TUTORIAL 08:

ADDING RIVER RUNOFFS

In this tutorial, we add CROCO input forcing for river inflows. We will connect to the super-computer LENGAU and run Matlab to prepare a river forcing file for the default BENGUELA_LR configuration. We will then compile and run CROCO with seasonal Orange and Doring river runoffs.

In order to include rivers in your simulation domain, there are several variables to define, such as:

- the number of rivers: `Nsrc`
- the position of the rivers in your model grid: `Isrc` and `Jsrc`
- the direction of the river flow (zonal or meridional): `Dsrc`
- the flow rate of the river (in m3/s): `Qbar`
- the type of tracer advected by the river: `Lsrc` and the value/concentration: `Tsrc`

> **Warning**
>
> The sources points must be placed on U or V points on the C-grid and not on rho-points

For more details, please refer to the dedicated CROCO documentation at https://croco-ocean.gitlabpages.inria.fr/croco_doc/model/model.river.html and the River tutorial https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.12.rivers.html.

STEP 1: Logging onto the Lengau HPC cluster and create a new CROCO working directory

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

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

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

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

→ Go into your **croco-v2.0.1** directory (**lustre/CROCO/croco-v2.0.1**):

```
[login@cnode0220 ~]$ cd lustre/CROCO/croco-v2.0.1
[login@cnode0220 croco-v2.0.1]$
```

NODES

→ Create a new CROCO configuration called **Run**. For this, you have to repeat **STEP 2** from **#TUTORIAL02**, i.e. edit **create_config.bash** and execute it:

```
[login@cnode0220 croco-v2.0.1]$ nedit create_config.bash &
[login@cnode0220 croco-v2.0.1]$ ./create_config.bash
[login@cnode0220 croco-v2.0.1]$ cd Run
```

NODES

STEP 2: Creating CROCO boundary input files and the river runoff file

The simplest example of CROCO configuration is the configuration called **BENGUELA_LR** which corresponds to a domain in the Benguela upwelling region off 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). We will run this configuration with climatological forcings and include the Orange and Doring river runoffs.

→ Launch **MATLAB**:

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

NODES

→ Create climatological model forcing without editing **crocotools_param.m** (see TUTO 04):

```
>> start
>> make_grid; make_forcing; make_clim; make_bry; make_ini
```

➥ **frc** and **bry** are the files associated with the default **cppdef.h** `CPPKEYS`.

→ Create the river forcing flows by using the Matlab script make_runoff.m:

➥ This Matlab script detects the main rivers located in your configuration domain. Data come from the RUNOFF_DAI dataset (Dai and Trenberth, 2000), a global monthly runoff climatology containing the 925 largest rivers over the world.

→ Edit the Matlab script make_runoff:

```
>> edit make_runoff
>>
```

❶ For this climatological simulation, we choose the following flag:

```
% Choose the monthly runoff forcing time and cycle in days
clim_run=1;
%     - times and cycles for runoff conditions:
%           - clim_run = 1 % climato forcing experiments with climato calendar
%                     qbar_time=[15:30:365];
%                     qbar_cycle=360;
%
%           - clim_run = 0 % interanual forcing experiments with real calendar
%                     qbar_time=[15.2188:30.4375:350.0313];
%                     qbar_cycle=365.25;
```

❷ We choose seasonally varying river discharges:

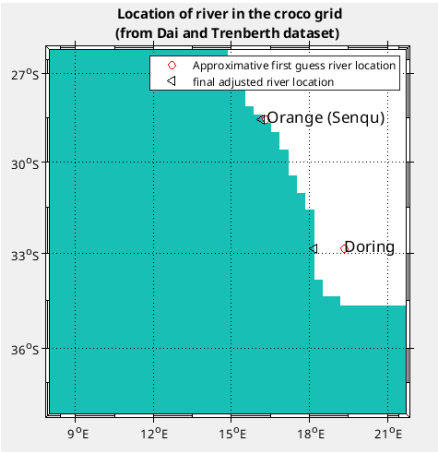
```
psource_ncfile_ts=1;

%        - pource_ncfile_ts = 1  => Variable runoff tracers
%                                   concentration  processing is activated.
%                                     - psource_ts_auto : auto definition using
%                                                         the nearest point in the climatlogy file
```

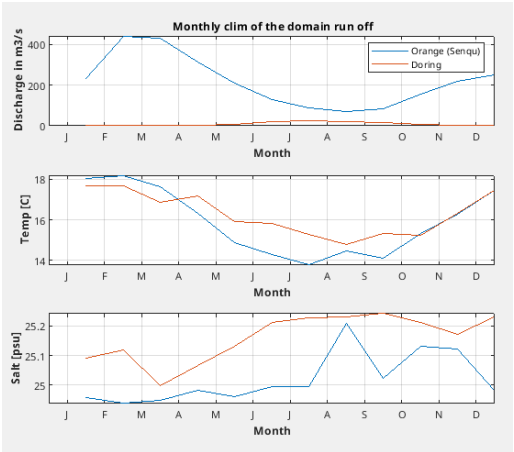
→ Execute the Matlab script make_runoff:

```
>> make_runoff
>>
```

➥ The script will estimate the right location of the river mouth in your croco_grid, regarding the direction and orientation you defined.

For the BENGUELA_LR test case, we have 2 rivers detected, Orange and Doring. We recommend to define them as zonal (0) and oriented from East to West (-1).





➥ At the end of the execution, the script provides the lines to enter in the psource_ncfile section (line 262) of the **croco_inter.in** file (position, direction, …):

```
Line to enter in the croco.in file in the psource_ncfile section :
-----------------------------------------------------------------
psource_ncfile:   Nsrc  Isrc  Jsrc  Dsrc qbardir  Lsrc  Tsrc   runoff file name
                            CROCO_FILES/croco_runoff.nc(.#nestlevel)
                2
                       25  34  0  -1   T  T   15.8982 25.0203
                       31  19  0  -1   T  T   16.2873 25.16
-----------------------------------------------------------------
```

➥ The NetCDF river runoff input file will be stored in the NetCDF file **CROCO_FILES/croco_runoff.nc**

→ You are finished with step 2. You can exit Matlab

```
>> exit
>>
```

RIVER FORCING

STEP 3: Compiling CROCO model with runoffs

→ To compile CROCO on Lengau, you need to copy my **jobcomp_lengau** into your **Run** directory:

```
cp /home/apps/chpc/earth/CROCCO_Workshop/CROCO_TRAINING_Basic/3_Some_files/jobcomp_lengau .
```

→ Edit the CROCO parameter file **param.h**:

```
[login@cnode0220 Run]$ nedit param.h &
[login@cnode0220 Run]$
```

➥ Check the parameters for the **parallelisation**. We want: NP_XI=1, NP_ETA=4

→ Edit the **cppdefs.h**:

```
[login@cnode0220 Run]$ nedit cppdefs.h &
[login@cnode0220 Run]$
```

➥ Activate MPI parallelization at line 74 (**# define MPI**).

➥ Activate the rivers: **#define PSOURCE** (line 293),

using variable river transport in croco_runoff.nc: **#define PSOURCE_NCFILE** (line 294), with seasonally varying T, S concentrations: **#define PSOURCE_NCFILE_TS** (line 296)

PSOURCE	Activate point sources (rivers)
ANA_PSOURCE	use analytical vertical profiles for point sources (set in set_global_definitions.h)
PSOURCE_NCFILE	Read variable river transports in netcdf file
PSOURCE_NCFILE_TS	Read variable river concentration in netcdf file

→ Compile CROCO using the script **jobcomp_lengau**:

```
[login@cnode0220 Run]$ ./jobcomp_lengau
[login@cnode0220 Run]$ ls
```

STEP 4: Running CROCO with runoffs

→ To launch your simulation, you need to copy my **run_croco.pbs** file into your **Run** directory:

```
cp /home/apps/chpc/earth/CROCCO_Workshop/CROCO_TRAINING_Basic/3_Some_files/run_croco.pbs .
```

→ Edit the script **run_croco.pbs**:

```
[login@cnode0220 Run]$ nedit run_croco.pbs &
[login@cnode0220 Run]$
```

➥ Check the PBS scheduler parameters (`mpiprocs` consistent with **param.h**) and email.

➥ Check the path of your **Run** working directory.

→ Edit **croco_inter.in** and fix the the psource_ncfile section (line 262) using the parameters written at the end of the execution of the Matlab script make_runoff (see **STEP 2**):

```
psource_ncfile:   Nsrc   Isrc   Jsrc   Dsrc qbardir   Lsrc   Tsrc    runoff file name
                            CROCO_FILES/croco_runoff.nc
                  2
                         25   34   0   -1    T   T    15.8982 25.0203
                         31   19   0   -1    T   T    16.2873 25.16
```

→ Launch the model with the runoffs activated:

```
[login@cnode0220 Run]$ qsub run_croco.pbs
[login@cnode0220 Run]$
```


➥ The model log (*.out) and outputs (*.nc) will be stored in the directory: **./SCRATCH**

STEP 5: Visualising model outputs

→ This can be done with **MATLAB**

→ Launch **`matlab -nodesktop`** (or the alias **`mat`**) to visualize your outputs using croco_gui:

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

STEP 6: Exiting

→ When you are done, exit Matlab, exit the interactive node, and logout from lengau.

STEP 7: Check List

→ Here is the list of the essential commands that you must execute during this hands-on session. The following table can help you confirm that you have executed all of them:

		Commands	✌
STEP 1	1	ssh -X login@lengau.chpc.ac.za	
	2	qsubi1	
	3	cd lustre/CROCO/croco-v2.0.1	
	4	nedit create_config.bash &	
	5	./create_config.bash	
	6	cd Run	
STEP 2	1	matlab -nodesktop	
	2	start	
	3	make_grid; make_forcing; make_bry; make_ini	
	4	edit make_runoff	
	5	make_runoff	
	6	exit	
STEP 3	1	cp /home/apps/chpc/earth/CROCCO_Workshop/CROCO_TRAINING_Basic/3_Some_files/jobcomp_lengau .	
	2	nedit param.h &	
	3	nedit cppdef.h &	
	4	./jobcomp_lengau	
STEP 5	1	cp /home/apps/chpc/earth/CROCCO_Workshop/CROCO_TRAINING_Basic/3_Some_files/run_croco.pbs	
	2	nedit run_croco.pbs &	
	3	nedit croco_inter.in &	
	4	qsub run_croco.pbs	
STEP 6	1	matlab -nodesktop	
	2	start	
	3	croco_gui	
STEP 7	1	exit	
	2	exit	
	3	exit	