Ocean model ROMS_AGRIF
&
Processing-tools ROMSTOOLS

http://roms.mpl.ird.fr

Gildas Cambon, Pierrick Penven,
Patrick Marchesiello, Laurent Debreu

gildas.cambon@ird.fr

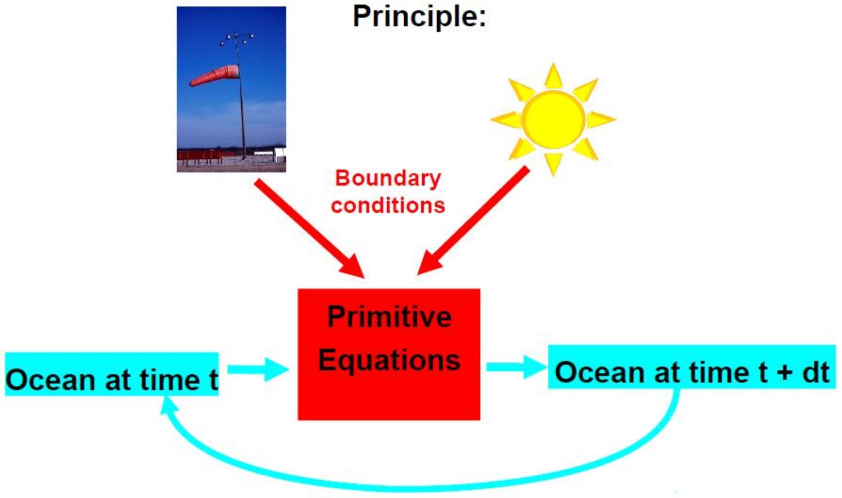
Ocean modeling principle

If we know:

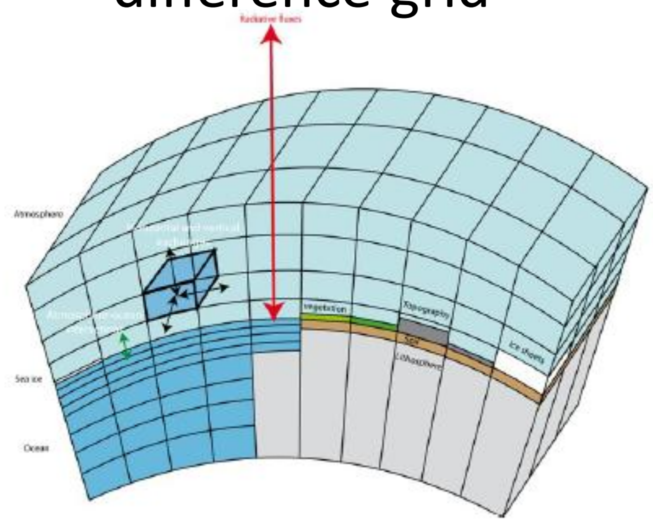
- The ocean state at time t (u,v,w,T,S, …)
- Boundary conditions (surface, bottom, lateral sides)

We can compute the ocean state at t+dt using numerical approximations of Primitive Equations

For that we need to proceed **discretization** of the equations in time and space:



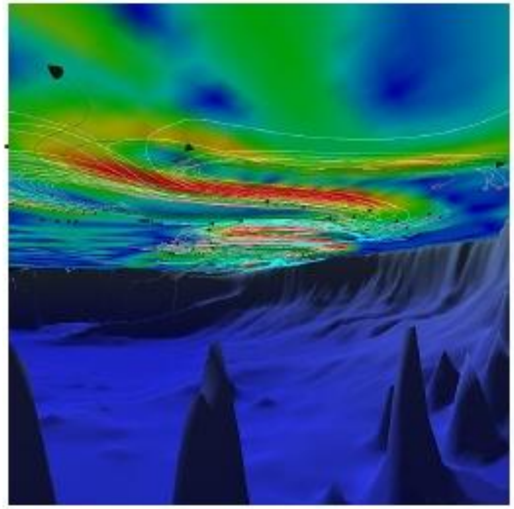
Example of a finite difference grid



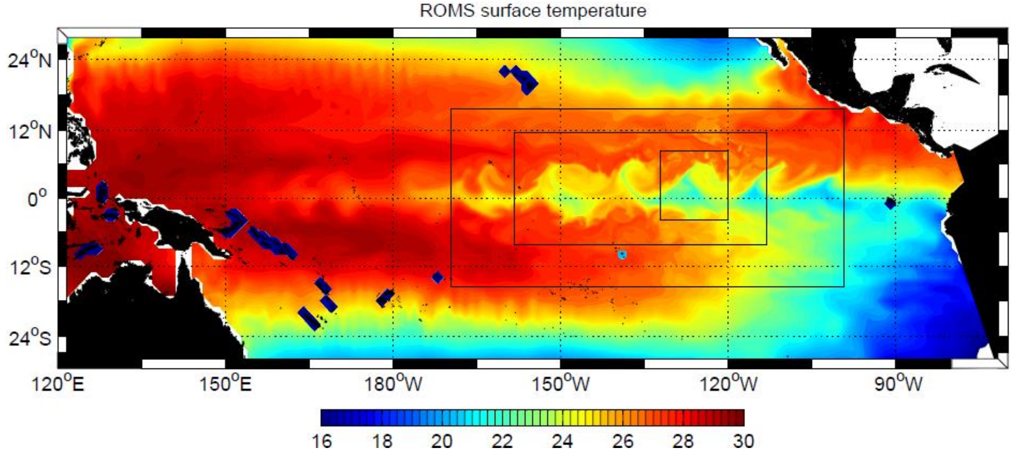
Introduction

Why use of ocean model :

- Cost effective
- Synoptic view
- Equilibrium diagnostics
- Test hypothesis
- Hindcast and forecast
- Coupling with different models
- …



Downloaded from Miami Isopycnic Coordinate model website



4-AGRIF nested grid, SST. *Marchesiello et al, Ocean modelling, 2011.*

Introduction

Equations to solve:

- Navier Stoke equations but with approximations !

- Hypothesis:
  - ✓ Hydrostatic : $H/L \ll 1$, (aspect ratio low)
    - neglect vertical acceleration
    - neglect Coriolis term associated to vertical velocities
  - ✓ Water is incompressible
  - ✓ Boussinesq : $\rho=\rho_o=cste$ for horizontal gradient pressure

Primitive equations

Primitive equations

Momentum conservation

$$\frac{\partial u}{\partial t} + \vec{u}.\nabla u - fv = -\frac{1}{\rho_0}\frac{\partial P}{\partial x} + \nabla_h\left(K_{Mh}.\nabla_h u\right) + \frac{\partial}{\partial z}\left(K_{Mv}\frac{\partial u}{\partial z}\right)$$

$$\frac{\partial v}{\partial t} + \vec{u}.\nabla v + fu = -\frac{1}{\rho_0}\frac{\partial P}{\partial y} + \nabla_h\left(K_{Mh}.\nabla_h v\right) + \frac{\partial}{\partial z}\left(K_{Mv}\frac{\partial v}{\partial z}\right)$$

Hydrostatic

$$0 = -\frac{\partial P}{\partial z} - \rho g$$

Continuity

$$0 = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}$$

Tracer conservation

$$\frac{\partial T}{\partial t} + \vec{u}.\nabla T = \nabla_h\left(K_{Th}.\nabla_h T\right) + \frac{\partial}{\partial z}\left(K_{Tv}\frac{\partial T}{\partial z}\right)$$

$$\frac{\partial S}{\partial t} + \vec{u}.\nabla S = \nabla_h\left(K_{Sh}.\nabla_h S\right) + \frac{\partial}{\partial z}\left(K_{Sv}\frac{\partial S}{\partial z}\right)$$

Equation of state

$$\rho = \rho(S, T, p)$$

Primitive equations : Boundary conditions

With surface boundary conditions (z=η):

$$\frac{\partial \eta}{\partial t} = w$$

Kinematic

$$K_{Mv}\frac{\partial u}{\partial z} = \frac{\tau_x}{\rho_0}$$

$$K_{Mv}\frac{\partial v}{\partial z} = \frac{\tau_y}{\rho_0}$$

Wind stress

$$K_{Tv}\frac{\partial T}{\partial z} = \frac{Q}{\rho_0 C_p}$$

Heat flux

$$K_{Sv}\frac{\partial S}{\partial z} = \frac{S(E-P)}{\rho_0}$$

Salt flux : evap - rain

And bottom boundary conditions (z=-H):

$$\vec{u}.\nabla(-H) = w$$

Kinematic

$$K_{Mv}\frac{\partial u}{\partial z} = \frac{-C_d|\vec{u}|u}{\rho_0}$$

$$K_{Mv}\frac{\partial v}{\partial z} = \frac{-C_d|\vec{u}|v}{\rho_0}$$

Bottom friction

$$K_{Tv}\frac{\partial T}{\partial z} = 0$$

$$K_{Sv}\frac{\partial S}{\partial z} = 0$$

Bottom-flux

Unknowns:

- Prognostic variables: u,v,T,S (+η)
- Diagnostic variables: w, P, ρ

- Parameters: K_{Mh}, K_{Mv}, K_{Th}, K_{Tv}, K_{Sh}, K_{Sv}

ROMS ocean model

ROMS: Regional Ocean Model System

3 version of ROMS:
•Rutger University, USA : http://www.myroms.org
•UCLA University : http://ww.atmos.ucla.edu.cesr.ROMS_page.html
•IRD/INRIA version called 'ROMS_AGRIF' version : http://roms.mpl.ird.fr

The practical clases will focus on this last **ROMS_AGRIF version :** its main specialty is the online nesting capability using AGRIF *(Penven et al, Ocean Modelling, 2006 ; Debreu et al, Ocean Modelling, 2011)*

To perform of ROMS simulation, you need :

- Horizontal grid
- Bottom topography
- Land mask
- Surface forcing
- Initial Conditions
- Lateral boundary conditions

} Data computed using the ROMSTOOLS

Summary of ROMS characteristics

- Solve the primitive equation : Boussinesq approximation + hydrostatic vertical momentum balance
- Coastline an terrain following curvilinear coordinate
- Split explicit, free surface ocean model : short time steps for barotropic dynamic (ssh and 2D momentum) and a much larger time steps for baroclinic dynamic (T,S, 3D momentum)

- Parallelization by two-dimensional subdomain partitioning
- Shared and distributed parallelization (OpenMP and MPI)

- High advection scheme
- Rotated tensors to reduce diapycnal mixing
- Improved calculation of horizontal pressure gradient
- KPP turbulent closure model (surface and bottom KPP)

- Open boundary : Adaptative mixed radiations/nudging open boundary conditions (Marchesiello et al, 2001).
- Nesting capability : AGRIF (Adaptative Grid Refinement in Fortran) library .

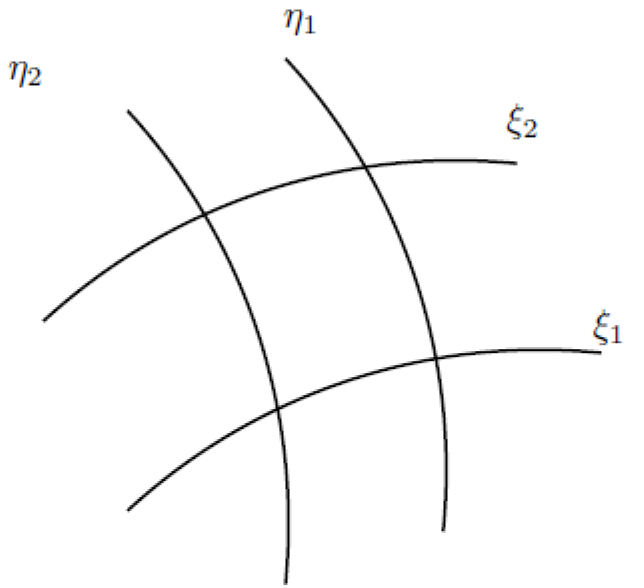
- Sediment module
- Several BGC model
- Float tracking module

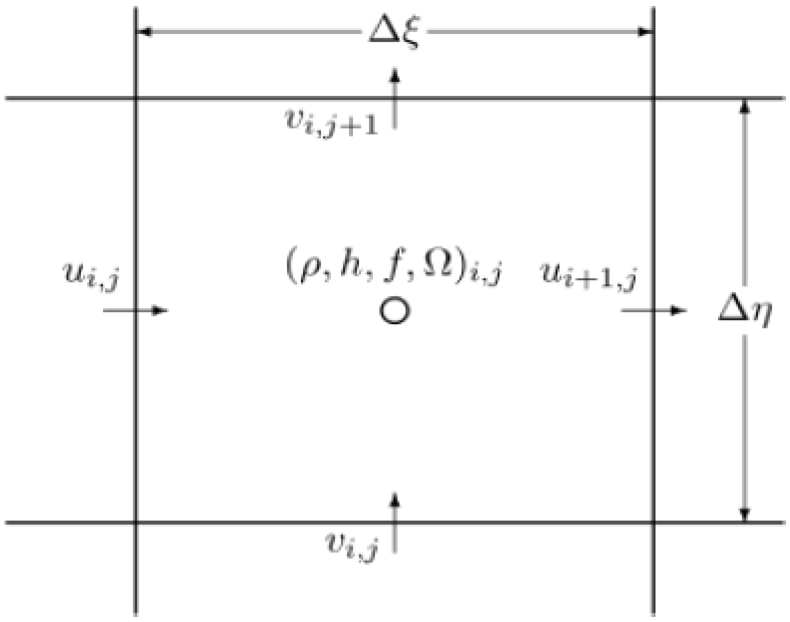
ROMS horizontal grid

- Discretized in coastline-and **terrain-following curvilinear coordinate**
- **Arakawa C-grid**

$$(ds)_\xi = \left(\frac{1}{m}\right) d\xi$$

$$(ds)_\eta = \left(\frac{1}{n}\right) d\eta$$





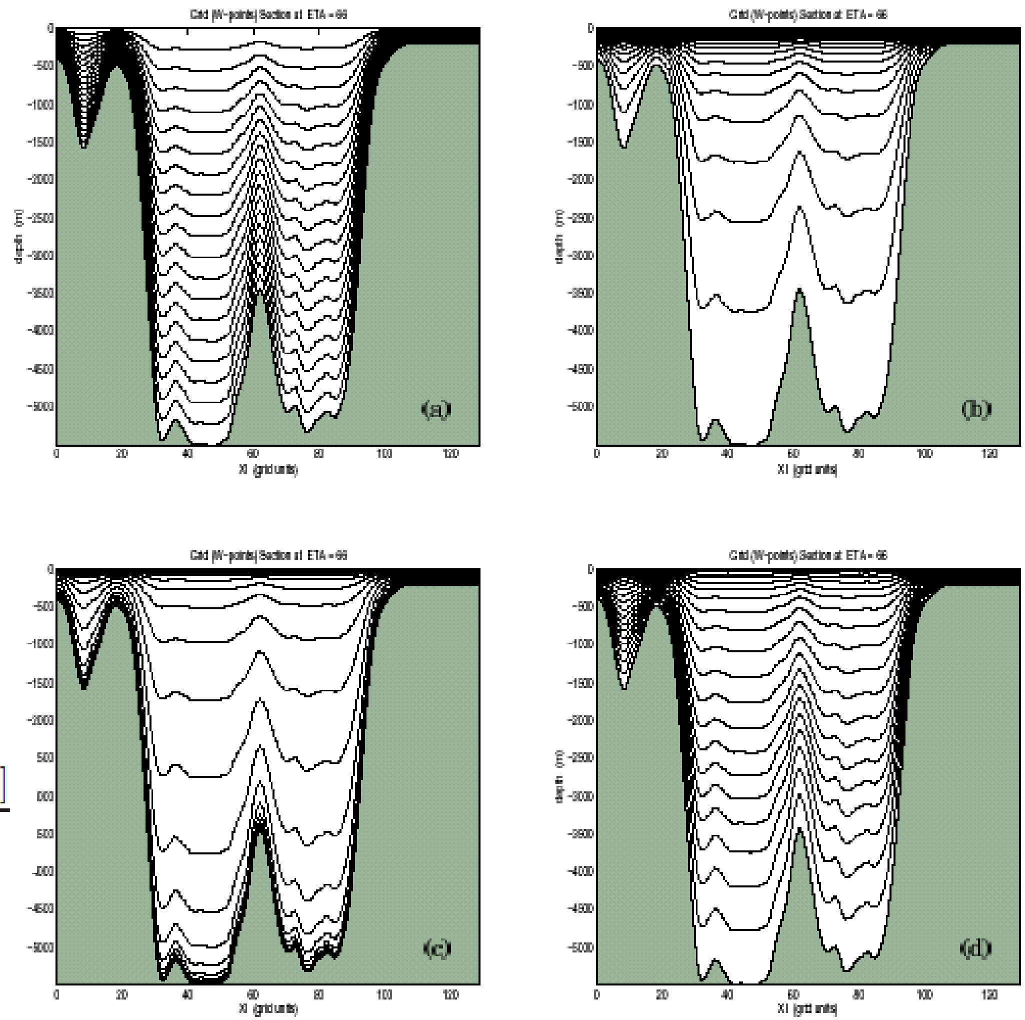
(a) Grille C d'Arakawa

ROMS verical grid : σ generalized coordinate

GENERALIZED σ-COORDINATE

Stretching & condensing of vertical resolution

- (a) Ts=0, Tb=0
- (b) Ts=8, Tb=0
- (c) Ts=8, Tb=1
- (d) Ts=5, Tb=0.4



$$z = \zeta(1+s) + h_c s + (h - h_c)C(s)$$

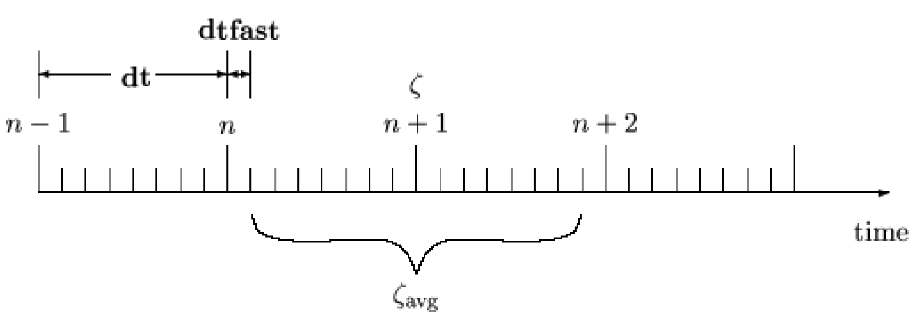
$$C(s) = (1-b)\frac{\sinh[\theta s]}{\sinh\theta} + b\frac{\tanh[\theta(s+\frac{1}{2})] - \tanh[\frac{1}{2}\theta]}{2\tanh[\frac{1}{2}\theta]}$$

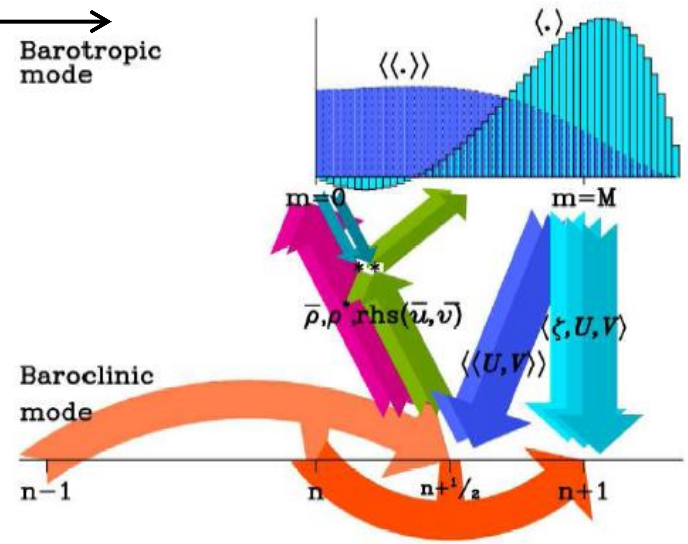
- !! Take care of the Pressure Gradient Error on steep slopes !!

Time splitting and free surface

ROMS resolve free surface using the time-splitting method :

- ❑ Direct integration of the barotropic equations
- ❑ **Getting the free surface is straight forward**
- ❑ Good parallelization performances
- ❑ BUT difficulties to separated fast and slow modes : possible instabilties. To avoid that
  - time averaging over the barotropic sub-cycle
  - finer mode coupling



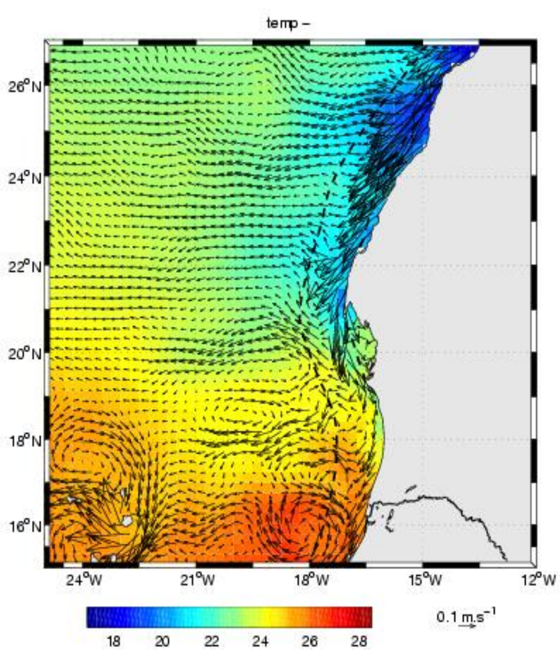


ROMS_AGRIF use 3rd order predictor (Leap frog)/corrector (Adam-Moulton) time stepping

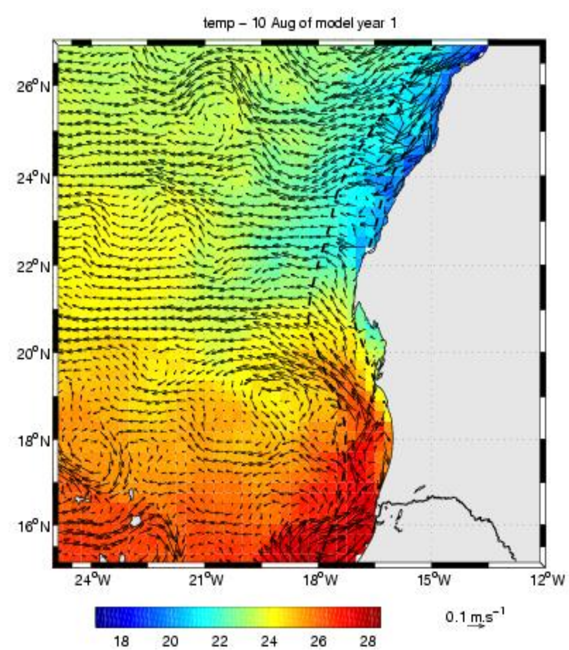
Advection schemes

- By default, 3rd order, upstream-biased advection scheme : allows the generation of steep gradient, with a weak dispersion and weak diffusion.
- No need to impose explicit diffusion/ viscosity to avoid numerical noise (in case of 3D modeling)
- Effective resolution is improved :

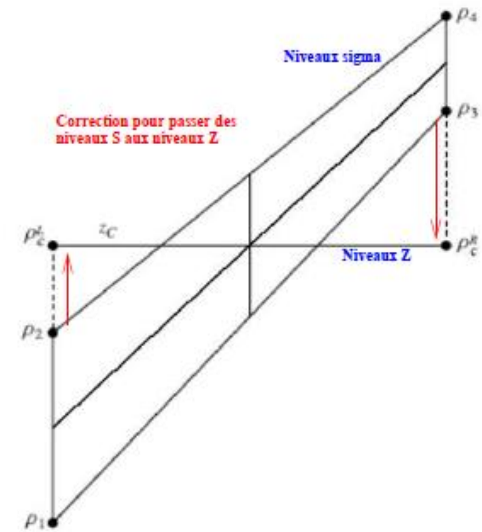
OPA - 0.25 deg



ROMS – 0.25 deg



Pressure Gradient force



- Truncation errors are made from calculating the baroclinic pressure gradients across sharp topographic changes such as the continental slope

- Difference between 2 large terms

$$-\frac{1}{\rho_0}\frac{\partial P}{\partial x}\Big|_z = -\frac{1}{\rho_0}\frac{\partial P}{\partial x}\Big|_s + \frac{1}{\rho_0}\cdot\frac{\partial P}{\partial z}\cdot\frac{\partial z}{\partial x}\Big|_s,$$

$$\epsilon \equiv \frac{\left|\frac{\partial P}{\partial x}\Big|_s - \frac{\partial P}{\partial z}\cdot\frac{\partial z}{\partial x}\Big|_s\right|}{\left|\frac{\partial P}{\partial x}\Big|_s\right| + \left|\frac{\partial P}{\partial z}\cdot\frac{\partial z}{\partial x}\Big|_s\right|} \ll 1,$$

- Errors can appear in the unforced flat stratification experiment

Reducing PGF Truncation Errors

- Smoothing the topography using a nonlinear filter and a criterium: → ***r = Δh / h < 0.2***
- Using a density formulation →

$$-\frac{1}{\rho_0}\frac{\partial P}{\partial x}\bigg|_z = -\frac{1}{\rho_0}\frac{\partial P}{\partial x}\bigg|_{z=\zeta} - \frac{g}{\rho_0}\int_z^{\zeta}\frac{\partial \rho}{\partial x}\bigg|_z dz'$$

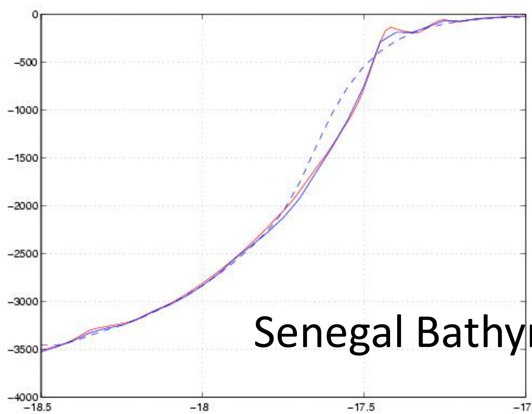
$$= -\frac{g\rho(\zeta)}{\rho_0}\frac{\partial \zeta}{\partial x} - \frac{g}{\rho_0}\int_z^{\zeta}\left[\frac{\partial \rho}{\partial x}\bigg|_s - \frac{\partial \rho}{\partial z'}\frac{\partial z'}{\partial x}\bigg|_s\right]dz',$$

- Using high order schemes to reduce the truncation error (4th order, McCalpin, 1994)
- Gary, 1973: substracting a reference horizontal averaged value from density ($\rho' = \rho - \rho_a$) before computing pressure gradient
- Rewritting Equation of State: reduce passive compressibility effects on pressure gradient

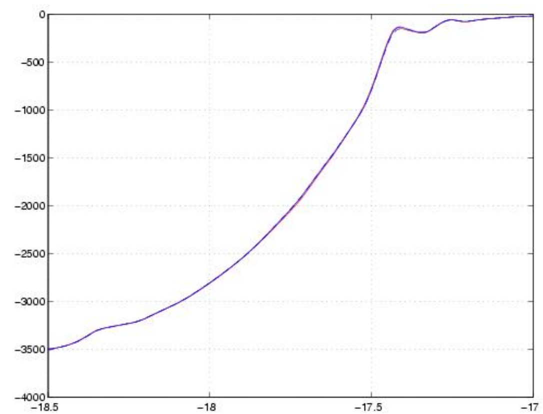
Smoothing methods

- ***r = Δh / h*** is the slope of the logarithm of *h*
- One method (ROMS) consists of smoothing *ln(h)* until $r < r_{max}$

Res: 5 km
r < 0.25

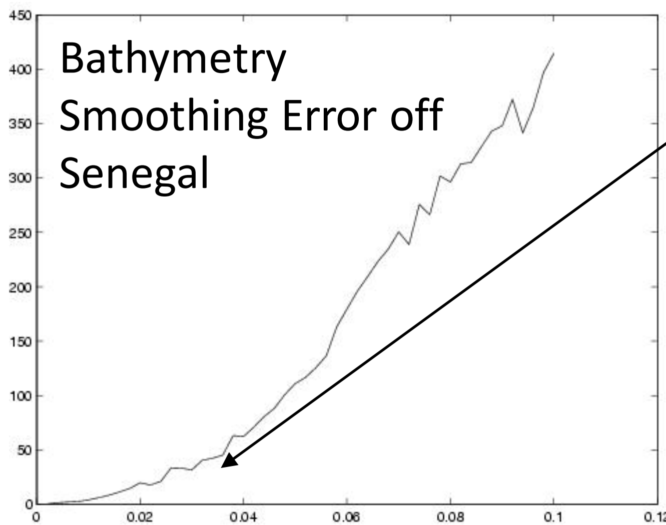


Res: 1 km
r < 0.25



Senegal Bathymetry Profil

Standard Deviation [m]



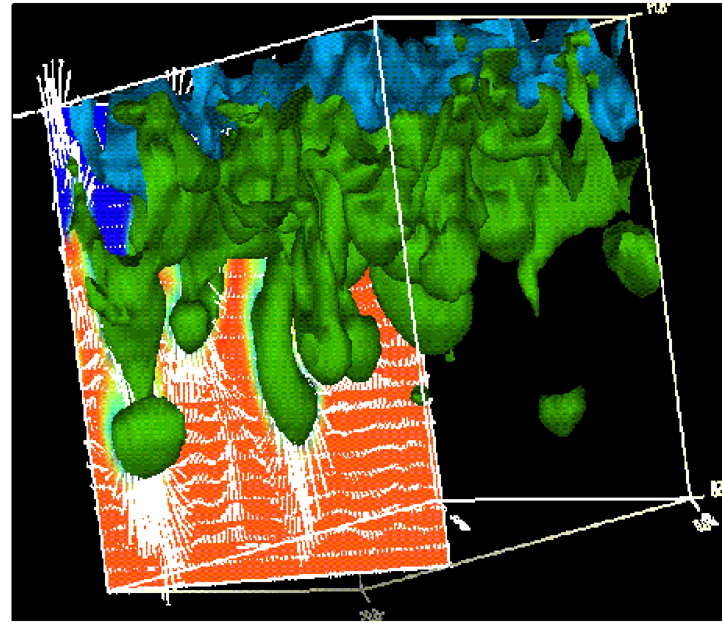
Bathymetry Smoothing Error off Senegal

Convergence at ~ *4 km* resolution

Grid Resolution [deg]

Boundary Layer Parameterization

- Boundary layers are characterized by strong turbulent mixing



- Turbulent Mixing depends on:
  - Surface/bottom forcing:
    - Wind / bottom-shear stress stirring
    - Stable/unstable buoyancy forcing
  - Interior conditions:
    - Current shear instability
    - Stratification

Reynolds term:

$$\frac{\partial}{\partial s}\,\overline{w'T'}$$

- We look for to determine the vertical profile of **K mixing parameter**

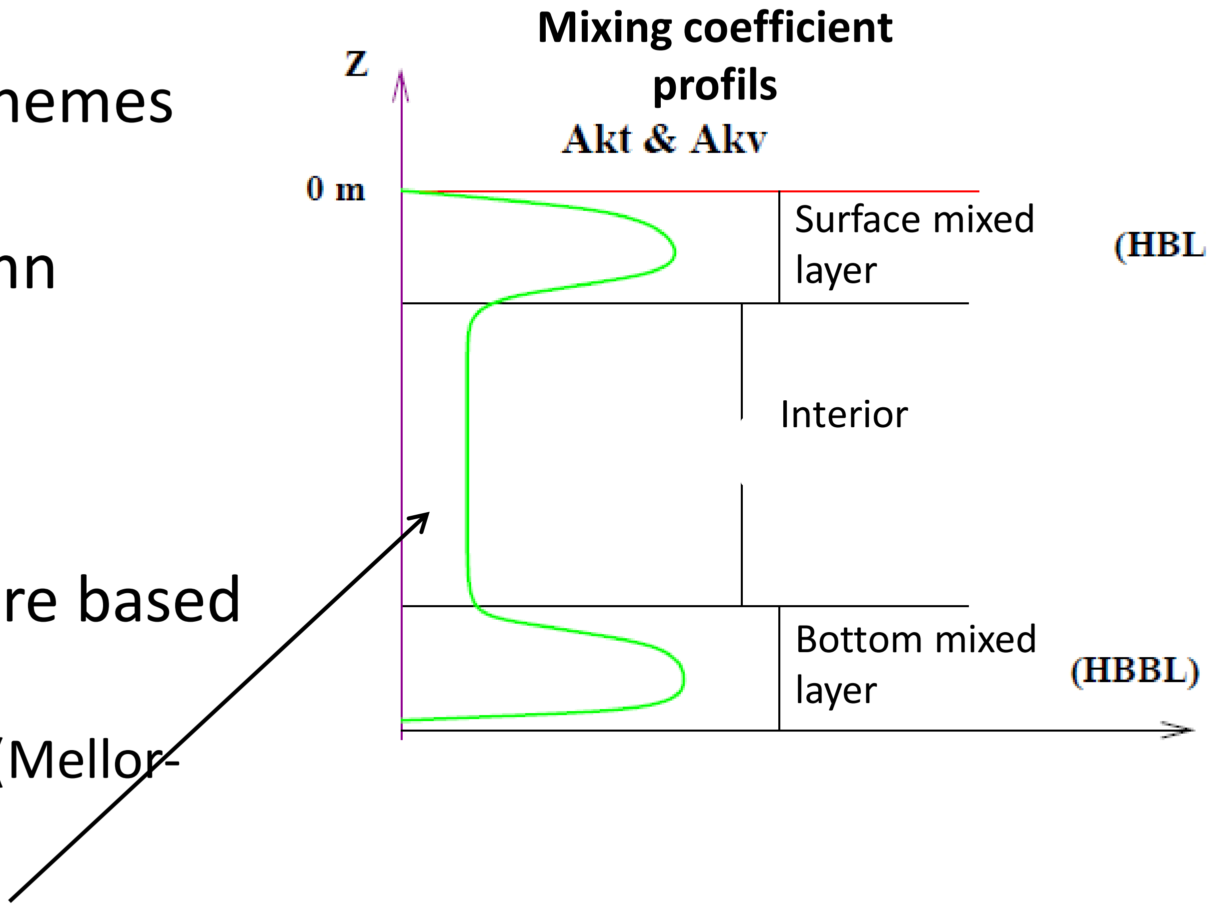
K theory

$$\frac{\partial}{\partial s}\left(\frac{\kappa}{H_z m n}\frac{\partial \phi}{\partial s}\right)$$

Boundary layer parametrization: KPP

- ❑All mixed layer schemes are based on one-dimentional « column physics »

- ❑Boundary layer parameterizations are based either on:
  - •Turbulent closure (Mellor-Yamada, TKE)
  - **• K profile (KPP)**



Principle scheme of KPP turbulent closure

Surface and Bottom forcing

$$\frac{\partial}{\partial s}\left(\frac{\kappa}{H_z mn}\frac{\partial \phi}{\partial s}\right)$$

The vertical boundary conditions can be prescribed as follows:

top $(z = \zeta(x,y,t))$

$$\nu \frac{\partial u}{\partial z} = \tau_s^x(x,y,t)$$

$$\nu \frac{\partial v}{\partial z} = \tau_s^y(x,y,t)$$

Wind stress

$$\kappa_T \frac{\partial T}{\partial z} = \frac{Q_T}{\rho_o c_P} + \frac{1}{\rho_o c_P}\frac{dQ_T}{dT}(T - T_{\text{ref}})$$

Heat flux

$$\kappa_S \frac{\partial S}{\partial z} = \frac{(E-P)S}{\rho_o}$$

Salt flux

$$w = \frac{\partial \zeta}{\partial t}$$

and bottom $(z = -h(x,y))$

$$\nu \frac{\partial u}{\partial z} = \tau_b^x(x,y,t)$$

$$\nu \frac{\partial v}{\partial z} = \tau_b^y(x,y,t)$$

$$\kappa_T \frac{\partial T}{\partial z} = 0$$

$$\kappa_S \frac{\partial S}{\partial z} = 0$$

$$-w + \vec{v}\cdot\nabla h = 0.$$

Bottom stress

$$\tau_b^x = (\gamma_1 + \gamma_2\sqrt{u^2+v^2})u$$

$$\tau_b^y = (\gamma_1 + \gamma_2\sqrt{u^2+v^2})v$$

Drag Coefficient C_D:

γ_1=3.10-4 m/s Linear

γ_2=2.5 10-3 Quadratic

Bottom friction parametrization

✓Linear friction, with r friction velocities [m/s] →

$$(\tau_b^x, \tau_b^y) = -r\ (u_b, v_b)$$

✓Quadratic friction, controled by a constant drag coefficient **Cd** →

$$(\tau_b^x, \tau_b^y) = C_d \sqrt{u_b^2 + v_b^2}\ (u_b, v_b)$$

✓Quadratic friction coefficient, using variable **Cd** (Von Karman log layer) →

$$(\tau_b^x, \tau_b^y) = C_d \sqrt{u_b^2 + v_b^2}\ (u_b, v_b)$$

$$C_d = \left(\frac{\kappa}{\log[\Delta z_b / z_r]}\right)^2 \text{ si } C_d^{min} < C_d < C_d^{max}$$

$$C_d = C_d^{min} \text{ ou } C_d^{max}$$

$$\kappa = 0.41$$

z_r = Rugosity scale

Δz_b = thickness of the first bottom level

Air Sea Interaction

Heat fluxes & Freshwater fluxes

- Directly read the forcing files
- Use of a bulk formulae :
  - Heat flux : compute total heat flux from latent, sensible, solar and longwave fluxes and model SST
  - Freshwater flux : compute from evap, prate and model SSS

- **Wind stress:**
  - Directly read the forcing files
  - Compute the windstress from the Cd drag coefficient, model SST and wind stress

bulk_flux.F

Open boundary conditions I (OBC type)

Adaptative mixed radiations/nudging open boundary conditions (Marchesiello et al, 2001).

$$\frac{\partial \phi}{\partial t} + c_x \frac{\partial \phi}{\partial x} + c_y \frac{\partial \phi}{\partial y} = -\frac{1}{\tau}(\phi - \phi_{ext})$$

Radiation, (Orlanski, 1982)

▪Possibility to **use "Flather" OBC conditions for barotropic mode :**

Specially designed for tidal applications

Adaptative nudging term :

Adaptitivity

▪Ingoing signal (Cx >0) : strong nudging toward external data using $\tau = \tau_{in}^{-1}$

▪Outgoing signal (Cx >0) : weak nudging toward ext. Data $\tau = \tau_{out}^{-1}$

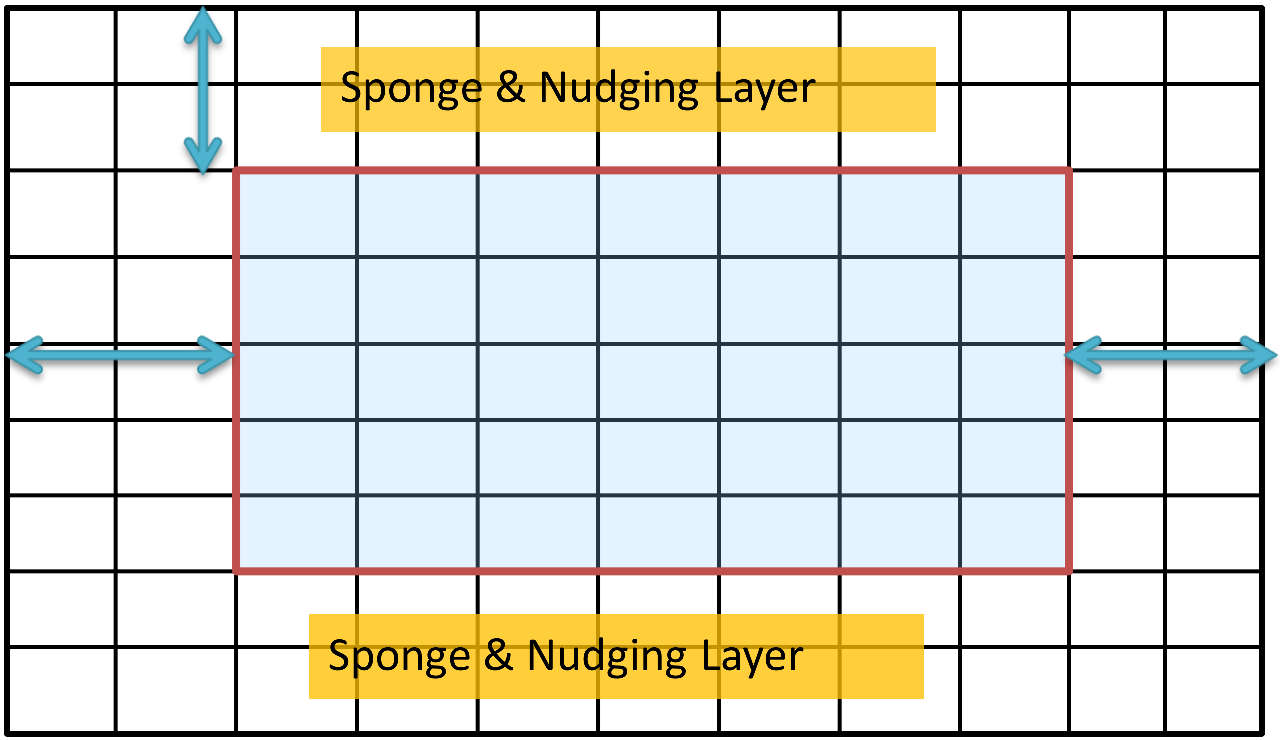
$\tau_{out} \approx 180$ days

$\tau_{in} \approx 1$ days

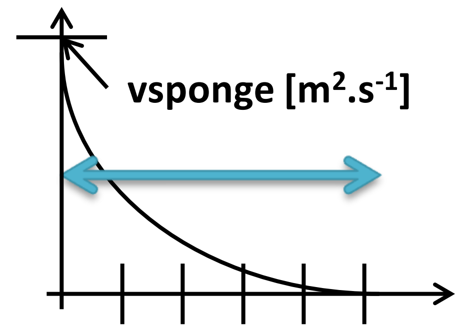
τ_{M_in} , τ_{M_out} : momentum

τ_{T_in} , τ_{T_out} : tracer

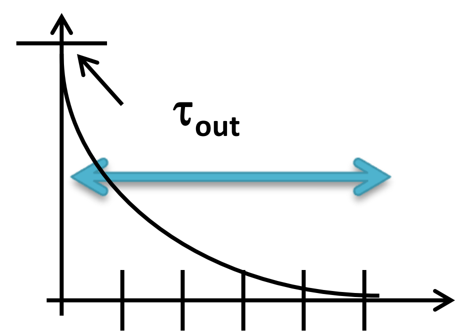
Sponge/Nudging Layer



$K^T h$, $K^m h$ profil across sponge layer



τ_{out} profil cross nudging layer



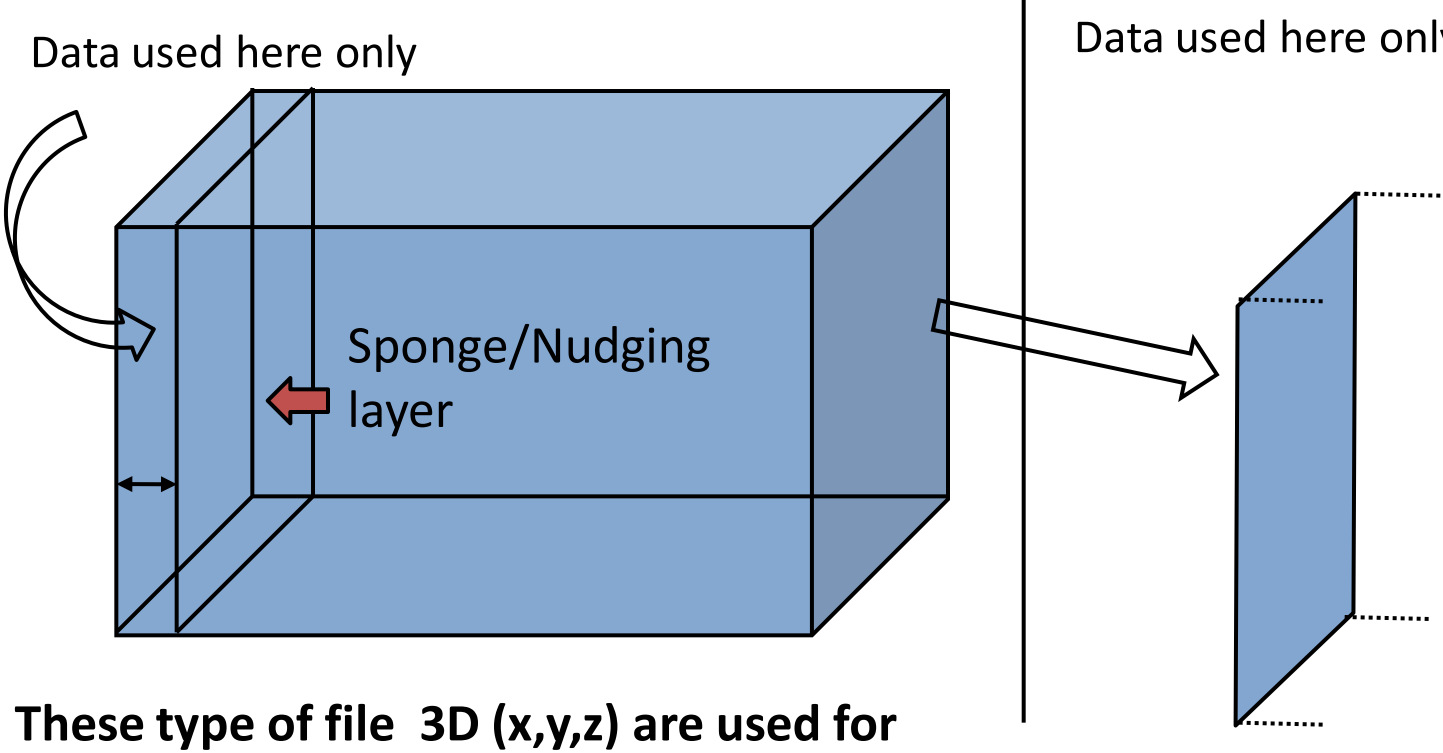
- Sponge : Additional viscosity/diffusivity
- Nudging : Add a weak nudging, $\tau = 0 \rightarrow \tau_{out}^{-1}$ toward climatology, if available (see after)

Open boundary forcing (Clim or Bry)

Different ways to impose OBC

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

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



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

Tides

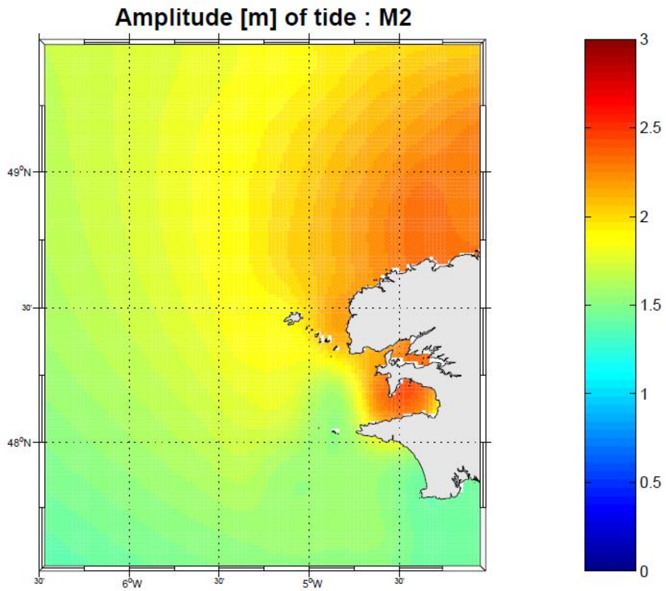
•The tides are imposed at open boundaries, using the Flather OBC.
(**OBC_M2FLATHER** cppkeys)

- ζ_{tides}, $\bar{u}_{tides}$ et $\bar{v}_{tides}$ h and depth averaged zonal and meridian currents) are added at the open boundaries ζ_{clim}, $\bar{u}_{clim}$ et $\bar{v}_{clim}$
- ζ_{tides}, $\bar{u}_{tides}$ et $\bar{v}_{tides}$ computed from the tidal harmonics given by some tidal model, in our case TPXO7 (0.25° resolution, 10 tidal components : M2, N2, S2, K2, K1, O1, P1,Q1,Ln, Mm
- The global tidal model gives harmonics constants for all the principal tidal waves. These constants permits to compute at every time t, ζ^N_{tides} :(t), $\bar{u}^N_{tides}$(t), $\bar{v}^N_{tides}$ (t) of the tidal wave component N.

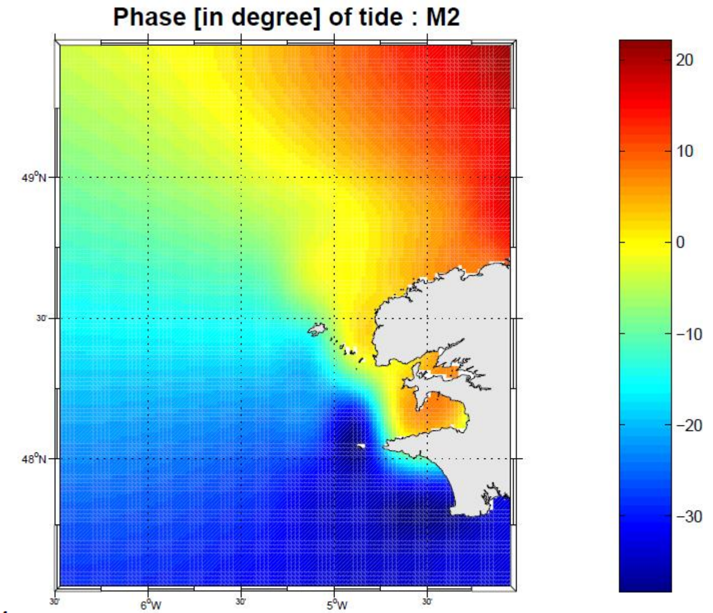
•You need :
  - Choose the number of tidal wave component you want.
  - Interpole on the grid the different harmonic constants

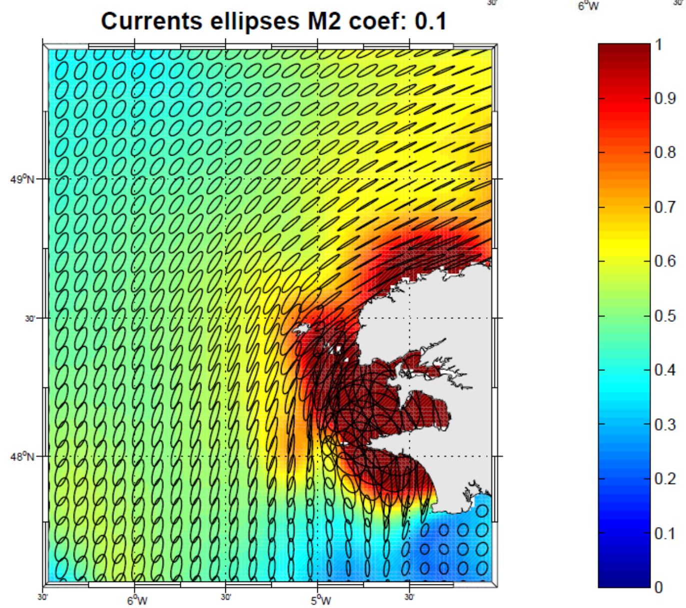
•For the moment, there is no generator potential (for the moment)

Tides


Amplitude [m] of tide : M2
49°N
30'
48°N
30'
6°W
30'
5°W
30'
3
2.5
2
1.5
1
0.5
0

M2


Phase [in degree] of tide : M2
49°N
30'
48°N
30'
6°W
30'
5°W
30'
20
10
0
−10
−20
−30


Currents ellipses M2 coef: 0.1
49°N
30'
48°N
30'
6°W
30'
5°W
30'
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

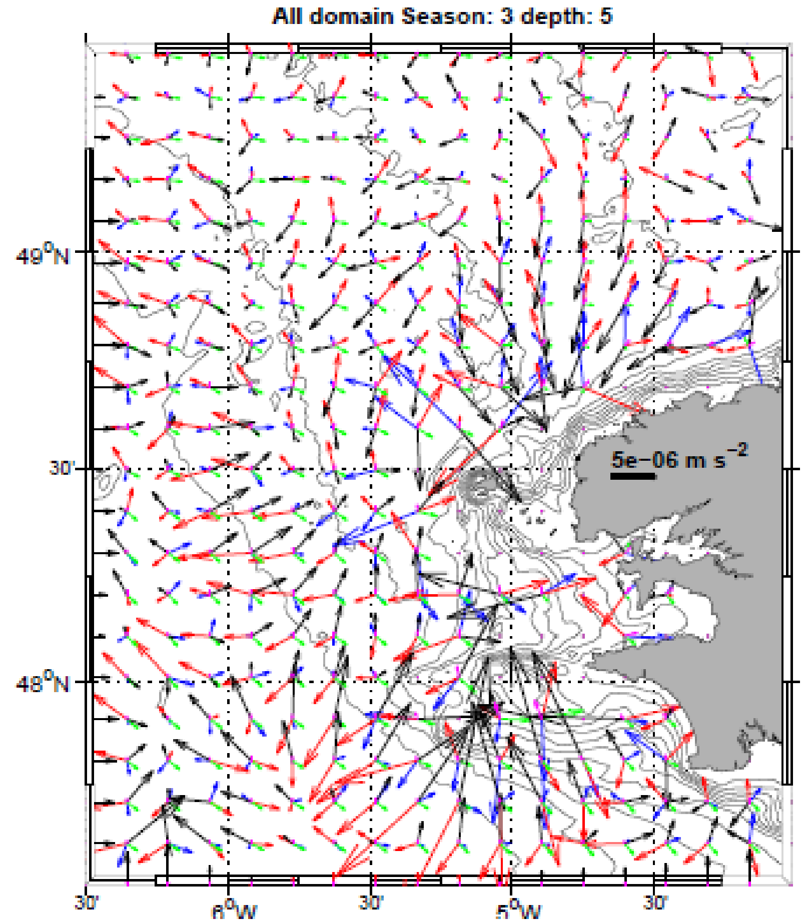
Online Diagnostics : Momentum equations terms

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

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

- 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 : $\partial_t \vec{u}$



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

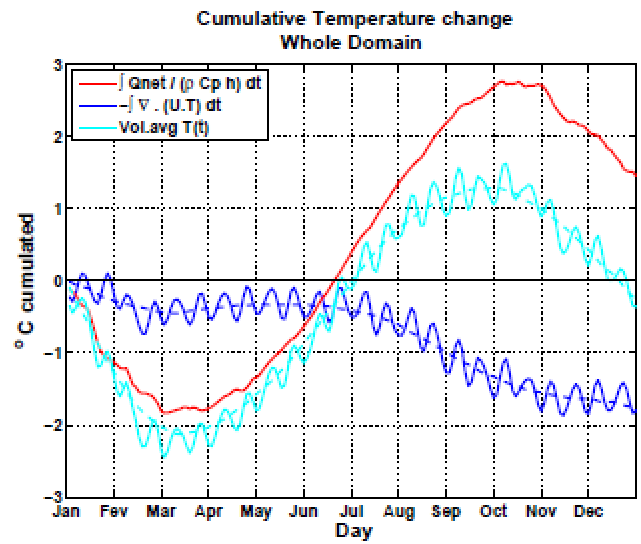
Storage:

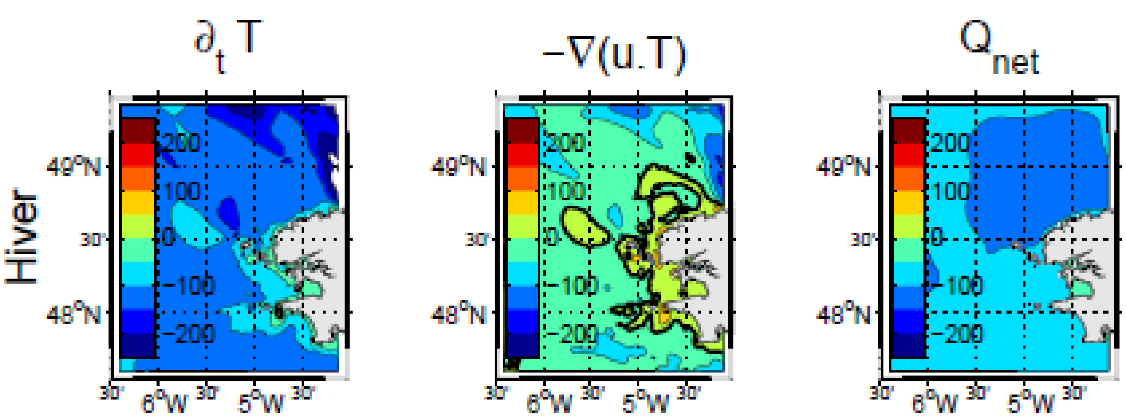
- roms_diaM.nc and roms_diaM_avg.nc files.
- Can choose in roms.in the writing frequency and the terms to store

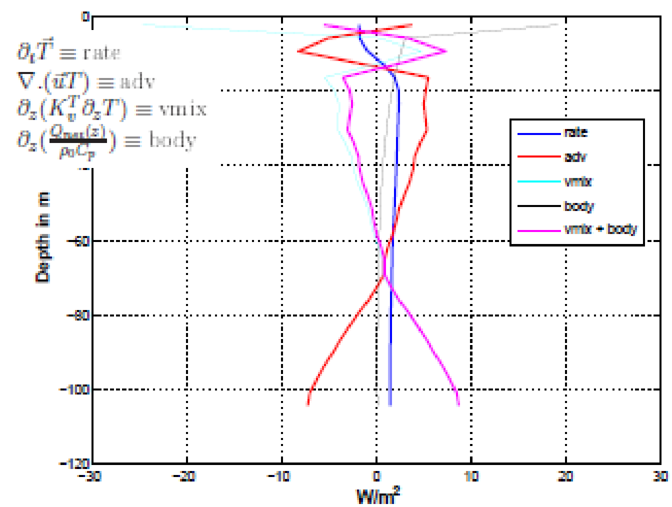
Online Diagnostics : Tracer equations terms

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

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

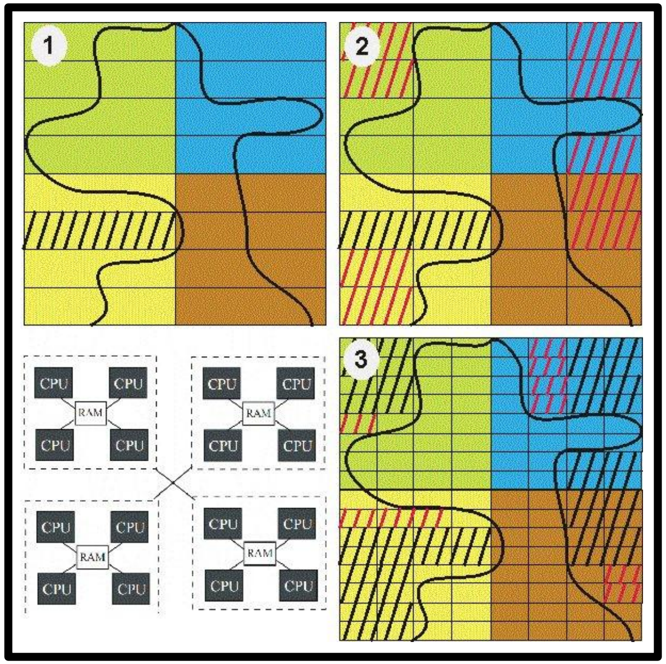






Parallelization

- ❑Parallelization by 2 dimensional subdomain partition.
- ❑Multiple subdomain can be assigned to each processor.
- ❑Parallelization operational on shared and distributed memory computer
- ❑Good scaling performance



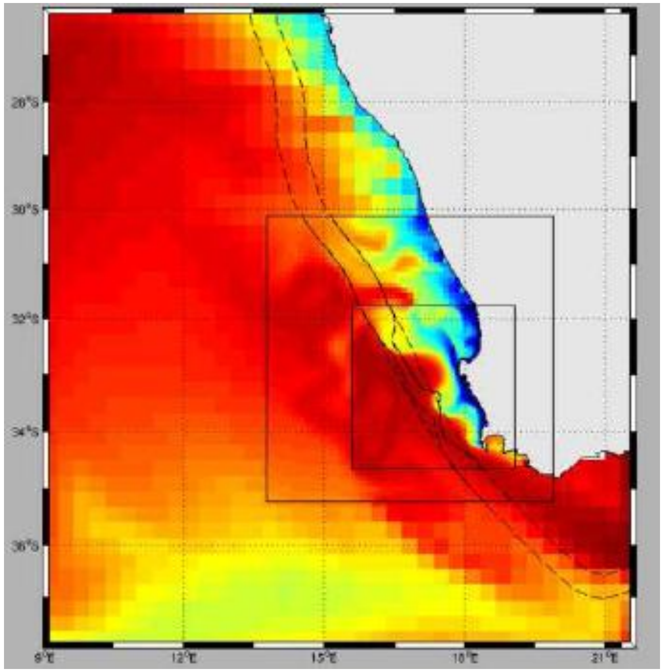
ROMS CPPKEYS :

- #define OPENMP
- #define MPI

Ported successfully on different computing center :

Earth simulator, Idris, Caparmor, Calmip, regional cluster, …

AGRIF Nesting

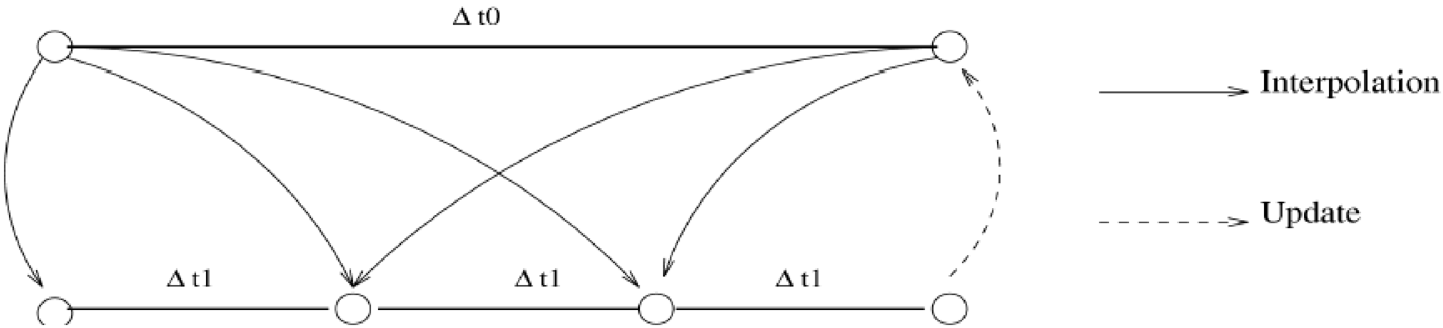


Nesting capability of ROMS: ROMS_AGRIF

- Manage arbitrary number of fixed grid and embedded levels
- AGRIF : Adaptative Mesh Refinement (http://www ljk.imag.fr/MOISE/AGRIF/)

- 1-way and 2 way nesting capability:
  - ✓1 way coarse grid feed fine grid
  - ✓ 2 way nesting : feed back of the fine grid on the coarse grid

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



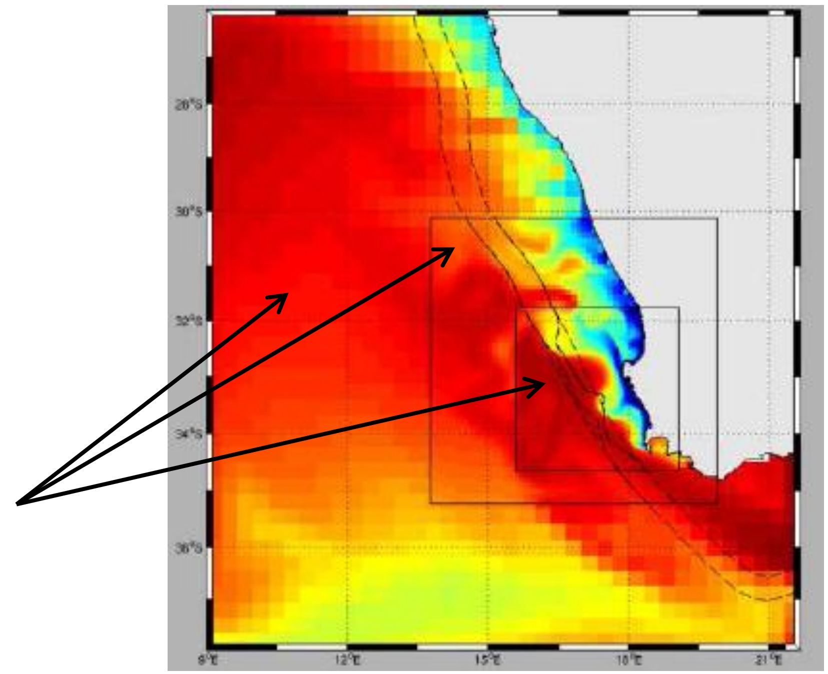
AGRIF Nesting

The file Agrif_FixedGrids

```
 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!]
```



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

Needs to run an embedded model :

Surface forcing and initial conditions datas files.

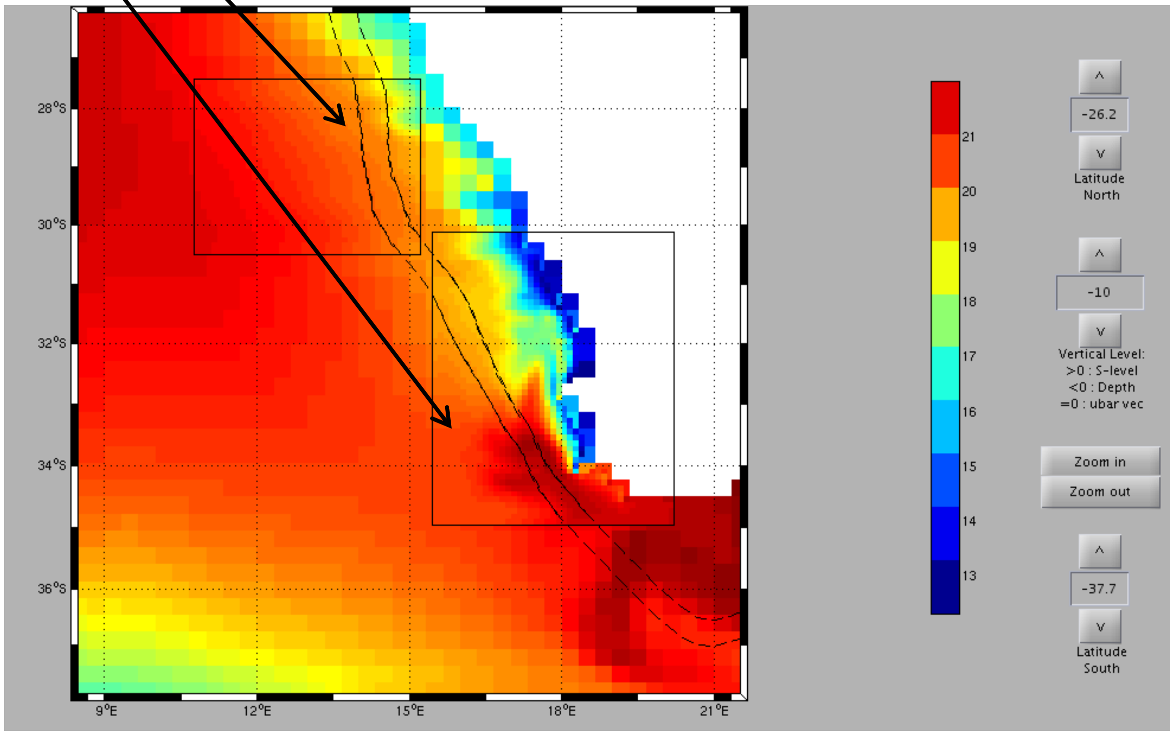
For grid #xx :

- roms_grd.nc.xx
- roms_frc.nc.xx
- roms_blk.nc.xx
- roms.ini.nc.xx
- roms.in.xx

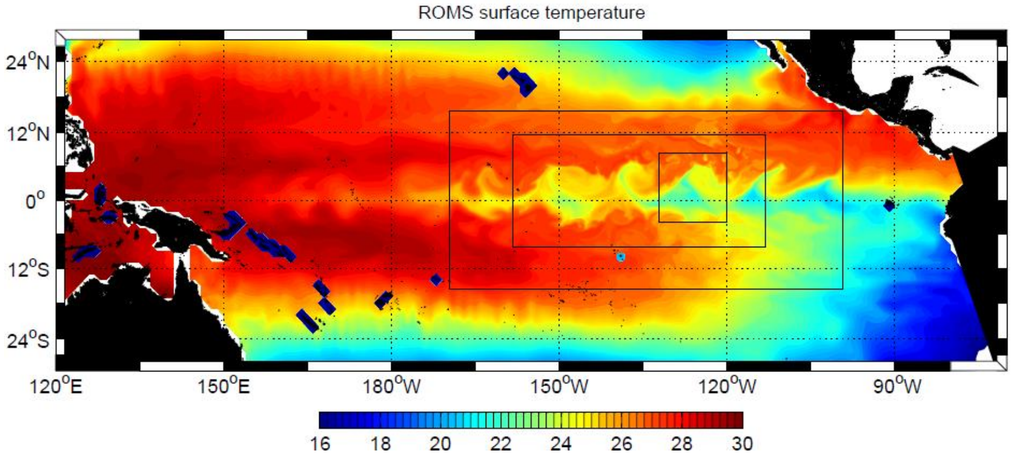
AGRIF Nesting

```
  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

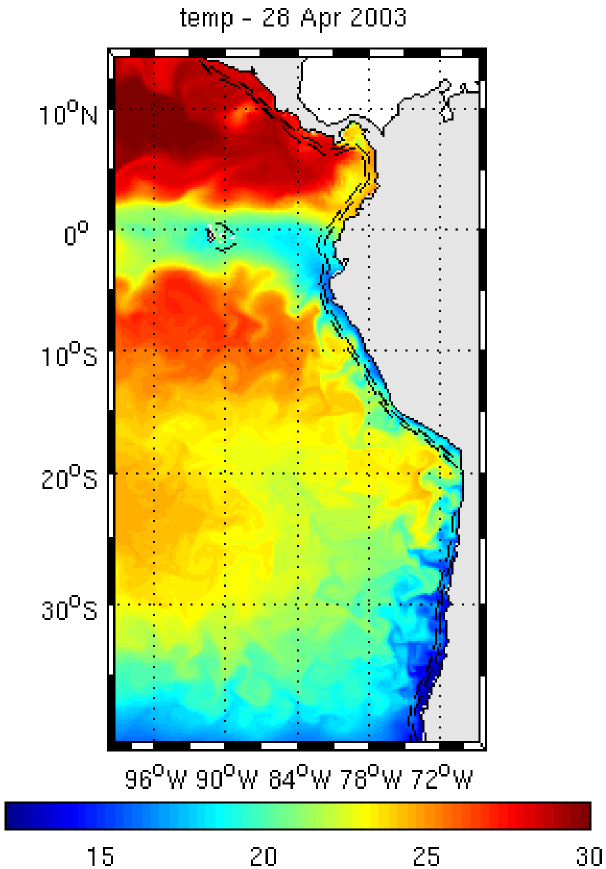


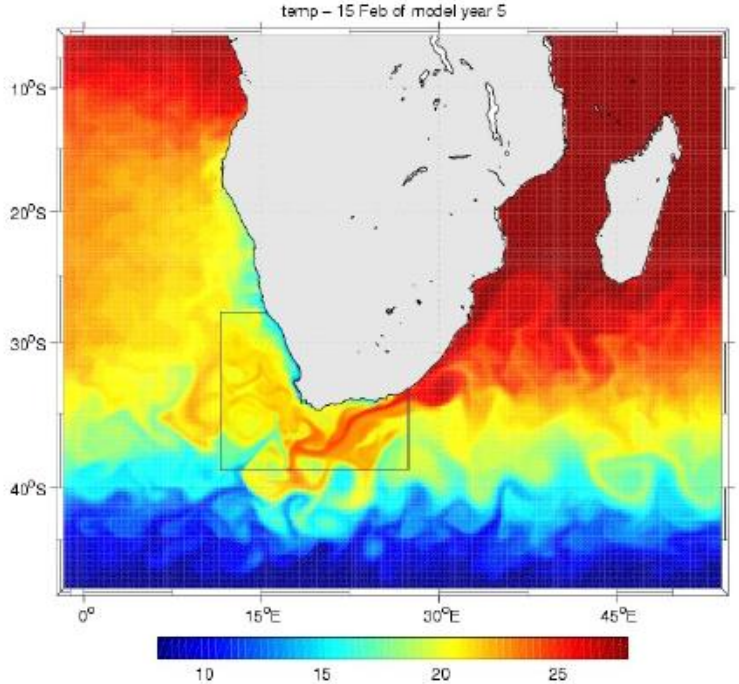
Many realistic applications



4-AGRIF nested grid, SST. *Marchesiello et al, Ocean modelling, 2011.*

Peru-Chile configuration, *Cambon et al*

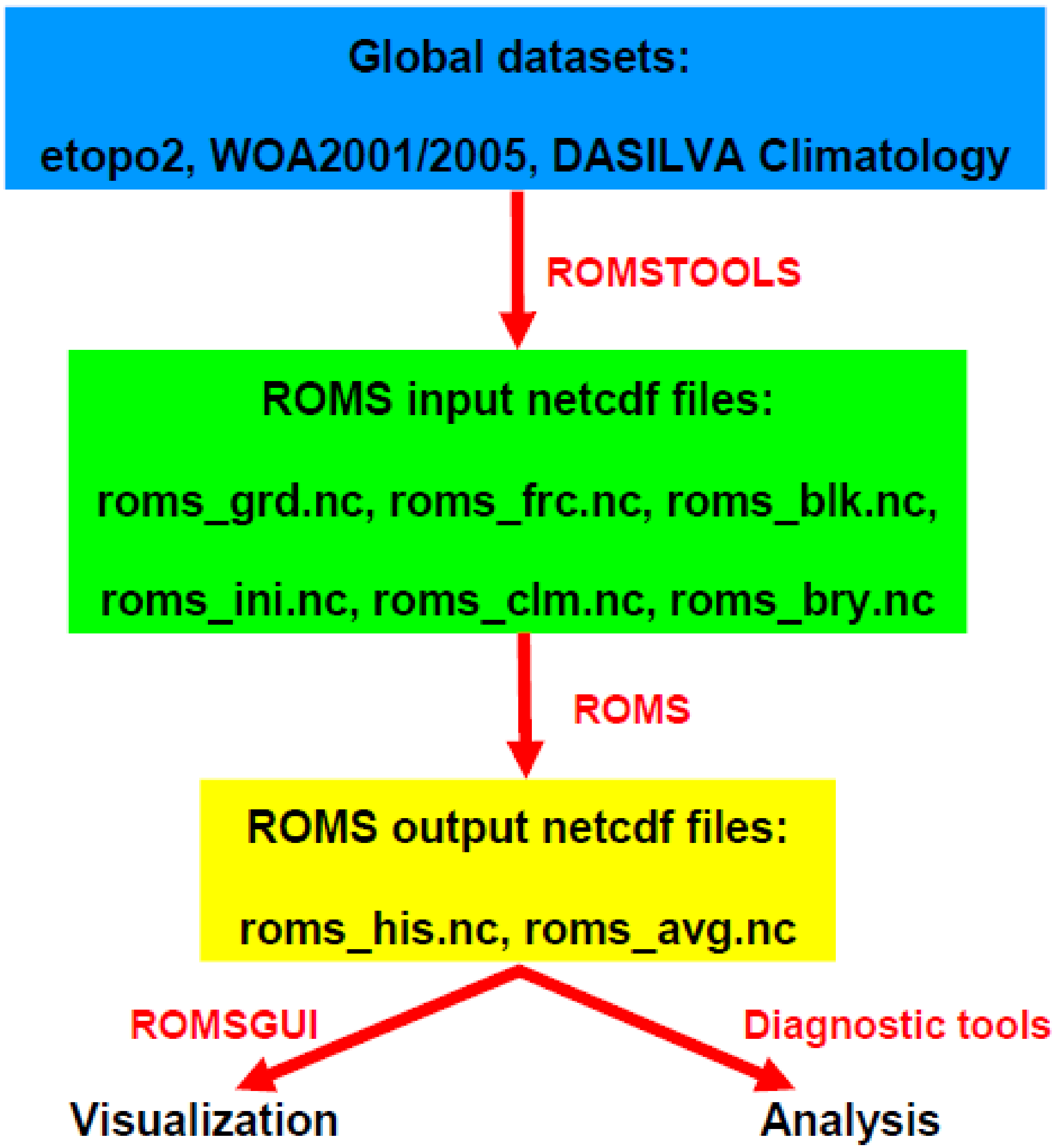




'SAFE' Configuration, *Penven et al*

Others ….

Strategy to build a configuration



ROMS_AGRIF & ROMSTOOLS modelling system

- **ROMS** : download and untar the file Roms_Agrif_v2.1.
- It provides the ROMS_AGRIF fortran code
- **ROMSTOOLS** : download and untar the file ROMSTOOLS_v2.1.tar.gz files.
- It provides the matlab routines to pre-process the input file and to visualize them
- **Utilities_ROMSTOOLS** : download and untar the file Utilities ROMSTOOLS.tar.gz files.
- It provides the netcdf librairy, mapping toolbox etc …
- **Datasets** : It provides the global datasets needed by the ROMSTOOLS

Directory tree of the system

Work is done in Roms_tools/Run dir

```
Roms_tools
  |- Aforc_NCEP
  |- Aforc_QuikSCAT
  |- air_sea
  |- COADS05
  |- Compile
  |- Diagnostic_tools
  |- Documentation
  |    |- User_guide
  |- Forecast_tools
  |- mask
  |- mex60
  |- mexnc
  |    |- tests
  |- m_map
  |    |- private
  |- Nesting_tools
  |- netcdf_g77
  |- netcdf_ifc
  |- netcdf_matlab
  |    |- listpick
  |    |- ncatt
  |    |- ncbrowser
  |    |- ncdim
  |    |- ncfiles
  |    |- ncitem
  |    |- ncrec
  |    |- nctype
  |    |- ncutility
  |    |- ncvar
  |    |- netcdf
  |- netcdf_x86_64
  |- Oforc_OGCM
  |- Opendap_tools
  |    |- FEDORA
  |    |- FEDORA_X64
  |- Preprocessing_tools
  |- Roms_Agrif
  |    |- AGRIFZOOM
  |    |    |- AGRIF_FILES
  |    |    |- AGRIF_INC
  |    |    |- AGRIF_OBJS
  |    |    |- AGRIF_YOURFILES
  |    |    |- LIB.clean
  |- Run
  |    |- DATA
  |    |- FORECAST
  |    |- ROMS_FILES
  |    |- SCRATCH
  |    |- TEST_CASES
  |- SeaWifs
  |- SST_pathfinder
  |- Tides
  |- Topo
  |    |- Matlab
  |- TPX06
  |- TPX07
  |- Visualization_tools
  |- WOA2001
  |- WOA2005
```

Steps to prepare and run a regional model

Presentation of the main step to run a model of the Southern Benguela at low resolution test case (1/3°) *with climatological forcings.*

1. Pre-processing data : for that use of ROMSTOOLS matlab toolbox
   - Grid and mask creation : **make_grid.m**
   - Atmospheric forcing : **make_forcing.m** (**make_bulk.m**)
   - Initialization and oceanic forcing at OBC : **make_clim.m** (**make_bry.m**)
2. Preparing and compiling the model : Compilation using jobcomp script
   - → executable **roms** is compiled
3. Running the model : **./roms roms.in**
4. Vizualize the results : for that use of matlab and the ROMSTOOLS
   - Vizualization gui : **roms_gui.m**

1) Pre-processing data

First : edit general parameters in **romstools_param.m** file

```
%%%%%%%%%%%%%%%%%%%%%%%%
%
%   Grid parameters
%   used by make_grid.m (and others..)
%
%%%%%%%%%%%%%%%%%%%%%%%%
%
% Grid dimensions:
%
lonmin =  12.3;  % Minimum longitude [degree east]
lonmax = 20.45;   % Maximum longitude [degree east]
latmin = -35.5;   % Minimum latitude  [degree north]
latmax = -26.5;   % Maximum latitude  [degree north]
%
% Grid resolution [degree]
%
dl = 13;
%
```

Horizontal grid parameters

```
% Number of vertical Levels (! should be the same in
param.h !)
%
N = 32;
%
%Vertical grid parameters (! should be the same in
roms.in !)
%
theta_s = 6.;
theta_b = 0.;
hc      =10.;
```

Vertical grid parameters

```
% Minimum depth at the shore [m] (depends on the resolution,
% rule of thumb: dl=1, hmin=300, dl=14, 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;
%
% Topography netcdf file name (ETOPO 2 or any other
netcdf file
% in the same format)
%
topofile = [ROMSTOOLS\_dir,'Topo/etopo2.nc'];
%
% Slope parameter (r=grad(h)/h) maximum value for
topography smoothing
```

$$r = \frac{\nabla h}{2h} = \frac{h_{+1/2} - h_{-1/2}}{h_{+1/2} + h_{-1/2}}$$

```
%
rtarget = 0.25;
%
% Number of pass of a selective filter to reduce the
isolated
% seamounts on the deep ocean.
%
n_filter_deep_topo=4;
%
% Number of pass of a single hanning filter at the end of
the
% smoothing procedure to ensure that there is no 2DX
noise in the
% topography.
%
```

1) Pre-processing data

- Go in the /CMS2011-TP/TP_ROMS/Roms_tools/Run
- Launch matlab –nodesktop
- >> start : *Add all the needed matlab path of the system*

- >> make_grid
  - Horizontal grid : position of the grid points, size of the grid cells
  - Bottom topography ⇨ /ROMS_FILES/roms_grd.nc
  - Land mask
- >> make_forcing :
  - Surface forcing : wind stress, surface heat flux, surface freshwater flux

    ⇨ ROMS_FILES/roms_frc.nc
- >> make_ini :
  - initial conditions : T, S, currents , SSH

    ⇨ ROMS_FILES/roms_ini.nc
- >> make_clim (resp. make_bry)
  - Lateral oceanic boundary conditions : T, S, currents , SSH

    ⇨ /ROMS_FILES/roms_clm.nc (resp. roms_bry.nc)

2) Preparing and compiling the model

Edit the param.h and cppdefs.h file to set-up the model

param.h defines the size of the arrays in ROMS:

```
...
#elif defined REGIONAL
#  if defined  BENGUELA
      parameter (LLm0=23, MMm0=31, N=32)  <----  Southern Benguela test Model
#  else
      parameter (LLm0=??,  MMm0=??,  N=??)
#  endif
...
```

Given by running make_grid (LLm0, MMm0)

Defined in romstools_param.m (N)

cppdefs.h:
- Basic options
- More advanced options

- 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.

2-Preparing and compiling the model

View cppdef.h file

```
!--------------------------------------------------------------
!                 BASIC OPTIONS
!--------------------------------------------------------------
*/
/*                        Configuration Name */
# define BENGUELA
/*                        Parallelization */
# undef  OPENMP
# undef  MPI
/*                        Embedding */
# undef AGRIF
/*                        Open Boundary Conditions */
# undef  TIDES
# define OBC_EAST
# undef OBC_WEST
# define OBC_NORTH
# define OBC_SOUTH
*/                         Embedding conditions */
# ifdef AGRIF
#  undef AGRIF_OBC_EAST
#  define AGRIF_OBC_WEST
#  define AGRIF_OBC_NORTH
#  define AGRIF_OBC_SOUTH
# endif
/*                       Applications */
# undef  BIOLOGY
# undef  FLOATS
# undef  STATIONS
# undef  PASSIVE_TRACER
# undef  SEDIMENTS
# undef  BBL
```

```
!--------------------------------------------------------------
!          MORE ADVANCED OPTIONS
!--------------------------------------------------------------
*/
/*                    Model dynamics */
# define SOLVE3D
# define UV_COR
# define UV_ADV
# ifdef TIDES
#  define SSH_TIDES
#  define UV_TIDES
#  define TIDERAMP
# endif
/*                    Grid configuration */
# define CURVGRID
# define SPHERICAL
# define MASKING
/*                         Input/Output & Diagnostics */
# define AVERAGES
# define AVERAGES_K
# define  DIAGNOSTICS_TS
# define  DIAGNOSTICS_UV
/*                         Equation of State  */ ...
/*                         Surface Forcing */ ...
/*                         Lateral Forcing  */ ...
/*                         Input/Output & Diagnostics */ ...
*                          Bottom Forcing  */ ...
/*                         Point Sources – Rivers */ ...
/*                         Lateral Mixing  */ ...
/*                         Vertical Mixing */ ...
/*                         Open Boundary Conditions  */ ...
/*                         Embedding conditions  */ ...
```

2) Preparing and compiling the model

For that use the the jobcomp tcsh file
./jobcomp

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 defintions (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 librairy (Netcdf, MPI, AGRIF) --
   - --> produce the executable roms

3) Running the model

The namelist roms.in

roms.in provides the run time parameters for ROMS:

```
title:
      Southern Benguela
time_stepping: NTIMES   dt[sec]  NDTFAST  NINFO
          480      5400     60      1
S-coord: THETA_S,   THETA_B,    Hc (m)
        6.0d0      0.0d0      10.0d0
grid:  filename
                   ROMS_FILES/roms_grd.nc
forcing: filename
                   ROMS_FILES/roms_frc.nc
bulk_forcing: filename
                   ROMS_FILES/roms_blk.nc
climatology: filename
                   ROMS_FILES/roms_clm.nc
boundary: filename
                   ROMS_FILES/roms_bry.nc
initial: NRREC  filename
       1
                   ROMS_FILES/roms_ini.nc
restart:        NRST, NRPFRST / filename
            480   -1
                   ROMS_FILES/roms_rst.nc
```

Warning ! These should be identical to the ones in romstools_param.m

```
history: LDEFHIS, NWRT, NRPFHIS / filename
        T    480    0
                 ROMS_FILES/roms_his.nc
averages: NTSAVG, NAVG, NRPFAVG / filename
        1     48    0
                 ROMS_FILES/roms_avg.nc

primary_history_fields: zeta UBAR VBAR  U  V   wrtT(1:NT)
                 T   F  F   F  F    10*F
auxiliary_history_fields:   rho Omega  W  Akv  Akt  Aks  HBL  Bostr
                   F  F    F  F    F    F    F    F
primary_averages: zeta UBAR VBAR  U  V   wrtT(1:NT)
             T   T    T    T  T   10*T
auxiliary_averages: rho Omega  W  Akv  Akt  Aks  HBL  Bostr
             F  T    T  F    T    F    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]
          10*0.d0
bottom_drag:    RDRG [m/s],  RDRG2,  Zob [m],  Cdb_min, Cdb_max
           0.0d-04     0.d-3    1.d-2     1.d-4    1.d-1
gamma2:
           1.d0
sponge:         X_SPONGE [m],    V_SPONGE [m^2/sec]
            100.e3         800.

nudg_cof:    TauT_in, TauT_out, TauM_in, TauM_out  [days for all]
          1.      360.      10.      360.
```

3) Running the model

Launching the simulation

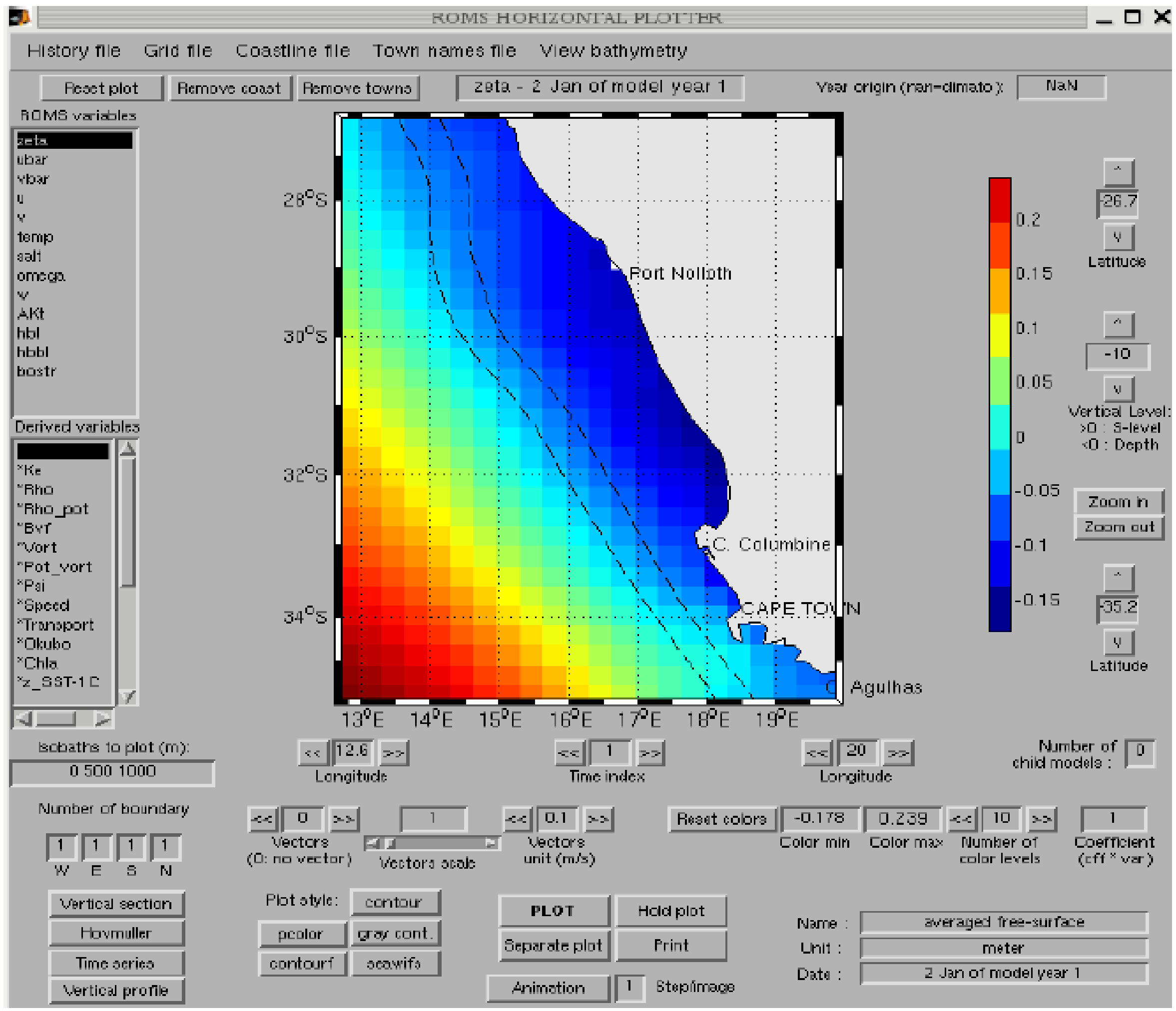
./roms roms.in

```
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=6.10483688831244468689OE+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 !!!

Vizualization

In /CMS2011-TP/TP_ROMS/Roms_tools/Run
$matlab
>> roms_gui



Embedding : Introduction

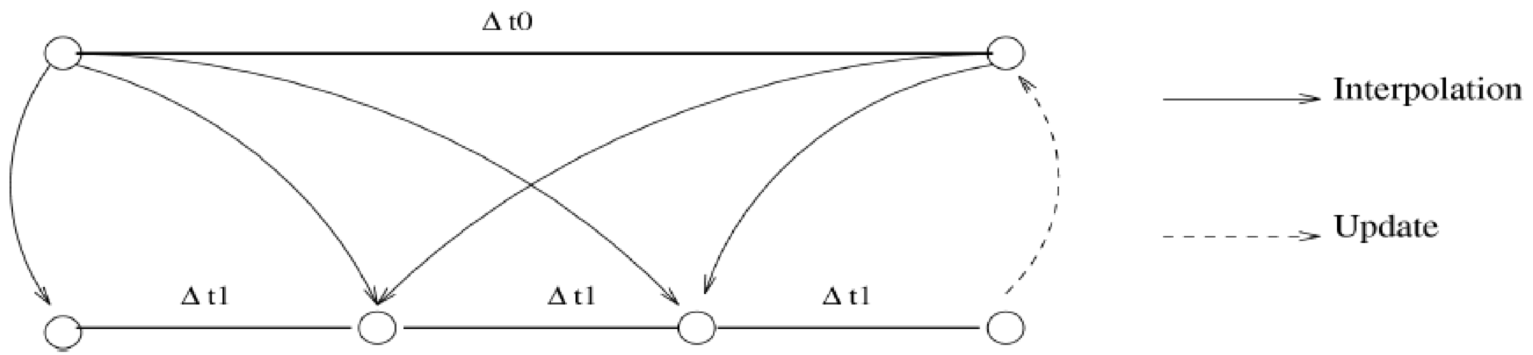
Nesting capability added to ROMS

→ 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 :

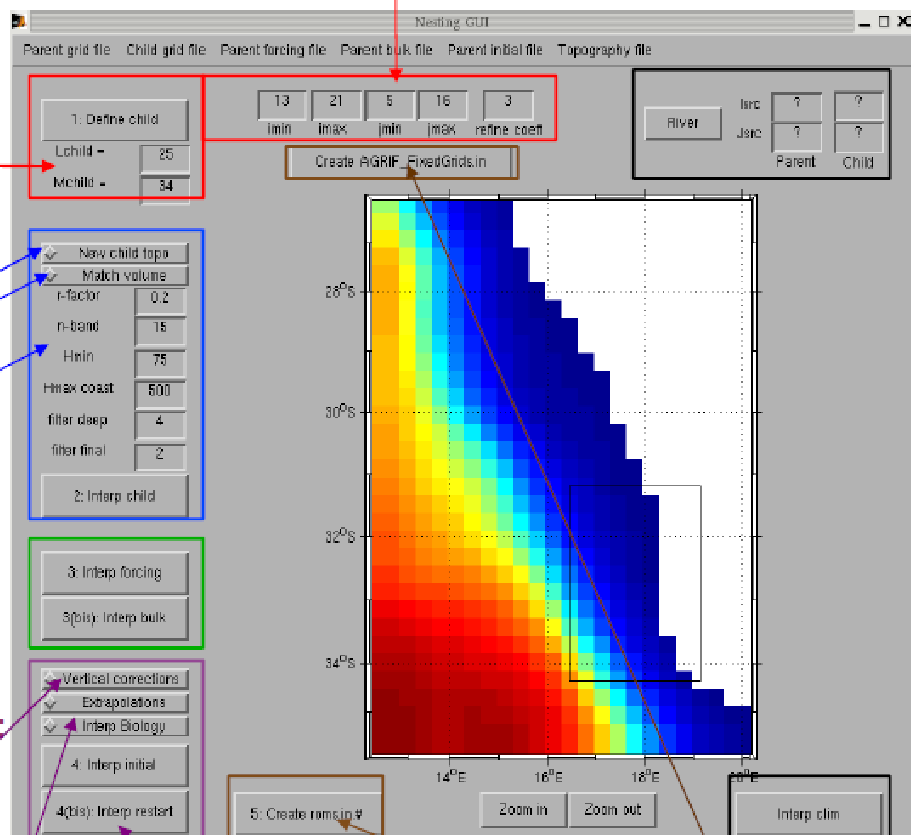
Parent file names : XXX.nc → First child file names : XXX.nc.1
second child file names : XXX.nc.2
. . .

Embedding

In the benguela test case, for the parent grid file, select in the entrance window of NestGUI ~/Run/ROMSFILES/roms_grd.nc 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

--> roms_grd.nc.*.

3- Create the surface forcing file:

Select roms_frc.nc or roms_blk.nc

--> 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

Other ROMSTOOLS processing tools …

✓Process the tides forcings : make_tides.m

✓ Process the biological forcing : make_biol.m, make_bgc.m, make_pisces.m

✓Process interanual forcing (atmopsheric and oceanic) using Opendap connection : make_ncep.m, make_OGCM.m

✓Diagnostics tools

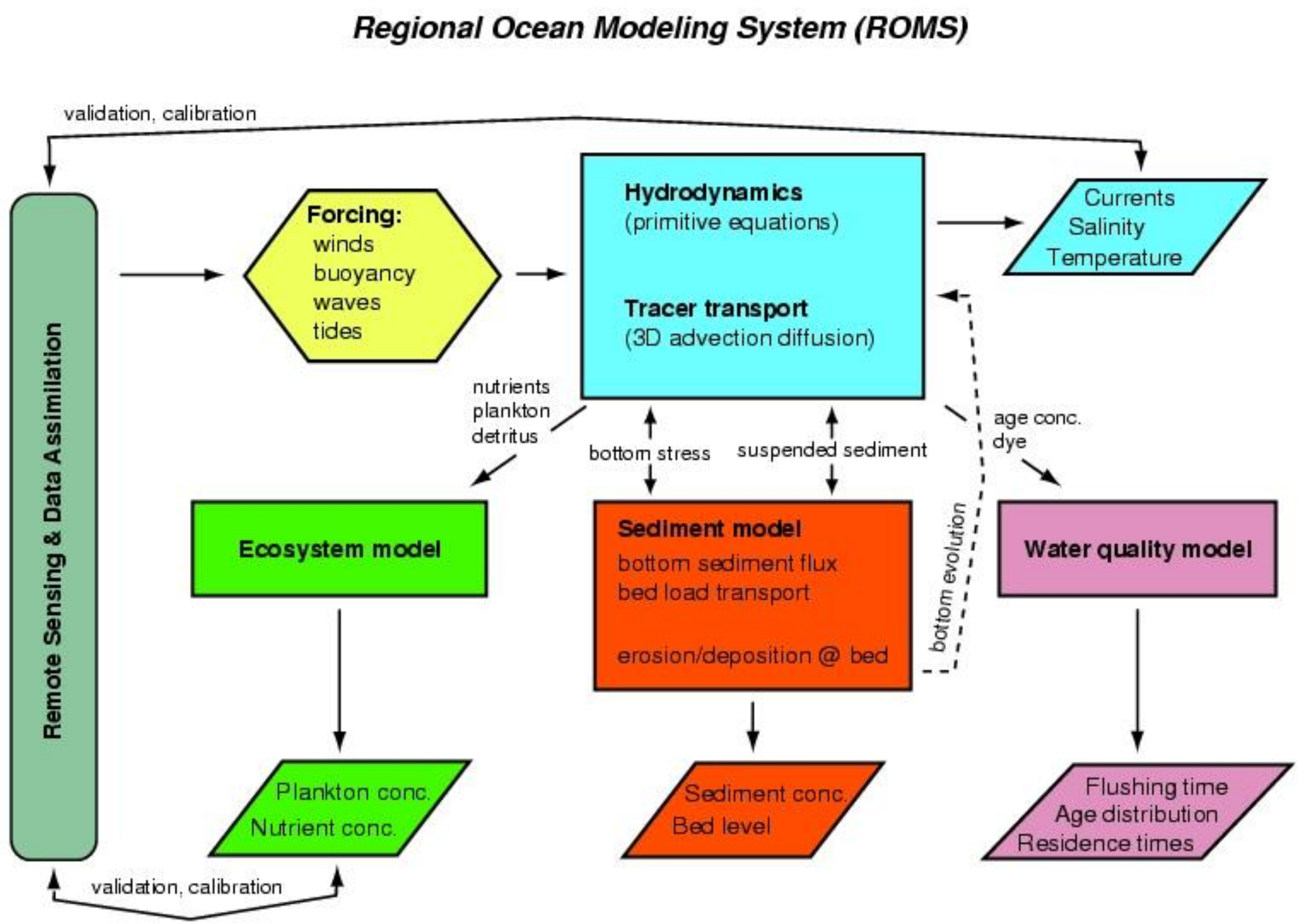
✓Script to run long simulation (→ roms_YxxMxx.nc) :

- ✓ Climatological runs : run_roms.csh
- ✓ Interannual run : run_roms_inter.csh

✓ Forecast system using Mercator and NCEP data: make_forecast.m

✓ …

Application module: Overview

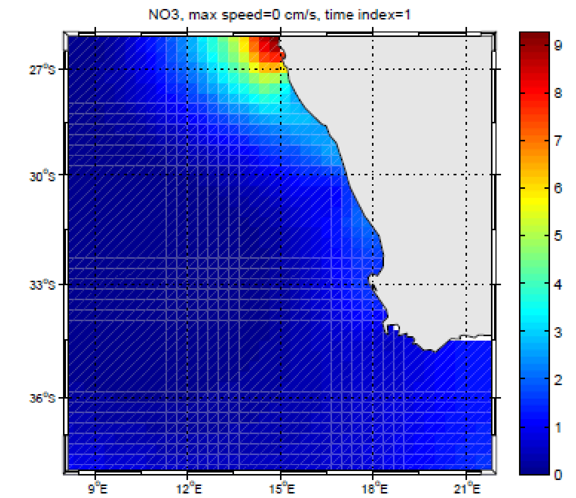


Applications module : BGC

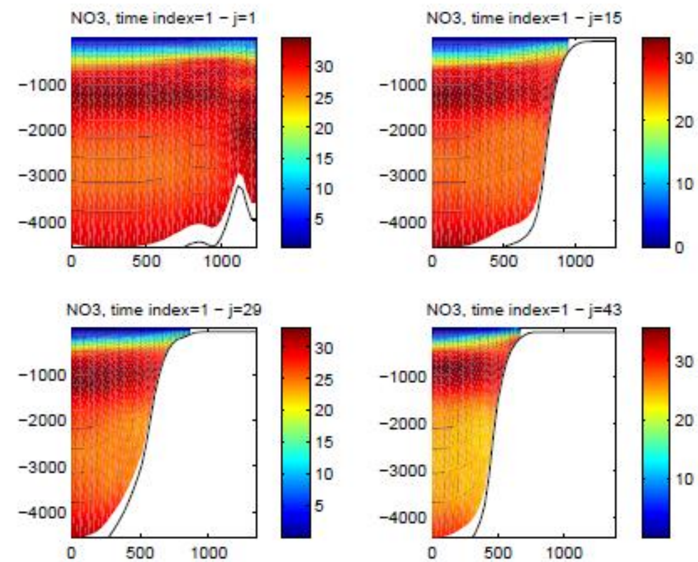
Different type of BGC model

- NPZD type model
  - ✓ NPZD (4 boxes)
  - ✓ N2PZD2 (6 boxes)
  - ✓ PISCES (24 boxes)

↓ *Complexity*



The BGC variables advected and diffused as tracer in the ROMS kernel.
Treated specifically in bio.F module, after the physical stepping.



CPPKEYS : BIOLOGY

```
# ifdef BIOLOGY
#  undef  PISCES
#  define BIO_NChlPZD
#  undef  BIO_N2P2Z2D2
#  undef  BIO_N2ChlPZD2
```

Application modules

Sediments (*#defined SEDIM*ENT)

⇨ Compute sediment variables as :

- silt & sand concentration (mg/l)
- porosity of sediment bed layer (m)
- thickness of sediment bed layer
- volume fraction of sand in sediment bed layer
- volume fraction of silt in sediment bed layer

Bottom Boundary Layer (*#defined BBL*)

⇨ Compute bottom stresses for combined wave and current *(using Soulsby, 1997) :*

- Bed Wave excursion amplitude (m)
- Bed ripple height and length (m)
- Physical hydraulic bottom roughness (m)
- Apparent hydraulic bottom roughness (m)
- Wave induced kinematic bottom stress (N/m^2)

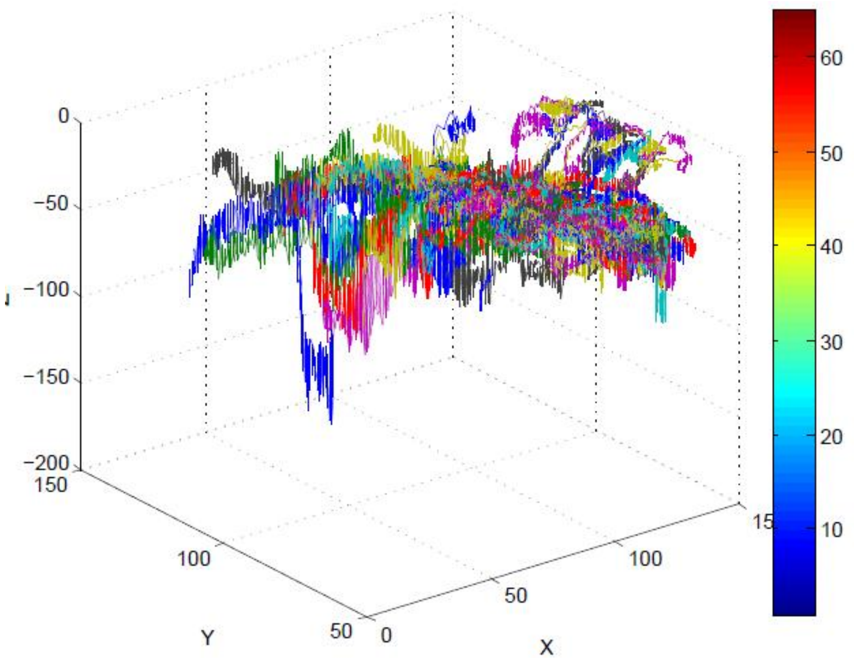
Application modules

Lagrangian floats (*#defined FLOATS*)

⇨ Tracking of lagrangian particules after injected in the domain :

- position
- depth
- temperature, salinity, density
- neutral or not density
- …

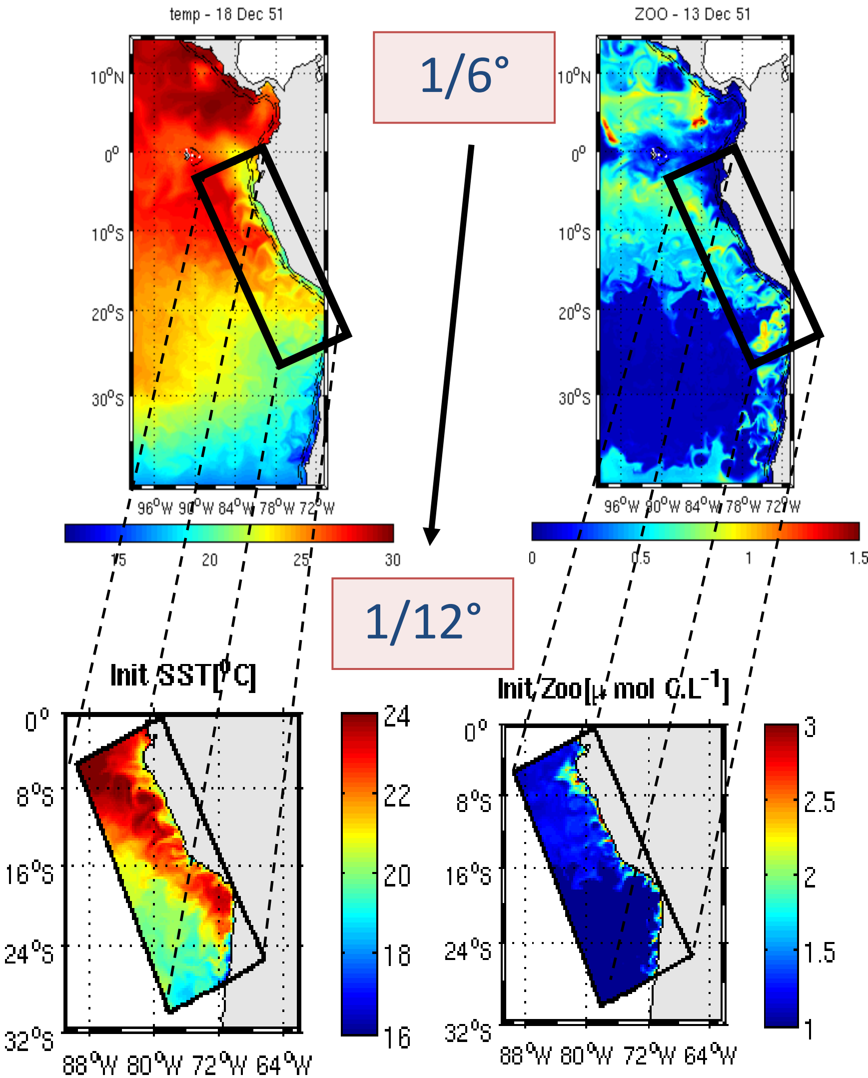
Initialisation cane be done using initfloats.m in Roms_tools/Roms_Agrif



Offline tools

✓ **"Offline" nesting Roms2Roms** (*Evan et al, 2010, Ocean Modeling*):

- Processing of roms OBC using the output of a larger roms simulation.
- Enable offline oceanic downscalling



Offline tools

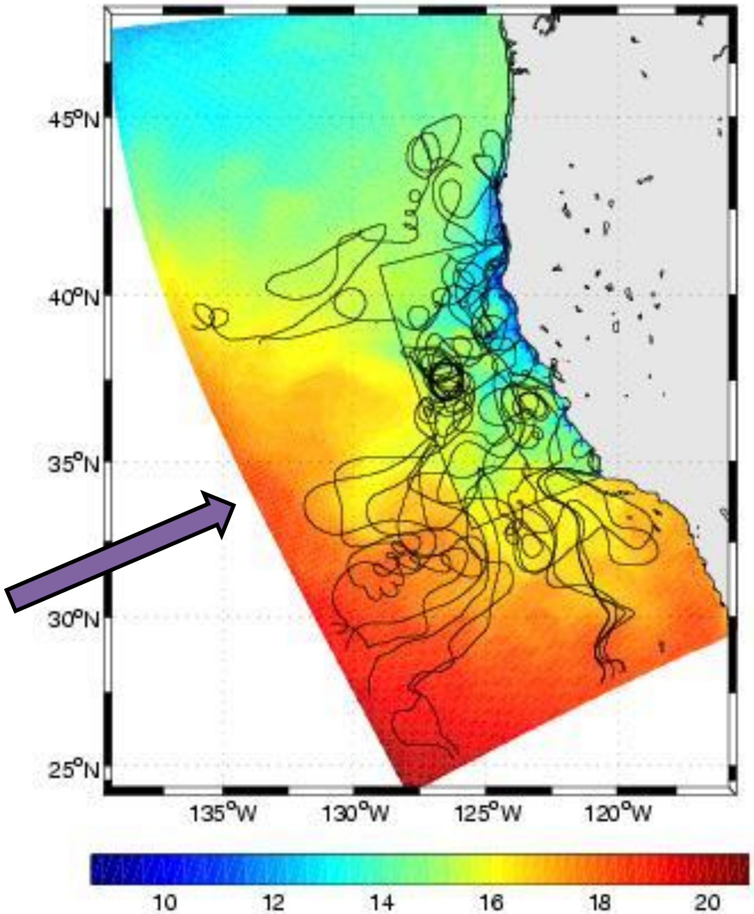
✓**Offline Lagrangian tracking :**

- Tracking of floats injected in the domain, but using the output to process the advection.
  - ✓More flexible
  - ✓Can reprocess the numerical experiment
  - ✓ But loss of information (offline !)

▪Tools available

○ **Roff** (*Capet et al*) :
http://www.atmos.ucla.edu/~capet/index.html

○ **ARIANE** (*Blanke et Grima, LPO, France*)
http://stockage.univ-brest.fr/~grima/Ariane/

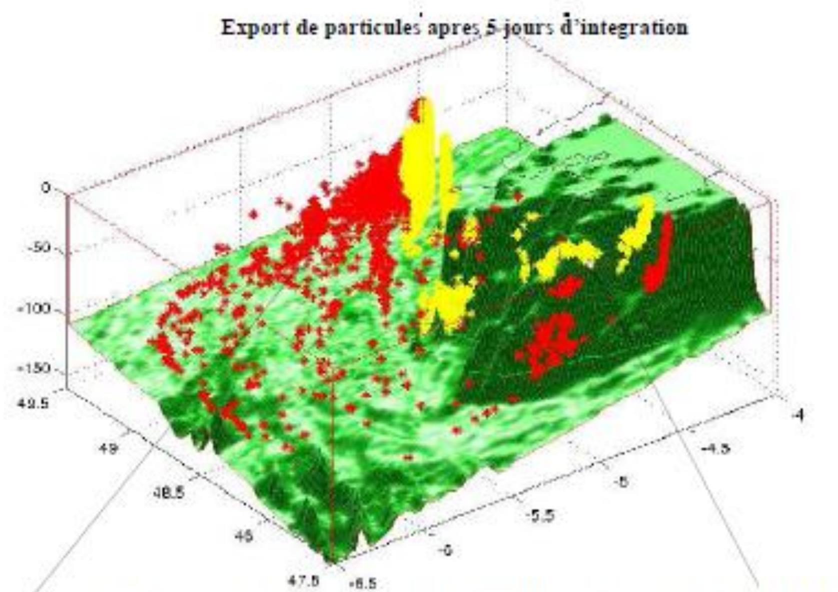


Offline tools

Example of ARIANE use :

Tidal front in Iroise Sea, PhD Gildas Cambon

http://www.legos.obs-mip.fr/~cambon/



En jaune les particules injectees au niveau de la ligne frontale

En rouge, les particules exportes travers le front (export cross−front) apres 5 jours d'integratione