Numerical Modelling

Jonathan GULA
gula@univ-brest.fr

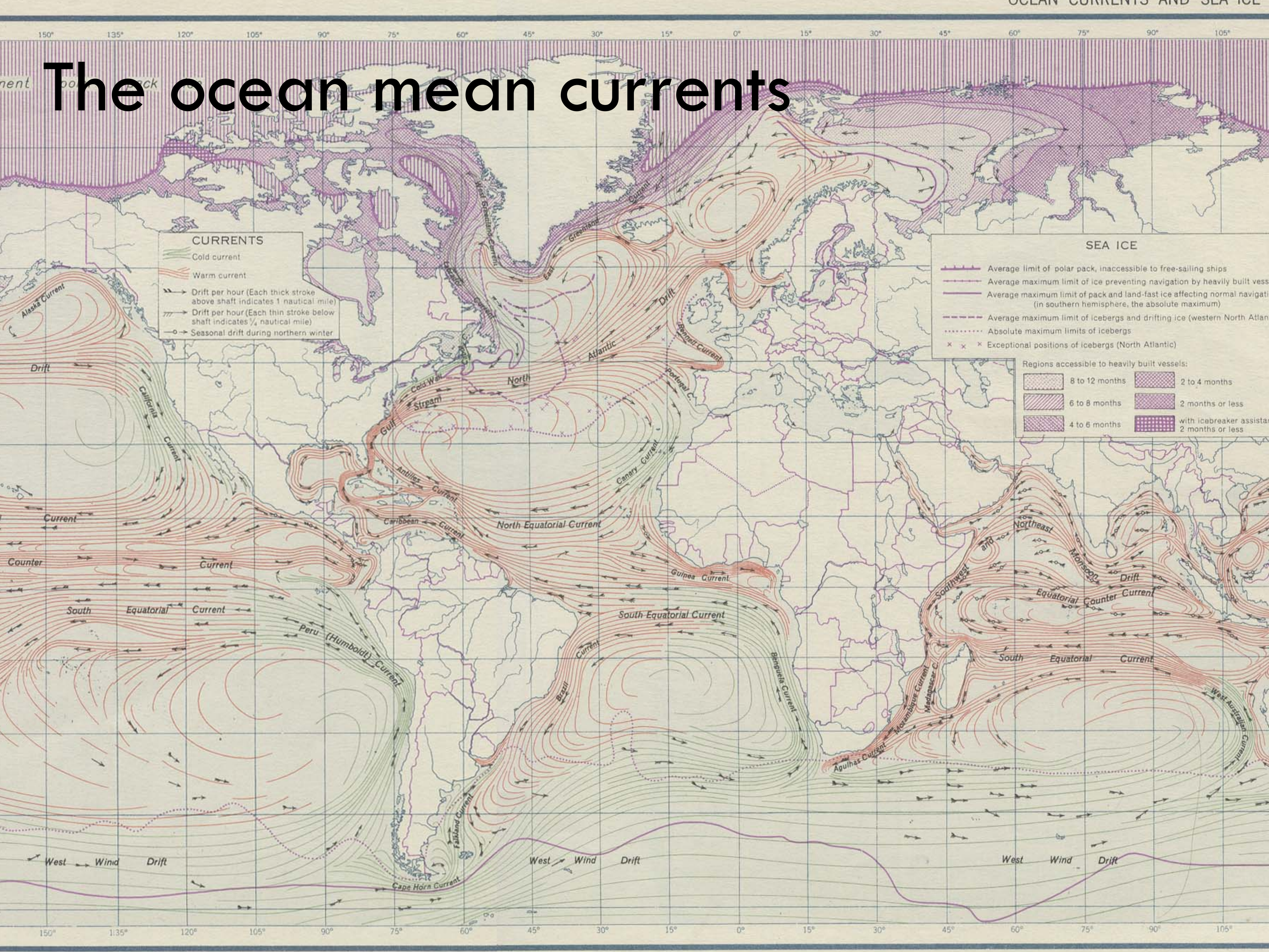
the anatomy of an ocean model

Introduction

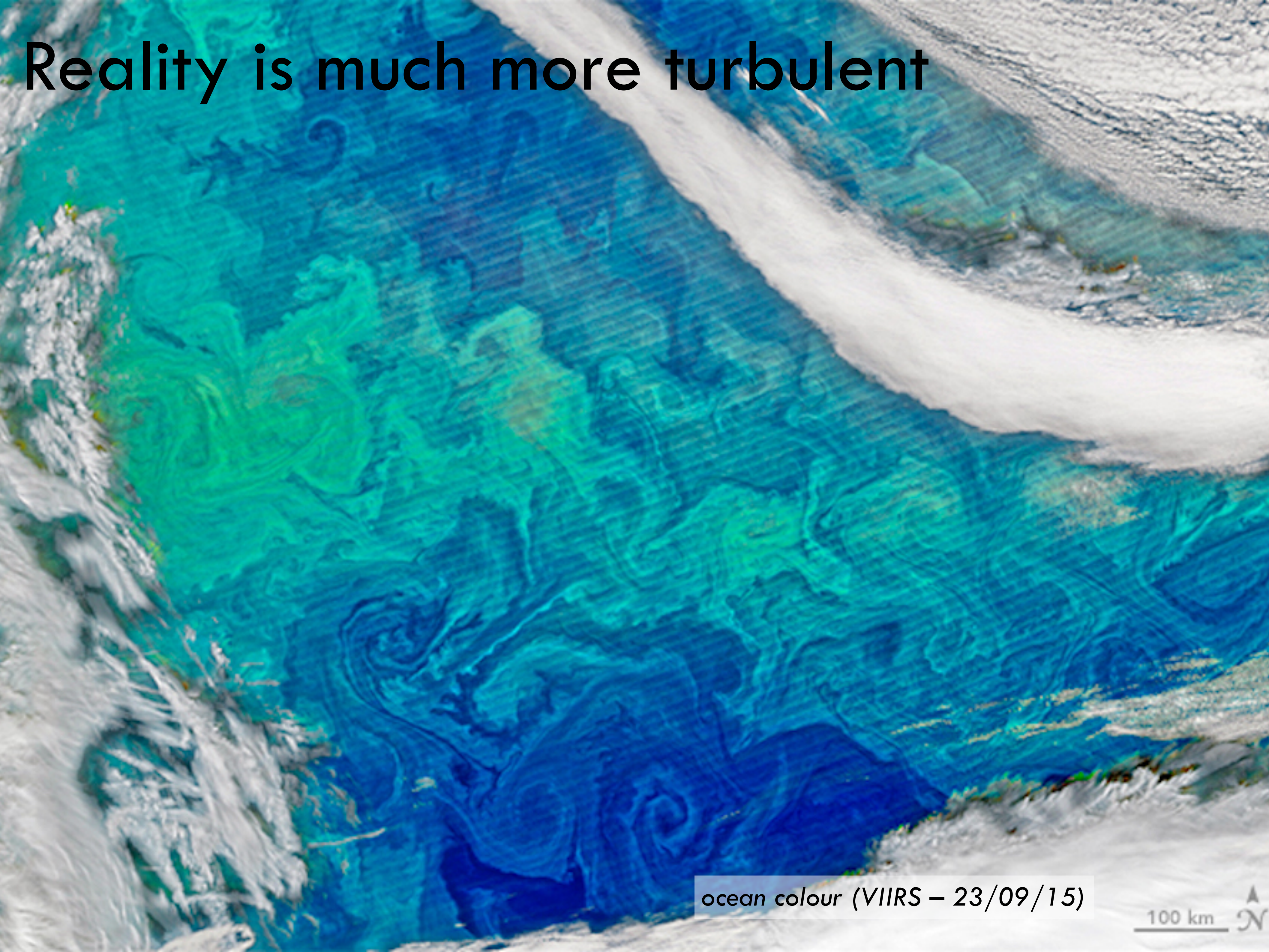
Master's degree 2nd year Marine Physics

Jonathan GULA
gula@univ-brest.fr

The ocean mean currents

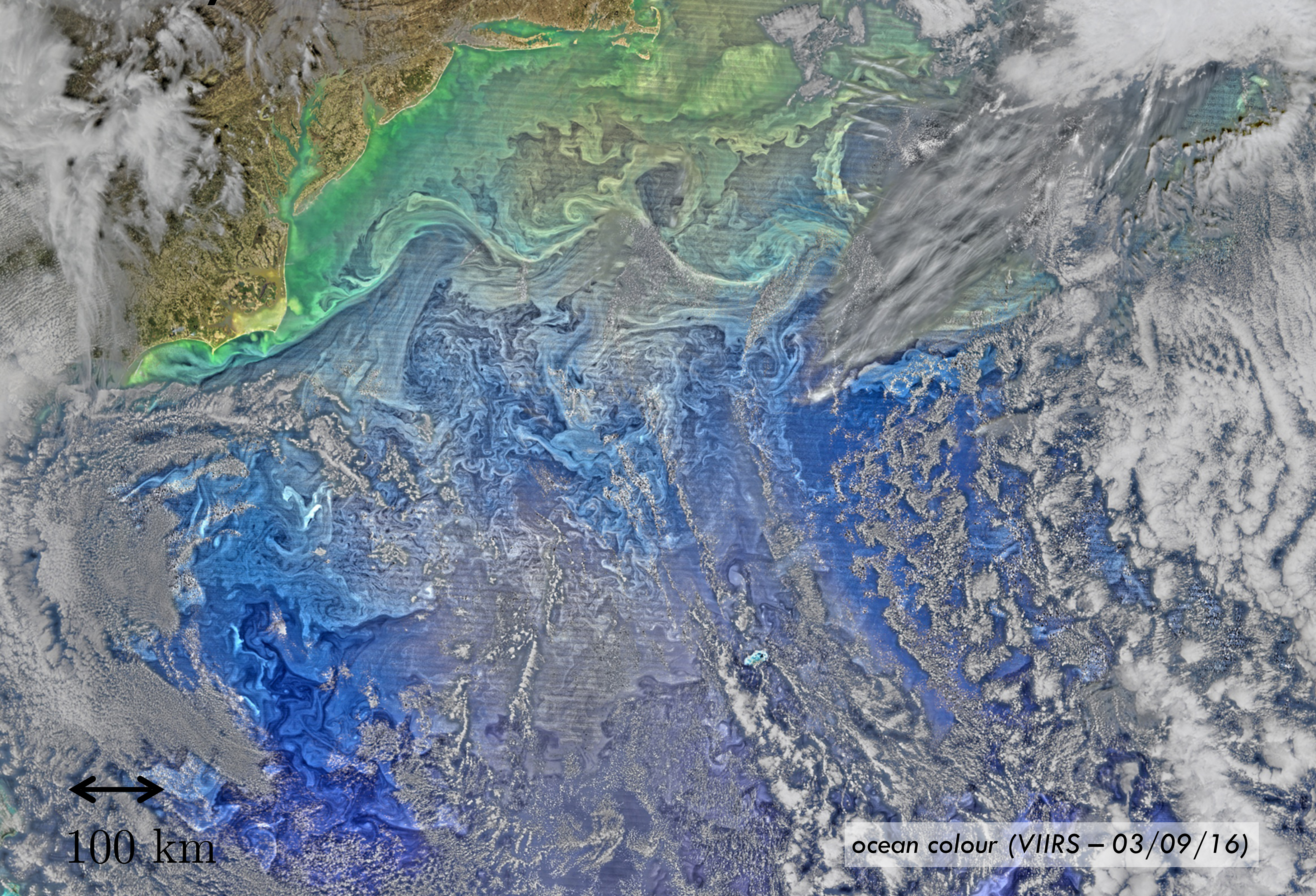


Reality is much more turbulent



ocean colour (VIIRS – 23/09/15)

Reality is much more turbulent



We know the equations:

Navier-Stokes
[Momentum equations]

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

Conservation of mass

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

Conservation of heat

$$\frac{DT}{Dt} = \mathcal{S}_T$$

Conservation of salinity

$$\frac{DS}{Dt} = \mathcal{S}_S$$

Equation of state

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

We know the equations:

Navier-Stokes
[Momentum equations]

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

Conservation of mass

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

Conservation of heat

$$\frac{DT}{Dt} = \mathcal{S}_T$$

Conservation of salinity

$$\frac{DS}{Dt} = \mathcal{S}_S$$

Equation of state

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

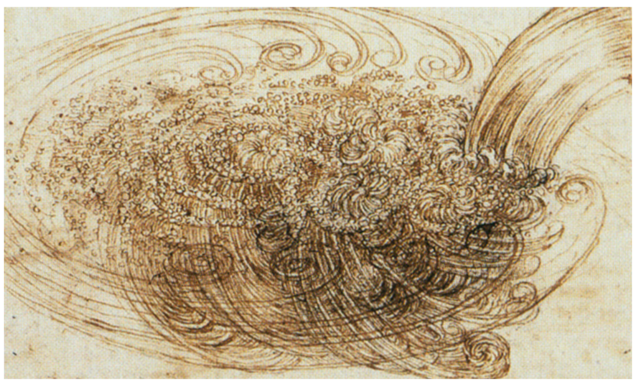
But we don't know the solutions…

We know the equations:

Navier-Stokes
[Momentum equations]

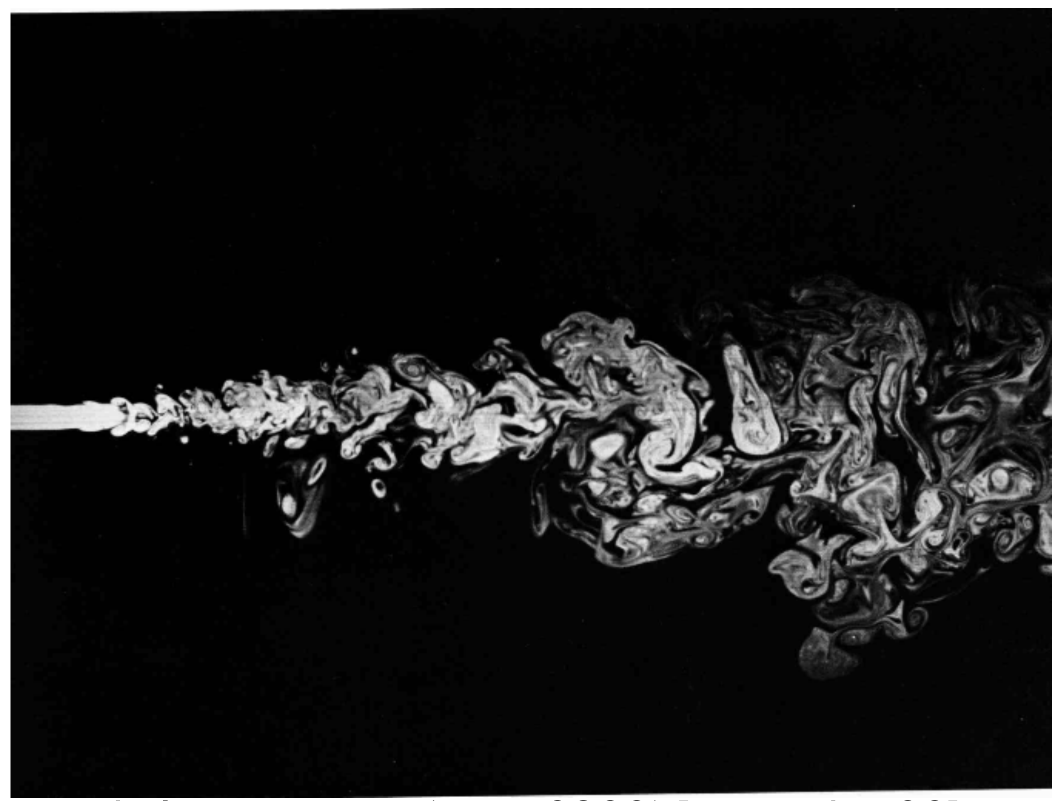
$$\frac{\partial \vec{u}}{\partial t} + \boxed{\vec{u} \cdot \vec{\nabla} \vec{u}} + 2\vec{\Omega} \times \vec{u} + g\vec{k} = -\frac{\vec{\nabla} P}{\rho} + \vec{\mathcal{F}}$$

Non-linear terms = Turbulence



turbolenza by da Vinci [1507]

Big whorls have little whorls,
which feed on their velocity;
And little whorls have lesser whorls,
And so on to viscosity.
L.F. Richardson [1922]

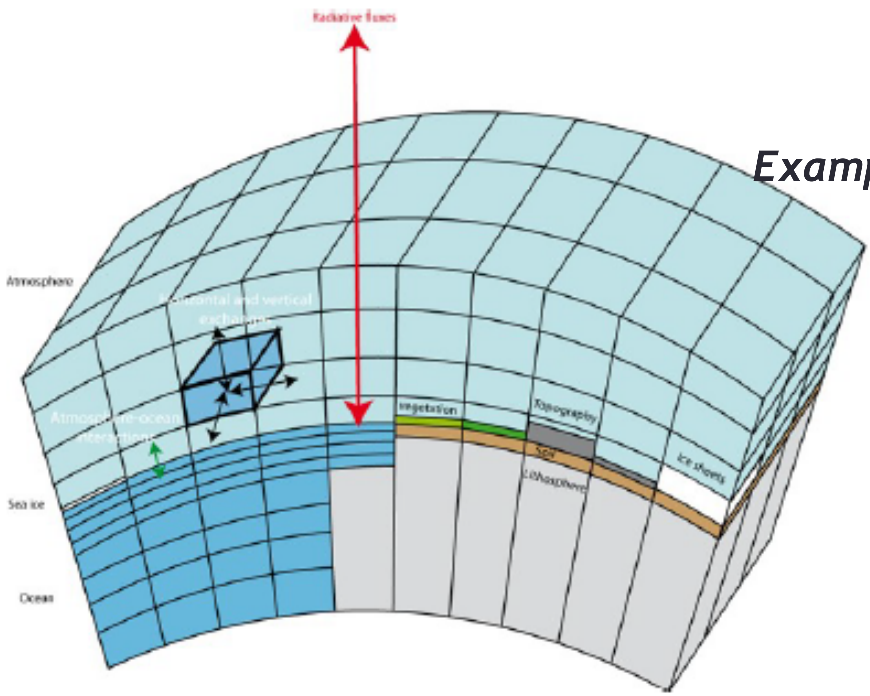


Turbulent water jet (Re = 2300) [Van Dyke, 82]

Ocean modeling principle

Ocean model = simplified representation of physical processes that take place in the ocean.

The ocean is divided into boxes : discretization



Example of a finite difference grid

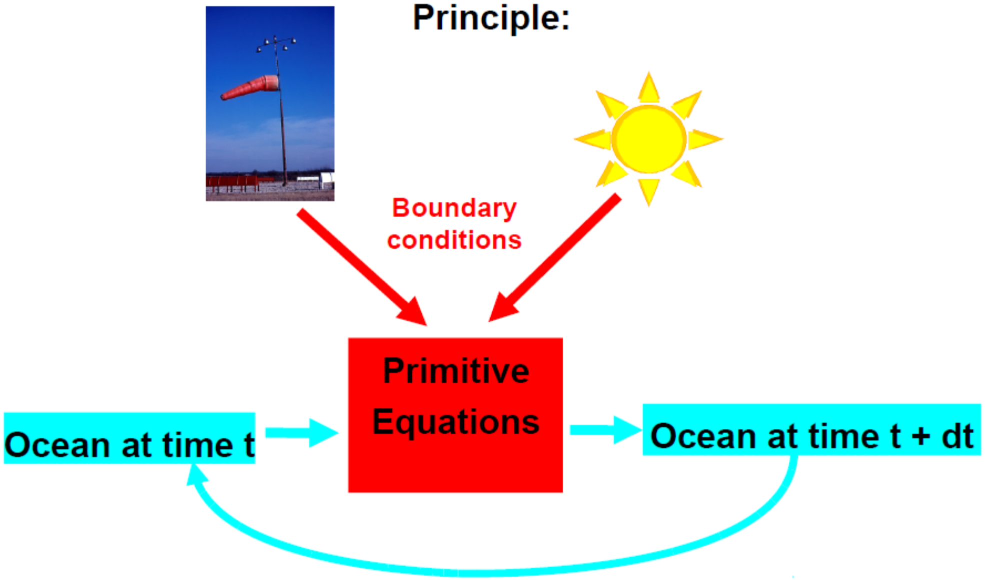
Ocean modeling principle

If we know:

- The **ocean state at time *t* :**

  u, v, w, T, S, …

- **Boundary conditions :** surface, bottom, lateral sides



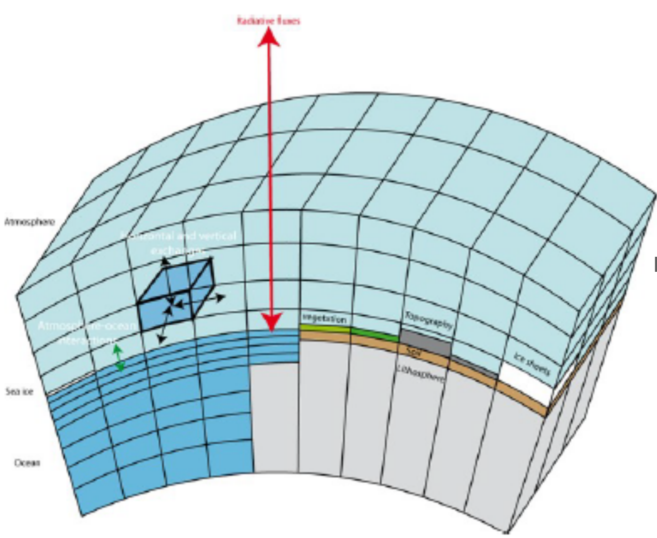
→ We can **compute the ocean state at time *t+dt*** using **numerical approximations of fluid dynamics equations**

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

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

Navier Stokes equations

+



Grid Discretization

+

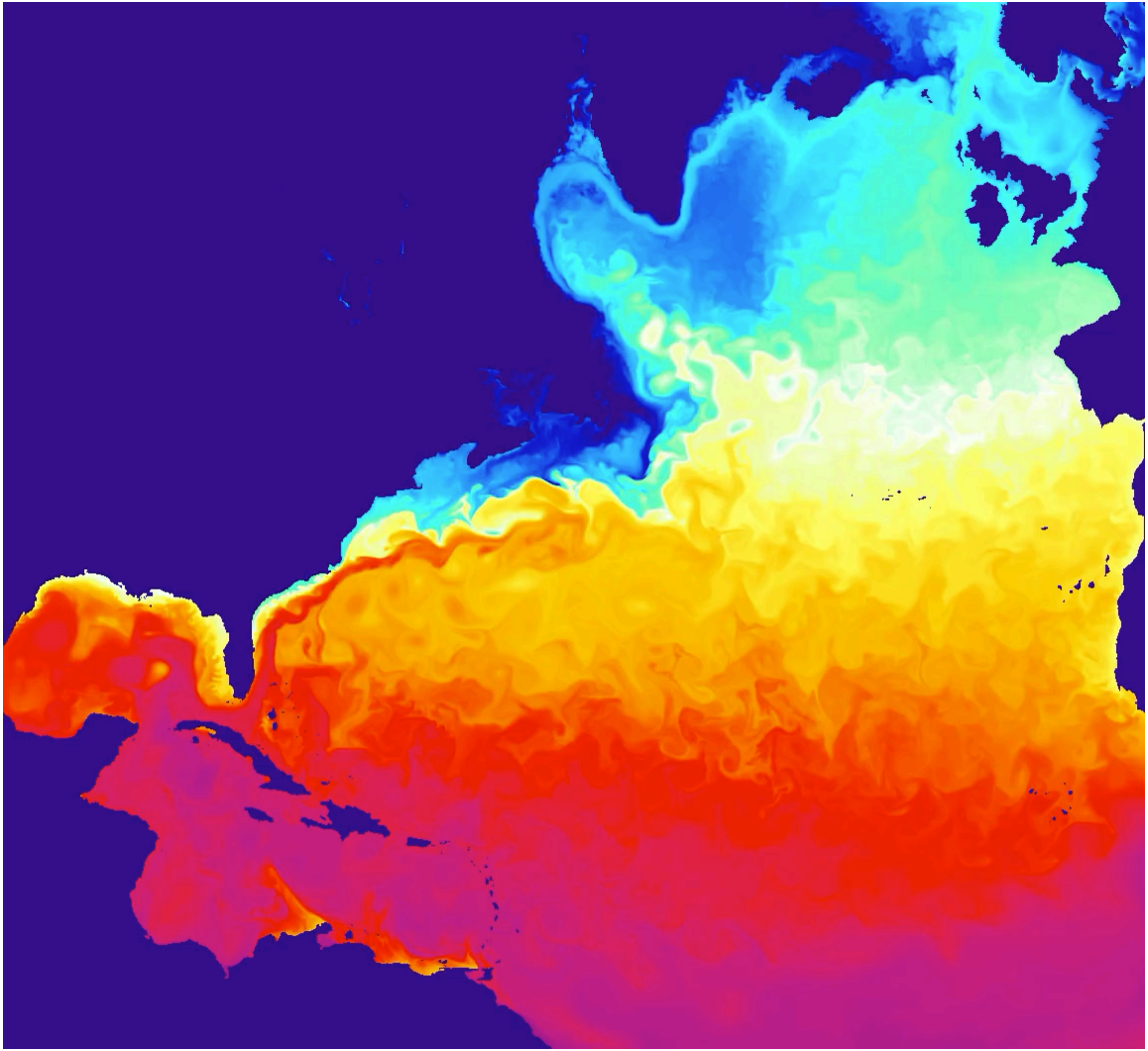
Supercomputer (Curie – CEA)

One way to solve them (approximately): **Numerical modelling**

Realistic Modelling:

North-Atlantic coupled simulations:

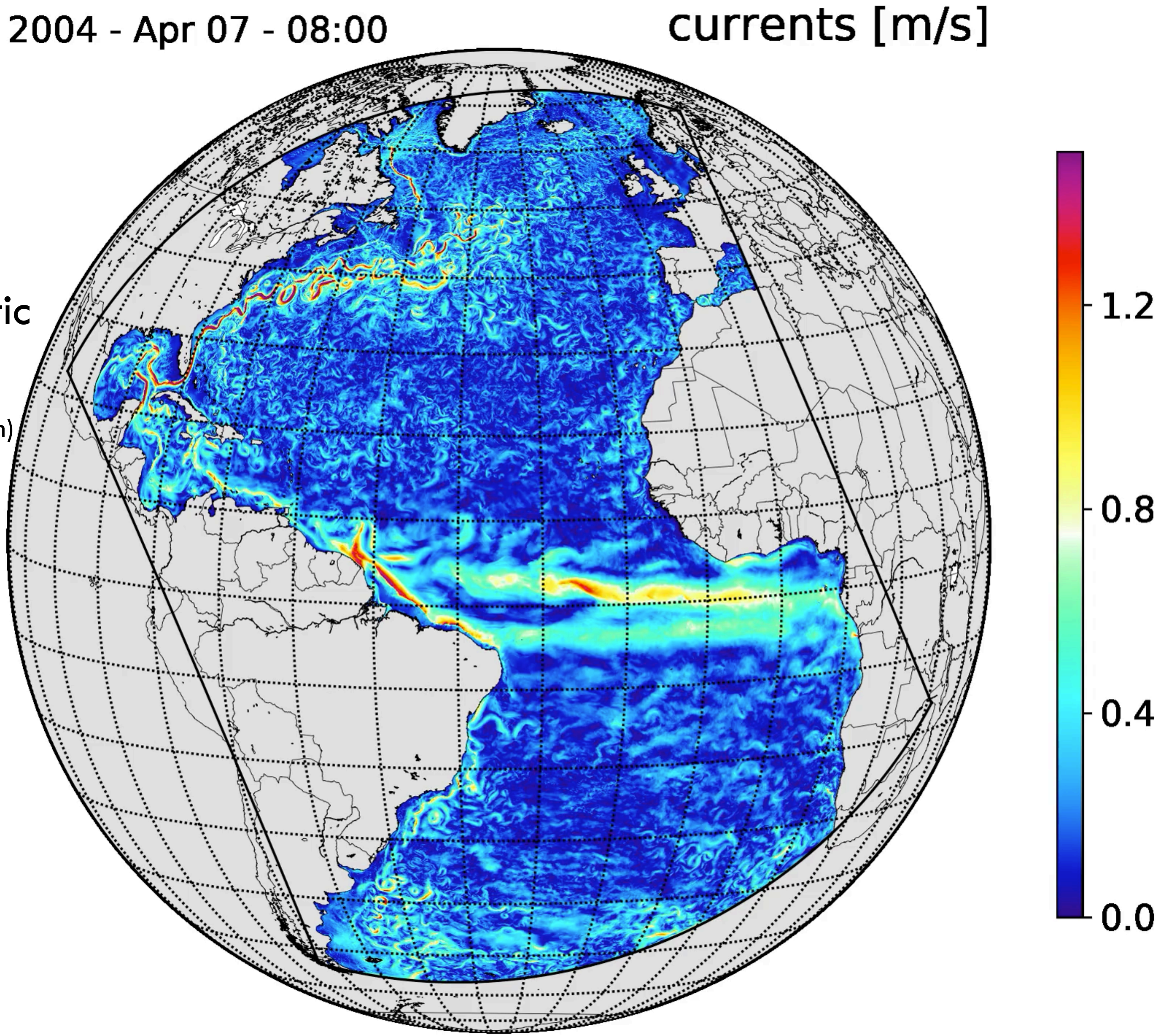
- oceanic model (6 km)
- atmospheric model (18 km)



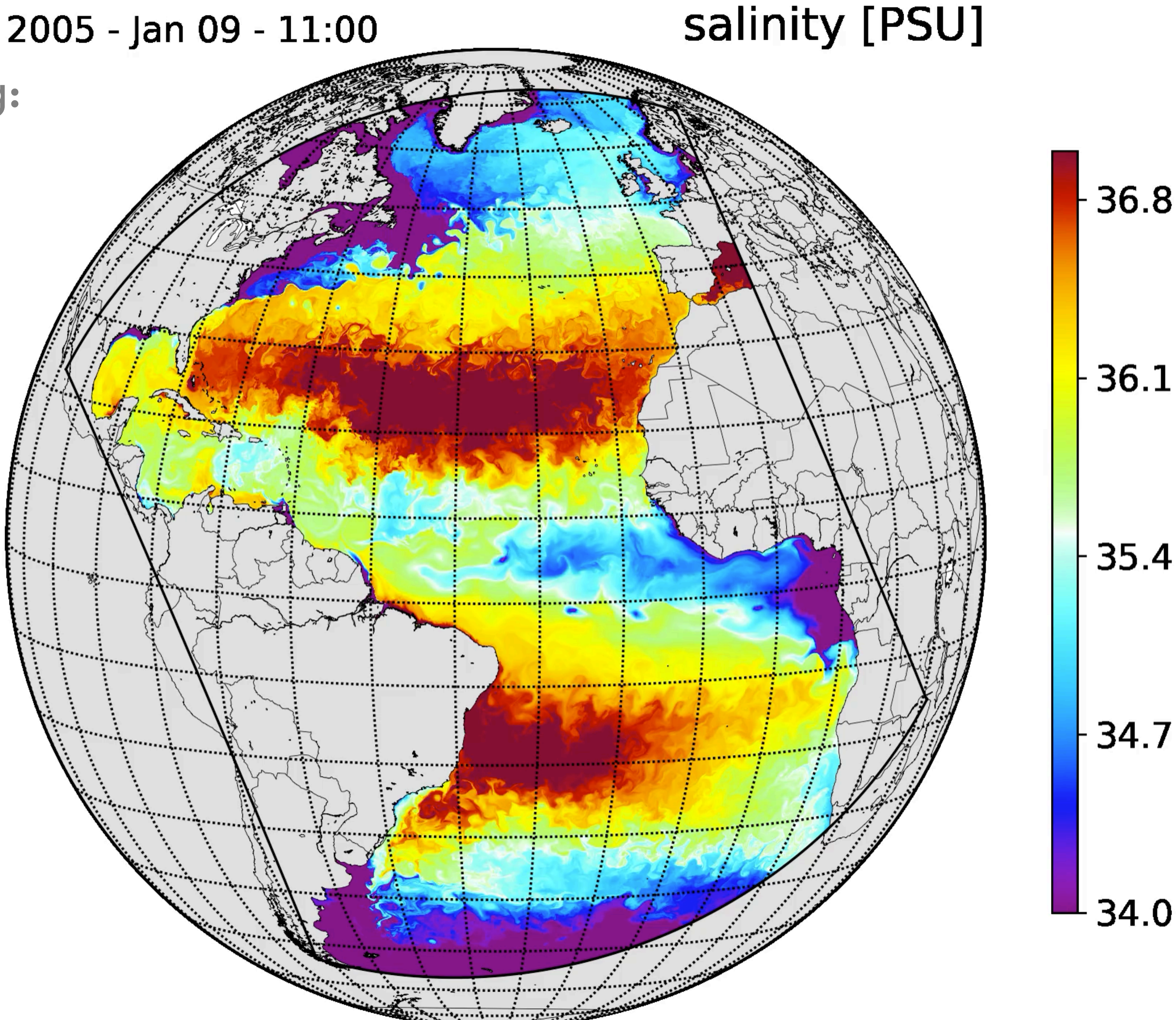
Realistic Modelling:

Forced Atlantic simulations:

- oceanic model (3 km)



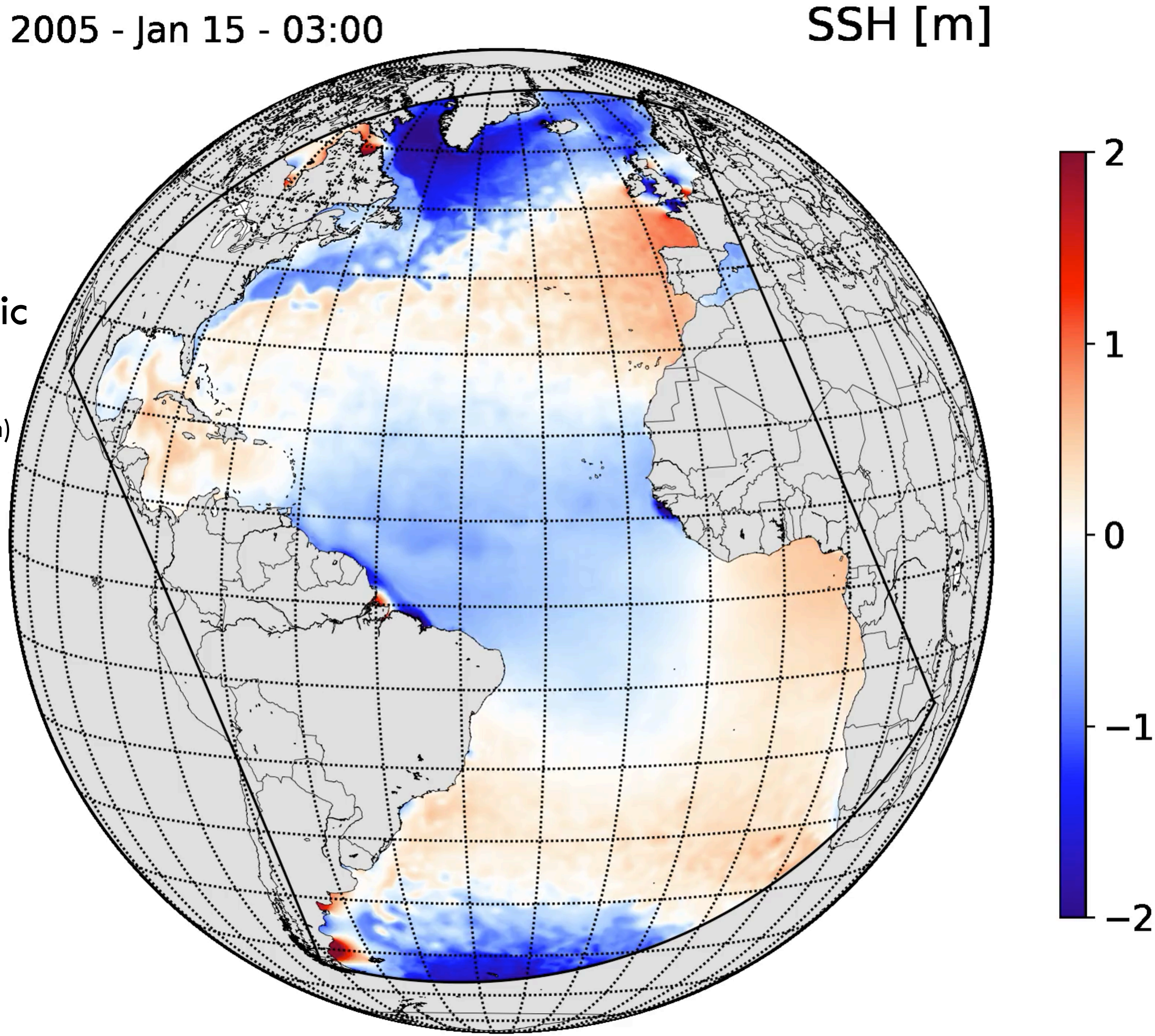
Realistic Modelling:



Realistic Modelling:

Forced Atlantic simulations:

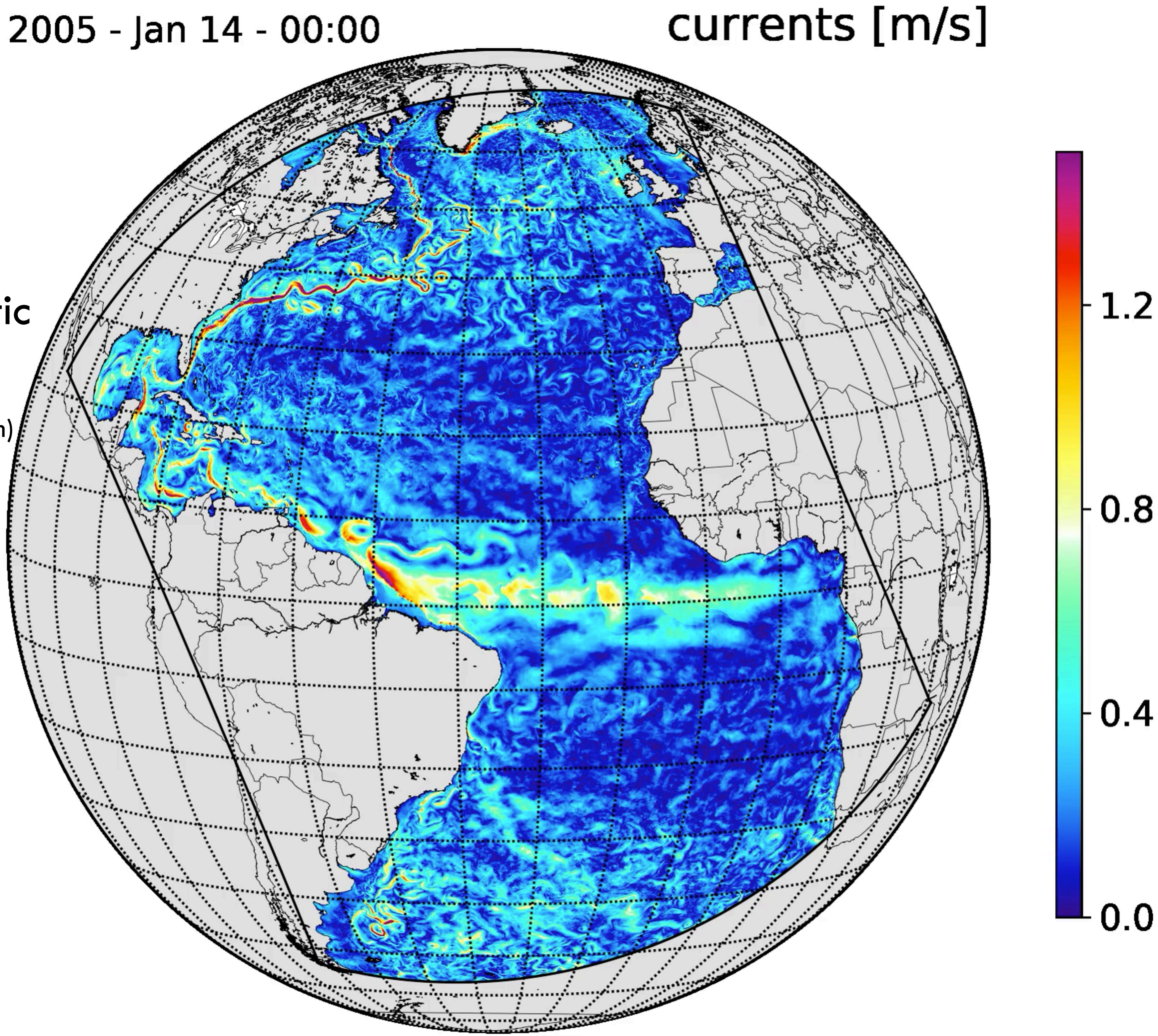
- oceanic model (3 km)



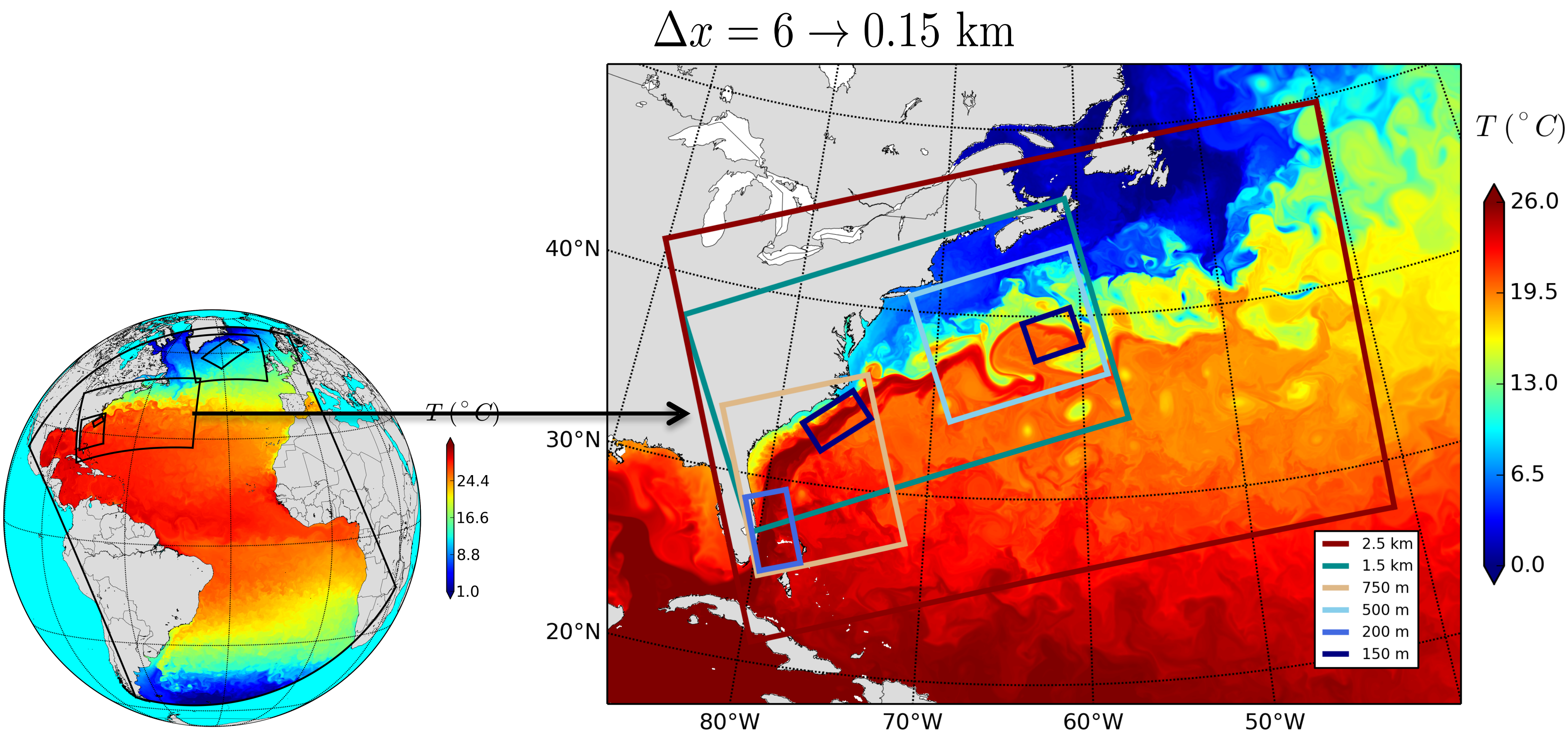
Realistic Modelling:

Forced Atlantic simulations:

- oceanic model (3 km)

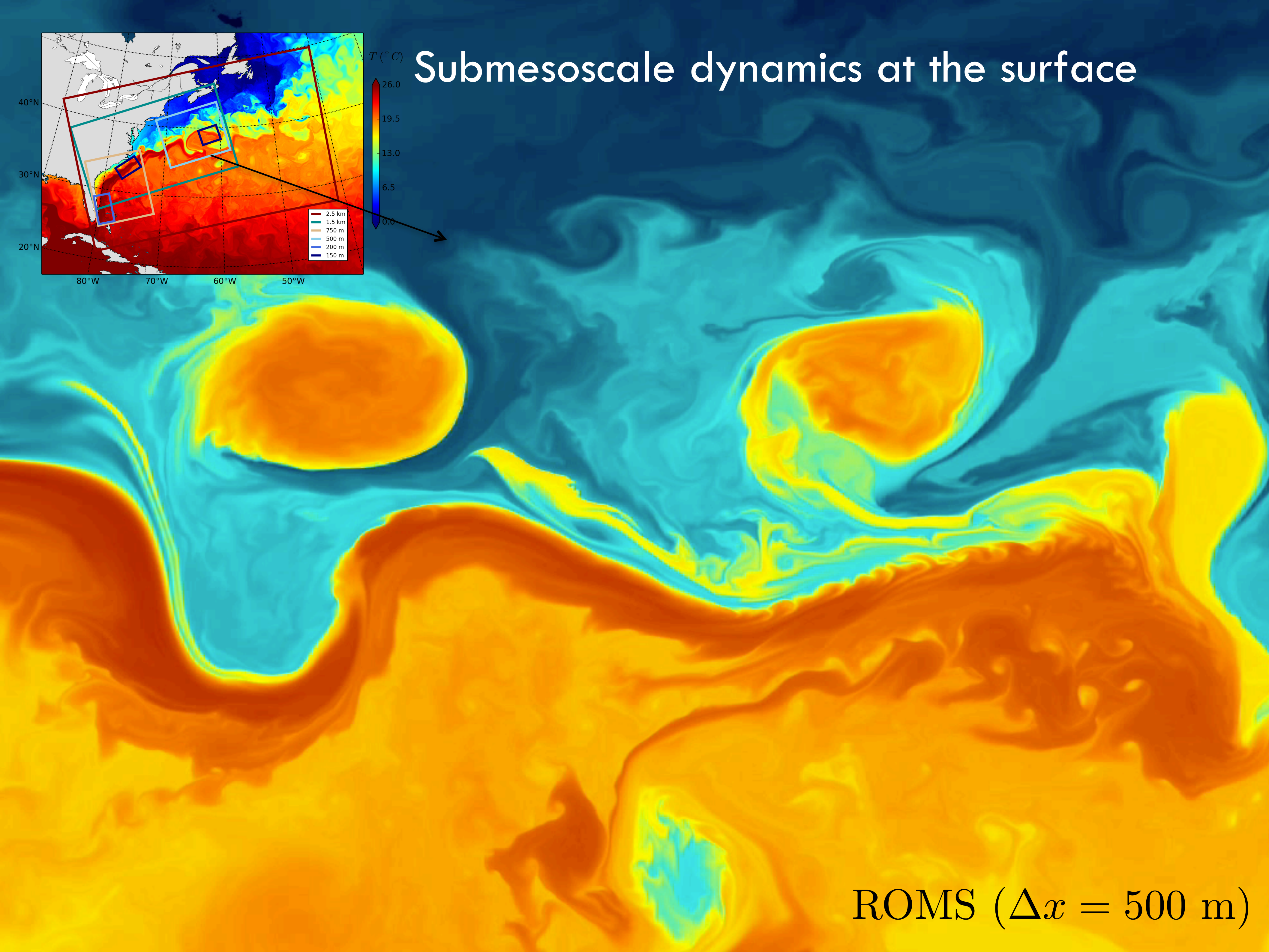


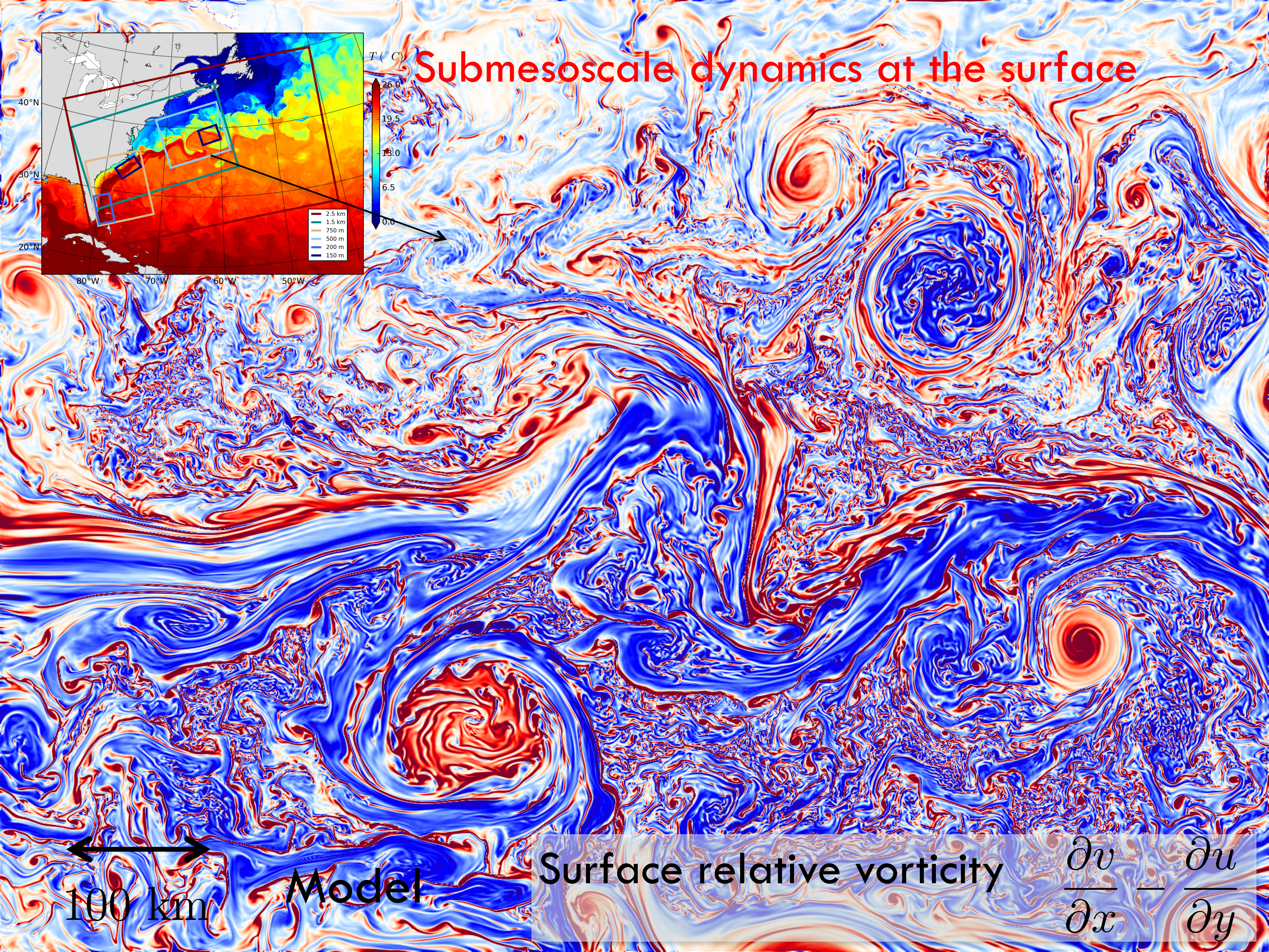
… Down to submesoscales (< 10 km)

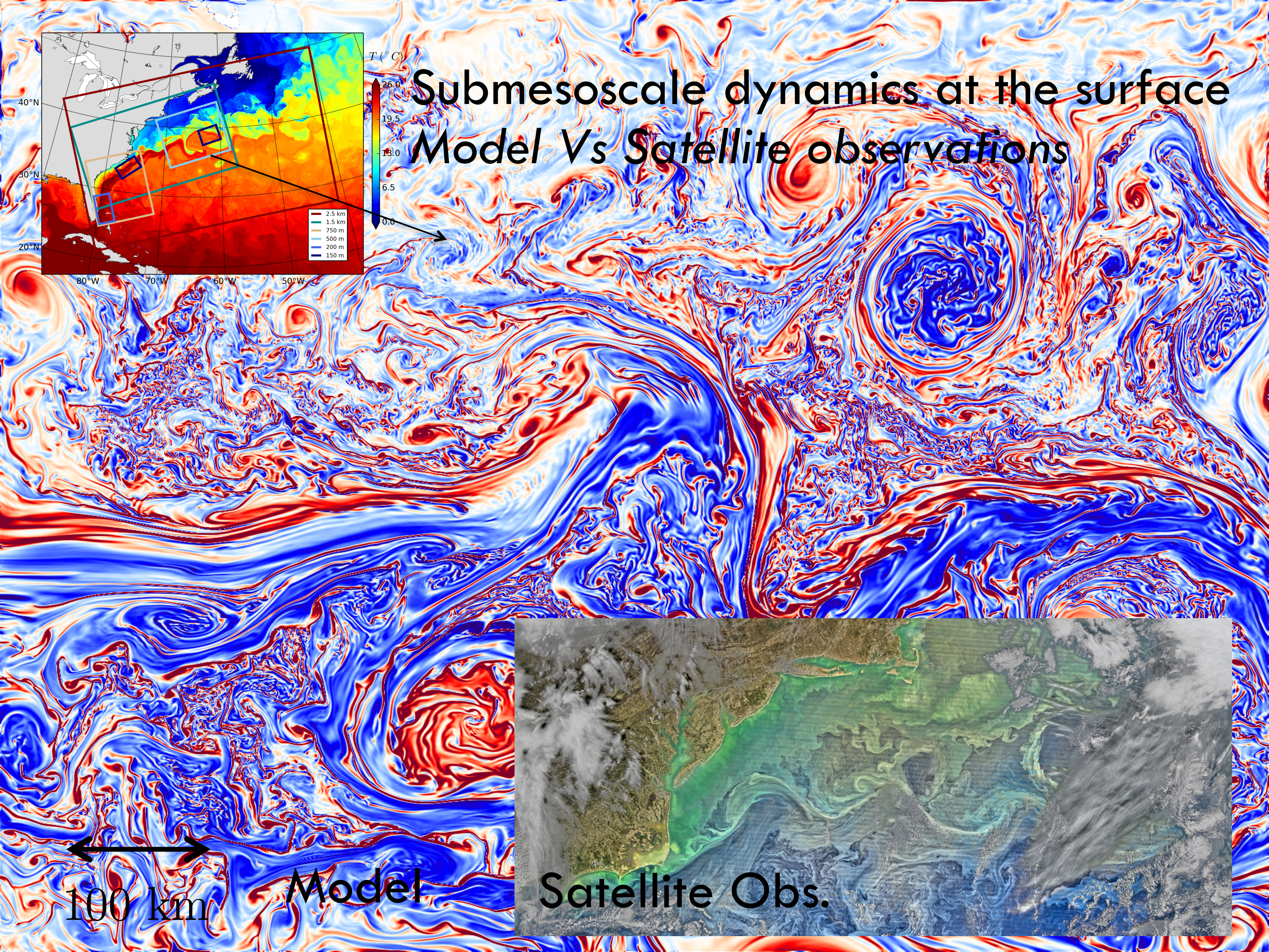


A portion of the Atlantic domain showing mean SST and several (1-way) nested grids:

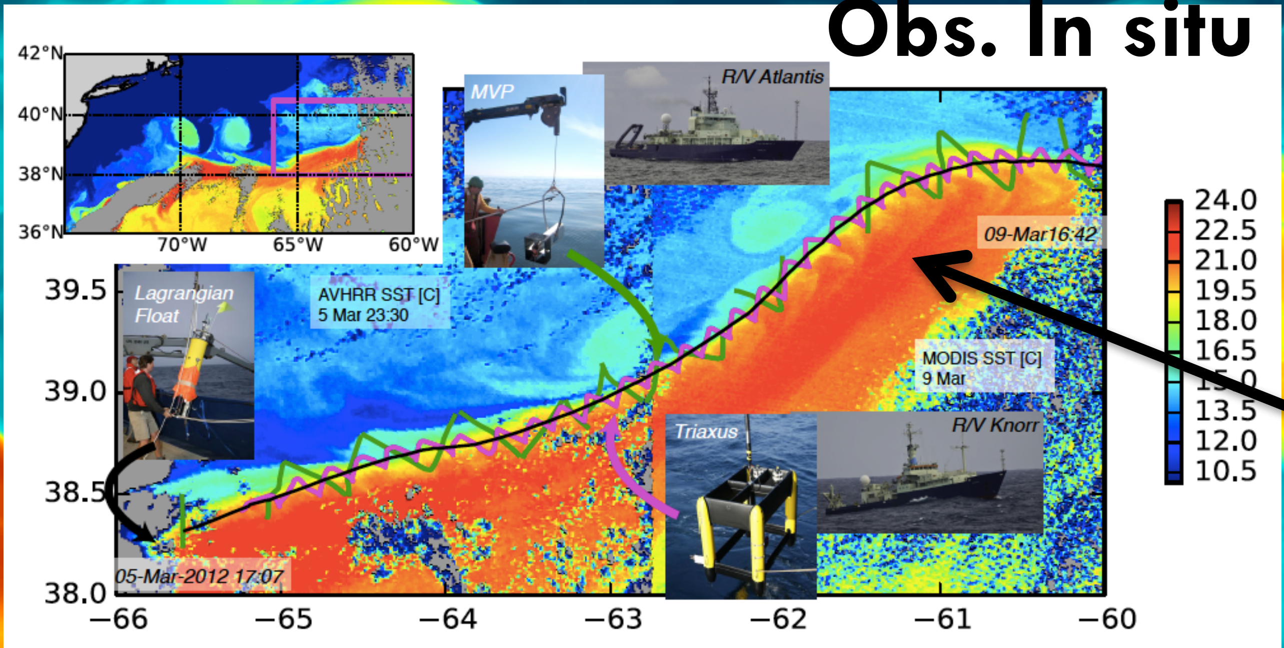
Submesoscale dynamics at the surface



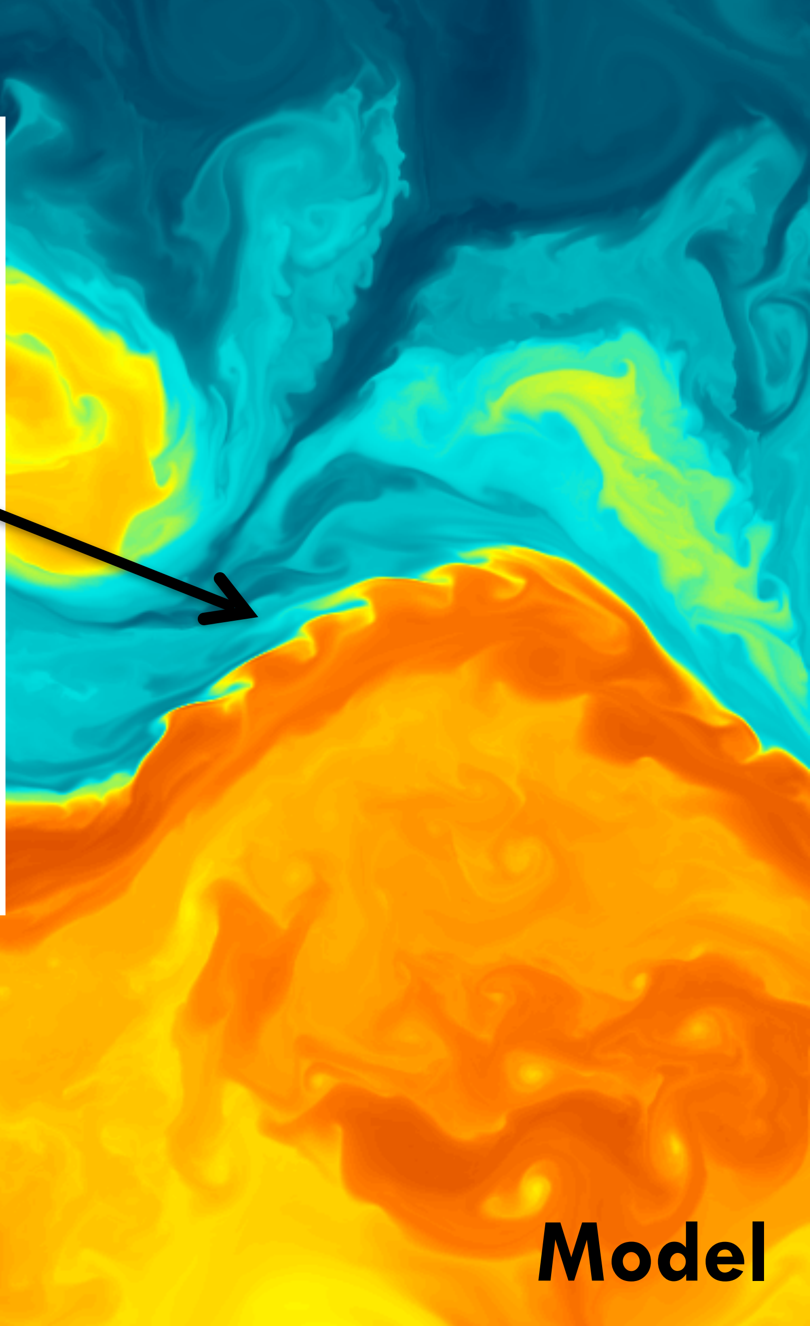

Submesoscale dynamics at the surface
$T(^\circ C)$
26.0
19.5
13.0
6.5
0.0
40°N
30°N
20°N
80°W
70°W
60°W
50°W
2.5 km
1.5 km
750 m
500 m
200 m
150 m
100 km
Model
Surface relative vorticity $\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$


Submesoscale dynamics at the surface
Model Vs Satellite observations
T(°C)
26.0
19.5
13.0
6.5
0.0
40°N
30°N
20°N
80°W
70°W
60°W
50°W
2.5 km
1.5 km
750 m
500 m
200 m
150 m
100 km
Model
Satellite Obs.

Submesoscale dynamics at the surface
Model Vs Satellite observations

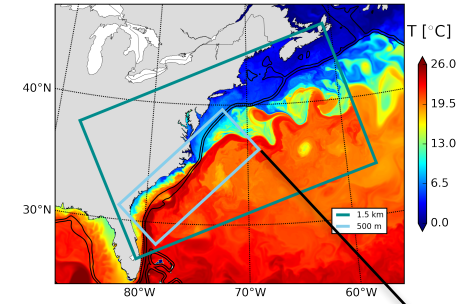


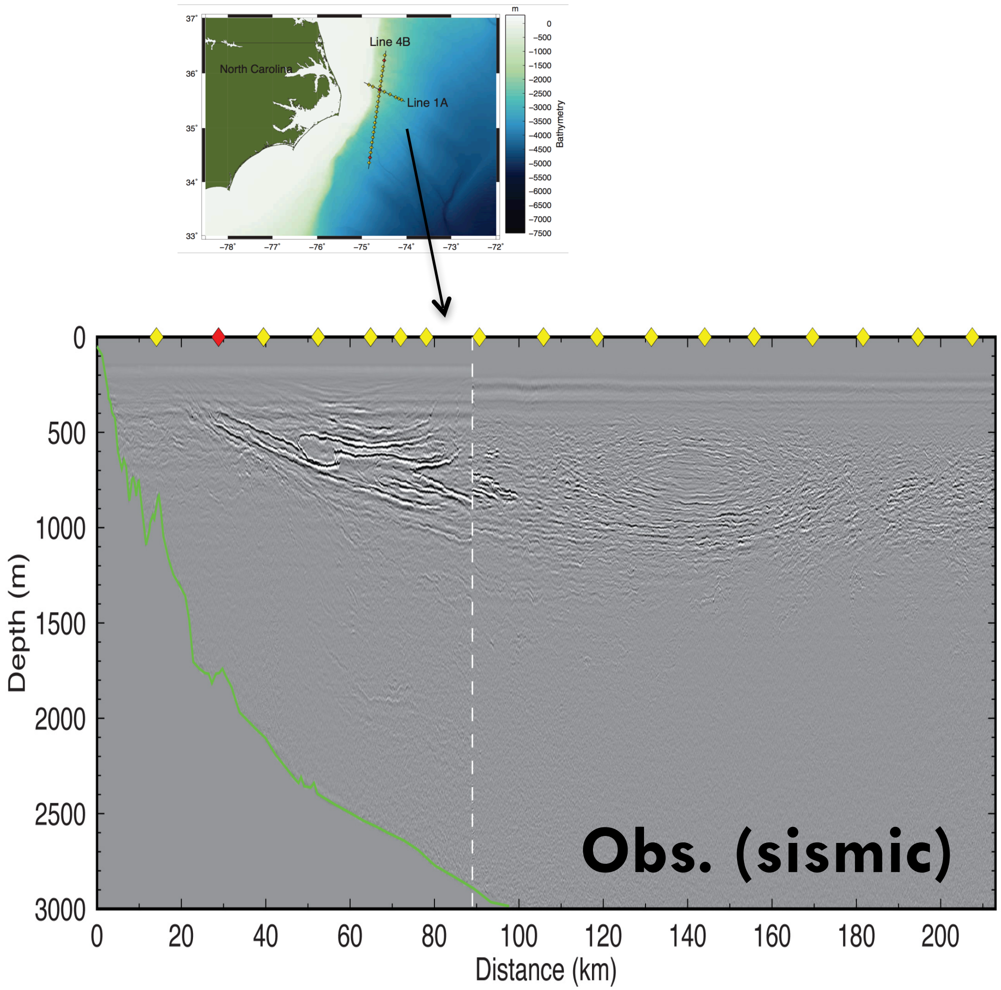
LATMIX 2012 Campaign
(Scalable Lateral Mixing and Coherent Turbulence)

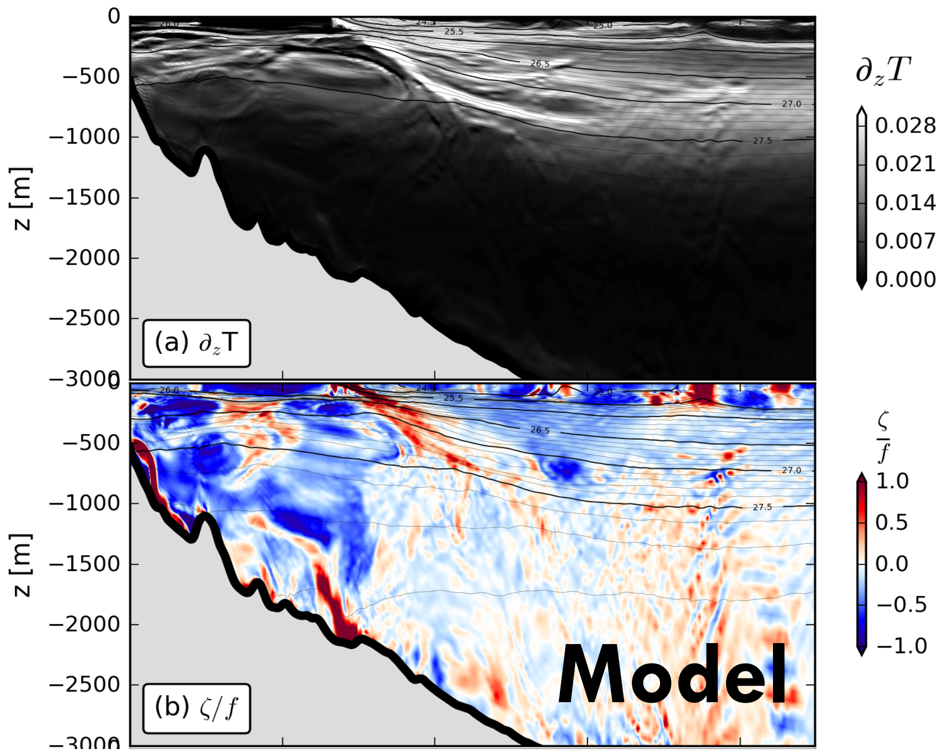


Submesoscale dynamics in the interior

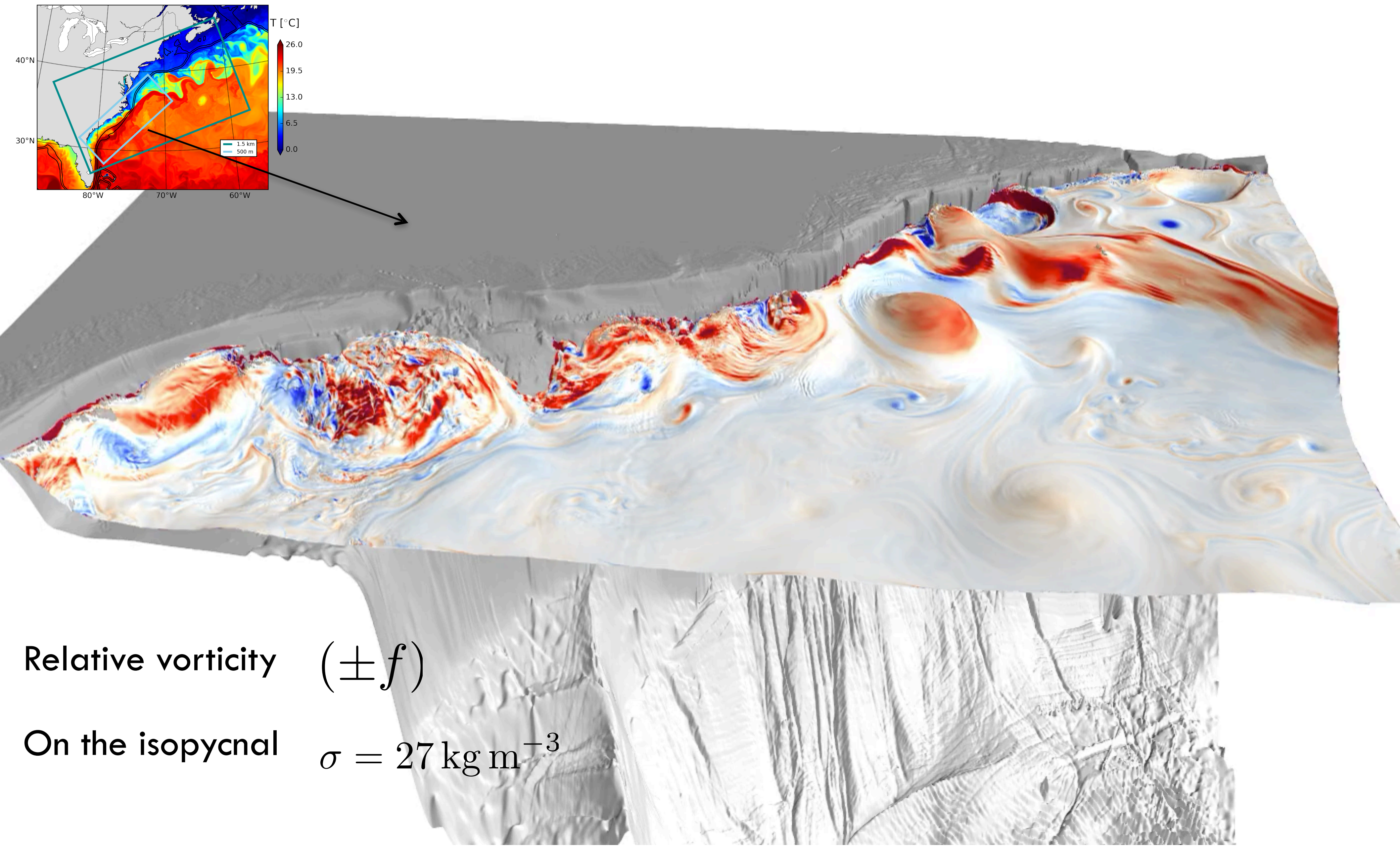
- Generation of submesoscale coherent vortices:







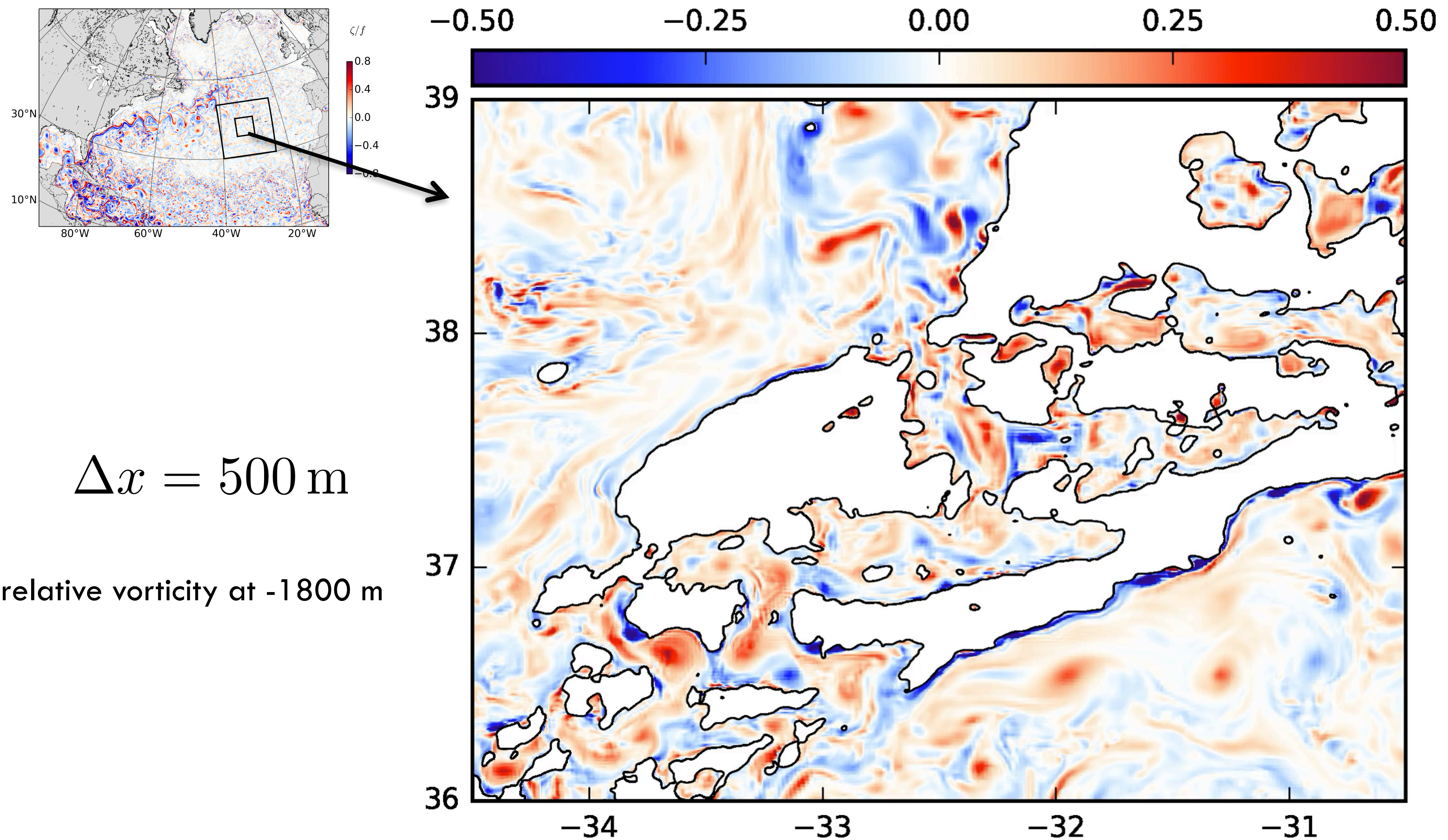
Submesoscale dynamics in the interior



Relative vorticity $(\pm f)$

On the isopycnal $\sigma = 27\,\mathrm{kg\,m^{-3}}$

Even the deep abyssal ocean is turbulent!

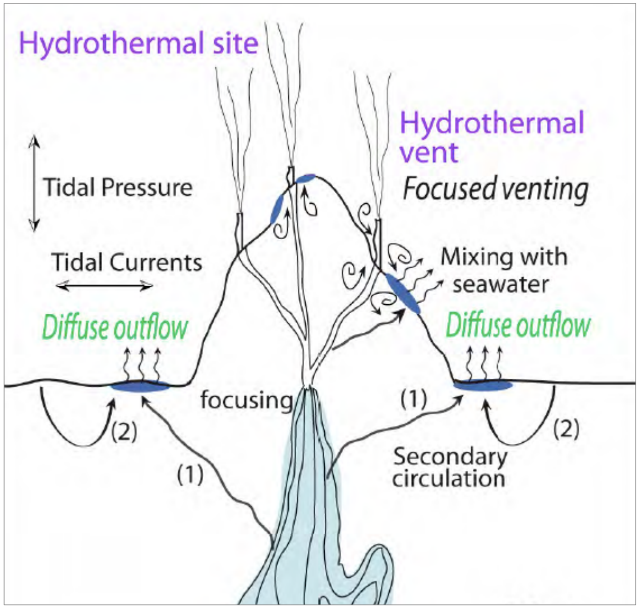


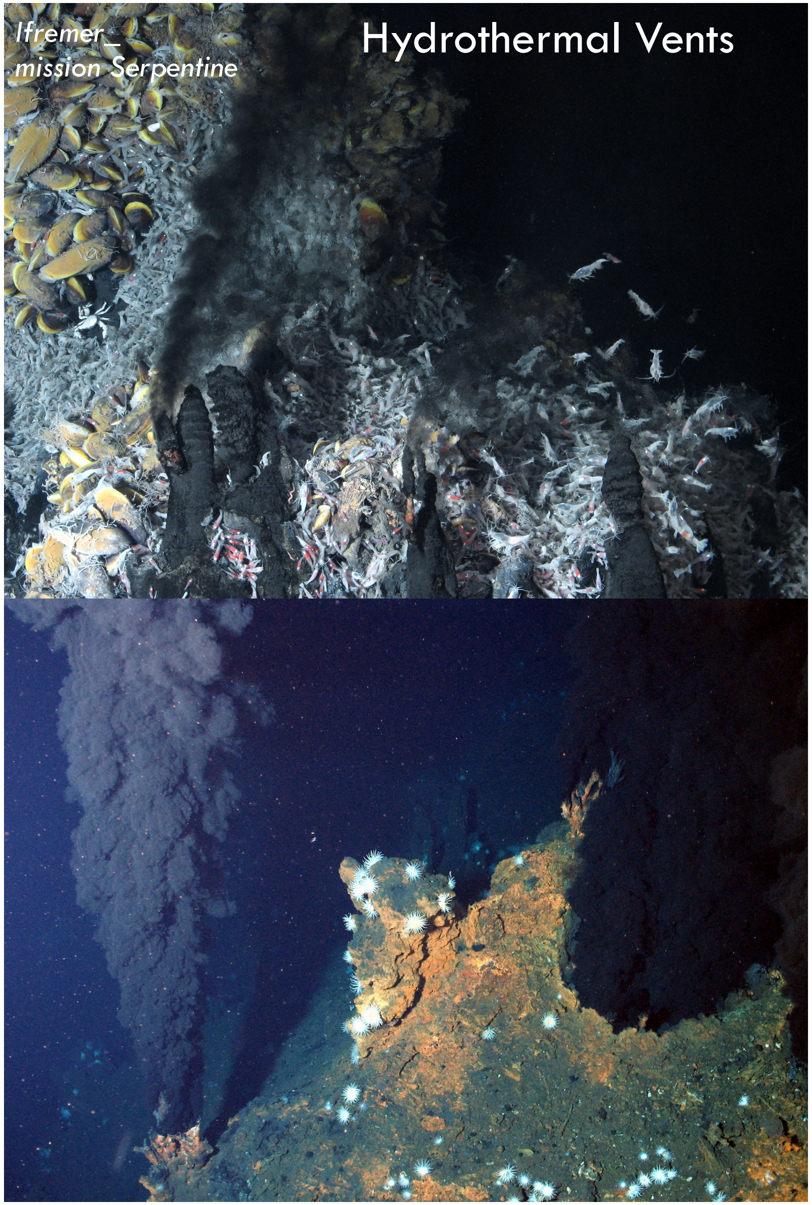
$$\Delta x = 500\,\mathrm{m}$$

relative vorticity at -1800 m

And the abyssal turbulence impacts a lot of things:

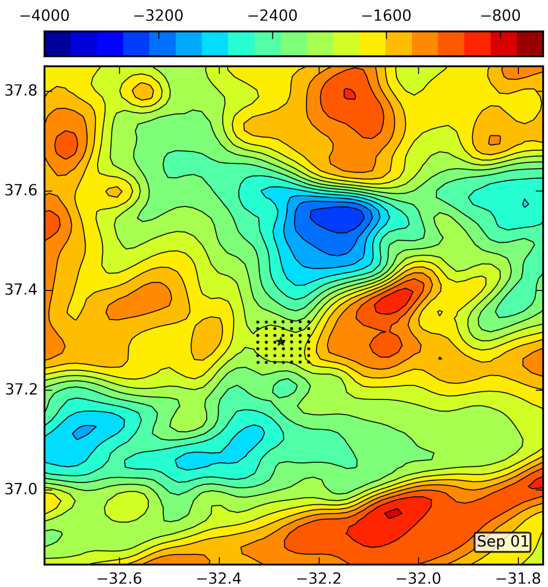
- *Dispersion of hydrothermal effluents*
- *Transport of biogeochemical tracers*
- *Connectivity between deep ecosystems*

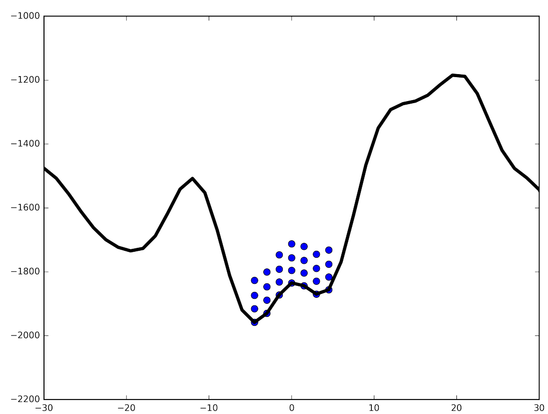


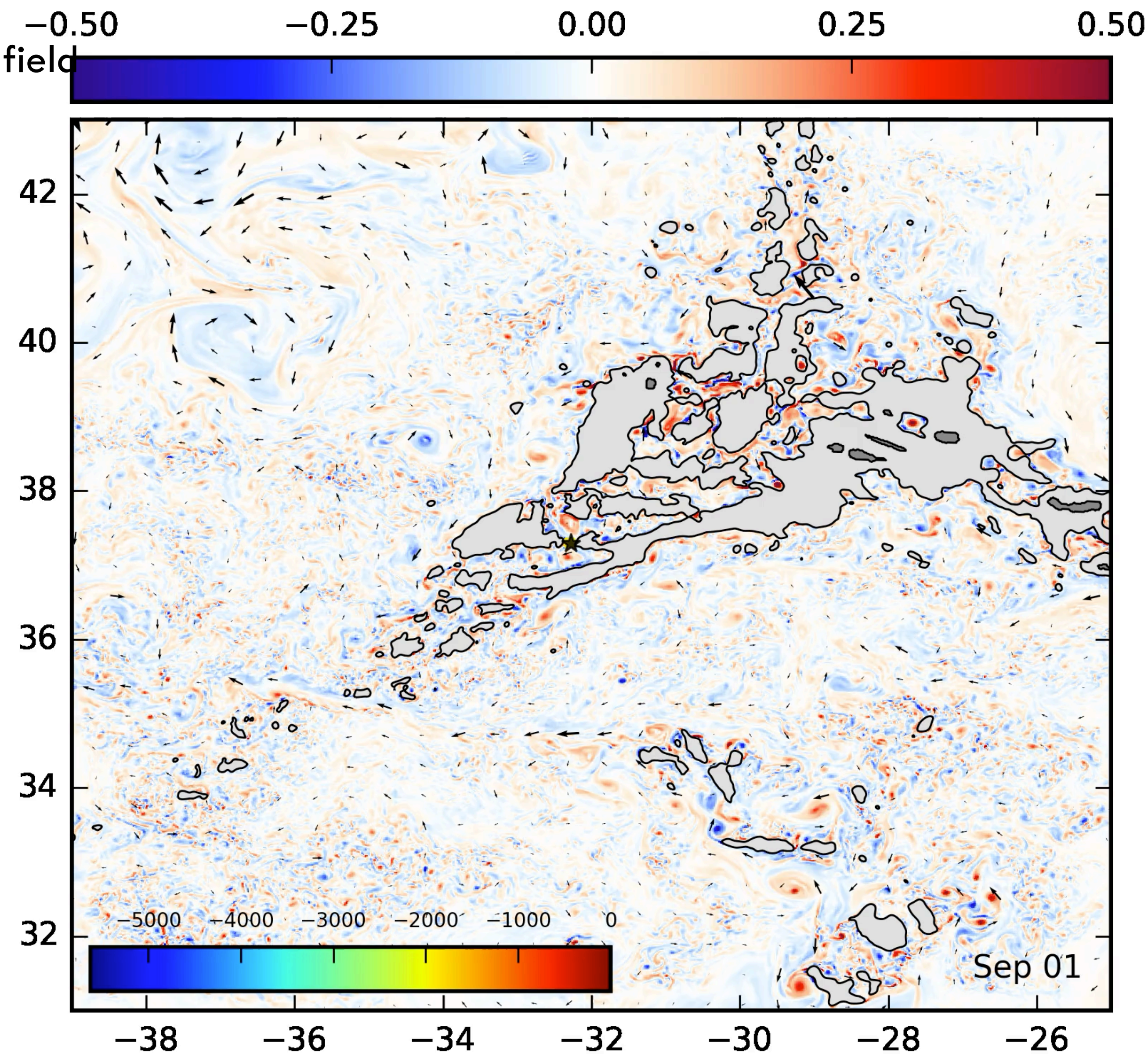


Dispersion of larvae by abyssal turbulence

Continuous release of Lagrangian particles near the Lucky Strike vent field (1.5 km res)







Dispersion of larvae by abyssal turbulence

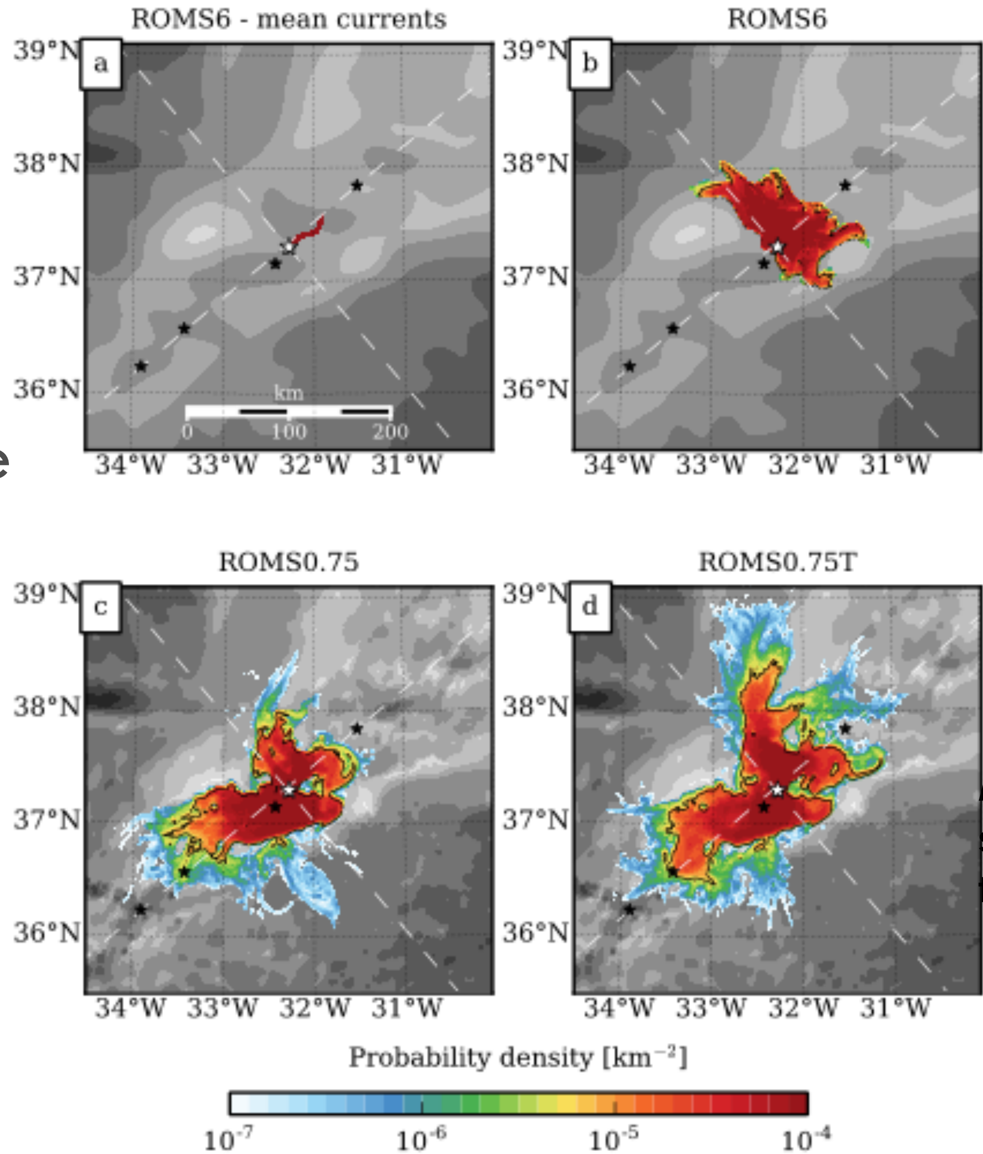
Larvae dispersion from the Lucky Strike vent after 30 days.

Mean currents only

Mesoscale

Mesoscale + submesoscale

Mesoscale + submesoscale + tides



Ocean Circulation Models

- ROMS ➔ CROCO https://www.croco-ocean.org/
- NEMO http://www.nemo-ocean.eu/
- MITgcm http://mitgcm.org/
- HYCOM http://hycom.org/
- POP http://www.cesm.ucar.edu/models/cesm1.0/pop2/
- OFES http://www.jamstec.go.jp/esc/ofes/eng/
- MOM http://www.gfdl.noaa.gov/ocean-model
- POM http://www.ccpo.odu.edu/POMWEB/
- etc

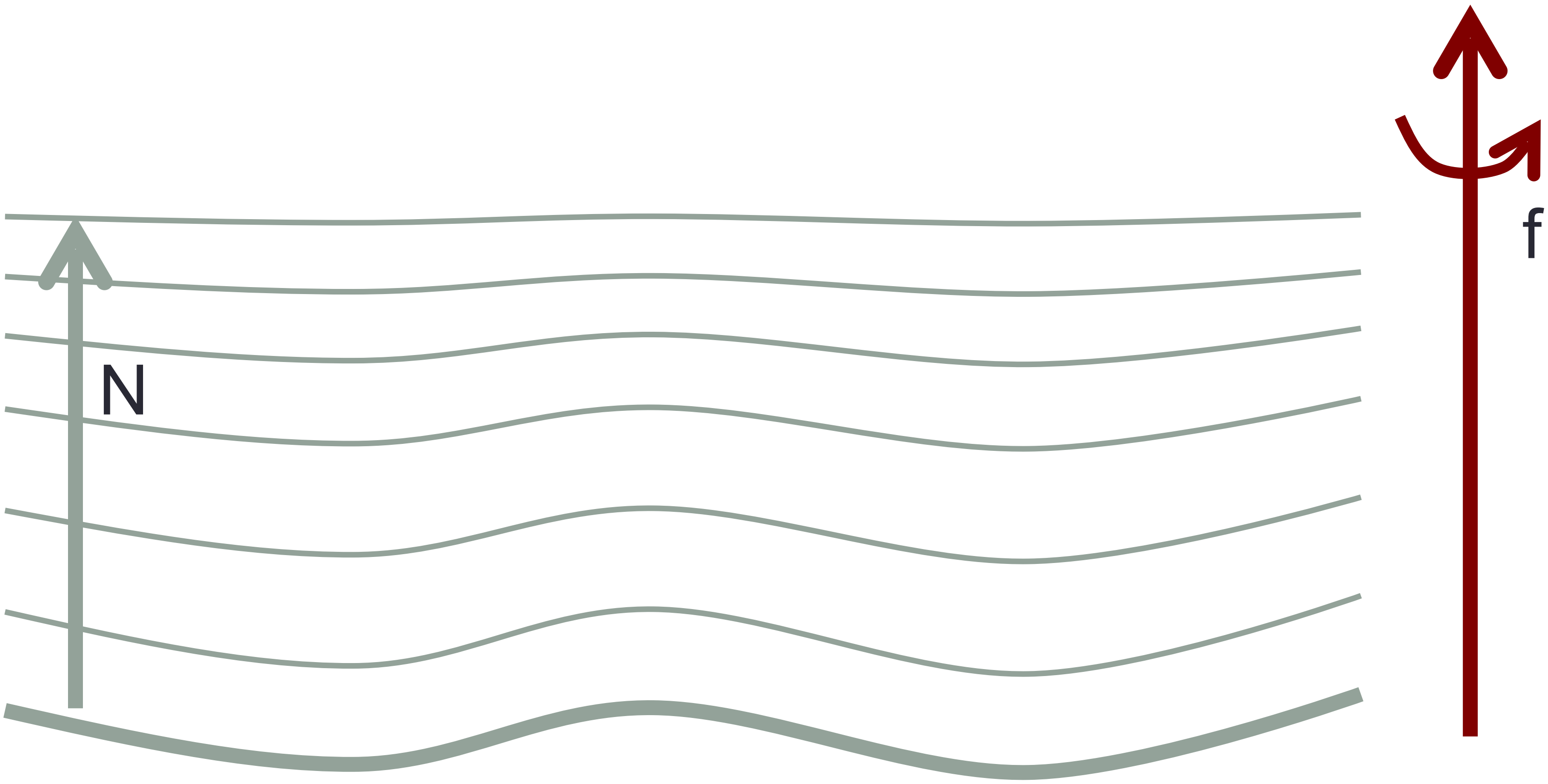
#1
Which Equations?

Master's degree 2^{nd} year Marine Physics

Jonathan GULA
gula@univ-brest.fr

Which Equations?

Ingredients : rotation + stratification



Which Equations?

- Momentum equations (3d) $$\frac{D\vec{u}}{Dt} = \ldots$$
- Conservation of mass $$\frac{D\rho}{Dt} = \mathcal{S}_\rho$$
- 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?

- Momentum equations (3d) $$\frac{D\vec{u}}{Dt} = \ldots$$
- Conservation of mass $$\frac{D\rho}{Dt} = \mathcal{S}_\rho$$
- 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]

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.

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:

$$\begin{aligned} \rho\vec{u} \quad &\rightarrow \quad \rho_0\vec{u} \\ \rho g \quad &\rightarrow \quad \rho g \end{aligned}$$

Equations for momentum/mass?

Boussinesq Approximation :

[+ incompressibility or adiabatic]

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

$$\Big\downarrow \quad \text{Continuity equation}$$

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

Equations for momentum/mass?

Boussinesq Approximation:

Note: EOS needs to be modified for consistency

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

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

$$\frac{\partial w}{\partial t} + \vec{u} \cdot \vec{\nabla} w + 2\Omega \cos\phi v + \boxed{\frac{\rho}{\rho_0} g = -\frac{\partial_z P}{\rho_0}} + \frac{\mathcal{F}_w}{\rho_0} + \frac{\mathcal{D}_w}{\rho_0}$$

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

H ~3000 m
L ~3000 km

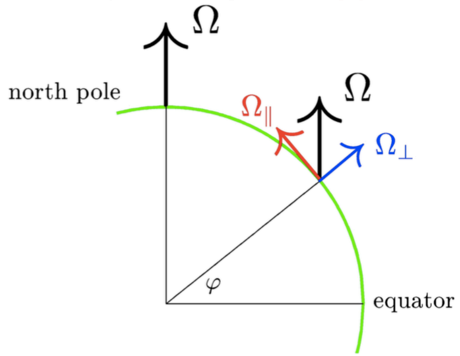
$$\frac{\partial P}{\partial z} = -\rho g$$

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 $$\frac{DT}{Dt} = \boxed{\mathcal{S}_T} \qquad \frac{DS}{Dt} = \boxed{\mathcal{S}_S}$$
- 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.)

#2
Subgrid-scale parameterization

Master's degree 2nd year Marine Physics

Jonathan GULA
gula@univ-brest.fr

Incompressible Navier-Stokes Equations:

- Dissipation of energy/momentum in the NS equations:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \vec{\nabla} \vec{u} + f \vec{k} \times \vec{u} + g \vec{k} = -\frac{\vec{\nabla} P}{\rho} + \boxed{\nu \vec{\nabla}^2 \vec{u}}$$

Viscosity

Type of models

Navier Stokes

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

CFD

Process studies

PE

- PE = Primitive Equations models

SW

- SW = Shallow-Water models

Ocean Circulation Models

SQG

- SQG = Surface Quasi-Geostrophic models

QG

- QG = Quasi-Geostrophic models
- Etc.

Idealized models

Incompressible Navier-Stokes Equations:

$$\frac{\partial \vec{u}}{\partial t} + \boxed{\vec{u} \cdot \vec{\nabla} \vec{u}} + f\vec{k} \times \vec{u} + g\vec{k} = -\frac{\vec{\nabla} P}{\rho} + \boxed{\nu \vec{\nabla}^2 \vec{u}}$$

Non-linear terms

Viscosity

- Importance of NL terms and viscosity = Reynolds Number

$$Re = \frac{UL}{\nu}$$

Where U is a typical velocity of the flow and L is a typical length describing the flow.

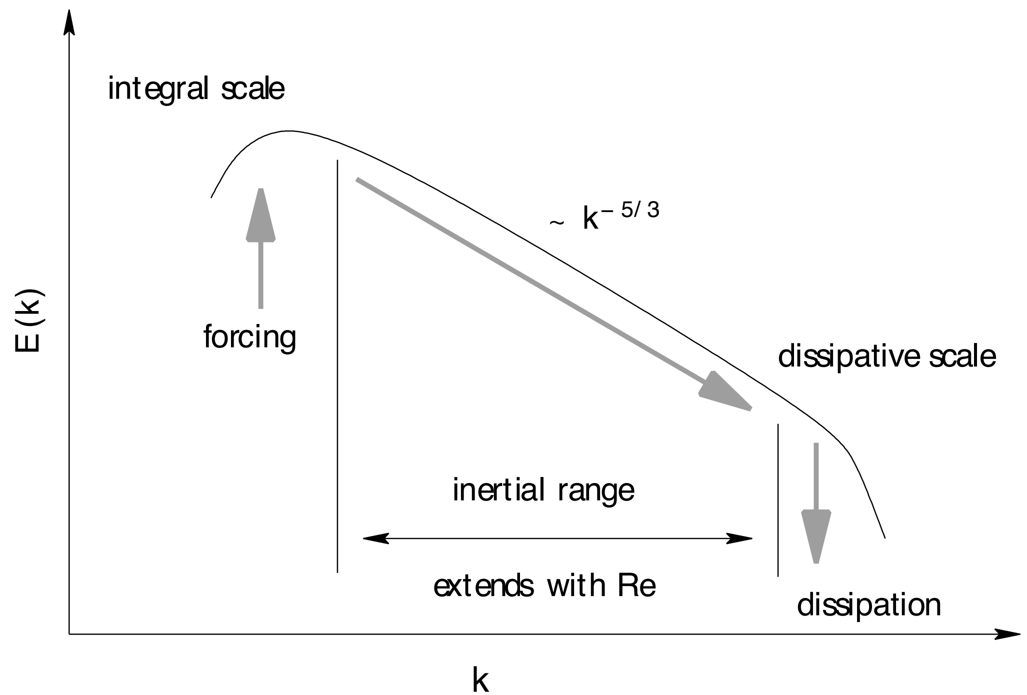
Direct numerical simulation (DNS)

DNS resolves the entire range of turbulent length scales down to the smallest dissipative scales (Kolmogorov scale):

$$\eta = \left(\frac{\nu^3}{\epsilon}\right)^{1/4} \approx \left(\frac{\nu^3 L}{U^3}\right)^{1/4} = Re^{-3/4} L$$

where ν is the kinematic viscosity

And ϵ the rate of kinetic energy dissipation



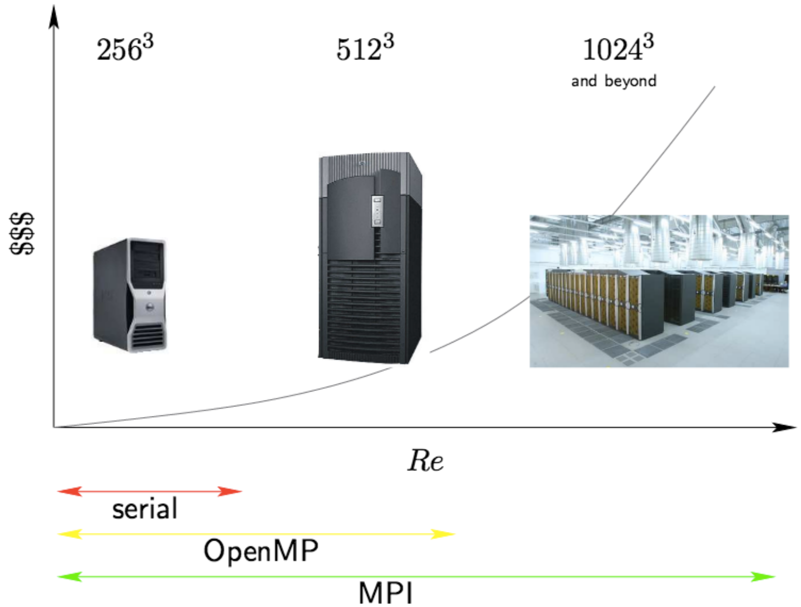
Direct numerical simulation (DNS)

The number of floating-point operations required to complete the simulation is proportional to the number of mesh points:

$$N_x = \frac{L}{\eta} = Re^{3/4}$$

and the number of time steps: $$\frac{T}{\Delta t} = \frac{TU}{\eta} = \frac{TU}{L} Re^{3/4}$$

It is extremely expensive as the computational cost scales as Re^3



Type of models

Navier Stokes

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

CFD

Process studies

PE

SW

SQG

QG

- PE = Primitive Equations models
- SW = Shallow-Water models
- SQG = Surface Quasi-Geostrophic models
- QG = Quasi-Geostrophic models
- Etc.

Ocean Circulation Models

Idealized models

Reynolds averaging

To reduce the computational cost, one need to reduce the range of time- and length-scales that are being solved for.

The idea is based on separation of mean and turbulent component:

$$u = \overline{u} + u'$$

Where

$$\overline{u} = \frac{1}{T}\int_0^T u\;dt \text{ or } \overline{u} = \frac{1}{X}\int_0^X u\;dx$$

With by definition

$$\overline{u'} = 0$$

Reynolds averaging

- Activity:

Adapt the momentum equation:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \vec{\nabla} \vec{u} + f\vec{k} \times \vec{u} + g\vec{k} = -\frac{\vec{\nabla} P}{\rho} + \nu \vec{\nabla}^2 \vec{u}$$

For the mean velocity: $\frac{\partial \overline{u_i}}{\partial t}$ = ?

Reynolds averaging

So we resolve only the equations for the mean variables:

$$\frac{\partial \overline{u_i}}{\partial t} + \overline{u_j \frac{\partial u_i}{\partial x_j}} + f\mathbf{k} \times \overline{u_i} + \frac{\overline{\rho}}{\rho_0} g\mathbf{k} = -\frac{1}{\rho_0}\frac{\partial \overline{P}}{\partial x_j} + \nu \frac{\partial^2 \overline{u_i}}{\partial x_i \partial x_j}$$

$$\frac{\partial \overline{u_i}}{\partial t} + \overline{u_j}\frac{\partial \overline{u_i}}{\partial x_j} + f\mathbf{k} \times \overline{u_i} + \frac{\overline{\rho}}{\rho_0} g\mathbf{k} = -\frac{1}{\rho_0}\frac{\partial \overline{P}}{\partial x_j} + \nu \frac{\partial^2 \overline{u_i}}{\partial x_i \partial x_j} \boxed{- \frac{\partial \overline{u_i' u_j'}}{\partial x_j}}$$

Advection for the averaged flow

Reynolds stress
= effect of subgrid-scale turbulence

Turbulence closure

The Closure Problem :

- The number of unknowns is larger than the number of equations
- When we start deriving equations for unknowns, more variables appear.

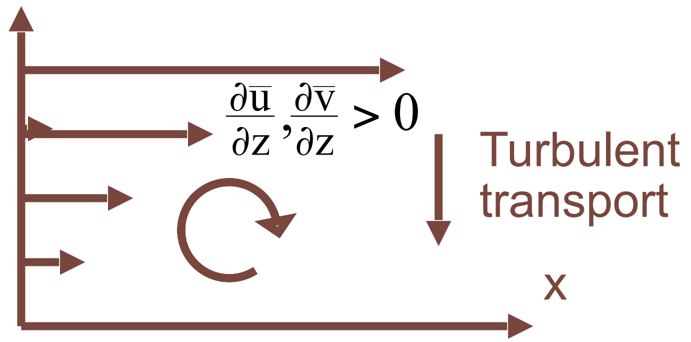
Progn. Eq. for:	Moment	Equation	No. Eqs.	No. Unknowns
$\overline{U_i}$	First	$\frac{\partial \overline{U_i}}{\partial t} = \ldots - \frac{\partial \overline{u'_i u'_j}}{\partial x_j}$	3	6
$\overline{u'_i u'_j}$	First	$\frac{\partial \overline{u'_i u'_j}}{\partial t} = \ldots - \frac{\partial \overline{u'_k u'_i u'_j}}{\partial x_k}$	6	10
$\overline{u'_i u'_j u'_k}$	First	$\frac{\partial \overline{u'_i u'_j u'_k}}{\partial t} = \ldots - \frac{\partial \overline{u'_k u'_i u'_j u'_m}}{\partial x_m}$	10	15

Turbulence closure

- In PE models the equations are closed by parameterizing the Reynolds stresses as:

$$\overline{u'w'} = -K_{Mv}\frac{\partial u}{\partial z}$$

$$\overline{v'w'} = -K_{Mv}\frac{\partial v}{\partial z}$$



Turbulence closure

In ROMS:

$$
\begin{aligned}
\mathcal{F}_u &= \frac{\partial}{\partial z}\left(K_{Mv}\frac{\partial u}{\partial z}\right) + \nabla_h(K_{Mh}.\nabla_h u) \\
\mathcal{F}_v &= \frac{\partial}{\partial z}\left(K_{Mv}\frac{\partial v}{\partial z}\right) + \nabla_h(K_{Mh}.\nabla_h v) \\
\mathcal{S}_T &= \frac{\partial}{\partial z}\left(K_{Tv}\frac{\partial T}{\partial z}\right) + \nabla_h(K_{Th}.\nabla_h T) \\
\mathcal{S}_S &= \frac{\partial}{\partial z}\left(K_{Sv}\frac{\partial S}{\partial z}\right) + \nabla_h(K_{Sh}.\nabla_h S)
\end{aligned}
$$

Vertical mixing

Horizontal diffusion

Turbulence closure

Vertical Mixing:

$$K_{Mv}, K_{Tv}, K_{Sv}$$

- Local Turbulence closure: GLS, k-kl (MY2.5), k-ε, κ-ω, etc. *[e.g. Warner et al, 2005, Ocean Modelling]*
- Non local K-profile parameterization (KPP) *[Large et al, 1994, Rev. of Geophysics]*

Horizontal diffusion:

$$K_{Mh}, K_{Th}, K_{Sh}$$

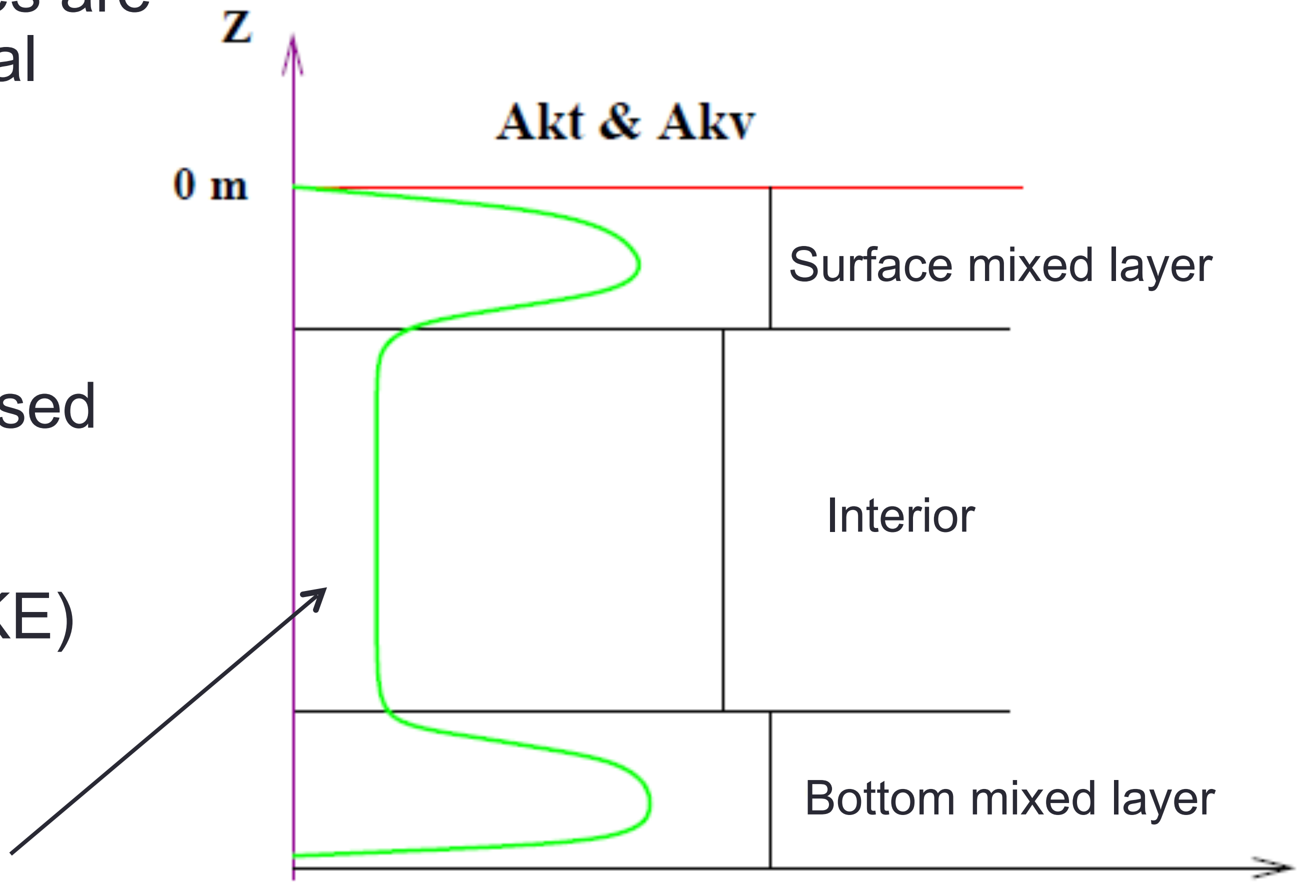
- Explicit diffusion
- Implicit (comes from the advective scheme)

Non local K-profile parameterization

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

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

Mixing coefficient profils



Principle scheme of KPP turbulent closure

Boundary conditions

- **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

- **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

Bottom friction parametrization

1. Linear friction, with r ***friction velocities*** [m/s] → $$(\tau_b^x, \tau_b^y) = -r\ (u_b, v_b)$$

2. Quadratic friction, controled by a constant drag coefficient **Cd** → $$(\tau_b^x, \tau_b^y) = C_d\sqrt{u_b^2 + v_b^2}\ (u_b, v_b)$$

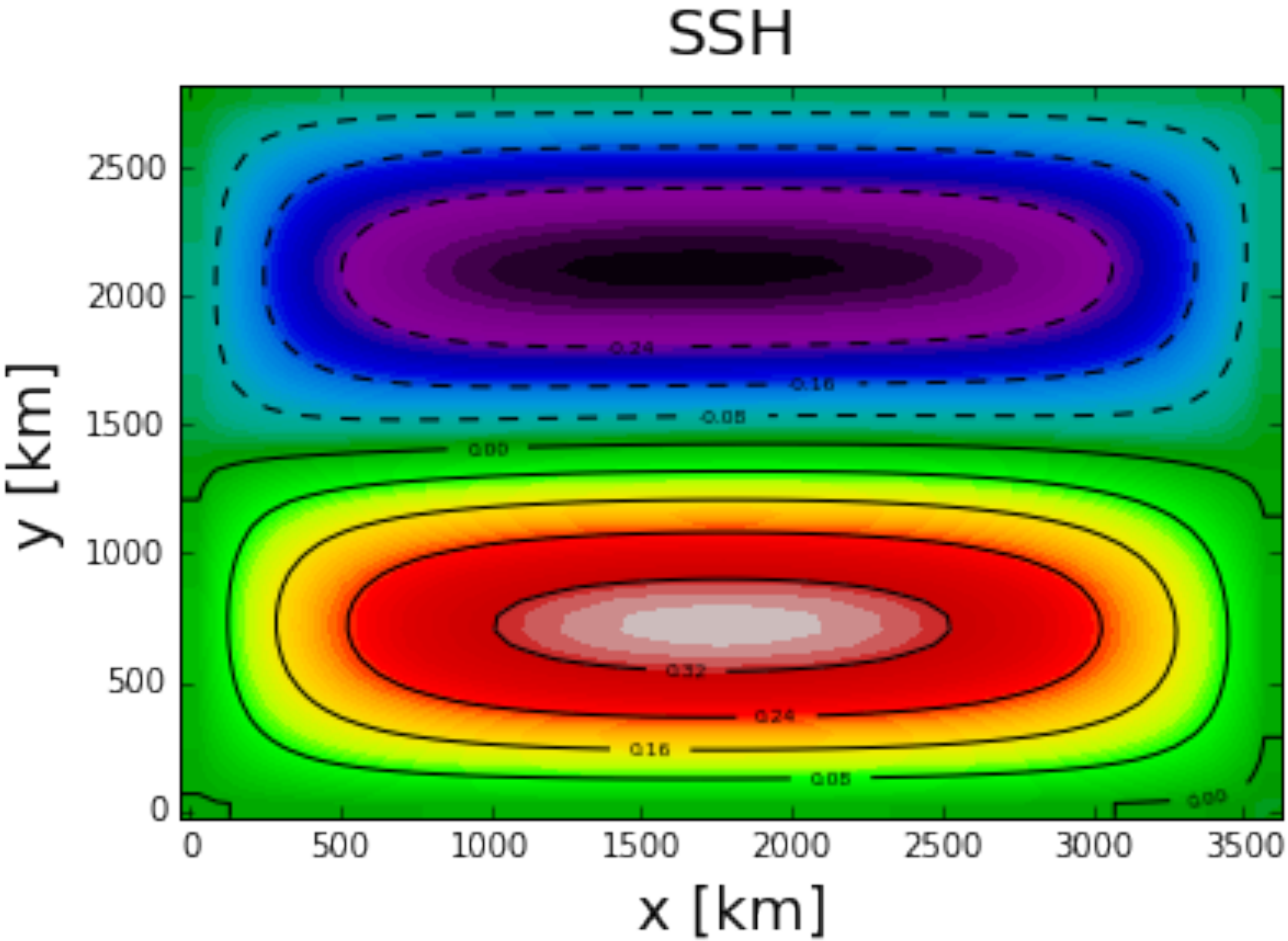
3. Quadratic friction coefficient, using variable **Cd** (Von Karman log. layer) →

$$\begin{cases} (\tau_b^x, \tau_b^y) = C_d\sqrt{u_b^2 + v_b^2}\ (u_b, v_b) \\ C_d = \left(\frac{\kappa}{\log[\Delta z_b / z_r]}\right)^2 \text{ si } C_d^{min} < C_d < C_d^{max} \\ C_d = C_d^{min} \text{ ou } C_d^{max} \\ \kappa = 0.41 \end{cases}$$

z_r = Rugosity scale

Δz_b = thickness of the first bottom level

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:

Given by running make_grid

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

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.
```