140101.nc`, etc.

- (8) **GRIB** (variant ECMWF, DWD, COSMO, GFS): The GRIB files are essentially standardized but there are small differences between DWD, ECMWF, COSMO and GFS. Also the grid is part of the GRIB file so there is no need for GribFile here.

```
MODELNAME = "GRIB_ECMWF",
HisPrefix = "/home/mathieu/Forecast_input/ECMWF_coarse/",
```

The `GridFile` does not need to be set because the grid is part of the data itself.

4. TIME CONVENTIONS

The times are entered in following format:

```

BEGTC = "20150410.000000",
ENDTC = "20150421.120000",
DELTC = 7200,
UNITC = "SEC",

```

as it this means that we do output from 2015-04-10 00:00:00 to 2015-04-21 12:00:00 at 2 hours of interval.

The possible values for UNITC are SEC, MIN, HOUR and DAY. The Default value is SEC.

The value of DELTC specifies the interval between time output. This is a double precision number so for example the following

```

DELTC = 0.5,
UNITC = "HOUR",

```

means that output is done every 30 minutes.

An alternative for the set up is to put BEGTC and/or ENDTC to ‘‘earliest’’ and/or ‘‘latest’’ in order to put the plots of all data that are available.

For example the following for the plotting software:

```

BEGTC = "earliest",
ENDTC = "latest",
DELTC = 3600,
UNITC = "SEC",

```

means that we plot from the earliest time available in the file to the latest time available.

5. VARIABLES

There are many variables possible. The possible variables are of 4 kinds:

- (1) kind “rho”: those are scalar variables over the domain. For example the bathymetry.
- (2) kind “uv”: those are vector variables over the domain. For example current on the ocean surface.
- (3) kind “3Drho”: those are scalar variables over the domain with a vertical slant. For example the salinity.
- (4) kind “3Duv”: those are vector variables over the domain. For example the horizontal component of the current.

Most variables are of type rho. Thus we indicate the type only if different from rho.

The variables that are available for use and plots are:

Generic model kind of variables:

- (1) **CFL1**: CFL variable number 1

- (2) **CFL2**: CFL variable number 2
- (3) **CFL3**: CFL variable number 3
- (4) **FieldOut1**: Generic Field out 1
- (5) **ThreeDfield1** (3Drho): Generic 3D field out 1
- (6) **IOBPWW3**: The IOBP of the WW3 model
- (7) **MAPSTA**: The MAPSTA of the WW3 model

Atmospheric variables:

- (1) **Uwind**: eastward wind component at 10m. (WWM : Uwind)
- (2) **Vwind**: northward wind component at 10m. (WWM : Vwind)
- (3) **WIND10** (uv): 10m wind speed (WWM : Uwind/Vwind)
- (4) **WINDMAG**: Wind magnitude (WWM : WINDMAG)
- (5) **AIRT2**: 2m air temperature (in deg C)
- (6) **AIRT2K**: 2m air temperature (in deg K)
- (7) **Rh2**: 2m air humidity
- (8) **AIRD**: Surface air density
- (9) **rain**: rainfall rate
- (10) **swrad**: shortwave radiation
- (11) **lwrad**: longwave radiation
- (12) **latent**: latent heat flux
- (13) **sensible**: sensible heat flux
- (14) **shflux**: surface heat flux
- (15) **ssflux**: surface salinity flux (ah ah)
- (16) **evaporation**: evaporation
- (17) **SurfPres**: surface air pressure.
- (18) **AirZ0**: air roughness length (WWM : Z0)
- (19) **AirFricVel**: air friction velocity (WWM : UFRIC)

Oceanic variables:

- (1) **SurfCurr** (uv): Surface currents (WWM : CURTX/CURTY or UsurfCurr/VsurfCurr)
- (2) **UsurfCurr**: Surface current eastward component (WWM : as above)
- (3) **VsurfCurr**: Surface currents northward component (WWM : as above)
- (4) **SurfCurrMag**: Surface currents magnitude (WWM : as above)
- (5) **TempSurf**: Surface sea temperature
- (6) **SaltSurf**: Surface sea salinity
- (7) **ZetaOcean**: free surface elevation (WWM : WATLEV)
- (8) **Curr** (3Duv): Current (WWM : Curr)

Wave variables:

- (1) **CGwave**: group velocity of the waves (WWM: CG)

- (2) **Hwave**: Significant Wave height (WWM: HS)
- (3) **BreakingFraction**: quotient Hwave / Depth (WWM : needs HS and WATLEV)
- (4) **TM02**: zero crossing wave period (WWM : TM02)
- (5) **CdWave**: drag coefficient from the wave model (WWM : CD)
- (6) **AlphaWave**: Charnock coefficient from the wave model (WWM : ALPHA.CH)
- (7) **MeanWaveDir**: mean wave direction (WWM : DM)
- (8) **PeakWaveDir**: peak wave direction (WWM : PEAKD)
- (9) **MeanWaveDirVect** (uv): mean wave direction as direction plot (WWM : DM)
- (10) **PeakWaveDirVect** (uv): peak wave direction as direction plot (WWM : PEAKD)
- (11) **DiscPeakWaveDir**: discrete peak wave direction (WWM : DPEAK)
- (12) **MeanWavePer**: mean wave period (WWM : TM01)
- (13) **PeakWavePer**: peak wave period (WWM : TPP)
- (14) **MeanWaveFreq**: Mean wave frequency (WWM : needs TM01)
- (15) **PeakWaveFreq**: Peak wave frequency (WWM : needs TPP)
- (16) **MeanWaveLength**: Mean wave length (WWM : WLM)
- (17) **PeakWaveLength**: Peak wave length (WWM : LPP)
- (18) **MeanWaveNumber**: Mean wave Number (WWM : KLM)
- (19) **PeakWaveNumber**: Peak wave Number (WWM : KPP)
- (20) **MeanWaveDirSpread**: Mean wave directional spreading (WWM : DSPR)
- (21) **PeakWaveDirSpread**: Peak wave directional spreading (WWM : PEAKDSPR)
- (22) **ZetaSetup**: free surface elevation from the solution of the setup equation (WWM : ZETA_SETUP)
- (23) **TotSurfStr**: total surface stress (WWM : TAUTOT)
- (24) **WaveSurfStr**: wave supported surface stress (WWM : TAUW)
- (25) **SurfStrHF**: high frequency surface stress (WWM : TAUHF)

It is important to note that this is a uniform interface. You specify the model, the variables you need and the software tries to recover them by using specific methodology of the model (GRIB, netCDF, etc.). The same names are used for all models.

The variable can also be composed. For example **Hwave_MeanWaveDirVect** means u/v set to the mean wave direction and the F value set to the significant wave height.

6. PROGRAMS AVAILABLE

The following programs are available:

- (1) **PLOT_results**: It is for plotting data files.
- (2) **PLOT_diff_results**: It is for plotting the difference of model results (models have to be identical and share the same grid).
- (3) **AltimeterComparison**: This is for comparison of model results with altimeter
- (4) **CREATE_sflux**: This is for creating sflux files from finite difference model usable by the SELFE model.
- (5) **INTERPOL_field**: This is for merging several different forcing files and creating an input file for the WWM model.

Other programs can be written by basing oneself on this architecture.

6.1. PLOT_result and PLOT_diff_results. The **PLOT_result** and **PLOT_diff_results** programs are designed to plot model output and plot it. The difference is that **PLOT_result** is for single model output, while **PLOT_diff_results** is for finding the difference between two model outputs.

A minimal example file for **PLOT_results** is given below:

```
&PROC
  MODELNAME = "GRIB_DWD",
  BEGTC = "20150410.000000",
  ENDTC = "20150421.120000",
  DELTC = 7200,
  UNITC = "SEC",
  HisPrefix = "/home/mathieu/Forecast_input/DWD/",
  PicPrefix = "/home/mathieu/Forecast_input/DWD/PlotWind/",
  Extension="png",
/

&VARS
  WIND10 = F,
  WINDMAG = T,
/
```

It means that model output of DWD from the directory `/home/mathieu/Forecast_input/DWD/` are being plotted every 2 hours and that the output directory is `/home/mathieu/Forecast_input`. The extension of the images is `png`.

Variable that is plotted is **WINDMAG** only as selected by the T/F switches. If a variable is not put then it is false by default.

A small example input file for **PLOT_diff_results** is the following:

```
&PROC
```

```

MODELNAME = "WAM",
BEGTC = "20110915.000000",
DELTC = 3600,
UNITC = "SEC",
ENDTC = "20110925.000000",
HisPrefix1 = "WAM_output_",
Name1 = "3 models",
HisPrefix2 = "../RUN_cosmowam/WAM_output_",
Name2 = "2 models",
PicPrefix = "Pictures/WAM_Hwave_coupled_uncoupled/",
Extension= "png",
KeepNC_NCL = T,
NPROC = 2,
/

&VARS
WIND10 = F,
UVsurf = F,
Hwave = T
/

```

It is possible to plot combination of variables. For example, if we write

```

&VARS
Hwave_WIND10 = T,
/

```

then a vector plot of the wind is done with superimposed significant wave height.

6.1.1. *PROC entries meaning.* The `PROC` entry contains all instructions relative to the process itself. Where to access data, where to write it, stuff like that.

List of all options:

- (1) `KeepNC_NCL` (logical): whether to keep the `.ncl` files and input `.nc` files in the temporary directory. Default is false. Put it as true if you want to improve your pictures and/or solve a bug in the `.ncl` program.
- (2) `Extension` (logical): types of the picture file. Default is `png`.
- (3) `ListNatureQuery`: see Section 6.1.2 for explanation.
- (4) `PicPrefix` (string): Where you put your picture. Default value is `Pictures/DIR_plot/`.

- (5) **FirstCleanDirectory** (logical): whether to remove existing file in the directory. Default is true.
- (6) **WriteITimeInFileName** (logical): specifies whether we write the iTime entry in the filename. Default is true.
If true the filename are typically of the form `Hwave_0071_20160208_230000.png` and if false the filename are of the form `Hwave_20160208_230000.png`.
- (7) **OverwritePrevious** (logical): if set to false then previous picture files are not overwritten. Default is true.
- (8) **NPROC** (integer): Number of processors used for making the plots. Default is 1. If **KeepNC_NCL** is selected then NPROC should be equal to 1. The parallelization is over the NCL programs.

6.1.2. *Nature query option.* We have following possible queries for the **PLOT_result**:

```
ListNatureQuery = 'instant', 'average', 'swathMax', 'swathMin'
TimeFrameDay = 30
```

By default only 'instant' is selected and value of the field at the precise time are plot. If 'average' is selected then we have the plots of average value of the field for the period $[t, t + T]$ with t the time just as for instant field and T the value of **TimeFrameDay**. So a $T = 30$ corresponds to monthly averages. If 'swathMax' or 'swathMin' is selected then the maximum and minimum values over the time intervals $[t, t + T]$ are also plot.

Above options are also possible for **PLOT_diff_result**. But two more options 'MaxDiff' and 'MinDiff' are also possible. MaxDiff/MinDiff plots the maximum/minimum difference over the interval $[t, t + T]$. This is different from swathMax/swathMin which would plot the difference between the maximum over the interval of the models.

6.1.3. *Colorbar selections.* We have following options for the colormaps:

```
&PLOT
  DoColorBar = T
  BoundSingle_var = Hwave, TM02
  BoundSingle_min = 0,    5
  BoundSingle_max = 4,    20
  BoundDiff_var = Hwave, TM02
  BoundDiff_min = -0.2, -2
  BoundDiff_max = 0.2,  2
  VariableColormap = F
/
```

By default a fixed colorbar is used for the plots. The range is fixed in the code for each variable and is generally reasonable. If you want to change the minimum/maximum, then you need to use `BoundSingle_var/min/max` for `PLOT_result` and `BoundDiff_var/min/max` for `PLOT_diff_results`.

If you want a colorbar that varies from one time to the next then use `VariableColormap` and the maximum and minimum of the field will be used. If you do not want a colorbar to be plotted, then select `DoColorBar = F`.

6.1.4. *Selection of subregion.* By default we plot the data over a rectangle defined by the longitude and latitude of the grid considered. This grid is called the **main** grid. The relevant entries to this issue are:

```
&PLOT
  DoMain = F,
  ListFrameMinLon = -10, -5
  ListFrameMinLat = -10, 5
  ListFrameMaxLon = 10, 20
  ListFrameMaxLat = 10, 20
/
```

The variable **DoMain** specifies whether we plot the main region or not. By default it is true. The **ListFrame** variables specify the other rectangle considered. With the above choice we have two regions being considered. One is from longitude -10 to 10 and latitude -10 to 10. The other is from longitude -5 to 20 and latitude 5 to 20.

6.1.5. *Miscellaneous options.* Following options are available in the `PLOT` section:

- (1) `DoTitle` (logical): draw the title of the figures. Default is true.
- (2) `PlotDepth` (logical): draw the bathymetry of the domain if available. Default is true.
- (3) `PrintMMA` (logical): Print the minimum / maximum / average values of the fields considered. Default is false
- (4) `LocateMM` (logical): locate the position of maximum and minimum values. Default is false.
- (5) `FillLand` (logical): Fill the land by using the geographic database of NCL.
- (6) `cnSmoothingOn` (string): whether to apply smoothing. Default is true.
- (7) `cnFillMode` (string): Method used for drawing. Available methods are:

- (a) "RasterFill": It plots values of the cell corresponding to the field considered.
 - (b) "AreaFill": It plots smoothed values of the fields
 - (c) "CellFill": It is a smoothed field (only for finite difference)
- Default is RasterFill.
- (8) **UseFDgrid** (logical): sometimes it is better to do plots using finite difference grids, even if the grid is unstructured. Default is false.
- (9) **RenameVariable_VarName1** and **RenameVariable_VarName2** (list of string): Those two variables can be used to rename the variable used for output.

For example

```
&PLOT
  RenameVariable_VarName1 = "Hwave",
  RenameVariable_VarName2 = "Wave height",
/
```

will use the word "Wave height" in the picture title for the plots of the Hwave variable instead of the standard name "Significant wave height".

6.2. AltimeterComparison. The data set used is the one from IFREMER. It needs first to be downloaded. First one needs to set the environment variable **ALTIMETER_DIRECTORY** as in for example

```
export ALTIMETER_DIRECTORY=/home/mathieu/Altimeter/
```

Then one has to download the data file with the perlscript **Download-AltimeterIfremer**.

See below an example of nml files

```
&PROC
  MODELNAME = "COSMO",
  GridFile = "COSMO_output_0001.nc",
  HisPrefix = "COSMO_output_",
  PicPrefix = "./AltimeterStat/"
  Extension="png",
  KeepNC_NCL = T,
/
```

```
&SELECT
  GEOSELECTION = 2,
  MinLON = -7,
  MaxLON = 37,
  MinLAT = 30,
```

```

MaxLAT = 46
LONPOLY = 14, -6, -6, 40, 40, 28
LATPOLY = 49, 39, 26, 29, 39, 39
MinWIND = 0
MaxWIND = 300
MinHS = 0
MaxHS = 998
BEGTC = 20101101.000000
ENDTC = 20101231.000000
MinimalTrackSize=30,
EliminationShortTrack = F,
DoTrackSmoothing = F,
/

&PROCESS
USE_CORRECTED = T
DO_WNDMAG = T
DO_HS = T
DO_STAT = T
DO_NCOUT = T
DO_TXTRAW = F
DO_SCATTERPLOT = T,
PLOT_TRACKS = T,
PLOT_ALL_TRACKS = T,
DO_SAVE_TXT = F,
SPATIALAVER = F
/

```

The section **PROC** is standard for the model input. Also added is the Prefix for the pictures, their extension and whether we keep the ncl files for further work.

The section **SELECT** contains information on the processing done:

- (1) **GEOSELECTION** is for geographical selection of the zone of interest. **GEOSELECTION=1** means using MinLon, MinLat, MaxLon, MaxLat for the selection. **GEOSELECTION=2** means using LONPOLY/LATPOLY that defines a polygon.
- (2) **MinWind**, **MaxWind** is for thresholding the wind values.
- (3) **MinHS**, **MaxHS** is the same for significant wave height.
- (4) **BEGTC**, **ENDTC** is for specifying the period of interest.
- (5) **AllowedSatellites** is for selecting the satellites of interest. If not then all satellites are used.

- (6) **EliminationShortTrack** is a logical for whether we eliminates the short tracks and **MinimalTrackSize** is the minimal allowed track size.
- (7) **DoTrackSmoothing** is for whether we smooth the tracks to the length scale of the models.
- (8) **DoMinDistCoast** is for whether we filter the tracks by the distance to the coast. **MinDistCoastKM** is the minimal distance to the coast that is specified. Also needed is the **LonLatDiscFile** for specifying the longitude/latitude of the coast.

The section **PROCESS** specifies what will be done with the data:

- (1) **DO_WNDMAG**: specifies whether we compare wind speed or not.
- (2) **DO_HS**: specifies whether we compare significant wave height or not.
- (3) **DO_STAT**: specifies whether we do raw statistics comparison (Mean Error, Root Mean Square Error, etc.). It is the cheapest possible comparison.
- (4) **DO_SCATTERPLOT**: specifies whether we do scatter plot of the data or not.
- (5) **USE_CORRECTED**: specifies whether we use corrected data from the altimeter or not.
- (6) **PLOT_ALL_TRACKS**: specified whether we plot all tracks used in the geographical domain.
- (7) **PLOT_TRACKS**: We can plot the model and the altimeter interpolation. **MinEntryTrackPlot** specifies the minimal length for the plot to be made.

6.3. CREATE_sflux. This program is for creating sflux files that can be used by the SELFIE program.

```
&PROC
MODELNAME = "GRIB_ECMWF",
BEGTC = "20150410.000000",
ENDTC = "20150421.120000",
DELTC = 21600,
UNITC = "SEC",
HisPrefix = "/home/mathieu/Forecast_input/ECMWF_coarse/",
OutPrefix = "/home/mathieu/Forecast_input/ECMWF_coarse/sflux/sflux_",
AnalyticWind = F,
AnalyticPRMSL = F,
AnalyticSPFH = T,
AnalyticSTMP = F,
AnalyticPRATE = T,
```

```

AnalyticDLWRF = T
AnalyticDSWRF = T
/

```

There is only a PROC entry, which specifies the model that is used. DELTC contains the interval between data output. The sflux files are daily files so DELTC must be a divisor of the length of the day. OutPrefix contains the prefix where the data is written.

The AnalyticWind and others specifies whether we use analytic values for the field used in the model. More precisely:

- (1) AnalyticWind: If true the wind is send to zero
- (2) AnalyticSPFH: If true the 2m air humidity is set to zero
- (3) AnalyticSTMP: If true the 2m air temperature is set to 15 deg C.
- (4) AnalyticPRMSL: If true the surface air pressure is set to 105384.9 Pascal.
- (5) AnalyticPRATE: If true the rate of precipitation is set to 0.
- (6) AnalyticDLWRF: If true the downward longwave flux is set to 0
- (7) AnalyticDSWRF: If true the downward shortwave flux is set to 0

6.4. MERGE_field. This program is for interpolating the fields from several models, and merging them in order to create one single series of forcing file that can be used for running WWM (only WWM right now).

An example of input file is below:

```

&INPUT
ListMODELNAME = "GRIB_ECMWF", "GRIB_ECMWF"
ListHisPrefix = "/home/mathieu/Forecast_input/ECMWF_coarse/", "/home/mathieu/F
ListSpongeSize = 4, 4
ListFatherGrid = -1, 0
/

&OUTPUT
MODELNAME = "WWM"
GridFile = "/home/mathieu/Forecast_input/MERGE_ECMWF/hgrid.gr3",
HisPrefix = "File.nc",
HisPrefixOut = "/home/mathieu/Forecast_input/MERGE_ECMWF/Forc",
BEGTC = "20150410.000000",
ENDTC = "20150421.120000",
DE LTC = 3600,
UNITC = "SEC",

```

```

    DEFINETC = 86400,
    WriteIFile = T,
    WriteDate = F,
/

```

```

&VARS
    WIND10 = T,
/

```

The **INPUT** field contains the list of model runs available. So we have a list of model names, grid files and prefix.

The **OUTPUT** array contains the description of the model output. We have a **MODELNAME**, **GridFile** and **HisPrefix** for the description of the model output. We also have a **HisPrefixOut** which is the prefix of the forcing files. Similarly, we have the **BEGTC**, **ENDTC** for the time frame of the data output, **DELTC** for the frequency of the output and **DEFINETC** for how many seconds a file will contain.

The **WriteIFile** specifies whether we put the file number in the filename. The **WriteDate** specifies whether we put the first file data in the filename. For example

```

&OUTPUT
    HisPrefixOut = "Forc",
    WriteIFile = T,
    WriteDate = F,
/

```

will give you file names of the form **Forc_0001.nc** while

```

&OUTPUT
    HisPrefixOut = "Forc",
    WriteIFile = F,
    WriteDate = T,
/

```

will give you file names of the form **Forc_20160415_000000.nc**

The **VARS** array contains the list of variables that are used for the output.

MATHIEU DUTOUR SIKIRIĆ, RUDJER BOSKOVIĆ INSTITUTE, BIJENICKA 54,
 10000 ZAGREB, CROATIA, FAX: +385-1-468-0245
E-mail address: mdsikir@irb.hr