

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

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](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](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>

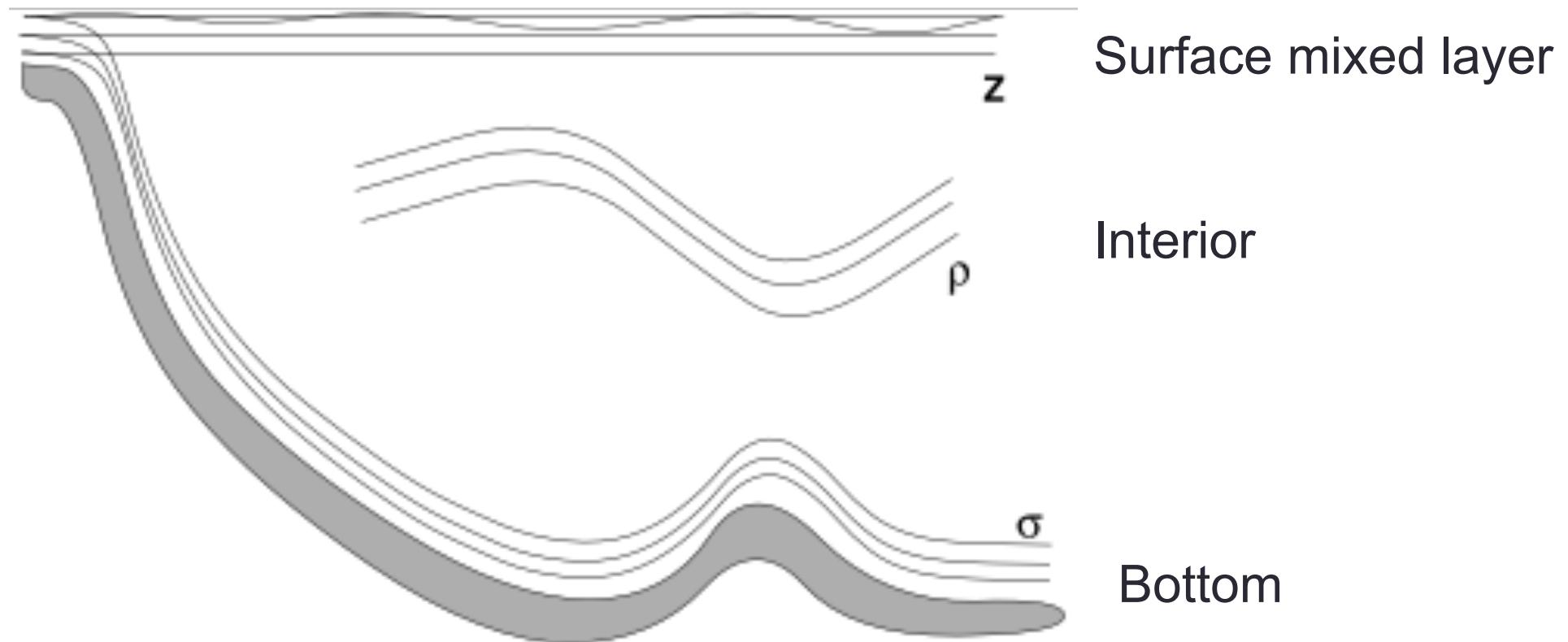
# #5 Vertical Discretization

Master's degree 2<sup>nd</sup> year Marine Physics

Jonathan GULA  
[gula@univ-brest.fr](mailto:gula@univ-brest.fr)

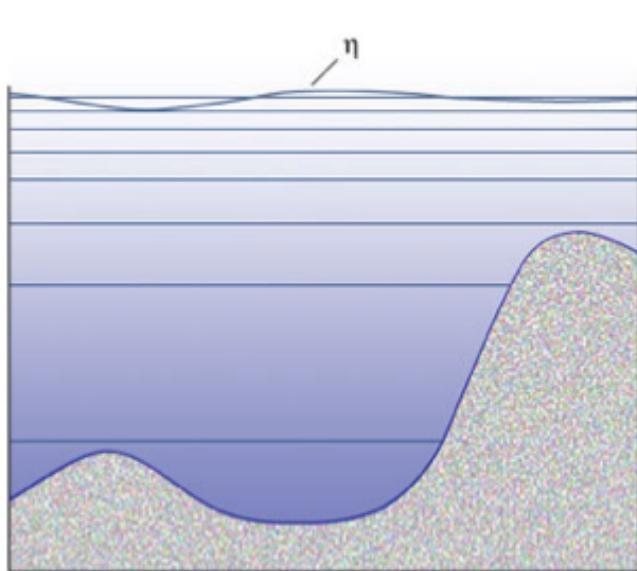
# Vertical discretization

Depending on the depth, motions in the ocean will be mostly aligned along **geopotentials, isopycnal or topography**.



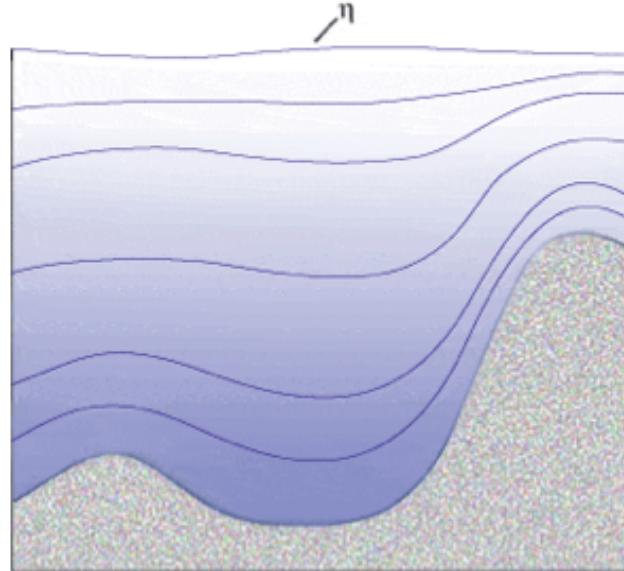
# Vertical discretization

Several choices are possible to define the vertical coordinates system:



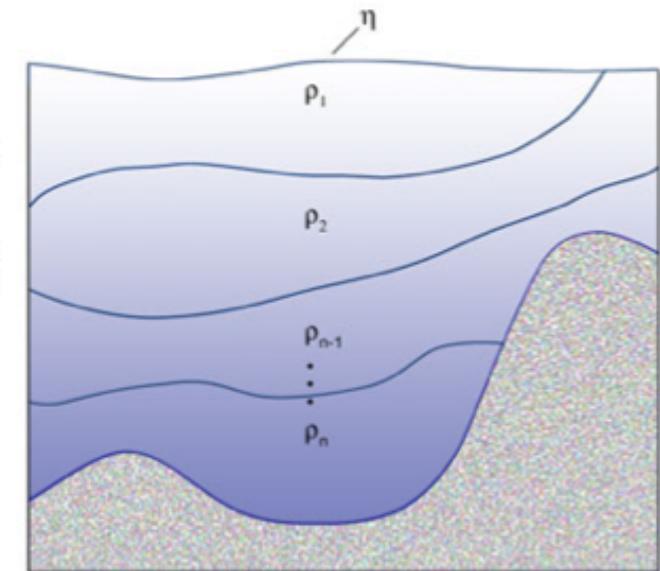
**z-coordinates**

Vertical coordinate is height (or depth)



**sigma-coordinates**

The vertical coordinate follows the bathymetry



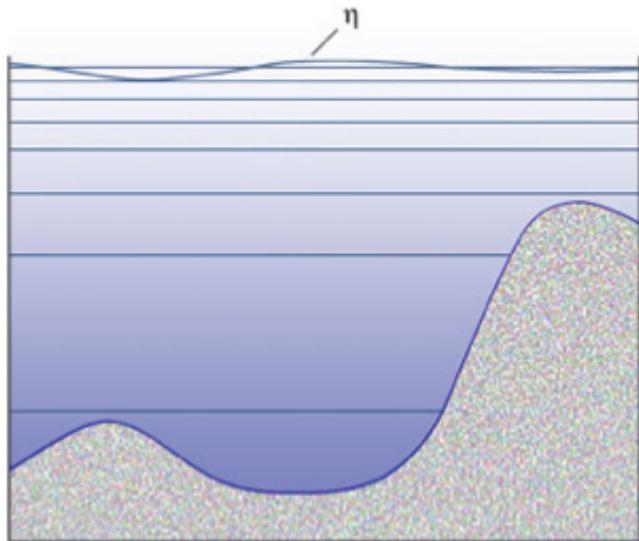
**isopycnal-coordinates**

The vertical coordinate is the potential density

The vertical coordinate is the major difference between models.

# Vertical discretization

## z-coordinates

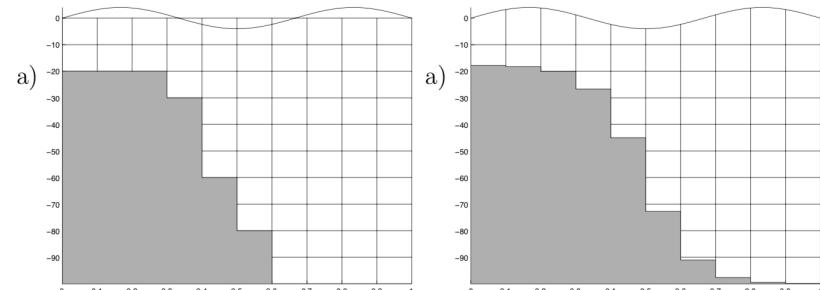


### PROS

- Natural in the upper ocean and for mixed-layer processes
- Ideal to compute horizontal (pressure) gradients
- Easier to implement and use

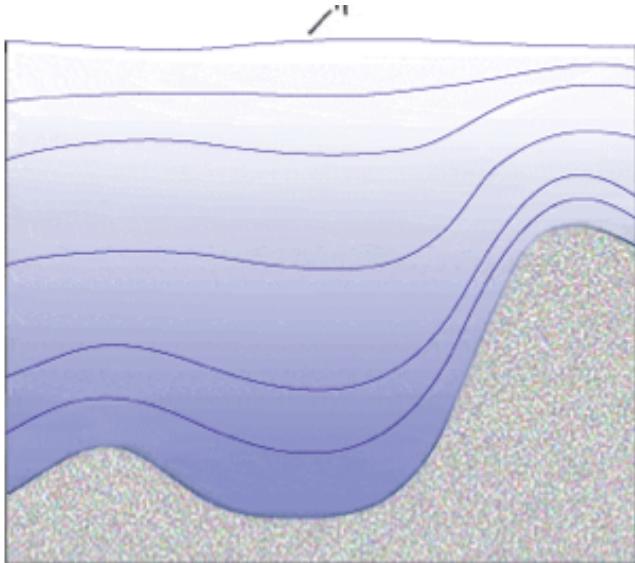
### CONS

- The representation of tracer advection and diffusion along inclined density surfaces in the ocean interior is cumbersome.
- Representation of bottom topography is difficult.
- Need for bottom and lateral conditions
- Representation and parameterization of the BBL is unnatural.



# Vertical discretization

## sigma-coordinates



### PROS

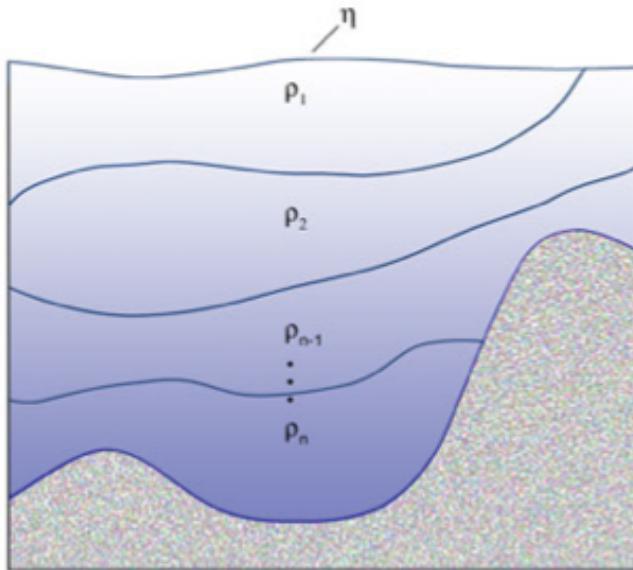
- Representation of bottom topography is natural (only bottom boundary condition)
- Representation and parameterization of the BBL is natural (more vertical res. in BBL)

### CONS

- The representation of tracer advection and diffusion along inclined density surfaces in the ocean interior is cumbersome.
- Pressure gradient errors can be a problem

# Vertical discretization

## isopycnal-coordinates



### PROS

- The representation of tracer advection and diffusion along inclined density surfaces in the ocean interior is natural.
- Water mass characteristics are preserved over long time scales

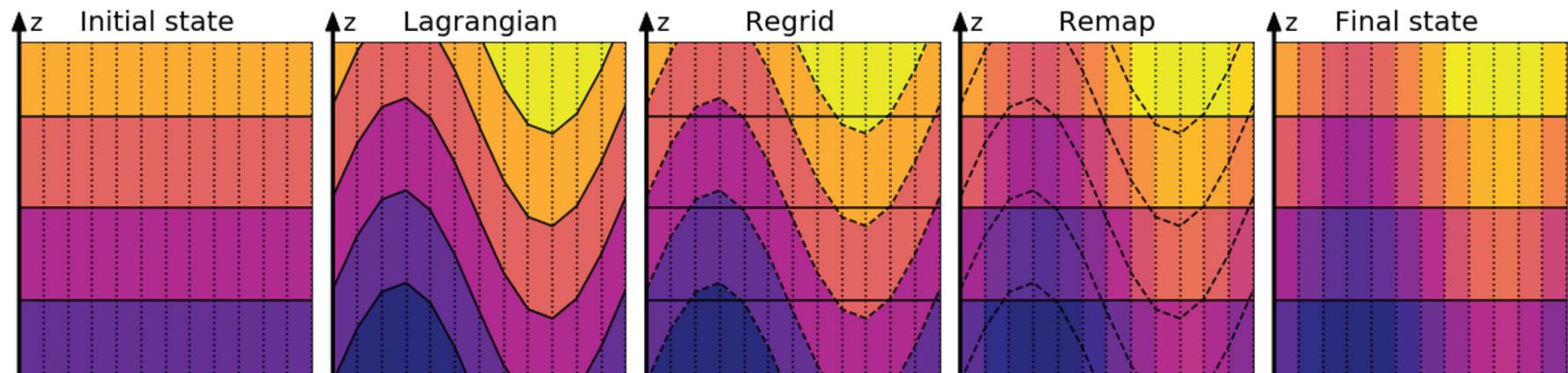
### CONS

- Representing the effects of a realistic (non-linear) equation of state is cumbersome.
- Inappropriate for representing the surface mixed layer or BBL which are mostly unstratified.
- Non-hydrostatic effects/dynamics are not possible.
- Vertical and horizontal resolution are tightly connected in regions where isopycnals outcrop. This can lead to inadequate horizontal resolution in regions such as the ACC.

# Vertical discretization

More recent developments use **Lagrangian vertical coordinates**:

- Arbitrary Lagrangian-Eulerian (ALE) method
- Lagrangian-remap method

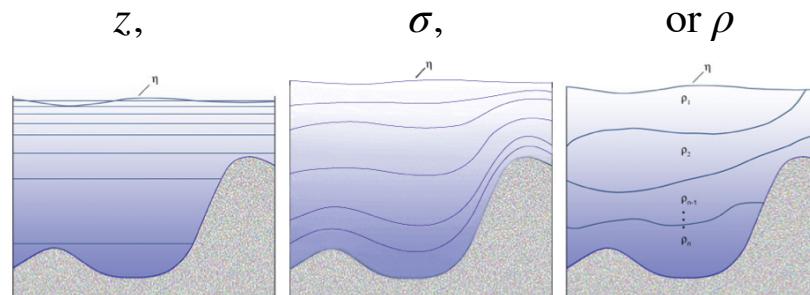


# Vertical discretization

- Equations for a generalized coordinate system:

Consider a general vertical coordinate,  $r$ , which is assumed to be a monotonic function of height,  $z$ .

$r$  can be for example:



Any variable can then be written in the new coordinate system:

$$A = A(x, y, z(x, y, r, t), t)$$

And vertical derivatives can be written

$$\frac{\partial A}{\partial r} = \frac{\partial z}{\partial r} \frac{\partial A}{\partial z}$$

$$\frac{\partial A}{\partial z} = \frac{\partial r}{\partial z} \frac{\partial A}{\partial r}$$

# Vertical discretization

- Equations for a generalized coordinate system:

Other derivatives (for a horizontal coordinate  $s = x, y$ ) can be written using the **chain rule**:

$$\frac{\partial A}{\partial s} \Big|_z = \frac{\partial A}{\partial s} \Big|_r - \frac{\partial A}{\partial z} \frac{\partial z}{\partial s} \Big|_r$$

Which gives:

$$\frac{\partial A}{\partial s} \Big|_z = \frac{\partial A}{\partial s} \Big|_r - \frac{\partial r}{\partial z} \frac{\partial A}{\partial r} \frac{\partial z}{\partial s} \Big|_r$$

# Vertical discretization

- Equations for a generalized coordinate system:

Using the chain rule, you can write the horizontal gradient:

$$\nabla_z A = \nabla_r A - \frac{\partial A}{\partial r} \frac{\partial r}{\partial z} \nabla_r z$$

And the vertical velocity can be written:

$$\begin{aligned} w = D_t z &= \left. \frac{\partial z}{\partial t} \right|_r + \left. \frac{\partial z}{\partial x} \right|_r D_t x + \left. \frac{\partial z}{\partial y} \right|_r D_t y + \left. \frac{\partial z}{\partial r} \right|_r D_t r \\ &= \left. \frac{\partial z}{\partial t} \right|_r + \vec{v} \cdot \nabla_r z + \dot{r} \frac{\partial z}{\partial r} \end{aligned}$$

# Vertical discretization

- Equations for a generalized coordinate system:

And the total derivative becomes:

$$\begin{aligned} D_t A &= \frac{\partial A}{\partial t} \Big|_z + \vec{v} \cdot \nabla_z A + w \frac{\partial A}{\partial z} \\ &= \frac{\partial A}{\partial t} \Big|_r + \vec{v} \cdot \nabla_r A + \left( w - \frac{\partial z}{\partial t} \Big|_r - \vec{v} \cdot \nabla_r z \right) \frac{\partial r}{\partial z} \frac{\partial A}{\partial r} \\ &= \frac{\partial A}{\partial t} \Big|_r + \vec{v} \cdot \nabla_r A + \dot{r} \frac{\partial A}{\partial r} \end{aligned}$$

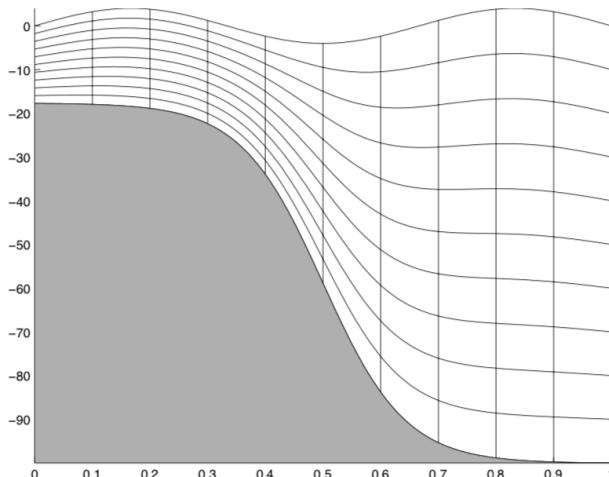
And the horizontal pressure gradient:

$$\begin{aligned} \nabla_z p &= \nabla_r p - \frac{\partial p}{\partial z} \nabla_r z \\ &= \nabla_r p + \rho \nabla_r g z \end{aligned}$$

# Vertical grid : $\sigma$ generalized coordinate

- Sigma-coordinate system:

Sigma-coordinates are terrain-following coordinates defined as:



$$r = \sigma = \frac{z}{H_z(x, y)}$$

$$\frac{\partial \sigma}{\partial z} = \frac{1}{H_z}$$

# Vertical grid : $\sigma$ generalized coordinate

- Horizontal and vertical derivatives can be written:

$$\left( \frac{\partial}{\partial x} \right)_z = \left( \frac{\partial}{\partial x} \right)_\sigma - \left( \frac{1}{H_z} \right) \left( \frac{\partial z}{\partial x} \right)_\sigma \frac{\partial}{\partial \sigma}$$

$$\left( \frac{\partial}{\partial y} \right)_z = \left( \frac{\partial}{\partial y} \right)_\sigma - \left( \frac{1}{H_z} \right) \left( \frac{\partial z}{\partial y} \right)_\sigma \frac{\partial}{\partial \sigma}$$

$$\frac{\partial}{\partial z} = \left( \frac{\partial s}{\partial z} \right) \frac{\partial}{\partial \sigma} = \frac{1}{H_z} \frac{\partial}{\partial \sigma}$$

# Vertical grid : $\sigma$ generalized coordinate

- Example: relative vorticity would be

$$\zeta = \vec{\nabla} \times \vec{u} \cdot \vec{k} = \frac{\partial v}{\partial x} \Big|_z - \frac{\partial u}{\partial y} \Big|_z$$

- Beware:  $\frac{\partial v}{\partial x} \Big|_{\sigma} - \frac{\partial u}{\partial y} \Big|_{\sigma} \neq \frac{\partial v}{\partial x} \Big|_z - \frac{\partial u}{\partial y} \Big|_z$

# Vertical grid : $\sigma$ generalized coordinate

- Example: (Hydrostatic) Potential vorticity (PV) is defined as

$$q = \left( f \vec{k} + \vec{\nabla} \times \vec{u} \right) \cdot \vec{\nabla} \rho$$

$$q = \left[ f + \frac{\partial v}{\partial x} \Big|_z - \frac{\partial u}{\partial y} \Big|_z \right] \frac{\partial \rho}{\partial z} - \frac{\partial v}{\partial z} \frac{\partial \rho}{\partial x} \Big|_z + \frac{\partial u}{\partial z} \frac{\partial \rho}{\partial y} \Big|_z$$

- Question: How do you compute PV in sigma-coordinates?

- Write the expression of  $q$  using only derivatives along sigma coordinates:

$$\frac{\partial}{\partial x} \Big|_{\sigma} \quad \frac{\partial}{\partial y} \Big|_{\sigma} \quad \frac{\partial}{\partial \sigma}$$

# Vertical grid : $\sigma$ generalized coordinate

$$\left(\frac{\partial}{\partial x}\right)_z = \left(\frac{\partial}{\partial x}\right)_\sigma - \left(\frac{1}{H_z}\right) \left(\frac{\partial z}{\partial x}\right)_\sigma \frac{\partial}{\partial \sigma}$$

$$\left(\frac{\partial}{\partial y}\right)_z = \left(\frac{\partial}{\partial y}\right)_\sigma - \left(\frac{1}{H_z}\right) \left(\frac{\partial z}{\partial y}\right)_\sigma \frac{\partial}{\partial \sigma}$$

$$\frac{\partial}{\partial z} = \left(\frac{\partial s}{\partial z}\right) \frac{\partial}{\partial \sigma} = \frac{1}{H_z} \frac{\partial}{\partial \sigma}$$

•

$$q = \left[ f + \left. \frac{\partial v}{\partial x} \right|_z - \left. \frac{\partial u}{\partial y} \right|_z \right] \frac{\partial \rho}{\partial z} - \left. \frac{\partial v}{\partial z} \right|_z \left. \frac{\partial \rho}{\partial x} \right|_z + \left. \frac{\partial u}{\partial z} \right|_z \left. \frac{\partial \rho}{\partial y} \right|_z$$

# Vertical grid : $\sigma$ generalized coordinate

- Example: (Hydrostatic) Potential vorticity (PV) is defined as

$$q = \left[ f + \frac{\partial v}{\partial x} \Big|_{\sigma} - \frac{\partial u}{\partial y} \Big|_{\sigma} \right] \frac{\partial \rho}{\partial z} - \frac{\partial v}{\partial z} \frac{\partial \rho}{\partial x} \Big|_{\sigma} + \frac{\partial u}{\partial z} \frac{\partial \rho}{\partial y} \Big|_{\sigma}$$

- Even if:  $\frac{\partial v}{\partial x} \Big|_{\sigma} - \frac{\partial u}{\partial y} \Big|_{\sigma} \neq \frac{\partial v}{\partial x} \Big|_z - \frac{\partial u}{\partial y} \Big|_z$

# Vertical grid : isopycnal coordinates

- For isopycnal coordinates, the vertical coordinate is:  $r = \rho$
- Potential vorticity (PV) is easily defined as :

$$q = \frac{1}{h} \left[ f + \left. \frac{\partial v}{\partial x} \right|_{\rho} - \left. \frac{\partial u}{\partial y} \right|_{\rho} \right] \quad \text{with} \quad h = \frac{\partial z}{\partial \rho}$$

# Vertical grid : $\sigma$ generalized coordinate

- The vertical velocity in sigma-coordinates is:

$$\Omega(x, y, \sigma, t) = \frac{1}{H_z} \left[ w - \left( \frac{z + h}{\zeta + h} \right) \frac{\partial \zeta}{\partial t} - u \frac{\partial z}{\partial x} - v \frac{\partial z}{\partial y} \right]$$

- And the “true” vertical velocity:

$$w = \frac{\partial z}{\partial t} + u \frac{\partial z}{\partial x} + v \frac{\partial z}{\partial y} + \Omega H_z.$$

# Vertical grid : $\sigma$ generalized coordinate

- Equations for ROMS/CROCO become:

$$\frac{\partial u}{\partial t} - fv + \vec{v} \cdot \nabla u = -\frac{\partial \phi}{\partial x} - \left( \frac{g\rho}{\rho_o} \right) \frac{\partial z}{\partial x} - g \frac{\partial \zeta}{\partial x} + \frac{1}{H_z} \frac{\partial}{\partial \sigma} \left[ \frac{K_m}{H_z} \frac{\partial u}{\partial \sigma} \right] + \mathcal{F}_u + \mathcal{D}_u$$

$$\frac{\partial v}{\partial t} + fu + \vec{v} \cdot \nabla v = -\frac{\partial \phi}{\partial y} - \left( \frac{g\rho}{\rho_o} \right) \frac{\partial z}{\partial y} - g \frac{\partial \zeta}{\partial y} + \frac{1}{H_z} \frac{\partial}{\partial \sigma} \left[ \frac{K_m}{H_z} \frac{\partial v}{\partial \sigma} \right] + \mathcal{F}_v + \mathcal{D}_v$$

$$\frac{\partial C}{\partial t} + \vec{v} \cdot \nabla C = \frac{1}{H_z} \frac{\partial}{\partial \sigma} \left[ \frac{K_C}{H_z} \frac{\partial C}{\partial \sigma} \right] + \mathcal{F}_T + \mathcal{D}_T$$

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

$$\frac{\partial \phi}{\partial \sigma} = \left( \frac{-gH_z\rho}{\rho_o} \right)$$

$$\frac{\partial H_z}{\partial t} + \frac{\partial(H_z u)}{\partial x} + \frac{\partial(H_z v)}{\partial y} + \frac{\partial(H_z \Omega)}{\partial \sigma} = 0$$

# Vertical grid + curvilinear grid

$$\begin{aligned} \frac{\partial}{\partial t} \left( \frac{H_z u}{mn} \right) + \frac{\partial}{\partial \xi} \left( \frac{H_z u^2}{n} \right) + \frac{\partial}{\partial \eta} \left( \frac{H_z u v}{m} \right) + \frac{\partial}{\partial \sigma} \left( \frac{H_z u \Omega}{mn} \right) \\ - \left\{ \left( \frac{f}{mn} \right) + v \frac{\partial}{\partial \xi} \left( \frac{1}{n} \right) - u \frac{\partial}{\partial \eta} \left( \frac{1}{m} \right) \right\} H_z v = \\ - \left( \frac{H_z}{n} \right) \left( \frac{\partial \phi}{\partial \xi} + \frac{g \rho}{\rho_o} \frac{\partial z}{\partial \xi} + g \frac{\partial \zeta}{\partial \xi} \right) + \frac{1}{mn} \frac{\partial}{\partial \sigma} \left[ \frac{K_m}{H_z} \frac{\partial u}{\partial \sigma} \right] + \frac{H_z}{mn} (\mathcal{F}_u + \mathcal{D}_u) \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial t} \left( \frac{H_z v}{mn} \right) + \frac{\partial}{\partial \xi} \left( \frac{H_z u v}{n} \right) + \frac{\partial}{\partial \eta} \left( \frac{H_z v^2}{m} \right) + \frac{\partial}{\partial \sigma} \left( \frac{H_z v \Omega}{mn} \right) \\ + \left\{ \left( \frac{f}{mn} \right) + v \frac{\partial}{\partial \xi} \left( \frac{1}{n} \right) - u \frac{\partial}{\partial \eta} \left( \frac{1}{m} \right) \right\} H_z u = \\ - \left( \frac{H_z}{m} \right) \left( \frac{\partial \phi}{\partial \eta} + \frac{g \rho}{\rho_o} \frac{\partial z}{\partial \eta} + g \frac{\partial \zeta}{\partial \eta} \right) + \frac{1}{mn} \frac{\partial}{\partial \sigma} \left[ \frac{K_m}{H_z} \partial v \partial \sigma \right] + \frac{H_z}{mn} (\mathcal{F}_v + \mathcal{D}_v) \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial t} \left( \frac{H_z C}{mn} \right) + \frac{\partial}{\partial \xi} \left( \frac{H_z u C}{n} \right) + \frac{\partial}{\partial \eta} \left( \frac{H_z v C}{m} \right) + \frac{\partial}{\partial \sigma} \left( \frac{H_z \Omega C}{mn} \right) = \\ \frac{1}{mn} \frac{\partial}{\partial s} \left[ \frac{K_C}{H_z} \frac{\partial C}{\partial \sigma} \right] + \frac{H_z}{mn} (\mathcal{F}_C + \mathcal{D}_C) \end{aligned}$$

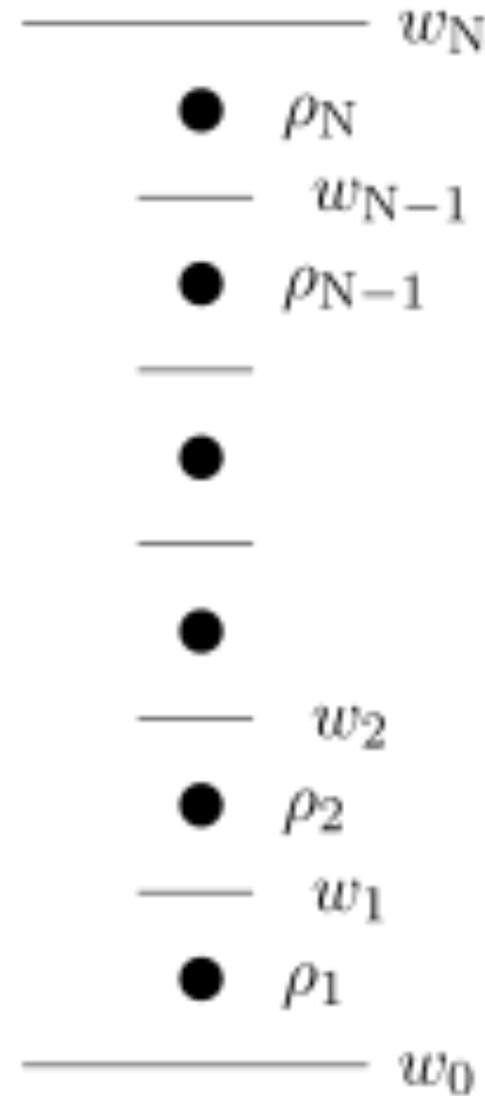
$$\rho = \rho(T, S, P)$$

$$\frac{\partial \phi}{\partial \sigma} = - \left( \frac{g H_z \rho}{\rho_o} \right)$$

$$\frac{\partial}{\partial t} \left( \frac{H_z}{mn} \right) + \frac{\partial}{\partial \xi} \left( \frac{H_z u}{n} \right) + \frac{\partial}{\partial \eta} \left( \frac{H_z v}{m} \right) + \frac{\partial}{\partial \sigma} \left( \frac{H_z \Omega}{mn} \right) = 0.$$

# Vertical discretization

ROMS: Staggered vertical grid



# Vertical grid : $\sigma$ generalized coordinate

ROMS: Generalized  $\sigma$ -Coordinate

Stretching &  
condensing of  
vertical resolution:

$$z(x, y, \sigma, t) = \zeta(x, y, t) + [\zeta(x, y, t) + h(x, y)] S(x, y, \sigma),$$
$$S(x, y, \sigma) = \frac{h_c \sigma + h(x, y) C(\sigma)}{h_c + h(x, y)}$$

$$S(x, y, \sigma) = \begin{cases} 0, & \text{if } \sigma = 0, \\ -1, & \text{if } \sigma = -1, \end{cases} \quad C(\sigma) = \begin{cases} 0, & \text{at the free-surface;} \\ -1, & \text{at the ocean bottom.} \end{cases}$$

Surface refinement function:

$$C(\sigma) = \frac{1 - \cosh(\theta_S \sigma)}{\cosh(\theta_S) - 1}, \quad \text{for } \theta_S > 0, \quad C(\sigma) = -\sigma^2, \quad \text{for } \theta_S \leq 0$$

Bottom refinement function:

$$C(\sigma) = \frac{\exp(\theta_B C(\sigma)) - 1}{1 - \exp(-\theta_B)}, \quad \text{for } \theta_B > 0$$

# Vertical grid : $\sigma$ generalized coordinate

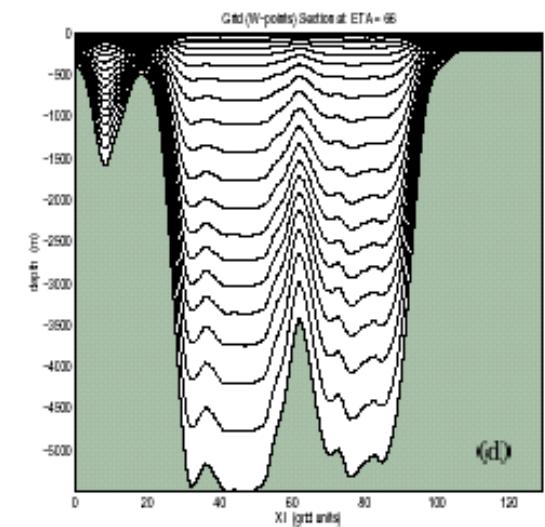
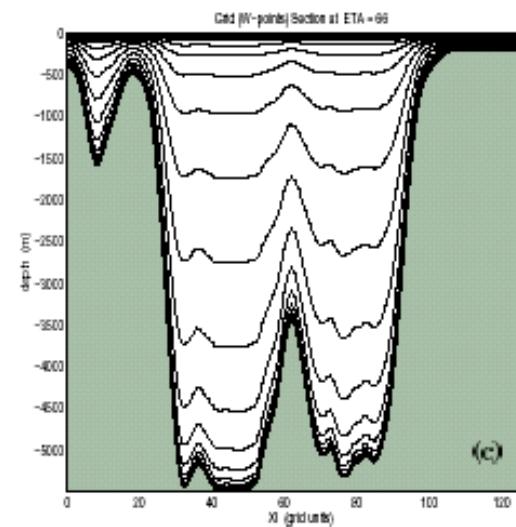
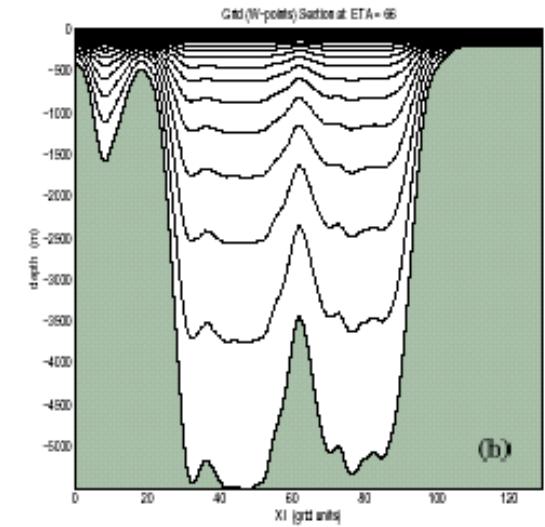
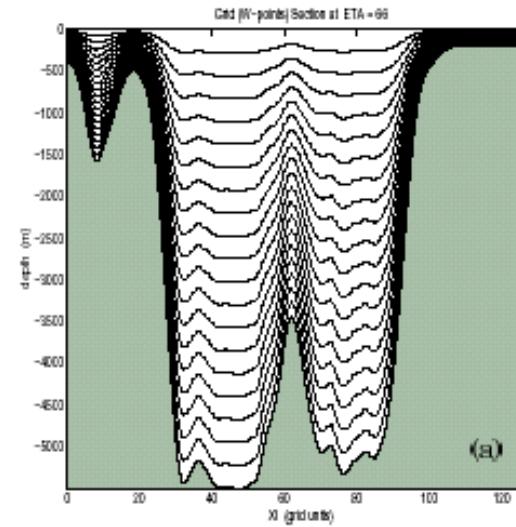
## ROMS: Generalized $\sigma$ -Coordinate

Stretching & condensing of vertical resolution

$\theta$  and b: surface and bottom

parameters:

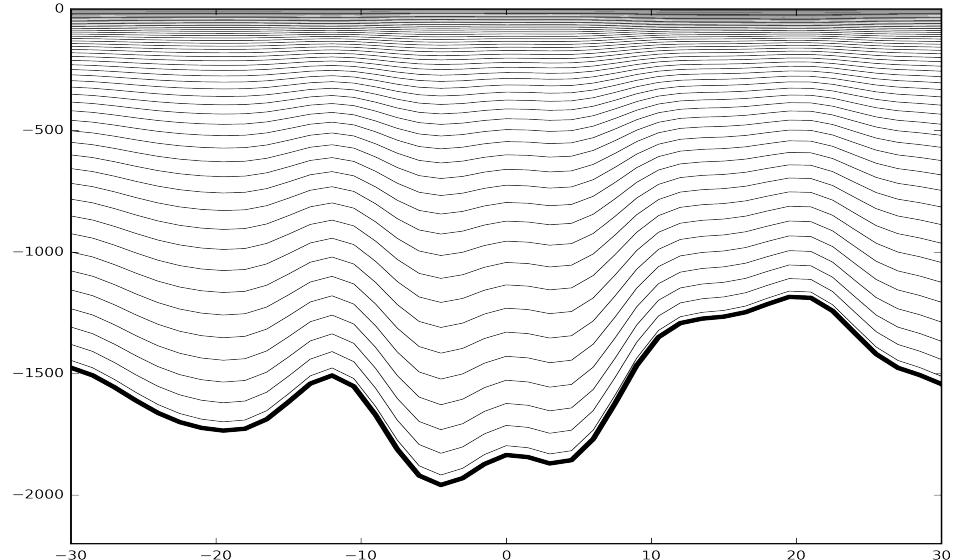
- (a)  $\theta=0, b=0$
- (b)  $\theta=8, b=0$
- (c)  $\theta=8, b=1$
- (d)  $\theta=5, b=0.4$



# Vertical grid : $\sigma$ generalized coordinate

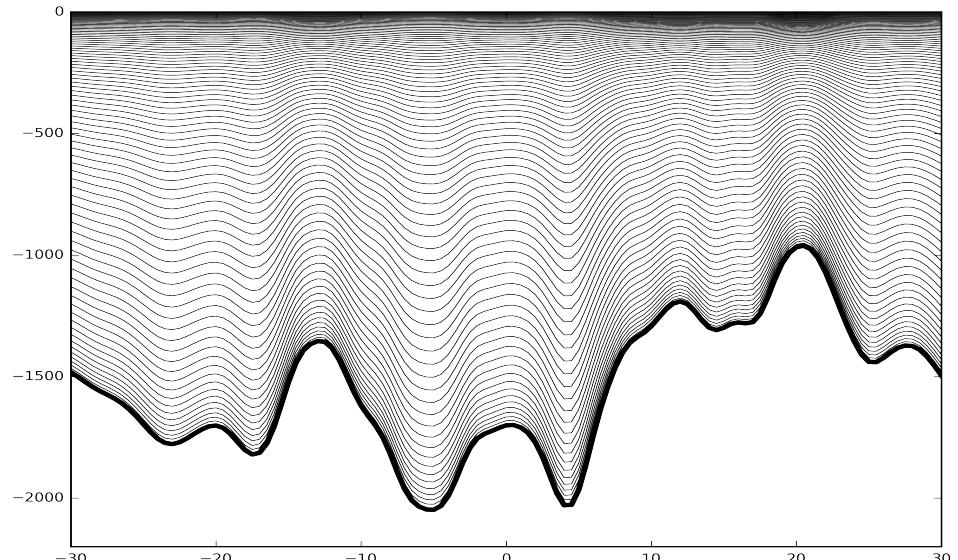
- 50 vertical levels

$$\theta=7, b=2, h_c=300 \text{ m}$$



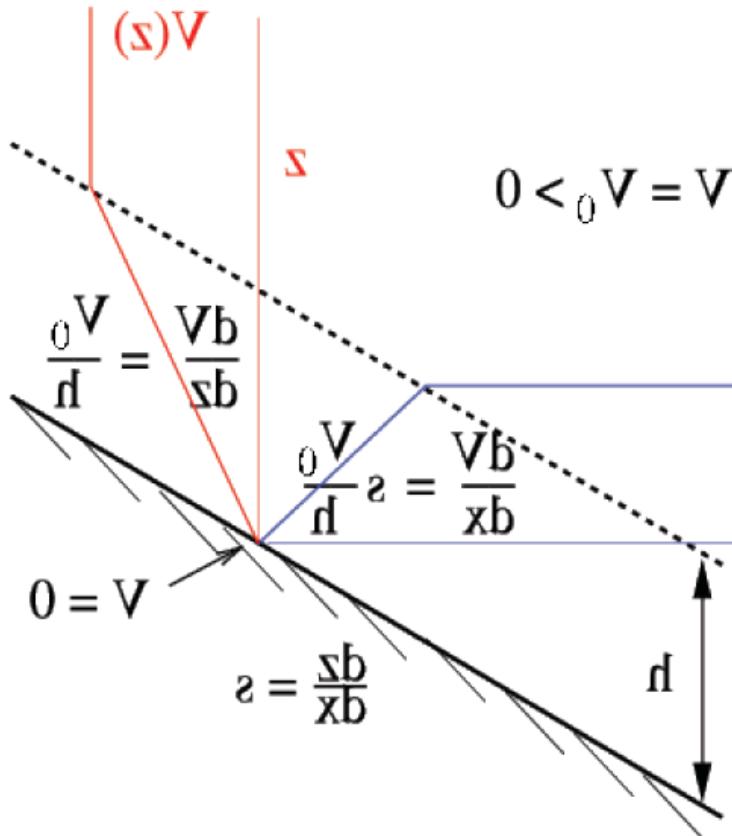
- 80 vertical levels

$$\theta=6, b=4, h_c=300 \text{ m}$$

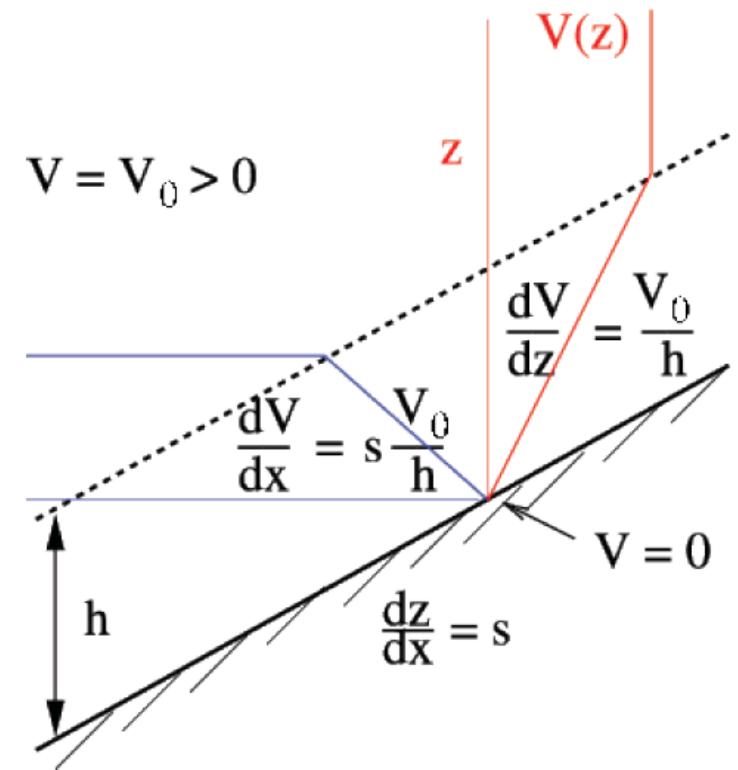


# Ex: Topographic vorticity generation

Generation of vertical vorticity within the bottom boundary layer:



Current flowing with the coast on its left in the Northern hemisphere  
(opposite to Kelvin wave propagation)  
= **Positive vorticity generation**



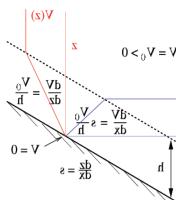
Current flowing with the coast on its right in the Northern hemisphere  
(same than Kelvin wave propagation)  
= **Negative vorticity generation**

# Ex: Topographic vorticity generation

## Positive vorticity generation :

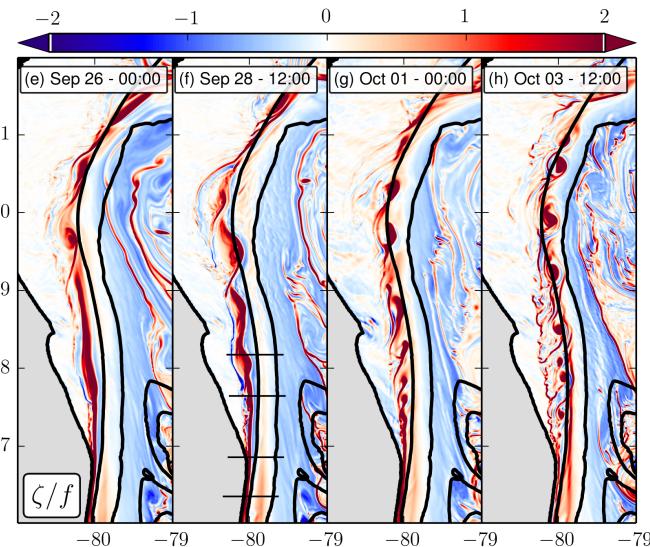
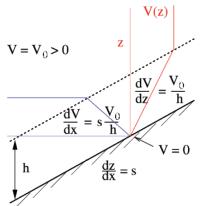
- Horizontal shear instability
- Formation of submesoscale cyclones

e.g.: *Gulf Stream along the slope*

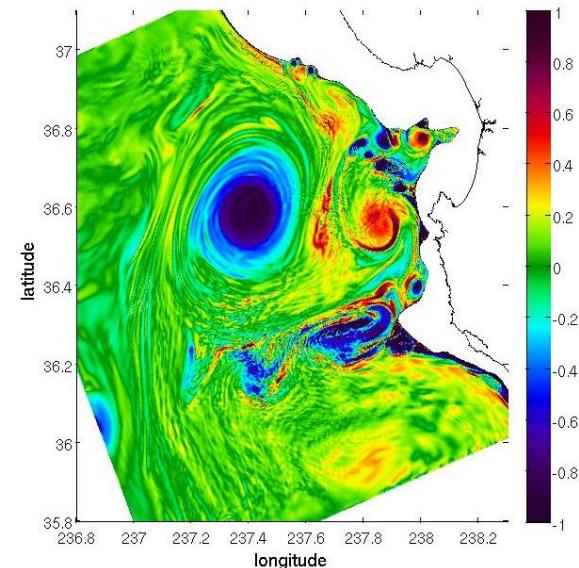


## Negative vorticity generation:

- Centrifugal instability
- Small-scale turbulence, Mixing and dissipation
- Formation of submesoscale anticyclones



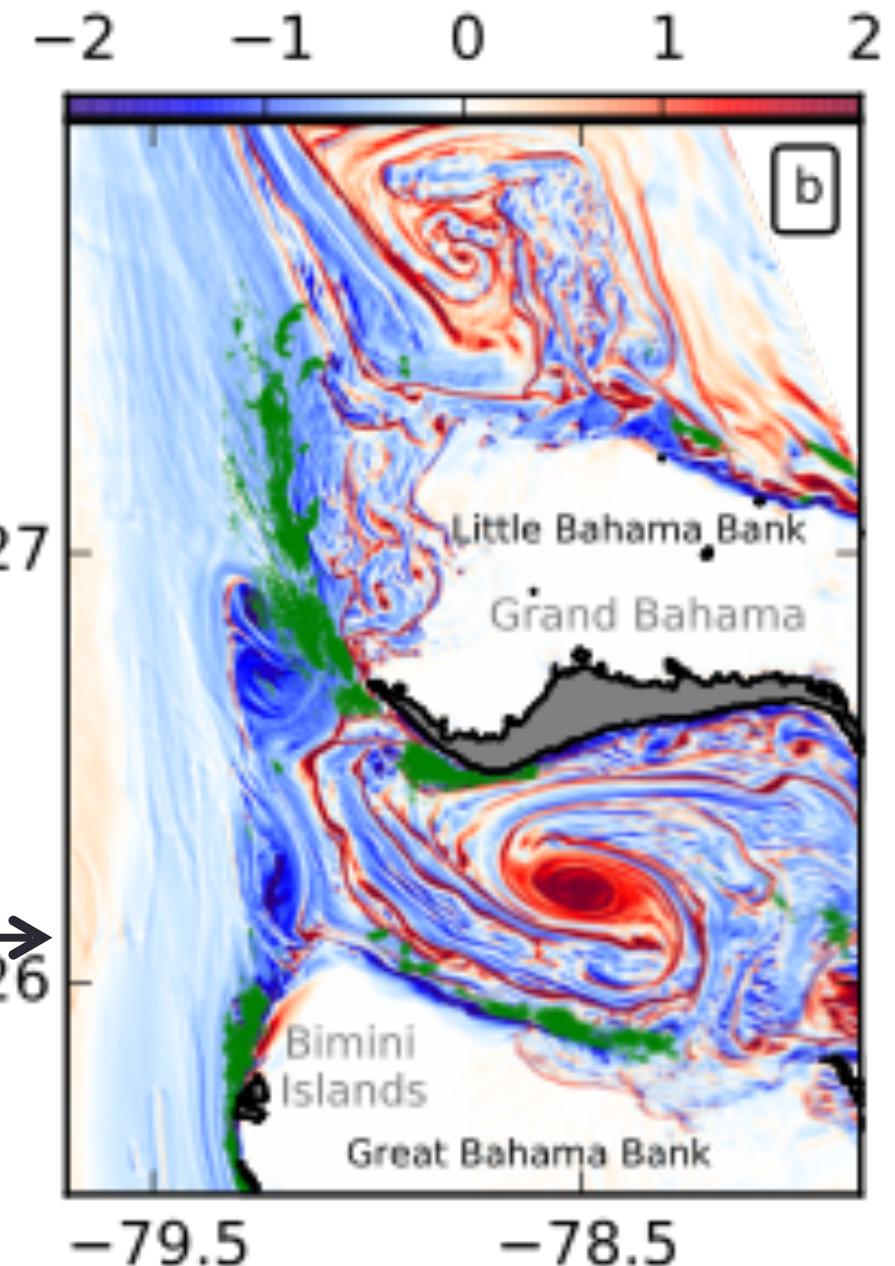
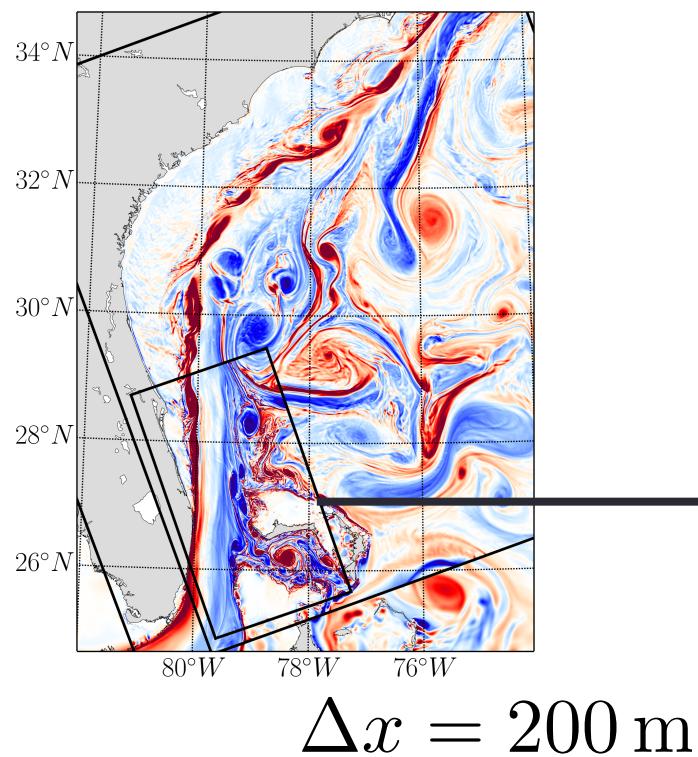
[Gula et al., GRL, 2015]



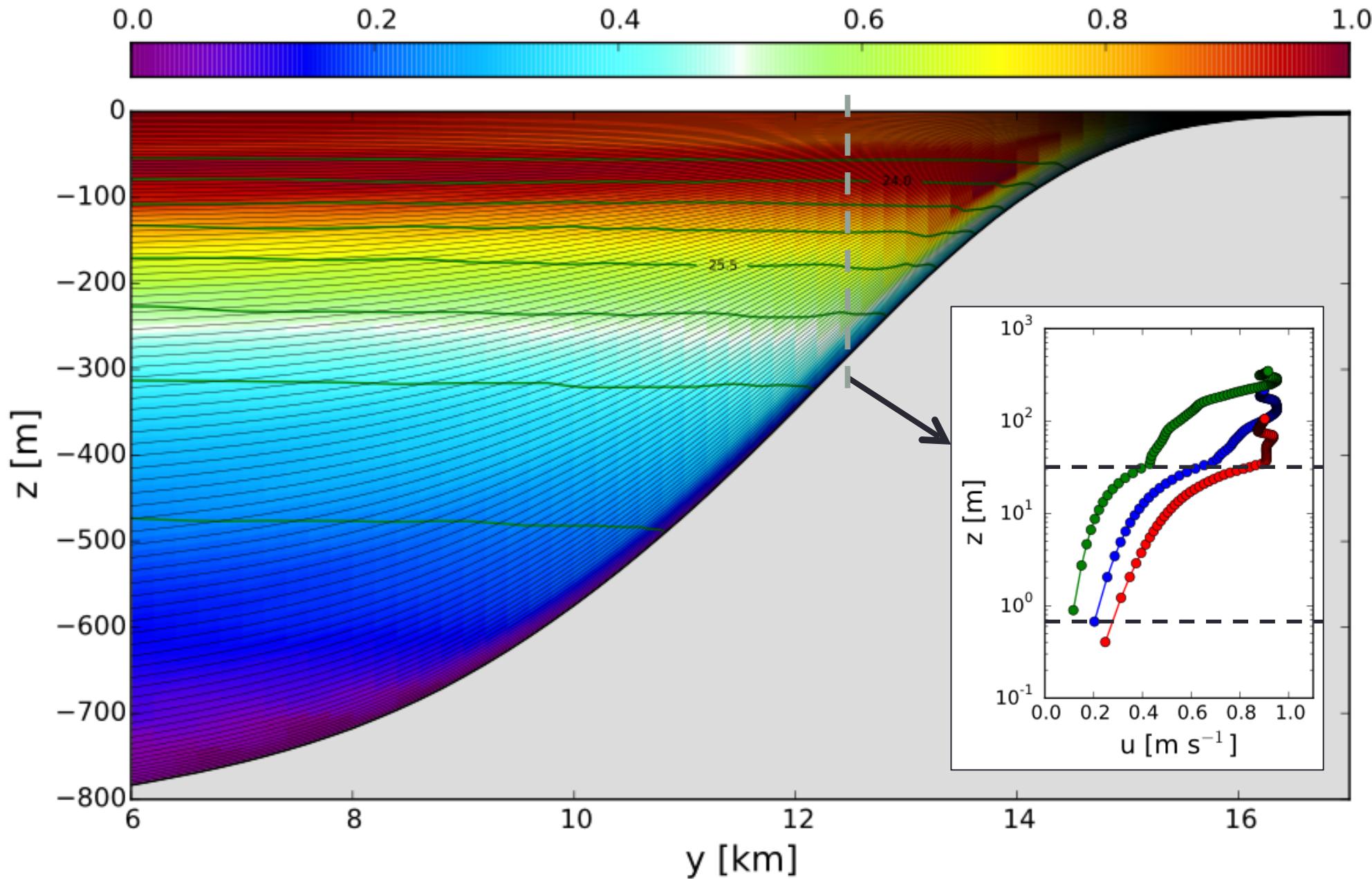
[Molemaker et al., JPO, 2015]

# Ex: Topographic vorticity generation

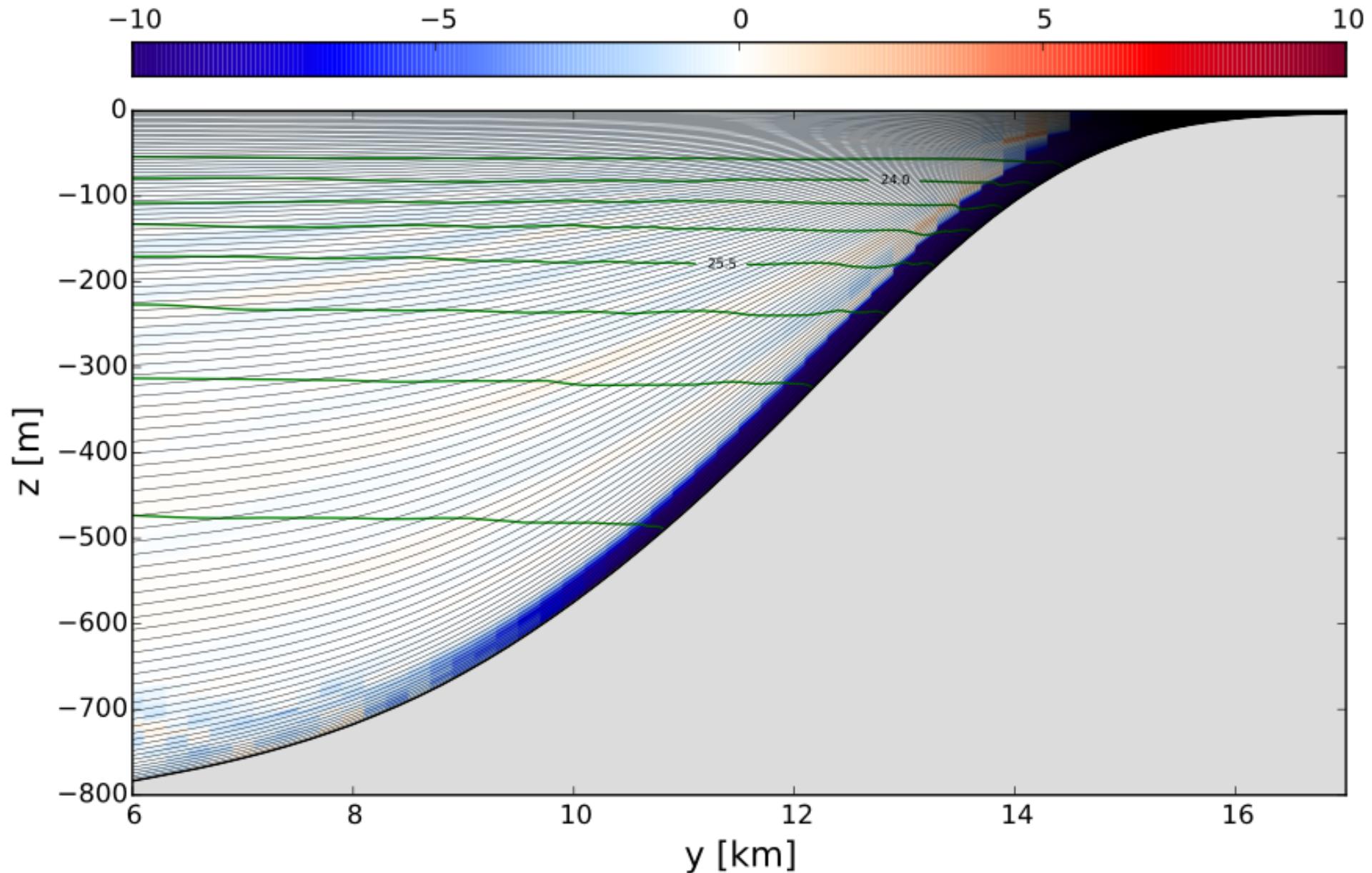
Anticyclonic vorticity generation by bottom drag on the slope on Bahamas slope



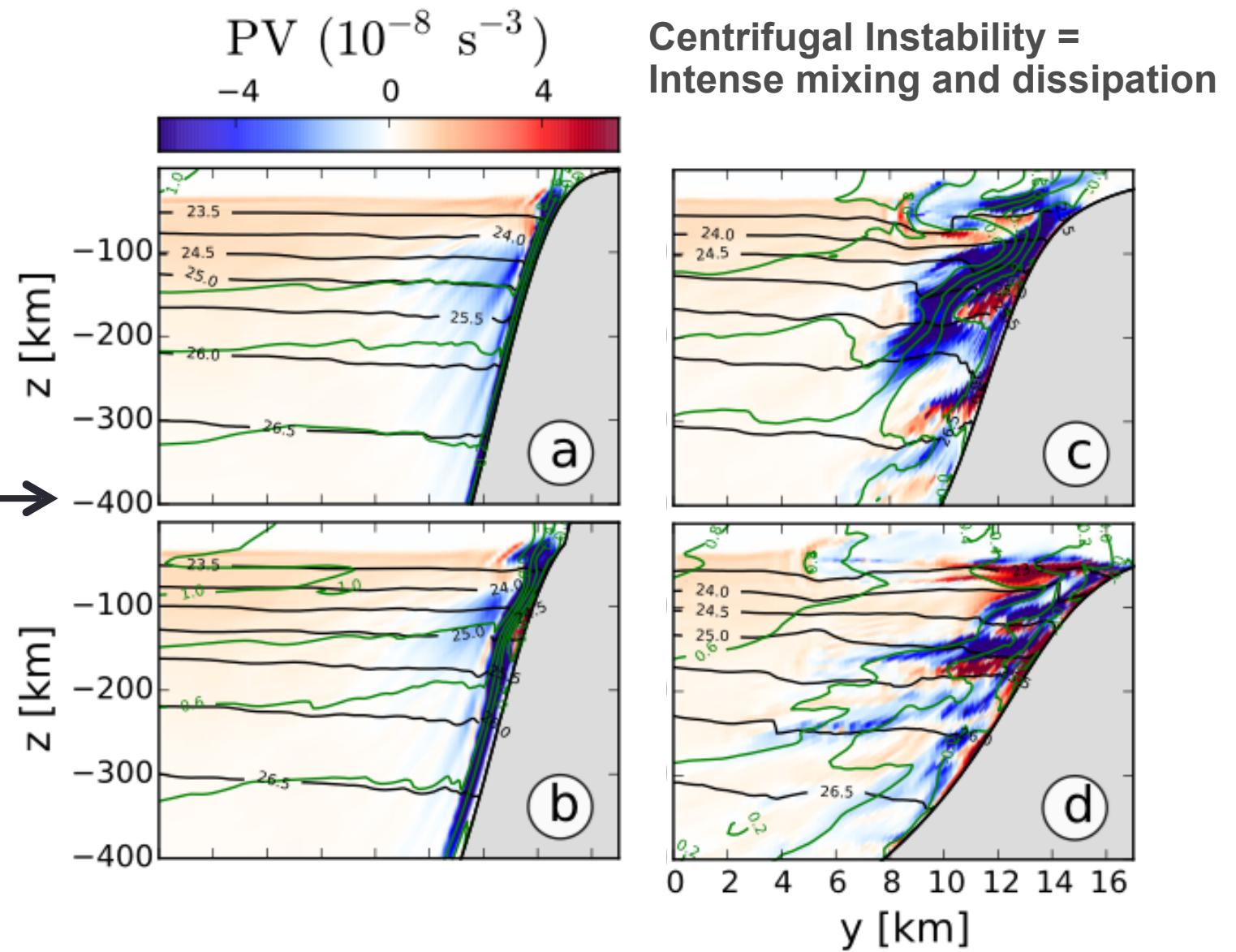
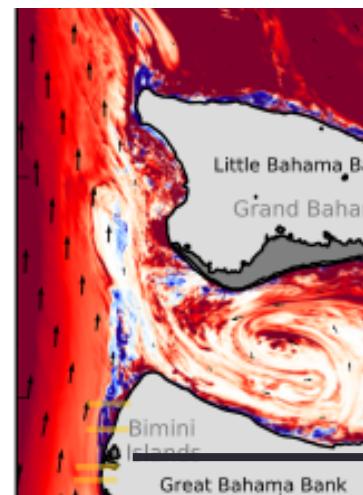
# Ex: Topographic vorticity generation



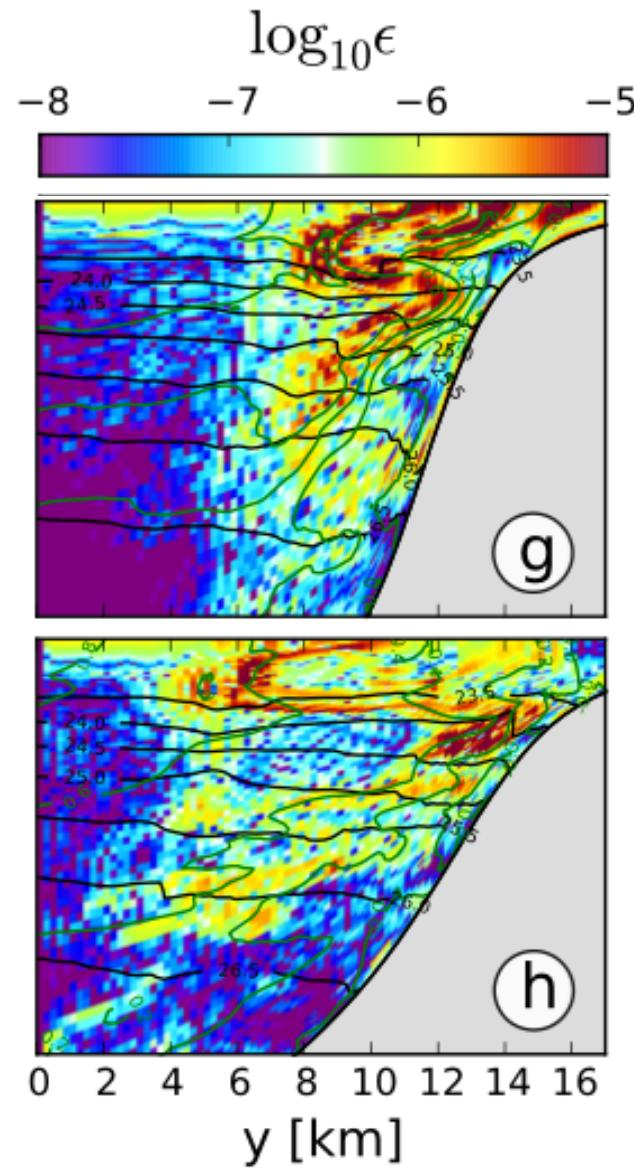
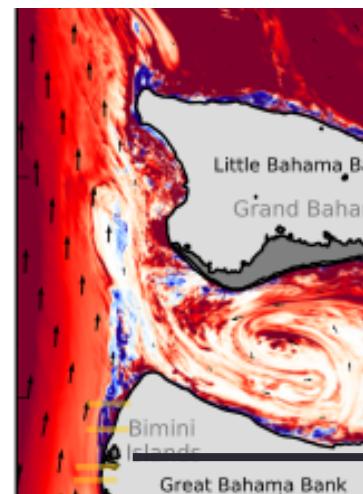
# Ex: Topographic vorticity generation



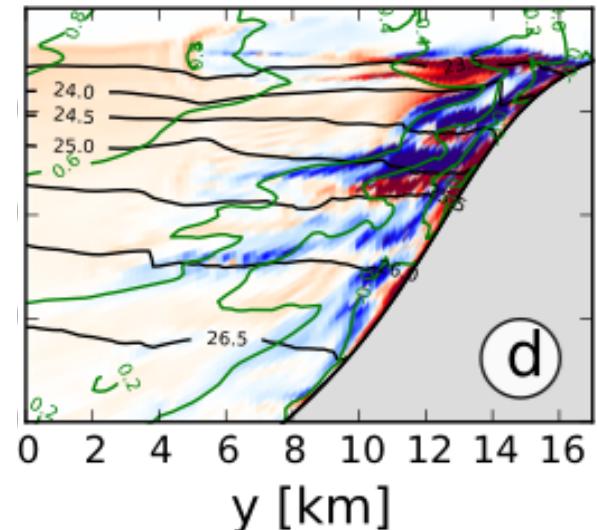
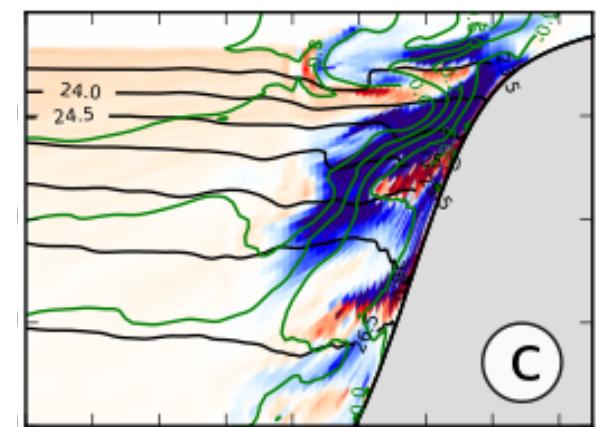
# Ex: Topographic vorticity generation



# Ex: Topographic vorticity generation



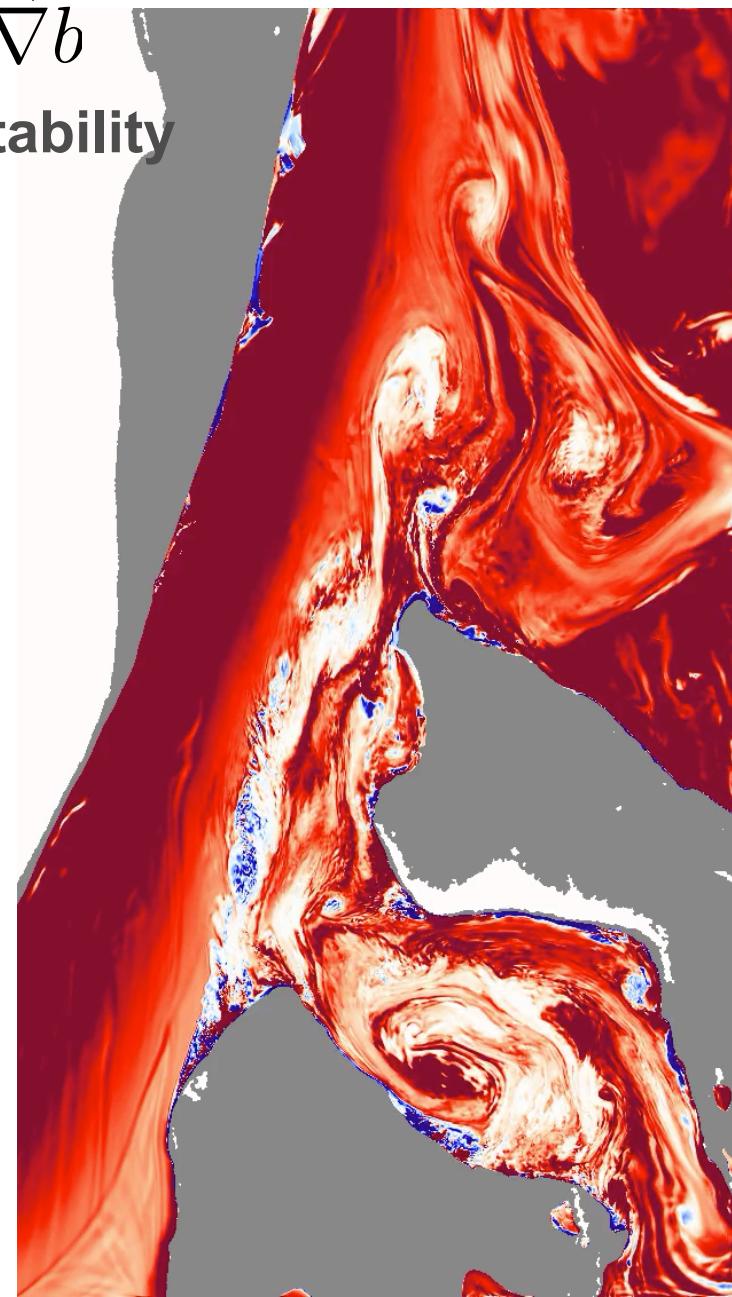
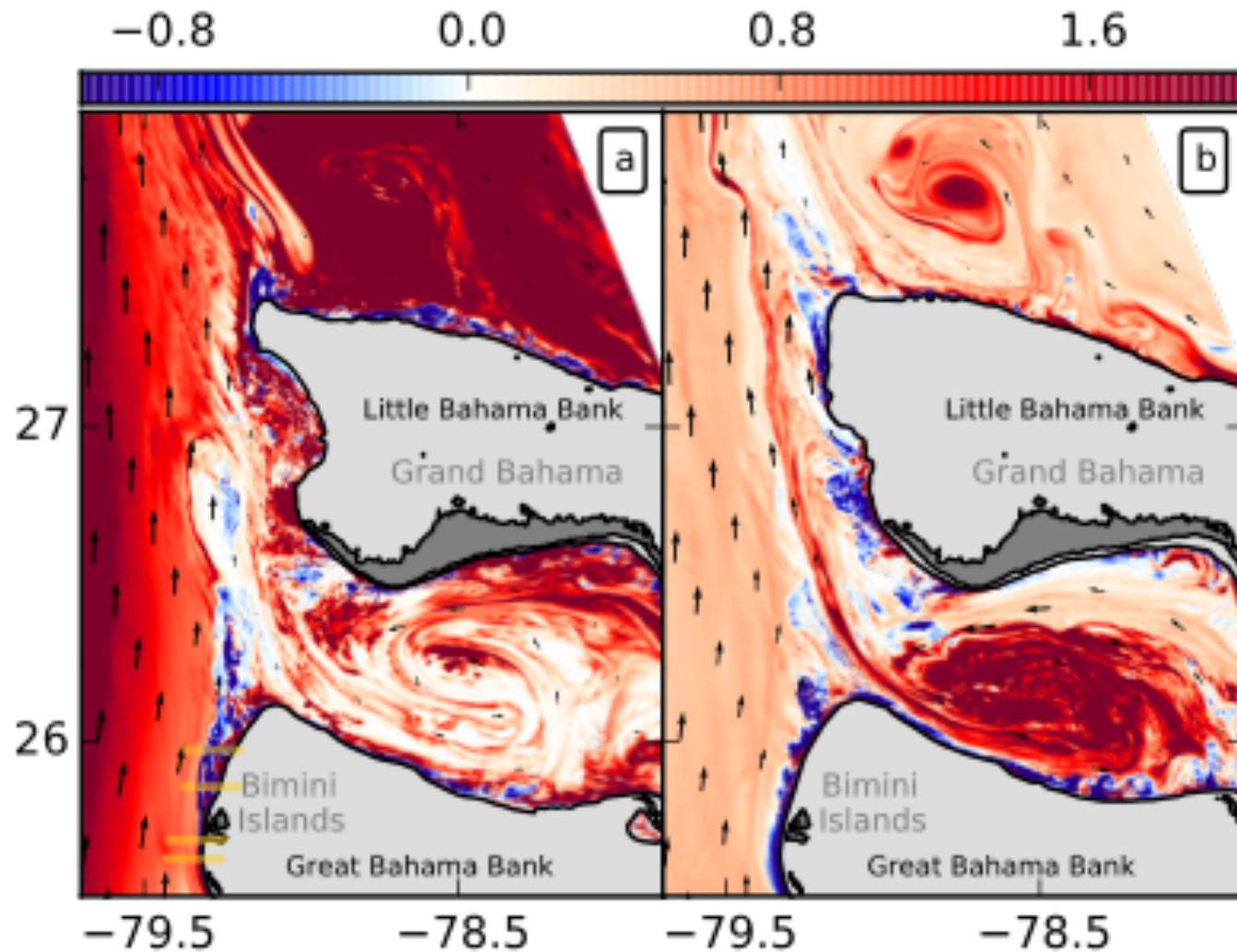
**Centrifugal Instability =  
Intense mixing and dissipation**



# Ex: Topographic vorticity generation

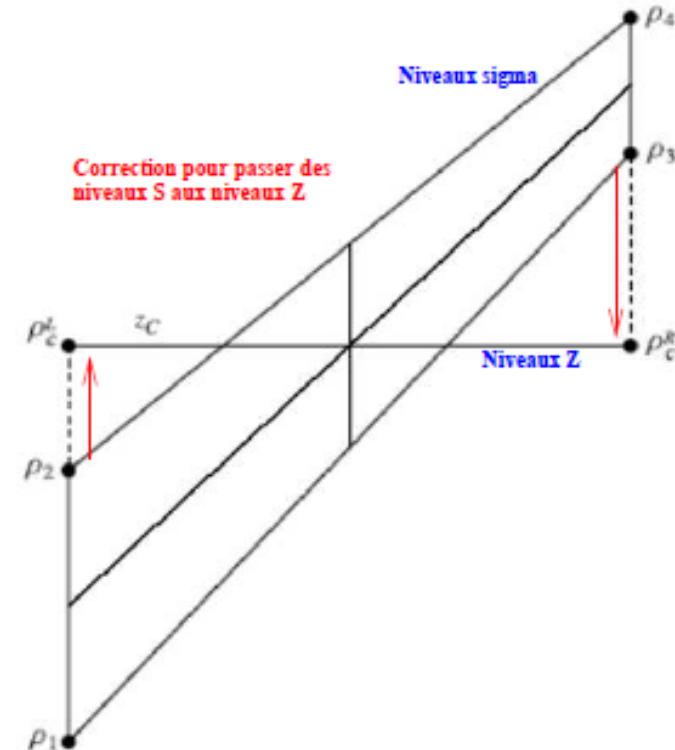
$$q = (f \vec{z} + \vec{\nabla} \times \vec{u}) \cdot \vec{\nabla} b$$

Negative PV ( $f q < 0$ ) = centrifugal (inertial) instability



# 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
- Errors can appear in the unforced flat stratification experiment



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

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

# Reducing PGF Truncation Errors

- Smoothing the topography using a nonlinear filter and a criterium:

$$r = \Delta h / h < 0.2$$

- Using a "density formulation"

- Using high order schemes to reduce the truncation error (4th order, McCalpin, 1994)

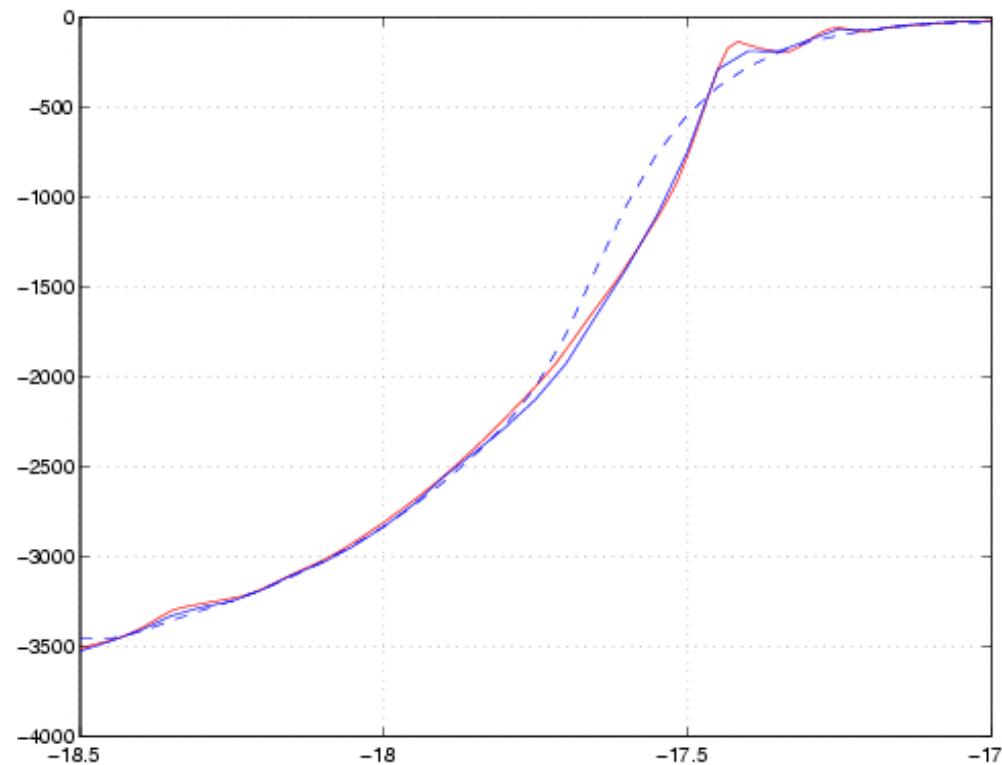
$$\begin{aligned} -\frac{1}{\rho_0} \frac{\partial P}{\partial x} \Big|_z &= -\frac{1}{\rho_0} \frac{\partial P}{\partial x} \Big|_{z=\zeta} - \frac{g}{\rho_0} \int_z^{\zeta} \frac{\partial p}{\partial x} \Big|_z dz' \\ &= -\frac{gp(\zeta)}{\rho_0} \frac{\partial \zeta}{\partial x} - \frac{g}{\rho_0} \int_z^{\zeta} \left[ \frac{\partial p}{\partial x} \Big|_s - \frac{\partial p}{\partial z'} \frac{\partial z'}{\partial x} \Big|_s \right] dz', \end{aligned}$$

- Gary, 1973: subtracting a reference horizontal averaged value from density ( $\rho' = \rho - \rho_a$ ) before computing pressure gradient
- Rewriting Equation of State: reduce passive compressibility effects on pressure gradient

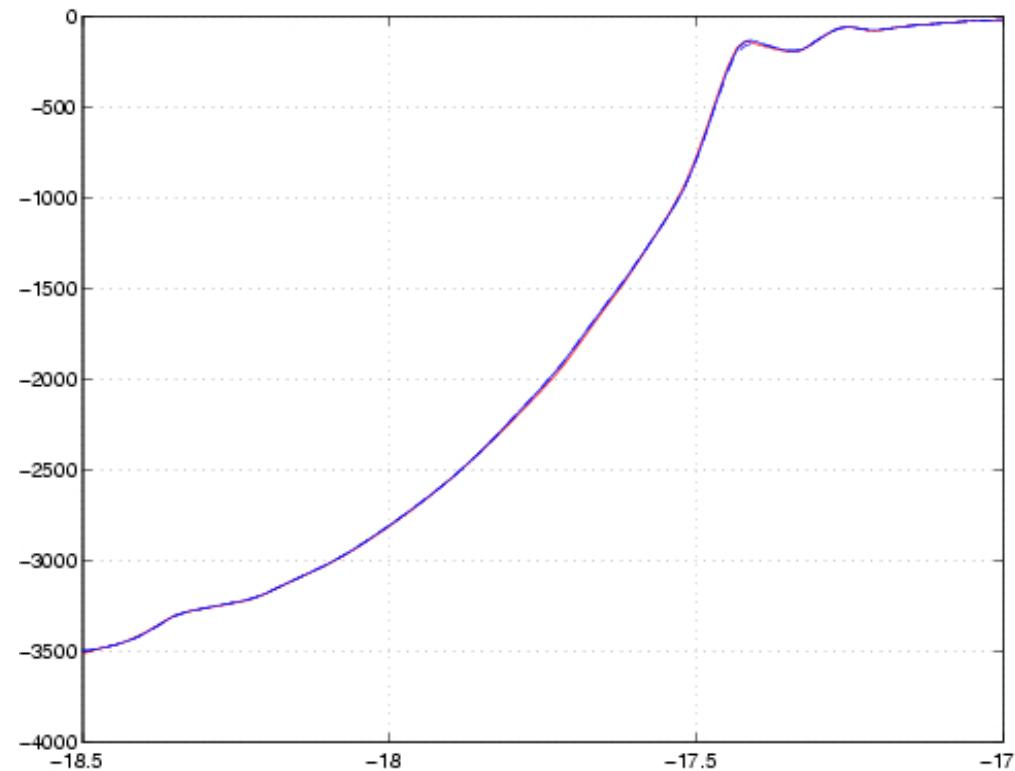
# Smoothing methods

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

*Res: 5 km ;  $r_{max} = 0.25$*



*Res: 1 km ;  $r_{max} = 0.25$*



# Turbulence closure

In ROMS:

$$\mathcal{F}_u = \frac{\partial}{\partial z} \left( K_{Mv} \frac{\partial u}{\partial z} \right) + \nabla_h (K_{Mh} \cdot \nabla_h u)$$

$$\mathcal{F}_v = \frac{\partial}{\partial z} \left( K_{Mv} \frac{\partial v}{\partial z} \right) + \nabla_h (K_{Mh} \cdot \nabla_h v)$$

$$\mathcal{S}_T = \frac{\partial}{\partial z} \left( K_{Tv} \frac{\partial T}{\partial z} \right) + \nabla_h (K_{Th} \cdot \nabla_h T)$$

$$\mathcal{S}_S = \frac{\partial}{\partial z} \left( K_{Sv} \frac{\partial S}{\partial z} \right) + \nabla_h (K_{Sh} \cdot \nabla_h S)$$

Vertical mixing

Horizontal diffusion

# Turbulence closure

Vertical Mixing:

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

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

Horizontal diffusion:

• Explicit diffusion

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

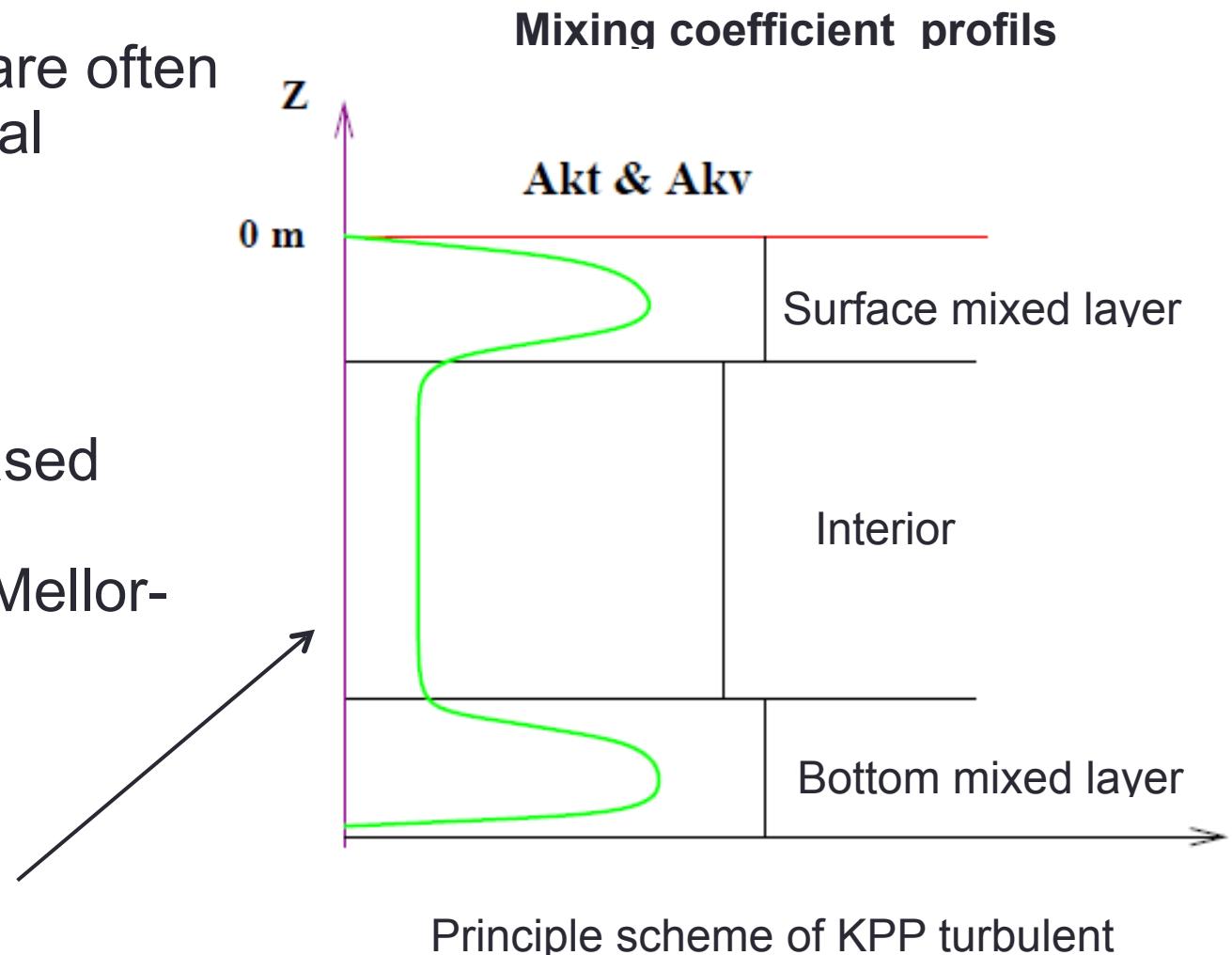
• Implicit (comes from the advective scheme)

# Non local K-profile parameterization

- Mixed layer schemes are often based on one-dimensional « column physics »

- Boundary layer parameterizations are based either on:

- Turbulent closure (Mellor-Yamada, TKE)
- **K profile (KPP)**



# Vertical grid : $\sigma$ generalized coordinate

- Boundary conditions become:

$$\begin{aligned} \text{top } (\sigma = 0) \quad & \left( \frac{K_m}{H_z} \right) \frac{\partial u}{\partial \sigma} = \tau_s^x(x, y, t) \\ & \left( \frac{K_m}{H_z} \right) \frac{\partial v}{\partial \sigma} = \tau_s^y(x, y, t) \\ & \left( \frac{K_C}{H_z} \right) \frac{\partial C}{\partial \sigma} = \frac{Q_C}{\rho_o c_P} \\ & \Omega = 0 \end{aligned}$$

$$\begin{aligned} \text{and bottom } (\sigma = -1) \quad & \left( \frac{K_m}{H_z} \right) \frac{\partial u}{\partial \sigma} = \tau_b^x(x, y, t) \\ & \left( \frac{K_m}{H_z} \right) \frac{\partial v}{\partial \sigma} = \tau_b^y(x, y, t) \\ & \left( \frac{K_C}{H_z} \right) \frac{\partial C}{\partial \sigma} = 0 \\ & \Omega = 0. \end{aligned}$$

# Bottom friction parametrization

1. Linear friction, with  
**r** **friction**  
**velocities** [m/s]  $\rightarrow (\tau_b^x, \tau_b^y) = -r (u_b, v_b)$
2. Quadratic friction,  
 controled by a  
 constant drag  
 coefficient **Cd**  $\rightarrow (\tau_b^x, \tau_b^y) = C_d \sqrt{u_b^2 + v_b^2} (u_b, v_b)$
3. Quadratic friction  
 coefficient, using  
 variable **Cd** (Von  
 Karman log. layer)  $\rightarrow$ 
  - $(\tau_b^x, \tau_b^y) = C_d \sqrt{u_b^2 + v_b^2} (u_b, v_b)$
  - $C_d = \left( \frac{\kappa}{\log[\Delta z_b/z_r]} \right)^2$  si  $C_d^{min} < C_d < C_d^{max}$
  - $C_d = C_d^{min}$  ou  $C_d^{max}$
  - $\kappa = 0.41$
  - $z_r$  = Roughness Length
  - $\Delta z_b$  = thickness of the first bottom level

# For Activity : Computing z and vertical interpolations

- See example in :

[https://github.com/Mesharou/mesharou.github.io/blob/master/ModNum/example\\_croco.ipynb](https://github.com/Mesharou/mesharou.github.io/blob/master/ModNum/example_croco.ipynb)

```
# some tools to interpolate croco outputs vertically
```

```
import tools as to
```

```
#####
#Load variables and parameters
#####
```

```
nc = Dataset(ncfile, 'r')
```

```
temp3d=np.array(nc.variables['temp'][[-1,:,:,:]])
```

```
#Load some parameters
```

```
zeta=nc.variables['zeta'][[-1,:,:]]  
topo=nc.variables['h'][::]  
pm=nc.variables['pm'][::]  
pn=nc.variables['pn'][::]
```

```
hc = nc.hc  
Cs_r = nc.Cs_r  
Cs_w = nc.Cs_w
```

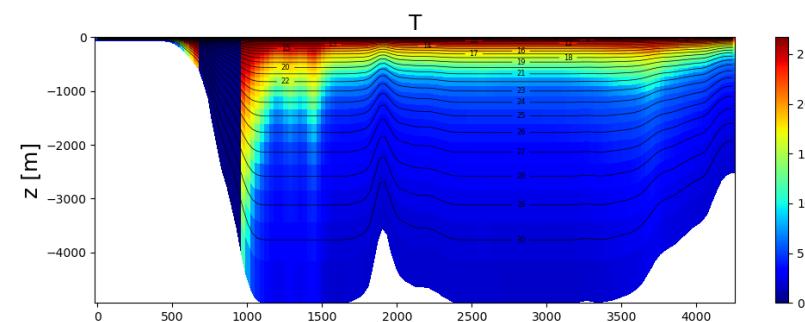
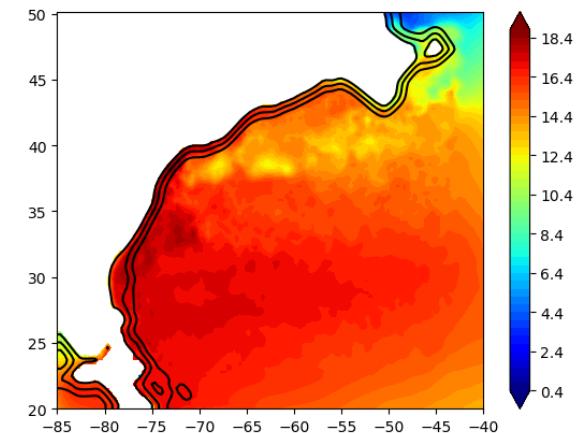
```
#close netcdf file  
nc.close()
```

```
#####
#Compute vertical coordinates
#####
```

```
(z_r,z_w) = to.zlevs(topo,zeta, hc, Cs_r, Cs_w)
```

```
#####
#Interpolate a variable on a given depth
#####
```

```
t400 = to.vinterp(temp3d,z_r,-500,topo=topo,cubic=1)
```



## For Activity : Computing z and vertical interpolations

- ## □ An another using xarray in

[https://github.com/Mesharou/mesharou.github.io/blob/master/ModNum/example\\_croco\\_xarray.ipynb](https://github.com/Mesharou/mesharou.github.io/blob/master/ModNum/example_croco_xarray.ipynb)

**Example use of xarray/dask/xgcm with CIOCO files**

- works with xarray/LBGRID
- works with xarray/dask
- works with older XIOS `ioBroker` files

- Interpolation to horizontal grids
- Computation of derivatives (relative/rectified)
- Plotting vertical sections
- Interpolation on a preexisting grid

**Examples adapted from:** <https://github.com/agentx/xiosfileexamples>, [xiosfileexamples](https://github.com/agentx/xiosfileexamples)

## Open the dataset

```
import xarray as xr
import numpy as np
import xiosfile

# Read CIOCO file dataset
dataset = xiosfile.open('cioco.nc')
# Open the dataset
# Open the dataset
# Open the dataset
```

---

## Open regular CIOCO files

```
# Example 1: CIOCO file without dask/GD (most cases)
# Example 2: CIOCO file with dask/GD (most cases)
# Example 3: regular xiosfile CIOCO file
# Example 4: dask/GD CIOCO file
# Example 5: regular CIOCO file
# Example 6: dask/GD CIOCO file
# Example 7: regular CIOCO file
# Example 8: dask/GD CIOCO file
```

---

```
# Open the dataset
# Open the dataset
# Open the dataset
```

