OASIS3-MCT tutorial

A. <u>Compiling OASIS3-MCT and running an empty "tutorial" toy model</u>

1. Download the OASIS3-MCT sources using :

    ➢ mkdir oasis3-mct ; cd oasis3-mct

    ➢ svn checkout http://oasis3mct.cerfacs.fr/svn/trunk/oasis3-mct .

2. To compile the OASIS3-MCT coupler:
    * Go into directory oasis3-mct/util/make_dir
    * Load the module pgi with the command: "module load pgi". You have to type this command every time you open a new terminal. You need to do load this module to compile with pgi Fortran compiler.
    * 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

3. 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.

4. To run the two tutorial component models:
    * Adapt to your platform 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. <u>Interfacing and running the "tutorial" toy model with OASIS3-MCT</u>

After your modifications, 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 a logically-rectangular (182x149) grid. Model2 has a time step of 1800 seconds and runs on a logically-rectangular (96x72) grid.

The coupling exchanges to implement are as follows:
- Every 2 hours, model1 sends the coupling field "FSENDOCN" to model2 that receives it as "FRECVATM".
- Every 3 hours, model2 sends the coupling field "FSENDATM" to model1 that receives it as "FRECVOCN".
- In addition, model1 also outputs to a file the field "FSENDOCN" every 2 hours.

1. Have a look at the OASIS3-MCT configuration file "namcouple", located in the directory oasis3-mct/examples/tutorial/data_oasis3, written to perform the exchanges described above. The interpolation specified to transform "FSENDATM" into "FRECVOCN" is the SCRIPR/BILINEAR one (i.e. the weight and addresses file used for the interpolation will be calculated at the first coupling time step with the SCRIP library). Interpolation from "FSENDOCN" into "FRECVATM" will use a pre-existing weight and addresses file (my_remapping_file_bilinear.nc).

2. 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 calls.

3. 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.

4. Try to 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 will occur in the coupled toy model. 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 one of the master process of the models, contains the information read in the configuration file "namcouple" by all the processes. The OASIS3-MCT debug files debug.root.01 and debug.root.02 are written by the processor of each model. The level of debug information in these files depends on the value of NLOGPRT defined in the namcouple. Here NLOGPRT=10 and the files contain normal diagnostics production, with initial debug info and calling trees of the routines (see the section 3.2 of the User Guide for more details).

➢ You can visualize the results 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".