

Activity 1 - Run an idealized ocean gyre

1. Get the model

- Download from <http://igula.fr/ModNum/croco.tar.gz> or from [/forums/public/pub/Data/croco.tar.gz](http://forums.public/pub/Data/croco.tar.gz)
- Put it in a folder of your choice: [for example `~/ModNum/`]
- untar it: `tar xf croco.tar.gz`
- the croco source code is in the folder `~/ModNum/croco/OCEAN`

2. Compile the model

- Create a folder where you will run the model [for example `~/ModNum/case1`]
- we need to edit the following files: `jobcomp`, `cppdefs.h`, `param.h`, `croco.in` so copy them into the folder you just created:

```
cp ~/ModNum/croco/OCEAN/jobcomp ~/ModNum/case1/  
cp ~/ModNum/croco/OCEAN/cppdefs.h ~/ModNum/case1/  
cp ~/ModNum/croco/OCEAN/param.h ~/ModNum/case1/  
cp ~/ModNum/croco/TEST_CASES/croco.in.Basin ~/ModNum/case1/croco.in
```

- the model needs a **fortran compiler and compatible netcdf libraries**:

- If you are working with **Linux on IUEM computers**:

- ★1. unload everything: `module purge`
- ★2. then load intel compilers and netcdf library
`module load intel/12.1 netcdf/4.1.3-intel12`

- If you are working with **MacOS**:

- ★ you need to have gcc and netcdf installed (using **homebrew** or **macports**): `brew install gcc; brew install netcdf`
- ★ you may need to edit the jobcomp to specify where to find the netcdf libraries (*nf-config doesn't work with homebrew version of netcdf-fortran*)

```
NETCDFLIB="-L/usr/local/lib -lnetcdff -lnetcdf"  
NETCDFINC="-I/usr/local/include"  
#NETCDFLIB=$(nf-config --flibs)  
#NETCDFINC=-I$(nf-config --includedir)
```

- edit the `jobcomp` to specify the location of the source code:

```
SOURCE=../croco/OCEAN
```

- edit the `cppdefs.h` and choose the predefined test case **Basin**:

```
#define BASIN  
  
...  
  
#undef REGIONAL
```

- Compile the code: `./jobcomp`

3. Run the model

- run the model: `./croco croco.in`
- look at the output: `ncview basin_his.nc`

4. Modify the namelist (`croco.in`)

See https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.08.run.html

- Modify the namelist (`croco.in`) to:
 - a. make the model run for 20 years (*you can approximate 1 year = 360 days*)
 - b. output of history files every 30 days
 - c. outputs of variables averaged over 5 years

- Note that you also need to modify the `cppdefs.h` to output averaged files:

```
#elif defined BASIN
/*
!               Basin      Example
!               =====
*/
# define AVERAGES
```

5. Run the model using openMP

- we need to edit the file `cppdefs.h`,
- find the part of the file where the BASIN case is defined (`#elif defined BASIN`) and change `# undef OPENMP` into `# define OPENMP`

```
#elif defined BASIN
/*
!               Basin      Example
!               =====
*/
# define OPENMP
# undef MPI
```

- edit the `param.h` file to choose the number of processors (NPP) you want to use:

```
#elif defined OPENMP
    parameter (NPP=8)
```

- reCompile the code: `./jobcomp`
- define the number of threads to use for your computer (the number should be the same than NPP defined in the `param.h` file), type in your terminal:
 - in bash: `export OMP_NUM_THREADS=8`
 - in csh/tcsh: `setenv OMP_NUM_THREADS 8`
- run the model: `./croco croco.in`

- What can you tell about the configuration? What is the forcing? What is the Coriolis parameter? What is the boundary condition? What is the bottom condition?

- What can you tell about the circulation? Is it similar to Stommel's gyre?

(<http://jgula.fr/ModNum/Stommel48.pdf>)