

# CROCO – training 2024

## CROCO Architecture

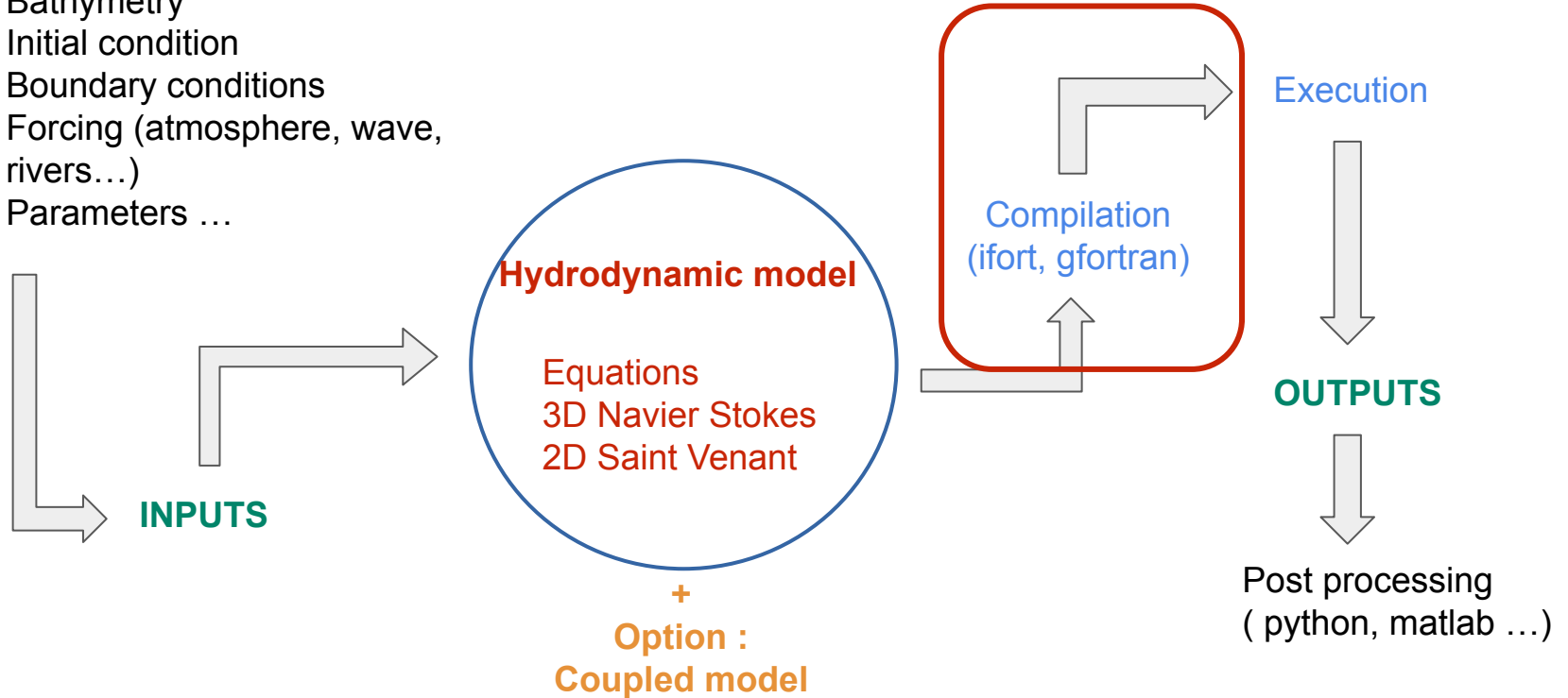


Camille Mazoyer, IRD, [camille.mazoyer@ird.fr](mailto:camille.mazoyer@ird.fr)

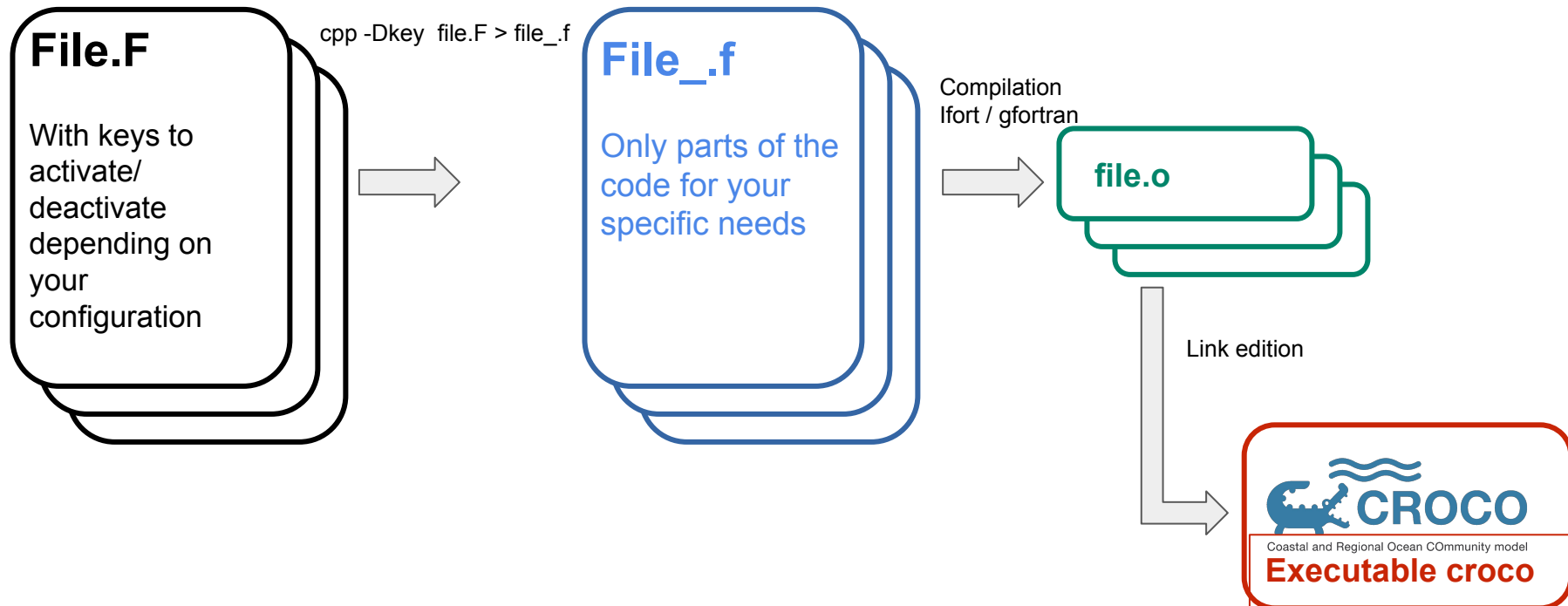
# How does it work?

## Préprocessing

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



# Compilation in 2 steps



# Compilation with CPP keys

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

file.F

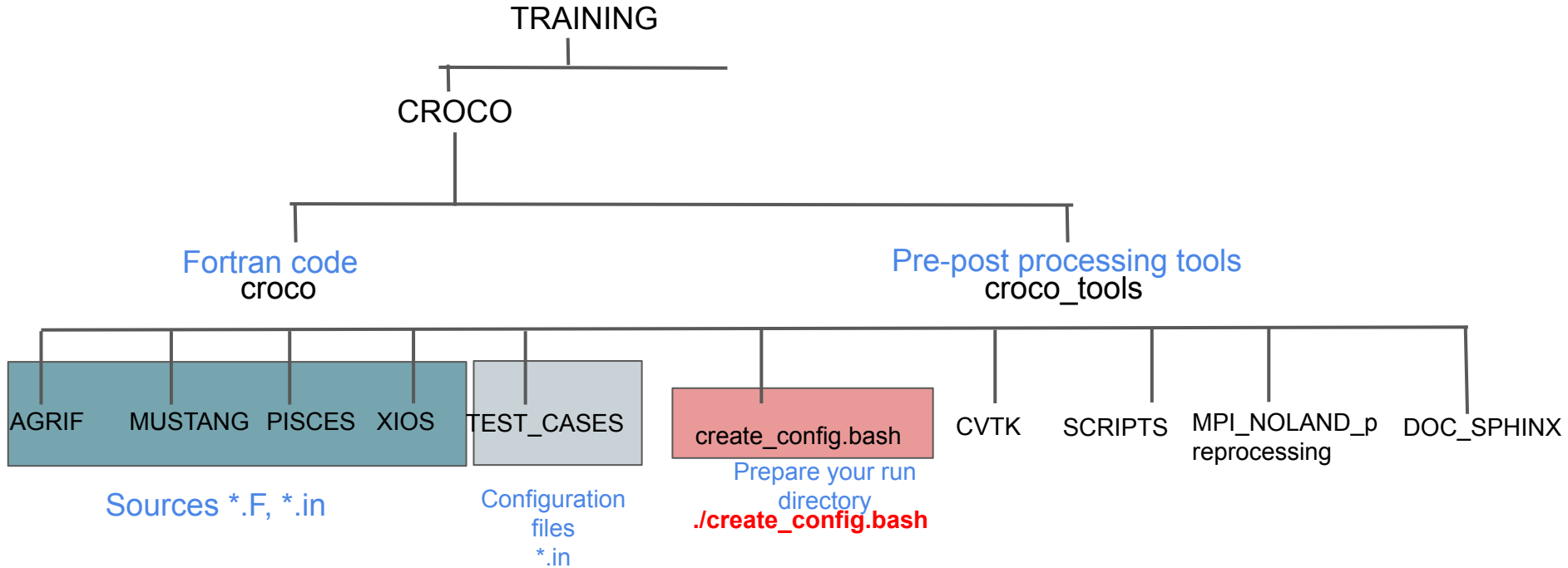
```
16 !
17 #include "cppdefs.h"
18     subroutine step()
19         implicit none
20 #include "param.h"
21 #include "scalars.h"
22 #include "zoom.h"
23 #include "grid.h"
24 #include "coupling.h"
25 #include "ocean3d.h"
26 #include "ocean2d.h"
27 #include "mpi_cpl.h"
28 #ifdef MUSTANG
29 # include "coupler_define_MUSTANG.h"
30 #endif
31
32 #ifdef AGRIF
33     IF (agrif_fixed().NE.sortedint(nbtimes)) return
34     nbtimes = nbtimes + 1
35 #endif
36
37 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
38 #ifdef SOLVE3D
39
40 #if defined OA_COUPLING || defined OW_COUPLING
41 !--Get phase of OASIS3
42     if ( (iif==-1).and.(oasis_time>=0).and.(nbstep3d
43 !         print *, 'oasis time before get = ', oasis_ti
44         call cpl_prism_get (oasis_time)
45 #endif AGRIF
46     else if ((.not.agrif_root()).and.(iif==-1).and.
47 &         (oasis_time>=0).and.(nbtimes<=Agrif_Max
48 &         (nbprtime<nbmaxprtime)) then
49         call cpl_prism_get (oasis_time)
```

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-dev »

```
# BEGIN USER MODIFICATIONS

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

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

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

# Configuration name
# -----
MY_CONFIG_NAME= BENGUELA_LR

# Home and Work configuration directories
# -----
MY_CONFIG_HOME= /home/userX/TRAINING/CONFIGS
MY_CONFIG_WORK= /home/userX/TRAINING/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-dev )

## example for production run architecture and coupling with external models :
#options=( all-prod-cpl )
```

# TRAINING

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

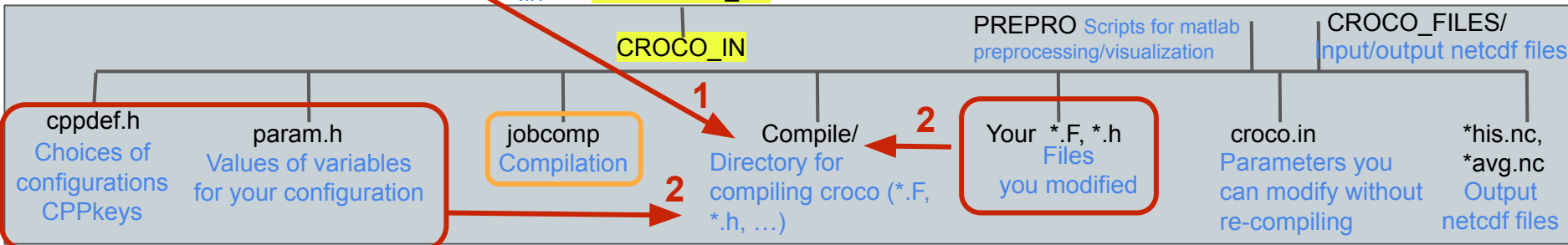
jobcomp  
Compilation

1  
Compile/  
Directory for  
compiling croco (\*.F,  
\*.h, ...)

2  
Your \*.F, \*.h  
Files  
you modified

croco.in  
Parameters you  
can modify without  
re-compiling

\*his.nc,  
\*avg.nc  
Output  
netcdf files



## 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
```