Numerical Modelling

the anatomy of an ocean model

Jonathan GULA
gula@univ-brest.fr

Outline

- **Lesson 1 :** [D109]
  - Introduction
  - Equations of motions
  - *Activity 1 [run an ocean model]*
- **Lesson 2 :** [D109]
  - Subgrid-scale parameterization
  - Dynamics of the ocean gyre
  - *Activity 2 [Dynamics of an ocean gyre]*
- **Lesson 3 :** [D109]
  - Horizontal Discretization
  - Vertical coordinates
  - *Activity 2 [Dynamics of an ocean gyre]*
  - *Activity 3 [Impacts of numerics / topography]*
- **Lesson 4 :** [D109]
  - Numerical schemes
  - Presentation of the model CROCO
  - *Activity 3 [Impacts of numerics / topography]*
- **Lesson 5 :** [D109]
  - Boundary Forcings
  - *Activity 5 [Design a realistic simulation]*
- **Lesson 6 :** [D109]
  - Diagnostics and validation
  - *Activity 6 [Analyze a realistic simulation]*
- **Lesson 7 :** [D109]
  - *Project*

Presentations and material will be available at :

jgula.fr/ModNum/

#3 Discretization

Master's degree 2nd year Marine Physics

Jonathan GULA
gula@univ-brest.fr

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

CROCO/CROCO:

•https://www.myCROCO.org/wiki/

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

Discretization

We cannot solve the continuous equations.

How to represent them with a finite set of numbers?

Discretization

We cannot solve the continuous equations.

How to represent them with a finite set of numbers?

Two basic strategies:

- Series expansion methods (spectral, finite element)
- **Grid-point methods** (finite difference, finite volume)

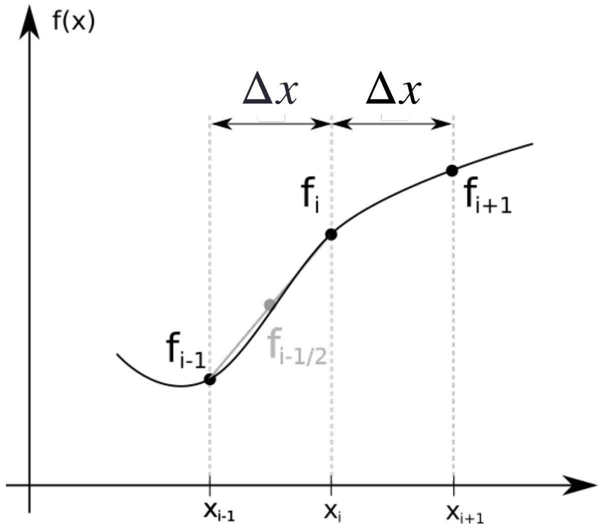
Discretization

Spectral methods:

- Variables are represented in a **spectral basis** (Fourier, eigenfunctions, orthogonal polynomials).
- The equations are solved **scale by scale**, eg: $\frac{d\hat{\mathbf{u}}_{\mathbf{k}}}{dt} = -i \sum_{\mathbf{p}+\mathbf{q}=\mathbf{k}} (\mathbf{k} \cdot \hat{\mathbf{u}}_{\mathbf{q}})\, \hat{\mathbf{u}}_{\mathbf{p}} - \nu k^2 \hat{\mathbf{u}}_{\mathbf{k}} + \hat{\mathbf{f}}_{\mathbf{k}}$
- *The nonlinear term is sometimes computed in physical space (= pseudo-spectral methods)*
- **Pros:** Very high accuracy / ideal for idealized studies in periodic domains
- **Cons:** Difficult for complex geometrics, High cost in 3d...

Exemple: https://journals.ametsoc.org/view/journals/atot/21/1/1520-0426_2004_021_0069_asmfps_2_0_co_2.xml

Discretization



<u>Grid-point methods</u>:

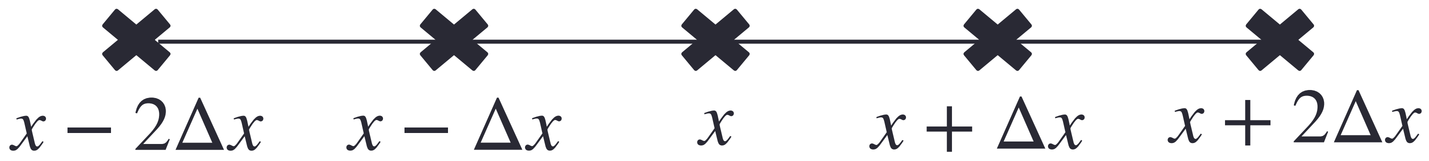
- Use of Taylor series to estimate truncation errors

$$f(x_{i+1}) = f(x_i) + \frac{f'(x)}{1!}\Delta x + \frac{f^{(2)}(x)}{2!}\Delta x^2 + \ldots + \frac{f^{(n)}(x)}{n!}\Delta x^n + R_n$$

- Order of accuracy = **lower order** of the error of a scheme

Discretization

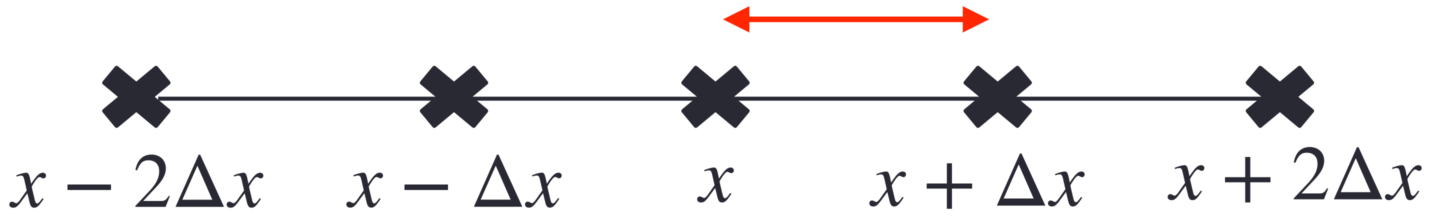
- Constructing a difference operator using Taylor series:



$$f(x + \Delta x) = f(x) + \frac{f'(x)}{1!}\Delta x + \frac{f^{(2)}(x)}{2!}\Delta x^2 + \ldots + \frac{f^{(n)}(x)}{n!}\Delta x^n + R_n$$

Discretization

- Constructing a difference operator using Taylor series:



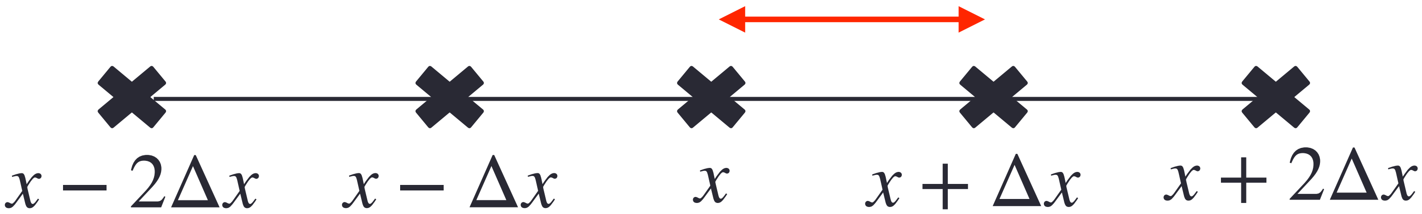
$$f(x + \Delta x) = f(x) + \frac{f'(x)}{1!}\Delta x + \frac{f^{(2)}(x)}{2!}\Delta x^2 + \ldots + \frac{f^{(n)}(x)}{n!}\Delta x^n + R_n$$

- 1st order approx. for derivative:

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} - \frac{f^{(2)}(x)}{2!}\Delta x + \ldots$$

Discretization

- Constructing a difference operator using Taylor series:



$$f(x+\Delta x) = f(x) + \frac{f'(x)}{1!}\Delta x + \frac{f^{(2)}(x)}{2!}\Delta x^2 + \ldots + \frac{f^{(n)}(x)}{n!}\Delta x^n + R_n$$

- 1st order approx. for derivative (Downstream):

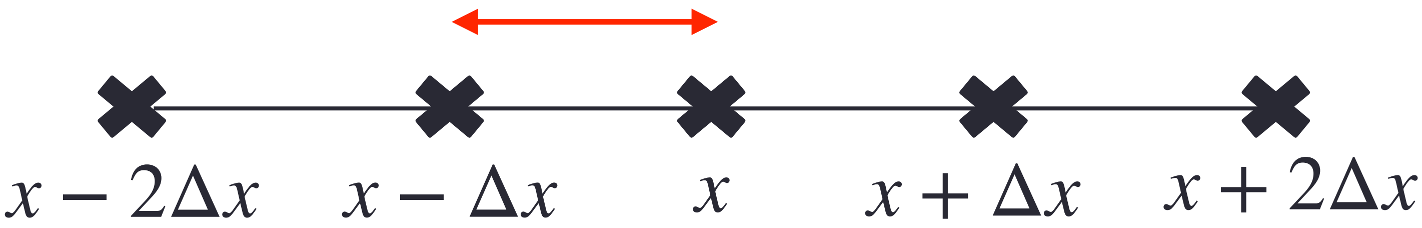
Truncation Error

$$\boxed{f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x}} - \frac{f^{(2)}(x)}{2!}\Delta x + \ldots$$

$O(\Delta x)$

Discretization

- Constructing a difference operator using Taylor series:



$$f(x - \Delta x) = f(x) - \frac{f'(x)}{1!}\Delta x + \frac{f^{(2)}(x)}{2!}\Delta x^2 + \ldots + (-1)^n \frac{f^{(n)}(x)}{n!}\Delta x^n + R_n$$

- 1st order approx. for derivative (upstream):

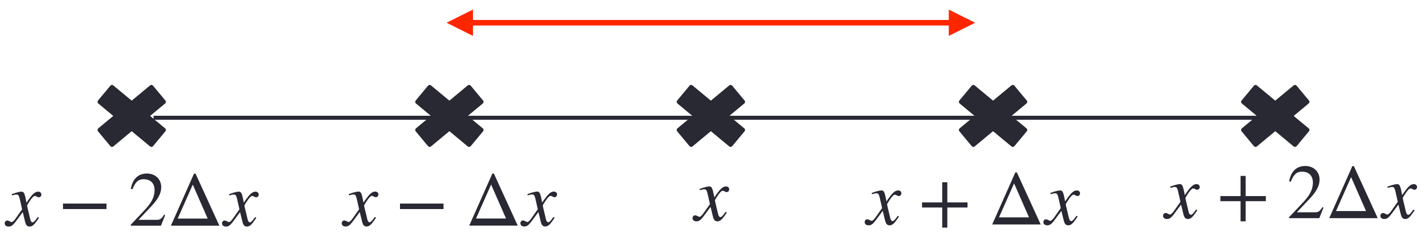
Truncation Error

$$\boxed{f'(x) = \frac{f(x) - f(x - \Delta x)}{\Delta x}} + \frac{f^{(2)}(x)}{2!}\Delta x + \ldots$$

$$O(\Delta x)$$

Discretization

- Constructing a difference operator using Taylor series:



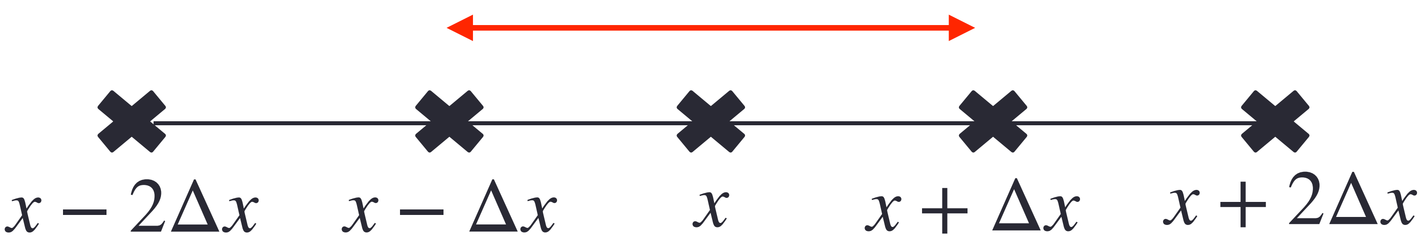
$$f(x+\Delta x) = f(x) + \frac{f'(x)}{1!}\Delta x + \frac{f^{(2)}(x)}{2!}\Delta x^2 + \ldots + \frac{f^{(n)}(x)}{n!}\Delta x^n + R_n$$

$$f(x-\Delta x) = f(x) - \frac{f'(x)}{1!}\Delta x + \frac{f^{(2)}(x)}{2!}\Delta x^2 + \ldots + (-1)^n\frac{f^{(n)}(x)}{n!}\Delta x^n + R_n$$

- 2nd order approx. for derivative?

Discretization

- Constructing a difference operator using Taylor series:



$$f(x+\Delta x) = f(x) + \frac{f'(x)}{1!}\Delta x + \frac{f^{(2)}(x)}{2!}\Delta x^2 + \ldots + \frac{f^{(n)}(x)}{n!}\Delta x^n + R_n$$

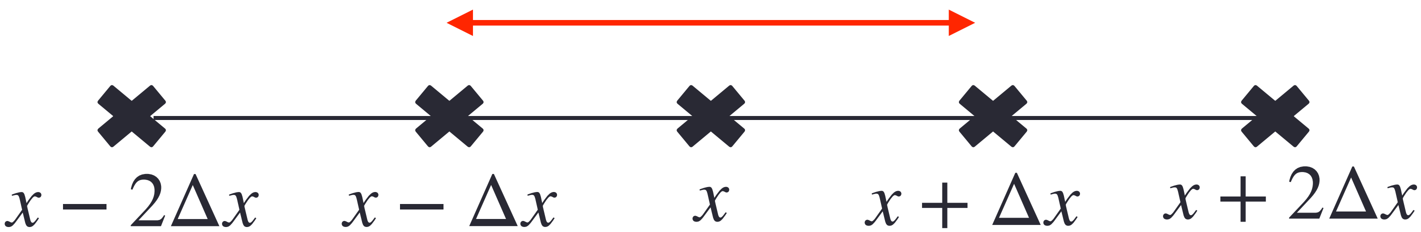
$$f(x-\Delta x) = f(x) - \frac{f'(x)}{1!}\Delta x + \frac{f^{(2)}(x)}{2!}\Delta x^2 + \ldots + (-1)^n\frac{f^{(n)}(x)}{n!}\Delta x^n + R_n$$

- 2nd order approx. for derivative?

$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} - \frac{f^{(3)}(x)}{3!}\Delta x^2 + \ldots$$

Discretization

- Constructing a difference operator using Taylor series:



$$f(x+\Delta x) = f(x) + \frac{f'(x)}{1!}\Delta x + \frac{f^{(2)}(x)}{2!}\Delta x^2 + \ldots + \frac{f^{(n)}(x)}{n!}\Delta x^n + R_n$$

$$f(x-\Delta x) = f(x) - \frac{f'(x)}{1!}\Delta x + \frac{f^{(2)}(x)}{2!}\Delta x^2 + \ldots + (-1)^n\frac{f^{(n)}(x)}{n!}\Delta x^n + R_n$$

- 2nd order approx. for derivative?

Truncation Error

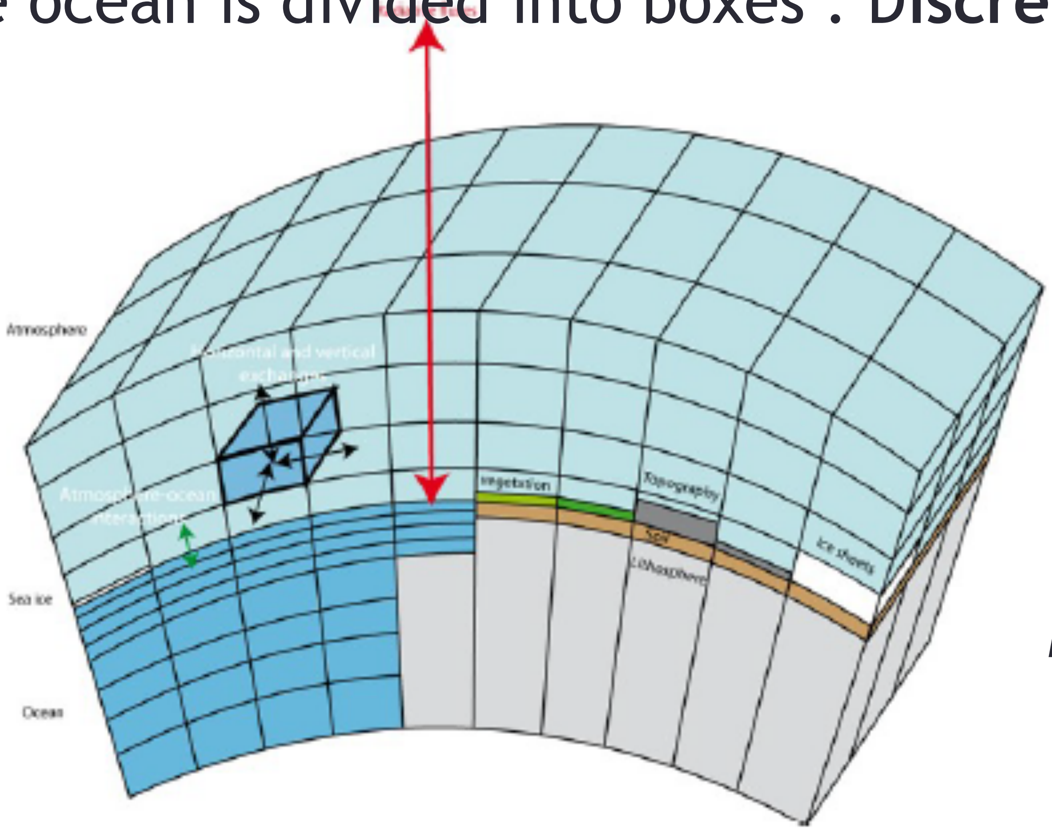
$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} - \frac{f^{(3)}(x)}{3!}\Delta x^2 + \ldots$$

$$O(\Delta x^2)$$

Discretization

<u>For grid-point methods</u>:

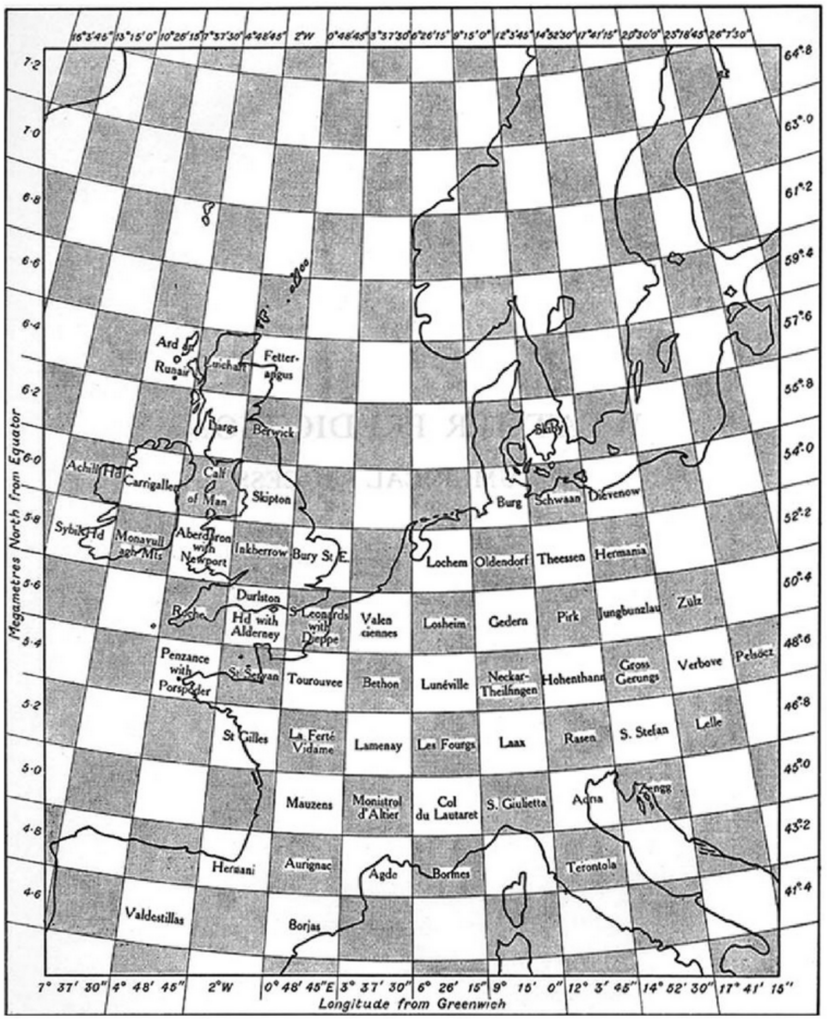
The ocean is divided into boxes : **Discretization**



Example of a finite difference grid

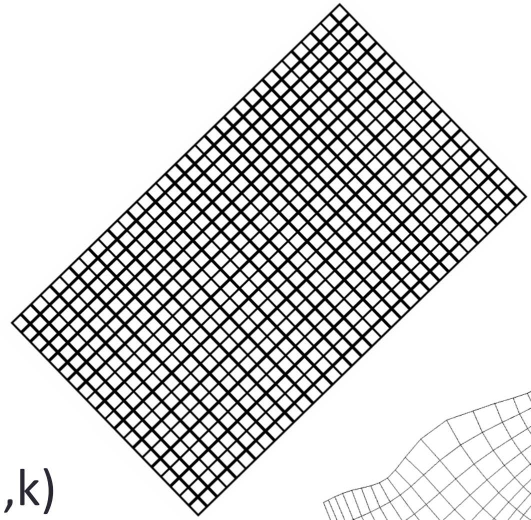
Discretization

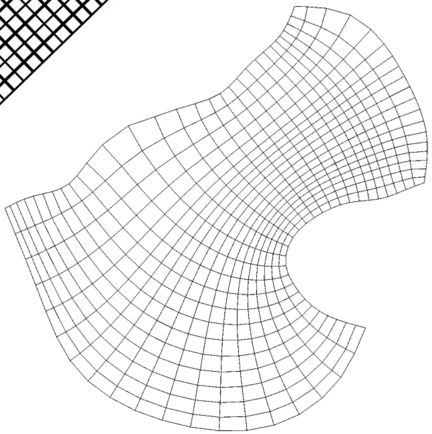
The ocean is divided into boxes : **discretization**



Richardson's 1922 first grid designed for weather prediction

Discretization



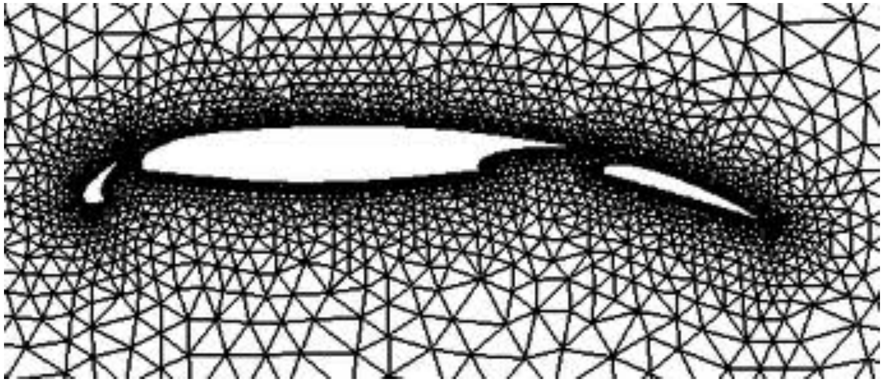


Structured grids

Identified by regular connectivity = can be adressed by (i,j,k)

Unstructured grids

The domain is tiled using more general geometrical shapes (triangles, hexagons …) pieced together to optimally fit details of the geometry.



Discretization

Structured grids

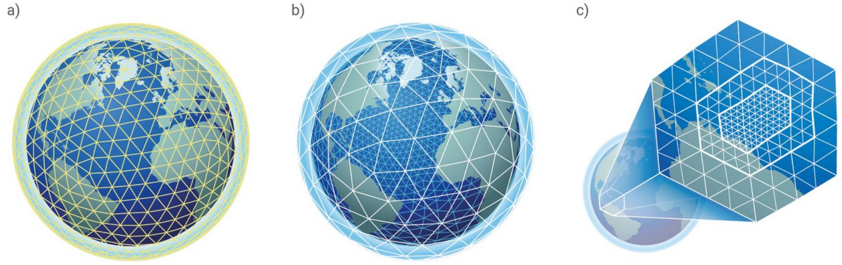
= simple to design / used by most ocean models

Unstructured grids

= Good for tidal modeling / engineering applications.

But problems with geostrophic balance accuracy, conservation and positivity properties, post-processing, etc.

Ex: ICON (https://mpimet.mpg.de/en/research/modeling)



Discretization

Structured grids

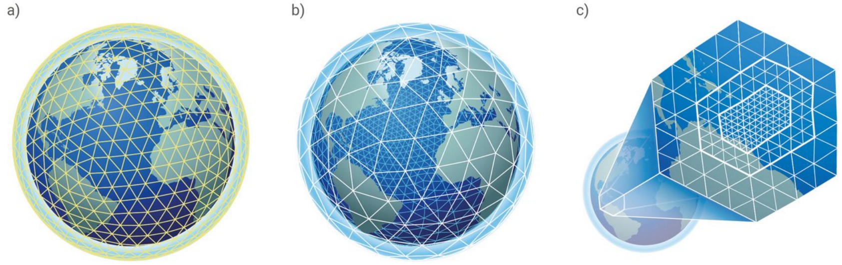
= simple to design / used by most ocean models

Unstructured grids

= Good for tidal modeling / engineering applications.

But problems with geostrophic balance accuracy, conservation and positivity properties, post-processing, etc.

Ex: ICON (https://mpimet.mpg.de/en/research/modeling)



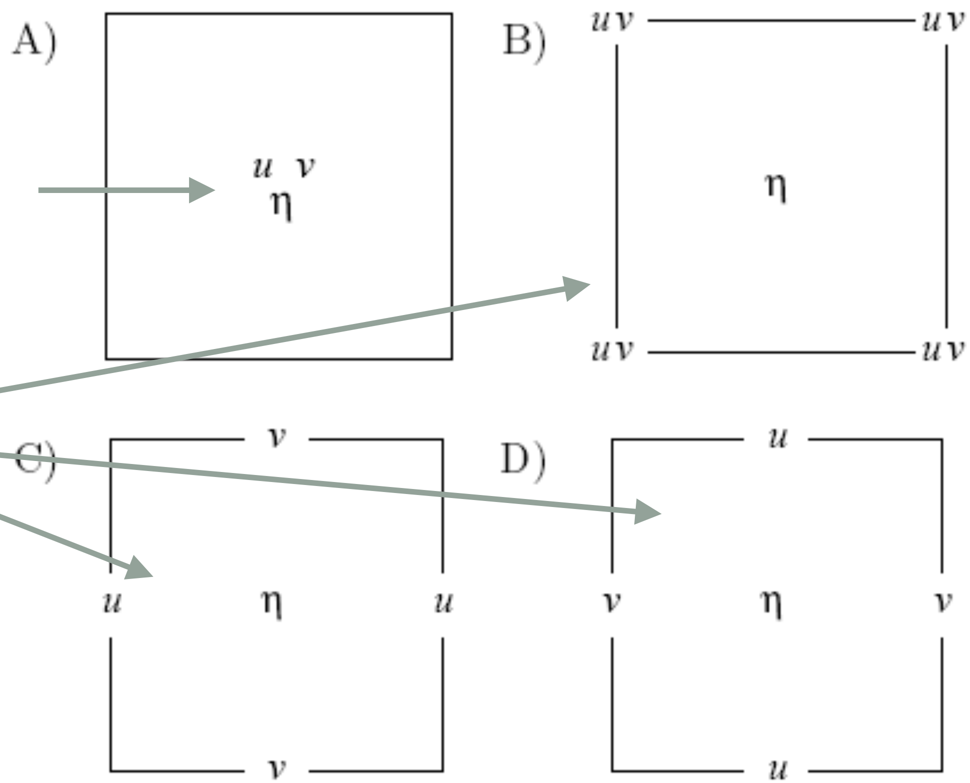
Horizontal discretization

Different types of Horizontal Grids (Arakawa Grids):

Non-staggered
(= collocated variables)

Or

Staggered



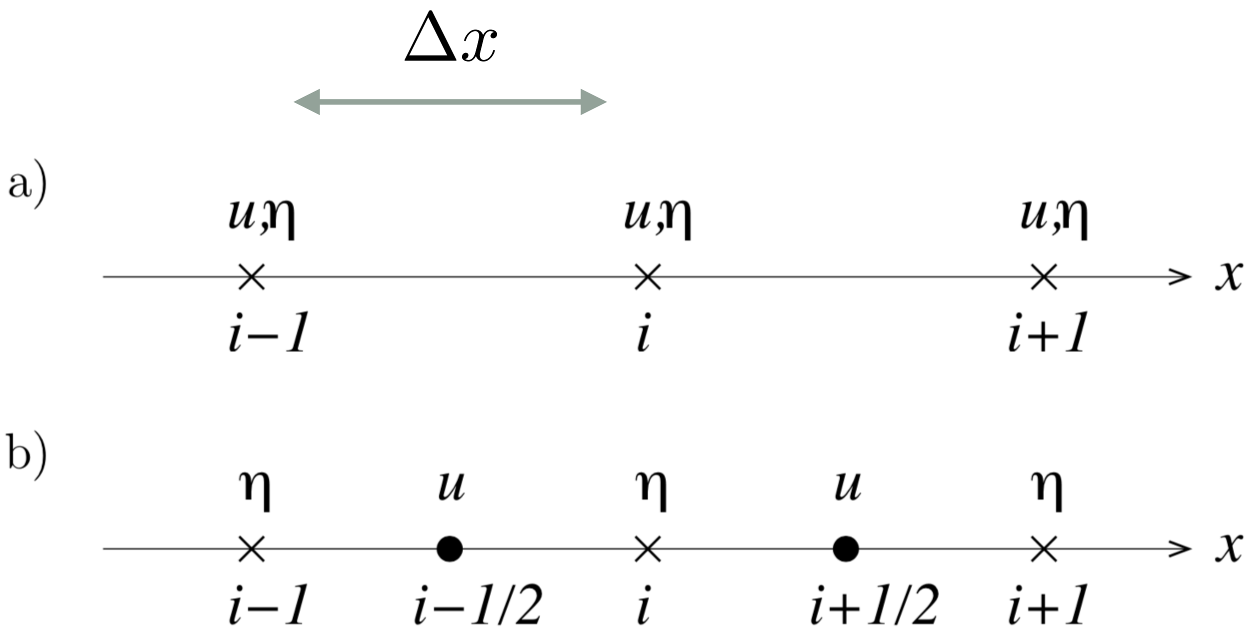
Horizontal discretization

Staggered Vs unstaggered : the 1D problem

Non-staggered

Vs

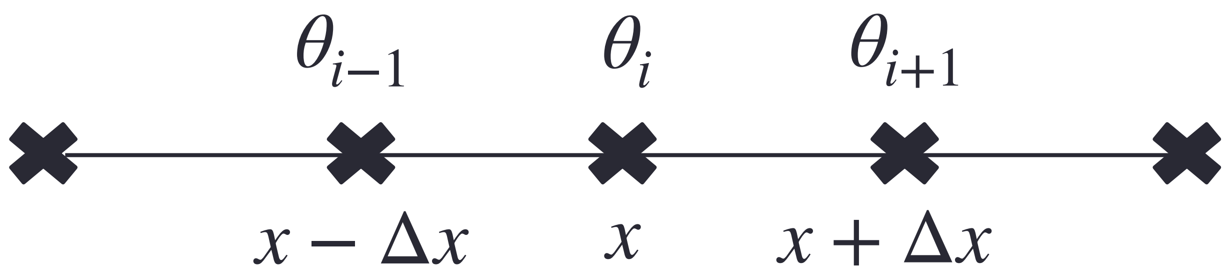
Staggered



Use of staggered or unstaggered grids strongly affects stability, accuracy, and spurious modes.

Horizontal discretization

Example of numerical dispersive errors:



1. The advection equation $\partial_t \theta + c \partial_x \theta = 0$

Solutions of the continuous equations are non-dispersive waves

$\theta(x,t) = \theta_o e^{i(kx - \omega t)}$ *with dispersion relation* $\omega = ck$

Horizontal discretization

Example of numerical dispersion:

1. The advection equation $\partial_t \theta + c\partial_x \theta = 0$

Solutions of the ***continuous equations*** *are non-dispersive waves*

$\theta(x,t) = \theta_o e^{i(kx-\omega t)}$ *with dispersion relation* $\omega = ck$

Discretized equations with the centered second order derivative are

$$d_t \theta + \frac{c}{\Delta x} \delta_i \overline{\theta}^i = 0$$

$$d_t \theta_i + \frac{c}{2\Delta x} \left(\theta_{i+1} - \theta_{i-1}\right) = 0$$

Horizontal discretization

Example of numerical dispersion:

1. The advection equation $\partial_t \theta + c\partial_x \theta = 0$

Substituting in our solution:

$$\theta_i(x,t) = \theta_0 e^{i(kx-\omega t)}$$

$$\theta_{i-1}(x,t) = \theta_0 e^{i(k(x-\Delta x)-\omega t)}$$

$$\theta_{i+1}(x,t) = \theta_0 e^{i(k(x+\Delta x)-\omega t)}$$

Horizontal discretization

Example of numerical dispersion:

1. The advection equation $\partial_t \theta + c\partial_x \theta = 0$

Substituting in the equation:

$$d_t \theta_i + \frac{c}{2\Delta x}\left(\theta_{i+1} - \theta_{i-1}\right) = 0$$

gives

$$\begin{aligned} -i\omega &= -\frac{c}{2\Delta x}\left(e^{ik\Delta x} - e^{-ik\Delta x}\right) \\ &= -\frac{ci}{\Delta x}\sin k\Delta x \end{aligned}$$

Horizontal discretization

Example of numerical dispersion:

1. The advection equation $\partial_t \theta + c\partial_x \theta = 0$

$$\begin{aligned} -i\omega &= -\frac{c}{2\Delta x}\left(e^{ik\Delta x} - e^{-ik\Delta x}\right) \\ &= -\frac{ci}{\Delta x}\sin k\Delta x \end{aligned}$$

Now the solution is **dispersive!!!**

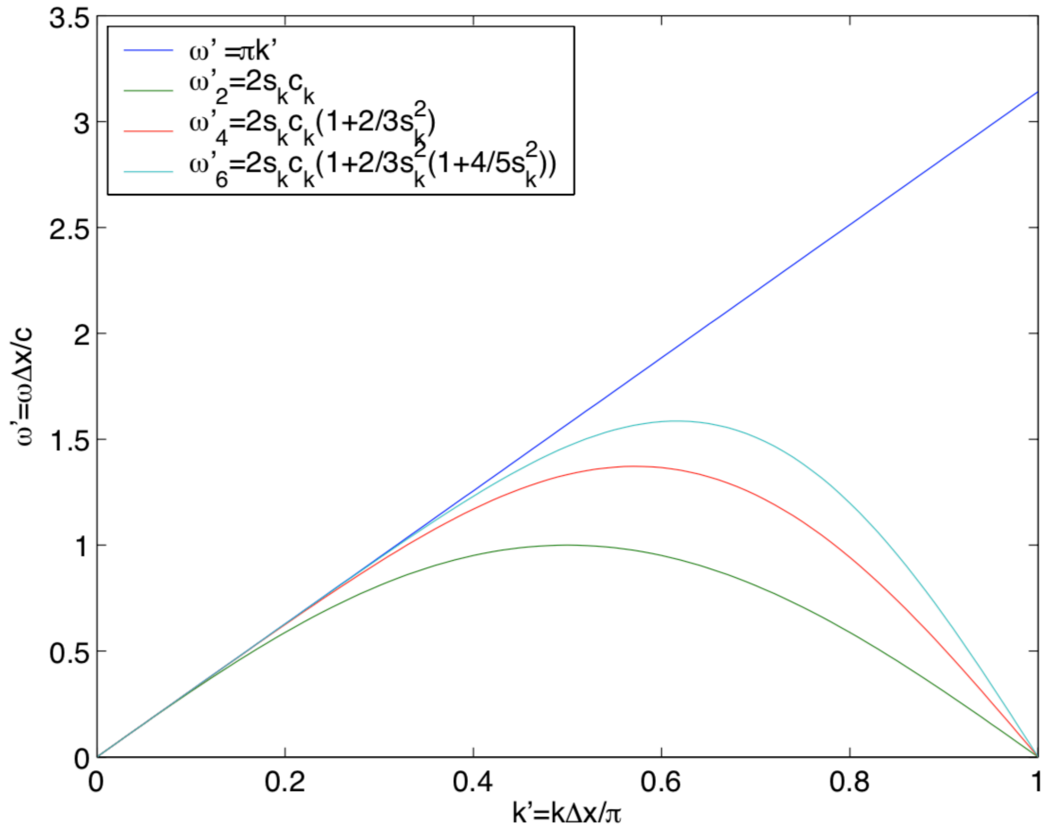
Even it will converge to the non-dispersive solution in the limit of small Δx

$$\omega = \frac{c}{\Delta x}\sin k\Delta x \overset{\Delta x \to 0}{=} ck$$

Horizontal discretization

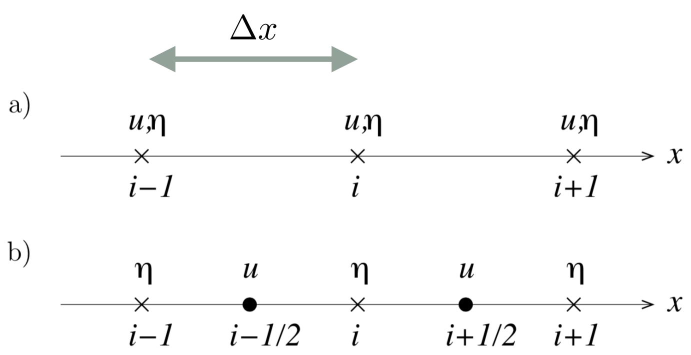
Example of numerical dispersion:

1. The advection equation $\partial_t \theta + c\partial_x \theta = 0$



Dispersion relations for constant flow advection using second, fourth, and sixth order spatial differences.

Horizontal discretization



Staggered Vs unstaggered : the 1D problem

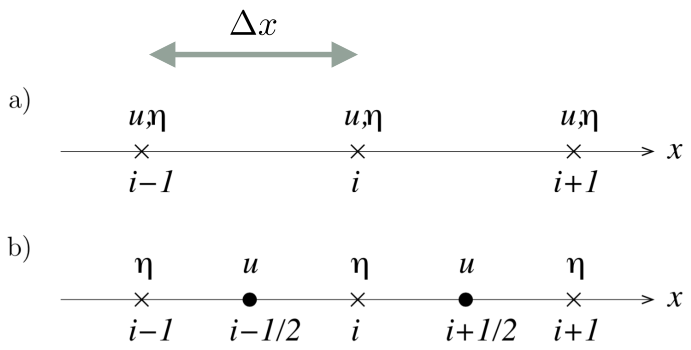
2. Gravity waves

$$\begin{aligned} \partial_t u &= -g\partial_x \eta \\ \partial_t \eta &= -H\partial_x u \end{aligned} \longrightarrow \partial_{tt}\eta = gH\partial_{xx}\eta$$

Solutions of the continuous equations are non-dispersive waves

$\eta = \eta_o e^{i(kx-\omega t)}$ *with dispersion relation* $\omega = \pm\sqrt{gH}k$

Horizontal discretization



Staggered Vs unstaggered : the 1D problem

2. Gravity waves

$$\begin{aligned} \partial_t u &= -g\partial_x \eta \\ \partial_t \eta &= -H\partial_x u \end{aligned} \longrightarrow \partial_{tt}\eta = gH\partial_{xx}\eta$$

Solutions of the continuous equations are non-dispersive waves

$\eta = \eta_o e^{i(kx-\omega t)}$ *with dispersion relation* $\omega = \pm\sqrt{gH}k$

Discretized equations with the centered second order derivative on the **unstaggered grid** are:

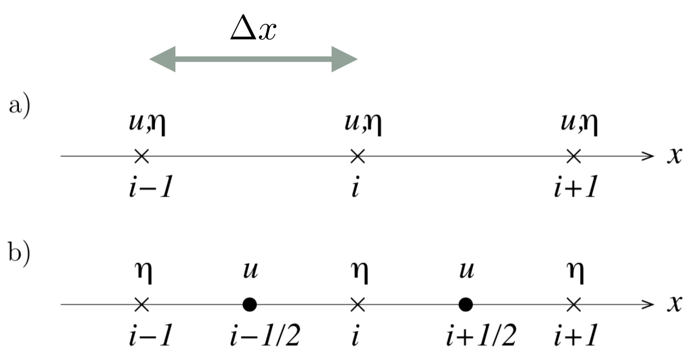
$$\begin{aligned} \partial_t u &= -\frac{g}{\Delta x}\delta_i \overline{\eta}^i \\ \partial_t \eta &= -\frac{H}{\Delta x}\delta_i \overline{u}^i \end{aligned}$$

$$\longrightarrow \partial_{tt}\eta = \frac{gH}{\Delta x^2}\delta_{ii}\overline{\eta}^{ii}$$

with

$$\delta_{ii}\overline{\eta}^{ii} = \frac{1}{4}(\eta_{i-2} - 2\eta_i + \eta_{i+2})$$

Horizontal discretization



Staggered Vs unstaggered : the 1D problem

2. Gravity waves

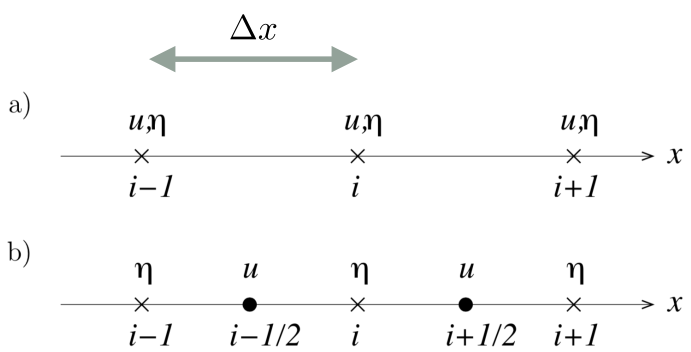
$$\begin{aligned} \partial_t u &= -g\partial_x \eta \\ \partial_t \eta &= -H\partial_x u \end{aligned} \longrightarrow \partial_{tt}\eta = gH\partial_{xx}\eta$$

Substituting in our solution on the unstaggered grid gives :

$$\begin{aligned} -\omega^2 &= \frac{gH}{4\Delta x^2}\left(e^{-i2k\Delta x} - 2 + e^{i2k\Delta x}\right) \\ &= \frac{gH}{4\Delta x^2}\left(2\cos 2k\Delta x - 2\right) \\ &= -\frac{4gH}{\Delta x^2}\sin^2\frac{k\Delta x}{2}\cos^2\frac{k\Delta x}{2} \end{aligned}$$

- Question:
  - What is the dispersion relation on the staggered grid?

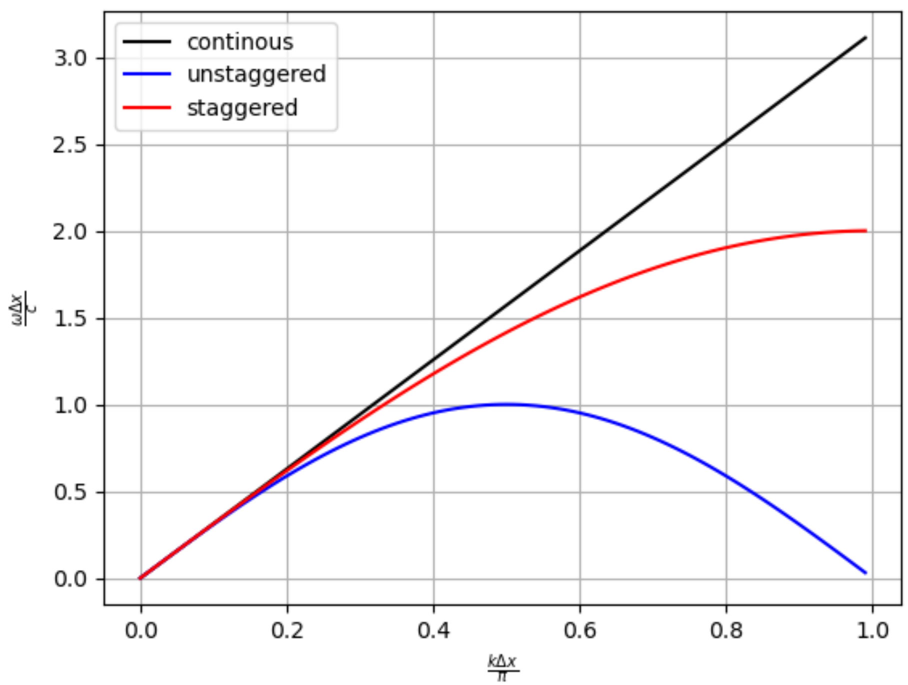
Horizontal discretization



Staggered Vs unstaggered : the 1D problem

2. Gravity waves

$$\begin{aligned}\partial_t u &= -g\partial_x \eta \\ \partial_t \eta &= -H\partial_x u\end{aligned} \longrightarrow \partial_{tt}\eta = gH\partial_{xx}\eta$$



Dispersion of numerical gravity wave for the unstaggered grid (green) and the staggered grid (red). The continuum (= k) is plotted for comparison (blue).

When compared to the continuum we see that the numerical modes are still dispersive on the staggered grid, but:

there is no false extrema, unlike the non-staggered grid,

the group speed is of the correct sign everywhere, even if reduced.

$$v_g = \partial_k \omega$$

Horizontal discretization

Staggered Vs unstaggered : the 1D problem

2. Inertia-Gravity waves

$$\begin{aligned} \partial_t u - fv + g\partial_x \eta &= 0 \\ \partial_t v + fu &= 0 \\ \partial_t \eta + H\partial_x u &= 0 \end{aligned}$$

Solutions of the continuous equations are waves following the dispersion relation:

$$\left| \begin{pmatrix} -i\omega & -f & gik \\ f & -i\omega & 0 \\ Hik & 0 & -i\omega \end{pmatrix} \right| = 0 \Rightarrow \begin{cases} \omega = 0 \\ \omega^2 = f^2 + gHk^2 \end{cases}$$

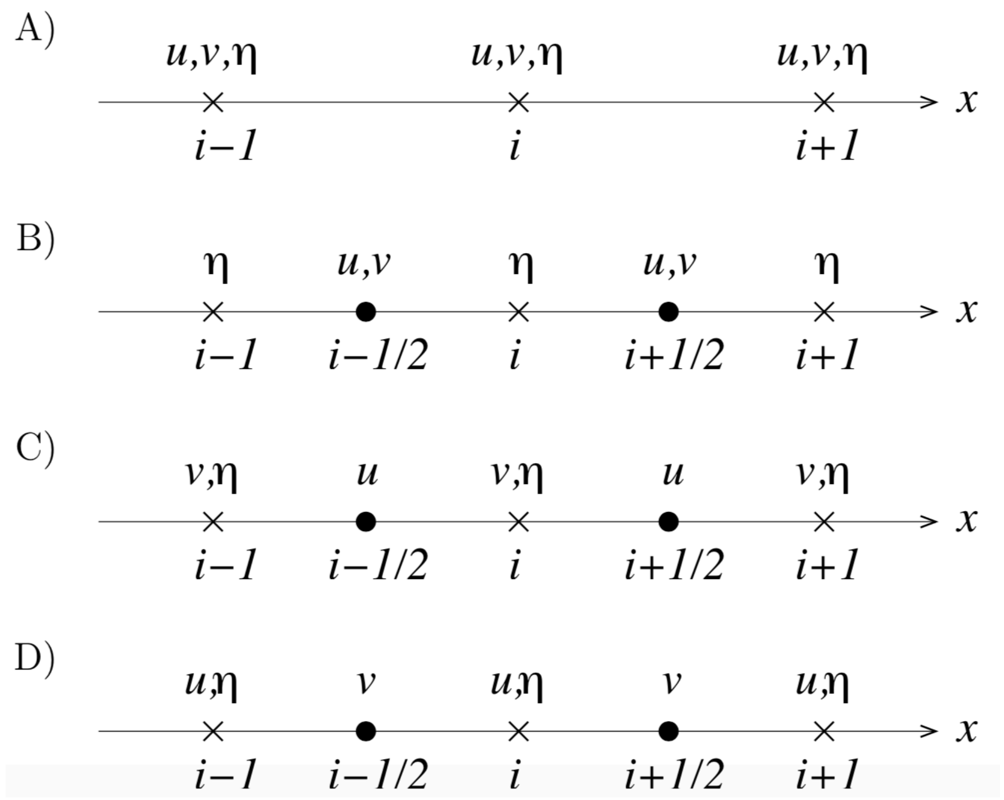
Horizontal discretization

Staggered Vs unstaggered : the 1D problem

2. Inertia-Gravity waves

$$
\begin{aligned}
\partial_t u - f v + g \partial_x \eta &= 0 \\
\partial_t v + f u &= 0 \\
\partial_t \eta + H \partial_x u &= 0
\end{aligned}
$$

Now, 4 different grids are possible:



Horizontal discretization

Staggered Vs unstaggered : the 1D problem

2. Inertia-Gravity waves

- A-grid model

$$
\begin{aligned}
\partial_t u - f v + \frac{g}{\Delta x}\delta_i \overline{\eta}^i &= 0 \\
\partial_t v + f u &= 0 \\
\partial_t \eta + \frac{H}{\Delta x}\delta_i \overline{u}^i &= 0
\end{aligned}
$$

- B-grid model

$$
\begin{aligned}
\partial_t u - f v + \frac{g}{\Delta x}\delta_i \eta &= 0 \\
\partial_t v + f u &= 0 \\
\partial_t \eta + \frac{H}{\Delta x}\delta_i u &= 0
\end{aligned}
$$

- C-grid model

$$
\begin{aligned}
\partial_t u - f \overline{v}^i + \frac{g}{\Delta x}\delta_i \eta &= 0 \\
\partial_t v + f \overline{u}^i &= 0 \\
\partial_t \eta + \frac{H}{\Delta x}\delta_i u &= 0
\end{aligned}
$$

- D-grid model

$$
\begin{aligned}
\partial_t u - f \overline{v}^i + \frac{g}{\Delta x}\delta_i \overline{\eta}^i &= 0 \\
\partial_t v + f \overline{u}^i &= 0 \\
\partial_t \eta + \frac{H}{\Delta x}\delta_i \overline{u}^i &= 0
\end{aligned}
$$

Horizontal discretization

Staggered Vs unstaggered : the 1D problem

2. Inertia-Gravity waves

The corresponding dispersion relations are :

A: $$\frac{\omega^2}{f^2} = 1 + \frac{4L_d^2}{\Delta x^2} s_k^2 c_k^2$$

B: $$\frac{\omega^2}{f^2} = 1 + \frac{4L_d^2}{\Delta x^2} s_k^2$$

C: $$\frac{\omega^2}{f^2} = c_k^2 + \frac{4L_d^2}{\Delta x^2} s_k^2$$

D: $$\frac{\omega^2}{f^2} = c_k^2 + \frac{4L_d^2}{\Delta x^2} s_k^2 c_k^2$$

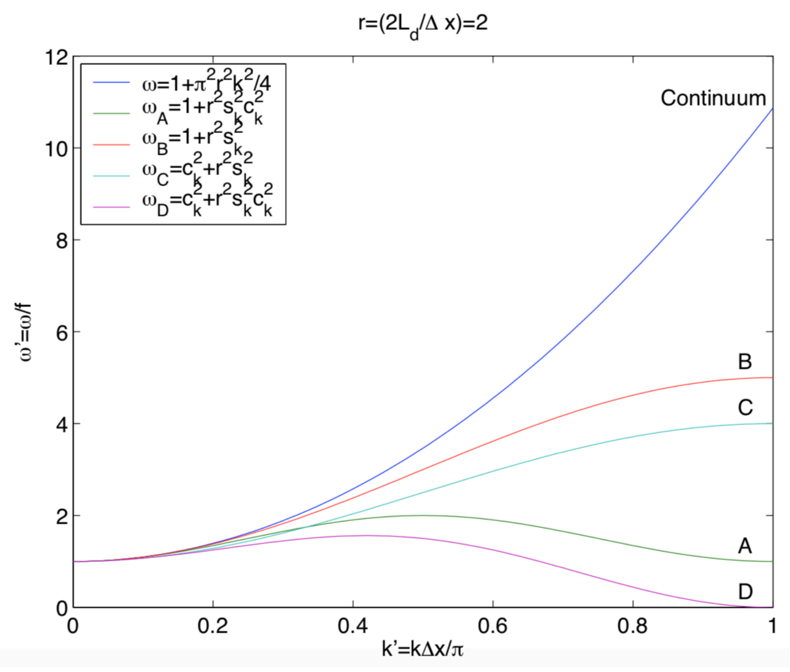
$$s_k = \sin \frac{k\Delta x}{2} \qquad c_k = \cos \frac{k\Delta x}{2}$$

$$L_d = \sqrt{gH}/f$$

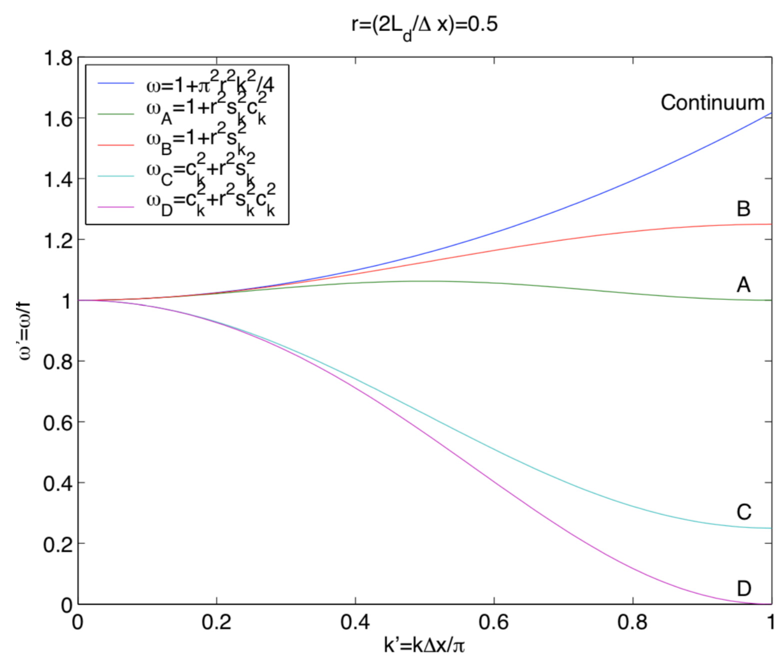
Horizontal discretization

Staggered Vs unstaggered : the 1D problem

2. Inertia-Gravity waves



deformation radius is resolved

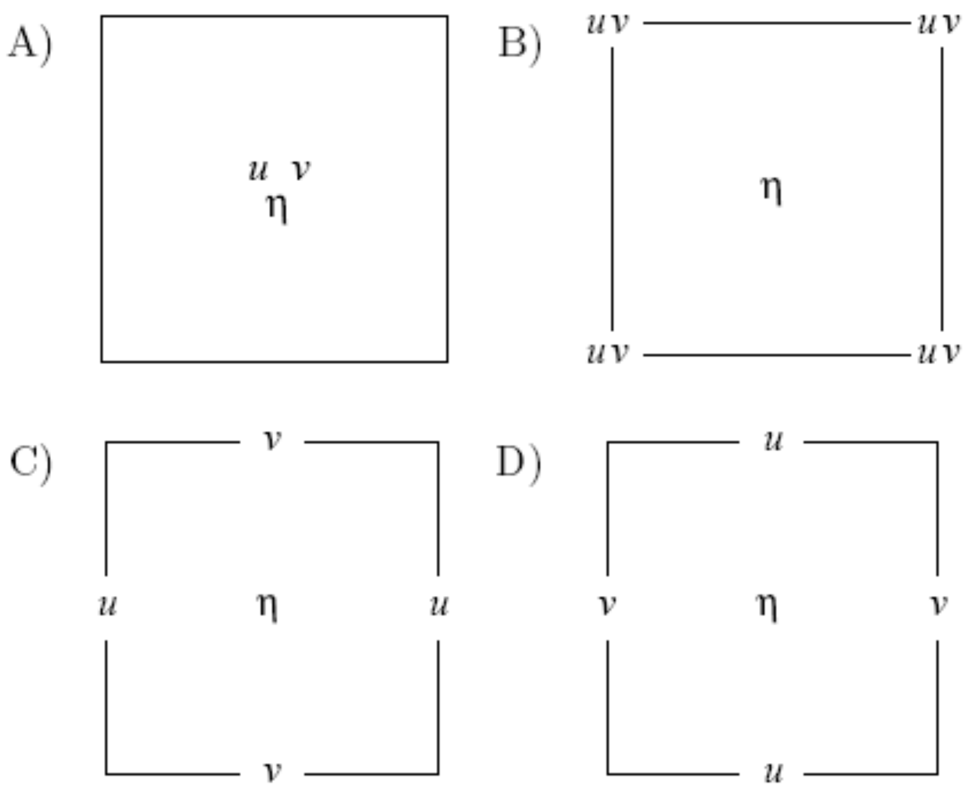


deformation radius is not resolved

Staggering variables in the form of the B grid is most likely to avoid computational modes when solving one-dimensional shallow water equations.

Horizontal discretization

Horizontal Arakawa Grids:

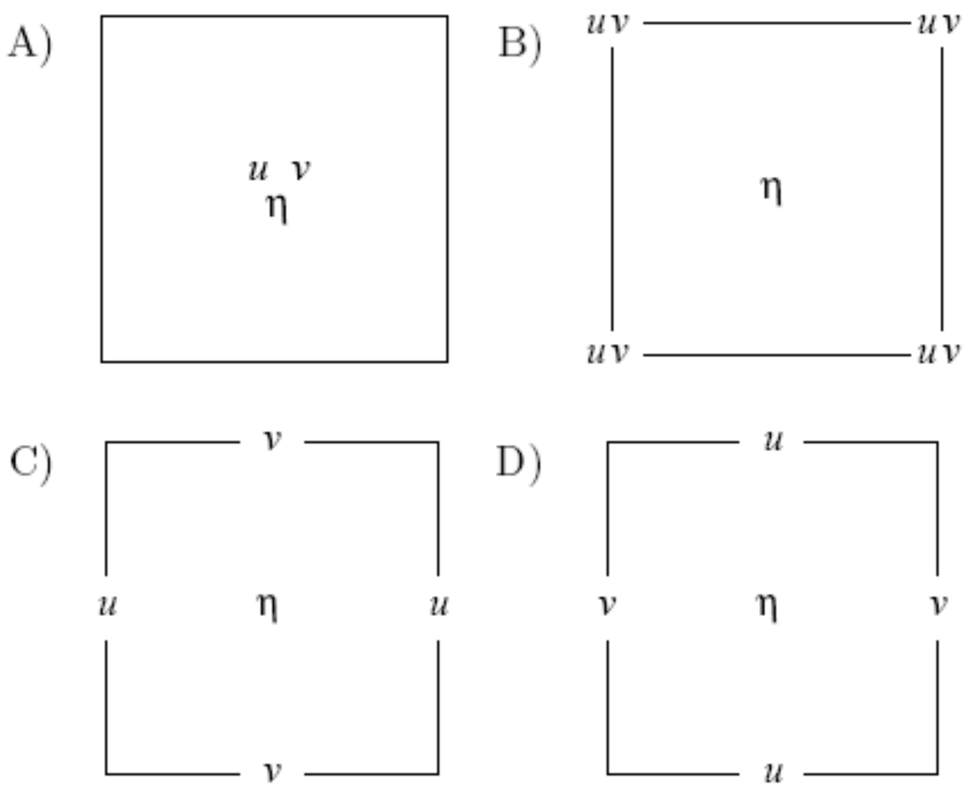


Linear shallow water equation:

$$
\begin{aligned}
\partial_t u - fv + \frac{g}{\Delta x}\delta_i \eta &= 0 \\
\partial_t v + fu + \frac{g}{\Delta y}\delta_j \eta &= 0 \\
\partial_t \eta + \frac{H}{\Delta x}\delta_i u + \frac{H}{\Delta y}\delta_y \eta &= 0
\end{aligned}
$$

Horizontal discretization

Horizontal Arakawa Grids:

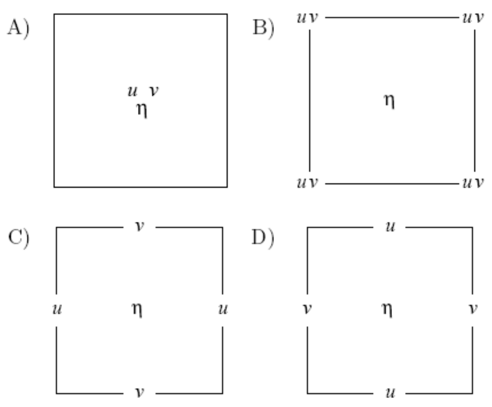


Linear shallow water equation:

$$
\begin{aligned}
\partial_t u - f v + \frac{g}{\Delta x}\delta_i \eta &= 0 \\
\partial_t v + f u + \frac{g}{\Delta y}\delta_j \eta &= 0 \\
\partial_t \eta + \frac{H}{\Delta x}\delta_i u + \frac{H}{\Delta y}\delta_y \eta &= 0
\end{aligned}
$$

- Question:
  - Which grid minimises the number of averaging between points when solving linear SW equations in 2d?

Horizontal discretization



Response of each operator:

$$
\begin{aligned}
R(\delta_i \phi) &= 2i \sin\frac{k\Delta x}{2} = 2i s_k \\
R(\delta_j \phi) &= 2i \sin\frac{l\Delta y}{2} = 2i s_l \\
R(\overline{\phi}^i) &= \cos\frac{k\Delta x}{2} = c_k \\
R(\overline{\phi}^j) &= \cos\frac{l\Delta y}{2} = c_l
\end{aligned}
$$

Dispersion relations:

- A grid:

$$
\begin{aligned}
\omega^2 &= f^2 + \frac{4gH}{\Delta x^2} s_k^2 c_k^2 + \frac{4gH}{\Delta y^2} s_l^2 c_l^2 \\
\text{or} \qquad \left(\frac{\omega}{f}\right)^2 &= 1 + r_x^2 s_k^2 c_k^2 + r_y^2 s_l^2 c_l^2
\end{aligned}
$$

- B grid:

$$
\begin{aligned}
\omega^2 &= f^2 + \frac{4gH}{\Delta x^2} s_k^2 c_l^2 + \frac{4gH}{\Delta y^2} s_l^2 c_k^2 \\
\text{or} \qquad \left(\frac{\omega}{f}\right)^2 &= 1 + r_x^2 s_k^2 c_l^2 + r_y^2 s_l^2 c_k^2
\end{aligned}
$$

- C grid:

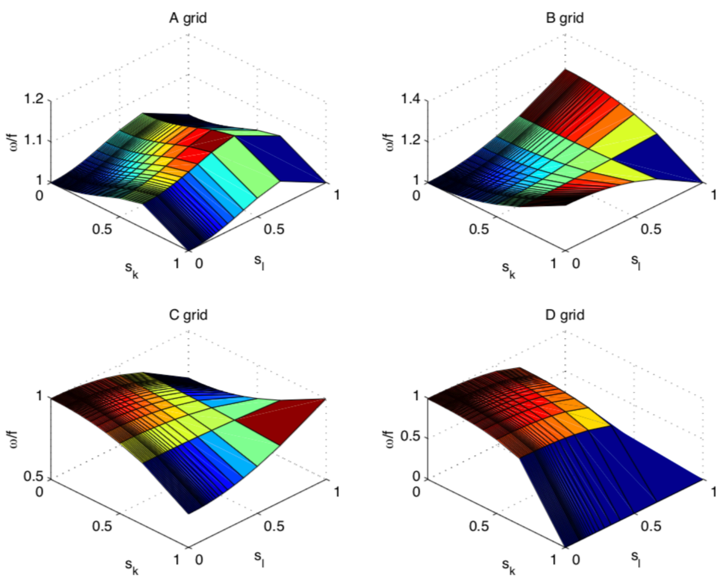
$$
\begin{aligned}
\omega^2 &= f^2 c_k^2 c_l^2 + \frac{4gH}{\Delta x^2} s_k^2 + \frac{4gH}{\Delta y^2} s_l^2 \\
\text{or} \qquad \left(\frac{\omega}{f}\right)^2 &= c_k^2 c_l^2 + r_x^2 s_k^2 + r_y^2 s_l^2
\end{aligned}
$$

- D grid:

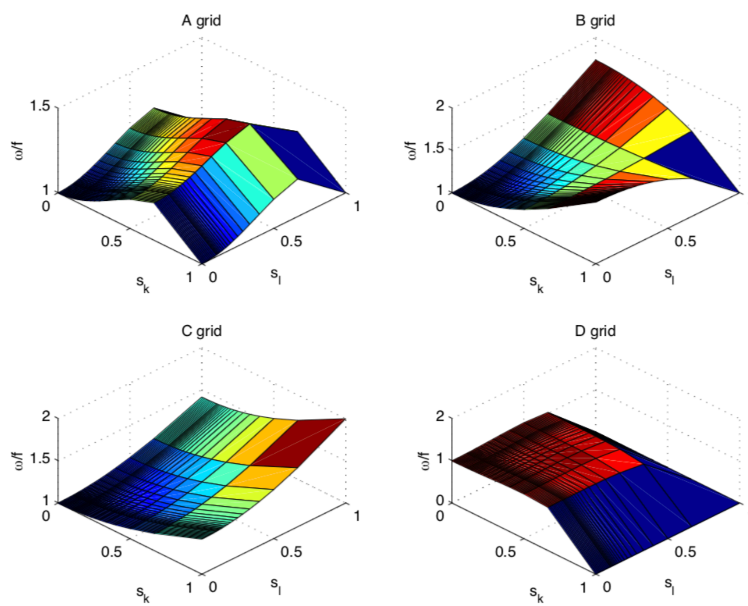
$$
\begin{aligned}
\omega^2 &= f^2 c_k^2 c_l^2 + \frac{4gH}{\Delta x^2} s_k^2 c_k^2 c_l^2 + \frac{4gH}{\Delta y^2} s_l^2 c_k^2 c_l^2 \\
\text{or} \qquad \left(\frac{\omega}{f}\right)^2 &= (1 + r_x^2 s_k^2 + r_y^2 s_l^2) c_k^2 c_l^2
\end{aligned}
$$

Horizontal discretization

Coarse resolution:



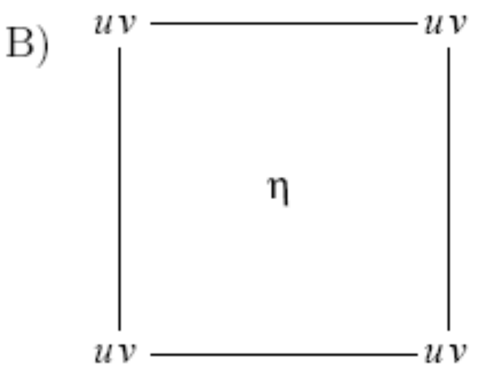
High resolution:



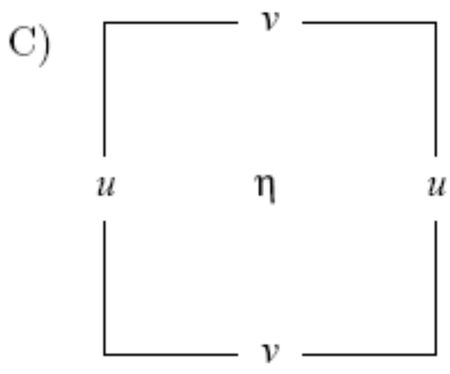
D is always bad.
B underestimates frequency for short two-dimensional waves
C is the only grid with monotonically increasing frequency (i.e. right sign of group velocity) at high res.

Horizontal discretization

- B grid is prefered at coarse resolution, when Coriolis is important:
  - Superior for poorly resolved inertia-gravity waves.
  - Good for Rossby waves: collocation of velocity points.
  - Bad for gravity waves: computational checkerboard mode



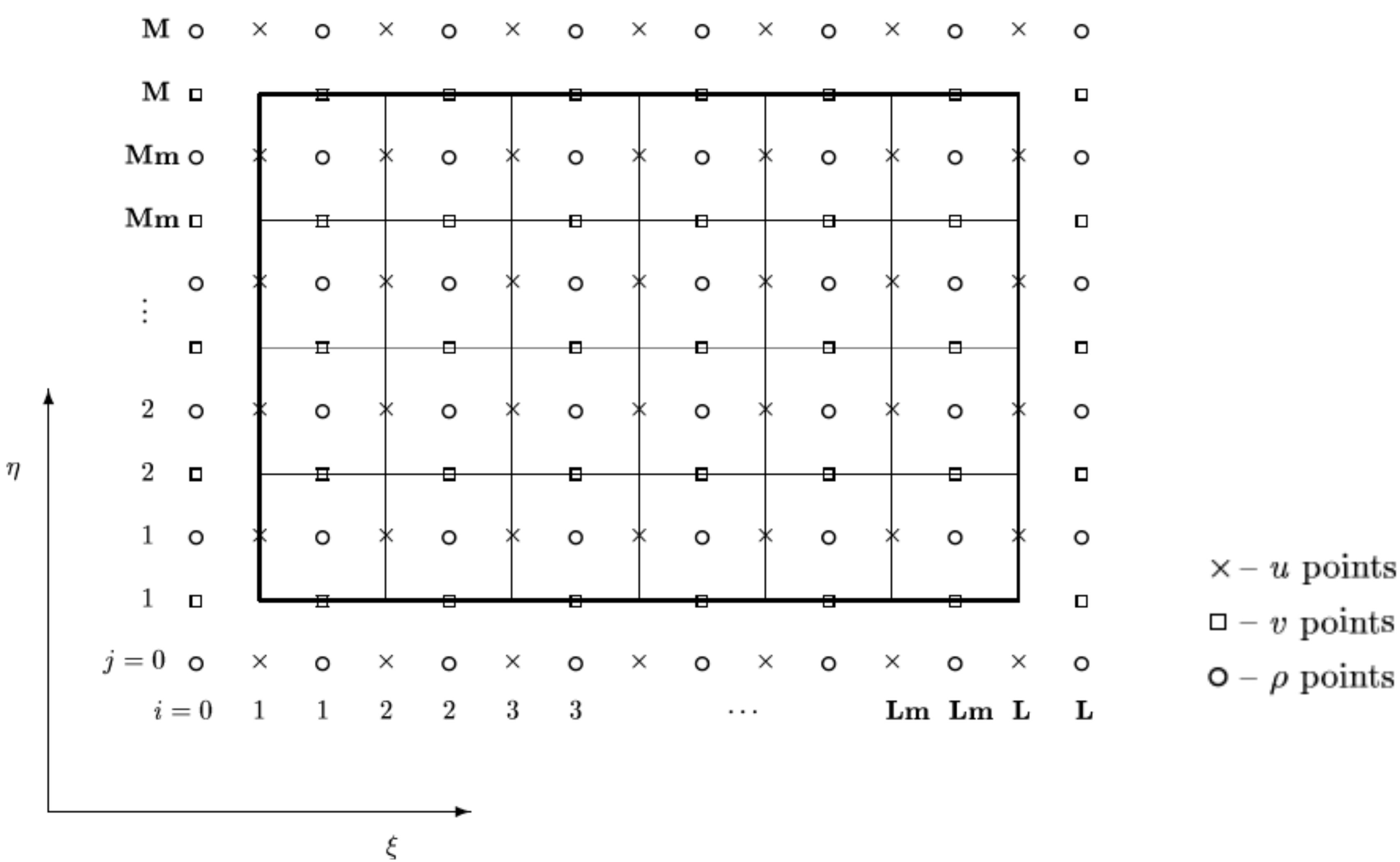
- C grid is prefered at fine resolution, when Coriolis is less important
  - Superior for gravity waves.
  - Good for well resolved inertia-gravity waves.
  - Bad for poorly resolved waves: Rossby waves (computational checkerboard mode) and inertia-gravity waves due to averaging the Coriolis force.



CROCO

Horizontal discretization

CROCO: Arakawa C-grid



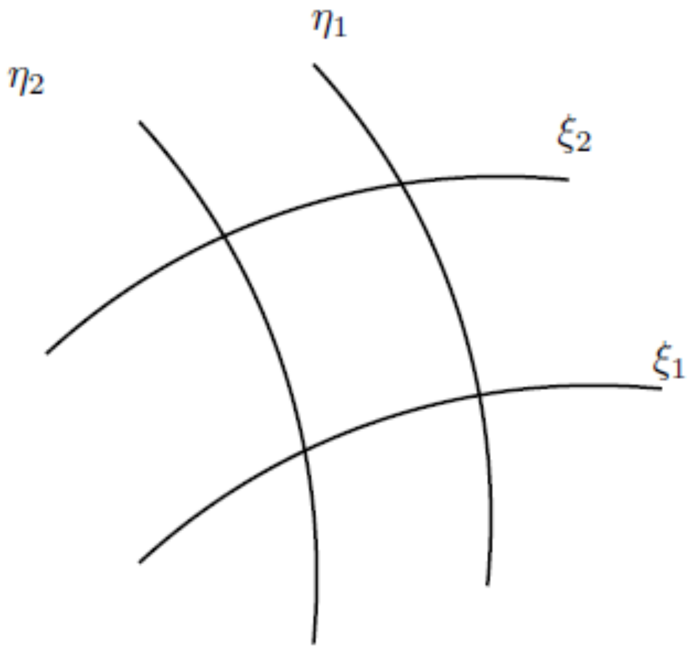
Horizontal curvilinear grid

- **CROCO:** is formulated in general horizontal curvilinear coordinates:

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

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

m, n: scale factors relating the differential distances to the physical arc lengths



$$\vec{v} \cdot \hat{\xi} = u$$

$$\vec{v} \cdot \hat{\eta} = v$$

Horizontal curvilinear grid

- **CROCO:** is formulated in general horizontal curvilinear coordinates:

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

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

With classical formulas for div, grad, curl and lap in curvilinear coordinates:

$$\nabla\phi = \hat{\xi} m \frac{\partial\phi}{\partial\xi} + \hat{\eta} n \frac{\partial\phi}{\partial\eta}$$

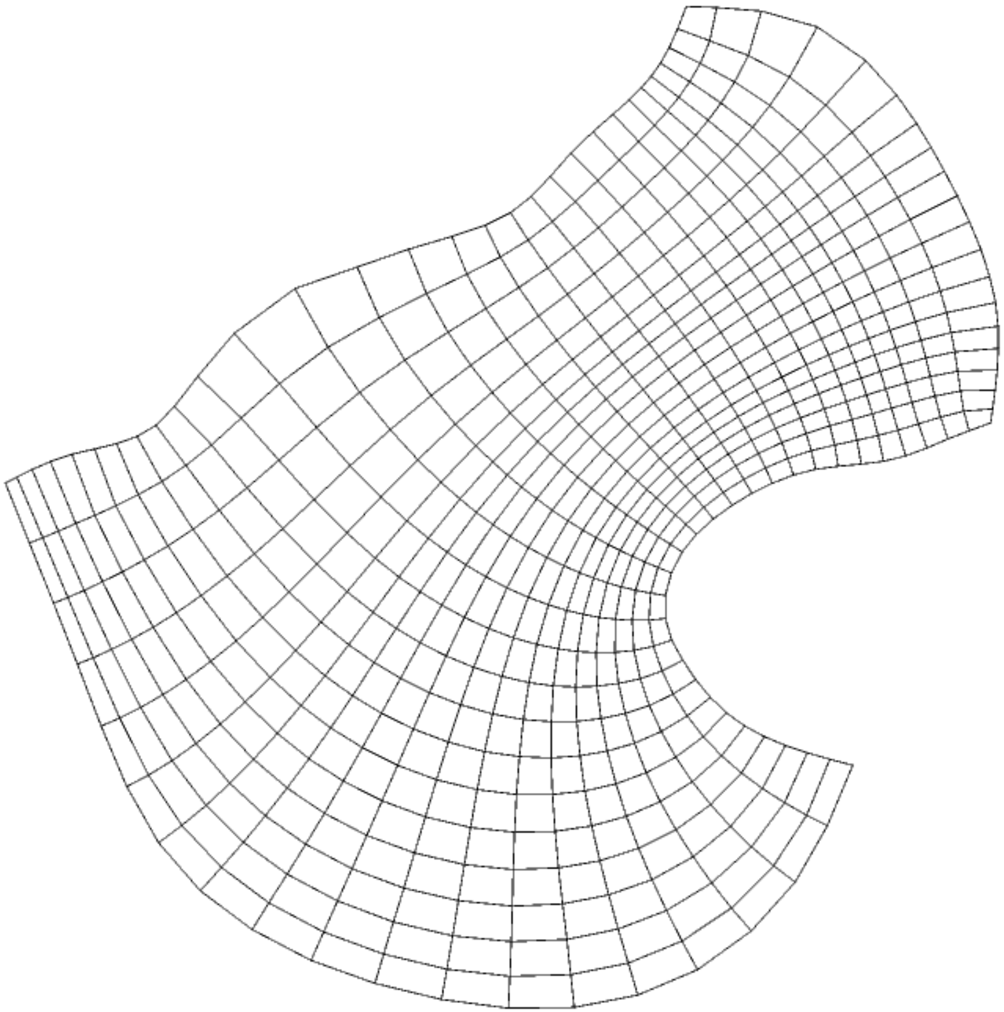
$$\nabla\cdot\vec{a} = mn\left[\frac{\partial}{\partial\xi}\left(\frac{a}{n}\right) + \frac{\partial}{\partial\eta}\left(\frac{b}{m}\right)\right]$$

$$\nabla\times\vec{a} = mn\begin{vmatrix} \frac{\hat{\xi}_1}{m} & \frac{\hat{\xi}_2}{n} & \hat{k} \\ \frac{\partial}{\partial\xi} & \frac{\partial}{\partial\eta} & \frac{\partial}{\partial z} \\ \frac{a}{m} & \frac{b}{n} & c \end{vmatrix}$$

$$\nabla^2\phi = \nabla\cdot\nabla\phi = mn\left[\frac{\partial}{\partial\xi}\left(\frac{m}{n}\frac{\partial\phi}{\partial\xi}\right) + \frac{\partial}{\partial\eta}\left(\frac{n}{m}\frac{\partial\phi}{\partial\eta}\right)\right]$$

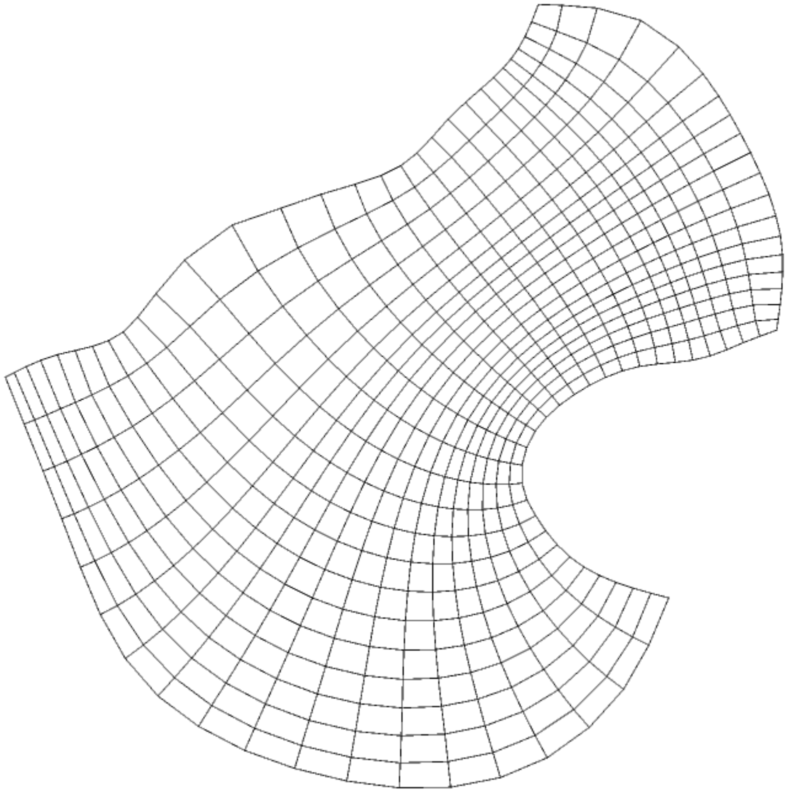
Horizontal curvilinear grid

- This is a possible gr



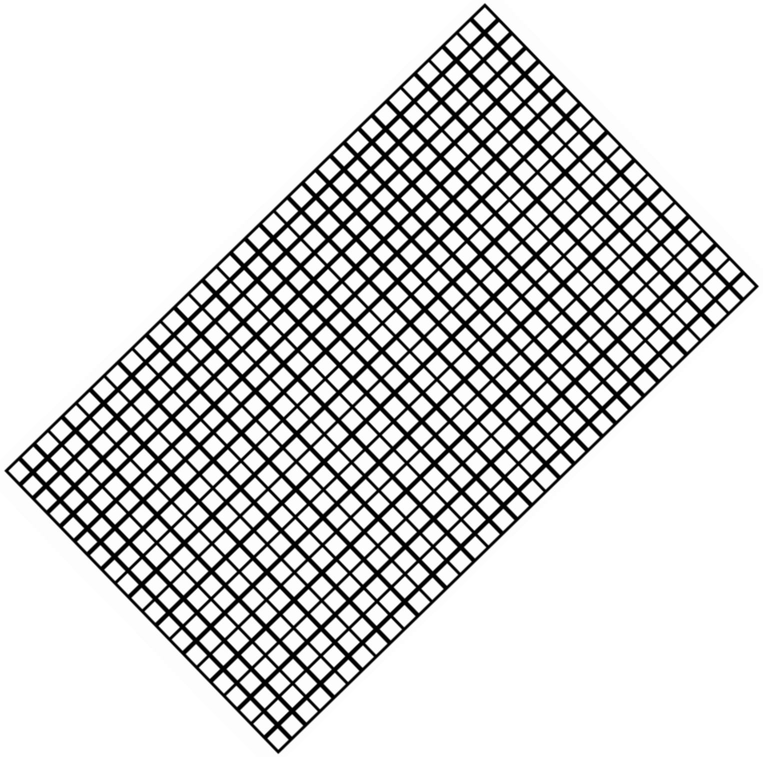
Horizontal curvilinear grid

- This is a possible grid:



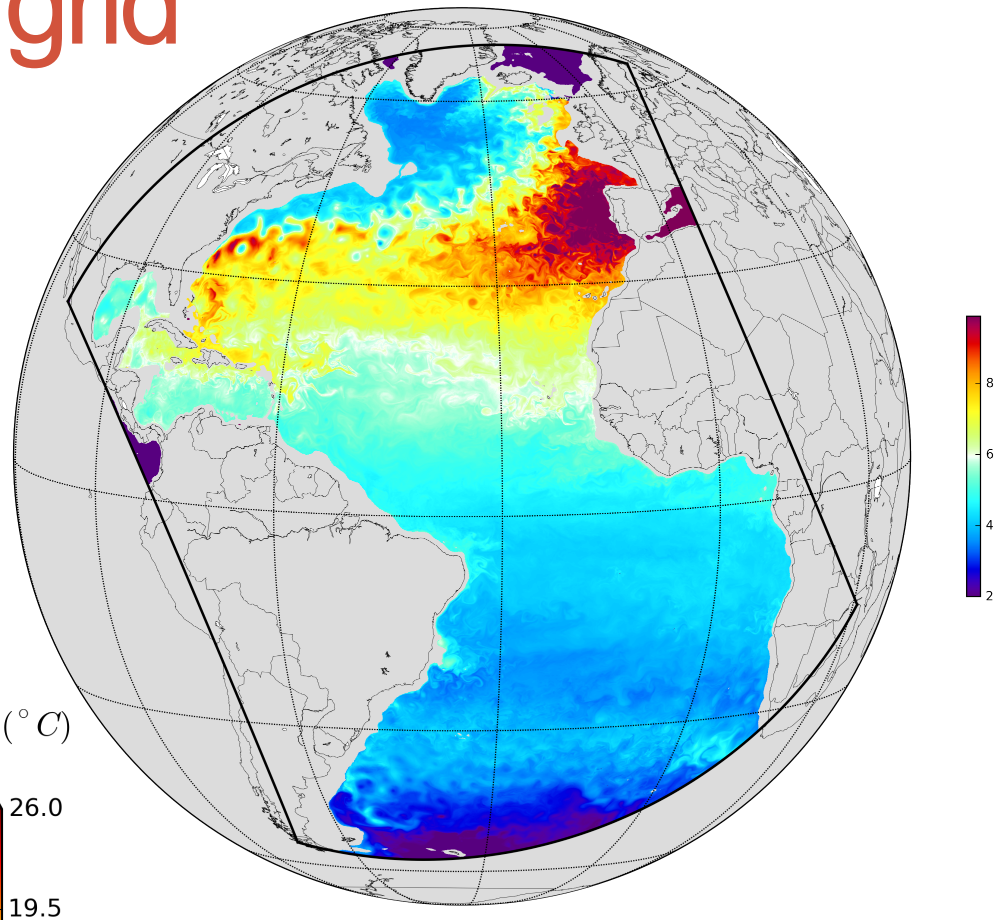
In practice variations in *dx* and *dy* should be minimized to minimize errors and optimize computation time.

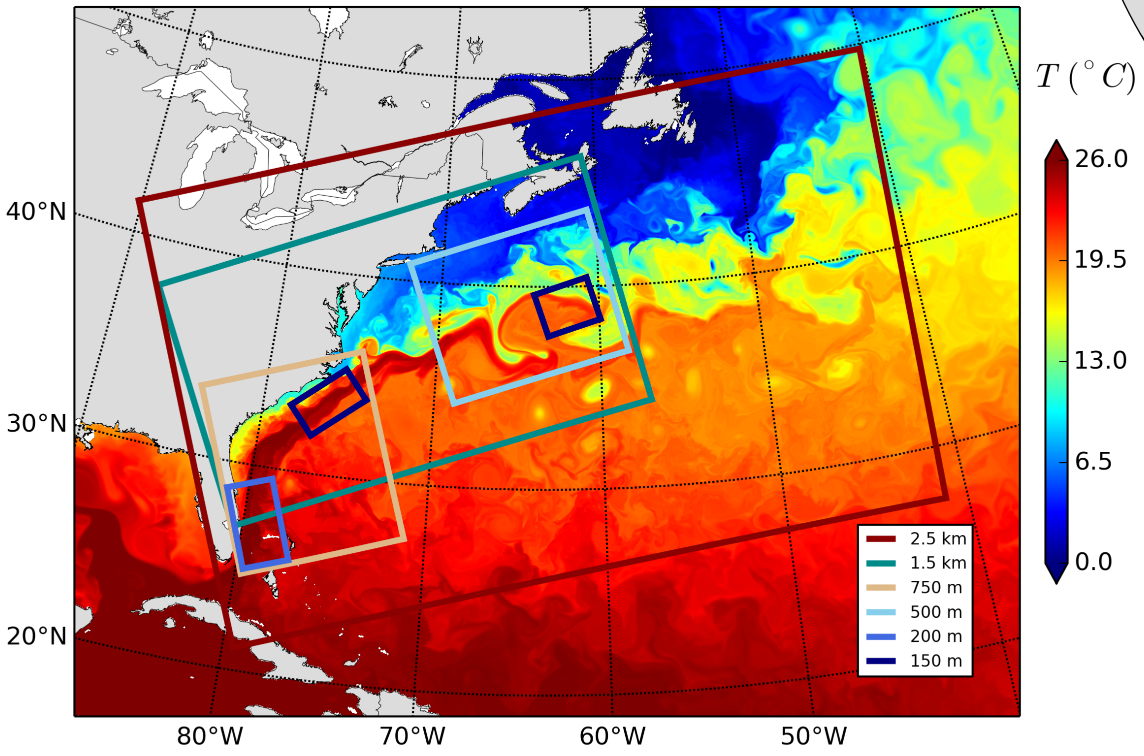
So avoir extreme distorsions and be as close as rectangular grids as possible (+ use land masks)



Horizontal curvilinear grid

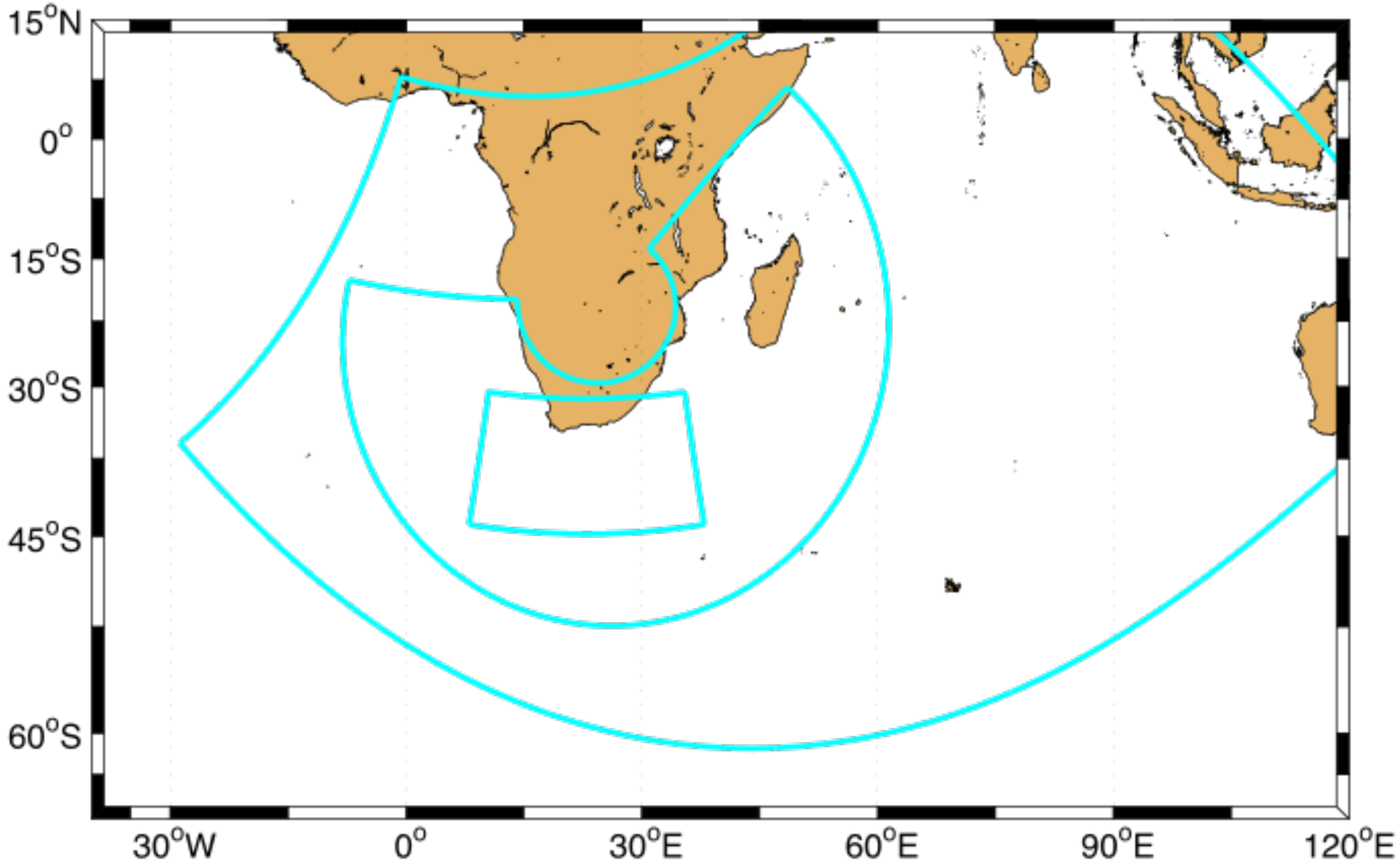
- Example of realistic domains:





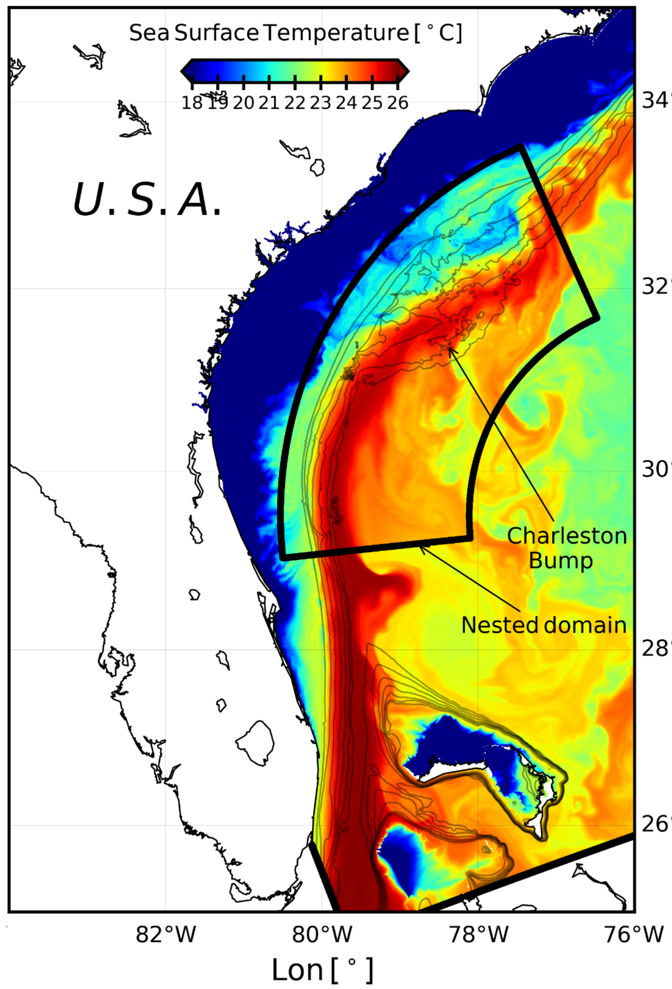
Horizontal curvilinear grid

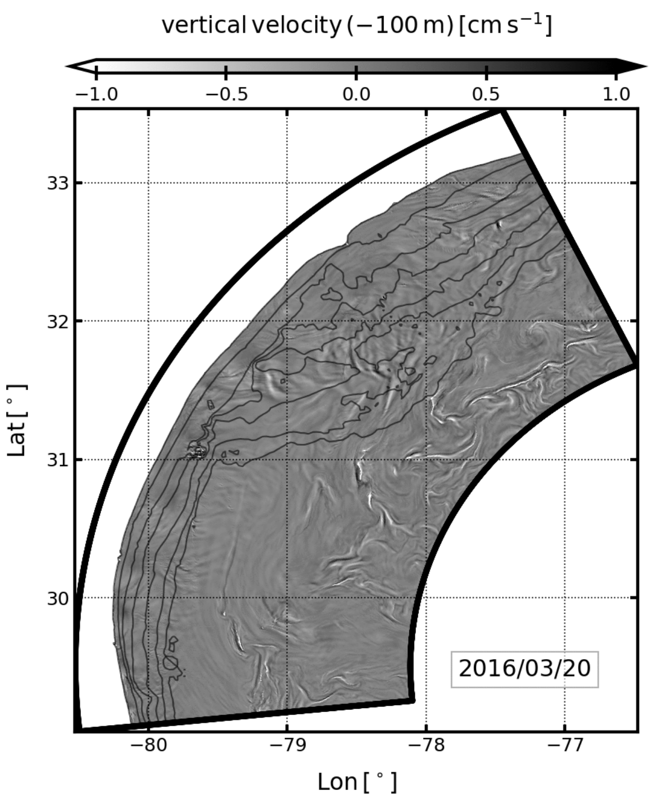
- Example of realistic domains (with gentle bendings):



Horizontal curvilinear grid

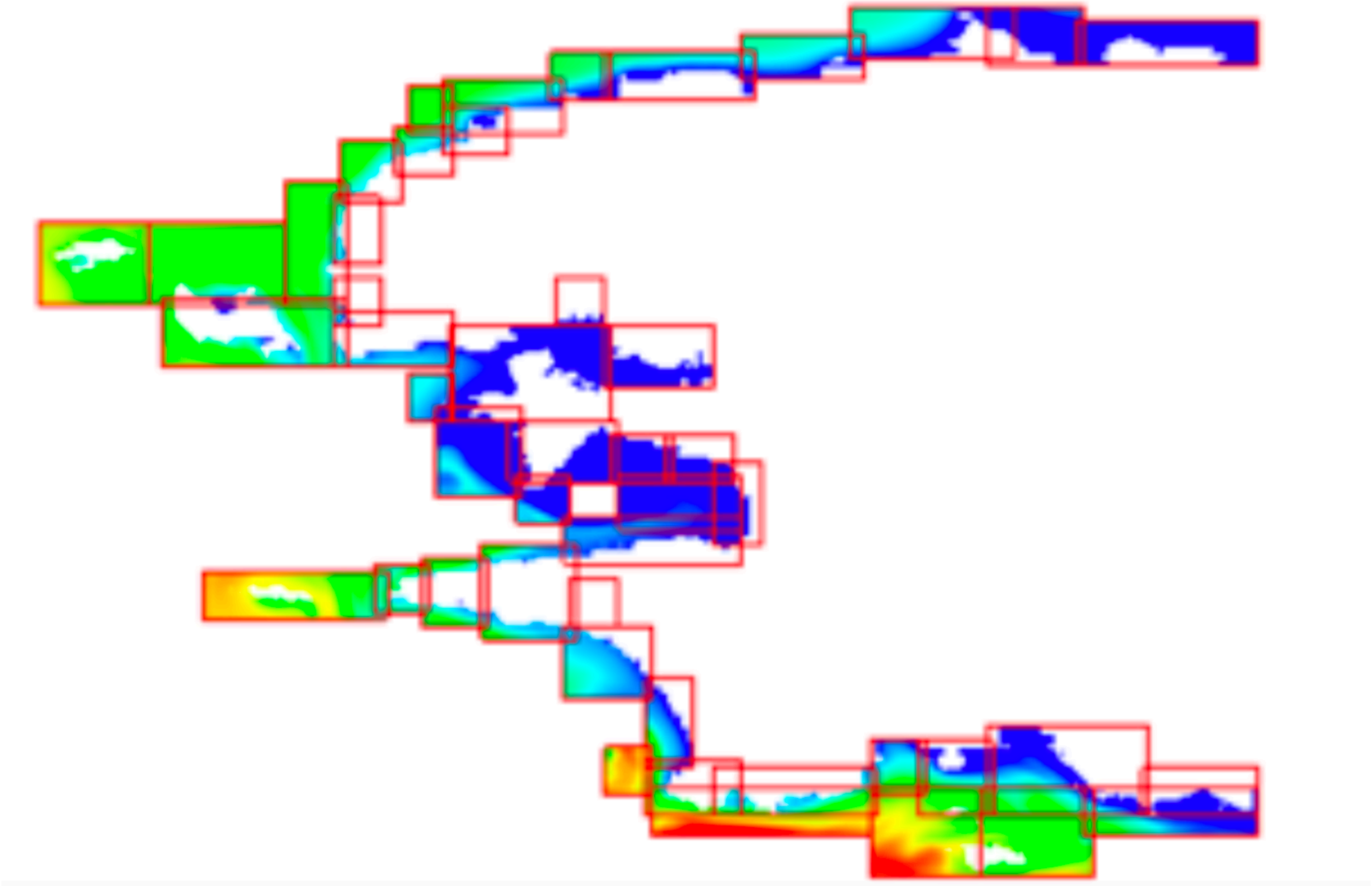
- Example of realistic domains (with gentle bendings):





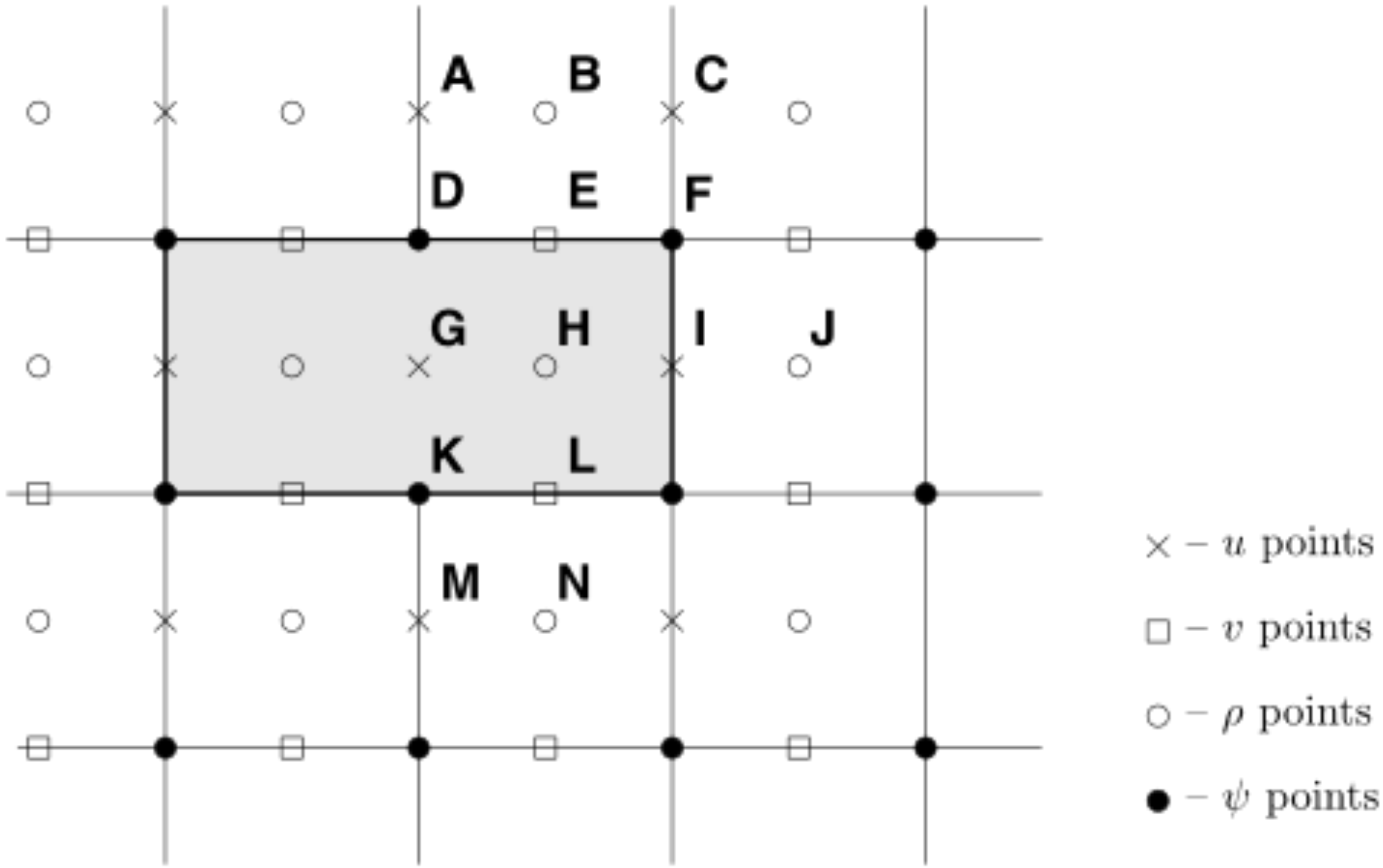
Horizontal curvilinear grid

- Another method = massive multigrain



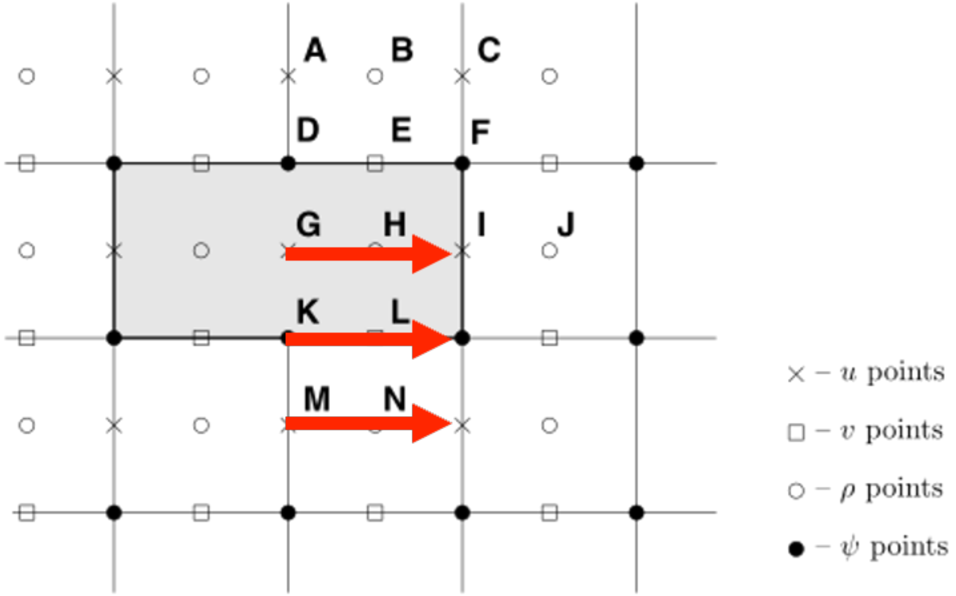
Land/sea Mask

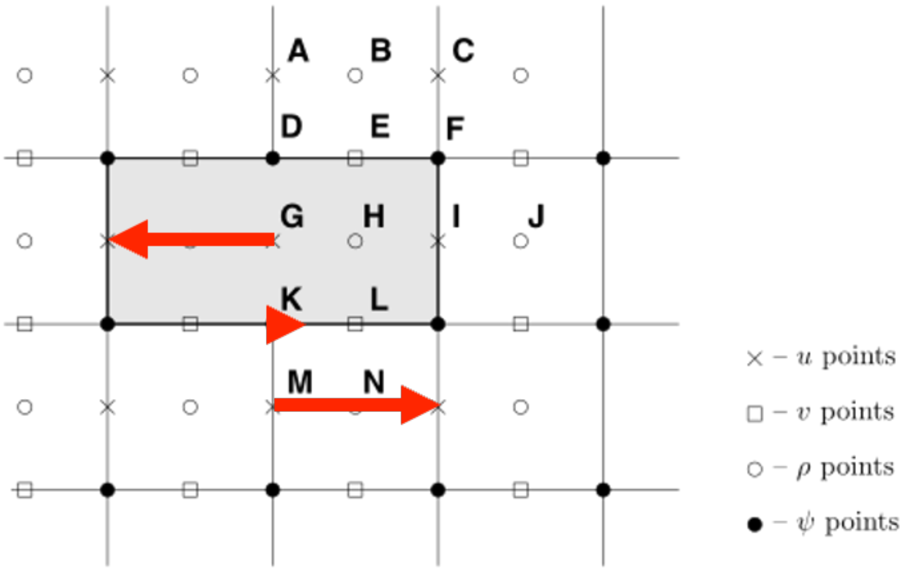
Variables within the masked region are set to zero by multiplying by the mask for either the u, v or rho points :



Land/sea Mask

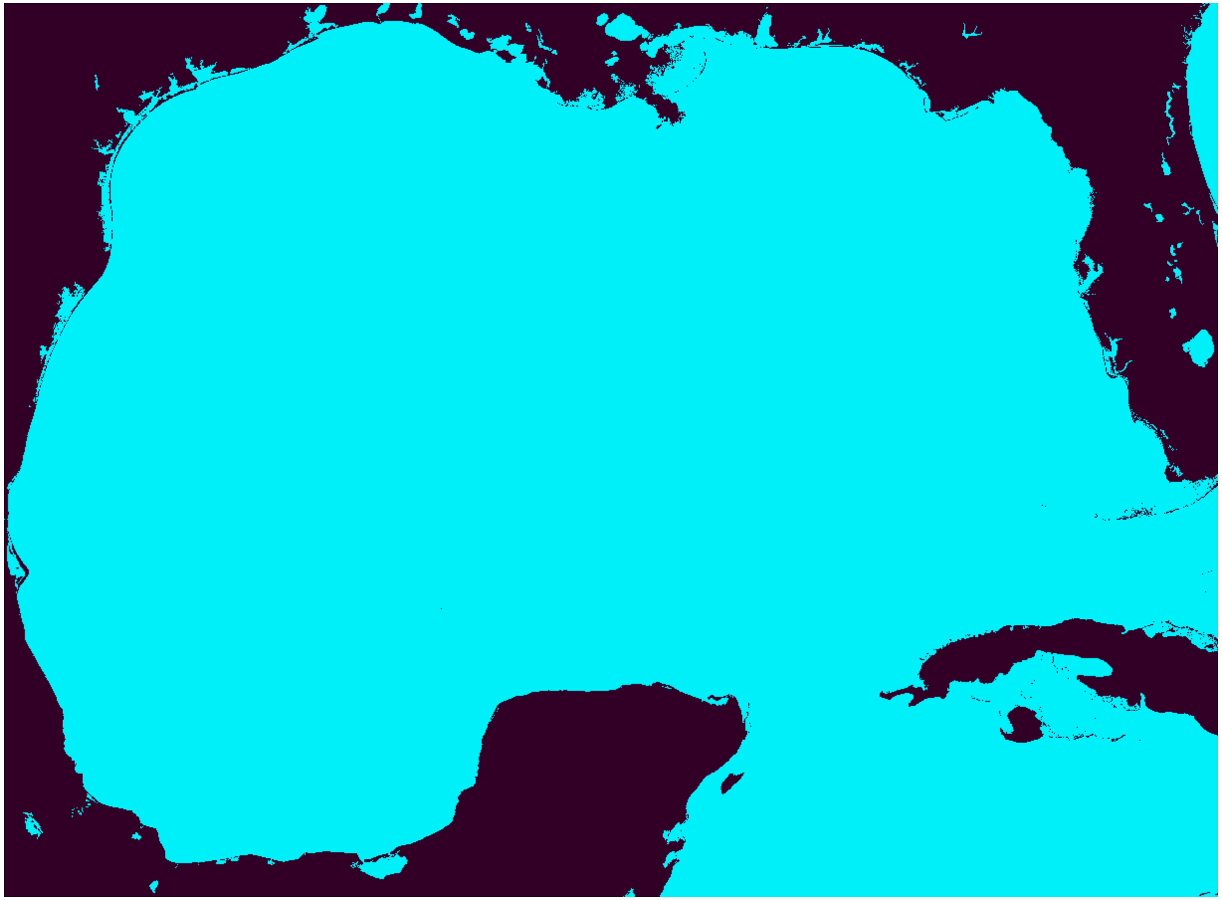
Free-slip versus **No-Slip**





Land/sea Mask

Variables within the masked region are set to zero by multiplying by the mask for either the u, v or rho points :



Land/sea Mask

See the code routines:

```
#ifdef MASKING
# define SWITCH *
#else
# define SWITCH !
#endif

!###########################
        do k=1,N
          do i=IstrU,Iend
            u(i,j,k,nnew)=(DC(i,k)-DC(i,0)) SWITCH umask(i,j)
```