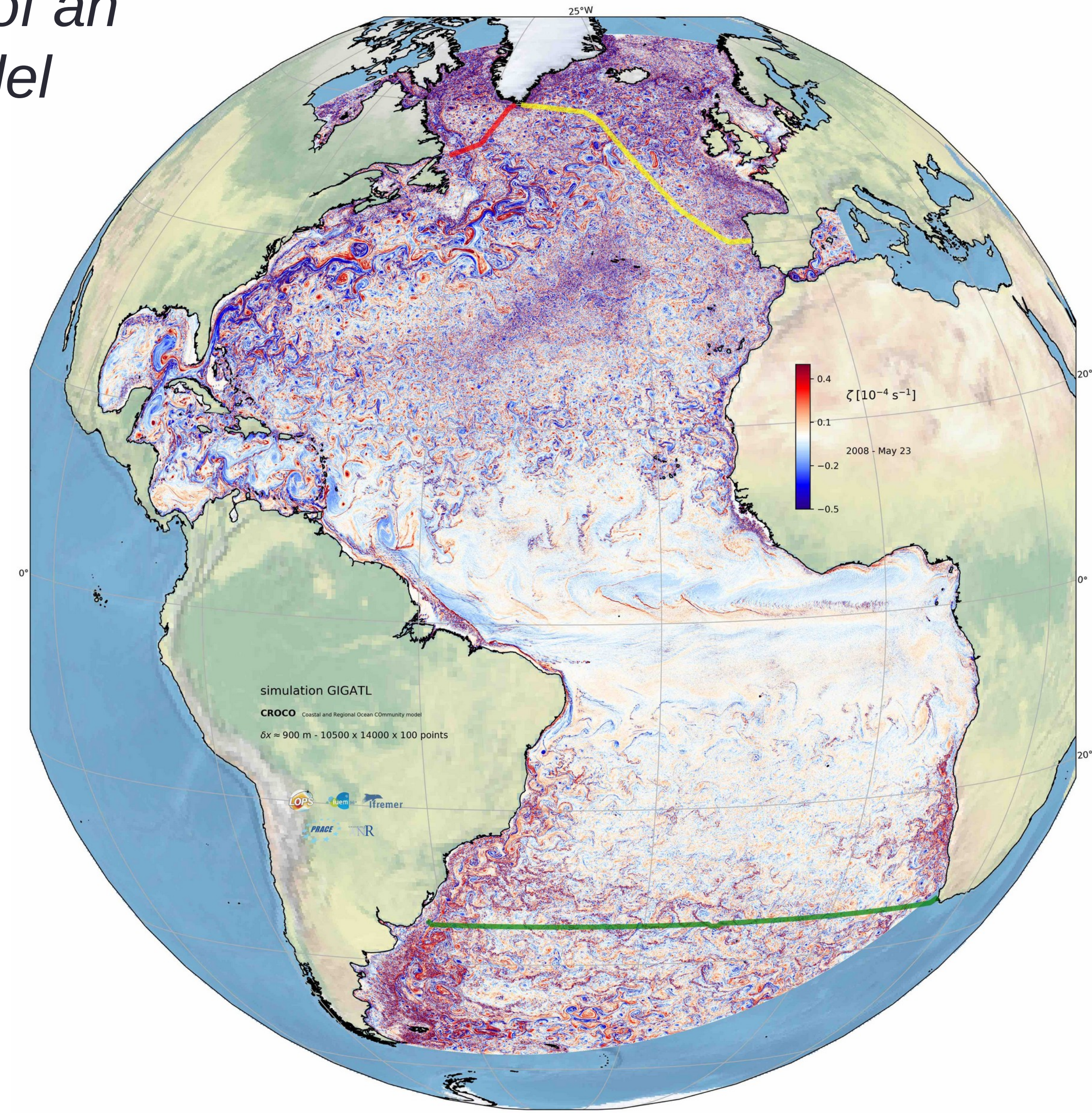
Numerical Modelling

Jonathan GULA
gula@univ-brest.fr

the anatomy of an ocean model



- **Lesson 1 :** [D109]
  - Introduction
  - Equations of motions
  - *Activity 1 [run an ocean model]*
- **Lesson 2 :** [B012]
  - Horizontal Discretization
  - *Activity 2 [Dynamics of an ocean gyre]*
- **Lesson 3 :** [B012]
  - Presentation of the model CROCO
  - Dynamics of the ocean gyre
  - *Activity 2 [Dynamics of an ocean gyre]*
- **Lesson 4 :** [D109]
  - Numerical schemes
  - *Activity 3 [Impacts of numerics]*
- **Lesson 5 :** [D109]
  - Vertical coordinates
  - *Activity 3 [Impact of topography]*
- **Lesson 6 :** [D109]
  - Boundary Forcings
  - *Activity 4 [Design a realistic simulation]*
- **Lesson 7 :** [D109]
  - Diagnostics and validation
  - *Activity 5 [Analyze a realistic simulation]*
- **Lesson 8 :** [D109]
  - *Work on your projet*

Presentations and material will be available at :

jgula.fr/ModNum/

Useful references

Extensive courses:

- MIT: https://ocw.mit.edu/courses/earth-atmospheric-and-planetary-sciences/12-950-atmospheric-and-oceanic-modeling-spring-2004/lecture-notes/
- Princeton: https://stephengriffies.github.io/assets/pdfs/GFM_lectures.pdf

Overview on ocean modelling and current challenges:

- Griffies et al., 2000, Developments in ocean climate modelling, Ocean Modelling. http://jgula.fr/ModNum/Griffiesetal00.pdf
- Griffies, 2006, "Some Ocean Model Fundamentals", In "Ocean Weather Forecasting: An Integrated View of Oceanography", 2006, Springer Netherlands. http://jgula.fr/ModNum/Griffies_Chapter.pdf
- Fox-Kemper et al, 19, “Challenges and Prospects in Ocean Circulation Models" http://jgula.fr/ModNum/FoxKemperetal19.pdf

ROMS/CROCO:

- https://www.myroms.org/wiki/
- Shchepetkin, A., and J. McWilliams, 2005: The Regional Oceanic Modeling System (ROMS): A split-explicit, free-surface, topography-following- coordinate ocean model. Ocean Modell. http://jgula.fr/ModNum/ShchepetkinMcWilliams05.pdf

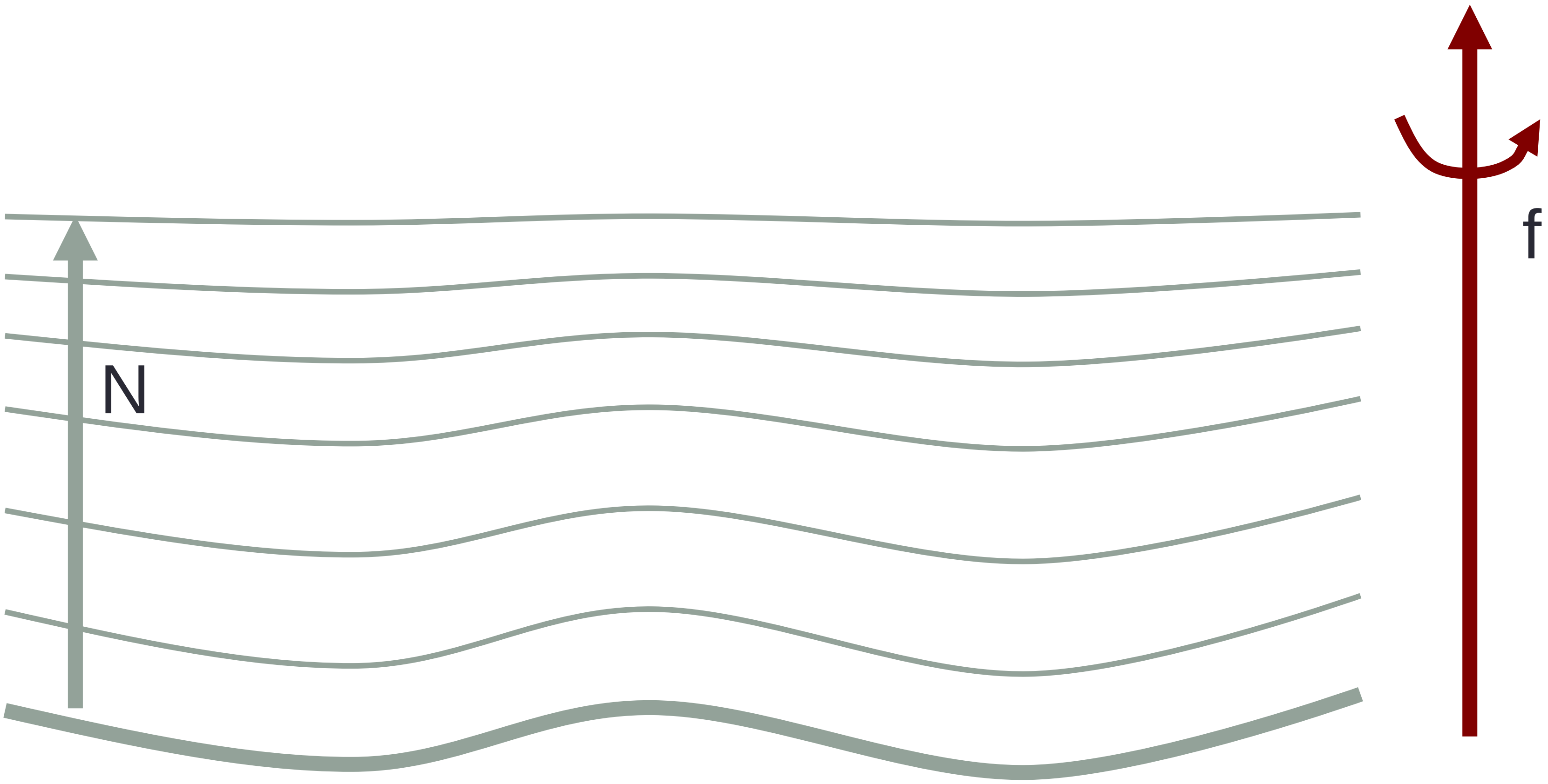
#1
Which Equations?

Jonathan GULA
gula@univ-brest.fr

Which Equations?

(See chapter 3 of Cushman-Roisin and Beckers)

Ingredients : rotation + stratification



Which Equations?

(See chapter 3 of Cushman-Roisin and Beckers)

-
-
-
-
-

[XX equations for the XX variables: ...]

Which Equations?

(See chapter 3 of Cushman-Roisin and Beckers)

- Momentum equations (3d) $$\frac{D\vec{u}}{Dt} = \ldots$$
- Conservation of mass $$\partial_t \rho + \nabla . (\rho \vec{u}) = 0$$
- Conservation of heat $$\frac{DT}{Dt} = \mathcal{S}_T$$
- Conservation of salinity $$\frac{DS}{Dt} = \mathcal{S}_S$$
- Equation of state : $$\rho = \rho(T, S, p)$$

[**7** equations for the **7** variables: u,v,w,p,T,S,ρ]

Which Equations?

(See chapter 3 of Cushman-Roisin and Beckers)

- Momentum equations (3d) $$\frac{D\vec{u}}{Dt} = \ldots$$
- Conservation of mass $$\partial_t \rho + \nabla . (\rho \vec{u}) = 0$$

- Conservation of heat $$\frac{DT}{Dt} = \mathcal{S}_T$$
- Conservation of salinity $$\frac{DS}{Dt} = \mathcal{S}_S$$
- Equation of state : $$\rho = \rho(T, S, p)$$

[**7** equations for the **7** variables: u,v,w,p,T,S,ρ]

→ Cannot be integrated forward in time consistently
We need further approximations

Which Equations?

(See chapter 3 of Cushman-Roisin and Beckers)

- The different approximations to obtain HPE:
  - Adiabatic motion $$\partial_\theta \rho|_{S,P} = \partial_S \rho|_{\theta,P} = 0$$
  - Boussinesq approximation
    (Incompressible flow) $$\rho(x,y,z,t) = \rho_0 + \rho'(x,y,z,t)$$
  - Hydrostatic approximation $$\delta = \frac{H}{L} = \frac{W}{U} \ll 1$$
  - Traditional approximation
    (Horizontal Coriolis) $$(\tilde{f}w, \tilde{f}u) \ll (fu, fv)$$

Equations for momentum/mass?

- Navier-Stokes Equations (NS)
- Non-hydrostatic Primitive Equations (NH)
- Hydrostatic Primitive Equations (PE)
- Shallow-water (SW)
- Quasi-geostrophic (QG)
- 2D Euler equations
- Etc.

Type of models

Navier Stokes

- DNS = Direct Numerical Simulation
- LES = Large Eddy Simulation

CFD

Process studies

PE

- PE = Primitive Equations models

Ocean Circulation Models

SW

- SW = Shallow-Water models

SQG

- SQG = Surface Quasi-Geostrophic models

QG

- QG = Quasi-Geostrophic models
- Etc.

Idealized models

Equations for momentum/mass?

Navier-Stokes Equations:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \vec{\nabla} \vec{u} + 2\vec{\Omega} \times \vec{u} + g\vec{k} = -\frac{\vec{\nabla} P}{\rho} + \vec{\mathcal{F}}$$

Momentum equations

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot \rho \vec{u} = 0$$

Mass conservation
(no source/sink)

Equations for momentum/mass?

Navier-Stokes Equations:

$$\boxed{\frac{\partial \vec{u}}{\partial t}} + \boxed{\vec{u} \cdot \vec{\nabla} \vec{u}} + \boxed{2\vec{\Omega} \times \vec{u}} + \boxed{g\vec{k}} = \boxed{-\frac{\vec{\nabla} P}{\rho}} + \boxed{\vec{\mathcal{F}}}$$

Time variation

Advection (inertia)

Rotation

Gravity

Pressure gradient

Forcings + Dissipation

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot \rho \vec{u} = 0$$

Mass conservation (no source/sink)

Equations for momentum/mass?

Navier-Stokes Equations:

Linearized momentum equations

+ continuity equation

+ adiabatic motion :

= Acoustic modes (sound waves)

$$\rho_0 \frac{\partial \vec{u}}{\partial t} = -\vec{\nabla} P$$

$$\frac{\partial P}{\partial t} = -\rho_0 c_s^2 \vec{\nabla} P \cdot \vec{u}$$

$$\partial_{tt} P = c_s^2 \nabla^2 P$$

With $c_s \approx 1500\,\mathrm{m\,s^{-1}}$ in water,
a model requires a very small time-step to solve these equations.

→ *Ex: sound waves would take about 1/30 sec to cross a 50-m long swimming pool*

Equations for momentum/mass?

Boussinesq Approximation:

Density perturbations small compared to mean background value:

$$\rho = \rho_0 + \rho' \qquad\qquad \rho' << \rho_0$$

Linearize all terms involving a product with density,

except the gravity term which is already linear:

$$\rho\vec{u} \quad \rightarrow \quad \rho_0\vec{u}$$

$$\rho g \quad \rightarrow \quad \rho g$$

Equations for momentum/mass?

Boussinesq Approximation :

[+ incompressibility or adiabatic]

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot \rho \vec{u} = 0$$

Continuity equation

$$\begin{aligned} \partial_t \rho + \nabla . \rho \vec{u} &= \partial_t \rho + \rho \nabla . \vec{u} + \vec{u} \nabla \rho \\ &= (\rho_0 + \rho') \nabla . \vec{u} + D_t \rho' \\ &\sim \rho_0 \nabla . \vec{u} \end{aligned}$$

$$\vec{\nabla} \cdot \vec{u} = 0$$

Mass conservation → Volume conservation

Equations for momentum/mass?

Non hydrostatic boussinesq (NH):

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \vec{\nabla} \vec{u} + 2\vec{\Omega} \times \vec{u} + \frac{\rho}{\rho_0} g\vec{k} = -\frac{\vec{\nabla} P}{\rho_0} + \frac{\vec{\mathcal{F}}}{\rho_0} + \frac{\vec{\mathcal{D}}}{\rho_0}$$

$$\vec{\nabla} \cdot \vec{u} = 0$$

Easier to solve than Navier-Stokes, but still requires to invert a 3d elliptic equation for P (computationally expansive)

Equations for momentum/mass?

Hydrostatic balance:

The vertical component of the Boussinesq momentum equations is

$$\partial_t w + \vec{u} . \vec{\nabla} w + 2\Omega \cos\phi u + \boxed{\frac{\rho}{\rho_0} g = \frac{1}{\rho_0} \partial_z P} + F_w + D_w$$

For long horizontal motions (L >> H) the dominant balance is

H ~3000 m
L ~3000 km

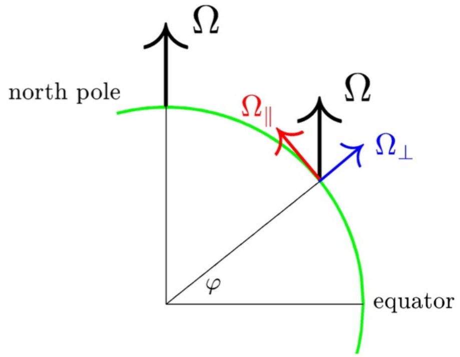
$$\frac{\partial P}{\partial z} = -\rho g$$

Such that pressure is just a vertical integral: $P = \int_z^\eta g\rho dz$

Equations for momentum/mass?

Traditional approximation

= neglect horizontal Coriolis term



$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \vec{\nabla} \vec{u} + 2\vec{\Omega} \times \vec{u} + \frac{\rho}{\rho_0} g \vec{k} = -\frac{\vec{\nabla} P}{\rho_0} + \vec{\mathcal{F}} + \vec{\mathcal{D}}$$

$$\downarrow$$

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \vec{\nabla} \vec{u} + f\vec{k} \times \vec{u} + \frac{\rho}{\rho_0} g \vec{k} = -\frac{\vec{\nabla} P}{\rho_0} + \vec{\mathcal{F}} + \vec{\mathcal{D}}$$

Equations for momentum/mass?

Hydrostatic Primitive Equations (PE)

- 2d momentum with Boussinesq approximation:

$$\frac{\partial u}{\partial t} + \vec{u}\cdot\vec{\nabla}_H u + w\frac{\partial u}{\partial z} - fv = -\frac{\partial_x P}{\rho_0} + \mathcal{F}_u + \mathcal{D}_u$$

$$\frac{\partial v}{\partial t} + \vec{u}\cdot\vec{\nabla}_H v + w\frac{\partial v}{\partial z} + fu = -\frac{\partial_y P}{\rho_0} + \mathcal{F}_v + \mathcal{D}_v$$

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

- Continuity equation for an incompressible fluid: $\vec{\nabla}\cdot\vec{u} = 0$

Equations for PE Ocean Models

Hydrostatic Primitive Equations (PE)

- 2d momentum with Boussinesq approximation:

$$\begin{aligned}
\frac{\partial u}{\partial t} + \vec{u} \cdot \vec{\nabla}_H u + w \frac{\partial u}{\partial z} - fv &= -\frac{\partial_x P}{\rho_0} + \mathcal{F}_u + \mathcal{D}_u \\
\frac{\partial v}{\partial t} + \vec{u} \cdot \vec{\nabla}_H v + w \frac{\partial v}{\partial z} + fu &= -\frac{\partial_y P}{\rho_0} + \mathcal{F}_v + \mathcal{D}_v
\end{aligned}$$

- Hydrostatic: $$\frac{\partial P}{\partial z} = -\rho g$$
- Continuity equation for an incompressible fluid: $$\vec{\nabla} \cdot \vec{u} = 0$$
- Conservation of heat and salinity $$\frac{DT}{Dt} = \mathcal{S}_T \qquad \frac{DS}{Dt} = \mathcal{S}_S$$
- Equation of state : $$\rho = \rho(T, S, z)$$

Equations for PE Ocean Models

Hydrostatic Primitive Equations (PE)

- 4 prognostics equations for **u, v, T, S**
- 3 diagnostics equations for **w, ρ, P**

Equations for PE Ocean Models

Hydrostatic Primitive Equations (PE)

- 2d momentum with Boussinesq approximation:

?

$$\begin{aligned}
\frac{\partial u}{\partial t} + \vec{u}\cdot\vec{\nabla}_H u + w\,\frac{\partial u}{\partial z} - fv &= -\frac{\partial_x P}{\rho_0} \boxed{+ \mathcal{F}_u + \mathcal{D}_u} \\
\frac{\partial v}{\partial t} + \vec{u}\cdot\vec{\nabla}_H v + w\,\frac{\partial v}{\partial z} + fu &= -\frac{\partial_y P}{\rho_0} \boxed{+ \mathcal{F}_v + \mathcal{D}_v}
\end{aligned}$$

- Hydrostatic: $$\frac{\partial P}{\partial z} = -\rho g$$
- Continuity equation for an incompressible fluid: $\vec{\nabla}\cdot\vec{u} = 0$
- Conservation of heat and salinity $\frac{DT}{Dt} = \boxed{\mathcal{S}_T}$ $\frac{DS}{Dt} = \boxed{\mathcal{S}_S}$
- Equation of state : $\rho = \rho(T, S, z)$

Equations for PE Ocean Models

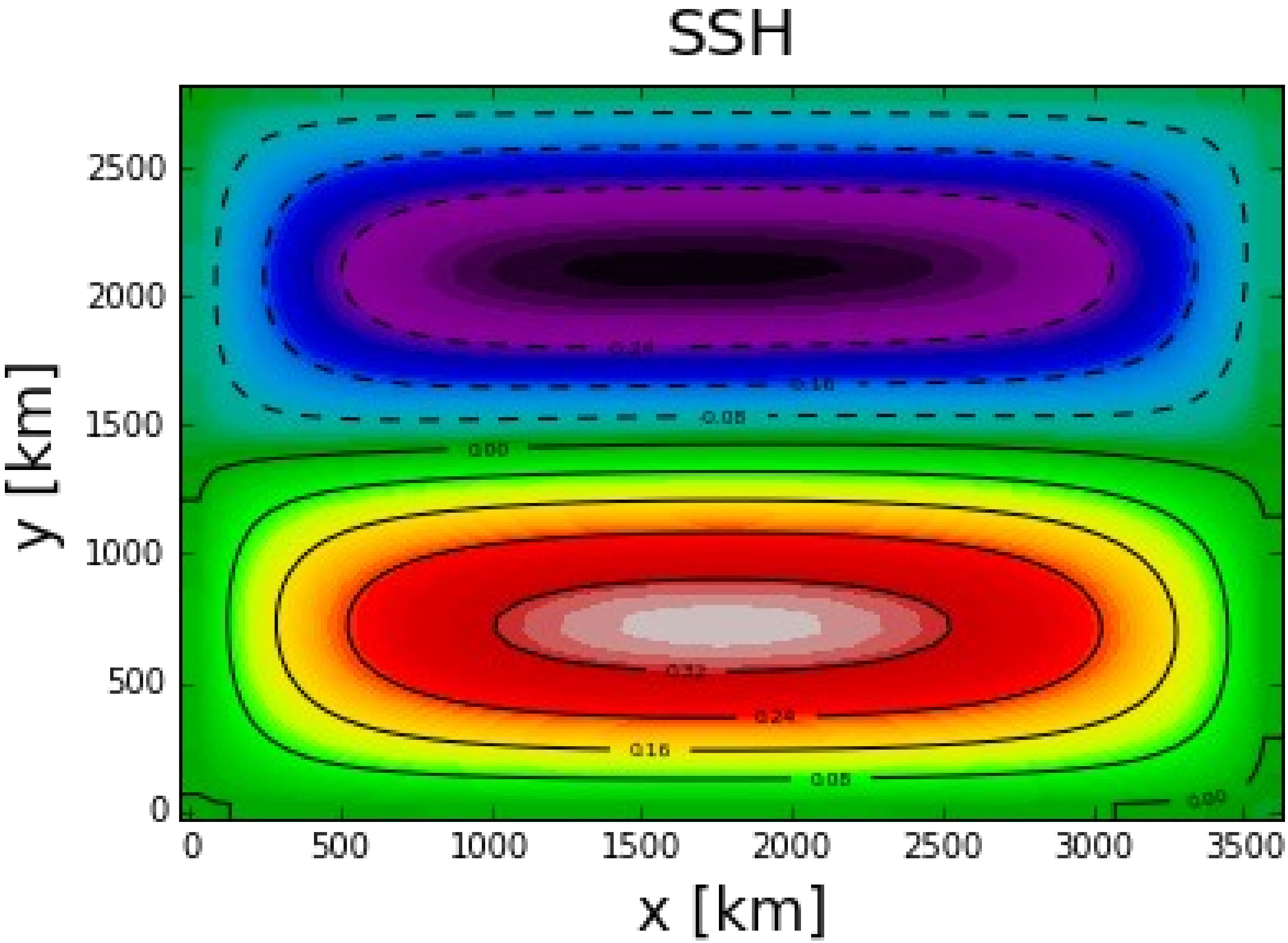
Hydrostatic Primitive Equations (PE)

- 4 prognostics equations for **u, v, T, S**
- 3 diagnostics equations for **w, ρ, P**

+ Forcings (wind, heat flux)

+ sub-grid scale parameterizations (bottom drag, mixing, etc.)

Activity 1 – Run an idealized ocean basin



Activity 1 – Run an idealized ocean basin

- **Jobcomp** (compilation)
- **cppdefs.h** (Numerical/physical options)
- **param.h** (gris size/ parallelisation)
- **croco.in** (choice of variables, parameter values, etc.)

1) Preparing and compiling the model

For that use the the jobcomp bash 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

1) 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

Defined in romstools_param.m

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.

1) 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) 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.
```

Activity 1 – Run an idealized ocean basin

- **param.h**

```
parameter (LLm0=60,   MMm0=50,   N=10)
```

- **cppdefs.h**

```
# define UV_COR
# define UV_VIS2
# define TS_DIF2

# define ANA_GRID
# define ANA_INITIAL
```

- **ana_grid.F**

```
f0=1.E-4
beta=0.
```

- **croco.in**

```
bottom_drag:     RDRG(m/s),      RDRG2, Zob [m],  Cdb_min, Cdb_max
                 3.e-4           0.      0.        0.      0.
gamma2:
                 1.
lin_EOS_cff:  R0 [kg/m3], T0 [Celsius], S0 [PSU], TCOEF [1/Celsius], SCOEF [1/PSU]
              30.         0.            0.        0.28               0.
lateral_visc:   VISC2    [m^2/sec ]
                 1000.  0.
tracer_diff2: TNU2         [m^2/sec]
                 1000. 0.
```

Homework

- For next time:
  - Read https://www.jgula.fr/ModNum/Stommel48.pdf
  - Read https://www.jgula.fr/ModNum/Munk50.pdf