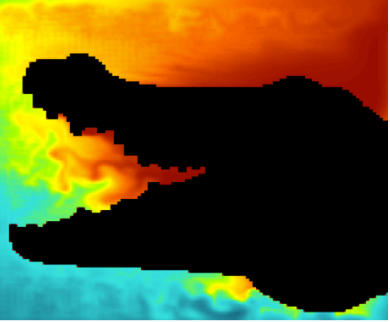
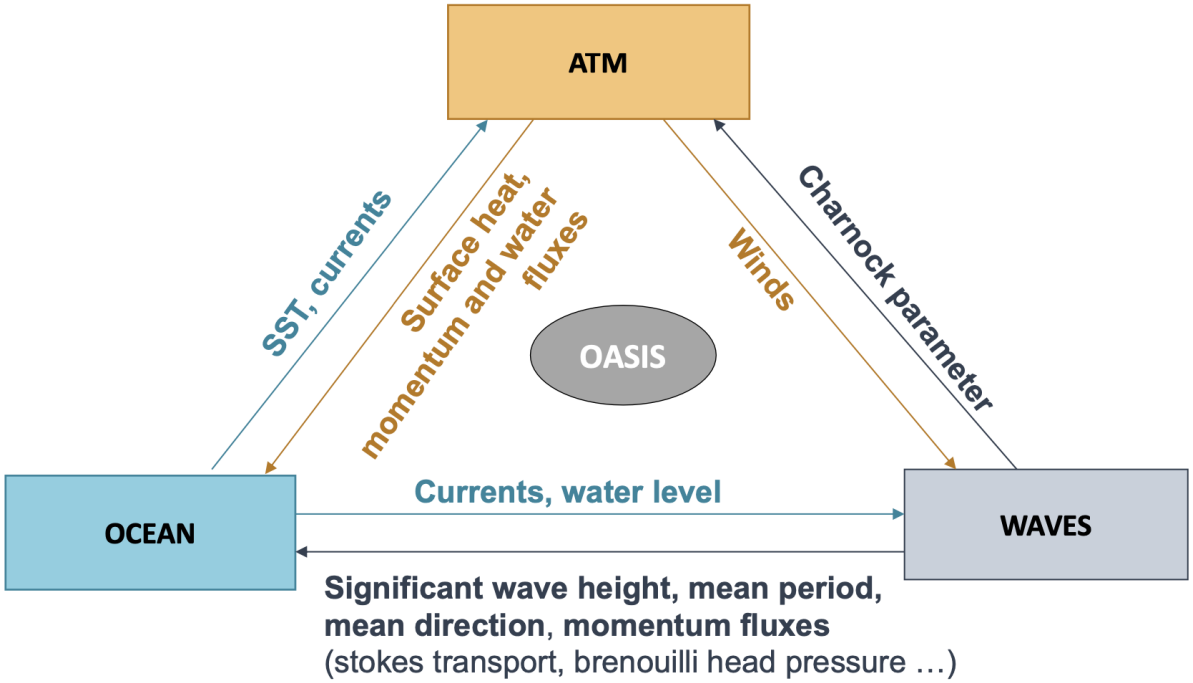


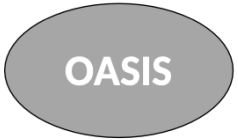
Coupling with OASIS

Introduction: coupling philosophy

Coupling with OASIS



Coupling with OASIS



OASIS-MCT (Ocean-Atmosphere-Sea-Ice-Soil, Model Coupling Toolkit) is a coupler developed at CERFACS, Toulouse, France.

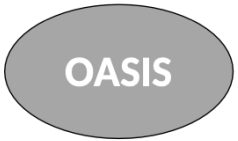
It is a **set of libraries** (not an executable file) providing functions which are called in the models themselves:

- Exchange of variables and time interpolations (PSMILE library)
- Parallel exchanges (MCT library)
- Grid interpolations (SCRIPR library)

It has the advantage of being:

- **non-intrusive, easy implementation**: only a few calls in the model time stepping, and a few additional routines
- A **common** interface for a variety of models (e.g. CROCO, NEMO, SURFEX, WAVEWATCHIII...)

Coupling with OASIS



On the developer side:

- a few additional routines in each model to specify exchanges
- a few calls in the main model routines (initialization, time stepping, finalization)

On the user side:

- **compilation** with options for coupling in each code and link to the OASIS library
- coupling settings (variables to exchange, coupling frequency, grids...) are controlled thanks to an external file: **namcouple**
- **additional restart files** are created for the coupler
- models are **launched together** at the same time
- “a few” additional log files to check

Implementation in models

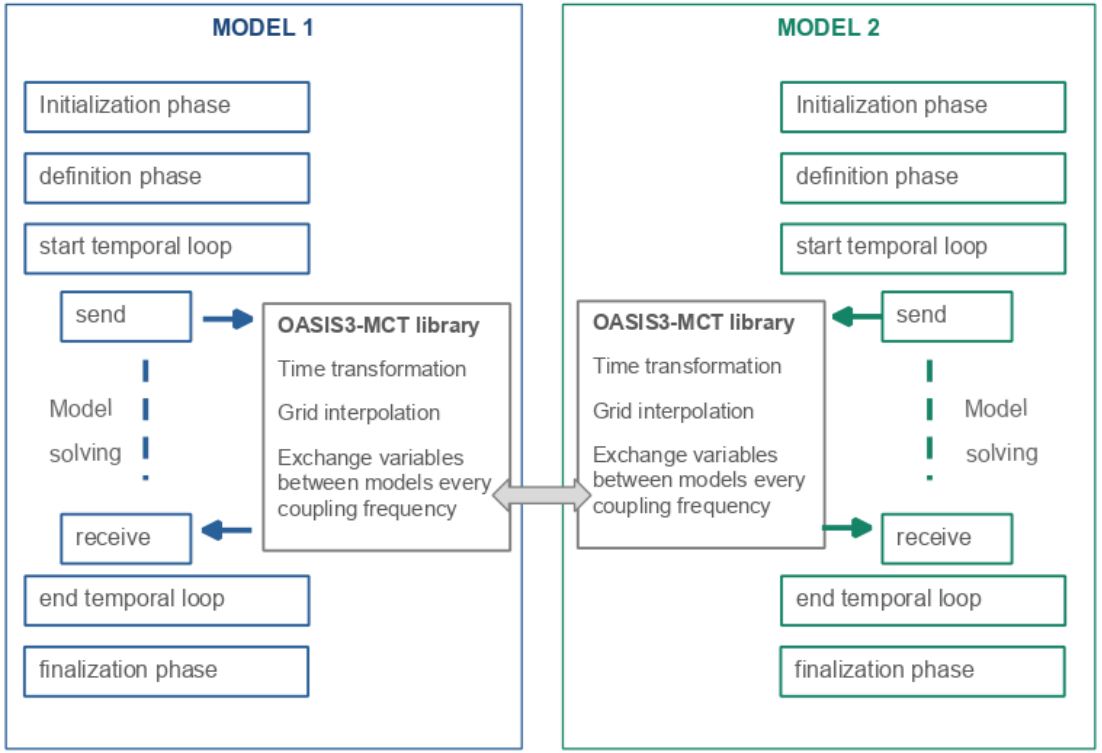
OASIS-MCT implementation calls:

Initialization
oasis_init_comp(...)
oasis_get_localcomm(..)

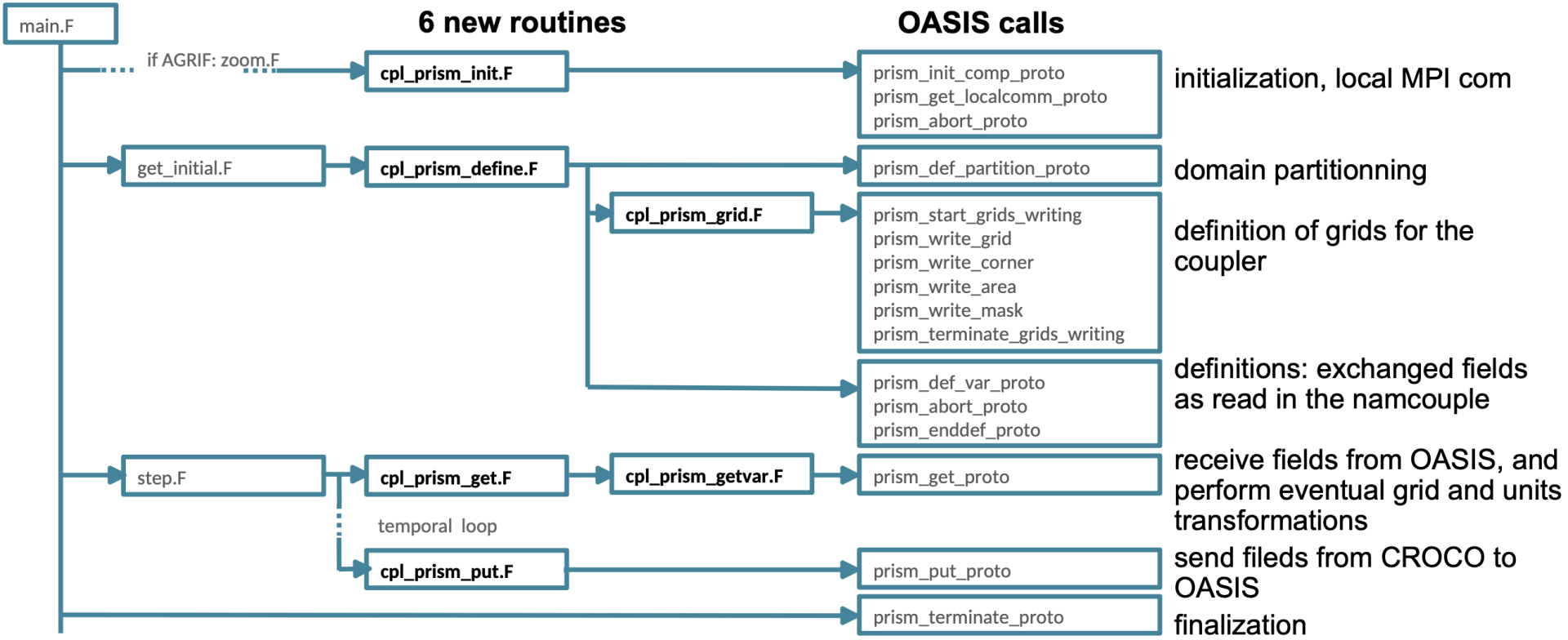
Definitions
oasis_write_grid(...) oasis_def_partitions(...)
oasis_def_var(...)

Exchange fields(within time stepping)
oasis_put(...)
oasis_get(...)

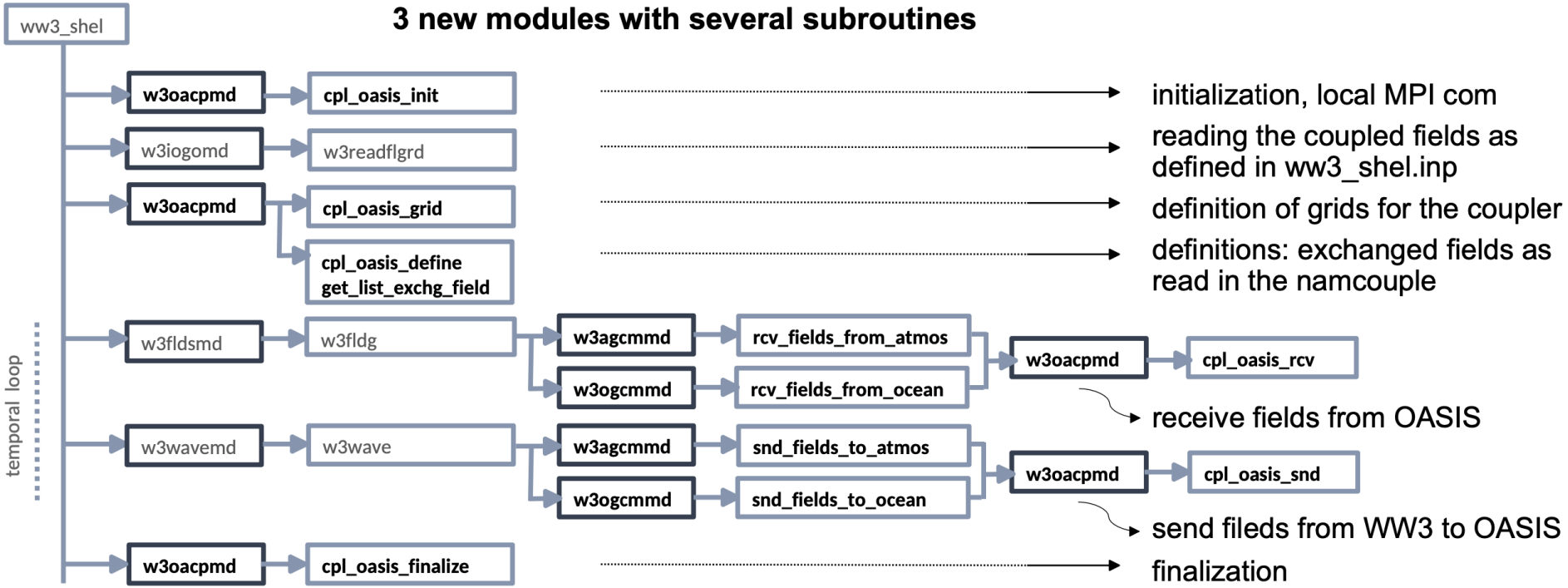
Finalization
oasis_terminate(...)



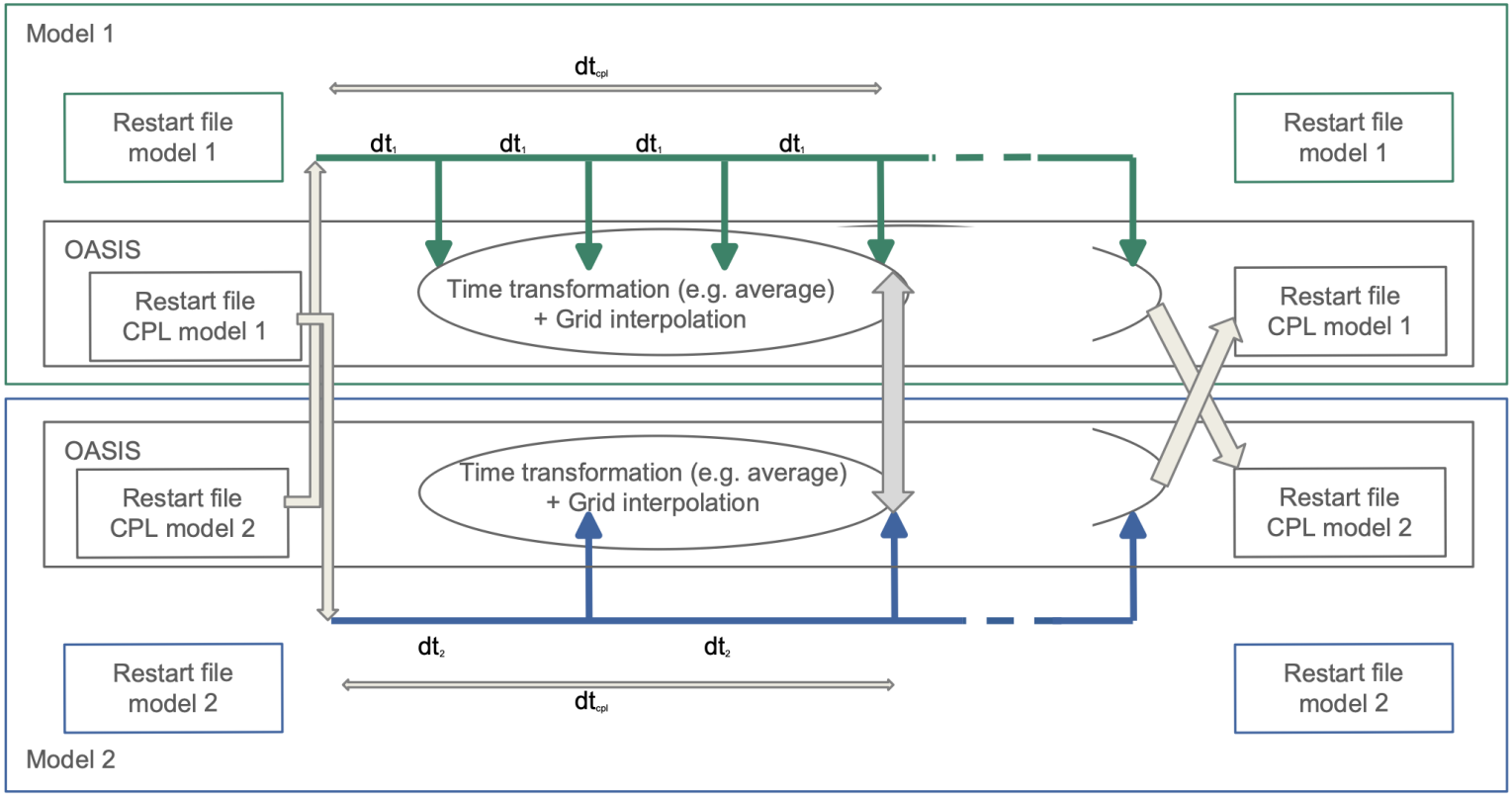
Implementation in CROCO



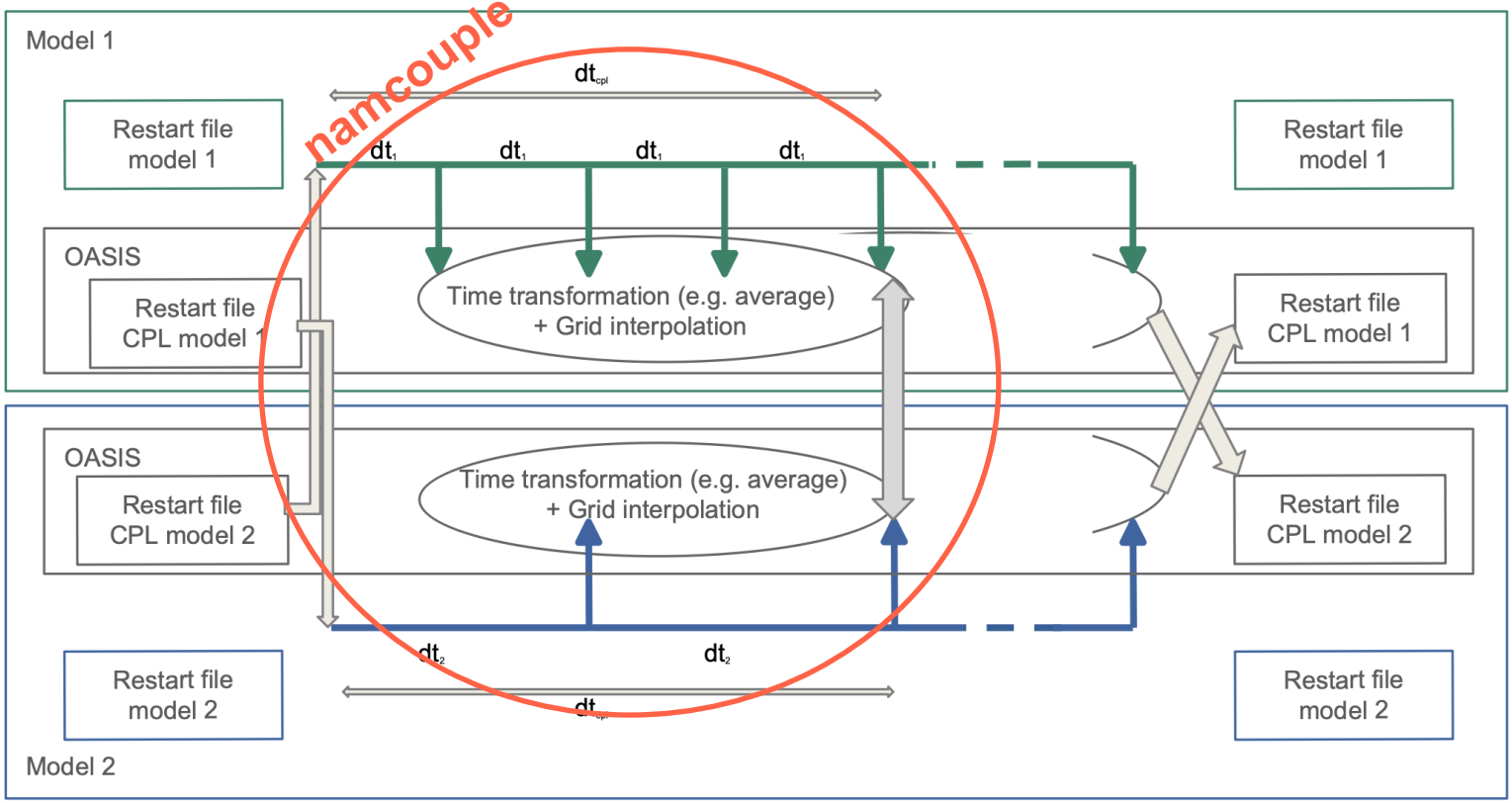
Implementation in WW3



Coupling sequence



Coupling sequence



The namcouple

The **namcouple** is the text file through which you will specify which fields will be coupled, and how...

A first section gathers the general settings:

```
# NFIELDS: total number of fields being exchanged
 $NFIELDS
 6
#############################################################################
# NBMODEL: number of models and their names (6 characters)
 $NBMODEL
 2 wwatch crocox
#############################################################################
# RUNTIME: total simulated time for the actual run in seconds (<I8)
 $RUNTIME
 2678400
#############################################################################
# NLOGPRT: debug and time statistics informations printed in log file
 $NLOGPRT
 1 1
#############################################################################
```

The namcouple

The **namcouple** is the text file through which you will specify which fields will be coupled, and how...

A second section provides the information on exchanged fields.
A typical sub-section for one exchanged field looks like:

```
#~~~~~~~~~~~~
# UOCE : sea surface zonal currents (m.s-1)
#~~~~~~~~~~~~
CROCO_EOCE WW3_OSSU 1 3600 2 oce.nc EXPORTED
95 96 95 96 ocnu ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
```

`CROCO_EOCE WW3_OSSU` : OASIS variable name in sending / receiving model
`1` : unused digit
`3600` : coupling frequency
`2` : number of transformation (temporal and grid interp.)
`oce.nc` : OASIS restart file name
`EXPORTED` : keyword : EXPORTED = field written only at the end of the run
EXPOUT = field written at every coupling time step

`95 96 95 96` : number of points of the sending / receiving grids
`ocnu ww3t` : OASIS names of the sending / receiving grids
`LAG=3600` : LAG=dt of sending model

`R 0 R 0` : type of grid (Regional or Periodical) and number of overlapping points for sending / receiving model

`LOCTRANS` : keyword for time transformation
`AVERAGE` : type of time transformation

`SCRIPR` : keyword for grid interpolation
`DISTWGT LR SCALAR LATLON 1 4` : type and parameters of grid interpolation (cf. OASIS manual)

Variables to be exchanged

OASIS NAME	Description	Unit
CROCO_SST	sea surface temperature	K
CROCO_UOCE / VOCE	X-dir / y-dir ocean surface current	m/s
CROCO_EOCE / NOCE	Eastward / northward ocean surface current	m/s
CROCO_SSH	Sea surface elevation	m

CROCO sent fields

OASIS NAME	Description	Unit
CROCO_HS	Significant wave height	m
CROCO_T0M1	Mean wave period	m/s
CROCO_DIR	Mean wave direction	rad
CROCO_FOC	Wave-to-ocean TKE flux	W/m2
CROCO_LM	Mean wavelength	m
CROCO_BHD	Bernoulli head pressure	N/m
CROCO_UBRX / UBRY	Wave orbital bottom velocity	m/s
CROCO_USSX / USSY	Stokes drift surface velocity	m/s
CROCO_UTWO / VTWO	y-dir /y-dir component of wave-to-ocean stress	m2/s2
CROCO_ETWO / NTWO	eastward /northward component of wave-to-ocean stress	m2/s2
CROCO_UTAW / VTAW	y-dir /y-dir component of atm-to-wave stress	m2/s2
CROCO_ETAW / NTAW	eastward /northward component of atm-to-wave stress	m2/s2

CROCO received fields

Variables to be exchanged

OASIS NAME	Description	Unit
WW3__OHS	Significant wave height	m
WW3_T0M1	Mean wave period	m/s
WW3__DIR	Mean wave direction	rad
WW3__FOC	Wave-to-ocean TKE flux	W/m2
WW3__LM	Mean wavelength	m
WW3__BHD	Bernoulli head pressure	N/m
WW3_UBRX / UBRY	Wave orbital bottom velocity	m/s
WW3_USSX / USSY	Stokes drift surface velocity	m/s
WW3_TWOX / TWOY	eastward /northward component of wave-to-ocean stress	m2/s2
WW3_TAWX / TAWY	eastward /northward component of atm-to-wave stress	m2/s2

WW3 sent fields

OASIS NAME	Description	Unit
WW3_OSSU / OSSV	Eastward / northward ocean surface current	m/s
WW3__SSH	Sea surface elevation	m

WW3 received fields

Time and grid transformations

The OASIS3-MCT coupler can process:

=> See OASIS manual for more detailed information

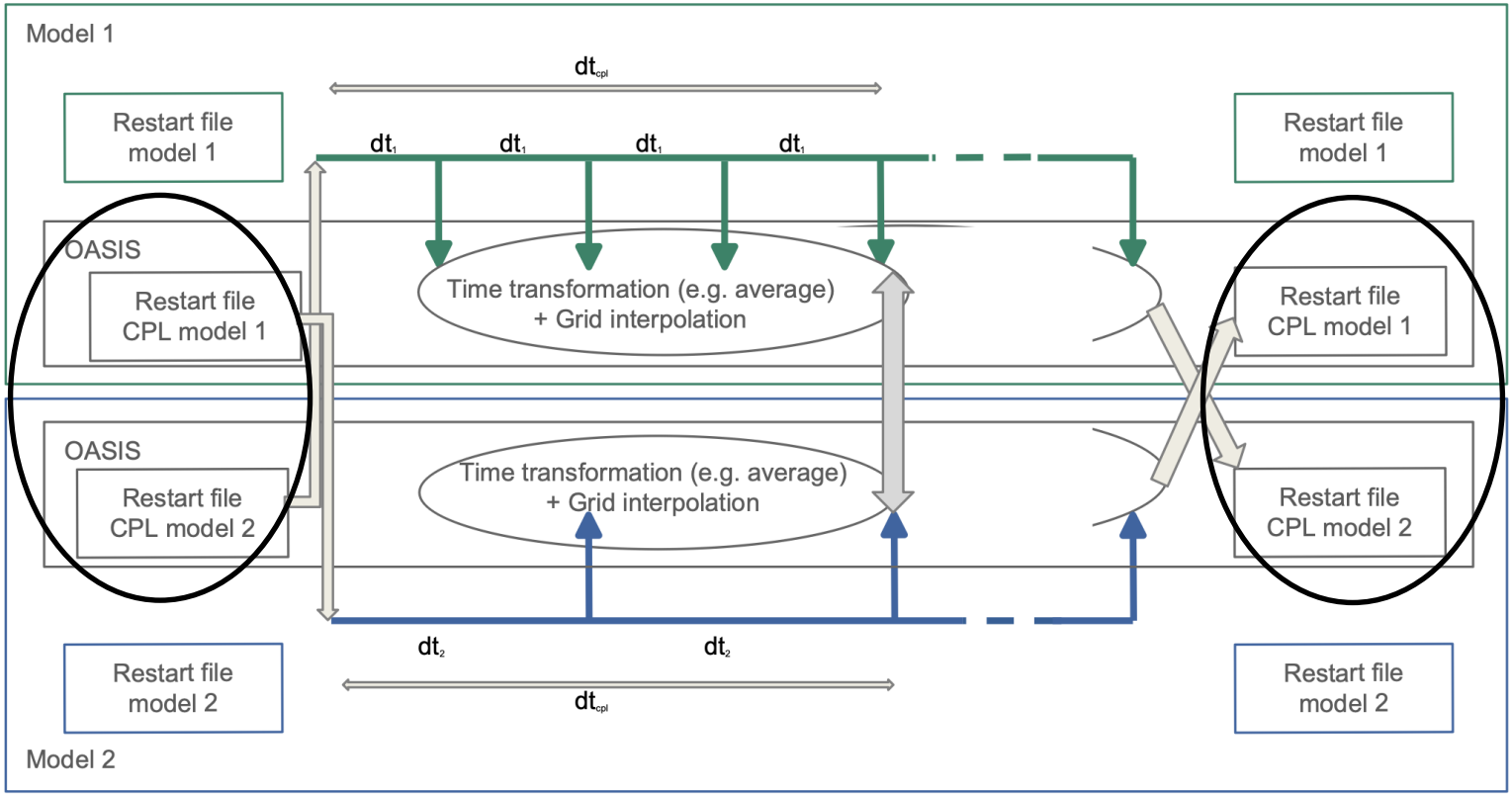
- time transformations (LOCTRANS):

INSTANT no time transformation, the instantaneous field is transferred
ACCUMUL the accumulated field over the coupling period is exchanged
AVERAGE the averaged field over the coupling period is transferred
T_MIN the minimum value of the field for each source grid point over the coupling period is transferred
T_MAX the maximum value of the field for each source grid point over the coupling period is transferred

- 2D spatial interpolations (SCRIPR):

BILINEAR interpolation based on a local bilinear approximation
BICUBIC interpolation based on a local bicubic approximation
CONSERV 1st or 2nd order conservative remapping
DISTWGT distance weighted nearest-neighbour interpolation (N neighbours)
GAUSWGT N nearest-neighbour interpolation weighted by their distance and a Gaussian function

OASIS restart files



Summary of OASIS-MCT additional files

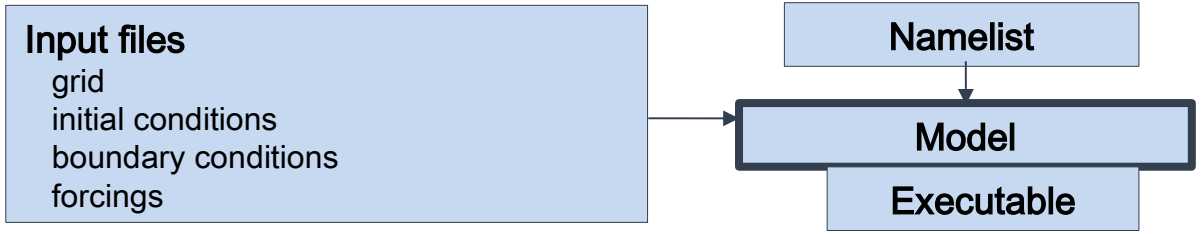
- **Input files:**
  - **Restart files**: they have to be build for initialization, and they will be automatically written at the end of the simulation for the next one: oce.nc, atm.nc, wav.nc
  - **Namelist** file: namcouple

- **Grid generated files**:
  these files are requested for interpolations, they are automatically built at the beginning of the simulation by all models: grids.nc, masks.nc, areas.nc

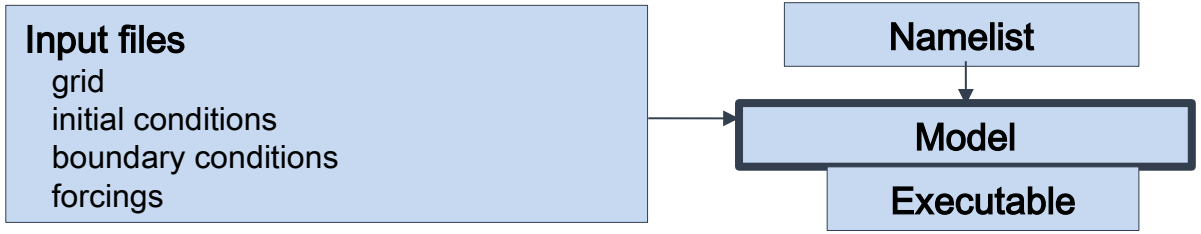
- **Grid interpolation generated files**:
  these files are built automatically by OASIS according to the previously cited grid files, and to SCRIPR settings in the namcouple: rmp_ww3t_to_ocnt_DISTWGT.nc, rmp_ocnt_to_ww3t_DISTWGT.nc ...

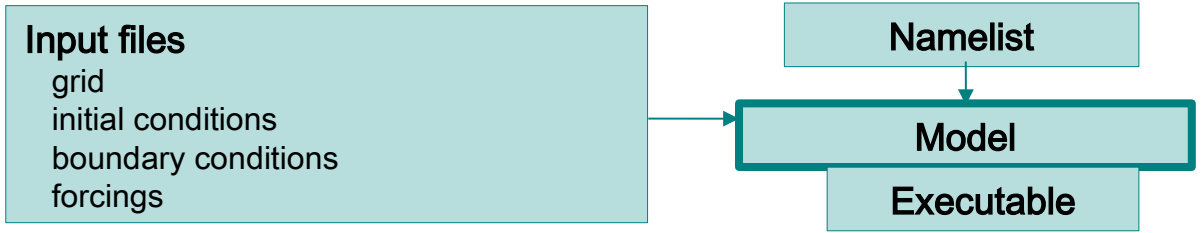
- **Log files**:
  several log files are produced by OASIS, they should be checked at the end of the simulation or if something goes wrong during the simulation:
  nout.000000 : OASIS log file
  crocox.timers_0000, wwatch.timers_0000 : OASIS log file for time statistics
  debug.root.01, debug.root.02 : log files for the master processor for each model
  debug.notroot.01 : log files for other processors for each model

Steps for setting up a coupled simulation

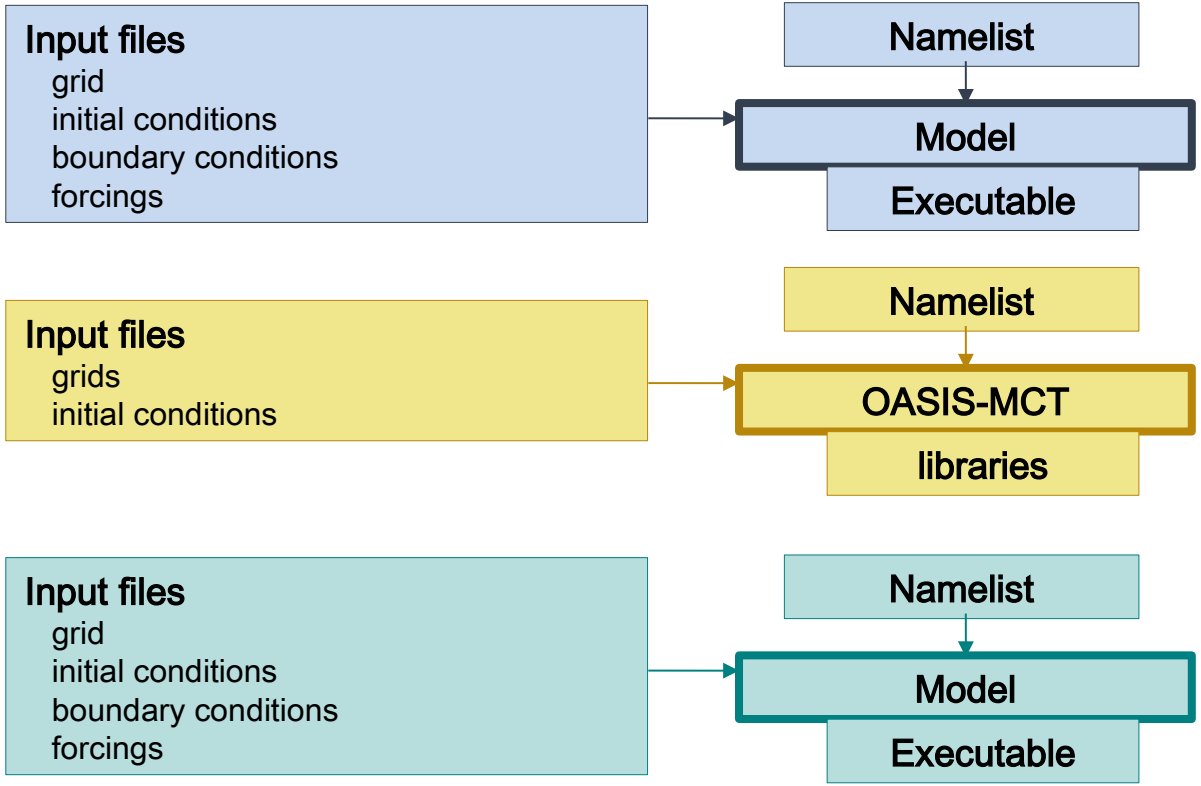


Steps for setting up a coupled simulation

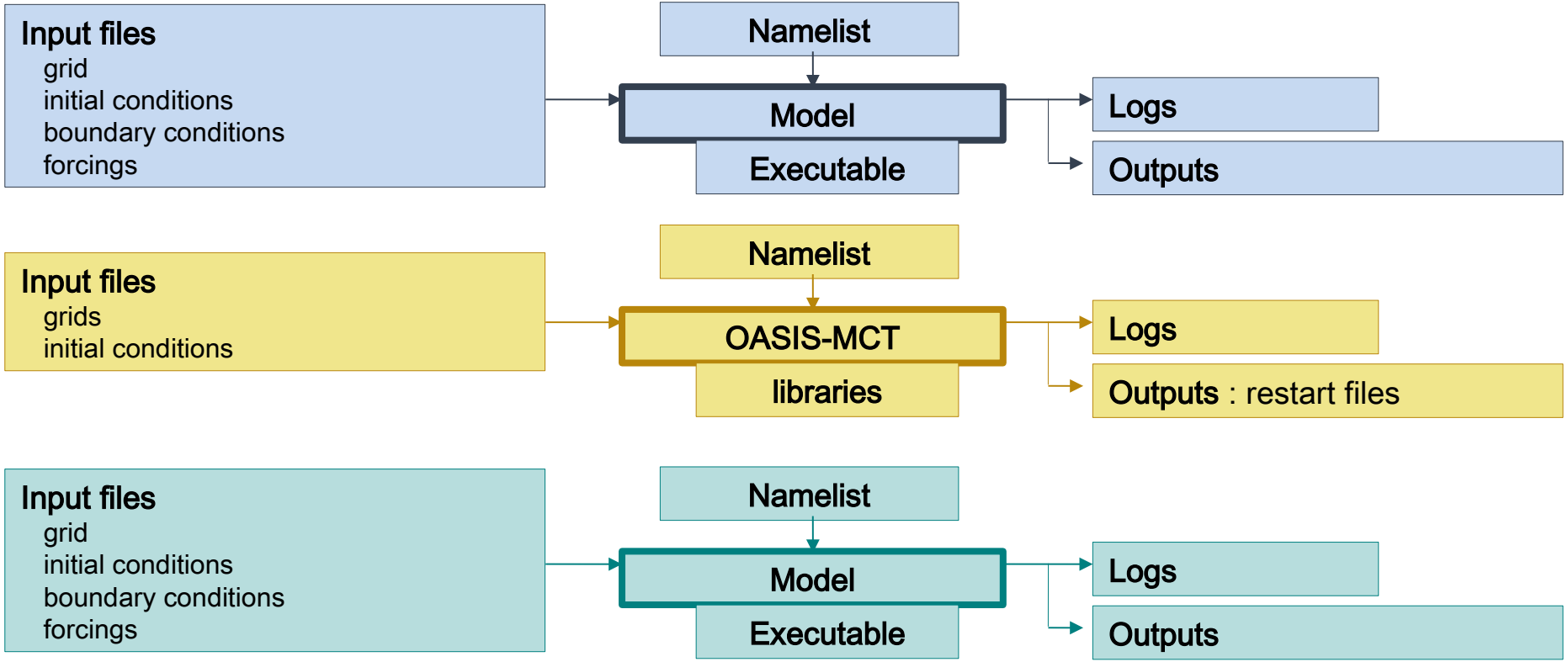




Steps for setting up a coupled simulation

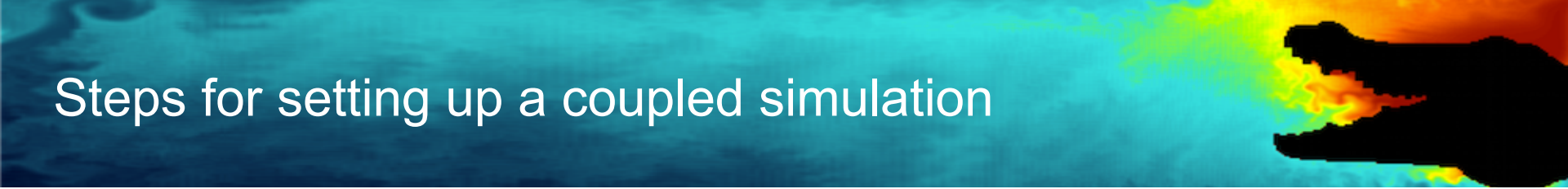


Steps for setting up a coupled simulation



In practice

Steps for setting up a coupled simulation



(1) Get the source codes

(2) Set-up your configuration architecture and environment

(3) **Compile** first OASIS and then your codes (CROCO, WRF...) **with the same netcdf libraries and compilers**

(4) Perform **pre-processing** for your different models

(5) Define the **namelists** and **input files** for OASIS and the different models

(6) **Run** the simulation : all models are simultaneously launched

(7) **Outputs**: check log and ouptut files

We can alternatively **use the coupling toolbox** to perform steps 5-6 more easily (and CROCO compilation)

The BENGUELA configuration

For this tutorial, we will use the BENGUELA_LR grid of CROCO, which is on the South Africa West coast, and is at a 1/3° resolution. The settings for this grid and the time step for CROCO, WW3, and the coupling frequency are:

CROCO
lonmin = 8
lonmax = 22
latmin = -38
latmax = -26
Grid resolution : 1/3°
LLm0=41 MMm0=42
Vertical Levels : N = 32
theta_s = 7.
theta_b = 2.
hc = 200.
vtransform = 2.
Hmin = 75

Time steps :
Baroclinic = 3600s
Barotropic = 60s

WW3
grid = same as CROCO

Time steps :
TMAX (global) = 3600s
TCFL (adv.) = 1200s
TREF (refr.) = 1800s
TSRC (source) = 10s

OASIS
Coupling frequency :
6h → 21600s

Run of 30 days, from Jan 1st 2005

On 2 CPUs for CROCO and 2 CPUs for WW3

1. get the source codes and useful data
   → copy from instructor account

(1) Get the source codes

For this tutorial: login to LEFTRARU:

```
ssh yourlogin@leftraru.nlhpc.cl
```

Get the source codes and useful data

→ generally you can get the codes from the models websites and git repositories (see the 'Download' tutorial on CROCO documentation to do it by yourself on your machine)

→ Here we already got the codes on the machine, you will copy them from instructor01 account

```
cp -r /home/courses/instructor01/CROCO .
cp -r /home/courses/instructor01/WW3 .
cp -r /home/courses/instructor01/OASIS .
ln -s /home/courses/instructor01/DATA .
mkdir CONFIGS
```

Create a CONFIGS directory, where all your simulation configurations will be setup:

```
mkdir CONFIGS
```

(1) Get the source codes

You should now have in your HOME:

```
CROCO
OASIS
WW3
DATA
CONFIGS
```

Suggested work architecture:

$HOME/CROCO
- croco-v1.3.1
- croco_tools-v1.31

$HOME/OASIS
- oasis3-mct

$HOME/WW3

Source codes

$HOME/DATA
- DATASETS_CROCOTOOLS
- CROCO_FILES
- WW3_FILES

Datasets from global reanalyses

$HOME/CONFIGS

Your model configuration directories

(1) Get the source codes

CROCO

OASIS-MCT

WW3

2. set-up the configuration and environment

→ create_config.bash

(2) set-up the configuration

1. **set-up the configuration**
   → create_config.bash

```
cd CONFIGS
cp ../CROCO/croco-v1.3.1/create_config.bash .
```

Edit create_config.bash (e.g. with vi) →

- **set-up the paths,**
- **your configuration name,**
- **the option « all-prod-cpl »**

Run the create_config script:

```
./create_config.bash
```

```
MACHINE="LEFTRARU"

# croco source directory
# ----------------------
CROCO_DIR=~/CROCO/croco-v1.3.1

# croco_tools directory
# ---------------------
TOOLS_DIR=~/CROCO/croco_tools-v1.3.1

# Configuration name
# ------------------
MY_CONFIG_NAME=BENGUELA_CPL

# Home and Work configuration directories
# ---------------------------------------
MY_CONFIG_HOME=~/CONFIGS
MY_CONFIG_WORK=~/CONFIGS

# Options of your configuration
# -----------------------------

## example for production run architecture and
coupling with external models:
options=( all-prod-cpl )
```

(2) set-up the configuration

You should now have in your CONFIGS/BENGUELA_CPL directory:

```
cd BENGUELA_CPL
ll
```

```
create_config.bash.bck  ## Backup of create_config script

myenv_mypath.sh         ## Environment file

mynamelist.sh           ## Script for setting up the coupled experiment (models, inuts, time stepping...)
myjob.sh                ## Script for setting up the coupled experiment (time period, MPI/CPU settings...)

submitjob.sh            ## Script for launching the job
SCRIPTS_TOOLBOX         ## Coupling toolbox

PREPRO                  ## Directory for preprocessing

CROCO_IN                ## Directory for CROCO compilation and settings
CROCO_FILES             ## Directory for CROCO inputs and outputs files

WRF_IN                  ## Directory for WRF compilation and settings/namelist
WRF_FILES               ## Directory for WRF inputs and outputs files

OASIS_IN                ## Directory for OASIS namelists
```

(2) set-up the configuration

Prepare the environment: edit **myenv_mypath.sh**

This file set all the necessary environment variables, modules, and paths for the machine.
→ check the file and edit some paths

```
export CPL="${HOME}/OASIS/oasis3-mct/compile_oasis3-mct_intel_LEFTRARU"
export OCE="${HOME}/CROCO/croco-v1.3.1/OCEAN"
export ATM="${HOME}/WRF/WRF-4.2.1"
export WAV="${HOME}/WW3/WW3-v6.07/model"
```

```
export OCE_FILES_ONLINEDIR="$CWORK/CROCO_FILES"
```

Then source it:

```
source myenv_mypath.sh
```

→ This file is your environment file, it is called at the beginning of all scripts. It also give you the necessary modules. **Source it each time you re-log into the cluster**, in the current config your are working on.

3. compile OASIS and the models

→ (Compile first OASIS and then your codes WW3, CROCO ...)

(3) Compilation

- First compile **OASIS:**

  → already done here by us. Library is in **`$HOME/OASIS/oasis3-mct/compile_oasis3-mct_intel_LEFTRARU`**
  refer to the CROCO online tutorials to do it on your own machine

  → check in **`myenv_mypath.sh`** that the path to the OASIS library is properly set:

```
export CPL="${HOME}/OASIS/oasis3-mct/compile_oasis3-mct_intel_LEFTRARU"
```

```
echo $CPL
ll $CPL
ll $CPL/lib
```

```
libmct.a
libmpeu.a
libpsmile.MPI1.a
libscrip.a
```

(3) Compilation

- Then, compile the models in coupled mode
  NB: with the same netcdf libraries and compilers

  - For **WW3**:
    → already done here by us. Executables are in `$HOME/WW3/WW3-v6.07/model/exe_ow`
    → check in **`myenv_mypath.sh`** that the path to the WW3 executables is properly set:

```
export WAV="${HOME}/WW3/WW3-v6.07/model"
```

```
echo $WAV
```

→ you can check that the executables directories are there:

```
ll $WAV
```

You should have:

```
exe_ow
exe_frc
exe_aw
exe_owa
```

```
ll $WAV/exe_ow
```

```
ww3_grid
ww3_prnc
ww3_shel
...
```

(3) Compilation

- Then, compile the models in coupled mode
  NB: with the same netcdf libraries and compilers
  - For **CROCO**: `cd CROCO_IN`

in `cppdefs.h`, `define OW_COUPLING`
`define MRL_WCI`
`define MPI` (**mandatory**)

```
#if defined REGIONAL
/*
!======================================================
!               REGIONAL (realistic) Configurations
!======================================================
!
!----------------------
! BASIC OPTIONS
!----------------------
!
*/
                        /* Configuration Name */
# define BENGUELA_LR
                        /* Parallelization */
# undef  OPENMP
# define  MPI
                        /* Non-hydrostatic option */
# undef  NBQ
# undef  CROCO_QH
                        /* Nesting */
# undef  AGRIF
# undef  AGRIF_2WAY
                        /* OA and OW Coupling via OASIS (MPI) */
# undef  OA_COUPLING
# define  OW_COUPLING
# ifdef OW_COUPLING
#  undef OW_COUPLING_FULL
# endif
                        /* Wave-current interactions */
# define  MRL_WCI
```

(3) Compilation

- Then, compile the models in coupled mode
  NB: with the same netcdf libraries and compilers

  - For **CROCO**: `cd CROCO_IN`

in `cppdefs.h`, `define OW_COUPLING`
`define MRL_WCI`
`define MPI` (**mandatory**)

`define BULK_FLUX`

```
                    /* Surface Forcing */
/*
! Bulk flux algorithms (options)
! by default : COARE3p0 paramet with GUSTINESS effects
!
! To change bulk param, define one the following keys
(exclusive) :
! - define BULK_ECUMEV0 : ECUME_v0 param
! - define BULK_ECUMEV6 : ECUME_v6 param
! - define BULK_WASP    : WASP param
! Note : gustiness effects can be added for all params
!        by defining BULK_GUSTINESS
*/
# define BULK_FLUX
```

(3) Compilation

- Then, compile the models in coupled mode
  NB: with the same netcdf libraries and compilers

  - For **CROCO**: `cd CROCO_IN`

in **cppdefs.h**, **define OW_COUPLING**
define MRL_WCI
define MPI (**mandatory**)

define BULK_FLUX

in **param.h**: edit the number of CPUs:
NP_XI=1, NP_ETA=2

```
#ifdef MPI
      integer NP_XI, NP_ETA, NNODES
      parameter (NP_XI=1,  NP_ETA=2,  NNODES=NP_XI*NP_ETA)
```

(3) Compilation

- Then, compile the models in coupled mode
  NB: with the same netcdf libraries and compilers

  - For **CROCO**: `cd CROCO_IN`

in **cppdefs.h**, **define OW_COUPLING**
define MRL_WCI
define MPI (**mandatory**)

define BULK_FLUX

in **param.h**: edit the number of CPUs:
NP_XI=1, NP_ETA=2

in **jobcomp**: check OASIS library path:
PRISM_ROOT_DIR=$CPL

```
# set OASIS-MCT (or OASIS3) directories if needed
#
PRISM_ROOT_DIR=${CPL}
```

(3) Compilation

- Then, compile the models in coupled mode
  NB: with the same netcdf libraries and compilers

  - For **CROCO**: `cd CROCO_IN`

    in `cppdefs.h`, `define OW_COUPLING`
    `define MRL_WCI`
    `define MPI` (**mandatory**)

    `define BULK_FLUX`

    in `param.h`: edit the number of CPUs:
    `NP_XI=1, NP_ETA=2`

    in `jobcomp`: check OASIS library path:
    `PRISM_ROOT_DIR=$CPL`

  → Compile: `./jobcomp`

(3) Compilation

- Then, compile the models in coupled mode
  NB: with the same netcdf libraries and compilers

  - For **CROCO**: `cd CROCO_IN`

in `cppdefs.h`, `define OW_COUPLING`
`define MRL_WCI`
`define MPI` (**mandatory**)

`define BULK_FLUX`

in `param.h`: edit the number of CPUs:
`NP_XI=1, NP_ETA=2`

in `jobcomp`: check OASIS library path:
`PRISM_ROOT_DIR=$CPL`

→ Compile: `./jobcomp` → If compilation went well, you should have the executable `croco`

(3) get the CROCO and WW3 exe files in your run directory

→ Prepare a directory where you will run your first simulation:

```
cd ..
mkdir run_test_ow
```

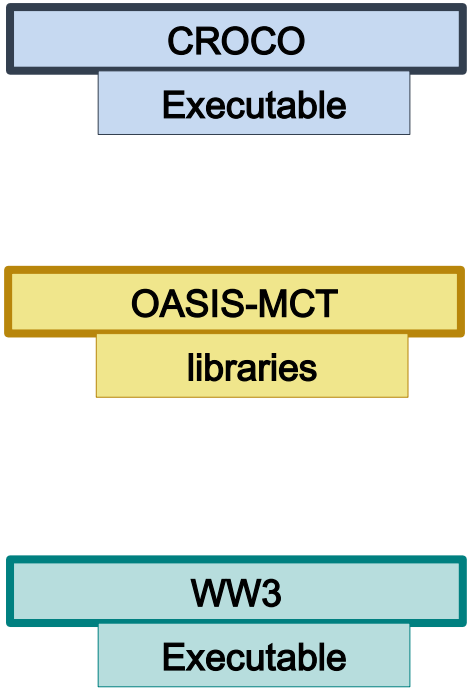
→ Let's go into the run directory:

```
cd run_test_ow
```

→ Copy the executables for CROCO and WW3 and change their name:

```
cp ../CROCO_IN/croco .
mv croco crocox
cp $WAV/exe_ow/ww3* .
mv ww3_shel wwatch
```

(4) get the CROCO and WW3 input files



4. get the CROCO and WW3 input files

→ (here pre-processing has previously been done)

(4) get the CROCO and WW3 input files

→ For **CROCO**:

NB : you normally should prepare the input files following CROCO usual pre-processing
Here we will use files that we have already prepared

First link the inputs files to your **CROCO_FILES** directory:

```
cd ../CROCO_FILES
ln -s $HOME/DATA/CROCO_FILES/* .
```

Then, link them into your run directory with appropriate names:

```
cd ../run_test_ow
ln -s ../CROCO_FILES/croco_grd.nc .
ln -s ../CROCO_FILES/croco_ini_SODA_Y2005M01.nc croco_ini.nc
ln -s ../CROCO_FILES/croco_bry_SODA_Y2005M01.nc croco_bry.nc
ln -s ../CROCO_FILES/croco_blk_ERA5_Y2005M01.nc croco_blk.nc
```

(4) get the CROCO and WW3 input files

→ For **WW3**:

NB: you normally should prepare the input files following WW3 usual pre-processing (gridgen) or eventually using CROCO files with the following given scripts provided in **`WW3_IN`**:

`make_ww3_grd_input_files_from_croco_grd.m`
`script_make_CROCO_current_and_level_for_ww3.sh`
`script_make_CFSR_wind_for_ww3.sh` **or** `script_make_WRF_wind_for_ww3.sh`

Here we will use files that we have already prepared.

First link the inputs files to your **`CROCO_FILES`** directory:

```
cd ../WW3_FILES
ln -s $HOME/DATA/WW3_FILES/* .
```

Then, link them into your run directory with appropriate names:

```
cd ../run_test_ow

ln -s ../WW3_FILES/*.inp .
ln -s ../WW3_FILES/ERA5_wind_Y2005M01.nc wind.nc
```

(4) get the CROCO and WW3 input files

Input files
- grid
- initial conditions
- boundary conditions
- forcings

→ CROCO Executable

OASIS-MCT libraries

Input files
- grid
- initial conditions
- boundary conditions
- forcings

But… text and netcdf files that will need to be transformed to binaires for WW3

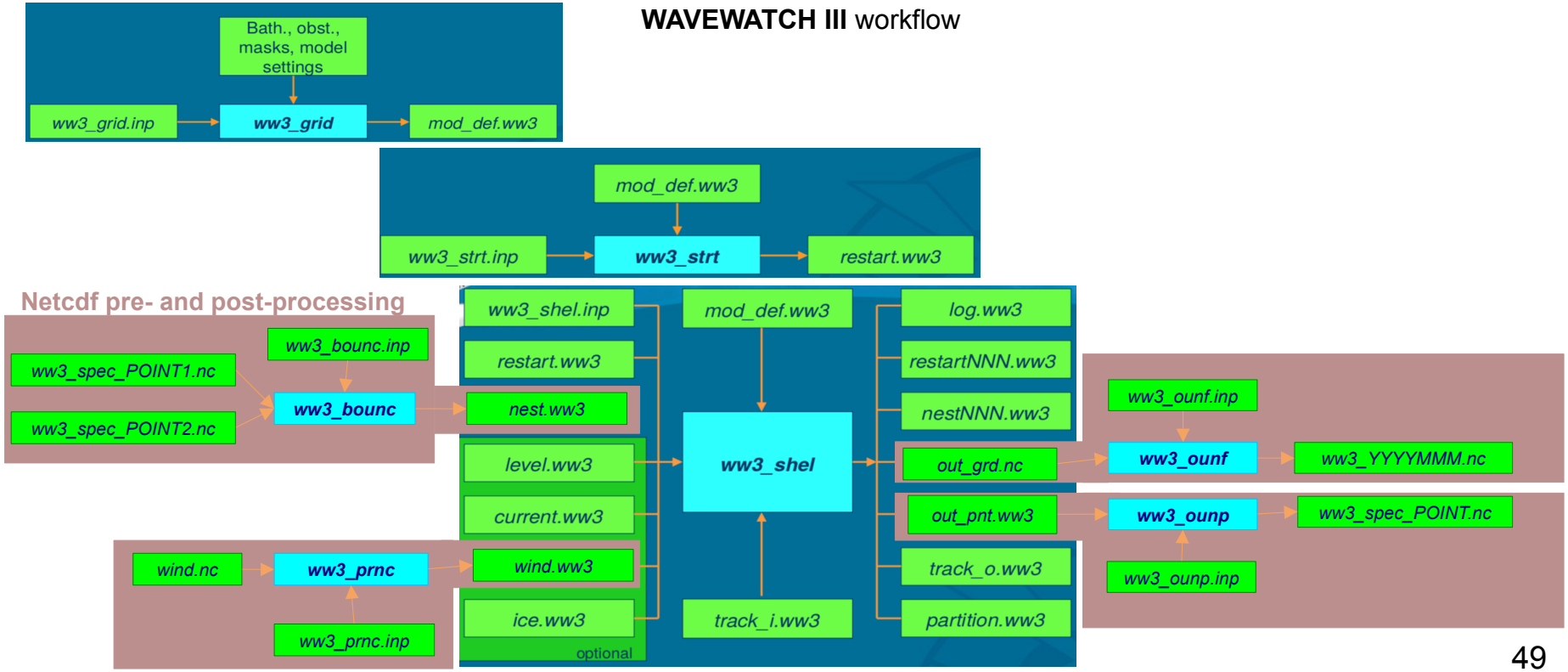
→ WW3 Executable

5. define namelists and create input files

→ (here CROCO and WW3 pre-processing has previously been done)

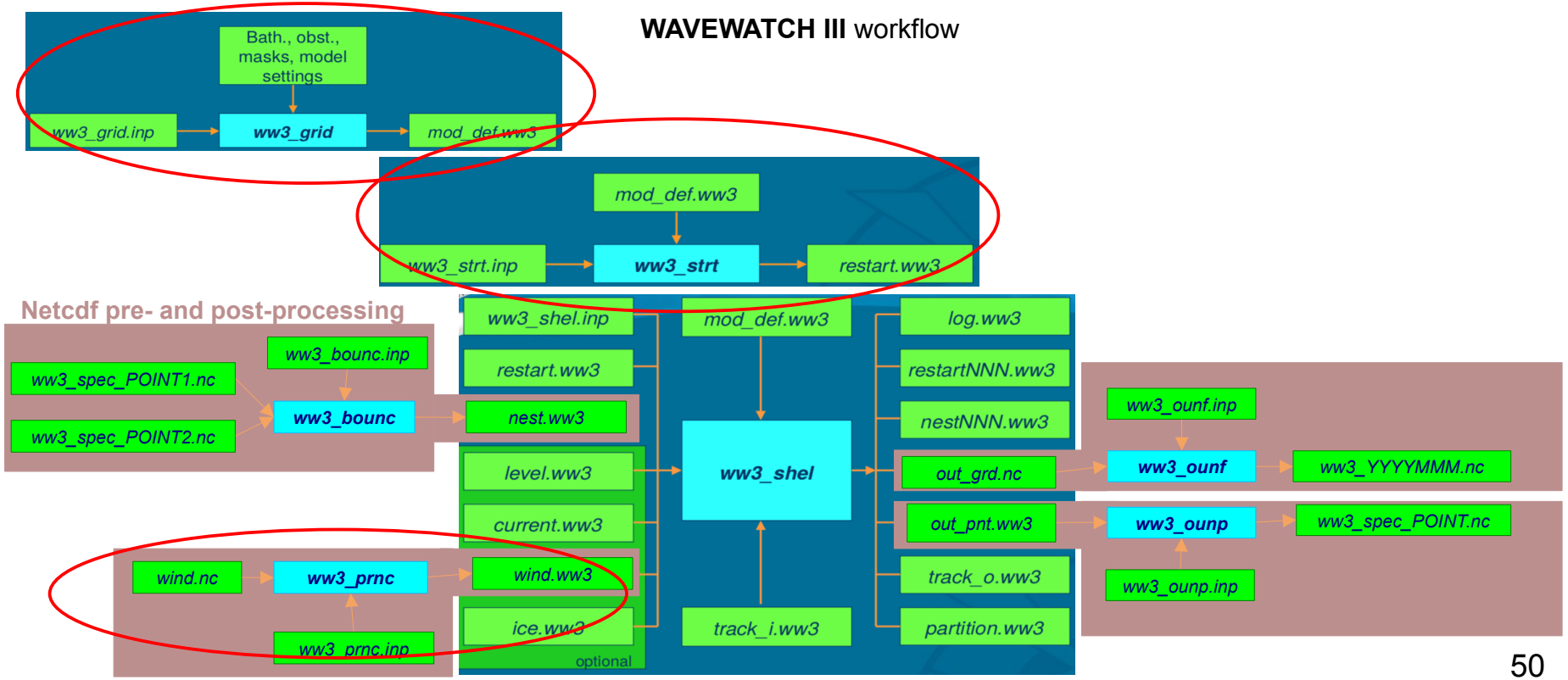
WW3 in practice

WAVEWATCH III workflow



WW3 in practice

WAVEWATCH III workflow



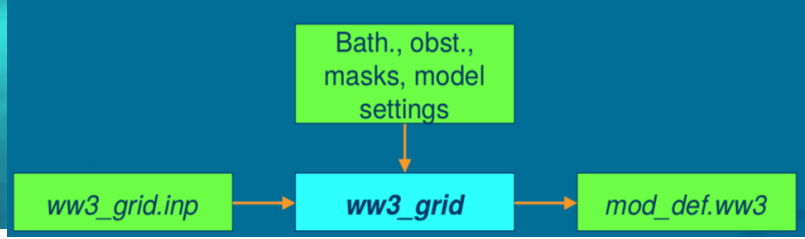
(5) Define namelist and input files

WW3

Copy WW3 namelists into your run directory:

```
cp ../WW3_IN/ww3_grid.inp.base ww3_grid.inp
cp ../WW3_IN/ww3_prnc.inp.wind ww3_prnc.inp
cp ../WW3_IN/ww3_strt.inp .
cp ../WW3_IN/ww3_shel.inp.base.ow ww3_shel.inp
cp ../WW3_IN/ww3_ounf.inp.base ww3_ounf.inp
```

(5) Define namelist and input files



WW3

Then edit the **ww3_grid.inp** :

```
$ Grid name (C*30, in quotes) ----------------------------------------- $
$
  'BENGUELA_CPL'
```

```
$ Time steps ------------------------------------------------------------ $
$ max global dt, max propag. dt, refraction dt, min source term dt --- $
$   TMAX            TCFL            TREF           TSRC             -- $
$ (all in seconds, 2nd and 3rd dt should be at least half of global) - $
$ As a first guess you can try (and decrease if not ok):             -- $
$ TMAX = 3 x TCFL                                                    -- $
$ TCFL = 0.8 x dx/(g/fmin4pi) with fmin=0.0373 => 3-4 % of dx        -- $
$ TREF = TMAX / 2 or much less if strong current area like TMAX / 10 - $
$ TSRC = usually 10s  (could be between 5s and 60s)                  -- $
  3600 1200 1800 10
```

```
     'CURV'  T  'NONE'
     41  42
     21 1 0 1 1 '(....)' 'NAME' 'x.inp'
     22 1 0 1 1 '(....)' 'NAME' 'y.inp'
$
     -75 10 23 1 1 1 '(....)' 'NAME' 'bottom.inp'
$              24 1 1 1 '(....)'  'NAME' 'obst.inp'
              25 1 1 '(....)'  'NAME' 'mask_and_bdy.inp'
```

Choose your config title

Set model time steps
NB : global time step should be a multiple of the coupling time step

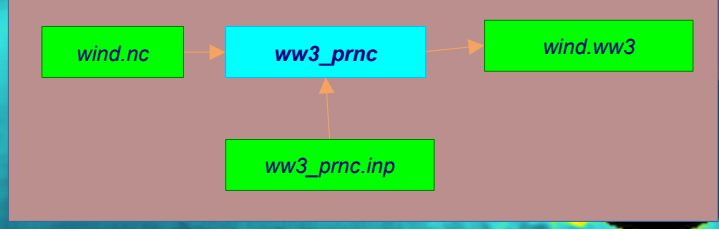
Grid settings (according to your WW3 grid preprocessing)

Run **ww3_grid** :

```
./ww3_grid > ww3_grid.out
```

→ you should now have **mod_def.ww3**

(5) Define namelist and input files



WW3

Then prepare the wind forcing : check and eventually edit the `ww3_prnc.inp` :

```
  'WND' 'LL' T T
$
$ Name spatial of dimensions ------------------------------------------ $
$ NB: time dimension is expected to be called 'time' ----------------- $
$
  longitude latitude
$
$ Variables to use ---------------------------------------------------- $
$
  u10 v10
$
$ Additional time input ----------------------------------------------- $
$ If time flag is F, give time of field in yyyymmdd hhmmss format ---- $
$
$   20160101 060000
$
$ Define data files --------------------------------------------------- $
$ Filename(s) of the forcing field (in quotes, one per line)        -- $
$
  'wind.nc'
```

lon/lat variables names found in the wind netcdf file

Variables names found in the wind netcdf file

Name of the wind netcdf file

Run `ww3_prnc` : `srun ww3_prnc > ww3_prnc.out`

→ you should now have `wind.ww3`

(5) Define namelist and input files

WW3

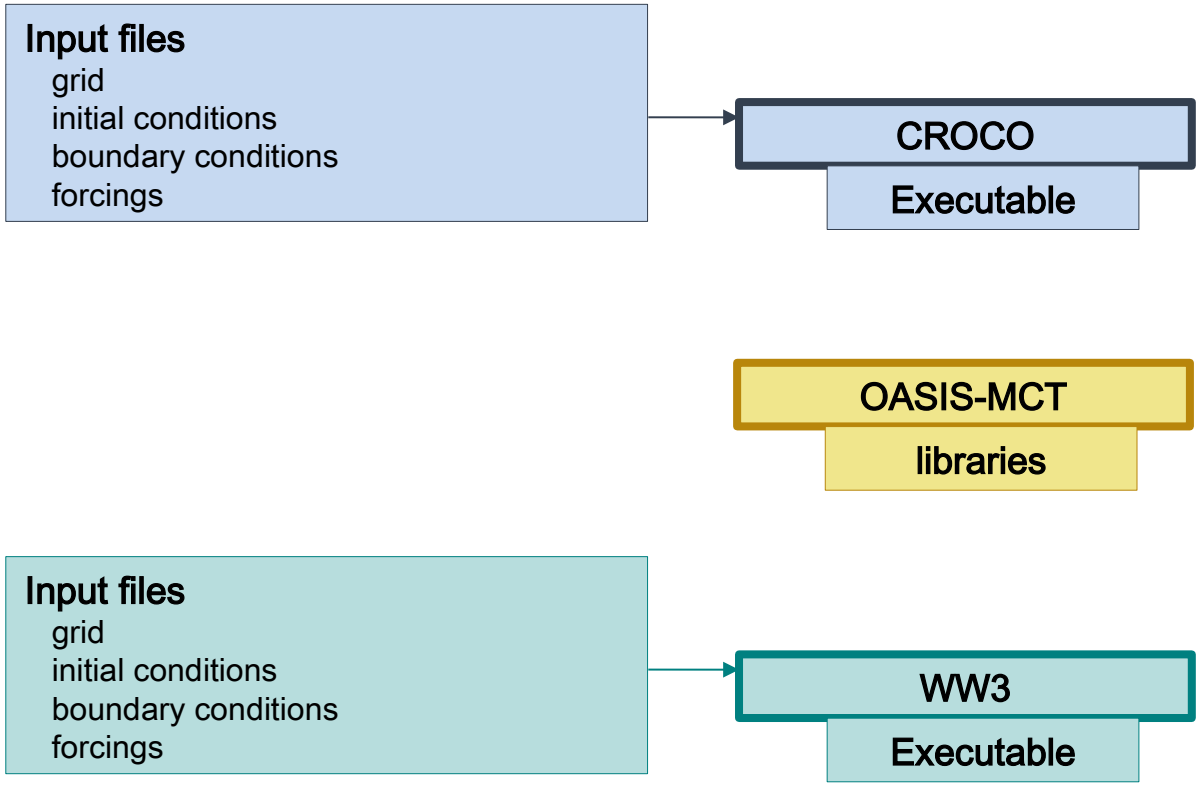
Then prepare the initial conditions : check and eventually edit the `ww3_strt.inp` :

```
$ type of initial field                                              -- $
$                                                                    -- $
$ 1 : Gaussian in freq and space, cos type in direction              -- $
$     fp, spread, mean dir, cos power, Xm, spread, Ym, spread, Hmax   - $
$                                                                    -- $
$ 2 : JONSWAP spectrum with Hasselmann et al. (1980) dir. distrib.   -- $
$     alfa, fp, mean dir, gamma, sigA, sigB, Xm, spread, Ym, spread   - $
$                                                                    -- $
$ 3 : Fetch-limited JONSWAP                                          -- $
$     no additional data (computed from local wind)                  -- $
$                                                                    -- $
$ 4 : User-defined spectrum                                          -- $
$     scale factor                                                   -- $
$     spectrum                                                       -- $
$                                                                    -- $
$ 5 : Calm conditions                                                -- $
$     no additional data
$
  3
$
```

Here we start with a JONSWAP spectrum computed from the local wind

Run `ww3_strt` : `srun ww3_strt > ww3_strt.out` → you should now have `restart.ww3`

(4) get the CROCO and WW3 input files



(5) Define namelist and input files

WW3

Then edit the `ww3_shel.inp` :

```
   C F     Water levels
   C F     Currents
   T F     Winds
```

```
$ Time frame of calculations ------------------------------------------ $
$   Starting time in yyyymmdd hhmmss format                            -- $
$   Ending time in yyyymmdd hhmmss format                              -- $
$
   20050101 000000
   20050131 000000
```

```
$ Type 1 : Fields of mean wave parameter                               -- $
$          Define outputs fields as namelists:                         -- $
$                 one line set to N                                    -- $
$                 next line contains parameter name                    -- $
$
  20050101 000000  21600  20050131 000000
```

Here **C** means that water levels and currents are **coupled**
T means that wind is **forced**

Set dates

Set dates and output frequency

(5) Define namelist and input files

WW3

Then edit the `ww3_shel.inp` :

```
$ Type 2 : Point output                                                -- $
$          Lines with lon, lat, name (C*10) of output points           -- $
$          The list is closed by defining a point named 'STOPSTRING' - $
$
  20050101 000000   0   20050131 000000
$
$ ---------------------------------------------------------------------- $
$ Type 3 : Output along track : flag T/F                               -- $
$
  20050101 000000   0   20050131 000000
$
$ ---------------------------------------------------------------------- $
$ Type 4 : Restart files (no additional data required)                 -- $
$
  20050101 000000   2592000   20050131 000000
$
$ ---------------------------------------------------------------------- $
$ Type 5 : Boundary data (no additional data required)                 -- $
$
  20050101 000000   0   20050131 000000
$
$ ---------------------------------------------------------------------- $
$ Type 6 : Separated wave field data (dummy for now)                   -- $
$
  20050101 000000   0   20050131 000000
```

Set dates and output frequency :

Here no point outputs, no track outputs

And no boundary data

Only one restart output after 30 days

(5) Define namelist and input files

WW3

Then edit the `ww3_shel.inp` :

```
$ ------------------------------------------------------------------- $
$ Type 7 : Coupled fields                                             -- $
$          (must be fully commented if not used with switch COU)      -- $
$          Time stride must be the same as the maximum global time step$
$          Namelist type selection is used here                       -- $
$                   start with N                                      -- $
$                   Possibly sent fields by ww3 :                     -- $
$                      T0M1 OHS DIR THM BHD TWO UBR FOC TAW TUS USS -- $
$                      LM DRY ACHA OCHA AHS TP (or FP)                -- $
$                   Possibly received fields by ww3 :                 -- $
$                      SSH CUR WND                                    -- $
$                   NB: SSH and CUR cannot be used separately         -- $
$
  20050101 000000  3600  20050131 000000
  N
  T0M1 OHS THM
  SSH CUR
```

→ your WW3 configuration is now ready

Here are the detailed coupled settings :

- Time and frequency : here it is mandatory to put the frequency to be the **same than the WW3 global time step**
- WW3 keywords for variables to be sent and received

(5) Define namelist and input files

CROCO

Copy CROCO namelist `croco.in` into your run directory:

```
cp ../CROCO_IN/croco.in.base .
mv croco.in.base croco.in
```

Then edit the `croco.in` :

```
title:
       BENGUELA_CPL
time_stepping: NTIMES   dt[sec]  NDTFAST  NINFO
               720      3600      60        1
S-coord: THETA_S,   THETA_B,    Hc          (m)
          7          2          200

start_date:
<dstart>-<mstart>-<ystart> 00:00:00
origin_date:
<dorig>-<morig>-<yorig> 00:00:00
```

Choose your config title

- Number time steps to run 30 days
- Baroclinic time step
- Number of baroytropic time steps for one baroclinic time step

Vertical grid settings (according to your CROCO grid preprocessing)

(5) Define namelist and input files

CROCO

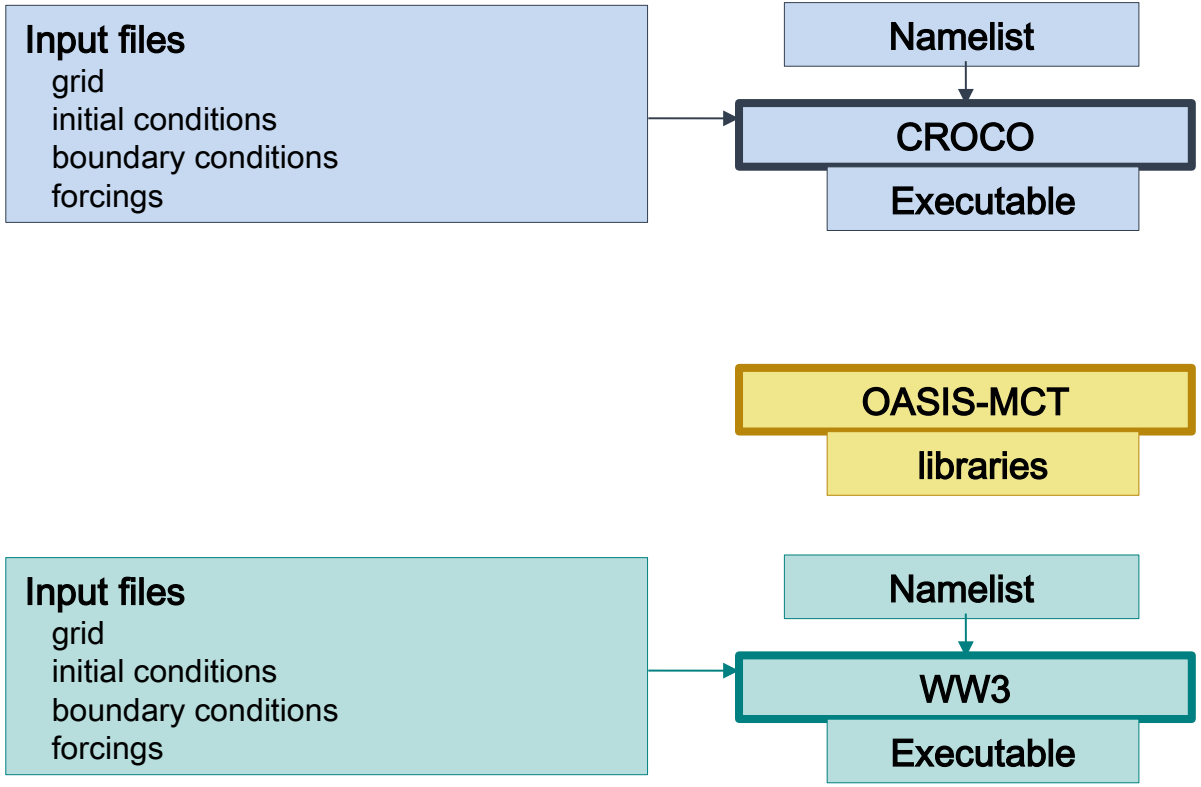
```
grid:  filename
                          ./croco_grd.nc
forcing: filename
                          ./croco_frc.nc
bulk_forcing: filename
                          ./croco_blk.nc
climatology: filename
                          ./croco_clm.nc
boundary: filename
                          ./croco_bry.nc
initial: NRREC  filename
          1
                          ./croco_ini.nc
```

Check that you point towards local CROCO files

```
restart:          NRST, NRPFRST / filename
                  720    -1
                           ./croco_rst.nc
history: LDEFHIS, NWRT, NRPFHIS / filename
            T     24    0
                           ./croco_his.nc
averages: NTSAVG, NAVG, NRPFAVG / filename
            1     24    0
                           ./croco_avg.nc
```

Check when you want restart and outputs (avg and his)

Steps for setting up a coupled simulation



(5) Define namelist and input files

OASIS

Get the OASIS `namcouple` into your run directory:

```
cp ../OASIS_IN/namcouple.base.ow .
mv namcouple.base.ow namcouple
```

Then edit the `namcouple` :

```
# NFIELDS: total number of fields being exchanged
$NFIELDS
6
#########################################################################
# NBMODEL: number of models and their names (6 characters)
$NBMODEL
2 wwatch crocox
#########################################################################
# RUNTIME: total simulated time for the actual run in seconds (<I8)
$RUNTIME
2592000
#########################################################################
# NLOGPRT: debug and time statistics informations printed in log file
$NLOGPRT
1 1
#########################################################################
```

Number of fields to be exchanged

Number of models and names of the executables

Time of the simulation in seconds (here 30 days)

Debug level

(5) Define namelist and input files

```
#                    ------------------------------------
#                      CROCO (crocox) ==> WW3 (wwatch)
#                    ------------------------------------
#
#~~~~~~~~~~~~
# SSH : sea surface height (m)
#~~~~~~~~~~~~
CROCO_SSH WW3__SSH 1 <cpldt> 2 oce.nc EXPORTED
<ocenx> <oceny> <wavnx> <wavny> ocnt ww3t LAG=<ocedt>
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# UOCE : sea surface zonal currents (m.s-1)
#~~~~~~~~~~~~
CROCO_EOCE WW3_OSSU 1 <cpldt> 2 oce.nc EXPORTED
<ocenx> <oceny> <wavnx> <wavny> ocnu ww3t LAG=<ocedt>
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# VOCE : sea surface meridional currents (m.s-1)
#~~~~~~~~~~~~
#
CROCO_NOCE WW3_OSSV 1 <cpldt> 2 oce.nc EXPORTED
<ocenx> <oceny> <wavnx> <wavny> ocnv ww3t LAG=<ocedt>
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
```

Variable names for OASIS for Sending and Receiving models :

- SSH
- Ocean eastward velocity
- Ocean northward velocity

(5) Define namelist and input files

```
#                    -------------------------------------
#                      CROCO (crocox) ==> WW3 (wwatch)
#                    -------------------------------------
#
#~~~~~~~~~~~~
# SSH : sea surface height (m)
#~~~~~~~~~~~~
CROCO_SSH WW3__SSH 1 21600 2 oce.nc EXPORTED
<ocenx> <oceny> <wavnx> <wavny> ocnt ww3t LAG=<ocedt>
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# UOCE : sea surface zonal currents (m.s-1)
#~~~~~~~~~~~~
CROCO_EOCE WW3_OSSU 1 21600 2 oce.nc EXPORTED
<ocenx> <oceny> <wavnx> <wavny> ocnu ww3t LAG=<ocedt>
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# VOCE : sea surface meridional currents (m.s-1)
#~~~~~~~~~~~~
#
CROCO_NOCE WW3_OSSV 1 21600 2 oce.nc EXPORTED
<ocenx> <oceny> <wavnx> <wavny> ocnv ww3t LAG=<ocedt>
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
```

Coupling time step in seconds (here =6h)

(5) Define namelist and input files

```
#                    ------------------------------------
#                      CROCO (crocox) ==> WW3 (wwatch)
#                    ------------------------------------
#
#~~~~~~~~~~~~
# SSH : sea surface height (m)
#~~~~~~~~~~~~
CROCO_SSH WW3__SSH 1 21600 2 oce.nc EXPORTED
 41  42  <wavnx> <wavny> ocnt ww3t LAG=<ocedt>
R   0   R   0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# UOCE : sea surface zonal currents (m.s-1)
#~~~~~~~~~~~~
CROCO_EOCE WW3_OSSU 1 21600 2 oce.nc EXPORTED
 41  42  <wavnx> <wavny> ocnu ww3t LAG=<ocedt>
R   0   R   0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# VOCE : sea surface meridional currents (m.s-1)
#~~~~~~~~~~~~
#
CROCO_NOCE WW3_OSSV 1 21600 2 oce.nc EXPORTED
 41  42  <wavnx> <wavny> ocnv ww3t LAG=<ocedt>
R   0   R   0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
```

Ocean grid
NX NY
(same as in param.h :
`LLm0=41,  MMm0=42`)

(5) Define namelist and input files

```
#                    ------------------------------------
#                      CROCO (crocox) ==> WW3 (wwatch)
#                    ------------------------------------
#
#~~~~~~~~~~~~
# SSH : sea surface height (m)
#~~~~~~~~~~~~
CROCO_SSH WW3__SSH 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnt ww3t LAG=<ocedt>
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# UOCE : sea surface zonal currents (m.s-1)
#~~~~~~~~~~~~
CROCO_UOCE WW3_OSSU 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnu ww3t LAG=<ocedt>
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# VOCE : sea surface meridional currents (m.s-1)
#~~~~~~~~~~~~
#
CROCO_VOCE WW3_OSSV 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnv ww3t LAG=<ocedt>
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
```

Wave model grid
NX, NY
(here similar to
CROCO grid)

(5) Define namelist and input files

```
#                    ------------------------------------
#                      CROCO (crocox) ==> WW3 (wwatch)
#                    ------------------------------------
#
#~~~~~~~~~~~~
# SSH : sea surface height (m)
#~~~~~~~~~~~~
CROCO_SSH WW3__SSH 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnt ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# UOCE : sea surface zonal currents (m.s-1)
#~~~~~~~~~~~~
CROCO_EOCE WW3_OSSU 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnu ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# VOCE : sea surface meridional currents (m.s-1)
#~~~~~~~~~~~~
#
CROCO_NOCE WW3_OSSV 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnv ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
```

Grid names for OASIS

(5) Define namelist and input files

```
#                    -------------------------------------
#                      CROCO (crocox) ==> WW3 (wwatch)
#                    -------------------------------------
#
#~~~~~~~~~~~~
# SSH : sea surface height (m)
#~~~~~~~~~~~~
CROCO_SSH WW3__SSH 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnt ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# UOCE : sea surface zonal currents (m.s-1)
#~~~~~~~~~~~~
CROCO_EOCE WW3_OSSU 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnu ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# VOCE : sea surface meridional currents (m.s-1)
#~~~~~~~~~~~~
#
CROCO_NOCE WW3_OSSV 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnv ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
```

Sending model time step = CROCO baroclinic time step

(5) Define namelist and input files

```
#                    -------------------------------------
#                      CROCO (crocox) ==> WW3 (wwatch)
#                    -------------------------------------
#
#~~~~~~~~~~~~
# SSH : sea surface height (m)
#~~~~~~~~~~~~
CROCO_SSH WW3__SSH 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnt ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# UOCE : sea surface zonal currents (m.s-1)
#~~~~~~~~~~~~
CROCO_EOCE WW3_OSSU 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnu ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# VOCE : sea surface meridional currents (m.s-1)
#~~~~~~~~~~~~
#
CROCO_NOCE WW3_OSSV 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnv ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
```

Types of grids : R = regional
Number of overlapping points = 0
For each model grid

(5) Define namelist and input files

```
#                    -------------------------------------
#                      CROCO (crocox) ==> WW3 (wwatch)
#                    -------------------------------------
#
#~~~~~~~~~~~~
# SSH : sea surface height (m)
#~~~~~~~~~~~~
CROCO_SSH WW3__SSH 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnt ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# UOCE : sea surface zonal currents (m.s-1)
#~~~~~~~~~~~~
CROCO_EOCE WW3_OSSU 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnu ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# VOCE : sea surface meridional currents (m.s-1)
#~~~~~~~~~~~~
#
CROCO_NOCE WW3_OSSV 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnv ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
```

Number of transformations

(5) Define namelist and input files

```
#                    -------------------------------------
#                      CROCO (crocox) ==> WW3 (wwatch)
#                    -------------------------------------
#
#~~~~~~~~~~~~
# SSH : sea surface height (m)
#~~~~~~~~~~~~
CROCO_SSH WW3__SSH 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnt ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# UOCE : sea surface zonal currents (m.s-1)
#~~~~~~~~~~~~
CROCO_EOCE WW3_OSSU 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnu ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# VOCE : sea surface meridional currents (m.s-1)
#~~~~~~~~~~~~
#
CROCO_NOCE WW3_OSSV 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnv ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
```

Keywords for transformations :
LOCTRANS = time
SCRIPR = grid

(5) Define namelist and input files

```
#                    -------------------------------------
#                      CROCO (crocox) ==> WW3 (wwatch)
#                    -------------------------------------
#
#~~~~~~~~~~~~
# SSH : sea surface height (m)
#~~~~~~~~~~~~
CROCO_SSH WW3__SSH 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnt ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# UOCE : sea surface zonal currents (m.s-1)
#~~~~~~~~~~~~
CROCO_EOCE WW3_OSSU 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnu ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# VOCE : sea surface meridional currents (m.s-1)
#~~~~~~~~~~~~
#
CROCO_NOCE WW3_OSSV 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnv ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
```

Type of time transformation :
AVERAGE of all model time steps within a coupling time step

(5) Define namelist and input files

```
#                    ------------------------------------
#                      CROCO (crocox) ==> WW3 (wwatch)
#                    ------------------------------------
#
#~~~~~~~~~~~~
# SSH : sea surface height (m)
#~~~~~~~~~~~~
CROCO_SSH WW3__SSH 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnt ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# UOCE : sea surface zonal currents (m.s-1)
#~~~~~~~~~~~~
CROCO_EOCE WW3_OSSU 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnu ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# VOCE : sea surface meridional currents (m.s-1)
#~~~~~~~~~~~~
#
CROCO_NOCE WW3_OSSV 1 21600 2 oce.nc EXPORTED
41 42 41 42 ocnv ww3t LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
```

Type of grid interpolation : here distributed weight on a lon-lat grid

(5) Define namelist and input files

```
#                     -------------------------------------
#                       WW3 (wwatch) ==> CROCO (crocox)
#                     -------------------------------------
#
#~~~~~~~~~~~~
# T0M1 / ww_tmn : mean period (s)
#~~~~~~~~~~~~
WW3_T0M1 CROCO_T0M1 1 <cpldt> 2 wav.nc EXPORTED
<wavnx> <wavny> <ocenx> <oceny> ww3t ocnt LAG=<wavdt>
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# HS / ww_hs : HS significant wave height (m)
#~~~~~~~~~~~~
WW3__OHS CROCO_HS 1 <cpldt> 2 wav.nc EXPORTED
<wavnx> <wavny> <ocenx> <oceny> ww3t ocnt LAG=<wavdt>
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# THM / ww_thm : mean direction
#~~~~~~~~~~~~
WW3__DIR CROCO_DIR 1 <cpldt> 2 wav.nc EXPORTED
<wavnx> <wavny> <ocenx> <oceny> ww3t ocnt LAG=<wavdt>
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
```

Variable names for OASIS for Sending and Receiving models :

- Mean wave period
- Significant wave height
- Mean wave direction

(5) Define namelist and input files

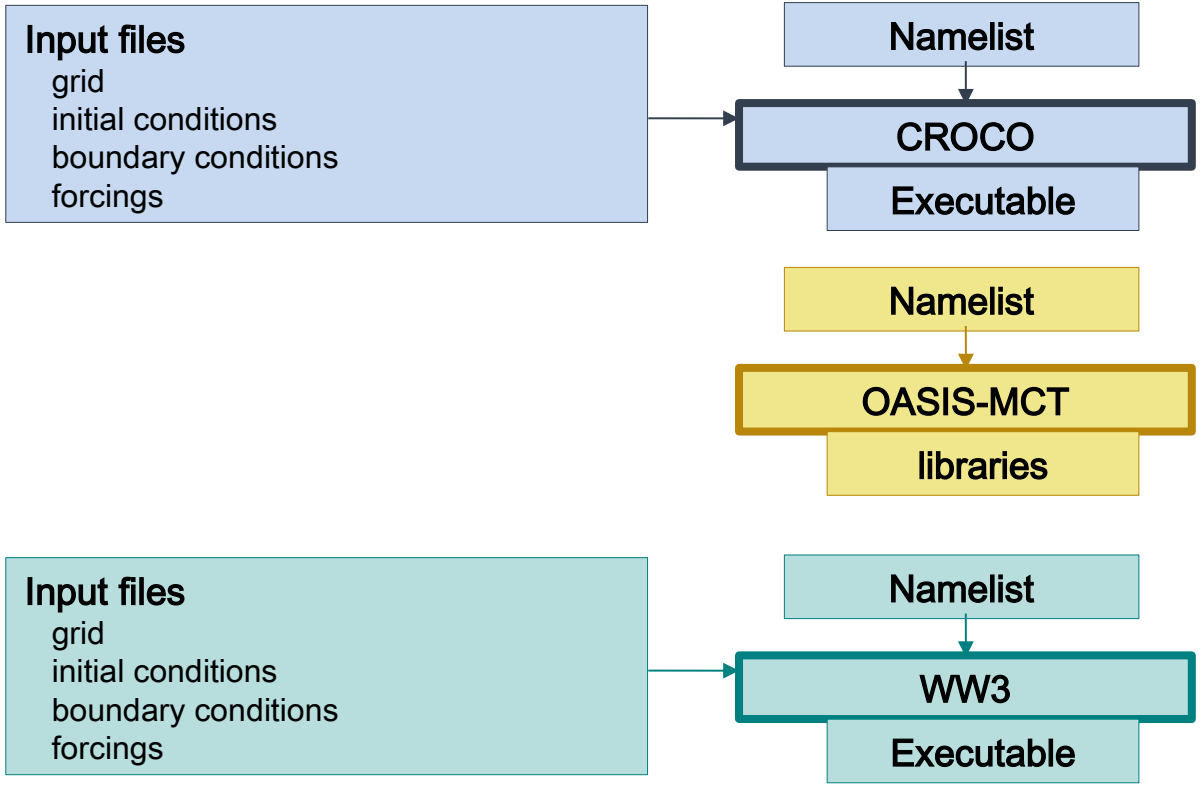
```
#                    -------------------------------------
#                      WW3 (wwatch) ==> CROCO (crocox)
#                    -------------------------------------
#
#~~~~~~~~~~~~
# T0M1 / ww_tmn : mean period (s)
#~~~~~~~~~~~~
WW3_T0M1 CROCO_T0M1 1 21600 2 wav.nc EXPORTED
41 42 41 42 ww3t ocnt LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# HS / ww_hs : HS significant wave height (m)
#~~~~~~~~~~~~
WW3__OHS CROCO_HS 1 21600 2 wav.nc EXPORTED
41 42 41 42 ww3t ocnt LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
#
#~~~~~~~~~~~~
# THM / ww_thm : mean direction
#~~~~~~~~~~~~
WW3__DIR CROCO_DIR 1 21600 2 wav.nc EXPORTED
41 42 41 42 ww3t ocnt LAG=3600
R  0  R  0
LOCTRANS SCRIPR
AVERAGE
DISTWGT LR SCALAR LATLON 1 4
```

Replace accordingly :

- the coupling time step
- the grids' number of points
- the WW3 model global time step for the LAG

Sending model time step = WW3 global time step

Steps for setting up a coupled simulation



(5) Define namelist and input files

OASIS

Prepare OASIS restart files:

→ OASIS needs restart files with the variables to exchange for the first time step
These files (one for each model) should contain variables with OASIS names
→ 2 scripts are provided to build these files in `SCRIPTS_TOOLBOX/OASIS_SCRIPTS`

`create_oasis_restart_from_calm_conditions.sh`
`create_oasis_restart_from_preexisting_output_files.sh`

We will start from calm conditions.

```
cp ../SCRIPTS_TOOLBOX/OASIS_SCRIPTS/create_oasis_restart_from_calm_conditions.sh .

ln -s ../WW3_FILES/ww3_200501.nc .
./create_oasis_restart_from_calm_conditions.sh ww3_200501.nc wav.nc ww3 "WW3_T0M1 WW3__OHS WW3__DIR"

./create_oasis_restart_from_calm_conditions.sh croco_grd.nc oce.nc croco "CROCO_SSH CROCO_NOCE CROCO_EOCE"
```

File where to find the mask

OASIS restart file names

From which model

Which variables

→ you should now have : `wav.nc` and `oce.nc`

(5) Define namelist and input files

OASIS

→ you should now have : **wav.nc** and **oce.nc**

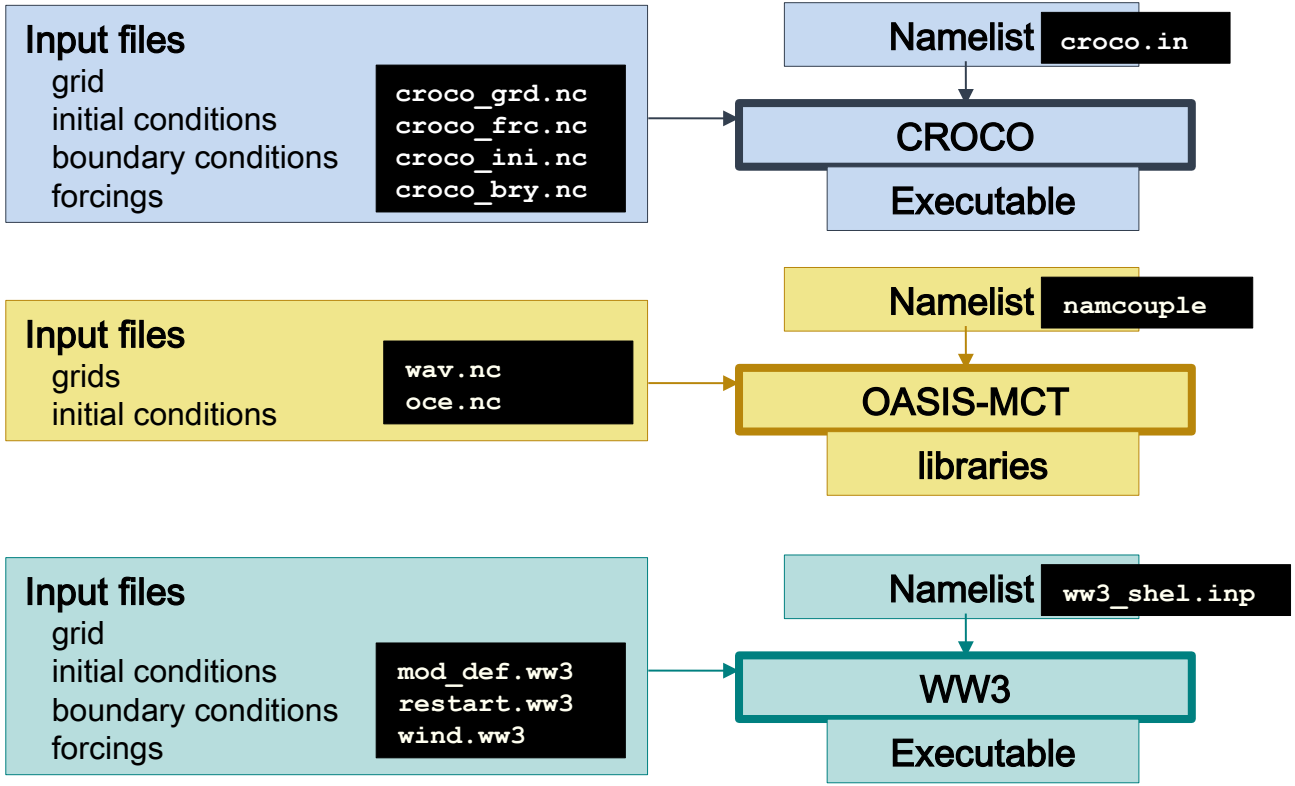
ncdump -h oce.nc

```
ncdump -h oce.nc
netcdf oce {
dimensions:
        eta_rho = 42 ;
        xi_rho = 41 ;
variables:
        double CROCO_SSH(eta_rho, xi_rho) ;
                CROCO_SSH:long_name = "mask on RHO-points" ;
                CROCO_SSH:option_0 = "land" ;
                CROCO_SSH:option_1 = "water" ;
        double CROCO_NOCE(eta_rho, xi_rho) ;
                CROCO_NOCE:long_name = "mask on RHO-
points" ;
                CROCO_NOCE:option_0 = "land" ;
                CROCO_NOCE:option_1 = "water" ;
        double CROCO_EOCE(eta_rho, xi_rho) ;
                CROCO_EOCE:long_name = "mask on RHO-
points" ;
                CROCO_EOCE:option_0 = "land" ;
                CROCO_EOCE:option_1 = "water" ;
```

ncdump -h wav.nc

```
ncdump -h wav.nc
netcdf wav {
dimensions:
        latitude = 42 ;
        longitude = 41 ;
variables:
        float latitude(latitude, longitude) ;
                latitude:units = "degree_north" ;
                latitude:long_name = "latitude" ;
                latitude:standard_name = "latitude" ;
                latitude:valid_min = -90.f ;
                latitude:valid_max = 180.f ;
                latitude:axis = "Y" ;
        float longitude(latitude, longitude) ;
                longitude:units = "degree_east" ;
                longitude:long_name = "longitude" ;
                longitude:standard_name = "longitude" ;
                longitude:valid_min = -180.f ;
                longitude:valid_max = 360.f ;
                longitude:axis = "X" ;
        double WW3_T0M1(latitude, longitude) ;
                WW3_T0M1:long_name = "status map" ;
                WW3_T0M1:standard_name = "status map" ;
                WW3_T0M1:units = "1" ;
                WW3_T0M1:valid_max = 32 ;
                WW3_T0M1:valid_min = -32 ;
        double WW3__OHS(latitude, longitude) ;
                WW3__OHS:long_name = "status map" ;
```

Steps for setting up a coupled simulation



6. launch the simulation and look at outputs

→ models are launched together

(6) Run the simulation

Now you are ready to launch the run.

The models are launched simultaneously:

```
srun -n 2 wwatch : -n 2 crocox
```

Number of processors for CROCO should be consistent with settings in param.h

To launch is in « batch » mode:

create a job: `job.slurm`

```
#!/bin/bash
#SBATCH --job-name=BENGUELA_CPL_ow
#SBATCH --partition=slims
#SBATCH --ntasks=4
#SBATCH --time=00:30:00

cd ${SLURM_SUBMIT_DIR}
source ../myenv_mypath.sh

echo $PWD

srun --multi-prog app.conf
```

create a file for multiple executable launch: `app.conf`

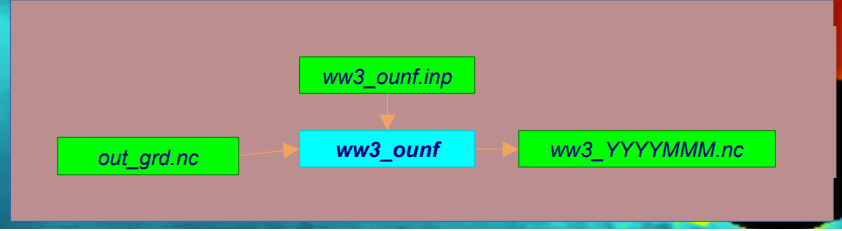
```
0-1 ./crocox
2-3 ./wwatch
```

Then you can launch the job:

```
sbatch job.slurm
```

7. Check outputs and logs

(7) Check the outputs



WW3

WW3 outputs are binaries `out_grd.ww3` that can be converted to netcdf format with the executable `ww3_ounf`.
To do so, first edit the namelist `ww3_ounf.inp` :

```
$ -------------------------------------------------------------------- $
$ WAVEWATCH III Grid output post-processing                             $
$-------------------------------------------------------------------- $
$
$ 1st output time (yyyymmdd hhmmss), increment (s), nb of outputs   -- $
$
  20050101 000000  21600  20050131 000000
$
$ Fields requested ------------------------------------------------- $
$  Defined as namelist:                                            -- $
$      one line set to N                                           -- $
$      next line contains parameter name                           -- $
$
  N
  DPT WND HS FP DP T0M1 LM SPR DIR TWO TAW FOC BHD WLV CUR TUS USS
$  DPT WND HS FP DP T0M1 THM LEV DIR TWO TAW FOC BHD
$  DPT WND HS FP DP
```

Run `ww3_ounf` : `srun ww3_ounf > ww3_ounf.out` → you should now have `ww3.200501.nc`

(7) Check the outputs

Outputs: check log and ouptut files

OASIS

Generated grid files:
- grids.nc
- masks.nc
- areas.nc

Generated grid interpolation files:
- rmp_ww3t_to_ocnt_DISTWGT.nc
- rmp_ocnt_to_ww3t_DISTWGT.nc …

Generated restart files:
- oce.nc, wav.nc
- (overwritten at the end of each simulation)

Logs:
- nout.000000
- crocox.timers_0000, wwatch.timers_0000
- debug.root.01, debug.root.02, …
- debug.notroot.01, debug.notroot.02, …

CROCO

Output files:
- croco_his.nc
- croco_avg.nc
- croco_rst.nc

Logs:
- croco.log
- + eventually standard output redirected to batch log file if LOGFILE cppkey not defined

WW3

Output files:
- out_grd.ww3 => ww3.DATE.nc
- out_pnt.ww3 => ww3.DATE_spec.nc

Logs:
- log.ww3
- output.ww3

Appendices

Tools: the coupling toolbox

Tutorials available here :
https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.16.coupling.html

(1) Get the source codes

On your own computer/cluster:

```
git clone https://github.com/NOAA-EMC/WW3
git clone https://github.com/wrf-croco/WRF.git
git clone --branch v1.3.1 https://gitlab.inria.fr/croco-ocean/croco.git croco-v1.3.1
```

(3) Compilation

(3) Compile first OASIS and then your codes (CROCO, WRF...)

- First compile **OASIS**

in **`$HOME/oasis/oasis3-mct/oasis3-mct/util/make_dir`**

- settings (path, libraries) to edit in **`make.MACHINETYPE`**
- include to do in **`make.inc`** : **`include $(home)/oasis/oasis3-mct/util/make_dir/make.MACHINETYPE`**
- clean and compile:

```
make realclean -f TopMakefileOasis3 > oasis_clean.out
make -f TopMakefileOasis3 > oasis_make.out
```

(3) Compilation

(3) Compile first OASIS and then your codes (CROCO, WW3...)

- Then, compile your models in coupled mode (**with the same netcdf libraries and compilers**)

  * For **WW3**:

    in **`$HOME/WW3`**

    WW3 compilation requests 3 files:

    - switch file: examples for coupling with ocean / atmosphere : **`switch_OASOCM / _OASACM`**

      Mandatory switches for coupling are: **`MPI DIST COU OASIS OASOCM OASACM and CRT0 WNT0`**
    - **`comp.COMPILER`** file, e.g. for Intel comp.Intel
    - **`link.COMPILER`** file : provide useful options and links for compilers

    Clean, setup and compile:

    ```
    ./w3_clean -c
    ./w3_setup .. -c Intel -s OASOCM
    ./w3_automake
    ```

    Or use the script provided in WW3_IN: **`make_WW3_compil`**

Tips in case of error during compilation

In case of strange errors during compilation (e.g. “catastrophic error: could not find …”), try one of these solutions:
• check your home space is not full ;-)
• check your paths to compilers and libraries (especially Netcdf library)
• check that you have the good permissions, and check that your executable files (configure, make…) do are executable • check that your shell scripts headers are correct or add them if necessary (e.g. for bash: #!/bin/bash)
• try to exit/log out the machine, log in back, clean and restart compilation

In case of ‘segmentation fault’ error:
• try to allocate more memory with “unlimited -s unlimited”
• try to launch the compilation as a job (batch) with more allocated memory

Errors and tips related to netcdf library:
• with netcdf 4.3.3.1: need to add the following compilation flag for all models: -mt_mpi
the error associated to a missing -mt_mpi flag is of this type: ” /opt/intel//impi/4.1.1.036/intel64/lib/libmpi_mt.so.4: could not read symbols: Bad value “
• with netcdf 4.1.3: do NOT add -mt_mpi flag
• with netcdf4, need to place hdf5 library path in your environment: export LD_LIBRARY_PATH=YOUR_HDF5_DIR/lib:$LD_LIBRARY_PATH • with netcdf 4, if you use the library splitted in 2: C part and Fortran part, you need to place links to C library before links to Fortran library and need to put both path in this same order in your LD_LIBRARY_PATH