TUTORIAL 02:
CREATE MY CROCO GRIDS

In this tutorial, we do the step to prepare our first model grid. We will connect to the super-computer LENGAU, and run Matlab to do the first CROCO pre-processing step. We will then analyse CROCO sigma vertical coordinate system.

STEP 1: Logging onto the Lengau HPC cluster

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

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

Replace **login** with your corresponding account number.

→ Reserve one interactive processor to do this pre-processing step (Step 4 from **#TUTORIAL01**):

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

→ Go directly into your **lustre** directory:

```
[login@cnode0220 ~]$ cd lustre
[login@cnode0220 lustre]$ ls
croco croco_tools
[login@cnode0220 lustre]$
```

STEP 2: Creating the work directory of your first CROCO configuration

The simplest example of CROCO configuration is the configuration called **BENGUELA_LR** which corresponds to a domain in the Benguela upwelling zone of the coasts of Namibia and South-Africa with a relatively Low Resolution (LR). This configuration is the one that comes by default in the CROCO code and it is similar to the one described in Penven et al. (2001). Before creating your own configuration, let's begin by recreating the BENGUELA_LR Grid.

→ Go into the **croco** source code directory **croco**:

```
[login@cnode0220 ~]$ cd croco
[login@cnode0220 croco]$ ls
 AGRIF                 DOC_SPHINX                 OCEAN       SCRIPTS
 create_config.bash    MPI_NOLAND_preprocessing   PISCES      TEST_CASES
 CVTK                  MUSTANG                    README.md   XIOS
[login@cnode0220 croco]$
```

→ Edit the file **create_config.bash** using the Linux command **vi** or the **nedit** software:

```
[login@cnode0220 croco]$ vi create_config.bash
[login@cnode0220 croco]$
                                          :
# Configuration name
# ------------------
MY_CONFIG_NAME=Run

# Home and Work configuration directories
# ---------------------------------------
                                          :
```

→ Using the up ⮝ and down ⮟ arrows, locate the line 27 where the MY_CONFIG_NAME is. This name will be the one used for your first configuration directory and the name of the configuration. Go into the **vi** insert mode by typing **i**. Use the horizontal arrows ⮜⮞ and the delete keys to replace the string "Run" by "Run_BENGUELA_LR". Quit the insert mode by pressing the Esc key.

→ Save and exit **vi** by tying the following keys: **:wq**. Then, execute the script:

```
[login@cnode0220 croco]$ ./create_config.bash
[login@cnode0220 croco]$
```

→ Here is what you will see in your terminal:

```
oce-dev is defined. all-in architecture and no external codes considered

Your choices :
 - CROCO_DIR         : /home/login/lustre/croco
 - TOOLS_DIR         : /home/login/lustre/croco_tools
 - CONFIG_HOME_DIR   : /home/login/lustre/croco/
 - CONFIG_WORK_DIR   : /home/login/lustre/croco/
 - CONFIG_NAME       : BENGUELA_LR
 - OPTIONS           : oce-dev xios test_cases agrif inter forc pisces … prepro
 Do you want to proceed ? [Y/n]
```

→ Press the key **Y** to accept. Then you will see:

```
Creating configuration ...

Copy CROCO useful scripts and input files
-----------------------------------------
[login@cnode0220 croco]
```

→ This script creates a directory within **croco** with the name that you defined in MY_CONFIG_NAME with all the necessary codes to create your input files and launch your simulation, this will be your working directory. The contents of this folder should be similar to the following:

```
[login@cnode0220 croco]$ cd Run_BENGUELA_LR
[login@cnode0220 Run_BENGUELA_LR]$ ls
AGRIF_FixedGrids.in      croco_inter.in          oct_start.m
cppdefs_dev.h            croco_inter.in.1        param.h
cppdefs.h                croco_stations.in       run_croco.bash
create_config.bash.bck   crocotools_param.m      run_croco_forecast.bash
CROCO_FILES              DATA                    run_croco_inter.bash
croco.in                 download_glorys_data.sh start.m
croco.in.1               jobcomp                 town.dat
[login@cnode0220 Run_BENGUELA_LR]$
```

→ In steps 3 and 4, you will need one of my Matlab script called **draw_zonal_section.m**. You can copy it from my lustre directory into you CROCO working directory:

```
cp /mnt/lustre/users/sillig/CROCO_TRAINING_Week1/3_Some_files/
                draw_zonal_section.m .
```

STEP 3: Recreating BENGUELA_LR CROCO Grid

→ It is done with **MATLAB**

→ Go into your new CROCO working directory (**cd croco/Run_BENGUELA_LR**)

→ Launch **matlab -nodesktop** (or the alias **mat**)

```
[login@cnode0220 Run_BENGUELA_LR]$ matlab -nodesktop
[login@cnode0220 Run_BENGUELA_LR]$
```

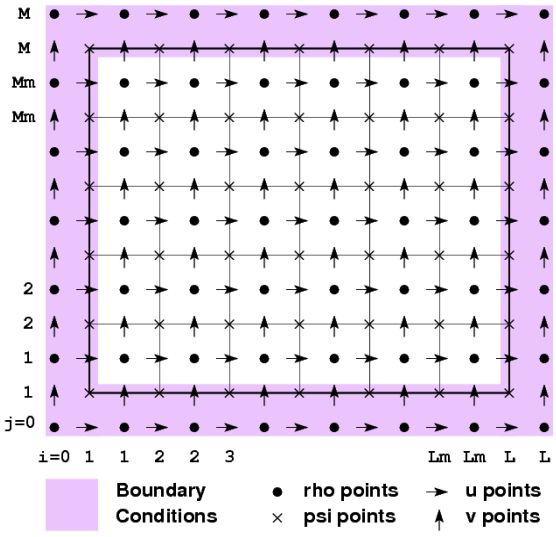
→ Inside MATLAB, execute the command start that will add some paths to Matlab (= tell Matlab where are CROCOTOOLS useful programs), and then open the **crocotools_param.m** file using the command edit:

```
                            < M A T L A B (R) >
                  Copyright 1984-2020 The MathWorks, Inc.
                R2020a Update 8 (9.8.0.1873465) 64-bit (glnxa64)
                               February 3, 2022

To get started, type doc.
For product information, visit www.mathworks.com.
>> start
>> edit crocotools_param
```

→ Analyse the parameters of the section "1- Configuration parameters used by make_grid.m"

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% 1  - Configuration parameters
%      used by make_grid.m (and others..)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
isoctave=exist('octave_config_info');
%
%  CROCO title names and directories
%
CROCO_title  = 'Benguela Model';
CROCO_config = 'Benguela_LR';
%
% Grid dimensions:
%
lonmin =   8;    % Minimum longitude [degree east]
lonmax =  22;    % Maximum longitude [degree east]
latmin = -38;    % Minimum latitude  [degree north]
latmax = -26;    % Maximum latitude  [degree north]
%
% Grid resolution [degree]
%
dl = 1/3;
%
% Number of vertical Levels (! should be the same in param.h !)
%
N = 32;
%
%  Vertical grid parameters (! should be the same in croco.in !)
%
theta_s     =  7.;
theta_b     =  2.;
hc          = 200.;
vtransform =   2.; % s-coordinate type (1: old- ; 2: new- coordinates)
                   % ! take care to define NEW_S_COORD cpp-key in cppdefs.h
```



→ (Off Matlab) You can look at the global topographic input data, i.e. the etopo2 data set:

```
[login@cnode0220 Run_BENGUELA_LR]$ cd ../../croco_tools/Topo
[login@cnode0220 Topo]$ ncdump -h etopo2.nc
[login@cnode0220 Topo]$ ncview etopo2.nc
```

→ (Back in Matlab) Launch the command make_grid: this will create your horizontal/vertical grid (position of the grid point, size of the grid cells, bottom topography, land mask, etc…) using information from the global etopo2 data base.

```
>> make_grid

 Making the grid: Run_BENGUELA_LR/CROCO_FILES/croco_grd.nc

 Title: Benguela Model

 Resolution: 1/3 deg

 Do you want to use interactive grid maker ?
 (e.g., for grid rotation or parameter adjustments) : y,[n] n

 :

 Do you want to use editmask ? y,[n] n

 :

 Write it down...
>>
```

→ For now, type "**no**" when asked to use interactive grid maker or to use editmask.

↳ The grid will be stored in the NetCDF file: **CROCO_FILES/croco_grd.nc**

→ (Off Matlab) You can look at the CROCO grid file:

```
[login@cnode0220 Topo]$ cd ../../croco/Run_BENGUELA_LR/CROCO_FILES
[login@cnode0220 CROCO_FILES]$ ncdump -h croco_grid.nc
[login@cnode0220 CROCO_FILES]$ ncview croco_grid.nc
```

→ (Back in Matlab) Inspect your vertical grid, using the Matlab script that has been copied in your Run directory in Step 2:

draw_zonal_section(N,theta_s,theta_b,hc,vtransform,lat_index)

```
>> help draw_zonal_section
>> draw_zonal_section(32,7,2,200,2,12)
>>
```

↳ Try different values for N, theta_s, theta_b, hc, and vtransform

STEP 4: Creating your CROCO Grid

→ Redo **steps 2 & 3** choosing the region of the world you would like to simulate. Call your configuration **Run_Clim**. You can play with the grid rotation and the editmask.

↳ Remember the size of your grid (LLm, MMm)

Do not exceed a 100x100 grid

Choose a place where there is a river

Do not overlap the equatorial zone by less than 2°

STEP 5: Exiting

→ When you are satisfied with your grid, give back the compute node and logout from Lengau:

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