CROCO REGIONAL MODELING TUTORIALS
INTRODUCTION

Serena ILLIG (serena.illig@ird.fr)
http://sillig.free.fr/Courses_CROCO.html

These tutorials, available at https://doi.org/10.5281/zenodo.17091356, guide learners through building and running a regional ocean configuration with the **CROCO — Coastal and Regional Ocean COmmunity model** (https://www.croco-ocean.org/).

The pack combines lecture presentations, step-by-step hands-on tutorials, and a set of dedicated scripts and programs used throughout the training. The materials cover environment setup, grid, open boundary and surface forcing preparation, model compilation/execution, and basic post-processing. They are designed to be portable across high-performance computing (HPC) systems and workstations.

Intended audience and prerequisites:

This training is designed for researchers, graduate students, and professionals interested in numerical ocean modelling. Prior exposure to physical oceanography (undergraduate level) is recommended. Experience working with computer programming with MATLAB/Octave (optionally Python) and a basic knowledge of the Linux operating system are required.

1: Training Overview

The course is composed of a suite of lectures plus hands-on sessions. Each practical follows a specific step in preparing a CROCO regional configuration, ensuring that each step is explained in detail. Beyond the technical aspects of running the model, the course also introduces the foundations of ocean modelling — from primitive equations to numerical schemes — and explains how the ocean is forced. Particular attention is given to applying lecture content immediately in practice: concepts introduced in the morning are directly implemented during the afternoon practicals.

→ **TUTORIAL 00**: ***Basic Linux/PBS commands & vi editor*** : CROCO presentation and quick refresher on Linux system: navigation, useful commands, and job schedulers.

→ **TUTORIAL 01**: ***Preparation of CROCO and CROCO_Tools environment***
Log onto the HPC system and set up your CROCO project tree; copy CROCO source code and CROCO tools; configure your shell environment; verify that MATLAB can read NetCDF files.

→ **TUTORIAL 02**: ***Create your first CROCO grid and explore sigma vertical coordinates***
Execute the first preprocessing steps to create a model configuration directory and your CROCO grid. Define your domain extension and vertical level properties.

→ **TUTORIAL 03**: ***Independent activity on finite difference schemes***
With Matlab, model the temperature evolution of the 1D diffusion case.

The complete solution is provided in the Zenodo archive

→ **TUTORIAL 04**: ***Create your first CROCO configuration and execute the model***
Generate initial conditions and surface and boundary forcings using climatological datasets; Compile and launch the model; Perform basic output analyses.

→ **TUTORIAL 05**: ***Independent activity on the stability of numerical schemes***
Pen and paper exercise to explore the consistency and stability of the 1D advection problem; Introduction to the CFL condition.

The complete solution is provided in the Zenodo archive

→ **TUTORIAL 06**: ***CROCO nesting with AGRIF***
Learn how to create and run a CROCO configuration embedded with a zoom.

→ **TUTORIAL 07**: ***CROCO interannual simulation***
Build a CROCO configuration forced with interannual forcings generated with interannual datasets.

→ **TUTORIAL 08**: ***Adding river runoff***
Create river inflow forcing and run the `BENGUELA_LR` configuration with the Orange/Doring rivers.

2: File Organization in the Zenodo Archive

The archive is organized into several directories to facilitate navigation and reuse of the materials:

- **T00_CROCO_INTRO/**, **T01_CROCO_ENV/**, **T02_CROCO_GRID/**, **T03_FiniteDiff/**, **T04_CROCO_CLIM/**, **T05_Stability/**, **T03_CROCO_NESTING/**, **T07_CROCO_INTER/**, and **T08_CROCO_RUNOFF/**
↪ Directories containing training materials associated with each tutorial. It includes the tutorial document distributed during the training (**TXX_HandsOn_Tutorial.pdf**), and the lecture slides (**TXX_HandsOn_Presentation.pdf**).

- **Tools_and_Scripts/**
↪ For the Zenodo archive, all files not included in the official CROCO releases are provided in this directory. This ensures that learners outside CHPC can access the same supporting scripts and materials needed to reproduce the training exercises.
[For the summer schools held in Cape Town at the CHPC, the CROCO source code, CROCO tools, and the dedicated scripts/programs used during the training are stored in a shared directory on the HPC: **/home/apps/chpc/earth/CROCCO_Workshop/CROCO_TRAINING_Basic**]
This directory contains:
  - **.bashrc_example** adapted for the CHPC Lengau cluster (this hidden file can be listed using the **ls -a** command),
  - Example scripts for compilation and execution on HPC: **jobcomp_lengau**, **run_croco.pbs**, and **run_croco_inter.pbs**,
  - Specific tools created for Tutorials 01, 02, and 03 (**TP0_test_file.nc**, **TP0_test_script.m**, **draw_zonal_section.m**, and **My_diffusion_1D.m**),
  - Slight modifications for CROCO files **create_config.bash** and **crocoin.m**.

- **Activity_Solutions/**
↪ Directory containing the solutions for the numerical aspect activities for Tutorials 03 and 05.

3: Computing Environment and Portability

These tutorials were prepared for the **CHPC Lengau cluster** (Cape Town, South Africa), where I led the annual CROCO Summer School since 2022. However, the training is designed to be **portable**, providing some modifications. Below are the assumptions that are *hidden* during the in-person course.

Reference toolchain:

- **Compilers** : ifort / gfortran
- **Libraries**: NetCDF-Fortran; optional: MPI (OpenMPI or Intel MPI)
- **Pre-/post-processing**: MATLAB (preferred in class) or GNU Octave, Python
- **Utilities**: make, meld, a text editor (vi / nedit) / ▪ **Optional tools**: NCO, ncview.

↪ For the training, students are provided with an adapted **.bashrc** script that loads the required modules and defines useful aliases (see **STEP2** in **#TUTORIAL01**). Below is the script provided for the class on the CHPC Lengau cluster that you can adapt to your system:

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# A nice prompt (red on the frontal node and green on the interactive node):
if [ "`echo $HOSTNAME | cut -c1-5`" = login ]; then
   PS1="\e[0;31m[\u@\h \W]$ \e[m"
fi
if [ "`echo $HOSTNAME | cut -c1-5`" = cnode ]; then
   PS1="\e[0;32m[\u@\h \W]$ \e[m"
fi
```

```
# Some useful aliases:
alias rm='rm -i'
alias mv='mv -i'
alias cp='cp -i'
alias  qsubi1='qsub  -X  -I  -l  select=1:ncpus=1:mpiprocs=1  -q  R6060705  -l walltime=10:00:00 -P WCHPC'
alias qs="qstat -u $USER"
alias mat='matlab -nodesktop -nosplash'

# User specific aliases and functions
module purge
module load chpc/earth/nedit/5.7                    # load nedit text editor
module load gcc                                     # load gcc compiler
module load chpc/earth/ncview/2.1.7-gcc             # load ncview software
module load chpc/math/matlab/R2020a                 # load matlab software
module load chpc/earth/NCO/4.9.3                    # load nco tools
module load chpc/compmech/meld-3.22.1               # load meld tool (visual diff)
module load chpc/python/3.7.0                       # load python library
module load chpc/parallel_studio_xe/2020u1          # load mpiifort
module load chpc/earth/netcdf/4.7.4/intel2020u1 # load NetCDF library
```

☝ If **MATLAB** is not available on your cluster, perform the preprocessing (grid/forcing/OBC generation) on a workstation/laptop with MATLAB or Octave, then copy the resulting NetCDF files to the HPC system for model runs. Runtime steps (compile/execute) remain on the cluster.

Parallelization and Scheduler:

➥ In the class, CROCO is compiled with MPI (**`mpiifort`**) for parallel execution. A customized **CROCO compilation script** **`jobcomp_lengau`** is distributed. Differences from the standard **`jobcomp`** script (provided with the CROCO source code) include compiler settings and source path:

line	Original **jobcomp**	**jobcomp_lengau**
26	SOURCE=../croco/OCEAN	SOURCE=../OCEAN
39	FC=gfortran	FC=ifort
44	MPIF90="mpif90"	MPIF90="mpiifort"

➥ The CHPC Lengau cluster uses **PBS Pro** as its job scheduler. Model execution jobs are submitted to a batch queuing system and are executed only when the requested resources become available using the command **`qsub`**.

• The **CROCO execution script** provided for the class, namely **`run_croco.pbs`**, is adapted from the original version provided with CROCO source code (**`run_croco.bash`**).

The run command (*line 34*) is chosen: **RUNCMD**="mpirun -np $NBPRPCS" (*line 34*)
and PBS headers are added:

```
#!/bin/bash
#PBS -l select=1:ncpus=24:mpiprocs=4
#PBS -P WCHPC
#PBS -q serial
#PBS -l walltime=2:00:00
#PBS -m abe
#PBS -M put.your@email.here
cd /home/$USER/lustre/CROCO/croco-v2.0.1/Run_Clim
#
```

• Also, on the CHPC Lengau cluster, preprocessing and compilation are performed on interactive compute nodes, since **login/front-end nodes forbid heavy work** on this HPC system. To simplify this process, we create an alias in the **`.bashrc`** script that requests an interactive session:

```
# Example alias placed in ~/.bashrc
alias qsubi1='qsub -I -X -l select=1:ncpus=1:mpiprocs=1 -l walltime= 08:00:00 -q small'
# Usage:
qsubi1 # then load modules, run MATLAB/Octave, compile CROCO, etc…
```

➥ To help avoid mistakes, the **`.bashrc`** script also modifies the shell prompt (see page 2): it appears red when connected to a login node and green when connected to an interactive node. This visual cue reminds users where heavy computations can safely be run on the CHPC Lengau cluster.

⇨ **Adapt the commands and headers to your local system** (PBS, Slurm, etc.).

✋ Note that the architecture used during the training is not fully adapted for HPC systems, as all files are stored in a space that is not saved. For production runs, it is recommended to store the sources and scripts in your home directory, which is usually backed up but may be limited in size. Conversely, CROCO input and output files should be written to the temporary data storage system of the HPC (i.e., the **`lustre`** system on the CHPC).

4: History and Contributions

I created this course in 2015, with the help of Emeline Cadier, and first delivered at the Department of Oceanography, University of Cape Town (UCT). Since then, it has been refined based on feedback from students and collaborators. Since 2022, it has been part of the annual IRN-Sud CROCO Summer Schools held at the CHPC in Cape Town.