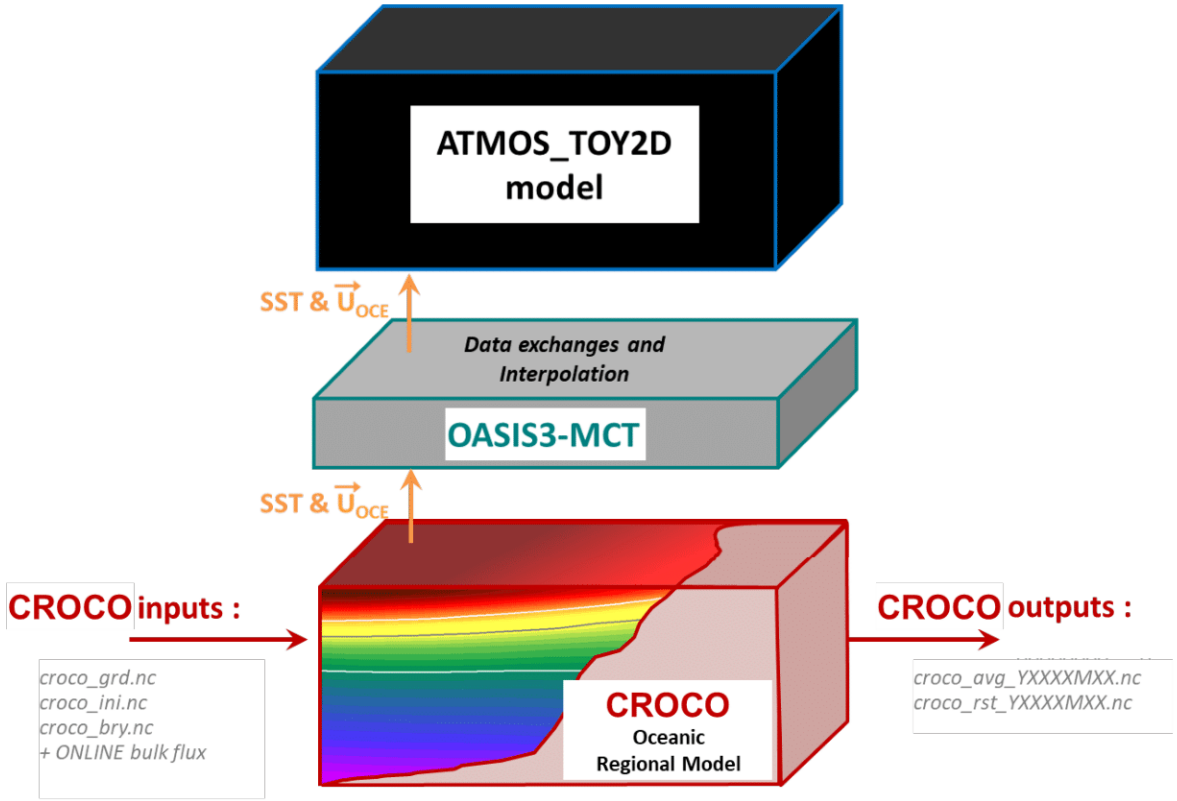
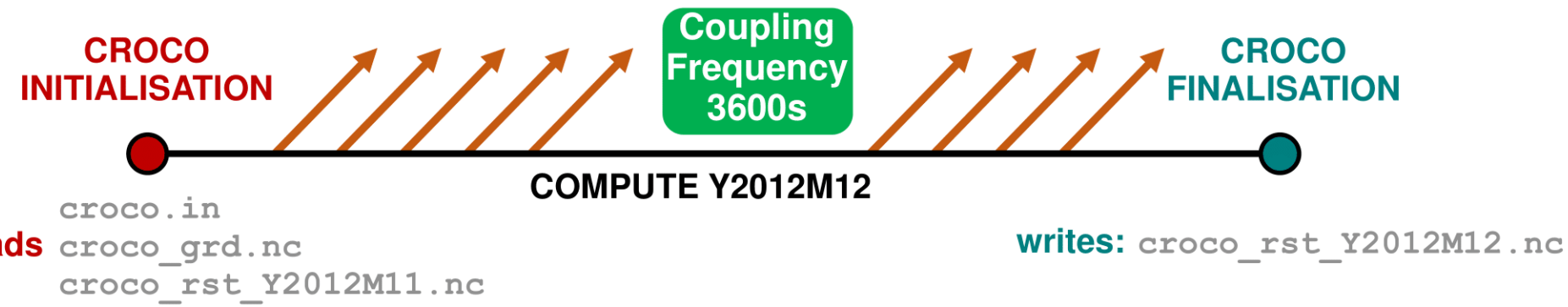
Outlines for the training week

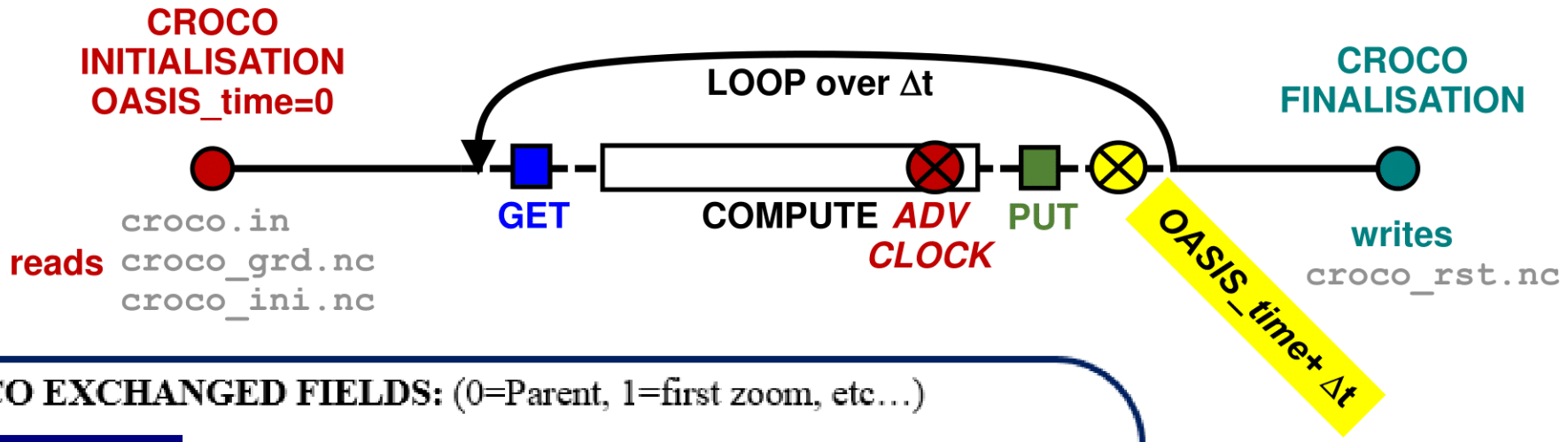
1. Introduction – Presentation of the Model & coupler components
   – Installation of the coupling environment
2. Run an inter-annual **CROCO** simulation (**RUN1**)
3. **Coupling CROCO with an atmospheric TOY model – spatial regridding** (**RUN2**)
4. Run a **WRF** inter-annual simulation (**RUN3**)
5. Coupling **WRF** with an ocean **TOY** model – time transformation (**RUN4**)
6. Coupling **CROCO** with **WRF** : parallel coupling (**RUN5**)

Coupling CROCO with a TOY MODEL





Coupling in CROCO code



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

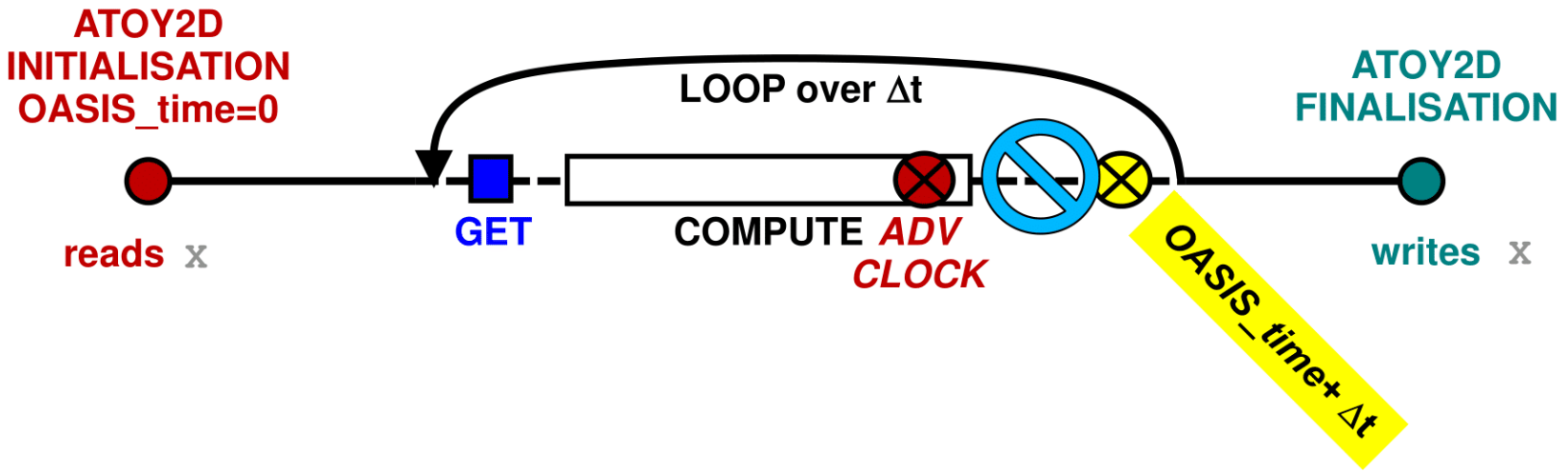
SENT

- **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**

Coupling in ATOY2D code



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

Coupling CROCO with a TOY MODEL

- **STEP1:** Compile OASIS3-MCT coupler library
- **STEP2:** Compile CROCO in a coupled context
- **STEP3:** Compile the atmospheric TOY model
- **STEP4:** Prepare all the input files for CROCO and OASIS
- **STEP5:** Run CROCO/ATOY with `RUN2_croco_inter_atoy.pbs`
- **STEP6:** Visualize outputs in `OUTPUT_FILES/RUN2`

STEP1-3: Compilation

❶ Compile OASIS3-MCT coupler library:

- This is done in **OASIS/oasis3-mct/util/make_dir**
- Execute the command **make -f TopMakefineOasis3**

❷ Compile CROCO in a semi-coupled context in **WORK_MCC/Run**

- **cppdefs.h**: Activate **#define OA_COUPLING**
  and **#define OA_COUPLING_CPLMASKS**
- Compile with **jobcomp_legau** and rename the executable **croco run2**

❸ Compile ATOY component in **TOY_MODELS/ATMOS_TOY2D**

- **atmos_toy2d.F**: Adjust **spatial grid size**
- Compile with **jobcomp_legau** to create the executable **atoy2d**

STEP1-3: Compilation

Nov 21, 24 10:51 **cppdefs.h** Page 1/3

```
!======================================================================
! CROCO is a branch of ROMS developped at IRD, INRIA,
! Ifremer, CNRS and Univ. Toulouse III  in France
! The two other branches from UCLA (Shchepetkin et al)
! and Rutgers University (Arango et al) are under MIT/X style license.
! CROCO specific routines (nesting) are under CeCILL-C license.
!
! CROCO website : http://www.croco-ocean.org
!======================================================================
!
/*
   This is "cppdefs.h": MODEL CONFIGURATION FILE
   ==== == =========== ===== ============ ====

        SELECT ACADEMIC TEST CASES
*/
#undef  BASIN             /* Basin Example */
#undef  CANYON            /* Canyon Example */
...
#define REGIONAL          /* REGIONAL Applications */

#if defined REGIONAL
/*
!======================================================================
!               REGIONAL (realistic) Configurations
!======================================================================
!
!----------------------
! BASIC OPTIONS
!----------------------
*/
                          /* Configuration Name */
# define MCC
                          /* 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
# ifdef OA_COUPLING
#  undef OA_COUPLING_CPLMASKS
# endif
# undef  OW_COUPLING
...
                          /* Wave-current interactions */
# undef  MRL_WCI
                          /* Open Boundary Conditions */
# undef  TIDES
# define OBC_EAST
# define OBC_WEST
# define OBC_NORTH
# define OBC_SOUTH
                          /* Applications */
# undef  BIOLOGY
# undef  FLOATS
# undef  STATIONS
# undef  PASSIVE_TRACER
# undef  SEDIMENT
# undef  MUSTANG
# undef  BBL
                          /* I/O server */
# undef  XIOS
                          /* Calendar */
# undef  USE_CALENDAR
                          /* dedicated croco.log file */
# undef  LOGFILE
/*
!------------------------------------------------
! PRE-SELECTED OPTIONS
!
! ADVANCED OPTIONS ARE IN CPPDEFS_DEV.H
!------------------------------------------------
*/
                          /* Parallelization */
# ifdef MPI
#  undef  PARALLEL_FILES
#  undef  NC4PAR
#  undef  MPI_NOLAND
```

Nov 21, 24 10:51 **cppdefs.h** Page 2/3

```
#  undef  MPI_TIME
# endif
# undef  AUTOTILING
                          /* Non-hydrostatic options */
# ifdef NBQ
#  define W_HADV_TVD
#  define W_VADV_TVD
# endif
                          /* Grid configuration */
# define CURVGRID
# define SPHERICAL
# define MASKING
# undef  WET_DRY
# define NEW_S_COORD
                          /* Model dynamics */
# define SOLVE3D
# define UV_COR
# define UV_ADV
                          /* Equation of State */
# define SALINITY
# define NONLIN_EOS
                          /* Surface Forcing */
/*
! Bulk flux algorithms (options)
! by default : COARE3p0 paramet with GUSTINESS effects
!
!Â 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
*/
# undef  ABL1D
# ifdef  ABL1D
#  define BULK_FLUX
#  undef  ANA_ABL_LSDATA
#  undef  ANA_ABL_VGRID
#  define STRESS_AT_RHO_POINTS
#  define ABL_NUDGING
#  define ABL_NUDGING_DYN
#  define ABL_NUDGING_TRA
#  undef  ABL_DYN_RESTORE_EQ
#  undef  SFLUX_CFB
# else
#  define BULK_FLUX
# endif
# ifdef BULK_FLUX
#  undef  BULK_ECUMEV0
#  undef  BULK_ECUMEV6
#  undef  BULK_WASP
#  define BULK_GUSTINESS
#  define BULK_LW
#  undef  SST_SKIN
#  undef  ANA_DIURNAL_SW
#  undef  ONLINE
#  ifdef ONLINE
#   undef  AROME
#   undef  ERA_ECMWF
#  endif
#  undef READ_PATM
#  ifdef READ_PATM
#   define OBC_PATM
#  endif
# else
#  define QCORRECTION
#  define SFLX_CORR
#  undef  SFLX_CORR_COEF
#  define ANA_DIURNAL_SW
# endif
# undef  SFLUX_CFB
# undef  SEA_ICE_NOFLUX
                          /* Lateral Forcing */
# undef CLIMATOLOGY
# ifdef CLIMATOLOGY
...
# endif

# define FRC_BRY
# ifdef FRC_BRY
...
# endif
                          /* Lateral Momentum Advection (default UP3) */
```

Nov 21, 24 10:51 **cppdefs.h** Page 3/3

```
# define UV_HADV_UP3
# undef  UV_HADV_UP5
# undef  UV_HADV_WENO5
# undef  UV_HADV_TVD
                          /* Lateral Explicit Momentum Mixing */
# undef  UV_VIS2
# ifdef UV_VIS2
#  define UV_VIS_SMAGO
# endif
                          /* Vertical Momentum Advection */
# define UV_VADV_SPLINES
# undef  UV_VADV_WENO5
# undef  UV_VADV_TVD
                          /* Lateral Tracer Advection (default UP3) */
...
                          /* Lateral Explicit Tracer Mixing */
...
                          /* Vertical Tracer Advection  */
...
                          /* Sponge layers for UV and TS */
# define SPONGE
                          /* Semi-implicit Vertical Tracer/Mom Advection */
# undef  VADV_ADAPT_IMP
                          /* Bottom friction in fast 3D step */
# define LIMIT_BSTRESS
# undef  BSTRESS_FAST
                          /* Vertical Mixing */
...
                          /* Wave-current interactions */
# ifdef OW_COUPLING
...
# endif
                          /* Bottom Forcing */
...
                          /* Point Sources - Rivers */
...
                          /* Open Boundary Conditions */
...
                          /* Input/Output */
...
                         /* Exact restart */
# undef EXACT_RESTART
                          /* Parallel reproducibility or restartabilty test
 */
...
/*
!                          Diagnostics
!------------------------------------------
! 3D Tracer & momentum balance
! 2D Mixing layer balance
! Depth-mean vorticity and energy balance
! Eddy terms
!------------------------------------------
...
/*
!           Applications:
!-----------------------------------
! Biology, floats, Stations,
! Passive tracer, Sediments, BBL
!-----------------------------------
!
*/

#elif defined COASTAL
/*
!======================================================================
!                 COASTAL (realistic) Configurations
!======================================================================
...
/*
!
!=====================================================
!                IDEALIZED CONFIGURATIONS
!=====================================================
!
*/
#elif defined BASIN
...

#endif /* END OF CONFIGURATION CHOICE */

#include "cppdefs_dev.h"
#include "set_global_definitions.h"
```

STEP4: Prepare CROCO and OASIS inputs

- CROCO input files have already been prepared for **RUN2** (**Y2012M12**) !
- Let's prepare OASIS auxiliary input files:
  - ➥ This is done in **`WORK_MCC/SCRIPT`**

❶ Edit **`STEP3_make_oasis_files.scr`** and adjust:

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

❷ Execute **`STEP3_make_oasis_files.scr`**

STEP4: Prepare CROCO and OASIS inputs

- OASIS auxiliary input files:

```
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)

STEP5: Create OASIS *namcouple* input file

- OASIS *namcouple* file in **../INPUT_FILES/OASIS_FILES**

namcouple is a text file :

- the keywords can appear in any order
- all lines beginning with # are comments
- blank lines not allowed.

```
###################################################################
#
#       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
 NUMSECS
#
###################################################################
#
# NLOGPRT: printing level in output file cplout:
#   0/1/2 = no printing/intermediate printing/complete output
#
 $NLOGPRT
  2
#
###################################################################
 $STRINGS
###################################################################
#                        OCEAN  --->>>  ATMOS
#                        --------------------
###################################################################
# Field 1.1 : Sea surface temperature (o->a 1) from CROCO Parent to
ATOY2D
#
CROCO_SST ATOYSST0 1 86400 1 Rcroco0.nc EXPOUT
41 42 41 42 crn0 crp0 LAG=0
R 0 R 0
SCRIPR
BILINEAR LR SCALAR LATLON 1
###################################################################
```

STEP5: Create OASIS *namcouple* input file

➤ OASIS *namcouple* file in **../INPUT_FILES/OASIS_FILES**

```
#########################################################
$NFIELDS
 1
$RUNTIME
 86400
$NLOGPRT
  2
#########################################################
$STRINGS
# Field 1.1 : Sea surface temperature (o->a 1) from CROCO Parent to ATOY2D
CROCO_SST ATOYSST0 1 86400 1 Rcroco0.nc EXPOUT
41 42 41 42 crn0 crp0 LAG=0
R 0 R 0
SCRIPR
BILINEAR LR SCALAR LATLON 1
#########################################################
```

Number of Fields Exchanged

Run Length (in seconds)

Debug Level (-1-0-1-2-5-10-12-15-20-30 0-1-2-3)

List of Coupling Interactions
- field names
- coupling periods
- grid names
- coupling lags
- transforms

First section of the *namcouple* :

- Only 3 keywords are really used by OASIS3-MCT_3.0
  - ⇨ **$NFIELDS**: number of coupling fields to be exchanged
  - ⇨ **$RUNTIME**: total simulated time for the run (in seconds)
  - ⇨ **$NLOGPRT**: amount of debug and time statistic information

STEP5: Create OASIS *namcouple* input file

➤ OASIS *namcouple* file in **../INPUT_FILES/OASIS_FILES**

```
#########################################################
$NFIELDS
 1
$RUNTIME
 86400
$NLOGPRT
  2
#########################################################
$STRINGS
# Field 1.1 : Sea surface temperature (o->a 1) from CROCO Parent to ATOY2D
CROCO_SST ATOYSST0 1 3600 3 Rcroco0.nc EXPOUT
41 42 41 42 crn0 crp0 LAG=0
R 0 R 0
CHECKIN SCRIPR CHECKOUT
INT=1
CONSERV LR SCALAR LATLON 1 FRACNNEI FIRST
INT=1
#
# Field 1.2 : Zonal Surface currents (o->a 1) from CROCO Parent to ATOY2D
CROCO_UOCE ATOYUOC0 1 3600 1 Rcroco.nc EXPORTED
87 107 87 107 cun0 crp0 LAG=0
R 0 R 0
#
SCRIPR
#
BILINEAR LR SCALAR LATLON 1
#########################################################
```

Number of Fields Exchanged

Run Length (in seconds)

Debug Level (-1-0-1-2-etc)

List of Coupling Interactions
- field names
- coupling periods
- grid names
- coupling lags
- transforms

Possible transformations:
- time transformations: **LOCTRANS**
- pre-proc: **CHECKIN**, **BLASOLD**
- remapping: **MAPPING** or **SCRIPR**
- post-proc: **CONSERV**, **BLASNEW**, **CHECKOUT**

Second section of the *namcouple* :

↳ After the keyword **$STRINGS**: coupling information for each coupling (or I/O) field.

⇨ The format depends on the field status given by the last entry on the field first line (**EXPORTED**, **EXPOUT**, **INPUT**, **OUTPUT**).

⇨ For **EXPORTED** and **EXPOUT** 3 lines are expected…

⇨ The 4th line lists the transformation to be performed

⇨ The following lines describe the parameter for each transformation.

STEP5: Create OASIS *namcouple* input file

➢ OASIS *namcouple* file in

../INPUT_FILES/OASIS_FILES

Possible transformations:
- time transformations: **LOCTRANS**
- pre-proc: **CHECKIN**, **BLASOLD**
- remapping: **MAPPING** or **SCRIPR**
- post-proc: **CONSERV**, **BLASNEW**, **CHECKOUT**

⇨ The **MAPPING** keyword is used to specify the NetCDF input file to be read and used for mapping. It contain weights and addresses, and must follow the SCRIPR format.

⇨ The **SCRIPR** keyword allow to use the interpolation techniques offered by SCRIPR library (1.4):

DISTWGT
GAUSSWGT
BILINEAR
BICUBIC
CONSERV

⇨ **NB**: OASIS3-MCT supports multiple fields coupling via single communication (see user's guide)

```
#############################################################
$NFIELDS
 1
$RUNTIME
 86400
$NLOGPRT
  2
#############################################################
$STRINGS
# Field 1.1 : Sea surface temperature (o->a 1) from CROCO Parent to
CROCO_SST ATOYSST0 1 3600 3 Rcroco0.nc EXPOUT
41 42 41 42 crn0 crp0 LAG=0
R 0 R 0
CHECKIN SCRIPR CHECKOUT
INT=1
CONSERV LR SCALAR LATLON 1 FRACNNEI FIRST
INT=1
#
# Field 1.2 : Zonal Surface currents (o->a 1) from CROCO Parent to A
CROCO_UOCE ATOYUOC0 1 3600 1 Rcroco.nc EXPORTED
41 42 41 42 cun0 crp0 LAG=0
R 0 R 0
SCRIPR
#
BILINEAR LR SCALAR LATLON 1
#
# Field 1.3 : Meridional Surface currents (o->a 1) from CROCO Parent
CROCO_VOCE ATOYVOC0 1 3600 1 Rcroco.nc EXPORTED
41 42 41 42 cvn0 crp0 LAG=0
R 0 R 0
SCRIPR
#
DISTWGT LR SCALAR LATLON 1 4
#############################################################
```

STEP5: Run CROCO/ATOY on Lengau

- This is done in **`WORK_MCC/SCRIPT`**
  1. Edit **`RUN2_croco_inter.pbs`** script to compile CROCO and adjust:

     email, procs, paths, CROCO executable name, NB_procs, DT

     start/end of the simulation [2012 12 →12]
  2. Launch **RUN2** using the command **`qsub`**
- Check for possible bugs:
  1. Edit the output and error file from the PBS job: **`job.o* job.e*`**
  2. Edit the CROCO log file: **`croco_Y2012M10.out`**

STEP5: Run CROCO/ATOY on Lengau

Nov 21, 24 11:06 **RUN2_croco_inter_atoy.pbs** Page 1/4

```
#!/bin/bash
#PBS -l select=1:ncpus=24:mpiprocs=4
#PBS -P WCHPC
#PBS -q serial
#PBS -l walltime=2:00:00
#PBS -m abe
#PBS -M put.your@email.here
cd /home/$USER/lustre/WORK_MCC/SCRIPTS
#
#####################################################
#  Define files and run parameters
#####################################################
#
MODEL=croco                                 # Name used for the input files. For example croco_grd.nc
SCRATCHDIR=`pwd`/../OUTPUT_FILES/RUN2       # Scratch directory where the model is run
INPUTDIR=`pwd`/../Run                       # Input directory where the croco_inter.in input file is located
AGRIF_FILE=AGRIF_FixedGrids.in              # AGRIF input file which defines the position of child grids
MSSDIR=`pwd`/../INPUT_FILES/CROCO_FILES     # Directory where the croco input NetCDF files (croco_grd.nc, ...) are stored
MSSOUT=$MSSDIR                              # Directory where the croco output and restart NetCDF files (croco_his.nc, ...) are
 stored
CODFILE=croco_run2                          # CROCO executable
NBPROCS_CROCO=4                             # number of processors for MPI run
#
OASIS_INPUT_DIR=`pwd`/../INPUT_FILES/OASIS_FILES
#
ATOY2DEXE_PATH=`pwd`/../../TOY_MODELS/ATMOS_TOY2D # Input directory where the ATOY2D executable file is located
ATOY2DEXE=atoy2d                            # ATOY2D executable
NBPROCS_ATOY2D=1                            # number of processors for MPI run
#
# command for running the mode : ./ for sequential job, mpirun -np NBPROCS for mpi run
# WARNING: for mpi run command, it is needed to add a space at the end!
#RUNCMD='./'
RUNCMD="mpirun "
#RUNCMD="$MPI_LAUNCH "
#RUNCMD="srun "

#  Define environment variables for OPENMP
...

# Define which type of inputs are used
BULK_FILES=1
FORCING_FILES=0
CLIMATOLOGY_FILES=0
BOUNDARY_FILES=1
RUNOFF_FILES=0

# Atmospheric surface forcing dataset used for the bulk formula (NCEP)
ATMOS_BULK=ERA5
# Atmospheric surface forcing dataset used for the wind stress (NCEP, QSCAT)
ATMOS_FRC=QSCAT
# Oceanic boundary and initial dataset (SODA, ECCO,...)
OGCM=SODA
# Runoff dataset (Dai and Trenberth,...)
RUNOFF_DAT=DAI

# Model time step [seconds]
DT=3600
# Number of barotropic time steps within one baroclinic time step [number], NDTFAST in croco.in
NFAST=60

# Number total of grid levels (1: No child grid)
NLEVEL=1
# AGRIF nesting refinement coefficient
AGRIF_REF=3

# Start and End year
NY_START=2005
NY_END=2005
# Start and End month
NM_START=1
NM_END=3
# Set month format at 1 or 2 digits (for input and output files): "%01d" = 1 digit/ "%02d" = 2 digit
MTH_FORMAT="%02d"
#  Time Schedule  -  TIME_SCHED=0 --> yearly files
#                    TIME_SCHED=1 --> monthly files
TIME_SCHED=1

# Number of year that are considered to be part of the spin-up (i.e. 365 days per year)
NY_SPIN=0

# Output frequency [days]
#  average
ND_AVG=3
#   history (if = -1 set equal to NUMTIMES)
ND_HIS=-1
#   restart (if = -1 set equal to NUMTIMES)
ND_RST=-1

#  Restart file - RSTFLAG=0 --> No Restart
#                 RSTFLAG=1 --> Restart
RSTFLAG=0
#  Exact restart - EXACT_RST=0 --> Exact restart OFF
#                  EXACT_RST=1 --> Exact restart ON
EXACT_RST=0

#unalias cp
#unalias mv
#limit coredumpsize unlimited
CP=/bin/cp
MV=/bin/mv
```

Nov 21, 24 11:06 **RUN2_croco_inter_atoy.pbs** Page 2/4

```
LN=/bin/ln
#
#####################################################
#  END USER CHANGE
#####################################################
#
if [ ! -e ${SCRATCHDIR} ]; then mkdir -p ${SCRATCHDIR}; fi
echo "Getting ${ATOY2DEXE} from ${ATOY2DEXE_PATH}"
cp ${ATOY2DEXE_PATH}/${ATOY2DEXE} $SCRATCHDIR
#
echo "Getting grids.nc,masks.nc,areas.nc,coupling_masks.nc, and namcouple from ${OASIS_INPUT_DIR}"
cp ${OASIS_INPUT_DIR}/grids.nc $SCRATCHDIR
cp ${OASIS_INPUT_DIR}/masks.nc $SCRATCHDIR
cp ${OASIS_INPUT_DIR}/areas.nc $SCRATCHDIR
cp ${OASIS_INPUT_DIR}/coupling_masks_zeros.nc $SCRATCHDIR/coupling_masks.nc
cp ${OASIS_INPUT_DIR}/namcouple_CROCO_ATOY2D $SCRATCHDIR/namcouple.generic
#
if [[ $TIME_SCHED == 0 ]]; then
  NM_START=1979
  NM_END=1979
fi
#
# netcdf file prefixes
#
GRDFILE=${MODEL}_grd
FRCFILE=${MODEL}_frc
BLKFILE=${MODEL}_blk
INIFILE=${MODEL}_ini
CLMFILE=${MODEL}_clm
BRYFILE=${MODEL}_bry
RNFFILE=${MODEL}_runoff
#
if [ ! -e $MSSOUT ] ; then
 mkdir $MSSOUT
fi
#
if [[ $RSTFLAG != 0 ]]; then
  NY=$NY_START
  NM=$NM_START
  if [[ $TIME_SCHED == 0 ]]; then
    NY=$((NY - 1))
    TIME=Y${NY}
  else
    NM=$((NM - 1))
    if [[ $NM == 0 ]]; then
      NM=12
      NY=$((NY - 1))
    fi
    TIME=Y${NY}M$( printf ${MTH_FORMAT} ${NM})
  fi
  RSTFILE=${MODEL}_rst_${TIME}
fi
#
if [[ $TIME_SCHED == 0 ]]; then
  TIME=Y${NY_START}
else
  TIME=Y${NY_START}M$( printf ${MTH_FORMAT} ${NM_START})
fi
#
# Get the code
#
if [ ! -e $SCRATCHDIR ] ; then
 mkdir $SCRATCHDIR
fi
cd $SCRATCHDIR
echo "Getting $CODFILE from $INPUTDIR"
$CP -f $INPUTDIR/$CODFILE $SCRATCHDIR
chmod u+x $CODFILE
echo "Getting $AGRIF_FILE from $INPUTDIR"
$CP -f $INPUTDIR/$AGRIF_FILE $SCRATCHDIR
#
# Get the netcdf files
#
LEVEL=0
while [ $LEVEL != $NLEVEL ]; do
  if [[ ${LEVEL} == 0 ]]; then
    ENDF=
  else
    ENDF=.${LEVEL}
  fi
  echo "Getting ${GRDFILE}.nc${ENDF} from $MSSDIR"
  $LN -sf $MSSDIR/${GRDFILE}.nc${ENDF} $SCRATCHDIR
  echo "Getting ${MODEL}_inter.in${ENDF} from $INPUTDIR"
  $CP -f $INPUTDIR/${MODEL}_inter.in${ENDF} $SCRATCHDIR
  if [[ $RSTFLAG == 0 ]]; then
    echo "Getting ${INIFILE}_${OGCM}_${TIME}.nc${ENDF} from $MSSDIR"
    $CP -f $MSSDIR/${INIFILE}_${OGCM}_${TIME}.nc${ENDF} $SCRATCHDIR
    $CP -f ${INIFILE}_${OGCM}_${TIME}.nc${ENDF} ${INIFILE}.nc${ENDF}
  else
    echo "Getting ${RSTFILE}.nc${ENDF} from $MSSOUT"
    $CP -f $MSSOUT/${RSTFILE}.nc${ENDF} $SCRATCHDIR
    $CP -f ${RSTFILE}.nc${ENDF} ${INIFILE}.nc${ENDF}
  fi
  LEVEL=$((LEVEL + 1))
done
#####################################################
#  Compute
#####################################################
#
NY_END=$((NY_END + 1))
```

STEP5: Run CROCO/ATOY on Lengau

Nov 21, 24 11:06 **RUN2_croco_inter_atoy.pbs** Page 3/4

```
NM_END=$((NM_END + 1))
NY=$NY_START
while [ $NY != $NY_END ]; do
  if [[ $NY == $NY_START ]]; then
    NM=$NM_START
  else
    NM=1
  fi
  MY_YEAR=$NY
  MY_YEAR=$((MY_YEAR + 1))
  if [[ $MY_YEAR == $NY_END ]]; then
    MONTH_END=$NM_END
  else
    MONTH_END=13
  fi
  if [[ $TIME_SCHED == 0 ]]; then
    MONTH_END=2
  fi
  while [ $NM != $MONTH_END ]; do
    if [[ $TIME_SCHED == 0 ]]; then
      TIME=Y${NY}
      echo "Computing YEAR $NY"
    else
      TIME=Y${NY}M$( printf ${MTH_FORMAT} ${NM})
      echo "Computing YEAR $NY MONTH $( printf ${MTH_FORMAT} ${NM})"
    fi
#
# Get forcing and clim for this time
#
    LEVEL=0
    while [ $LEVEL != $NLEVEL ]; do
      if [[ ${LEVEL} == 0 ]]; then
        ENDF=
      else
        ENDF=.${LEVEL}
      fi
      if [[ ${FORCING_FILES} == 1 ]]; then
        echo "Getting ${FRCFILE}_${ATMOS_FRC}_${TIME}.nc${ENDF} from $MSSDIR"
        $LN -sf $MSSDIR/${FRCFILE}_${ATMOS_FRC}_${TIME}.nc${ENDF} ${FRCFILE}.nc${ENDF}
      fi
      if [[ ${BULK_FILES} == 1 ]]; then
        echo "Getting ${BLKFILE}_${ATMOS_BULK}_${TIME}.nc${ENDF} from $MSSDIR"
        $LN -sf $MSSDIR/${BLKFILE}_${ATMOS_BULK}_${TIME}.nc${ENDF} ${BLKFILE}.nc${ENDF}
      fi
     if [[ ${RUNOFF_FILES} == 1 ]]; then
        echo "Getting ${RNFFILE}.nc${ENDF} from $MSSDIR"
        $LN -sf $MSSDIR/${RNFFILE}.nc${ENDF} ${RNFFILE}.nc${ENDF}
      fi

      LEVEL=$((LEVEL + 1))
    done
#
# No child climatology or boundary files
#
    if [[ ${CLIMATOLOGY_FILES} == 1 ]]; then
      echo "Getting ${CLMFILE}_${OGCM}_${TIME}.nc from $MSSDIR"
      $LN -sf $MSSDIR/${CLMFILE}_${OGCM}_${TIME}.nc ${CLMFILE}.nc
    fi
    if [[ ${BOUNDARY_FILES} == 1 ]]; then
      echo "Getting ${BRYFILE}_${OGCM}_${TIME}.nc from $MSSDIR"
      $LN -sf $MSSDIR/${BRYFILE}_${OGCM}_${TIME}.nc ${BRYFILE}.nc
    fi
#
# Set the number of time steps for each month
# (30 or 31 days + 28 or 29 days for february)
#

...
...
...

    #
    # Put the number of time steps in the .in files
    #
    echo "YEAR = $NY MONTH = $NM DAYS = $NDAYS DT = $DT NTIMES = $NUMTIMES"
    NUMSECS=$((NDAYS * 24 * 3600))
    NUMTIMES=$((NUMSECS / DT))

    DT0=$DT
    LEVEL=0
    while [[ $LEVEL != $NLEVEL ]]; do
      if [[ ${LEVEL} == 0 ]]; then
        ENDF=
      else
        ENDF=.${LEVEL}
        NUMTIMES=$((AGRIF_REF * NUMTIMES))
        DT=$((DT / AGRIF_REF))
      fi
      NUMAVG=$((ND_AVG * 86400 / DT ))
      if [[ ${ND_HIS} -ne -1 ]]; then
        NUMHIS=$((ND_HIS * 86400 / DT ))
      else
        NUMHIS=$NUMTIMES
      fi
      if [[ ${ND_RST} -ne -1 ]]; then
        NUMRST=$((ND_RST * 86400 / DT ))
      else
        NUMRST=$NUMTIMES
      fi
```

Nov 21, 24 11:06 **RUN2_croco_inter_atoy.pbs** Page 4/4

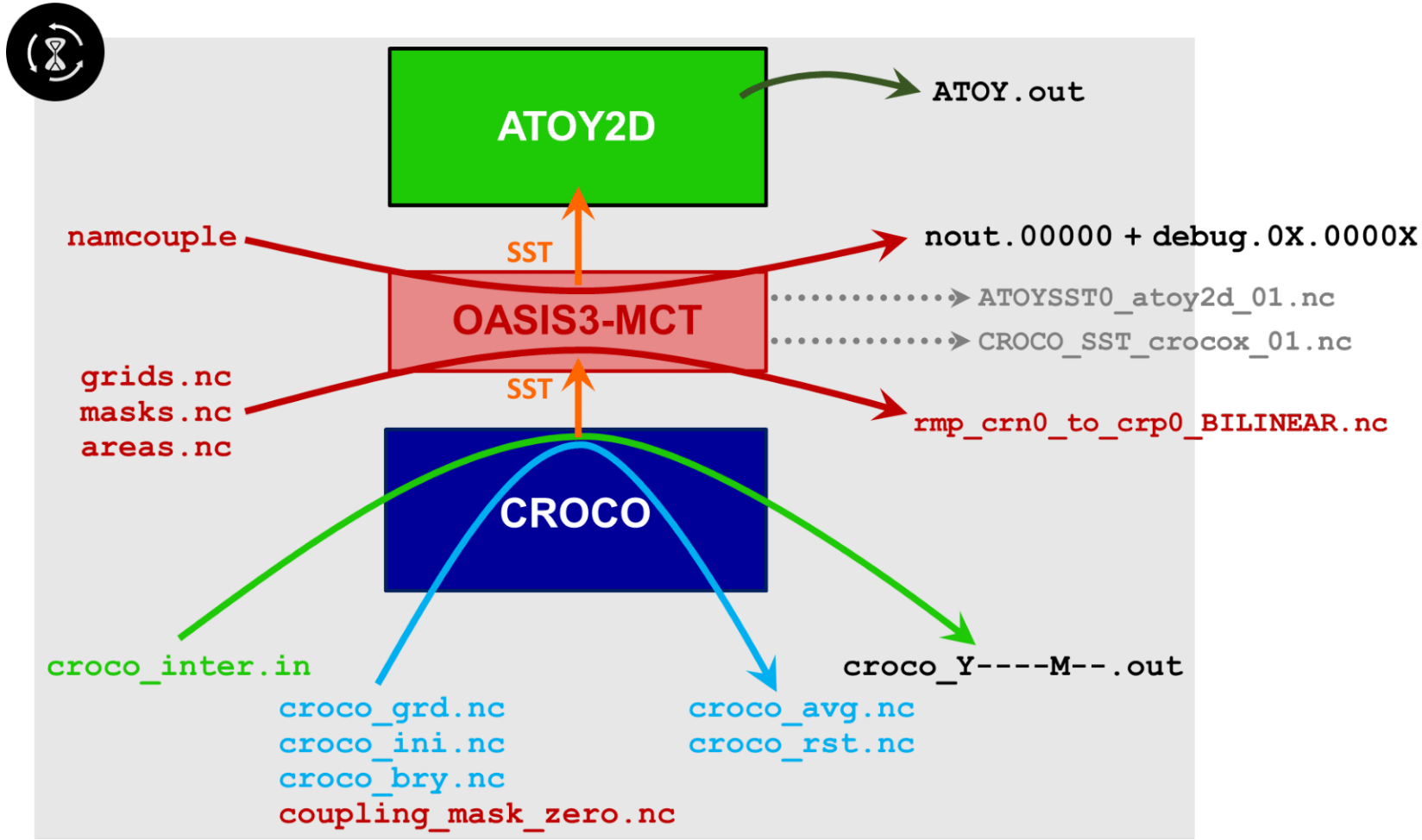
```
      if [[ $EXACT_RST == 1 ]]; then
        echo "Exact restart defined"
        if [[ $NY == $NY_START && $NM == $NM_START ]]; then
          NUMRECINI=1
          echo "set NUMRECINI = $NUMRECINI"
        else
          NUMRECINI=2
          echo "set NUMRECINI = $NUMRECINI"
        fi
      else  # no exact restart
        echo "No exact restart"
        NUMRECINI=1
        echo "set NUMRECINI = $NUMRECINI"
      fi

      echo " "
      echo "Writing in ${MODEL}_inter.in${ENDF}"
      echo "USING DT      = $DT"
      echo "USING NFAST   = $NFAST"
      echo "USING NUMTIMES = $NUMTIMES"
      echo "USING NUMAVG  = $NUMAVG"
      echo "USING NUMHIS  = $NUMHIS"
      echo "USING NUMRST  = $NUMRST"
      echo "USING NUMRECINI = $NUMRECINI"

      if [ ! -f ${MODEL}_inter.in${ENDF} ]; then
        echo "=="
        echo "=> ERROR : miss the ${MODEL}_inter.in${ENDF} file"
        echo "=="
        exit 1
      fi
      sed -e 's/NUMTIMES/'$NUMTIMES'/' -e 's/TIMESTEP/'$DT'/' -e 's/NFAST/'$NFAST'/' \
          -e 's/\bNUMAVG\b/'$NUMAVG'/' -e 's/\bNUMHIS\b/'$NUMHIS'/' -e 's/\bNUMRST\b/'$NUMRST'/' \
          -e 's/NUMRECINI/'$NUMRECINI'/' \
          -e 's/NYONLINE/'$NY'/' -e 's/NMONLINE/'$NM'/' < ${MODEL}_inter.in${ENDF} > ${MODEL}_${TIME}_inter.in${ENDF}
      $CP -f ${MODEL}_${TIME}_inter.in${ENDF} croco.in${ENDF}

      LEVEL=$((LEVEL + 1))
    done
    DT=$DT0
    #
    sed -e 's/NUMSECS/'$NUMSECS'/g' namcouple.generic > namcouple
    #
    #  COMPUTE
    #
    echo " "
    echo "Computing for $TIME"
    ${RUNCMD} -np ${NBPROCS_CROCO} ./${CODFILE} : -np ${NBPROCS_ATOY2D} ./${ATOY2DEXE} > ${MODEL}_${TIME}.out
    date
    #
    # Test if the month has finised properly
    echo "Test ${MODEL}_${TIME}.out"
    status=`tail -2 ${MODEL}_${TIME}.out | grep DONE | wc -l`
    if [[ $status == 1 ]]; then
      echo "All good"
      echo "XXXX${MYTEST}XXXX"
    else
      echo
      echo "Warning: month not finished properly"
      echo
      tail -20 ${MODEL}_${TIME}.out
      echo
      echo "Month ${TIME} did not work"
      echo
      exit 1
    fi
#
#  Archive
#
    LEVEL=0
    while [[ $LEVEL != $NLEVEL ]]; do
      if [[ ${LEVEL} == 0 ]]; then
        ENDF=
      else
        ENDF=.${LEVEL}
      fi
          $CP -f ${MODEL}_rst.nc${ENDF} ${INIFILE}.nc${ENDF}
          $MV -f ${MODEL}_his.nc${ENDF} ${MSSOUT}/${MODEL}_his_${TIME}.nc${ENDF}
          $MV -f ${MODEL}_rst.nc${ENDF} ${MSSOUT}/${MODEL}_rst_${TIME}.nc${ENDF}
          $MV -f ${MODEL}_avg.nc${ENDF} ${MSSOUT}/${MODEL}_avg_${TIME}.nc${ENDF}
      LEVEL=$((LEVEL + 1))
    done
    NM=$((NM + 1))
  done
  NY=$((NY + 1))
done
#
##########################################################
```

STEP5: Run CROCO/ATOY on Lengau

RUN2_croco_inter.pbs: ⟶ job.o-------
job.e-------

❶ Copy executables + croco.in.template + OASIS files

❷ Loop over time : copy monthly input files, adjust croco.in, run CROCO:



STEP6: Visualizing the model outputs

- CROCO ouputs from **RUN2** are in `OUTPUT_FILES/RUN2`
- Verify that the final restart file has been written, copy it in `../RUN5`
- To visualize the outputs:
  - you can use ncview in `OUTPUT_FILES/RUN2`
  - you can use **croco_gui** in **WORK/Run**

STEP 7: Exiting

- Exit Matlab:

```
exit
```

- Give back the compute node:

```
exit
```

NODES

- Logoff the Lengau cluster:

```
exit
```