CROCO – training 2022

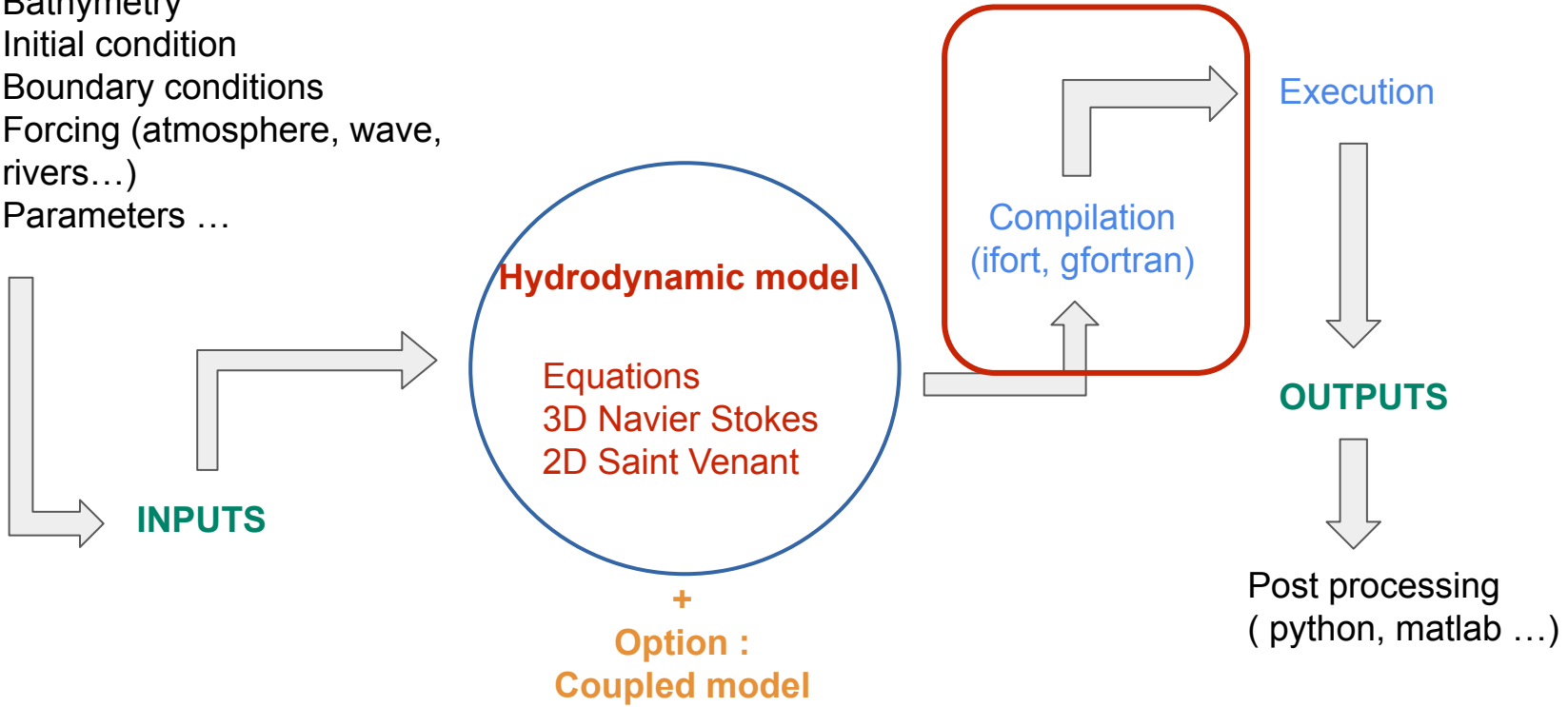
Architecture CROCO

CROCO
Coastal and Regional Ocean COmmunity model

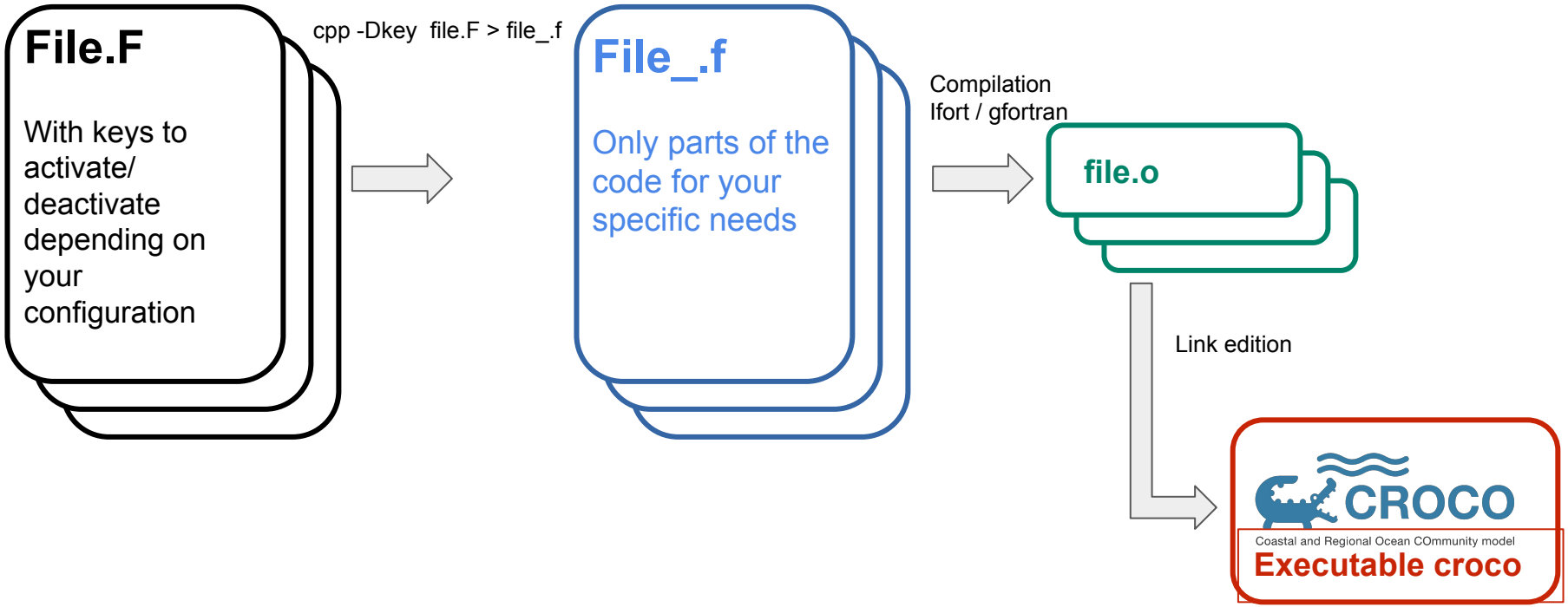
Camille Mazoyer, IRD, camille.mazoyer@ird.fr

How does it work?

Préprocessing
- Bathymetry
- Initial condition
- Boundary conditions
- Forcing (atmosphere, wave, rivers…)
- Parameters …



Compilation in 2 steps



Compilation with CPP keys

Coastal and Regional Ocean COmmunity model

file.F → cpp -DCAS1 file.F > file_.f → file_.f

```
PROGRAM test_clefCPP

IMPLICIT NONE
integer :: a,b


! b va etre remplace par 2*a dans le code dans le fichier .f90
#define b 2*a


a=0
#ifdef CAS1
        a=1
        WRITE(*,*)''
        WRITE(*,*)'clef activee=CAS1'
        WRITE(*,*)'a=',a
        WRITE(*,*)''
#elif CAS2
        a=2
        WRITE(*,*)''
        WRITE(*,*)'clef activee=CAS2'
        WRITE(*,*)'a=',a
        WRITE(*,*)''
#endif

WRITE(*,*)''
write(*,*) 'b est remplace par 2*a dans le code! b=',b
WRITE(*,*)''

END PROGRAM test_clefCPP
```

```
PROGRAM test_clefCPP

IMPLICIT NONE
integer :: a,b


! b va etre remplace par 2*a dans le code dans le fichier .f90


a=0

    a=1
    WRITE(*,*)''
    WRITE(*,*)'clef activee=CAS1'
    WRITE(*,*)'a=',a
    WRITE(*,*)''

empty


WRITE(*,*)''
write(*,*) 'b est remplace par 2*a dans le code! b=:',2*a
WRITE(*,*)''

END PROGRAM test_clefCPP
```

Compilation with CPP keys

CROCO
Coastal and Regional Ocean COmmunity model

file.F

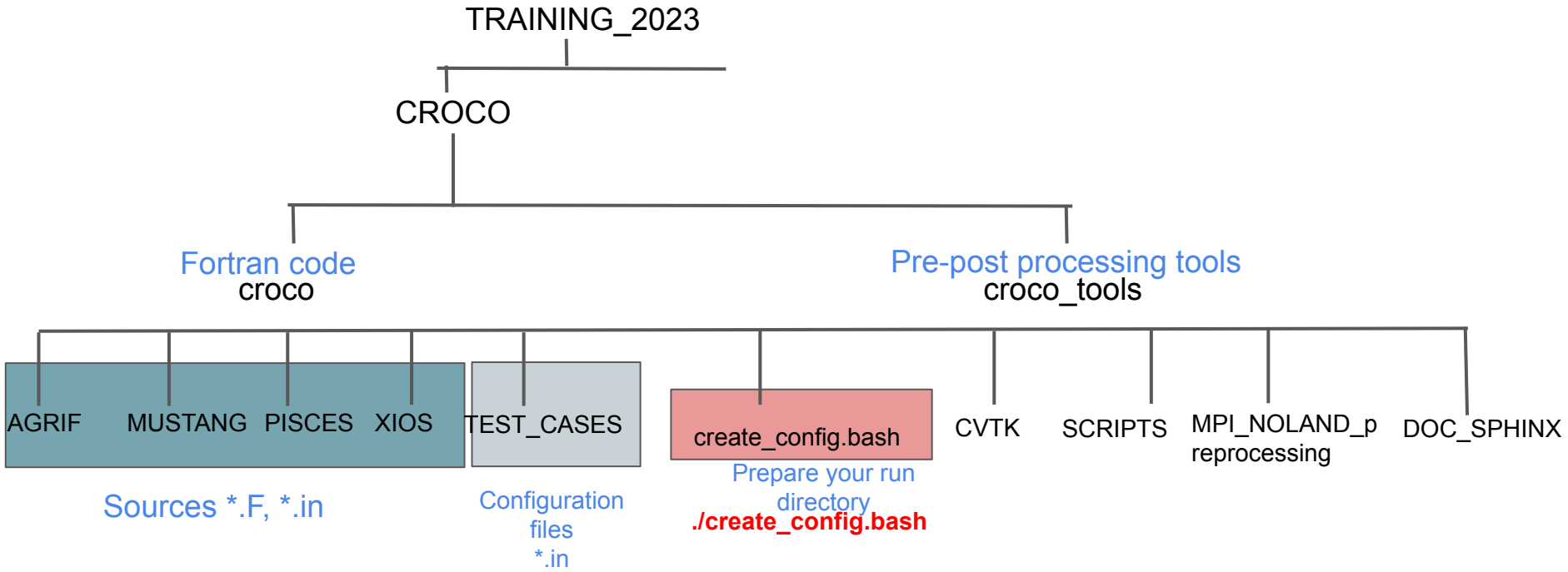
```
!
#include "cppdefs.h"
      subroutine step()
      implicit none
#include "param.h"
#include "scalars.h"
#include "zoom.h"
#include "grid.h"
#include "coupling.h"
#include "ocean3d.h"
#include "ocean2d.h"
#include "mpi_cpl.h"
#ifdef MUSTANG
# include "coupler_define_MUSTANG.h"
#endif

#ifdef AGRIF
      IF (agrif_fixed().NE.sortedint(nbtimes)) return
      nbtimes = nbtimes + 1
#endif

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#ifdef SOLVE3D

#if defined OA_COUPLING || defined OW_COUPLING
!--Get phase of OASIS3
       if ( (iif==-1).and.(oasis_time>=0).and.(nbstep3d
!         print *, 'oasis time before get = ', oasis_ti
          call cpl_prism_get (oasis_time)
#ifdef AGRIF
       else if ((.not.agrif_root()).and.(iif==-1).and.
     &          (oasis_time>=0).and.(nbtimes<=Agrif_Max
     &          (nbprttime<nbmaxprttime)) then
```

file_.f

```
      real zeta(GLOBAL_2D_ARRAY,4)
      real ubar(GLOBAL_2D_ARRAY,4)
      real vbar(GLOBAL_2D_ARRAY,4)
      common /ocean_zeta/zeta
      common /ocean_ubar/ubar
      common /ocean_vbar/vbar
```

Each include is paste in file_.f

Architecture of CROCO sources

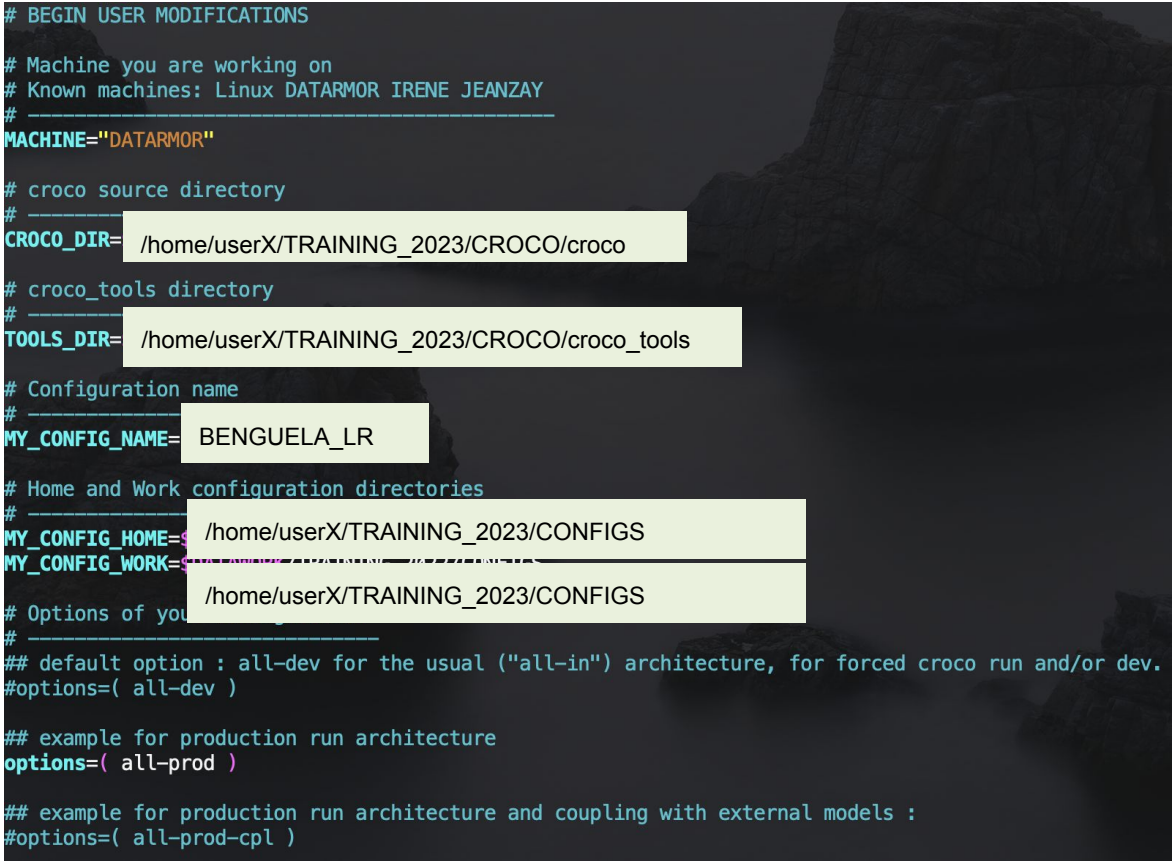


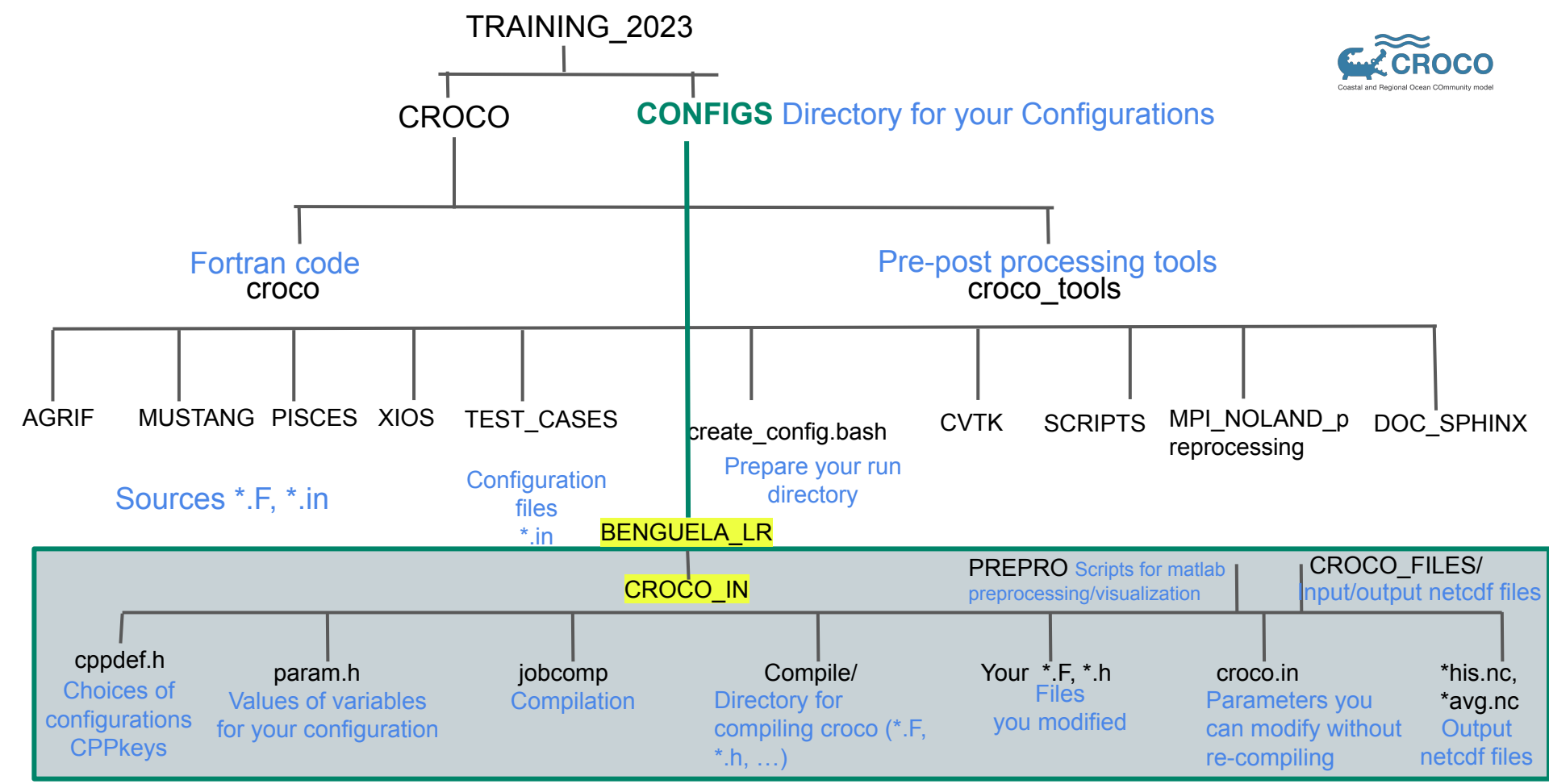
create _config.bash

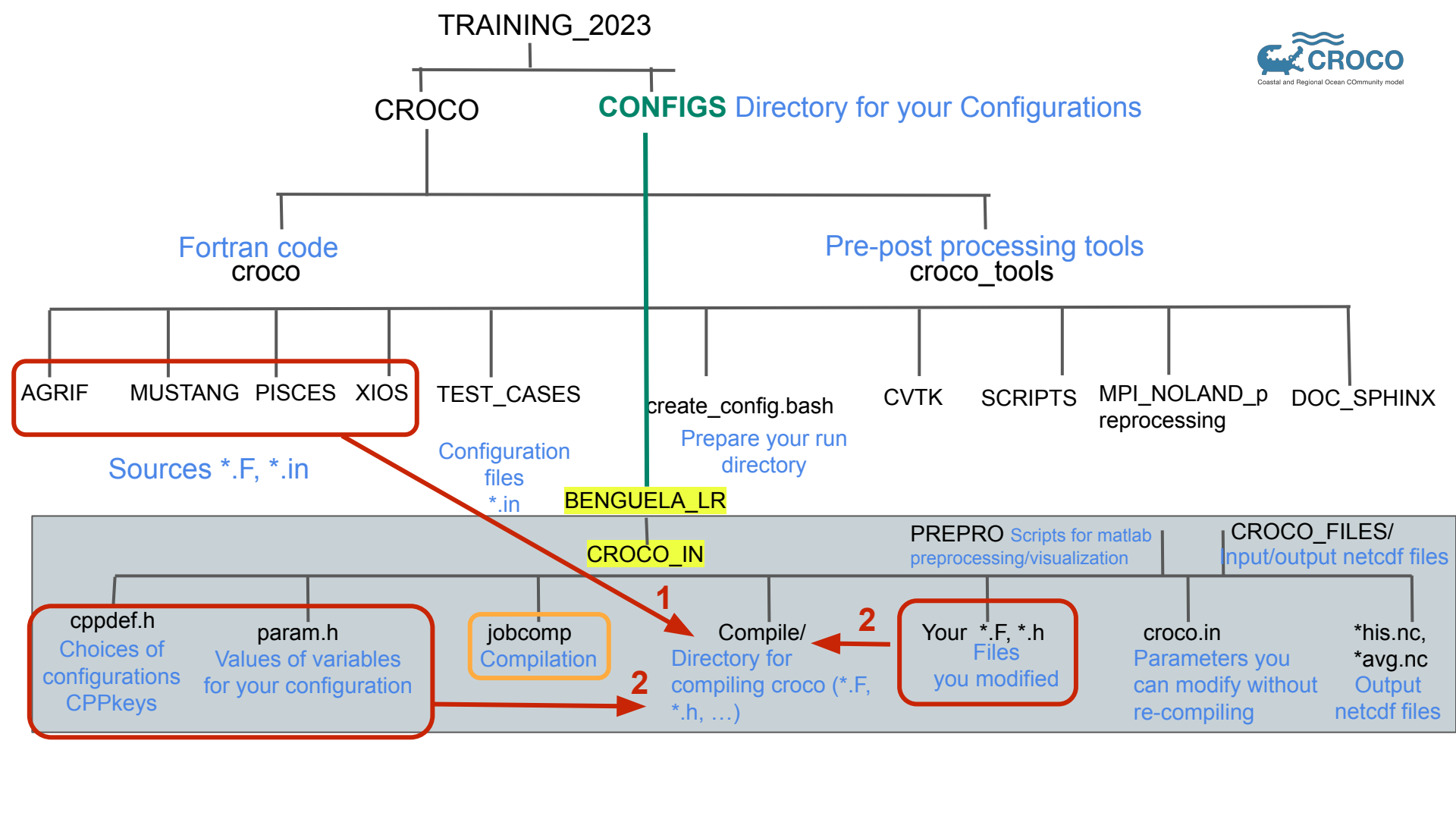
cp croco/create_config.bash .

=> 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 »




TRAINING_2023
CROCO
CONFIGS Directory for your Configurations
Fortran code
croco
Pre-post processing tools
croco_tools
AGRIF
MUSTANG
PISCES
XIOS
TEST_CASES
create_config.bash
CVTK
SCRIPTS
MPI_NOLAND_p
reprocessing
DOC_SPHINX
Sources *.F, *.in
Configuration
files
*.in
Prepare your run
directory
BENGUELA_LR
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


TRAINING_2023
CROCO
CONFIGS Directory for your Configurations
Fortran code
croco
Pre-post processing tools
croco_tools
AGRIF
MUSTANG
PISCES
XIOS
TEST_CASES
create_config.bash
CVTK
SCRIPTS
MPI_NOLAND_p
reprocessing
DOC_SPHINX
Sources *.F, *.in
Configuration
files
*.in
Prepare your run
directory
BENGUELA_LR
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
1
Compile/
Directory for
compiling croco (*.F,
*.h, …)
2
2
Your *.F, *.h
Files
you modified
croco.in
Parameters you
can modify without
re-compiling
*his.nc,
*avg.nc
Output
netcdf files

Execution, scripts

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
CROCO_IN
DATA
myenv_mypath.sh
myjob.sh
mynamelist.sh
PREPRO
README_coupling_tools
run_croco.bash
run_croco_forecast.bash
run_croco_inter.bash
SCRIPTS_TOOLBOX
submitjob.sh
```