

A regional model coupling
with OASIS3-MCT
Eric Maisonnave
WN/CMGC/13/34

Table of Contents

Configuration set up.....	3
Debug and performance measure options.....	5
Coupling fields.....	5
Verbosity.....	5
Load balancing and performances.....	6

Configuration set up

Models, libraries and scripts described in this document are available on CSCS rosa supercomputer (mentioned directories were up-to-date in May 2013)

A new configuration of an OASIS3-MCT based coupled model must be set up providing input informations (grid definition, partitioning, initial conditions, forcing) for each component of the system, and to the coupling library itself. This chapter indicates the different steps the user have to follow to start his simulation on a new configuration.

For COSMO, there is no need to change the input information that should be normally provided for a forced run.

At the opposite, input files of CLM (and of the DATM dummy atmosphere model including in CESM) required modification to optimally contribute to the coupled system¹.

Actually, due to the properties of their domain limits (lat-lon rectangular for CLM, rotated grid for COSMO), both domain necessarily differs. We implemented a methodology to select only the subset of CLM original grid points that fits COSMO domain limits.

STEP 1

Goal:

to produce a netcdf file (mask_clm.nc) including a land/sea mask variable (mask) on CLM grid

Actions:

- recompile CLM after modifying oas_clm_vardef.F90 file:
INTEGER :: IOASISDEBUGLVL = 2
- recompile COSMO after modifying oas_cos_vardef.F90 file:
INTEGER :: IOASISDEBUGLVL = 2
- launch a test simulation (could takes a few seconds or several hours, depending on model resolution) after modifying launching script parameter in user_defined file:
> set CUT_OUT_CLM = 1

Using a new namcouple (namcouple.base.define_clm_limits), reducing to 1 the number of sub-domain dedicated to DATM, OASIS interpolates the COSMO domain limits on the CLM grid.

¹ Additionally, the OASIS3/OASIS3-MCT upgrade has an impact on CLM coupling. Due to the fact that CESM (CLM) already uses MCT as an internal coupling library, it was necessary to transform the OASIS released version. All the modules of the MCT library used by OASIS has to be renamed (_oasis suffix), every calls to those module have to be redefined on PSMILE library and the names of the two mct/mpeu libraries have to be changed (_oasis suffixes). All the necessary operations has been gathered in two shell script /project/s433/emaision/oasis3-mct/sh_cesm_compliance and sh_duplicate_lib

STEP 2 (optional)

Goal:

Customize mask produced on step 1 and get the result on a new file (new_mask_clm.nc)

Actions:

- launch "persona" tool to modify binary variable thanks its GUI interface (see <http://www.cerfacs.fr/~maisonna/Persona/readme.html> User guide)
> /users/emaizon/bin/persona mask_clm.nc

STEP 3

Goal:

Substitute original CLM/DATM land/sea mask input files by newly created variable

Actions:

- use mask variable from mask_clm.nc file to replace (with nco tools, for example) frac et mask variables in surf_datm.nc file and LANDFRAC et LANDMASK surfdata_0122x0276.nc file (be careful with dimension names)

STEP 4

Goal:

Produce OASIS auxiliary (masks.nc, grids.nc and areas.nc) and interpolation weights (rmp*.nc) files

Actions:

- recompile CLM after modifying oas_clm_vardef.F90 file:
INTEGER :: IOASISDEBUGLVL = 0
- recompile COSMO after modifying oas_cos_vardef.F90 file:
INTEGER :: IOASISDEBUGLVL = 0
- models are now ready to be launched for a second test simulation (again, it could takes a few seconds or several hours, depending on model resolution) after modifying launching script parameter in user_defined file:
> set CUT_OUT_CLM = 2
- save auxiliary and weights files produced. You will copy them in the working directory at the beginning of each new simulation

STEP 5

Goal:

launch a production run

Actions:

Modify the script parameter

```
> set CUT_OUT_CLM = 0
```

and re-launch the model

Debug and performance measure options

Coupling fields

If you want to visualize your exchanged coupling fields after and before interpolation, you can modify the script parameter in `user_defined` file

```
> set CUT_OUT_CLM = 3
```

This will replace the EXPORTED instructions by EXPOUT in `namcouple` parameter file (see OASIS User guide). OASIS library will output variables in netcdf files (two per coupling field, after/before interpolation) at coupling time step frequency.

Notice that those files can be read by OASIS and used to force one of the two models. An example of scripts necessary to launch such “forced” simulation can be found here: `/project/s433/emaision/cclm4.8_19.2_oas_mct/run/low_res/forcing`

Forcing variable has to be defined on `namcouple` as follow:

```
CLMUWIND CLMUWIND 182 _cpl_freq_ 0 CLMUWIND_clmxxx_02.nc INPUT
```

where `CLMUWIND_clmxxx_02.nc` is the name of the netcdf file produced with EXPOUT option in a previous run.

Verbosity

If you need a verbose version of OASIS routines, change NLOGPRT value (up to 50) in `namcouple` parameter file. For more verbosity of the model/coupler interface:

- recompile CLM after modifying `oas_clm_vardef.F90` file:
INTEGER :: IOASISDEBUGLVL = 1
- recompile COSMO after modifying `oas_cos_vardef.F90` file:
INTEGER :: IOASISDEBUGLVL = 1

Load balancing and performances

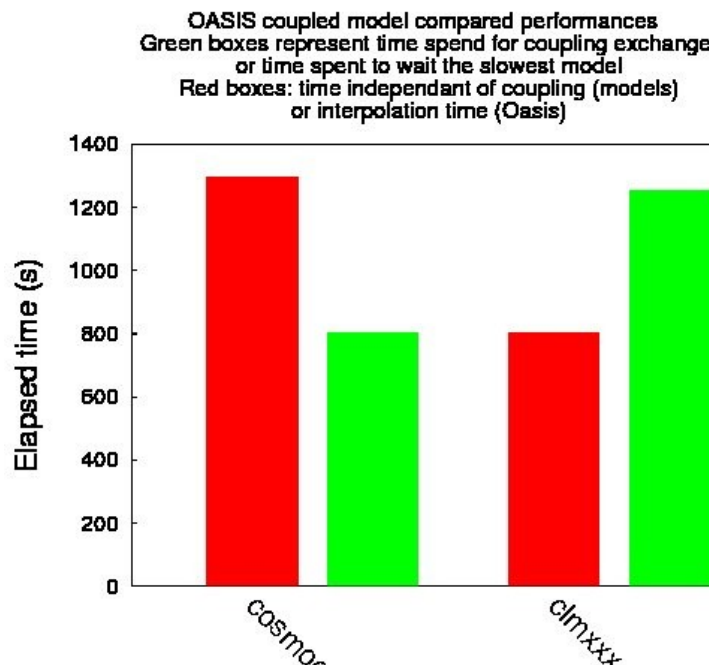
Due to its particular configuration (2 executables are running at the same time), it is more difficult to finely measure performances of an OASIS based coupled model than a single executable. When two models run sequentially (as it is the case with COSMO-CLM²), one model is waiting the results of other's calculations. It could be of some interest to determine how much time take calculations of each model without running them previously on a stand alone mode.

To measure this load balance between COSMO and CLM:

- compile OASIS with "balance" CPP key
- link COSMO and CLM with the OASIS library you produced
- launch model (preferentially without producing OASIS coupling fields in netcdf files, that could slow down the simulation)
- launch "lucia" script on your working directory (where debug.0?.?????? files are produced)
> /users/emaision/Lucia/lucia-mct

This tool produces 2 different kinds of output:

- An eps file with computing time / waiting time for COSMO and CLM components (see picture below as an example for 256 cores allocated for each)



- A list of informations on standard output (screen) including:
 - the list of exchanged fields
 - the same information than provided graphically but with exact figures (and error margin)

- o additional information such as OASIS communication time and model jitter

The so called “computing” time (red boxes) represents the total elapsed time of each model, excluding initialization and restart/ending time and the time needed to perform OASIS put and get operations. On the other hand, this time needed to perform OASIS put and get operations (when a model is waiting the result of the other model) is shown in green boxes.

The computing time can be used to determine scalability of each model (as part of the coupled system). This has been done for the CORDEX 11km configuration of the model. In the figure below, we show how different scalability (left) or parallel efficiency (right) can be calculated (varying core number allocated for each model, adding the two to get coupled model scalability)

