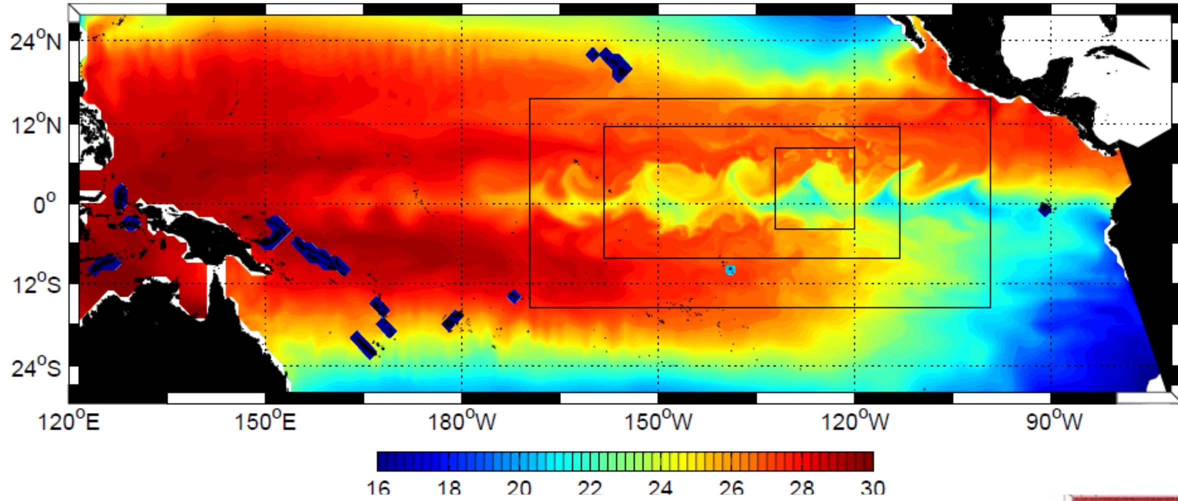
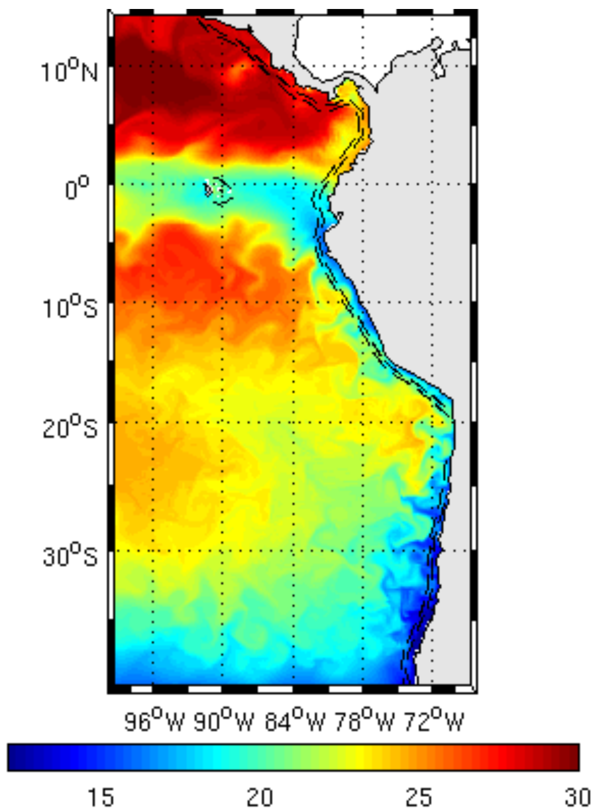
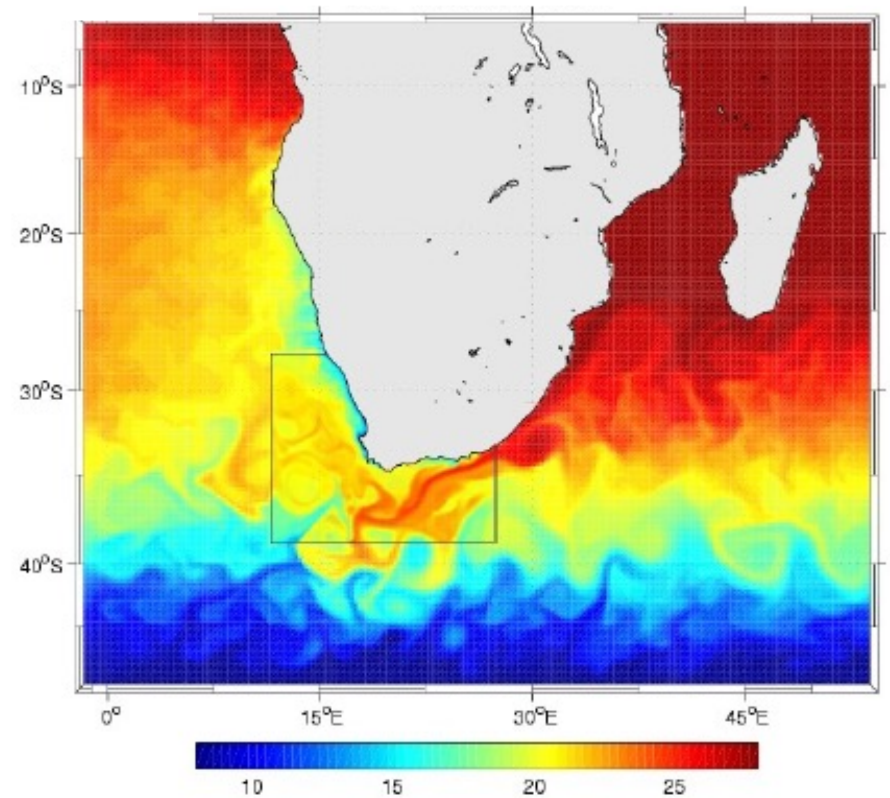
Introduction to CROCO / CROCO_TOOLS ocean modeling system



CROCO
Coastal and Regional Ocean COmmunity model





Gildas Cambon (gildas.cambon@ird.fr)
Swen Jullien (swen.jullien@ifremer.fr)

Strategy to build a configuration

Global datasets

- GEBCO 1', Etopo2, SRTM30, ….
- COADS, QuikSCAT, CFSR, …
- WOA_2009, SODA, ECCO, Mercator, …

CROCOTOOLS

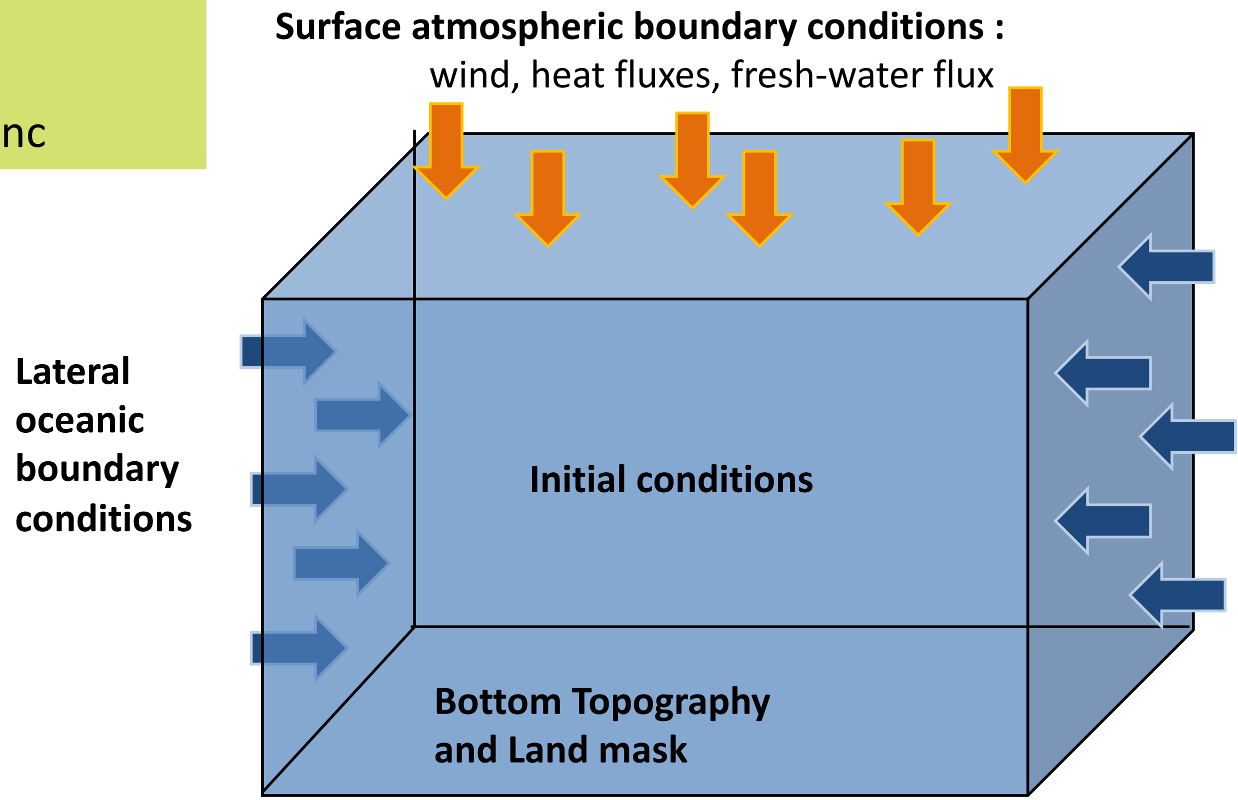
CROCO input netcdf files

- croco_grd.nc,
- croco_frc.nc, croco_blk.nc
- croco_ini.nc, croco_clm.nc, croco_bry.nc

I Pre-processing phase :

Input file creation :

- Topography
- Lateral oceanic boundaries
- Surface atmospheric boundaries



Strategy to build a configuration

Global datasets

- GEBCO 1’, Etopo2, SRTM30, ….
- COADS, QuikSCAT, CFSR, …
- WOA_2009, SODA, ECCO, Mercator, …

CROCOTOOLS

CROCO input netcdf files

- croco_grd.nc,
- croco_frc.nc, croco_blk.nc
- croco_ini.nc, croco_clm.nc, croco_bry.nc

CROCO

I Pre-processing phase :

Input file creation :

- Topography
- Lateral oceanic boundaries
- Surface atmospheric boundaries

II Compilation phase :

- Create the croco executable with you options : numeric, physic and domain size

Strategy to build a configuration

Global datasets

- GEBCO 1', Etopo2, SRTM30, ….
- COADS, QuikSCAT, CFSR, …
- WOA_2009, SODA, ECCO, Mercator, …

CROCOTOOLS

CROCO input netcdf files

- croco_grd.nc,
- croco_frc.nc, croco_blk.nc
- croco_ini.nc, croco_clm.nc, croco_bry.nc

CROCO

CROCO output netcdf files

- croco_his.nc, croco_avg.nc
- croco_diags.nc, croco_diags_avg.nc
- croco_diagsM.nc, croco_diagsM_nc

…

I Pre-processing phase :

Input file creation :

- Topography
- Lateral oceanic boundaries
- Surface atmospheric boundaries

II Compilation phase :

- Create the croco executable with you options : numeric, physic and domain size

III Computing phase :

- the croco model run, resolve the primitive equation
- produce the outputs in croco_avg.nc, croco_his.nc

Strategy to build a configuration

Global datasets

- GEBCO 1', Etopo2, SRTM30, ….
- COADS, QuikSCAT, CFSR, …
- WOA_2009, SODA, ECCO, Mercator, …

CROCOTOOLS

CROCO input netcdf files

- croco_grd.nc,
- croco_frc.nc, croco_blk.nc
- croco_ini.nc, croco_clm.nc, croco_bry.nc

CROCO

CROCO output netcdf files

- croco_his.nc, croco_avg.nc
- croco_diags.nc, croco_diags_avg.nc
- croco_diagsM.nc, croco_diagsM_nc

…

CROCOTOOLS

analysis

visualization

I Pre-processing phase :

Input file creation :

- Topography
- Lateral oceanic boundaries
- Surface atmospheric boundaries

II Compilation phase :

- Create the croco executable with you options : numeric, physic and domain size

III Computing phase :

- the croco model run, resolve the primitive equation
- produce the outputs in croco_avg.nc, croco_his.nc

IV - Post processing / Analyse phase :

- Visualize and analyze model results …

System requirement

OS: LINUX or UNIX
CROCO : fortran program ~ (55000 lines of FORTRAN code)
CROCOTOOLS : Matlab (Octave) scripts + datasets (18 Gb)

PC configuration:

- Fortran Compiler : gfortran, Intel Fortran Compiler
- NetCDF : http://www.unidata.ucar.edu
- OpenMPI : http://www.open-mpi.org
- Matlab (or Octave)

Library needed :

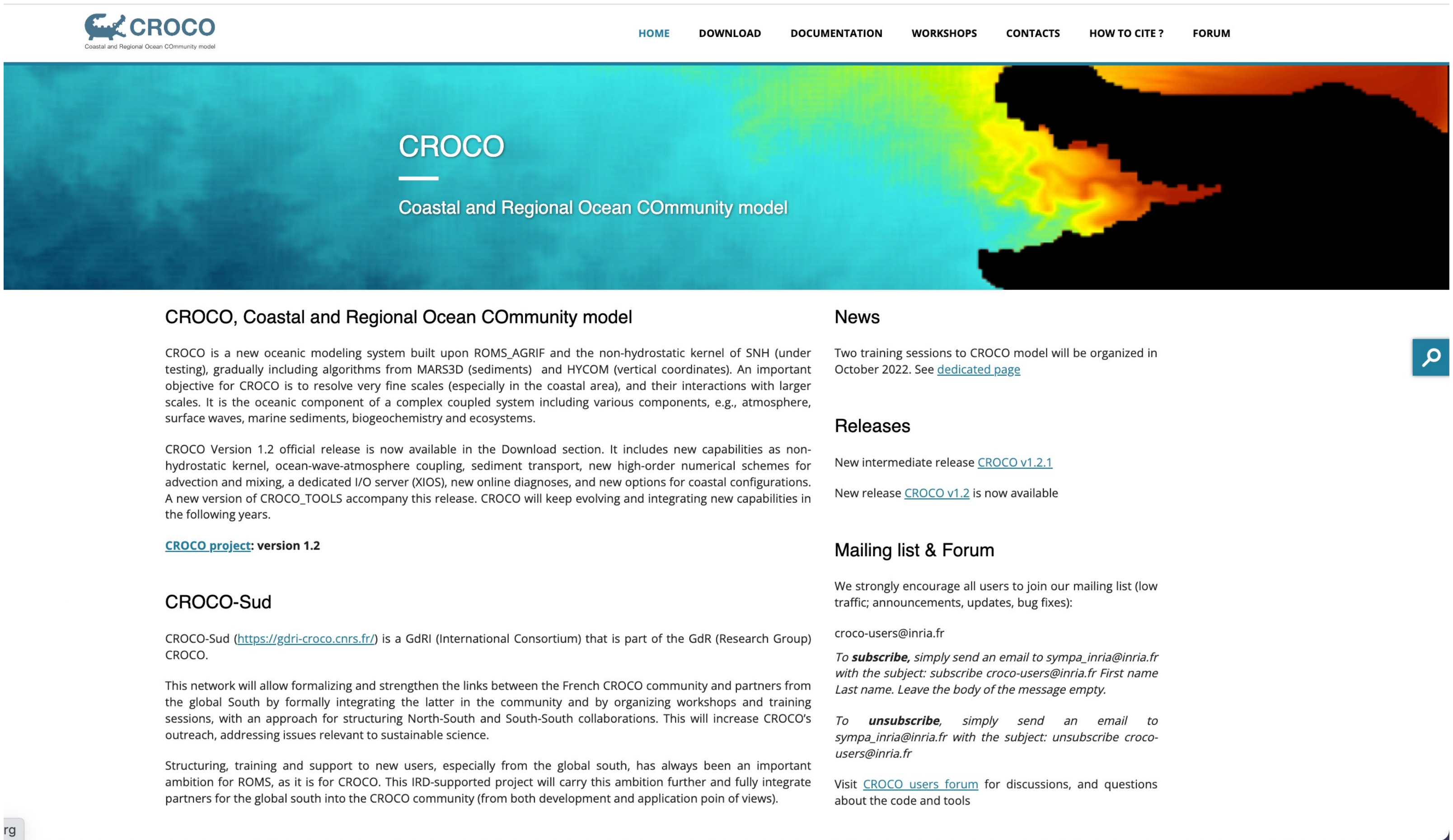
- NetCDF : http://www.unidata.ucar.edu

Additional software

- Ncview
- NCO
- Ferret
- Octave-3.6.1 + octcdf 1.1.5

How to get the tools?

http://www.croco-ocean.org



→ **In the download section**

- **CROCO** : Fortran source code
- **CROCO_TOOLS** : Pre- and post
- processing toolbox (Matlab)
- **DATASETS** [18 Gb] : the global datasets needed by the CROCO_TOOLS
- **UTILITIES** : netCDF library, mapping toolbox etc …

A modeling package

Coastal and Regional Ocean COmmunity model

croco_tools : Matlab pre-post processing toolbox

- Aforc_NCEP
- Aforc_QuikSCAT
- Diagnostic_tools
- Nesting_tools
- Opendap_tools
- Opendap_tools_no_loaddap
- RUNOFF_DAI
- Tides
- Town
- UTILITIES
- Visualization_tools
- croco_pytools
- croco_pyvisu
- start.m
- Oforc_OGCM
- Aforc_CFSR
- Aforc_ECMWF
- Coupling_tools
- Rivers
- crocotools_param.m
- readme_version_croco_tools.txt
- Forecast_tools
- Preprocessing_tools

UTILITIES:

- air_sea
- export_fig
- m_map1.4h
- mask
- mex60
- mexcdf
- netcdf_matlab_60
- netcdf_x86_64

CROCO : Fortran source code

- AGRIF
- CVTK
- OCEAN
- PISCES
- Run
- XIOS
- create_config.bash

DATA_SETS : Global datasets to build the regional forcing

- CARS2009
- COADS05
- GOT99.2
- m_map1.4f
- QuikSCAT_clim
- RUNOFF_DAI
- SeaWifs
- SST_pathfinder
- Topo
- TPXO6
- TPXO7
- WOA2009
- WOAPISCES

How to deploy a configuration ?

Create your the BENGUELA configuration

- *Edit create_config.sh*
- *define the directory where are the CROCO, the CROCO_TOOLS sources and where the simulations*
- *Define the name of the configuration : BENGUELA_LR*
- *Execute ./create_config.sh*

Go in the BENGUELA_LR configuration directory

- *cd [where_are_the simulations]/CONFIGS/BENGUELA_LR*

Edit the parameter file crocotools_param.m

- *gedit crocotools_param.m :* ***General parameters of the croco_tools***

[where_are_the simulations]/CONFIGS/BENGUELA_LR **:**
Your own croco configuration.

- CROCO_FILES
- Misc
- NAMELIST_OANALYSIS
- README_XIOS
- TEST_CASES
- cppdefs.h
- create_myconfig.bash.BCK
- croco.in
- croco.in.1
- croco_inter.in
- crocotools_param.m
- domain_def.xml
- field_def.xml_full
- jobcomp
- param.h
- run_croco.bash
- run_croco_forecast.bash
- run_croco_inter.bash
- sediment.in
- start.m
- xios_launch.file

Step to build a configuration ?

The **climatological** Southern Benguela configuration at low resolution example (1/3°)

- General parameter
- Building the grid
- Processing the forcing (atmo + oceanic + initial conditions)
- Preparing the model
- Compiling the model
- Running the model
- Visualization of the results

How to build a input files using CROCO_TOOLS ?

Documentation :
https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.04.config.html

7. Regional: Preprocessing (Matlab)

CROCO preprocessing tools have been developed under Matlab software by IRD researchers (former Roms_tools). Note: These tools have been made to build easily regional configurations using climatological data. To use interannual data, some facilities are available (NCEP, CFSR, QuickScat data for atmospheric forcing, SODA and ECCO for lateral boundaries). However, to use other data, you will need to adapt the scripts. All utilities/toolbox requested for matlab crocotools programs are provided within the UTILITIES directory, or can be downloaded here: http://www.croco-ocean.org/download/utilities/

Previous

Next

The crocotools_param.m files

`crocotools_param.m` is separated into several sections:

1 - Configuration parameters	used by make_grid.m (and others..)
2 - Generic file and directory names	need to match your work architecture
3 - Surface forcing parameters	used by make_forcing.m and by make_bulk.m
4 - Open boundaries and initial conditions parameters	used by make_clim.m, make_biol.m, make_bry.m make_OGCM.m and make_OGCM_frcst.m
5 - Parameters for tidal forcing	used by make_tides.m
6 - Reference date and simulation times	used for make_tides, make_CFSR (or make_NCEP), make_OGC
7 - Parameters for Interannual forcing	SODA, ECCO, CFSR, NCEP, ...
8 - Parameters for the forecast system	used by make_forecast.m
9 - Parameters for the diagnostic tools	used by scripts in `Diagnostic_tools`

https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.05.prepro.matlab.clim.html

Building the grid

Launch matlab session in$confs/CONFIGS/BENGUELA_LR

make_grid.m → create the input file **croco_grd.nc**

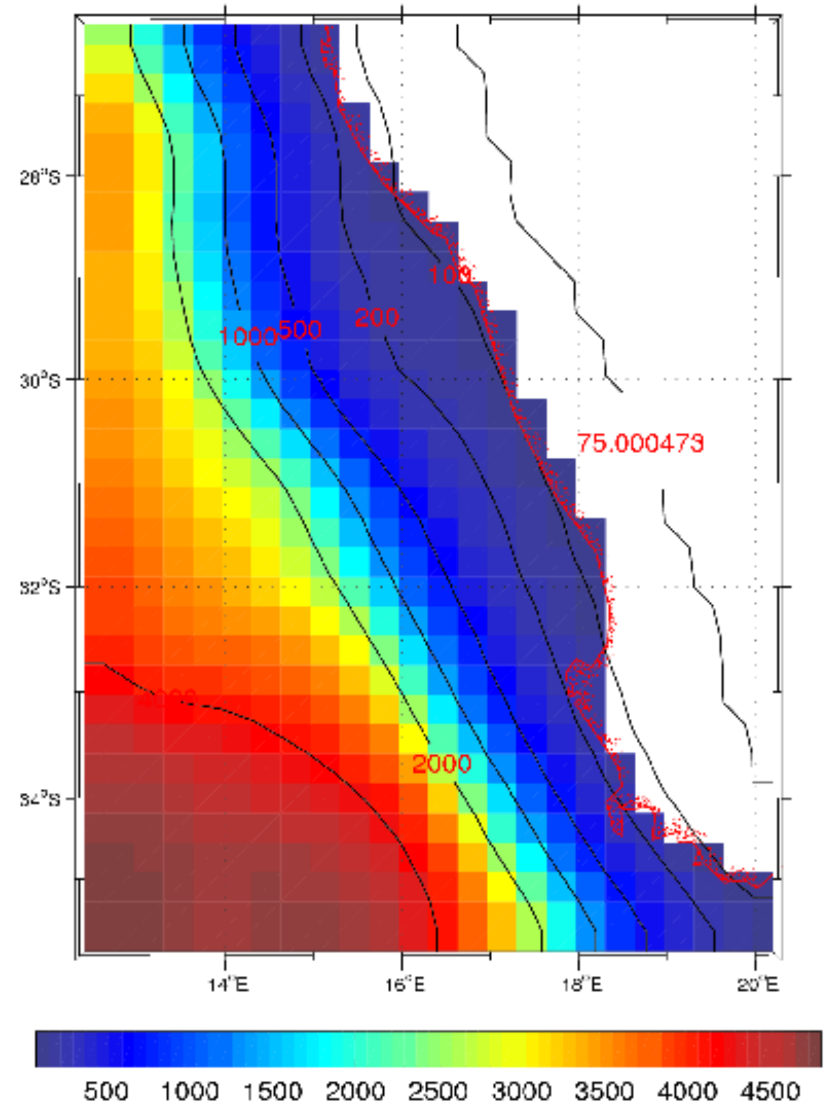
- Horizontal grid : position of the grid points, size of the grid cells
- Bottom topography
- Land mask

Section 1 in crocotools_param.m

```
%
% 1 - Configuration parameters
% used by make_grid.m (and others..)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
isoctave=exist('octave_config_info');
%
% CROCO title names and directories
%
CROCO_title = 'Benguela Model';
CROCO_config = 'Benguela_LR';
%
% Grid dimensions:
%
lonmin = 8; % Minimum longitude [degree east]
lonmax = 22; % Maximum longitude [degree east]
latmin = -38; % Minimum latitude [degree north]
latmax = -26; % Maximum latitude [degree north]
%
% Grid resolution [degree]
%
dl = 1/3;
%
% Number of vertical Levels (! should be the same in param.h !)
%
N = 32;
%
% Vertical grid parameters (! should be the same in croco.in !)
%
theta_s = 7.;
theta_b = 2.;
hc = 200.;
vtransform = 2.; % s-coordinate type (1: old- ; 2: new- coordinates)
% ! take care to define NEW_S_COORD cpp-key in cppdefs.h
%
% Topography: choice of filter
%
topo_smooth = 1; % 1: old ; 2: new filter (better but slower)
%
% Minimum depth at the shore [m] (depends on the resolution,
% rule of thumb: dl=1, hmin=300, dl=1/4, hmin=150, ...)
% This affect the filtering since it works on grad(h)/h.
%
hmin = 75;
%
% Maximum depth at the shore [m] (to prevent the generation
% of too big walls along the coast)
%
hmax_coast = 500;

....
```

Building the grid

```
Making the grid:
/home1/datawork/gcambon/TRAINING_2019/CONFIGS/Run_BENGUELA_LR/CROCO_FILES/croco_grd.nc

 Title: Benguela Model
 Resolution: 1/3 deg

 Do you want to use interactive grid maker ?
 (e.g., for grid rotation or parameter adjustments) : y,[n] n

 Create the grid file...
 LLm = 41
 MMm = 42

 Fill the grid file...
 Compute the metrics...
 Min dx=29.1913 km - Max dx=33.3244 km
 Min dy=29.2434 km - Max dy=33.1967 km
 Fill the grid file...
 Add topography...
   CROCO resolution : 31.3 km
   Topography data resolution : 3.42 km
   Topography resolution halved 4 times
   New topography resolution : 54.6 km
Processing coastline_l.mat ...

 Do you want to use editmask ? y,[n]
...
```



Build the atmospheric surface boundary forcing

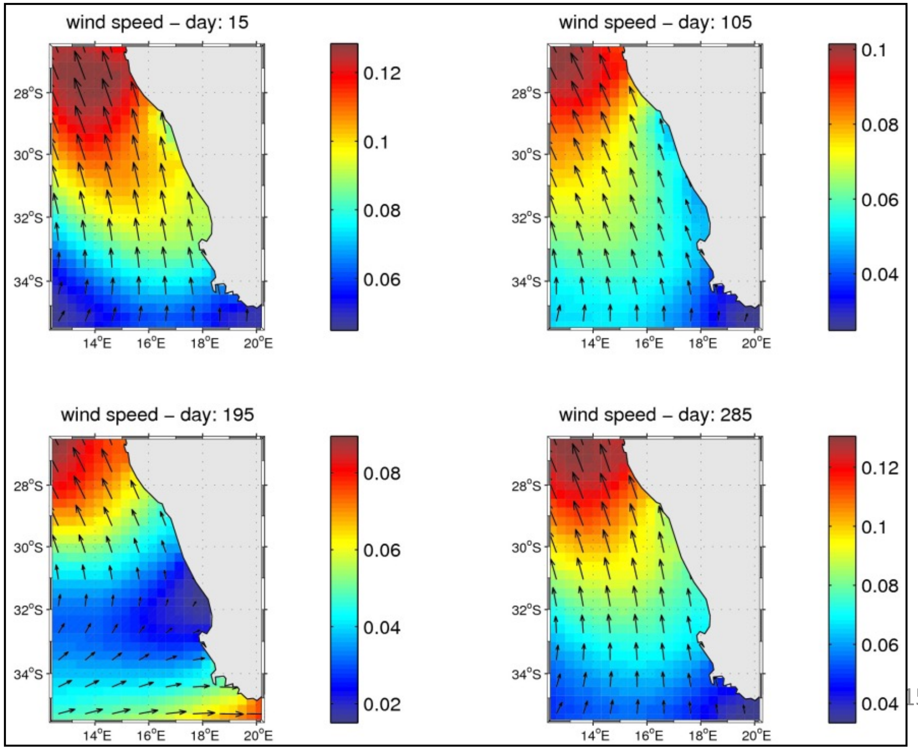
CROCO Coastal and Regional Ocean COmmunity model

make_forcing .m → create the input file **croco_frc.nc**

Section 3 crocotools_param.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% COADS directory (for climatology runs)
%
coads_dir=[DATADIR,'COADS05/'];
%
% COADS time (for climatology runs)
%
coads_time=(15:30:345); % days: middle of each month
coads_cycle=360; % repetition of a typical year of 360 days
%
%coads_time=(15.2188:30.4375:350.0313); % year of 365.25 days in case
%coads_cycle=365.25; % interannual QSCAT winds
% % are used with clim. heat flux
%
% Pathfinder SST data used by pathfinder_sst.m
%
pathfinder_sst_name=[DATADIR,...
'SST_pathfinder/climato_pathfinder.nc'];
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Benguela Model

 Read in the grid...

 Create the forcing file...
Getting taux for time index 1
Getting tauy for time index 1
Getting taux for time index 2
Getting tauy for time index 2
Getting taux for time index 3
Getting tauy for time index 3
...
Getting shortrad for time index 9
Getting shortrad for time index 10
Getting shortrad for time index 11
Getting shortrad for time index 12
```



Build the initial/open boundary oceanic conditions

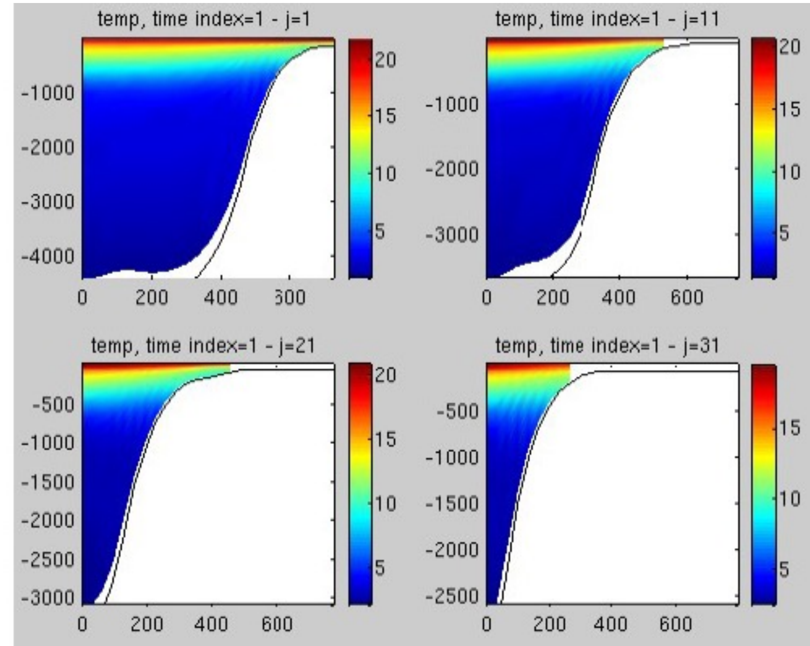
Section 4 in crocotools_param.m

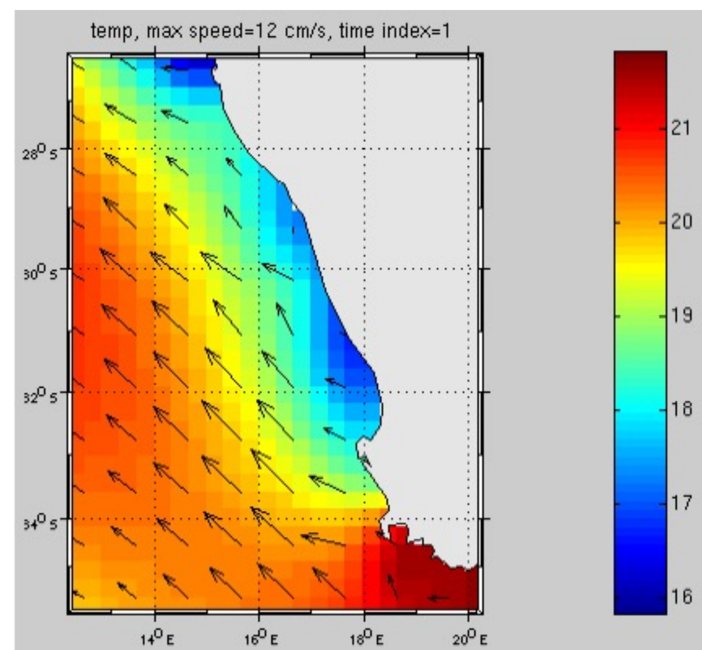
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Open boundaries switches (! should be consistent with cppdefs.h !)
%
obc = [1 1 1 1]; % open boundaries (1=open , [S E N W])
%
%  Level of reference for geostrophy calculation
%
zref = -1000;
%
%  initial/boundary data options (1 = process)
%  (used in make_clim, make_biol, make_bry,
%   make_OGCM.m and make_OGCM_frcst.m)
%
makeini    = 1;   % initial data
makeclim   = 1;   % climatological data (for boundaries and nudging layers)
makebry    = 1;   % lateral boundary data
makenpzd   = 0;   % initial and boundary data for NChlPZD and N2ChlPZD2 models
makebioebus= 0;   % initial and boundary data for BioEBUS model
makepisces = 0;   % initial and boundary data for PISCES model
%
%
makeoa     = 1;   % oa data (intermediate file)
makeZbry   = 1;   % boundary data in Z coordinate (intermediate file)
insitu2pot = 1;   % transform in-situ temperature to potential temperature
%
%  Day of initialisation for climatology experiments (=0 : 1st january 0h)
%
tini=0;
%
```

```
%
% Pisces biogeochemical seasonal climatology
%
woapisces_dir = [DATADIR,'WOAPISCES/'];  % only compatible with woa_dir
%
% Surface chlorophyll seasonal climatology (SeaWifs)
%
chla_dir=[DATADIR,'SeaWifs/'];
%
% Runoff monthly seasonal climatology (Dai and Trenberth)
%
global_clim_riverdir=[DATADIR,'RUNOFF_DAI/'];
global_clim_rivername=[global_clim_riverdir,'Dai_Trenberth_runoff_global_clim.nc'];
%
%  Set times and cycles for the boundary conditions:
%   monthly climatology
%
woa_time=(15:30:345); % days: middle of each month
woa_cycle=360;       % repetition of a typical year of 360 days
%
```

> **make_ini.m** → create **croco_ini.nc**

```
Benguela Model
Making initial file:
/home1/datawork/gcambon/TRAINING_2019/CONFIGS/Run_BENGUELA
_LR/CROCO_FILES/croco_ini.nc

 Title: Climatology

 Creating the file :
/home1/datawork/gcambon/TRAINING_2019/CONFIGS/Run_BENGUELA
_LR/CROCO_FILES/croco_ini.nc
 VTRANSFORM = 2

 Interpolations / extrapolations

 Temperature...

 Ext tracers: ro = 0 km - default value = NaN
   ext_tracers_ini: time index: 1 of total: 12
   ext_tracers_ini: horizontal interpolation of seasonal data
   ext_tracers_ini: vertical interpolation

 Salinity...

 Ext tracers: ro = 0 km - default value = NaN
   ext_tracers_ini: time index: 1 of total: 12
   ext_tracers_ini: horizontal interpolation of seasonal data
   ext_tracers_ini: vertical interpolation

 Compute potential temperature from in-situ...
   getpot: Time index: 1 of total: 1
```





Build the Open boundary oceanic conditions : clm type [old way]

> **make_clim.m** create **croco_clm.nc**

```
Creating the file :
/home1/datawork/gcambon/TRAINING_2019/CONFIGS/Run_BENGUELA_LR/CROCO_FILES/croco_clm.nc

 VTRANSFORM = 2

          NetCDF_File: '/home1/datawork/gcambon/TRAINING_2019/CONFIGS/Run_BENGUELA_LR/CROCO_FILES/croco_clm.nc'
          nDimensions: 20
           nVariables: 31
    nGlobalAttributes: 0
      RecordDimension: ''
             nRecords: 0
           Permission: 'clobber'
           DefineMode: 'define'
             FillMode: 'fill'
           MaxNameLen: 0

 Create the OA file...

 Creating the file :
/home1/datawork/gcambon/TRAINING_2019/CONFIGS/Run_BENGUELA_LR/CROCO_FILES/croco_oa.nc

 Horizontal extrapolations
 Temperature...

 Ext tracers: Roa = 0 km - default value = NaN
 Ext tracers: horizontal interpolation of the annual data
 Ext tracers: horizontal interpolation of the seasonal data
time index: 1 of total: 12
…
time index: 12 of total: 12
```

```
Salinity...

 Ext tracers: Roa = 0 km - default value = NaN
 Ext tracers: horizontal interpolation of the annual data
 Ext tracers: horizontal interpolation of the seasonal data
time index: 1 of total: 12
…
Vertical interpolations

 Temperature...
 Time index: 1 of total: 12
…

 Compute potential temperature from in-situ...
   getpot: Time index: 1 of total: 12
…
Compute geostrophic currents
time index: 1 of total: 12
Flux correction : -7.2065
Mask: 16 iterations
…
```

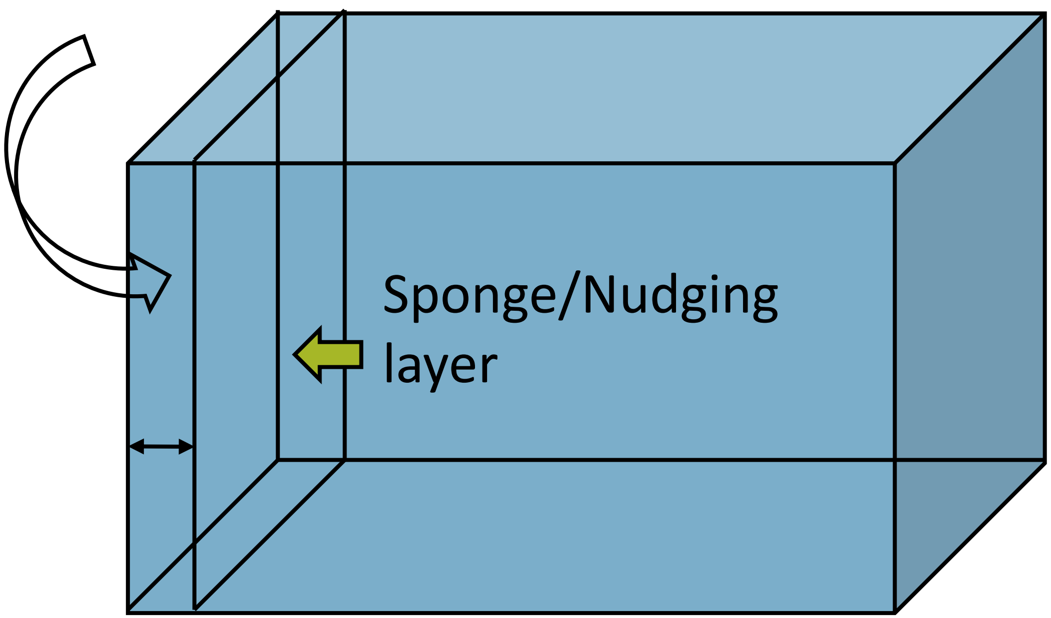
> **make_bry.m** create **croco_bry.nc**

```
Create the boundary file...
 Creating the file :
/home1/datawork/gcambon/TRAINING_2019/CONFIGS/Run_BENGUEL
A_LR/CROCO_FILES/croco_bry.nc

 VTRANSFORM = 2

nc =

         NetCDF_File:
'/home1/datawork/gcambon/TRAINING_2019/CONFIGS/Run_BENGUEL
A_LR/CROCO_FILES/croco_bry.nc'
         nDimensions: 21
          nVariables: 10
    nGlobalAttributes: 0
      RecordDimension: ''
            nRecords: 0
          Permission: 'clobber'
          DefineMode: 'define'
            FillMode: 'fill'
          MaxNameLen: 0

 Create the boundary Z-file...

 Creating the file :
/home1/datawork/gcambon/TRAINING_2019/CONFIGS/Run_BENGUEL
A_LR/CROCO_FILES/croco_bry_Z.nc

 Horizontal extrapolations
 Processing southern boundary...
  Temperature...
 Ext tracers: horizontal interpolation of the annual data
 Ext tracers: horizontal interpolation of the seasonal data
time index: 1 of total: 12
```

Different ways to impose OBC

CLIM (make_clim) : '3D+time' files (x,y,z,t) only used at boundaries point + sponge/nudging layer : large amount of data unused.

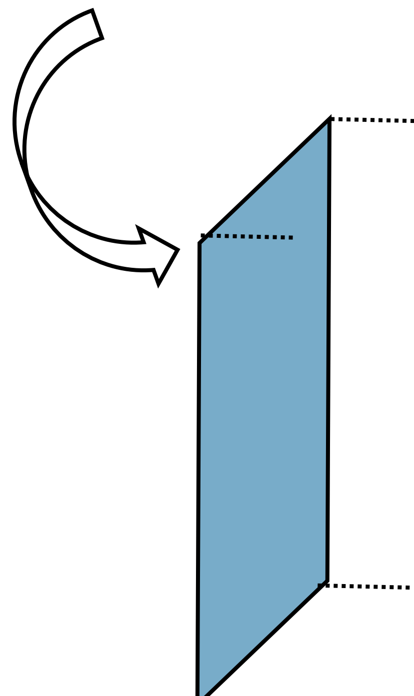
Data used here only



These type of file 3D (x,y,z) are used for initialization

BRY (make_bry): '2D+time' file (x,z,t) only used at boundaries point : much less data needed !! **but no nudging layer** (for the moment)

Data used only on the x-z or y-z face



Summary for pre-processing input files [climato]

cd $confs/CONFIGS/BENGUELA_LR

matlab &

>> **start** : Add all the needed matlabpath of the system

>> **make_grid**

=> creates CROCO_FILES/croco_grd.nc

- Horizontal grid : position of the grid points, size of the grid cells
- Bottom topography
- Land mask

>> **make_forcing**

=> create CROCO_FILES/croco_frc.nc

- Surface forcing : wind stress, surface heat flux, surface freshwater flux

>> **make_ini**

=> creates CROCO_FILES/croco_ini.nc

initial conditions : T, S, currents , SSH

>> **make_clim** (or **make_bry**)

=> creates CROCO_FILES/croco_clm.nc (or . croco_bry.nc)

- Lateral oceanic boundary conditions T, S, currents , SSH

Configuring the model CROCO

Documentation :
https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.07.compil.html

Physical and numerical choices : cppdefs.h

cppdefs.h : define physical and numerical choices

- Define CPP keys used by the C-preprocessor when compiling the model
- Reduce the code to its minimal size : fast compilation
- Avoid FORTRAN logical statements: efficient coding

Configuring the model CROCO

```
#if defined REGIONAL
/*
!====================================================================
REGIONAL (realistic) Configurations
!====================================================================
!
!----------------------
! BASIC OPTIONS
!----------------------
!
*/
                  /* Configuration Name */
# define BENGUELA_LR
                  /* Parallelization */
# undef  OPENMP
# undef  MPI
# undef  MPI_NOLAND
                  /* I/O server */
# undef  XIOS
# undef  XIOS2
                  /* Non-hydrostatic option */
# undef  NBQ
                  /* Nesting */
# undef  AGRIF
# undef  AGRIF_2WAY
                  /* OA and OW Coupling via OASIS (MPI) */
# undef  OA_COUPLING
# undef  OW_COUPLING
                  /* Wave-current interactions */
# undef  MRL_WCI
                  /* Open Boundary Conditions */
# undef  TIDES
# define OBC_EAST
# define OBC_WEST
# define OBC_NORTH
# define OBC_SOUTH
 /* Applications */
# undef  BIOLOGY
# undef  FLOATS
# undef  STATIONS
# undef  PASSIVE_TRACER
# undef  SEDIMENT
# undef  BBL
                  /* dedicated croco.log file */
# undef  LOGFILE
                  /* Calendar */
# undef  USE_CALENDAR
/*!
!-------------------------------------------------
! PRE-SELECTED OPTIONS
!
! ADVANCED OPTIONS ARE IN CPPDEFS_DEV.H
!-------------------------------------------------
*/
                  /* Parallelization */
# ifdef MPI
#  undef  PARALLEL_FILES
# endif
# undef  NC4PAR
# undef  AUTOTILING
                  /* Non-hydrostatic options */
# ifdef NBQ
#  define W_HADV_TVD
#  define W_VADV_TVD
# endif
                  /* Grid configuration */
# define CURVGRID
# define SPHERICAL
# define MASKING
# undef  WET_DRY
# define NEW_S_COORD
                  /* Model dynamics */
# define SOLVE3D
# define UV_COR
# define UV_ADV
```

cppdefs.h file (1/2)

```
/* Equation of State */
# define SALINITY
# define NONLIN_EOS
             /* Lateral Momentum Advection (default UP3) */
# define UV_HADV_UP3
# undef  UV_HADV_UP5
# undef  UV_HADV_WENO5
# undef  UV_HADV_TVD
             /* Lateral Explicit Momentum Mixing */
# undef  UV_VIS2
# ifdef UV_VIS2
#  define UV_VIS_SMAGO
# endif
             /* Vertical Momentum Advection  */
# define UV_VADV_SPLINES
# undef  UV_VADV_WENO5
# undef  UV_VADV_TVD
             /* Lateral Tracer Advection (default UP3) */
# undef  TS_HADV_UP3
# define TS_HADV_RSUP3
# undef  TS_HADV_UP5
# undef  TS_HADV_WENO5
             /* Lateral Explicit Tracer Mixing */
# undef  TS_DIF2
# undef  TS_DIF4
# undef  TS_MIX_S
             /* Vertical Tracer Advection  */
# undef  TS_VADV_SPLINES
# define TS_VADV_AKIMA
# undef  TS_VADV_WENO5
             /* Sponge layers for UV and TS */
# define SPONGE
             /* Semi-implicit Vertical Tracer/Mom Advection */
# undef  VADV_ADAPT_IMP
             /* Bottom friction in fast 3D step */
# undef  BSTRESS_FAST
             /* Vertical Mixing */
# undef  BODYFORCE
# undef  BVF_MIXING
# define LMD_MIXING
# undef  GLS_MIXING
# ifdef LMD_MIXING
#  define LMD_SKPP
#  define LMD_BKPP
#  define LMD_RIMIX
#  define LMD_CONVEC
#  undef  LMD_DDMIX
#  define LMD_NONLOCAL
#  undef  MLCONVEC
# endif
             /* Surface Forcing */
# undef BULK_FLUX
# ifdef BULK_FLUX
#  define BULK_FAIRALL
#  define BULK_LW
#  define BULK_EP
#  define BULK_SMFLUX
#  undef  SST_SKIN
#  undef  ANA_DIURNAL_SW
#  undef  ONLINE
#  ifdef ONLINE
#   undef  AROME
#   undef  ERA_ECMWF
#  endif
#  undef READ_PATM
#  ifdef READ_PATM
#   define OBC_PATM
#  endif
# else
#  define QCORRECTION
#  define SFLX_CORR
#  undef  SFLX_CORR_COEF
#  define ANA_DIURNAL_SW
# endif
# undef SMFLUX_CFB
# undef  SEA_ICE_NOFLUX
             /* Wave-current interactions */
# ifdef OW_COUPLING
#  define MRL_WCI
#  define BBL
# endif
# ifdef MRL_WCI
#  ifndef OW_COUPLING
#   define WAVE_OFFLINE
#   undef  WKB_WWAVE
#  endif
#  undef  WAVE_ROLLER
#  define WAVE_STREAMING
#  define WAVE_FRICTION
#  define WAVE_RAMP
#  ifdef WKB_WWAVE
#   undef  WKB_OBC_NORTH
#   undef  WKB_OBC_SOUTH
#   undef  WKB_OBC_WEST
#   undef  WKB_OBC_EAST
#  endif
# endif
```

cppdefs.h file (2/2)

Configuring the model CROCO

Array dimensions and parameters (param.h)

```
!----------------------------------------------------------------------
! Dimensions of Physical Grid and array dimensions
!----------------------------------------------------------------------
!
! LLm,MMm  Number of the internal points of the PHYSICAL grid.
!        in the XI- and ETA-directions [physical side boundary
!        points and peroodic ghost points (if any) are excluded].
!
! Lm,Mm    Number of the internal points [see above] of array
!        covering a Message Passing subdomain. In the case when
!        no Message Passing partitioning is used, these two are
!        the same as LLm,MMm.
!
! N        Number of vertical levels.
!
     integer  LLm,Lm,MMm,Mm,N, LLm0,MMm0
#if defined AGRIF
     integer LLmm2, MMmm2
#endif
#if defined BASIN
     parameter (LLm0=60,   MMm0=50,   N=10)
#elif defined CANYON_A
     parameter (LLm0=65,   MMm0=48,   N=16)
#elif defined CANYON_B
     parameter (LLm0=66,   MMm0=48,   N=16)
#elif defined EQUATOR
     parameter (LLm0=40,   MMm0=32,   N=32)   ! 100 km resolution
```

```
#elif defined KH_INST
....
#  elif defined  BENGUELA_LR
     parameter (LLm0=41,   MMm0=42,   N=32)   ! BENGUELA_LR
#  elif defined  BENGUELA_HR
     parameter (LLm0=83,   MMm0=85,   N=32)   ! BENGUELA_HR
#  elif defined  BENGUELA_VHR
     parameter (LLm0=167,  MMm0=170,  N=32)   ! BENGUELA_VHR
#  elif defined MENOR
     parameter (LLm0=1059, MMm0=447,  N=40)   ! MENOR
#  elif defined SEINE
     parameter (LLm0=411,  MMm0=181,  N=20)   ! SEINE
#  else
     parameter (LLm0=94,   MMm0=81,   N=40)
#  endif
#else
     parameter (LLm0=xx, MMm0=xx, N=xx)
#endif
...
!
...
```

Configuring the model

Domain decomposition for parallelization : param.h

```
!
!----------------------------------------------------------------------
! MPI related variables
!----------------------------------------------------------------------
!
      integer Lmmpi,Mmmpi,iminmpi,imaxmpi,jminmpi,jmaxmpi
      common /comm_setup_mpi1/ Lmmpi,Mmmpi
      common /comm_setup_mpi2/ iminmpi,imaxmpi,jminmpi,jmaxmpi
!
! Domain subdivision parameters
! ====== =========== ==========
!
! NPP          Maximum allowed number of parallel threads;
! NSUB_X,NSUB_E  Number of SHARED memory subdomains in XI- and
!                                 ETA-directions;
! NNODES       Total number of MPI processes (nodes);
! NP_XI,NP_ETA  Number of MPI subdomains in XI- and ETA-directions;
!
      integer NSUB_X, NSUB_E, NPP
#ifdef MPI
      integer NP_XI, NP_ETA, NNODES
      parameter (NP_XI=1,  NP_ETA=4,  NNODES=NP_XI*NP_ETA)
      parameter (NPP=1)
      parameter (NSUB_X=1, NSUB_E=1)
#elif defined OPENMP
      parameter (NPP=4)
# ifdef AUTOTILING
      common/distrib/NSUB_X, NSUB_E
# else
      parameter (NSUB_X=1, NSUB_E=NPP)
# endif
#else
      parameter (NPP=1)
# ifdef AUTOTILING
      common/distrib/NSUB_X, NSUB_E
# else
      parameter (NSUB_X=1, NSUB_E=NPP)
# endif
#endif
....
```

Compiling the model

For that use the the jobcomp bash file ./jobcomp.bash

1. Set library path
2. Automatic selection of option accordingly the platform used
3. Use of makefile
   - C-preprocessing step : .F → .f using the CPP keys definitions (in cppdefs.h file, "customization" of the code)
   - Compilation step : .f → .o (object) using Fortran compiler
   - Linking step : link all the .o file and the library (Netcdf, MPI, AGRIF) --
   - --> produce the executable croco

Input file croco.in

Documentation :

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

```
title:
      BENGUELA TEST MODEL
time_stepping: NTIMES   dt[sec]  NDTFAST  NINFO
           8640      1200      60     1
time_stepping_nbq: NDTNBQ    CSOUND_NBQ    VISC2_NBQ
                1        1000        0.01
S-coord: THETA_S,   THETA_B,    Hc (m)
         7.0d0     2.0d0      200.0d0
run_start_date:
01/04/2014 00:00:00
run_end_date:
01/01/2016 00:00:00
output_time_steps: DT_HIS(H), DT_AVG(H), DT_RST(H)
               1       6       12
grid:  filename
   CROCO_FILES/croco_grd.nc
forcing: filename
   CROCO_FILES/croco_frc.nc
bulk_forcing: filename
   CROCO_FILES/croco_blk.nc
climatology: filename
   CROCO_FILES/croco_clm.nc
boundary: filename
   CROCO_FILES/croco_bry.nc
initial: NRREC  filename
        1
   CROCO_FILES/croco_ini.nc
restart:          NRST, NRPFRST / filename
               2160    -1
   CROCO_FILES/croco_rst.nc
history: LDEFHIS, NWRT, NRPFHIS / filename
         T     72     0
   CROCO_FILES/croco_his.nc
averages: NTSAVG, NAVG, NRPFAVG / filename
         1     72     0
   CROCO_FILES/croco_avg.nc

primary_history_fields: zeta UBAR VBAR  U  V   wrtT(1:NT)
                    T    T   T   T  T    30*T
auxiliary_history_fields:   rho Omega  W  Akv  Akt  Aks Visc3d Diff3d  HBL HBBL Bostr Wstr
Ustr Vstr Shfl Swfl rsw rlw lat sen  HEL
                      F   F    T   F    T    F    F     F      T   T    T     T    T    T    T    T   10*T
gls_history_fields:   TKE  GLS  Lscale
                  T     T    T

primary_averages: zeta UBAR VBAR  U  V   wrtT(1:NT)
               T    T    T     T  T   30*T
auxiliary_averages: rho Omega  W  Akv  Akt  Aks  Visc3d Diff3d HBL HBBL Bostr Wstr Ustr
Vstr Shfl Swfl rsw rlw lat sen HEL
                 F   T     T   F    T    F     F     F      T   T    T     T   T    T     T    T   10*T
gls_averages:   TKE  GLS  Lscale
             T     T    T
rho0:
     1025.d0
lateral_visc:   VISC2,    VISC4    [m^2/sec for all]
             0.      0.
tracer_diff2: TNU2(1:NT)          [m^2/sec for all]
            30*0.d0

tracer_diff4: TNU4(1:NT)          [m^4/sec for all]
            30*0.d11

vertical_mixing: Akv_bak, Akt_bak [m^2/sec]
               0.d0    30*0.d0

bottom_drag:     RDRG [m/s],  RDRG2,  Zob [m],  Cdb_min, Cdb_max
             3.0d-04      0.d-3    0.d-3     1.d-4    1.d-1
gamma2:
             1.d0
sponge:         X_SPONGE [m],    V_SPONGE [m^2/sec]
                XXX              XXX

nudg_cof:    TauT_in, TauT_out, TauM_in, TauM_out  [days for all]
             1.      360.      3.      360.
```

./croco croco.in

```
Southern Benguela
     480  ntimes   Total number of timesteps for 3D equations.
 5400.00  dt       Timestep [sec] for 3D equations
      60  ndtfast  Number of 2D timesteps within each 3D step.
       1  ninfo    Number of timesteps between runtime diagnostics.
...
Activated C-preprocessing Options:
...
Spherical grid detected
hmin          hmax          grdmin          grdmax          Cu_min      Cu_max
 75.000000  4803.032721 .301836927E+05 .331215714E+05  0.12176008  0.91533005
 volume=9.5239860932610875000000E+14   open_cross=6.1048368883124446868900E+09
...
MAIN: started time-steping.
STEP   time[DAYS] KINETIC_ENRG    POTEN_ENRG    TOTAL_ENRG    NET_VOLUME   trd
   0     0.00000 0.000000000E+00 2.1475858E+01 2.1475858E+01 9.5239861E+14  0
   1     0.06250 1.306369099E-04 2.1476230E+01 2.1476361E+01 9.5239208E+14  0
...
```

Courant number: evaluation of the CFL criterion: dx/dt > fastest waves (here gravity waves).

Cu_max < 1 !!!

Running the model : Multi-months launching

In many studies, there is a need for long simulations : spin-up , statistical equilibrium.

For regional models, 10 years appears to be a reasonable model duration.

In tis case, it is easier to do several simulations of 1 month, using restart at the end of each month to start the next month.

run_croco.bash for climatological forcing or **run_croco_inter.bash** for interannual forcing do it automatically

- It get the grid, the initial file and the boundary file
- In case of a interannual forcing, we have realistic forcing, segmented month by month, from various atmospheric/oceanic reanalysis [for example : croco_ini_SODA_Y2004M1.nc, croco_frc_CFSR_Y2004M1.nc or croco_bry_ECCO_Y2004M1.nc]
- It runs the model for 1 month
- It store the files in a specific for in the directory SCRATCH
    - For climatological forcing : roms_avg_Y4M3.nc (i.e march of year 4)
    - For interannual forcing : roms_avg_Y2004M3.nc (i.e march of year 2004)
- It replace the initial file by the restart file (croco_rst.nc) which has been generated at the end of the month.
- It relaunch the model for the next month

Multi-months simulations with climatological forcings

Documentation :
https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.10.run.inter.html

11. Running with interannual forcing

11.1. Run after classical interannual pre-processing

Before running you should prepare your interannual inputs files following the Interannual Preprocessing tutorial.

To run a plurimonth simulation, we provide the following scripts in `~/croco/croco/SCRIPTS/Plurimonths_scripts`:

- `run_croco.bash` : Plurimonth run with climatological forcing

Multi-months simulations with interannual forcings

Documentation :
https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.10.run.inter.html

11. Running with interannual forcing

11.1. Run after classical interannual pre-processing

Before running you should prepare your interannual inputs files following the Interannual Preprocessing tutorial.

To run a plurimonth simulation, we provide the following scripts in `~/croco/croco/SCRIPTS/Plurimonths_scripts`:

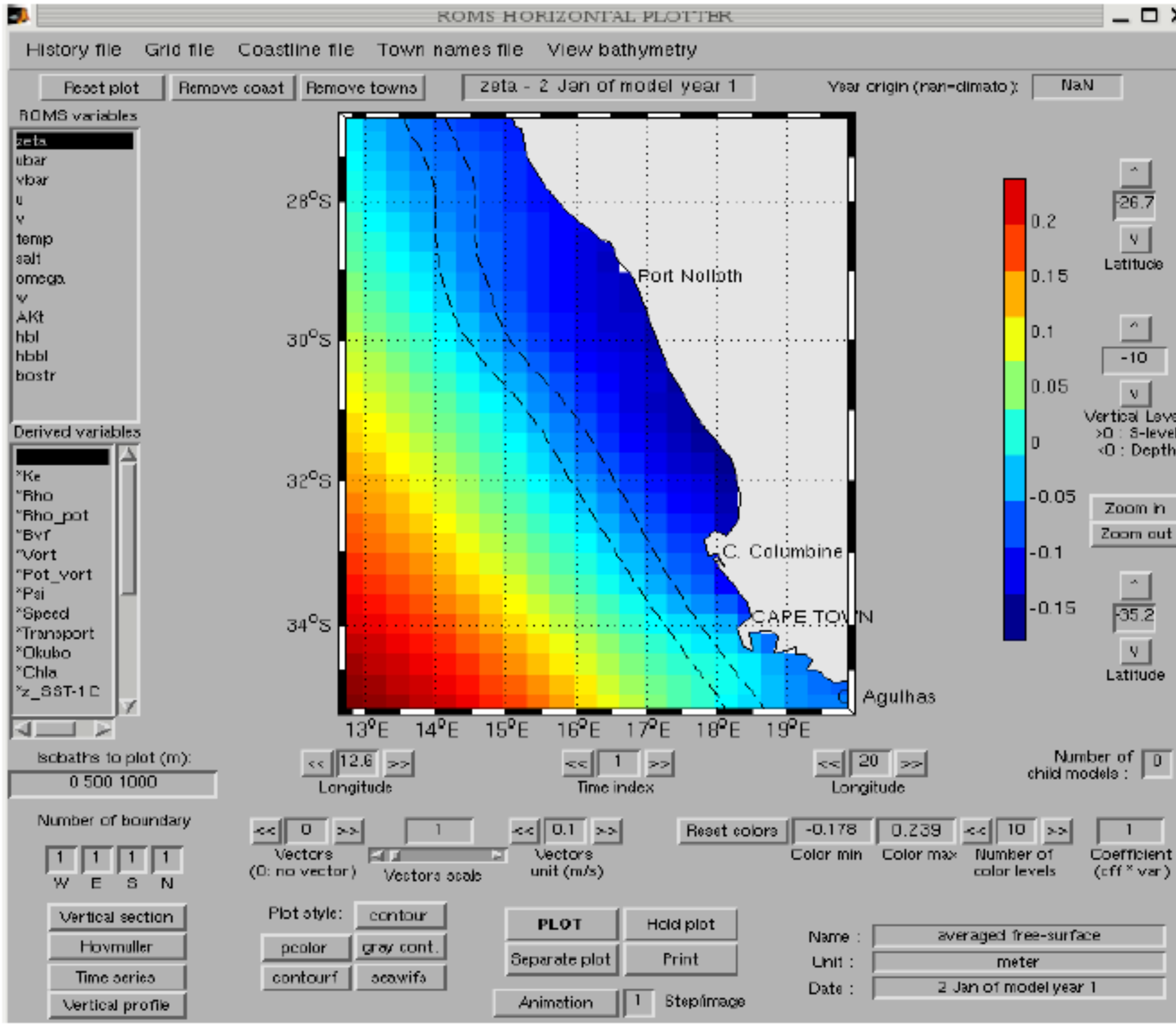
- `run_croco_inter.bash`: Plurimonth run with interannual forcing

Visualization

CROCO
Coastal and Regional Ocean COmmunity model

Documentation :
https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.14.visu.matlab.html

```
$matlab
>> croco_gui
```

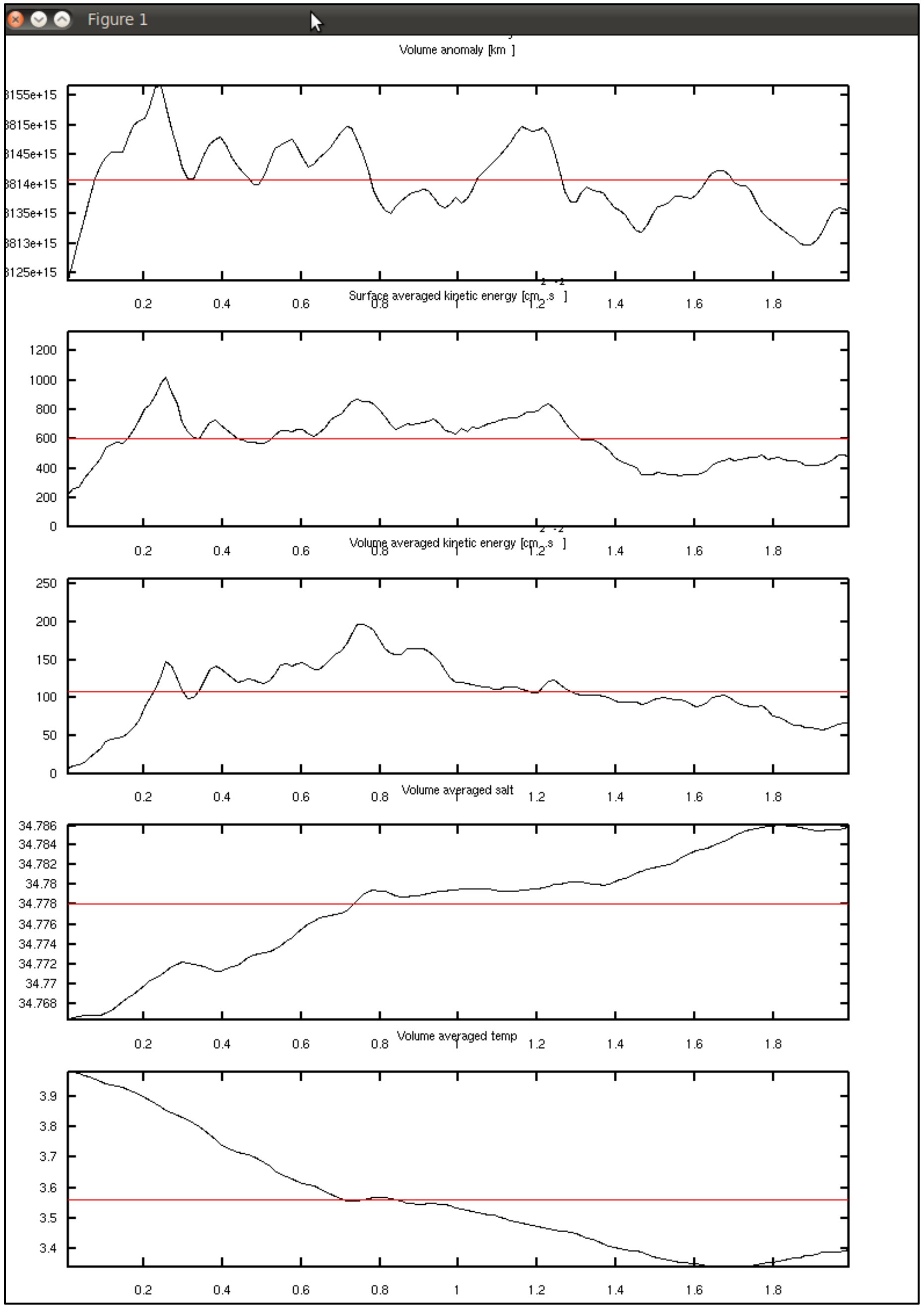


Global Diagnostics

CROCO
Coastal and Regional Ocean COmmunity model

Scripts for the simulation analysis in croco_tools/Diagnostics_tools

croco_diags	Get volume and surface averaged quantities
plot_diags	Plot averaged quantities
get_Mmean	Get monthly mean climatology
get_Smean	Get seasonal and annual mean climatology
get_Meddy	Get seasonal and annual variance climatology
get_Seddy	Get seasonal and annual variance climatology



Adding Tides

Documentation :
https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.13.tides.html

The tides are imposed at open boundaries using the characteristic open boundary conditions (define **OBC_M2CHARACT** cpp-keys)

- ξ_{tides}, $\overline{u}_{tides}$, $\overline{u}_{tides}$: ssh and depth averaged zonal and meridian currents are added at the open boundaries
- ξ_{tides}, $\overline{u}_{tides}$, $\overline{u}_{tides}$ are computed from the tidal harmonics given by some tidal model, in our case TPXO9 (0.25° resolution, 10 tidal components : M2, N2,S2,K2, K1, 01,P1, Q1, Lm, Mm)

- The global tidal model gives harmonics constants for all the principal tidal waves. These constants permits to compute at every time t, $\xi^N_{tides}(t)$, $\overline{u}^N_{tides}(t), and\ \overline{u}^N_{tides}(t)$ of the tidal wave component N.

You need :

- Choose the number of tidal wave component you want
- Interpolate on the grid the different harmonic constants
- Possibility to add the generator potential

Adding Tides

```
matlab >> make_tides

Start date for nodal correction : 1-Jan-2000
Reading CROCO grid parameters ...
Tidal components : M2 S2 N2 K2 K1 O1 P1 Q1 Mf Mm
Processing tide : 1 of 10
  ssh...
Getting ssh_r for time index 1
Getting ssh_i for time index 1
  u...
Getting u_r for time index 1
Getting u_i for time index 1
  v...
Getting v_r for time index 1
Getting v_i for time index 1
Convert to tidal ellipse parameters...
Process equilibrium tidal potential...
Process tidal loading and self-attraction potential...
Get total tidal potential...
Processing tide : 2 of 10
  ssh...
Getting ssh_r for time index 2
Getting ssh_i for time index 2
  u...
Getting u_r for time index 2
Getting u_i for time index 2
  v...
Getting v_r for time index 2
Getting v_i for time index 2
Convert to tidal ellipse parameters...
Process equilibrium tidal potential...
Process tidal loading and self-attraction potential...
Get total tidal potential...
```

The tidal forcings are added in the croco_frc.nc file.

To define the tides, in cppdefs.h

```
# ifdef TIDES
#  define SSH_TIDES
#  define UV_TIDES
#  define POT_TIDES
#  undef  TIDES_MAS
#  ifndef UV_TIDES
#   define OBC_REDUCED_PHYSICS
#  endif
#  define TIDERAMP
# endif
```

Adding Tides

Tidal forcing parameter & temporal parameters

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% 5 - Parameters for tidal forcing
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% TPXO file name (TPXO6 or TPXO7)
%
tidename=[DATADIR,'TPXO7/TPXO7.nc'];
%
% Self-Attraction and Loading GOT99.2 file name
%
sal_tides=1;
salname=[DATADIR,'GOT99.2/GOT99_SAL.nc'];
%
% Number of tides component to process
%
Ntides=10;
%
% Chose order from the rank in the TPXO file :
% "M2 S2 N2 K2 K1 O1 P1 Q1 Mf Mm"
% " 1  2  3  4  5  6  7  8  9 10"
%
tidalrank=[1 2 3 4 5 6 7 8 9 10];
%
% Compare with tidegauge observations
%
lon0 =  18.37;   % Example:
lat0 = -33.91;   % Cape Town location
Z0   =  1;       % Mean depth of tide gauge
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

For real time runs, with tides forcing , a procedure correct phases and amplitudes (nodal corrections) (Egbert and Erofeeva (2002))

Adding a biogeochemical forcing

In CROCO , several biogeochemical model : 1D vertical equation (subroutine biology.F)

- Nitrogen -based model of increasing complexity:
  - **NPZD**
  - **N2PZDD2**
  - **BIOBUS**
- Not only Nitrogen-based (including Iron, etc..): **PISCES**

For initial and lateral boundary condition :

- Seasonal climatology of nitrate provided from WOA (2005 or 2009)
- Seasonal climatology of surface chlorophyll provided from SeasWifs

Adding a biological forcings

Choose the BGC model you want (section Applications in cppdefs.h)

```
# define BIOLOGY
…
 /*   Choice of Biology models   */
# ifdef BIOLOGY
#  undef  PISCES
#  undef BIO_NChlPZD
#  undef  BIO_N2ChlPZD2
#  define BIO_BioEBUS
                    /*   Biology options    */
#  ifdef PISCES
#   undef  DIURNAL_INPUT_SRFLX
#   define key_pisces
#  endif
#  ifdef BIO_NChlPZD
#   define  OXYGEN
#  endif
#  ifdef BIO_BioEBUS
#   define NITROUS_OXIDE
#  endif
```

Adding a biological forcings

croco_tools :

- make_biol : for climatology file
- make_bry_npzd :
  _bioebus :
  _pisces : for boundary files

crocotools_param.m

```
%
% Pisces biogeochemical seasonal climatology
%
woapisces_dir = [DATADIR,'WOAPISCES/'];  % only
compatible with woa_dir
%
% Surface chlorophyll seasonal climatology (SeaWifs)
%
chla_dir=[DATADIR,'SeaWifs/'];
%
```

```
> make_biol
Add_no3: creating variables
and attributes for the OA file
write no3time
Add_no3: creating variables
and attributes for the
Climatology file

 Ext tracers: Roa = 0 km -
default value = NaN
 Ext tracers: horizontal
interpolation of the annual
data
 Ext tracers: horizontal
interpolation of the seasonal
data
time index: 1 of total: 4
time index: 2 of total: 4
time index: 3 of total: 4
time index: 4 of total: 4


 ….
```

Adding a river forcing

Documentation :
https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.12.rivers.html

Section Applications in cppdefs.h

```
/* Point Sources - Rivers */
# undef PSOURCE
# define PSOURCE_NCFILE
# ifdef PSOURCE_NCFILE
#   define PSOURCE_NCFILE_TS
# endif
```

crocotools_param.m

```
%
% Runoff monthly seasonal climatology (Dai and
Trenberth)
%
global_clim_riverdir=[DATADIR,'RUNOFF_DAI/'];
global_clim_rivername=[global_clim_riverdir,'Dai_Trenb
erth_runoff_global_clim.nc'];
%
%  Set times and cycles for the boundary conditions:
%   monthly climatology
%%
%   Set times and cycles for runoff conditions:
%   monthly climatology
%
qbar_time=[15:30:365];
qbar_cycle=360;
%
%   Tracer runoff concentration processing flag
%     pource_ts = 1  => Runoff tracers concentration
processing is activated.
%                      It needs the climatology file created with
make_clim.m
%     psource_ts = 0 => No Runoff tracers concentration
processing
%                      It reads analytical values in croco.in
%                      or use default value defined in analytical.F
%
psource_ts=0;
%
```

Adding a river forcing

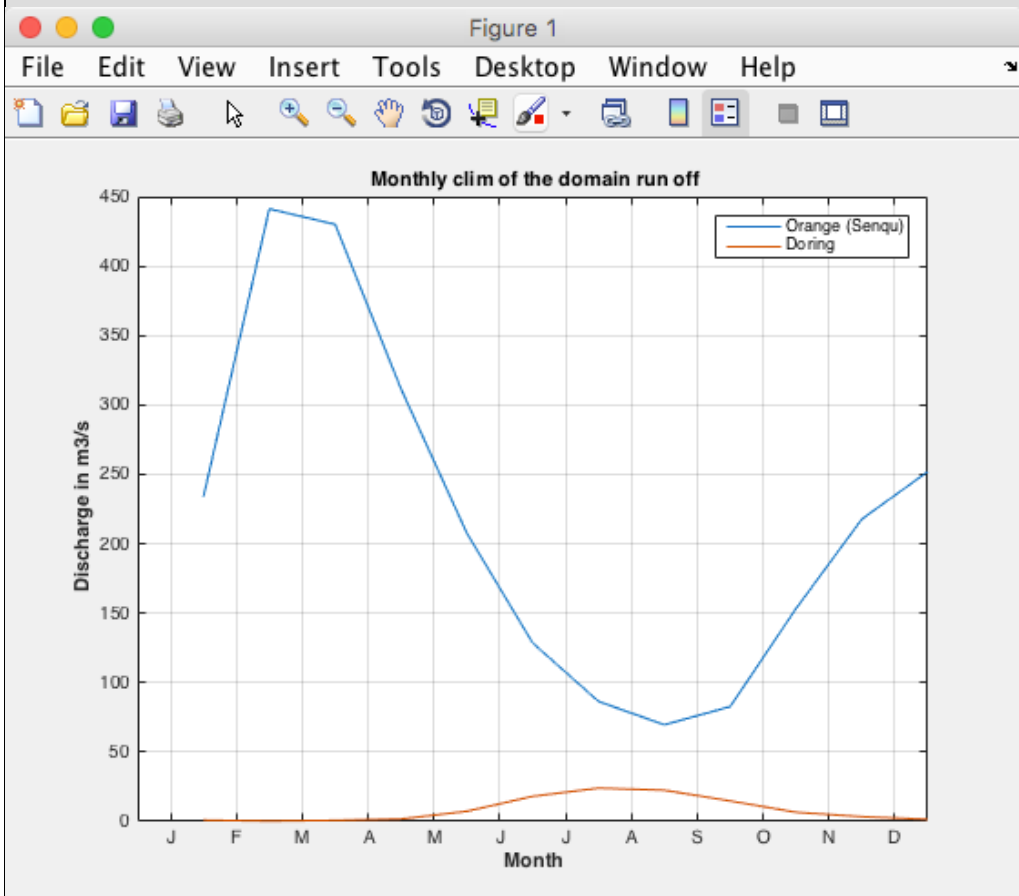
croco_tools : **make_runoff.m**

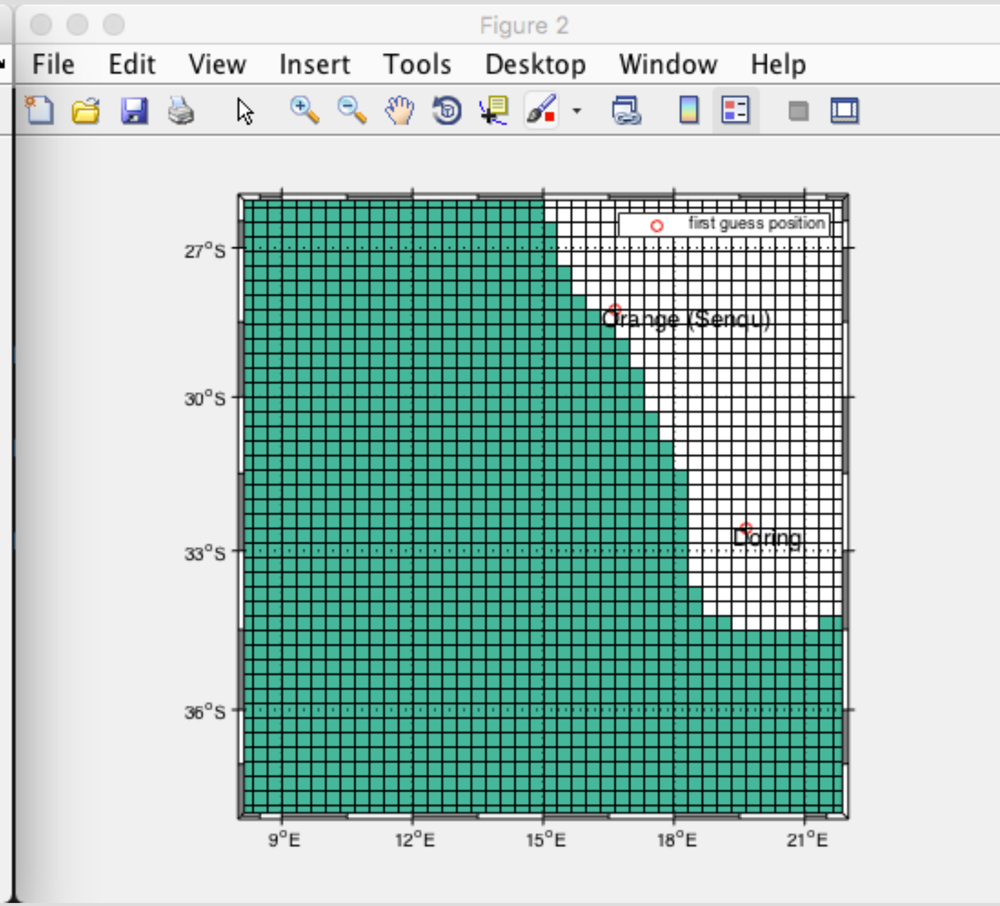
```
> In make_runoff (line 29)
mkdir:
/Users/gcambon/DATA/for_asi18_sess2/CONFIGS/BENGUELA_LR/CROCO_FILES/: File exists

Create runoff forcing from Dai and
Trenberth's global monthly climatological
run-off dataset

Reading the global monthly climatological
run-off dataset...

There are 2 rivers in the domain :
Domain contains rivers :
1 - Orange (Senqu)
flowing in ocean ATL
2 - Doring
flowing in ocean ATL

First guess:
============

- Process river #1: Orange (Senqu)
  Position is approximetly  J=35 and I=26
lon src in grid (rho point) ~16.3333
lat src in grid (rho point) ~-28.5604

- Process river #2: Doring
  Position is approximetly  J=20 and I=35
lon src in grid (rho point) ~19.3333
lat src in grid (rho point) ~-32.8514
Do you want to use river (Yes[1], No[0])
?  Orange (Senqu)

…
...
```

Adding a river forcing

croco_tools : **make_runoff.m**

- \> In make_runoff (line 29)
- mkdir:
- – Process river #2: Doring



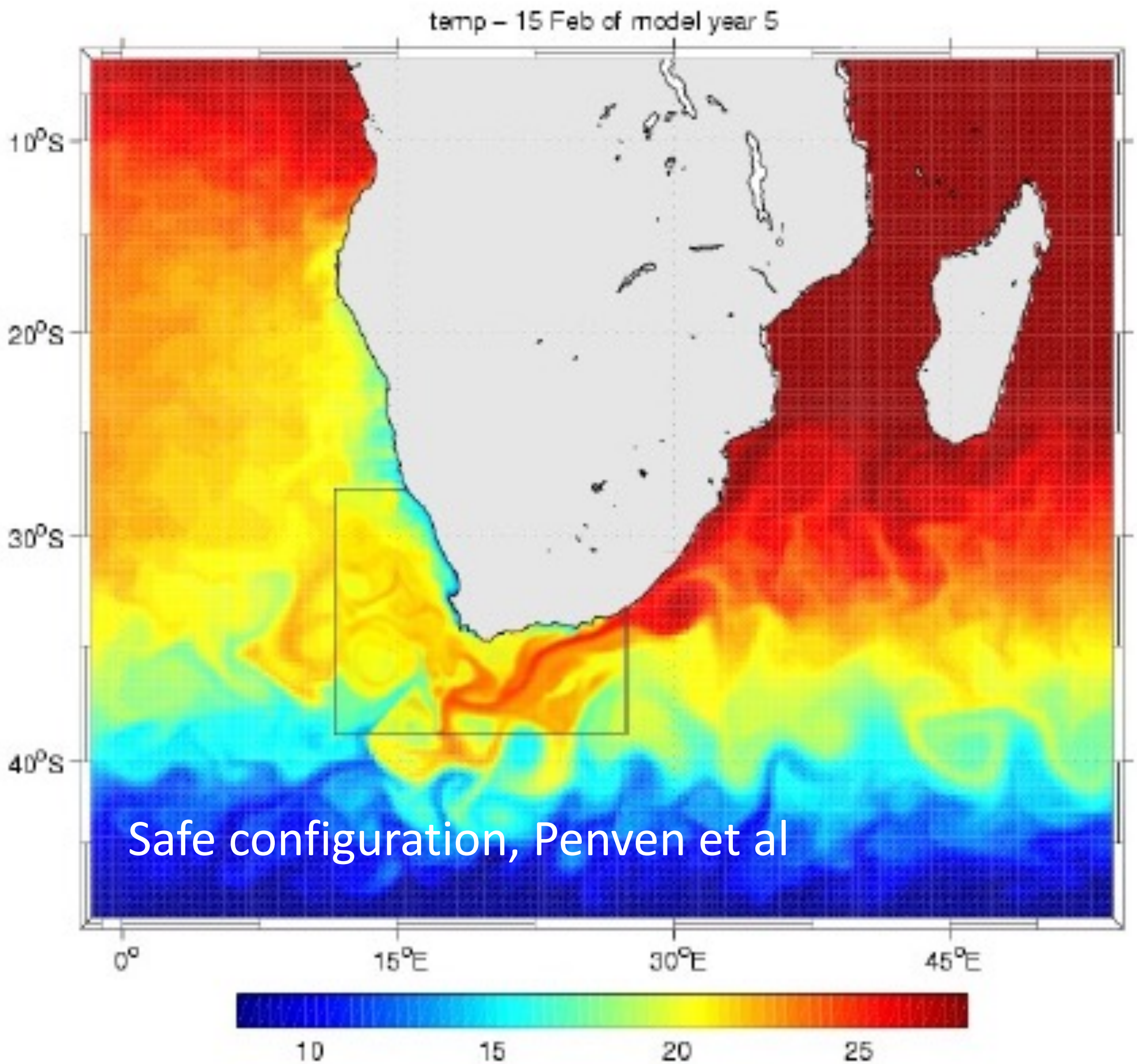


- – Process river #1: Orange (Senqu)
- Position is approximetly J=35 and I=26
- lon src in grid (rho point) ~16.3333
- lat src in grid (rho point) ~-28.5604

Adding an AGRIF nest (online nesting)

Documentation :
https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.11.nesting.html

Example :



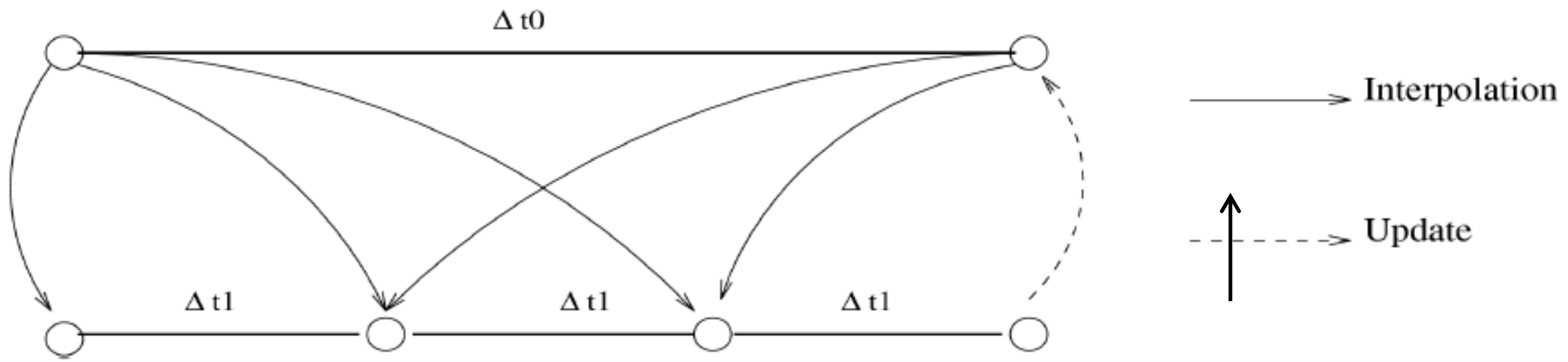
Nesting capability added to ROMS

→ AGRIF package

→ Adaptive Mesh Refinement

→ Manage arbitrary number of fixed grid and embedding level

Temporal coupling between a parent and a child grid for a refinement factor of 3 :



Needs to run an embedded model : Surface forcing and initial conditions datas files.

AGRIF names the different datas files as :

Parent file names : XXX.nc → First child file names : XXX.nc.1
second child file names : XXX.nc.2
. . .

Adding an AGRIF nest (online nesting)

CROCO
Coastal and Regional Ocean COmmunity model

With matlab

In the benguela test case, for the parent grid file, select in the entrance window of NestGUI and click 'open '

Follow the steps :

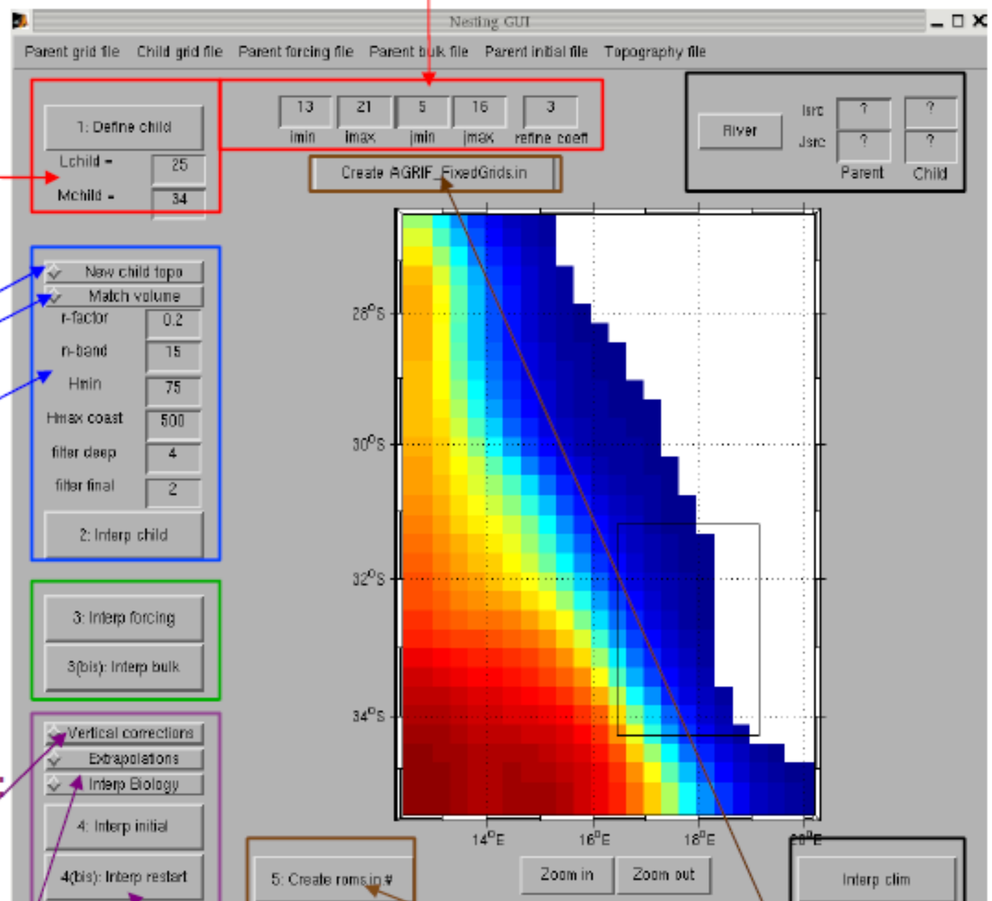
1- Tune the child domain

1- Define the child domain :
Size of the child grid

2- Create the child grid file :
What topography file?
Child grid volume
Parameters to change
-->

3- Create the surface forcing file:
Select _ _
--> roms_frc.nc.*. or roms_blk.nc.*.

4- Create the initial condition file:
Select roms_ini.nc
If different topography
Interpolate parent biological variables
--> roms_ini.nc.*.



4- Select roms_rst.nc
--> roms_rst.nc.*.

5- Generate roms.in.*.
Create AGRIF_fixedGrids.in

Locate river on the coast

Generate boundary condition to test the child model alone

Adding an AGRIF nest (online nesting)

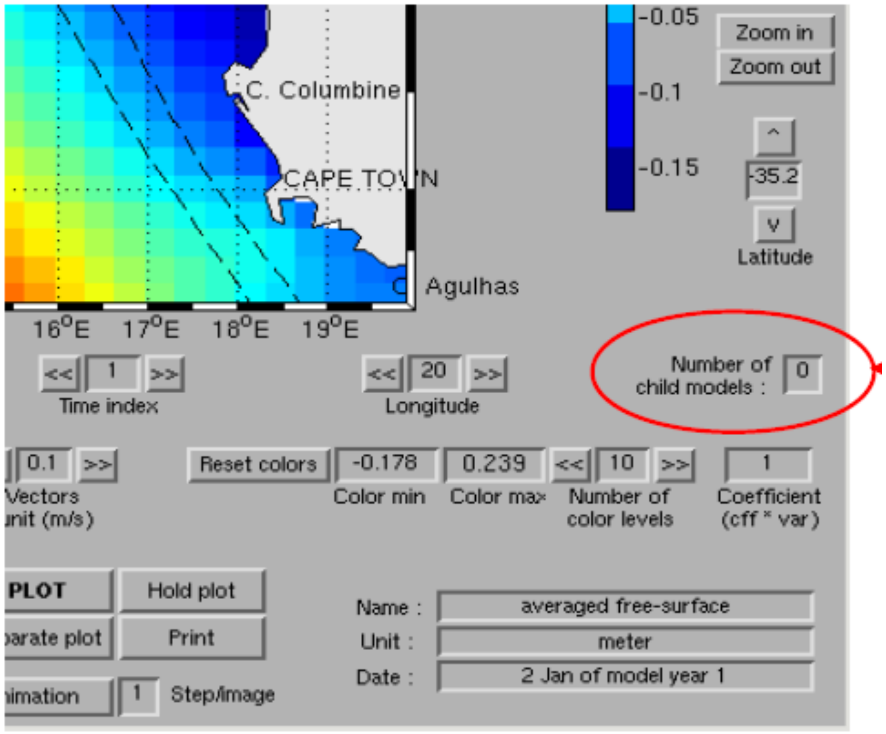
- To run a simulation with nesting, define the CPP keys and compile (./jobcomp)
  - AGRIF
  - AGRIF_2W

- Position of the different grid in AGRIF_FixedGrids.in file

```
  1
  23   37   12   29   3   3   3   3
  0
# number of children per parent
# imin imax jmin jmax spacerefx spacerefy timerefx timerefy
# [all coordinates are relative to each parent grid!]
```

- Namelist relative to the different nest level croco.in.1, croco.in.2 etc …

- Visualization (in Matlab) :
  - >>matlab
  - >>croco_gui



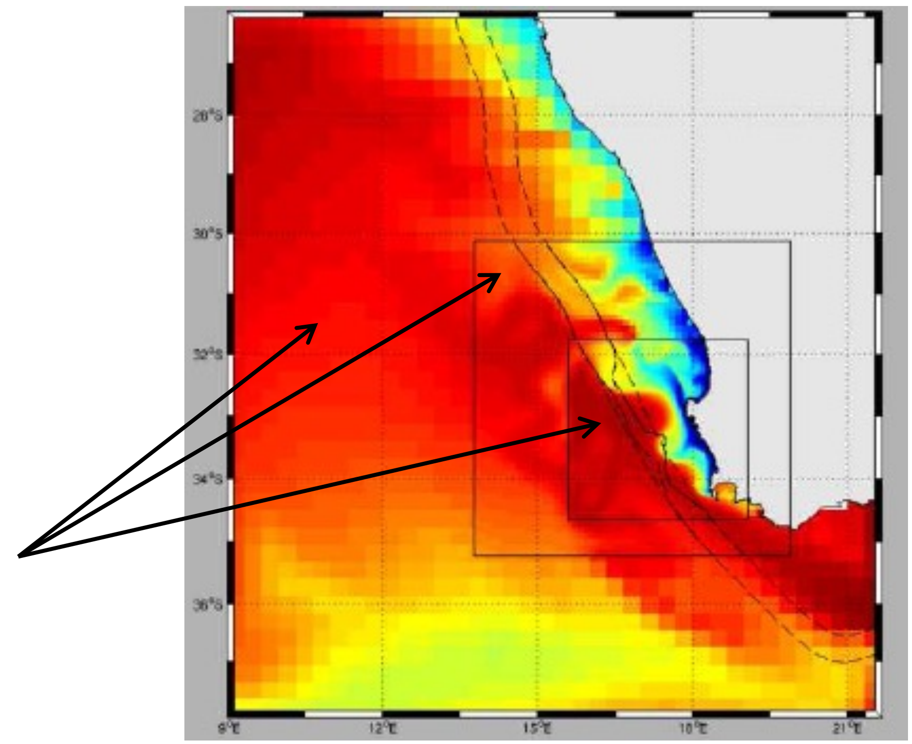
Adding an AGRIF nest (online nesting)

The file Agrif_FixedGrids.in define the position of the nested grid

```
 1
  23   37   12   29   3   3   3   3
  0
# number of children per parent
# imin imax jmin jmax spacerefx spacerefy timerefx timerefy
# [all coordinates are relative to each parent grid!]
```

2 grids : #0 and #1
#1 is embedded in #0

```
 1
  23   37   12   29   3   3   3   3
  1
  12   28   15  33    3   3   3   3
  0
# number of children per parent
# imin imax jmin jmax spacerefx spacerefy timerefx timerefy
# [all coordinates are relative to each parent grid!]
```



Needs to run an embedded model :

3 grids : #0,#1 and #2
#1 embedded in #0 ;
#2 is embedded in the #1

Surface forcing and initial conditions datas files.

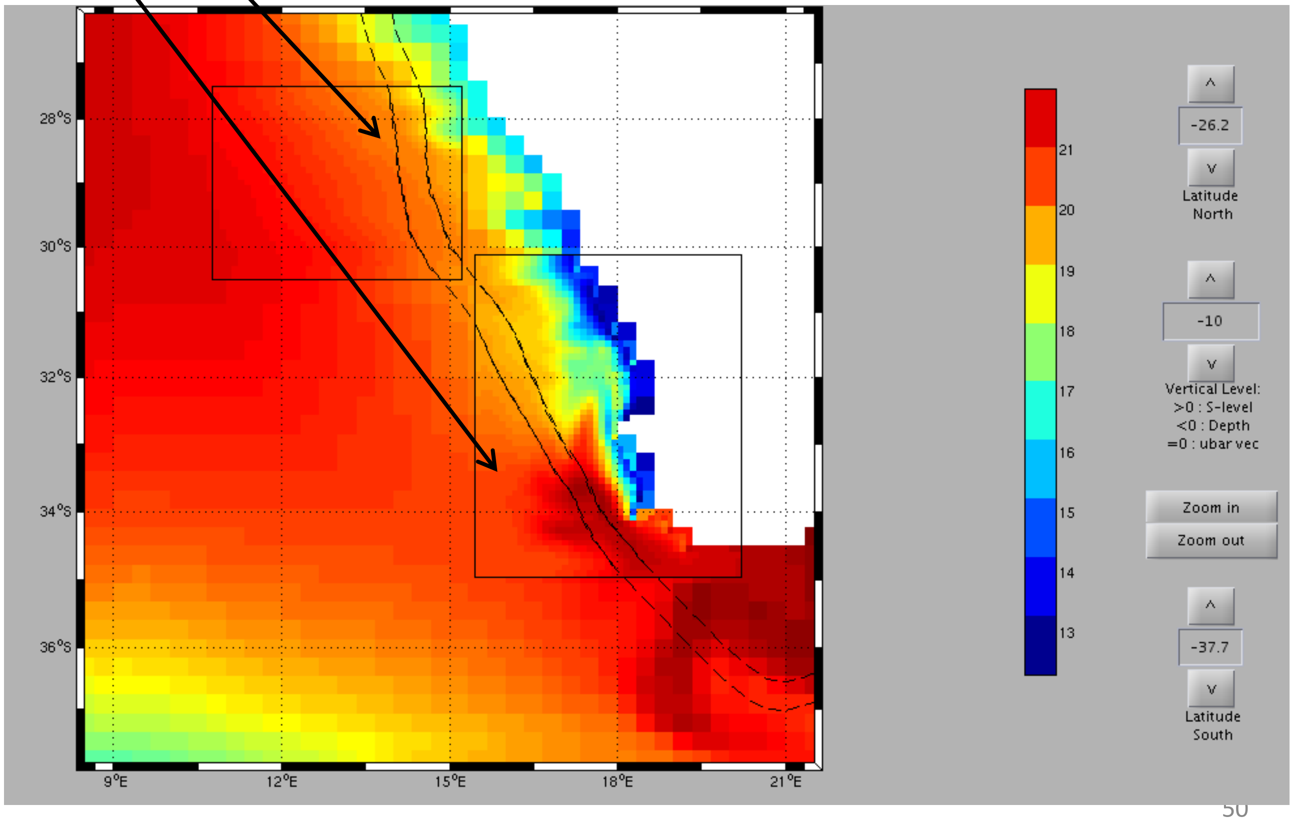
For grid #xx :

- croco_grd.nc.xx
- croco_frc.nc.xx
- croco_blk.nc.xx
- croco.ini.nc.xx
- croco.in.xx

Adding an AGRIF nest (online nesting)

```
  2
  23   37   12   29   3   3   3   3
  9    22   28   38   3   3   3   3
  0
  0
#number of children per parent
# …
```

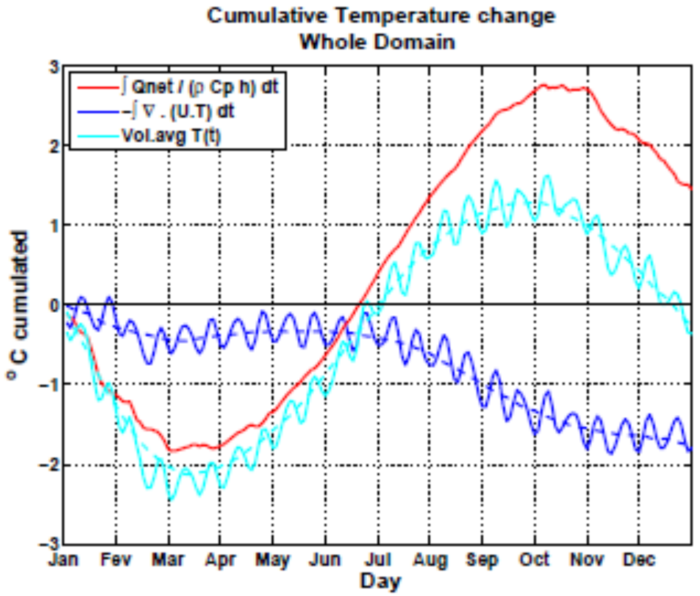
3 grids : #0,#1 and #2
- #1 embedded in #0 ;
- #2 is embedded in #0 : independent grids

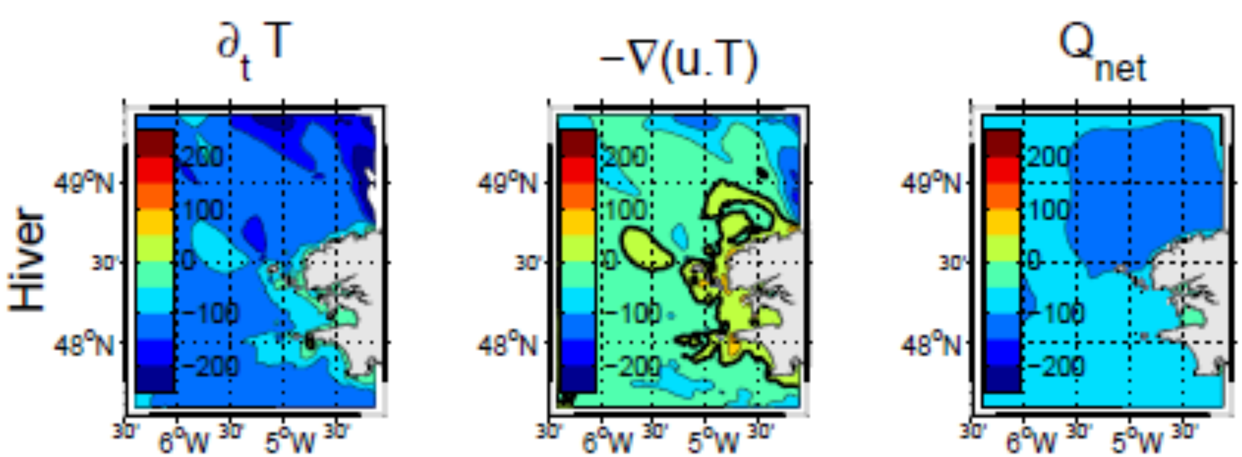


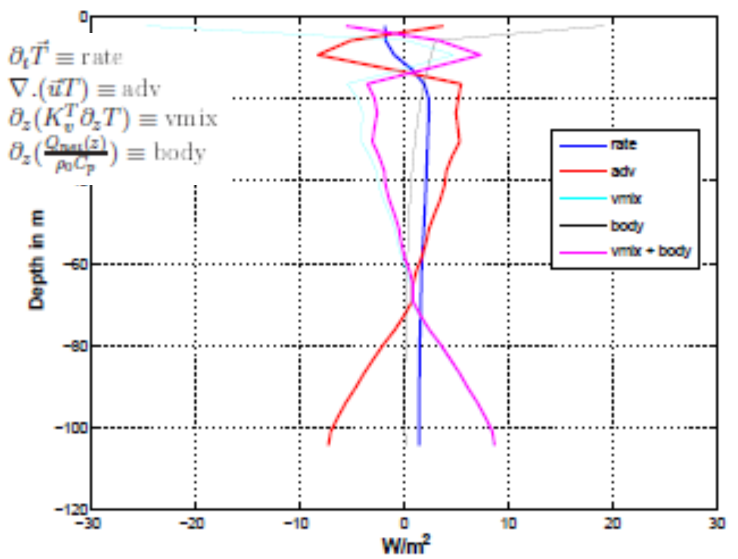
Online Diagnostics : Tracer equations terms

$$\partial_t T + \nabla.(\vec{u}T) = F^T + D^T$$
$$= K_h^T \Delta T + \partial_z\left(K_v^T \partial_z T\right) + \frac{1}{\rho_0 C_p}\partial_z\left(Q_{net}(z)\right)$$

- $\partial_t \vec{T}$: = Time rate (rate)
- $\nabla.(\vec{u}\mathbf{T})$: = Advection (adv)
- $\partial_z\left(K_v^T \partial_z T\right)$: Vert. mixing (vmix)
- $K_h^T \triangle \vec{T}$: Hori. mixing (hmix)
- $\partial_z\left(\frac{Q_{net}(z)}{\rho_0 C_p}\right)$ Solar heating forcing (body)







CROCO
Coastal and Regional Ocean COmmunity model

Getting the terms of the equations, stored in netCDF files

- Momentum equation terms : define CPP keys DIAGNOSTICS_TS
- Tracer equation terms : define CPP keys DIAGNOSTICS_UV
- Biological fluxes terms : define CPP keys DIAGNOSTICS_BIO

In croco.in :

```
diagnostics:   ldefdia   nwrtdia    nrpfdia /filename
                  T       72        0
                                 CROCO_FILES/croco_dia.nc
diag_avg: ldefdia_avg  ntsdia_avg  nwrtdia_avg  nprfdia_avg /filename
               T         1          72           0
                                 CROCO_FILES/croco_dia_avg.nc
diag3D_history_fields:    diag_tracers3D(1:NT)
                          30*T
diag2D_history_fields:    diag_tracers2D(1:NT)
                          30*T
diag3D_average_fields:    diag_tracers3D_avg(1:NT)
                          30*T
diag2D_average_fields:    diag_tracers2D_avg(1:NT)
                          30*T
```

```
diagnosticsM:   ldefdiaM   nwrtdiaM    nrpfdiaM /filename
                   T        72         0
                                 CROCO_FILES/croco_diaM.nc
diagM_avg: ldefdiaM_avg  ntsdiaM_avg  nwrtdiaM_avg  nprfdiaM_avg
/filename
               T         1          72           0
                                 CROCO_FILES/croco_diaM_avg.nc
diagM_history_fields: diag_momentum(1:2)
                          T T
diagM_average_fields: diag_momentum_avg(1:2)
                                 T T
diagnostics_bio:   ldefdiabio   nwrtdiabio    nrpfdiabio /filename
                       T        72           0
                                 CROCO_FILES/croco_diabio.nc
diagbio_avg: ldefdiabio_avg  ntsdiabio_avg  nwrtdiabio_avg
nprfdiabio_avg /filename
                  T          1           72           0
                                 CROCO_FILES/croco_diabio_avg.nc
```

Online Diagnostics : Tracer equations terms

Some precisions on advection

$\nabla.(\vec{u}T)$

$$u\partial_x T \neq \partial_x(u.T)$$

$$only$$

$$\partial_x(uT) + \partial_y(v.T) + \partial_z(w.T) = u\partial_x T + v\partial_y T + w\partial_z u$$

CPPKEYS: DIAGNOSTICS_TS

Storage:

- croco_diaM.nc and croco_diaM_avg.nc files.

- Can choose in croco.in the writing frequency and the terms to store

- By default : the different momentum terms with advection terms in “flux forms” :

$$T_{xadv} = \partial_x(u.T)$$

- With cppkey DIAGNOSTICS_TS_ADV : compute the “advective” forms :

$$T_{xadv-II} = u.\partial_x(T)$$

- with cppkey DIAGNOSTICS_TS_MLD : integration of the differents terms over the mixed layer HBL, computed by the KPP model

Online Diagnostics : Momentum equations terms

CROCO
Coastal and Regional Ocean COmmunity model

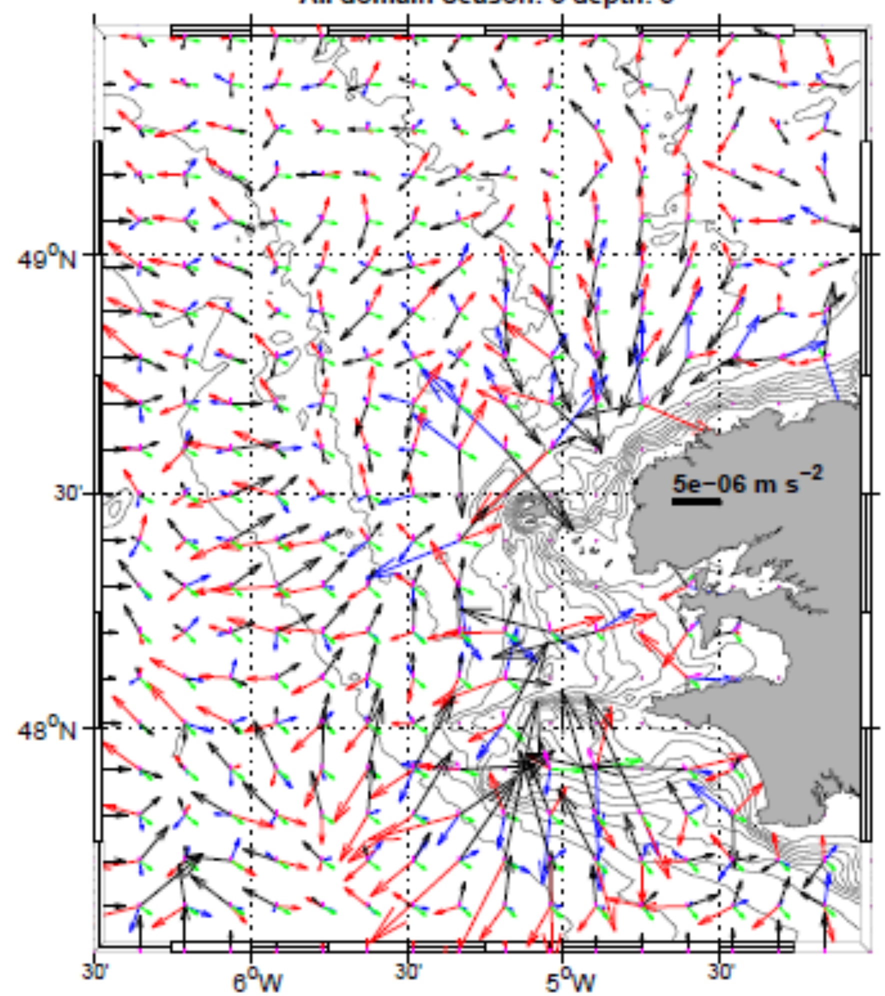
$$\frac{\partial u}{\partial t} + \vec{v} \cdot \nabla u - fv = -\frac{\partial \phi}{\partial x} - \frac{\partial}{\partial z}\left(\overline{u'w'} - \nu \frac{\partial u}{\partial z}\right) + \mathcal{F}_u + \mathcal{D}_u$$

$$\frac{\partial v}{\partial t} + \vec{v} \cdot \nabla v + fu = -\frac{\partial \phi}{\partial y} - \frac{\partial}{\partial z}\left(\overline{v'w'} - \nu \frac{\partial v}{\partial z}\right) + \mathcal{F}_v + \mathcal{D}_v$$

- Rate change term : $\partial_t \vec{u}$
- Coriolis term : $\begin{pmatrix} -fv \\ +fu \end{pmatrix}$
- Advection term : $\vec{u}.\nabla\vec{u}$
- Pressure gradient term : $-\nabla P/\rho_0.$
- Vertical mixing term : $\partial_z\left(K_v \partial_z \vec{u}\right)$
- Horizontal mixing term : $K_H \Delta \vec{u}$

Termes d'accélérations en $m.s^{-2}$:
Advection : $\vec{u}.\nabla\vec{u}$
Coriolis : $f\vec{u}$
Mélange vertical : $\partial_z\left(K_v \partial_z \vec{u}\right)$
Gradient de pression : $-\partial P/\rho_0$
Tendance temporelle : $\partial_t \vec{u}$

All domain Season: 3 depth: 5



Online Diagnostics : Momentum equations terms

$$\vec{u}.\nabla\vec{u} \equiv \begin{pmatrix} u\partial_x u + v\partial_y u + w\partial_z u \\ u\partial_x v + v\partial_y v + w\partial_z v \end{pmatrix} = \begin{pmatrix} \partial_x(u.u) + \partial_y(u.v) + \partial_z(u.w) \\ \partial_x(u.v) + \partial_y(v.v) + \partial_z(v.w) \end{pmatrix}$$

$$= \begin{pmatrix} u_{xadv} + u_{yadv} + u_{vadv} \\ v_{xadv} + v_{yadv} + v_{vadv} \end{pmatrix}$$

Formulation flux

BUT take care :

$$u\partial_x u \neq \partial_x(u.u)$$

$$only$$

$$\left(\partial_x(u.u) + \partial_y(u.v) + \partial_z(u.w)\right) = \left(u\partial_x u + v\partial_y u + w\partial_z u\right)$$

CPPKEYS: DIAGNOSTICS_UV