

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. MODEL DATA INPUT

The supported models are: WWM, WAM, COSMO, WW3, ROMS, GRIB_ECMWF, 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**, **.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) 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)

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

```

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

```

- (4) 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)

- (5) ROMS: The structured model for data output.

```

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

```

- (6) 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/",

```

3. TIME AND VARIABLES

The times are entered just as in WWM. Example:

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

The variables that are available for use and plots are:

- (1) **WIND10**: 10m wind speed (as a vector with Uwind, Vwind components)
- (2) **SurfCurr**: Surface currents
- (3) **Hwave**: Significant Wave height
- (4) **WINDMAG**: Wind magnitude
- (5) **TempSurf**: Surface sea temperature
- (6) **SaltSurf**: Surface sea salinity
- (7) **AIRT2**: 2m air temperature
- (8) **Rh2**: 2m air humidity
- (9) **ZetaOcean**: free surface elevation
- (10) **MwaveFreq**: Mean wave frequency
- (11) **PwaveFreq**: Peak wave frequency
- (12) **AIRD**: Surface air density
- (13) **CdWave**: drag coefficient from the wave model
- (14) **AlphaWave**: Charnock coefficient from the wave model
- (15) **rain**: rainfall rate
- (16) **swrad**: shortwave radiation
- (17) **lwrad**: longwave radiation
- (18) **latent**: latent heat flux
- (19) **sensible**: sensible heat flux
- (20) **shflux**: surface heat flux
- (21) **ssflux**: surface salinity flux (ah ah)
- (22) **evaporation**: evaporation
- (23) **MwavePer**: mean wave period
- (24) **PwavePer**: Peak wave period
- (25) **SurfPres**: surface air pressure.

4. PROGRAMS AVAILABLE

The following programs are available:

- (1) **PLOT_result**: 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 SELFIE 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.

4.1. **PLOT_result**. A minimal example file 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.

4.2. **PLOT_diff_results**. This is plotting results of two different models. An example file:

```
&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,
/

```

```

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

```

This is the plot of coupled vs uncoupled WAM results

4.3. 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.

4.4. **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,
/

```

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 wind (and hence zero) if no wind is available or whether we use the data available from the model. Same for `AnalyticPRMSL`, `AnalyticSPFH` and `AnalyticSTMP` switches.

4.5. **INTERPOL_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",
  DELTC = 3600,
  UNITC = "SEC",
  DEFINETC = 86400,
/

&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, **DELT**C for the frequency of the output and **DEFIN**ETC for how large a single file should be.

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