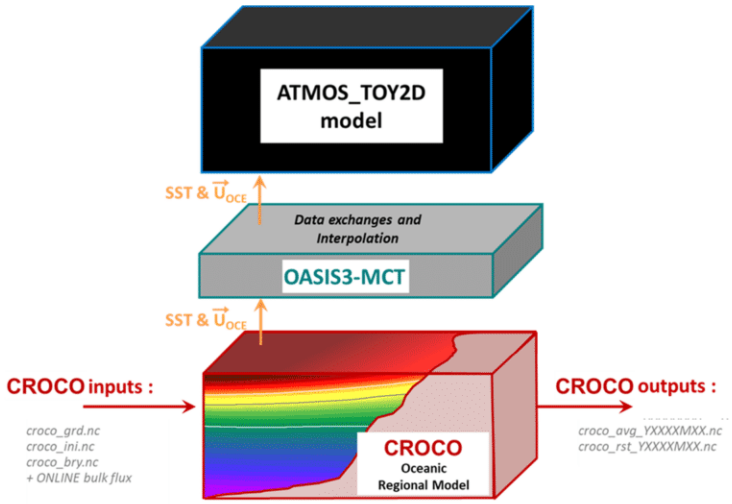
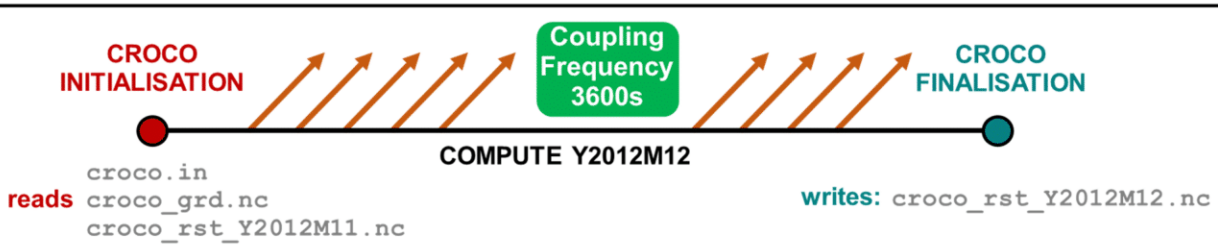
# OA COUPLING TUTORIAL 02:
# Coupling CROCO with an atmospheric TOY model – RUN2

In this tutorial, we run a **semi-coupled** interannual CROCO simulation for the month of December 2012: `croco` will be forced by ERA5 bulk forcings, meanwhile it will be coupled to an atmospheric TOY model (`atoy2d`) that will get CROCO SST and surface currents.




To run this simulation, we will connect to the Lengau supercomputer (see STEP1). We will need the OASIS3-MCT coupler (see STEP2), and the `croco` and `atoy2d` models compiled in a (semi-) coupled context (see STEP3 and STEP4). Our CROCO **MCC** grid and forcings for the month of December 2012 have already been created in #**CPL_TUT01**). CROCO input files are in `../INPUT_FILES/CROCO_FILES`. We will create the files required for the coupler (see STEP5 and STEP6) and then run the simulation (see STEP7).

## STEP 1: Logging onto the Lengau HPC cluster

→ From a terminal/konsole, execute the following instruction:

```
ssh -X login@lengau.chpc.ac.za
```

✋ Replace `login` with your corresponding account number.

→ Reserve one interactive processor:

```
[login@login2 ~]$ qsubi1
[login@cnode0220 ~]$
```

→ Go directly into your `lustre` directory:

```
[login@cnode0220 ~]$ cd lustre
```

## STEP 2: Compiling the OASIS3-MCT coupler library on Lengau

Here, we create the oasis3-mct coupler libraries.

→ Go into the **OASIS** directory:

```
[login@cnode0220 lustre]$ cd OASIS
[login@cnode0220 OASIS]$ ls
 oasis3-mct compiled_oa3-mct_lengau readme
```

➥ The source code of the latest version of the coupler has already been downloaded from https://oasis.cerfacs.fr/en/downloads/ and is stored in the **oasis3-mct** directory.

➥ **compiled_oa3-mct_lengau** is an empty directory, where the compilation outputs will be stored (`libmct.a  libmpeu.a  libpsmile.MPI1.a  libscrip.a`).

→ To compile the oasis3-mct coupler libraries, follow these steps:

- Go into the directory **oasis3-mct/util/make_dir**
- If needed: edit **make.inc**
  - Specify the compiling platform (ex: **make.intel_lengau**)
- If needed: edit **make.intel_lengau**
  - Make the appropriate changes (input/output paths and compilation options)

```
COUPLE  = /home/${USER}/lustre/OASIS/oasis3-mct
ARCHDIR = $(COUPLE)/../compiled_oa3-mct_lengau
```

- Execute the command: **make -f TopMakefileOasis3**

→ Verify that the 4 libraries have been created in **compiled_oa3-mct_lengau/lib**

## STEP 3: Preparation of the CROCO component in a semi-coupled context

Our CROCO grid and forcings for the month of December 2012 have already been created in #**CPL_TUT01**). **MCC** CROCO input files are in **WORK_MCC/INPUT_FILES/CROCO_FILES**. Moreover, CROCO restart (**croco_rst_Y2012M11.nc**) should have been copied in **INPUT_FILES/CROCO_FILES** (see **STEP8** in #**CPL_TUT01**)

→ Compile CROCO in a **semi-coupled** context.

- Go in the directory **WORK_MCC/Run**
- If needed: edit **param.h**:
  - Check the **MCC** grid size and **parallelisation** params (NP_XI=1, NP_ETA=4)
- Edit **cppdefs.h**:
  - Activate the coupling **#define OA_COUPLING** (line 82)
  - Activate the **semi-coupled** context: **#define OA_COUPLING_CPLMASKS** (line 84)
  - See **#CPL_TUT01** for additional CPP KEY to define/undefine for MPI and CROCO forcing choices (bry+online bulk).
    - ➥ Save your **cppdefs.h** to **cppdefs_run2.h**:

```
[login@cnode0220 Run]$ cp cppdefs.h cppdefs_run2.h
```

    - ➥ Compare **cppdefs_run2.h** to **cppdefs_run1.h** using the command **meld**:

```
[login@cnode0220 SCRIPTS]$ meld cppdefs_run2.h cppdefs_run1.h
```

- If needed: edit **jobcomp_lengau** and adjust the OASIS3-MCT library path
- Compile the code: **./jobcomp_lengau**

→ Rename CROCO executable **croco** → **croco_run2** (sc for semi-coupled)

➥ You can also add the Nb of proc at the end of the executable name

```
[login@cnode0220 Run]$ mv croco croco_run2
```

## STEP 4: Preparation of the ATOY2D component

The ATMOS_TOY2D model (in **TOY_MODELS/ATMOS_TOY2D** directory) is a TOY model that mimic WRF component. It will be coupled to CROCO and will receive its SST and surface currents.

→ Go in the directory **TOY_MODELS/ATMOS_TOY2D**

→ Here is the core of the **atoy2d** component **TOY_MODELS/ATMOS_TOY2D/atmos_toy2d.F**:

*Atoy2d params*

```
parameter(nx0=88,ny0=108)  !Size of the spatial grid
open(unit=271,file='ATOY.out',form="formatted")
call oasis_cpl_init
call oasis_cpl_define(nx0,ny0,nmaxfld,krcv0,krcv,prcv_nid,oasis_runtime)
idt=60   !Time step (second)
do time=0,oasis_runtime-idt,idt
  call oasis_cpl_get(nx0,ny0,...,krcv0,krcv,time,prcv_nid)
enddo
call oasis_terminate(ierr)
```

**ATOY2D EXCHANGED FIELDS:** (0=Parent, 1=first zoom, etc…)

*RECEIVED*

- **ATOYSST0** → 2D Field, just like an SST field
- **ATOYUOC0** → 2D Field, just like a Zonal current field
- **ATOYVOC0** → 2D Field, just like a Meridional current field

→ Edit the **atmos_toy2d.F** and change the **spatial** grid size with your MCC grid (LLm, MMm).

➥ You can go in **WORK_MCC/SCRIPTS** and execute **./STEP1_inspect_CROCO_Grid.scr**

→ If needed: edit **jobcomp_lengau** and adjust NetCDF, MPI and OASIS-MCT library paths

→ Compile the code: **./jobcomp_lengau** and verify that the executable **atoy2d** has been created.

## STEP 5: Create the OASIS Files

→ It is done with some **bash scripts** in the **WORK_MCC/SCRIPTS** directory:

→ Edit **STEP3_make_oasis_files.scr**

- Adjust CROCO grid path (**../INPUT_FILES/CROCO_FILES**)
- Adjust WRF grid path (**../INPUT_FILES/WRF_FILES**) [file does not exist yet]
- Adjust OUTPUT path (**../INPUT_FILES/OASIS_FILES**)
- Adjust the number of domains: **max_domains_CROCO=1** and **max_domains_WRF=1**

→ Execute **./STEP3_make_oasis_files.scr**: this will create the OASIS3-MCT auxiliary NetCDF files required for **interpolation** of fields between your different model grids:

**INPUT_FILES/OASIS_FILES/grids.nc**
**INPUT_FILES/OASIS_FILES/masks.nc**
**INPUT_FILES/OASIS_FILES/areas.nc**
**INPUT_FILES/OASIS_FILES/coupling_masks_zero.nc**

**CROCO GRIDS/MASKS:** (0=Parent, 1=first zoom, etc…)

- **crn0** → CROCO ***rho*** Normal (Masked)
- **crp0** → CROCO ***rho*** Processed (No Mask)
- **cun0** → CROCO ***U*** Normal (Masked)
- **cup0** → CROCO ***U*** Processed (No Mask)
- **cvn0** → CROCO ***V*** Normal (Masked)
- **cvp0** → CROCO ***V*** Processed (No Mask)

## STEP 6: Create the OASIS namcouple file

→ This is a simple ascii file
→ An example is given:
**INPUT_FILES/OASIS_FILES/namcouple_CROCO_ATOY2D_example**

```
#########################################################################
#
#       Input file for OASIS3-MCT
#
#########################################################################
#
#       Input delimiters have to occupy position 1 to 9 !
#       No blank lines allowed !
#       Length of input lines <= 80 !
#
#########################################################################
#
# NFIELDS : total number of fields being exchanged.
#
 $NFIELDS
 1
#
#########################################################################
#
# RUNTIME: total simulated time for the actual run in seconds (<I8)
#
 $RUNTIME
 86400
#
#########################################################################
#
# NLOGPRT: printing level in output file cplout:
#               0 = no printing
#               1 = main routines and field names when treated
#               2 = complete output
#
 $NLOGPRT
  2
 $END
#########################################################################
 $STRINGS
#########################################################################
#
#                       OCEAN  --->>>  ATMOS
#                       --------------------
#########################################################################
#
# Field 1.1 : Sea surface temperature (o->a 1) from CROCO Parent to ATOY2D
#
CROCO_SST ATOYSST0 1 3600 1 ssto0.nc EXPOUT
87 107 87 107 crn0 crp0 LAG=0
R 0 R 0
SCRIPR
BILINEAR LR SCALAR LATLON 1
#########################################################################
$END
```

→ For RUN2, the **namcouple** file need to be named **namcouple_CROCO_ATOY2D** [as specified in **RUN2_croco_inter_atoy.pbs**].

```
[login@cnode0220 OASIS_FILES]$ cp namcouple_CROCO* namcouple_CROCO_ATOY2D
```

➥ Edit this new file to change dimensions same as in **param.h** and **atoy2d** and save.

➥ Later you will make more changes in this **namcouple_CROCO_ATOY2D** file.

**CROCO EXCHANGED FIELDS:** (0=Parent, 1=first zoom, etc…)

*SENT*

- **CROCO_SST** → CROCO SST on grid **crn0**
- **CROCO_UOCE** → CROCO Surface Zonal Current on grid **cun0**
- **CROCO_VOCE** → CROCO Surface Meridional Current on grid **cvn0**

*RECEIVED*

- **CROCO_SRFL** → CROCO Solar Heat Flux on grid **crp0**
- **CROCO_EVPR** → CROCO Evaporation minus Precipitations on grid **crp0**
- **CROCO_STFL** → CROCO Non-Solar Heat Flux on grid **crp0**
- **CROCO_UTAU** → CROCO Stress along X axis on grid **crp0** or **cup0**
- **CROCO_VTAU** → CROCO Stress along Y axis on grid **crp0** or **cvp0**

## STEP 7: Launch CROCO/ATOY simulation

The simulation will be launched from the **WORK_MCC/SCRIPTS** directory using the **RUN2_croco_inter_atoy.pbs** script.

→ Compare the script for **RUN2** with the **RUN1_croco_inter.pbs** script with the Linux command **meld**:

```
[login@cnode0220 SCRIPTS]$ meld RUN2_croco_inter_atoy.pbs RUN1_croco_inter.pbs
```

→ Modify the script **RUN2_croco_inter_atoy.pbs**:

- PBS header:
  - Check the PBS scheduler parameters (`mpiprocs` = NBcroco+NBatoy) and email.
- RUN2 output directory:
  - Check the path of your output directory: **SCRATCHDIR=../OUTPUT_FILES/RUN2**
- CROCO paths and run parameters:
  - Adjust the name of you CROCO executable: **CODFILE=croco_run2**
  - Check the path of the **Run** directory where **croco_inter.in** and **croco_run2** are.
  - Check the path of your CROCO input files: **../INPUT_FILES/CROCO_FILES**.
  - Adjust **NBPROCS_CROCO** (consistent with **param.h**)
  - Check which input files are used (only BOUNDARY_FILES=1)
  - Put **OGCM=mercator**
  - Adjust **DT**, **NAVG**, start (**2012M12**), end (**2012M12**), with **RSTFLAG=1**.
- OASIS input files:
  - Check the path of your OASIS auxiliary files: **INPUT_FILES/OASIS_FILES**.
- ATOY2D paths and run parameters:
  - Adjust the path where the **atoy2d** executable is.
  - Adjust **NBPROCS_ATOY2D=1**

→ Launch the model:

```
[login@cnode0220 SCRIPTS]$ qsub RUN2_croco_inter_atoy.pbs
```

**Debug**:
- The output and error file from the **PBS** job are **SCRIPTS/job.pbs.oxxxxx** **SCRIPTS/job.pbs.exxxxx**
- The **croco** log file is **OUTPUT_FILES/RUN2/croco_Y2012M12.out**
- The **atoy** log file is **OUTPUT_FILES/RUN2/ATOY.out**
- The **oasis** log files are:
  1) Oasis: **OUTPUT_FILES/RUN2/nout.000000**
  2) CROCO exchanges: **OUTPUT_FILES/RUN2/debug.01.00000[0-3]**
  3) ATOY2D exchanges **OUTPUT_FILES/RUN2/debug.02.000000**

**Outputs**:
- CROCO outputs are: **OUTPUT_FILES/RUN2/croco_avg_Y2012M12.nc** **OUTPUT_FILES/RUN2/croco_his_Y2012M12.nc**
- OASIS remapping weights are: **OUTPUT_FILES/RUN2/ rmp_crn0_to_crp0_BILINEAR.nc**
- If you used the EXPOUT in oasis exchanges (in the **namcouple** file), outputs are: **OUTPUT_FILES/RUN2/ATOYSST0_atoy2d_01.nc** **OUTPUT_FILES/RUN2/CROCO_SST_crocox_01.nc**

## STEP 8: Visualising model outputs

→ CROCO/OASIS output files from **RUN2** are in **OUTPUT_FILES/RUN2**

➥ Verify that the final restart file (**croco_rst_Y2012M12.nc**) has been written.

➥ Copy the restart in **RUN5** input directory (**OUTPUT_FILES/RUN5**):

```
[login@cnode0220 SCRIPTS]$ cd ../OUPUT_FILES/RUN2
[login@cnode0220 RUN2]$ cp croco_rst_Y2012M12.nc ../RUN5
[login@cnode0220 RUN2]$
```


→ Visualization can be done:

- With **ncview** in **OUTPUT_FILES/RUN2**
- With **MATLAB croco_gui** (for CROCO outputs) in **WORK_MCC/Run**

➥ Launch **matlab -nodesktop** (or the alias **mat**) to visualize your outputs:

```
>> start
>> croco_gui
```

## STEP 9: Exiting

→ Give back the interactive node and logout from Lengau:

```
[login@cnode0220 CROCO]$ exit
logout
qsub: job 4416950.sched01 completed
[login@login2 ~]$ exit
```