CROCO – Training Barcelonette 2023

Setup a CROCO simulation

Coastal and Regional Ocean COmmunity model

Outline

- Architecture reminder
- Setup architecture reminder
- Setup the BENGUELA_LR (default) configuration
- Input file pre-processing
- Compilation
- Setup multi-month simulation

Architecture reminder

Suggested work architecture:

$HOME/CROCO
- croco
- croco_tools

Source codes

$HOME/OASIS
- oasis3-mct

$HOME/WRF
- WRF
- WPS

$HOME/DATA
- CROCO_DATASETS
- CFSR_GRIB2
- others …

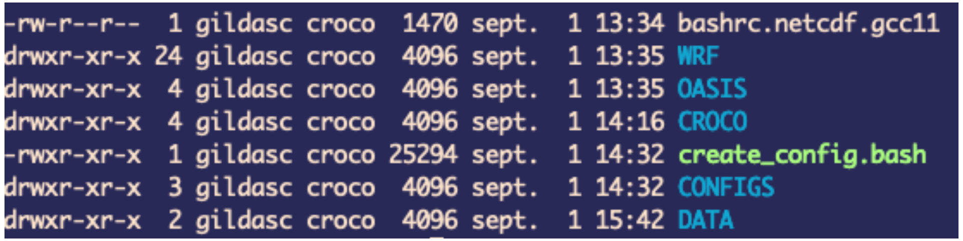
Datasets from global reanalyses

$HOME/CONFIGS
- BENGUELA_LR

Your model configurations

Setup architecture

In /home/login/



```
-rw-r--r--  1 gildasc croco  1470 sept.  1 13:34 bashrc.netcdf.gcc11
drwxr-xr-x 24 gildasc croco  4096 sept.  1 13:35 WRF
drwxr-xr-x  4 gildasc croco  4096 sept.  1 13:35 OASIS
drwxr-xr-x  4 gildasc croco  4096 sept.  1 14:16 CROCO
-rwxr-xr-x  1 gildasc croco 25294 sept.  1 14:32 create_config.bash
drwxr-xr-x  3 gildasc croco  4096 sept.  1 14:32 CONFIGS
drwxr-xr-x  2 gildasc croco  4096 sept.  1 15:42 DATA
```

ssh -X login@croco

```
cp /home/COMMONDATA/bashrc.netcdf.gcc11 .
cp -r /home/COMMONDATA/codes/CROCO/   .
[ cp -r /home/COMMONDATA/codes/WRF .    #If coupling ]
[ cp -r /home/COMMONDATA/codes/OASIS . #If coupling ]

mkdir DATA

cp -Rf /home/COMMONDATA/data_tutos/CROCO_FILES DATA/
[ln -s /home/COMMONDATA/data_tutos/WRF_FILES DATA/    #If coupling ]
[ln -s /home/COMMONDATA/data_tutos/TOY_FILES  DATA/   #If coupling ]

mkdir CONFIGS
```

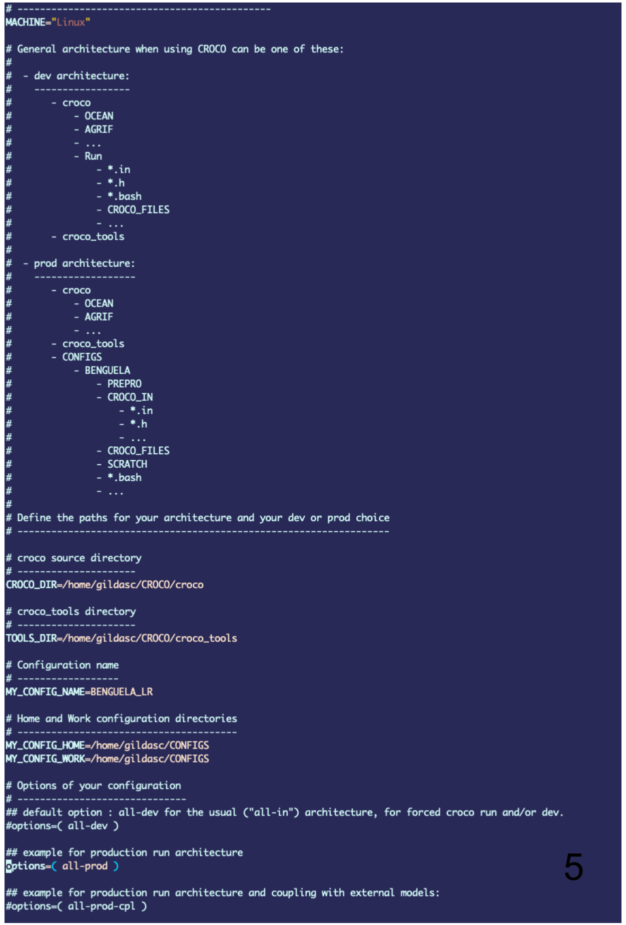
Generalities to deploy a configuration

Go in /home/login

cp CROCO/croco/create_config.bash .

=> Edit create_config.bash
(e.g. with vi)

Note : 3 options of configuration architectures available :
"all-dev": for dev of analytical tests
"all-prod": for production climatological / interannual simulations => provides additional scripts
"all-prod-cpl" : for coupled simulations (ww3, wrf)=> provides additional scripts
=> choose « all-prod »



```
# --------------------------------------------
MACHINE="Linux"

# General architecture when using CROCO can be one of these:
#
#  - dev architecture:
#    ----------------
#       - croco
#           - OCEAN
#           - AGRIF
#           - ...
#           - Run
#               - *.in
#               - *.h
#               - *.bash
#               - CROCO_FILES
#               - ...
#       - croco_tools
#
#  - prod architecture:
#    -----------------
#       - croco
#           - OCEAN
#           - AGRIF
#           - ...
#       - croco_tools
#       - CONFIGS
#           - BENGUELA
#               - PREPRO
#               - CROCO_IN
#                   - *.in
#                   - *.h
#                   - ...
#               - CROCO_FILES
#               - SCRATCH
#               - *.bash
#               - ...
#
# Define the paths for your architecture and your dev or prod choice
# ------------------------------------------------------------------

# croco source directory
# --------------------
CROCO_DIR=/home/gildasc/CROCO/croco

# croco_tools directory
# --------------------
TOOLS_DIR=/home/gildasc/CROCO/croco_tools

# Configuration name
# -----------------
MY_CONFIG_NAME=BENGUELA_LR

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

# Options of your configuration
# -----------------------------
## default option : all-dev for the usual ("all-in") architecture, for forced croco run and/or dev.
#options=( all-dev )

## example for production run architecture
options=( all-prod )

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

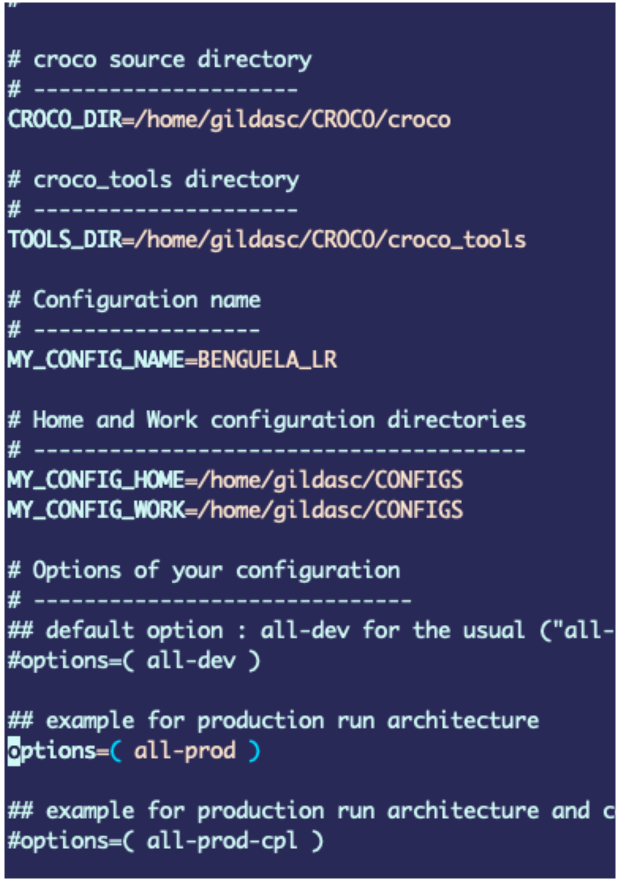
Generalities to deploy a configuration

Go in /home/login

cp CROCO/croco/create_config.bash .

=> Edit create_config.bash
(e.g. with vi)

Note : 3 options of configuration architectures available :
"all-dev": for dev of analytical tests
"all-prod": for production climatological / interannual simulations => provides additional scripts
"all-prod-cpl" : for coupled simulations (ww3, wrf)=> provides additional scripts
=> choose « all-prod »



```
# croco source directory
# ---------------------
CROCO_DIR=/home/gildasc/CROCO/croco

# croco_tools directory
# ---------------------
TOOLS_DIR=/home/gildasc/CROCO/croco_tools

# Configuration name
# ------------------
MY_CONFIG_NAME=BENGUELA_LR

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

# Options of your configuration
# -----------------------------
## default option : all-dev for the usual ("all-
#options=( all-dev )

## example for production run architecture
options=( all-prod )

## example for production run architecture and c
#options=( all-prod-cpl )
```

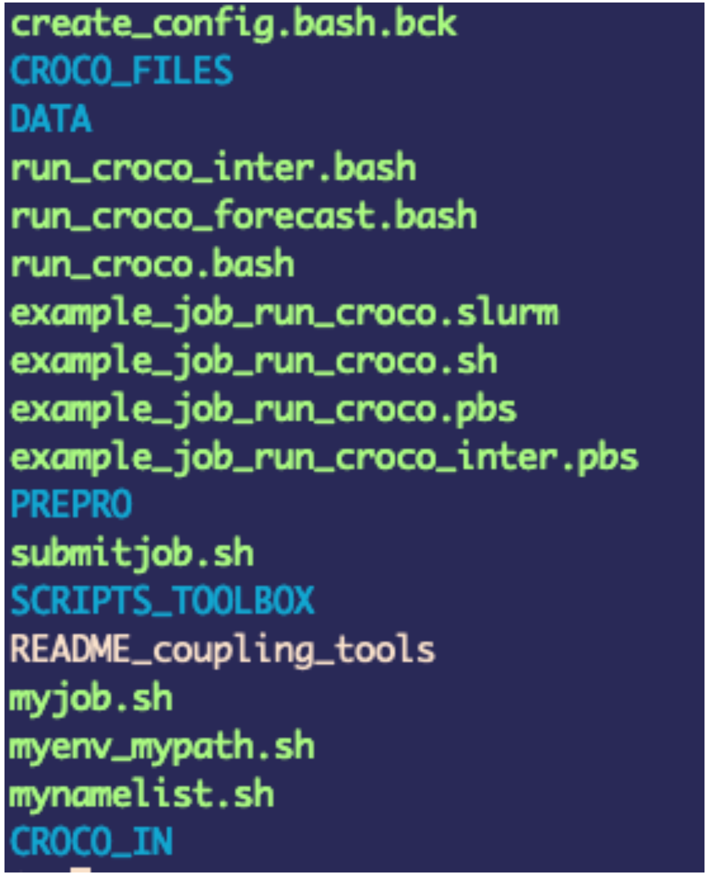
Deploy the BENGUELA_LR (default) configuration

Run the create_config script:
```
./create_config.bash
```

=> It will create a BENGUELA_LR configuration in your CONFIGS directory

```
cd CONFIGS/BENGUELA_LR
ls -l
```

/home/login/CONFIGS/BENGUELA_LR



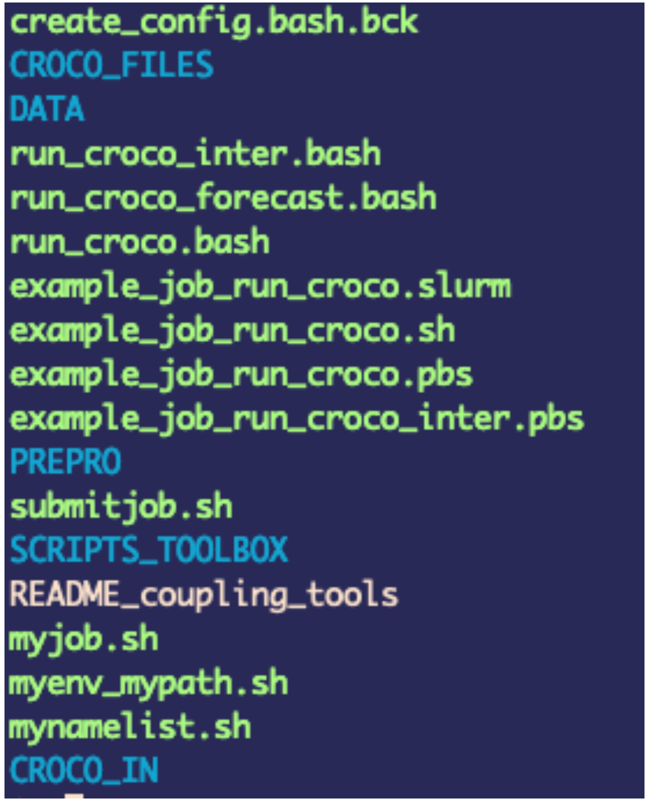
```
create_config.bash.bck
CROCO_FILES
DATA
run_croco_inter.bash
run_croco_forecast.bash
run_croco.bash
example_job_run_croco.slurm
example_job_run_croco.sh
example_job_run_croco.pbs
example_job_run_croco_inter.pbs
PREPRO
submitjob.sh
SCRIPTS_TOOLBOX
README_coupling_tools
myjob.sh
myenv_mypath.sh
mynamelist.sh
CROCO_IN
```

Deploy the BENGUELA_LR (default) configuration

/home/login/CONFIGS/BENGUELA_LR

General architecture of the configuration folder:

create_config.bash.bck	---------------- Backup of create_config script
myenv_mypath.sh	---------------- Environment file
PREPRO	---------------- Directory for preprocessing
CROCO_IN	---------------- Directory for CROCO compilation and settings
CROCO_FILES	---------------- Directory for CROCO inputs and outputs files
SCRATCH	---------------- Directory where the run is executed
run_croco.bash	---------------- Script for launching climatological runs
run_croco_inter.bash	---------------- Script for launching interannual runs
run_croco_forecast.bash	------------ Script for launching forecast runs
mynamelist.sh myjob.sh submitjob.sh SCRIPTS_TOOLBOX	---------------- Scripts for setting and launching simulation with the coupling toolbox



```
create_config.bash.bck
CROCO_FILES
DATA
run_croco_inter.bash
run_croco_forecast.bash
run_croco.bash
example_job_run_croco.slurm
example_job_run_croco.sh
example_job_run_croco.pbs
example_job_run_croco_inter.pbs
PREPRO
submitjob.sh
SCRIPTS_TOOLBOX
README_coupling_tools
myjob.sh
myenv_mypath.sh
mynamelist.sh
CROCO_IN
```

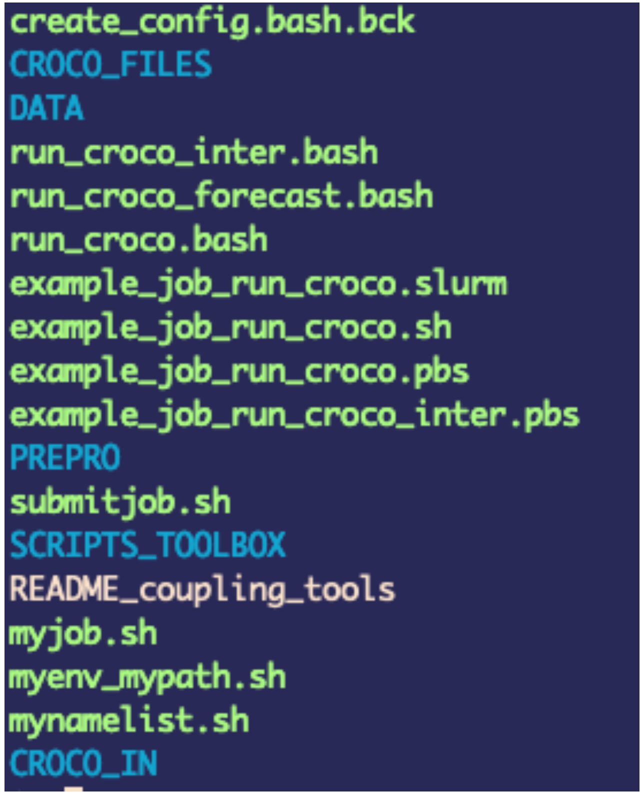
Deploy the BENGUELA_LR (default) configuration

/home/login/CONFIGS/BENGUELA_LR

=> The myenv_mypath.sh file will set all the necessary environment variables, modules, and paths for the machine. Check the file and eventually edit paths if necessary, and source it:

```
source myenv_mypath.sh
```

=> If no error, your environment is now set up.

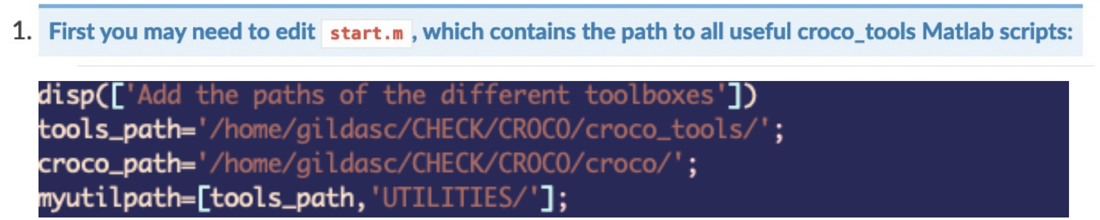


```
create_config.bash.bck
CROCO_FILES
DATA
run_croco_inter.bash
run_croco_forecast.bash
run_croco.bash
example_job_run_croco.slurm
example_job_run_croco.sh
example_job_run_croco.pbs
example_job_run_croco_inter.pbs
PREPRO
submitjob.sh
SCRIPTS_TOOLBOX
README_coupling_tools
myjob.sh
myenv_mypath.sh
mynamelist.sh
CROCO_IN
```

Input files preprocessing

Coastal and Regional Ocean COmmunity model

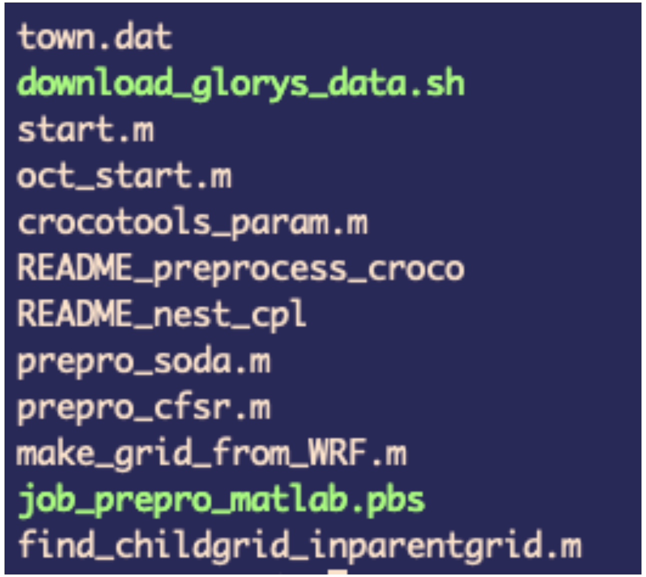
In /home/login/CONFIGS/BENGUELA_LR/PREPRO/CROCO



1. First you may need to edit `start.m`, which contains the path to all useful croco_tools Matlab scripts:

```
disp(['Add the paths of the different toolboxes'])
tools_path='/home/gildasc/CHECK/CROCO/croco_tools/';
croco_path='/home/gildasc/CHECK/CROCO/croco/';
myutilpath=[tools_path,'UTILITIES/'];
```

=> this should be already good for us, as it was set up by create_config.bash



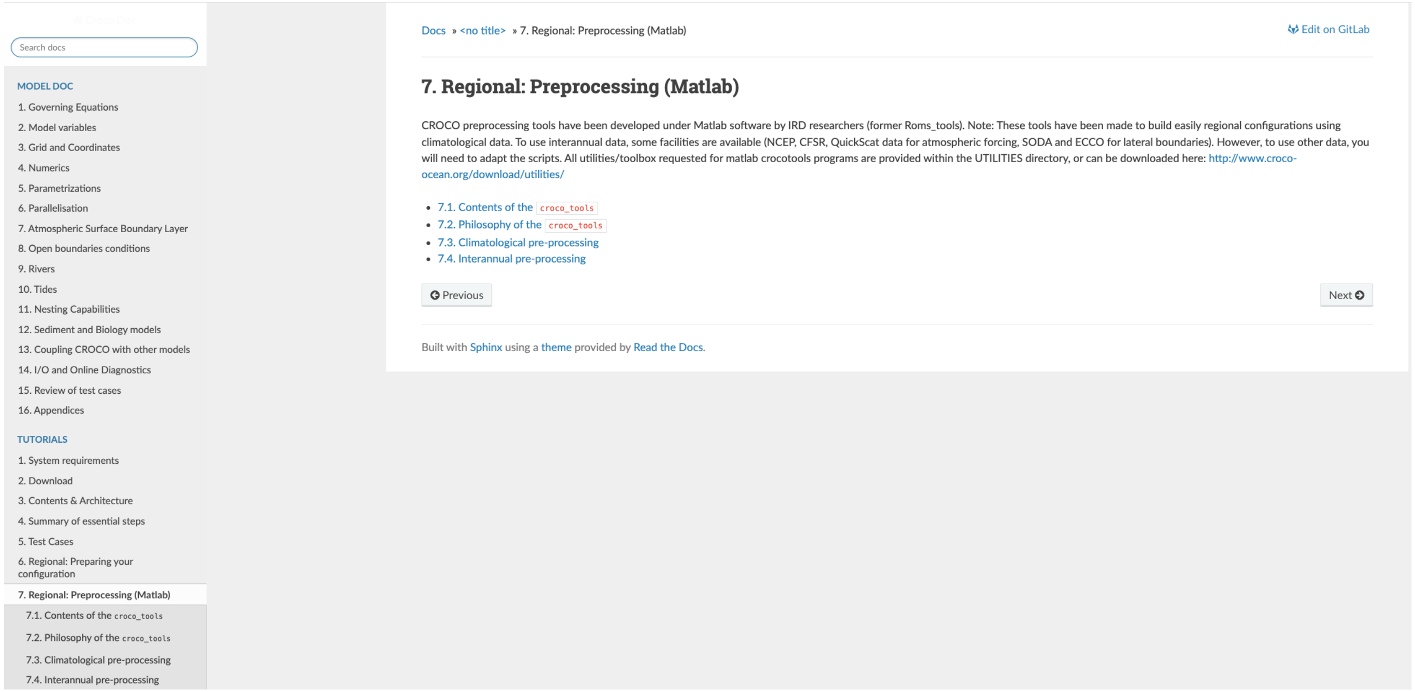
```
town.dat
download_glorys_data.sh
start.m
oct_start.m
crocotools_param.m
README_preprocess_croco
README_nest_cpl
prepro_soda.m
prepro_cfsr.m
make_grid_from_WRF.m
job_prepro_matlab.pbs
find_childgrid_inparentgrid.m
```

Input files preprocessing

Coastal and Regional Ocean COmmunity model

/home/login/CONFIGS/BENGUELA_LR/PREPRO/CROCO

For advanced users, refer to the documentation : (need slight adjustement)
https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.05.prepro.matlab.html



Search docs

MODEL DOC
1. Governing Equations
2. Model variables
3. Grid and Coordinates
4. Numerics
5. Parametrizations
6. Parallelisation
7. Atmospheric Surface Boundary Layer
8. Open boundaries conditions
9. Rivers
10. Tides
11. Nesting Capabilities
12. Sediment and Biology models
13. Coupling CROCO with other models
14. I/O and Online Diagnostics
15. Review of test cases
16. Appendices

TUTORIALS
1. System requirements
2. Download
3. Contents & Architecture
4. Summary of essential steps
5. Test Cases
6. Regional: Preparing your configuration
7. Regional: Preprocessing (Matlab)
7.1. Contents of the `croco_tools`
7.2. Philosophy of the `croco_tools`
7.3. Climatological pre-processing
7.4. Interannual pre-processing

Docs » <no title> » 7. Regional: Preprocessing (Matlab) Edit on GitLab

7. Regional: Preprocessing (Matlab)

CROCO preprocessing tools have been developed under Matlab software by IRD researchers (former Roms_tools). Note: These tools have been made to build easily regional configurations using climatological data. To use interannual data, some facilities are available (NCEP, CFSR, QuickScat data for atmospheric forcing, SODA and ECCO for lateral boundaries). However, to use other data, you will need to adapt the scripts. All utilities/toolbox requested for matlab crocotools programs are provided within the UTILITIES directory, or can be downloaded here: http://www.croco-ocean.org/download/utilities/

- 7.1. Contents of the `croco_tools`
- 7.2. Philosophy of the `croco_tools`
- 7.3. Climatological pre-processing
- 7.4. Interannual pre-processing

Previous Next

Built with Sphinx using a theme provided by Read the Docs.

Input files preprocessing : interannual

/home/login/CONFIGS/BENGUELA_LR/PREPRO/CROCO

Let’s build interannual/plurimonth SODA (ocean) and ERA5 (atmo.) forcings for the BENGUELA_LR configuration

To build these interannual forcings, you need:

- to download the SODA and ERA5 files
- to re-format the files so that croco_tools can read them

=> tools for these 2 steps are available in croco_tools/Aforc_ERA5 and croco_tools/Oforc_OGCM

=> For today, this is already done and the re-formatted data are available in :

/home/COMMONDATA/data_tutos/ERA5_Benguela_LR

/home/COMMONDATA/data_tutos/DATA/SODA_Benguela_LR

=> to use them :

```
cp –Rf /home/COMMONDATA/data_tutos/ERA5_Benguela_LR /home/login/CONFIGS/BENGUELA_LR/DATA
cp –Rf /home/COMMONDATA/data_tutos/SODA_Benguela_LR /home/login/CONFIGS/BENGUELA_LR/DATA
```

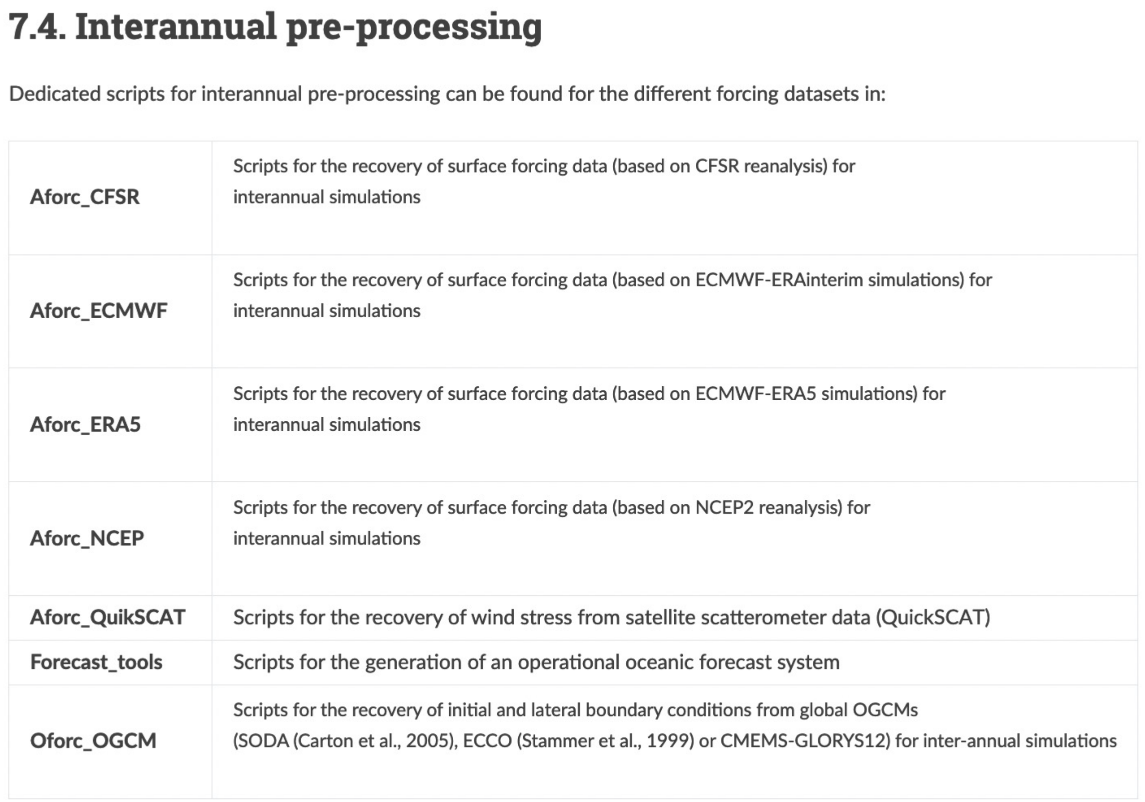
Reminder :

- For climatology (beginners) : make_grid – make_forcing – make_bulk – make_bry – make_ini
- For interannual : we will use : make_grid – make_ERA5 (equivalent to make_bulk) – make_OGCM_mercator (equivalent to make_bry + make_ini)

Input files preprocessing : interannual

Coastal and Regional Ocean COmmunity model

/home/login/CONFIGS/BENGUELA_LR/PREPRO/CROCO



7.4. Interannual pre-processing

Dedicated scripts for interannual pre-processing can be found for the different forcing datasets in:

Aforc_CFSR	Scripts for the recovery of surface forcing data (based on CFSR reanalysis) for interannual simulations
Aforc_ECMWF	Scripts for the recovery of surface forcing data (based on ECMWF-ERAinterim simulations) for interannual simulations
Aforc_ERA5	Scripts for the recovery of surface forcing data (based on ECMWF-ERA5 simulations) for interannual simulations
Aforc_NCEP	Scripts for the recovery of surface forcing data (based on NCEP2 reanalysis) for interannual simulations
Aforc_QuikSCAT	Scripts for the recovery of wind stress from satellite scatterometer data (QuickSCAT)
Forecast_tools	Scripts for the generation of an operational oceanic forecast system
Oforc_OGCM	Scripts for the recovery of initial and lateral boundary conditions from global OGCMs (SODA (Carton et al., 2005), ECCO (Stammer et al., 1999) or CMEMS-GLORYS12) for inter-annual simulations

Input files preprocessing : interannual

Let’s build interannual/plurimonth SODA (ocean) and ERA5 (atmo.) forcings for the BENGUELA_LR configuration

3- in **PREPRO/CROCO**:

- edit crocotools_param.m :
  - Section 1 : adapt/check domain defintion
  - Section 2 adapt path to DATASETS_CROCOTOOLS to
    `DATADIR='/home/COMMONDATA/data_tutos/DATASETS_CROCOTOOLS/';`
  - Section 4 : adapt flag for initial/boundary data options : we need initial and bry oceanic forcing
  - Section 6 adapt/check interannual period
  - Section 7 : adapt/check interannual forcing parameters (we focus here on SODA ans ERA5)
    - Have a look on the overlap parameter
    - `Download_data = 0`

Note

An important aspect is the definition of time and especially the choice of a time origin. The origin of time Yorig should be kept the same for all the preprocessing and postprocessing steps.

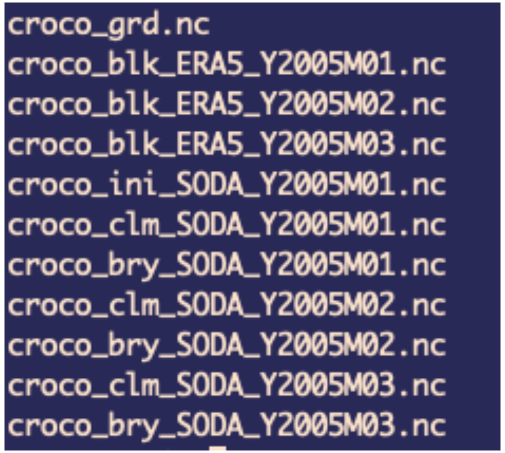
Input files preprocessing : interannual

PREPRO/CROCO run in matlab :

- start
- make_grid (No to all questions)
- make_OGCM
- make_ERA5

4- Check files in CROCO_FILES

```
ls –l /home/login/CONFIGS/BENGUELA_LR/CROCO_FILES
```



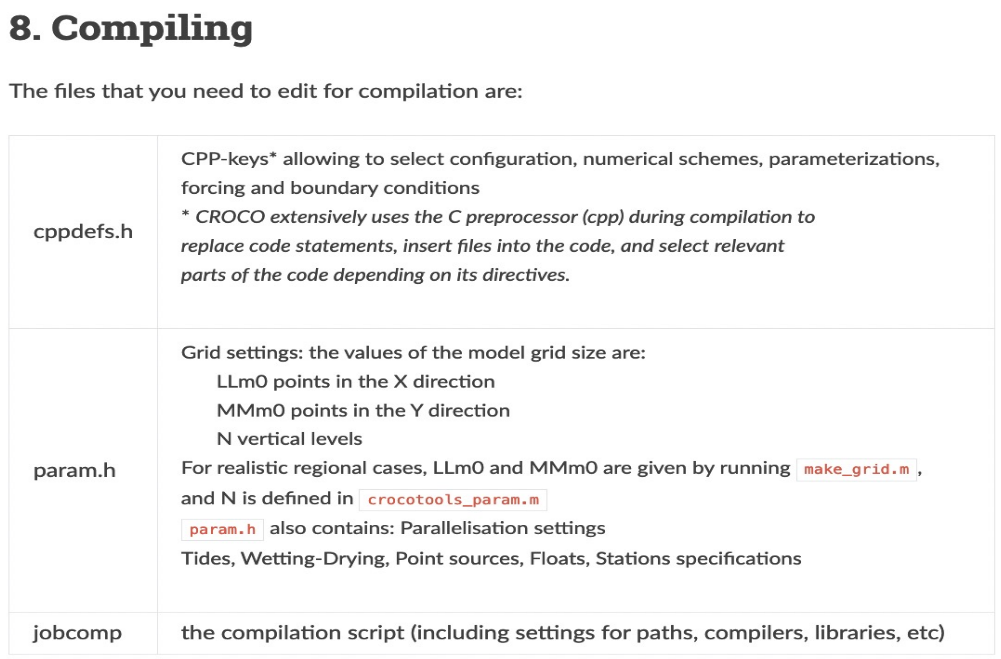
```
croco_grd.nc
croco_blk_ERA5_Y2005M01.nc
croco_blk_ERA5_Y2005M02.nc
croco_blk_ERA5_Y2005M03.nc
croco_ini_SODA_Y2005M01.nc
croco_clm_SODA_Y2005M01.nc
croco_bry_SODA_Y2005M01.nc
croco_clm_SODA_Y2005M02.nc
croco_bry_SODA_Y2005M02.nc
croco_clm_SODA_Y2005M03.nc
croco_bry_SODA_Y2005M03.nc
```

Compilation

Coastal and Regional Ocean COmmunity model

/home/login/CONFIGS/BENGUELA_LR/CROCO_IN

For advanced users, refer to the documentation :
https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.07.compil.html



8. Compiling

The files that you need to edit for compilation are:

cppdefs.h	CPP-keys* allowing to select configuration, numerical schemes, parameterizations, forcing and boundary conditions ** CROCO extensively uses the C preprocessor (cpp) during compilation to replace code statements, insert files into the code, and select relevant parts of the code depending on its directives.*
param.h	Grid settings: the values of the model grid size are: LLm0 points in the X direction MMm0 points in the Y direction N vertical levels For realistic regional cases, LLm0 and MMm0 are given by running `make_grid.m`, and N is defined in `crocotools_param.m` `param.h` also contains: Parallelisation settings Tides, Wetting-Drying, Point sources, Floats, Stations specifications
jobcomp	the compilation script (including settings for paths, compilers, libraries, etc)

Compilation

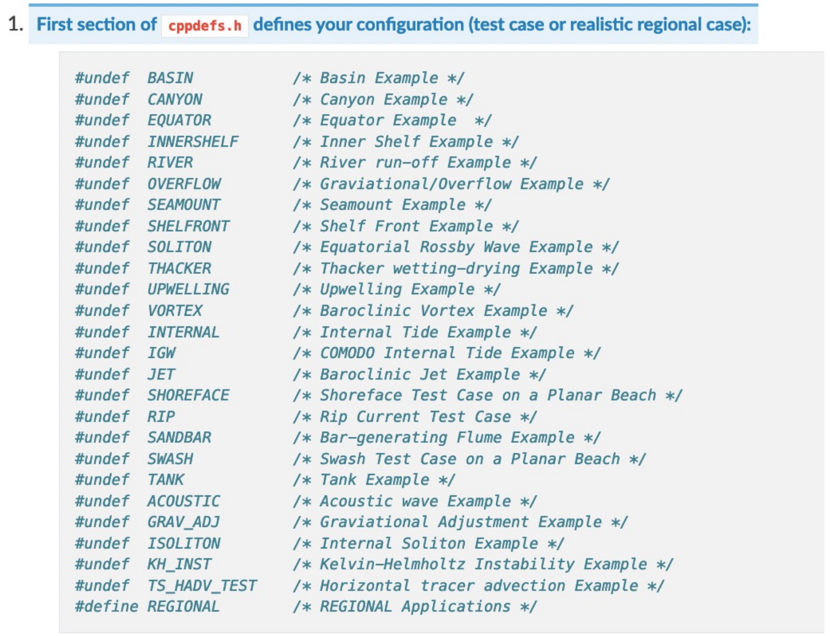
/home/login/CONFIGS/BENGUELA_LR/CROCO_IN

8.1. cppdefs.h

Physical and numerical choices : cppdefs.h

cppdefs.h : define physical and numerical choices

- Define CPP keys used by the C-preprocessor when compiling the model
- Reduce the code to its minimal size : fast compilation
- Avoid FORTRAN logical statements: efficient coding



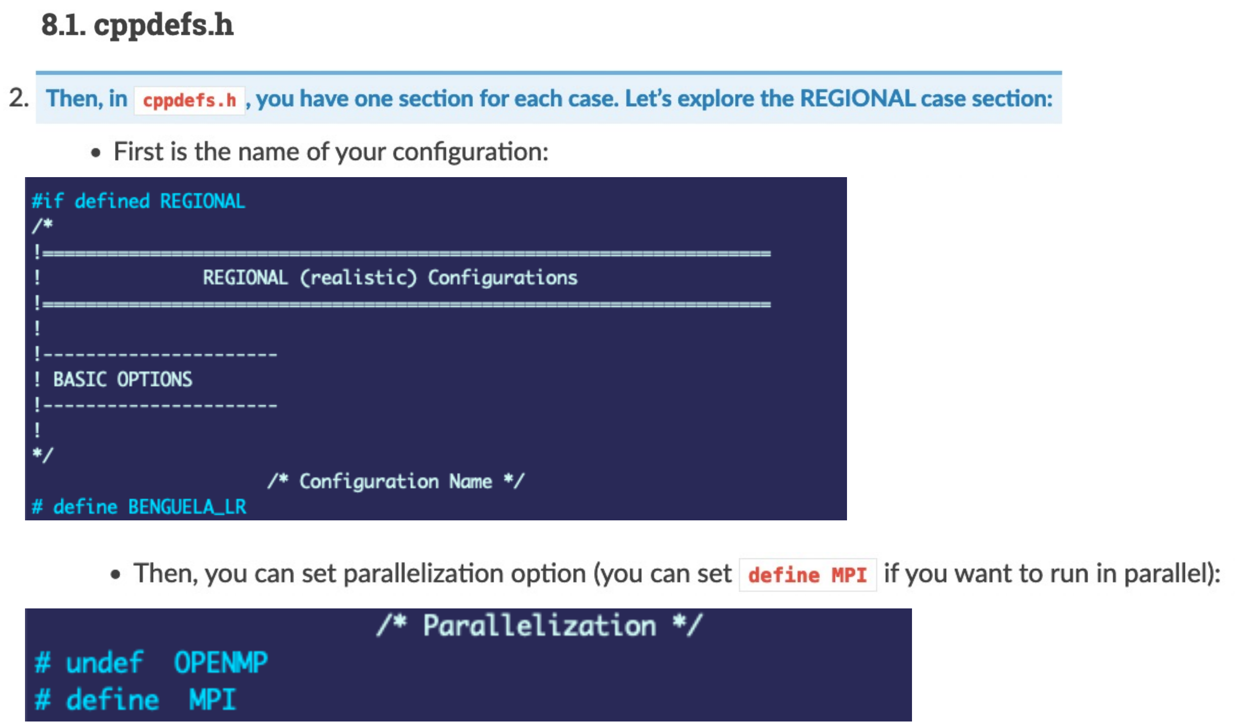
1. First section of `cppdefs.h` defines your configuration (test case or realistic regional case):

```
#undef  BASIN           /* Basin Example */
#undef  CANYON          /* Canyon Example */
#undef  EQUATOR         /* Equator Example  */
#undef  INNERSHELF      /* Inner Shelf Example */
#undef  RIVER           /* River run-off Example */
#undef  OVERFLOW        /* Graviational/Overflow Example */
#undef  SEAMOUNT        /* Seamount Example */
#undef  SHELFRONT       /* Shelf Front Example */
#undef  SOLITON         /* Equatorial Rossby Wave Example */
#undef  THACKER         /* Thacker wetting-drying Example */
#undef  UPWELLING       /* Upwelling Example */
#undef  VORTEX          /* Baroclinic Vortex Example */
#undef  INTERNAL        /* Internal Tide Example */
#undef  IGW             /* COMODO Internal Tide Example */
#undef  JET             /* Baroclinic Jet Example */
#undef  SHOREFACE       /* Shoreface Test Case on a Planar Beach */
#undef  RIP             /* Rip Current Test Case */
#undef  SANDBAR         /* Bar-generating Flume Example */
#undef  SWASH           /* Swash Test Case on a Planar Beach */
#undef  TANK            /* Tank Example */
#undef  ACOUSTIC        /* Acoustic wave Example */
#undef  GRAV_ADJ        /* Graviational Adjustment Example */
#undef  ISOLITON        /* Internal Soliton Example */
#undef  KH_INST         /* Kelvin-Helmholtz Instability Example */
#undef  TS_HADV_TEST    /* Horizontal tracer advection Example */
#define REGIONAL        /* REGIONAL Applications */
```

Compilation

CROCO
Coastal and Regional Ocean COmmunity model

/home/login/CONFIGS/BENGUELA_LR/CROCO_IN



8.1. cppdefs.h

2. Then, in `cppdefs.h` , you have one section for each case. Let's explore the REGIONAL case section:

   - First is the name of your configuration:

```
#if defined REGIONAL
/*
!====================================================================
!                REGIONAL (realistic) Configurations
!====================================================================
!
!----------------------
! BASIC OPTIONS
!----------------------
!
*/
                                /* Configuration Name */
# define BENGUELA_LR
```

   - Then, you can set parallelization option (you can set `define MPI` if you want to run in parallel):

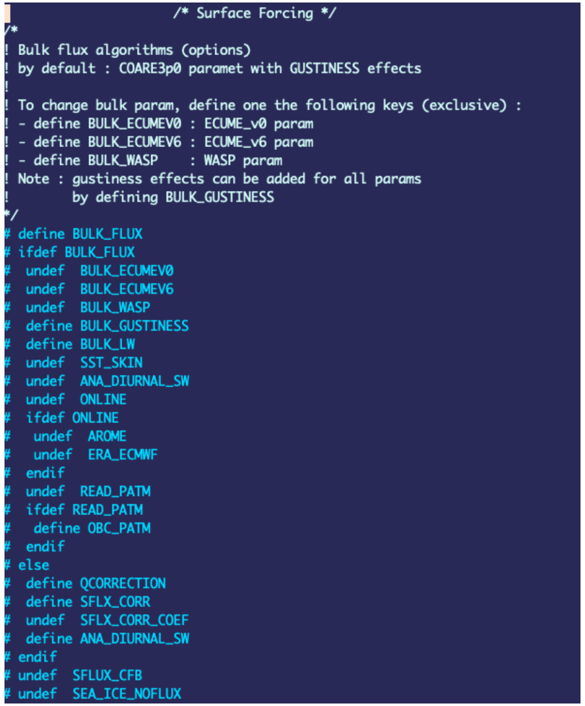
```
                        /* Parallelization */
# undef  OPENMP
# define  MPI
```

Compilation

/home/login/CONFIGS/BENGUELA_LR/CROCO_IN

8.1. cppdefs.h

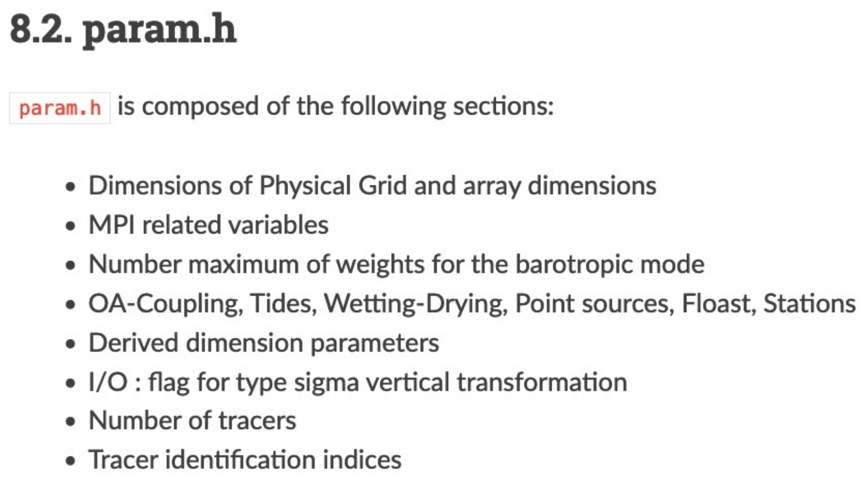
Then define BULK_FLUX to use ERA5 interannual forcing



```
                        /* 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
# 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
```

Compiling

/home/login/CONFIGS/BENGUELA_LR/CROCO_IN



8.2. param.h

`param.h` is composed of the following sections:

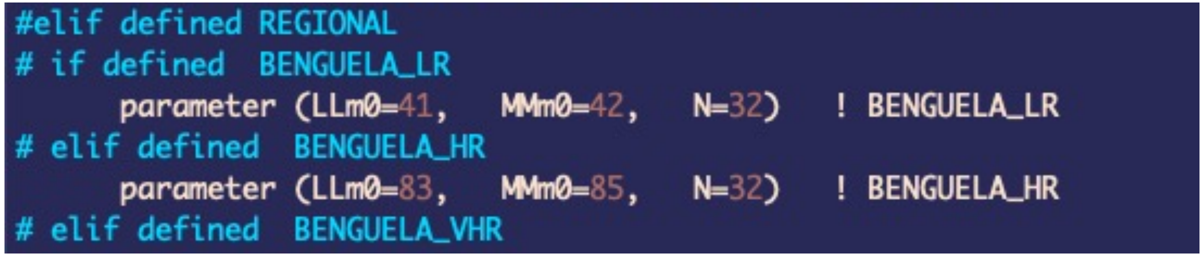
- Dimensions of Physical Grid and array dimensions
- MPI related variables
- Number maximum of weights for the barotropic mode
- OA-Coupling, Tides, Wetting-Drying, Point sources, Floast, Stations
- Derived dimension parameters
- I/O : flag for type sigma vertical transformation
- Number of tracers
- Tracer identification indices

Compilation

/home/login/CONFIGS/BENGUELA_LR/CROCO_IN

8.2. param.h

1. **Check the grid settings:**



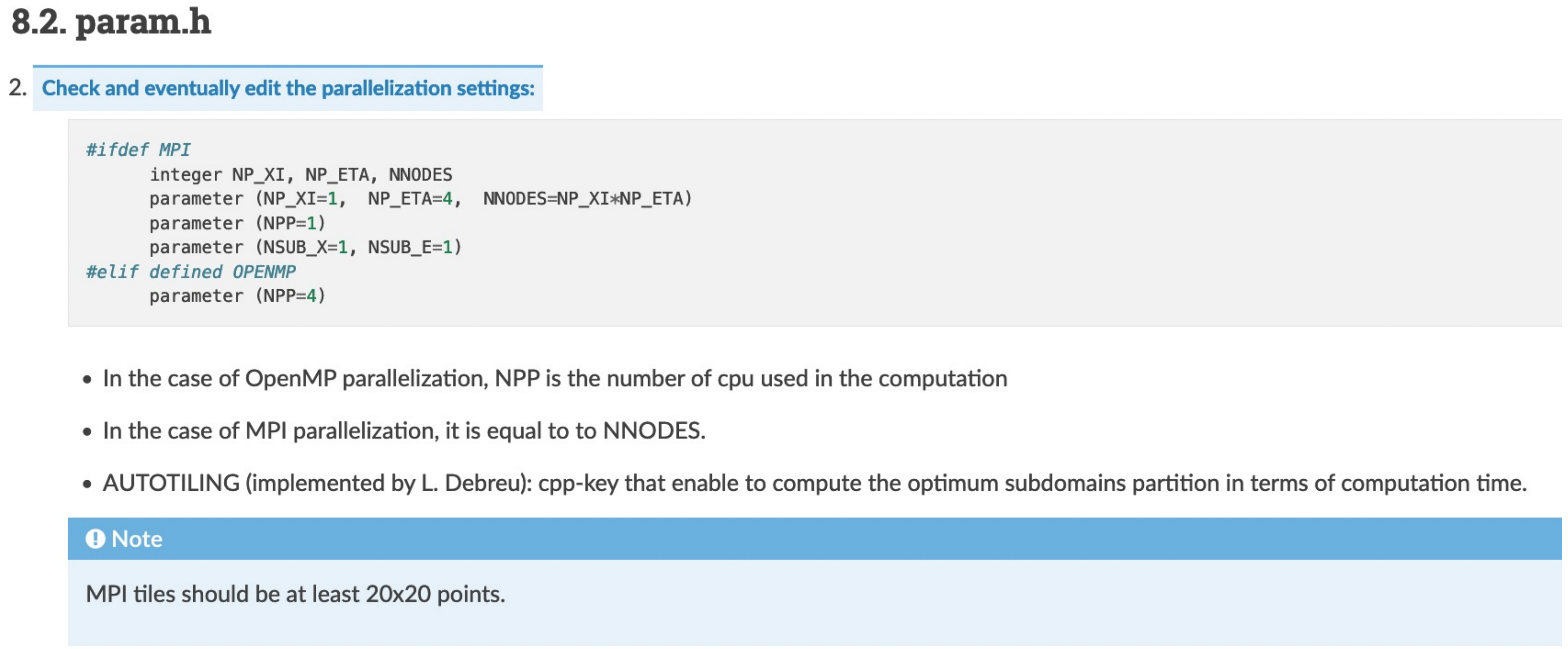
```
#elif defined REGIONAL
# if defined  BENGUELA_LR
      parameter (LLm0=41,   MMm0=42,   N=32)   ! BENGUELA_LR
# elif defined  BENGUELA_HR
      parameter (LLm0=83,   MMm0=85,   N=32)   ! BENGUELA_HR
# elif defined  BENGUELA_VHR
```

- **LLm0**: Dimension (ghost points included) in the ξ direction.
- **MMm0**: Dimension (ghost points included) in the η direction.
- **N**: Number of ρ-vertical points, in the vertical grid.

Compilation

Coastal and Regional Ocean COmmunity model

/home/login/CONFIGS/BENGUELA_LR/CROCO_IN



8.2. param.h

2. **Check and eventually edit the parallelization settings:**

```
#ifdef MPI
      integer NP_XI, NP_ETA, NNODES
      parameter (NP_XI=1,  NP_ETA=4,  NNODES=NP_XI*NP_ETA)
      parameter (NPP=1)
      parameter (NSUB_X=1, NSUB_E=1)
#elif defined OPENMP
      parameter (NPP=4)
```

- In the case of OpenMP parallelization, NPP is the number of cpu used in the computation
- In the case of MPI parallelization, it is equal to to NNODES.
- AUTOTILING (implemented by L. Debreu): cpp-key that enable to compute the optimum subdomains partition in terms of computation time.

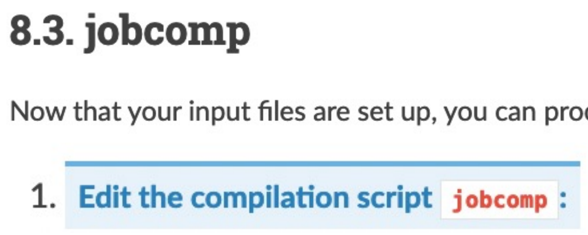
Note

MPI tiles should be at least 20x20 points.

Compilation

Coastal and Regional Ocean COmmunity model

/home/login/CONFIGS/BENGUELA_LR/CROCO_IN

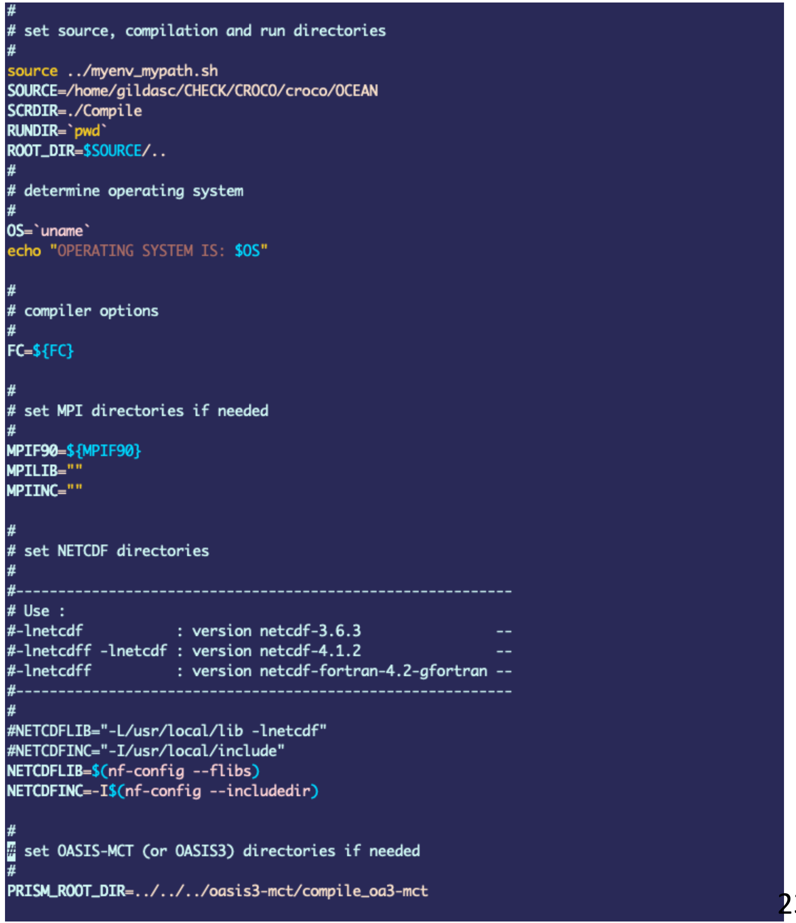


8.3. jobcomp

Now that your input files are set up, you can proceed to compilation:

1. Edit the compilation script `jobcomp` :

you should have nothing to do as everything should be already set-up by myenv_mypath



```
#
# set source, compilation and run directories
#
source ../myenv_mypath.sh
SOURCE=/home/gildasc/CHECK/CROCO/croco/OCEAN
SCRDIR=./Compile
RUNDIR=`pwd`
ROOT_DIR=$SOURCE/..
#
# determine operating system
#
OS=`uname`
echo "OPERATING SYSTEM IS: $OS"

#
# compiler options
#
FC=${FC}

#
# set MPI directories if needed
#
MPIF90=${MPIF90}
MPILIB=""
MPIINC=""

#
# set NETCDF directories
#
#-----------------------------------------------------------
# Use :
#-lnetcdf           : version netcdf-3.6.3               --
#-lnetcdff -lnetcdf : version netcdf-4.1.2               --
#-lnetcdff          : version netcdf-fortran-4.2-gfortran --
#-----------------------------------------------------------
#
#NETCDFLIB="-L/usr/local/lib -lnetcdf"
#NETCDFINC="-I/usr/local/include"
NETCDFLIB=$(nf-config --flibs)
NETCDFINC=-I$(nf-config --includedir)

#
# set OASIS-MCT (or OASIS3) directories if needed
#
PRISM_ROOT_DIR=../../../oasis3-mct/compile_oa3-mct
```

Running with interannual forcing (ERA5 + SODA)

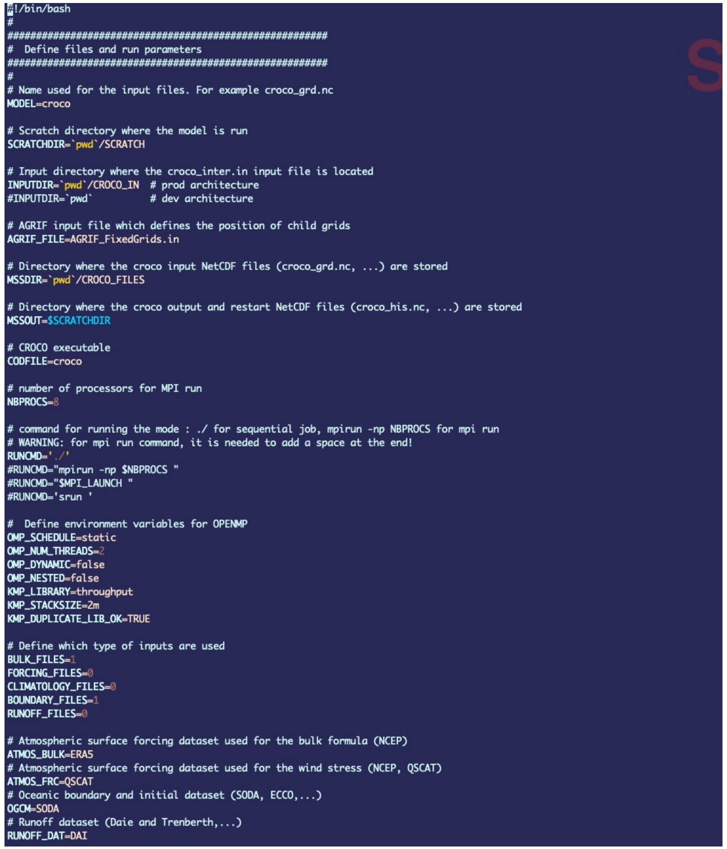
Coastal and Regional Ocean COmmunity model

/home/login/CONFIGS/BENGUELA_LR/

For advanced users, refer to the documentation (some adaptation needed):
https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.10.run.inter.html

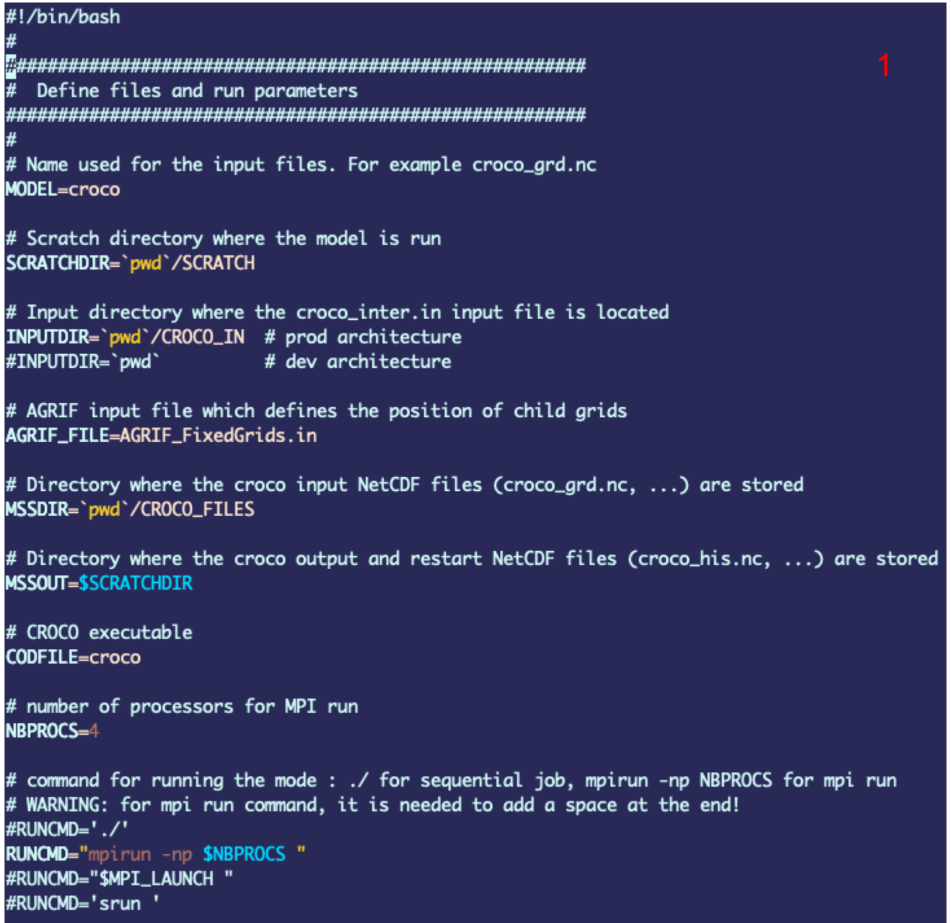
Edit run_croco_inter.bash

- Check/adapt the various sections
- RUNCMD='./'
- DT
- ATMOS_BULK / OGCM / …
- ...

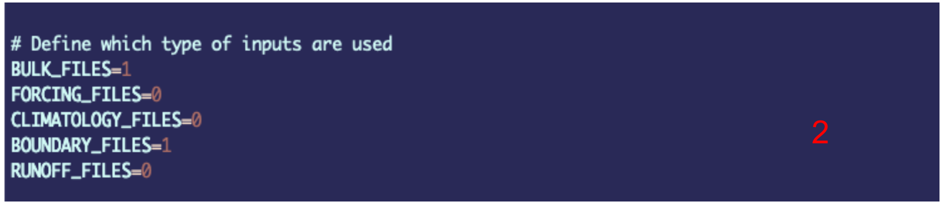


```
#!/bin/bash
#
########################################################
#  Define files and run parameters
########################################################
#
# Name used for the input files. For example croco_grd.nc
MODEL=croco

# Scratch directory where the model is run
SCRATCHDIR=`pwd`/SCRATCH

# Input directory where the croco_inter.in input file is located
INPUTDIR=`pwd`/CROCO_IN  # prod architecture
#INPUTDIR=`pwd`          # dev architecture

# AGRIF input file which defines the position of child grids
AGRIF_FILE=AGRIF_FixedGrids.in

# Directory where the croco input NetCDF files (croco_grd.nc, ...) are stored
MSSDIR=`pwd`/CROCO_FILES

# Directory where the croco output and restart NetCDF files (croco_his.nc, ...) are stored
MSSOUT=$SCRATCHDIR

# CROCO executable
CODFILE=croco

# number of processors for MPI run
NBPROCS=8

# 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 -np $NBPROCS "
#RUNCMD="$MPI_LAUNCH "
#RUNCMD='srun '

#  Define environment variables for OPENMP
OMP_SCHEDULE=static
OMP_NUM_THREADS=2
OMP_DYNAMIC=false
OMP_NESTED=false
KMP_LIBRARY=throughput
KMP_STACKSIZE=2m
KMP_DUPLICATE_LIB_OK=TRUE

# 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 (Daie and Trenberth,...)
RUNOFF_DAT=DAI
```

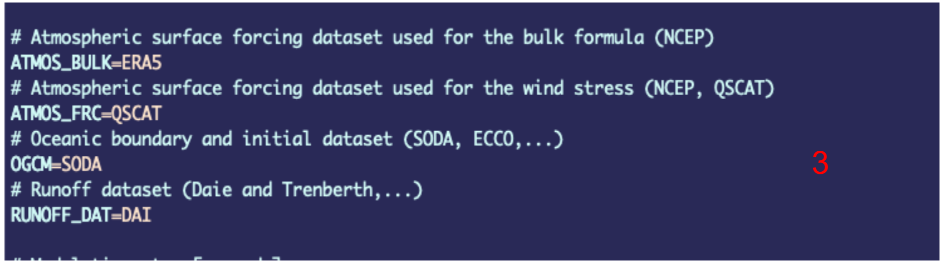
Running with interannual forcing (ERA5 + SODA)



```
#!/bin/bash
#
#####################################################
#  Define files and run parameters
#####################################################
#
# Name used for the input files. For example croco_grd.nc
MODEL=croco

# Scratch directory where the model is run
SCRATCHDIR=`pwd`/SCRATCH

# Input directory where the croco_inter.in input file is located
INPUTDIR=`pwd`/CROCO_IN  # prod architecture
#INPUTDIR=`pwd`          # dev architecture

# AGRIF input file which defines the position of child grids
AGRIF_FILE=AGRIF_FixedGrids.in

# Directory where the croco input NetCDF files (croco_grd.nc, ...) are stored
MSSDIR=`pwd`/CROCO_FILES

# Directory where the croco output and restart NetCDF files (croco_his.nc, ...) are stored
MSSOUT=$SCRATCHDIR

# CROCO executable
CODFILE=croco

# number of processors for MPI run
NBPROCS=4

# 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 -np $NBPROCS "
#RUNCMD="$MPI_LAUNCH "
#RUNCMD='srun '
```

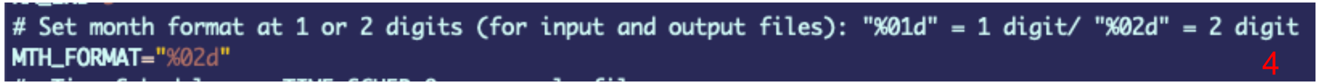


```
# Define which type of inputs are used
BULK_FILES=1
FORCING_FILES=0
CLIMATOLOGY_FILES=0
BOUNDARY_FILES=1
RUNOFF_FILES=0
```



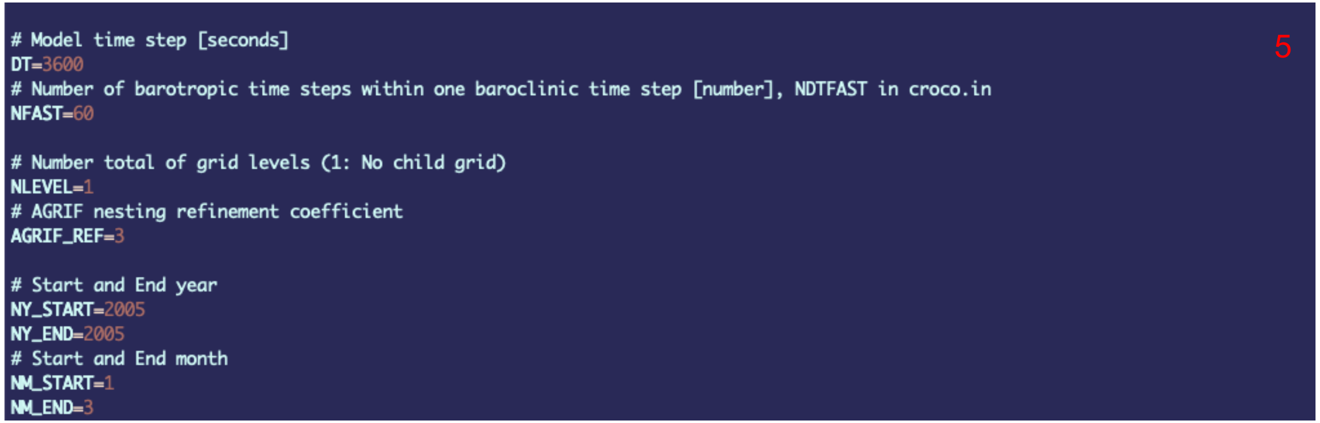
```
# Atmospheric surface forcing dataset used for the bulk formula (NCEP)
ATMOS_BULK=ERA5
# Atmospheric surface forcing dataset used for the wind stress (NCEP, QSCAT)
ATMOS_FRC=QSCAT
# Oceanic boundary and initial dataset (SODA, ECCO,...)
OGCM=SODA
# Runoff dataset (Daie and Trenberth,...)
RUNOFF_DAT=DAI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

Running with interannual forcing (ERA5 + SODA)



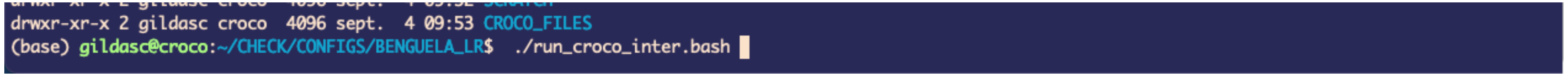
```
# Set month format at 1 or 2 digits (for input and output files): "%01d" = 1 digit/ "%02d" = 2 digit
MTH_FORMAT="%02d"
```

4



5

```
# 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
```



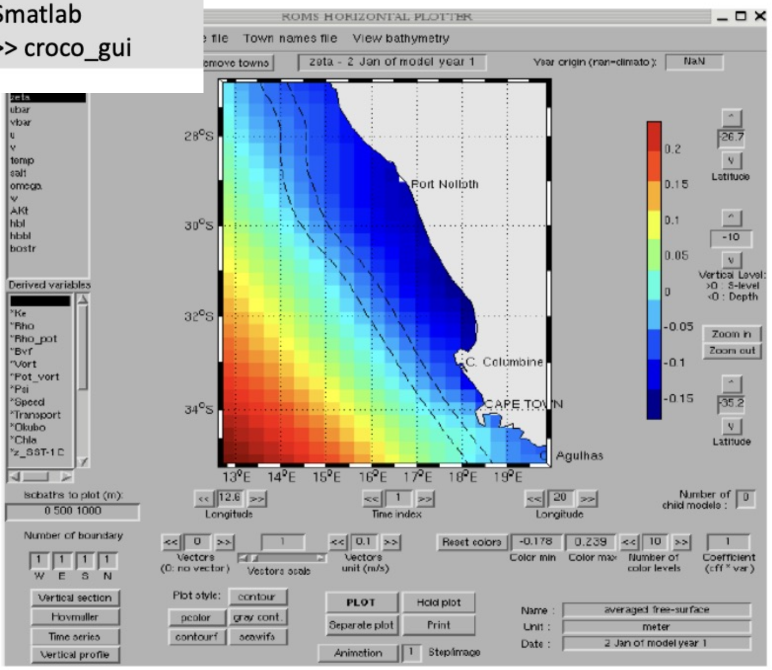
```
drwxr-xr-x 2 gildasc croco  4096 sept.  4 09:53 CROCO_FILES
(base) gildasc@croco:~/CHECK/CONFIGS/BENGUELA_LR$  ./run_croco_inter.bash
```

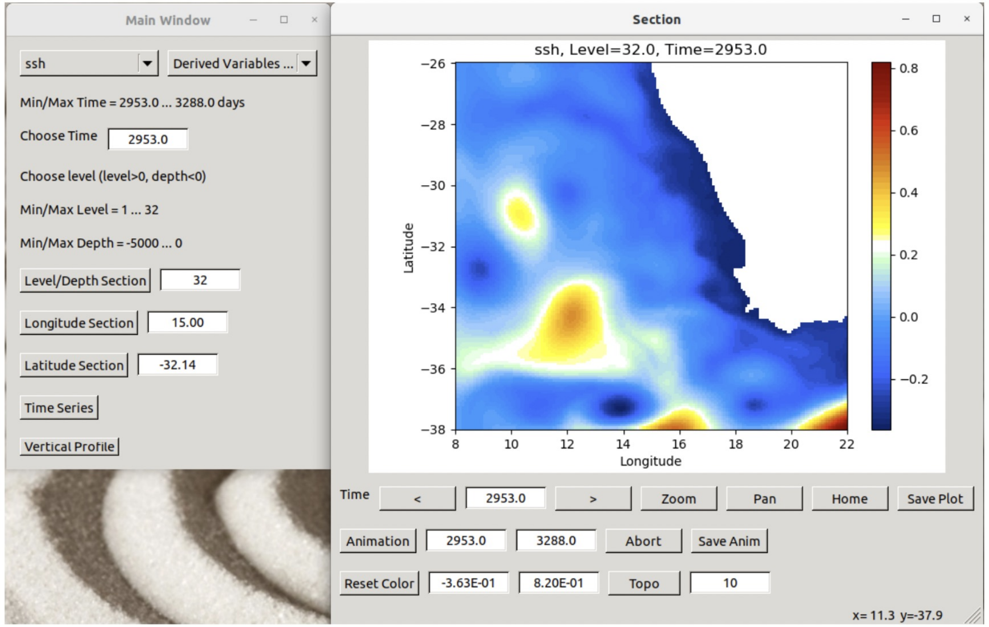
Visualisation of Outputs

/home/login/CONFIGS/BENGUELA_LR/PREPRO/CROCO

You can explore your model outputs (croco_his.nc, croco_avg.nc) using different frameworks (ncview, ferret, etc). In the croco_tools, a matlab interface is offered to explore your data: `croco_gui`, as well as a Python interface `croco_pyvisu`. These are explored in other tutorials.

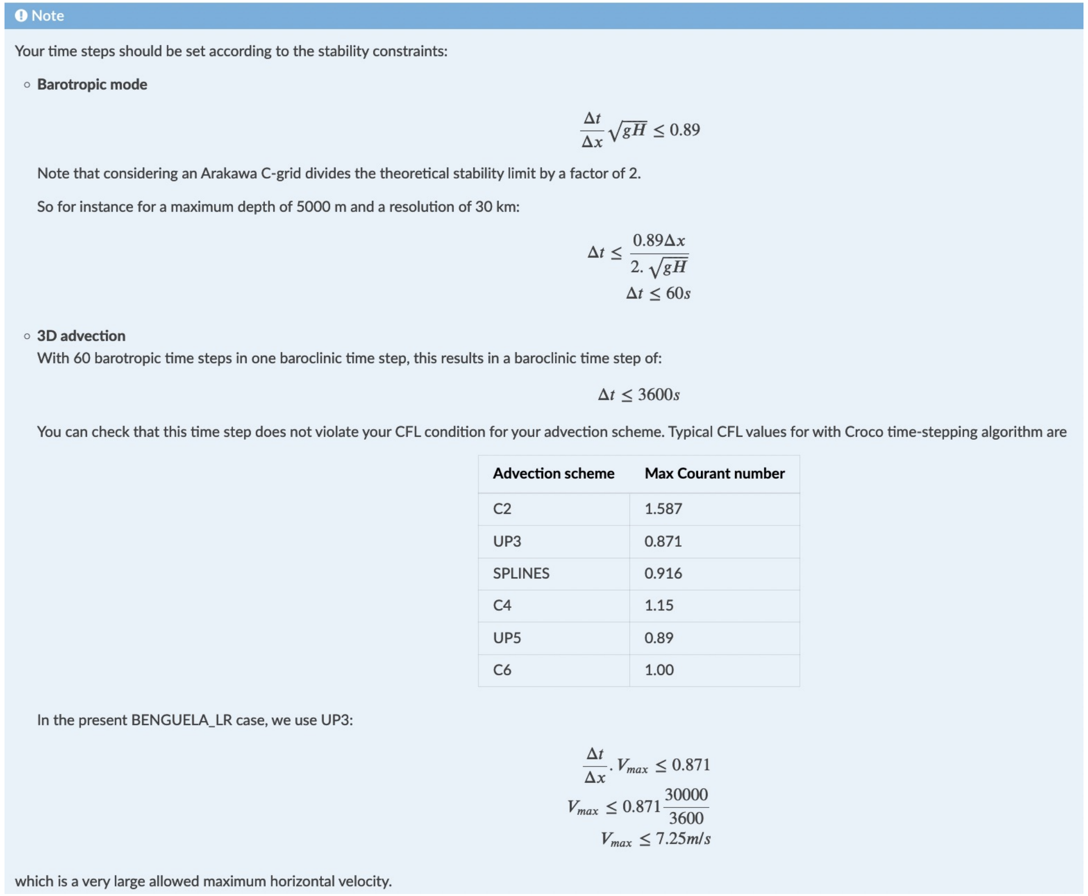
```
$matlab
>> croco_gui
```





Running

Coastal and Regional Ocean COmmunity model



Note

Your time steps should be set according to the stability constraints:

- **Barotropic mode**

$$\frac{\Delta t}{\Delta x}\sqrt{gH} \leq 0.89$$

Note that considering an Arakawa C-grid divides the theoretical stability limit by a factor of 2.

So for instance for a maximum depth of 5000 m and a resolution of 30 km:

$$\Delta t \leq \frac{0.89\Delta x}{2.\sqrt{gH}}$$
$$\Delta t \leq 60s$$

- **3D advection**
With 60 barotropic time steps in one baroclinic time step, this results in a baroclinic time step of:

$$\Delta t \leq 3600s$$

You can check that this time step does not violate your CFL condition for your advection scheme. Typical CFL values for with Croco time-stepping algorithm are

Advection scheme	Max Courant number
C2	1.587
UP3	0.871
SPLINES	0.916
C4	1.15
UP5	0.89
C6	1.00

In the present BENGUELA_LR case, we use UP3:

$$\frac{\Delta t}{\Delta x}.V_{max} \leq 0.871$$
$$V_{max} \leq 0.871\frac{30000}{3600}$$
$$V_{max} \leq 7.25m/s$$

which is a very large allowed maximum horizontal velocity.