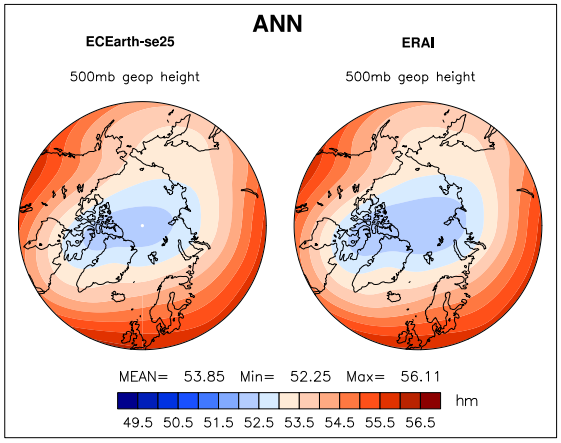
EMoP user manual

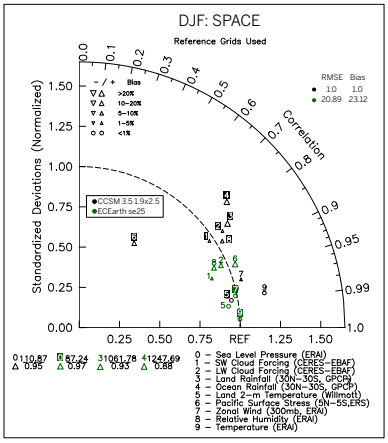
Laurent Brodeau[1] & Klaus Wyser[2]

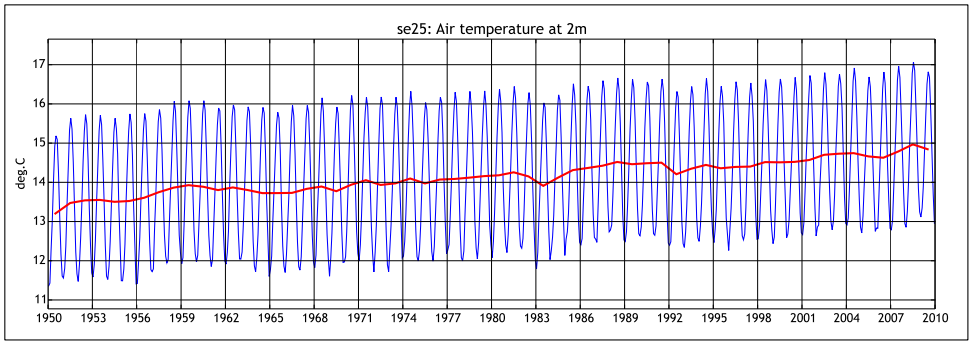
[1] Department of Meteorology, Stockholm University, Sweden

[2] Rossby Centre, SMHI, Sweden

September 26, 2014







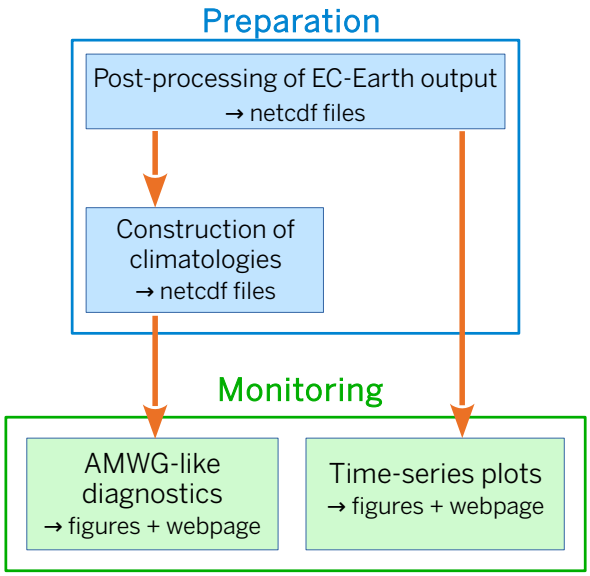
Contents

1 About EMoP

EMoP is a collection of scripts to produce a wide variety of diagnostics and plots of EC-Earth simulations. It includes a modified and adapted version of the AMWG diagnostic package developed at NCAR. EMoP relies solely on free software.

EMoP action is done through 2 successive phases:

- Phase I, the *preparation* phase in which monthly climatologies of variables of interest are extracted from raw IFS grib files and saved into Netcdf format on a regular Gaussian grid and pressure levels for the atmosphere. Relevant scripts are found in the `prepare_clim` sub-directory. The following collection of software is needed for phase I:
  - `NCO`
  - `GRIB_API`
  - `CDO` (compiled with `GRIB_API` support)
  - `CDFTOOLS` (if you are interested in ocean fields)
- Phase II, the *monitoring* phase in which high-quality figures and browsable HTML pages are created using the climatology files created during phase 1. The following collection of software is needed for this phase:
  - `NCL`
  - `Python` environment, with the following packages: `NumPy`, `Matplotlib`, `Netcdf4`. Only for the time-series tool



1.1 Installing EMoP

EMoP can simply be installed using Subversion. In a Unix environment, this is done with the following command:

```
svn co svn://svn.code.sf.net/p/emop/code/trunk emop
```

Since EMoP is a collection of script, no compilation step is needed.

1.2 The master configuration file

The idea behind this file is that all the case/machine-related information, like computer/architecture, paths to I/O directories, etc, is centralized and written once for all in a single file that will be used by all the different components of EMoP: `conf_<MY_SETUP>.bash`. A template version of this file is provided as `conf_example_machine.bash`. It is sufficiently commented.

2 Phase I: Construction of monthly climatologies

When you run EC-Earth, a minimum set of variable saved by IFS should be saved in order for the diagnostic tool to present a descent selection of diagnostics and for the AMWG diagnostic tool to work properly.

2D Atmospheric variables to save at the monthly frequency are:

```
ps msl uas vas tas e stl1 tcc totp cp lsp ewss nsss sshf slhf ssrd strd
ssr str tsr ttr tsrc ttrc ssrc strc lcc mcc hcc tcwv tclw tciw fal
```

3D Atmospheric variables to save at the monthly frequency are:

```
q rh t u v z
```

Also have a look into file `prepare_clim/1_post_process/ecearth.tab`

2.1 Post-processing tool

This set of scripts is developed by Klaus Wyser at SMHI and is found in the following sub-directory:

```
prepare_clim/1_post_process
```

The goal of this post-processing step is to extract data from IFS grib output files (and NEMO Netcdf output) and save as it as monthly means in human-friendly Netcdf files with the following convention: 1 file per field and per year. Each of these files containing 12 monthly records, the field can be either 2D or 3D.
This step is mandatory to be able to complete any of the monitoring described in this document.

The post-processing tool consists of the following scripts and files:

```
master-post.sh
ifs-post.sh
nemo-post.sh
ecearth.tab
META.sh
```

`master-post.sh` is the main script that you need to adjust for your experiment. It is documented and should be a problem to modify it for your needs (see section 2.1.3). `master-post.sh` then calls `ifs-post.sh` and `nemo-post.sh` to make the post-processing for atmosphere and ocean, respectively. No modifications are needed in these two scripts (as long as you do not need more variables or a different diagnostics).

Then there is `ecearth.tab` that contains the translation from GRIB parameters to Netcdf variable names and units, needed for cdo. The names and description of variables in this file are copied from ECMWF (with minor modifications), mind that it is not CMIP5 standard.

2.1.1 Requirements for extracting atmospheric fields from IFS

By setting `i_do_ifs=1` into `master-post.sh`, climatologies for atmospheric fields will be built.

Specifically, you need the following software installed on your system

Set GRIB_API_BINDIR to point to the directory containing grib_api

2.1.2 Requirement for extracting ocean fields from NEMO

If you are interested in any field originating NEMO, set i_do_nemo=1 into master-post.sh, climatologies for atmospheric fields will be built. The following files and software should be present on your system.

You will need the grid description files for NEMO that matches your current configuration: mask.nc, mesh_zgr.nc and mesh_hgr.nc. These files can be generated online with NEMO by setting the proper namelist parameter (nn_msh). Recent versions of NEMO only produce a mesh_mask.nc, in that case just replace mask.nc, mesh_zgr.nc, mesh_hgr.nc by symbolic links pointing to mesh_mask.nc. In addition a land-sea mask file containing the description of the different ocean basins is needed: new_maskglo.nc. Specify variable NEMO_MESH_DIR and MESH_MASK_ORCA into your conf_<MY_SETUP>.bash configuration file.

To rebuild NEMO output from processor-sub-domain files to global files, the script rebuild and (which relies on the flio_rbld.exe executable) should be installed. These two come along with any recent version of NEMO need to be compiled and installed. Specify variable RBLD_NEMO into conf_<MY_SETUP>.bash. The CDFTOOL package developed by J.M. Molines in Grenoble is the official set of diagnostic tools used within the DRAKKAR consortium it is downloadable via svn via http://servforge.legi.grenoble-inp.fr/projects/CDFTOOLS. Once compiled, specify the variable containing the CDFTOOLS executables into variable CDFTOOLS_BIN of your conf_<MY_SETUP>.bash.

2.1.3 Adjusting and launching master-post.sh

Make sure that the conf_<MY_SETUP>.bash in the root directory of EMoP is configured according to your needs and your system. master-post.sh relies mainly on information sourced into conf_<MY_SETUP>.bash, but a few adjustment must should be done in master-post.sh by setting up i_do_ifs and i_do_nemo.

You can launch a post-process job in batch mode to extract a given year (year_to_treat):

```
$ master-post.sh <MY_SETUP> <EXP> <reference_year> <year_to_treat>
```

EXP is the name of the experiment, reference_year is usually the first year of you simulation.

Since this step is resource-consuming we advise you to use a batch job manager (example is SLURM with the command sbatch). Within a batch job it is possible to work with a few nodes in parallel (example 4 cores):

```
$ sbatch -N 1-1 -n 4 master-post.sh <MY_SETUP> <EXP> \
                                          <reference_year> <year_to_treat>
```

(Note the -N 1-1 flag, very important! Refer to your own batch job manager documentation to find an equivalent setup)

In addition to master-post.sh, there is also a META.sh in the same directory that does the loop over several years. Comes in handy when processing long experiments. Configure META.sh to your needs and batch system and launch it in the background. It will run in the background and launch the job for the next year once the current year has been processed:

```
$ nohup ./META.sh > meta.out &
```

If everything went according to plan, the directory POST_DIR (defined into conf_<MY_SETUP>.bash) will contain 1 Netcdf file per year and per extracted field.

2.2 Building NCAR-friendly climatologies

The relevant script (NCARIZE_b4_AMWG.sh) is found in the following sub-directory:

```
prepare_clim/2_ncarize/
```

The goal of this second post-processing step is to create monthly, seasonal and annual climatologies for a given period of your run (usually the end) in a NCAR-friendly fashion. In order to create these climatologies, the Netcdf files produced via the former post-processing step (see previous section 2.1.3 must be present into <POST_DIR>. Usually, you only want to build a climatology out of the last years of your run, it is enough to base this climatology on the last 20 or even 10 years of your simulation.
Note that this second post-processing step is only needed if you plan on using the AMWG monitoring package (section 3.2). It is not needed if you only want to plot time-series (section 3.1).

To launch the creation of the climatologies for the period <y1>-<y2>, type the following command:

```
$ NCARIZE_b4_AMWG.sh -C <MY_SETUP> -R <EXP> -g <GAUSS_RES> \
                                -i <y1> -e <y2>
```

This will create and save the Netcdf climatology files that the AMWG monitoring package is expecting as input into directory:
<EMOP_CLIM_DIR>/clim_<EXP>_<y1>-<y2>
This directory should contain monthly, seasonal and annual climatologies. You can jump to section 3.2 to create the AMWG diagnostics and HTML pages.

3 Phase II: Creation of time-series, figures and HTML pages

3.1 Extracting and plotting global atmospheric time-series

In order to be able to produce the following diagnostics and plots, only the Netcdf files produced by the first "post-processed" stage (see section 2.1) must be present into <POST_DIR>.

Move to monitoring/time_series.

Needed: Python with the following modules: NumPy, Netcdf4 and Matplotlib with Basemap support. A way to easily get a proper and up-to-date Python installation is to install the *Enthough* Python environment: Canopy.

Extract time series into a single Netcdf file:

```
$ sbatch ./monitor_atmo.sh  -C <MY_SETUP> -R <EXP>
$ sbatch ./monitor_ocean.sh -C <MY_SETUP> -R <EXP>
```

These commands can be run over and over when new data are produced from EC-Earth and have been processed via the *Post* step (section 2.1). These 2 scripts will keep on appending new data to the 2 following Netcdf files:

```
<DIR_TIME_SERIES>/<EXP>/atmosphere/<EXP>_<y1>_<y2>_time-series_atmo.nc
<DIR_TIME_SERIES>/<EXP>/ocean/<EXP>_<y1>_<y2>_time-series_ocean.nc
```

When the former step has been successfully completed and you want to generate the *time-series* diagnostic HTML page for the atmosphere and the ocean, run the same commands but with adding the -e switch:

```
$ ./monitor_atmo.sh  -C <MY_SETUP> -R <EXP> -e
$ ./monitor_ocean.sh -C <MY_SETUP> -R <EXP> -e
```

Figures and HTML page are created in the same directories as for the Netcdf files containing the time-series: <DIR_TIME_SERIES>/<EXP>/<realm>/

They will also be exported to your web server if you have specified the following variables in the master configuration file:
RHOST: hostname of the remote host
RUSER: username on the remote host
RWWWD: full-path directory that should contain diagnostics on the remote host

3.2 AMWG plots and HTML pages

You should have successfully created the *NCAR-friendly* climatologies (as explained in section 2.2). They should be into <EMOP_CLIM_DIR>/clim_<EXP>_<y1>-<y2>.

You need ncl installed on your system (version 6.1.0 and later).

```
$ cd monitoring/amwg_diag/
```

3.2.1 Run against observations

Climatologies built from run <EXP> are compared against a wide collection of observations (reanalyzes, satellite...). These observations are stored into directory <DATA_OBS>, a tarball of this directory can be downloaded here: http://misu228.misu.su.se/EMoP/obs_data_5.5.tgz

To compare the climatology of run <EXP> under the period <y1>-<y2> against observations fire the following command:

```
$ sbatch ./diag_mod_vs_obs.csh <MY_SETUP> <EXP> <y1>-<y2>
```

Here, <y1>-<y2> obviously has to be the same as in the climatology you created:
<EMOP_CLIM_DIR>/clim_<EXP>_<y1>-<y2>

If everything goes well, browsable diagnostics should be created and stored into:
<EMOP_CLIM_DIR>/diag_<EXP>_<y1>-<y2>/<EXP>-obs_<y1>-<y2>
Just open the index.html file with your web browser.

3.2.2 Run A against Run B

It is also possible to compare two of your runs against one another, provided the *NCAR-friendly* processing step has been successfully completed for both of these runs. The periods on which climatologies are built do not necessarily have to be the same. For example, to compare run <EXPA> against (reference) run <EXPB>:

```
$ sbatch ./diag_mod_vs_mod.csh <MY_SETUP> <EXPA> <y1_a>-<y2_a> \
                                          <EXPB> <y1_b>-<y2_b>
```

If everything goes well, browsable diagnostics should be created and stored into:
diag_<EXPA>_<y1_a>-<y2_a>/<EXPA>_<y1_a>-<y2_a>_-_<EXPB>_<y1_b>-<y2_b>
Just open the index.html file with your web browser.