OASIS3-MCT tutorial

All the documentation about the coupler OASIS3-MCT can be found on the OASIS web site at http://oasis.enes.org and in the OASIS3-MCT sources in the oasis3-mct/doc directory.
The current OASIS3-MCT coupler uses internally the Model Coupling Toolkit (MCT) developed by the Argonne National Laboratory (http ://www.mcs.anl.gov/mct) to perform parallel regridding and parallel exchanges of the coupling fields.

After the modifications you will bring following the steps below, the "tutorial" toy model should reproduce the coupling between two simple codes, model1 and model2, with the OASIS3-MCT coupler. Note that after modifications, your sources model1.F90, model2.F90 and data_oasis3/namcouple should be similar to what you can find in model1.F90_TP, model2.F90_TP and data_oasis3/namcouple_TP. If you want to run directly the coupled configuration without modifying yourself the sources, just copy the *_TP sources removing the _TP suffix, recompile model1 and model2 (see A. below), adapt run_tutorial to your platform and execute this script.

I – Creation of the configuration file of OASIS3-MCT (namcouple) for the tutorial

The aim of this first part is to create the *namcouple* text file used for the ***tutorial*** toy coupled model you will have to interface with OASIS3-MCT in part II. The *namcouple* contains all the user-defined information necessary to configure a particular coupled run.

A. Extracting the OASIS3-MCT sources

1. Login with login and password provided and open a terminal session clicking Applications -> System Tools -> Terminal in the top bar. Download the OASIS3-MCT sources of the official version 2.0 :
   - mkdir oasis3-mct ; cd oasis3-mct
   - Open a browser window and go to the OASIS web site to register and download the sources, at the address: https://verc.enes.org/oasis/download/oasis-registration-form

B. Launching the OASIS3-MCT GUI

The OASIS3-MCT GUI is an application of C3SM, a graphical interface used by different codes, and developed at CERFACS by the Combustion team. It is located in the repository C3SM of your $HOME. A short description of the OASIS GUI is given in the README file.

1. Create a .csrhc file in your $HOME. Open it and define the following PATH and alias "oasisgui" to launch easily the GUI, using the commands:

- setenv C3SMHOME $HOME/C3SM
- setenv PYTHONPATH $C3SMHOME/XDRpy
- alias oasisgui 'wish $C3SMHOME/c3sm/c3sm.tcl -config $C3SMHOME/myconfig.xml &'

Update your environment using :

- source .cshrc

Untar C3SM.tar using the command :

- tar -xvf C3SM.tar

Go into the directory $HOME/C3SM. Copy the default configuration xml file in your configuration file using:

- cp default_config.xml myconfig.xml

Launch the GUI using the alias you created, and on the first window, click on the OASIS icon:

- oasisgui

You must fill up the different tabs from the left to the right. Each window must be validated (button "process" on the bottom on the right). An orange bullet in the tab means that no information was stored yet in the window, a red bullet in the tab means that there is an error in the window and a green bullet in the tab means that the data entered in the window is correct.

Validated data will be progressively stored in an xml file when filling the GUI. This file must be explicitly saved when leaving the GUI (you will be automatically asked for this). Once you will have saved your xml data, you will be able to reload it using the (File/Load) menu at the left top of the GUI.
You can find information on OASIS3-MCT by clicking on the blue bullets (?) in each window.

B. Creating the namcouple of the tutorial example

The "tutorial" toy model should reproduce the coupling between two simple codes, model1 and model2, with the OASIS3-MCT coupler.

The total run is 6 hours by default. Model1 has a time step of 3600 seconds and runs on the logically-rectangular ORCA2T grid (182x149). Model2 has a time step of 1800 seconds and runs on the logically-rectangular LMDz grid (96x72).

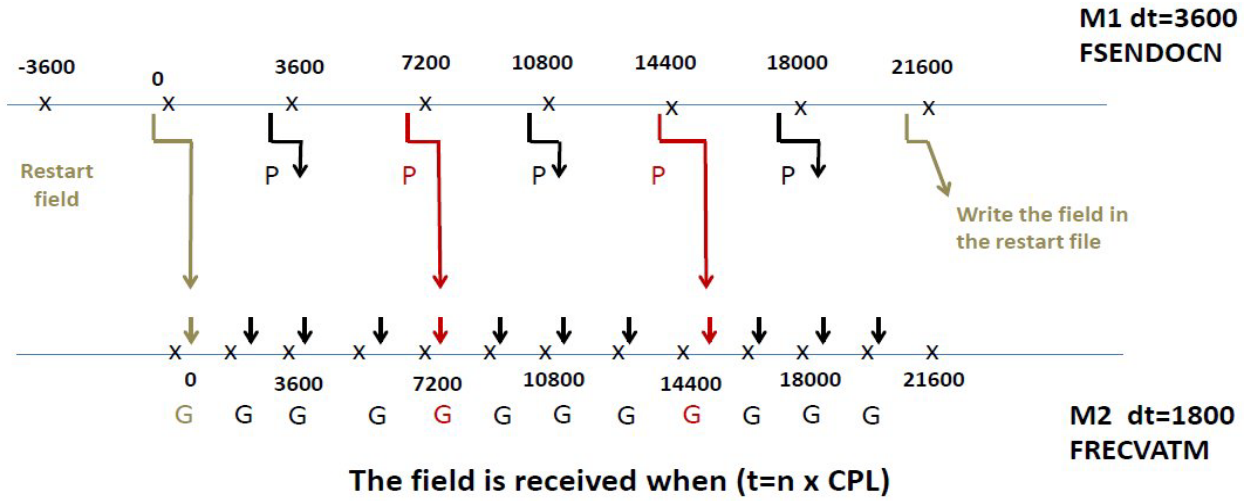
In the coupling to implement, a field with symbolic name “FSENDOCN” in model1 and “FRECVATM” in model2 is sent from model1 to model2 every 2 hours and a field with symbolic name “FSENDATM” in model2 and “FRECVOCN” in model1 is sent from model2 to model1 every 3 hours as follows:

- At the beginning of the run, model1 receives “FRECVOCN” coming from a restart file; at the end of the coupling period of 2 hours, model1 sends the coupling field “FSENDOCN” to model2 that receives it as “FRECVATM” at the beginning of the next coupling period (see figure fsendocn below).
- At the beginning of the run, model2 receives “FRECVATM” coming from a restart file; at the end of the coupling period of 3 hours, model2 sends the coupling field “FSENDATM” to model1 that receives it as “FRECVOCN” at the beginning of the next coupling period (see figure fsendatm below).
- In addition, model1 also outputs to a file the time averaged field "FSENDOCN" every 2 hours (see figure fsendocn_file).
- The interpolation from "FSENDOCN" into "FRECVATM" uses a pre-existing weight and addresses file called "my_remapping_file_bilinear.nc". The interpolation specified to transform "FSENDATM" into "FRECVOCN" is the SCRIPR/BILINEAR one (i.e. the weight and addresses file used for the interpolation are calculated at the first coupling time step with the SCRIP library).

Figure fsendocn

LAG=3600 for FSENDOCN dt=3600s, CPL=7200s, maxtime=21600s

The field is sent when (t+LAG=n x CPL)



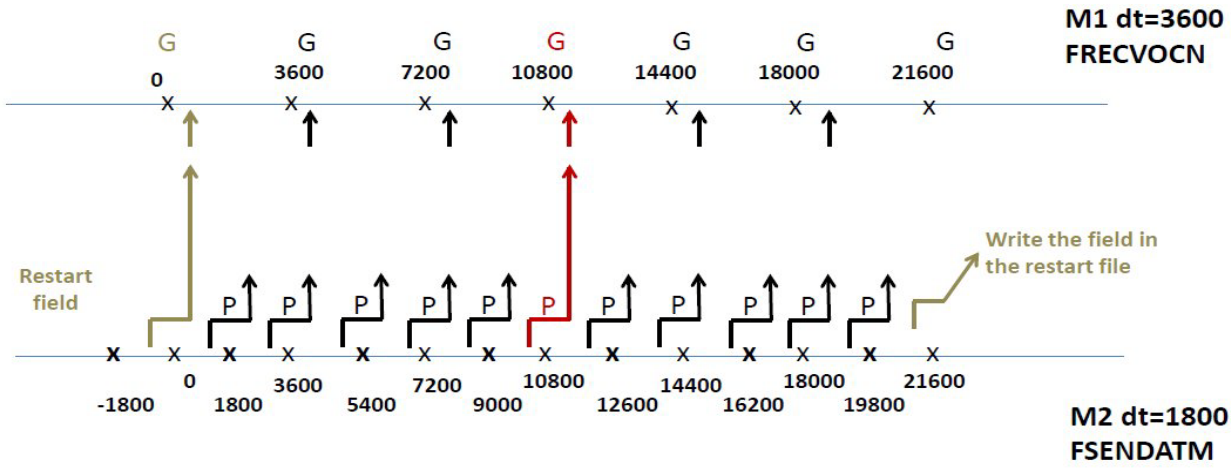
The field is received when (t=n x CPL)

P=oasis_put
G=oasis_get

Figure fsendatm

LAG=1800 for FSENDATM dt=1800s, CPL=10800s, maxtime=21600s

The field is received when (t=n x CPL)



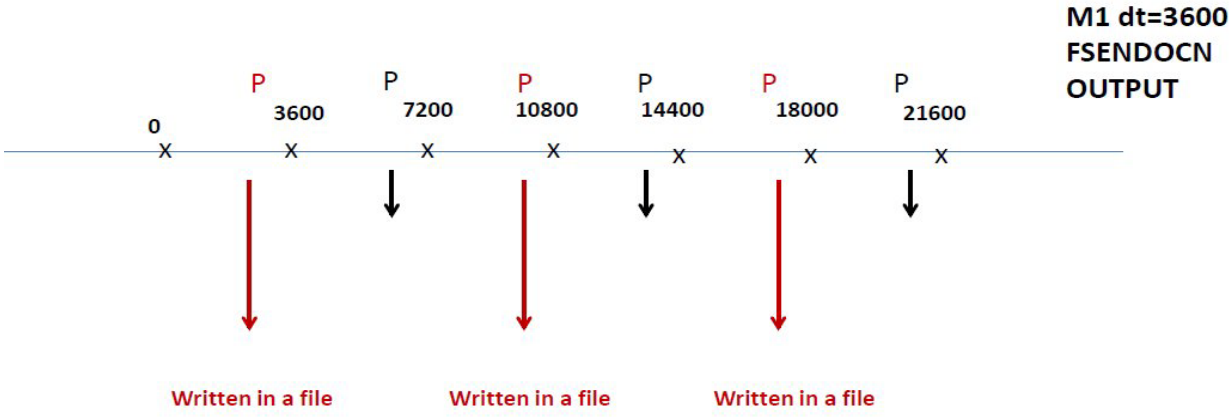
The field is sent when (t+LAG=n x CPL)

P=oasis_put
G=oasis_get

Figure fsendocn_file

FSENDOCN sent to a file CPL=7200s, maxtime=21600s

The field is sent when (t=n x CPL)



P=oasis_put

1. Read the description of the GUI in the first window and validate it by processing it using the "Process" button on the bottom left.
2. Window "Gen" (Generalities):
   - Enter the total simulation length corresponding to the total run of the tutorial of 6 hours.
   - Put the debug level at 1, which corresponds to debug information written only by the master of each model (see the blue bullet for more details).
   - Put the Time statistics option to 0 so to not perform any time statistics for the tutorial toy.
   - Do not activate the "Enable grouped fields" as there will be no fields sent together in the tutorial toy.
   - Validate the window by processing it using the "Process" button.
3. Window "Model":
   - Enter the names of the two models that will be coupled using the button "add Model in use", double clicking on the new row created and filling the part "no label" under "Model names". The names of the model in the namcouple are the ones that will be used when interfacing them with OASIS3-MCT. In your case please defined them as "model1" and "model2".
   - For each model you can also specify the maximum Fortran unit number used by the models but it is optional.
   - When processing the window, you are asked to save the data in an xml file. Save the new xml file in the data directory of the tutorial toy ***oasis3-mct/examples/tutorial/data_oasis3***. The namcouple text file will be saved when all the windows will be filled up and processed in a sub-directory named after the xml file in directory ***oasis3-mct/examples/tutorial/data_oasis3/***
4. Window "Grids":
   - Enter the names of the grids associated to the coupling fields of the different models, clicking on the button "add Grid in use", double clicking on the new row created and filling the part "no label" under "Grid names". In your case, the grids defined in the models are torc for model1 and lmdz for model2.
   - For each grid you must specify if the grid is global (periodical) or regional and the number of overlapping points. ORCA2T (torc) is periodical with 2 overlapping points and LMDz (lmdz) is also periodical but without any overlapping point.
   - You may define the dimensions of each grid or not, but if they are specified for one grid, they must be specified for all grids.
5. Window "Fld name" (Field name):
   - For coupling field , a "user name" must be given, clicking on the button "add Coupling field", double clicking on the new row created and filling the part "no label" under "Field label". Then for each coupling field, the symbolic name used in the source and target models must be specified as well as the source and target grids.
   - When processing the window to validate the data, an automatic Id is associated to each field by the GUI. This Id will be used in the last tabs to defined the different transformations associated to each field. If the coupling fields are not grouped the Id is the Field label.
6. From window "Fld status" to window "Cons" (for Conservation for global transformations):
   - Define all characteristics and transformations associated to each coupling field identified by its Id Id under the tab Fields in each window using the documentation available by clicking on the blue (?).

- The name of the restart file for the field FSENDOCN sent by model1 exists and is named fdocn.nc, the name of the restart file for the field FSENDATM sent by model2 exists and is named fdatm.nc. The field FSENDOCN will additionally be time averaged and written to a file; the name of a restart file (not used initially but used between two runs), for example f2avg.nc, must be also defined for this field. Defined the status so to be able to visualize and debug the results. For the field FSENDOCN written to a file, the status must be OUTPUT. In this case the output file and the debug file are the same.
- The interpolation from "FSENDOCN" into "FRECVATM" uses a pre-existing weight and addresses file called "my_remapping_file_bilinear.nc". The interpolation specified to transform "FSENDATM" into "FRECVOCN" is the SCRIPR/BILINEAR one (i.e. the weight and addresses file used for the interpolation are calculated at the first coupling time step with the SCRIP library).
- Process each window.

7. Window "Sum" (for Summary):
   - Process the window to obtain a summary of all the coupling fields and their transformations.
8. Window "Config" (for Configuration file):
   - Process the window to obtain the namcouple; the textfile will be saved in the sub-directory named after the xml file in directory ***oasis3-mct/examples/tutorial/data_oasis3/*** directory **.**
   - Copy the namcouple in the directory ***oasis3-mct/examples/tutorial/data_oasis3.***

II – Tutorial

A. Compiling OASIS3-MCT and running an empty "tutorial" toy model

1. To compile the OASIS3-MCT coupler:
   - Go into directory oasis3-mct/util/make_dir
   - Adapt the value of $ARCHDIR in the platform header makefile make.pgi_cerfacs to have the compiled sources in e.g. $(HOME)/oasis3-mct_comp (or another local directory)
   - Adapt the "make.inc" file to include your platform header makefile make.pgi_cerfacs.
   - Type "make realclean –f TopMakefileOasis3" and then "make –f TopMakefileOasis3"
   - The libraries "libmct.a", "libmpeu.a", "libpsmile.MPI1.a" and "libscrip.a" that need to be linked to the models are available in the directory $ARCHDIR/lib
2. To compile the tutorial models:
   - Go into directory oasis3-mct/examples/tutorial
   - Type "make clean; make" (note that the Makefile in this directory automatically includes your header makefile – see the first line in the Makefile)
   - ➢ The executables model1 and model2 are available in the current directory.

3. To run the two tutorial component models:
   - Adapt to your platform (“training_computer”) and execute the script run_tutorial:
     > ./run_tutorial
     The results of the component models are now in subdirectory $rundir defined in run_tutorial (i.e. ${HOME}/oasis3-mct/examples/tutorial/work_tutorial by default).
   - ➢ In their current form, "model1.F90" and "model2.F90" are not interfaced with the OASIS3-MCT library and do not perform any exchanges. They just run for 6 time steps of 1 hour each without doing anything specific.

B. Interfacing and running the "tutorial" toy model with OASIS3-MCT

1. Modify "model1.F90" and "model2.F90" so to implement the initialisation, definition and declaration calls of OASIS3-MCT library. The lines where to introduce the OASIS3-MCT specific calls are marked with ""!! TOCOMPLETE …". As a first step, suppose that each model will run on only one process. Most of the variables that will be used in the oasis routines calls are already defined at the beginning of the programs. A step-by-step approach is recommended, i.e. validate one-by-one each call implemented by compiling the toy coupled model and checking that the toy model runs fine until the call.
2. Modify "model1.F90" and "model2.F90" so to implement the coupling exchanges, as described above. In the time step of the models, try first by implementing the reception of the input coupling fields (“oasis_get”) at the beginning of the time step and then, the sending of the output coupling field (“oasis_put”), at the end of the timestep (see again figures fsendocn and fsendatm).
3. Run the toy coupled model with the script "run_tutorial". In this configuration, where both models call their oasis_get before their oasis_put, a deadlock may occur in the coupled toy model if the coupling fields received at the beginning of the run are not automatically read from the coupling restart files. If this happens, find which modifications are needed in the toy to remove this deadlock without changing the order of the oasis_get and oasis_put (see section 2.10.1 of User Guide for more details).

➢ The results of the tutorial coupled model are now in subdirectory ${HOME}/oasis3-mct/examples/tutorial/work_tutorial. In "work_tutorial", the file "nout.000000", written by the master process of one model, contains the information read in the configuration file "namcouple". The OASIS3-MCT debug files debug.root.01 and debug.root.02 are written by the master process of each model. The level of debug information in these files depends on the value of the first number on the line below $NLOGPRT in the namcouple (see the section 3.2 of the User Guide for more details).

➢ If you have put “EXPOUT” for the coupling field status, you can visualize the content of the netCDF debug files with ferret and the scripts script_ferret_*.jnl. Type "ferret" and then "go xxx.jnl" where xxx.jnl is the name of the ferret script you want to run. (Note that to run another ferret script, you have to restart ferret.)

➢ Explain why the number of time steps in the file FSENDATM_model2_02.nc is not the same than in the file FRECVOCN_model1_03.nc.

➢ Keep your results by renaming /work_tutorial into /work_tutorial_B4_mono

In this configuration, model1 and model2 run with one process each. This is indicated in the launching script "run_tutorial" (see "nproc_exe1" and "nproc_exe2").

C. Running the models in parallel with OASIS3-MCT

To run and couple model1 and model2 in parallel, one has to ensure that the partition definition transferred to OASIS3-MCT via the call to oasis_def_partition is properly done. In model1 and model2, the information to provide as arguments of the oasis_def_partition is calculated by the subroutine decomp_def that you can check in more detail. Then, the number of processes has to be modified in the launching procedure.

- Specify for example "nproc_exe1=3", "nproc_exe2=3" in the script run_tutorial .
- In oasis3-mct/examples/tutorial, execute "run_tutorial", which then launches the models on 3 processes each.
- Keep your results by renaming /work_tutorial into /work_tutorial_C_para

➢ Visually compare the results with the non-parallel case B4_mono.

OASIS3-MCT supports different parallel decompositions for the models (see the section 2.4 of the User Guide). Tutorial routine "decomp_def.F90" can define the local partition for the BOX or the APPLE decompositions. By default, the APPLE decomposition is used. To test the BOX decomposition, recompile the tutorial models with, in Makefile: "CPPKEYDECOMP_M1=DECOMP_BOX" or "CPPKEYDECOMP_M2=DECOMP_BOX".