CROCO – training 2023

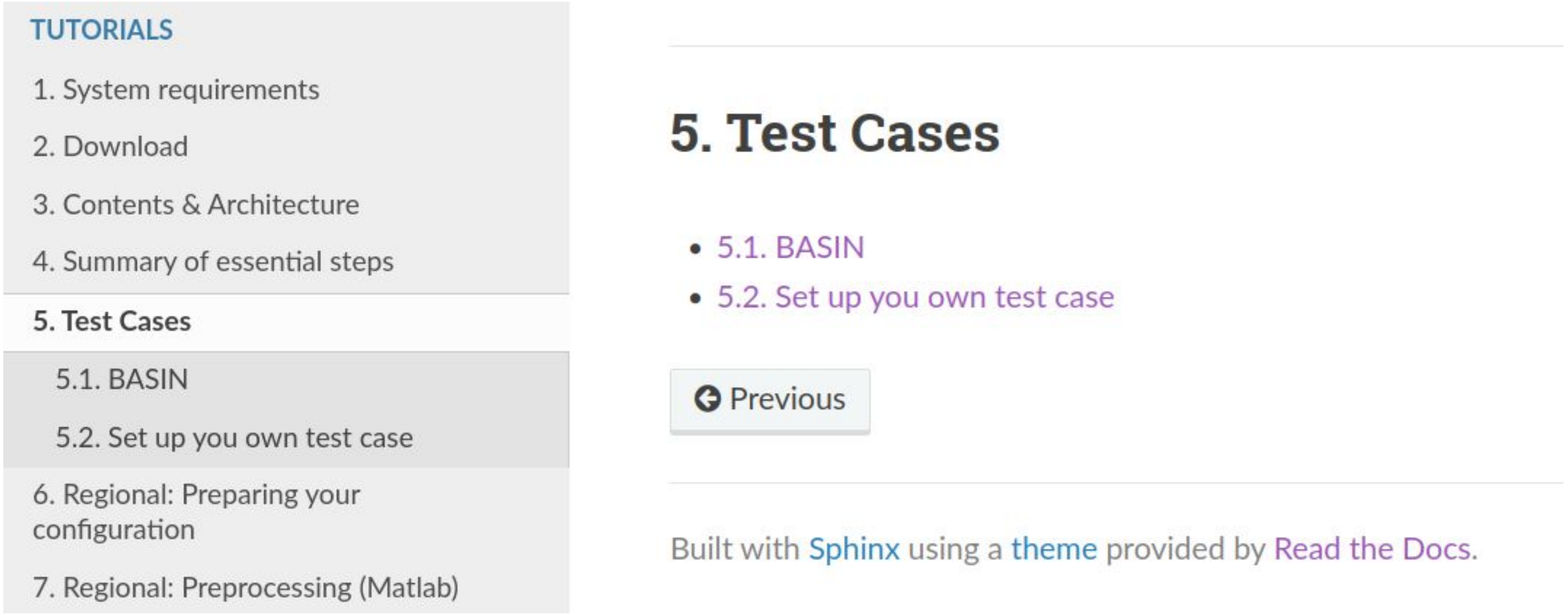
TP tests cases

Coastal and Regional Ocean COmmunity model

Camille Mazoyer, IRD, camille.mazoyer@ird.fr
Nicolas Ducousso, SHOM, nicolas.ducousso@shom.fr

Documentation

CROCO
Coastal and Regional Ocean COmmunity model



TUTORIALS

5. Test Cases

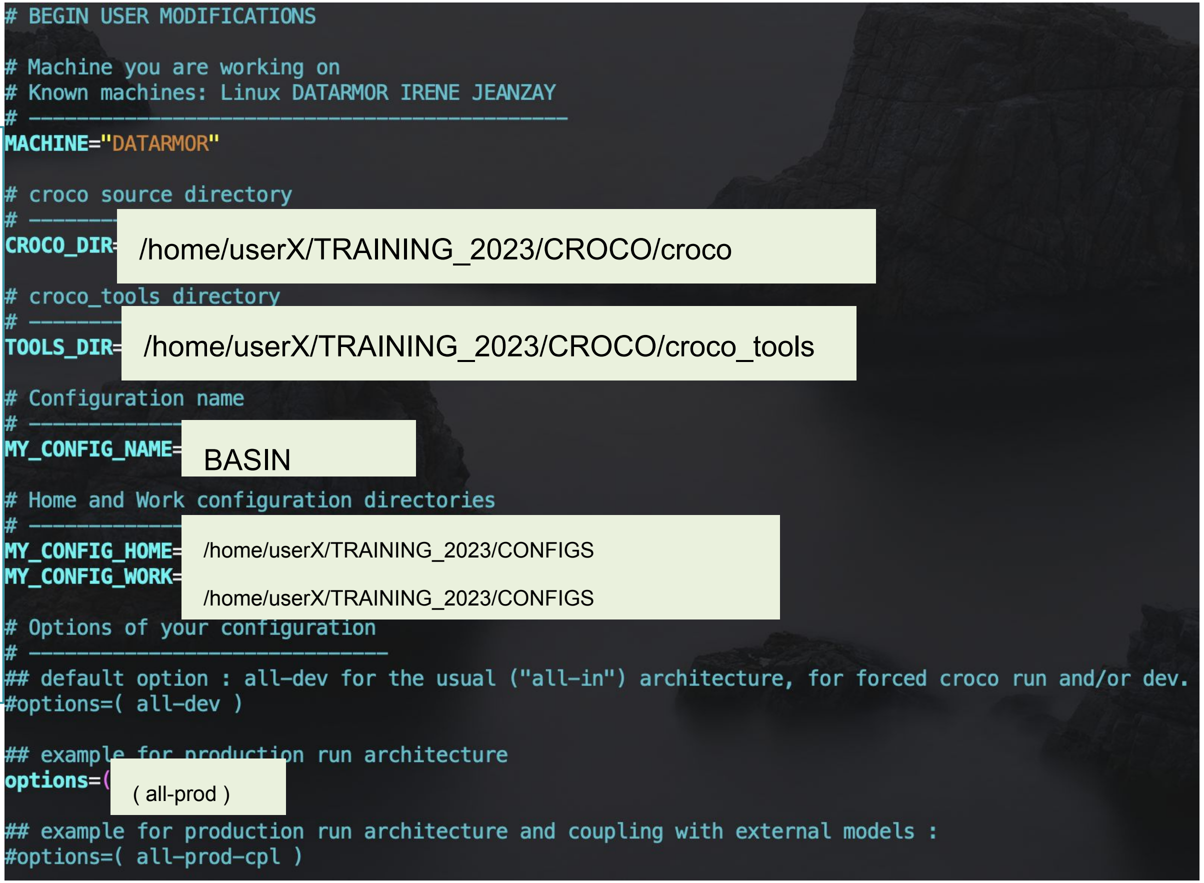
- 5.1. BASIN
- 5.2. Set up you own test case

Previous

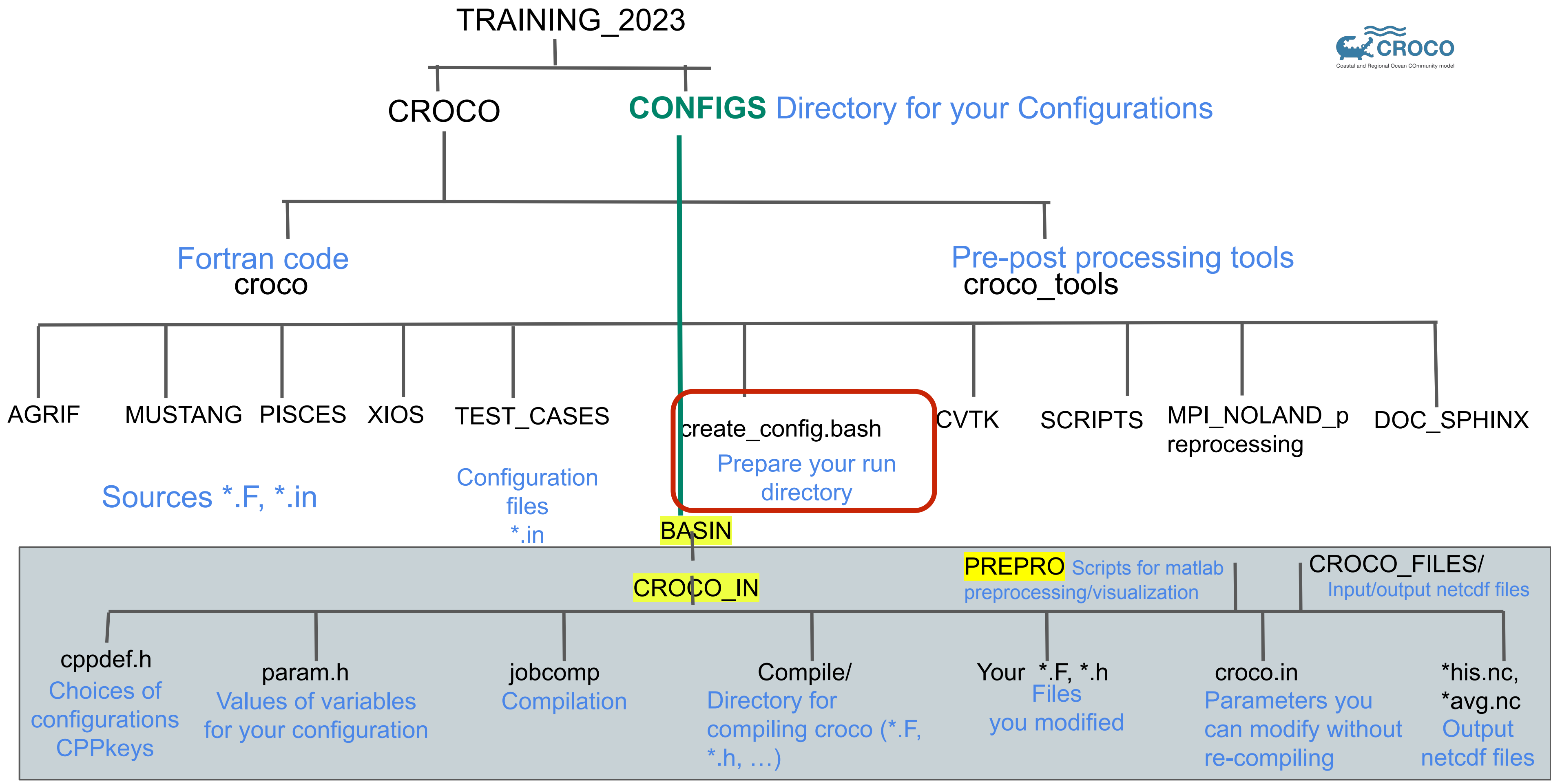
Built with Sphinx using a theme provided by Read the Docs.

Create config directory for basin testcase

```
ssh -X userX@croco
mkdir TRAINING_2023
cd TRAINING_2023
cp -r
/home/COMMONDATA/codes/CROCO .
cp CROCO/croco/create_config.bash .
mkdir CONFIGS
vim create_config.bash
./create_config.bash
```

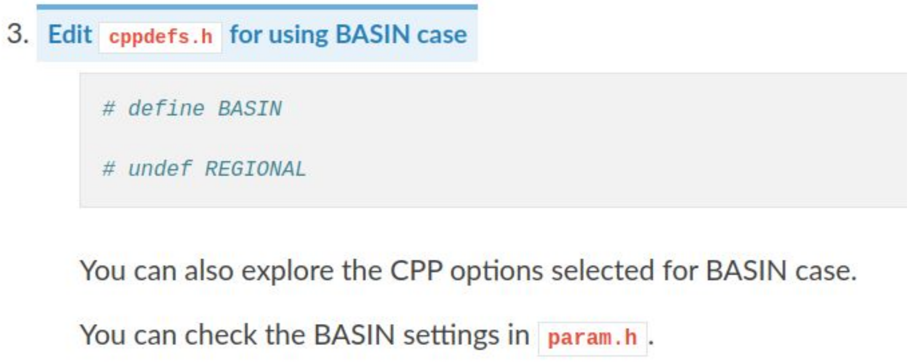


```
# BEGIN USER MODIFICATIONS

# Machine you are working on
# Known machines: Linux DATARMOR IRENE JEANZAY
# ---------------------------------------------
MACHINE="DATARMOR"

# croco source directory
# ---------
CROCO_DIR= /home/userX/TRAINING_2023/CROCO/croco

# croco_tools directory
# ---------
TOOLS_DIR= /home/userX/TRAINING_2023/CROCO/croco_tools

# Configuration name
# -------------
MY_CONFIG_NAME= BASIN

# Home and Work configuration directories
# -------------
MY_CONFIG_HOME= /home/userX/TRAINING_2023/CONFIGS
MY_CONFIG_WORK= /home/userX/TRAINING_2023/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 )
```


TRAINING_2023
CROCO
CONFIGS Directory for your Configurations
Fortran code
croco
Pre-post processing tools
croco_tools
AGRIF
MUSTANG
PISCES
XIOS
TEST_CASES
Sources *.F, *.in
Configuration files *.in
create_config.bash
Prepare your run directory
CVTK
SCRIPTS
MPI_NOLAND_preprocessing
DOC_SPHINX
BASIN
CROCO_IN
PREPRO Scripts for matlab preprocessing/visualization
CROCO_FILES/
Input/output netcdf files
cppdef.h
Choices of configurations CPPkeys
param.h
Values of variables for your configuration
jobcomp
Compilation
Compile/
Directory for compiling croco (*.F, *.h, …)
Your *.F, *.h Files you modified
croco.in
Parameters you can modify without re-compiling
*his.nc, *avg.nc
Output netcdf files

Basin test case

Cd TRAINING_2023/CONFIGS/BASIN/CROCO_IN



3. **Edit `cppdefs.h` for using BASIN case**

```
# define BASIN

# undef REGIONAL
```

You can also explore the CPP options selected for BASIN case.

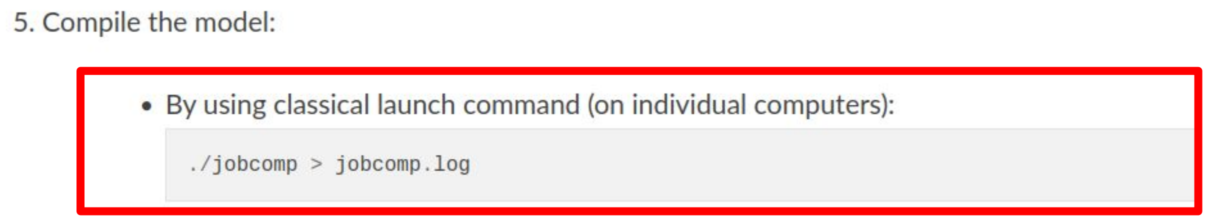
You can check the BASIN settings in `param.h`.

Compilation

Coastal and Regional Ocean COmmunity model

Cd TRAINING_2023/CONFIGS/BASIN

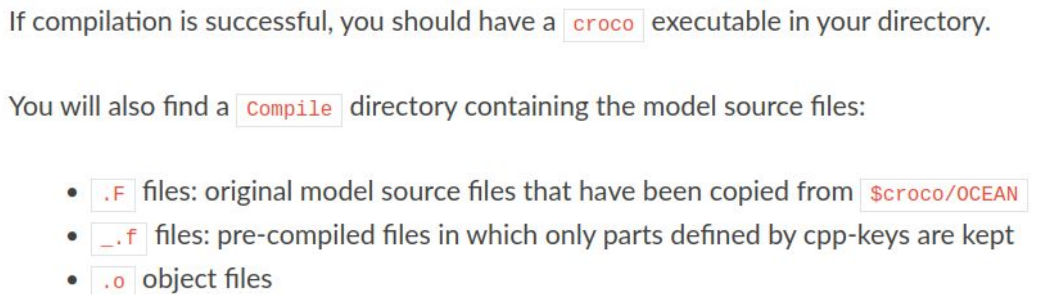
Jobcom should work automatically in Seolane : do not modify it



5. Compile the model:

- By using classical launch command (on individual computers):

```
./jobcomp > jobcomp.log
```



If compilation is successful, you should have a `croco` executable in your directory.

You will also find a `Compile` directory containing the model source files:

- `.F` files: original model source files that have been copied from `$croco/OCEAN`
- `_.f` files: pre-compiled files in which only parts defined by cpp-keys are kept
- `.o` object files

Execution and visualization

6. **Copy the namelist input file for BASIN case:**

```
Cp ~/TRAINING_2023/CROCO/croco/TEST_CASES/croco.in.Basin .
```

Eventually edit it.

7. **Run the model:**

```
./croco croco.in.Basin
```

If your run is successful you should obtain the following files:

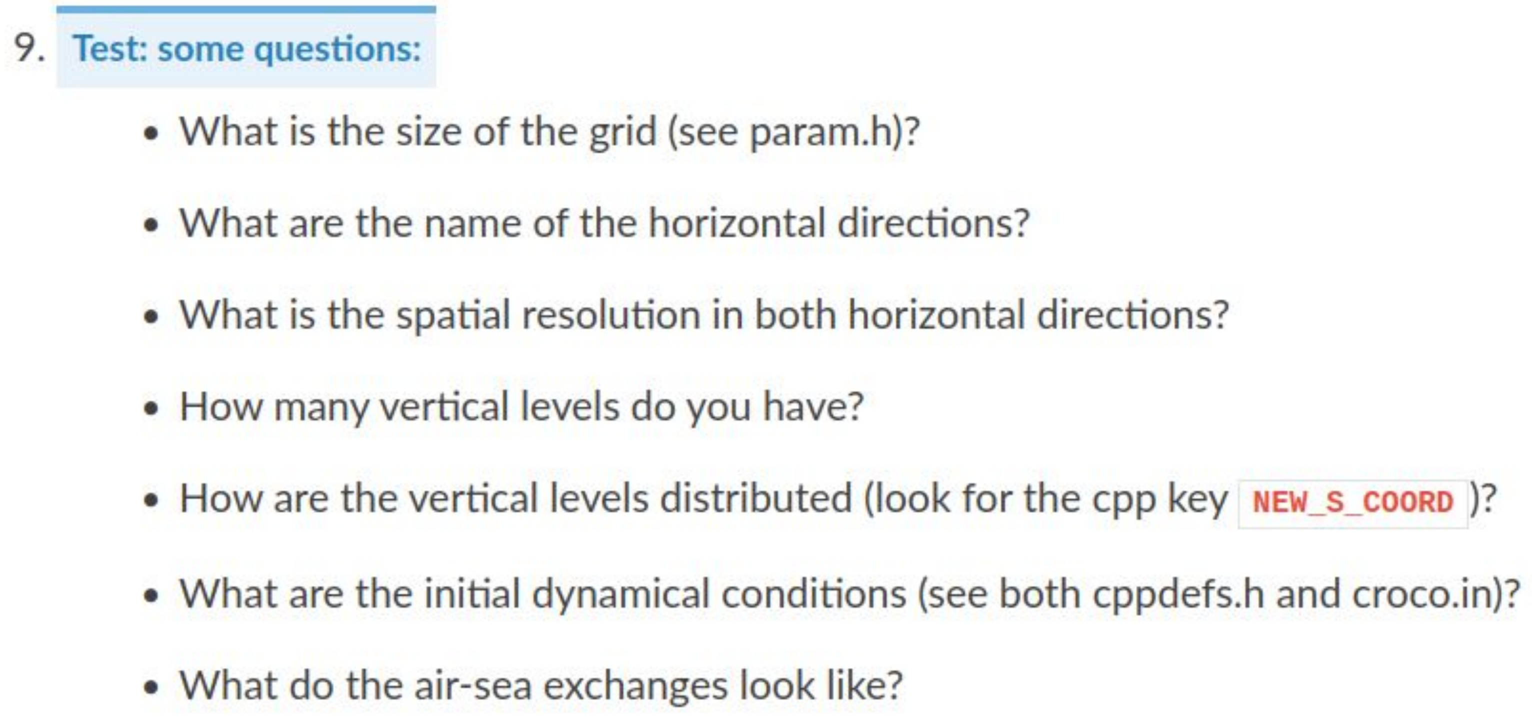
```
basin_rst.nc # restart file
basin_his.nc # instantaneous output file
```

8. **Have a look at the results:**

```
ncview basin_his.nc
```

Questions

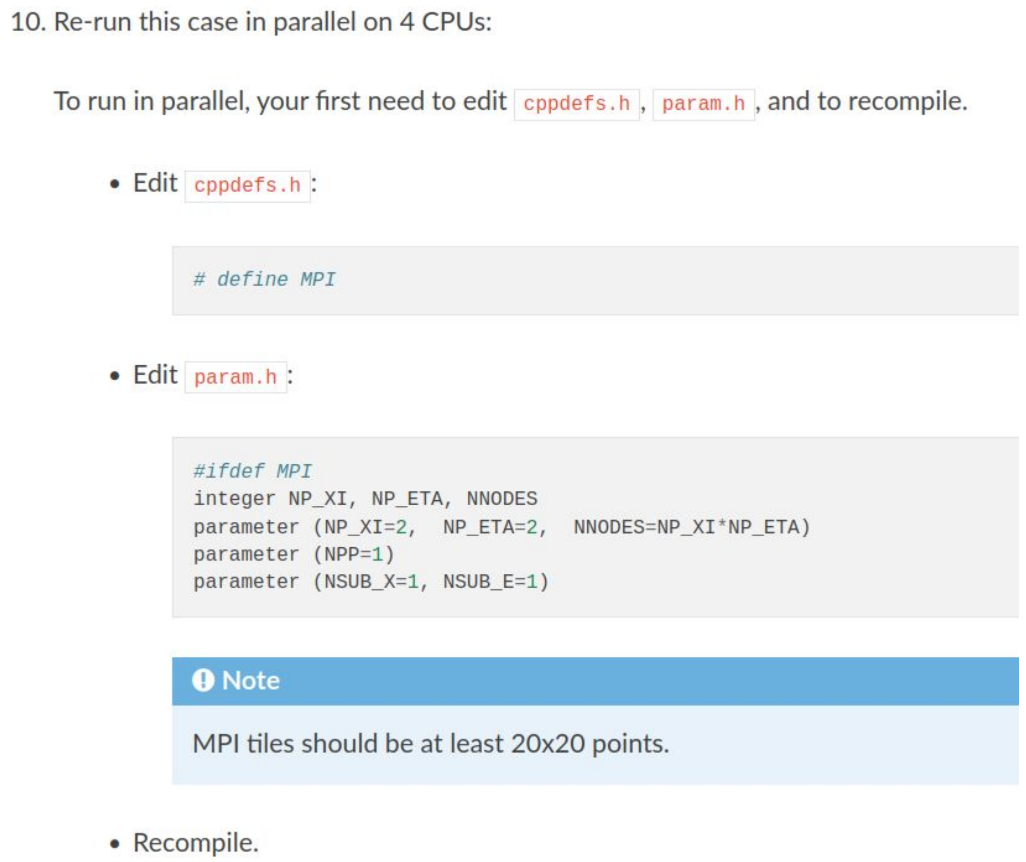
Cd TRAINING_2023/CONFIGS/BASIN



9. **Test: some questions:**
    - What is the size of the grid (see param.h)?
    - What are the name of the horizontal directions?
    - What is the spatial resolution in both horizontal directions?
    - How many vertical levels do you have?
    - How are the vertical levels distributed (look for the cpp key `NEW_S_COORD`)?
    - What are the initial dynamical conditions (see both cppdefs.h and croco.in)?
    - What do the air-sea exchanges look like?

Parallel compilation and execution

Cd TRAINING_2023/CONFIGS/BASIN/CROCO_IN



10. Re-run this case in parallel on 4 CPUs:

To run in parallel, your first need to edit `cppdefs.h`, `param.h`, and to recompile.

- Edit `cppdefs.h`:

```
# define MPI
```

- Edit `param.h`:

```
#ifdef MPI
integer NP_XI, NP_ETA, NNODES
parameter (NP_XI=2,  NP_ETA=2,  NNODES=NP_XI*NP_ETA)
parameter (NPP=1)
parameter (NSUB_X=1, NSUB_E=1)
```

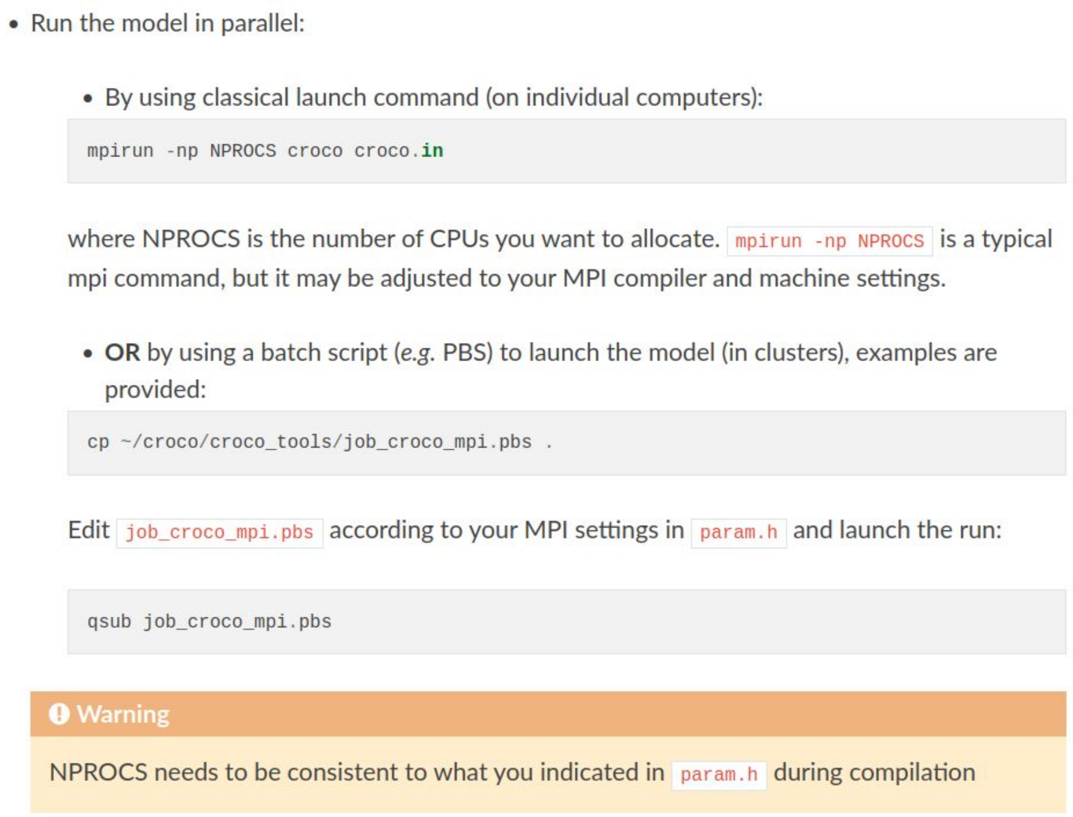
Note

MPI tiles should be at least 20x20 points.

- Recompile.

Parallel compilation and execution

Cd TRAINING_2023/CONFIGS/BASIN/CROCO_IN



- Run the model in parallel:

  - By using classical launch command (on individual computers):

```
mpirun -np NPROCS croco croco.in
```

where NPROCS is the number of CPUs you want to allocate. `mpirun -np NPROCS` is a typical mpi command, but it may be adjusted to your MPI compiler and machine settings.

  - **OR** by using a batch script (*e.g.* PBS) to launch the model (in clusters), examples are provided:

```
cp ~/croco/croco_tools/job_croco_mpi.pbs .
```

Edit `job_croco_mpi.pbs` according to your MPI settings in `param.h` and launch the run:

```
qsub job_croco_mpi.pbs
```

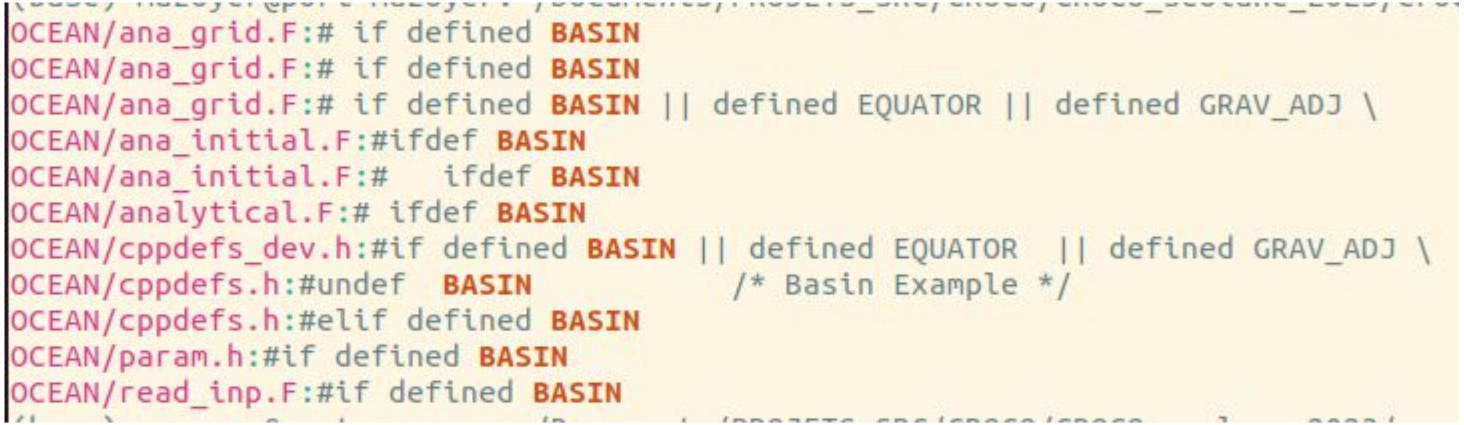
Warning

NPROCS needs to be consistent to what you indicated in `param.h` during compilation

CPP key BASIN inside the source code

Cd TRAINING_2023/CONFIGS/BASIN/CROCO_IN

grep BASIN Compile/*



```
OCEAN/ana_grid.F:# if defined BASIN
OCEAN/ana_grid.F:# if defined BASIN
OCEAN/ana_grid.F:# if defined BASIN || defined EQUATOR || defined GRAV_ADJ \
OCEAN/ana_initial.F:#ifdef BASIN
OCEAN/ana_initial.F:#   ifdef BASIN
OCEAN/analytical.F:# ifdef BASIN
OCEAN/cppdefs_dev.h:#if defined BASIN || defined EQUATOR  || defined GRAV_ADJ \
OCEAN/cppdefs.h:#undef  BASIN               /* Basin Example */
OCEAN/cppdefs.h:#elif defined BASIN
OCEAN/param.h:#if defined BASIN
OCEAN/read_inp.F:#if defined BASIN
```

CPP key BASIN inside the source code

Cd TRAINING_2023/CONFIGS/BASIN/CROCO_IN

```
grep -i BASIN Compile/*
```

Key activation and clefs CPP liées
cppdef.h

Parameters related to the key
param.h

Grid
ana_grid.F

Initialisation: U, V, …
ana_initial.F

Analytic functions or parameters: bathymetry, coriolis, …
analytical.F

Cppkey activated (hidden)
cppdef_dev.h

What *.in to read automatically
read_inp.F

CPP key BASIN inside the source code

Cd TRAINING_2023/CONFIGS/BASIN/CROCO_IN

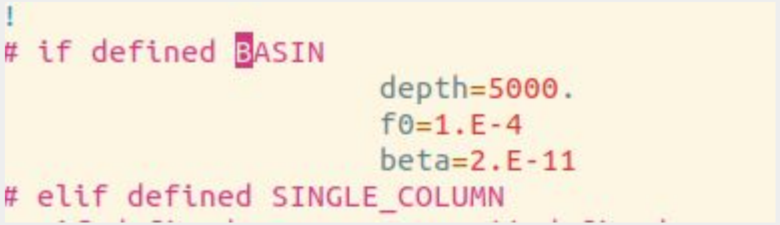
grep -i BASIN Compile/*

Key activation and clefs CPP liées
cppdef.h

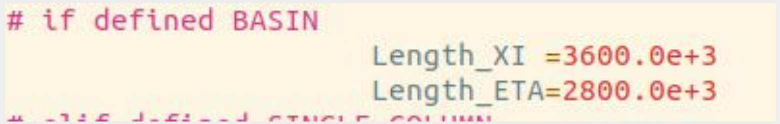
Parameters related to the key
param.h

Grid
ana_grid.F

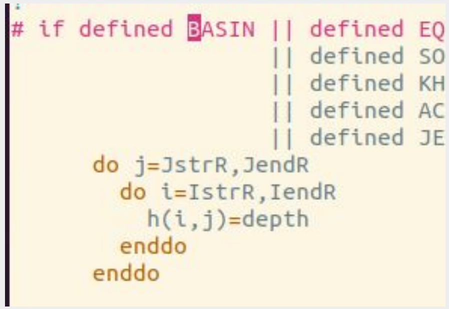
ana_grid.F



```
!
# if defined BASIN
                      depth=5000.
                      f0=1.E-4
                      beta=2.E-11
# elif defined SINGLE_COLUMN
```



```
# if defined BASIN
                     Length_XI =3600.0e+3
                     Length_ETA=2800.0e+3
```



```
# if defined BASIN || defined EQ
                   || defined SO
                   || defined KH
                   || defined AC
                   || defined JE
      do j=JstrR,JendR
        do i=IstrR,IendR
          h(i,j)=depth
        enddo
      enddo
```

CPP key BASIN inside the source code

Cd TRAINING_2023/CONFIGS/BASIN/CROCO_IN

```
grep -i BASIN Compile/*
```

Key activation and clefs CPP liées
cppdef.h

Parameters related to the key
param.h

Grid
ana_grid.F

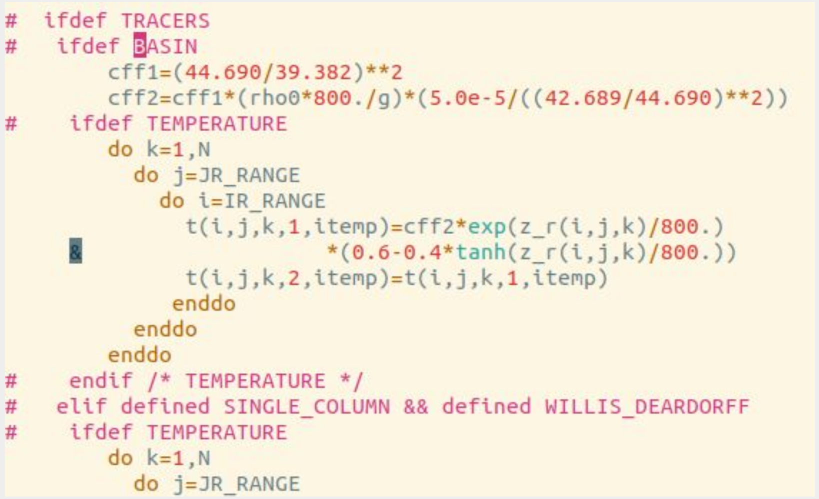
Initialisation: U, V, …
ana_initial.F

A
c
a

Cp
cp

Wh
rea

ana_initial.F



```
#  ifdef TRACERS
#   ifdef BASIN
        cff1=(44.690/39.382)**2
        cff2=cff1*(rho0*800./g)*(5.0e-5/((42.689/44.690)**2))
#    ifdef TEMPERATURE
        do k=1,N
          do j=JR_RANGE
            do i=IR_RANGE
              t(i,j,k,1,itemp)=cff2*exp(z_r(i,j,k)/800.)
     &                   *(0.6-0.4*tanh(z_r(i,j,k)/800.))
              t(i,j,k,2,itemp)=t(i,j,k,1,itemp)
             enddo
          enddo
        enddo
#    endif /* TEMPERATURE */
#   elif defined SINGLE_COLUMN && defined WILLIS_DEARDORFF
#    ifdef TEMPERATURE
        do k=1,N
          do j=JR_RANGE
```

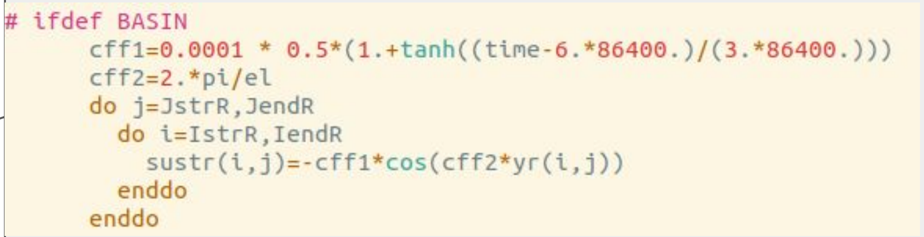
CPP key BASIN inside the source code

Cd TRAINING_2023/CONFIGS/BASIN/CROCO_IN

grep -i BASIN Compile/*

analytical.F

Stress du vent



```
# ifdef BASIN
      cff1=0.0001 * 0.5*(1.+tanh((time-6.*86400.)/(3.*86400.)))
      cff2=2.*pi/el
      do j=JstrR,JendR
        do i=IstrR,IendR
          sustr(i,j)=-cff1*cos(cff2*yr(i,j))
        enddo
      enddo
```

coriolis, …
analytical.F

Key activation and clefs CPP liées
cppdef.h

Parameters related to the key
param.h

Cppkey activated (hidden)
cppdef_dev.h

What *.in to read automatically
read_inp.F